TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Flugsystemdynamik

System Identification for Aerial Applications Using Optimal Control Methods

Christoph Göttlicher

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Oskar J. Haidn
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Florian Holzapfel
	2. Prof. Carlo L. Bottasso, Ph.D.

Die Dissertation wurde am 20.02.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 05.07.2019 angenommen.

Abstract

This thesis investigates the use of advanced optimal control methods in aircraft system identification with a focus on small, unmanned aerial vehicles.

The well established methods for aircraft dynamics characterization based on flight test data have been collected in textbooks and discussed on numerous occasions. However, the advances in modern optimal control theory have not yet been fully incorporated into this process, although the two fields share many common properties. This thesis shows the advantages of utilizing optimal control aspects when determining aircraft model parameters from flight test data.

The parameter estimation problem is first formulated in an optimal control context, both for deterministic and stochastic system descriptions. Then challenges are discussed, which arise in real life applications. Next to the well known solutions to standard issues, novel strategies are collected that improve several aspects of existing procedures. Amongst the former are suitable cost function formulations, their analytic derivatives and residual covariance estimation. The latter, novel ideas include an interpolation approach that is adapted to the dynamic system at hand. This decreases problem size and thus computation time. Further, a scaling strategy is presented that allows for straight forward output weighting and eventually enables the direct use of signals with different sampling frequencies. Besides, parallels are drawn between desirable properties of the cost function Hessian and favorable properties of the resulting parameter estimates in terms of its covariance and correlation.

Overall the application of optimal control methods to aircraft parameter estimation problems, especially transcription using full discretization, has definitive advantages. However, it also raises some issues, the most important of which is the a posteriori computation of parameter covariance estimates. The standard methods cannot be applied to fully discretized formulations, which is why a novel uncertainty quantification framework for constrained problems is developed.

All ideas that are presented in this work are implemented in an add-on to the optimal control toolbox FALCON.m, which is used to treat six representative examples. Those comprise of two cases covering unstable aircraft, and two cases where a stochastic treatment of the system is necessary. The most complex example case illustrates parameter estimation for a nonlinear, stochastic, six degree of freedom model, where the corresponding flight data was collected using the low-cost testbed SKYMULE.

Zusammenfassung

Die vorliegende Arbeit untersucht die Verwendung von Methoden der Optimalsteuerung zur Flugzeugsystemidentifikation. Der Fokus liegt hierbei auf kleinen, unbemannten Flugsystemen.

Die gängigen Methoden zur Charakterisierung von flugdynamischen Eigenschaften basierend auf Flugtestdaten wurden bereits in einigen Fachbüchern zusammengestellt, sowie bei einer Vielzahl von weiteren Gelegenheiten diskutiert. Jedoch haben die aktuellsten Erkenntnisse im Bereich der Optimalsteuerungstheorie noch keinen Eingang in diesen Prozess gefunden, obwohl die beiden Anwendungsfelder viele gemeinsame Merkmale aufweisen. Diese Dissertation zeigt die Vorteile auf, welche sich ergeben, wenn Aspekte der optimalen Steuerung verwendet werden, um Flugzeugmodellparameter basierend auf Flugtestdaten zu bestimmen.

Zunächst wird das Parameterschätzproblem im Kontext der Optimalsteuerung formuliert, was sowohl für deterministische wie auch stochastische Systembeschreibungen erfolgt. Dann werden Herausforderungen diskutiert, welche sich bei der praktischen Anwendung stellen. Neben den gängigen Lösungen für standardmäßig auftretende Fragen, werden neuartige Strategien zusammengetragen, welche verschiedene Aspekte etablierter Prozeduren verbessern. Erstere umfassen geeignete Formulierungen der Kostenfunktion, ihre analytischen Ableitungen, sowie die Schätzung der Residuenkovarianzmatrix. Letztere, neue Ideen beinhalten einen Interpolationsansatz, welcher speziell an die Dynamik des betrachteten Systems angepasst ist. Dieser reduziert die Problemgröße und spart damit Berechnungszeit. Weiterhin wird eine Skalierungsstrategie präsentiert, die es erlaubt die Systemausgänge gegeneinander zu gewichten. Diese erlaubt schließlich die direkte Verwendung von Signalen mit unterschiedlicher Aufzeichnungsrate. Außerdem werden Parallelen aufgezeigt, zwischen wünschenswerten Eigenschaften der Hessematrix der Kostenfunktion und vorteilhaften Eigenschaften der resultierenden, geschätzten Parameterkovarianzmatrix und -korrelationsmatrix.

Insgesamt hat die Anwendung von Optimalsteuerungsmethoden zur Flugzeugsystemidentifikation, speziell die Verwendung von vollständiger Diskretisierung, definitive Vorteile. Allerdings werden auch einige Fragen aufgeworfen. Die Wichtigste ist die nach der a posteriori Berechnung von Parameterkovarianzschätzungen. Die Standardvorgehensweisen sind nicht für die Verwendung bei vollständig diskretisierten

Formulierungen geeignet, weswegen eine neuartige Herangehensweise zur Unsicherheitsquantifizierung in beschränkten Problemen entwickelt wurde.

Alle Ideen, welche im Rahmen dieser Dissertation präsentiert werden, sind in einem Add-On zum Optimalsteuerungswerkzeug FALCON.m implementiert, welches benutzt wurde, um sechs repräsentative Beispiele zu behandeln. Diese beinhalten zwei Fälle, welche sich mit instabilen Flugzeugkonfigurationen beschäftigen und zwei weitere, in denen eine stochastische Systembeschreibung notwendig ist. Das komplexeste Beispielproblem illustriert die Parameterschätzung für ein nichtlineares, stochastisches, Modell mit sechs Freiheitsgraden, wobei die Datenerhebung hierfür mittels der low-cost Plattform SKYMULE erfolgte.

Danksagung

Diese Arbeit entstand größtenteils während meiner Zeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Flugsystemdynamik. Hier gilt mein Dank Prof. Florian Holzapfel: für die Chance meine Dissertation an seinem Lehrstuhl anfertigen zu können, sowie für seine Unterstützung vor, während und nach meiner Zeit am FSD. Weiterhin möchte ich Prof. Carlo L. Bottasso und Prof. Oskar J. Haidn danken, dass sie bereit waren das Zweitgutachten anzufertigen respektive den Prüfungsvorsitz zu übernehmen.

Diese Arbeit wäre nicht möglich gewesen, wenn ich mich nicht regelmäßig mit Stefan Hager, Christian Merkl, Barzin Hosseini, Javensius Sembiring und Xiang Fang über die Feinheiten der Systemidentifikation in allen ihren Facetten hätte austauschen können. Mindestens genauso unabdingbar waren die Unterstützung und die Vorarbeiten der Flugbahnoptimierungsgruppe um Benedikt Grüter, Johannes Diepolder, Patrick Piprek, Tuğba Akman, Matthias Bittner, und Matthias Rieck. Deren Werkzeug FALCON.m bot die perfekte Grundlage für meine eigenen Implementierungen.

Generell gilt mein Dank allen Kollegen am Lehrstuhl. Die offene und freundliche Atmosphäre, sowie die stete Hilfsbereitschaft haben diese Zeit zu einer ganz besonderen werden lassen. Die Erfahrungen, die ich dort machen durfte, hinterließen definitiv einen bleibenden Eindruck. Hervorheben möchte ich Frau Monica Kleinoth-Gross für die Unterstützung in allen organisatorischen Angelegenheiten. Danke, dass ich auch mit dummen Fragen vorbeikommen durfte. Weiterhin möchte ich mich bei Chong Wang, Christoph Krause, Markus Hochstrasser, Martin E. Kügler, Christopher Blum und Andreas Kleser bedanken. Die fachlichen und die weniger fachlichen Diskussionen mit euch waren stets eine Bereicherung.

Zuletzt kann ich mich nicht genug bei meinen Eltern Silvia Göttlicher, Dieter Göttlicher und meinem Großvater Alfred Ganz bedanken. Ohne deren Rückhalt, Hilfe und Unterstützung hätte ich niemals diesen Weg einschlagen, geschweige denn zu Ende gehen können. Abschließend gilt mein Dank meiner wunderbaren Partnerin Susanne Vogt. Danke, dass du immer für mich da warst, mir in den letzten Monaten den Rücken freigehalten und mich ertragen hast, wenn mir mal wieder alles zu viel wurde.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Examples

Abbreviations

Acronyms

FALCON.m	FSD optimAL CONtrol tool for MATLAB
A/C	aircraft
AIAA	American Institute of Aeronautics and Astronautics
BLDC	Brushless DC
BLUE	best, linear, unbiased estimator
BVLOS	beyond visual line of sight
c.g.	center of gravity
CFD	Computational Fluid Dynamics
DAE	Differential Algebraic Equation
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOF	degree of freedom
ECEF	Earth-Centered-Earth-Fixed
EKF	Extended Kalman Filter
ERTS	Extended Rauch-Tung-Striebel
FCC	Flight Control System
FCS	Flight Control System
FEM	Filter Error Method
FPR	Flight Path Reconstruction
FSD	Institute of Flight System Dynamics
GNSS	Global Navigation Satellite System
i.i.d.	independent and identically distributed
IMU	Inertial Measurement Unit
IPOPT	Interior Point Optimizer
ISA	International Standard Atmosphere
JMLE	joint maximum likelihood estimation
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LQR	Linear Quadratic Regulator
MAP	maximum a posteriori probability

MEMS	microelectromechanical systems
MMLE	marginal maximum likelihood estimate
MSE	mean square error
NED	North-East-Down
NLP	Non-Linear Programming
ODE	ordinary differential equation
OEM	Output Error Method
pdf	probability density function
RPAS	Remotely Piloted Aerial System
RTS	Rauch-Tung-Striebel
SNOPT	Sparse Nonlinear Optimizer
SQP	Sequential Quadratic Programming
SVD	Singular Value Decomposition
TUM	Technische Universität München
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VTOL	Vertical Take-Off and Landing
WGS84	World Geodetic System 1984
WLS	Weighted Least-Squares
WMM2015	World Magnetic Model 2015
WORHP	We Optimize Really Huge Problems

Symbols

Dynamic System Quantities

$\mathbf{A}$	continuous time system matrix
$\mathbf{B}$	continuous time input matrix
$\mathbf{C}$	output matrix for linear systems
${}_{\theta}\mathbf{C}$	jacobian of output equation w.r.t. parameter vector
$\mathbf{D}$	feedthrough matrix for linear systems
${}^{c}\mathbf{F}$	continuous time process noise distribution matrix
$\mathbf{F}$	process noise distribution matrix
${}^{d}\mathbf{f}_{k}[\,]$	discrete, non-linear system equation
$\boldsymbol{f}(\,)$	non-linear system equation
$\mathbf{G}$	measurement noise distribution matrix
$\boldsymbol{g}$	non-linear output equation
$\boldsymbol{h}$	non-linear static system
${}^{\square_1}_{\square_2}\mathbf{S}$	sensitivity of $\square_1$ w.r.t. $\square_2$, e.g. state sensitivity w.r.t model parameters ${}^{x}_{\theta}\mathbf{S} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}}$
$\mathbf{U}$	concatenated inputs $\mathbf{U} = [\boldsymbol{u}_1, \dots \boldsymbol{u}_N]$
$\boldsymbol{u}$	input vector
u	scalar input
$\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$	linearized regressor matrix in weighted least-squares problems
$\mathbf{X}$	concatenated states $\mathbf{X} = [\boldsymbol{x}_0, \dots \boldsymbol{x}_{N-1}]$
$\boldsymbol{x}$	state vector
x	state
$\boldsymbol{y}$	output vector
y	output
$\mathbf{\Gamma}$	input matrix for discrete time, linear systems
$\mathbf{\Phi}_{\boldsymbol{f}}(t_0, \dots, t_N \vert \boldsymbol{x}_0)$	numerical integration of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})$ on the time grid $t_0, \dots, t_N$, using the initial condition $\boldsymbol{x}_0$
${}_{\theta}\mathbf{\Phi}$	transition matrix of parameter states
$\mathbf{\Phi}$	state transition matrix

Estimation Quantities

$\bar{\square}_N$	sample average of $\square$, using N samples
$\hat{\boldsymbol{\mathcal{B}}}$	estimated residual covariance matrix
$\boldsymbol{\mathcal{B}}$	residual covariance matrix
$\mathrm{Cov}[\square_1 \vert \square_2]$	conditional covariance of $\square_1$, given $\square_2$

$\text{Corr}[\square_1]$	correlation Matrix of $\square$, i.e. correlations $\rho[\square_i, \square_j]$ arranged in a matrix
$\rho[\square_1, \square_2]$	correlation between $\square_1$ and $\square_2$
$\text{Cov}[\square]$	Covariance of $\square$
$\mathbf{D}$	output scaling matrix for maximum likelihood problems
$\text{E}[\square_1\|\square_2]$	conditional expected value of $\square_1$, given $\square_2$
$\text{E}[\square]$	Expected value of $\square$
$\hat{\square}$	estimated quantity
$\boldsymbol{\mathcal{F}}$	Fisher Information Matrix
f_N	Nyquist Frequency
f_s	Sampling Frequency
J_{FEM}	maximum likelihood cost function for filter error formulation
J_{LS}	least-squares cost function
J_{MAP}	maximum a posteriori probability (MAP) cost function
J_{ML}	maximum likelihood cost function
J_{NLS}	non-linear least-squares cost function
J_{OEM}	maximum likelihood cost function for output error formulation
J_{WLS}	weighted least-squares cost function
$\mathbf{K}$	Kalman gain
$\mathbb{L}$	Likelihood function
$\mathcal{M}$	Model structure
$\mathbf{M}$	RTS Smoother gain
${}^{m}\square$	measured quantity
$\bar{N}$	index of last sample, usually $\bar{N} = N - 1$
N	Sample Size
$Pr\{\omega\|\Lambda\}$	probability of the event ω, given the event Λ has been observed
$\mathbf{P}_0^{\mathrm{x}}$	initial state error covariance
$Pr\{\omega\}$	probability of the event ω
$\mathbf{P}_{k\|k}^{\tilde{\mathbf{x}}}$	corrected state error covariance estimate, incorporating all available information at time step k
$\mathbf{P}_{k+1\|k}^{\tilde{\mathbf{x}}}$	propagated state error covariance estimate, incorporating all available information until time step $k-1$
$\mathbf{P}_{k\|N}^{\tilde{\mathbf{x}}}$	smoothed state error covariance estimate, incorporating all available information up until the final time N
$p_{\mathrm{x}\|\mathrm{y}}(x\|y)$	conditional probability density function of x, given y, evaluated at the realization x, given y
$p_{\mathrm{x}}(x)$	probability density function of the random variable x, evaluated at the realization x
${}^{c}\mathbf{Q}$	continuous time process noise covariance matrix

$\tilde{\mathbf{Q}}$	extended process noise covariance matrix $\tilde{\mathbf{Q}} = \mathbf{F}\mathbf{Q}\mathbf{F}^\intercal$, including distribution matrix
$\mathbf{Q}^{\boldsymbol{\theta}}$	process noise covariance matrix of parameter estimate
$\mathbf{Q}$	process noise covariance matrix
$\tilde{\mathbf{R}}$	extended measurement noise covariance matrix $\tilde{\mathbf{R}} = \mathbf{G}\mathbf{R}\mathbf{G}^\intercal$, including distribution matrix
$\mathbf{R}$	measurement noise covariance matrix
$\boldsymbol{r}$	residuals
r	scalar residual
$\sigma_{\text{rel}}[\square]$	relative standard deviation of $\square$, i.e. relation between its standard deviation and numerical value $\sigma[\square]\,/\square$
$\boldsymbol{s}$	Score
$\sigma[\square]$	Standard deviation of $\square$
$\text{Var}[\square]$	Variance of $\square$
v	measurement noise
$\boldsymbol{v}$	measurement noise vector
$\mathbf{W}$	covariance scaling matrix for maximum likelihood problems
${}^c\boldsymbol{w}$	continuous time process noise vector
w	process noise
$\boldsymbol{w}^{\boldsymbol{\theta}}$	process noise vector driving parameter state
$\boldsymbol{w}$	process noise vector
$\mathbf{X}$	regressor matrix in linear least-squares problems
$\bar{\boldsymbol{x}}_0$	initial state estimate
$\hat{\boldsymbol{x}}$	estimated state vector
$\hat{\boldsymbol{y}}_{k\|l}$	output vector estimate at time k, incorporating all information until time l
$\tilde{\boldsymbol{x}}_{k\|l}$	estimation error at time k, incorporating all information until time l
$\hat{\boldsymbol{x}}_{x\|l}$	state vector estimate at time k, incorporating all information until time l
$\hat{\boldsymbol{y}}$	estimated output vector
$\mathbf{Z}$	concatenated measurements $\mathbf{Z} = [\boldsymbol{z}_1, \dots \boldsymbol{z}_N]$
$\boldsymbol{z}$	measurement vector
z	measurement
$\boldsymbol{\Sigma}$	covariance of Gaussian prior parameter distribution
$\hat{\boldsymbol{\theta}}_{LS}$	least-squares estimate of the parameter vector $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{MAP}$	maximum a posteriori probability estimate of the parameter vector $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{ML}$	Maximum Likelihood estimate of the parameter vector $\boldsymbol{\theta}$
$\hat{\boldsymbol{\theta}}_{WLS}$	weighted least-squares estimate of the parameter vector $\boldsymbol{\theta}$

$\hat{\boldsymbol{\theta}}$	estimated parameter vector
$\boldsymbol{\theta}_{prior}$	mean of Gaussian prior parameter distribution
θ	scalar parameter
$\boldsymbol{\theta}$	parameter vector
$\boldsymbol{\xi}$	regressor in linear least-squares problems

Optimization Quantities

$\mathcal{A}$	Active Set
$\square_{lb}$	lower bound on $\square$
$\square_{ub}$	upper bound on $\square$
$\tilde{\boldsymbol{c}}$	active constraint
c_{eq}	scalar equality constraint
$\boldsymbol{c}_{eq}$	equality constraint
c_{ineq}	scalar inequality constraint
$\boldsymbol{c}_{ineq}$	inequality constraint
$\boldsymbol{d}$	search direction in optimization problem
$\mathcal{H}$	Hamiltonian
J	cost function value
L	Lagrange term in optimal control cost function
$\mathcal{L}$	Lagrangian
$\square^*$	optimum value of $\square$
$\mathcal{Z}$	feasible region for an optimization problem
$\mathbf{z}$	optimization vector for numerical optimization problems
$\boldsymbol{\lambda}$	vector of Lagrange multipliers corresponding to equality constraints
λ	Lagrange multiplier corresponding to equality constraints
$\boldsymbol{\mu}$	vector of Lagrange multipliers corresponding to inequality constraints
μ	Lagrange multiplier corresponding to inequality constraint
ϕ	Mayer term in optimal control cost function
$\boldsymbol{\zeta}$	integration defect

Misc

$\square^a$	augmented quantity
b	span
C_i	aerodynamic coefficient i
$\bar{c}$	mean aerodynamic chord
δ_{ij}	Kronecker delta with $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise
$[\square]_{(l,m)}$	element of matrix $\square$ in l-th row and m-th column

$\boldsymbol{e}_j$	unit vector in j-th dimension
$[\square]_{(l)}$	l-th element of vector $\square$
h	altitude in World Geodetic System 1984 (WGS84) coordinates
$\mathbf{I}_{\square}$	identity matrix with dimension $\square$
$\left(\mathbf{I}^R\right)_B$	tensor of inertia at reference point R in B system
$n_{\square}$	Dimension of vector $\square$
p_{stat}	static pressure
$\bar{q}$	dynamic pressure
$\left[\vec{\square}\times\right]$	operator to transform cross product into skew symmetric matrix
S_{ref}	reference area
$\mathbf{T}$	matrix of eigenvectors
T	static temperature
$\boldsymbol{t}$	eigenvector
$\mathbf{Z}$	base for the null-space
${}_{\mathrm{S}}\mathbf{Z}$	base for the null-space built from solving the sensitivity equations
α	angle of attack
β	angle of sideslip
$\mathbf{\Theta}$	vector of euler angles
$\mathbf{\Lambda}$	matrix of eigenvalues
λ	eigenvalue
λ	longitude in WGS84 coordinates
μ	latitude in WGS84 coordinates
ϕ	bank angle
ψ	heading angle
ρ	air density
θ	pitch angle

So far as Mathematics do not tend to make men more sober and rational thinkers, wiser and better men, they are only to be considered as an amusement, which ought not to take us off from serious business.

Thomas Bayes, 1736 [Bay1736, p. 50]

1

Introduction

1.1 Motivation and Background

The main motivation for this work will be illustrated first. This is done by highlighting three aspects that are to play a fundamental role throughout this thesis. Namely these three aspects are modeling, system identification and optimal control.

1.1.1 Why Modeling?

In modern aircraft projects, mathematical models that may be analyzed and simulated become ever more important. Their application ranges from simple simulation tasks in order to get a basic feeling for the aircraft, to training simulations, mission planning, complex control design and optimization tasks.

Originally, simple, linear mathematical descriptions were mainly used to analyze the aircraft's flight mechanical characteristics in terms of natural frequencies and damping coefficients of the pertinent eigenmotions [MK2016, Ch. 1.1]. With this knowledge, simple automatic control systems could be designed and tuned on ground, before going the last, costly step towards flight testing. Nowadays, complex, high fidelity, nonlinear, six degree of freedom models play a central role, allowing for more complex control strategies and the possibility to reduce flight-test time. Much can already be tested in simulations, thus eventually reducing development costs [SL2003, Ch. 2].

This extensive use of simulation models has as of yet not fully arrived in the currently growing Unmanned Aerial Vehicle (UAV) market. Here, much work has been very "hands-on" so far: simple control strategies have been implemented and tuned manually, in-flight. However, the importance of simulation models is being recog-

nized, and their availability promises to make development efforts in this increasingly important aviation sector more structured.

Additionally, in both worlds the large, commercial aircraft market, and the UAV sector, many systems show a growing degree of automation. This is not only limited to the classical task of changing the flight mechanical characteristics via the inner feedback loops, but also encompasses high level flight management functionalities. Especially for these latter applications, the myriad of different combinations of scenarios, aircraft states, environmental conditions, applicable procedures and actual piloting tasks makes it impossible to test every possibility in flight. Thus, the only viable way is to automate testing based on simulated aircraft behavior.

Another important application of aircraft simulation models is operator training. It is not important if this "operator" is a commercial pilot flying the aircraft, or the ground control crew steering a UAV. In both cases the respective personnel needs to be well trained in order to safely and effectively perform their assigned tasks. This can be achieved using high-fidelity simulation models, together with real Flight Control System (FCS) hard- or software, and original input devices, combined in a realistic flight training environment [SL2003, Ch. 2].

Closely related to operator training are aspects of mission planning. Any mission that may achieve its goal on paper but is infeasible due to real-life limitations is bound to fail before it even begins. By using realistic simulation models, the mission can be simulated before its start in order to check its feasibility in terms of flight mechanics (necessary turn radii, minimum/maximum climb angles, acceleration profiles ...), flight performance (maximum flight time, necessary energy supply), and external disturbances (wind, thermals, turbulences). If any one of these aspects is not to the full satisfaction of the operator, he may freely adjust the respective mission parameters in order to keep well within the aircraft's limitations.

If this adjustment is not done manually, but left to an optimization algorithm, the threshold towards the field of aircraft trajectory optimization is crossed. Using algorithms from optimal control, complex high-level tasks may be solved. Classically, these comprise time or fuel minimization problems, but may also encompass the computation of noise-minimal approach trajectories, or optimal airport arrival sequence determination; just to name very few examples from this huge field.

Depending on the application at hand, different levels of fidelity are used: a rather basic 3-degree of freedom (DOF) model may suffice for trajectory and mission planning, whereas the development of an advanced FCS necessitates a very accurate flight dynamics model. In order to be cost efficient, one often strives to use that model formulation, which is as simple as possible, but still incorporates the characteristics, which are most relevant for the task to be performed. This is because the development effort and cost to formulate a model and determine its unknown parameters increase with growing fidelity.

In all of the listed applications, and for all the possible levels of model fidelity, the results are futile, if the underlying simulation model does not represent reality sufficiently well. It is thus imperative to have tools at one's disposal, which may not only be used to determine these models, but also to verify their quality by comparing them to real system responses.

1.1.2 Why System Identification?

In the above scenarios, it is of utmost importance that the mathematical aircraft description represents the real system as closely as possible. This is true both for rather simple models, with a limited number of parameters and for very complex, high fidelity models with many unknowns.

Rough approximations of the aircraft's behavior are relatively easy to come by. Many handbook methods are available for the preliminary determination of flight mechanical characteristics, based on geometry, airfoil information and the desired configuration [Ros2008, Ros2006, Fin1978]. Some of them are even readily available as computer programs [WV1979]. On the downside, these preliminary approximation methods are seldom accurate enough to fulfill the fidelity requirements of the applications listed in the previous section.

More accurate results may be obtained through Computational Fluid Dynamics (CFD) or wind-tunnel testing, which come at considerably higher costs though, and also do not deliver perfect results without uncertainty. A lot of experience is necessary to meaningfully apply CFD methods, and they only yield the most reliable results, if checked against real data. As for wind-tunnel testing: although it provides an environment where many of the influencing variables may be designed independently just as the analyst desires, several aircraft motions can only be replicated with great difficulties and some not at all. Any dynamic effects, due to e.g. rotational rates are complex to excite and to evaluate afterwards. Furthermore, the two approaches never consider the real system, rather an abstract mathematical representation (CFD) or a scaled model (wind tunnel testing), which introduces further uncertainty when translating them to a real-life setting.

A way to remedy some of those drawbacks is aerodynamic parameter estimation based on flight test data. Admittedly, there are some disadvantages to this approach as well.

First and foremost, the real system has to be available and ready for flight-testing. Thus some basic FCS, together with suitable recording devices need to be installed, which may be a show-stopper in the early stages of a development project.

What is more, the most accurate results for aerodynamic parameters are achieved, if the flow conditions around the aircraft are well known. If the analyst wants to avoid excessive post-processing to obtain these, special sensor equipment, such as flow vanes

or multiple hole probes, is required. This may not be available, especially in UAV development projects, putting further limits on the applicability of parameter estimation.

Further, the influencing factors for aerodynamic parameter estimation usually cannot be excited independently, but have to adhere to flight mechanics: a change in angle of attack will necessarily lead to a change in lift force and thus a change in climb angle. In contrast, during wind-tunnel experiments, some of these kinematic links may be broken up and the influencing factors analyzed independently.

Also, designing a suitable flight test program for parameter estimation, which reliably excites all aspects of the system under investigation, necessitates at least as much experience as successfully applying CFD or designing meaningful wind-tunnel experiments.

Despite these disadvantages, the estimation of aerodynamic parameters from flight test data is well-established in practice, and routinely applied in new development programs, since it also comes with major advantages. One of these is closely related to one of the aforementioned drawbacks: on the one hand, an operational version of the real systems needs to be available; but on the other hand, experiments can be conducted using the *real system* rather than some mathematical or scaled model. Thus, results will be based on the real reaction of the aircraft under investigation instead of theoretic predictions, or statements based on laboratory experiments.

Also, the final flight mechanical model, with all relevant subsystems, may directly be used in parameter estimation. No transfer of coefficients, coordinate systems, or units is necessary, which helps to reduce errors when converting test results into a simulation model.

In addition, although the aircraft can only be excited while adhering to flight mechanics, it will not suddenly stop to obey the basic laws of physics during its operation. Thus, even though only a subspace of independently influencing variables may be excited, one can be sure to cover the right subspace, since the data is collected during real flight tests. And no time is spent on analyzing combinations of the independent variables that do not appear in normal operation.

The last, and in many cases probably most pronounced, advantage is the cost and necessary infrastructure involved when applying aerodynamic parameter estimation rather than wind-tunnel testing or CFD analysis: If an implementation such as the one developed during the work on this thesis is available, no additional software licenses need to be procured (as would be the case for most CFD programs); and no additional infrastructure such as a wind-tunnel needs to be available. Instead, the necessary resources are: some time for the analysis; a consumer PC with a license for MATLAB (which is one of the most widely used tools for control design in aerospace applications nowadays, and is thus probably available); and flight data.

The last aspect, namely the collection of flight data, may be an issue, since carefully planned tests for the explicit purpose of parameter estimation are necessary. How-

ever, flight testing is an integral part of any development project, which is why some additional flight test time can usually be allocated to parameter estimation purposes.

Cost and the availability of specialized infrastructure may play only a secondary role for huge commercial or military aircraft development projects, which are usually funded by the government. In contrast, these aspects are critical for commercial Unmanned Aerial System (UAS) projects: Development times are shorter, available budget is smaller, and commonly no infrastructure or costly CFD licenses can be procured, leaving aerodynamic parameter estimation as the only practicable way of obtaining realistic simulation models.

Accordingly, the specific application aspects discussed here, will mostly be with UAV cases in mind. However, all of the methods presented throughout this work are applicable to parameter estimation for any size of aircraft, and most even to general dynamic systems.

1.1.3 Why Optimal Control?

As mentioned before, aerodynamic parameter estimation from flight test data is a well-established process that is routinely applied in aircraft development projects. However, most of the approaches go back to algorithms that have originally been developed in the 1970's and 1980's. The most important update during the last 30 years may have been the use of growing computational power to consider not only linear, but also non-linear model formulations. But the underlying solution algorithms are still the same. Since good results have been obtained during the last three decades, this should not be regarded as "bad" per se.

On the other hand, big advancements in the field of optimal control have been made since the 1980's. More and more computational power, available on standard consumer PCs, have enabled the solution of non-linear programming problems of increasing size. Now it is possible to tackle real-life optimal control tasks that result in huge optimization problems. However, they are comparatively straight forward to solve, if modern methods for their formulation are used. These methods also improve the robustness of the solution, together with its convergence radius, while the resulting, sizable problem stays manageable through the extensive exploitation of its sparsity characteristics [Bet2010, BA2010].

It thus seems natural to combine the two fields, and to try to benefit from the advancements in optimal control, also in the field of parameter estimation.

Together with UAV applications, this constitutes the main motivation for this thesis. The specific solution strategies and application aspects that have been adapted will be detailed in the following sections.

1.2 State of the Art

The following, first subsection refers to the main historical contributors to modern day estimation theory. After that, a brief overview of optimization and optimal control is given, and the section is concluded by some remarks on the state of the art of parameter and state estimation and its aircraft applications.

The detailed mathematical background will be presented in the subsequent chapter I "Preliminaries".

1.2.1 Historical Perspective of Estimation Theory

No work on parameter estimation from observation data can be complete without reference to CARL FRIEDRICH GAUSS. His invention of the method of least-squares in his work "Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium" ("Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections" [Gau1857]) laid the foundation for many estimation approaches to follow in the subsequent centuries. Not only did GAUSS use this method to solve linear least-squares problems: the equations he dealt with in the observation of heavenly bodies were inherently non-linear. In order to still obtain viable solutions, he developed an iterative procedure, based on first order approximations [Gau1857], which allowed for the solution of non-linear systems of equations; This approach is still at the heart of many modern solution algorithms.

Next to the method of least-squares, probably the most important contribution of GAUSS to modern estimation theory was the statement of the Gaussian or normal distribution. He based this distribution on the observation of measurement errors, for which he postulated three simple assumptions

1. Small measurement errors are more likely than large measurement errors
2. The likelihood for the positive and negative value of an error has to be equal (i.e. the distribution has to be symmetric)
3. If several observations are made, the most likely value of the quantity of interest is to be the arithmetic mean of the observations

Subsequently GAUSS used fundamental probabilistic arguments along with some basic calculus to arrive at the well-known definition of the normal distribution for some zero-mean measurement error Δ, with a "measure of precision"[1] h [Gau1857, Sec. 177]

$$\phi(\Delta) = \frac{h}{\sqrt{\pi}} \exp\Big(-h^2\Delta^2\Big)$$

The main importance of this distribution lies in the Central Limit Theorem (see Appendix A.4.9.2). It states that, independent of the type of distribution of a random

[1] GAUSS' measure of precision is related to the modern variance via $\sigma^2 = \frac{1}{2h^2}$

variable, its sample mean (for sufficiently large sample sizes) will be distributed according to a Gaussian or normal distribution. Thus, even if one cannot explicitly state the nature of the distribution of some random variable, as soon as an averaging process is involved, one can always argue that the result will be normally distributed. Some parallels to the engineer's fondness of linearization in order to simplify problems can be drawn: if no precise knowledge about a random component can be obtained, the analyst may always use the normal hypothesis and afterwards check if it was justified. GAUSS' ideas are still at the core of many modern approaches, as will be illustrated throughout this work, underlining the importance of his discoveries.

During the 18th and 19th century, other mathematicians published ideas, which are still in use today: DANIEL BERNOULLI applied the idea of maximizing a probability density in order to obtain the "most probable" value of the underlying, desired quantity [Ber1777, Ken1961]. Due to some unfortunate assumptions, he encountered technical difficulties, which eventually led to unsatisfactory results. Only in the beginning of the 20th century, RONALD AYLMER FISHER re-invented the *method of maximum likelihood*, and put it on a sound mathematical basis, which made it useful for real-life applications. In a series of papers [Fis1912, Fis1922, Fis1925], he also introduced other fundamental ideas of modern estimation theory, such as "efficient" and "consistent" estimates, and investigated his method on the grounds of this theory.

Another early pioneer in probability theory was THOMAS BAYES, even though he did not live to claim his fame. In his essay [Bay1763], which was posthumously published by his friend RICHARD PRICE, he stated a theorem about conditional probabilities that nowadays bears his name

$$Pr\{A|B\} = \frac{Pr\{B|A\}\, Pr\{A\}}{Pr\{B\}}$$

This seemingly simple formula can be found in many modern publications on statistical inference and lies at the core of a whole field entitled "Bayesian Statistics". It allows for a straightforward inclusion of prior knowledge about an event into the analysis and is thus not only central to iterative estimation procedures, such as the Kalman filter. It is also an important tool in parameter estimation, if prior knowledge about the parameters is to be used. Therefore, it is inevitable that this theorem will be employed at numerous occasions throughout this work.

The last major theoretic breakthrough that needs to be mentioned for the complete historic background of this work are the findings due to RUDOLF EMIL KÁLMÁN [Kal1960, KB1961]. He considered parameters that were allowed to vary in time according to a prescribed model, which incorporates a well-defined random component on one side. On the other side, he considered randomly disturbed measurements of linear combinations of those parameters, which can be used to correct propagated values. In this context, the border between parameter and state estimation is blurred, and appears as merely two sides of the same problem. KALMAN's approach to optimally

combine propagated values with measurements provided a solution methodology that was much-needed in the age of the space race. Although many other researchers at that time looked into state estimation, KALMAN managed to link his name to the topic as no other, by publishing his findings at the right time, and for a wide audience. This is mainly due to its ease of implementation, and the readily available approximate solutions even for non-linear model formulations [Sim2006, App. A]. The most prominent application of the latter is probably in the field of sensor data fusion, where the application due to SMITH, SCHMIDT, and MCGEE [SSM1962] was amongst the first, and notably contributed to the fame of the Kalman Filter.

The choice of milestones in estimation theory and inferential statistic in the foregoing section is mainly motivated by the applications to follow in this work. For a more comprehensive treatment of the historical aspects, see e.g. [Hal2008, Lai1974, Sor1970, Sor1980, Sim2006].

1.2.2 Optimization and Optimal Control

The problem of achieving a goal while minimizing the cost associated with this achievement is probably as old as the mathematical tools that may be used to describe the task and cost at hand. The field of optimization has grown to be very vast, with myriads of different possible problem and cost formulations. This is why a restriction is in order: the focus of this thesis will lie on cost and constraint functions that are continuously differentiable almost everywhere, and feasible sets that may be described by (possibly non-linear) algebraic equality and inequality constraints.

The two results that enable the structured search for solutions of those optimization (and consequently optimal control) problems are the Karush-Kuhn-Tucker (KKT) conditions on the one hand, and the Pontryagin maximum principle on the other.

The first are named after the publications by HAROLD W. KUHN and ALBERT W. TUCKER [KT1951] and WILLIAM KARUSH [Kar1939], which provide first order necessary conditions for optimality (under certain regularity conditions) in the presence of inequality constraints. The KKT conditions extend the notion of Lagrange multipliers to inequality constraints, and can eventually be exploited in the algorithmic search for optima in inequality constraint cases. Only these results enabled the solution of large scale, real-life optimization problems.

The second fundamental result in modern optimal control theory was another brainchild of the cold war: During the 1940's and 1950's, on the west side of the Atlantic Ocean, HESTENES, BELLMANN and ISAACS tried to solve a minimum time interception problem, while LEV SEMYONOVICH PONTRYAGIN and his students investigated a similar problem in the Soviet Union [Bit2017, Ch. 1]. The latter however, managed to link their names to what is nowadays called "Pontryagin's maximum (or minimum) principle", which was first made available for a large, English speaking audience in

[PBGM1962]. Therein, necessary conditions for optimality of a cost, subject to dynamic system constraints are given, providing the mathematical tools to solve optimal control problems.

With the advent of modern computers, both results have been brought together by first suitably transcribing the optimal control task, and subsequently solving the resulting non-linear programming problem based on consequences of the KKT conditions.

The literature on this topic today is abundant. The aspects collected here are largely based on the textbooks [Bet2010, BA2010, Ger2017, Ger2018, Ger2012] as well as the doctoral theses [Rie2017, Bit2017].

1.2.3 Modern Estimation Theory

Similar to optimization and optimal control, the field of system identification and statistical parameter estimation is huge nowadays. The sheer number of disciplines that routinely apply estimation in their respective fields, together with their discipline-specific solutions is immense. This again necessitates the restriction to a narrower field, namely parameter and state estimation in dynamic systems with aerospace applications. Still, an exhaustive overview over the topic is beyond the scope of this work; rather its main sources will be mentioned briefly.

As illustrated already, the field of state estimation began to advance quickly with the works of KALMAN, and its application to linear and non-linear systems. Further notable contributions are due to RAUCH, TUNG, and STRIEBEL, who considered the state estimation problem for linear systems from a maximum likelihood point of view [RTS1965]. They provided one of the most used batch algorithms: the Rauch-Tung-Striebel (RTS) smoother. It is able to not only include past, but also future measurements in order to obtain the best possible state estimate (in a statistical sense).

Many of the results at the time were compiled by JAZWINKSI [Jaz1970], which provides an exhaustive background on linear and non-linear filtering and smoothing. Some ten years later, SORENSON's book highlighted the close relation between state estimation, maximum likelihood and maximum a posteriori estimation on the one hand and least-squares approaches on the other hand [Sor1980]. Also, it provides a summary of the historical development of estimation theory up to its publication date.

The most modern books on state and parameter estimation that have been consulted here, are the ones by SIMON [Sim2006] and CRASSIDIS and JUNKINS [CJ2012]. Their focus lies mainly on state estimation, presenting the state of the art in a modern vector-matrix notation with many optimization based arguments. Statistical aspects are only considered when necessary.

One of the works on general system identification, which has been consulted for this thesis, is the one by WALTER [WP1997], focusing on the identification of *parametric* models. Since most aerospace applications are based on this type, it is especially rele-

vant. It provides a good, albeit application independent, overview over system identification, illustrating advanced concepts that go beyond what is usually discussed in aircraft parameter estimation.

Works that have not directly found their way into this thesis, but should be mentioned for an overview over the state of the art, are the books by LJUNG, which form the basis for MATLAB's own system identification toolbox [Lju2009, Lju2011]. Additionally, works that occur in most bibliographies in aircraft system identification are due to EYKHOFF [Eyk1987], SODERSTRÖM and STOICA [SS1989], and GOODWYN and PAYNE [GP1977].

Even though some arguments will need to be borrowed from frequency domain approaches, the overall field will not be discussed. The reason for this is mainly its inherent limitation to *linear* models. However, although the basic idea of considering the data in another domain seems fundamentally different, it is noteworthy that some of the statistical basics to be illustrated in this work are also routinely applied in frequency domain parameter estimation. Textbooks on the topic are available by SCHOUKENS and PINTELON [SPR2012, PS2012], where actual aircraft and rotorcraft applications are discussed by TISCHLER [TR2012].

1.2.4 Current Developments in Aircraft System Identification

The first attempts at numerically quantifying an aircraft's characteristics date back to the early 1920s, when a basic detection of frequencies and damping ratios was of interest. Already then, GLAUERT [Gla1919] and NORTON [Nor1924a, Nor1924b] engaged in the determination of stability and control derivatives.

During the 1940's and early 1950's, these aspects were investigated using frequency domain approaches. Prominent names during this era were those of GREENBERG [Gre1951], MILLIKEN [Mil1947], and SHINBROT [Shi1951].

Between the 1940's until the early 1960's most techniques were limited to frequency response methods, or suffered from other limitations [Jat2015, Ch. 1]. With the advent of modern computers, it became possible to apply more advanced, oftentimes maximum likelihood based methods, to identify more sophisticated models [CJ2012, Ch. 4.4] [MK2016, Ch. 1]. Especially the works of ILIFF et al. [IP1969], TAYLOR and ILIFF [TI1969], MEHRA [Meh1970, Meh1971], and GERLACH [Ger1971] are to be mentioned here.

During the 1980's, many of the approaches that are still in use today were compiled by MAINE and ILIFF [MIM1985, MI1986, IM1985, Ili1989], and MULDER et al. [MBI+1994] with a very complete overview over the mathematical background. Also, they collected an extensive bibliography over the works in aircraft parameter estimation up to 1986 [IM1986]; another one is available from the year 1996 by HAMEL and JATEGAONKAR [HJ1996]. A list of parameter estimation activities at NASA Dryden

Flight Research Center was compiled by WANG and ILIFF in [WI2004], the corresponding summary of activities at the Deutsches Zentrum für Luft- und Raumfahrt (DLR) was compiled by JATEGAONKAR [Jat2004].

Nowadays, the two main sources for theoretical background and applications advice for time-domain aircraft system identification are the works by MORELLI and KLEIN [MK2016] as well as the book by JATEGAONKAR [Jat2006], respectively the second edition [Jat2015]. Those two compile the current state of the art, with numerous application examples, accompanying software, and extensive bibliographies for further reading.

Publications on the application of system identification methods in aircraft development projects appear on a regular basis in respective conference proceedings and journals. As with most topics in the aerospace field, the American Institute of Aeronautics and Astronautics (AIAA) and its web presence are one possible starting point for research. They even dedicated two special sections in the Journal of Aircraft to flight vehicle system identification [Jat2004, Jat2005], which illustrate many applications. Further, the working group around MORELLI can look back on long years of practical experience and regularly share their results (e.g. [MS2009, Mor2012a, Mor2012b]). Their contribution to one of the aforementioned special sections is noteworthy [MK2005], since it already contains many major topics to later appear in their book, together with a detailed historical overview over aircraft system identification at NASA Langley Research Center.

The vast number of papers and their diverse applications again prohibit an exhaustive overview. Only some of the publications by BOTTASSO are mentioned here, due to their close relation to what is to be presented in the remainder of this work: In [BLM2009b, BLMM2010] the similarities between parameter estimation and trajectory optimization are pointed out, and a special single-multiple shooting method is illustrated. The latter has already been discussed earlier in [BM2009].

After more than five decades of a numerous activities in the field of aircraft system identification, the well-established methodologies can accomplish good results in a wide area of fields [MK2016, Ch. 1]

- it is possible to determine the structure of aerodynamic model equations along with the numerical values of the corresponding parameters, as well as uncertainty bounds; one prerequisite for this is that the data has a certain quality.
- filtering techniques provide the means to distinguish between measurement noise and external disturbances, such as atmospheric turbulences or modeling errors
- these filtering techniques, combined with parameter estimation approaches, can be used to determine systematic instrumentation errors
- prior knowledge can be included in many estimation algorithms
- as a by-product the approaches developed in aircraft system identification allow for the design of experiments with a maximum information content.

However, despite the fact that standard approaches yield very good results, there is still room for improvement. The following issues have been identified:

- Optimization has been understood to be a necessary step in most parameter estimation tasks. However, so far the capabilities of modern optimization and optimal control formulations are not fully exploited: the well-established solution methodologies for dynamic system identification usually consider a single shooting formulation. In modern optimal control however, full-discretization has advanced to be the state of the art, but has of yet not been widely used for parameter estimation tasks. Its increased robustness and better convergence properties may however greatly benefit any system identification project.

  Also, off-the-shelf tools such as the optimal control tool FSD optimAL CONtrol tool for MATLAB (FALCON.m) [RBG+2018] may then be used for the solution of parameter estimation tasks, with manageable additional implementation effort.
- The standard literature on flight vehicle system identification usually advises the analyst to try to obtain time-synchronized sensor readings with equal sampling frequencies. However, in many real-life scenarios, requirements on sensor equipment are mainly driven by automatic control necessities. Equal sampling frequencies are then of minor concern. This is especially true for Remotely Piloted Aerial System (RPAS) applications. Due to tight budgets and short development cycles often no explicit sensors for system identification purposes are installed.

  So far, no mathematical rigorous treatment of signals showing different sampling rates is available, at least to the best of the author's knowledge. This issue is of great concern, if raw sensor data is to be processed: typical sampling frequencies for inertial sensors may be 50 to 100 times higher compared to e.g. Global Navigation Satellite System (GNSS) data, resulting in a large imbalance of the information content.
- One of the great advantages of the standard maximum likelihood approaches is that *no* explicit weighting of the outputs is necessary to obtain meaningful results, i.e. that manual tuning is reduced. Nonetheless, sometimes it might still be interesting to force the algorithm to focus more on a specific set of signals. This has not been possible so far, at least not without violating the underlying statistics.
- Existing optimization algorithms are capable to incorporate constraints, however mainly they are restricted to constraints on model parameters only. When using a full discretization formulation, the complete system dynamics are formulated as constraints. This has two major consequences: on the one hand more complicated (path-) constraints, possibly including states and inputs, can be considered in the optimization stage. On the other hand, extended use of constraints necessitates their inclusion when determining parameter uncertainty estimates. Even

though there exist approaches to overcome this challenge, they have not yet been rigorously applied in flight vehicle system identification.

- As MORELLI states in [MK2016, Ch. 6.3], the maximum likelihood cost function is often far from quadratic, with multiple local minima. This usually necessitates very good initial parameter guesses, that already lie in the vicinity of the correct local minimum in order for gradient based optimization algorithms to yield meaningful results. The computation of these initial guesses is often cumbersome and very project specific.

All of these shortcomings of the current state of the art will be addressed in the course of this work, and solutions will be proposed.

1.3 Objectives

As mentioned, the overall aim of this thesis is to bring the fields of optimal control and parameter estimation closer together, with a special focus on UAV applications.

Currently, one can get the impression that for many scientists working in parameter estimation, the optimization part of their work flow is merely a necessary means to an end. On the other hand, some people in the field of optimization and optimal control seem to consider parameter estimation a simple special case that may easily be solved using their standard tools.

This thesis now attempts to show that both statements are not entirely true. Even though the optimization part is an inevitable step in parameter estimation, it is more than just a necessary tool to obtain a solution. If use is made of recent advancements in the field of optimal control, more robust algorithms with larger convergence radii can be obtained.

As to the optimization view on parameter estimation: although most problems do come down to some sort of weighted least-squares problem, whose solution is quite straight forward, this is not the complete picture. Numerical values for the parameter estimates can only be half the result in parameter estimation. The necessary second part are metrics to assess the quality of the result, such as parameter error covariances. Quantifying and analyzing these is a routine step in parameter estimation, and it will be shown that the tools used therein are also meaningful from an optimization point of view.

Specifically, the thesis sets out to meet the following objectives:

- develop a unified mathematical description that contains both the standard methods as well as applied optimal control approaches for state and parameter estimation
- show the applicability of advanced optimal control methods, namely the use of full discretization, to parameter and state estimation

- develop a unified framework for uncertainty quantification for all cases
- highlight both sides of the topic, i.e. the parameter estimation from an optimization point of view and optimization from a parameter estimation point of view
- develop solutions to challenges that appear when actually applying the algorithms to real life problems. These include interpolation approaches specifically tailored to the task at hand, scaling approaches to make convergence more robust, and several practical methods for uncertainty quantification to choose from
- collect existing solutions to challenges in real-life application, such as derivative determination, residual covariance estimation and sensitivity computations
- show the potential in the synergy between parameter estimation and optimal control for future work in e.g. model validation, optimal input design, or optimal control with parameter uncertainties.
- implement all of the above in a modular, easy to handle fashion that enables the user to quickly get started with parameter estimation.

1.4 Contributions

The main contributions of this thesis to the field of parameter estimation and to optimal control are to be illustrated next.

Identification of Optimal Control Methods Suitable for Parameter and State Estimation

In the current literature, optimal control and parameter estimation are commonly treated as two separate fields. This thesis investigates those aspects of direct optimal control that may advantageously be applied in the context of flight vehicle system identification.

Works both on optimization, optimal control, and estimation theory are abundant, but only few highlight the commonalities between these fields. Some steps in this direction have been taken [BLM2009b, BLMM2010], however not with the theoretical depth as will be presented here.

Part I: “Preliminaries” assembles this background information, where every piece is not new per se, but the overall collection, and its level of detail in both fields, has not yet been assembled in one work. The main result of this part, next to providing necessary background information, is the statement of a very generic state and parameter estimation task, formulated as an optimal control problem. At the same time, it is shown how the formulations that commonly appear in the literature (smoothing state estimation, output error parameter estimation, etc.) may be considered as special cases of this general formulation.

Application Details for Solution of Real-Life Parameter Estimation Problems
In this thesis, clear rules, algorithms and approaches are presented to overcome implementation considerations that arise from real-life applications.

Some of the issues that have to be solved in practical applications are: the determination of meaningful weighting matrices; the computation of cost function derivatives; the proper scaling of parameters, states, outputs, and cost; the reduction of the problem size; If these aspects are not addressed properly, results may remain unsatisfactory or computational times may increase beyond acceptable limits.

For some of these aspects, well-established solutions exist, whereas for others, new approaches have been developed during the work on this thesis. All are collected in Chapter 3 "Implementation Aspects of System Identification Using Optimal Control Methods", which will eventually help to make the aircraft system identification process more efficient, streamline and robust.

New results to be presented are an argument why the cost function Hessian is approximately equal for given versus estimated residual covariance matrices (section 3.2).

Further, an efficient interpolation scheme for the solution of the dynamic system equations is illustrated (section 3.9). It is able to reduce the problem size considerably, while keeping approximation errors small. This speeds up computations in the early trial-and-error phase when determining a proper model structure, thus enabling the handling of large data-sets in the first place.

As with many other optimization problems, proper scaling is imperative. Two approaches to scaling the cost function are investigated (section 3.10), where an approach to directly scale the residual covariance matrix is new. It may be used to account for different sampling frequencies in the residual covariance matrix estimate; or simply to have more weight on a specific output. Preliminary ideas have been published by the author [GH2016]. Here, they are developed further by conducting a thorough investigation of the scaling approach, together with its consequences for gradient and Hessian computations.

Mathematical Framework for Uncertainty Quantification in Direct Optimal Control
It is imperative for any parameter estimation algorithm to provide means of uncertainty quantification together with the actual estimates. This thesis presents a mathematical framework that is capable of quantifying the estimation uncertainties, also for the application of direct optimal control to parameter estimation.

Standard textbook approaches for aircraft parameter estimation cannot achieve this, since they are in general not able to properly incorporate constraints on the optimization variables. However, in direct optimal control the system's dynamics are enforced *purely* via constraints. To solve this issue, a new method is proposed that is able to incorporate arbitrary constraints. At the same time, it contains the standard approach

for unconstrained single-shooting as a special case.

The background on this is illustrated in section 3.5 "Unified Approach for Computation of Parameter Uncertainties". So far, parameter covariance estimates in flight vehicle system identification have only been provided for the unconstrained case. In other fields, there exist specific solution algorithms for multiple shooting formulations [Boc1987], but this has not yet been transferred to the aerospace field.

Detailed Formulation of Estimation Problem in Optimal Control Framework
Eventually, the parameter estimation problem is formalized as a fully discretized optimal control problem.

It is illustrated how standard methods for dynamic system identification for deterministic and stochastic systems constitute single shooting problems with a specific cost function; the latter being based on statistical considerations. Then the formalism is extended to a full discretization transcription technique for both deterministic and stochastic system formulations. The details are presented in Chapter 4 "Application of Optimization to Parameter Estimation Problems", combining all building blocks that have been discussed in the preceding chapters. Thus, the full array of optimal control tools becomes available for parameter estimation, including the use of high-end off-the-shelf optimization algorithms and improved convergence radii of modern approaches.

Steps toward using full discretization in parameter estimation have been taken in the past, by applying multiple-(together with single) shooting methods [BM2009, BLM2009b]. In an overview chapter, SCHITTKOWSKI lists many different formulations for parameter estimation in dynamics systems, with most examples being taken from chemistry applications. On a side-note he also mentions full discretization as a possible way for its solution. However, no more details are provided, and parameter estimation is again merely presented as a curve fitting problem, thereby neglecting all statistical interpretations [Sch2000].

To the best of the authors knowledge, using full-discretization for parameter estimation in the aerospace field has not been investigated thoroughly; merely preliminary results have been published by the author [GGBH2016].

Generic Approach for Initial Guess Improvement for Model Parameters
Most parameter estimation algorithms are iterative in nature. Thus, the quality of the results crucially depends on meaningful initial model parameter guesses. Section 3.8 "Automatic Improvement of Initial Guesses During Initial Optimization Iterations" shows two approaches to improve upon them. Both do not necessitate auxiliary information about the problem such as wind tunnel data, or previous estimation results. Merely meaningful state information and a suitable model structure need to be available. The first method had been reported already, whereas the second, which is again

closely related to the idea of full-discretization, has not yet been encountered.

Most of the examples in chapter 6 illustrate that applying these two approaches, together with the larger convergence radius of fully discretized problems, they provide a very high robustness against bad initial model parameter guesses. In most example cases it was enough to provide a vector of zeros to obtain good results.

Implementation of an Integrated Parameter Estimation Tool
Finally, all of the aspects developed throughout the theoretic chapters of the thesis have been implemented in an add-on to the MATLAB optimal control toolbox FALCON.m [RBG+2018], which is being developed at the Institute of Flight System Dynamics. In parameter estimation, no single algorithm will always perform perfectly. It was thus realized that the overall goal of any implementation has to be to provide a large range of possible solution methods, which may easily be interchanged.

The user is then able to apply different algorithms to the same problem definition, with minimal changes in the source code, in order to see which performs best for the problem at hand. The add-on is set-up such that

- all convenience features (interpolation, scaling, consideration of prior information, initial guess improvement, ...) can be quickly de-/activated;
- all necessary features (derivative computation, determination of uncertainty estimates) are tightly incorporated;
- with the change of a few lines of code, the same problem definition may be solved using standard or advanced optimal control methods.

1.5 Structure of the Thesis

The overall structure of the thesis is shown in figure 1.1. After this introduction, the main text is split in three parts.

The first **part I** is dedicated to the necessary mathematical preliminaries in the fields of optimization and optimal control, estimation theory and state estimation. The three subsections of this part are relatively independent, merely some optimization aspects will be used in the estimation chapters. Thus, readers familiar with the respective topics may only browse through them cursorily, to get acquainted with the nomenclature in use.

Chapter 2.1 summarizes the basics in continuous cost function optimization with continuous algebraic constraints, using gradient based algorithms. Additionally, discretization and transcription methods to translate a continuous time optimal control problem into a non-linear programming problem are discussed.

In **Chapter 2.2**, some basics in statistical estimation are summarized, and their use illustrated for maximum likelihood, least-squares, and Bayesian estimation. These

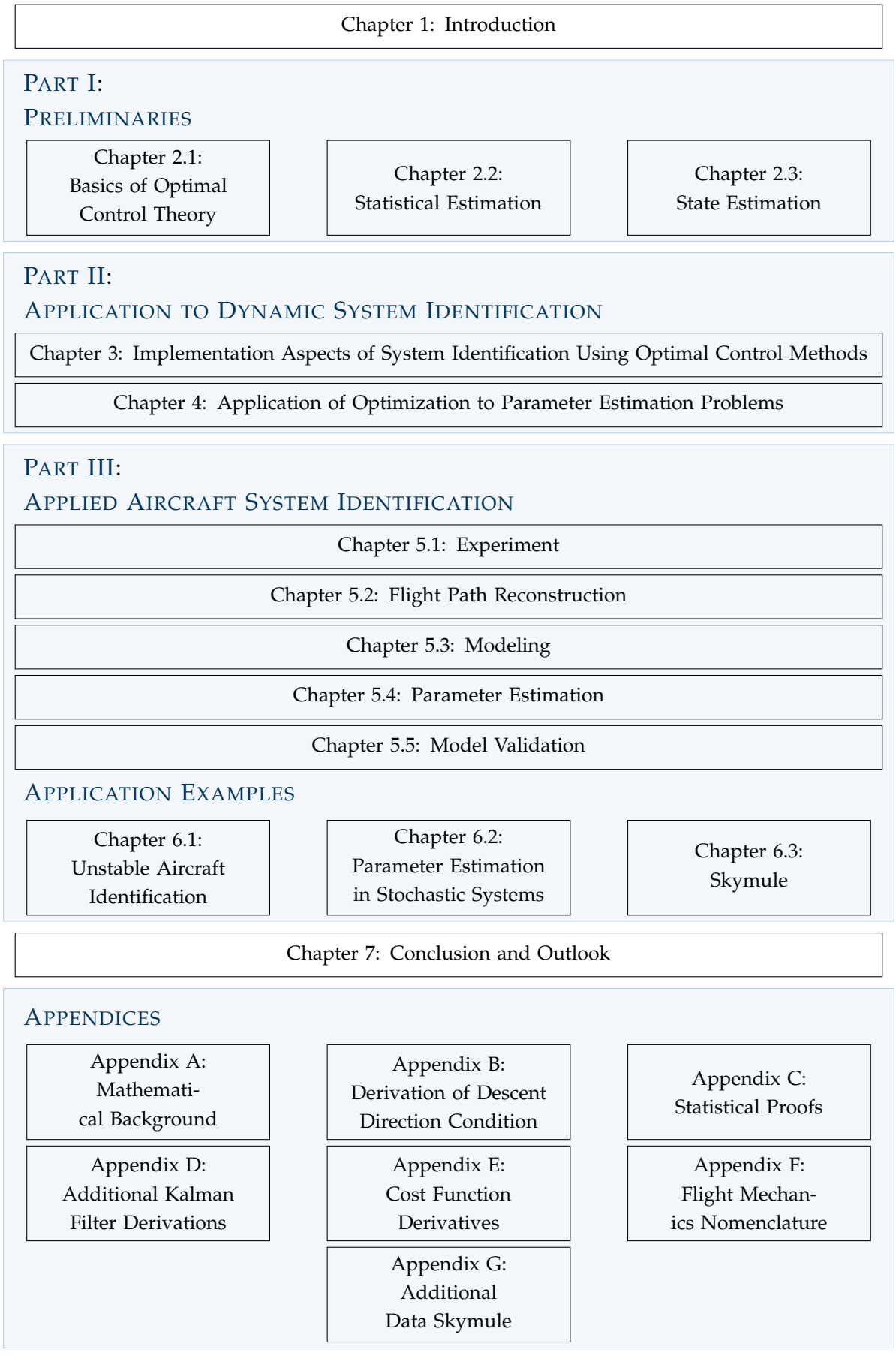


Figure 1.1: *Structure of the thesis*

three estimators will also be the most frequently used ones during the remainder of the thesis.

The first part is concluded by **Chapter 2.3**, which discusses the basics in linear and non-linear state estimation. The most prominent state estimation algorithm in this chapter is the Kalman filter, but also maximum likelihood state estimation (leading amongst others to the RTS smoother) will be considered. Towards the end of the chapter it will be shown that the maximum likelihood state and parameter estimation approach leads to a very general problem formulation that incorporates all of the aforementioned estimation approaches as special cases.

The second **part II** then covers aspects that are necessary for the actual application of the estimation algorithms. In this part, most of the contributions of this thesis will be discussed in detail.

Chapter 3 illustrates some more general aspects that are helpful irrespectively of the precise formulation of the problem. These aspects consider, amongst others, the estimation of covariance matrices, determination of derivatives, a general approach for uncertainty quantification, as well as specifically tailored interpolation and scaling considerations.

Finally, four possible problem formulations are stated and discussed in **Chapter 4**. They consider deterministic and stochastic systems, which are transcribed using single shooting and full discretization approaches.

The third **part III** collects information on the actual process of aircraft system identification, together with RPAS application examples. **Chapter 5** gives a brief overview over the major steps in a typical project, which namely are planning and executing experiments, preprocessing raw data, modeling, parameter estimation and model validation.

The following **Chapter 6** is dedicated to application examples and discusses three different, independent scenarios. At first, **Chapter 6.1** shows the performance of the full-discretization approach for unstable aircraft models. Another example in **Chapter 6.2** considers the application to stochastic systems. The last, most complex example shows an application using the low-cost flying testbed SKYMULE in **Chapter 6.3**. A full six degree of freedom model of the aircraft is derived, purely based on the available, low-cost sensor data.

A brief conclusion and possible future work is outlined in the last **Chapter 7**.

The appendices contain additional background information that is necessary for a complete discussion, but would unnecessarily disrupt the flow of the text:

- **Appendix A**: layout conventions, matrix derivative rules, statistical basics
  It is advised to at least browse through this appendix, since it assembles all background information that is assumed to be known in the main body of this text.
- **Appendix C**: statistical proofs e.g. for asymptotic characteristics of maximum likelihood estimates

- **Appendix E**: analytical forms for the gradient and Hessian of the cost functions in use
- **Appendix D**: relation between discrete and continuous time process noise, equivalence of maximum likelihood and minimum variance state estimation
- **Appendix B**: descent direction condition
- **Appendix F**: flight mechanics coordinate frames, designation rules
- **Appendix G**: complementary attitude filtering, additional SKYMULE figures

Part I

Preliminaries

It is remarkable that a science which began with the consideration of games of chance should have become the most important object of human knowledge.

Pierre-Simon Laplace, 1814 [Lap1814]

2

Mathematical Preliminaries

In the following, the mathematical basics for the application of optimal control methods to parameter estimation problems will be laid out. At first some basics of optimization and optimal control will be illustrated, followed by a discussion of the fundamentals in estimation theory. The chapter is concluded by a presentation of some state estimation concepts and their relation to optimization and estimation theory.

2.1 Basics of Optimal Control Theory

The solution to the estimation problems that will be discussed throughout this thesis eventually come down to solving an optimization problem. This is why a basic discussion of cost function optimization, together with some optimal control basics are to be presented in this chapter. The field of optimization is huge, thus only the few aspects necessary to understand the application of optimization in the parameter estimation context are illustrated. For more in-depth treatment the interested reader may consult the standard works on these topics, some of which are [Bet2010, Ger2017, Ger2018, Ger2012, Bit2017, Rie2017, BA2010].

The overall goal will eventually be to determine values for the *model parameters* $\boldsymbol{\theta}$, that are optimal with respect to some identification criterion. However, they are seldom the only values that need to be incorporated in the optimization. In order to get meaningful results, additional spurious parameters need to be considered. Commonly, these include initial conditions, or bias terms. However, if methods from the field of optimal control are to be used, complete state histories, and even auxiliary inputs may be subject to optimization.

In order to distinguish between these two groups of parameters, the model parameters will be denoted as $\boldsymbol{\theta}$, whereas the complete optimization vector is $\mathbf{z}$. The latter must not be confused with the measurement vector $\boldsymbol{z}$, which will be introduced in later chapters. Different typesetting and the context should make an easy distinction possible.

2.1.1 Basics of Cost Function Optimization

The focus of this work will be restricted to problems of the following type [Bet2010, Ch. 1]:
Find the

1. parameter vector $\mathbf{z}^* \in \mathbb{R}^{n_\mathbf{z}}$ of finite size $n_\mathbf{z}$,
2. such that the cost function $J(\mathbf{z})$ is optimized
3. without violating a set of constraints, which are formulated as equality and inequality constraints

$$\boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0} \tag{2.1}$$

$$\boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0} \tag{2.2}$$

where the inequality sign in above equation is to be understood element-wise.

A *feasible* point is one, where all constraints are satisfied. The set of all feasible points is called the *feasible region* $\mathcal{Z}$. If no constraints are imposed, i.e. if the feasible region is equal to $\mathbb{R}^{n_\mathbf{z}}$, the task is said to be an *unconstrained problem*.

The most general form of optimization problems to be considered here may be summed up as

$$\begin{pmatrix} & \min\limits_{\mathbf{z}} J(\mathbf{z}) \\ \text{s.t.} & \boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0} \\ & \boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0} \end{pmatrix} \tag{2.3}$$

Only minimization problems will be considered, since every maximization task can be converted by multiplying the cost function with -1. A local minimum is a point, for which it holds [Ger2017]

$$J(\mathbf{z}) \geq J(\mathbf{z}^*) \qquad \forall \mathbf{z} \in \mathcal{Z} \cap U_\delta(\mathbf{z}^*) = \{\mathbf{z} : \|\mathbf{z}^* - \mathbf{z}\| \leq \delta\} \tag{2.4}$$

i.e. a point in the feasible region $\mathcal{Z}$, where, in a small neighborhood U_δ around $\mathbf{z}^*$, no smaller function value of J can be obtained. If no smaller function value can be reached in the complete feasible region $\mathcal{Z}$, the minimum is called *global*. In general, global minima are hard to find, and most gradient-based optimization algorithms only provide local minima.

2.1.2 Unconstrained Optimization

First the unconstrained case is discussed, i.e. problems of the type

$$\min_{\mathbf{z}} J(\mathbf{z}) \tag{2.5}$$

where the feasible region is the whole of $\mathbb{R}^{n_z}$. Then, necessary conditions for a local minimum $\mathbf{z}^*$ are [Bet2010, Ch. 1]

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}^*} = \mathbf{0} \tag{2.6}$$

$$(\mathbf{z}^* - \mathbf{z}) \left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}^*} (\mathbf{z}^* - \mathbf{z})^\mathsf{T} \geq 0 \qquad \forall \mathbf{z} \in U_\delta(\mathbf{z}^*) = \{\mathbf{z} : \|\mathbf{z}^* - \mathbf{z}\| \leq \delta\} \tag{2.7}$$

whereas their sufficient version is

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}^*} = \mathbf{0} \tag{2.8}$$

$$(\mathbf{z}^* - \mathbf{z}) \left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}^*} (\mathbf{z}^* - \mathbf{z})^\mathsf{T} > 0 \qquad \forall \mathbf{z} \in U_\delta(\mathbf{z}^*) = \{\mathbf{z} : \|\mathbf{z}^* - \mathbf{z}\| \leq \delta\} \setminus \mathbf{z}^* \tag{2.9}$$

The first equation in both cases requires the gradient vector of J to be $\mathbf{0}$ at $\mathbf{z}^*$. Since the gradient points in the direction of the steepest ascend of J [Ger2017], a decrease of the cost function is always possible in the direction $-\frac{\partial J}{\partial \mathbf{z}}$ (see also the following discussion of the descent condition (2.16)). This is why it is intuitive that at any optimal point according to equation (2.6) the cost function gradient has to vanish.

The second equation in both cases enforces a positive (semi-)definite Hessian of the cost function in a small environment around the optimal $\mathbf{z}^*$. This can be illustrated using a second order Taylor expansion of J around the optimal point $\mathbf{z}^*$

$$\begin{aligned} J(\mathbf{z}) &\approx J(\mathbf{z}^*) + \underbrace{\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}^*}}_{=\mathbf{0}} (\mathbf{z} - \mathbf{z}^*) + (\mathbf{z} - \mathbf{z}^*) \left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}^*} (\mathbf{z} - \mathbf{z}^*)^\mathsf{T} \\ &= J(\mathbf{z}^*) + (\mathbf{z} - \mathbf{z}^*) \left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}^*} (\mathbf{z} - \mathbf{z}^*)^\mathsf{T} \end{aligned} \tag{2.10}$$

If in above equation the Hessian $\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}^*}$ is positive (semi-)definite, the second term is greater than (or equal to) zero and $J(\mathbf{z}^*)$ has to be a local minimum of $J(\mathbf{z})$ with the definition in (2.4).

2.1.2.1 Newton's Method in Optimization

A common way to solve an unconstrained optimization problem is then to find a point that satisfies equation (2.6). This can be achieved by applying Newton's algorithm to finding its root: linearize the gradient $\frac{\partial J}{\partial \mathbf{z}}$ about the current estimate $\mathbf{z}_i$, and use the root

of the linearization as the next iteration point [Bet2010] [MK2016, Ch. 6.1]

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_{i+1}} \approx \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} + \left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i} (\mathbf{z}_{i+1} - \mathbf{z}_i) \stackrel{!}{=} \mathbf{0} \tag{2.11}$$

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \left(\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i}\right)^{-1} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} \tag{2.12}$$

This linear approximation to the gradient corresponds to a quadratic approximation of the cost function; the gradient's root is then equal to the approximation's minimum. The expression

$$\boldsymbol{d}_i = \mathbf{z}_{i+1} - \mathbf{z}_i \tag{2.13}$$

is generally termed the *search direction*, i.e. the direction in the parameter space, in which the next, better iterate is searched for. If Newton's approach to root finding is used, it is

$$\boldsymbol{d}_i = -\left(\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i}\right)^{-1} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} \tag{2.14}$$

Since in above approach, merely a root of the gradient is determined, it cannot be guaranteed that a minimum is approached rather than a stationary point or a maximum. To avoid a cost function increase, it has to be enforced that $\boldsymbol{d}$ is a *descent direction* at $\mathbf{z}_i$, i.e. there has to exist a $\bar{\alpha}$ such that [Ger2017]

$$J(\mathbf{z}_i + \alpha \boldsymbol{d}_i) < J(\mathbf{z}_i) \qquad \forall 0 < \alpha \leq \bar{\alpha} \tag{2.15}$$

At every iteration, the step along $\boldsymbol{d}_i$ (possibly reduced by a factor of up to $\bar{\alpha}$) must lead to a cost function reduction.

A sufficient condition for $\boldsymbol{d}_i$ to be a descent direction is [Ger2017]

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \boldsymbol{d}_i < 0 \tag{2.16}$$

i.e. the directional derivative of the cost function along the search direction $\boldsymbol{d}_i$ has to be strictly negative. The proof can be found in Appendix B. Geometrically, this means, that the search direction and cost function gradient have to enclose an angle between $90°$ and $270°$. Thus the hyperplane through $\mathbf{z}_i$ and perpendicular to $\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}$ divides the solution space $\mathbb{R}^{n_\mathrm{z}}$ into two half-spaces. Any descent direction needs to have at least a small component in the negative gradient direction, i.e. needs to point into the half-space opposite the gradient direction.

For search directions based on Newton's algorithm, this results in [Bet2010]

$$-\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \left(\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i}\right)^{-1} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} < 0 \tag{2.17}$$

Thus, for a Newton based optimization step, the Hessian matrix needs to be positive definite in order to achieve a cost function reduction.

Close to the optimum, the Newton approach exhibits *quadratic convergence* [Bet2010, Ch. 1]. Loosely speaking this means that at every iteration step, the number of significant figures of the solution roughly doubles, which is very desirable. A more mathematically rigorous definition is [Ger2017, Def. 3.9.4]: the sequence of points $\mathbf{z}_i$ is said to converge quadratically against $\mathbf{z}^*$ if there is a $c > 0$ such that

$$|\mathbf{z}_{i+1} - \mathbf{z}^*| \leq c|\mathbf{z}_i - \mathbf{z}^*|^2 \qquad \forall i \in \mathbb{N} \tag{2.18}$$

$$\text{i.e. } |\mathbf{z}_{i+1} - \mathbf{z}^*| = \mathcal{O}\left(|\mathbf{z}_i - \mathbf{z}^*|^2\right) \tag{2.19}$$

i.e. at every iteration, the difference to the target value $|\mathbf{z}_{i+1} - \mathbf{z}^*|$ is of the order of magnitude of the difference in the last iteration, squared $|\mathbf{z}_i - \mathbf{z}^*|^2$.

However, quadratic convergence is only possible, if accurate information about the matrix of second derivatives is available. In general, this can be challenging for complex systems and cost functions, since computing first derivatives may already be very cumbersome; or if done using finite differences, prone to numerical error. For second order derivatives this problem grows even worse, which is why approximations to the Hessian are very common. Most of these approximations lead to a decrease in the convergence rate to a superlinear instead of quadratic convergence [Bet2010, Ch. 1].

However, for many of the parameter estimation algorithms to be discussed, a very good analytic approximation to the Hessian is possible, see section 3.2. Thus, at least close to the optimum, their convergence rate should be close to quadratic.

2.1.2.2 Stopping Criteria

Above iteration cannot go on forever, but meaningful abort criteria have to be defined. From a theoretical point of view, the cost function gradient needs to be zero. However, in numerical practice, this can usually not be achieved, due to the errors and approximations involved in the solution of real-life problems. Nevertheless, criteria of the form

$$\left\| \left. \frac{\partial J}{\partial \mathbf{z}} \right|_{\mathbf{z}=\mathbf{z}_i} \right\| \leq \epsilon \tag{2.20}$$

together with a user-definable threshold ϵ are usually included in the optimization algorithm [Ger2017, Ch. 3.6].

A downside of above criterion is its scale variance, i.e. when scaling the cost function, the norm of the gradient changes as well, even though the value of the optimiza-

tion vector does not. Thus, other, often relative, criteria are introduced [GMW1982]

$$\frac{J(\mathbf{z}_{i-1}) - J(\mathbf{z}_i)}{1 + |J(\mathbf{z}_i)|} \leq \epsilon \tag{2.21}$$

$$\frac{\|\mathbf{z}_{i-1} - \mathbf{z}_i\|}{1 + \|\mathbf{z}_i\|} \leq \sqrt{\epsilon} \tag{2.22}$$

$$\frac{\left\| \frac{\partial J}{\partial \mathbf{z}} \Big|_{\mathbf{z}=\mathbf{z}_i} \right\|}{1 + |J(\mathbf{z}_i)|} \leq \sqrt[3]{\epsilon} \tag{2.23}$$

For small, absolute values, these criteria lose their relative nature and become absolute themselves [Ger2017, Ch. 3.6]. Usually, a subset of them is used, which has to be fulfilled simultaneously, in order to determine when to stop the optimization process.

MORELLI provides some numerical values for the model parameters, which have been successfully used in aircraft system identification [MK2016, Ch. 6.3]

$$\begin{aligned}
\left| \left[\hat{\boldsymbol{\theta}}_i\right]_{(j)} - \left[\hat{\boldsymbol{\theta}}_{i-1}\right]_{(j)} \right| &< 1{\times}10^{-5} \qquad \forall j = 1 \ldots n_{\boldsymbol{\theta}} \\
\frac{\left\| \hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_{i-1} \right\|_2}{\left\| \hat{\boldsymbol{\theta}}_i \right\|_2} &< 1{\times}10^{-3} \\
\left| \frac{J\left(\hat{\boldsymbol{\theta}}_i\right) - J\left(\hat{\boldsymbol{\theta}}_{i-1}\right)}{J\left(\hat{\boldsymbol{\theta}}_{i-1}\right)} \right| &< 1{\times}10^{-3} \\
\left| \frac{\partial J\left(\hat{\boldsymbol{\theta}}_i\right)}{\partial \left[\hat{\boldsymbol{\theta}}\right]_{(j)}} \right| &< 5{\times}10^{-2} \qquad \forall j = 1 \ldots n_{\boldsymbol{\theta}}
\end{aligned}$$

The first criterion, being in terms of absolute model parameter values, is very application dependent. That threshold given by MORELLI may be adequate for aerodynamic parameter estimation, but may fail in other contexts, where the order of magnitude of the respective parameters is different.

2.1.2.3 Globalization Strategies for Newton's Method

As mentioned above, Newton's method exhibits quadratic convergence close to the optimum of the cost function. However, far away from it, the method might even diverge, e.g. if the inherent quadratic approximation to the cost function is locally not good enough. In these cases, precautions have to be taken in order to eventually find the optimal point.

One of the methods, which is easiest to implement in this context is *step-halving* [Jat2015, Ch. 4] [WP1997, Ch. 4.3]: if a full step in the search direction $\boldsymbol{d}$ does not yield a decrease of the cost function, then a reduced step of $\frac{1}{2^\alpha}\boldsymbol{d}_i$ might have the desired effect.

Algorithm 2.1: Newton Algorithm with Step Halving

Initialization:

I. initialize parameter values $\mathbf{z}_0$ from some other source

Main Part:

II. compute current cost function $J(\mathbf{z}_i)$, gradient $\frac{\partial J}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_i}$, and Hessian $\frac{\partial^2 J}{\partial \mathbf{z}^2}\Big|_{\mathbf{z}=\mathbf{z}_i}$

III. determine the parameter update

$$\boldsymbol{d}_i = -\left(\frac{\partial^2 J}{\partial \mathbf{z}^2}\bigg|_{\mathbf{z}=\mathbf{z}_i}\right)^{-1} \frac{\partial J}{\partial \mathbf{z}}\bigg|_{\mathbf{z}=\mathbf{z}_i}$$

IV. increase the halving parameter sequentially $\alpha = 0, 1, 2, \ldots$

(a) if a cost function reduction was achieved

$$J\left(\mathbf{z}_i + \frac{1}{2^\alpha}\boldsymbol{d}_i\right) < J(\mathbf{z}_i)$$

compute the new parameter values

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \frac{1}{2^\alpha}\boldsymbol{d}_i;$$

and go to step V.

(b) if the maximum number of halving steps was exceeded, report to the user and stop optimization

V. check convergence of the algorithm via any combination of criteria in section 2.1.2.2. If no convergence was achieved, go to step II.

This can be checked sequentially for increasing $\alpha = 0, 1, 2, \ldots$ until either a reduction in the cost function is achieved, or a maximum number of halving steps is exceeded. The latter would indicate more serious problems. In order to eventually come close to the (theoretically) quadratic convergence of Newton's method, it is desirable to aim for $\alpha = 0$ in this approach. The algorithm is illustrated in Algorithm 2.1, and implemented as possible solver for the parameter estimation extension to FALCON.m [RBG+2018].

An alternative would be a scalar line-search along the descent direction $\boldsymbol{d}_i$. However, most cost functions arising in parameter estimation problems have a somewhat quadratic form, which fits nicely with the basic assumptions in Newton's method. Thus the simple step-halving approach is often preferred over more complex line-search approaches.

An idea that is different from Newton's method would be a *steepest descent* approach: Consider the directional derivative $\frac{\partial J}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \boldsymbol{v}$ of the cost function J at $\mathbf{z}_i$ in a normalized direction $\boldsymbol{v}$, with $\boldsymbol{v}^\intercal \boldsymbol{v} = 1$. Using the SCHWARZ inequality, the magnitude

of the directional derivative is limited from above by [Sor1980, App. A.3]

$$\left(\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \boldsymbol{v}\right)^2 \leq \left(\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}\right) \underbrace{(\boldsymbol{v}^{\mathsf{T}}\boldsymbol{v})}_{=1} = \left(\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}\right) \tag{2.24}$$

Thus the maximum rate of change in the cost function, is obtained in the direction of the gradient $\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}$ [Ger2017]. Together with the descent condition (2.16), it is then clear that the negative gradient direction yields the "steepest descent". This gives rise to the idea of a search direction of the form

$$\boldsymbol{d}_i = -\beta \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} \qquad \text{with } \beta > 0 \tag{2.25}$$

This automatically satisfies the descent condition (2.16)

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \boldsymbol{d}_i = -\beta \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}^{\mathsf{T}} \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} < 0 \tag{2.26}$$

but β has to be chosen small enough to keep the local approximations valid. Thus a search direction can be determined, without having to compute a matrix of second derivatives.

The approach has two major disadvantages

1. although it provides a search direction, it does not provide information about the magnitude of the step, i.e. some line search along $-\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i}$ to determine a "good" step size β is necessary
2. according to [Mar1963], the algorithm exhibits serious convergence issues close to the minimum

Especially the second argument usually outweighs the benefit of the simplicity of the approach. Also, in parameter estimation problems, good approximations to the Hessian of the problem are available, with only minor additional computations.

A third possibility, namely the LEVENBERG-MARQUARDT algorithm, combines the advantages of a pure Newton iteration with the steepest descent idea. Instead of solving for a Newton search direction, the modified system of equations

$$\left(\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i} + \lambda_i \mathbf{S}_i\right) \boldsymbol{d}_i = -\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} \tag{2.27}$$

$$[\mathbf{S}_i]_{(m,n)} = \begin{cases} \left[\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i}\right]_{(m,n)} & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

is considered. The matrix $\mathbf{S}_i$ is a diagonal matrix with the same main diagonal elements as the Hessian $\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i}$.

LEVENBERG [Lev1944] first introduced the idea of having a damping factor λ_i in the context of non-linear least-squares. His reasoning was based on the argument that

in addition to minimizing the residuals, the iteration steps $\boldsymbol{d}_i$ should be minimized as well, in order to keep the validity of the underlying Taylor approximation. MARQUARDT [Mar1963] extended the idea further by realizing that this approach constitutes a combination of the pure Newton step, with a steepest descent step [Jat2015, Ch. 4]:

1. for $\lambda_i \to 0$ the update step becomes the Newton step
2. for $\lambda_i \to \infty$ the update step becomes a scaled steepest descent step

$$\boldsymbol{d}_i \approx -\frac{1}{\lambda_i}\mathbf{S}^{-1}\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_i} \tag{2.29}$$

Additionally, since the properties of the gradient descent approach are not scale invariant, MARQUARDT proposed the scaling matrix $\mathbf{S}_i$ based on statistical considerations[1] originating in least-squares estimation. However the idea was seen to work fine in other contexts, too [Jat2006].

The ultimate goal is to drive the damping parameter λ_i to zero towards the end of the optimization, in order to obtain the quadratic convergence of the Newton algorithm. However, in the beginning a large damping parameter might be necessary (making the step more steepest descent like) in order to achieve any cost function reduction at all. This may happen if the initial guess is outside the radius of convergence of the pure Newton method.

Since $\mathbf{S}_i$ is positive definite, there will eventually be a value for λ_i, which makes the sum $\left(\left.\frac{\partial^2 J}{\partial \mathbf{z}^2}\right|_{\mathbf{z}=\mathbf{z}_i} + \lambda_i \mathbf{S}_i\right)$ positive definite, and thus the descent condition (2.16) will be satisfied. MARQUARDT then proposes a heuristic, but intuitively appealing approach to choosing the damping parameter: The algorithm always tries to decrease λ_i, in order to make it more Newton-like and thus exploit its quadratic convergence property. Only if no cost function reduction can be achieved, the damping parameter is increased in order to obtain a valid parameter update. The magnitude of the decrease/increase in λ is controlled via the parameter ν [Mar1963] [CJ2012, Ch. 1]. The algorithm is summed up in Algorithm 2.2.

[1] In least-squares estimation above Hessian can be considered as covariance matrix of the parameters. MARQUARDT then proposed to scale the Hessian, gradient and update step by the parameter standard deviations using a matrix $\tilde{\mathbf{S}} = \operatorname{diag}\left(\left[\frac{\partial^2 J}{\partial \mathbf{z}^2}\right]_{(i,i)}^{-\frac{1}{2}}\right)$, thus $\tilde{\mathbf{S}}^{-1}\tilde{\mathbf{S}}^{-1} = \mathbf{S}$ [Mar1963]

$$\begin{aligned} & \left(\tilde{\mathbf{S}}\frac{\partial^2 J(\mathbf{z}_i)}{\partial \mathbf{z}^2}\tilde{\mathbf{S}} + \lambda_i \mathbf{I}_{n_{\mathbf{z}}}\right)\tilde{\mathbf{S}}^{-1}\boldsymbol{d}_i = -\tilde{\mathbf{S}}\frac{\partial J(\mathbf{z}_i)}{\partial \mathbf{z}} \\ \Leftrightarrow \quad & \left(\frac{\partial^2 J(\mathbf{z}_i)}{\partial \mathbf{z}^2} + \lambda_i \mathbf{S}\right)\boldsymbol{d}_i = -\frac{\partial J(\mathbf{z}_i)}{\partial \mathbf{z}} \end{aligned}$$

Algorithm 2.2: Levenberg-Marquardt Algorithm

Initialization:

I. initialize parameter values $\mathbf{z}_0$ from some other source

II. initialize the algorithm's parameters to e.g. $\lambda_0 = 0.001$ and $\nu = 10$

Main Part:

III. compute current cost function $J(\mathbf{z}_i)$, gradient $\frac{\partial J}{\partial \mathbf{z}}\Big|_{\mathbf{z}=\mathbf{z}_i}$, and Hessian $\frac{\partial^2 J}{\partial \mathbf{z}^2}\Big|_{\mathbf{z}=\mathbf{z}_i}$

IV. determine two parameter update values

$$ {}_{\lambda}\boldsymbol{d}_i = -\left(\frac{\partial^2 J}{\partial \mathbf{z}^2}\bigg|_{\mathbf{z}=\mathbf{z}_i} + \lambda_i \mathbf{S}_i\right)^{-1} \frac{\partial J}{\partial \mathbf{z}}\bigg|_{\mathbf{z}=\mathbf{z}_i} $$

$$ {}_{\nu}\boldsymbol{d}_i = -\left(\frac{\partial^2 J}{\partial \mathbf{z}^2}\bigg|_{\mathbf{z}=\mathbf{z}_i} + \frac{\lambda_i}{\nu} \mathbf{S}_i\right)^{-1} \frac{\partial J}{\partial \mathbf{z}}\bigg|_{\mathbf{z}=\mathbf{z}_i} $$

V. compare the cost function values for the two parameter updates

(a) if $J(\mathbf{z}_i + {}_{\nu}\boldsymbol{d}_i) \leq J(\mathbf{z}_i)$: ["accept reduced λ"]
set $\mathbf{z}_{i+1} = \mathbf{z}_i + {}_{\nu}\boldsymbol{d}_i$
reduce $\lambda_{i+1} = \frac{\lambda_i}{\nu}$ and continue with step VI.

(b) if $J(\mathbf{z}_i + {}_{\nu}\boldsymbol{d}_i) > J(\mathbf{z}_i)$, but $J(\mathbf{z}_i + {}_{\lambda}\boldsymbol{d}_i) \leq J(\mathbf{z}_i)$: ["keep λ unchanged"]
set $\mathbf{z}_{i+1} = \mathbf{z}_i + {}_{\lambda}\boldsymbol{d}_i$
keep $\lambda_{i+1} = \lambda_i$ and continue with step VI.

(c) if $J(\mathbf{z}_i + {}_{\nu}\boldsymbol{d}_i) > J(\mathbf{z}_i)$, and $J(\mathbf{z}_i + {}_{\lambda}\boldsymbol{d}_i) > J(\mathbf{z}_i)$: ["increase λ"]
increase $\lambda_i = \lambda_i \cdot \nu$
go back to step IV.

VI. check convergence of the algorithm via any combination of criteria in section 2.1.2.2. If no convergence was achieved, go to step III.

2.1.3 Constrained Optimization

Constrained optimization is necessary, whenever additional constraints on the optimization parameters have to be considered. Here, those constraints will be formulated as equalities and inequalities involving $\mathbf{z}$.

2.1.3.1 Equality Constraints

The optimization problems involving equality constraints, which are treated here, have the form

$$ \begin{pmatrix} \min\limits_{\mathbf{z}} J(\mathbf{z}) \\ \text{s.t.} \quad \boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0} \end{pmatrix} \tag{2.30} $$

Furthermore, the constraint gradients $\left.\frac{\partial [\boldsymbol{c}_{eq}]_{(l)}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}, l = 1 \dots n_{\boldsymbol{c}_{eq}}$ at stationary points of above problem are required to be linearly independent. The latter is commonly called Linear Independence Constraint Qualification (LICQ) [Ger2017]. LICQ also implies that the maximum number of equality constraints can be $n_{\mathbf{z}}$: then the solution is purely determined by the equality constraints, and the problem essentially transforms into a root finding task. For $n_{\boldsymbol{c}_{eq}} < n_{\mathbf{z}}$ it remains an actual optimization problem.

The classic solution approach to problems of the above type is to define the *Lagrangian* [Bet2010, Ch. 1.8]

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}) = J(\mathbf{z}) + \boldsymbol{\lambda}^\mathsf{T} \boldsymbol{c}_{eq}(\mathbf{z}) \tag{2.31}$$

where the λ_l are termed Lagrange multipliers (not to be confused with the LEVENBERG-MARQUARDT damping parameter of the foregoing section). This is then used as a cost function for an unconstrained optimization problem in the extended optimization vector $\begin{bmatrix} \mathbf{z}^\mathsf{T} & \boldsymbol{\lambda}^\mathsf{T} \end{bmatrix}$

$$\min_{\mathbf{z}, \boldsymbol{\lambda}} \mathcal{L}(\mathbf{z}, \boldsymbol{\lambda}) \tag{2.32}$$

Then the first order necessary conditions become

$$\mathbf{0} \overset{!}{=} \left.\frac{\partial \mathcal{L}(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} = \left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} + \left(\left.\frac{\partial \boldsymbol{c}_{eq}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}\right)^\mathsf{T} \boldsymbol{\lambda}^* \tag{2.33}$$

$$\mathbf{0} \overset{!}{=} \left.\frac{\partial \mathcal{L}(\mathbf{z}, \boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}^*} = \boldsymbol{c}_{eq}(\mathbf{z}^*) \tag{2.34}$$

By finding a solution to the unconstrained problem of equation (2.32), the constrained problem of equation (2.30) can be solved [Bet2010, Ch. 1.8].

The second equation above illustrates the fact that at an optimal point, the equality constraints have to be fulfilled. The first equation states that at the optimum, the cost function gradient may be non-zero, but it has to be a linear combination of the constraint gradients. This can be interpreted geometrically: The equality constraints, defined via the implicit functions $[\boldsymbol{c}_{eq}]_{(l)}(\mathbf{z}) = \mathbf{0}\ l = 1 \dots n_{\boldsymbol{c}_{eq}}$ represent hypersurfaces in $\mathbb{R}^{n_z}$, on which any feasible point has to lie. Now the constraint gradients $\left.\frac{\partial [\boldsymbol{c}_{eq}]_{(l)}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ always point in the direction that is perpendicular to the local tangent plane. In order to stay on the hypersurface, a feasible direction has to be *in* the tangent plane and thus perpendicular to the local constraint gradient.

$$\left(\left.\frac{\partial [\boldsymbol{c}_{eq}]_{(l)}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}\right)^\mathsf{T} \boldsymbol{d} = \mathbf{0} \tag{2.35}$$

Eventually a feasible search direction needs to be perpendicular to *all* constraint gradients[2] $l = 1 \dots n_{\boldsymbol{c}_{eq}}$.

[2] A different interpretation is that feasible descent directions need to be located in the *Null-Space* of the constraint Jacobian, a fact that will be relied on heavily when treating covariance estimates of equality constrained estimation problems in section 3.5.

Example 2.1: 2D example for equality constraint optimization
Consider the following, linear programming problem with elliptical equality constraints

$$\begin{pmatrix} \min\limits_{x_1,x_2} J = c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad \dfrac{1}{a_1^2} x_1^2 + \dfrac{1}{a_2^2} x_2^2 - 1 = 0 \end{pmatrix}$$

A graphical illustration can be seen in figure 2.1: At an optimal point (⊙), the cost function gradient can be completely compensated for by the constraint gradient and the corresponding optimal Lagrange multiplier. Also, since these conditions are merely necessary, not sufficient, there exists another stationary point (◀), where the cost function gradient is completely compensated for, but which has a larger cost function value.
At all other feasible points on the ellipse (✖), only parts of the cost function gradient can be compensated for by the constraint gradient, and a feasible descent direction $\boldsymbol{d} \neq \mathbf{0}$ remains.

It has been realized before that any descent direction must have a small component pointing in the negative gradient direction (see equation (2.16) and the following paragraph). However, if the cost function gradient $\frac{\partial J}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}$ can be represented as a linear combination of constraint gradients $\frac{\partial [\boldsymbol{c}_{eq}]_{(l)}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}$, it contains only directions that would violate one or the other constraint. The negative gradient direction then also consists only of those "forbidden" directions and no feasible descent direction can exist.

Lastly, it can be state that the sign of the Lagrange multipliers for equality constraints does not matter, since equating $\boldsymbol{c}_{eq}(\mathbf{z})$ to zero amounts to the same as equating $-\boldsymbol{c}_{eq}(\mathbf{z})$ to zero. The difference will only be the gradient direction, which is compensated for by a change in the sign of the Lagrange multiplier.

This set-up is illustrated in the simple example 2.1 and the accompanying figure 2.1.

2.1.3.2 Inequality Constraints

In this work, problems including inequality constraints are considered to be of the form

$$\begin{pmatrix} \min\limits_{\mathbf{z}} J(\mathbf{z}) \\ \text{s.t.} \quad \boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0} \end{pmatrix} \tag{2.36}$$

For these problems, the concept of an *active set* is fundamental [Bet2010, Ch. 1.9] [Ger2017, Ch. 5]: Inequality constraints that are satisfied at their boundaries, i.e. as equality constraints, are called *active*, as opposed to those, which are satisfied as strict inequalities and consequently called *inactive*. The active set $\mathcal{A}(\mathbf{z})$ then contains those indices for which the respective inequality constraint is active at point $\mathbf{z}$

$$\mathcal{A}(\mathbf{z}) = \left\{ l \,\middle|\, [\boldsymbol{c}_{ineq}(\mathbf{z})]_{(l)} = 0 \right\} \tag{2.37}$$

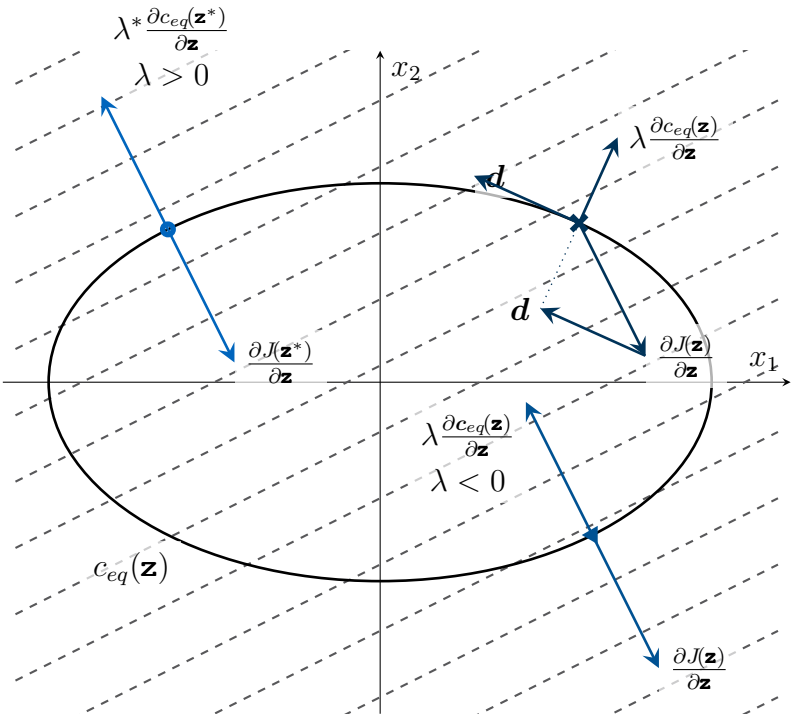


Figure 2.1: *Illustration of equality constraint optimization: cost contour (- - -) and elliptical equality constraints (——), optimal point (●), stationary but non-optimal point (◀), and an arbitrary, non-optimal, feasible point (✖).*

At an optimal point $\mathbf{z}^*$, the inequality constraints can thus be split into active and inactive constraints. The inactive constraints may be dropped from the problem formulation, since they are fulfilled anyway. For active inequality constraints, a reasoning similar to that of the last section can be applied: at an optimal point, the cost function gradient may be non-zero, but has to consist only of directions, which would violate the constraints. Otherwise a change in the cost function, while still keeping all constraints, would be possible. In the inequality case, those directions are (as before) the directions of the active inequality constraint gradients

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} = -\sum_{l \in \mathcal{A}(\mathbf{z}^*)} \mu_l^* \left.\frac{\partial [\boldsymbol{c}_{ineq}]_{(l)}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} \tag{2.38}$$

$$\mu_l^* \geq 0 \qquad \forall l \in \mathcal{A}(\mathbf{z}^*) \tag{2.39}$$

However, in the inequality case, only one direction violates the constraint, the other points into the feasible region. This fact manifests itself as above second condition on the Lagrange multipliers[3].

[3] some authors define the inequality constraints and Lagrangian using a different sign convention, as e.g. BETTS [Bet2010, Ch. 1.9]

$$\boldsymbol{c}_{ineq}(\mathbf{z}) \geq \mathbf{0}$$

$$\left.\frac{\partial J}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} = \sum_{l \in \mathcal{A}(\mathbf{z}^*)} \mu_l^* \left.\frac{\partial [\boldsymbol{c}_{ineq}]_{(l)}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$$

Example 2.2: 2D example for inequality constraint optimization
Consider the following, linear programming problem with elliptical inequality constraints

$$\begin{pmatrix} & \min\limits_{x_1,x_2} J = c_1 x_1 + c_2 x_2 \\ \text{s.t.} & \frac{1}{a_1^2} x_1^2 + \frac{1}{a_2^2} x_2^2 - 1 \leq 0 \end{pmatrix}$$

A graphical illustration can be seen in figure 2.2: At an optimal point (⊙), the cost function gradient is again compensated for by the inequality constraint gradient. Furthermore, the accompanying multiplier obeys the sign constraint $\mu^* > 0$. The second stationary point (◀) can be ruled out directly, since it is not possible to compensate the cost function gradient using positive multipliers, as indicated by the constraint gradient direction shown.
At all other feasible points within the ellipse (✖), the inequality constraint is inactive, thus no explicit treatment is necessary. The corresponding multiplier is $\mu = 0$.

Then a solution strategy is to apply the approach of the last section, together with an *active set strategy*. This keeps track of the active and inactive constraints during the iteration, and ensures the proper sign of the multipliers $\boldsymbol{\mu}$. As an illustration, example 2.2 presents an adjustment of the foregoing case to include inequality constraints. Figure 2.2 shows the corresponding graphical representation.

A simple example of an active set strategy for box constraint parameters will be illustrated in section 2.1.3.4 and algorithm 2.3.

2.1.3.3 Combined Equality and Inequality Constraints and KKT Conditions

Combining the information of the last two subsections, the so-called Karush-Kuhn-Tucker (KKT) Conditions arise [Ger2017, Ch. 5] [Bet2010, Ch. 1.12], which define the necessary conditions for an optimal point of a problem that is constrained by equality and inequality conditions:

Let $\mathbf{z}^*$ be a local optimum of the standard optimization problem

$$\begin{pmatrix} & \min\limits_{\mathbf{z}} J(\mathbf{z}) \\ \text{s.t.} & \boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0} \\ & \boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0} \end{pmatrix}$$

Let the functions J, $\boldsymbol{c}_{eq}$, and $\boldsymbol{c}_{ineq}$ be continuously differentiable, and the gradients of the equality and active inequality constraints at $\mathbf{z}^*$ be linearly independent (LICQ

This may affect the formulation of optimality conditions.

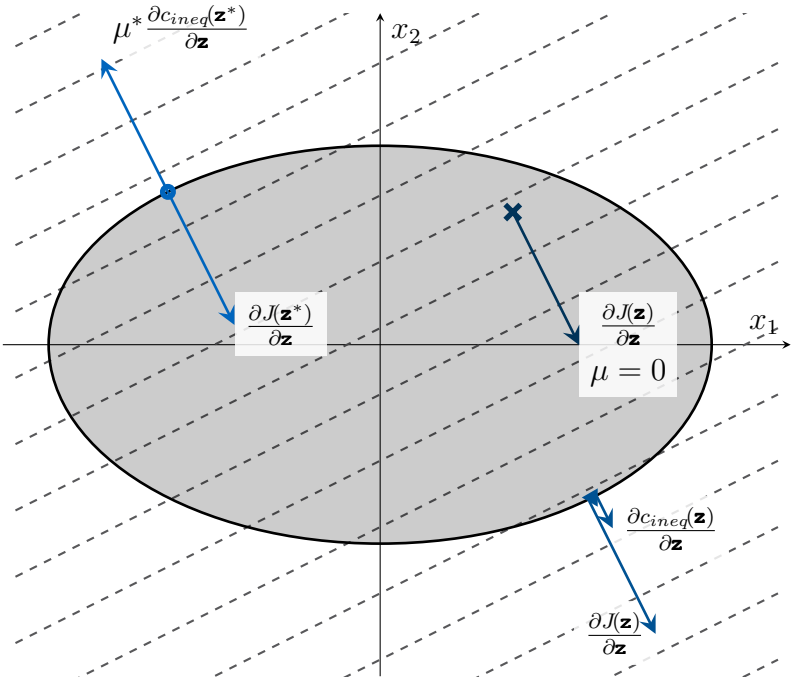


Figure 2.2: *Illustration of inequality constraint optimization: cost contour (- - -) and elliptical equality constraints (——), optimal point (⚬), non-KKT point (◀), and an arbitrary, feasible point (✖) with inactive constraint.*

regularity condition)

$$\mathbf{0} = \sum_{l=1}^{n_{c_{eq}}} \alpha_l \frac{\partial [\boldsymbol{c}_{eq}]_{(l)}}{\partial \mathbf{z}}\bigg|_{\mathbf{z}^*} + \sum_{m \in \mathcal{A}(\mathbf{z}^*)} \beta_m \frac{\partial [\boldsymbol{c}_{ineq}]_{(m)}}{\partial \mathbf{z}}\bigg|_{\mathbf{z}^*} \tag{2.40}$$
$$\Rightarrow \alpha_l, \beta_m = 0 \qquad \forall l = 1 \dots n_{\boldsymbol{c}_{eq}};\ m \in \mathcal{A}(\mathbf{z}^*)$$

Then there are multipliers $\boldsymbol{\lambda}^* \in \mathbb{R}^{n_{\boldsymbol{c}_{eq}}}$ and $\boldsymbol{\mu}^* \in \mathbb{R}^{n_{\boldsymbol{c}_{ineq}}}$ such that the following conditions hold

(i) Sign Condition:

$$\boldsymbol{\mu}^* \geq \mathbf{0} \tag{2.41}$$
$$\boldsymbol{\lambda}^* \neq \mathbf{0} \tag{2.42}$$

(ii) Optimality Condition

$$\frac{\partial J}{\partial \mathbf{z}}\bigg|_{\mathbf{z}^*} + \left(\frac{\partial \boldsymbol{c}_{eq}}{\partial \mathbf{z}}\bigg|_{\mathbf{z}^*}\right)^{\mathsf{T}} \boldsymbol{\lambda}^* + \left(\frac{\partial \boldsymbol{c}_{ineq}}{\partial \mathbf{z}}\bigg|_{\mathbf{z}^*}\right)^{\mathsf{T}} \boldsymbol{\mu}^* = \mathbf{0} \tag{2.43}$$

(iii) Complementary Condition[4]:

$$\boldsymbol{\mu}^{*\mathsf{T}} \boldsymbol{c}_{ineq}(\mathbf{z}^*) = 0 \tag{2.44}$$

(iv) Feasibility Condition:

$$\boldsymbol{c}_{eq}(\mathbf{z}^*) = \mathbf{0} \tag{2.45}$$
$$\boldsymbol{c}_{ineq}(\mathbf{z}^*) \leq \mathbf{0} \tag{2.46}$$

Above conditions are merely *necessary* for a local minimum in $\mathbf{z}^*$. Sufficient conditions exist, however since they usually incorporate second order derivatives of the cost function and constraints they are often impractical for real-life approaches. This is not a major drawback, since properly formulated problems only allow for local *minima* as the solution to above conditions. Thus second order considerations are often neglected with the argument, that the problem is "well-posed".

2.1.3.4 Box Constraints

An interesting simplification arises, if the inequality constraints take the form of simple box-constraints

$$\left(\begin{array}{cc} & \min\limits_{\mathbf{z}} J(\mathbf{z}) \\ \text{s.t.} & \mathbf{z} \leq \mathbf{z}_{ub} \\ & -\mathbf{z} \leq \mathbf{z}_{lb} \end{array} \right) \tag{2.47}$$

Then the constraint gradients are simply $+1$ for upper bound constraints , and -1 for lower bound constraints. The KKT conditions boil down to

$$0 = \left[\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}} \right]_{(j)} + \mu_j^* \cdot (\pm 1) \qquad\qquad j = 1, \ldots, n_{\mathbf{z}} \tag{2.48}$$

$$\Rightarrow \mu_j^* = \begin{cases} -\left[\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}} \right]_{(j)} & \text{if } [\mathbf{z}]_{(j)} = [\mathbf{z}_{ub}]_{(j)} \\ \left[\frac{\partial J(\mathbf{z})}{\partial \mathbf{z}} \right]_{(j)} & \text{if } [\mathbf{z}]_{(j)} = [\mathbf{z}_{lb}]_{(j)} \\ 0 & \text{otherwise} \end{cases} \tag{2.49}$$

Checking, if the resulting multipliers obey the KKT sign condition $\mu_j^* \geq 0$ can then be used in a simple active set strategy [Jat2006, Ch. 4], see algorithm 2.3. This algorithm can easily be combined with any of the algorithms for unconstrained optimization (section 2.1.2) in order to enable the consideration of box constraints.

For more complicated equality and inequality constraints, the workload to implement the necessary algorithmic details oneself is usually not justified, and it is commonly advised to use off-the-shelf algorithms [WP1997].

2.1.3.5 Remarks on Optimization

It is important to have a basic grasp of the concepts in use, in order to correctly apply off-the-shelf algorithms. The mathematical details presented so far enable the analyst to formulate an optimization problem, to implement solvers for simple tasks, and to interpret the results. However, in order to set up an efficient algorithm for the solution of complex, generic problems, the discussion of many more algorithmic details would be

[4] this condition implies, that for inactive constraints $[\boldsymbol{c}_{ineq}]_{(i)}(\mathbf{z}^*) < 0 \; i \notin \mathcal{A}(\mathbf{z}^*)$ it follows that $[\boldsymbol{\mu}^*]_{(i)} = 0$ and vice versa

Algorithm 2.3: Active set algorithm for box constraint model parameters
Initialization:

I. initialize the active set $\mathcal{A}(\mathbf{z}_0)$, using the initial guess $\mathbf{z}_0$ and the parameter bounds $\mathbf{z}_{lb}$, $\mathbf{z}_{ub}$. If some parameter elements violate their boundaries, the constraints have to be enforced.

Main Part:

II. use the cost function gradient to check the KKT sign condition (2.49), and drop those indices that violate it, i.e. drop an index j from the active set $\mathcal{A}(\mathbf{z}_i)$ if

$$\left[\frac{\partial J(\mathbf{z}_i)}{\partial \mathbf{z}}\right]_{(j)} > 0 \qquad \text{if } [\mathbf{z}_i]_{(j)} = [\mathbf{z}_{ub}]_{(j)}$$

$$\left[\frac{\partial J(\mathbf{z}_i)}{\partial \mathbf{z}}\right]_{(j)} < 0 \qquad \text{if } [\mathbf{z}_i]_{(j)} = [\mathbf{z}_{lb}]_{(j)}$$

III. compute a parameter update $[\boldsymbol{d}_i]_{(j)}$ (using one of the algorithms for unconstrained optimization) for the reduced problem involving the "free" model parameters $j \notin \mathcal{A}(\mathbf{z}_i)$ only; the fixed parameters stay at the respective bounds

$$[\mathbf{z}_{i+1}]_{(j)} = \begin{cases} [\mathbf{z}_i]_{(j)} + [\boldsymbol{d}_i]_{(j)} & \text{for } j \notin \mathcal{A}(\mathbf{z}_i) \\ [\mathbf{z}_i]_{(j)} & \text{for } j \in \mathcal{A}(\mathbf{z}_i) \end{cases}$$

IV. check if any of the updated parameters violate their bounds. If so, set them to the corresponding boundary value and include the respective index in the active set $\mathcal{A}(\mathbf{z}_{i+1})$ for the next iteration

necessary, which is beyond the scope of this thesis. Rather, the interested reader is referred to the respective literature, where those few works, which have been consulted for this work are [Ger2017, Ger2012, GMW1982, BA2010]. Since cost function optimization is an integral part of parameter estimation, the respective literature always contains chapters on this topic, too [WP1997, Sor1980, MIM1985, MK2016, Jat2006].

Several software implementations are available for different platforms, e.g.

- We Optimize Really Huge Problems (WORHP) [Ste2018],
- Sparse Nonlinear Optimizer (SNOPT) [GWMS2018], or
- Interior Point Optimizer (IPOPT) [KML+2015].

The availability of these mature implementations makes it unnecessary to implement them oneself, and further details regarding implementation aspects are omitted here.

2.1.4 Optimal Control

In the following section, the optimal control basics, necessary to understand its application to parameter estimation problems are illustrated. The goal is *not* to give a full

recapitulation of the necessary theoretical basics for optimal control in aircraft applications. For further details, the literature may be consulted [Bet2010, Ger2012, BA2010, Bit2017, Rie2017]. Moreover, the focus lies on those aspects, which are necessary to understand and use optimal control tools for parameter estimation purposes.

This entails certain simplifications compared to the most general optimal control problem formulation:

- The time grid is fixed, i.e. initial t_0 and final t_f are fixed and not subject to optimization
- In classical optimal control, different "phases" may arise, which may have their own time grid, state, input, and parameter vector as well as path constraints. These phases may even have different dynamic constraints, i.e. different underlying models. In general, these phases are linked together via specific constraints, e.g. to preserve continuity of states.

  In parameter estimation, those phases correspond to different maneuvers, which have been executed to excite different aspects of the model. They also have their own time grid, state, and input vector. However, the same model together with the same parameters are used for all maneuvers. Also, maneuvers are executed independently, rendering linking constraints unnecessary.
- Since inputs to the system were measured during experiments, in general they are fixed and not subject to optimization. However, for some special cases, where optimizable inputs will be considered, they are kept as part of the optimization vector in the following sections.
- Since parameter estimation always considers measured outputs, an output equation of the form

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t)\,, \boldsymbol{u}(t)\,, \boldsymbol{\theta}) \tag{2.50}$$

  is introduced. In the general considerations to follow, this output equation is often incorporated implicitly into constraints or cost functions.

2.1.4.1 Basics

In general, optimal control problems are used to find the optimal, continuous control trajectory $\boldsymbol{u}^*(t)$, and a set of optimal model parameters $\boldsymbol{\theta}^*$, together with the corresponding state trajectory $\boldsymbol{x}^*(t)$, which drive a performance index $J(\boldsymbol{u}(t)\,, \boldsymbol{x}(t)\,, \boldsymbol{\theta})$ to an extremal value. The optimization is to be subject to constraints in the form of system dynamics and possibly further, algebraic path constraints. Explicit constraints on the initial and final state may appear in the most general problem formulation, but are usually not necessary in the parameter estimation context. This yields the following

problem

$$\left(\begin{array}{ll} & \min\limits_{\boldsymbol{u}(t),\boldsymbol{\theta}} J(\boldsymbol{u}(t)\,,\boldsymbol{x}(t)\,,\boldsymbol{\theta}) \\ & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)\,,\boldsymbol{u}(t)\,,\boldsymbol{\theta}) \\ \text{s.t.} & \boldsymbol{c}_{eq}(\boldsymbol{x}(t)\,,\boldsymbol{u}(t)\,,\boldsymbol{\theta}) = \boldsymbol{0} \\ & \boldsymbol{c}_{ineq}(\boldsymbol{x}(t)\,,\boldsymbol{u}(t)\,,\boldsymbol{\theta}) \leq \boldsymbol{0} \end{array}\right) \tag{2.51}$$

The cost function formulations considered here are of *Bolza* type [Bit2017]

$$\begin{aligned} J(\boldsymbol{u}(t)\,,\boldsymbol{x}(t)\,,\boldsymbol{\theta}) = \phi(\boldsymbol{x}(t_f)\,,\boldsymbol{u}(t_f)\,,\boldsymbol{x}(t_0)\,,\boldsymbol{u}(t_0)\,,\boldsymbol{\theta}) \\ + \int_{t_0}^{t_f} L(\boldsymbol{x}(t)\,,\boldsymbol{u}(t)\,,\boldsymbol{\theta})\,dt \end{aligned} \tag{2.52}$$

where an explicit Mayer cost term on the initial and final conditions $\phi(\circ)$ is considered, along with a Lagrange cost term $L(\circ)$ that is integrated over time. Both cost terms can be converted in the respective other form as is e.g. illustrated in [Bit2017] and [Bet2010].

2.1.4.2 The Indirect Method

The indirect method, based on PONTRYAGIN's maximum principle [PBGM1962] is often too complicated to be applied in real-life problems. Nevertheless, for a complete discussion and in order to understand the differences compared to the direct method, the basic ideas are outlined here.

PONTRYAGIN's maximum principle may be used to define a set of necessary conditions for an optimal input trajectory $\boldsymbol{u}^*(t)$ and model parameter set $\boldsymbol{\theta}^*$. These may then be solved numerically. A central quantity in PONTRYAGIN's consideration is the Hamiltonian [Bit2017][5]

$$\mathcal{H}\Big(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda},\boldsymbol{\mu}_{eq},\boldsymbol{\mu}_{ineq}\Big) = L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\mu}_{eq}^{\mathsf{T}}\boldsymbol{c}_{eq}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\mu}_{ineq}^{\mathsf{T}}\boldsymbol{c}_{ineq}(\boldsymbol{x},\boldsymbol{u}) \tag{2.53}$$

In this formulation, the co-states $\boldsymbol{\lambda}(t)\,,\boldsymbol{\mu}_{eq}(t)$, and $\boldsymbol{\mu}_{ineq}(t)$ are all functions of time. PONTRYAGIN's maximum principle, together with the consequences of optimality criteria then state rules on how to determine the relationship between states, co-states and several partial derivatives of the Hamiltonian, in order to eventually obtain the optimal input trajectory $\boldsymbol{u}^*(t)$.

However the brute force application of this *indirect* method only leads to valid results for simple example cases. More often, it results in a two (or even multi-) point boundary value problem involving a Differential Algebraic Equation (DAE), which needs to be discretized and solved numerically [Bet2010, Ch. 4].

[5] For simplicity, only the explicit dependency of the terms on inputs and states is considered, since the problem can be easily transformed to incorporate parameters $\boldsymbol{\theta}$ and explicit dependencies on time t as additional states [Bit2017]

BETTS [Bet2010, Ch. 4.3] gives three main reasons, why the indirect approach is not generally used for the solution of practical optimal control problems

1. The computation of the necessary derivatives of the Hamiltonian is cumbersome at best, if not impossible for real world applications. Also, they are highly problem specific, i.e. have to be re-computed every time a new problem is formulated.

   The automatic differentiation approach of FALCON.m [RBG+2018], might alleviate this issue when applied to the Hamiltonian. However, the other disadvantages of the indirect method have so far discouraged from using it to this end.
2. Analogously to inactive inequality constraints and their corresponding Lagrange multipliers, the co-state $\boldsymbol{\mu}_{ineq}$ may be zero at times, when the corresponding path constraint is inactive, and non-zero otherwise. However, the number and sequence of the switches between these constrained and unconstrained arcs is hard to guess a priori, making the proper formulation of the resulting Non-Linear Programming (NLP) problem almost impossible.
3. Lastly, this method is not robust. Initial guesses for the co-states have to be provided, which is very unintuitive, since these are not physical quantities. What makes matters worse, is that the problem is quite sensitive to bad initial guesses in these co-states.

These are the main reasons, why the indirect method is seldom encountered in practice, which serve to motivate the use of a direct optimal control approach.

2.1.4.3 The Direct Method

In contrast to the indirect method, the direct method follows the following steps in solving an optimal control problem [Bet2010]

1. convert the infinite dimensional optimal control problem into a problem with a finite set of variables
2. solve the resulting non-linear programming problem
3. check that the solution to the finite dimensional problem approximates the infinite dimensional task reasonably well.

Since in parameter estimation, phases are in general independent of each other, the following remarks only focus on one phase, but can easily be extended to multiple phase problems.

Above mentioned "discretize then optimize" [Bet2010, p. 178] approach first discretizes the states and controls on a discrete (not necessarily uniform) time grid

$$\boldsymbol{x}(t_0 + k\Delta t_k) = \boldsymbol{x}_k \qquad \forall k = 0 \dots (N-1) \tag{2.54}$$

$$\boldsymbol{u}(t_0 + k\Delta t_k) = \boldsymbol{u}_k \qquad \forall k = 0 \dots (N-1) \tag{2.55}$$

Then the state and input vectors can be included in the optimization vector to different degrees, resulting in three commonly used problem formulations.

Single Shooting

The simplest approach for the transcription and solution of the infinite dimensional optimal control problem (2.51) is to ensure the dynamic constraint by propagating a set of initial states $\boldsymbol{x}_0$ forward in time, using the current values of the controls $\boldsymbol{u}_k$. This can be accomplished by any Runge-Kutta scheme, which transforms the continuous time system into a discretized version. For one integration step, this yields

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{\theta})\, d\tau \\ &\approx \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1} | \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) \end{aligned} \tag{2.56}$$

For two steps this would be

$$\begin{aligned} \boldsymbol{x}_{k+2} &= \boldsymbol{x}_k + \int_{t_k}^{t_{k+2}} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{\theta})\, d\tau \\ &\approx \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}, t_{k+2} | \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+2}, \boldsymbol{\theta}) \end{aligned} \tag{2.57}$$

Extending this scheme to the solution to an initial value problem over an arbitrary number of steps may then be formulated as

$$\boldsymbol{x}_k \approx \boldsymbol{\Phi}_{\boldsymbol{f}}(t_0 \dots t_k | \boldsymbol{x}_0, \boldsymbol{u}_0 \dots \boldsymbol{u}_k, \boldsymbol{\theta}) \qquad k = 1 \dots N \tag{2.58}$$

Where the right hand side of above equation is to be interpreted as the numerical solution of an initial value problem from time t_0 to time t_k, using the initial condition $\boldsymbol{x}_0$, model parameters $\boldsymbol{\theta}$, and input sequence $\boldsymbol{u}_0 \dots \boldsymbol{u}_k$.

No restriction on explicit integration methods is strictly necessary. However, since implicit methods necessitate the iterative solution of a non-linear root finding problem for every time step, it is usually sensible to choose explicit methods in the shooting cases in order to reduce computation times.

The vector of optimization variables, which the NLP solver may adjust in this case is

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \\ \boldsymbol{\theta} \end{bmatrix} \tag{2.59}$$

The path constraints appear as algebraic constraints, evaluated at the discretization points t_k. Eventually the discretized optimization problem becomes

$$\begin{pmatrix} \min\limits_{\mathbf{z}} J(\boldsymbol{x}_0 \dots \boldsymbol{x}_{N-1}, \boldsymbol{u}_0 \dots \boldsymbol{u}_{N-1}, \boldsymbol{\theta}) & & \\ \qquad\qquad \boldsymbol{x}_k = \boldsymbol{\Phi}_{\boldsymbol{f}}(t_0 \dots t_k | \boldsymbol{x}_0, \boldsymbol{u}_0 \dots \boldsymbol{u}_k, \boldsymbol{\theta}) & k = 1, \dots, N-1 \\ \text{s.t.} \quad \boldsymbol{c}_{eq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) = \mathbf{0} & k = 0, \dots, N-1 \\ \boldsymbol{c}_{ineq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \leq \mathbf{0} & k = 0, \dots, N-1 \end{pmatrix} \tag{2.60}$$

where the first constraint regarding the integration of the system equations is not explicitly enforced in the NLP problem, but directly evaluated, i.e. the intermediate $\boldsymbol{x}_k$ $k = 0, \ldots, N-1$ are computed based on the numerical integration scheme.
The solution strategy is then as follows [Bit2017]: At every NLP iteration

1. extract the initial condition, parameters and control history from the optimization vector
2. solve the arising initial value problem using any numeric integration scheme
3. evaluate the constraints at the discretization points $t_k \quad k = 0 \ldots N-1$, as well as the cost function
4. determine the gradients of the cost function and constraints; This can be achieved analytically, using the chain rule and sensitivity equations (see section 3.3.3). Alternatively, gradient information can be obtained numerically using finite differences [Bit2017, Fis2011].
5. provide cost function and constraint values, together with gradient information to the NLP algorithm; at the next iteration, start from 1.

This approach is generally termed *single shooting*, due to the parallels with shooting a cannonball at a specified target [Bet2010]: after the initial conditions – the cannons elevation and azimuth – and the parameters – the cannonball's shape and weight – are fixed, nothing can be done about its trajectory anymore (in this example, no control is possible after the shot). This thought experiment also illustrates one of the largest shortcomings of the approach: small changes in the initial conditions, or controls applied early in the integration, may have a significant effect on the trajectory. Since path constraints at later points in time depend on the initial condition through the recursive numerical solution of the differential equation, this dependency may become *very* non-linear [Bet2010], and thus pose serious problems for the NLP solver.

Nevertheless, single shooting has become the quasi standard for parameter estimation in the aircraft context [Jat2006, MK2016]. This is for several reasons: In classical applications, no path constraints and no constraints on the final state are considered, i.e. the above mentioned very non-linear dependencies simply do not exist for most parameter estimation problems. Next, the controls are usually considered fixed, since they were measured during the experiment. Thus, only the model parameters $\boldsymbol{\theta}$ shape the trajectory throughout the complete time history. This dependency also becomes more non-linear as the integration time progresses, however, together with the close to quadratic maximum likelihood cost function (as will be shown in section 2.2.2) the resulting problem is often treatable using a Newton-based solver.

The discussion of the details of this classical solution process in aircraft parameter estimation is deferred to chapter 4, when all the necessary basics have been laid out.

Multiple Shooting

One idea to overcome the problematic sensitivity on the initial conditions in the single shooting case, is to break the problem into smaller integration intervals, i.e. to introduce more "initial conditions" [Bet2010, Ch. 3.4] [Bit2017, Ch. 3.2]. Staying in above picture this would mean to "catch" the cannonball mid-flight and to perform a new shot. Then early changes in the trajectory only influence shorter segments, and the resulting dependencies are less non-linear. However, it has to be made sure that in the end, the starting and ending points of the segments are consistent.

This may be formulated more mathematically. If M multiple shooting segments are considered, the time grid is partitioned into M (not necessarily equal) segments

$$t_0 = t_{k_1} < t_{k_2} < \cdots < t_{k_{M+1}} = t_f \tag{2.61}$$

$$0 = k_1 < k_2 < \cdots < k_{M+1} = N - 1 \tag{2.62}$$

i.e. the partitioning takes place at time instants t_{k_l}, with the respective separation indices k_l $l = 1 \dots M$. The initial condition for every segment $\boldsymbol{x}_{k_l}$ $l = 1 \dots M$ are explicitly included in the optimization vector

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{x}_{k_1} \\ \vdots \\ \boldsymbol{x}_{k_M} \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \\ \boldsymbol{\theta} \end{bmatrix} \tag{2.63}$$

In order to keep the states continuous over segment borders, additional equality constraints have to be introduced for every segment [Bit2017, Ch. 3.2] [Bet2010, Ch. 3.4.]

$$\boldsymbol{\zeta}_{MS}\Big(\boldsymbol{x}_{k_l}, \boldsymbol{x}_{k_{l+1}}, \boldsymbol{u}_{k_l} \dots \boldsymbol{u}_{k_{l+1}}, \boldsymbol{\theta}\Big) = \boldsymbol{x}_{k_{l+1}} - \boldsymbol{\Phi}_{\boldsymbol{f}}\Big(t_{k_l} \dots t_{k_{l+1}} \Big| \boldsymbol{x}_{k_l}, \boldsymbol{u}_{k_l} \dots \boldsymbol{u}_{k_{l+1}}, \boldsymbol{\theta}\Big) = \mathbf{0} \qquad l = 1 \dots M-1 \tag{2.64}$$

They enforce equality between the final integration step of the previous segment and the initial condition at the current segment. These *defect* constraints may be violated in the course of the optimization; however, they need to be fulfilled in the end for a valid solution to the overall problem.

The multiple shooting problem to be solved is then

$$\left(\begin{array}{rll} & \min\limits_{\mathbf{z}} J(\boldsymbol{x}_0 \dots \boldsymbol{x}_{N-1}, \boldsymbol{u}_0 \dots \boldsymbol{u}_{N-1}, \boldsymbol{\theta}) & \\ & \boldsymbol{\Phi}_{\boldsymbol{f}}(t_{k_l} \dots t_k | \boldsymbol{x}_{k_l}, \boldsymbol{u}_{k_l} \dots \boldsymbol{u}_k, \boldsymbol{\theta}) = \boldsymbol{x}_k & k = k_l + 1 \dots k_{l+1} \\ & & l = 1 \dots M \\ \text{s.t.} & \boldsymbol{c}_{eq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) = \mathbf{0} & k = 0 \dots N-1 \\ & \boldsymbol{c}_{ineq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \leq \mathbf{0} & k = 0 \dots N-1 \\ & \boldsymbol{\zeta}_{MS}\Big(\boldsymbol{x}_{k_l}, \boldsymbol{x}_{k_{l+1}}, \boldsymbol{u}_{k_l} \dots \boldsymbol{u}_{k_{l+1}}, \boldsymbol{\theta}\Big) = \mathbf{0} & l = 1 \dots M-1 \end{array} \right) \tag{2.65}$$

Again, the first constraint is not directly enforced within the NLP solver, but explicitly computed as part of the solution strategy. This strategy is very similar to that of the single shooting case: the only difference is that in order to obtain the complete state trajectory, one initial value problem is solved per segment and continuity is enforced via additional equality constraints.

In addition to alleviating the problem with overly sensitive constraints, the multiple shooting approach also has favorable consequences regarding the problem sparsity: as the number of multiple shooting segments grows, the problem Jacobian becomes sparser. Not only is this practical for the NLP solver, since decoupling of the optimization parameters in general facilitates the problem solution; also there exist very efficient solution algorithms for large, sparse optimization problems [Bet2010, Ch. 3, Ch. 4].

Full Discretization

The idea of multiple shooting can be driven to an extreme, by taking as many shooting segments, as there are discretization points [Bit2017, Bet2010]. The optimization vector is then

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_{N-1} \\ \boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{u}_{N-1} \\ \boldsymbol{\theta} \end{bmatrix} \tag{2.66}$$

and contains all states and all inputs at all time instants, together with the model parameters. The defect constraints become quite simple, since they only link two time instants

$$\begin{aligned} \boldsymbol{\zeta}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) = \boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1} | \boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) = \mathbf{0} \\ k = 0 \dots N-2 \end{aligned} \tag{2.67}$$

In the resulting optimization problem, the dynamic constraint is fully translated into defect constraints and does not appear explicitly anymore.

$$\left(\begin{aligned} & \min_{\mathbf{z}} J(\boldsymbol{x}_0 \dots \boldsymbol{x}_{N-1}, \boldsymbol{u}_0 \dots \boldsymbol{u}_{N-1}, \boldsymbol{\theta}) \\ & \boldsymbol{c}_{eq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) = \mathbf{0} \quad k = 0 \dots N-1 \\ \text{s.t.} \quad & \boldsymbol{c}_{ineq}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \leq \mathbf{0} \quad k = 0 \dots N-1 \\ & \boldsymbol{\zeta}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) = \mathbf{0} \quad k = 0 \dots N-2 \end{aligned} \right) \tag{2.68}$$

The complete problem can be presented to an NLP solver as is, no explicit treatment of the dynamic constraints is necessary anymore.

Furthermore, the inclusion of implicit integration methods is easily possible in the defect constraints $\boldsymbol{\zeta}_k$: from the view of the NLP solver, the integration scheme is merely an equality constraint linking $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ and $\boldsymbol{x}_{k+1}$, $\boldsymbol{u}_{k+1}$. Thus, by fulfilling this constraint the lower level root-finding problem, which is necessary for implicit schemes, is automatically solved by the NLP solver [Bit2017].

Obviously, with many time instants, and possibly large state and input vectors, the problem size grows very quickly. However, as the problem size increases, it becomes more and more sparse. Thus when using specially tailored sparse optimization algorithms, the problem remains manageable.

Remarks on the Direct Method
The application of the direct method overcomes most of the problems that were listed for the indirect method in section 2.1.4.2 [Bet2010, Ch. 4.14] [Bit2017, Ch. 3.2.4]

1. no derivatives of the Hamiltonian need to be formulated, i.e. the user does not need to be familiar with PONTRYAGIN's maximum principle.
2. gradient information is easy to supply, either analytically (via sensitivity equations and the chain rule) or via (sparse) numeric finite differences. In the former case, it might even be possible to apply automatic differentiation algorithms, as is e.g. done in FALCON.m [RBG+2018]
3. no initial information about constrained arcs is necessary if path inequalities are present, since the NLP solver's active set strategy can determine them automatically through the corresponding multipliers
4. initial information about the Lagrange multipliers is easier to come by, as opposed to the necessary initial guess for the unphysical co-states in the indirect approach. For further information, see also the discussion on the relation between co-states and Lagrange multipliers in [Bet2010, Ch. 4.2.] [Bit2017, Ch. 3.2.5.].

2.1.4.4 Further Aspects in Optimal Control

For the sake of completeness, further relevant aspects of optimal control are briefly listed. However, they are not considered in detail, as they are not relevant for the further understanding of this thesis. Additional information can be obtained in the literature [Bet2010, Ger2017, Ger2018, Ger2012, Bit2017, Rie2017, BA2010].

Different Control and State Parameterization The time grid of the state and control discretization need not necessarily be the same. It just has to be made sure, that all necessary information for formulating the state integration is available. If the time grids do not match between states and controls, commonly some sort of interpolation is applied, to obtain the control values at the state discretization steps [Bit2017]. To achieve this, linear interpolation or more advanced approaches like B-splines can be

used [Ger2012]. This functionality is integrated in FALCON.m [RBG+2018] but will only be used sparingly when considering auxiliary inputs to the estimation problem.

This idea can be taken further to even use different discretization intervals for subsets of states, e.g. because one knows, that a certain subset has considerably "slower" dynamic than another subset. BOTTASSO then shows, that different integration methods can be used for these subsets, by combining single and multiple shooting [BM2009, BLM2009a]. In a similar manner, BITTNER combines multiple shooting with full discretization [Bit2017].

Choice of Integration Method It is intuitive, that the accuracy of the integration scheme in use, together with the step size selection significantly influence the quality of the solution. Here, a multitude of choices is possible: several explicit and implicit Runge-Kutta schemes are available, which differ in their truncation errors and the number of intermediate stages. The possibility of including the intermediate integration stages as additional optimization variables even increases the number of possibilities of formulating the same problem [Bet2010, Bit2017].

In this work, a restriction is made to two major methods: in the full discretization approach, a second order trapezoidal scheme is used; whereas for applications of the single shooting method, a Runge-Kutta scheme (usually fourth order) is employed.

Independent Variables All of the above methods may also be formulated for independent variables other than time, in order to e.g. solve partial differential equations. However, since the main focus of this work is on parameter estimation from flight test data, other independent variables are not discussed [Bet2010].

Path Constraints The proper inclusion of path constraints in an optimal control problem is not as easy, as it may seem when inspecting the foregoing sections. In the direct method using full discretization, they are formulated as equality and inequality constraints on state and control variables. However, no guarantee can be given, that the trajectory does not violate these constraints between the discretization points, which necessitates further analysis of the problem (in order to show that it cannot happen for the formulation at hand) or the result (in order to show that a violation did not happen for that specific solution) [Bit2017].

Scaling Considerations Scaling is a crucial point in every optimization problem. It is easy to construct simple problems, which become nearly impossible to solve, if the optimization variables are scaled disadvantageously. Some main aspects here are the numerical condition of the Hessian (or its approximation), and gradient, which can seriously inhibit the solution process. Thus it is usually favorable, to scale the involved parameters to a range between 0 and 1 as far as possible.

Most off-the-shelf optimization algorithms incorporate some sort of automatic scaling approach. However it can be more helpful to scale variables based on the analyst's knowledge of the problem characteristics, rather than trusting some automatic approach that cannot know the physics involved.

Sparsity Considerations Problems with several $100\,000$ optimization variables can nowadays be solved on a consumer PC. However, this is only possible by making use of the sparsity patterns of the constraints, cost function and their derivatives. Exploiting this sparsity considerably reduces computational load. This is why at every step of the process, this aspect should be taken into account in order to make the problem as "sparse" as possible [Bit2017, Bet2010].

2.2 Statistical Estimation

The following sections will discuss the basics of statistical parameter estimation. After introducing some general concepts, and discussing possible estimator characteristics, three parameter estimation approaches will be discussed in detail. Namely they are maximum likelihood estimation, Bayesian estimation, and least-squares estimation. The chapter is concluded by some remarks on estimated statistical characteristics of parameter estimates.

2.2.1 Basics in Estimation Theory

The probability theoretic basics, which are necessary to understand the following chapter, are summed up in Appendix A.4. There, a notational distinction between a random variable x and its realization x was made in order to illustrate the conceptual differences. However, this will be dropped, since it is usually clear from the context, if a random variable, or its realization is discussed.

Also, probability densities in Appendix A.4 have a superscript to indicate the random variable $p_{\mathrm{x}}(x)$ in question. However, if it is clear from the context, which random variable is meant, this will be omitted $p(x)$.

2.2.1.1 Estimator Definition

The basics of modern estimation theory and statistical inference were laid by RONALD AYLMER FISHER in the early 20th century [Fis1912, Fis1922, Fis1925]. The following presentation of his main results follow a similar structure as in [MK2016, Ch. 4, Ch. 6], [Jat2006, Ch. 4], and [HGH+2017, Ch. 4].

The overall problem to be considered is to infer the numerical values of a parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^{n_\theta}$, based on measurements $\boldsymbol{z} \in \mathbb{R}^{n_z}$ that are influenced by inputs $\boldsymbol{u} \in \mathbb{R}^{n_u}$ (which are assumed to be perfectly known) and distorted by some random component $\boldsymbol{\omega} \in \mathbb{R}^{n_\omega}$

$$\mathbf{z} = \mathbf{z}(\boldsymbol{\theta}, \boldsymbol{u}, \boldsymbol{\omega}) \tag{2.69}$$

To do so, an *estimator* is employed, which computes an *estimate* $\hat{\boldsymbol{\theta}}$ using the inputs $\boldsymbol{u}_k$ and the realizations of the measurements $\boldsymbol{z}_k$ for $k = 0, \ldots, N-1$. This estimator can be any function that maps $\mathbf{Z} = [\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{N-1}]$ and $\mathbf{U} = [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}]$ to the set $\boldsymbol{\Theta}$ [MIM1985]

$$\hat{\boldsymbol{\theta}} : \mathbb{R}^{n_z \times N} \times \mathbb{R}^{n_u \times N} \to \boldsymbol{\Theta} \tag{2.70}$$

The estimate $\hat{\boldsymbol{\theta}}$ arises, when a specific set of samples is processed

$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\mathbf{Z}, \mathbf{U}) \tag{2.71}$$

Example 2.3: Sample average as estimator for the mean
if z is a random number with mean μ, then the sample average over N samples (see also (A.158))

$$\bar{z}_N = \frac{1}{N}\sum_{k=0}^{N-1} z_k$$

can be considered an estimator for the true mean μ.

Example 2.4: Sample variance as estimator for the variance
if z is a random number with mean μ and variance σ^2, then the sample variance over N samples (see also (A.159))

$$s^2_{z,N} = \frac{1}{N}\sum_{k=0}^{N-1} (z_k - \mu)^2$$

can be considered an estimator for the true variance σ^2. If the true mean μ is unknown, the following provides an unbiased estimate of the variance, using the sample average $\bar{z}_N$ as estimator for the mean [HGH+2017, Ch. 4]

$$s^2_{z,N} = \frac{1}{N-1}\sum_{k=0}^{N-1} (z_k - \bar{z}_N)^2$$

Explicit dependency on the inputs $\mathbf{U}$ will usually be dropped for ease of notation.

Two widely used estimators are illustrated in examples 2.3 and 2.4. These are examples for linear (sample average) and non-linear (sample variance) algebraic relations between the measurements and the estimate. However, other mathematical dependencies are possible: most of the estimators employed in this work will be solutions to optimization problems.

It is important to note, that the estimates $\hat{\boldsymbol{\theta}}$ are a function of the random variable $\mathbf{Z}$ and thus are themselves a random variable [MK2016, Ch. 4]. Therefore they can be characterized by their statistical moments, where usually mean $\mathrm{E}\left[\hat{\boldsymbol{\theta}}\right]$ and Covariance $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]$ play the most important role. This holds true, even if the underlying true parameters $\boldsymbol{\theta}$ are assumed to be deterministic.

In order to meaningfully apply estimation algorithms based on above definition, some more details need to be specified. MORELLI lists four aspects, that need to be well defined [MK2016, Ch. 4]:

1. a model structure, relating the inputs $\boldsymbol{u}$ and parameters $\boldsymbol{\theta}$ to the model outputs
2. a mathematical model for the measurement process, i.e. relating the random component $\boldsymbol{\omega}$ to the measurements $\mathbf{z}$
3. a set of realizations of the measurement variables $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{N-1}$
4. assumptions about the random component $\boldsymbol{\omega}$, and possibly prior knowledge on the parameters $\boldsymbol{\theta}$; this is often provided in the form of a pdf in both cases

2.2.1.2 Estimator Properties

An estimator can have several properties, the definition of which are given next, for more details see [MK2016, Ch. 4], [Sor1980], [Jat2015, Ch. 4] or [MIM1985]. For asymptotic results, where the number of samples grows large, stochastic convergence concepts are used, details of which can be found in appendix A.4.8.

linear An estimator is called linear, if the estimates $\hat{\boldsymbol{\theta}}$ are affine in the measurements.

unbiased An estimator is called unbiased, if the expected value of $\hat{\boldsymbol{\theta}}$ is equal to the expected value of the true parameters $\boldsymbol{\theta}$, independent of the sample size N

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}\right] = \mathrm{E}[\boldsymbol{\theta}] \qquad \forall N, \boldsymbol{\theta} \tag{2.72}$$

In contrast, if it holds that

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}\right] = \mathrm{E}[\boldsymbol{\theta}] + \boldsymbol{b}(\boldsymbol{\theta}) \tag{2.73}$$

the estimator is called biased with bias $\boldsymbol{b}(\boldsymbol{\theta})$.

The bias is a systematic error of an estimator, which must not be confused with random fluctuations when processing different sets of measurements. For a single estimation result there might well be a significant difference between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$. However, if this estimation was performed several times (using several data sets, i.e. several different realizations of the random component), the mean over all estimates for an *unbiased* estimator would eventually be equal to the true parameter mean. The random components would average out.

This cannot be achieved for a biased estimator, which is in general undesirable.

minimum mean square error An estimator is called minimum mean square error (MSE) estimator, if it minimizes the MSE defined as

$$MSE = \mathrm{E}\left[\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)^{\mathsf{T}}\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right)\right] \tag{2.74}$$

In general, the MSE includes influences of the parameter variances (random error) as well as the parameter biases (estimator inherent, systematic error) [MK2016, Ch. 4]. For unbiased estimators with $\boldsymbol{b}(\boldsymbol{\theta}) = \mathbf{0}$ the MSE reduces to the trace of the parameter error covariance $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]$, i.e. the sum of the individual parameter variances.

Obviously it is desirable to use unbiased estimators, with small resulting parameter variances, thus yielding a small MSE. However, both aspects should be taken into account when evaluating different estimation algorithms: sometimes it may be favorable, to trade a small bias for a major decrease in parameter variances. This is especially true in cases, where an estimator might be unbiased, but yields estimates that are unusable in practice, see [MIM1985, Sec. 4.2.1] for an example.

efficient The obtainable covariance of the parameter estimates $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]$ for any unbiased estimator is bounded from below via the so called *Cramér-Rao* inequality [GP1977, Sor1980, WP1997]

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] \geq \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1} \tag{2.75}$$

where $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})$ is the *Fisher information matrix* defined as

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) = \mathrm{E}\left[\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\mathsf{T}}\right] \tag{2.76}$$

See Appendix C.2 for a derivation of the Cramér-Rao inequality. The inequality sign in the definition of the Cramér-Rao inequality is to be understood in the sense that the difference $\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] - \boldsymbol{\mathcal{F}}^{-1}\right)$ is positive semi-definite. Alternatively, the Fisher information matrix can be expressed as [GP1977, Sor1980, WP1997]

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) = -\mathrm{E}\left[\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \tag{2.77}$$

see Appendix C.1 for details[6].

Although the expression $p(\mathbf{Z}|\boldsymbol{\theta})$ plays a very important role in maximum likelihood estimation (as will be illustrated in section 2.2.2) the Cramér-Rao bound in this form provides a result that is valid way beyond that: it is a theoretical minimum for *any* kind of unbiased estimator, which cannot be undercut.

The Cramér-Rao inequality is especially useful, if an estimator proves to actually attain this lower bound

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1} \tag{2.78}$$

An estimator for which equation (2.78) holds is called *efficient*: It makes maximum use of the available information by providing the lowest possible covariance $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]$ [Sor1980, Ch. 3]. Lower covariances can only be obtained by introducing an estimation bias $\boldsymbol{b}(\boldsymbol{\theta})$.

If an estimator can be shown to be efficient, and an approximation for the Fisher information matrix was available, the parameter covariances can be estimated based on equation (2.78). This is in general an important aspect of the validation step, since

[6] The appearance of the natural logarithm in the Fisher information matrix might seem quite random. However, when carefully studying the proofs in Appendix C, one can realize that its derivative

$$\frac{\partial \ln p(x)}{\partial x} = \frac{1}{p(x)}\frac{\partial p(x)}{\partial x} \qquad \Leftrightarrow \qquad \frac{\partial p(x)}{\partial x} = p(x)\,\frac{\partial \ln p(x)}{\partial x}$$

is used several times to translate the integral over a derivative $\frac{\partial p(x)}{\partial x}$ into an expectation of $\frac{\partial \ln p(x)}{\partial x}$. Furthermore, in section 2.2.2 it will be shown, that this logarithm makes it possible to easily compute derivatives of the cost function analytically. These analytic expressions for gradient and Hessian greatly improve the convergence properties of the necessary optimization.

in addition to the parameter values, also their uncertainty can be quantified via their covariance matrix.

Efficiency for finite data sets is often difficult to prove, whereas results for $N \to \infty$ can be obtained more easily. An estimator for which the limit of equation (2.78) holds

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) \cdot \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] \xrightarrow[N\to\infty]{} \mathbf{I}_{n_\theta} \tag{2.79}$$

is called *asymptotically efficient*. For real-life applications, an infinite amount of data is impractical, however in aircraft system identification it is a valid assumption that the data sets are "large enough" to justify above approximation.

consistent An estimator is called *consistent* if it holds for growing N [CS2011, Ch. 4.4]

$$\hat{\boldsymbol{\theta}}(N) \xrightarrow[N\to\infty]{Pr} \boldsymbol{\theta} \tag{2.80}$$

That is, for increasing sample size, the estimate converges in probability to the true parameter value (see appendix A.4.8 for details on statistical convergence). This concept is related to the concept of an estimation bias: a consistent estimator is also an unbiased estimator. However, consistency only covers the limit case $N \to \infty$, whereas unbiasedness is valid independent of the sample size N.

normal An estimator is called normal, if estimates obtained from different sets of samples are clustered around the true value with a normal distribution, i.e. if it holds that

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{A}) \tag{2.81}$$

with covariance matrix $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] = \mathbf{A}$. *Asymptotic Normality* is given, if this behavior is reached in the limit case [CS2011, Ch. 4.4]

$$\hat{\boldsymbol{\theta}} \xrightarrow[N\to\infty]{D} \boldsymbol{r} \sim \mathcal{N}(\boldsymbol{\theta}, \mathbf{A}) \tag{2.82}$$

In summary, an unbiased, minimum MSE/variance, (asymptotically) efficient and consistent estimator is desirable. However, one should evaluate on a case to case basis, if one or more of those properties might have to be sacrificed in order to improve the overall result. As was mentioned before, one common trade-off is that between parameter variance an estimator bias.

2.2.2 Maximum Likelihood Estimation

Maximum likelihood estimation is definitely the most widely used time-domain approach for parametric identification of dynamic aircraft systems. Its basics will be illustrated in this chapter.

The probability of obtaining the measurements $\mathbf{Z}$ given a set of parameters $\boldsymbol{\theta}$ can be described via the so called *Likelihood* function [MK2016, Ch. 4]

$$\mathbb{L}(\mathbf{Z}, \boldsymbol{\theta}) = p(\mathbf{Z}|\boldsymbol{\theta}) \tag{2.83}$$

which is the probability density function of the measurements $\mathbf{Z}$ given certain values of the parameters $\boldsymbol{\theta}$. In many cases, the likelihood function can be stated to the extent that it belongs to a certain family of distributions, which is parameterized by the unknown parameter vector $\boldsymbol{\theta}$. The final task is then to "best" determine those parameters. Then, $\mathbb{L}(\mathbf{Z}, \boldsymbol{\theta})$ describes the combined effect of random components of the process under investigation (unmeasurable disturbances) together with the uncertainties introduced by measuring the quantities $\mathbf{Z}$ (measurement noise).

The basic idea of maximum likelihood estimation is now, to find those parameter values, that *maximize* this conditional probability, since "the probability of an event occurring is directly proportional to the value of its probability density function" [Sim2006, p. 301]. For a given set of measurements $\mathbf{Z}$ the parameters are chosen such that those $\mathbf{Z}$ are the *most likely* to occur [Sor1980, Ch. 5], i.e. are the most plausible.

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{Z}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \mathbb{L}(\mathbf{Z}, \boldsymbol{\theta}) \tag{2.84}$$

The maximum of a probability density is commonly called a "mode". If above probability density has more than one peak, i.e. is *multi-modal*, the uniqueness of the estimate is not given anymore. However, in this work, most probability densities are considered to be *uni-modal*.

The maximum likelihood approach must not be confused with maximizing $p(\boldsymbol{\theta}|\mathbf{Z})$, which would correspond to choosing the most plausible $\hat{\boldsymbol{\theta}}$ given the measurements $\mathbf{Z}$, as is done in Bayesian estimation, which will be detailed in section 2.2.3. For this latter formulation it is necessary that the prior distribution $p(\boldsymbol{\theta})$ is known, which is not the case for a maximum likelihood estimator.

Since $\mathbf{Z}$ is made up of N separate measurements, Bayes' rule (A.108) and the definition of conditional probability densities (A.102) can be applied repeatedly to arrive at [MK2016, Ch. 6.1]

$$\begin{aligned}
\mathbb{L}(\mathbf{Z}, \boldsymbol{\theta}) = p(\mathbf{Z}|\boldsymbol{\theta}) &= p(\boldsymbol{z}_{N-1}|\mathbf{Z}_{N-2}, \boldsymbol{\theta})\, p(\mathbf{Z}_{N-2}|\boldsymbol{\theta}) \\
&= p(\boldsymbol{z}_{N-1}|\mathbf{Z}_{N-2}, \boldsymbol{\theta})\, p(\boldsymbol{z}_{N-2}|\mathbf{Z}_{N-3}, \boldsymbol{\theta})\, p(\mathbf{Z}_{N-3}|\boldsymbol{\theta}) = \dots \\
&= \prod_{k=0}^{N-1} p(\boldsymbol{z}_k|\mathbf{Z}_{k-1}, \boldsymbol{\theta})
\end{aligned} \tag{2.85}$$

with the abbreviations

$$\begin{aligned}
\mathbf{Z}_k &= \begin{bmatrix} \boldsymbol{z}_0 & \dots & \boldsymbol{z}_k \end{bmatrix} \\
\mathbf{Z}_{N-1} &= \mathbf{Z} \\
p(\boldsymbol{z}_0|\mathbf{Z}_{-1}, \boldsymbol{\theta}) &= p(\boldsymbol{z}_0|\boldsymbol{\theta})
\end{aligned}$$

MAINE and ILIFF call the following the *Markov Criterion* [MIM1985]

$$p(\boldsymbol{z}_k|\mathbf{Z}_{k-1},\boldsymbol{\theta}) = p(\boldsymbol{z}_k|\boldsymbol{\theta}) \tag{2.86}$$

that is, if the conditional probability of $\boldsymbol{z}_k$ is independent of the previous measurements $\mathbf{Z}_{k-1}$, then the likelihood function can be expressed as

$$\mathbb{L}(\mathbf{Z},\boldsymbol{\theta}) = \prod_{k=0}^{N-1} p(\boldsymbol{z}_k|\boldsymbol{\theta}) \tag{2.87}$$

In other words, the random components in the samples are required to be independent and identically distributed (i.i.d.).

Instead of considering a maximization of equation (2.87) directly, commonly its negative natural logarithm is used as cost function in an optimization approach. This transformation does not change the location of local minima, since the natural logarithm is strictly monotonically increasing [MK2016, Ch. 6.1]. Merely the numeric value of the cost function is changed. Also the multiplication with -1 leads to a minimization instead of a maximization task, which is more common in optimization.

$$J_{ML}(\boldsymbol{\theta}) = -\ln(\mathbb{L}(\mathbf{Z},\boldsymbol{\theta})) = -\sum_{k=0}^{N-1} \ln(p(\boldsymbol{z}_k|\boldsymbol{\theta})) \tag{2.88}$$

The explicit dependency of J_{ML} on $\mathbf{Z}$ is dropped, for ease of notation. The first, obvious advantage of this transformation is that the logarithm of a product can be transformed into a sum of logarithms, which will be easier to handle when computing derivatives as the product rule does not have to be applied. Secondly, the probability density functions considered here are of exponential nature, which will cancel with the logarithm.

A necessary condition for a minimum of the cost function is then (see section 2.1.2)

$$\left.\frac{\partial J_{ML}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{ML}} = \mathbf{0} \tag{2.89}$$

which is often called the *likelihood equation* [CJ2012, Ch. 2].

2.2.2.1 Properties of Maximum Likelihood Estimates

First, a few interesting properties of a minimum of equation (2.88) are investigated: The second order necessary condition for a local minimum involves the Hessian of equation (2.88)

$$\left.\frac{\partial^2 J_{ML}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}_{ML}} = -\sum_{k=0}^{N-1} \left.\frac{\partial^2 \ln(p(\boldsymbol{z}_k|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}_{ML}} \tag{2.90}$$

Similar to the approach in [Jat2006, App. D], the strong law of Large Numbers (A.162) may be used to arrive at

$$\begin{aligned} -N\frac{1}{N}\sum_{k=0}^{N-1} \left.\frac{\partial^2 \ln(p(\boldsymbol{z}_k|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}_{ML}} &\xrightarrow[a.s.]{N\to\infty} N\mathrm{E}\left[-\left.\frac{\partial^2 \ln(p(\boldsymbol{z}_k|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^2}\right|_{\hat{\boldsymbol{\theta}}_{ML}}\right] = \\ &= N\boldsymbol{\mathcal{F}}_k\left(\hat{\boldsymbol{\theta}}_{ML}\right) = \boldsymbol{\mathcal{F}}\left(\hat{\boldsymbol{\theta}}_{ML}\right) \end{aligned} \tag{2.91}$$

Combining the last two equations one can see that for a sufficiently large N, the Hessian of J_{ML} can be considered an approximation for the Fisher information matrix $\boldsymbol{\mathcal{F}}\left(\hat{\boldsymbol{\theta}}_{ML}\right)$.

Furthermore, a maximum likelihood estimator exhibits favorable properties under the following hypotheses [WP1997, Ch. 3.3.3] [CS2011, Ch. 4.4]

- the model structure $\mathcal{M}(\boldsymbol{\theta})$ for data generation and parameter estimation is perfectly known (no so-called "characterization error")
- the model structure is *globally identifiable* under the experimental conditions, i.e.

$$\mathcal{M}(\boldsymbol{\theta}_1) = \mathcal{M}(\boldsymbol{\theta}_2) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \tag{2.92}$$

  If the model behaves in the same way for two parameters, this implies that the parameters are the same.
- the perturbation and noise that arise in the description of the measurement process can be modelled as i.i.d. random variable, possibly passed through some shaping filter.
- the second derivative of the log-likelihood function w.r.t. $\boldsymbol{\theta}$ exists and is continuous in $\boldsymbol{\theta}$, thus enabling a second order series expansion
- the expected value of the first and second derivative of the log-likelihood function is bounded

If these aspects are given, it can be shown that the maximum likelihood estimator is [MK2016, Ch. 6] [Sor1980] [WP1997, Ch. 3.3.3] [CJ2012, Ch. 2] [Jat2015, Ch. 3]

- asymptotically unbiased
- consistent
- asymptotically normal
- asymptotically efficient

See Appendix C.3 for detailed derivations. Especially the last property is advantageous, since it provides an estimate for the parameter variances through the asymptotic version of the Cramér-Rao bounds (2.79). As shown above, an approximation for the Fisher information matrix is readily available through the Hessian $\frac{\partial^2 J_{ML}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\Big|_{\hat{\boldsymbol{\theta}}_{ML}}$.

Apart from above mentioned advantages, it has to be noted, that most of the proofs rely heavily on the assumptions that the data samples are i.i.d. [Sor1980, Ch. 5], which may not be so easily guaranteed in practice. Also, perfect knowledge of the model structure, and its global identifiability are seldom given. Lastly above properties only hold in the limit as $N \to \infty$. Whereas this last aspect does not weigh so heavily in aircraft applications, since typically the number of samples is very large, perfect knowledge and identifiability pose a problem. Currently, this is mostly tackled by a trial- and error procedure, in order to find a model structures that best suits the problem at hand.

To summarize, WALTER notes on this topic that above mentioned properties merely show that maximum likelihood estimators have favorable properties under idealized conditions. However apart from some special cases (linear model structure and Gaussian, i.i.d. measurement noise), few theoretical results are available for real-life applications [WP1997, Ch. 3.3.3].

Nevertheless, one advantage of the maximum likelihood approach is the arising cost function structure. As will be shown, this is basically quadratic in nature, resembling a non-linear, weighted least-squares formulation. This is an intuitive approach to parameter estimation, if it was considered as a curve fitting problem. Thus, even neglecting the statistical roots of these cost function formulations, they have an intuitive appeal for the problem at hand. MAINE and ILLIF sum it up quite nicely [MIM1985, Ch. 5, p. 59]:

> "...even if some of the assumptions about the noise distribution are questionable, the estimators still make sense form a non-statistical viewpoint"

2.2.2.2 Solving the Maximum Likelihood Problem – The Output Error Method

So far, it has only been assumed that the samples $\boldsymbol{z}_k$ are i.i.d. Some further assumptions about the nature of the random component are necessary, in order to obtain a practical estimation algorithm. Assuming

- the deterministic model output is only distorted by an additive random component. This random component can often be interpreted as measurement noise.
- the additive random component is normally distributed, white, and has zero mean and constant covariance.

eventually leads to the so-called Output Error Method (OEM) [Jat2015, Ch. 4]. Above also implies that any input to the system needs to be considered as deterministic, and perfectly known.

Mathematically speaking, the measurements can then be described as

$$\boldsymbol{z}_k = \boldsymbol{y}_k(\boldsymbol{\theta}) + \boldsymbol{v}_k \tag{2.93}$$

$$\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2.94}$$

with the measurement noise covariance matrix $\mathbf{R}$. The residuals $\boldsymbol{r}_k$, describing the difference between the model output $\boldsymbol{y}_k(\boldsymbol{\theta})$ and measurements $\boldsymbol{z}_k$ are then equal to the measurement noise and are thus also normally distributed with zero mean and covariance $\boldsymbol{\mathcal{B}} = \mathbf{R}$

$$\boldsymbol{r}_k = \boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}) = \boldsymbol{v}_k \tag{2.95}$$

$$\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{B}}) \qquad \boldsymbol{\mathcal{B}} = \mathbf{R} \tag{2.96}$$

The assumption of Gaussian noise may be questioned here. However, a mathematically non-rigorous justification can be based on the Central Limit Theorem (see Appendix A.4.9.2): As soon as some sort of averaging is used in the sampling process, the samples approach a normal distribution, independently of the underlying, true distribution [WP1997, Ch. 3]. Also, this assumption leads to a cost function formulation, that closely resembles the least-squares cost function, which has yielded good results over centuries.

Using the expression for a multivariate Gaussian distribution (A.120), above assumptions can be used to express the conditional probability density $p(\boldsymbol{z}_k|\boldsymbol{\theta})$

$$p(\boldsymbol{z}_k|\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^{n_z}|\boldsymbol{\mathcal{B}}|}} \exp\Big(-\frac{1}{2}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}))^\intercal \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}))\Big) \tag{2.97}$$

Plugging $p(\boldsymbol{z}_k|\boldsymbol{\theta})$ in the expression for the negative log-likelihood function (2.88), the overall cost for the output error method arises

$$\begin{aligned} J_{OEM}(\boldsymbol{\theta}) &= -\sum_{k=0}^{N-1} \ln(p(\boldsymbol{z}_k|\boldsymbol{\theta})) \\ &= \sum_{k=0}^{N-1} \frac{1}{2}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}))^\intercal \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta})) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| + \frac{Nn_z}{2}\ln(2\pi) \end{aligned} \tag{2.98}$$

The last term is a constant, which does not influence the location of the optimum of above function and is thus usually discarded

$$\begin{aligned} J_{OEM}(\boldsymbol{\theta}) &= \frac{1}{2}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}))^\intercal \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta})) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| \\ &= \frac{1}{2}\sum_{k=0}^{N-1}\boldsymbol{r}_k(\boldsymbol{\theta})^\intercal \boldsymbol{\mathcal{B}}^{-1}\boldsymbol{r}_k(\boldsymbol{\theta}) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| \end{aligned} \tag{2.99}$$

This is now the common formulation for the maximum likelihood cost function in the OEM context. The solution of the arising optimization problem will be discussed in part II.

Here, it has to be stressed, that this is only one way of formulating a maximum likelihood parameter estimation problem, although probably the most widely used. Alternatives may for example involve: different probability distributions, non-constant variance, or correlated measurement noise. Example 2.5 shows the use of a non-Gaussian probability distribution.

The output error method in this form is the most widely applied approach for time-domain aircraft system identification from flight data [Jat2015, Ch. 3], and will also be at the heart of most discussions here.

2.2.2.3 Solving the Maximum Likelihood Problem – The Equation Error Method

Usually, the $\boldsymbol{y}_k$ in the aircraft context are outputs of a dynamical system. Thus their determination usually involves the solution of a system of differential equations, of-

Example 2.5: Maximum Likelihood Estimation with Non-Gaussian Noise
This example is rather academic in nature. It is intended to illustrate that the maximum likelihood principle is valid beyond additive, Gaussian noise models.
Consider a measurement model of the form

$$z_k = y_k(\boldsymbol{\theta}) + \ln v_k$$

where the noise is assumed to be distributed according to a Weibull distribution, with known and constant parameters $\lambda > 0$ and $\mu > 0$

$$p_{\mathrm{v}}(v) = \lambda\mu \cdot (\lambda v)^{\mu-1} \exp(-(\lambda v)^{\mu})$$

The residuals are then

$$r_k(\boldsymbol{\theta}) = z_k - y_k(\boldsymbol{\theta})$$
$$v_k = \exp(z_k - y_k(\boldsymbol{\theta})) = \exp(r_k(\boldsymbol{\theta}))$$

Using the transformation of probability densities of A.93 the probability density of the residuals is

$$p_{\mathrm{r}}(r_k(\boldsymbol{\theta})) = p_{\mathrm{v}}(\exp(r_k(\boldsymbol{\theta}))) \left| \left| \frac{\partial \exp(r_k(\boldsymbol{\theta}))}{\partial r_k(\boldsymbol{\theta})} \right| \right| = p_{\mathrm{v}}(\exp(r_k(\boldsymbol{\theta}))) \cdot \exp(r_k(\boldsymbol{\theta}))$$

For a set of N samples, similar derivations as in the previous section 2.2.2.2 can be performed to arrive at a negative log-likelihood cost function of the form

$$J = -\ln p(\mathbf{Z}|\boldsymbol{\theta}) = -\sum_{k=0}^{N-1} \ln(p_{\mathrm{v}}(\exp(r_k(\boldsymbol{\theta}))) \cdot \exp(r_k(\boldsymbol{\theta})))$$
$$= -N \ln(\mu\lambda^{\mu}) - \sum_{k=0}^{N-1} \mu \cdot r_k(\boldsymbol{\theta}) + \sum_{k=0}^{N-1} (\lambda \exp(r_k(\boldsymbol{\theta})))^{\mu}$$

Thus a maximum likelihood estimate can be obtained by solving the following optimization problem

$$\min_{\boldsymbol{\theta}} J = -\sum_{k=0}^{N-1} \mu \cdot r_k(\boldsymbol{\theta}) + \sum_{k=0}^{N-1} (\lambda \exp(r_k(\boldsymbol{\theta})))^{\mu}$$

ten formulated as initial value problem. If they were however the outputs of a static system, they could be determined purely based on model parameter and input values.

The first case, in which a system may be considered as "static" arises, if the internal dynamics may be neglected, e.g. because only steady state behavior is of interest. This is easily accommodated in the output error formulation

$$\boldsymbol{z}_k = \boldsymbol{y}_k(\boldsymbol{u}_k, \boldsymbol{\theta}) + \boldsymbol{v}_k \tag{2.100}$$

with the difference to the foregoing section being that $\boldsymbol{y}_k(\boldsymbol{u}_k, \boldsymbol{\theta})$ now does not arise from the solution of an initial value problem, but algebraically relates parameters $\boldsymbol{\theta}$ and inputs $\boldsymbol{u}$ to the outputs.

The second case covers "quasi-static" formulations: if measurements (or results from data post-processing) of the states and their derivatives were available, the need

for the solution of a system of differential equations would be eliminated. The augmented "outputs" in this case are a combination of state derivatives and original outputs $\begin{bmatrix} \dot{\boldsymbol{x}}_k^\mathsf{T} & \boldsymbol{y}_k^\mathsf{T} \end{bmatrix}^\mathsf{T}$, whereas states and original inputs $\begin{bmatrix} \boldsymbol{x}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} \end{bmatrix}^\mathsf{T}$ may be interpreted as "inputs" to the static system.

$$\boldsymbol{z}_k^a = \begin{bmatrix} \dot{\boldsymbol{x}}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \\ \boldsymbol{y}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_k^x \\ \boldsymbol{v}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \\ \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \end{bmatrix} + \boldsymbol{v}_k^a \tag{2.101}$$

Some call this approach the "equation error method" [MK2016, Ch. 6.4] [Jat2006, Ch. 6]. However, since this is merely a special case of the output error method (where the integration of the system equations is circumvented) all of the respective considerations may easily be applied and no special treatment of an equation error formulation is deemed necessary in this work.

The downside of this approach is, that often no information about the state derivatives is available. Sometimes, they may be obtained through post-processing approaches, as will be illustrated in section 3.8 as alternative to brute force finite differences. However, the quality of the derivative estimates is often insufficient, especially if low-cost sensors are used as is common for Remotely Piloted Aerial System (RPAS) applications.

2.2.2.4 Solving the Maximum Likelihood Problem – The Filter Error Method

Sometimes not all effects can be covered by a deterministic model formulation with additive, white, Gaussian noise. In these cases there exists the possibility to consider a random component *within* the model formulation, so-called process noise, leading to Filter Error Methods (FEMs) [Jat2015, Ch. 1].

This approach is more complicated and can oftentimes be circumvented by an advantageously chosen model formulation, or comprehensive data-preprocessing. Also, above mentioned "random component within the model structure" in aircraft applications is usually turbulent air, so the necessity of filtering approaches may be dropped, if flight tests are conducted on days with calm air, preferably very early in the morning.

Once a random component within the model is considered, determining the system outputs is not straightforward anymore. Usually a state estimation algorithm has to be employed in order to reconstruct outputs based on measurements. In the case of dynamic system identification, this is commonly done using a Kalman filter [MIM1985] for state and output estimation.

Once the estimated outputs $\hat{\boldsymbol{y}}_k(\boldsymbol{\theta})$ are determined, similar assumptions as in the OEM case of section 2.2.2.2 may be employed, namely that the residuals describing the difference between estimated model outputs $\hat{\boldsymbol{y}}_k(\boldsymbol{\theta})$ and measurements $\boldsymbol{z}_k$ are

- white (thus ensuring the Markov Criterion (2.86))
- normally distributed with

- zero mean
- known covariance.

Expressing this mathematically leads to

$$\boldsymbol{r}_k = \boldsymbol{z}_k - \hat{\boldsymbol{y}}_k(\boldsymbol{\theta}) \tag{2.102}$$

$$\mathbf{r}_\mathrm{k} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{B}}_k) \tag{2.103}$$

$$\mathrm{Cov}[\boldsymbol{r}_k, \boldsymbol{r}_j] = \delta_{ik}\boldsymbol{\mathcal{B}}_k \tag{2.104}$$

where the main difference to the result of section 2.2.2.2 is that the estimated system outputs $\hat{\boldsymbol{y}}_k(\boldsymbol{\theta})$ are used instead of $\boldsymbol{y}_k(\boldsymbol{\theta})$.

Especially stating that the residuals are white seems counter-intuitive, since the Kalman filter offers the best possible state estimate, based on all past measurements. Thus one might expect the residuals to be correlated in time. However, it can be shown, that the linear Kalman filter is a *whitening* filter, which guarantees that the resulting residuals are white [MIM1985, Ch. 7], if the propagated outputs $\hat{\boldsymbol{y}}_{k|k-1}(\boldsymbol{\theta})$ are used in above equation, see chapter 2.3 and Appendix D.2 for details.

With these considerations in mind, the same approach as in section 2.2.2.2 can be used to arrive at the cost function for the FEM

$$J_{FEM}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=0}^{N-1}\Big(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1}(\boldsymbol{\theta})\Big)^\intercal \boldsymbol{\mathcal{B}}_k^{-1}\Big(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1}(\boldsymbol{\theta})\Big) + \frac{1}{2}\sum_{k=0}^{N-1}\ln|\boldsymbol{\mathcal{B}}_k| \tag{2.105}$$

Often, the time-dependency of the residual noise covariance is dropped to simplify the problem. It is approximated using a constant $\boldsymbol{\mathcal{B}}$ that represents an average over the considered time span

$$J_{FEM}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=0}^{N-1}\Big(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1}(\boldsymbol{\theta})\Big)^\intercal \boldsymbol{\mathcal{B}}^{-1}\Big(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1}(\boldsymbol{\theta})\Big) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| \tag{2.106}$$

Then the only difference to the OEM case is that the model outputs are estimated using one of the state estimation approaches to be illustrated in chapter 2.3.

The unknowns of the problem may then be divided into three subsets: states, model parameters, and noise covariance matrices [MK2016, Ch. 6]. Especially the last group poses significant practical problems when algorithms are developed to automatically estimate noise covariance matrix entries. It has to be carefully chosen which elements are to be included: there exist formulations that include combinations of measurement noise, process noise, or Kalman gain matrix elements. Also, strategies on how to counteract the resulting convergence issues have to be incorporated.

One of the possible formulations has been applied to aircraft parameter estimation in the presence of process noise. There, the elements of the process noise covariance matrix are estimated as nuisance parameters, along with the aerodynamic characteristics. However, in order to obtain useful convergence rates, artificial corrections of this covariance matrix have to be introduced. Further, additional constraints have to be

considered, in order to keep Kalman filter parameters meaningful [MI1981b] [Jat2015, Ch. 5]. These aspects usually discourage from using the FEM in connection with process noise covariance estimation, although there have been some successful applications [JP1989, JP1990].

Because of the difficulties just mentioned the focus in this work lies on state and parameter estimation, for cases where the noise characteristics are given. This usually means that they have to be tuned manually for good results. The resulting algorithm then obtains estimates for the states using the Kalman filter, whereas model parameters are estimated using a maximum likelihood based approach.

It seems tempting to replace the Kalman filter output estimate in above formulation by a Rauch-Tung-Striebel (RTS) smoother result, using the algorithm of section 2.3.3.2. This offers smoother state estimates with lower covariances, with only minor additional computations. However, a theoretical problem arises: the smoother residuals will not be white anymore, which is a basic prerequisite for the derivations in this section. This is probably the reason, why MAINE and ILIFF state, that "the direct use of a smoother in [above cost function formulation] is simply incorrect" [MIM1985, p. 99]. On the other hand, the conditions under which the Kalman filter residuals can be considered "white" (perfectly known model, correct noise characteristics,...) are hard to meet in practice to begin with. This is why it will have to be considered on a case to case basis, if above theoretical flaw might be accepted in favor of better state estimates.

In some simple example cases, that arose during the work on this thesis, an optimization solution using a filter instead of a smoother was achieved in fewer iterations. This may be, because the smoother "hides" the effect of parameter changes, making it harder to determine their optimum value.

2.2.3 Bayesian Estimation

The fundamental difference between maximum likelihood methods, and Bayesian estimation is the treatment of the unknown parameters $\boldsymbol{\theta}$. So far, $\boldsymbol{\theta}$ has been assumed to be an unknown, but constant vector of model parameters. In the Bayesian approach, it is considered to be the realization of a random variable, which can be characterized by a probability density function $p(\boldsymbol{\theta})$ [CS2011, Ch. 2.4]. Here, a conundrum arises: determining exactly this characterization of the parameters via $p(\boldsymbol{\theta})$ tends to be difficult, when the original task is to determine "good" numeric values for them [MIM1985, Ch. 4]. In other words, knowledge of the result is necessary in order to be able to solve the problem.

However, if the situation arises that knowledge about some of the parameters is available, but cannot be fully trusted, i.e. is more or less "uncertain". Then, the tools to be presented next offer the possibility to include this information in the estimation process.

2.2.3.1 Estimator Definition

The basic idea of maximizing a probability density function remains the same, i.e. the mode of a density function is considered to be the estimate. However, the probability density function to be considered is now the posterior distribution $p(\boldsymbol{\theta}|\mathbf{Z})$. Applying Bayes' Theorem (A.108) to relate $p(\boldsymbol{\theta}|\mathbf{Z})$ to $p(\mathbf{Z}|\boldsymbol{\theta})$ via the joint distribution $p(\mathbf{Z}, \boldsymbol{\theta})$ yields [MIM1985, Ch. 4.1] [MK2016, Ch. 4]

$$p(\boldsymbol{\theta}|\mathbf{Z}) = \frac{p(\mathbf{Z}, \boldsymbol{\theta})}{p(\mathbf{Z})} = \frac{p(\mathbf{Z}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{Z})} \tag{2.107}$$

As is discussed in [MIM1985], the posterior distribution $p(\boldsymbol{\theta}|\mathbf{Z})$ reflects the information about the parameters *after* performing the experiment: It combines the prior information ($p(\boldsymbol{\theta})$) and the information gained by the experiment ($p(\mathbf{Z}|\boldsymbol{\theta})$, $p(\mathbf{Z})$) and thus constitutes the most information one can obtain about the parameters from a statistical point of view.

Another way of looking at this is presented by CZADO: some prior knowledge, in the form of the prior distribution $p(\boldsymbol{\theta})$, is adjusted to the "new knowledge" about $\boldsymbol{\theta}$, which can be inferred from the data $\mathbf{Z}$ via the conditional probability $p(\mathbf{Z}|\boldsymbol{\theta})$ (and the marginal probability $p(\mathbf{Z})$) [CS2011, Ch. 2.4].

The maximum a posteriori probability (MAP) estimate according to MAINE, ILIFF [MIM1985] is then the mode of above conditional probability density

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{MAP} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{Z}) = \arg\max_{\boldsymbol{\theta}} p(\mathbf{Z}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} -\ln p(\mathbf{Z}|\boldsymbol{\theta}) - \ln p(\boldsymbol{\theta}) \end{aligned} \tag{2.108}$$

The second equality holds, since $p(\mathbf{Z})$ is not a function of the parameters and thus does not influence the maximizing argument.

If the parameters' probability distribution $p(\boldsymbol{\theta})$ is continuous, and non-zero in a neighborhood of the estimate $\hat{\boldsymbol{\theta}}_{MAP}$, the MAP estimator inherits the maximum likelihood estimator's properties, such as asymptotic consistency and efficiency [WP1997, Ch. 3.5.1].

Other estimator definitions exist, which are optimal in the Bayesian sense (e.g. the a posteriori expected value, Bayesian minimum risk [MIM1985, Ch 4.3]), however only the MAP estimator will be treated here, due to its close relation to maximum likelihood estimation. For different assumptions about the prior distribution, different relations result, which will be discussed next.

2.2.3.2 Uniform Prior Parameter Distribution

If the prior parameter distribution $p(\boldsymbol{\theta})$ is assumed to be the uniform distribution between two boundaries $\boldsymbol{\theta}_{lb}$ and $\boldsymbol{\theta}_{ub}$, i.e. if the probability density can be written as

$$p\Big([\boldsymbol{\theta}]_{(l)}\Big) = \begin{cases} \frac{1}{[\boldsymbol{\theta}_{ub}]_{(l)} - [\boldsymbol{\theta}_{lb}]_{(l)}} & [\boldsymbol{\theta}]_{(l)} \in \Big[[\boldsymbol{\theta}_{lb}]_{(l)}, [\boldsymbol{\theta}_{ub}]_{(l)}\Big] \\ 0 & \text{otherwise} \end{cases} \qquad l = 1, \ldots, n_{\boldsymbol{\theta}} \tag{2.109}$$

Then, as long as the parameters stay within their boundaries, the prior distribution is merely a constant, and does not influence the location of the minimum of the MAP estimate. This is called a non-informative prior [CS2011, Ch. 2.4], and it essentially makes the estimate the same as the maximum likelihood estimate [MIM1985, Ch. 4], with additional box constraints on the parameters. The estimator then maximizes both, the a posteriori probability distribution, and the likelihood function.

A discussion on the necessity of box constraints on parameter estimates can be found in section 3.4. An illustration on how they are incorporated into an optimization algorithm is shown in section 2.1.3.4 and algorithm 2.3.

2.2.3.3 Constant Parameter

Another link to the maximum likelihood estimates of the last section is to think of a (possibly unknown) constant $\boldsymbol{\theta}$ as a discrete random variable, with the following probability mass function $m(\boldsymbol{\theta}_{prior})$. For details on the relation between discrete and continuous random variables, see Appendix A.4.1.

$$m(\boldsymbol{\theta}_{prior}) = \begin{cases} 1 & \boldsymbol{\theta}_{prior} = \boldsymbol{\theta} \\ 0 & \text{otherwise} \end{cases} \tag{2.110}$$

For this type of random variable, it holds

$$\mathrm{E}[\boldsymbol{\theta}_{prior}] = \boldsymbol{\theta} \tag{2.111}$$

$$\mathrm{Cov}[\boldsymbol{\theta}_{prior}] = \mathbf{0} \tag{2.112}$$

Although plugging this into the definition of the posterior probability (2.107) may not be allowed when looking at it too mathematically rigorous[7], but it makes for an intuitive link to maximum likelihood estimation

$$p(\boldsymbol{\theta}_{prior}|\mathbf{Z}) = \frac{p(\mathbf{Z}|\boldsymbol{\theta}_{prior})}{p(\mathbf{Z})} m(\boldsymbol{\theta}_{prior}) = \begin{cases} \frac{p(\mathbf{Z}|\boldsymbol{\theta})}{p(\mathbf{Z})} & \boldsymbol{\theta}_{prior} = \boldsymbol{\theta} \\ 0 & \text{otherwise} \end{cases} \tag{2.113}$$

Thus the only way to maximize the posterior density function is to chose the value of the constant $\boldsymbol{\theta}$ such, as to maximize the likelihood function $p(\mathbf{Z}|\boldsymbol{\theta})$. Even though the result is the same (i.e. the maximum likelihood estimator) the prior knowledge involved is different, compared to the foregoing section 2.2.3.2: here, it is assumed, that the parameters are constants, whereas in the last section, they were assumed to be uniformly distributed.

[7] mixing continuous and discrete random variables necessitate some more elaborate measure theoretic tools, like distributions in the sense of generalized functions

2.2.3.4 Gaussian Parameter Distributions

Assuming that the prior parameter distribution is Gaussian with mean $\boldsymbol{\theta}_{prior}$ and covariance $\boldsymbol{\Sigma}$ yields [MK2016, Ch. 4]

$$p(\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^{n_{\boldsymbol{\theta}}}|\boldsymbol{\Sigma}|}} \exp\Big(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{prior})^{\intercal}\, \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_{prior})\Big) \tag{2.114}$$

Combining this with derivations similar to those of sections 2.2.2.2 and 2.2.2.4, the resulting cost function is

$$\begin{aligned} J_{MAP}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{k=0}^{N-1} \boldsymbol{r}_k(\boldsymbol{\theta})^{\intercal}\, \boldsymbol{\mathcal{B}}^{-1}\boldsymbol{r}_k(\boldsymbol{\theta}) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| \\ + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{prior})^{\intercal}\, \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_{prior}) + \frac{1}{2}\ln|\boldsymbol{\Sigma}| \end{aligned} \tag{2.115}$$

Thus a direct dependency of the cost on the parameters $\boldsymbol{\theta}$ is introduced via a quadratic penalty, in addition to the indirect dependency via the residuals $\boldsymbol{r}(\boldsymbol{\theta})$.

Apart from its statistical interpretation, this may help to "regularize" the problem, since the positive (semi-) definite matrix $\boldsymbol{\Sigma}$ is added to the Hessian of the cost function in the maximum likelihood estimation

$$\frac{\partial^2 J_{MAP}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} = \frac{\partial^2 J_{ML}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} + \boldsymbol{\Sigma}^{-1} \tag{2.116}$$

From an optimization point of view, the second term helps to keep the Hessian "well behaved" by always adding a positive definite term $\boldsymbol{\Sigma}^{-1}$. It thus aids in overcoming intermediate problems when e.g. inverting the matrix of second derivatives. These problems may arise due to an unfavorable intermediate choice of parameter values by the optimization algorithm. In these cases, the quadratic parameter term "pulls" the estimates towards the prior estimates, thus reducing the risk of numerical and algorithmic problems, e.g. close to saddle points.

From a statistical point of view, the resulting MAP estimate will eventually lie between the pure maximum likelihood estimate and the prior estimate, where both information sources are used "optimally", in the sense that they are combined according to their covariances. Also the two limiting cases arise:

- for very "sure" prior estimates (i.e. very "small" $\boldsymbol{\Sigma}$), the term related to the prior estimates will dominate the optimization, and the result will lie very close to the prior mean $\boldsymbol{\theta}_{prior}$.
- For very "unsure" prior estimates, with hardly any information, the inverse prior covariance matrix will be negligible and the result will essentially be the maximum likelihood estimate.

In this framework, it is also easy to include information on only a subset of parameters. If no prior information is available for a set of parameters, the corresponding rows

and columns of $\boldsymbol{\Sigma}^{-1}$ are set to zero. Statistically speaking, this corresponds to an infinite variance of those parameters, thus no prior information is available [MIM1985]. From an optimization point of view, setting some rows and columns of $\boldsymbol{\Sigma}^{-1}$ to zero removes the effect of the corresponding element from the cost function. Consequently, only information based on the experimental data will be used to determine the value of the respective parameter.

2.2.3.5 Statistical Properties of Bayesian Estimates

The MAP estimate converges to the maximum likelihood estimate for large sample sizes. Thus it shares some of the favorable asymptotic properties of the maximum likelihood estimator: namely asymptotic consistency and efficiency [CJ2012, Ch. 2.6]. Uncertainties in the parameter estimates may then be quantified based purely on the use of the Fisher information matrix, as will be illustrated in section 2.2.5.

However, since these are only asymptotic properties, WALTER proposes to include the prior information on the parameters if the number of samples is finite [WP1997, Ch. 5.3.2]. The resulting inequality, which resembles the Cramér-Rao bounds [WP1997, Ch. 5.3.2] [Sor1980], is however not usable in practice, since it involves the expectation of the Fisher information over the complete parameter space $\mathrm{E}[\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})]$. MAINE and ILIFF propose the following approximation for a Gaussian prior [MIM1985, Ch. 5.4.]

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{MAP}\middle|\mathbf{Z}\right] \approx \left(\frac{\partial^2 J_{MAP}\left(\hat{\boldsymbol{\theta}}_{MAP}\right)}{\partial \boldsymbol{\theta}^2}\right)^{-1} = \left(\frac{\partial^2 J_{ML}\left(\hat{\boldsymbol{\theta}}_{MAP}\right)}{\partial \boldsymbol{\theta}^2} + \boldsymbol{\Sigma}^{-1}\right)^{-1} \tag{2.117}$$

Here, the Hessian of the maximum likelihood cost function is used as approximation for $\mathrm{E}[\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})]$. The validity of above approximation may be based on two intuitive arguments:

As will be shown in section 3.2, $\frac{\partial^2 J_{ML}}{\partial \boldsymbol{\theta}^2}$ may very well be approximated as a sum over the output sensitivities for all samples. Thus it grows with the sample size N and will eventually dominate over the prior estimate (given that the experiment is performed such as to actually collect new information about the parameters). This illustrates once more the fact, that the MAP estimates converge towards the maximum likelihood estimates.

Secondly, appendix A.4.7 illustrates, how two independent estimates $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ of a parameter $\boldsymbol{\theta}$ may be optimally combined to yield an estimate $\hat{\boldsymbol{\theta}}_c$ with minimum parameter variances. The resulting covariance estimate is

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right] = \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1}$$

which is similar in form as the MAP covariance estimate. This underlines once more, that the Bayesian approach can be considered as the "optimal" combination of prior information and new information, introduced via the measurements in the maximum likelihood part.

2.2.3.6 Remarks on the Bayesian Approach

The Bayesian approach is not often encountered in aircraft parameter estimation, due to the difficulty of obtaining the prior density $p(\boldsymbol{\theta})$ [Jat2015, Ch. 4]. Sometimes, prior knowledge is available from other sources (wind-tunnel experiments, prior estimations), however then also the chosen model formulation needs to be consistent with this prior data in order for it to be useful.

Here, it is mainly interesting for two reasons: on the one hand, ill-posed problems may be regularized by an "artificial" prior, in order to enlarge the radius of convergence of the optimization algorithm in use. On the other hand, the fundamentals of state estimation may also be formulated in the Bayesian sense, as will be shown in chapter 2.3. The considerations of this chapter form the basis for this.

2.2.4 Least-Squares Estimation

The main aspects of the least-squares technique date back to C.F. GAUSS [Gau1857]. Here, they are here presented using modern vector matrix notation, mainly following the presentation in [Sor1980, Ch. 2]. Similar ideas are illustrated in [MK2016, Jat2006, CJ2012, Sim2006].

2.2.4.1 Non-Statistical Linear Problem and Its Solution

The basic setting in linear least-squares is as follows: Assume a scalar model that is linear in the parameters

$$y = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{\theta} \tag{2.118}$$

Considering a scalar, linear model is usually enough, since tasks involving multi-dimensional outputs can be easily partitioned into several, independent, scalar sub-problems, as long as the model is linear in the parameters.

Now the *dependent variable* y can only be observed with an additive error v resulting in the measurements z. This observation is done multiple times, to arrive at a set of N samples

$$z_k = \boldsymbol{x}_k\boldsymbol{\theta} + v_k \qquad k = 0 \dots N-1 \tag{2.119}$$

$$\begin{aligned} \boldsymbol{z} = \begin{bmatrix} z_0 \\ \vdots \\ z_{N-1} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{x}_0^{\mathsf{T}} \\ \vdots \\ \boldsymbol{x}_{N-1}^{\mathsf{T}} \end{bmatrix} \boldsymbol{\theta} + \boldsymbol{v} \\ &= \begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_{n_{\boldsymbol{\theta}}} \end{bmatrix} \boldsymbol{\theta} + \boldsymbol{v} \\ &= \mathbf{X}\boldsymbol{\theta} + \boldsymbol{v} \end{aligned} \tag{2.120}$$

In the applications to be presented here, the sorting index k usually refers to time, but in other applications this might not be true. The columns of the *regressor matrix* $\mathbf{X}$

are commonly referred to as "regressors" $\boldsymbol{\xi}$, whereas its rows[8] $\boldsymbol{x}_k^\intercal$ represent the values of the regressors at one sampling instant. The regressors are assumed to be perfectly known, and errors are only allowed to enter via $\boldsymbol{v}$. Furthermore, the approach presented here is only meaningful if $N \gg n_{\boldsymbol{\theta}}$ and if $\mathbf{X}$ has full column rank. A precondition for $\mathbf{X}$ to have full column rank is for the regressors to be linearly independent.

The fact that only linear relations of the dependent variable and the parameters can be included may seem overly restrictive. However, since the regressors may be any non-linear function of the independent variables, a great model variability is still possible. Quite advanced approaches involving splines or radial basis functions can be boiled down to linear least-squares problems with only slight restrictions.

Now a "good" parameter value $\hat{\boldsymbol{\theta}}_{LS}$ is determined such, that the 2-norm of the error vector is minimum, which results in the following cost function

$$J_{LS} = \frac{1}{2}\boldsymbol{e}^\intercal\boldsymbol{e} = \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta})^\intercal(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}) \tag{2.121}$$

In order to find a minimum of this cost function, its gradient w.r.t. the parameter vector is set to zero, and its Hessian is investigated further [Sor1980, Ch. 2] [MK2016, Ch. 5]

$$\frac{\partial J_{LS}}{\partial \boldsymbol{\theta}} \overset{!}{=} \mathbf{0} = -\mathbf{X}^\intercal\left(\boldsymbol{z} - \mathbf{X}\hat{\boldsymbol{\theta}}_{LS}\right) = -\mathbf{X}^\intercal\boldsymbol{z} + \mathbf{X}^\intercal\mathbf{X}\hat{\boldsymbol{\theta}}_{LS} \tag{2.122}$$

$$\frac{\partial^2 J_{LS}}{\partial \boldsymbol{\theta}^2} = \mathbf{X}^\intercal\mathbf{X} \tag{2.123}$$

From the second equation it follows that the Hessian in this case is always positive definite, since $\mathbf{X}$ has full rank. Thus the resulting optimum is a minimum.

The first equation results in a linear system of equations, which can be solved for the least-squares solution $\hat{\boldsymbol{\theta}}_{LS}$

$$\begin{aligned}(\mathbf{X}^\intercal\mathbf{X})\,\hat{\boldsymbol{\theta}}_{LS} &= \mathbf{X}^\intercal\boldsymbol{z} \\ \Rightarrow \hat{\boldsymbol{\theta}}_{LS} &= (\mathbf{X}^\intercal\mathbf{X})^{-1}\,\mathbf{X}^\intercal\boldsymbol{z}\end{aligned} \tag{2.124}$$

The first of above equations is commonly known as the "normal" equations. Here the fact that $\mathbf{X}$ is assumed to have full column rank comes into play: only then is a unique solution $\hat{\boldsymbol{\theta}}_{LS}$ possible. Numerical issues may arise, not only if the regressors $\boldsymbol{\xi}$ are perfectly linearly dependent, but also if they resemble each other too closely. Then $\mathbf{X}^\intercal\mathbf{X}$ will not be well conditioned and solving above linear system of equations may become difficult. This situation may be interpreted as follows: the algorithm is not able to map a variation in the response uniquely to one regressor, since there are several combinations that may amount to almost the same effect [MK2016, Ch. 5.1].

[8] A remark on notation is in order: even though the regressors are not to be confused with the states of a dynamic system, they still fully describe the output of the model at one sampling instant. Due to this close relation it was chosen to use the same symbol $\boldsymbol{x}$ (and its concatenated version $\mathbf{X}$) both for the states of dynamic systems in other chapters, and the regressors at one sampling instant in linear least-squares problems.

Despite an ill-conditioned matrix $\mathbf{X}^\intercal\mathbf{X}$, results may still be obtained, since there exist numerically stable algorithms for the solution of least squares problems. Orthogonal decomposition methods such as the QR decomposition or the Singular Value Decomposition (SVD) may be employed, which do not necessitate the explicit inversion of $\mathbf{X}^\intercal\mathbf{X}$.

The difference between measurements $\boldsymbol{z}$ and estimated model outputs $\hat{\boldsymbol{y}}$ are called "residuals" $\boldsymbol{r}$

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\theta}}_{LS} = \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{z} \tag{2.125}$$

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{z} - \hat{\boldsymbol{y}} = \boldsymbol{z} - \mathbf{X}\hat{\boldsymbol{\theta}}_{LS} \\ &= \left(\mathbf{I}_N - \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\right)\boldsymbol{z}\end{aligned} \tag{2.126}$$

It is interesting to note that according to the definition of the minimum via a root of the cost function gradient (2.122), the residuals need to be orthogonal to all columns of the regressor matrix $\mathbf{X}$

$$\mathbf{X}^\intercal\boldsymbol{r} = \mathbf{X}^\intercal\left(\mathbf{I}_N - \mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\right)\boldsymbol{z} = \mathbf{0} \tag{2.127}$$

Further, it can be verified that the matrix $\mathbf{X}(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal$ is an orthogonal projection of a vector in $\mathbb{R}^N$ onto the subspace spanned by the columns of the regressor matrix $\mathbf{X}$ [Sor1980, App. B] [CJ2012, Ch. 1].

Thus the measurement vector $\boldsymbol{z}$ is partitioned into two parts: the estimated model outputs $\hat{\boldsymbol{y}}$ are the projection of the measurements $\boldsymbol{z}$ onto the space spanned by the regressor vectors $\boldsymbol{\xi}_i$; the residuals are the projection of $\boldsymbol{z}$ onto a space orthogonal to $\mathbf{X}$ [Sor1980, Ch. 2]. Figure 2.3 illustrates this. It also shows the geometrical necessity of linear independence between the regressor vectors $\boldsymbol{\xi}_i$: if they were not independent, the example plane would degrade to a line and in *two* dimensions no meaningful projection would be possible. This orthogonal projection characteristics were, amongst others, used by KALMAN in his original derivation of the filter bearing his name [Kal1960].

Sometimes it is known that the quality of some measurements is different from that of other measurements. This can be taken into consideration by introducing a positive definite, symmetric, weighting matrix $\mathbf{W}$ in the cost function. This leads to generalized least-squares, whose solution can be obtained in exactly the same manner as above [Sor1980, Ch. 2]

$$J_{WLS} = \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta})^\intercal\,\mathbf{W}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}) \tag{2.128}$$

$$\hat{\boldsymbol{\theta}}_{WLS} = (\mathbf{X}^\intercal\mathbf{W}\mathbf{X})^{-1}\,\mathbf{X}^\intercal\mathbf{W}\boldsymbol{z} \tag{2.129}$$

However, it is often difficult to determine reasonable values for the weighing matrix when applying this approach.

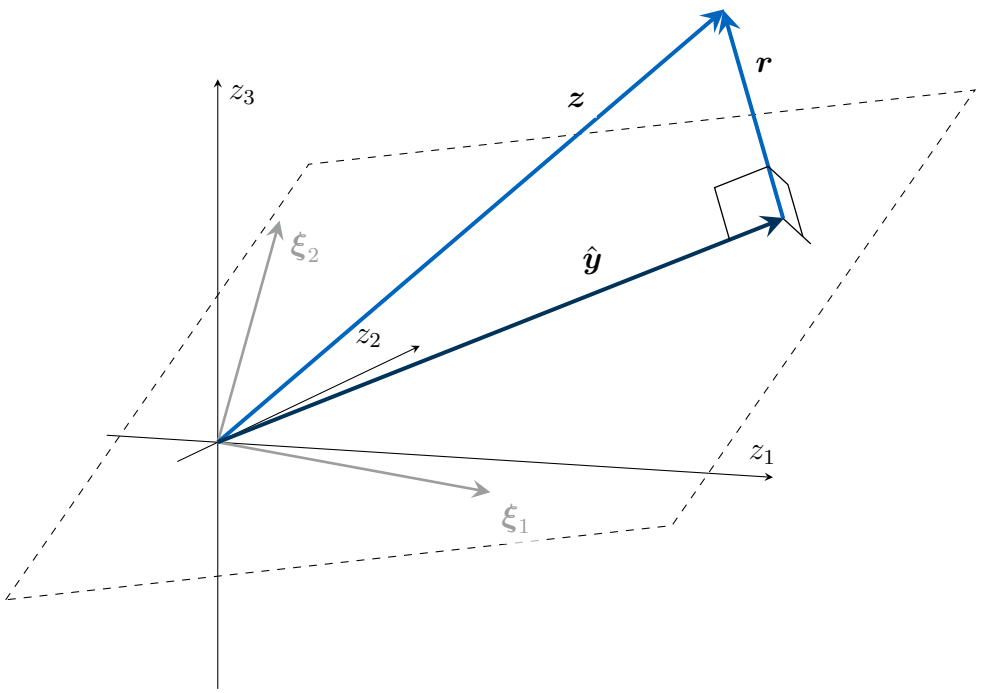


Figure 2.3: *Illustration of the projection characteristics in least-squares: the measurement vector $\boldsymbol{z}$ is projected onto the hyper-surface spanned by the regressor matrix $\mathbf{X} = [\boldsymbol{\xi}_1\ \boldsymbol{\xi}_2]$ in order to obtain the estimated model output $\hat{\boldsymbol{y}} = \boldsymbol{\xi}_1\hat{\theta}_1 + \boldsymbol{\xi}_2\hat{\theta}_2$. The remaining residuals $\boldsymbol{r}$ are then perpendicular to $\hat{\boldsymbol{y}}$ (and all regressors $\boldsymbol{\xi}_i$)*

2.2.4.2 Linear least-squares with Equality Constraints

Sometimes linear relationships between the model parameters are known, which may be included in least-squares optimization. The problem to be solved can then be formulated as

$$\begin{pmatrix} \min\limits_{\boldsymbol{\theta}} \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta})^\mathsf{T}\mathbf{W}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}) \\ \text{s.t.} \quad \mathbf{A}\boldsymbol{\theta} = \boldsymbol{b} \end{pmatrix} \tag{2.130}$$

where $\mathbf{A}$ is assumed to have full rank

$$\text{rank}(\mathbf{A}^\mathsf{T}) < n_{\boldsymbol{\theta}} \tag{2.131}$$

i.e. the possible solutions are constrained to a linear subspace of $\mathbb{R}^{n_{\boldsymbol{\theta}}}$. One standard way of going about this problem is the Lagrangian approach, as was illustrated in section 2.1.3.1. However, a different approach can be taken, if a solution $\boldsymbol{\theta}^{eq}$ to the constraint equation was available

$$\mathbf{A}\boldsymbol{\theta}^{eq} = \boldsymbol{b} \tag{2.132}$$

It has to be emphasized that $\boldsymbol{\theta}^{eq}$ is an *arbitrary*, but *known* parameter that solves the linear constraint equation. Next, a basis for the null-space of $\mathbf{A}$ is needed, i.e. a full-

rank matrix $\mathbf{Z}$, such that

$$\mathbf{A}\mathbf{Z} = \mathbf{0} \tag{2.133}$$

$$\mathbf{Z} \in \mathbb{R}^{n_{\boldsymbol{\theta}} \times (n_{\boldsymbol{\theta}} - \text{rank}(\mathbf{A}^\intercal))} \tag{2.134}$$

All possible solutions to the equality constraint equation may then be expressed based on the known solution $\boldsymbol{\theta}^{eq}$ together with linear combinations of the columns of $\mathbf{Z}$

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{eq} + \mathbf{Z}\boldsymbol{\theta}^{free} \tag{2.135}$$

since they automatically fulfill the constraint equation

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{A}\boldsymbol{\theta}^{eq} + \mathbf{A}\mathbf{Z}\boldsymbol{\theta}^{free} = \mathbf{A}\boldsymbol{\theta}^{eq} = \boldsymbol{b} \tag{2.136}$$

Thus the admissible parameter space is built from an offset $\boldsymbol{\theta}^{eq}$ together with the feasible directions in $\mathbf{Z}$. Then a transformation of the cost function is possible, by using this parameterization of the feasible space

$$\begin{aligned} J_{WLS} &= \frac{1}{2}\left(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq} - \mathbf{X}\mathbf{Z}\boldsymbol{\theta}^{free}\right)^\intercal \mathbf{W}\left(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq} - \mathbf{X}\mathbf{Z}\boldsymbol{\theta}^{free}\right) \\ &= \frac{1}{2}\left(\tilde{\boldsymbol{z}} - \tilde{\mathbf{X}}\boldsymbol{\theta}^{free}\right)^\intercal \mathbf{W}\left(\tilde{\boldsymbol{z}} - \tilde{\mathbf{X}}\boldsymbol{\theta}^{free}\right) \\ \text{with} \quad \tilde{\boldsymbol{z}} &= \boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq} \\ \tilde{\mathbf{X}} &= \mathbf{X}\mathbf{Z} \end{aligned} \tag{2.137}$$

Above cost function automatically includes only solutions, which fulfill the equality constraints, and can thus be treated as unconstrained problem. Also, the parameter dimension is reduced to the remaining degrees of freedom $n_{\boldsymbol{\theta}^{free}} = n_{\boldsymbol{\theta}} - \text{rank}(\mathbf{A}^\intercal)$ that are not constrained via $\mathbf{A}\boldsymbol{\theta} = \boldsymbol{b}$. This constitutes the so-called null-space method of quadratic programming with equality constraints [SS2011]. It searches for possible solutions only in the reduced space of free directions, implemented in terms of the null-space basis $\mathbf{Z}$.

The solution to above reduced problem is then straight-forward

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{LS}{}^{free} &= \left(\tilde{\mathbf{X}}^\intercal \mathbf{W} \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\intercal \mathbf{W} \tilde{\boldsymbol{z}} \\ &= \left(\mathbf{Z}^\intercal \mathbf{X}^\intercal \mathbf{W} \mathbf{X} \mathbf{Z}\right)^{-1} \mathbf{Z}^\intercal \mathbf{X}^\intercal \mathbf{W} (\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq}) \end{aligned} \tag{2.138}$$

The null-space basis $\mathbf{Z}$ has, per definition, $n_{\boldsymbol{\theta}} - \text{rank}(\mathbf{A}^\intercal)$ linearly independent columns, thus above matrix inversion poses no problem. The solution to the original, equality constraint problem is then

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{LS}{}^{c} &= \boldsymbol{\theta}^{eq} + \mathbf{Z}\hat{\boldsymbol{\theta}}_{LS}{}^{free} \\ &= \boldsymbol{\theta}^{eq} + \mathbf{Z}\left(\mathbf{Z}^\intercal \mathbf{X}^\intercal \mathbf{W} \mathbf{X} \mathbf{Z}\right)^{-1} \mathbf{Z}^\intercal \mathbf{X}^\intercal \mathbf{W} (\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq}) \end{aligned} \tag{2.139}$$

This illustrates, that the computational cost for solving an augmented problem in the Lagrangian method is replaced by the computational cost of determining the null-space of $\mathbf{A}$, which may be significantly lower, if $n_{\boldsymbol{\theta}} - \text{rank}(\mathbf{A}^\intercal)$ is small, i.e. if $\mathbf{Z}$ has many more rows than columns.

2.2.4.3 Statistical Interpretation

So far, the least-squares approach has only been presented from an optimization point of view, with some geometric interpretations. However, under suitable assumptions about the additive noise term, linear least-squares problems can be interpreted from a statistical point of view, too.

Assume, the measurement noise is zero mean, with known covariance matrix

$$\mathrm{E}[\boldsymbol{v}] = \mathbf{0} \tag{2.140}$$

$$\mathrm{Cov}[\boldsymbol{v}] = \mathbf{R} \tag{2.141}$$

The assumption of zero mean does not pose a significant restriction, since a non-zero mean can easily be included as bias term in the regressor matrix and estimated along with the model parameters. Here, the complete noise vector for all time instants $\boldsymbol{v} = \begin{bmatrix} v_0 & \cdots & v_{N-1} \end{bmatrix}^\intercal$ is considered, which also allows for noise that is correlated between the samples.

With above assumption, it can be shown that the least-squares estimator (ordinary and weighted) is unbiased [Sor1980, Ch. 2]

$$\begin{aligned} \mathrm{E}\left[\hat{\boldsymbol{\theta}}_{WLS}\right] &= \mathrm{E}\left[\left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}\boldsymbol{z}\right] = \left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}\cdot\mathrm{E}[\mathbf{X}\boldsymbol{\theta}+\boldsymbol{v}] \\ &= \left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta} = \boldsymbol{\theta} \end{aligned} \tag{2.142}$$

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}_{LS}\right] = \mathrm{E}\left[\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\boldsymbol{z}\right] = \cdots = \left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{X}\boldsymbol{\theta} = \boldsymbol{\theta} \tag{2.143}$$

In above derivation, it has been used that the regressor matrix $\mathbf{X}$, the parameter vector $\boldsymbol{\theta}$ and the weighting matrix $\mathbf{W}$ are deterministic, and that the noise $\boldsymbol{v}$ has zero mean.

Furthermore, the parameter covariance matrix in the weighted least-squares case is [Sor1980, Ch. 2]

$$\begin{aligned} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{WLS}\right] &= \mathrm{Cov}\left[\left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}(\mathbf{X}\boldsymbol{\theta}+\boldsymbol{v})\right] = \mathrm{Cov}\left[\left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}\boldsymbol{v}\right] \\ &= \left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{W}\mathbf{R}\mathbf{W}\mathbf{X}\left(\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\right)^{-1} \end{aligned} \tag{2.144}$$

and for ordinary least-squares ($\mathbf{W} = \mathbf{I}_N$) the following results

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}\right] = \left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{R}\mathbf{X}\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1} \tag{2.145}$$

An interesting special case arises, if the weighting matrix in the weighted least-squares case is chosen to be the inverse of the measurement noise covariance matrix $\mathbf{W} = \mathbf{R}^{-1}$

$$\begin{aligned} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{WLS}\right] &= \left(\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{X}\left(\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\right)^{-1} \\ &= \left(\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\right)^{-1} \end{aligned} \tag{2.146}$$

This results in the best, linear, unbiased estimator (BLUE), see [Sor1980] for details and the corresponding proof of optimality. A further special case may be formulated for white noise for which it holds that $\mathbf{R} = \sigma_v^2\mathbf{I}_N$

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{WLS}\right] = \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}\right] = \sigma_v^2\left(\mathbf{X}^\intercal\mathbf{X}\right)^{-1} \tag{2.147}$$

This last case allows for an especially useful approach, since for large sample sizes, the sample variance (A.159) of the residuals $s_{r,N}^2$ constitutes a good estimate for the noise variance σ_v^2. Thus it is possible to obtain the covariance matrix of the estimated parameters without having to specify the noise characteristics.

2.2.4.4 Statistical Estimation With Equality Constraints

Using the same assumptions as above (zero-mean noise with known covariance) least-squares estimation involving deterministic equality constraints may be treated in a similar manner. First, the true, deterministic parameter vector is split in three parts without loss of generality

$$\boldsymbol{\theta} = \boldsymbol{\theta}^{eq} + \mathbf{Z}\boldsymbol{\theta}^{free} + \boldsymbol{\theta}^{\perp} \tag{2.148}$$

The first part solves the equality constraints, the second is built from the basis vectors of the null-space of $\mathbf{A}$, and the last, remaining part covers the parts which may not be explained by the columns of $\mathbf{Z}$, i.e. for which it holds $\mathbf{Z}^\intercal\boldsymbol{\theta}^{\perp} = \mathbf{0}$. This last part would actually violate the equality constraints, the consequences of which will be discussed later. The measurements can then be expressed as

$$\boldsymbol{z} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{v} = \mathbf{X}\boldsymbol{\theta}^{eq} + \mathbf{X}\mathbf{Z}\boldsymbol{\theta}^{free} + \mathbf{X}\boldsymbol{\theta}^{\perp} + \boldsymbol{v} \tag{2.149}$$

Using this, the expected value of the constraint estimate (2.139) is

$$\begin{aligned}
\mathrm{E}\left[\hat{\boldsymbol{\theta}}_{LS^c}\right] &= \mathrm{E}\left[\boldsymbol{\theta}^{eq} + \mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}^{eq})\right] \\
&= \boldsymbol{\theta}^{eq} + \mathrm{E}\left[\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\left(\mathbf{X}\boldsymbol{\theta}^{eq} + \mathbf{X}\mathbf{Z}\boldsymbol{\theta}^{free} + \mathbf{X}\boldsymbol{\theta}^{\perp} + \boldsymbol{v} - \mathbf{X}\boldsymbol{\theta}^{eq}\right)\right] \\
&= \boldsymbol{\theta}^{eq} + \mathbf{Z}\boldsymbol{\theta}^{free} + \mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\boldsymbol{\theta}^{\perp} \\
&= \boldsymbol{\theta} + \left(\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X} - \mathbf{I}_{n_{\boldsymbol{\theta}}}\right)\boldsymbol{\theta}^{\perp}
\end{aligned} \tag{2.150}$$

in above derivation, apart from simple matrix multiplications, it was only used that the measurement noise is assumed to have zero mean, and that all components of the parameter vector ($\boldsymbol{\theta}^{eq}$, $\boldsymbol{\theta}^{free}$, $\boldsymbol{\theta}^{\perp}$), as well as the regressor matrix $\mathbf{X}$ and null-space basis $\mathbf{Z}$ are deterministic.

This shows, that the weighted least-squares estimator with equality constraints can only be unbiased, if the true parameter also fulfills the equality constraints (i.e. $\boldsymbol{\theta}^{\perp} = \mathbf{0}$). This intuitively makes sense, since if the true parameter is not located within the feasible set of the optimization problem, the estimation bias will necessarily consist of components perpendicular to this feasible set. This situation may arise, if the equality constraints do not capture the real nature of the problem, but are wrongly specified by the analyst.

The covariance of the estimate can be obtained by first computing the covariance

in the free directions (using the decomposition (2.135) and estimation results (2.138))

$$\begin{aligned}\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right] &= \mathrm{Cov}\left[(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\,\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}(\boldsymbol{z}-\mathbf{X}\boldsymbol{\theta}^{eq})\right] \\ &= (\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\,\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\underbrace{\mathrm{Cov}[(\mathbf{X}\boldsymbol{\theta}+\boldsymbol{v}-\mathbf{X}\boldsymbol{\theta}^{eq})]}_{=\mathbf{R}}\mathbf{W}^\intercal\mathbf{X}\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1} \\ &= (\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\,\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\cdot\mathbf{R}\cdot\mathbf{W}^\intercal\mathbf{X}\mathbf{Z}(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{W}\mathbf{X}\mathbf{Z})^{-1}\end{aligned} \tag{2.151}$$

Then the covariance of the constraint estimate can be determined

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{c}\right] = \mathrm{Cov}\left[\boldsymbol{\theta}^{eq}+\mathbf{Z}\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right] = \mathbf{Z}\cdot\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right]\cdot\mathbf{Z}^\intercal \tag{2.152}$$

Again considerable simplification can be achieved, if the weighting matrix $\mathbf{W}$ is chosen to be the inverse of the measurement noise covariance matrix

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right] = \left(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\mathbf{Z}\right)^{-1} \tag{2.153}$$

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{c}\right] = \mathbf{Z}\left(\mathbf{Z}^\intercal\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\mathbf{Z}\right)^{-1}\mathbf{Z}^\intercal \tag{2.154}$$

Above equations first condense the unconstrained information matrix $\mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}$ onto the space of free directions via $\mathbf{Z}$. This condensed information is then used to compute the covariance matrix of the parameters in the free directions $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right]$. Thus the uncertainty involved may only exist in those directions, that are not constrained. To obtain the parameter error covariance matrix of the complete problem, $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{LS}{}^{free}\right]$ is then mapped onto the original parameter space, again via $\mathbf{Z}$. This mapping necessarily introduces high correlations between the estimates of the original problem, which fits intuition very well: linear equality constraints enforce linear relationships between parameters, which have to be reflected in the correlations between their estimates.

2.2.4.5 Extension to Non-Linear Models

If there is no way to formulate the model in a linear fashion, one has to resort to non-linear least-squares of the form

$$\boldsymbol{z}_k = \boldsymbol{y}_k(\boldsymbol{\theta}) + \boldsymbol{v}_k \qquad k = 0\ldots N-1 \tag{2.155}$$

$$J_{NLS} = \frac{1}{2}\sum_{k=0}^{N-1}(\boldsymbol{z}_k-\boldsymbol{y}_k(\boldsymbol{\theta}))^\intercal\,\mathbf{R}_k^{-1}(\boldsymbol{z}_k-\boldsymbol{y}_k(\boldsymbol{\theta})) \tag{2.156}$$

In the non-linear case, it is often not possible to decouple the output variables, thus consideration of all outputs at once is necessary. In addition to weighting individual sampling instants, it also becomes possible to weigh different output signals via the elements of $\mathbf{R}_k$. Here, the case is illustrated, where the outputs are weighted according to the measurement noise covariance matrix, and more/less importance can be given to sampling instants via the explicit time-dependency of $\mathbf{R}_k$. It should be noted, that it is usually challenging enough to determine proper values for the measurement noise

covariance matrix at one point in time, not to mention its evolution throughout the period of interest.

This problem can in general not be solved analytically, and one has to employ iterative methods, as were presented in chapter 2.1 on cost function optimization.

2.2.4.6 Consideration of Prior Knowledge

The least-squares problem formulation allows for a straight forward way to include prior knowledge on the parameters: they can be considered as additional, artificial measurement [Sor1980, Ch. 2].

Assume, prior knowledge on the parameters is available in the form of its mean $\boldsymbol{\theta}_{prior}$ and covariance matrix $\boldsymbol{\Sigma}$. Further assume that the prior estimate is statistically independent of the measurement noise. Then, using the identity $\boldsymbol{\theta}_{prior} = \boldsymbol{\theta} + (\boldsymbol{\theta}_{prior} - \boldsymbol{\theta})$, the weighted least-squares approach can be modified

$$\tilde{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\theta}_{prior} \\ z_0 \\ \vdots \\ z_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} \\ \boldsymbol{x}_0^\intercal \\ \vdots \\ \boldsymbol{x}_{N-1}^\intercal \end{bmatrix} \boldsymbol{\theta} + \begin{bmatrix} \boldsymbol{\theta}_{prior} - \boldsymbol{\theta} \\ v_0 \\ \vdots \\ v_{N-1} \end{bmatrix} = \tilde{\mathbf{X}}\boldsymbol{\theta} + \tilde{\boldsymbol{v}} \tag{2.157}$$

$$\mathrm{Cov}[\tilde{\boldsymbol{v}}] = \tilde{\mathbf{R}} = \begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{2.158}$$

Only the use of a weighted least-squares approach makes sense here, since otherwise the knowledge about $\boldsymbol{\Sigma}$ would be lost. Choosing $\mathbf{W} = \tilde{\mathbf{R}}^{-1}$ yields

$$\begin{aligned} J_{WLS} &= \frac{1}{2}\left(\tilde{\boldsymbol{z}} - \tilde{\mathbf{X}}\boldsymbol{\theta}\right)^\intercal \tilde{\mathbf{R}}^{-1}\left(\tilde{\boldsymbol{z}} - \tilde{\mathbf{X}}\boldsymbol{\theta}\right) \\ &= \frac{1}{2}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta})^\intercal \mathbf{R}^{-1}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}) \end{aligned} \tag{2.159}$$

$$\hat{\boldsymbol{\theta}}_{WLS} = \left(\boldsymbol{\Sigma}^{-1} + \mathbf{X}^\intercal\mathbf{R}^{-1}\mathbf{X}\right)^{-1}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta}_{prior} + \mathbf{X}^\intercal\mathbf{R}^{-1}\boldsymbol{z}\right) \tag{2.160}$$

For a non-informative prior, i.e. $\boldsymbol{\Sigma} \to \infty$, the estimator of section 2.2.4.1 on weighted least-squares can be obtained. The extension to non-linear least-squares is straight forward, and results in the following cost function

$$J_{NLS} = \frac{1}{2}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta}) + \frac{1}{2}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta}))^\intercal \mathbf{R}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\boldsymbol{\theta})) \tag{2.161}$$

which again needs to be solved iteratively.

2.2.4.7 Connection to Other Estimation Approaches

Comparing the cost function formulations of this section, with the output error formulation presented in section 2.2.2.2, a close relationship between the two becomes

evident: Under the following assumptions, the (non-) linear least-squares estimator is a maximum likelihood estimator [Sor1980, Ch. 5] [CJ2012, Ch. 2]

- the noise $\boldsymbol{v}$ is additive
- the noise $\boldsymbol{v}$ is Gaussian, with known covariance $\mathbf{R} = \boldsymbol{\mathcal{B}}$
- the parameters $\boldsymbol{\theta}$ are unknown, but deterministic constants

Then, the negative log-likelihood function and Fisher information matrix become

$$-\ln(\mathbb{L}(\boldsymbol{\theta})) = \frac{1}{2}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta})^{\intercal}\,\mathbf{R}^{-1}(\boldsymbol{z} - \mathbf{X}\boldsymbol{\theta}) + \frac{N}{2}\ln|\mathbf{R}| + \frac{Nn_z}{2}\ln(2\pi) \tag{2.162}$$

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) = -\mathrm{E}\left[\frac{\partial^2 \ln \mathbb{L}}{\partial \boldsymbol{\theta}^2}\right] = \mathbf{X}^{\intercal}\mathbf{R}^{-1}\mathbf{X} \tag{2.163}$$

Now, the expression for the covariance of the parameters in the weighted least-squares case (2.146) is actually equal to above Fisher information matrix. Thus it is shown, that in this setting, the weighted least-squares estimator attains the Cramér-Rao lower bound and is thus efficient, even in the non-asymptotic case [CJ2012, Ch. 2].

If prior knowledge on the parameters is to be included, the comparison should take place with the cost functions arising in the Bayesian approach of section 2.2.3.4. Then the least-squares estimator becomes a MAP estimator under the following additional conditions [Sor1980, Ch. 5]

- the parameters $\boldsymbol{\theta}$ are unknown, random constants (replacing above condition of an unknown, deterministic constant)
- the parameters $\boldsymbol{\theta}$ are Gaussian, with known mean and covariance

If the least-squares estimator can be considered to be a maximum likelihood / MAP estimator, its favorable asymptotic properties (consistency, asymptotic normality, asymptotic efficiency) apply as well.

2.2.4.8 Comments on Least-Squares Estimation in Aircraft Parameter Estimation

In practical application, some of the basic assumptions of least-squares estimation are often violated.

For instance, it is often very difficult to explicitly specify the measurement noise covariance $\mathbf{R}$. Then, to simplify computations, measurement errors are often assumed to be white, i.i.d. noise. This assumption is widely spread, albeit seldom justified in practice. The result is still an unbiased parameter estimate, but the covariance estimate tends to be overly confident [MK2016, Ch. 5.1]. Some remedies to this will be presented in section 2.2.5, where the possibility to include noise coloring in the covariance estimate is discussed.

Another violated assumption, at least in aircraft applications, is that of perfectly known regressors. They are often functions of measured quantities, such as angle of

attack, rotational rates, or velocity components. Those measurements can show significant noise influences. Application of ordinary least-squares techniques then lead to a biased estimate [CJ2012, Ch. 2.7.3]. There exists an approach to take care of errors in the regressors, namely "total least-squares". However, it exhibits some disadvantageous characteristics, which limit is practical applicability: total least-squares problems are inherently worse conditioned compared to the respective ordinary least-squares problem. Furthermore, the parameter covariances are commonly larger, and the advantage of the total least-squares approach is only useful, if the bias term of the ordinary case becomes dominant in the result [CJ2012, Ch. 2.7.3].

Another problem arises with the regressors $\boldsymbol{\xi}$: One wishes to have regressors, that are as different from each other as possible, in order to have less numerical problems when inverting the information matrix $\mathbf{X}^\intercal\mathbf{X}$. Eventually this also results in only weakly correlated parameter estimates. However, in aircraft applications, the typical regressors (body rotational rates, control surface deflections) may not be excited independently, since they are coupled via flight mechanics: It is impossible to, e.g. excite aileron deflections without provoking a roll rate, which necessarily leads to correlations of the respective parameters. This is not only problematic for linear least squares estimation, but also for the more general, non-linear maximum likelihood case. This effect has to be kept in mind, and investigated on a case by case basis.

In summary, one can conclude that even though some of the basic assumptions of the least-squares estimator may be violated in practice, it is still a very important approach in parameter estimation problems for aircraft. This is mainly due to two reasons: On the one hand, its robust, simple an non-iterative nature allows for quick results. On the other hand, least-squares results are commonly used as initial guesses for more advanced parameter estimation algorithms [Jat2015, Ch. 6]. Thus small systematic errors, introduced by violated assumptions, will be compensated for in later stages of the parameter estimation process. It is then more important that initial guesses can be obtained in a robust fashion, quick, and within reasonable limits of the final result.

2.2.5 Statistical Properties of Parameter Estimates

2.2.5.1 The Fisher Information Matrix

In all of the three discussed estimation approaches, it was shown that the cost function Hessian presents a viable approximation or is equal to the respective Fisher information matrix. Since all approaches are (asymptotically) efficient, its inverse may be used as approximation to the parameter covariance matrix [Jat2015, Ch. 4].

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1} \overset{(2.77)}{=} \left(-\mathrm{E}\left[\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right]\right)^{-1} \approx \left(\frac{\partial^2 J\left(\hat{\boldsymbol{\theta}}\right)}{\partial \boldsymbol{\theta}^2}\right)^{-1} \tag{2.164}$$

WALTER illustrates the approximations, which are involved here in more detail, specifically they are [WP1997, Ch. 5.3.1]

- often the efficiency of the estimators is only achieved asymptotically, whereas the sample sizes are finite.
- $\boldsymbol{\mathcal{F}}\left(\hat{\boldsymbol{\theta}}\right)^{-1}$ is used as approximation to the Fisher information at the true parameter value $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1}$
- The Hessian of the cost function $\frac{\partial^2 J\left(\hat{\boldsymbol{\theta}}\right)}{\partial \boldsymbol{\theta}^2}$ is used as approximation for the Fisher information matrix $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1}$. However, this relation is based on some assumptions about the noise (Markov criterion, known covariance), which are never fully satisfied in reality.

The consequences of above approximations depend strongly on the case at hand, but should always be kept in the back of one's head. They can be alleviated somewhat if [WP1997, Ch. 5.3.1]

- the number of samples is large
- the parameter's influence on the output is close to linear
- the assumptions about the noise are correct (Markov-criterion, constant covariance)
- the noise influence is small, i.e. the signal-to-noise ratio is large (JATEGAONKAR proposes a ratio of $10:1$ for aircraft applications [Jat2006, Ch. 2.5])

Further criticism on the Fisher information matrix as source of estimates for the parameter uncertainties results from the assumptions about the underlying noise process [WP1997, Ch. 5.4]: The estimators, which are used here, are based on the assumption of mutually independent realizations of the noise processes involved. In many practical applications, this assumption does not hold. Especially, if the true model structure is unknown, there results a structural, deterministic error, which is perfectly repeatable.

Nevertheless, the inspection of $\frac{\partial^2 J\left(\hat{\boldsymbol{\theta}}\right)}{\partial \boldsymbol{\theta}^2}$ is convenient, since it is computed as a by-product of the optimization process anyway. This partly explains the fact that it is probably the most widely used characterization method for parameter uncertainty. Additionally, there are other interpretations, which are not based on notions of efficiency, which underline its usefulness: In all of the discussed applications, the Fisher information matrix is a quadratic function of the *output sensitivities* (see section 3.3 for details). Output sensitivities relate the magnitude of a change in a model parameter to a change in the actual output, i.e. if the output sensitivity is large, a small change in a parameter entails significantly modified outputs. Parameter estimation routines will then have less problems to accurately locate an estimate, since small changes will significantly alter the response error. The opposite is true for small output sensitivities: the estimation routine will be able to significantly change a parameter value, without

modifying the model output too much, making accurate determination of its numerical value harder.

Thus it makes sense, to investigate a quadratic function of the "amount of output sensitivities accumulated over an experiment", e.g. a sum of squared output sensitivities. This can be used to assess, if the algorithm had a chance of precisely determining the parameter value [MK2016, Ch. 9]. Further merits of the Fisher information matrix, presented from an optimization point of view will be illustrated in section 3.7.

2.2.5.2 Residual Coloring

It has been observed quite early in the history of aircraft parameter estimation that the scatter in parameter estimates from different experiments do not fit well with the covariance estimates based on the Fisher information matrix. The latter was often observed to be too optimistic, which is why sometimes a fudge factor of 5 to 10 was used in the uncertainty estimates.

MAINE and ILIFF traced this anomaly back to noise coloring [IM1977, MI1981a]: a basic assumption is that the noise processes involved are white. However, this is not true in real-life applications, especially if unmodeled dynamics remain. Then, a considerable part of the residual power will be located in the frequency range of aircraft rigid body motion.

This situation can nicely be illustrated for the general least squares case, of section 2.2.4.3. It has been shown, that if the weighting matrix $\mathbf{W}$ was chosen as the inverse of the noise covariance $\mathbf{R}$, the best, linear, unbiased estimator (BLUE) results. However, if a noise covariance $\hat{\mathbf{R}}$ is assumed that only covers parts of the true noise characteristics

$$\mathbf{R} = \hat{\mathbf{R}} + \Delta\mathbf{R} \qquad \Leftrightarrow \qquad \mathbf{W} = \hat{\mathbf{R}}^{-1} \tag{2.165}$$

the estimator covariance based on equation (2.144) is

$$\begin{aligned} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{WLS}\right] &= \left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1} \mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{R}\hat{\mathbf{R}}^{-1}\mathbf{X}\left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1} \\ &= \left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1} + \left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1} \mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\Delta\mathbf{R}\hat{\mathbf{R}}^{-1}\mathbf{X}\left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1} \end{aligned} \tag{2.166}$$

Thus, the covariance estimate $\left(\mathbf{X}^\intercal \hat{\mathbf{R}}^{-1}\mathbf{X}\right)^{-1}$ would be too optimistic, since the second part in above equation is ignored. Nevertheless, the actual estimate is still unbiased, as is also shown in section 2.2.4.3.

This is exactly what happens if residual coloring is ignored and the noise is wrongly assumed to be white: the estimated noise matrix $\hat{\mathbf{R}}$ will be (block-) diagonal in general, and any off-diagonal elements in $\Delta\mathbf{R}$, due to coloring, will be ignored.

MORELLI and KLEIN show how this effect may be incorporated a posteriori with little additional computations [MK1997, MK2016]. They propose to estimate the resid-

ual autocorrelation after the parameter estimates have been determined.

$$\hat{\mathbf{R}}_i^{\mathrm{r}} = \frac{1}{N} \sum_{k=0}^{N-1-i} (\boldsymbol{r}_k - \bar{\boldsymbol{r}}_N)(\boldsymbol{r}_{k+i} - \bar{\boldsymbol{r}}_N)^\intercal = \left(\hat{\mathbf{R}}_{-i}^{\mathrm{r}}\right)^\intercal \qquad i = 0 \dots r \tag{2.167}$$

where r implements a band-limitation on residual coloring, since only contributions up to r samples in the past are considered. This autocorrelation can then be used for a correction, by approximating the true noise covariance with the following band-diagonal matrix

$$\mathbf{R} \approx \mathbf{R}^{corr} = \begin{bmatrix} \hat{\mathbf{R}}_0^{\mathrm{r}} & \cdots & \hat{\mathbf{R}}_r^{\mathrm{r}} & & \\ \vdots & \hat{\mathbf{R}}_0^{\mathrm{r}} & & \ddots & \\ \hat{\mathbf{R}}_{-r}^{\mathrm{r}} & & \ddots & & \hat{\mathbf{R}}_r^{\mathrm{r}} \\ & \ddots & & & \vdots \\ & & \hat{\mathbf{R}}_{-r}^{\mathrm{r}} & \cdots & \hat{\mathbf{R}}_0^{\mathrm{r}} \end{bmatrix} \tag{2.168}$$

This approximation may then be used in above equation 2.166 to correct the covariance estimate which was based on the assumption of white noise.

Naturally, the question arises, why above approximation is not used as new weighting matrix $\mathbf{W}$ for a new solution to the original problem. MORELLI and KLEIN state that the approximated noise covariance matrix is large and often ill-conditioned for typical flight test data, which inhibits convergence. Thus they propose to compute parameter estimates using generalized or ordinary least-squares, and merely correct the covariance estimate in above fashion [MK2016, Ch. 5.2].

In this work, residual coloring will be treated along with the general uncertainty quantification approach in section 3.5. There some remarks on its application to maximum likelihood estimation will also be presented. Alternatively, a correction factor of 5 or 10 may be applied to the parameter standard error estimates in order to account for the overly confident covariance estimates when residual coloring is neglected [MK2016, Ch. 5].

2.2.5.3 Parameter Characteristics

From the parameter covariance matrix, some very important statistical characteristics of the estimates $\hat{\boldsymbol{\theta}}$ can be extracted.

Parameter Standard Deviations

For one, the main diagonal elements of $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]$ are the variances of the parameter estimates, i.e. their standard deviations can be extracted as [MK2016, Ch. 5]

$$\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right] = \sqrt{\left[\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}\right]\right]_{(j,j)}} \tag{2.169}$$

See the definition of the standard deviation in equation (A.69). Relating the standard deviations to the actual parameter value results in a relative standard deviation [Jat2015, Ch. 10]

$$\sigma_{\text{rel}}\left[\left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right] = \frac{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right]}{\left[\hat{\boldsymbol{\theta}}\right]_{(j)}} \cdot 100\% \tag{2.170}$$

Large values of the relative standard deviation should be investigated closer. The two most common explanations are that either the information content of the data with respect to this particular parameter is too low, thus it cannot be determined very exactly. Alternatively, if the numerical value of the particular parameter is small, its relative standard deviation also grows rapidly. Both aspects are undesirable and should be circumvented, by either trying to collect more informative data, or by investigating, if the respective parameter could be dropped from the model.

Parameter Correlation

The second quantity, which can be extracted from the parameter covariance matrix $\text{Cov}\left[\hat{\boldsymbol{\theta}}\right]$ are the parameter correlation coefficients [MK2016, Ch. 5] [Jat2015, Ch. 11]

$$\rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(i)}, \left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right] = \frac{\left[\text{Cov}\left[\hat{\boldsymbol{\theta}}\right]\right]_{(i,j)}}{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(i)}\right] \cdot \sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right]} \tag{2.171}$$

They are the cross-variances of two parameter estimates, normalized with the respective standard deviations. The $\rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(i)}, \left[\hat{\boldsymbol{\theta}}\right]_{(j)}\right]$ describe the linear relation between the two parameters, see equation (A.85).

Arranging the correlation coefficients in a matrix yields the correlation matrix, see equation (A.86) [MK2016, Ch. 5]

$$\begin{aligned}\text{Corr}\left[\hat{\boldsymbol{\theta}}\right] = &\begin{bmatrix} \rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(1)}, \left[\hat{\boldsymbol{\theta}}\right]_{(1)}\right] & \cdots & \rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(1)}, \left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}\right] \\ \vdots & \ddots & \vdots \\ \rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}, \left[\hat{\boldsymbol{\theta}}\right]_{(1)}\right] & \cdots & \rho\left[\left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}, \left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}\right] \end{bmatrix} = \\ &\begin{bmatrix} \frac{1}{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(1)}\right]} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}\right]} \end{bmatrix} \text{Cov}\left[\hat{\boldsymbol{\theta}}\right] \begin{bmatrix} \frac{1}{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(1)}\right]} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma\left[\left[\hat{\boldsymbol{\theta}}\right]_{(n_{\boldsymbol{\theta}})}\right]} \end{bmatrix}\end{aligned} \tag{2.172}$$

Absolute values of the correlation coefficients close to 1 indicate that the respective parameters might be linearly dependent on each other. In this case, the estimation algorithm may have problems to find independent estimates of the parameters. This is also an undesirable situation, and modification of the model formulation, or inclusion

of prior knowledge can be remedies. Alternatively, new, more informative data can be gathered, to make the effect of certain parameters more distinct, thus reducing their correlation.

Since in aircraft applications many influencing factors are connected closely via flight mechanics (e.g. aileron deflection cannot be investigated without the resulting roll-rate), comparatively high correlations may appear. In one example, MORELLI accepts correlations that have an absolute value just below 0.9 [MK2016, Ex. 5.1], where JATEGAONKAR states that correlations over 0.9 should be investigated, and those over 0.95 can be considered linearly dependent [Jat2015, Ch. 11].

Further Approaches for Uncertainty Quantification

Apart from a simple investigation of the Fisher information matrix, and the derived parameter covariances, many more approaches to investigate the uncertainty in the parameters exist. WALTER illustrates the use of confidence regions and cost contours. However they quickly become cumbersome if the parameter space is of dimension higher than four [WP1997, Ch. 5], since drawing point clouds cannot be achieved in three dimensional space anymore. There exist possibilities to reduce the dimensions involved (e.g. by projection on subspaces), or to investigate only principal components. But all of these approaches require a large number of model runs, which is another reason why they were not pursued further in this work.

Furthermore, WALTER illustrates approaches based on Monte-Carlo simulations, the use of which could be part of a future work on this topic [WP1997, Ch. 5.2]. Their merit lies in the fact that they characterize the uncertainty in the parameters for finite sample sizes, and adapted to the estimation problem at hand.

2.3 State Estimation

As soon as the system under investigation is not deterministic, simple integration in time cannot be used to obtain the model outputs any longer. Moreover, filtering approaches have to be employed to reconstruct a valid state and output trajectory based on the available information. In the context of aircraft system identification, the Kalman Filter has shown to yield appropriate results.

The next sections will lay out the basics necessary to understand state estimation in the context of system identification. Most of the content will be based on ideas commonly associated with KALMAN and RAUCH, TUNG, and STRIEBEL. However, this is by no means intended to give an extensive overview over the topic. A good starting point for further inquiries are [Sim2006, Jaz1970, CJ2012, RTS1965, Kal1960].

After some remarks on the naming convention in use, the following section will introduce the basic ideas for linear systems. Subsequent sections will deal with the maximum likelihood interpretation of state estimation, smoothing, as well as non-linear problems and the approximations necessary to solve them.

2.3.1 Notation

Propagation, Correction, Smoothing

Naming conventions for the presentation of state estimation topics differ greatly from publication to publication. The basic problem being, that per time-step, usually two state estimates are available: one that incorporates the information contained in past measurements, the second incorporates the information at the current point in time as well. The former is often termed the *propagated* or *a priori* state estimate, and is usually given with its covariance matrix as quantity describing the uncertainty involved. The latter is known as the *corrected* or *updated* state estimate, and is also computed with its own covariance matrix.

Some authors use different accent characters, or superscripts to indicate the difference. However, here two time scales in the subscripts will be used, to indicate the current time step (first part), and the time up until which measurements are processed (second part). The advantage of this notation is that it may easily be extended to the smoothing problem, where all information up to the last sample is incorporated.

This results in the following expression for the state estimate at time k, including measurement information up until time l

$$\hat{\boldsymbol{x}}_{k|l} \tag{2.173}$$

Depending on the relation between k and l, there arise three different scenarios to be investigated [MK2016, Ch. 4]:

1. $k > l$: in the *prediction* case, only measurement information until a point l in the past is included, and the current state is merely predicted.

2. $k = l$: in the *filter* case, all information, up to the current point in time is used, to obtain the currently optimal state estimate
3. $k < l$: in the *smoothing* case, also future values are incorporated to improve the current state estimate. This can only be achieved offline for a batch of data.

In a probabilistic framework, as e.g. JAZWINSKI presents it [Jaz1970], the current state estimate can be based on the knowledge of the following conditional probability density

$$p(\boldsymbol{x}_k|\boldsymbol{z}_0, \dots \boldsymbol{z}_l) = p(\boldsymbol{x}_k|\mathbf{Z}_l) \tag{2.174}$$

Then, different approaches to arrive at an actual estimate can be taken. Two widely used ones are

1. the "mode", i.e. peak value of $p(\boldsymbol{x}_k|\mathbf{Z}_l)$, resulting in a maximum likelihood state estimate. This approach is related to the aspects illustrated in section 2.2.2 on maximum likelihood estimation.
2. the conditional mean of $p(\boldsymbol{x}_k|\mathbf{Z}_l)$

For unimodal, symmetric distributions (as e.g. the Gaussian distribution) the mode and mean are the same [Cox1963], thus the estimate is

$$\hat{\boldsymbol{x}}_{k|l} = \mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_l] \tag{2.175}$$

The general estimation error is then

$$\tilde{\boldsymbol{x}}_{k|l} = \hat{\boldsymbol{x}}_{k|l} - \boldsymbol{x}_k \tag{2.176}$$

The accompanying covariance matrix is

$$\begin{aligned} \mathbf{P}^{\mathbf{x}}_{k|l} &= \mathrm{Cov}[\boldsymbol{x}_k|\mathbf{Z}_l] = \mathrm{E}[(\boldsymbol{x}_k - \mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_l])(\boldsymbol{x}_k - \mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_l])^{\mathsf{T}}|\mathbf{Z}_l] \\ &= \mathrm{E}\Big[\Big(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|l}\Big)\Big(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|l}\Big)^{\mathsf{T}}\Big|\mathbf{Z}_l\Big] \\ &= \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|l}\tilde{\boldsymbol{x}}^{\mathsf{T}}_{k|l}\Big|\mathbf{Z}_l\Big] = \mathrm{Cov}\Big[\tilde{\boldsymbol{x}}_{k|l}\Big|\mathbf{Z}_l\Big] = \mathbf{P}^{\tilde{\mathbf{x}}}_{k|l} \end{aligned} \tag{2.177}$$

The next to last equality is based on the assumption that the estimation error $\tilde{\boldsymbol{x}}_{k|l}$ is zero mean, as will be shown in the subsequent sections. Above shows, that with the definition of the estimate as conditional mean, the state covariance $\mathbf{P}^{\mathbf{x}}_{k|l}$ and state estimation error covariance $\mathbf{P}^{\tilde{\mathbf{x}}}_{k|l}$ are the same. The above applies to all cases, i.e. prediction ($k > l$), filtering ($k = l$), and smoothing ($k < l$).

Indices in Difference Equations

The usual formulation for finite time difference equations over N samples will be

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k\boldsymbol{x}_k + \boldsymbol{\Gamma}_k\boldsymbol{u}_k \qquad\qquad k = 0 \dots \bar{N} - 1 \tag{2.178}$$

where in order to keep notation short, the abbreviation

$$\bar{N} = N - 1 \tag{2.179}$$

will be used.

2.3.2 Linear Filtering

KALMAN first introduced the basic ideas for the discrete time filter based on orthogonal projections [Kal1960]. Later, he published the continuous time equivalent together with BUCY [KB1961]. Since it better fits in the overall setting of this work, the basic derivation of the Kalman filter will follow the thoughts as presented in [Jaz1970, Sim2006, CJ2012], which rely on the minimization of estimation errors, as opposed to orthogonal projection.

2.3.2.1 System Description

The system, which will be considered for the linear Kalman filter case, is a discrete time, linear state space system, with two noise sources: measurement noise $\boldsymbol{v}$ at the system's output and process noise $\boldsymbol{w}$ at the system's input.

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \mathbf{F}_k \boldsymbol{w}_k \qquad k = 0 \dots \bar{N} - 1 \tag{2.180}$$

$$\boldsymbol{y}_k = \mathbf{C}_k \boldsymbol{x}_k + \mathbf{D}_k \boldsymbol{u}_k \qquad k = 0 \dots \bar{N} \tag{2.181}$$

$$\begin{aligned} \boldsymbol{z}_k &= \mathbf{C}_k \boldsymbol{x}_k + \mathbf{D}_k \boldsymbol{u}_k + \mathbf{G}_k \boldsymbol{v}_k \\ &= \boldsymbol{y}_k + \mathbf{G}_k \boldsymbol{v}_k \end{aligned} \qquad k = 0 \dots \bar{N} \tag{2.182}$$

The two noise terms are assumed to be white, zero-mean, uncorrelated with known covariance matrices

$$\mathrm{E}[\boldsymbol{w}_k] = \mathbf{0} \tag{2.183}$$

$$\mathrm{E}[\boldsymbol{v}_k] = \mathbf{0} \tag{2.184}$$

$$\mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_{k+l}] = \mathbf{Q}_k \delta_l \tag{2.185}$$

$$\mathrm{Cov}[\boldsymbol{v}_k, \boldsymbol{v}_{k+l}] = \mathbf{R}_k \delta_l \tag{2.186}$$

$$\mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{v}_{k+l}] = \mathbf{0} \tag{2.187}$$

where δ_l is the Kronecker delta with

$$\delta_l = \begin{cases} 1 & \text{if } l = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.188}$$

Usually, it is assumed that the inputs to above system (2.180) - (2.182) are perfectly known. The case, where the input measurements are distorted by white, zero-mean noise of known covariance, can easily be covered, by treating it as an additional process noise source.

Furthermore, the mean and covariance of the initial condition estimate is assumed to be available, with

$$\mathrm{E}[\boldsymbol{x}_0] = \bar{\boldsymbol{x}}_0 \tag{2.189}$$

$$\mathrm{Cov}[\boldsymbol{x}_0] = \mathbf{P}_0^{\mathrm{x}} \tag{2.190}$$

It is assumed to be statistically independent of the noise processes ($\mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{x}_0] = \mathbf{0}$, $\mathrm{Cov}[\boldsymbol{v}_k, \boldsymbol{x}_0] = \mathbf{0}\ \forall k = 0 \dots \bar{N}$).

Since we are mostly dealing with sampled data, above discrete time system description is well suited. However, a similar system may be formulated in continuous time

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}(t)\,\boldsymbol{x}(t) + \mathbf{B}(t)\,\boldsymbol{u}(t) + {}^{c}\mathbf{F}(t)\,{}^{c}\boldsymbol{w}(t) \tag{2.191}$$

$$\boldsymbol{y}(t) = \mathbf{C}(t)\,\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t) \tag{2.192}$$

$$\boldsymbol{z}_k = \boldsymbol{y}(t_k) + \mathbf{G}_k\boldsymbol{v}_k \qquad k = 0 \dots \bar{N} \tag{2.193}$$

$$\mathrm{E}[{}^{c}\boldsymbol{w}(t)] = \mathbf{0} \tag{2.194}$$

$$\mathrm{Cov}[{}^{c}\boldsymbol{w}(t)\,, {}^{c}\boldsymbol{w}(t+\tau)] = {}^{c}\mathbf{Q}(t)\,\delta(\tau) \tag{2.195}$$

where $\delta(\tau)$ is the Dirac impulse. It can be approximated in discrete time using the following results (see Appendix D.1 for details)

$$\boldsymbol{\Phi}_k = \exp(\mathbf{A}_k \cdot (t_{k+1} - t_k)) = \sum_{j=0}^{\infty} \frac{(\mathbf{A}_k \Delta t_k)^j}{j!} \tag{2.196}$$

$$\boldsymbol{\Gamma}_k = \int_0^{\Delta t_k} \exp(\mathbf{A}_k \cdot \tau)\, d\tau \mathbf{B}_k = \sum_{j=0}^{\infty} \frac{\mathbf{A}_k^j \Delta t_k^{j+1}}{(j+1)!} \mathbf{B}_k \tag{2.197}$$

Above representation of continuous time white noise is approximated with a discrete time, white process noise term with

$$\mathrm{E}[\boldsymbol{w}_k] = 0 \tag{2.198}$$

$$\mathrm{Cov}\left[\boldsymbol{w}_k \boldsymbol{w}_{k+l}^{\intercal}\right] \approx \Delta t_k\, {}^{c}\mathbf{Q}(t_k)\,\delta_l \tag{2.199}$$

$$\mathbf{F}_k = {}^{c}\mathbf{F}(t_k) \tag{2.200}$$

For details, see Appendix D.1. The above aims at approximating the effect of continuous time white noise ${}^{c}\boldsymbol{w}$ on the continuous system, with a suitable discrete time white noise sequence $\boldsymbol{w}_k$ acting on the corresponding discrete time system. It may be interpreted as follows: if the sampling time Δt_k grows, the continuous time white noise has "more time to influence the system", i.e. the outcome will be somewhat "more uncertain". This is reflected in the scaling of the discrete time covariance matrix $\mathrm{Cov}\left[\boldsymbol{w}_k \boldsymbol{w}_{k+l}^{\intercal}\right]$ with the sampling time Δt_k. Thus it is ensured that the transition from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$ involves more uncertainty, the longer the considered time step.

A similar approach leads to relations for the measurement noise covariance matrices for discrete and continuous time [CJ2012, Ch. 5.4]. This is not pursued further, since only sampled outputs and thus only discrete time measurement noise is to be considered. For the explicit treatment of continuous time filters, see [Jaz1970, Sim2006, CJ2012], or the original work by KALMAN and BUCY [KB1961].

2.3.2.2 Propagation

The first step during a Kalman filter iteration is to predict the current state estimate $\hat{\boldsymbol{x}}_{k|k}$ and corresponding covariance matrix $\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}$. This prediction is done by taking the conditional expectation on both sides of the linear system equation (2.180)

$$\begin{aligned} \mathrm{E}[\boldsymbol{x}_{k+1}|\mathbf{Z}_k] &= \boldsymbol{\Phi}_k \mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_k] + \boldsymbol{\Gamma}_k \mathrm{E}[\boldsymbol{u}_k|\mathbf{Z}_k] + \mathbf{F}_k \mathrm{E}[\boldsymbol{w}_k|\mathbf{Z}_k] \\ &= \boldsymbol{\Phi}_k \mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_k] + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \end{aligned} \tag{2.201}$$

The second equality is due to the fact that the inputs $\boldsymbol{u}_k$ (as well as all system matrices) are considered to be deterministic, and the process noise $\boldsymbol{w}_k$ has zero mean. Thus the state propagation is

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \tag{2.202}$$

This can also be interpreted as using the "best available" estimate of the unknown quantities: for the state $\boldsymbol{x}_k$ this would be its corrected estimate $\hat{\boldsymbol{x}}_{k|k}$, the input is assumed to be known, and the best available guess for the process noise $\boldsymbol{w}_k$ is its mean $\mathrm{E}[\boldsymbol{w}_k] = \mathbf{0}$ [MK2016, Ch. 4].

The estimation error during propagation can be quantified as

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1|k} &= \hat{\boldsymbol{x}}_{k+1|k} - \boldsymbol{x}_{k+1} = \\ &= \left(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right) - (\boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \mathbf{F}_k \boldsymbol{w}_k) \\ &= \boldsymbol{\Phi}_k \tilde{\boldsymbol{x}}_{k|k} - \mathbf{F}_k \boldsymbol{w}_k \end{aligned} \tag{2.203}$$

Thus, if it held that $\mathrm{E}\left[\tilde{\boldsymbol{x}}_{k|k}\right] = \mathbf{0}$, then above propagation equation guarantees that the propagated error has zero-mean $\mathrm{E}\left[\tilde{\boldsymbol{x}}_{k+1|k}\right] = \mathbf{0}$. This can again be shown by applying the expectation operator to both sides of the equation. The circumstances, under which the Kalman filter yields estimates with zero-mean error, are illustrated in the next section.

From above equation, the propagation of the corresponding covariance matrices can be computed straight forward [MK2016, Ch. 4]

$$\begin{aligned} \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} &= \mathrm{Cov}\left[\boldsymbol{\Phi}_k \tilde{\boldsymbol{x}}_{k|k} - \mathbf{F}_k \boldsymbol{w}_k \middle| \mathbf{Z}_k\right] \\ &= \boldsymbol{\Phi}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k} \middle| \mathbf{Z}_k\right] \boldsymbol{\Phi}_k^\intercal + \mathbf{F}_k \mathrm{Cov}[\boldsymbol{w}_k|\mathbf{Z}_k] \, \mathbf{F}_k^\intercal \\ &- \boldsymbol{\Phi}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k}, \boldsymbol{w}_k \middle| \mathbf{Z}_k\right] \mathbf{F}_k^\intercal - \mathbf{F}_k \mathrm{Cov}\left[\boldsymbol{w}_k, \tilde{\boldsymbol{x}}_{k|k} \middle| \mathbf{Z}_k\right] \boldsymbol{\Phi}_k^\intercal \end{aligned} \tag{2.204}$$

Since the error at time-step k can at maximum contain influences of the (white) process noise up to time-step $k-1$ ($\boldsymbol{w}_k$ "hasn't had the time to act yet", see equation (2.180)). the cross variances in above equation vanish $\mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k}, \boldsymbol{w}_k \middle| \mathbf{Z}_k\right] = \mathrm{Cov}\left[\boldsymbol{w}_k, \tilde{\boldsymbol{x}}_{k|k} \middle| \mathbf{Z}_k\right]^\intercal = \mathbf{0}$. Thus the covariance propagation becomes

$$\begin{aligned} \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} &= \boldsymbol{\Phi}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k} \middle| \mathbf{Z}_k\right] \boldsymbol{\Phi}_k^\intercal + \mathbf{F}_k \mathrm{Cov}[\boldsymbol{w}_k|\mathbf{Z}_k] \, \mathbf{F}_k^\intercal \\ &= \boldsymbol{\Phi}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \boldsymbol{\Phi}_k^\intercal + \tilde{\mathbf{Q}}_k \end{aligned} \tag{2.205}$$

with the abbreviation

$$\tilde{\mathbf{Q}}_k = \mathbf{F}_k \mathbf{Q}_k \mathbf{F}_k^\intercal \tag{2.206}$$

JAZWINSKI notes on the form of the covariance that it doesn't explicitly depend on $\mathbf{Z}_k$, which is why some authors equivalently use the covariance $\mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k}\right]$, instead of its conditional form $\mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k}\middle|\mathbf{Z}_k\right]$ [Jaz1970, Ch. 7].

2.3.2.3 Correction

Whenever a measurement becomes available, the current state estimate can be corrected. This does not have to happen at every sampling instant, however, the extension of the following derivation to the more general case based on $\hat{\boldsymbol{x}}_{k+l|k}$ $l > 1$ is straight forward.

The basic idea is to correct the state using linear feedback of the residuals, i.e. the difference between the estimated model output, and the actual measurements

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}\right) \tag{2.207}$$

Using equations (2.181) and (2.182) for measurements and estimated output, the estimation error becomes

$$\begin{aligned}
\tilde{\boldsymbol{x}}_{k+1|k+1} &= \hat{\boldsymbol{x}}_{k+1|k+1} - \boldsymbol{x}_{k+1} \\
&= \hat{\boldsymbol{x}}_{k+1|k} - \boldsymbol{x}_{k+1} \\
&+ \mathbf{K}_{k+1}\Big(\underbrace{\mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1}}_{=\boldsymbol{z}_{k+1}} - \underbrace{\left(\mathbf{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}\right)}_{=\hat{\boldsymbol{y}}_{k+1|k}}\Big) \\
&= \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\tilde{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\mathbf{G}_{k+1}\boldsymbol{v}_{k+1}
\end{aligned} \tag{2.208}$$

Based on above equation, its covariance matrix is

$$\begin{aligned}
\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1} &= \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)^\intercal + \mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\mathbf{K}_{k+1}^\intercal \\
\text{with} \qquad \tilde{\mathbf{R}}_{k+1} &= \mathbf{G}_{k+1}\mathbf{R}_{k+1}\mathbf{G}_{k+1}^\intercal
\end{aligned} \tag{2.209}$$

Again, the fact was used that $\mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k+1|k}, \boldsymbol{v}_{k+1}\middle|\mathbf{Z}_k\right] = \mathrm{Cov}\left[\boldsymbol{v}_{k+1}, \tilde{\boldsymbol{x}}_{k+1|k}\middle|\mathbf{Z}_k\right]^\intercal = \mathbf{0}$, since the white measurement noise at time $k+1$ has not yet had a chance to act on the error $\tilde{\boldsymbol{x}}_{k+1|k}$.

For a complete solution of the filtering problem, the gain matrix $\mathbf{K}_{k+1}$ needs to be determined. The reasoning here is based on minimizing the sum of the error variances for the elements of the state vector. These are exactly the main diagonal elements of the covariance matrix $\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1}$, the sum of which can be expressed as its trace

$$\min_{\mathbf{K}_{k+1}} \sum_{j=1}^{n_x} \mathrm{Cov}\left[\left[\tilde{\boldsymbol{x}}_{k+1|k+1}\right]_{(j)}\middle|\mathbf{Z}_{k+1}\right] = \min_{\mathbf{K}_{k+1}} \mathrm{tr}\left[\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1}\right] \tag{2.210}$$

The first order necessary conditions for optimality (2.6) can then be used to determine $\mathbf{K}_{k+1}$. Some matrix calculus results of section A.1 are necessary, especially equations (A.11) and (A.13)

$$\begin{aligned}
&\frac{\partial \mathrm{tr}\left[\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1}\right]}{\partial \mathbf{K}_{k+1}} = \\
&= \frac{\partial}{\partial \mathbf{K}_{k+1}} \mathrm{tr}\left[\left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)^{\intercal} + \mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\mathbf{K}^{\intercal}_{k+1}\right] \qquad (2.211)\\
&= -\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + 2\mathbf{K}_{k+1}\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + 2\mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\\
&= -2\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + 2\mathbf{K}_{k+1}\left(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + \tilde{\mathbf{R}}_{k+1}\right) \overset{!}{=} \mathbf{0}\\
\Rightarrow &\mathbf{K}_{k+1} = \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1}\left(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + \tilde{\mathbf{R}}_{k+1}\right)^{-1} \qquad (2.212)
\end{aligned}$$

An intuitive interpretation for this form of gain is deferred to the end of section 2.3.3.1. There, a different, but algebraically equivalent, form for the correction step will be obtained based on maximum likelihood considerations. This form will be easier to interpret.

The result, after substituting $\mathbf{K}_{k+1}$ back into equation (2.209) can be formulated in several different ways, all of which are algebraically equivalent, see [CJ2012, Ch. 5]

$$\begin{aligned}
\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1} &= \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)^{\intercal} + \mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\mathbf{K}^{\intercal}_{k+1}\\
&= \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\\
&= \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1}\left(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}^{\intercal}_{k+1} + \tilde{\mathbf{R}}_{k+1}\right)^{-1}\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} \qquad (2.213)\\
&= \left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1} + \mathbf{C}^{\intercal}_{k+1}\tilde{\mathbf{R}}^{-1}_{k+1}\mathbf{C}_{k+1}\right)^{-1}
\end{aligned}$$

However, the first form is usually preferred: for symmetric, and positive definite $\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}$ it also yields a symmetric and positive definite $\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1}$ per construction, while keeping the number of matrix inversions small. The other formulations are either computationally more expensive, or their positive definiteness cannot be guaranteed in the presence of numerical inaccuracies.

In order to investigate the properties of the estimation error, its propagation equation needs to be determined. Combining the corrected estimation error (2.208) with its propagation equation (2.203) yields the one-step propagated $\tilde{\boldsymbol{x}}_{k+1|k}$ and corrected $\tilde{\boldsymbol{x}}_{k+1|k+1}$ estimation errors

$$\tilde{\boldsymbol{x}}_{k+1|k} = \boldsymbol{\Phi}_k(\mathbf{I}_{n_x} - \mathbf{K}_k\mathbf{C}_k)\,\tilde{\boldsymbol{x}}_{k|k-1} + \boldsymbol{\Phi}_k\mathbf{K}_k\mathbf{G}_k\boldsymbol{v}_k - \mathbf{F}_k\boldsymbol{w}_k \qquad (2.214)$$

$$\tilde{\boldsymbol{x}}_{k+1|k+1} = \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\boldsymbol{\Phi}_k\tilde{\boldsymbol{x}}_{k|k} - \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{F}_k\boldsymbol{w}_k + \mathbf{K}_{k+1}\mathbf{G}_{k+1}\boldsymbol{v}_{k+1} \qquad (2.215)$$

It was initially assumed, that an unbiased estimate of the initial state $\bar{\boldsymbol{x}}_0$ was available, which may now be used as[9]

$$\hat{\boldsymbol{x}}_{0|-1} = \bar{\boldsymbol{x}}_0 \qquad (2.216)$$

[9] the notation $\hat{\boldsymbol{x}}_{0|-1}$ implies the state estimate *before* any measurements are available [Sor1980, Ch. 1].

Then, the mean of the initial estimation error is

$$\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{0|-1}\Big] = \mathrm{E}\Big[\hat{\boldsymbol{x}}_{0|-1} - \boldsymbol{x}_0\Big] = \mathrm{E}[\bar{\boldsymbol{x}}_0] - \mathrm{E}[\boldsymbol{x}_0] = \mathbf{0} \tag{2.217}$$

By applying the expectation operator to both sides of equation (2.214) it can be shown that the estimation error will be zero-mean for all times. If no unbiased estimate is available, the mean of the estimation error will approach zero asymptotically, given the system $\boldsymbol{\Phi}_k(\mathbf{I}_{n_x} - \mathbf{K}_k\mathbf{C}_k)$ is stable, i.e. has only eigenvalues within the complex unit circle. This is usually given for the systems under consideration.

For proper initialization of the algorithm, a corresponding initial state covariance matrix needs to be given, too

$$\mathbf{P}^{\tilde{\mathrm{x}}}_{0|-1} = \mathrm{Cov}\Big[\tilde{\boldsymbol{x}}_{0|-1}\Big] = \mathbf{P}^{\mathrm{x}}_0 \tag{2.218}$$

Then the correction scheme of this section may be used to consider an initial measurement $\boldsymbol{z}_0$, in order to correct the initial estimate $\hat{\boldsymbol{x}}_{0|-1}$ with $\mathbf{P}^{\tilde{\mathrm{x}}}_{0|-1}$ to eventually start the iteration with $\hat{\boldsymbol{x}}_{0|0}$ and $\mathbf{P}^{\tilde{\mathrm{x}}}_{0|0}$. If no initial measurement is available, the initial state estimate may be used as corrected version, which may be interpreted as "infinitely uncertain" measurement, yielding $\mathbf{K}_0 = \mathbf{0}$.

The Algorithm is summed up in Algorithm 2.4.

2.3.2.4 Characteristics

The Kalman filter as presented here exhibits some interesting characteristics [Sim2006, Ch. 5.2]. Given the system (2.180) - (2.182) is the true description of the process under investigation, the following hold

- if the noise processes $\boldsymbol{w}$ and $\boldsymbol{v}$ are zero-mean, uncorrelated, and white then the Kalman filter is the best linear filter in the sense, that it minimizes the mean squared estimation error $\tilde{\boldsymbol{x}}_{k|k}$. *Linear* in this context means, that it is a linear combination of the measurements.
- In above derivations, no assumption about the shape of the noise processes' probability density function (pdf) was necessary, i.e. they also hold for non-Gaussian noise.
- if the noise processes $\boldsymbol{w}$ and $\boldsymbol{v}$ are Gaussian, then estimation errors and state estimates are Gaussian, too, since they arise by linear combinations of Gaussian random variables.
- one special form of residual, which is called *innovation* in the Kalman filter context is based on the predicted outputs

$$\boldsymbol{r}_{k+1|k} = \boldsymbol{z}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k} \tag{2.219}$$

Algorithm 2.4: Linear, discrete time Kalman filter

Initialization:

I. Initialize state and covariance estimates

$$\begin{aligned}\hat{\boldsymbol{x}}_{0|-1} &= \bar{\boldsymbol{x}}_0\\ \mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1} &= \mathbf{P}^{\mathrm{x}}_0\end{aligned}$$

II. if an initial measurement is to be considered (i.e. $\tilde{\mathbf{R}}_0$ is finite), perform initial correction step

$$\begin{aligned}\mathbf{K}_0 &= \mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\mathbf{C}_0^\intercal\Big(\mathbf{C}_0\mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\mathbf{C}_0^\intercal + \tilde{\mathbf{R}}_0\Big)^{-1}\\ \mathbf{P}^{\tilde{\mathbf{x}}}_{0|0} &= (\mathbf{I}_{n_x} - \mathbf{K}_0\mathbf{C}_0)\,\mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\,(\mathbf{I}_{n_x} - \mathbf{K}_0\mathbf{C}_0)^\intercal + \mathbf{K}_0\tilde{\mathbf{R}}_0\mathbf{K}_0^\intercal\\ \hat{\boldsymbol{x}}_{0|0} &= (\mathbf{I}_{n_x} - \mathbf{K}_0\mathbf{C}_0)\,\hat{\boldsymbol{x}}_{0|-1} + \mathbf{K}_0(\boldsymbol{z}_0 - \mathbf{D}_0\boldsymbol{u}_0)\end{aligned}$$

otherwise set $\hat{\boldsymbol{x}}_{0|0} = \hat{\boldsymbol{x}}_{0|-1}$, and $\mathbf{P}^{\tilde{\mathbf{x}}}_{0|0} = \mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}$

Main Part:

III. prediction step

$$\begin{aligned}\hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k\boldsymbol{u}_k\\ \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} &= \boldsymbol{\Phi}_k\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\boldsymbol{\Phi}_k^\intercal + \tilde{\mathbf{Q}}_k\end{aligned}$$

IV. correction step

(a) compute Gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^\intercal\Big(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^\intercal + \tilde{\mathbf{R}}_{k+1}\Big)^{-1}$$

(b) correct estimates

$$\begin{aligned}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1} &= (\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1})\,\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\,(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1})^\intercal + \mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\mathbf{K}_{k+1}^\intercal\\ \hat{\boldsymbol{x}}_{k+1|k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\Big(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}\Big)\end{aligned}$$

V. increment k and iterate from step III. while $k \leq \bar{N} - 1$

These innovations describe the part of the measurement, that cannot yet be explained by the filter and thus contains new information. In Appendix D.2 it is shown, that they are zero-mean and white, with auto-covariance matrix

$$\mathrm{E}\Big[\boldsymbol{r}_{k+1|k}\Big] = \mathbf{0} \tag{2.220}$$

$$\mathrm{Cov}\Big[\boldsymbol{r}_{k+1|k}, \boldsymbol{r}_{j+1|j}^\intercal\Big] = \delta_{k-j}\Big(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^\intercal + \tilde{\mathbf{R}}_{k+1}\Big) \tag{2.221}$$

Especially this last property of the Kalman filter is quite important in the context of parameter estimation: if the innovations $\boldsymbol{r}_{k+1|k}$ are used as the residuals $\boldsymbol{r}_{k+1}$ in an estimation algorithm (as e.g. depicted in section 2.2.2.4), due to their whiteness and zero-mean, the Markov-Criterion (2.86) holds. Then the application of the maximum likelihood algorithms of section 2.2.2.4 becomes possible, with

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_{k+1|k} = \boldsymbol{z}_{k+1} - \hat{\boldsymbol{y}}_{k|k-1} \tag{2.222}$$

$$\mathrm{Cov}\Big[\boldsymbol{r}_{k+1|k}\Big] = \mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^\intercal + \tilde{\mathbf{R}}_{k+1} = \boldsymbol{\mathcal{B}}_{k+1} \tag{2.223}$$

A simple, second order system is illustrated in Example 2.6 and the accompanying figure 2.4 to be found on pages 101 and 102. Additionally, the result of the respective Rauch-Tung-Striebel (RTS) smoother is illustrated, which will be discussed in the next sections.

2.3.3 Linear Maximum Likelihood State Estimation

For offline applications it can be interesting, to not only consider the measurements up to a time t_{k+1} for correction, but to consider all available data up to the final time $t_{\bar{N}}$, i.e. base the state estimation on

$$p(\boldsymbol{x}_{k+1}|\mathbf{Z}_{\bar{N}}) \tag{2.224}$$

which is then called "smoothing" [Jaz1970]. In this approach, all of the available data is used to improve the state estimates at all times. A solution for such a state estimation problem, which can be easily implemented on a digital computer, was presented by RAUCH, TUNG, and STRIEBEL [RTS1965]. It is still widely used today. They base their algorithm on the method of maximum likelihood and manipulations of above probability density according to Bayes' law (A.108). At the same time, COX obtained similar results based on a dynamic programming approach and the explicit solution of a two point boundary value problem for linear systems [Cox1963, Cox1964].

Modern treatment of the topic is based on considering two filters, running forward and backward in time, and optimally combining their results (see [Sim2006, CJ2012]). Here, the maximum likelihood approach is illustrated as in the original publications [RTS1965, Jaz1970]. This is done, because it fits nicely with the results derived in chapter 2.2, despite the significant drawback of any maximum likelihood approach, as mentioned by JAZWINSKI: "We note that maximum likelihood estimation is of questionable value unless the density function is unimodal and concentrated near the mode." [Jaz1970, p. 157]. For the applications considered here, this is given, but for other fields this basic assumption might be questioned.

2.3.3.1 Maximum Likelihood Filtering

In order to illustrate some of the main principles employed by RAUCH et al. the filtering step is re-derived, based on maximum likelihood considerations [RTS1965]. For the solution of the smoothing problem, see the following section 2.3.3.2.

Instead of minimizing the trace of the state error covariance matrix as is done in equation (2.210), consider the filtered estimate to be that value, which maximizes $p(\boldsymbol{x}_{k+1}|\mathbf{Z}_{k+1})$. That is the filtered estimate is that state value $\hat{\boldsymbol{x}}_{k+1|k+1}$, which is the most likely given the observations $\mathbf{Z}_{k+1}$ up to time $k+1$. Similar considerations as illustrated in section 2.2.2 on maximum likelihood estimation, together with Bayes' Theorem lead

to

$$\begin{aligned}\hat{\boldsymbol{x}}_{k+1|k+1} &= \arg\max_{\boldsymbol{x}_{k+1}} p(\boldsymbol{x}_{k+1}|\mathbf{Z}_{k+1}) = \arg\min_{\boldsymbol{x}_{k+1}} -\ln p(\boldsymbol{x}_{k+1}|\mathbf{Z}_{k+1}) \\ &= \arg\min_{\boldsymbol{x}_{k+1}} J(\boldsymbol{x}_{k+1})\end{aligned} \tag{2.225}$$

$$\begin{aligned}J(\boldsymbol{x}_{k+1}) &= -\ln p(\boldsymbol{x}_{k+1}|\mathbf{Z}_{k+1}) = -\ln \frac{p(\boldsymbol{x}_{k+1}, \mathbf{Z}_{k+1})}{p(\mathbf{Z}_{k+1})} \\ &= -\ln p(\boldsymbol{x}_{k+1}, \mathbf{Z}_{k+1}) + \ln p(\mathbf{Z}_{k+1})\end{aligned} \tag{2.226}$$

In order to determine an optimal value of $\boldsymbol{x}_{k+1}$, the second term in above cost function may be dropped, since it does not depend on $\boldsymbol{x}_{k+1}$. Using the definition of conditional probability densities (A.106), above joint probability density can be further modified to read [RTS1965]

$$\begin{aligned}p(\boldsymbol{x}_{k+1}, \mathbf{Z}_{k+1}) &= p(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \mathbf{Z}_k) = p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}, \mathbf{Z}_k)\, p(\boldsymbol{x}_{k+1}, \mathbf{Z}_k) \\ &= p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}, \mathbf{Z}_k)\, p(\boldsymbol{x}_{k+1}|\mathbf{Z}_k)\, p(\mathbf{Z}_k)\end{aligned} \tag{2.227}$$

This can be plugged in the expression for the cost function.

Then it is used that the $\boldsymbol{z}_{k+1}$ given $\boldsymbol{x}_{k+1}$ are statistically independent of past measurements $\mathbf{Z}_k$, i.e.

$$p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}, \mathbf{Z}_k) = p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}) \tag{2.228}$$

Above can be seen from the measurement equation (2.182)

$$\boldsymbol{z}_k = \mathbf{C}_k \boldsymbol{x}_k + \mathbf{D}_k \boldsymbol{u}_k + \mathbf{G}_k \boldsymbol{v}_k$$

If the states $\boldsymbol{x}_k$ are given, and the measurement noise is white, there remains nothing that could depend on $\mathbf{Z}_k$. Additionally, $p(\mathbf{Z}_k)$ does not offer any more information on $\boldsymbol{x}_{k+1}$. Thus, the cost function to minimize is

$$J(\boldsymbol{x}_{k+1}) = -\ln p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}) - \ln p(\boldsymbol{x}_{k+1}|\mathbf{Z}_k) \tag{2.229}$$

Now, assume that both of the above probability density functions describe Gaussian random variables, then they can be characterized by their mean and covariance values. This assumption is justified, if all involved processes are Gaussian, since the dynamics are linear [10].

The first term of above equation can be characterized by using the system's output equation (2.182), retaining the as of yet unknown $\boldsymbol{x}_{k+1}$ [RTS1965]

$$\begin{aligned}\mathrm{E}[\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}] &= \mathrm{E}[\mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1}|\boldsymbol{x}_{k+1}] \\ &= \mathbf{C}_{k+1}\mathrm{E}[\boldsymbol{x}_{k+1}|\boldsymbol{x}_{k+1}] + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} + \mathbf{G}_{k+1}\mathrm{E}[\boldsymbol{v}_{k+1}] \\ &= \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}\end{aligned} \tag{2.230}$$

[10] see also the part about Gaussian variables in the Appendix, especially, equations (A.132) and (A.120)

The corresponding covariance matrix is

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}] &= \mathrm{E}\left[\underbrace{(\mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} - \mathrm{E}[\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}])}_{=\beta} \cdot \beta^{\intercal} \middle| \boldsymbol{x}_{k+1}\right] \\ &= \mathbf{G}_{k+1}\mathrm{E}\left[\boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^{\intercal}\right]\mathbf{G}_{k+1}^{\intercal} \\ &= \mathbf{G}_{k+1}\mathbf{R}_{k+1}\mathbf{G}_{k+1}^{\intercal} = \tilde{\mathbf{R}}_{k+1}\end{aligned} \tag{2.231}$$

With this, the distribution of $\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}$ is

$$\mathbf{z}_{\mathrm{k+1}}|\mathbf{x}_{\mathrm{k+1}} \sim \mathcal{N}\left(\mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}, \tilde{\mathbf{R}}_{k+1}\right) \tag{2.232}$$

JAZWINSKI argues a little differently to obtain the same result: He uses the transformation of probability densities (A.93) and the fact that $\boldsymbol{v}_{k+1}$ is Gaussian to arrive at [Jaz1970, Ch. 5]

$$\mathbf{G}_{k+1}\boldsymbol{v}_{k+1} = \boldsymbol{z}_{k+1} - \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} \tag{2.233}$$

$$p_{\mathbf{z}_{\mathrm{k+1}}|\mathbf{x}_{\mathrm{k+1}}}(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}) \overset{(A.93)}{=} p_{\mathbf{G}_{\mathrm{k+1}}\mathbf{v}_{\mathrm{k+1}}}(\boldsymbol{z}_{k+1} - \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}) \tag{2.234}$$

which is the same result as illustrated above: now the distribution of $\mathbf{z}_{\mathrm{k+1}}|\mathbf{x}_{\mathrm{k+1}}$ is directly defined via the probability density $p_{\mathbf{z}_{\mathrm{k+1}}|\mathbf{x}_{\mathrm{k+1}}}(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1})$ instead of mean and covariance $\mathcal{N}\left(\mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}, \tilde{\mathbf{R}}_{k+1}\right)$.

The second conditional probability density in the cost function (2.229) describes the propagated state. Its mean and covariance are, with the definitions in section 2.3.1

$$\mathrm{E}[\boldsymbol{x}_{k+1}|\mathbf{Z}_k] = \hat{\boldsymbol{x}}_{k+1|k} \tag{2.235}$$

$$\mathrm{Cov}[\boldsymbol{x}_{k+1}|\mathbf{Z}_k] = \mathbf{P}_{k+1|k}^{\tilde{\mathbf{x}}} \tag{2.236}$$

$$\mathbf{x}_{\mathrm{k+1}}|\mathbf{Z}_{\mathrm{k}} \sim \mathcal{N}\left(\hat{\boldsymbol{x}}_{k+1|k}, \mathbf{P}_{k+1|k}^{\tilde{\mathbf{x}}}\right) \tag{2.237}$$

Together with the definition of the multi-dimensional Gaussian distribution (A.120), the non-constant parts of above cost function (2.229) eventually evaluate to

$$\begin{aligned}J(\boldsymbol{x}_{k+1}) &= -\ln p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}) - \ln p(\boldsymbol{x}_{k+1}|\mathbf{Z}_k) = \ldots \\ &= \frac{1}{2}(\boldsymbol{z}_{k+1} - \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1})^{\intercal}\,\tilde{\mathbf{R}}_{k+1}^{-1}(\boldsymbol{z}_{k+1} - \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1}) \\ &+ \frac{1}{2}\left(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}\right)^{\intercal}\left(\mathbf{P}_{k+1|k}^{\tilde{\mathbf{x}}}\right)^{-1}\left(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}\right) + c\end{aligned} \tag{2.238}$$

where c is a constant that does not depend on $\boldsymbol{x}_{k+1}$.

It is interesting to note that above cost function formulation can also be obtained, if the propagated state is considered as prior knowledge in a weighted least-squares problem, as was illustrated in section 2.2.4.6. The model that corresponds to above

quadratic cost function is

$$\underbrace{\begin{bmatrix} \hat{\boldsymbol{x}}_{k+1|k} \\ \boldsymbol{z}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} \end{bmatrix}}_{\boldsymbol{z}_{filt}} = \underbrace{\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{x}}} \\ \mathbf{C}_{k+1} \end{bmatrix}}_{\mathbf{X}_{filt}} \boldsymbol{x}_{k+1} + \underbrace{\begin{bmatrix} \hat{\boldsymbol{x}}_{k+1|k} - \boldsymbol{x}_{k+1} \\ \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} \end{bmatrix}}_{\boldsymbol{v}_{filt}} \tag{2.239}$$

$$\mathrm{Cov}\left[\begin{bmatrix} \hat{\boldsymbol{x}}_{k+1|k} - \boldsymbol{x}_{k+1} \\ \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} \end{bmatrix}\right] = \underbrace{\begin{bmatrix} \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}_{k+1} \end{bmatrix}}_{\mathbf{W}^{-1}_{filt}} \tag{2.240}$$

which shows the close relationship between least-squares estimation, maximum likelihood estimation, and Kalman filtering, if the noise terms are assumed to be Gaussian. Applying the weighted least-squares solution of sections 2.2.4.1 and 2.2.4.3 eventually yields the corrected state estimate

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k+1|k+1} &= \left(\mathbf{X}^{\intercal}_{filt}\mathbf{W}_{filt}\mathbf{X}_{filt}\right)^{-1}\mathbf{X}^{\intercal}_{filt}\mathbf{W}_{filt}\boldsymbol{z}_{filt} \\
&= \left(\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{C}^{\intercal}_{k+1} \end{bmatrix}\begin{bmatrix} \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}^{-1}_{k+1} \end{bmatrix}\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{x}}} \\ \mathbf{C}_{k+1} \end{bmatrix}\right)^{-1} \\
&\qquad \left(\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{C}^{\intercal}_{k+1} \end{bmatrix}\begin{bmatrix} \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{R}}^{-1}_{k+1} \end{bmatrix}\begin{bmatrix} \hat{\boldsymbol{x}}_{k+1|k} \\ \boldsymbol{z}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} \end{bmatrix}\right) \\
&= \left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1} + \mathbf{C}^{\intercal}_{k+1}\tilde{\mathbf{R}}^{-1}_{k+1}\mathbf{C}_{k+1}\right)^{-1} \\
&\qquad \left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1}\hat{\boldsymbol{x}}_{k+1|k} + \mathbf{C}^{\intercal}_{k+1}\tilde{\mathbf{R}}^{-1}_{k+1}(\boldsymbol{z}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1})\right)
\end{aligned} \tag{2.241}$$

Using some matrix identities, it can be shown that above expression is actually equal to the filtering step (2.207) using the gain matrix $\mathbf{K}_{k+1}$ from (2.212), details can be seen in Appendix D.4.

In contrast to the expression obtained in equation (2.207) before, this result is somewhat more intuitive to interpret. Consider the two limiting cases [CJ2012, Ch. 2]

- *high uncertainty in propagation* [$\tilde{\mathbf{R}}_{k+1}$ finite; $\mathbf{Q}_k$, $\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} \to \infty$]
  If the propagated result is considered to be very unreliable, the process noise covariance matrix, and consequently the propagated state covariance matrix become very "large" (in some matrix sense). This can happen for model formulations that are badly adapted to the task at hand, or for significant, unmeasurable disturbances. Then the corrected state can be approximated by

  $$\hat{\boldsymbol{x}}_{k+1|k+1} \approx \left(\mathbf{C}^{\intercal}_{k+1}\tilde{\mathbf{R}}^{-1}_{k+1}\mathbf{C}_{k+1}\right)^{-1}\left(\mathbf{C}^{\intercal}_{k+1}\tilde{\mathbf{R}}^{-1}_{k+1}(\boldsymbol{z}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1})\right) \tag{2.242}$$

  This is essentially the weighted least-squares solution to

  $$\begin{aligned} \boldsymbol{z}_{k+1} - \mathbf{D}_{k+1}\boldsymbol{u}_{k+1} &= \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} \\ \mathrm{Cov}[\mathbf{G}_{k+1}\boldsymbol{v}_{k+1}] &= \tilde{\mathbf{R}}_{k+1} \end{aligned} \tag{2.243}$$

  i.e. the state estimate is computed purely based on measurement information.

- *high uncertainty in measurement* [$\mathbf{Q}_k$, $\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}$ finite; $\tilde{\mathbf{R}}_{k+1} \to \infty$]
  If the measurements are considered to be very unreliable, i.e. if their covariance matrix is "large", the corrected state can be approximated by the propagated state

$$\hat{\boldsymbol{x}}_{k+1|k+1} \approx \hat{\boldsymbol{x}}_{k+1|k} \tag{2.244}$$

  i.e. the state estimate is solely based on the propagation as obtained by the model.

2.3.3.2 Marginal Maximum Likelihood State Estimation

The same approach as was used in the last section to re-derive the filtering equation for the Kalman filter can be used to obtain a smoothed state estimate based on all available measurements. RAUCH et al. consider an overall cost function J, which relates the state estimates to the true states. Based on the form of this cost function, several estimator formulations are possible (which, for linear systems and Gaussian noise result in the same estimates [RTS1965]). If the cost function J can be separated into a sum, where every term is only related to a state estimate at one time

$$J\left(\mathbf{X}_{\bar{N}}, \hat{\mathbf{X}}_{\bar{N}|\bar{N}}\right) = \sum_{k=0}^{\bar{N}} J\left(\boldsymbol{x}_k, \hat{\boldsymbol{x}}_{k|\bar{N}}\right) \tag{2.245}$$

then the cost function is "separable" with respect to sampling instants, and there exists the chance to obtain a recursive relation linking two time steps. Maximum likelihood arguments, as illustrated before, can then be based on $p(\boldsymbol{x}_k|\mathbf{Z}_{\bar{N}})$

$$\hat{\boldsymbol{x}}_{k|\bar{N}} = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\mathbf{Z}_{\bar{N}}) = \arg\min_{\boldsymbol{x}_k} -\ln p(\boldsymbol{x}_k|\mathbf{Z}_{\bar{N}}) \tag{2.246}$$

The solution to above problem, according to RAUCH et al. is called the "marginal maximum likelihood estimate (MMLE)" of the state. It is based on the marginal probability density $p(\boldsymbol{x}_k|\mathbf{Z}_{\bar{N}})$, as opposed to the joint probability density $p(\mathbf{X}_{\bar{N}}|\mathbf{Z}_{\bar{N}})$ of the next section [RTS1965].

In order to eventually obtain a recursive algorithm, RAUCH et al. do not only consider the conditional probability density of one point in time, but the joint conditional probability density of two consecutive points in time, where information about $\hat{\boldsymbol{x}}_{k+1|\bar{N}}$ will be assumed given [RTS1965]

$$\left\{\hat{\boldsymbol{x}}_{k|\bar{N}}, \hat{\boldsymbol{x}}_{k+1|\bar{N}}\right\} = \arg\min_{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}} -\ln p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}|\mathbf{Z}_{\bar{N}}) \tag{2.247}$$

Again, applying Bayes' Theorem yields

$$\begin{aligned}\left\{\hat{\boldsymbol{x}}_{k|\bar{N}}, \hat{\boldsymbol{x}}_{k+1|\bar{N}}\right\} &= \arg\min_{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}} \left(-\ln p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_{\bar{N}}) + \ln p(\mathbf{Z}_{\bar{N}})\right) \\ &= \arg\min_{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}} -\ln p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_{\bar{N}}) = \arg\min_{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}} J(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})\end{aligned} \tag{2.248}$$

Where it was used in the second equality that $p(\mathbf{Z}_{\bar{N}})$ does not influence the maximization w.r.t. the states and may thus be neglected. The main idea when further simplifying the joint probability density $p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_{\bar{N}})$ is then based on separating the measurements in two parts: the first part contains the measurements up to time k, whose information will eventually be contained in the filtered estimate $\hat{\boldsymbol{x}}_{k|k}$ and covariance matrix $\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}$. The information of the second part of the measurements $\{\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}}\}$ will be contained in a smoothed estimate $\hat{\boldsymbol{x}}_{k+1|\bar{N}}$. Combining the the two respective probability densities results in a cost function formulation, which can be optimized for the smoothed state estimate $\hat{\boldsymbol{x}}_{k|\bar{N}}$ at time k.

Repeated application of the definition of conditional probabilities (A.106) yields

$$\begin{aligned} p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_{\bar{N}}) &= p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_k, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}}) \\ &= p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \mathbf{Z}_k)\, p(\mathbf{Z}_k) \\ &= p(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_k, \mathbf{Z}_k)\, p(\boldsymbol{x}_k | \mathbf{Z}_k)\, p(\mathbf{Z}_k) \end{aligned} \tag{2.249}$$

Now, RAUCH et al. argue that the joint probability density of $\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}}$ given the states $\boldsymbol{x}_k$ at time k, is independent of the measurements $\mathbf{Z}_k$ up to time k.

$$p(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_k, \mathbf{Z}_k) = p(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_k) \tag{2.250}$$

This consequently allows for the following, final modifications [RTS1965] [Jaz1970, Ch. 7]

$$\begin{aligned} p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \mathbf{Z}_{\bar{N}}) &= p(\boldsymbol{x}_{k+1}, \boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_k)\, p(\boldsymbol{x}_k | \mathbf{Z}_k)\, p(\mathbf{Z}_k) \\ &= p(\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_k, \boldsymbol{x}_{k+1})\, p(\boldsymbol{x}_{k+1} | \boldsymbol{x}_k)\, p(\boldsymbol{x}_k | \mathbf{Z}_k)\, p(\mathbf{Z}_k) \\ &= p(\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_{k+1})\, p(\boldsymbol{x}_{k+1} | \boldsymbol{x}_k)\, p(\boldsymbol{x}_k | \mathbf{Z}_k)\, p(\mathbf{Z}_k) \end{aligned} \tag{2.251}$$

The last equality is again due to the independence of $\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}}$ on $\boldsymbol{x}_k$, if conditioned on $\boldsymbol{x}_{k+1}$. The resulting cost function is then (dropping $p(\mathbf{Z}_k)$ since it does not influence the estimate)

$$J(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}) = -\ln p(\boldsymbol{x}_{k+1} | \boldsymbol{x}_k) - \ln p(\boldsymbol{x}_k | \mathbf{Z}_k) - \ln p(\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}} | \boldsymbol{x}_{k+1}) \tag{2.252}$$

Again assuming Gaussian distributions, the probability densities can be characterized by their mean and covariance alone. Namely, for the first of above terms this is (keeping the system definition (2.180) in mind) [RTS1965] [Jaz1970, Ch. 7]

$$\begin{aligned} \mathrm{E}[\boldsymbol{x}_{k+1} | \boldsymbol{x}_k] &= \mathrm{E}[\boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \mathbf{F}_k \boldsymbol{w}_k | \boldsymbol{x}_k] = \boldsymbol{\Phi}_k \mathrm{E}[\boldsymbol{x}_k | \boldsymbol{x}_k] + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \mathbf{F}_k \mathrm{E}[\boldsymbol{w}_k] \\ &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \end{aligned} \tag{2.253}$$

$$\begin{aligned} \mathrm{Cov}[\boldsymbol{x}_{k+1} | \boldsymbol{x}_k] &= \mathrm{E}\left[\underbrace{\left(\boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k + \mathbf{F}_k \boldsymbol{w}_k - \mathrm{E}[\boldsymbol{x}_{k+1} | \boldsymbol{x}_k] \right)}_{=\boldsymbol{\beta}} \cdot \boldsymbol{\beta}^{\mathsf{T}} \middle| \boldsymbol{x}_k \right] \\ &= \mathbf{F}_k \mathrm{E}[\boldsymbol{w}_k \boldsymbol{w}_k^{\mathsf{T}}]\, \mathbf{F}_k^{\mathsf{T}} = \mathbf{F}_k \mathbf{Q}_k \mathbf{F}_k^{\mathsf{T}} = \tilde{\mathbf{Q}}_k \end{aligned} \tag{2.254}$$

$$\mathbf{x}_{k+1} | \mathbf{x}_k \sim \mathcal{N}\left(\boldsymbol{\Phi}_k \boldsymbol{x}_k + \boldsymbol{\Gamma}_k \boldsymbol{u}_k, \tilde{\mathbf{Q}}_k \right) \tag{2.255}$$

Again, JAZWINSKI argues that the rule about the transformation of probability densities (A.93) may alternatively be used to arrive at

$$\mathbf{F}_k \boldsymbol{w}_k = \boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k \tag{2.256}$$

$$p_{\mathbf{x}_{k+1}|\mathbf{x}_k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \overset{(A.93)}{=} p_{\mathbf{F}_k \mathbf{w}_k}(\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k) \tag{2.257}$$

which, for Gaussian random variables, is equivalent to the result above,

The second term in above cost function describes the result of the filtered state estimate from section 2.3.3.1

$$\mathrm{E}[\boldsymbol{x}_k|\mathbf{Z}_k] = \hat{\boldsymbol{x}}_{k|k} \tag{2.258}$$

$$\mathrm{Cov}[\boldsymbol{x}_k|\mathbf{Z}_k] = \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \tag{2.259}$$

For now, we are primarily interested in a solution of the optimization problem w.r.t. $\boldsymbol{x}_k$, thus the third term does not need to be specified, since it does not depend on $\boldsymbol{x}_k$. The cost function to be minimized then becomes

$$\begin{aligned} J(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}) = &\frac{1}{2}(\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k)^\intercal \tilde{\mathbf{Q}}_k^{-1} (\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k) \\ &+ \frac{1}{2}\left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\right)^\intercal \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\right)^{-1} \left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\right) + c(\boldsymbol{z}_{k+1}, \ldots, \boldsymbol{z}_{\bar{N}}, \boldsymbol{x}_{k+1}) \end{aligned} \tag{2.260}$$

Now a recursion equation for $\hat{\boldsymbol{x}}_{k|\bar{N}}$ can be obtained. First, assume that the estimate $\hat{\boldsymbol{x}}_{k+1|\bar{N}} = \mathrm{E}[\boldsymbol{x}_{k+1}|\boldsymbol{z}_{\bar{N}}]$ is available. Then the gradient of $J\left(\boldsymbol{x}_k, \hat{\boldsymbol{x}}_{k+1|\bar{N}}\right)$ w.r.t. $\boldsymbol{x}_k$ may be set to zero

$$\begin{aligned} \mathbf{0} &\overset{!}{=} \frac{\partial J\left(\hat{\boldsymbol{x}}_{k|\bar{N}}, \hat{\boldsymbol{x}}_{k+1|\bar{N}}\right)}{\partial \hat{\boldsymbol{x}}_{k|\bar{N}}} = \\ &= -\boldsymbol{\Phi}_k^\intercal \tilde{\mathbf{Q}}_k^{-1}\left(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|\bar{N}} - \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right) + \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\right)^{-1}\left(\hat{\boldsymbol{x}}_{k|\bar{N}} - \hat{\boldsymbol{x}}_{k|k}\right) \end{aligned} \tag{2.261}$$

$$\begin{aligned} \Rightarrow \hat{\boldsymbol{x}}_{k|\bar{N}} = &\left(\boldsymbol{\Phi}_k^\intercal \tilde{\mathbf{Q}}_k^{-1} \boldsymbol{\Phi}_k + \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\right)^{-1}\right)^{-1} \\ &\left(\boldsymbol{\Phi}_k^\intercal \tilde{\mathbf{Q}}_k^{-1}\left(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right) + \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\right)^{-1} \hat{\boldsymbol{x}}_{k|k}\right) \end{aligned} \tag{2.262}$$

Realizing, that the structure of above equation is exactly the same as the maximum likelihood filtering result (2.241) of the foregoing section, a derivation completely analogous to that in Appendix D.4 can be performed. This results in the following backwards recursion in corrector form

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|\bar{N}} &= \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\left(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \left(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right)\right) \\ &\overset{(2.202)}{=} \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\left(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \hat{\boldsymbol{x}}_{k+1|k}\right) \end{aligned} \tag{2.263}$$

$$\begin{aligned} \mathbf{M}_k &= \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \boldsymbol{\Phi}_k^\intercal \left(\boldsymbol{\Phi}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \boldsymbol{\Phi}_k^\intercal + \tilde{\mathbf{Q}}_k\right)^{-1} \\ &\overset{(2.205)}{=} \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \boldsymbol{\Phi}_k^\intercal \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1} \end{aligned} \tag{2.264}$$

Algorithm 2.5: Linear Rauch-Tung-Striebel (RTS) smoother

Initialization:

I. Initialize smooth state and covariance estimates at the end of the considered batch using the Kalman Filter results

$$\hat{\boldsymbol{x}}_{\bar{N}|\bar{N}} = \mathrm{E}[\boldsymbol{x}_{\bar{N}}|\mathbf{Z}_{\bar{N}}]$$
$$\mathbf{P}^{\tilde{\mathbf{x}}}_{\bar{N}|\bar{N}} = \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{\bar{N}|\bar{N}}\right]$$

Main Part:

II. recursively compute smoothed state and covariance backwards in time

(a) compute Gain

$$\begin{aligned}\mathbf{M}_k &= \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\boldsymbol{\Phi}_k^\intercal\left(\boldsymbol{\Phi}_k\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\boldsymbol{\Phi}_k^\intercal + \tilde{\mathbf{Q}}_k\right)^{-1}\\ &= \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\boldsymbol{\Phi}_k^\intercal\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)^{-1}\end{aligned}$$

(b) correct estimates

$$\hat{\boldsymbol{x}}_{k|\bar{N}} = \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\left(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \hat{\boldsymbol{x}}_{k+1|k}\right)$$
$$\mathbf{P}^{\tilde{\mathbf{x}}}_{k|\bar{N}} = \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|\bar{N}} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)\mathbf{M}_k^\intercal$$

III. decrement k and iterate from step II. while $k >= 0$

It is interesting to note that (in contrast to pure forward filtering), the gain $\mathbf{M}_k$ does not depend on the smoother covariance, it only depends on the covariances obtained during the forward pass. This would allow to compute and store only the smoother gain during the forward pass, if only the smoothed state is of interest [CJ2012, Ch. 6.1].

Determining the smoothed state error covariance estimate is more intricate compared to the formulations discussed so far. A detailed derivation may be found in appendix D.5, where only the resulting, final backwards recursion for the smoother covariance estimate is shown here

$$\mathbf{P}^{\tilde{\mathbf{x}}}_{k|\bar{N}} = \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|\bar{N}} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\right)\mathbf{M}_k^\intercal \tag{2.265}$$

The algorithm constitutes a backwards recursion and thus needs to be initialized at the end of the considered time span. However, there the forward filter result with $\hat{\boldsymbol{x}}_{\bar{N}|\bar{N}}$ and $\mathbf{P}^{\tilde{\mathbf{x}}}_{\bar{N}|\bar{N}}$ is readily available.

A summary of the algorithm for linear, discrete time dynamic systems can be found in Algorithm 2.5. This is commonly called the RTS smoother, an example for which is given in Example 2.6.

Example 2.6: Comparison of Kalman Filter and RTS-smoother

Consider the second order, discrete dynamic system

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k + \boldsymbol{w}_k$$
$$\boldsymbol{y}_k = \mathbf{C}\boldsymbol{x}_k + \mathbf{D}\boldsymbol{u}_k + \boldsymbol{v}_k$$

with the following definitions

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 \\ K \end{bmatrix}$$
$$\mathbf{C} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 5 \\ 15 \end{bmatrix}$$
$$\boldsymbol{\Phi} = \exp(\mathbf{A}\Delta t) \qquad \boldsymbol{\Gamma} = \mathbf{A}^{-1}(\exp(\mathbf{A}\Delta t) - \mathbf{I}_{n_x})\,\mathbf{B}$$
$$\mathrm{Cov}\left[\boldsymbol{w}_k\boldsymbol{w}_{k-l}^\intercal\right] = \mathbf{Q}\delta_l \qquad \mathrm{Cov}\left[\boldsymbol{v}_k\boldsymbol{v}_{k-l}^\intercal\right] = \mathbf{R}\delta_l$$

The corresponding numerical values are

Parameter	Value	Parameter	Value	Parameter	Value
ω_0	$5\frac{rad}{s}$	t_0	$0s$	$\boldsymbol{x}_0$	$\begin{bmatrix} 3 & -20 \end{bmatrix}^\intercal$
ζ	0.2	Δt	$0.01s$	$\bar{\boldsymbol{x}}_0$	$\begin{bmatrix} 2 & -15 \end{bmatrix}^\intercal$
K	25	t_{end}	$15s$	$\mathbf{P}_0^{\mathrm{x}}$	$\mathrm{diag}\left(\begin{bmatrix} 1 & 100 \end{bmatrix}\right)$
C_1	2			$\mathbf{Q}$	$\mathrm{diag}\left(\begin{bmatrix} 0.001 & 0.1 \end{bmatrix}\right)$
C_2	3			$\mathbf{R}$	$\mathrm{diag}\left(\begin{bmatrix} 1 & 7 \end{bmatrix}\right)$

The estimation results, when subject to a doublet input, are illustrated in figure 2.4 for a linear, discrete Kalman Filter and a RTS smoother. The estimated state trajectories and estimation errors are plotted, along with the 3σ confidence bounds.

Since the model and all noise characteristics are perfectly known, both algorithms manage to track the true state trajectory very good. The RTS result is smoother, and its covariance is always lower than the Kalman filter equivalent. This illustrates the fact that it uses more samples (i.e. more information) to obtain a result. Only at the very last sample, it can be seen how the RTS smoother is initialized with the Kalman filter result, i.e. there the estimate and confidence bounds are equal.

The solution obtained by solving the quadratic problem to be illustrated in section 2.3.3.3 is not shown, since it is numerically equivalent to the RTS-solution (numerical difference is the range of 1×10^{-14}).

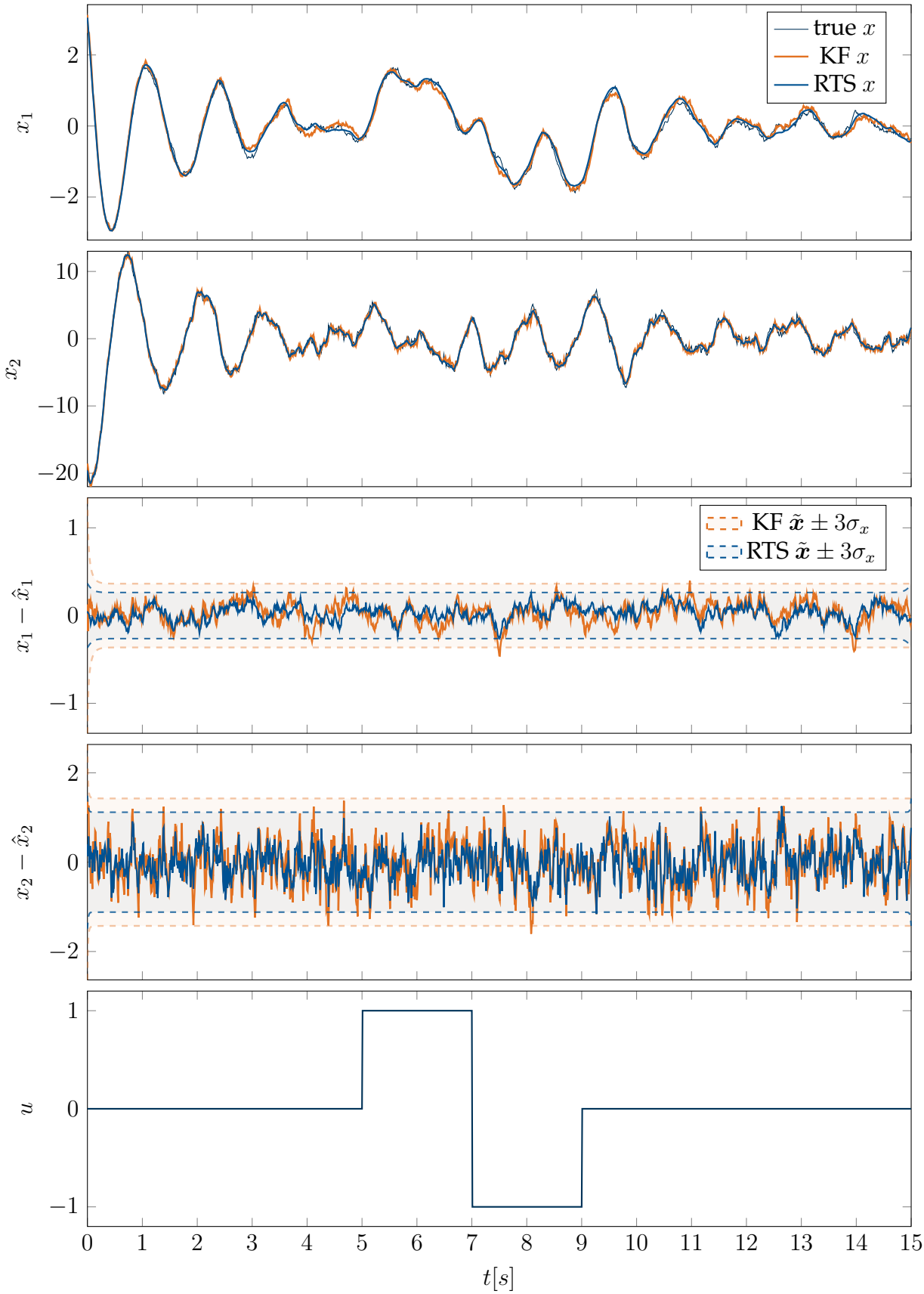


Figure 2.4: *Example for Kalman Filter and RTS-Smoother based state estimation; comparison of states and the respective estimates (subplot 1 and 2) as well as the estimation errors (subplot 3 and 4), given a doublet control input (subplot 5)*

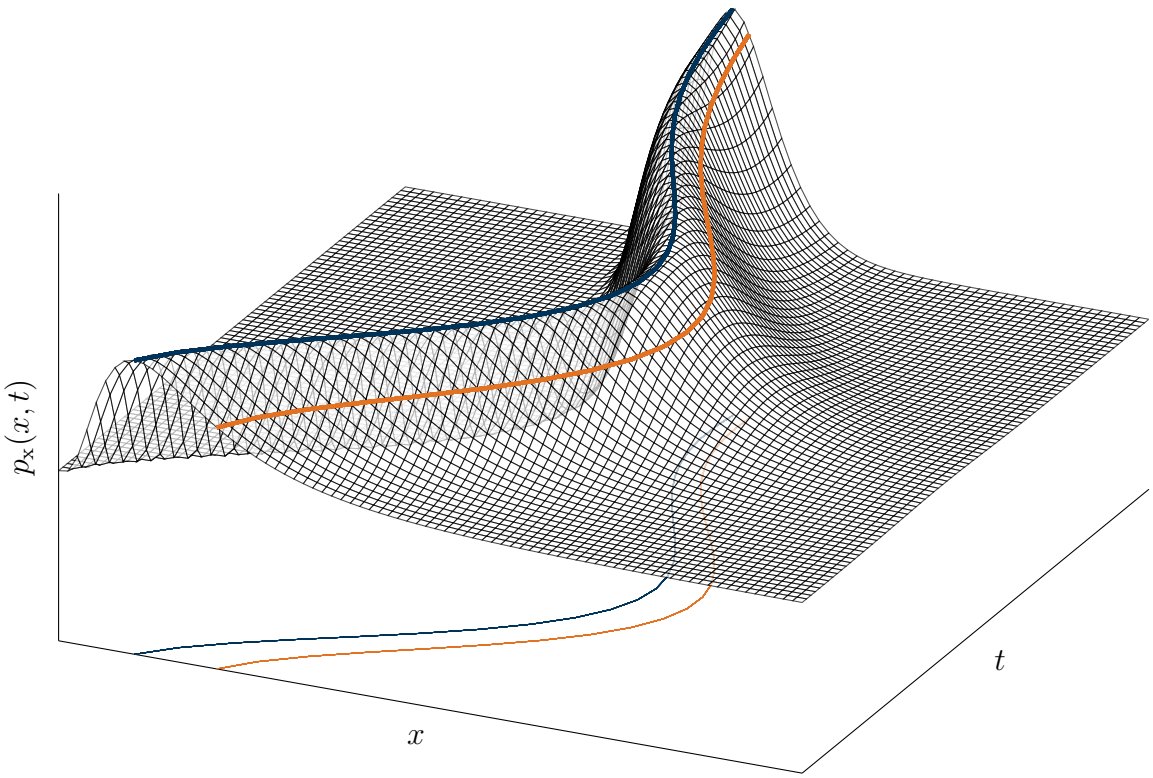


Figure 2.5: *Illustration of a modal trajectory (———) as the peak of (non-symmetric) state pdf over time; an expected value trajectory (———) is shown for comparison.*

2.3.3.3 Joint Maximum Likelihood State Estimation

In the last section, a recursive solution based on the joint conditional probability density $p(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}|\mathbf{Z}_{\bar{N}})$ was given. If the defining cost function cannot be divided into separate parts for the respective estimates, or one does not aim for such a formulation, the same arguments may be extended to the joint conditional probability density of all states, eventually resulting in joint maximum likelihood estimation (JMLE)

$$p(\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}|\mathbf{Z}_{\bar{N}}) = p(\mathbf{X}_{\bar{N}}|\mathbf{Z}_{\bar{N}}) = \frac{p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}})}{p(\mathbf{Z}_{\bar{N}})} \tag{2.266}$$

$$\begin{aligned} \left\{\hat{\boldsymbol{x}}_{0|\bar{N}} \dots \hat{\boldsymbol{x}}_{\bar{N}|\bar{N}}\right\} &= \arg\max_{\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}} p(\mathbf{X}_{\bar{N}}|\mathbf{Z}_{\bar{N}}) = \arg\min_{\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}} -\ln p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) \\ &= \arg\min_{\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}} J(\mathbf{X}_{\bar{N}}) \end{aligned} \tag{2.267}$$

The sequence $\left\{\hat{\boldsymbol{x}}_{0|\bar{N}} \dots \hat{\boldsymbol{x}}_{\bar{N}|\bar{N}}\right\}$ is sometimes called the *modal trajectory*, since it constitutes the trajectory of maxima or *modes* of the underlying pdf [Jaz1970, p. 156]. Figure 2.5 illustrates this: as the state pdf progresses over time, its mode forms a trajectory, which may be used as state estimate. In contrast, another state estimate could be based on the mean, which is also shown. In figure 2.5 the two are distinct, since the example pdf is *not* symmetric. For a Gaussian distribution the two would coincide.

Again, applying the definition of conditional probabilities (A.106) to the joint prob-

ability distribution yields [RTS1965]

$$p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) = p(\mathbf{Z}_{\bar{N}}|\mathbf{X}_{\bar{N}})\, p(\mathbf{X}_{\bar{N}}) \tag{2.268}$$

Now, if the measurement noise is considered to be white, the conditional probability density of $\boldsymbol{z}_k$ is independent, both of the states and measurements at all other times $\{\boldsymbol{z}_j, \boldsymbol{x}_j\}\, \forall j \neq k$ [Jaz1970, Ch. 5]

$$p(\mathbf{Z}_{\bar{N}}|\mathbf{X}_{\bar{N}}) = p(\boldsymbol{z}_0 \dots \boldsymbol{z}_{\bar{N}}|\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}) = \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_0 \dots \boldsymbol{x}_{\bar{N}}) = \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k) \tag{2.269}$$

Additionally, Bayes' Theorem can be recursively applied to the second part of equation (2.268), eventually resulting in

$$\begin{aligned} p(\mathbf{X}_{\bar{N}}) &= p(\boldsymbol{x}_{\bar{N}}|\mathbf{X}_{\bar{N}-1})\, p(\mathbf{X}_{\bar{N}-1}) = p(\boldsymbol{x}_{\bar{N}}|\mathbf{X}_{\bar{N}-1})\, p(\boldsymbol{x}_{\bar{N}-1}|\mathbf{X}_{\bar{N}-2})\, p(\mathbf{X}_{\bar{N}-2}) = \dots \\ &= \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}|\mathbf{X}_k) \cdot p(\boldsymbol{x}_0) = \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_0) \end{aligned} \tag{2.270}$$

The last equality arises from the assumption that all processes considered here are Markov (see Appendix A.5.4), i.e. the probability density of $\boldsymbol{x}_{k+1}$ only depends on the realization of its direct predecessor $\boldsymbol{x}_k$. The joint probability of all states and all measurements may then be expressed as

$$p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) = \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_0) \tag{2.271}$$

The expressions for above probability densities have been derived in the last sections, where $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ involves the system's output equation (see (2.230) and (2.232)), and the system's propagation equation (see (2.253) and (2.254)) is reflected in $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$. The distribution of the initial condition is assumed to be known, too $\mathbf{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \mathbf{P}_0^{\mathrm{x}})$.

Eventually, the negative log-likelihood function based on above considerations can be determined. Here, only the terms that depend on any of the states are kept, and it is assumed that the noise covariances $\tilde{\mathbf{R}}_k$, $\tilde{\mathbf{Q}}_k$, and $\mathbf{P}_0^{\mathrm{x}}$ have full rank [RTS1965]. The case of singular covariance matrices is treated in subsection 2.3.5.2

$$\begin{aligned} J(\mathbf{X}_{\bar{N}}) &= -\ln p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) = -\sum_{k=0}^{\bar{N}} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k) - \sum_{k=0}^{\bar{N}-1} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) - \ln p(\boldsymbol{x}_0) \\ &= \frac{1}{2} \sum_{k=0}^{\bar{N}} (\boldsymbol{z}_k - \mathbf{C}_k \boldsymbol{x}_k - \mathbf{D}_k \boldsymbol{u}_k)^{\intercal}\, \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{C}_k \boldsymbol{x}_k - \mathbf{D}_k \boldsymbol{u}_k) \\ &+ \frac{1}{2} \sum_{k=0}^{\bar{N}-1} (\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k)^{\intercal}\, \tilde{\mathbf{Q}}_k^{-1} (\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_k \boldsymbol{x}_k - \boldsymbol{\Gamma}_k \boldsymbol{u}_k) \\ &+ \frac{1}{2} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^{\intercal} (\mathbf{P}_0^{\mathrm{x}})^{-1} (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0) \end{aligned} \tag{2.272}$$

JAZWINKSI gives an interpretation of above cost function, which also makes it meaningful from a non-statistical point of view: "Roughly speaking, we want to pass the

solution of [the system's difference equation], as closely as possible, through the observations." [Jaz1970, Ch. 5, p. 151]. As will be illustrated later, this interpretation even holds in the case of a non-linear system model. COX' interpretation is somewhat more statistical: using the minimum of above cost function as estimate for the state trajectory "... corresponds to choosing the $\mathbf{X}_{\bar{N}}$ of the most probable (least unlikely) of all possible sequences of states given the observation and the a priori distribution" [Cox1963, p. 36].

Another interesting fact about above cost-function formulation is the role of $\bar{\boldsymbol{x}}_0$ and $\mathbf{P}_0^{\mathrm{x}}$: as the number of samples grows, their influence on the cost function becomes smaller and smaller. Thus, for the quality of the final result the accuracy of the estimates of the initial statistics are of minor importance. This is good news, since they are often difficult to determine.

For linear systems, above is a cost function that is quadratic in the state estimate, which can thus be solved analytically. The corresponding weighted least squares measurement equation can be formulated as

$$\underbrace{\begin{bmatrix} \bar{\boldsymbol{x}}_0 \\ \boldsymbol{z}_0 - \mathbf{D}_0\boldsymbol{u}_0 \\ \vdots \\ \boldsymbol{z}_{\bar{N}-1} - \mathbf{D}_{\bar{N}-1}\boldsymbol{u}_{\bar{N}-1} \\ \boldsymbol{z}_{\bar{N}} - \mathbf{D}_{\bar{N}}\boldsymbol{u}_{\bar{N}} \\ -\boldsymbol{\Gamma}_0\boldsymbol{u}_0 \\ \vdots \\ -\boldsymbol{\Gamma}_{\bar{N}-1}\boldsymbol{u}_{\bar{N}-1} \end{bmatrix}}_{\boldsymbol{z}_{JMLE}} = \underbrace{\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{x}}} & & & & \\ \mathbf{C}_0 & & & & \\ & \ddots & & & \\ & & \mathbf{C}_{\bar{N}-1} & & \\ & & & & \mathbf{C}_{\bar{N}} \\ \boldsymbol{\Phi}_0 & -\mathbf{I}_{n_{\boldsymbol{x}}} & & & \\ & \ddots & \ddots & & \\ & & \boldsymbol{\Phi}_{\bar{N}-1} & -\mathbf{I}_{n_{\boldsymbol{x}}} \end{bmatrix}}_{\mathbf{x}_{JMLE}} \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_{\bar{N}-1} \\ \boldsymbol{x}_{\bar{N}} \end{bmatrix} + \underbrace{\begin{bmatrix} \bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0 \\ \mathbf{G}_0\boldsymbol{v}_0 \\ \vdots \\ \mathbf{G}_{\bar{N}-1}\boldsymbol{v}_{\bar{N}-1} \\ \mathbf{G}_{\bar{N}}\boldsymbol{v}_{\bar{N}} \\ \mathbf{F}_0\boldsymbol{w}_0 \\ \vdots \\ \mathbf{F}_{\bar{N}-1}\boldsymbol{w}_{\bar{N}-1} \end{bmatrix}}_{\boldsymbol{v}_{JMLE}} \tag{2.273}$$

with corresponding noise covariance

$$\mathrm{Cov}[\boldsymbol{v}_{JMLE}] = \begin{bmatrix} \mathbf{P}_0^{\mathrm{x}} & & & & & & \\ & \tilde{\mathbf{R}}_0 & & & & & \\ & & \ddots & & & & \\ & & & \tilde{\mathbf{R}}_{\bar{N}} & & & \\ & & & & \tilde{\mathbf{Q}}_0 & & \\ & & & & & \ddots & \\ & & & & & & \tilde{\mathbf{Q}}_{\bar{N}-1} \end{bmatrix} \tag{2.274}$$

In section 2.2.4.6 it was shown, how different information sources may be optimally combined in the linear least-squares context. Here the three sources of information for the state estimates can be identified as follows: the first row of above problem corresponds to prior information on the initial state; the second block represents infor-

mation obtained through the measurements; and the last block links the state estimates via the system dynamics.

Eventually, the resulting state estimate can be computed using the weighted least-squares approach of section 2.2.4.3, with weighting matrix $\mathbf{W}_{JMLE} = \mathrm{Cov}[\boldsymbol{v}_{JMLE}]^{-1}$

$$\begin{bmatrix} \hat{\boldsymbol{x}}_{0|\bar{N}} \\ \vdots \\ \hat{\boldsymbol{x}}_{\bar{N}-1|\bar{N}} \\ \hat{\boldsymbol{x}}_{\bar{N}|\bar{N}} \end{bmatrix} = \left(\mathbf{X}_{JMLE}^{\intercal}\mathbf{W}_{JMLE}\mathbf{X}_{JMLE}\right)^{-1}\mathbf{X}_{JMLE}^{\intercal}\mathbf{W}_{JMLE}\boldsymbol{z}_{JMLE} \tag{2.275}$$

The corresponding state covariance estimates are

$$\mathrm{Cov}\left[\begin{bmatrix} \hat{\boldsymbol{x}}_{0|\bar{N}} \\ \vdots \\ \hat{\boldsymbol{x}}_{\bar{N}-1|\bar{N}} \\ \hat{\boldsymbol{x}}_{\bar{N}|\bar{N}} \end{bmatrix}\right] = \left(\mathbf{X}_{JMLE}^{\intercal}\mathbf{W}_{JMLE}\mathbf{X}_{JMLE}\right)^{-1} \tag{2.276}$$

Above is in general a fully dense matrix, which also covers covariances between state estimates at different sampling instants. The $n_{\boldsymbol{x}} \times n_{\boldsymbol{x}}$-blocks on the main diagonal of above covariance matrix correspond to the commonly considered state covariance matrices $\mathbf{P}_{k|\bar{N}}^{\tilde{\mathbf{x}}}$.

Since the result of JMLE is numerically identical to the solution obtained using the recursive RTS smoother (both in state and covariance estimate), it is not explicitly shown in example 2.6 and figure 2.4.

In their publication, RAUCH et al. show this equivalence of JMLE and their recursive formulation analytically for linear, discrete time systems. However, they also state that for the general, nonlinear case the two solutions should be expected to differ [RTS1965]. A more detailed proof of the equivalence of RAUCH et al.'s solution and the solution to above joint maximum likelihood state estimation problem for linear, discrete time systems can be found in [CJ2012, Ch. 6.4.1].

RAUCH et al. also state, that above formulation is the discrete version of a problem stated earlier by BRYSON and FRAZIER [BF1963]: They formulate the cost function in continuous time (essentially replacing above sums with the respective integrals) and solve it using the indirect method. This eventually leads to a Two-Point-Boundary Value problem, as is illustrated in section 2.1.4.2 for general optimal control problems. The formulation here is then the discrete time equivalent using full discretization, which retains the advantages of the direct method (robustness, no need for derivatives of Hamiltonian, versatility, ...).

2.3.4 Non-Linear Filtering

Since no system is truly linear, non-linear extensions to state estimation have to be found. Unfortunately, the few results that are available for non-linear filtering are in general too complex to be of practical use. This makes approximations necessary, which are often problem-specific. The main focus of this work will lie on the Extended Kalman Filter (EKF), which is widely used in aircraft applications, and non-linear JMLE, which together with the optimal control point of view is a novel approach. The EKF was first published on the context of position and velocity determination for a circumlunar vehicle [SSM1962] and has since been applied successfully in many other areas.

The basic idea behind the EKF is to linearize the system equations around the last corrected state update, and then to apply the standard Kalman filter equations to this linearized system. The details that will be presented in the next subsections are based on the derivations in [Sim2006, CJ2012].

2.3.4.1 Extended Kalman Filter - System Description

Many different system descriptions for the EKF are available, e.g. a complete, continuous time formulation, a hybrid continuous discrete formulation, or, as considered here, a fully discrete system formulation [Sim2006, Ch. 13.2.1. & 13.2.2]. The complete discrete formulation is selected, due to its rather simple covariance propagation. It results in one algebraic expression, rather than the numerical solution to a matrix-valued Riccatti equation. Furthermore, due to the sampled nature of the available data, a discrete time approach is more intuitive.

Thus the system is

$$\boldsymbol{x}_{k+1} = {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] \qquad k = 0 \dots \bar{N} - 1 \tag{2.277}$$

$$\boldsymbol{y}_k = \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \mathbf{0}) \qquad k = 0 \dots \bar{N} \tag{2.278}$$

$$\boldsymbol{z}_k = \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{v}_k) \tag{2.279}$$

The noise terms are characterized the same way as in equations (2.183) - (2.187), i.e. they are assumed to be zero mean, white noise processes with known covariance matrices. Furthermore, the distribution of the initial condition is assumed to be known $\mathbf{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \mathbf{P}_0^{\mathrm{x}})$, see (2.189) and (2.190).

${}^{d}\mathbf{f}[\,]$ describes the non-linear, discrete system equation, which results as the integration of the non-linear continuous time system $\boldsymbol{f}(\,)$ over one time step. This integration may be approximated through one step of any explicit Runge-Kutta numerical integration scheme, see Appendix D.3 for details.

The non-linear, discrete system equations are closely related to the state integration schemes used in the section on optimal control 2.1.4. Repeated, iterative application of

the discrete dynamic system may be used to make up an integration scheme

$$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{\Phi}_{\boldsymbol{f}}(t_0 \dots t_k | \boldsymbol{x}_0, \boldsymbol{u}_0 \dots \boldsymbol{u}_{k-1}) \\ &= {}^d\mathbf{f}_{k-1}\Big[{}^d\mathbf{f}_{k-2}\Big[\dots {}^d\mathbf{f}_0[\boldsymbol{x}_0, \boldsymbol{u}_0] \dots, \boldsymbol{u}_{k-2}\Big], \boldsymbol{u}_{k-1}\Big]\end{aligned} \qquad k = 1 \dots \bar{N} \tag{2.280}$$

The fundamental difference is that the discrete system only describes the evolution through *one* time step, whereas the integration scheme may be used for a whole interval. Thus only for one time-step it holds

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1} | \boldsymbol{x}_k, \boldsymbol{u}_k) = {}^d\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k] \tag{2.281}$$

2.3.4.2 Extended Kalman Filter - Application

Linearizing the system equation (2.277) at the current state estimate $\hat{\boldsymbol{x}}_{k|k}$, and the mean of the process noise term $\mathrm{E}[\boldsymbol{w}_k] = \mathbf{0}$ yields

$$\begin{aligned}\boldsymbol{x}_{k+1} &= {}^d\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] + \left.\frac{\partial\, {}^d\mathbf{f}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{w}_k = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k|k}}} \Big(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\Big) + \left.\frac{\partial\, {}^d\mathbf{f}_k}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{w}_k = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k|k}}} \boldsymbol{w}_k + \text{h.o.t.} \\ &= {}^d\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] + \boldsymbol{\Phi}_k\Big(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}\Big) + \mathbf{F}_k \boldsymbol{w}_k + \text{h.o.t.} \\ &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \Big({}^d\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k}\Big) + \mathbf{F}_k \boldsymbol{w}_k + \text{h.o.t.} \\ &= \boldsymbol{\Phi}_k \boldsymbol{x}_k + \tilde{\boldsymbol{u}}_k + \tilde{\boldsymbol{w}}_k\end{aligned} \tag{2.282}$$

In above derivation, the following abbreviations were used

$$\boldsymbol{\Phi}_k = \left.\frac{\partial\, {}^d\mathbf{f}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{w}_k = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k|k}}} \tag{2.283}$$

$$\mathbf{F}_k = \left.\frac{\partial\, {}^d\mathbf{f}_k}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{w}_k = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k|k}}} \tag{2.284}$$

$$\tilde{\boldsymbol{u}}_k = {}^d\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} \tag{2.285}$$

$$\tilde{\boldsymbol{w}}_k = \mathbf{F}_k \boldsymbol{w}_k + \text{h.o.t.} \tag{2.286}$$

In this way, the non-linear system can be transformed into a linear system, where the deterministic part of the non-linearity is "hidden" in the deterministic input $\tilde{\boldsymbol{u}}_k$, and the linearization errors are attributed to an increase in process noise via $\tilde{\boldsymbol{w}}_k$ [Sim2006].

Similarly, the output equation (2.278) is linearized at the predicted state estimate $\hat{\boldsymbol{x}}_{k+1|k}$, and the mean of the measurement noise term $\mathrm{E}[\boldsymbol{v}_{k+1}] = \mathbf{0}$

$$\begin{aligned}\boldsymbol{z}_{k+1} &= \boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big) + \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{v}_{k+1} = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k+1|k}}} \Big(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}\Big) + \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{v}}\right|_{\substack{\boldsymbol{v}_{k+1} = \mathbf{0} \\ \hat{\boldsymbol{x}}_{k+1|k}}} \boldsymbol{v}_{k+1} + \text{h.o.t.} \\ &= \boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big) + \mathbf{C}_{k+1}\Big(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k}\Big) + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} + \text{h.o.t.} \\ &= \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \Big(\boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big) - \mathbf{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\Big) + \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} + \text{h.o.t.} \\ &= \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \tilde{\boldsymbol{y}}_{k+1} + \tilde{\boldsymbol{v}}_{k+1}\end{aligned} \tag{2.287}$$

The following abbreviations were used

$$\mathbf{C}_{k+1} = \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{v}_{k+1}=\mathbf{0} \\ \hat{\boldsymbol{x}}_{k+1|k}}} \tag{2.288}$$

$$\mathbf{G}_{k+1} = \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{v}}\right|_{\substack{\boldsymbol{v}_{k+1}=\mathbf{0} \\ \hat{\boldsymbol{x}}_{k+1|k}}} \tag{2.289}$$

$$\tilde{\boldsymbol{y}}_{k+1} = \boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big) - \mathbf{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k} \tag{2.290}$$

$$\tilde{\boldsymbol{v}}_{k+1} = \mathbf{G}_{k+1}\boldsymbol{v}_{k+1} + \text{h.o.t.} \tag{2.291}$$

Again, the non-linearity is "hidden" in a deterministic feed-through-type of term $\tilde{\boldsymbol{y}}_{k+1}$, and the linearization errors in the output are attributed to an increase in measurement noise via $\tilde{\boldsymbol{v}}_{k+1}$ [Sim2006].

Now, the linear Kalman filter (see algorithm 2.4) of the last section can be applied to the system

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_k \boldsymbol{x}_k + \tilde{\boldsymbol{u}}_k + \tilde{\boldsymbol{w}}_k \tag{2.292}$$

$$\boldsymbol{z}_{k+1} = \mathbf{C}_{k+1}\boldsymbol{x}_{k+1} + \tilde{\boldsymbol{y}}_{k+1} + \tilde{\boldsymbol{v}}_{k+1} \tag{2.293}$$

Using equations (2.283) - (2.285), the propagation step becomes

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \tilde{\boldsymbol{u}}_k \\ &= \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \Big({}^{d}\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] - \boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k}\Big) \\ &= {}^{d}\mathbf{f}_k\Big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\Big] \end{aligned} \tag{2.294}$$

The correction step is very similar to the linear case, shown in equations (2.288) - (2.290)

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\Big(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{y}}_{k+1|k}\Big) \\ &= \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\Big(\boldsymbol{z}_{k+1} - \Big(\mathbf{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k} + \Big(\boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big) - \mathbf{C}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\Big)\Big)\Big) \\ &= \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\Big(\boldsymbol{z}_{k+1} - \boldsymbol{g}_{k+1}\Big(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\Big)\Big) \end{aligned} \tag{2.295}$$

The propagation and correction of the state covariance matrix, as well as the computation of the Kalman gain are all based on the linearized system. Thus, the same equations as in the linear case result. The algorithm is summed up in Algorithm 2.6.

2.3.4.3 Extended Kalman Filter - Consquences for Characteristics

There are some difficulties that arise from the linear approximation of the non-linear dynamics:

1. even if the linearization errors could be treated as "random" process noise, assuming they are white is in general not possible.
2. the increase in process and measurement noise due to the linearization errors Cov[h.o.t.] is difficult to quantify numerically.

Algorithm 2.6: Discrete time, extended Kalman filter

Initialization:

I. Initialize state and covariance estimates

$$\hat{\boldsymbol{x}}_{0|-1} = \bar{\boldsymbol{x}}_0$$
$$\mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1} = \mathbf{P}^{\mathrm{x}}_0$$

II. if an initial measurement is to be considered (i.e. $\tilde{\mathbf{R}}_0$ is finite), perform initial correction step

$$\mathbf{C}_0 = \left.\frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{v}_0=\mathbf{0}\\ \hat{\boldsymbol{x}}_{0|-1}}} \qquad \mathbf{G}_0 = \left.\frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{v}}\right|_{\substack{\boldsymbol{v}_0=\mathbf{0}\\ \hat{\boldsymbol{x}}_{0|-1}}}$$
$$\mathbf{K}_0 = \mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\mathbf{C}_0^{\intercal}\left(\mathbf{C}_0\mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\mathbf{C}_0^{\intercal} + \tilde{\mathbf{R}}_0\right)^{-1}$$
$$\mathbf{P}^{\tilde{\mathbf{x}}}_{0|0} = \left(\mathbf{I}_{n_x} - \mathbf{K}_0\mathbf{C}_0\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}\left(\mathbf{I}_{n_x} - \mathbf{K}_0\mathbf{C}_0\right)^{\intercal} + \mathbf{K}_0\tilde{\mathbf{R}}_0\mathbf{K}_0^{\intercal}$$
$$\hat{\boldsymbol{x}}_{0|0} = \hat{\boldsymbol{x}}_{0|-1} + \mathbf{K}_0\left(\boldsymbol{z}_0 - \boldsymbol{g}_0\left(\hat{\boldsymbol{x}}_{0|-1}, \boldsymbol{u}_0, \mathbf{0}\right)\right)$$

otherwise set $\hat{\boldsymbol{x}}_{0|0} = \hat{\boldsymbol{x}}_{0|-1}$, and $\mathbf{P}^{\tilde{\mathbf{x}}}_{0|0} = \mathbf{P}^{\tilde{\mathbf{x}}}_{0|-1}$

Main Part:

III. linearize for prediction

$$\boldsymbol{\Phi}_k = \left.\frac{\partial\, {}^{d}\mathbf{f}_k}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{w}_k=\mathbf{0}\\ \hat{\boldsymbol{x}}_{k|k}}} \qquad \mathbf{F}_k = \left.\frac{\partial\, {}^{d}\mathbf{f}_k}{\partial \boldsymbol{w}}\right|_{\substack{\boldsymbol{w}_k=\mathbf{0}\\ \hat{\boldsymbol{x}}_{k|k}}}$$

IV. prediction step

$$\hat{\boldsymbol{x}}_{k+1|k} = {}^{d}\mathbf{f}_k\left[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\right]$$
$$\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} = \boldsymbol{\Phi}_k\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k}\boldsymbol{\Phi}_k^{\intercal} + \tilde{\mathbf{Q}}_k$$

V. linearize for correction

$$\mathbf{C}_{k+1} = \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{x}}\right|_{\substack{\boldsymbol{v}_{k+1}=\mathbf{0}\\ \hat{\boldsymbol{x}}_{k+1|k}}} \qquad \mathbf{G}_{k+1} = \left.\frac{\partial \boldsymbol{g}_{k+1}}{\partial \boldsymbol{v}}\right|_{\substack{\boldsymbol{v}_{k+1}=\mathbf{0}\\ \hat{\boldsymbol{x}}_{k+1|k}}}$$

VI. correction step

(a) compute Gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^{\intercal}\left(\mathbf{C}_{k+1}\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\mathbf{C}_{k+1}^{\intercal} + \tilde{\mathbf{R}}_{k+1}\right)^{-1}$$

(b) correct estimates

$$\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k+1} = \left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k}\left(\mathbf{I}_{n_x} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\right)^{\intercal} + \mathbf{K}_{k+1}\tilde{\mathbf{R}}_{k+1}\mathbf{K}_{k+1}^{\intercal}$$
$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \mathbf{K}_{k+1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{g}_{k+1}\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_{k+1}, \mathbf{0}\right)\right)$$

VII. increment k and iterate from step III. while $k \leq \bar{N} - 1$

3. if Gaussian noise was assumed, in the linear case all random terms stay Gaussian. This is no longer true in the EKF case, since a non-linear transformation of Gaussian noise is in general no longer Gaussian.

Although these aspects violate some of the basic assumptions in the derivation of the linear Kalman filter (white noise processes and known covariance matrices), the overall EKF approach has been found to work well in practice.

Furthermore, the notion of increasing the noise processes' covariances to cover linearization errors, can be extended to include modeling deficiencies and computational errors, too. SORENSON puts it nicely: "it is naive at best to assume that any physical system can be modeled precisely, so it is necessary to account for model errors" [Sor1980, p. 24]. Thus increased process noise can be a useful, albeit artificial method to cover these aspects [Jaz1970, Ch. 8]. Assuming the true system ${}^{d}\mathbf{f}_{k}^{true}$ is modeled using the formulation ${}^{d}\mathbf{f}_{k}$, then the true evolution of the system can be written as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= {}^{d}\mathbf{f}_{k}^{true}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] \\ &= {}^{d}\mathbf{f}_{k}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] + \Big({}^{d}\mathbf{f}_{k}^{true}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] - {}^{d}\mathbf{f}_{k}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k]\Big) \end{aligned} \tag{2.296}$$

where the error term ${}^{d}\mathbf{f}_{k}^{true} - {}^{d}\mathbf{f}_{k}$ may be treated as additional process noise source. However, the difficulty of quantifying its covariance remains, together with the fact that in general, systematic modeling errors cannot be characterized as random and white.

2.3.5 Non-Linear Maximum Likelihood State Estimation

Also the presented maximum likelihood state estimation approaches may be extended to non-linear systems. The final solution may be based on similar linearization ideas as shown in the last section, or on consideration of the non-linear dynamics directly.

2.3.5.1 Non-Linear Marginal Maximum Likelihood State Estimation

Using a similar approach as illustrated in section 2.3.4, i.e. linearizing the system equations about the current estimate and treating the non-linear part as artificial input, can be used to derive the extended version of the RTS smoother, the Extended Rauch-Tung-Striebel (ERTS) smoother. Using the backward recursion (2.263) together with the linearized EKF system (2.282) yields

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|\bar{N}} &= \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\Big(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \big(\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k} + \tilde{\boldsymbol{u}}_k\big)\Big) \\ &\overset{(2.285)}{=} \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\Big(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - {}^{d}\mathbf{f}_{k}\big[\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k, \mathbf{0}\big]\Big) \\ &\overset{(2.294)}{=} \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k\Big(\hat{\boldsymbol{x}}_{k+1|\bar{N}} - \hat{\boldsymbol{x}}_{k+1|k}\Big) \end{aligned} \tag{2.297}$$

This is based on the same reasoning as for the EKF: The corrected state estimate $\hat{\boldsymbol{x}}_{k|k}$ is assumed to be "good enough" to justify a linear treatment of the error between true and estimated state trajectory. The resulting state recursion is exactly the same as in the linear case. The smoother gain and covariance equation stay the same as in equations (2.264) and (2.265), since they are based on the linear systems to begin with. Thus the ERTS smoother comprises of the same equations as are summed up in Algorithm 2.5.

When it comes to the applicability of the ERTS smoother, the same limitations and difficulties as mentioned in section 2.3.4.3 apply, i.e. linearization errors are not white, their covariance is difficult to specify, and the error probabilities are non-Gaussian.

2.3.5.2 Non-Linear Joint Maximum Likelihood State Estimate

In the derivation of the joint probability density function for all states and measurements (2.271), the only assumption was, that the generating process be Markovian (see Appendix A.5.4 for a definition). Even the most general dynamic system description, in the form of a first order difference equation (as e.g. the system in equation 2.277) has this property, since it only depends on the last state, and the realization of the process noise [Jaz1970].

In the development of the joint maximum likelihood cost function for linear systems in section 2.3.3.3, the following probability densities needed to be specified: $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$, $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$, and $p(\boldsymbol{x}_0)$. Two different cases will be treated here, to illustrate the approach in the non-linear case: first, purely additive Gaussian process noise with non-singular covariance is treated; then a more general setting is investigated, where the process noise is allowed to enter non-linearly.

Additive Gaussian Process Noise

If one is to assume purely additive process and measurement noise, the system becomes

$$\begin{aligned}\boldsymbol{x}_{k+1} &= {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k] + \mathbf{F}_k\boldsymbol{w}_k \\ \boldsymbol{y}_k &= \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{G}_k\boldsymbol{v}_k\end{aligned}$$

Then, the same joint-pdf and cost function as in equations (2.271) and (2.272) may be used for non-linear systems as well

$$\begin{aligned}p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) &= \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_0) \\ J(\mathbf{X}_{\bar{N}}) &= -\sum_{k=0}^{\bar{N}} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k) - \sum_{k=0}^{\bar{N}-1} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) - \ln p(\boldsymbol{x}_0)\end{aligned}$$

Assuming Gaussian distributions, the necessary conditional probability densities can be characterized by their means and covariances alone: Based on the computational

rules for conditional expectations (A.113) - (A.117), the development for $\mathrm{z_k}|\mathrm{x_k}$ in equations (2.230), (2.232), and the development for $\mathrm{x_{k+1}}|\mathrm{x_k}$ in equations (2.253), (2.254), the desired means and covariances are

$$\mathrm{E}[\boldsymbol{x}_{k+1}|\boldsymbol{x}_k] = \mathrm{E}\Big[\Big({}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k] + \mathbf{F}_k\boldsymbol{w}_k\Big)\Big|\boldsymbol{x}_k\Big] \overset{\text{(A.116)}}{=} {}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k] \tag{2.298}$$

$$\mathrm{Cov}[\boldsymbol{x}_{k+1}|\boldsymbol{x}_k] = \mathrm{Cov}\Big[\Big({}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k] + \mathbf{F}_k\boldsymbol{w}_k\Big)\Big|\boldsymbol{x}_k\Big] = \mathbf{F}_k\mathbf{Q}_k\mathbf{F}_k^\intercal = \tilde{\mathbf{Q}}_k \tag{2.299}$$

$$\mathrm{E}[\boldsymbol{z}_k|\boldsymbol{x}_k] = \mathrm{E}[(\boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k) + \mathbf{G}_k\boldsymbol{v}_k)|\boldsymbol{x}_k] = \boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k) \tag{2.300}$$

$$\mathrm{Cov}[\boldsymbol{z}_k|\boldsymbol{x}_k] = \mathrm{Cov}[(\boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k) + \mathbf{G}_k\boldsymbol{v}_k)|\boldsymbol{x}_k] = \mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^\intercal = \tilde{\mathbf{R}}_k \tag{2.301}$$

$$\mathrm{E}[\boldsymbol{x}_0] = \bar{\boldsymbol{x}}_0 \tag{2.302}$$

$$\mathrm{Cov}[\boldsymbol{x}_0] = \mathbf{P}_0^{\mathrm{x}} \tag{2.303}$$

Above results can again be obtained using the transformation of probability densities (A.93) and the assumption of Gaussian process and measurement noise

$$p_{\mathbf{z_k}|\mathbf{x_k}}(\boldsymbol{z}_k|\boldsymbol{x}_k) = p_{\mathbf{G_k v_k}}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k)) \tag{2.304}$$

$$p_{\mathbf{x_{k+1}}|\mathbf{x_k}}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = p_{\mathbf{F_k w_k}}\Big(\boldsymbol{x}_{k+1} - {}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k]\Big) \tag{2.305}$$

For now, assume $\tilde{\mathbf{Q}}_k$ to be non-singular. Furthermore, assume $\tilde{\mathbf{R}}_k$ to have full rank. Then the joint maximum likelihood estimation problem, assuming Gaussian densities, is similar to equation (2.272) [Jaz1970, Ch. 5]

$$\min_{\mathbf{X}_{\bar{N}}} J(\mathbf{X}_{\bar{N}}) \tag{2.306}$$

$$\begin{aligned}
J(\mathbf{X}_{\bar{N}}) &= -\ln p(\boldsymbol{x}_0) - \sum_{k=0}^{\bar{N}} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k) - \sum_{k=0}^{\bar{N}-1} \ln p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \\
&= -\ln p(\boldsymbol{x}_0) - \sum_{k=0}^{\bar{N}} \ln p_{\mathbf{G_k v_k}}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k)) \\
&\qquad - \sum_{k=0}^{\bar{N}-1} \ln p_{\mathbf{F_k w_k}}\Big(\boldsymbol{x}_{k+1} - {}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k]\Big) \\
&= \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^\intercal (\mathbf{P}_0^{\mathrm{x}})^{-1}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0) \\
&+ \frac{1}{2}\sum_{k=0}^{\bar{N}} (\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k))^\intercal \tilde{\mathbf{R}}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k,\boldsymbol{u}_k)) \\
&+ \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \Big(\boldsymbol{x}_{k+1} - {}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k]\Big)^\intercal \tilde{\mathbf{Q}}_k^{-1}\Big(\boldsymbol{x}_{k+1} - {}^d\mathbf{f}_k[\boldsymbol{x}_k,\boldsymbol{u}_k]\Big)
\end{aligned} \tag{2.307}$$

Other than a model formulation having the Markov property, and only additive, Gaussian noise sources, no constraints apply. This makes this approach very versatile.

Non-Linearly Entering Process Noise

A more general formulation was developed by FRIEDLAND and BERNSTEIN [FB1966], which relaxes the requirements on the transition probability distributions (Gaussian,

and additive). Here, their results are presented for non-additive, Gaussian process noise, and additive, Gaussian measurement noise, where process and measurement noise sources are independent[11]

$$\begin{aligned}\boldsymbol{x}_{k+1} &= {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] \\ \boldsymbol{y}_k &= \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \mathbf{G}_k \boldsymbol{v}_k\end{aligned}$$

At first, assume that there exists a non-singular transformation

$$\boldsymbol{w}_k = {}^{d}\mathbf{f}_k^{-1}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{k+1}] \tag{2.308}$$

Then the theorem about the transformation of probability densities (A.93) can be used to arrive at an expression for the transition probability density $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = p_{\mathbf{w}_\mathrm{k}}\left({}^{d}\mathbf{f}_k^{-1}[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{k+1}]\right)\left|\left|\frac{\partial\, {}^{d}\mathbf{f}_k^{-1}}{\partial \boldsymbol{x}_k}\right|\right| \tag{2.309}$$

Here attention has to be paid to the fact that $\big|\,|\square|\,\big|$ is the *absolute value of the determinant*, not a norm. In the first equality, it is used that $\boldsymbol{w}_k$ and $\boldsymbol{x}_k$ are independent

$$p_{\mathbf{w}_\mathrm{k}}(\boldsymbol{w}_k) = p_{\mathbf{w}_\mathrm{k}|\mathbf{x}_\mathrm{k}}(\boldsymbol{w}_k|\boldsymbol{x}_k) \tag{2.310}$$

The second equality uses the inverse function theorem (A.15).

Now, instead of explicitly computing the inverse of the system dynamics ${}^{d}\mathbf{f}_k^{-1}$, FRIEDLAND et al. propose to introduce the process noise terms as optimization variables and the system dynamics as equality constraint in the optimization problem [FB1966]

$$\left(\begin{array}{cc} \min\limits_{\substack{\boldsymbol{x}_0, \boldsymbol{w}_0, \ldots \\ \boldsymbol{x}_{\bar{N}-1}, \boldsymbol{w}_{\bar{N}-1}, \boldsymbol{x}_{\bar{N}}}} & J(\boldsymbol{x}_0, \boldsymbol{w}_0 \ldots \boldsymbol{x}_{\bar{N}-1}, \boldsymbol{w}_{\bar{N}-1}, \boldsymbol{x}_{\bar{N}}) \\ \text{s.t.} & \boldsymbol{x}_{k+1} = {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] \end{array}\right) \tag{2.311}$$

$$\begin{aligned} J(\boldsymbol{x}_0, \boldsymbol{w}_0 \ldots \boldsymbol{x}_{\bar{N}-1}, \boldsymbol{w}_{\bar{N}-1}, \boldsymbol{x}_{\bar{N}}) &= -\ln p(\boldsymbol{x}_0) - \sum_{k=0}^{\bar{N}} \ln p_{\mathbf{G}_\mathrm{k}\mathbf{v}_\mathrm{k}}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)) \\ &\quad - \sum_{k=0}^{\bar{N}-1} \ln\left(p_{\mathbf{w}_\mathrm{k}}(\boldsymbol{w}_k)\left|\left|\frac{\partial\, {}^{d}\mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right|\right|\right) \\ &= \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^\intercal (\mathbf{P}_0^\mathrm{x})^{-1}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0) \\ &\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}} (\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k))^\intercal \tilde{\mathbf{R}}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)) \\ &\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \boldsymbol{w}_k^\intercal \mathbf{Q}_{k-1}^{-1} \boldsymbol{w}_k - \sum_{k=0}^{\bar{N}-1} \ln\left|\left|\frac{\partial\, {}^{d}\mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right|\right| \end{aligned} \tag{2.312}$$

Some special cases are worth mentioning:

[11] for their fully general case (with both noise sources possibly non-additive and correlated with an arbitrary, but known, probability density), see [FB1966].

Independent Process Noise Jacobian If the Jacobian $\frac{\partial\, {}^{d}\mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}$ does neither depend on the states $\boldsymbol{x}_k$ nor on the process noise $\boldsymbol{w}_k$, the last term in above equation does not depend on the optimization variables and may be dropped.

This is e.g. the case for purely additive process noise, with a full-rank $\mathbf{F}_k$. In this case, by using

$$\boldsymbol{w}_k = \mathbf{F}_k^{-1} \cdot \left(\boldsymbol{x}_{k+1} - {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k] \right) \tag{2.313}$$

the process noise terms can be eliminated as optimization variables and the cost function formulation of the previous section arises.

Singular Transformation FRIEDLAND et al. show that the above formulation in equation 2.309 is also valid, if the Jacobian $\frac{\partial\, {}^{d}\mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}$ is singular, either because of an unfavorable choice of states $\boldsymbol{x}_k$, or because $n_{\boldsymbol{w}} < n_{\boldsymbol{x}}$ [FB1966]. The only prerequisite is that the system equation can be partitioned in one part that is influenced by process noise ${}^{d}\mathbf{f}_k^{I}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k)$ (for which a non-singular transformation exists) and a second part that is purely deterministic ${}^{d}\mathbf{f}_k^{II}(\boldsymbol{x}_k, \boldsymbol{u}_k)$. By using the approach via the inclusion of the system equations as equality constraints, the second part of the dynamics ${}^{d}\mathbf{f}_k^{II}$ will be fulfilled anyway, and for the first part, the transformation can be performed as illustrated above. Then the respective term in the cost function is

$$-\sum_{k=0}^{\bar{N}-1} \ln \left| \left| \frac{\partial^{d} \mathbf{f}_k^{-1\,I}}{\partial \boldsymbol{w}_k} \right| \right| \tag{2.314}$$

Conclusion

As mentioned in the context of joint maximum likelihood state estimation for linear models, above cost function also has its merits from a non-statistical point of view. The goal of the optimization problem can be interpreted as follows: on the one hand a solution is passed as closely as possible through the observations. On the other hand, the system's dynamics are to be fulfilled "as good as possible", by minimizing the energy of a random input. Both the measurement noise and the process noise terms are weighted by matrices according to the "trust" one puts in the respective information. A more statistical interpretation is that above is "... the problem of finding the most likely random input after making noisy observations of the output" [Cox1963, p. 96]. This idea has been used by BACH, together with a cost function similar to the above, for Flight Path Reconstruction (FPR) problems (see section 5.2 for further details on FPR). However, he then uses arguments based on variational calculus and an adjoint equation rather than full-discretization to eventually find a solution to the non-linear smoothing problem [Bac1982, BW1985].

The measurement noise terms are closely related to the residuals, as illustrated in the context of parameter estimation using the output error method in section 2.2.2.2.

In contrast, the process noise terms are closely related to integration defects in the full discretization approach to the solution of optimal control problems, see section 2.1.4.3.

Above results, can be extended to non-linearly entering measurement noise and singular transformations thereof, too. However, in this work it was decided to only consider additive measurement noise, since this provides a realistic enough setting: additive, white, Gaussian measurement noise is the standard approach to modeling random sensor errors and has yielded good results over decades. The case of singular transformations of the measurement noise is disregarded, since this would mean perfect measurements or perfectly correlated noise sequences, which is unheard of in real-life problems.

2.3.6 Joint Parameter and State Estimation

Some authors consider the problem of parameter estimation as a special case of the problem of state estimation [CJ2012, Sim2006], amongst which the first were [KO1963, Cox1964]. The former problem has been discussed at length in the foregoing sections. Parameter estimation may then be incorporated by augmenting the system with a model, describing the parameters' evolution

$$\boldsymbol{x}_{k+1}^{a} = \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{\theta}_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}_k] \\ {}^{d}\mathbf{f}_k^{\boldsymbol{\theta}}\left[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}^{\boldsymbol{\theta}}{}_k\right] \end{bmatrix} \qquad k = 0 \dots \bar{N} - 1 \tag{2.315}$$

$$\boldsymbol{z}_k = \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k) + \mathbf{G}_k \boldsymbol{v}_k \qquad k = 0 \dots \bar{N} \tag{2.316}$$

with the following definition of the white noise processes involved

$$\mathbf{w}_{\mathrm{k}}, \mathbf{w}_{\mathrm{k}}^{\theta}, \mathbf{v}_{\mathrm{k+1}} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{\boldsymbol{\theta}}{}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{k+1} \end{bmatrix}\right) \tag{2.317}$$

$$\mathbf{x}_0, \theta_0 \sim \mathcal{N}\left(\begin{bmatrix} \bar{\boldsymbol{x}}_0 \\ \boldsymbol{\theta}_{prior} \end{bmatrix}, \begin{bmatrix} \mathbf{P}_0^{\mathrm{x}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma} \end{bmatrix}\right) \tag{2.318}$$

Above formulation makes some assumptions, which are usually well justified: process and measurement noise are considered to be independent; so are the process noise contributions for the states and parameters. The same is true for the distributions of the initial conditions. The last assumption is that all noise processes are Gaussian, which can be questioned in general, but yielded good application results in the past.

In contrast to the approaches illustrated in Chapter 2.2, above formulation is also able to accommodate time-varying parameters. Application of an EKF, and an ERTS smoother to above system is straightforward. Alternatively, similar manipulations as

in sections 2.3.3 and 2.3.5 using the augmented state vector yield

$$\begin{aligned} p(\mathbf{X}_{\bar{N}}^a, \mathbf{Z}_{\bar{N}}) &= \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k^a) \cdot \prod_{k=0}^{\bar{N}-1} p\Big(\boldsymbol{x}_{k+1}^a \Big| \boldsymbol{x}_k^a\Big) \cdot p(\boldsymbol{x}_0^a) \\ &= \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}, \boldsymbol{\theta}_{k+1}|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot p(\boldsymbol{x}_0, \boldsymbol{\theta}_0) \\ &= \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{\theta}_{k+1}|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot p(\boldsymbol{x}_0) \cdot p(\boldsymbol{\theta}_0) \end{aligned} \tag{2.319}$$

The last equality results from the assumption that the process noise terms driving the system are independent for $\boldsymbol{x}_k$ and $\boldsymbol{\theta}_k$; so are the distributions of the initial conditions.

The resulting optimization problem is

$$\left(\begin{array}{ll} \min\limits_{\substack{\boldsymbol{x}_0^a, \boldsymbol{w}_0^a, \dots \\ \boldsymbol{x}_{\bar{N}-1}^a, \boldsymbol{w}_{\bar{N}-1}^a, \boldsymbol{x}_{\bar{N}}^a}} & J\Big(\boldsymbol{x}_0^a, \boldsymbol{w}_0^a \dots \boldsymbol{x}_{\bar{N}-1}^a, \boldsymbol{w}_{\bar{N}-1}^a, \boldsymbol{x}_{\bar{N}}^a\Big) \\ \text{s.t.} & \begin{aligned} \boldsymbol{x}_{k+1} &= {}^d\mathbf{f}_k\big[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}_k\big] \\ \boldsymbol{\theta}_{k+1} &= {}^d\mathbf{f}_k^{\boldsymbol{\theta}}\Big[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}^{\boldsymbol{\theta}}{}_k\Big] \end{aligned} \end{array} \right) \tag{2.320}$$

$$\begin{aligned} &J\Big(\boldsymbol{x}_0^a, \boldsymbol{w}_0^a \dots \boldsymbol{x}_{\bar{N}-1}^a, \boldsymbol{w}_{\bar{N}-1}^a, \boldsymbol{x}_{\bar{N}}^a\Big) = \\ &\quad = \frac{1}{2}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0)^\intercal (\mathbf{P}_0^{\mathrm{x}})^{-1}(\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0) + \frac{1}{2}(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{prior})^\intercal \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{prior}) \\ &\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k))^\intercal \tilde{\mathbf{R}}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k)) + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\Big|\tilde{\mathbf{R}}_k\Big| \\ &\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \boldsymbol{w}_k^\intercal \mathbf{Q}_{k-1}^{-1} \boldsymbol{w}_k - \sum_{k=0}^{\bar{N}-1} \ln\left|\left|\frac{\partial\, {}^d\mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right|\right| \\ &\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \boldsymbol{w}^{\boldsymbol{\theta}}{}_k{}^\intercal \Big(\mathbf{Q}^{\boldsymbol{\theta}}{}_{k-1}\Big)^{-1} \boldsymbol{w}^{\boldsymbol{\theta}}{}_k - \sum_{k=0}^{\bar{N}-1} \ln\left|\left|\frac{\partial^d \mathbf{f}_k^{-1\boldsymbol{\theta}}}{\partial \boldsymbol{w}^{\boldsymbol{\theta}}{}_k}\right|\right| \end{aligned} \tag{2.321}$$

This is the most general problem formulation to be used in this work.

The term $\ln\Big|\tilde{\mathbf{R}}_k\Big| = \ln|\mathbf{G}_k\mathbf{R}_k\mathbf{G}_k^\intercal|$ has thus far been neglected, since the measurement noise covariance matrix $\mathbf{R}_k$ was considered to be given. Then, the term would be a constant, irrelevant to the maximization. However, here it is included for the sake of completeness, and to provide a link to section 3.1 on covariance estimation. It stems from a similar reasoning as was presented in section 2.2.2.2 to arrive at the pure output error cost function. As will be elaborated in the end of section 3.1, the covariance estimation approach is only viable on the output side of the problem, thus the corresponding term involving $\ln|\mathbf{Q}_k|$ is still neglected.

Here the estimation of states and parameters is combined in one, huge optimization problem. In contrast to that, the "classical" approaches illustrated in section 2.2.2.4 solve the state estimation part iteratively, and determine the parameter values in a disconnected maximum likelihood setting.

Several interesting special cases may be derived:

Linear Gauss-Markov Parameter Model If no special knowledge on the evolution of the parameters is available, linear Gauss-Markov models pose a very flexible way of modeling parameter variation

$$\boldsymbol{\theta}_{k+1} = {}^{d}\mathbf{f}_k^{\boldsymbol{\theta}}\left[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k, \boldsymbol{w}^{\boldsymbol{\theta}}{}_k\right] = \boldsymbol{\Phi}_k^{\boldsymbol{\theta}}\left(\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^{const}\right) + \boldsymbol{w}^{\boldsymbol{\theta}}{}_k \tag{2.322}$$

Prior knowledge can be incorporated through the state transition matrix $\boldsymbol{\Phi}_k^{\boldsymbol{\theta}}$ and forcing term $\boldsymbol{\theta}_k^{const}$. This formulation may be used to implemented a colored noise behavior of the parameters, and to limit the bandwidth of parameter changes.

If no prior knowledge is available, a pure random walk model may be used

$$\begin{aligned} \boldsymbol{\Phi}^{\boldsymbol{\theta}} &= \mathbf{I}_{n_{\boldsymbol{\theta}}} \qquad \boldsymbol{\theta}_k^{const} = \mathbf{0} \\ \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \boldsymbol{w}^{\boldsymbol{\theta}}{}_k \end{aligned} \tag{2.323}$$

Constant Random Parameter If the parameter vector is a random constant, all available knowledge enters the estimation problem via the parameters' initial condition

$$\boldsymbol{w}^{\boldsymbol{\theta}}{}_k = \mathbf{0} \qquad \Rightarrow \qquad \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k = \boldsymbol{\theta} \tag{2.324}$$

This formulation is closely related to the maximum a posteriori probability (MAP) estimate of section 2.2.3.4. The difference is that in addition to the parameters, state values are estimated as well.

Constant Parameter If one is to assume that the parameter vector $\boldsymbol{\theta}$ is a constant, this can formally be expressed via

$$\mathrm{E}\left[\tilde{\boldsymbol{\theta}}\right] = \boldsymbol{\theta} \tag{2.325}$$

$$\mathrm{Cov}\left[\tilde{\boldsymbol{\theta}}\right] = \mathbf{0} \tag{2.326}$$

The relation between maximum likelihood and MAP estimates resulting from this is similar to what had been discussed in section 2.2.3.3 in the context of Bayesian parameter estimation. Then, all of the above probability densities only make sense if evaluated at $\boldsymbol{\theta}$.

Still, slight differences exist compared to section 2.2.2.4, since states and parameters are estimated in one problem, instead of using a Kalman filter for state and a maximum likelihood estimator for parameter estimation.

Deterministic System If the system under consideration is deterministic, i.e. the process noise term vanishes

$$\boldsymbol{w}_k = \mathbf{0} \tag{2.327}$$

the problem reduces to that of chapter 2.2. To be fully compatible with the results of chapter 2.2, the initial states $\boldsymbol{x}_0$ can be integrated in the parameter vector. Then, only

the assumptions about the prior distribution of the parameters decide, if a maximum likelihood problem arises (constant $\boldsymbol{\theta}$) or a MAP estimate results (other prior distributions, e.g. $\theta \sim \mathcal{N}(\boldsymbol{\theta}_{prior}, \boldsymbol{\Sigma})$).

Conclusion Above special cases are summarized in table 2.1. Due to the excessive additional complexity that would be added by considering non-linearly entering process noise via $\ln\left|\left|\frac{\partial^d \mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right|\right|$, the exact formulation is mentioned for the sake of completeness in table 2.1, but will not be pursued further. For most practical cases, the influence of the non-linearity may be incorporated in the process noise covariance, which needs to be tuned manually anyways. Also, optimization vector components are given for every formulation, in addition to the governing probability density and the resulting cost function.

Most commonly uncertainty quantification is based on parameter covariance approximations via the inverse of the Fisher information matrix, the pro's and con's of which were already discussed in section 2.2.5. Meaningful results can be obtained for the most general state and parameter estimation problem, too, but its discussion is deferred to section 3.5. Some more aspects, especially the proper treatment of active constraints, are necessary to present them with the suitable level of detail.

Table 2.1: *Possible state and parameter estimation formulations;*
$\|\boldsymbol{x}\|_{\mathbf{A}} = \boldsymbol{x}^\intercal \mathbf{A} \boldsymbol{x}$ *is used for brevity;*
terms due to non-linearly entering process noise are shown in gray and will be dropped from here on

case	probability density cost function optimization vector	
general	$p(\mathbf{X}_{\bar{N}}^a, \mathbf{Z}_{\bar{N}}) =$	$p(\boldsymbol{x}_0) \cdot p(\boldsymbol{\theta}_0) \cdot \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1} \vert \boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{\theta}_{k+1} \vert \boldsymbol{x}_k, \boldsymbol{\theta}_k)$
	$J(\mathbf{z}) =$	$\frac{1}{2}\|\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0\|_{\left(\mathbf{P}_0^{\mathrm{x}}\right)^{-1}} + \frac{1}{2}\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{prior}\|_{\boldsymbol{\Sigma}^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \|\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k)\|_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$+$	$\frac{1}{2}\sum_{k=0}^{\bar{N}-1} \|\boldsymbol{w}_k\|_{\mathbf{Q}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \left\|\boldsymbol{w}^{\boldsymbol{\theta}}{}_k\right\|_{\left(\mathbf{Q}^{\boldsymbol{\theta}}{}_k\right)^{-1}} - \sum_{k=0}^{\bar{N}-1} \ln\left\vert\left\vert\frac{\partial^d \mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right\vert\right\vert - \sum_{k=0}^{\bar{N}-1} \ln\left\vert\left\vert\frac{\partial^d \mathbf{f}_k^{-1\boldsymbol{\theta}}}{\partial \boldsymbol{w}^{\boldsymbol{\theta}}{}_k}\right\vert\right\vert$
	$\mathbf{z} =$	$\left(\boldsymbol{x}_0^a, \boldsymbol{w}_0^a \ldots \boldsymbol{x}_{\bar{N}-1}^a, \boldsymbol{w}_{\bar{N}-1}^a, \boldsymbol{x}_{\bar{N}}^a\right)$
linear Gauss-Markov parameter model	$p(\mathbf{X}_{\bar{N}}^a, \mathbf{Z}_{\bar{N}}) =$	$p(\boldsymbol{x}_0) \cdot p(\boldsymbol{\theta}_0) \cdot \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1} \vert \boldsymbol{x}_k, \boldsymbol{\theta}_k) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{\theta}_{k+1} \vert \boldsymbol{\theta}_k)$
	$J(\mathbf{z}) =$	$\frac{1}{2}\|\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0\|_{\left(\mathbf{P}_0^{\mathrm{x}}\right)^{-1}} + \frac{1}{2}\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}_{prior}\|_{\boldsymbol{\Sigma}^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \|\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_k)\|_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$+$	$\frac{1}{2}\sum_{k=0}^{\bar{N}-1} \|\boldsymbol{w}_k\|_{\mathbf{Q}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}-1} \left\|\boldsymbol{w}^{\boldsymbol{\theta}}{}_k\right\|_{\left(\mathbf{Q}^{\boldsymbol{\theta}}{}_k\right)^{-1}} - \sum_{k=0}^{\bar{N}-1} \ln\left\vert\left\vert\frac{\partial^d \mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right\vert\right\vert$
	$\mathbf{z} =$	$\left(\boldsymbol{x}_0^a, \boldsymbol{w}_0^a \ldots \boldsymbol{x}_{\bar{N}-1}^a, \boldsymbol{w}_{\bar{N}-1}^a, \boldsymbol{x}_{\bar{N}}^a\right)$

constant random parameter (Gaussian prior)	$p(\mathbf{X}_{\bar{N}}, \boldsymbol{\theta}, \mathbf{Z}_{\bar{N}}) =$	$p(\boldsymbol{x}_0) \cdot p(\boldsymbol{\theta}) \cdot \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{x}_k, \boldsymbol{\theta}) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1} \vert \boldsymbol{x}_k, \boldsymbol{\theta})$
	$J(\mathbf{z}) =$	$\frac{1}{2}\Vert\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0\Vert_{\left(\mathbf{P}_0^{\mathrm{x}}\right)^{-1}} + \frac{1}{2}\Vert\boldsymbol{\theta} - \boldsymbol{\theta}_{prior}\Vert_{\boldsymbol{\Sigma}^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}}\Vert\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})\Vert_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$+$	$\frac{1}{2}\sum_{k=0}^{\bar{N}-1}\Vert\boldsymbol{w}_k\Vert_{\mathbf{Q}_k^{-1}} - \sum_{k=0}^{\bar{N}-1} \ln\left\vert\left\vert\frac{\partial^d \mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right\vert\right\vert$
	$\mathbf{z} =$	$(\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{w}_0 \ldots \boldsymbol{x}_{\bar{N}-1}, \boldsymbol{w}_{\bar{N}-1}, \boldsymbol{x}_{\bar{N}})$
constant deterministic parameter	$p(\mathbf{X}_{\bar{N}}, \mathbf{Z}_{\bar{N}}) =$	$p(\boldsymbol{x}_0) \cdot \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{x}_k, \boldsymbol{\theta}) \cdot \prod_{k=0}^{\bar{N}-1} p(\boldsymbol{x}_{k+1} \vert \boldsymbol{x}_k, \boldsymbol{\theta})$
	$J(\mathbf{z}) =$	$\frac{1}{2}\Vert\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0\Vert_{\left(\mathbf{P}_0^{\mathrm{x}}\right)^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}}\Vert\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})\Vert_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$+$	$\frac{1}{2}\sum_{k=0}^{\bar{N}-1}\Vert\boldsymbol{w}_k\Vert_{\mathbf{Q}_{k-1}^{-1}} - \sum_{k=0}^{\bar{N}-1} \ln\left\vert\left\vert\frac{\partial^d \mathbf{f}_k^{-1}}{\partial \boldsymbol{w}_k}\right\vert\right\vert$
	$\mathbf{z} =$	$(\boldsymbol{\theta}, \boldsymbol{x}_0, \boldsymbol{w}_0 \ldots \boldsymbol{x}_{\bar{N}-1}, \boldsymbol{w}_{\bar{N}-1}, \boldsymbol{x}_{\bar{N}})$
deterministic system [MAP] (Gaussian prior)	$p(\boldsymbol{\theta}, \mathbf{Z}_{\bar{N}}) =$	$p(\boldsymbol{\theta}) \cdot \prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{\theta})$
	$J(\mathbf{z}) =$	$\frac{1}{2}\Vert\boldsymbol{\theta} - \boldsymbol{\theta}_{prior}\Vert_{\boldsymbol{\Sigma}^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}}\Vert\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})\Vert_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$\mathbf{z} =$	$(\boldsymbol{\theta}, \boldsymbol{x}_0 \ldots \boldsymbol{x}_{\bar{N}})$ [single shooting: $\mathbf{z} = (\boldsymbol{\theta}, \boldsymbol{x}_0)$]
deterministic system [OEM] (constant parameter)	$p(\boldsymbol{\theta}, \mathbf{Z}_{\bar{N}}) =$	$\prod_{k=0}^{\bar{N}} p(\boldsymbol{z}_k \vert \boldsymbol{\theta})$
	$J(\mathbf{z}) =$	$\frac{1}{2}\sum_{k=0}^{\bar{N}}\Vert\boldsymbol{z}_k - \boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})\Vert_{\tilde{\mathbf{R}}_k^{-1}} + \frac{1}{2}\sum_{k=0}^{\bar{N}} \ln\left\vert\tilde{\mathbf{R}}_k\right\vert$
	$\mathbf{z} =$	$(\boldsymbol{\theta}, \boldsymbol{x}_0 \ldots \boldsymbol{x}_{\bar{N}})$ [single shooting: $\mathbf{z} = (\boldsymbol{\theta}, \boldsymbol{x}_0)$]

Part II

Application to Dynamic System Identification

On two occasions I have been asked, "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" [...] I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.

Charles Babbage, 1864 [Bab1864, Ch. 5, p. 67]

3

Implementation Aspects of System Identification Using Optimal Control Methods

In the actual application of parameter estimation algorithms, many implementation issues and possible pitfalls arise that have not yet been discussed. The following chapter collects details on those, hints at possible challenges, and proposes solutions to tackle these. Most of the following approaches are applicable to different formulations of the problem, formulated as a single shooting problem, or using full discretization. The main difference will be the components of the optimization vector $\mathbf{z}$. In the single shooting-case, it will mainly consist of model parameters $\boldsymbol{\theta}$ (and possibly initial conditions $\boldsymbol{x}_0$), whereas for fully discretized problems, $\mathbf{z}$ will additionally include state variables at all time instants $\boldsymbol{x}_k$ $k = 0 \dots \bar{N}$, and possibly process noise inputs $\boldsymbol{w}_k$ $k = 0 \dots \bar{N}$.

The remainder of the chapter is structured as follows: first some aspects regarding the cost function and its derivatives are treated (estimation of residual covariance; gradient and Hessian determination; sensitivity computations). Then aspects concerning uncertainty quantification are illustrated (parameter covariance at a constrained solution; framework for covariance computation using full discretization; singular information matrix), before implementation aspects of the optimization part are discussed (initial guess improvement; interpolation; scaling).

It has to be noted that the estimates are always extremal points of an optimization problem. This may lead to some confusion in the notation, since estimated parameters $\hat{\mathbf{z}}$ are at the same time optimal solutions $\mathbf{z}^*$. In the remainder of this work, these two

notations are interchanged rather freely, depending on the context: If interpretations stem from an estimation point of view $\hat{\square}$ will be used, whereas for optimization-based arguments $\square^*$ is sometimes more practical.

3.1 Estimation of Residual Covariance

In the foregoing chapters, it was often assumed that all covariance matrices of the problem were known, or may be specified explicitly. For many real-life applications this is cumbersome at best, if not impossible. Thus, whenever there is a possibility to obtain reasonable values otherwise, it is usually preferred. In the case of maximum likelihood (or maximum a posteriori probability (MAP)) estimation, some of the covariance matrices may actually be estimated along with the actual parameters of the problem.

Unfortunately, for reasons that will be elaborated in the end of this section, this approach may only be used for covariances on the "output-side" of the problem, i.e. the residual covariance matrix $\boldsymbol{\mathcal{B}}$ (in contrast to process noise that acts on the input side). Depending on the problem formulation, this may be:

- the covariance of the residuals $\boldsymbol{z}_k - \boldsymbol{y}_k$ of the deterministic output error problem as illustrated in section 2.2.2.2, which only consists of measurement noise
- the covariance of the residuals $\boldsymbol{z}_k - \hat{\boldsymbol{y}}_k$ of the filter error problem as illustrated in section 2.2.2.4 (if $\boldsymbol{\mathcal{B}}$ is assumed to be constant over all sampling instants, or if its average may be used); it then incorporates effects of both process and measurement noise
- the covariance of the residuals $\boldsymbol{z}_k - \hat{\boldsymbol{y}}_k$ of the joint state and parameter estimation problem of section 2.3.6, which is dominated by the measurement noise covariance $\tilde{\mathbf{R}}_k = \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^\intercal$ (again this works only if it is considered to be constant for all sampling instants, or if its average may be used)

Similar arguments as were used to determine the actual parameter values may be used to estimate $\boldsymbol{\mathcal{B}}$. Namely, an extremal point of the cost function is considered an estimate for it. To do so, the parameter vector is first partitioned in two disjoint sets corresponding to optimization and covariance parameters [MIM1985]

$$\mathbf{z}^{total} = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^{\boldsymbol{\mathcal{B}}} \end{bmatrix} \tag{3.1}$$

$$J\left(\mathbf{z}^{total}\right) = \frac{1}{2} \sum_{k=0}^{N-1} (\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^\intercal \, \boldsymbol{\mathcal{B}}\left(\mathbf{z}^{\boldsymbol{\mathcal{B}}}\right)^{-1} (\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) + \frac{N}{2} \ln\left|\boldsymbol{\mathcal{B}}\left(\mathbf{z}^{\boldsymbol{\mathcal{B}}}\right)\right| \tag{3.2}$$

where $\mathbf{z}^{\boldsymbol{\mathcal{B}}}$ represents the independent elements of $\boldsymbol{\mathcal{B}}$. Only the upper/lower triangular part of $\boldsymbol{\mathcal{B}}$ needs to be considered, since it is symmetric by definition. Thus only $\sum_l^{n_z} l = \frac{l(l+1)}{2}$ independent elements remain. In the case of combined parameter and state estimation, above expression represents only that part of the cost function, which

is relevant for covariance estimation. Whereas for output error estimation, this is the complete cost function[1].

Trying to minimize above cost function directly leads to problems due to the strong interdependence between $\mathbf{z}$ and $\mathbf{z}^{\boldsymbol{\mathcal{B}}}$ [Jat2006, Ch. 4V]. To circumvent this one can realize that both the derivatives with respect to the optimization parameters $\mathbf{z}$ as well as the covariance parameters $\mathbf{z}^{\boldsymbol{\mathcal{B}}}$ will have to vanish at the optimum. For a fixed value of $\mathbf{z}$, the stationary point of J w.r.t. $\mathbf{z}^{\boldsymbol{\mathcal{B}}}$ may be stated analytically: The gradient $\frac{\partial J}{\partial \mathbf{z}^{\boldsymbol{\mathcal{B}}}}$ is zero, if the gradient $\frac{\partial J}{\partial \boldsymbol{\mathcal{B}}}$ is zero, since $\frac{\partial \boldsymbol{\mathcal{B}}}{\partial \mathbf{z}^{\boldsymbol{\mathcal{B}}}}$ only contains information about the distribution of the parameters within $\boldsymbol{\mathcal{B}}$. Thus, only the stationary point of J w.r.t. $\boldsymbol{\mathcal{B}}$ needs to be determined. Using the fact that for a scalar a it holds $a = \mathrm{tr}[a]$, the cost function derivative becomes

$$\begin{aligned}\frac{\partial J}{\partial \boldsymbol{\mathcal{B}}} &= \frac{\partial}{\partial \boldsymbol{\mathcal{B}}}\mathrm{tr}[J] \\ &= \frac{\partial}{\partial \boldsymbol{\mathcal{B}}}\mathrm{tr}\left[\frac{1}{2}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^{\intercal}\,\boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}|\right]\end{aligned} \tag{3.3}$$

Now, the trace is a linear operator, and using its cyclic permutation property yields [MK2016, Ch. 6.1]

$$\frac{\partial J}{\partial \boldsymbol{\mathcal{B}}} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\mathcal{B}}}\mathrm{tr}\left[\boldsymbol{\mathcal{B}}^{-1}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^{\intercal}\right] + \frac{\partial}{\partial \boldsymbol{\mathcal{B}}}\underbrace{\left(\frac{N}{2}\ln|\boldsymbol{\mathcal{B}}|\right)}_{=\mathrm{tr}\left[\frac{N}{2}\ln|\boldsymbol{\mathcal{B}}|\right]} \tag{3.4}$$

Appendix A.1 lists some useful equalities for matrix derivatives, where equations (A.9) and (A.10) may be used to eventually arrive at [MIM1985, Ch. 5] [WP1997, Example 3.4] [GP1977, Ch. 5.4] [CJ2012, Ch. 2]

$$\frac{\partial J}{\partial \boldsymbol{\mathcal{B}}} = -\frac{1}{2}\boldsymbol{\mathcal{B}}^{-1}\left(\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^{\intercal}\right)\boldsymbol{\mathcal{B}}^{-1} + \frac{N}{2}\boldsymbol{\mathcal{B}}^{-1} \stackrel{!}{=} \mathbf{0} \tag{3.5}$$

$$\Rightarrow \hat{\boldsymbol{\mathcal{B}}} = \frac{1}{N}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^{\intercal} = \frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{r}_k(\mathbf{z})\,\boldsymbol{r}_k(\mathbf{z})^{\intercal} \tag{3.6}$$

This resembles the estimate for the sample variance (A.160) closely and is often called the "maximum likelihood variance estimate".

The stationary point of J w.r.t. $\mathbf{z}$ can in general not be stated analytically, and has to be determined numerically. This leads to the following two-step optimization algorithm, which also overcomes above mentioned problems due to the strong interdependence of $\mathbf{z}$ and $\mathbf{z}^{\boldsymbol{\mathcal{B}}}$ [MIM1985] [MK2016, Ch. 6.1] [Jat2015, Ch. 4]:

1. compute an estimate of the covariance matrix $\hat{\boldsymbol{\mathcal{B}}}_i$ for the current value of the parameters $\mathbf{z}_i$ and consider it as constant, independent of $\mathbf{z}_i$ for this step

[1] If the covariance matrix $\boldsymbol{\mathcal{B}}$ were known, the cost function simplifies further, since $\frac{N}{2}\ln\left|\boldsymbol{\mathcal{B}}(\mathbf{z}^{\boldsymbol{\mathcal{B}}})\right|$ will be constant and may thus be dropped. The task then reduces to a weighted, non-linear least-squares problem as illustrated in section 2.2.4.5 [WP1997, Example 3.4].

2. compute a parameter update $\Delta\mathbf{z}_i$ using the current estimate of the covariance matrix $\hat{\boldsymbol{\mathcal{B}}}_i$
3. set $\mathbf{z}_{i+1} = \mathbf{z}_i + \Delta\mathbf{z}_i$ and iterate from 1. until convergence

Not only is this approach of estimating $\boldsymbol{\mathcal{B}}$ interesting for an unknown covariance matrix. It may also add more flexibility to the solution algorithm: during the iteration, when parameters are possibly far away from their optimum, the resulting model deficiencies are automatically included in $\hat{\boldsymbol{\mathcal{B}}}$. This may improve convergence of the overall approach, compared to a fixed $\boldsymbol{\mathcal{B}}$, which does not take the current characteristics of the residuals into account.

Above two steps can be combined by plugging the maximum likelihood estimate of the covariance matrix (3.6) back into the expression for the cost function (3.2). Again, using the trace and its cyclic permutation property leads to [MIM1985, Ch. 5]

$$\begin{aligned} J(\mathbf{z}) = \mathrm{tr}[J(\mathbf{z})] &= \frac{1}{2}\mathrm{tr}\left[\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^\intercal\, \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))\right] + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| \\ &= \underbrace{\frac{1}{2}\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1} N \frac{1}{N}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z}))^\intercal\right]}_{=\frac{N}{2}\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}\hat{\boldsymbol{\mathcal{B}}}\right]} + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| \\ &= \frac{N n_{\boldsymbol{z}}}{2} + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| \end{aligned} \tag{3.7}$$

Dropping the constant term results in

$$\hat{J}(\mathbf{z}) = \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| \tag{3.8}$$

Optimizing above formulation directly leads to the same result as the two step approach mentioned before. Sometimes, an even simpler form of cost function is used, that solely considers the determinant of $\hat{\boldsymbol{\mathcal{B}}}$

$$\tilde{J}(\mathbf{z}) = \exp\left(\frac{2}{N}\cdot\hat{J}\right) = \left|\hat{\boldsymbol{\mathcal{B}}}\right| \tag{3.9}$$

$$\hat{J}(\mathbf{z}) = \frac{N}{2}\ln\left(\tilde{J}\right) \tag{3.10}$$

However, since a non-linear transformation is applied, above formulation of $\tilde{J}$ is not compatible with the combined state and parameter estimation formulations of section 2.3.6. There, the separate contributions due to process noise $\boldsymbol{w}_k$, and prior information $\boldsymbol{\theta}_{prior}$, $\bar{\boldsymbol{x}}_0$ appear as summand of the cost function, which would need to be incorporated in the non-linear transformation. The additional complexity then discourages from its application.

Both formulations are valid cost functions for maximum likelihood estimation and aid to solve the same problem. Also, their gradients and Hessians are closely connected, as will be seen in section 3.2. The choice of cost function eventually depends on the numerical properties of the solution algorithm, and the actual problem at hand.

As was already mentioned, this idea for covariance estimation may only be applied on the output side of the problem. If process noise elements $\boldsymbol{w}_k$ are determined directly during combined state and parameter estimation, their covariance matrix $\mathbf{Q}$ cannot be estimated along with their numerical values: The cost function would involve the determinant of $\mathbf{Q}$, a minimum of which can always be achieved by perfectly correlating two noise sequences. Then the covariance matrix estimate $\frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{w}_k\boldsymbol{w}_k^\mathsf{T}$ will have linearly dependent rows/columns, and its determinant will be at its absolute minimum of zero. Some simple tests have shown that the Non-Linear Programming (NLP) solvers in use find this undesired solution quite robustly, thus showing that the approach of this section cannot work for process noise covariance estimation.

This section showed, how good covariance estimates can be obtained, based on the maximum likelihood idea. The approach relieves the analyst from having to specify them explicitly, while still keeping the other aspects of the problem unchanged.

3.2 Computation of Derivatives

The arising NLP problems will exclusively be solved using gradient based optimization algorithms (see section 2.1 for details). Convergence properties of these algorithms are greatly improved, if analytic gradients and Hessians of the cost function are supplied [WP1997, Bet2010].

Most of the derivatives of the cost functions in table 4.1 are straight forward to compute, since they are merely weighted sums of squares. Also the cost function formulations themselves are very modular in the sense that the influences of process noise, outputs, and prior information are separated in distinct terms.

Only the dependency of the cost function gradient on the optimization parameters via the model outputs is more involved. It appears in a term of the form

$$J(\mathbf{z}) = \frac{1}{2}\sum_{k=0}^{N-1}(\boldsymbol{z}_k - \boldsymbol{y}_k)^\mathsf{T}\boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k) + \frac{1}{2}\sum_{k=0}^{N-1}\ln|\boldsymbol{\mathcal{B}}| \tag{3.11}$$

possibly together with other summands incorporating the effect of prior knowledge or process noise.

In order to obtain meaningful results, the first step is to apply the chain rule, while keeping the layout convention illustrated in appendix A.1.1 in mind

$$\frac{\partial J}{\partial \mathbf{z}} = \sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T}\frac{\partial J}{\partial \boldsymbol{y}_k} \tag{3.12}$$

The terms $\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}$ represent the output sensitivities, which indicate, how sensitive an output is to a change in a certain parameter. More details on their computation will be presented in section 3.3.

Taking the second derivative of the gradient results in the Hessian

$$\frac{\partial^2 J}{\partial \mathbf{z}^2} = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T} \frac{\partial^2 J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\mathsf{T}}\left(\frac{\partial \boldsymbol{y}_l(\mathbf{z})}{\partial \mathbf{z}}\right) + \sum_{k=0}^{N-1}\sum_{m=1}^{n_y} \frac{\partial^2 [\boldsymbol{y}_k(\mathbf{z})]_{(m)}}{\partial \mathbf{z}^2}\left[\frac{\partial J}{\partial \boldsymbol{y}_k}\right]_{(m)} \tag{3.13}$$

The second part of above equations, involving second order output sensitivities, is often neglected. This simplification is justified, since for the cost functions in use, the higher order term is multiplied by the residuals $\boldsymbol{r}(\mathbf{z}) = \boldsymbol{z}_k - \hat{\boldsymbol{y}}_k(\mathbf{z})$, as will be seen in the following section. This difference will be small near an optimum (and of zero mean if the statistical background is valid), thus the approximation gets better the closer the optimization is to the final result [MK2016, Ch. 6.2]. It greatly simplifies computation, since higher order derivatives are costly to compute, and if done numerically, are prone to round-off errors. Eventually, the Hessian is approximated as [GP1977, Ch. 5.4] [Jat2015, Ch. 4]

$$\frac{\partial^2 J}{\partial \mathbf{z}^2} \approx \sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T} \frac{\partial^2 J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\mathsf{T}}\left(\frac{\partial \boldsymbol{y}_l(\mathbf{z})}{\partial \mathbf{z}}\right) \tag{3.14}$$

If mixed second derivatives vanish $\frac{\partial^2 J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\mathsf{T}} = \mathbf{0} \quad k \neq l$, the formulation simplifies further

$$\frac{\partial^2 J}{\partial \mathbf{z}^2} \approx \sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T} \frac{\partial^2 J}{\partial {\boldsymbol{y}_k}^2}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right) \tag{3.15}$$

3.2.1 Derivatives for Constant Residual Covariance Matrix

If the covariance matrix $\boldsymbol{\mathcal{B}}$ is known and constant, the cost function gradient is

$$\frac{\partial J}{\partial \boldsymbol{y}_k} = -\boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.16}$$

$$\Rightarrow \frac{\partial J}{\partial \mathbf{z}} = -\sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T} \boldsymbol{\mathcal{B}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.17}$$

Together with the approximation of the Hessian (3.14), the matrix of second derivatives becomes

$$\frac{\partial^2 J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\mathsf{T}} = \boldsymbol{\mathcal{B}}^{-1} \cdot \delta_{kl} = \begin{cases} \boldsymbol{\mathcal{B}}^{-1} & \text{for } k = l \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{3.18}$$

$$\Rightarrow \frac{\partial^2 J}{\partial \mathbf{z}^2} = \sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^\mathsf{T} \boldsymbol{\mathcal{B}}^{-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right) \tag{3.19}$$

3.2.2 Derivatives for Combined Residual Covariance and Parameter Estimation

If the covariance matrix is estimated along with the parameters (using its maximum likelihood estimate (3.6)), the simplified cost function of equation (3.8) may be used for

derivative computations. Details on the derivation of the gradient expression can be found in Appendix E. It results in

$$\frac{\partial \hat{J}}{\partial \boldsymbol{y}_k} = -\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.20}$$

$$\Rightarrow \frac{\partial \hat{J}}{\partial \mathbf{z}} = -\sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^{\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.21}$$

It is noteworthy, that the result is the same as in the case of a constant residual covariance matrix, i.e. the same as if $\hat{\boldsymbol{\mathcal{B}}}$ was independent of the model outputs.

The gradient of the simplified cost function $\tilde{J} = \left|\hat{\boldsymbol{\mathcal{B}}}\right|$ of equation (3.9) arises via the chain rule

$$\frac{\partial \tilde{J}}{\partial \hat{J}} = \frac{\partial \exp\left(\frac{2}{N}\cdot \hat{J}\right)}{\partial \hat{J}} = \frac{2}{N}\exp\left(\frac{2}{N}\cdot \hat{J}\right) = \frac{2}{N}\left|\hat{\boldsymbol{\mathcal{B}}}\right| \tag{3.22}$$

$$\frac{\partial \tilde{J}}{\partial \boldsymbol{y}_k} = \frac{\partial \tilde{J}}{\partial \hat{J}}\cdot\frac{\partial \hat{J}}{\partial \boldsymbol{y}_k} = -\frac{2}{N}\left|\hat{\boldsymbol{\mathcal{B}}}\right| \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.23}$$

$$\Rightarrow \frac{\partial \tilde{J}}{\partial \mathbf{z}} = -\frac{2}{N}\left|\hat{\boldsymbol{\mathcal{B}}}\right| \sum_{k=0}^{N-1}\left(\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial \mathbf{z}}\right)^{\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) \tag{3.24}$$

Details on the derivation of the Hessian in this case can be found in Appendix E, the result is

$$\frac{\partial^2 \hat{J}}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^{\intercal}} = -\frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^{\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^{\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\right) + \hat{\boldsymbol{\mathcal{B}}}^{-1}\delta_{kj} \tag{3.25}$$

None of the elements of the term in parenthesis grows with sample size N, thus for large N the first part of above equation will shrink. Furthermore, close to the optimum, the residuals $\boldsymbol{r}_l$ and $\boldsymbol{r}_k$ will be small themselves. Additionally, in some cases it might be necessary to store the complete Hessian, which quickly becomes unfeasible for large problems, if all elements are considered, since it is of dimension $(N \cdot n_{\boldsymbol{y}})^2$.

Eventually, neglecting the first term, the Hessian can be approximated as

$$\frac{\partial^2 \hat{J}}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_j^{\intercal}} \approx \hat{\boldsymbol{\mathcal{B}}}^{-1}\delta_{kj} \tag{3.26}$$

which is the same expression as resulted from a constant, known residual covariance matrix $\boldsymbol{\mathcal{B}}$ and is commonly found in textbooks on the topic [Jat2015, Ch. 4]. Above argument (using a correct derivative and inspecting the terms to eventually obtain a tractable Hessian approximation) has not yet been encountered by the author. However, it underlines that the common approximation is a valid one.

For the simplified cost function this becomes

$$\frac{\partial^2 \tilde{J}}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_j^{\intercal}} = \frac{\partial^2 \tilde{J}}{\partial \hat{J}^2}\cdot\frac{\partial \hat{J}}{\partial \boldsymbol{y}_k}\cdot\frac{\partial \hat{J}}{\partial \boldsymbol{y}_j}^{\intercal} + \left(\frac{\partial \tilde{J}}{\partial \hat{J}}\right)^2 \frac{\partial^2 \hat{J}}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^{\intercal}} \tag{3.27}$$

which shows another decisive disadvantage over the original formulation: Due to the first term in above formulation, the Hessian will in general be fully dense and may not as easily be reduced to its main diagonal elements. Thus for large problems, if the Hessian is to be used, the computational burden for this formulation is considerably larger compared to simply using $\hat{J}$.

This section showed, how the analytic gradient and Hessian of different cost functions formulation w.r.t. the model outputs may be computed. Also, it was shown that the resulting cost function gradient has the same form for pre-defined and estimated residual covariance. Further, a novel argument was presented, why the cost function Hessian may be computed in the same manner for both cases as well. Thus, no algorithmic changes are necessary whether the residual covariance matrix is defined explicitly or estimated from the data. Together with the output sensitivities, which are to be discussed in the next section, this forms the basis for an analytic determination of gradient and Hessian w.r.t. the optimization parameters. This greatly improves the convergence properties of the underlying NLP solvers.

3.3 Output Sensitivities

In all cases to be discussed, the notion of "sensitivities" is central. In general, they give a hint at how the output or state trajectory would respond to a change in any of the parameters, be it a model parameter, or a state or input at any point in time. This information, together with the results of the last section, is then used to compute gradient and Hessian information.

When useful, the output sensitivities will be abbreviated as

$$ {}^{y}_{\mathbf{z}}\mathbf{S}_k = \frac{\partial \boldsymbol{y}_k}{\partial \mathbf{z}} \tag{3.28} $$

One output sensitivity term $[{}^{y}_{\mathbf{z}}\mathbf{S}_k]_{(l,j)} = \frac{\partial [\boldsymbol{y}_k]_{(l)}(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}}$ has three influencing dimensions:

1. the sample index $k = 0, \ldots, N-1$,
2. the index of the output $l = 1, \ldots, n_{\boldsymbol{y}}$,
3. the index of the parameter $j = 1, \ldots, n_{\mathbf{z}}$.

They essentially describe, how a change in an optimization parameter influences the model outputs at different times. Thus the output sensitivity may describe how changing a model parameter $[\boldsymbol{\theta}]_{(\tilde{j})} = [\mathbf{z}]_{(j)}$ may influence the l-th output at time k (this would be captured in $[{}^{y}_{\mathbf{z}}\mathbf{S}_k]_{(l,j)}$). For fully discretized problems, the sensitivities may also contain measures on how changing a state at one point in time t_n, $[\boldsymbol{x}_n]_{(\tilde{j})} = [\mathbf{z}]_{(j)}$, may influence the outputs at another point in time.

In above sections on the computation of the cost function derivatives, it was assumed that output sensitivities can be determined, which is now elaborated in more detail.

3.3.1 Numeric Determination

The brute-force method to determine output sensitivities is via numeric, finite differences. The definition of the partial derivative with respect to one element of $[\mathbf{z}]_{(j)}$ $j = 1 \dots n_{\mathbf{z}}$ is

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} = \lim_{h \to 0} \frac{\boldsymbol{y}_k(\mathbf{z} + \boldsymbol{e}_j \cdot h) - \boldsymbol{y}_k(\mathbf{z})}{h} \tag{3.29}$$

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} = \lim_{h \to 0} \frac{\boldsymbol{y}_k(\mathbf{z}) - \boldsymbol{y}_k(\mathbf{z} - \boldsymbol{e}_j \cdot h)}{h} \tag{3.30}$$

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} = \lim_{h \to 0} \frac{\boldsymbol{y}_k(\mathbf{z} + \boldsymbol{e}_j \cdot h) - \boldsymbol{y}_k(\mathbf{z} - \boldsymbol{e}_j \cdot h)}{2h} \tag{3.31}$$

Approximating the limit case with a small, finite h, forward, backward or central finite differences arise [Bet2010, Ch. 1] [Jat2006, Ch. 4] [MK2016, Ch. 6.3]

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} \approx \frac{\boldsymbol{y}_k(\mathbf{z} + \boldsymbol{e}_j \cdot h) - \boldsymbol{y}_k(\mathbf{z})}{h} \tag{3.32}$$

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} \approx \frac{\boldsymbol{y}_k(\mathbf{z}) - \boldsymbol{y}_k(\mathbf{z} - \boldsymbol{e}_j \cdot h)}{h} \tag{3.33}$$

$$\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}} \approx \frac{\boldsymbol{y}_k(\mathbf{z} + \boldsymbol{e}_j \cdot h) - \boldsymbol{y}_k(\mathbf{z} - \boldsymbol{e}_j \cdot h)}{2h} \tag{3.34}$$

In order to actually compute those sensitivities, the model has to be evaluated once (twice for central finite differences) per parameter with perturbed values. This approach is very flexible, since only model evaluations are involved, and no information about the internal structure of the model is necessary. However this flexibility comes at the cost of $n_{\mathbf{z}} + 1$ for forward/backward, and $2n_{\mathbf{z}}$ model evaluations for central finite differences. Usually, despite the additional computations, central finite differences are preferred, because of their significantly lower approximation error [MK2016, Ch. 6.3].

Nevertheless, the additional computations are the greatest downside of the approach, since it might take considerable additional time, if $n_{\mathbf{z}}$ becomes large. Also, round-off errors are very common since h has to be chosen very small, and thus some differences might fall below a computers numerical resolution, distorting the result.

For the choice of h, some authors propose relative perturbations, e.g. MORELLI and KLEIN mention that 1% of a parameters nominal value works well for many aircraft applications [MK2016, Ch. 6.3]. Thus for the part of the optimization parameter vector pertaining to aerodynamic coefficients they advise to use

$$h = 0.01 \cdot [\mathbf{z}]_{(j)} = 0.01 \cdot [\boldsymbol{\theta}]_{(\tilde{j})} \tag{3.35}$$

as perturbation for the computation of $\frac{\partial \boldsymbol{y}_k(\mathbf{z})}{\partial [\mathbf{z}]_{(j)}}$. If the nominal value is too close to zero, the relative is replaced by an absolute perturbation.

3.3.2 Analytic Determination - Static Models

For static systems, i.e. systems that can be modeled as purely analytically depending on inputs and parameters

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{u}_k, \mathbf{z}) \tag{3.36}$$

computing analytic output sensitivities might be cumbersome but is straightforward.

$${}^{y}_{\mathbf{z}}\mathbf{S}_k = \frac{\partial \boldsymbol{y}_k}{\partial \mathbf{z}} = \frac{\partial \boldsymbol{h}(\boldsymbol{u}_k, \mathbf{z})}{\partial \mathbf{z}} \tag{3.37}$$

Problems may arise with dependencies that are not differentiable (e.g. saturations) or lookup table formulations. These pose problems for numerical differentiation, too, but are hidden behind the finite difference approach. Especially the cumbersome differentiation by hand can be circumvented by using automatic differentiation algorithms, as is done in the simulation model builder of FALCON.m [RBG+2018].

3.3.3 Analytic Determination - Dynamic Models

For dynamic systems in the form of first order ordinary differential equations (ODEs), the computation of analytic output sensitivities is not as straightforward anymore. A possible solution can be obtained via the sensitivity equations [CJ2012, Ch. 3.3] [Jat2006, Ch. 4] [Ger2018, Ch. 5.2] [WP1997, Ch. 4.3].

They are based on an initial value problem with a system of first order ODEs in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{3.38}$$

$$\boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \tag{3.39}$$

$$k = 0, \ldots, N-1 \qquad t_k \in \{t_0, \ldots t_{N-1}\}$$

Applying the chain rule, the individual influences on the output sensitivities due to a variation in states, inputs or model parameters can be stated as

$$\begin{aligned} {}^{y}_{\mathbf{z}}\mathbf{S}_k = \frac{\partial \boldsymbol{y}_k}{\partial \mathbf{z}} &= \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{x}_k} \cdot \frac{\partial \boldsymbol{x}_k}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{u}_k} \cdot \frac{\partial \boldsymbol{u}_k}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}} \\ &= \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{x}_k} \cdot {}^{x}_{\mathbf{z}}\mathbf{S}_k + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{u}_k} \cdot {}^{u}_{\mathbf{z}}\mathbf{S}_k + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \mathbf{z}} \cdot {}^{\theta}_{\mathbf{z}}\mathbf{S} \end{aligned} \tag{3.40}$$

Both, ${}^{u}_{\mathbf{z}}\mathbf{S}_k$ and ${}^{\theta}_{\mathbf{z}}\mathbf{S}_k$ consist of matrices that are mainly zero. Only at those indices, where the respective parameters ($\boldsymbol{u}_k$ or $\boldsymbol{\theta}$) are located within the optimization vector $\mathbf{z}$, an identity matrix is to be found. Assuming the following structure for $\mathbf{z}$

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\theta}^\intercal & \boldsymbol{x}_0^\intercal & \boldsymbol{u}_0^\intercal & \cdots & \boldsymbol{x}_{N-1}^\intercal & \boldsymbol{u}_{N-1}^\intercal \end{bmatrix}^\intercal \tag{3.41}$$

they are

$$ {}^{\theta}_{\mathbf{z}}\mathbf{S} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \end{bmatrix} \tag{3.42} $$

$$ {}^{u}_{\mathbf{z}}\mathbf{S}_k = \begin{bmatrix} \overset{\boldsymbol{\theta}}{\mathbf{0}} & \overset{\boldsymbol{x}_0}{\mathbf{0}} & \overset{\boldsymbol{u}_0}{\mathbf{0}} & \cdots & \overset{\boldsymbol{x}_k}{\mathbf{0}} & \overset{\boldsymbol{u}_k}{\mathbf{I}_{n_{\boldsymbol{u}}}} & \overset{\boldsymbol{x}_{k+1}}{\mathbf{0}} & \cdots & \overset{\boldsymbol{x}_{N-1}}{\mathbf{0}} & \overset{\boldsymbol{u}_{N-1}}{\mathbf{0}} \end{bmatrix} \tag{3.43} $$

If none of the inputs $\boldsymbol{u}_k$ are part of the optimization vector $\mathbf{z}$ (i.e. all inputs at all time instants are perfectly known), the input sensitivity vanishes ${}^{u}_{\mathbf{z}}\mathbf{S}_k = \mathbf{0}\ \forall k$ and the formulation simplifies to

$$ \begin{aligned} {}^{y}_{\mathbf{z}}\mathbf{S} &= \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{x}_k} \cdot \frac{\partial \boldsymbol{x}_k}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}} \\ &= \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{x}_k} \cdot {}^{x}_{\mathbf{z}}\mathbf{S} + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot {}^{\theta}_{\mathbf{z}}\mathbf{S} \end{aligned} \tag{3.44} $$

This is the common case for standard approaches in system identification, since all inputs are measured and thus not subject to optimization. Only if process noise inputs are part of the problem, the input sensitivities ${}^{y}_{w}\mathbf{S}_k$ will have to be considered explicitly.

Again, the computation of the partial derivatives of $\boldsymbol{g}$ may be quite cumbersome, but are straightforward. This only leaves the determination of the state sensitivities ${}^{x}_{\mathbf{z}}\mathbf{S} = \frac{\partial \boldsymbol{x}}{\partial \mathbf{z}}$. Taking the derivative with respect to time, and assuming sufficient differentiability, the order of differentiation can be interchanged to arrive at

$$ \begin{aligned} \frac{d}{dt}\,{}^{x}_{\mathbf{z}}\mathbf{S} = \frac{d}{dt}\frac{\partial \boldsymbol{x}}{\partial \mathbf{z}} &= \frac{\partial}{\partial \mathbf{z}}\frac{d\boldsymbol{x}}{dt} = \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \mathbf{z}} = \\ &= \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{u}} \cdot \frac{\partial \boldsymbol{u}}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}} \\ &= \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{x}} \cdot {}^{x}_{\mathbf{z}}\mathbf{S} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{u}} \cdot {}^{u}_{\mathbf{z}}\mathbf{S} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot {}^{\theta}_{\mathbf{z}}\mathbf{S} \end{aligned} \tag{3.45} $$

with the definitions of ${}^{u}_{\mathbf{z}}\mathbf{S}$ and ${}^{\theta}_{\mathbf{z}}\mathbf{S}$ as in above equations (3.42) and (3.43). Again, for perfectly known inputs ${}^{u}_{\mathbf{z}}\mathbf{S} = \mathbf{0}$ the expression simplifies to

$$ \begin{aligned} \frac{d}{dt}\,{}^{x}_{\mathbf{z}}\mathbf{S} &= \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \mathbf{z}} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}} \\ &= \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \boldsymbol{x}} \cdot {}^{x}_{\mathbf{z}}\mathbf{S} + \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})}{\partial \mathbf{z}} \cdot {}^{\theta}_{\mathbf{z}}\mathbf{S} \end{aligned} \tag{3.46} $$

This now is a linear(!) time-variant matrix-valued ODE, which can be integrated along with the system equations using any numerical integration method. The initial condition of the state sensitivity equation is given by

$$ {}^{x}_{\mathbf{z}}\mathbf{S}_0 = \frac{\partial \boldsymbol{x}_0}{\partial \mathbf{z}} \tag{3.47} $$

which is zero, if the initial conditions $\boldsymbol{x}_0$ are not subject to optimization, and is structured similarly to ${}^{\theta}_{\mathbf{z}}\mathbf{S}$ and ${}^{u}_{\mathbf{z}}\mathbf{S}$ otherwise.

The solution to the state sensitivity equation thus consists of a transient of $\frac{\partial x_0}{\partial \mathbf{z}}$, which abates according to the linearized system equations (homogeneous part of the solution). The other part of the solution is influenced by a forcing term ($\frac{\partial f}{\partial u} \cdot {}^{u}_{\mathbf{z}}\mathbf{S} + \frac{\partial f}{\partial \mathbf{z}} {}^{\theta}_{\mathbf{z}}\mathbf{S}$), which covers the direct dependency of the system matrix on the optimization variables (particular part of the solution). It is noteworthy that even for non-linear systems, the state sensitivity ODE remains linear, but time-varying. Furthermore, the differential equations for different entries of $\mathbf{z}$, i.e. different columns of ${}^{x}_{\mathbf{z}}\mathbf{S}$ are *independent* [WP1997, Ch. 4.3].

The analytic determination of the partial derivatives of the system and output equations might be cumbersome and suffer from the same difficulties as in the case of static systems, see section 3.3.2. Furthermore, integrating a matrix-valued ODE in time can be computationally expensive. However, for large numbers of parameters this approach is faster compared to numeric finite differences. The greatest advantage is that the results are analytically correct and no approximations are involved. This last aspect is especially useful if the optimization algorithm, which is used to solve the estimation problem, is sensitive to the quality of the computed gradient, as is e.g. the case for quasi-Newton approaches [WP1997, Ch. 4.3].

The foregoing section illustrated, how to determine state and output sensitivities, either numerically, or by explicitly integrating the matrix valued sensitivity differential equations. The results may then be used to eventually determine analytic expressions for cost function gradient and Hessian. Furthermore, they play an important role in the determination of parameter covariances.

3.4 Consequences of Constrained Parameter Estimation on Uncertainty Quantification

If constraints on the parameter estimates are to be considered, the question arises on how to incorporate those in the uncertainty quantification process, i.e. how to formulate the Fisher information matrix in these cases. GORMAN and HERO developed an approach to do so in the case of maximum likelihood estimation [GH1990].

A "constraint Fisher information matrix" is derived, based on the Jacobian of the active equality constraints $\tilde{c}$, and the unconstrained Fisher information matrix. Here, $\tilde{c}$ comprises of all constraints that are active at the final solution, i.e. equality constraints (including integration defects) and active inequality constraints (bounds on optimization parameters, path constraints, etc.). The details of their work are beyond the scope of this thesis, only their main results will be illustrated and linked to the corresponding content in section 2.2.4.4 on weighted least squares estimation.

GORMAN and HERO first consider the constraint Jacobian $\left.\frac{\partial \tilde{c}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$, which may be used to define the tangent hyperplane $\mathcal{T}(\mathbf{z}^*)$ onto the hyper-surface spanned by the con-

straints $\tilde{\boldsymbol{c}}$

$$\mathcal{T}(\mathbf{z}^*) = \left\{ \mathbf{z} \in \mathbb{R}^{n_z} : \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} (\mathbf{z} - \mathbf{z}^*) = \mathbf{0} \right\} \tag{3.48}$$

Then, an orthogonal projection matrix is built, which projects any direction in $\mathbb{R}^{n_z}$ onto that tangent hyperplane, i.e. all vectors are mapped onto the free directions that are *not* constrained via $\tilde{\boldsymbol{c}}$ [GH1990, Remark 2]

$$\mathbf{P}_{\tilde{c}} = \mathbf{I}_{n_\mathbf{z}} - \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}^\intercal \left(\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}^\intercal \right)^\dagger \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} \tag{3.49}$$

where $\mathbf{A}^\dagger$ indicates the Moore-Penrose pseudo inverse. This is then used to project the rows and columns of the unconstrained Fisher information matrix onto that hyperplane [GH1990], computing the pseudo-inverse of the result and then project it back onto the original parameter space

$$\boldsymbol{\mathcal{F}}^{-1}_{constr.} = \mathbf{P}_{\tilde{c}} (\mathbf{P}_{\tilde{c}}^\intercal \boldsymbol{\mathcal{F}} \mathbf{P}_{\tilde{c}})^\dagger \mathbf{P}_{\tilde{c}}^\intercal \tag{3.50}$$

GORMAN and HERO give an intuitive geometrical interpretation [GH1990]: The Fisher information matrix for the unconstrained case is related to the expected value of the Hessian of the negative log-likelihood function. It thus can be interpreted as the average curvature of the log-likelihood function (see section 2.2.2.1). If the parameter space is now equality-constrained to be a hypersurface in $\mathbb{R}^{n_z}$, only certain constrained trajectories of the parameters are possible, and thus the average curvature changes. By projecting the rows- and columns of the Fisher information matrix onto the tangent plane of this hyper-surface, this change in curvature is taken into account.

This result is closely related to equality constrained least-squares estimation as was illustrated in section 2.2.4.4: Consider a linear approximation of the active equality constraints $\tilde{\boldsymbol{c}}$ at the final solution of the estimation problem

$$\tilde{\boldsymbol{c}}(\mathbf{z}) \approx \tilde{\boldsymbol{c}}(\mathbf{z}^*) + \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} (\mathbf{z}^* - \mathbf{z}) = \mathbf{0} \tag{3.51}$$

If $\mathbf{Z}$ then is a basis for the null-space of $\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ it is shown in Appendix C.4 that GORMAN and HERO's constrained Fisher information matrix may also be expressed as[2]

$$\boldsymbol{\mathcal{F}}^{-1}_{constr.} = \mathbf{Z} (\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z})^{-1} \mathbf{Z}^\intercal \tag{3.52}$$

This is the same result as that of section 2.2.4.4. The reason is that the weighted least-squares estimator is a maximum likelihood estimator, thus the results for the covariance estimate need to be equivalent.

Above formulation involving $\mathbf{Z}$ has a few computational advantages over the explicit use of the projection approach: the expression for the condensed Fisher information matrix $\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z}$ has in general full rank, i.e. no pseudo inverse needs to be computed. Furthermore, it has the size of the free directions, which may be considerably

[2] A proof for the special case of ortho-normal $\mathbf{Z}$ can be found in [SN1998].

Example 3.1: Covariance matrix modification for fixed parameters
If a parameter element $[\mathbf{z}]_{(j)}$ is fixed, the corresponding constraint gradient is

$$\frac{\partial[\tilde{\boldsymbol{c}}]_{(l)}}{\partial \mathbf{z}} = \boldsymbol{e}_j$$

A null-space basis can be constructed as

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{j-1} & \boldsymbol{e}_{j+1} & \cdots & \boldsymbol{e}_{n_{\mathbf{z}}} \end{bmatrix}$$

The condensed Fisher information matrix $\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z}$ then only consists of the free elements, and $\mathbf{Z}(\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z})^{-1} \mathbf{Z}^\intercal$ will consist only of zeros in the j-th row and column. This can be interpreted such that there is no uncertainty in the respective parameter, which is intuitive, since it had been fixed a priori.

smaller than the original, unconstrained problem, making its inversion computationally cheaper.

Also, above constrained Fisher information matrix $\boldsymbol{\mathcal{F}}_{constr.}$ is densely populated, which may lead to difficulties for large problems. However, the formulation via the null-space basis $\mathbf{Z}$ allows for an economical way to store that information, in the smaller matrix $(\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z})^{-1}$ and the null-space basis $\mathbf{Z}$. Relevant sub-matrices are only computed if necessary. All these aspects will come in very handy, when treating estimation in a full discretization setting, since then the problems will become very large, but only few free directions in the parameter space will remain. This leaves the formulation via the null-space matrix the only feasible way. Two simple example cases (example 3.1 and example 3.2) are used to illustrate the consequences of a constrained Fisher information matrix.

The necessity of considering constraints in uncertainty quantification is obvious for fully discretized problems, where the system dynamics are incorporated as equality constraints. However, the consideration of inequalities on the model parameters $\boldsymbol{\theta}$ may be inadvisable according to WALTER: If the inequality constraints are active in the final solution, the model structure $\mathcal{M}(\boldsymbol{\theta})$ is at the boundary of its valid domain and thus may be unsuitable for the task at hand. Reformulating the model may be a more promising way, compared to enforcing inequality constraints [WP1997, Ch. 3.6.1].

If these inequality constraints are inactive at the final solution, the question has to be raised as to why the additional computational load was introduced in the first place. However, as with many aspects in system identification, absolute statements are difficult: Sometimes, inequality constraints may actually help convergence, e.g. to prevent unstable models at intermediate stages of the optimization. Then also inactive inequality constraints may be of use.

This section presented the basis for including constraint information in maximum likelihood estimation. The original result by GORMAN and HERO was summed up, and a relation to what has been elaborated in the context of weighted least-squares

Example 3.2: Covariance matrix modification for equal parameters
If two parameter elements are constrained to be equal, the corresponding constraint gradient is

$$[\tilde{\boldsymbol{c}}]_{(l)} = [\mathbf{z}]_{(i)} - [\mathbf{z}]_{(j)} = 0$$
$$\frac{\partial [\tilde{\boldsymbol{c}}]_{(l)}}{\partial \mathbf{z}} = \boldsymbol{e}_i - \boldsymbol{e}_j$$

A null-space basis is then

$$\mathbf{Z} = \begin{bmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_{i-1} & \boldsymbol{e}_{i+1} & \cdots & \boldsymbol{e}_{j-1} & \boldsymbol{e}_i + \boldsymbol{e}_j & \boldsymbol{e}_{j+1} & \cdots & \boldsymbol{e}_{n_\mathbf{z}} \end{bmatrix}$$

The unit vector in one of the two directions (here $\boldsymbol{e}_i$) does not appear in $\mathbf{Z}$, whereas the other one is replaced by the sum of the unit vectors (here $\boldsymbol{e}_i + \boldsymbol{e}_j$). In the condensed Fisher information matrix $\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}} \mathbf{Z}$ the "information" w.r.t. $[\mathbf{z}]_{(i)}$ and $[\mathbf{z}]_{(j)}$ is then summed up, before inverting it and projecting it back to the original parameter space.
This eventually leads to a constrained parameter covariance matrix $\boldsymbol{\mathcal{F}}^{-1}_{constr.}$, whose i-th and j-th rows and columns are equal, i.e. where the uncertainty in the i-th and j-th parameter are equal, and $[\mathbf{z}]_{(i)}$ and $[\mathbf{z}]_{(j)}$ are perfectly (positively) correlated, which again fits intuition.

estimation has been shown. This result is at the heart of the next section, when it is shown how it may be used to compute parameter uncertainties if a full-discretization approach is used for the solution of the estimation problem.

3.5 Unified Approach for Computation of Parameter Uncertainties

It has been pointed out a few times already that to properly assess the quality of estimated parameter values, a measure for the uncertainty involved should be presented. Despite the errors introduced when using the inverse of the Fisher information matrix to this end (as discussed in section 2.2.5), it still remains one of the easiest and best accessible means of evaluating the uncertainty of the estimates. Its computation has been illustrated for some special cases (maximum likelihood model parameter estimation section 2.2.5; MAP model parameter estimation section 2.2.3.5).

Here, the major challenge lies in the fact that the standard textbook approaches for uncertainty quantification in aircraft system identification are not capable of taking constraints into account. However, when using full discretization, the dynamic system characteristics are enforced *purely* via constraints. Thus for a meaningful application to parameter estimation, a new approach was necessary. This latter is "unified" in the sense that it is on the one hand able to yield meaningful results for fully discretized

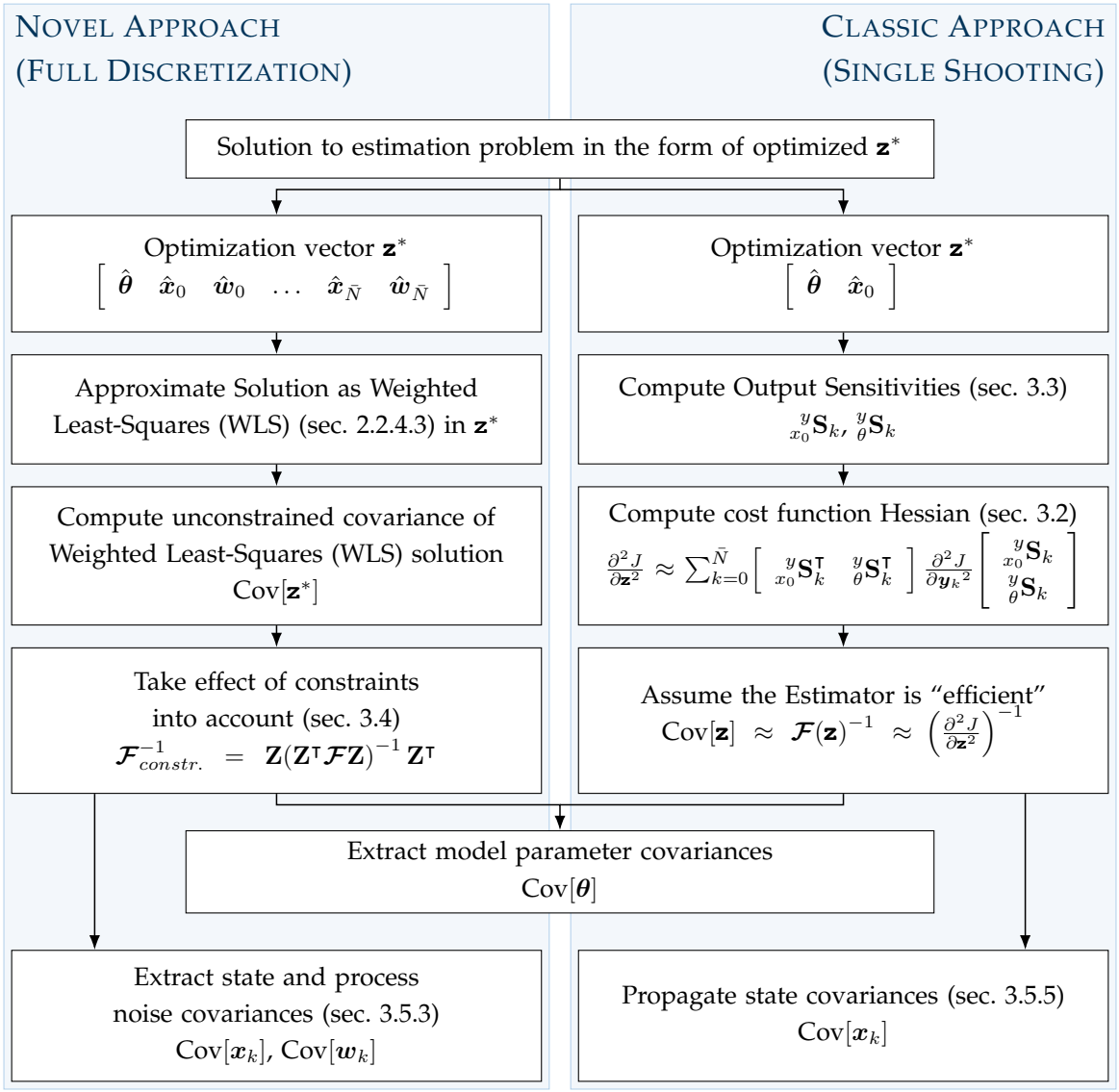


Figure 3.1: *Classic and novel approach to uncertainty quantification in dynamic system identification*

problems. On the other hand, it actually contains the standard textbook solutions as special cases, if applied to a single shooting formulation.

Figure 3.1 contrasts the novel, and the classical approach. The classic approach relies on explicitly solving the sensitivity equations, computing the cost function Hessian and using its inverse as approximation to the model parameter covariances. Most of the computations are part of the optimization iteration anyway, so the additional effort to obtain model parameter covariances is small.

The novel approach, developed during the work on this thesis, follows a different argument: first, the estimation problem is locally approximated as a WLS problem at the final estimation result $\mathbf{z}^*$. Then the methods of section 2.2.4.3 on least squares estimation may be used to obtain the unconstrained Fisher information matrix of the complete optimization vector. Lastly, the approach of the last section will be used to

properly take any active constraint into account. Since the novel approach is able to explicitly consider constraints both on states, process noise, and model parameters, it is very well applicable to fully discretized problems. After the theoretical development, algorithm 3.1 will summarize the key steps.

The main steps to be illustrated in the following subsections are

1. approximate the state and parameter estimation task as weighted least squares problem and determine its parameter covariance matrix (section 3.5.1)
2. formulate the linearized regressor matrix for different problem statements (section 3.5.2)
3. use the result of the last section to obtain the constrained parameter covariance matrix using a null-space basis of the constraint Jacobian (section 3.5.3)
4. present ways to analytically determine this null-space basis (section 3.5.4)
5. show that the classical methods for uncertainty quantification are contained in the novel approach, if the problem is formulated in a special way (section 3.5.5)

3.5.1 Unconstrained Covariance of the Optimal Solution

The following section presents the solution for one experiment only. Extensions to the simultaneous consideration of multiple experiments are straight-forward, but considerably complicate the notation, which would be detrimental to understanding the basic ideas. Also, the last sample $N-1$ will again be indicated by $\bar{N}$ to keep the notation shorter.

Even though the most general cost function formulation in table 2.1 involves an augmented state vector including the time-varying model parameters, here we will consider the case of random, but time-invariant parameters as the most general formulation. The reason is, that this is the most complex problem to treat notation-wise, since time-varying parameters may easily be included in an augmented state vector.

In this most general case, the optimization vector consists of model parameters, states at all points in time, as well as process noise inputs[3]

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\theta}^\intercal & \boldsymbol{x}_0^\intercal & \boldsymbol{w}_0^\intercal & \cdots & \boldsymbol{x}_{\bar{N}}^\intercal & \boldsymbol{w}_{\bar{N}}^\intercal \end{bmatrix}^\intercal \tag{3.53}$$

Then, one can realize that all of the cost functions in table 2.1 are a combination of several weighted least-squares cost functions, apart form the logarithmic term on the output side of the problem. However, this term is only relevant if the residual covariance is to be estimated and does not influence gradient computations (see section 3.2

[3] So far, only explicit system formulations $\boldsymbol{x}_{k+1} = {}^d\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k, \boldsymbol{\theta}]$ $k = 0 \ldots, \bar{N}-1$ have been considered when discussing process noise. Then, a process noise term $\boldsymbol{w}_{\bar{N}}$ at the end of the period of interest cannot influence the state trajectory. Now, implicit schemes are easily incorporated in the formulation of integration defects $\boldsymbol{\zeta}_k(\boldsymbol{x}_k, \hat{\boldsymbol{x}}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{w}_k, \boldsymbol{w}_{k+1}, \boldsymbol{\theta})$. This may result in a noticeable effect of $\boldsymbol{w}_{\bar{N}}$, which is why it is included in the optimization vector.

for details), which is why it is disregarded for the *first-order* expansion used here. Thus for this section, the cost function is considered to consist of weighted least-squares terms only

$$
\begin{aligned}
J(\mathbf{z}) &= \frac{1}{2}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta}(\mathbf{z}^*))^\mathsf{T}\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}_{prior} - \boldsymbol{\theta}(\mathbf{z}^*)) + \frac{1}{2}(\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0(\mathbf{z}^*))^\mathsf{T}(\mathbf{P}_0^{\mathrm{x}})^{-1}(\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0(\mathbf{z}^*)) \\
&\quad + \frac{1}{2}\sum_{k=0}^{\bar{N}}(\boldsymbol{z}_k - \boldsymbol{g}_k(\mathbf{z}^*))^\mathsf{T}\,\tilde{\mathbf{R}}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{g}_k(\mathbf{z}^*)) + \frac{1}{2}\sum_{k=0}^{\bar{N}}\boldsymbol{w}_k(\mathbf{z}^*)^\mathsf{T}\,\mathbf{Q}_k^{-1}\boldsymbol{w}_k(\mathbf{z}^*) \qquad (3.54) \\
&= \frac{1}{2}\boldsymbol{\beta}^\mathsf{T}\begin{bmatrix}
\boldsymbol{\Sigma}^{-1} & & & & & & & \\
& (\mathbf{P}_0^{\mathrm{x}})^{-1} & & & & & & \\
& & \tilde{\mathbf{R}}_0^{-1} & & & & & \\
& & & \ddots & & & & \\
& & & & \tilde{\mathbf{R}}_{\bar{N}}^{-1} & & & \\
& & & & & \mathbf{Q}_0^{-1} & & \\
& & & & & & \ddots & \\
& & & & & & & \mathbf{Q}_{\bar{N}}^{-1}
\end{bmatrix}\underbrace{\begin{bmatrix}
\boldsymbol{\theta}_{prior} - \boldsymbol{\theta}(\mathbf{z}^*) \\
\bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0(\mathbf{z}^*) \\
\boldsymbol{z}_0 - \boldsymbol{g}_0(\mathbf{z}^*) \\
\vdots \\
\boldsymbol{z}_{\bar{N}} - \boldsymbol{g}_{\bar{N}}(\mathbf{z}^*) \\
\mathbf{0} - \boldsymbol{w}_0(\mathbf{z}^*) \\
\vdots \\
\mathbf{0} - \boldsymbol{w}_{\bar{N}}(\mathbf{z}^*)
\end{bmatrix}}_{=\boldsymbol{\beta}} \\
&= (\boldsymbol{z}^a - \boldsymbol{y}^a(\mathbf{z}^*))^\mathsf{T}\,\mathbf{W}(\boldsymbol{z}^a - \boldsymbol{y}^a(\mathbf{z}^*)) \\
\boldsymbol{z}^a &= \begin{bmatrix} \boldsymbol{\theta}_{prior}{}^\mathsf{T} & \bar{\boldsymbol{x}}_0{}^\mathsf{T} & \boldsymbol{z}_0^\mathsf{T} & \cdots & \boldsymbol{z}_{\bar{N}}^\mathsf{T} & \mathbf{0}^\mathsf{T} & \cdots & \mathbf{0}^\mathsf{T} \end{bmatrix}^\mathsf{T} \qquad (3.55) \\
\boldsymbol{y}^a(\mathbf{z}^*) &= \begin{bmatrix} \boldsymbol{\theta}(\mathbf{z}^*)^\mathsf{T} & \boldsymbol{x}_0(\mathbf{z}^*)^\mathsf{T} & \boldsymbol{g}_0(\mathbf{z}^*)^\mathsf{T} & \cdots & \boldsymbol{g}_{\bar{N}}(\mathbf{z}^*)^\mathsf{T} & \boldsymbol{w}_0(\mathbf{z}^*)^\mathsf{T} & \cdots & \boldsymbol{w}_{\bar{N}}(\mathbf{z}^*)^\mathsf{T} \end{bmatrix}^\mathsf{T} \qquad (3.56)
\end{aligned}
$$

In above equation the expressions $\boldsymbol{\theta}(\mathbf{z}^*)$, $\boldsymbol{x}_0(\mathbf{z}^*)$ and $\boldsymbol{w}_k(\mathbf{z}^*)$ are to be interpreted as "extract the relevant elements from the complete optimization vector $\mathbf{z}^*$". Thus it is shown how the problem actually is a weighted least squares formulation for an augmented measurement vector $\boldsymbol{z}^a$ and the augmented, non-linear model outputs $\boldsymbol{y}^a(\mathbf{z}^*)$.

In order to eventually apply the methods of section 2.2.4.3 to obtain the covariance of $\mathbf{z}^*$, the augmented, non-linear output equations are linearized around the optimum estimate $\mathbf{z}$. This is directly done in the context of an augmented residual vector $\boldsymbol{v}^a$

$$
\boldsymbol{v}^a = \boldsymbol{z}^a - \boldsymbol{y}^a(\mathbf{z}) \approx \boldsymbol{z}^a - \boldsymbol{y}^a(\mathbf{z}^*) - \left.\frac{\partial \boldsymbol{y}^a}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}(\mathbf{z} - \mathbf{z}^*) = \tilde{\boldsymbol{z}}^a - \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\delta\mathbf{z} \qquad (3.57)
$$

where the augmented measurements $\boldsymbol{z}^a$ and the model outputs at the linearization point $\boldsymbol{y}^a(\mathbf{z}^*)$ are collected in $\tilde{\boldsymbol{z}}^a = \boldsymbol{z}^a - \boldsymbol{y}^a(\mathbf{z}^*)$; and the partial derivatives of the augmented output $\left.\frac{\partial \boldsymbol{y}^a}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ are contained in the regressor matrix $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$. A similar idea is often used to motivate the Gauss-Newton optimization algorithm via this "quasi linearization" [Jat2015, Ch. 4].

To have a link to the original cost function formulation, it is instructive to consider

the elements (i.e. rows) of $\boldsymbol{v}^a$

$$\boldsymbol{v}^a \approx \tilde{\boldsymbol{z}}^a - \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] \delta\mathbf{z} = \begin{bmatrix} \boldsymbol{\theta}_{prior} - \boldsymbol{\theta}(\mathbf{z}^*) \\ \bar{\boldsymbol{x}}_0 - \boldsymbol{x}_0(\mathbf{z}^*) \\ \boldsymbol{z}_0 - \boldsymbol{g}_0(\mathbf{z}^*) \\ \vdots \\ \boldsymbol{z}_{\bar{N}} - \boldsymbol{g}_{\bar{N}}(\mathbf{z}^*) \\ -\boldsymbol{w}_0(\mathbf{z}^*) \\ \vdots \\ -\boldsymbol{w}_{\bar{N}}(\mathbf{z}^*) \end{bmatrix} - \begin{bmatrix} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{x}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{w}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{w}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \end{bmatrix} \delta\mathbf{z} \tag{3.58}$$

which are directly related to prior information on model parameters $\boldsymbol{\theta}_{prior}$ and initial values $\bar{\boldsymbol{x}}_0$, as well as the original measurements $\boldsymbol{z}_k$ and the process noise terms $\boldsymbol{w}_k$.

In summary, above cost function, and linearized measurement equation may now be interpreted as a weighted least squares problem (see section 2.2.4.3) with

- "measurements" $\tilde{\boldsymbol{z}}^a$
- "regressor matrix" $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$
- "residuals" $\boldsymbol{v}^a$ with covariance matrix

$$\mathrm{Cov}[\boldsymbol{v}^a] = \begin{bmatrix} \boldsymbol{\Sigma} & & & & & & & \\ & \mathbf{P}_0^{\mathrm{x}} & & & & & & \\ & & \tilde{\mathbf{R}}_0^{\intercal} & & & & & \\ & & & \ddots & & & & \\ & & & & \tilde{\mathbf{R}}_{\bar{N}}^{\intercal} & & & \\ & & & & & \mathbf{Q}_0 & & \\ & & & & & & \ddots & \\ & & & & & & & \mathbf{Q}_{\bar{N}} \end{bmatrix} \tag{3.59}$$

- and cost function

$$J \approx \left(\tilde{\boldsymbol{z}}^a - \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] \delta\mathbf{z}\right)^{\intercal} \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \left(\tilde{\boldsymbol{z}}^a - \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] \delta\mathbf{z}\right) \tag{3.60}$$

The solution to above linearized WLS problem is (see section 2.2.4.3)

$$\delta\mathbf{z}^* = \left(\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal} \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\right)^{-1} \underbrace{\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal} \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \tilde{\boldsymbol{z}}^a}_{=\mathbf{0}} = \mathbf{0} \tag{3.61}$$

The second part here is zeros, because it actually *is* the gradient of the original cost function, which was used to find $\mathbf{z}^*$ in the first place. This was to be expected: a problem, which stems from an approximation around the optimum, should have its own optimal solution at zero.

More interesting is the approximation to the covariance of $\mathbf{z}$, which may also be obtained from the weighted least-squares solution, see section 2.2.4.3

$$\mathrm{Cov}[\delta\mathbf{z}] = \left(\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal} \cdot \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \cdot \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\right)^{-1} = \boldsymbol{\mathcal{F}}(\mathbf{z})^{-1} \tag{3.62}$$

Since the weighted least-squares estimator is unbiased and efficient, above also constitutes the inverse of the Fisher information matrix, which is hinted at in the last equality. The above may be used to eventually quantify the uncertainty in $\mathbf{z}$, since for these considerations, the solution to the optimization problem $\mathbf{z}^*$ may be considered deterministic

$$\mathrm{Cov}[\mathbf{z}] = \mathrm{Cov}[\mathbf{z}^* + \delta\mathbf{z}] = \mathrm{Cov}[\delta\mathbf{z}] \tag{3.63}$$

In summary, the last section presents a way to obtain an estimate for the unconstrained covariance matrix of the optimization parameters at the optimum. This was achieved by a linear approximation of the problem, which resulted in a WLS formulation, for which an uncertainty estimate could be obtained. This serves as a basis, which will be expanded by considering constraints in section 3.5.3.

3.5.2 Elements of the Linearized Regressor Matrix

Depending on the elements contained in $\mathbf{z}$, and the terms considered in the cost function, the regressor matrix $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ may take on special forms. The most general formulation is again that, which includes model parameters, states and process noise terms in the optimization vector,

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\theta}^{\intercal} & \boldsymbol{x}_0^{\intercal} & \boldsymbol{w}_0^{\intercal} & \cdots & \boldsymbol{x}_{\bar{N}}^{\intercal} & \boldsymbol{w}_{\bar{N}}^{\intercal} \end{bmatrix}^{\intercal} \tag{3.64}$$

and uses prior information on initial conditions, and model parameters in the cost function. Then $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ has the following form

$$\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] = \begin{bmatrix} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{x}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_1}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{w}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{w}_1}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{w}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} & & & & & \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{0} & & & & \\ \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{w}_0}\Big|_{\mathbf{z}^*} & \mathbf{0} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & & \mathbf{0} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{w}_1}\Big|_{\mathbf{z}^*} & \ddots & \\ \vdots & & & \ddots & \ddots & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & & & & \mathbf{0} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{w}_{\bar{N}}}\Big|_{\mathbf{z}^*} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{w}}} & & & & \\ \mathbf{0} & & \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{w}}} & & \\ \vdots & & & \ddots & \ddots & \ddots & \\ \mathbf{0} & & & & \mathbf{0} & \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{w}}} \end{bmatrix} \tag{3.65}$$

If no process noise is to be considered, the $\boldsymbol{w}$ terms are removed from the optimization vector and regressor matrix

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\theta}^\intercal & \boldsymbol{x}_0^\intercal & \cdots & \boldsymbol{x}_{\bar{N}}^\intercal \end{bmatrix}^\intercal \tag{3.66}$$

$$\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] = \begin{bmatrix} \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{x}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_1}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0}\Big|_{\mathbf{z}^*} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & \mathbf{0} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1}\Big|_{\mathbf{z}^*} & \ddots & \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & & & \mathbf{0} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}}\Big|_{\mathbf{z}^*} \end{bmatrix} \tag{3.67}$$

Further simplifications may be obtained, if prior information on model parameters and initial conditions are not to be considered

$$\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] = \begin{bmatrix} \frac{\partial \boldsymbol{g}_0}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \frac{\partial \boldsymbol{g}_1}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \\ \vdots \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0}\Big|_{\mathbf{z}^*} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & \mathbf{0} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1}\Big|_{\mathbf{z}^*} & \ddots & \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}}\Big|_{\mathbf{z}^*} & & & \mathbf{0} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}}\Big|_{\mathbf{z}^*} \end{bmatrix} \tag{3.68}$$

Residual Coloring

The same considerations as already presented in section 2.2.5.2 on residual coloring may be incorporated in the present framework, too. If the covariance of the augmented noise vector $\mathrm{Cov}[\boldsymbol{v}^a]$ is wrongly specified, it may be corrected by using sample auto covariance estimates $\mathbf{R}^{corr}$. The adjusted parameter covariance is then

$$\begin{aligned} \mathrm{Cov}[\mathbf{z}] = \Big(\Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big]\Big)^{-1} \\ \Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \cdot \mathbf{R}^{corr} \cdot \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big] \\ \Big(\Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \Big[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\Big]\Big)^{-1} \end{aligned} \tag{3.69}$$

The downside of this formulation is the considerable size of the matrices involved. Currently, straight forward application of the residual coloring approach is only possible, for the standard single shooting formulation. This yields essentially the same results as proposed in [MK1997]. Otherwise, the inversion of very large, dense matrices becomes necessary, thus making it impossible to obtain results within reasonable computational times.

Further work may find advantageous formulations of above equation or good approximations thereof, making computations possible for practical problems.

3.5.3 Constrained Covariance of the Optimal Solution

In order to fully characterize the uncertainty of a solution, possible equality constraints need to be taken into account. To do so, a linear approximation is used. Consider the

concatenation of all equality and *active* inequality constraints, as well as integration defects (see section 2.1.4.3 for details)

$$\tilde{\boldsymbol{c}}(\mathbf{z}^*) = \begin{bmatrix} \boldsymbol{c}_{eq}(\boldsymbol{x}_k^*, \boldsymbol{w}_k^*, \boldsymbol{\theta}^*) & k = 0 \dots \bar{N} \\ \boldsymbol{c}_{ineq}(\boldsymbol{x}_i^*, \boldsymbol{w}_i^*, \boldsymbol{\theta}^*) & i \in \mathcal{A}(\mathbf{z}^*) \\ \boldsymbol{\zeta}_k\Big(\boldsymbol{x}_k^*, \boldsymbol{x}_{k+1}^*, \boldsymbol{w}_k^*, \boldsymbol{w}_{k+1}^*, \boldsymbol{\theta}^*\Big) & k = 0 \dots \bar{N} - 1 \end{bmatrix} = \mathbf{0} \tag{3.70}$$

which is then linearized at the optimum point

$$\tilde{\boldsymbol{c}}(\mathbf{z}^*) = \tilde{\boldsymbol{c}}(\mathbf{z}^*) + \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} \delta\mathbf{z} = \mathbf{0} \tag{3.71}$$

as was shown in the preceding section, the solution to the overall problem is $\delta\mathbf{z}^* = \mathbf{0}$, which then also ensures that the constraints are kept.

Now the results of section 3.4 (on constraints in maximum likelihood estimation), respectively section 2.2.4.4 (on constrained linear least squares estimation) may be used to incorporate the effects of $\tilde{\boldsymbol{c}}$. Let $\mathbf{Z}$ be a basis for the null-space of $\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$

$$\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} \cdot \mathbf{Z} = \mathbf{0}; \qquad \mathbf{Z} \neq \mathbf{0} \tag{3.72}$$

then the constraint covariance is according to equation (3.52) of the last section

$$\mathrm{Cov}[\mathbf{z}]_{constr} = \mathbf{Z}\Big(\mathbf{Z}^\intercal \mathrm{Cov}[\mathbf{z}]^{-1}\, \mathbf{Z}\Big)^{-1} \mathbf{Z}^\intercal \overset{(3.62)}{=} \mathbf{Z}(\mathbf{Z}^\intercal \boldsymbol{\mathcal{F}}(\mathbf{z})\, \mathbf{Z})^{-1}\, \mathbf{Z}^\intercal \tag{3.73}$$

Again, one can see that first the "information" in the problem is condensed onto the remaining degrees of freedom, i.e. the null-space of the constraint Jacobian. Then the corresponding covariance in the free directions is computed and projected onto the original parameter space.

The result is the covariance of the complete optimization vector $\mathbf{z}$. Thus, even if the system under consideration was deterministic, the resulting, propagated states are random variables! The reason for this is that the parameter estimates are computed from measurements, which themselves are random variables. This is especially true for the model parameters $\boldsymbol{\theta}$, but also extends to the states $\boldsymbol{x}$: either because they are directly part of the optimization vector (full-discretization case), or because they are propagated, based on "random" model parameter estimates $\hat{\boldsymbol{\theta}}$ (single-shooting case).

The structure of $\mathrm{Cov}[\mathbf{z}]_{constr}$ may vary, again depending on the elements that are directly included in $\mathbf{z}$. For the most general case, with model parameter, state, and process noise components, it has the following structure

$$\mathrm{Cov}[\mathbf{z}]_{constr} = \mathrm{Cov}\left[\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{x}_0 \\ \boldsymbol{w}_0 \\ \vdots \\ \boldsymbol{x}_{\bar{N}} \\ \boldsymbol{w}_{\bar{N}} \end{bmatrix}\right] = \begin{bmatrix} \mathbf{P}^{\boldsymbol{\theta}} & \mathbf{P}^{\boldsymbol{\theta}\mathbf{x}_0} & \mathbf{P}^{\boldsymbol{\theta}\mathbf{w}_0} & \dots & \mathbf{P}^{\boldsymbol{\theta}\mathbf{x}_{\bar{N}}} & \mathbf{P}^{\boldsymbol{\theta}\mathbf{w}_{\bar{N}}} \\ \mathbf{P}^{\mathbf{x}_0\boldsymbol{\theta}} & \mathbf{P}_0^{\mathbf{x}} & & & \mathbf{P}^{\mathbf{x}_0\mathbf{x}_{\bar{N}}} & \mathbf{P}^{\mathbf{x}_0\mathbf{w}_{\bar{N}}} \\ \mathbf{P}^{\mathbf{w}_0\boldsymbol{\theta}} & & \mathbf{P}_0^{\mathbf{w}} & & & \\ \vdots & & & \ddots & & \vdots \\ \mathbf{P}^{\mathbf{x}_{\bar{N}}\boldsymbol{\theta}} & \mathbf{P}^{\mathbf{x}_{\bar{N}}\mathbf{x}_0} & & & \mathbf{P}_{\bar{N}}^{\mathbf{x}} & \\ \mathbf{P}^{\mathbf{w}_{\bar{N}}\boldsymbol{\theta}} & \mathbf{P}^{\mathbf{w}_{\bar{N}}\mathbf{x}_0} & \dots & & & \mathbf{P}_{\bar{N}}^{\mathbf{w}} \end{bmatrix} \tag{3.74}$$

From this, the other cases are derived by omitting the respective rows and columns. The implementation developed during the work on this thesis only extracts the main diagonal blocks of above matrix for further analysis. They contain the model parameter covariance, as well as state and possibly process noise covariances, after the estimation has finished. Thus, these covariances represent the "posterior" information, i.e. they combine possible prior uncertainty ($\boldsymbol{\Sigma}$, $\mathbf{P}_0^{\mathrm{x}}$, $\mathbf{R}$, $\mathbf{Q}$) with measurement and system information to arrive at the above uncertainty estimate.

Necessarily, model parameters, states and process noise at different points in time (i.e. the off-diagonal blocks) will show significant correlation: only their combined effect leads to the resulting trajectory, thus they will be linked via the system dynamics. It is a possible point for future research, if this information can be meaningfully used to assess the quality of an estimate.

Above covariance matrix $\mathrm{Cov}[\mathbf{z}]_{constr}$ may become very large, and is always densely populated. However, as was mentioned before, by only storing the condensed information matrix $\mathbf{Z}^{\intercal}\mathrm{Cov}[\mathbf{z}]^{-1}\,\mathbf{Z}$ together with the null-space basis $\mathbf{Z}$ all relevant submatrices may be computed when desired, using only a fraction of the memory necessary to store the complete $\mathrm{Cov}[\mathbf{z}]_{constr}$.

The last section thus completes the theoretic considerations by properly incorporating constraints into the aircraft system identification process. The approach illustrated so far is now capable of computing parameter covariances for all cases that are considered in this work, which is one of the major contributions of the author. It is well adjusted to using full discretization for parameter estimation by ensuring that the same mechanisms are employed for the solution of the system equations (via equality constraints) and the determination of uncertainty estimates (via the null-space of the constraint Jacobian). One remaining question, namely how to efficiently compute the null space basis of the constraint Jacobian, will be addressed next.

3.5.4 Determination of the Null-Space Basis of the Constraint Jacobian

To actually compute the null-space basis of the constraint Jacobian, two solutions are possible. Either, one can use a brute force, numeric approach and use a computer program to determine $\mathbf{Z}$, or one can exploit the special structure of the problem and determine at least some parts of $\mathbf{Z}$ analytically, before resorting to numerical methods.

The implementation that was developed during the work on this thesis checks that a valid null space basis was found by inspecting the largest magnitude entry of the product $\left(\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*} \cdot \mathbf{Z}\right)$. In all examples that are considered here, this value was in the region of machine precision ϵ. The software issues a warning if it is larger than $\epsilon \times 10^4$, which may hint at some problems with the model or data at hand.

3.5.4.1 Numeric Determination

When actually determining the null-space basis, one can make use of the fact that typically, in parameter estimation problems, the number of remaining degrees of freedom is considerably smaller than the size of the optimization vector $\mathbf{z}$; at least for deterministic systems without process noise inputs $\boldsymbol{w}_k$, which fortunately constitute the majority of cases. Additionally, the constraint Jacobian in fully discretized problems is commonly sparse [Bit2017], which can be utilized in the search for the null-space basis.

The Singular Value Decomposition (SVD) explicitly contains an ortho-normal null-space basis as part of the right singular matrix (see appendix A.2.1 or [BIG2003] for background on the SVD.)

$$\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\intercal \\ \mathbf{Z}^\intercal \end{bmatrix} \tag{3.75}$$

Unfortunately, for practical cases it cannot be computed directly in MATLAB, since the `svd` function cannot handle sparse matrices (at least in the version R2016b which was used for the implementation at hand), which is why a detour via the eigenvalue decomposition was implemented. The idea is to compute the $n_{\mathbf{z}} - n_{\tilde{\boldsymbol{c}}}$ smallest magnitude eigenvectors of $\left(\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}^\intercal \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}\right)$ via MATLAB's `eigs` function.

$$\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}^\intercal \left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*} = \begin{bmatrix} * & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} * \\ \mathbf{Z}^\intercal \end{bmatrix} \tag{3.76}$$

The null space of $\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ can have at maximum dimension $n_{\mathbf{z}} - n_{\tilde{\boldsymbol{c}}}$ (if $\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ has full rank, i.e. if the Linear Independence Constraint Qualification (LICQ) holds) and `eigs` is capable of exploiting the sparsity of $\left.\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\right|_{\mathbf{z}^*}$ in order to compute a pre-defined number of eigenvectors. Those associated with eigenvalues at 0 then constitute the null-space basis.

Above approach is especially appealing, if there are few degrees of freedom, and many constraints: the null-space can be computed numerically in a straight forward manner, and no information about the exact form of the constraints is necessary. Thus relatively arbitrary problems may be solved, with e.g. path constraints that are only active at certain sampling instants, which otherwise have to be considered explicitly.

3.5.4.2 Analytic Determination

The analytic solution to the problem of finding a null-space basis of the constraint Jacobian is based on the fact that in parameter estimation problems most of the active constraints will be related to integration defects. Thus the focus of this section will be on finding an analytic null-space basis ${}_{\mathrm{S}}\mathbf{Z}$ for the integration defects $\boldsymbol{\zeta}$, before incorporating the remaining constraints separately.

Deterministic Problem Formulation

First, consider a deterministic problem formulation, i.e. the integration defect at one point in time only depends on adjacent state, and (fixed) input values, as well as model parameters. Then, the derivatives of the integration defects at the optimum $\mathbf{z}^*$ are[4]

$$\begin{aligned}\mathbf{0} &= \frac{\partial}{\partial \tilde{\mathbf{z}}} \boldsymbol{\zeta}_k \left(\boldsymbol{x}_k^*, \boldsymbol{x}_{k+1}^*, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}^* \right) \\ &= \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_k} \frac{\partial \boldsymbol{x}_k}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_{k+1}} \frac{\partial \boldsymbol{x}_{k+1}}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \tilde{\mathbf{z}}} \\ &= \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_k} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_k + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_{k+1}} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{k+1} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{\theta}} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S}\end{aligned} \tag{3.77}$$

For ease of notation, the explicit evaluation point at $\mathbf{z}^*$ of the Jacobians, i.e. $\square|_{\mathbf{z}^*}$ is dropped. Above derivatives are taken w.r.t. an optimization vector $\tilde{\mathbf{z}}$, which does not necessarily have to correspond to the $\mathbf{z}$ of the original problem. Suitably choosing $\tilde{\mathbf{z}}$ will provide a proper basis for the null-space, which will be shown further down.

Above equation relates the state and parameter sensitivities via the (possibly implicit) numerical integration scheme that was chosen to discretize the problem. Now this can be stacked vertically to arrive at

$$\begin{bmatrix} \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_0} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_1} & & & & \\ \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{\theta}} & & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_2} & & & \\ \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{\theta}} & & & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_2} & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_3} & & \\ \vdots & & & & \ddots & \ddots & \\ \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{\theta}} & & & & & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix} \begin{bmatrix} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S} \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_0 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_3 \\ \vdots \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix} = \mathbf{0} \tag{3.78}$$

However, the matrix on the left is exactly the constraint Jacobian $\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}$ of the integration defects. Thus the solution to the sensitivity equations actually contains a basis for the null-space of the constraint Jacobian!

This basis can be extracted, if the elements of the optimization vector $\tilde{\mathbf{z}}$ are chosen to match the number of degrees of freedom of the constraint Jacobian: Since the LICQ hold for any reasonable parameter estimation problem formulation, this number of degrees of freedom is equal to the number of columns minus the number of rows of $\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}$. The integration defects have $(N-1)\, n_{\boldsymbol{x}}$ rows and the optimization vector has $N n_{\boldsymbol{x}} + n_{\boldsymbol{\theta}}$ elements, leaving $n_{\boldsymbol{x}} + n_{\boldsymbol{\theta}}$ degrees of freedom.

One choice for $\tilde{\mathbf{z}}$ that comes to mind is related to the single shooting approach, i.e. only considering the model parameters $\boldsymbol{\theta}$ and initial conditions $\boldsymbol{x}_0$ as part of the optimization vector $\tilde{\mathbf{z}}$. Thus their sensitivities ${}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S}$ and ${}^{x_0}_{\tilde{\mathbf{z}}}\mathbf{S}$ only contain identity and

[4] This constitutes an alternative to solving the sensitivity equations of section 3.3: here the integration is performed first (via the defect equations), before taking derivatives; whereas in section 3.3 derivatives are taken first, before solving the resulting matrix-valued ODEs using a numerical integration scheme.

zero matrices, and above system can be transformed into a linear system of equations with full rank. This is done by bringing the first two columns (related to $\boldsymbol{\theta}$ and $\boldsymbol{x}_0$) to the right

$$\tilde{\mathbf{z}} = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{x}_0 \end{bmatrix} \tag{3.79}$$

$$\begin{bmatrix} \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_1} & & & & \\ \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_2} & & & \\ & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_2} & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_3} & & \\ & & \ddots & \ddots & \\ & & & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix} \begin{bmatrix} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_3 \\ \vdots \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{\theta}} & -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_0} \\ -\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{\theta}} & \mathbf{0} \\ -\frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{\theta}} & \vdots \\ \vdots & \\ -\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{\theta}} & \mathbf{0} \end{bmatrix} \underbrace{\begin{bmatrix} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S} \\ {}^{x_0}_{\tilde{\mathbf{z}}}\mathbf{S} \end{bmatrix}}_{=\mathbf{I}_{n_{\boldsymbol{x}}+n_{\boldsymbol{\theta}}}} \tag{3.80}$$

Thus one choice for the null-space basis ${}_{\mathrm{S}}\mathbf{Z}$, and consequently the solution to equation (3.78), can be constructed by augmenting the solution to above system of equations with ${}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S}$ and ${}^{x_0}_{\tilde{\mathbf{z}}}\mathbf{S}$ as follows

$$\underbrace{\begin{bmatrix} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S} \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_0 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_3 \\ \vdots \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix}}_{={}_{\mathrm{S}}\mathbf{Z}} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & & & & & & & \\ & \mathbf{I}_{n_{\boldsymbol{x}}} & & & & & & \\ & & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_1} & & & & & \\ & & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_2} & & & & \\ & & & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_2} & \frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{x}_3} & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{\theta}} & -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_0} \\ -\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{\theta}} & \mathbf{0} \\ -\frac{\partial \boldsymbol{\zeta}_2}{\partial \boldsymbol{\theta}} & \vdots \\ \vdots & \\ -\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{\theta}} & \mathbf{0} \end{bmatrix} \tag{3.81}$$

which from here on out will be referred to as ${}_{\mathrm{S}}\mathbf{Z}$, i.e. that part of the null-space related to the integration defects, which may be constructed from the sensitivities. Above (non-ortho-normal) basis for the null-space of the constraint Jacobian may be computed quite efficiently, possibly more efficiently than using the numeric eigenvector approach, especially if the number of degrees of freedom becomes large.

The basic approach of section 3.4 to incorporate constraints in the uncertainty quantification is based on an *arbitrary* null-space basis. Different choices of $\mathbf{Z}$ then correspond to different subsets of optimization parameters (or linear combination thereof) onto which the information is first condensed via $\mathbf{Z}^{\intercal}\boldsymbol{\mathcal{F}}(\mathbf{z})\,\mathbf{Z}$. The numeric approach chooses this subset of optimization parameters such that an ortho-normal basis $\mathbf{Z}$ results. The analytic approach presented here, explicitly chooses $\boldsymbol{\theta}$ and $\boldsymbol{x}_0$ for a different version of $\mathbf{Z}$.

Consideration of Process Noise

If process noise is to be included explicitly in the problem formulation, a similar approach as described above for the deterministic case may be applied. However, the number of degrees of freedom, which directly relates to the dimension of the null-space basis in this case, is considerably larger. Every process noise entry in the defect

constraints will be an additional degree of freedom, increasing the null-space dimension to $n_{\boldsymbol{\theta}} + n_{\boldsymbol{x}} + N n_{\boldsymbol{w}}$.

Relying on the numeric approach via the SVD of $\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}$ may then fail due to the sheer size of the problem. Fortunately, the approach via the sensitivity equations may be extended to provide an analytic way to compute the null-space basis in the presence of explicitly considered process noise terms.

As before, the sensitivities are computed via the integration defects, now taking process noise elements into account

$$\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial \tilde{\mathbf{z}}} \boldsymbol{\zeta}_k \left(\boldsymbol{x}_k^*, \boldsymbol{x}_{k+1}^*, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{w}_k^*, \boldsymbol{w}_{k+1}^*, \boldsymbol{\theta}^* \right) \\
&= \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_k} \frac{\partial \boldsymbol{x}_k}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_{k+1}} \frac{\partial \boldsymbol{x}_{k+1}}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_k} \frac{\partial \boldsymbol{w}_k}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_{k+1}} \frac{\partial \boldsymbol{w}_{k+1}}{\partial \tilde{\mathbf{z}}} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \tilde{\mathbf{z}}} \\
&= \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_k} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_k + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_{k+1}} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{k+1} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_k} {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_k + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_{k+1}} {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_{k+1} + \frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{\theta}} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S}
\end{aligned} \tag{3.82}$$

For ease of notation, the explicit evaluation point at $\mathbf{z}^*$ of the Jacobians, i.e. $\square|_{\mathbf{z}^*}$ is again dropped.

The concatenated sensitivity equations then become

$$\begin{bmatrix}
\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_0} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{w}_0} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{w}_1} & & & & & & \\
\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{\theta}} & & & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{w}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_2} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{w}_2} & & & & \\
\vdots & & & & & & \ddots & \ddots & & & \\
\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{\theta}} & & & & & & & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{w}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{w}_{\bar{N}}}
\end{bmatrix}
\begin{bmatrix}
{}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S} \\
{}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_0 \\
{}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_0 \\
{}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\
{}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\
{}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\
{}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\
\vdots \\
{}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \\
{}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}}
\end{bmatrix} = \mathbf{0} \tag{3.83}$$

$$\text{with } {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_k = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{x}_0 & \boldsymbol{w}_0 & & \boldsymbol{x}_k & \boldsymbol{w}_k & \boldsymbol{x}_{k+1} & & \boldsymbol{x}_{\bar{N}} & \boldsymbol{w}_{\bar{N}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{w}}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.84}$$

Since the matrix on the left is again the constraint Jacobian (including the process noise contributions), the solution to the sensitivity equations is again a basis for the null space of $\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}$. The increased number of degrees of freedom manifests itself in the solution to the sensitivity equations: in addition to $\boldsymbol{\theta}$ and $\boldsymbol{x}_0$, the $\boldsymbol{w}_k$ $k = 0 \dots \bar{N}$ are chosen as

elements of $\tilde{\mathbf{z}}$, again with known sensitivities.

$$\tilde{\mathbf{z}} = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{x}_0 & \boldsymbol{w}_0 & \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{\bar{N}} \end{bmatrix} \tag{3.85}$$

$$\begin{bmatrix} \frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_1} & & & \\ \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_1} & \frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{x}_2} & & \\ & \ddots & \ddots & \\ & & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}-1}} & \frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix} \begin{bmatrix} {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2 \\ \vdots \\ {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix} = \tag{3.86}$$

$$\left[\begin{array}{cc|cccccc} -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{\theta}} & -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{x}_0} & -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{w}_0} & -\frac{\partial \boldsymbol{\zeta}_0}{\partial \boldsymbol{w}_1} & & & & \\ -\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{\theta}} & \mathbf{0} & & -\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{w}_1} & -\frac{\partial \boldsymbol{\zeta}_1}{\partial \boldsymbol{w}_2} & & & \\ \vdots & \vdots & & & \ddots & \ddots & & \\ -\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{\theta}} & \mathbf{0} & & & & -\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{w}_{\bar{N}-1}} & -\frac{\partial \boldsymbol{\zeta}_{\bar{N}-1}}{\partial \boldsymbol{w}_{\bar{N}}} \end{array}\right] \underbrace{\begin{bmatrix} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S} \\ {}^{x_0}_{\tilde{\mathbf{z}}}\mathbf{S} \\ {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_0 \\ {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_1 \\ \vdots \\ {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix}}_{=\mathbf{I}_{n_{\boldsymbol{x}}+n_{\boldsymbol{\theta}}+N\cdot n_{\boldsymbol{w}}}}$$

where only the ${}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}$ are computed via this equation; the ${}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}$ contain zero and identity matrices only and are directly incorporated in the right-hand side. Eventually, equation (3.84) and the solution to (3.86) may be combined to arrive at the null-space basis in this case

$${}_{\mathrm{S}}\mathbf{Z} = \begin{bmatrix} {}^{\theta}_{\tilde{\mathbf{z}}}\mathbf{S}^{\mathsf{T}} & {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_0^{\mathsf{T}} & {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_0^{\mathsf{T}} & {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_1^{\mathsf{T}} & {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_1^{\mathsf{T}} & {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_2^{\mathsf{T}} & {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_2^{\mathsf{T}} & \cdots & {}^{x}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}}^{\mathsf{T}} & {}^{w}_{\tilde{\mathbf{z}}}\mathbf{S}_{\bar{N}}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} \tag{3.87}$$

The interpretation of the foregoing paragraph may be extended to the case incorporating process noise inputs, i.e. choose a subset of optimization parameters as degrees of freedom and condense information onto those parameters.

The last two paragraphs illustrated, how solutions to the problem of finding a null-space basis for the integration defect Jacobian can be constructed analytically. This was done for both cases with and without the explicit consideration of process noise. Admittedly, the additional computations are not as minor as in the standard single-shooting case, but they are not excessive either: in FALCON.m [RBG+2018], the value of the constraint Jacobian at the optimum is directly available as part of the solution. Then it is only necessary, to extract the parts related to integration defects, assemble them in the above illustrated linear systems of equations and solve for the null-space ${}_{\mathrm{S}}\mathbf{Z}$. Care has to be taken to exploit the sparsity of the problem, since matrix sizes may become too large to handle on a standard consumer PC otherwise.

The connection between the null-space basis of the constraint Jacobian and the solution to the sensitivity equations may have been investigated in the past. However, its use in the context of uncertainty quantification for aircraft system identification is new, and a major contribution of this work. Only in this way is it possible to actually compute covariances of states and parameters for real-life problem sizes, where the brute force numeric approach is bound to fail due to the sizes of the matrices involved.

Consideration of Remaining Constraints

The set of active constraints at the optimal solution of an estimation problem may not only comprise of integration defects. This makes it necessary to modify the null space basis ${}_{\mathrm{S}}\mathbf{Z}$, which was determined in the last paragraphs, in order to take those auxiliary constraints into account, too. The latter may consist of active path constraints, optimization parameter bounds, known relations between parameters etc.

Let $\boldsymbol{c}_{aux}$ be the vector of remaining, non-defect constraints with the Jacobian $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$. Then a transformation $\mathbf{E}$ of ${}_{\mathrm{S}}\mathbf{Z}$ of the last section is sought, such that ${}_{\mathrm{S}}\mathbf{Z}$ is mapped to the null space of $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$, i.e.

$$\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}} \cdot {}_{\mathrm{S}}\mathbf{Z} \cdot \mathbf{E} = \mathbf{0} \tag{3.88}$$

Since $\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}$, of which $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$ is a part, will have full rank (LICQ!), $\mathbf{E}$ necessarily reduces the column dimension of ${}_{\mathrm{S}}\mathbf{Z}$.

The arbitrary nature of $\boldsymbol{c}_{aux}$ makes it impossible to define general algorithms to analytically compute its null space basis $\mathbf{Z}_G$, and leaves its numeric determination as the only practical way. However, most of the active constraints at an optimum will be related to integration defects, thus the dimension $n_{\boldsymbol{c}_{aux}}$ of the remaining auxiliary constraints $\boldsymbol{c}_{aux}$ should be small. Further, the $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$ will have only few non-zero elements: they commonly only link optimization variables at one sampling instant (path constraints) or model parameters. Thus the all-zero columns of $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$ may also be disregarded in the search for $\mathbf{Z}_G$, further reducing the sizes of the matrices involved.

As was noted above, $\mathbf{E}$ needs to transform ${}_{\mathrm{S}}\mathbf{Z}$ such that it lies in the null-space of $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$, i.e. only consists of linear combinations of the columns of $\mathbf{Z}_G$

$$\begin{aligned} {}_{\mathrm{S}}\mathbf{Z}\mathbf{E} &= \mathbf{Z}_G\mathbf{V} \\ \Leftrightarrow \begin{bmatrix} {}_{\mathrm{S}}\mathbf{Z} & \mathbf{Z}_G \end{bmatrix} &\begin{bmatrix} \mathbf{E} \\ -\mathbf{V} \end{bmatrix} = \mathbf{0} \end{aligned} \tag{3.89}$$

where $\mathbf{V}$ is the representation of ${}_{\mathrm{S}}\mathbf{Z}\mathbf{E}$ in terms of the null space basis of $\frac{\partial \boldsymbol{c}_{aux}}{\partial \mathbf{z}}$. The desired transformation $\mathbf{E}$ is then the top part of the null-space of the compound matrix $\begin{bmatrix} {}_{\mathrm{S}}\mathbf{Z} & \mathbf{Z}_G \end{bmatrix}$. Again, this will in general have to be determined numerically, but should stay tractable, due to the reduced sizes, which have been elaborated above.

The overall approach to include auxiliary constraints can be interpreted as finding a basis for the intersection of

- the null-space of the integration defect constraint Jacobian ${}_{\mathrm{S}}\mathbf{Z}$ on the one hand,
- and the null-space of the auxiliary constraint Jacobian $\mathbf{Z}_G$ on the other hand.

For large problems, solving two sub-steps (integration defects first; then modify to include auxiliary constraints) may be considerably easier compared to the numerical solution of the overall problem as was illustrated in section 3.5.4.1.

This was the last missing part, which was necessary to determine uncertainty measures when using full discretization in parameter estimation. This novel approach is now able to consider arbitrary constraints on states, model parameters and possibly process noise, which constitutes a major advantage over the methods in use so far. Algorithm 3.1 sums up the key points to obtain an uncertainty estimate from a solved estimation problem.

3.5.5 Link to Standard Approaches

For every novel approach it is reassuring, if it can be linked meaningfully to well established methods. This is the goal of the following section, i.e. to provide a link between the textbook methods for single shooting, and the formulation via the null-space of the constraint Jacobian presented here.

The standard textbook methods are usually formulated for deterministic systems, where the model parameters $\boldsymbol{\theta}$ and possibly the initial values $\boldsymbol{x}_0$ are to be determined. Using the approach developed in this section the "condensed" covariance matrix in the free directions is $\left(\mathbf{Z}^\intercal \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] \mathbf{Z}\right)^{-1}$. Since the system is considered to be deterministic, no process noise contributions are considered and the $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ of equation (3.67) arises

$$\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}} & \mathbf{0} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1} & \ddots & \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}} & & & \mathbf{0} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix}$$

with the corresponding augmented noise covariance (assuming a constant residual covariance matrix)

$$\mathrm{Cov}[\boldsymbol{v}^a] = \begin{bmatrix} \boldsymbol{\Sigma} & & & & \\ & \mathbf{P}_0^{\mathrm{x}} & & & \\ & & \boldsymbol{\mathcal{B}} & & \\ & & & \ddots & \\ & & & & \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{3.90}$$

For this deterministic model formulation, an analytic null-space basis is provided via

Algorithm 3.1: Uncertainty quantification in parameter estimation using full discretization

I. extract the Jacobian matrices of the output equation $\boldsymbol{g}_k(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k, \boldsymbol{\theta})$ to assemble the linearized regressor matrix $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ (see section 3.5.2 for possible formulations)

II. assemble prior covariances $\boldsymbol{\Sigma}$, $\mathbf{P}_0^{\mathrm{x}}$, process noise covariances $\mathbf{Q}_k$ and measurement noise covariances $\tilde{\mathbf{R}}_k$ (pre-defined, or estimated) to obtain the augmented noise vector covariance $\mathrm{Cov}[\boldsymbol{v}^a]$ (equation (3.59))

III. compute the unconstrained covariance of the optimization vector (equation (3.62))

$$\mathrm{Cov}[\mathbf{z}]^{-1} = \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal} \cdot \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \cdot \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] = \boldsymbol{\mathcal{F}}(\mathbf{z})$$

IV. determine a null space basis for the active constraint Jacobian

- Either use the numeric approach (section 3.5.4.1) via the eigen decomposition of $\left(\frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}^{\intercal} \frac{\partial \tilde{\boldsymbol{c}}}{\partial \mathbf{z}}\Big|_{\mathbf{z}^*}\right)$
- or use the analytic construction (section 3.5.4.2)
  (a) extract the integration defect Jacobian matrices $\frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_k}$, $\frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{x}_{k+1}}$, $\frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_k}$, $\frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{w}_{k+1}}$, $\frac{\partial \boldsymbol{\zeta}_k}{\partial \boldsymbol{\theta}}$ at the optimal solution from FALCON.m
  (b) assemble the linear system of equations (eq. (3.81) for deterministic systems or eq. (3.86) for stochastic descriptions)
  (c) solve the system and assemble the null space basis of the integration defect constraints ${}_{\mathrm{S}}\mathbf{Z}$
  (d) possibly adjust for auxiliary, non-defect constraints $\boldsymbol{c}_{aux}$

V. modify the unconstrained covariance to take constraints into account (equation (3.73))

$$\begin{aligned}\mathrm{Cov}[\mathbf{z}]_{constr} &= \mathbf{Z}\left(\mathbf{Z}^{\intercal}\mathrm{Cov}[\mathbf{z}]^{-1}\,\mathbf{Z}\right)^{-1}\mathbf{Z}^{\intercal} \\ &= \mathbf{Z}\left(\mathbf{Z}^{\intercal}\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal} \cdot \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \cdot \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\mathbf{Z}\right)^{-1}\mathbf{Z}^{\intercal}\end{aligned}$$

VI. extract and store covariances for model parameters $\mathrm{Cov}[\boldsymbol{\theta}]$, states $\mathrm{Cov}[\boldsymbol{x}_k]$ and possibly process noise terms $\mathrm{Cov}[\boldsymbol{w}_k]$

equation (3.81)

$$ {}_{\mathrm{S}}\mathbf{Z} = \begin{bmatrix} {}^{\theta}_{\tilde{\boldsymbol{z}}}\mathbf{S} \\ {}^{x}_{\tilde{\boldsymbol{z}}}\mathbf{S}_0 \\ \vdots \\ {}^{x}_{\tilde{\boldsymbol{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ {}^{x}_{\theta}\mathbf{S}_1 & {}^{x}_{x_0}\mathbf{S}_1 \\ \vdots & \\ {}^{x}_{\theta}\mathbf{S}_{\bar{N}} & {}^{x}_{x_0}\mathbf{S}_{\bar{N}} \end{bmatrix} $$

Now it can be realized that large parts of the product $\left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]\mathbf{Z}$ actually consist of the discrete output sensitivities (see section 3.3)

$$ \begin{aligned} \left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]\mathbf{Z} &= \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} & & & \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0} & \mathbf{0} & & \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}} & \mathbf{0} & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1} & \ddots & \\ \vdots & & \ddots & \ddots & \mathbf{0} \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}} & & & \mathbf{0} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ {}^{x}_{\theta}\mathbf{S}_1 & {}^{x}_{x_0}\mathbf{S}_1 \\ \vdots & \\ {}^{x}_{\theta}\mathbf{S}_{\bar{N}} & {}^{x}_{x_0}\mathbf{S}_{\bar{N}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \boldsymbol{g}_0}{\partial \boldsymbol{x}_0} \\ \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1}\,{}^{x}_{\theta}\mathbf{S}_1 & \frac{\partial \boldsymbol{g}_1}{\partial \boldsymbol{x}_1}\,{}^{x}_{x_0}\mathbf{S}_1 \\ \vdots & \\ \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}}\,{}^{x}_{\theta}\mathbf{S}_{\bar{N}} & \frac{\partial \boldsymbol{g}_{\bar{N}}}{\partial \boldsymbol{x}_{\bar{N}}}\,{}^{x}_{x_0}\mathbf{S}_{\bar{N}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ {}^{y}_{\theta}\mathbf{S}_0 & {}^{y}_{x_0}\mathbf{S}_0 \\ {}^{y}_{\theta}\mathbf{S}_1 & {}^{y}_{x_0}\mathbf{S}_1 \\ \vdots & \\ {}^{y}_{\theta}\mathbf{S}_{\bar{N}} & {}^{y}_{x_0}\mathbf{S}_{\bar{N}} \end{bmatrix} \end{aligned} \tag{3.91} $$

Eventually, the condensed covariance is

$$ \begin{aligned} &\left(\mathbf{Z}^{\intercal}\left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]^{\intercal}\mathrm{Cov}[\boldsymbol{v}^a]^{-1}\left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]\mathbf{Z}\right)^{-1} = \\ &= \left(\beta^{\intercal}\cdot \begin{bmatrix} \boldsymbol{\Sigma}^{-1} & & & & \\ & \mathbf{P}_0^{\mathrm{x}-1} & & & \\ & & \boldsymbol{\mathcal{B}}^{-1} & & \\ & & & \ddots & \\ & & & & \boldsymbol{\mathcal{B}}^{-1} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_{\boldsymbol{x}}} \\ {}^{y}_{\theta}\mathbf{S}_0 & {}^{y}_{x_0}\mathbf{S}_0 \\ {}^{y}_{\theta}\mathbf{S}_1 & {}^{y}_{x_0}\mathbf{S}_1 \\ \vdots & \\ {}^{y}_{\theta}\mathbf{S}_{\bar{N}} & {}^{y}_{x_0}\mathbf{S}_{\bar{N}} \end{bmatrix}}_{=\beta}\right)^{-1} \\ &= \left(\begin{bmatrix} \boldsymbol{\Sigma}^{-1} & \\ & \mathbf{P}_0^{\mathrm{x}-1} \end{bmatrix} + \sum_{k=0}^{\bar{N}} \begin{bmatrix} {}^{y}_{\theta}\mathbf{S}_k^{\intercal} \\ {}^{y}_{x_0}\mathbf{S}_k^{\intercal} \end{bmatrix} \boldsymbol{\mathcal{B}}^{-1} \begin{bmatrix} {}^{y}_{\theta}\mathbf{S}_k & {}^{y}_{x_0}\mathbf{S}_k \end{bmatrix}\right)^{-1} \end{aligned} \tag{3.92} $$

which is exactly the same expression as in section 2.2.3.5 on uncertainty quantification for Bayesian MAP estimation using a Gaussian prior. Thus it holds

$$ \left(\mathbf{Z}^{\intercal}\left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]^{\intercal}\mathrm{Cov}[\boldsymbol{v}^a]^{-1}\left[\mathbf{X}_{\frac{\partial}{\partial \tilde{\boldsymbol{z}}}}\right]\mathbf{Z}\right)^{-1} = \begin{bmatrix} \mathrm{Cov}[\boldsymbol{\theta}] & \mathrm{Cov}[\boldsymbol{\theta}, \boldsymbol{x}_0] \\ \mathrm{Cov}[\boldsymbol{x}_0, \boldsymbol{\theta}] & \mathrm{Cov}[\boldsymbol{x}_0] \end{bmatrix} \tag{3.93} $$

which provides the desired linke between the null-space based methods illustrated here, and the standard methods. If prior information is not available, the above reduces to

$$\left(\mathbf{Z}^\intercal\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1}\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\mathbf{Z}\right)^{-1} = \left(\sum_{k=0}^{\bar{N}}\begin{bmatrix} {}^{y}_{\theta}\mathbf{S}_k^\intercal \\ {}^{y}_{x_0}\mathbf{S}_k^\intercal \end{bmatrix}\boldsymbol{\mathcal{B}}^{-1}\begin{bmatrix} {}^{y}_{\theta}\mathbf{S}_k & {}^{y}_{x_0}\mathbf{S}_k \end{bmatrix}\right)^{-1} \tag{3.94}$$

which is exactly the expression used in the computation of the Hessian according to section 3.2. Section 2.2.2 shows, that this may be used as approximation of the Fisher information matrix. Thus, the approach via the approximation as a linearized, weighted least-squares problem, together with the null-space method to incorporate integration defect constraints actually contains the two textbook approaches as special cases.

It was noted before that even if the system under consideration is assumed to be deterministic, the propagated states will themselves be random variables. A link to the standard methods used in aircraft system identification may be established by considering the states to be a function of the model parameters and initial values $\boldsymbol{x}_k = \boldsymbol{x}_k(\boldsymbol{\theta}, \boldsymbol{x}_0)$. A first order Taylor expansion around the estimates then yields

$$\boldsymbol{x}_k(\boldsymbol{\theta}, \boldsymbol{x}_0) \approx \hat{\boldsymbol{x}}_k\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0\right) + \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\right) + \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{x}_0}(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0) \tag{3.95}$$

$$= \hat{\boldsymbol{x}}_k\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0\right) + \begin{bmatrix} \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{x}_0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \end{bmatrix} \tag{3.96}$$

Since the estimates $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0$ are considered to be unbiased, the resulting estimation error has zero mean (at least to first order)

$$\mathrm{E}\left[\boldsymbol{x}_k(\boldsymbol{\theta}, \boldsymbol{x}_0) - \hat{\boldsymbol{x}}_k\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0\right)\right] = \begin{bmatrix} \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \boldsymbol{x}_0} \end{bmatrix}\mathrm{E}\left[\begin{bmatrix} \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \end{bmatrix}\right] = \mathbf{0} \tag{3.97}$$

Now, the model parameter error $\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ and state estimation errors for all samples $\boldsymbol{x}_k(\boldsymbol{\theta}, \boldsymbol{x}_0) - \hat{\boldsymbol{x}}_k\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_k\right)$ may be stacked vertically, and their covariance computed

$$\mathrm{Cov}\left[\begin{bmatrix} \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{x}_0(\boldsymbol{\theta}, \boldsymbol{x}_0) - \hat{\boldsymbol{x}}_0\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0\right) \\ \vdots \\ \boldsymbol{x}_{\bar{N}}(\boldsymbol{\theta}, \boldsymbol{x}_0) - \hat{\boldsymbol{x}}_{\bar{N}}\left(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{x}}_0\right) \end{bmatrix}\right] = \underbrace{\begin{bmatrix} \mathbf{I}_{n_{\boldsymbol{\theta}}} & \mathbf{0} \\ \frac{\partial \hat{\boldsymbol{x}}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{x}}_0}{\partial \boldsymbol{x}_0} \\ \vdots & \\ \frac{\partial \hat{\boldsymbol{x}}_{\bar{N}}}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{x}}_{\bar{N}}}{\partial \boldsymbol{x}_0} \end{bmatrix}}_{={}_{\mathrm{S}}\mathbf{Z}} \mathrm{Cov}\left[\begin{bmatrix} \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \end{bmatrix}\right] \cdot {}_{\mathrm{S}}\mathbf{Z}^\intercal \tag{3.98}$$

The matrices on the left and right are exactly the solutions to the sensitivity equations ${}_{\mathrm{S}}\mathbf{Z} = \begin{bmatrix} {}^{\theta}_{\tilde{\boldsymbol{z}}}\mathbf{S} & {}^{x}_{\tilde{\boldsymbol{z}}}\mathbf{S}_0 & \cdots & {}^{x}_{\tilde{\boldsymbol{z}}}\mathbf{S}_{\bar{N}} \end{bmatrix}$, see equation (3.81). Thus the multiplication of the condensed covariance with the null-space basis of equation (3.73) corresponds to a propagation of the parameter and initial value uncertainties, which again relates standard methods to the null-space method developed here.

Similar ideas may be used to obtain the output covariances, using the output sensitivities instead

$$\mathrm{Cov}\left[\left[\begin{array}{c} \boldsymbol{y}_0(\boldsymbol{\theta},\boldsymbol{x}_0) - \hat{\boldsymbol{y}}_0\left(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{x}}_0\right) \\ \vdots \\ \boldsymbol{y}_{\bar{N}}(\boldsymbol{\theta},\boldsymbol{x}_0) - \hat{\boldsymbol{y}}_{\bar{N}}\left(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{x}}_0\right) \end{array}\right]\right] = \underbrace{\left[\begin{array}{cc} \frac{\partial \hat{\boldsymbol{y}}_0}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{y}}_0}{\partial \boldsymbol{x}_0} \\ \vdots & \\ \frac{\partial \hat{\boldsymbol{y}}_{\bar{N}}}{\partial \boldsymbol{\theta}} & \frac{\partial \hat{\boldsymbol{y}}_{\bar{N}}}{\partial \boldsymbol{x}_0} \end{array}\right]}_{=\boldsymbol{\beta}} \mathrm{Cov}\left[\left[\begin{array}{c} \boldsymbol{\theta} - \hat{\boldsymbol{\theta}} \\ \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0 \end{array}\right]\right] \cdot \boldsymbol{\beta}^\intercal \tag{3.99}$$

The only difference then may arise, if the interpolation approach illustrated in section 3.9 is used. Then the interpolated output sensitivities on the fine measurement grid (see e.g. equation (3.129)) have to be used in above equation.

The last section provided a link between the novel approach based on the null space basis of the constraint Jacobian, and the standard methods for single shooting problem formulations. It was shown, that the classic approach is actually contained within the null space formulation as a special case, thus reassuring the correctness of the basic idea. The two major steps when taking constraints into account via $\mathbf{Z}$, have their counterpart in the classic method:

1. "condensing" the information on the free directions via

   $$\mathrm{Cov}[\boldsymbol{x}_0,\boldsymbol{\theta}] = \left(\mathbf{Z}^\intercal\left[\mathbf{X}_{\frac{\partial}{\partial \boldsymbol{z}}}\right]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1}\left[\mathbf{X}_{\frac{\partial}{\partial \boldsymbol{z}}}\right]\mathbf{Z}\right)^{-1}$$

   corresponds to solving the sensitivity equations and computing the cost function Hessian as approximation to the overall Fisher information matrix
2. "projecting" the condensed covariance ${}_{\mathrm{S}}\mathbf{Z} \cdot \mathrm{Cov}[\boldsymbol{x}_0,\boldsymbol{\theta}]^{-1} \cdot {}_{\mathrm{S}}\mathbf{Z}^\intercal$ corresponds to a propagation of the parameter and initial state uncertainty to obtain uncertainty estimates for all states $\boldsymbol{x}_k$

Further, the validity was checked for some simple numeric examples: the model parameter covariances $\mathrm{Cov}[\boldsymbol{\theta}]$ were computed using the null space approach and the classical formulation. The two results were equal to within the expected tolerance.

Only if a filter is used in combination with the single-shooting method, some differences arise: the filter sees the model parameters as deterministic constants, and computes state and output covariances based on this. It could be modified to to include model parameter uncertainties, too, but then two filter implementations would be necessary: one to be used in the estimation process, and one for covariance computations.

3.6 Nearly Singular Information Matrix

All of the approaches to compute approximate parameter uncertainties sooner or later involve the inversion of a matrix, which is quadratic in some sensitivity. This matrix can often be interpreted as a Hessian of the problem or its approximation. Now, there may arise situations, where the Hessian is nearly singular, and thus its inverse is not

easily computed. The following section discusses situations, how this may arise, and possible remedies.

At the heart of many of the following arguments lies the covariance approximation via a linearization of the problem

$$\mathrm{Cov}[\mathbf{z}] = \mathrm{Cov}[\delta\mathbf{z}] = \left(\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^{\intercal}\mathrm{Cov}[\boldsymbol{v}^a]^{-1}\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]\right)^{-1} \tag{3.100}$$

Thus many of the arguments used in linear least-squares estimation may be applied here, too.

MORELLI and KLEIN list several aspects that may result in an ill-conditioning of the Hessian [MK2016, Ch. 6.3]. The first would be over-parameterization, i.e. having too many model parameters to estimate, for the limited amount of information in the data. Then the estimates cannot be computed with sufficient accuracy.

Another problem is called "misspecification of the model" [MK2016, Ch. 6.3], i.e. the setting where several model parameters have more or less the same effect on the output. Then the respective columns of the concatenated output sensitivities $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ may be almost linearly dependent, which yields problems when inverting the Hessian. This is equivalent to having almost linearly dependent regressors in linear least-squares problems. A special case of this arises, if a parameter is to describe the influence of a quantity, which was constant during the experiment. Then the corresponding column of the concatenated output sensitivities $\left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]$ will be almost linearly dependent on possible bias parameters. Similarly, if a parameter does not significantly influence the outputs, the corresponding column in the concatenated output sensitivities will be close to zero, thus complicating the inversion of the Hessian.

If a nearly singular information matrix appears during the iterative solution of the optimization problem, this may be tolerated. Common solution algorithms such as the Levenberg-Marquardt algorithm 2.2 are able to cope with singular Hessians. Alternatively, a rank deficient inverse may be used, where only eigenvalues of the Hessian, that are significantly larger than zero are involved in the actual inversion [MK2016, Ch. 6.3]. This idea closely resembles the pseudo-inverse (see Appendix C.4), where not only singular values at zero, but also those smaller than a pre-defined threshold are neglected in the inversion.

More problematic is a nearly singular information matrix as final result of the optimization process, because it basically invalidates the result: Either has the model been misspecified, or there is not enough information for a certain subset of parameters [MIM1985, Ch. 5.3]. Both of which should be carefully questioned and might necessitate the collection of new data, or a different model formulation. As will be discussed in section 3.7, ill-conditioning of the Hessian will result in large covariance estimates, which can be used as an indication for near-singularity.

Another method of regularization, at least w.r.t. the model parameters, is inherent to Bayesian estimation with a Gaussian prior: Then, there is always a positive definite

matrix that is added to the concatenated, squared, output sensitivities in computing the Hessian, see equation (2.116). Thus its conditioning is improved for those parameters, where there might not be enough information in the data to determine them accurately.

Overall, although the Hessian of the problem should not be singular from an optimization point of view, it may happen in real-life application. The foregoing section lists some possible reasons for this, which may help to overcome the issue. Alternatively, arguments are presented as the basis for a reasoning, why the situation is tolerable.

3.7 An Optimization View on Parameter Uncertainties

One of the overall goals of this thesis is to bring the fields of parameter estimation and optimal control closer together. After an estimation has been performed, it is often instructive to inspect the parameter covariances, which is now illuminated from an optimization point of view.

The general discussion on parameter uncertainties in section 2.2.5 was based on statistical considerations, the Fisher information matrix and the notion of efficiency. In the applications discussed here, these parameter covariances are computed from the inverse of the cost function Hessian. The remainder of this section now investigates the desirable statistical properties of parameter estimates (low standard deviations, low pair-wise correlations) from an optimization point of view.

The content of this section is easiest understood, if only model parameters $\boldsymbol{\theta}_0$, and possibly initial conditions $\boldsymbol{x}_0$ are considered as elements of the optimization vector. The arguments may be extended to the full-discretization case. However, instead of considering the complete parameter covariance, the condensed version in the free directions $\left(\mathbf{Z}^\intercal \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right]^\intercal \mathrm{Cov}[\boldsymbol{v}^a]^{-1} \left[\mathbf{X}_{\frac{\partial}{\partial \mathbf{z}}}\right] \mathbf{Z}\right)^{-1}$ should then be considered. Since this significantly complicates notation and may hide the key points here, the focus will lie on the interpretation of the classic single shooting case.

Most of the discussion to follow will be based on the fact that the Fisher information matrix is approximated as the cost function Hessian

$$\mathrm{Cov}[\mathbf{z}^*] = \boldsymbol{\mathcal{F}}(\mathbf{z}^*)^{-1} \approx \left(\frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2}\right)^{-1} \tag{3.101}$$

Since the Hessian is necessarily symmetric, its eigendecomposition may be used in above expression

$$\begin{aligned} \frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2} &= \begin{bmatrix} \boldsymbol{t}_1 & \cdots & \boldsymbol{t}_{n_\mathbf{z}} \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n_\mathbf{z}} \end{bmatrix} \begin{bmatrix} \boldsymbol{t}_1^\intercal \\ \vdots \\ \boldsymbol{t}_{n_\mathbf{z}}^\intercal \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{t}}_1^\intercal \\ \vdots \\ \tilde{\boldsymbol{t}}_{n_\mathbf{z}}^\intercal \end{bmatrix} \boldsymbol{\Lambda} \begin{bmatrix} \tilde{\boldsymbol{t}}_1 & \cdots & \tilde{\boldsymbol{t}}_{n_\mathbf{z}} \end{bmatrix} \\ &= \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^\intercal = \sum_{j=1}^{n_\mathbf{z}} \lambda_j \boldsymbol{t}_j \boldsymbol{t}_j^\intercal \end{aligned} \tag{3.102}$$

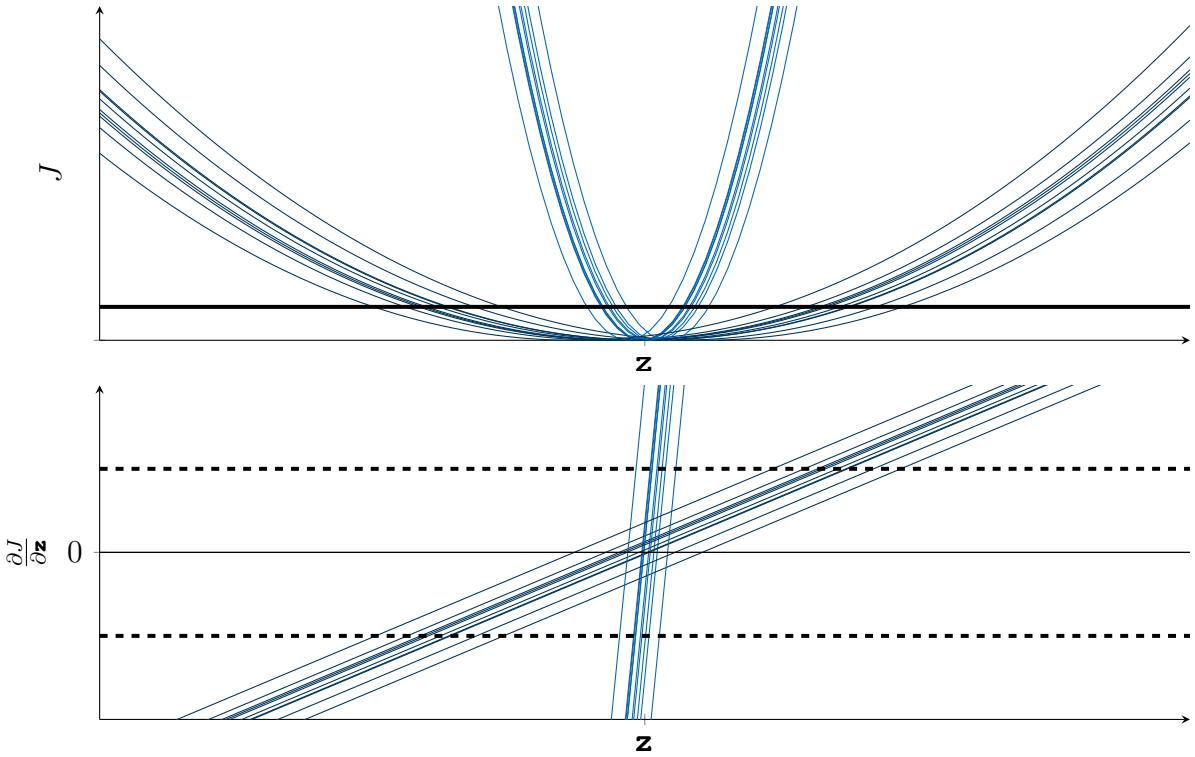


Figure 3.2: *Two cost functions and their gradients with high (——) and low (——) curvature for several realizations of the underlying noise process; an exemplary threshold is given, below which the cost function (——) is considered to not change significantly, and/or its gradient (- - -) is considered to be "small enough"*

where $\boldsymbol{t}_j$ denotes the j-th column of $\mathbf{T}$, and $\tilde{\boldsymbol{t}}_j$ the j-th row.

Now, from an optimization point of view, it is desirable to have a high curvature of the cost function at the optimum, since this leads to a well-defined minimum. Figure 3.2 illustrates this: the higher the curvature, the "better defined" is the minimum, i.e. the smaller is the parameter range, for which the cost function may be considered "to not change significantly anymore" by the optimization algorithm in order to stop the iteration. Also, random variations in the cost function, due to different realizations of the underlying noise process have a lower influence on the resulting minimum, if the curvature is higher (and thus the gradient steeper).

In general, the signed curvature of a function $f(x)$ is defined as [MV2003, Ch. 5]

$$\kappa(x) = \frac{\frac{d^2 f(x)}{dx^2}}{\left(1 + \frac{df(x)}{dx}\right)^{\frac{3}{2}}} \tag{3.103}$$

This can be applied to a quadratic approximation of the cost function, at the optimum, in the direction $\boldsymbol{v}$, where $\boldsymbol{v}$ is assumed to be of unit length

$$J(\gamma, \boldsymbol{v}) \approx J(\mathbf{z}^*) + \frac{1}{2}(\gamma \boldsymbol{v})^\intercal \frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2}(\gamma \boldsymbol{v}) \tag{3.104}$$

The first and second derivatives, in the direction $\boldsymbol{v}$ w.r.t. the *scalar* argument γ are now

$$\frac{dJ(\gamma, \boldsymbol{v})}{d\gamma} = \gamma \boldsymbol{v}^\mathsf{T} \frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2} \boldsymbol{v} \tag{3.105}$$

$$\frac{d^2 J(\gamma \boldsymbol{v})}{d\gamma^2} = \boldsymbol{v}^\mathsf{T} \frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2} \boldsymbol{v} \tag{3.106}$$

which may be combined to obtain the curvature

$$\kappa_J(\gamma, \boldsymbol{v}) \approx \frac{\frac{d^2 J(\gamma \boldsymbol{v})}{d\gamma^2}}{\left(1 + \frac{dJ(\gamma, \boldsymbol{v})}{d\gamma}\right)^{\frac{3}{2}}} \tag{3.107}$$

Evaluating this at the optimum (i.e. $\gamma = 0$) can then be used to investigate the cost function curvature in different directions $\boldsymbol{v}$

$$\kappa_J(0, \boldsymbol{v}) \approx \boldsymbol{v}^\mathsf{T} \frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2} \boldsymbol{v} = \boldsymbol{v}^\mathsf{T} \left(\sum_{j=1}^{n_\mathbf{z}} \lambda_j \boldsymbol{t}_j \boldsymbol{t}_j^\mathsf{T} \right) \boldsymbol{v} \tag{3.108}$$

From above equation, and the ortho-normality of the eigenvectors, it can be seen, that the cost function curvature is at maximum equal to the largest eigenvalue of the Hessian λ_{max} for $\boldsymbol{v} = \boldsymbol{t}_{max}$, where $\boldsymbol{t}_{max}$ is the corresponding eigenvector. The inverse is also true, i.e. the smallest curvature λ_{min} results for $\boldsymbol{v} = \boldsymbol{t}_{min}$. Thus for a well-defined minimum, it is in general desirable to have eigenvalues, that are as large as possible.

Now, relating the eigenvalues and eigenvectors of the Hessian to the parameter covariance estimate yields

$$\mathrm{Cov}[\mathbf{z}^*] \approx \left(\frac{\partial^2 J(\mathbf{z}^*)}{\partial \mathbf{z}^2} \right)^{-1} = \mathbf{T} \boldsymbol{\Lambda}^{-1} \mathbf{T}^\mathsf{T} = \sum_{j=1}^{n_\mathbf{z}} \frac{\boldsymbol{t}_j \boldsymbol{t}_j^\mathsf{T}}{\lambda_j} \tag{3.109}$$

From this, variances and cross variances may be extracted

$$\mathrm{Cov}\left[[\mathbf{z}^*]_{(k)}, [\mathbf{z}^*]_{(l)} \right] \approx \sum_{j=1}^{n_\mathbf{z}} \frac{[\boldsymbol{t}_j]_{(k)} [\boldsymbol{t}_j]_{(l)}}{\lambda_j} = \tilde{\boldsymbol{t}}_k^\mathsf{T} \boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{t}}_l \tag{3.110}$$

$$\mathrm{Var}\left[[\mathbf{z}^*]_{(k)} \right] = \mathrm{Cov}\left[[\mathbf{z}^*]_{(k)}, [\mathbf{z}^*]_{(k)} \right] \approx \sum_{j=1}^{n_\mathbf{z}} \frac{[\boldsymbol{t}_j]_{(k)}^2}{\lambda_j} = \tilde{\boldsymbol{t}}_k^\mathsf{T} \boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{t}}_k \tag{3.111}$$

which may then be combined to obtain the correlation coefficient

$$\begin{aligned} \rho\left[[\mathbf{z}^*]_{(k)}, [\mathbf{z}^*]_{(l)} \right] &= \frac{\mathrm{Cov}\left[[\mathbf{z}^*]_{(k)}, [\mathbf{z}^*]_{(l)} \right]}{\sqrt{\mathrm{Var}\left[[\mathbf{z}^*]_{(k)} \right]} \sqrt{\mathrm{Var}\left[[\mathbf{z}^*]_{(l)} \right]}} \\ &\approx \frac{\sum_{j=1}^{n_\mathbf{z}} \frac{[\boldsymbol{t}_j]_{(k)} [\boldsymbol{t}_j]_{(l)}}{\lambda_j}}{\sqrt{\sum_{j=1}^{n_\mathbf{z}} \frac{[\boldsymbol{t}_j]_{(k)}^2}{\lambda_j}} \sqrt{\sum_{j=1}^{n_\mathbf{z}} \frac{[\boldsymbol{t}_j]_{(l)}^2}{\lambda_j}}} = \frac{\tilde{\boldsymbol{t}}_k^\mathsf{T} \boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{t}}_l}{\sqrt{\tilde{\boldsymbol{t}}_k^\mathsf{T} \boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{t}}_k} \sqrt{\tilde{\boldsymbol{t}}_l^\mathsf{T} \boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{t}}_l}} \end{aligned} \tag{3.112}$$

MORELLI and KLEIN present a similar approach for detection of data-collinearity in linear least-squares estimation [MK2016, Ch. 5.5], however the ideas behind that are now generalized to the solution of non-linear estimation problems.

As mentioned before, comparatively large eigenvalues are desirable from an optimization point of view. They also influence the parameter variance estimates in a positive manner: equation (3.111) shows, that if the Hessian's eigenvalues are large, only small terms will be summed up to yield the parameter variance estimate [MK2016, Ch. 5.5]. Also, if the Hessian is well-conditioned, i.e. if the relation between the largest and smallest eigenvalue is close to 1, the cross-variances between the parameters will be close to zero, due to the orthogonality of the eigenvectors

$$\frac{\lambda_{max}}{\lambda_{min}} \approx 1 \qquad \Leftrightarrow \qquad \lambda_{max} \approx \lambda_{min} \approx \lambda$$

$$\mathrm{Cov}\left[\left[\mathbf{z}^*\right]_{(k)}, \left[\mathbf{z}^*\right]_{(l)}\right] \approx \frac{1}{\lambda}\tilde{\boldsymbol{t}}_k^{\mathsf{T}}\mathbf{I}_{n_{\mathbf{z}}}\tilde{\boldsymbol{t}}_l = \begin{cases} \frac{1}{\lambda} & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \tag{3.113}$$

If on the other hand, one eigenvalue λ_j is considerably smaller than the rest, this will lead to high variances for those parameters $\mathrm{Var}\left[\left[\mathbf{z}^*\right]_{(k)}\right]$, where the corresponding element of the eigenvector $\left[\boldsymbol{t}_j\right]_{(k)}$ is significantly non-zero. To quantify this adverse effect, MORELLI and KLEIN present the *condition index*

$$\frac{\lambda_{max}}{\lambda_j} \geq 1 \tag{3.114}$$

where the condition index of the smallest eigenvalue is known as the Hessian's condition number [MK2016, Ch. 5.5]. Again, they present this in the context of linear least-squares estimation, but it can be applied to non-linear problems, too. As a rule of thumb MORELLI and KLEIN mention that condition indices in the range 100-1000 may indicate data collinearity problems, and that in some cases problems already arose with condition indices lower than 100 [MK2016, Ch. 5.5].

If a $\boldsymbol{t}_j$ has several elements, that are significantly different from zero, they will contribute significantly in the correlation coefficient via the $\sum_{j=1}^{n_{\mathbf{z}}} \frac{\left[\boldsymbol{t}_j\right]_{(k)}\left[\boldsymbol{t}_j\right]_{(l)}}{\lambda_j}$ term in equation (3.112) [MK2016, Ch. 5.5]. To quantify the influence of an eigenvalue λ_j on the k-th parameter variance estimate, MORELLI and KLEIN mention the variance proportion[5] [MK2016, Ch. 5] for linear least squares problems which is generalized here

$$\pi_{k,j} = \frac{\frac{\left[\boldsymbol{t}_j\right]_{(k)}^2}{\lambda_j}}{\sum_{i=1}^{n_{\mathbf{z}}} \frac{\left[\boldsymbol{t}_i\right]_{(k)}^2}{\lambda_i}} = \frac{\frac{\left[\boldsymbol{t}_j\right]_{(k)}^2}{\lambda_j}}{\mathrm{Var}\left[\left[\mathbf{z}^*\right]_{(k)}\right]} \tag{3.115}$$

MORELLI and KLEIN then state that if one eigenvalue is comparatively small, and if several variance proportions are larger than 0.5, this might indicate a data collinearity problem [MK2016, Ch. 5.5]. The variance proportions thus offer another possibility of analyzing a result, and to detect possible problems.

Thus, favorable properties of the optimization problem, in terms of eigenvalues and eigenvectors, map to favorable properties of the estimation problem. Then, even

[5] the original equation in the first edition [KM2006] misses the squares for the $\left[\boldsymbol{t}_j\right]_{(k)}^2$, which is corrected in [KM2018]

if the underlying theoretical basics for statistic interpretations may be questioned, it may still make sense to have a look at parameter variances and correlations in order to detect problems with the resulting *optimization* problem.

This discussion provides a further beneficial link between the fields of optimization and parameter estimation. Although MORELLI and KLEIN discussed many of the above ideas in the context of data collinearity in regression analysis, here it is shown how these concepts may be extended to the general estimation case. Further, basing the discussion on the cost function curvature has not yet been encountered by the author.

3.8 Automatic Improvement of Initial Guesses During Initial Optimization Iterations

The availability of meaningful initial guesses is crucial in any iterative optimization procedure: only if the initial guess is in the region of attraction of the desired optimum, the NLP solver will provide meaningful results. Thus, any additional effort to provide "good" initial guesses, or to improve upon existing ones is worth the effort.

Two approaches will be illustrated that are capable of improving upon the initial parameter guess $\boldsymbol{\theta}_0$, which helps to make the overall process more robust against disadvantageously chosen initial model parameters. Then, for some cases even an initial guess in the form of the zero-vector may eventually yield good results. Both of the approaches require an auxiliary source for aircraft state information ${}^{m}\boldsymbol{x}_k \; k = 0 \ldots \bar{N}$, which can be the result of a Flight Path Reconstruction (FPR) step (see section 5.2), or direct measurement of the states. This auxiliary state information is then used instead of integrated states, until the model parameters $\boldsymbol{\theta}$ are considered "good enough" so that the propagated states actually provide a benefit in the solution of the overall problem.

3.8.1 Sensitivity Computation Using Measured States

The first approach is based on an idea by TYLOR and ILIFF [TI1972]. They consider the case of linear dynamic models and the classic single shooting approach based on solving the system equation, computing output sensitivities and applying a Newton-Raphson algorithm to find the best model parameter estimates. Their proposition is to initially compute the output sensitivities ${}^{y}_{\mathbf{z}}\mathbf{S}$ of equations (3.46) and (3.44) based on ${}^{m}\boldsymbol{x}$ rather than the solution of the system equation, i.e. propagated $\boldsymbol{x}$

$$\frac{d}{dt}\,{}^{x}_{\theta}\mathbf{S} = \frac{\partial \boldsymbol{f}({}^{m}\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_i)}{\partial \boldsymbol{x}} \cdot {}^{x}_{\theta}\mathbf{S} + \frac{\partial \boldsymbol{f}({}^{m}\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_i)}{\partial \boldsymbol{\theta}} \tag{3.116}$$

$${}^{y}_{\theta}\mathbf{S} = \frac{\partial \boldsymbol{g}({}^{m}\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_i)}{\partial \boldsymbol{x}_k} \cdot {}^{x}_{\theta}\mathbf{S} + \frac{\partial \boldsymbol{g}({}^{m}\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}_i)}{\partial \boldsymbol{\theta}} \tag{3.117}$$

These are then used for computing the cost function gradient and Hessian (see section 3.3.3). The propagated solution $\hat{\boldsymbol{x}}_{k+1}$ to the system equation is only used in the computation of the residuals, which enter the gradient and actual cost function terms.

The reasoning behind this is that these "measured" sensitivities will be closer to the final solution, compared to those, that are computed based on a (possibly diverging) solution of the system equation, using bad initial parameter guesses. Thus the corresponding update step can put the parameter vector in close proximity to the final solution, independent of the quality of the initial guess [MK2016, Ch. 6.3]. This effect is more pronounced, the "more convex" the cost function is.

In their original report [TI1972] TYLOR and ILIFF state, that they apply this correction only in the first iteration. However, depending on the problem at hand, it might be of use to do several iterations using above approach, to make the most of the auxiliary state information.

3.8.2 Equation Error Approach

The second, novel approach is also based on using measured ${}^{m}\boldsymbol{x}$ instead of propagated states, but is closer related to the full discretization parameter estimation point of view. The basic idea behind it is to minimize a cost function that is quadratic in the integration defects (using measured ${}^{m}\boldsymbol{x}$), purely by adjusting the model parameter values $\boldsymbol{\theta}$. Since the trajectory is then necessarily close to the measured states, the "improved" model parameters are more likely to yield a valid state trajectory when being used in the propagation.

If all state variables were known in a fully discretized optimal control problem, only the model parameters $\boldsymbol{\theta}$ remain free and can thus be used to fulfill one (possibly implicit) integration step (see the discussion of equation (2.56))

$$ {}^{m}\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\, {}^{m}\boldsymbol{x}_k, {}^{m}\boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) \tag{3.118} $$

Combining this with the output equations, an augmented measurement and model output vector can be built

$$ \tilde{\boldsymbol{z}}_k = \begin{bmatrix} {}^{m}\boldsymbol{x}_{k+1} \\ \boldsymbol{z}_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\, {}^{m}\boldsymbol{x}_k, {}^{m}\boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta}) \\ \boldsymbol{g}_k({}^{m}\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta}) \end{bmatrix} + \tilde{\boldsymbol{v}}_k = \tilde{\boldsymbol{y}}(\boldsymbol{\theta})_k + \tilde{\boldsymbol{v}}_k \tag{3.119} $$

where the first part of $\tilde{\boldsymbol{v}}$ acts on the system dynamics in a fashion comparable to process noise; and the second part can be identified as measurement noise. Forming a quadratic cost function then resembles a relaxation approach to the fully discretized problem, where instead of exactly fulfilling the equality constraints, they have been included as quadratic cost via terms involving $({}^{m}\boldsymbol{x}_{k+1} - \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\circ, \boldsymbol{\theta}))$.

If the interpolation scheme of section 3.9 is used, there will be fewer state values ${}^{m}\boldsymbol{x}_k$, than measurement $\boldsymbol{z}_k$ and output $\boldsymbol{y}_k$ values. However, this can be tackled

with the covariance scaling of section 3.10.2, or by simply considering states and outputs separately in the quadratic cost.

Above is a static system, which can be solved using the equation error approach illustrated in section 2.2.2.3, because the state integration has been circumvented by using the measured states instead. Also, the output sensitivities simplify considerably, since no integrations are involved

$$\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\, {}^m\boldsymbol{x}_k, {}^m\boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{g}_k({}^m\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \end{bmatrix} \tag{3.120}$$

where the derivative of $\boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\circ)$ depends on the scheme in use. For example in the case of trapezoidal integration, this yields

$$\frac{\partial \boldsymbol{\Phi}_{\boldsymbol{f}}(t_k, t_{k+1}|\, {}^m\boldsymbol{x}_k, {}^m\boldsymbol{x}_{k+1}, \boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{t_{k+1} - t_k}{2} \left(\frac{\partial \boldsymbol{f}({}^m\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{f}({}^m\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) \tag{3.121}$$

An approach with some similarities to the one above has been reported in [Jat2015, Ch. 4], with the big difference being its treatment in continuous time. This in turn necessitates measurements (or numerical approximations) of the state derivative, introducing more uncertainty. In contrast, here the discretized state equations are used directly, avoiding the computation / measurement of state derivatives.

For both of the approaches it is paramount to have a reliable, auxiliary source for state information at one's disposal, in connection with a suitable model structure. If this can be considered given, especially the second approach has seen to be very robust: errors do not propagate by solving the system equation, since every sampling instant is considered independently.

Apart from being robust, this also alleviates correlations between parameters of the state and output equation that may have a comparable effect when the system is propagated: e.g. in a linear system, input and output gains influence the output in the same way, and are thus often almost perfectly correlated. However, in above approach, they appear at different places in the equations, and thus an independent determination may be possible.

The downside is, that the integration defects can probably not be driven to zero, thus the system equation will not be fulfilled, and a subsequent estimation using the methods illustrated earlier can in general not be avoided. However, having a robust method for the improvement of disadvantageously chosen initial parameter guesses at ones disposal, is definitely worth the additional implementation effort. Especially since above problem is well behaved, and should converge to meaningful results within few iterations, in addition to the justified hope of ending up within the region of attraction of the final result.

BOCK briefly describes a similar approach of using the model parameters $\boldsymbol{\theta}$ to improve the "consistency" of the problem (i.e. the degree to which the equality discretization constraints are fulfilled) [Boc1987, Ch. 2], however he only discusses problems that are then linear in the parameters. The approach as illustrated here, is applicable to general, non-linear systems. For linear systems, the problem is easy enough to almost guarantee convergence to good starting values. Unfortunately this cannot be said for general, non-linear problems.

In summary, especially the equation error approach, which has in this form not been encountered in aircraft system identification, provides a robust means to improve upon initial parameter guesses. In most of the examples to be illustrated in chapter 6, this approach was used, together with $\boldsymbol{\theta}_0 = \mathbf{0}$, to obtain meaningful parameter estimates. Apart from its robustness, a further advantage is that the model structure to be used in the actual estimation is directly incorporated. No translation of auxiliary initial parameters to the model formulation at hand is necessary. This may be necessary if other sources were to be used, that do not fully comply with the desired structure.

3.9 Efficient Interpolation of Large Sample Sizes

In common flight data analyses, the sample size can grow very large. It is not uncommon to have between ten and fifteen outputs to match, at a sample rate of 50Hz to 100Hz and for several maneuvers simultaneously, each maneuver lasting around 10 to 60 seconds or even longer. The resulting large sample sizes then result in very large optimization problems and consequently comparatively long computation times. Then, the inevitable testing of different model structures becomes quite cumbersome: the analyst has to wait comparatively long before being able to make an informed decision whether to discard or investigate the current model formulation further. Here, an approach is illustrated that is able to reduce the problem size, while still producing reasonable results.

3.9.1 Interpolation via B-Splines

A possibility to obtain results faster while still getting meaningful results in the trial and error phase of determining a suitable model formulation is proposed next. The main idea is to compute the model outputs on a rather coarse grid first, before efficiently interpolating them onto the finer measurement grid. This is possible, since all of the approaches treated here only consider a model output, nowhere is it ever required to integrate the state equations of the underlying dynamic system on the measurement time grid. The idea was first introduced in the master's thesis [Gno2016], which was co-supervised by the author.

For the actual implementation, B-splines of a given order n are used, details on

which can be found in appendix A.3. The basis for interpolation is a function, defined in terms of a knot sequence $\mathbf{t}$ (and thus spline basis function ${}_{\mathbf{t}}B_j^n(t)$) and coefficient vector $\boldsymbol{\alpha}$

$$ {}_{\mathbf{t}}S^n(t) = \sum_j \alpha_j \, {}_{\mathbf{t}}B_j^n(t) \qquad \alpha_j \in \mathbb{R} \tag{3.122} $$

Now, if the spline representation of a data-set of the l-th output $[\boldsymbol{y}_k]_{(l)}$ $k = 0 \dots N-1$ using the knot sequence $\mathbf{t}$ is sought, the coefficients α_j have to follow

$$ [\boldsymbol{y}_k]_{(l)} = {}_{\mathbf{t}}S^n(t_k) = \sum_j \alpha_j \, {}_{\mathbf{t}}B_j^n(t_k) \qquad k = 0, \dots, N-1 \tag{3.123} $$

which can be noted in matrix form as

$$ \begin{bmatrix} {}_{\mathbf{t}}B_0^n(t_0) & {}_{\mathbf{t}}B_1^n(t_0) & \dots \\ {}_{\mathbf{t}}B_0^n(t_1) & {}_{\mathbf{t}}B_1^n(t_1) & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \end{bmatrix} = \mathbf{A}\boldsymbol{\alpha} = \begin{bmatrix} [\boldsymbol{y}_0]_{(l)}(\mathbf{z}) \\ \vdots \\ [\boldsymbol{y}_{N-1}]_{(l)}(\mathbf{z}) \end{bmatrix} \tag{3.124} $$

Due to the local support of the B-spline functions, the matrix $\mathbf{A}$ has a band-diagonal structure and can be stored efficiently. If the knot sequence for the B-spline functions is chosen to contain exactly $N_{knots} = N$ knots, above equation has a unique solution for α, since $\mathbf{A} \in \mathbb{R}^{n_N \times n_N}$.

Returning to the introductory idea of computing the model outputs on a coarse grid, before interpolating them onto the measurement grid to be used in the cost function, the approach is now as follows: Given the time-grid of measurement points $t_0, \dots t_{N-1}$ a coarser grid ${}^c t_0, \dots {}^c t_{{}^c N-1}$ with ${}^c N < N$, together with a suitable knot sequence ${}^c\mathbf{t}$ and spline order n is defined. Then, the values of the l-th output on the coarse grid $[{}^c\boldsymbol{y}_k]_{(l)}$ and the values of the l-th output on the fine grid $[\boldsymbol{y}_k]_{(l)}$ can both be expressed via the same coefficient vector ${}^c\boldsymbol{\alpha}$ of the coarse grid

$$ \begin{bmatrix} {}_{{}^c\mathbf{t}}B_0^n({}^c t_0) & \dots & {}_{{}^c\mathbf{t}}B_{{}^c N-1}^n({}^c t_0) \\ \vdots & \ddots & \vdots \\ {}_{{}^c\mathbf{t}}B_0^n\left({}^c t_{{}^c N-1}\right) & \dots & {}_{{}^c\mathbf{t}}B_{{}^c N-1}^n\left({}^c t_{{}^c N-1}\right) \end{bmatrix} \begin{bmatrix} {}^c\alpha_0 \\ \vdots \\ {}^c\alpha_{{}^c N-1} \end{bmatrix} = {}^c\mathbf{A}\,{}^c\boldsymbol{\alpha} = \begin{bmatrix} [{}^c\boldsymbol{y}_0]_{(l)}(\mathbf{z}) \\ \vdots \\ \left[{}^c\boldsymbol{y}_{{}^c N-1}\right]_{(l)}(\mathbf{z}) \end{bmatrix} \tag{3.125} $$

$$ \begin{bmatrix} {}_{{}^c\mathbf{t}}B_0^n(t_0) & \dots & {}_{{}^c\mathbf{t}}B_{{}^c N-1}^n(t_0) \\ \vdots & \ddots & \\ {}_{{}^c\mathbf{t}}B_0^n(t_{N-1}) & \dots & {}_{{}^c\mathbf{t}}B_{{}^c N-1}^n(t_{N-1}) \end{bmatrix} \begin{bmatrix} {}^c\alpha_0 \\ \vdots \\ {}^c\alpha_{{}^c N-1} \end{bmatrix} = \mathbf{A}\,{}^c\boldsymbol{\alpha} = \begin{bmatrix} [\boldsymbol{y}_0]_{(l)}(\mathbf{z}) \\ \vdots \\ [\boldsymbol{y}_{N-1}]_{(l)}(\mathbf{z}) \end{bmatrix} \tag{3.126} $$

with the two collocation matrices ${}^c\mathbf{A} \in \mathbb{R}^{{}^c N \times {}^c N}$ and $\mathbf{A} \in \mathbb{R}^{N \times {}^c N}$ based on the same B-spline basis functions of the knot sequence associated with the coarse grid ${}^c\mathbf{t}$. The matrices arise by evaluating the ${}_{{}^c\mathbf{t}}B_j^n(t)$ once on the coarse time grid ${}^c t_k$ and a second time on the fine time grid t_k. The relation between the outputs on the different grids is

then

$$\begin{bmatrix} [\boldsymbol{y}_0]_{(l)} \\ \vdots \\ [\boldsymbol{y}_{N-1}]_{(l)} \end{bmatrix} = \mathbf{A} \cdot {}^c\boldsymbol{\alpha} = \mathbf{A} \cdot {}^c\mathbf{A}^{-1} \begin{bmatrix} [{}^c\boldsymbol{y}_0]_{(l)} \\ \vdots \\ \left[{}^c\boldsymbol{y}_{{}^cN-1}\right]_{(l)} \end{bmatrix} \tag{3.127}$$

which is merely a linear mapping.

Furthermore, the collocation matrices $\mathbf{A}$ and ${}^c\mathbf{A}$ are valid for all outputs that were measured on the same grid, i.e. no separate interpolation matrices have to be determined for the separate outputs. Only if several maneuvers are to be considered simultaneously, each needs its own interpolation matrices.

Thus incorporating the interpolation scheme for all outputs at all sampling instants comes down to modifying the model outputs using a constant factor

$$\begin{bmatrix} \boldsymbol{y}_0(\mathbf{z}) & \cdots & \boldsymbol{y}_{N-1}(\mathbf{z}) \end{bmatrix} = \begin{bmatrix} {}^c\boldsymbol{y}_0(\mathbf{z}) & \cdots & {}^c\boldsymbol{y}_{{}^cN-1}(\mathbf{z}) \end{bmatrix} {}^c\mathbf{A}^{-\intercal}\mathbf{A}^{\intercal} \tag{3.128}$$

Since the collocation matrices are constant, the output sensitivities are interpolated in the same fashion, where the biggest remaining challenge is to get the indexing right. One possibility is to consider one optimization variable at a time

$$\begin{bmatrix} \frac{\partial \boldsymbol{y}_0}{\partial [\mathbf{z}]_{(j)}}(\mathbf{z}) & \cdots & \frac{\partial \boldsymbol{y}_{N-1}}{\partial [\mathbf{z}]_{(j)}}(\mathbf{z}) \end{bmatrix} = \begin{bmatrix} \frac{\partial\, {}^c\boldsymbol{y}_0}{\partial [\mathbf{z}]_{(j)}}(\mathbf{z}) & \cdots & \frac{\partial\, {}^c\boldsymbol{y}_{{}^cN-1}}{\partial [\mathbf{z}]_{(j)}}(\mathbf{z}) \end{bmatrix} {}^c\mathbf{A}^{-\intercal}\mathbf{A}^{\intercal} \qquad j = 1 \dots n_{\mathbf{z}} \tag{3.129}$$

Thus the spline interpolation approach may be incorporated, by simply adjusting model outputs and output sensitivities, after having integrated them on the coarse grid. All other aspects as illustrated so far are then still valid.

Finally, since both $\mathbf{A}$ and ${}^c\mathbf{A}$ are sparse, not only is this approach memory-saving, but above equations can also be solved efficiently. An example for the sparsity patterns of the two matrices can be seen in figure 3.3.

3.9.2 Knot Sequence Determination Using Model Dynamics

Additionally, interpolation accuracy can be improved by an advantageous choice of the knot sequence ${}^c\mathbf{t}$, which constitutes an advancement over the approach presented in [Gno2016]. The basic idea is to chose the knots such, that they are more densely located on the time grid, where higher resolution is necessary, and knot distances are increased where lower resolution may be tolerated. This can be achieved by employing ideas similar to those used in mesh-refinement for optimal control problems as illustrated in [Bit2017], where density functions are used to determine the necessity for finer/coarser meshes.

A first, intuitive idea would be to base the necessity for denser grid points on a "measure of activity" in the system in the form of the state derivatives $\dot{\boldsymbol{x}}$, i.e. to make

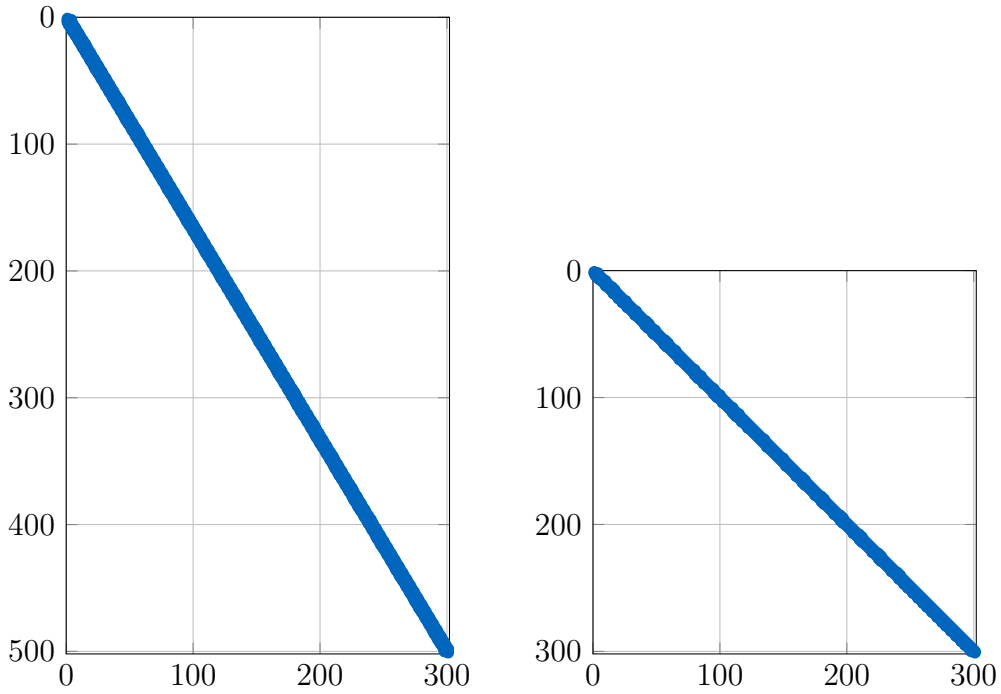


Figure 3.3: *Example sparsity patterns for* $\mathbf{A}$ *(left) and* $\tilde{\mathbf{A}}$ *(right)*

the grid more dense depending on a norm of $\dot{x}$. However, if many states are to be considered simultaneously, each having a different order of magnitude (e.g. velocity vs. rotational rate), the choice of proper scaling and norm quickly becomes challenging. This is why a different approach was pursued here: the time discretization of a variable step-size solver will be used to generate a density function for knot placement.

The so-called RUNGE-KUTTA-FEHLBERG class of algorithms exploit the characteristics of multi-step integration schemes to efficiently generate two estimated results of the same integration step. By comparing those two results, the following conclusions can be drawn:

- the integration step size may be increased, if the two results coincide with a very small error
- the integration step size should be kept constant, if the two results agree with a tolerable error
- the integration step size should be decreased, if the two results do not agree at the desired level

A widely used example of this approach is the DORMAND-PRINCE method [SBB+2010], also implemented in MATLAB's `ode45` solver. Further details can be found in the relevant textbooks on numerical methods.

These algorithms produce a time-grid, which is adapted to the dynamics of the model under investigation: the integration steps are smaller, if higher accuracy is necessary and are larger, if less accuracy may be tolerated. This is exactly what is needed

in order to find an appropriate knot sequence, which is why the proposed approach is as follows:

1. Do one forward simulation in time, using a variable step size solver belonging to the RUNGE-KUTTA-FEHLBERG class using the initial model parameters $\boldsymbol{\theta}_0$ to generate the sequence of integration times $t_k^{RKF}\ k = 0 \dots N-1$
2. interpolate this sequence using the desired number of samples ${}^cN$ to obtain the less dense sequence of integration times ${}^ct_k^{RKF}\ k = 0 \dots {}^cN-1$
3. determine a suitable knot sequence ${}^c\mathbf{t}$ from ${}^ct_k^{RKF}\ k = 0 \dots {}^cN-1$, which is in turn used to determine the collocation matrices ${}^c\mathbf{A}$ and $\mathbf{A}$

Even though the forward simulation increases the computational work load when determining the knot sequence, this is only done once in the beginning and is far outweighed by the reduction in computational time during the optimization iteration. Also, the knot sequence will be determined on the basis of an *initial guess* $\boldsymbol{\theta}_0$ of the model parameters. However for any parameter estimation to be successful, those parameters should have reasonable values, which is why the approximation of the desired density in the time grid is still assumed to be valid. If necessary, "bad" initial guesses may be improved upon by using the approaches of section 3.8.

An illustration can be found in figure 3.4: The variable step size integration time steps $t_k^{RKF}\ k = 0 \dots N-1$ (blue), are plotted over the normalized sample number $\frac{k}{N}$. Then fewer ${}^cN$ normalized sample numbers $\frac{{}^ck}{{}^cN}$ are distributed uniformly on the x-axis, and used to interpolate the original integration step times t^{RKF} (dashed orange). This eventually yields the thinned-out integration time sequence ${}^ct^{RKF}$ on the y-axis. The figure shows, that at around $0.5s$, $1.5s$, and $2.5s$, the integration step size had to be reduced in order to keep the error small. At these time instants, also the new knot sequence is denser than for the rest of the time.

For a few example simulation maneuvers, performed in the lateral plane of a simulation model for a small scale RC model, an approximation for the error using the spline interpolation approach can be seen in figure 3.5. The reference solution is obtained using an 8$^{\text{th}}$ order integration scheme on the fine grid. In the left column, knot sequences for reduction factors $\frac{{}^cN}{N}$ are generated uniformly over the maneuver time span, whereas in the right column they are generated using above described approach. Integration errors can be considered small overall, amounting to about 4% of the maximum amplitude in the respective signals. Furthermore, with the same reduction factors, a major decrease of the error compared to the baseline solution can be achieved.

To sum it up, this approach can be used to trade the possibly computationally costly determination of model outputs for an efficient interpolation of the outputs onto the time grid of the measurements. The advantages of this approach are comparable to those of variable step-size solvers: while keeping the accuracy high, computational costs can be reduced considerably. At the same time, the approach generates model

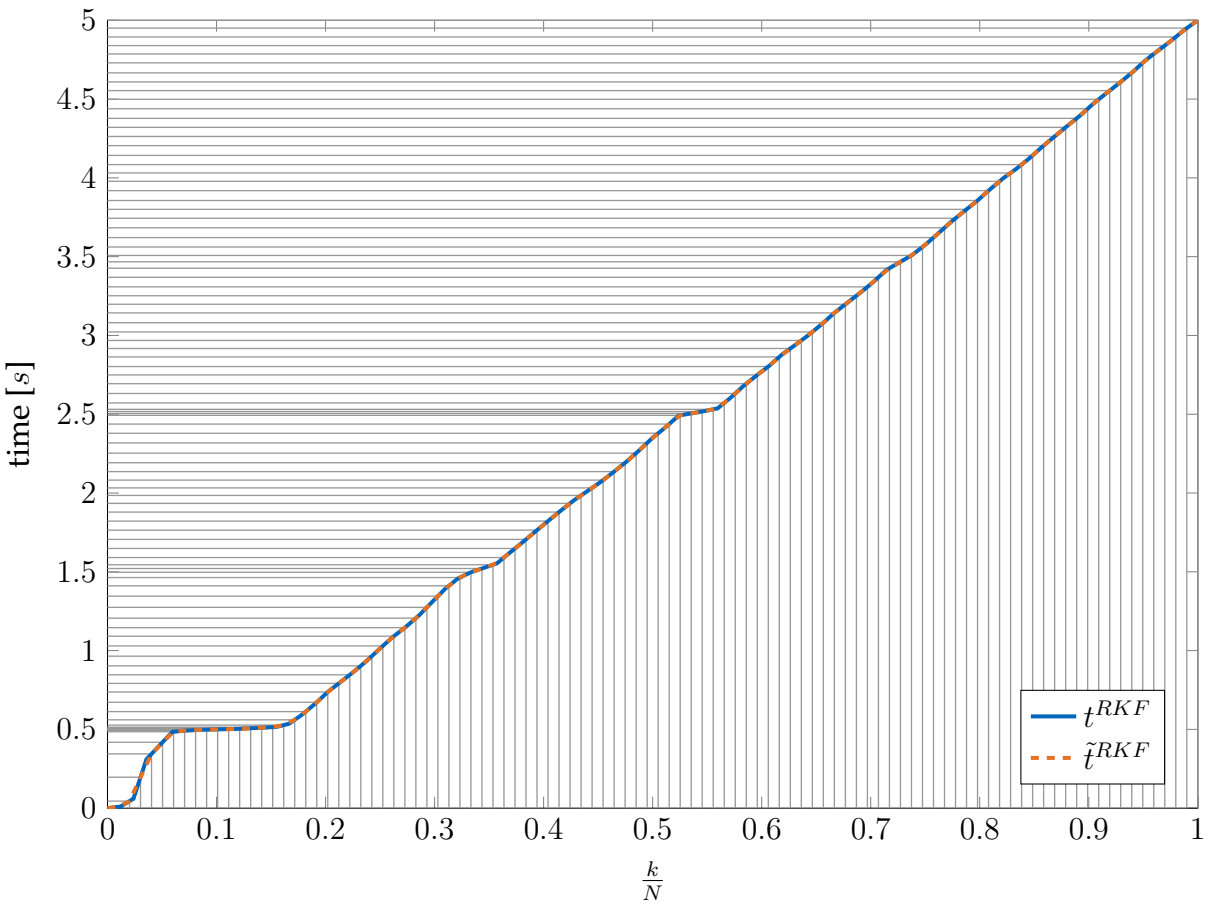


Figure 3.4: *Normalized sample instant over simulation time for variable step size solution (——) and interpolated grid (- - -) for knot generation*

outputs on the complete measurement grid, with only small errors. This is of great benefit especially in the beginning of the identification process, when many possible model formulations have to be considered, their parameters determined, and the results assessed. The interpolation approach, in connection with this novel knot determination scheme, thus improves the applicability of full discretization to parameter estimation problems. This marks another contribution of this thesis.

The price to pay, from an optimization point of view, is that the interpolation scheme links outputs (and thus eventually states) at several time instants together. This is opposed to the basic idea of full-discretization, which tries to *decouple* states at different points in time. If the Hessian of the problem were to be used, it would show a denser structure, with broader "bands" on its main diagonal. However, due to memory limitations, for the problems treated here, the analytic Hessian can only be used in the single shooting cases. For the full-discretization formulation, the solver's internal Hessian approximation scheme is employed, thus this effect is left to the NLP solver implementation to cope with.

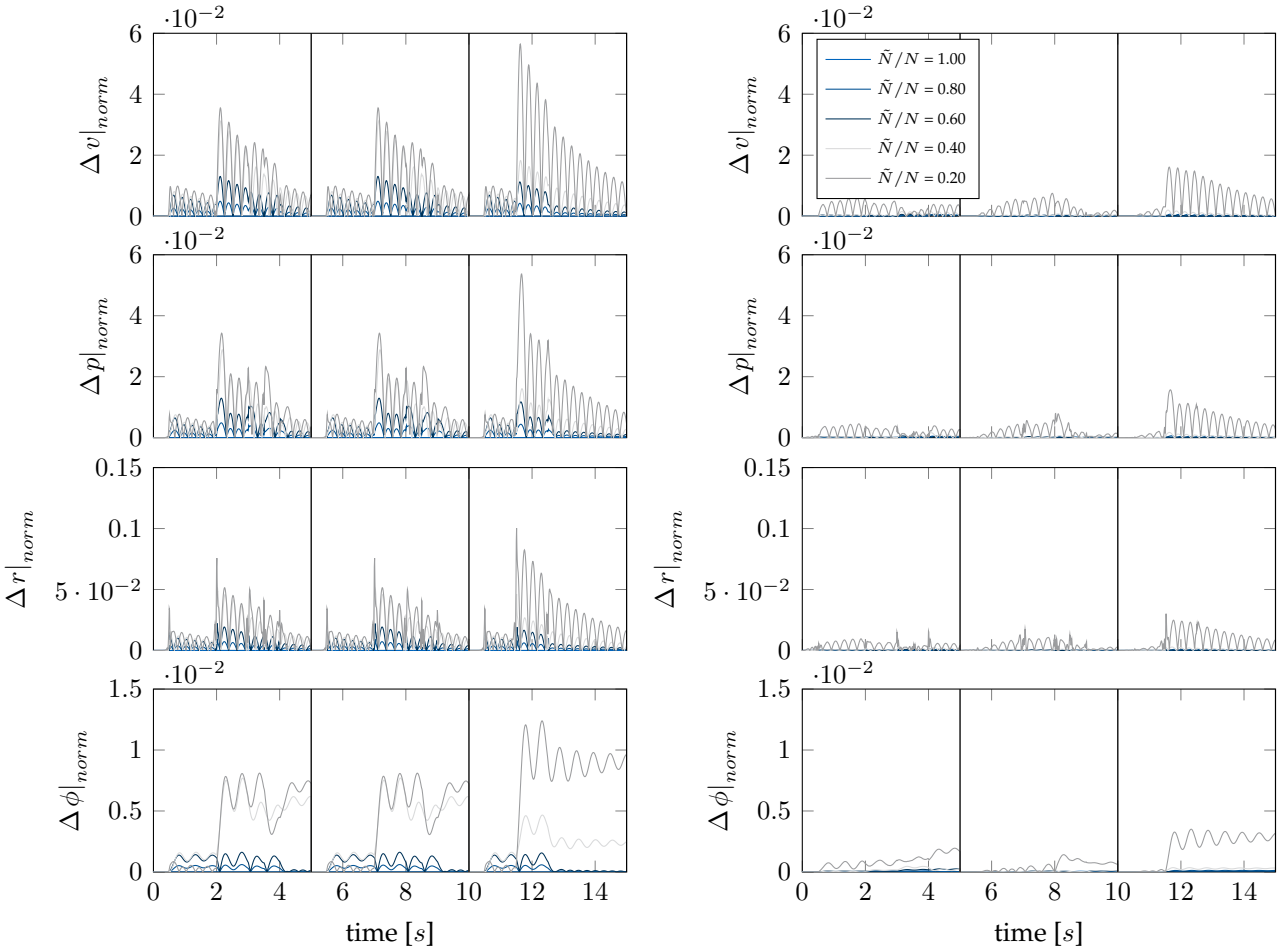


Figure 3.5: *Comparison of interpolation errors compared to 8th order integration scheme with uniform (left) and adapted (right) knot sequence for different reduction factors. Values are normalized with maximum data range* $\left(\max[\hat{\boldsymbol{y}}_k]_{(j)} - \min[\hat{\boldsymbol{y}}_k]_{(j)}\right)$.

3.10 Scaling Considerations

For every numeric optimization, scaling is of utmost importance. If some of the optimization variables exhibit largely different orders of magnitude, convergence is not easily achieved. In addition to scaling states, model parameters and possibly process noise variables (which is directly implemented in FALCON.m [RBG$^+$2018]), the following section presents different scaling approaches for the maximum likelihood problem, which are especially tailored to the ideas presented so far.

3.10.1 Residual Scaling

One very intuitive approach would be to scale the experiment's outputs and measurements, in order to bring them to roughly the same order of magnitude. If, instead of considering the real measurements and outputs, versions that are scaled with a con-

stant, full-rank matrix $\mathbf{D} \in \mathbb{R}^{n_y \times n_y}$ are taken into account, the residuals are

$$ {}_{scaled}\boldsymbol{r}_k = \mathbf{D}\boldsymbol{r}_k = \mathbf{D}(\boldsymbol{z}_k - \boldsymbol{y}_k(\mathbf{z})) = {}_{scaled}\boldsymbol{z}_k - {}_{scaled}\boldsymbol{y}_k(\mathbf{z}) \tag{3.130} $$

In order to be consistent with the statistic considerations leading to the maximum likelihood approach in the first place, the residual covariance matrix needs to be scaled, too.

$$ {}_{scaled}\boldsymbol{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{D}\boldsymbol{\mathcal{B}}\mathbf{D}^\intercal) \tag{3.131} $$

$$ \mathrm{Cov}[{}_{scaled}\boldsymbol{r}_i, {}_{scaled}\boldsymbol{r}_j] = \delta_{ij}\mathbf{D}\boldsymbol{\mathcal{B}}\mathbf{D}^\intercal = \delta_{ij} \cdot {}_{scaled}\boldsymbol{\mathcal{B}} \tag{3.132} $$

The residual part of the cost function may take on different forms, depending on the cost function in use. For standard maximum likelihood eq. (3.2) this is

$$ \begin{aligned} {}_{scaled}J(\mathbf{z}) &= \frac{1}{2}\sum_{k=0}^{\bar{N}} \cdot {}_{scaled}\boldsymbol{r}_k(\mathbf{z})^\intercal \cdot {}_{scaled}\boldsymbol{\mathcal{B}}^{-1} {}_{scaled}\boldsymbol{r}_k(\mathbf{z}) + \frac{N}{2}\ln|{}_{scaled}\boldsymbol{\mathcal{B}}| \\ &= \frac{1}{2}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k(\mathbf{z})^\intercal \mathbf{D}^\intercal(\mathbf{D}\boldsymbol{\mathcal{B}}\mathbf{D}^\intercal)^{-1}\mathbf{D}\boldsymbol{r}_k(\mathbf{z}) + \frac{N}{2}\ln|\mathbf{D}\boldsymbol{\mathcal{B}}\mathbf{D}^\intercal| \\ &= \frac{1}{2}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k(\mathbf{z})^\intercal \boldsymbol{\mathcal{B}}^{-1}\boldsymbol{r}_k(\mathbf{z}) + \frac{N}{2}\ln|\mathbf{D}\boldsymbol{\mathcal{B}}\mathbf{D}^\intercal| \\ &= \frac{1}{2}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k(\mathbf{z})^\intercal \boldsymbol{\mathcal{B}}^{-1}\boldsymbol{r}_k(\mathbf{z}) + \frac{N}{2}\ln|\boldsymbol{\mathcal{B}}| + N\ln|\mathbf{D}| \end{aligned} \tag{3.133} $$

For the other cost function formulations (parameter and covariance estimation eq. (3.8), modified parameter and covariance estimation eq. (3.9)), similar results can be obtained

$$ {}_{scaled}\hat{J}(\mathbf{z}) = \frac{N}{2}\ln\left|\hat{\tilde{\boldsymbol{\mathcal{B}}}}\right| = \frac{N}{2}\ln\left|\mathbf{D}\hat{\boldsymbol{\mathcal{B}}}\mathbf{D}^\intercal\right| = N\ln|\mathbf{D}| + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| \tag{3.134} $$

$$ {}_{scaled}\tilde{J}(\mathbf{z}) = \left|\hat{\tilde{\boldsymbol{\mathcal{B}}}}\right| = \left|\mathbf{D}\hat{\boldsymbol{\mathcal{B}}}\mathbf{D}^\intercal\right| = |\mathbf{D}|^2 \cdot \left|\hat{\boldsymbol{\mathcal{B}}}\right| \tag{3.135} $$

Thus this scaling approach constitutes merely a level shift for J and $\hat{J}$, which does neither influence the first nor the second derivative. For $\tilde{J}$ this approach equals a multiplication with a constant factor. In all of the above cases, the locations of the stationary points of the cost functions do not change, thus the scaling does not influence the estimates.

In summary, using a constant scaling matrix $\mathbf{D}$ for the outputs has no benefit. It can serve to shift the cost function value to more tractable regions, if it had been very small or large before. Additionally, if $\tilde{J}$ is considered, it can be used to alter the numerical values of the gradient and Hessian, which might improve the convergence properties of the optimization algorithm. However, only *one* scalar value is available for tuning the whole problem, which is easily introduced without going through the trouble of scaling residuals.

3.10.2 Direct Covariance Scaling

Another approach is to directly scale the estimated covariance matrix, which is novel in the context of aircraft system identification. This is obviously only meaningful, if it is estimated from the data, using the approach illustrated in section 3.1. It provides a scaling possibility, based on some statistic considerations. Instead of using the maximum likelihood estimate of the covariance matrix, equation (3.6), a version scaled with a full-rank matrix $\mathbf{W} \in \mathbb{R}^{n_y \times n_y}$ is used

$$\hat{\tilde{\boldsymbol{\mathcal{B}}}} = \mathbf{W}\left(\frac{1}{N}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k \boldsymbol{r}_k^\intercal\right)\mathbf{W}^\intercal = \mathbf{W}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^\intercal \tag{3.136}$$

The cost function then becomes

$${}_{scaled}J(\mathbf{z}) = \frac{1}{2}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k(\mathbf{z})^\intercal\, \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k(\mathbf{z}) + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| + N\ln|\mathbf{W}| \tag{3.137}$$

However, in this formulation, when using the expression for the estimated covariance matrix, the first term of the cost function cannot be dropped anymore, merely simplified to arrive at

$$\begin{aligned}
{}_{scaled}J(\mathbf{z}) &= \mathrm{tr}\left[\frac{1}{2}\sum_{k=0}^{\bar{N}} \boldsymbol{r}_k(\mathbf{z})^\intercal\, \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k(\mathbf{z})\right] + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| + N\ln|\mathbf{W}| \\
&= \frac{1}{2}\mathrm{tr}\left[\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}N\frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{r}_k(\mathbf{z})\,\boldsymbol{r}_k(\mathbf{z})^\intercal\right] + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| + N\ln|\mathbf{W}| \\
&= \frac{N}{2}\mathrm{tr}\left[\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\right] + \frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right| + N\ln|\mathbf{W}|
\end{aligned} \tag{3.138}$$

The derivations for the scaled versions of the first and second order derivatives w.r.t. the model outputs can be found in Appendix E. The cost function gradient with respect to the model outputs is

$$\frac{\partial\, {}_{scaled}J}{\partial \boldsymbol{y}_k} = \left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\boldsymbol{r}_k \tag{3.139}$$

For $\mathbf{W} = \mathbf{I}_{n_y}$ above expressions equals the result as was obtained in equation (3.7), and equation (3.16) for the unscaled case. The second order derivatives become

$$\begin{aligned}
\frac{\partial^2\, {}_{scaled}J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\intercal} = \frac{1}{N}\Big(&-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} - \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k \\
&+ \hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k \\
&- \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_l^\intercal\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k \\
&+ \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k \\
&- \hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\Big) \\
&+ \left(\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\delta_{kl}
\end{aligned} \tag{3.140}$$

$$\frac{\partial^2\, {}_{scaled}J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\intercal} \approx \left(\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\delta_{kl} \tag{3.141}$$

Both expressions for first and second order derivatives have been checked using complex step finite differences. The approximation in the second equation is based on the same reasoning as was mentioned in the section on the estimation of the covariance matrix: the first term will shrink with the number of samples N and can thus be neglected. Again, for $\mathbf{W} = \mathbf{I}_{n_y}$, above expressions are the same as obtained before.

Using this formulation, the estimates will differ from the estimates obtained using an unscaled version of the residual covariance estimate. However, the basic problem is still a maximum likelihood estimation problem, merely the way the residual covariance estimate is computed differs.

A scaling of this sort was used in the publication [GH2016] by the author to account for different numbers of samples: parts of the measurement vector $\boldsymbol{z}_k$ originated from a Global Navigation Satellite System (GNSS) receiver, whereas other parts originated from an Inertial Measurement Unit (IMU). Since the IMU had a far higher sampling rate, and the estimation was done on its discretization grid, the missing GNSS values had to be accounted for, which was done by scaling the residual covariance matrix. Here, the consequences of this scaling idea are investigated in more detail, especially their effect on the cost function derivatives.

An alternative solution is due to WALTER, who proposes to assume $\boldsymbol{\mathcal{B}}$ to be diagonal, which then leads to a cost function [WP1997, Example 3.4]

$$J(\mathbf{z}) = \sum_{j=1}^{n_y} \frac{M_j}{2} \ln \sum_{k=0}^{M_j-1} \left([\boldsymbol{y}_k]_{(j)} - [\boldsymbol{z}_k]_{(j)} \right)^2 \tag{3.142}$$

That is, to only use the M_j valid samples per output, and consider all outputs separately.

The novel scaling approach developed here allows for the assumption of a diagonal residual covariance matrix to be dropped, by accounting for missing samples through a scaling of $\boldsymbol{\mathcal{B}}$: First, set the residuals to 0 for invalid measurements

$$[\boldsymbol{r}_k]_{(j)} = \begin{cases} [\boldsymbol{y}_k]_{(j)} - [\boldsymbol{z}_k]_{(j)} & \text{if } [\boldsymbol{z}_k]_{(j)} \text{ is valid} \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, \ldots, n_y \tag{3.143}$$

Then account for the lower number of samples through scaling of the covariance matrix using the number of valid samples per output M_j

$$\mathbf{W} = \begin{bmatrix} \sqrt{\frac{N}{M_1}} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\frac{N}{M_{n_y}}} \end{bmatrix} \tag{3.144}$$

Above corrects the fact that in the maximum likelihood estimate of the covariance matrix (3.6), all elements are divided by the total number of samples N, whereas for the lower frequency signals, only M_j valid samples are available. Further, the square root

caters to the fact that $\mathbf{W}$ is multiplied from the left and from the right. In this way, elements of the covariance matrix related to outputs with $M_i < N$ are scaled up and thus their weighting corrected.

In summary, this novel direct covariance scaling approach offers the possibility to consider an output weighting, based on statistical considerations. Together with the detailed inspection of the consequences regarding cost function derivatives, this constitutes another of the main contributions of this thesis. The idea is especially appealing when simultaneously considering measurement signals of different sampling rate. However, it may also be applied if more weight is to be put on a specific output for other reasons. Additionally, with this scaling approach all outputs may be influenced independently of each other, where the pure residual scaling of the last section is reduced to a mere scalar weighting.

[...] our knowledge can be only finite, while our ignorance must necessarily be infinite.

Karl Raimund Popper, 1963 [Pop2002]

4

Application of Optimization to Parameter Estimation Problems

From the foregoing chapters it is obvious, that there exist many ways to actually formulate a parameter estimation problem. In this part, the four main formulations of this work will be discussed:

1. **Case I:** single shooting formulation for deterministic systems
   The optimization vector consists of the model parameters and initial conditions. This constitutes the standard case as is widely used in the literature
2. **Case II:** full-discretization formulation for deterministic systems
   The optimization vector consists of model parameters and all state values at all integration time steps. This will be the main formulation used in this thesis
3. **Case III:** single shooting formulation for stochastic systems using an iterative state estimation algorithm
   The optimization vector again only consists of model parameters and initial conditions. This formulation (and its extensions to also estimate noise covariances) is known as "Filter Error Method (FEM)" and is commonly used in the literature if a stochastic treatment of the system cannot be avoided.
4. **Case IV:** full-discretization formulation for stochastic systems based on Bayesian estimation
   The optimization vector consists of model parameters, as well as state values and process noise inputs at all integration time steps. This formulation is the extension of the full-discretization approach to stochastic systems, using an explicit process noise input.

Table 4.1: *Possible formulations of the parameter estimation problem from an optimal control point of view.*

case	process noise	measurement noise	residuals	state integration	optimization variables $\mathbf{z}$[1]	constraints
case I sec. 4.1	*not considered*	exclusive noise source	measurement noise; predefine or estimate covariance	single shooting	$\boldsymbol{\theta}$, $\boldsymbol{x}_0$	possibly box constraints $\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub}$
case II sec. 4.2	*not considered*	exclusive noise source	measurement noise; predefine or estimate covariance	full discretization	$\boldsymbol{\theta}$, $\boldsymbol{x}_k$ $k = 0 \dots \bar{N}$	possibly box constraints $\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub}$ integration defects $\boldsymbol{\zeta}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{\theta}) = \mathbf{0}$ possibly path constraints $\boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0}; \quad \boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0}$
case III sec. 4.3	predefine covariance	predefine covariance	state estimation residuals; predefine or estimate covariance	single shooting (using iterative state estimation)	$\boldsymbol{\theta}$, $\boldsymbol{x}_0$	possibly box constraints $\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub}$
case IV sec. 4.4	predefine covariance	predefine or estimate covariance	*not considered*	full discretization (with explicit process noise terms $\boldsymbol{w}_k$)	$\boldsymbol{\theta}$, $\boldsymbol{x}_k$, $\boldsymbol{w}_k$ $k = 0 \dots \bar{N}$	possibly box constraints $\mathbf{z}_{lb} \leq \mathbf{z} \leq \mathbf{z}_{ub}$ integration defects $\boldsymbol{\zeta}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \boldsymbol{w}_k, \boldsymbol{w}_{k+1}\boldsymbol{\theta}) = \mathbf{0}$ possibly path constraints $\boldsymbol{c}_{eq}(\mathbf{z}) = \mathbf{0}; \quad \boldsymbol{c}_{ineq}(\mathbf{z}) \leq \mathbf{0}$

[1] If states $\boldsymbol{x}$ or process noise $\boldsymbol{w}$ are part of the optimization vector, it is inherently assumed that they appear once per maneuver

Table 4.1 sums up the main characteristics of the four possible formulations, which were implemented for this work. The first three columns contain information about the treatment of noise sources within the problem. Those denoted "process noise" / "measurement noise" indicate, how the respective noise sources are treated and which information about them is necessary, i.e. if their covariance needs to be pre-specified, or may be estimated. The next column "residuals" indicates if the residuals play a major role in the problem formulation, and what they represent in the respective case. The last three columns sum up information about the optimal control aspects, i.e. which transcription method is employed, what the contents of the optimization vector are, and which type of constraints may be incorporated.

Case I and case III constitute the standard approaches commonly discussed in the literature for deterministic and stochastic systems respectively. The formulations of case II and case IV are two of the main contributions of this thesis.

Even though those four problem formulations are fundamentally different, some similarities exist

Fixed Inputs In all cases, the measured controls $\boldsymbol{u}$ are fixed and thus not subject to optimization, which is probably the largest difference compared to classical optimal control. Merely in the case of state and parameter estimation using full discretization, process noise inputs $\boldsymbol{w}$ may appear in the optimization vector.

Treatment of Maneuvers In the application of parameter estimation for aircraft, usually several maneuvers are considered simultaneously. They are typically designed to excite different aspects of the aircraft's dynamic characteristics. From the optimal control point of view, these maneuvers are treated as different, disconnected "phases" of the problem.

Cost Function Some different cost function formulations have been discussed, which are summed up in table 2.1. Further, their derivatives can be computed analytically (gradient) or approximated very well (Hessian), using analytic or finite-difference approximations of the output sensitivities [WP1997, Ch. 4]. See sections 3.2 and 3.3 for details. The different parts of the cost functions (measurements, prior information) may be combined with the formulations of table 4.1 rather freely, with the exception of the process noise term, which only makes sense in case IV.

Problem Convexity The arising problem is in general non-convex, i.e. local minima may exist that do not represent the best possible parameter estimates. This problem can be alleviated, if good initial guesses can be produced (see section 3.8), and if suitable experimental conditions were chosen. The latter need to show the effects of the parameters sufficiently well (see section 5.1) [WP1997, Ch. 4].

Consideration of Parameter Bounds The statistical considerations that may lead to the necessity of including model parameter bounds in the estimation problem were discussed in section 3.4. There, the necessary adjustments of the Fisher information matrix for constrained results are also shown.

From an optimization point of view, including equality and inequality constraints on parameters along with other optimization variables is straight forward, since the model parameters $\boldsymbol{\theta}$ usually form a part of the optimization vector $\mathbf{z}$.

4.1 Case I: Single Shooting with Deterministic Dynamic System

One of the two classical approaches, as illustrated in detail in the standard literature on the topic [MK2016, Jat2006], is summed up in the first row "case I" in table 4.1:

It considers a pure output-error formulation, based on a deterministic system, i.e. treatment of process noise is not possible. This makes the residuals equal to the measurement errors, for which a Gaussian distribution is assumed.

The optimal control problem is then transcribed into a NLP problem by using a single shooting scheme, where the only free optimization parameters are the model parameters $\boldsymbol{\theta}$, and possibly the initial conditions $\boldsymbol{x}_0$ of the maneuvers under consideration.

All aspects necessary for the solution of this problem have been discussed already

- If measurements of the states are directly available (or may be reconstructed), an efficient approach to improve upon initial model parameter guesses may be employed, see section 3.8. In many of the examples to be presented in chapter 6, the necessity to specify initial model parameter guesses $\boldsymbol{\theta}_0$ could be circumvented using this approach.
- At the i-th iteration, using the current model parameter value $\boldsymbol{\theta}_i$ and initial value $\boldsymbol{x}_{0i}$, the model outputs may be computed by solving the system equations forward in time. To this end, one of the integration method's mentioned in appendix D.3 may be used.
- This integration may be performed on a coarse grid, which is then interpolated onto the measurement grid using the approach of section 3.9. This may help to save time during the trial an error phase when determining a suitable model structure.
- With the outputs $\boldsymbol{y}_k$ available, an estimate of the residual covariance matrix $\hat{\boldsymbol{\mathcal{B}}}$ may be computed according to section 3.1; alternatively the residual covariance may be fixed a priori.
- If so desired, the estimated covariance matrix may be scaled, using the approach illustrated in section 3.10.2 to e.g. incorporate different sampling rates.

- Cost function derivatives w.r.t. model outputs $\frac{\partial J}{\partial \boldsymbol{y}_k}$ are computed using the derivatives of section 3.2
- The cost function gradient $\frac{\partial J}{\partial \mathbf{z}}$ and Hessian $\frac{\partial^2 J}{\partial \mathbf{z}^2}$ are eventually determined with the aid of the output sensitivities ${}^{y}_{\mathbf{z}}\mathbf{S}_k$ (section 3.3). They may be computed numerically from finite difference approximations, or analytically by solving the sensitivity differential equations
- With the gradient and Hessian information, the solution algorithms for unconstrained optimization, as illustrated in section 2.1.2, may be used to determine a parameter update $\mathbf{z}_{i+1} = \Delta\mathbf{z}_i + \mathbf{z}_i$
- In the implementation that was developed for this work, the consideration of simple box constraints on the optimization parameters $\mathbf{z}$ is possible by employing the active set strategy of section 2.1.3.4. More complicated constraints cannot be enforced using the implementation at hand.
- In a post-processing step, parameter uncertainties are computed based on the cost function Hessian, which involves the residual covariance matrix and output sensitivities w.r.t. model parameters and initial guesses, see sections 2.2.5 and 3.2.
- These parameter uncertainties may then be used as illustrated in section 3.5.5. There, a scheme was derived to use state and output sensitivities to propagate model parameter and initial value uncertainties in time, to eventually obtain state and output covariances at all sampling instants.

This procedure is often used iteratively with different model structures, to eventually find that, which best describes the data.

4.2 Case II: Full Discretization with Deterministic Dynamic System

This approach, as summed up in the row "case II" of table 4.1, is based on the classical approach as illustrated in the last section 4.1: the considered output error formulation, with a purely deterministic system, and the possible cost functions are the same. However, the transcription method that is used to obtain a NLP problem relies on full discretization rather than single or multiple shooting as was done in [BM2009, BM2009].

The most noteworthy steps in the solution are

- The improvement step for initial model parameter guesses of section 3.8 is independent of the transcription method and may thus be employed in this case as well.
- From the current value of the optimization vector $\mathbf{z}$, model parameters $\boldsymbol{\theta}$ as well as state values $\boldsymbol{x}_k$ are extracted; Using these, output values $\boldsymbol{y}_k$ as well as constraint values (integration defects, path constraints, box constraints etc.) may be computed.

- The optimization framework FALCON.m is able to compute most of the relevant derivatives automatically, and if desired analytically. It provides Jacobians of the model and output equation and, together with the integration scheme in use, it is able to assemble them to obtain the constraint Jacobian to be provided to the NLP solver.
- FALCON.m is also able to compute simple, analytic cost function derivatives automatically. However, the formulations used here are too complex, which is why cost function derivatives w.r.t. model outputs are computed manually using the results of section 3.2. FALCON.m then takes care of the application of the chain rule in order to map those to derivatives w.r.t the original optimization vector, which is then supplied to the NLP solver.
- The states (and thus also the corresponding outputs) in the optimization vector may be located on a rather coarse time grid. Using the interpolation methodology of section 3.9, they may be interpolated onto the measurement time grid to obtain residuals and eventually evaluate the cost function and its derivatives. In contrast to the approach of the foregoing section 4.1, the state values on the coarse grid are now directly incorporated in the optimization vector, rather than being numerically integrated on this coarse grid.
- Covariance estimation (section 3.1) and scaling (section 3.10.2) may be performed exactly as in case I
- FALCON.m then passes the discretized problem, together with the cost function gradient, and constraint Jacobian to the underlying NLP solver which takes care of the actual solution of the optimization problem.
- Optimization parameter uncertainty now needs to be computed while taking constraints into account. The methodology to do so, has been introduced in section 3.5.

This changes the characteristics of the problem compared to the single shooting transcription:

- Here, the size of the optimization vector $\mathbf{z}$ is comparable to that of the number of integration steps times the state dimension; However, the problem Jacobian is extremely sparse, which keeps the overall task tractable.
- Most of the optimization parameter values are constrained via the integration defects, so the number of remaining degrees of freedom of the problem is small (in the same order of magnitude as in the single shooting case)
- constrained optimization is necessary, in order to take care of integration defects and possibly further constraints. Here, off-the-shelf NLP solvers (e.g. Interior Point Optimizer (IPOPT), Sparse Nonlinear Optimizer (SNOPT)) are to be favored over custom implementations due to the maturity of those algorithms compared to anything an analyst might develop himself [WP1997].

- Constraints on model parameters can be easily included as constraints on the optimization vector $\mathbf{z}$. Still this should be done with the same care as always, see section 3.4 for a discussion.
- Furthermore, path constraints on outputs or (possibly non-linear) functions of states and inputs may be incorporated, both in the estimation as well as the uncertainty quantification.
- Since explicit integration of the system equations is avoided, also the sensitivity equations do not need to be solved. The information they contain is directly incorporated via the integration defects and Jacobian matrices.

Using full discretization may increase the radius of convergence of a problem considerably, compared to the standard, single shooting approach: For bad initial model parameter guesses, the single shooting approach can drift significantly during the time span under investigation. This may prohibit a meaningful convergence if the difference between measured and propagated outputs is too large. This issue is alleviated when using full discretization: using every propagation time step as "node" stabilizes the procedure numerically, and offers more freedom to the solver to keep the solution closer to the measured values. BOCK discusses this aspect for a multiple shooting approach [Boc1987, Ch. 2], which can readily be extended to full discretization.

Another advantage of the full discretization formulation is that it is directly applicable to unstable models. The standard, single shooting approach may very well lead to numerical problems and fail to converge if the system under consideration is unstable, which is why several artificial stabilization approaches are illustrated in [Jat2015, Ch. 9]. The full discretization approach does not suffer from this drawback, and meaningful results may be obtained, as is illustrated in the examples of chapter 6.1.

Lastly, the direct inclusion of path and optimization vector constraints offer some advantages over the standard methods in use: Using constraints, the state trajectory may be forced to lie in physically reasonable regimes, where the true solution is known to be located. The NLP solver is thus forced to look for model parameters only in meaningful regions. This type of constraint on outputs or states should typically only be active during the earlier iterations, when model parameter values are still far from their optimum. Thus, the consequences on parameter uncertainty will in general not need to be considered at the final solution.

However, the approaches illustrated here are also capable of incorporating the effects of active path constraints at the optimum. This may for example be necessary, if unit quaternions are used to parameterize an aircraft's attitude: they only yield a valid attitude representation if they are of unit length. This needs to be explicitly enforced, both in the estimation as well as uncertainty quantification process. So far, approximate methods using feedback or normalization have been used, however, now this constraint can directly be considered.

These advantages are bought at the expense of larger problems, and possibly longer

computation times. However, the very efficient model evaluation algorithms within FALCON.m, together with the interpolation approach of section 3.9, make it possible to obtain meaningful results within an acceptable amount of time, even for rather large problems.

4.3 Case III: Single Shooting with Stochastic Dynamic System

The formulation of case I is definitely the most widely used one for parameter estimation in the aircraft context. Nevertheless, sometimes process noise influences cannot be ignored and have to be considered explicitly in the problem formulation, which leads to joint parameter and state estimation problems.

Again, there exists a classical approach, which is summed up in the row "case III" of table 4.1. Here, the arising problem is solved by applying an iterative state estimation algorithm in order to obtain a solution to the stochastic system. The model parameters and possibly initial values are then estimated using similar ideas as in "case I". The problem is thus transcribed using a single shooting formulation, based on a stochastic filter.

This is also an output-error formulation: in the definition of the probability density to maximize, the only random component to be considered are the residuals

$$\boldsymbol{r}_{k|k-1} = \boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{4.1}$$

which are derived in equation (2.219) of section 2.3.2.4 on state estimation. The output error approach is possible, since a Kalman filter for a linear, discrete time, dynamic system yields *white* residuals $\boldsymbol{r}_{k|k-1}$. Then the derivations of section 2.2.2.2 can also be applied to stochastic systems treated with a Kalman filter, resulting in the same cost function formulations as before. The difference is that the random component in the residuals combines influences due to measurement *and* process noise. As was noted already in section 2.2.2.4, the use of smoothing algorithms for state estimation is questionable from a theoretic point of view, since the residuals may not be white anymore. However, for specific cases, algorithms like the Rauch-Tung-Striebel (RTS) smoother may provide superior results despite this theoretic flaw.

The same is true for non-linear state estimation involving Extended Kalman Filter (EKF) or Extended Rauch-Tung-Striebel (ERTS) smoother algorithms: then it cannot be guaranteed, that the residuals are white. Nevertheless, good results have been obtained in this case, too [GH2016], which justifies the application of the output error cost functions. The quadratic nature of the cost function seems to enable good results, even if some of the statistical assumptions are violated.

Again, most of the necessary steps have already been discussed, the most noteworthy aspects now are

- At each iteration, model parameters $\boldsymbol{\theta}$ and initial values $\boldsymbol{x}_0$ are extracted from the optimization vector. Model outputs are then computed using an iterative state estimation approach using these initial conditions and model parameters. In this work, only situations are considered, where the process and measurement noise covariance matrices are defined a priori, and possibly tuned manually for good results.
- Application of the interpolation approach of section 3.9 is possible. However, then the measurements have to be interpolated onto the coarser grid for the iterative state estimation step. This is necessary, since the minimum allowable time step width between two measurements is the integration step width, i.e. if there is more than one measurement per integration step, the state estimation algorithms as presented here are not able to handle these.

  This is a considerable drawback, since interpolation of measurement data (involving measurement noise!) has a smoothing effect [MK2016], which thus alters the measurement statistics for the state estimation. Care has to be taken, to eventually verify the results with all samples considered.
- As has been mentioned, the residuals now contain influences both of process and measurement noise. This leads to a situation, where it is more complex to predefine the residual covariance matrix. Thus, the possibility to estimate it from the data (see section 3.1) is quite useful.

  It would be possible to extract the residual covariance information from the internal workings of the state estimation algorithm via equations similar to eq. (2.221)

$$\mathrm{Cov}\left[\boldsymbol{r}_{k|k-1}\right] = \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^\intercal$$

  However, the additional complexity in the implementation at hand was circumvented by approximating $\mathrm{Cov}\left[\boldsymbol{r}_{k|k-1}\right]$ via an average residual covariance for all samples, either pre-defined or estimated from the data.
- If the residual covariance is estimated, it may then be scaled, using the approach illustrated in section 3.10.2
- Since the cost function in use is the same as before, its derivatives w.r.t. model outputs are computed using the derivatives of section 3.2.
- To eventually obtain the cost function gradient and Hessian w.r.t. the optimization parameters the output sensitivities (section 3.3) remain to be determined. Since state estimation already involves first order Jacobians of the system and output equation, analytically solving the sensitivity equations of the filter solution would necessitate second order derivatives of the system and output equations. Furthermore, the sensitivities of the state error covariance needs to be con-

sidered, too. These facts discourage from an analytical solution, which is why the output sensitivities for this case are computed using numeric finite differences.

- Using Hessian and gradient information, the same unconstrained optimization algorithms (section 2.1.2) and active set strategy for box constraints (2.1.3.4) as for case I may be used to actually solve the problem
- Optimization vector covariances may be computed using the same approach as in case I, i.e. using the resulting output sensitivities and the estimated residual covariance matrix in the approximation of the Fisher information matrix according to sections 2.2.5 and 3.2
- State covariance matrices are directly supplied by the state estimation algorithm, see sections 2.3 and the discussion in the end of section 3.5.5 for details.

Here, it is assumed that both process, as well as measurement noise covariances are pre-defined. There exist approaches, that estimate the involved noise characteristics along with the model parameters, see e.g. [Jat2006, Ch. 5]. However, the restrictions this imposes (only steady-state, linear Kalman filters can be easily treated; for non-linear approaches the complexity grows fast) and the necessary, heuristic corrections (pseudo-constraint optimization for Kalman Gain; F-correction for revised residual covariance matrix [Jat2006, Ch. 5.IV.B]) discourage from their application.

4.4 Case IV: Full Discretization with Stochastic Dynamic System

The last case to be treated is summed up in the row "case IV" of table 4.1. It considers the problem of combined state and parameter estimation in a full-discretization setting rather than employing an explicit state estimation algorithm. To transcribe the estimation into an NLP problem, the process noise terms are explicitly introduced as optimization variables, resulting in the largest, but most general problem formulation to be discussed here.

The price to pay for this very versatile formulation is an increased optimization parameter vector $\mathbf{z}$, increased complexity in the integration defects $\boldsymbol{\zeta}_k(\dots) = \mathbf{0}$, and additional terms in the cost function, which punish too much process noise activity. The advantages of this formulation from a theoretical point of view, is that it shows interesting connections between state estimation, parameter estimation, and optimal control approaches. It is also closely related to the idea of "inverse simulation" (see section 5.5.3 for details), i.e. having a feedback controller track the measurements with the final parameter estimates and evaluating the control effort. Here, this may be achieved by explicitly trying to minimize additional control inputs ($\hat{=}$ process noise) when trying to follow the measurements, rather than by constructing a feedback loop.

A practical advantage is, that since all information about the state trajectory is now contained in the optimization vector, very creative constraints may be formulated, e.g. to enforce a passive system (e.g. dissipating energy), or to enforce certain characteristics only on parts of the trajectory (e.g. ground contact).

In the light of the foregoing discussion, there are a few noteworthy aspects

- At each iteration, after extracting states $\boldsymbol{x}_k$, model parameters $\boldsymbol{\theta}$, and process noise terms $\boldsymbol{w}$ from the current optimization vector $\mathbf{z}$, output values $\boldsymbol{y}_k$ as well as constraints (integration defects, path constraints, box constraints etc.) may be computed.

  Again, FALCON.m takes care of the Jacobian matrices, applying the chain rule and eventually providing cost function and constraint values as well as their analytic derivatives to the NLP solver in order to have accurate derivative information.
- Cost function derivatives w.r.t. model outputs are computed using the derivatives of section 3.2; derivatives w.r.t. process noise terms are simple due to the quadratic nature of the corresponding cost function term. Again, this is necessary since the automatic derivative computation is not able to cope with large vector matrix computations in the cost function. FALCON.m then takes care of mapping the derivatives correctly to eventually obtain those w.r.t. the optimization parameters.
- Application of the interpolation methodology of section 3.9, is straight-forward.
- Covariance estimation (section 3.1) and scaling (section 3.10.2) may be performed for the *measurement* covariance matrix. However, estimation of the process noise covariance using the methods of section 3.1 is bound to fail, since the NLP solver will minimize its determinant by making two process noise signals perfectly correlated.
- If only parts of the process noise vector are to be considered (e.g. because some states can be considered as deterministic), the use of singular process noise matrices is included in the implementation. Then, to compute the weighed process noise contribution to the cost function, only the full-rank parts of $\mathbf{Q}$ are inverted and the remainder is set to zero. The corresponding process noise terms are fixed at zero and removed from the problem, thus reducing its size.
- Further, process noise inputs may be introduced on a third grid (in addition to the measurement and state time grids), which may be even coarser than the state time grid. The choice of grid points follows similar ideas as illustrated in section 3.9, i.e. the density of the state time grid is used to determine process noise grid points. Thus, whenever the integration step size is smaller, also more process noise variability is possible.

  To evaluate model outputs and constraints, $\boldsymbol{w}$ is then interpolated onto the state

time grid by FALCON.m [RBG+2018]. This approach may help to further reduce the problem size. However, it also effectively limits the bandwidth of the process noise inputs, which may be undesirable. If lower process noise bandwidth can be tolerated, reducing the number of process noise samples has another advantage: the same amount of "information" will always be used, resulting in a better "information per sample" ratio, which may help to make the problem more convex, and thus easier to solve.

- The computation of the optimization vector covariance, taking the effect of constraints into account, is detailed in section 3.5.3 and 3.5.4

The main characteristics are the same as in "case II" of section 4.2, with the obvious extensions where process noise is now explicitly involved. Also, the basic statistical considerations are very similar to what is done in case III, as was illustrated in chapter 2.3 on state estimation: now, instead of iteratively maximizing local likelihoods ("maximum likelihood filtering" as illustrated in section 2.3.3.1 and [RTS1965]), a global maximum likelihood estimate for all states at once ("joint maximum likelihood estimation (JMLE)" as shown in section 2.3.3.3 and [RTS1965]) is computed.

A difference is, that measurement and process noise appear explicitly in the cost function formulation. Thus there is no need to consider residuals, combining their effect.

The downside of the approach is the considerably larger problem size. However, carefully evaluating, which process noise inputs are really necessary, together with the possibility to reduce the number of process noise samples, may help to keep the size of the optimization vector tractable. Thus the approach may not be suited for pure state estimation with large state and process noise vectors over long time spans. Rather, it offers additional freedom in the formulation of parameter estimation problems, where including some few process noise terms is beneficial.

Also, having explicit values for the $\boldsymbol{w}$ at ones disposal opens new possibilities for future work: their whiteness may be explicitly enforced by punishing correlation in time in the cost function; alternatively inspection of the spectral characteristics of $\boldsymbol{w}$ may give hints on model deficiencies;

Part III

Applied Aircraft System Identification

System Identification is the "... determination, on the basis of observation of input and output, of a system within a specified class of systems to which the system under test is equivalent;"

Lotfi Aliasker Zadeh, 1962 [Zad1962]

5

The Aircraft System Identification Process

The process of system identification here, is split into five steps, which will be illustrated in the following sections. Their relationship is illustrated in figure 5.1, which is loosely based on [Jat2015, Fig. 1.5]. Although originally discussed for aircraft system identification for large commercial or military applications, they are also valid for the Remotely Piloted Aerial System (RPAS) cases to be the main focus of this work.

The initial activities are related to the design and execution of experiments (chapter 5.1). In system identification for aircraft, this is usually the most expensive step, since flight testing is a rather complex procedure. It necessitates various specialists, the aircraft, operating resources, and, if available, special flight testing equipment. Thus the experiments should be carefully planned in order to minimize the number of necessary iterations at this stage. Although flight testing for RPAS tends to be cheaper, most of the aspects of experiment design are still important.

After the experiments had been conducted and data gathered, the time consuming step of data pre-processing follows, where a major activity is Flight Path Reconstruction (FPR) (chapter 5.2). Here, data-logs have to be time-synchronized, screened, and the actual maneuver sequences extracted. A first, rough estimate of the data quality may be obtained in order to provide immediate feedback, if some tests need to be repeated. If non-negligible disturbances were acting on the aircraft (wind or turbulence), it is possible to obtain estimates for them, together with estimates for possible sensor errors, based on kinematic relationships.

Next, a mathematical model needs to be set up, that describes the aircraft's flight dynamics and is compatible with the estimation algorithms (chapter 5.3). Then the ac-

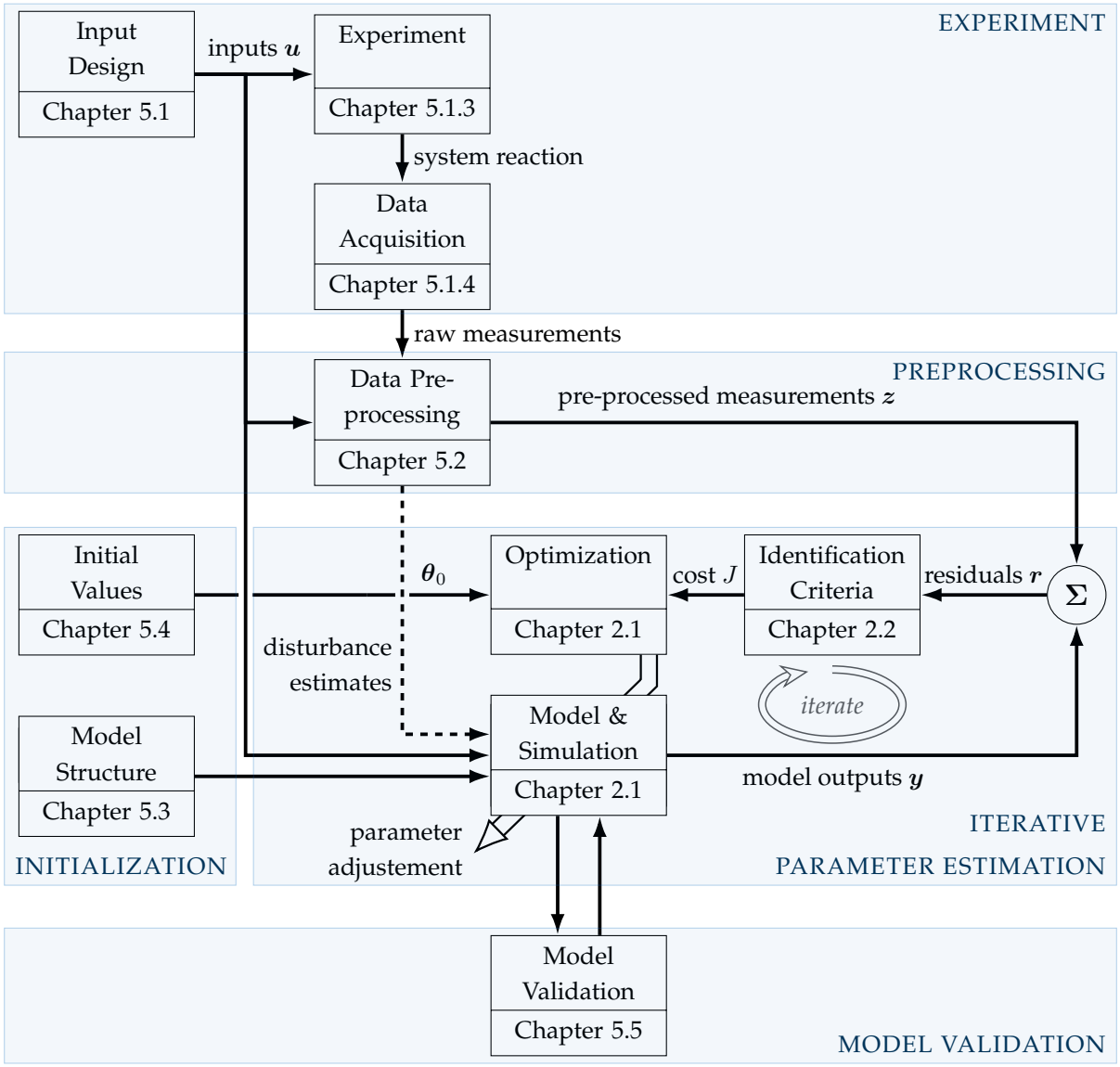


Figure 5.1: *Work flow in system identification; links to the chapters are given within the respective boxes*

tual parameter estimation (chapter 5.4) takes place, by choosing a model structure, choosing an identification criterion, and an optimization algorithm. These components, together with an initial guess for the optimization parameters, are eventually used to obtain the actual parameter estimates.

The quality of the results is assessed in the model validation step (chapter 5.5). Hardly ever is the analyst able to get it right the first time, which leads to many iterations of the validation and parameter estimation step using different model structures, initial guesses and identification criteria.

Each of these steps will be illustrated in the following chapters. However, since the focus of this work is on parameter estimation techniques, the discussion will be quite general, but references for further reading are given.

5.1 Experiment

A basic truism in system identification is, that "the identified model can only capture behavior that is exhibited by the system and embodied in the measured data" [MK2016, p. 324], or as WALTER puts it: "... a badly designed experiment may ruin any attempt at analyzing the data collected from the system" [WP1997, p. 285]. Although modern post-processing and parameter estimation algorithms are very capable at extracting reasonable results from seemingly unusable data, it is always preferable to have good quality data as a basis for further analysis.

For aircraft system identification this means that specific flight tests are inevitable, since data collected under normal operation conditions is in general not informative enough. The overall goal in experiment design is thus to increase the information content in the data, subject to practical constraints, such as safety, operational or financial limits. Data quality is one of the major limiting factors of the scope and accuracy of model development and parameter estimation [Jat2015, Ch. 2], which is why time spent on its improvement is usually well invested.

This translates to three distinct steps [MK2016, Ch. 9]:

1. define the flight condition and aircraft configuration
2. define the inputs for the identification maneuvers
3. specify the instrumentation and data acquisition system

During these steps, practical constraints have to be considered [MK2016, Ch. 9]

- input/output amplitudes are commonly limited. On the one hand to ensure validity of model structures, on the other hand for safety reasons.
- hardware limits are present in the form of sensor range/resolution, bandwidth for data storage, and sensor availability
- limited experimentation time
- regulatory aspects, e.g. for RPAS operations currently beyond visual line of sight (BVLOS) operation is strictly regulated, so is the flight above certain altitudes above ground

5.1.1 Test Condition

In aircraft system identification applications, a test point is usually defined as one point in the envelope, i.e. commonly a wing-level trim point with a pre-defined velocity (Mach or airspeed) and altitude, together with a specific aircraft configuration (center of gravity (c.g.) position, settings of high-lift devices) [Jat2015, Ch. 2]. Typically, local models at these envelope points are used to approximate the global functional dependencies of the aerodynamic forces and moments in the form of multivariate Taylor expansions in the states and controls [MK2016, Ch. 3].

The number of possible test points grows quickly with several speeds, altitudes and configurations. However, in RPAS applications it is limited by current legislation: if no explicit provisions are taken, flight activities are limited to several tens to some hundred meter above ground, which essentially reduces the test point selection to an airspeed selection, and the definition of aircraft configurations. Also, some combinations may be disregarded, since they do not arise in normal operations: If high lift devices are available, they will usually not be used at cruise speed, it is thus often enough to consider them for approach velocities.

5.1.2 Input Design

To complete a test point definition, the actual maneuver to be performed needs to be chosen. It has to be defined which control effector to use, how long the maneuver is to last, and which input form to use [MK2016, Ch. 9]. These choices all have one fundamental idea at heart: the aircraft's pertinent modes are to be excited independently and sufficiently; whenever one particular mode is treated, excitement of the others should be minimized as far as possible [Jat2015, Ch. 2].

Two approaches have to be distinguished when designing experiments for aircraft system identification: the case without an a priori model, and the case where some prior knowledge is available. The former is rather uncommon, since in the design phase of the aircraft usually some rough estimates of its control characteristics can be computed.

If some prior knowledge about the model is available, a more mathematically exact description of "data information content" may be achieved via the Fisher information matrix $\mathcal{F}(\boldsymbol{\theta}_{prior})$. For (asymptotically) efficient estimators, the latter is directly related to the obtainable parameter uncertainties via the Cramér-Rao inequality, see the discussion on statistical properties of estimates in section 2.2.5. It presents a theoretical lower limit on the parameter covariance for *any* estimator, and can thus be examined for the merit of an input sequence independent of the algorithm to be used [MK2016, Ch. 9] [Jat2015, Ch. 2]. The downside of the use of the Fisher information matrix is its dependency on prior parameter values and noise covariance definition: In order to generate "good" input signals for parameter estimation, parameter estimates need to be available [Jat2015, Ch. 2].

Further, certain boundaries must not be violated during maneuver execution: Inputs are usually constraint by mechanical or software limitations, whereas output limits are based on safety considerations (over-speed, acceleration limits,...) and model validity arguments (small perturbations for linear modeling) [MK2016, Ch. 9]. If automated systems are used, actuator bandwidth also puts limits on the realizable input signals for high frequencies.

Lastly, correlation between influencing quantities should be as small as possible, to

avoid the problem of data-collinearity [MK2016, Ch. 5]. Data-collinearity in general describes the problem of two or more influencing signals being too similar, so that the estimation algorithm has difficulties attributing an effect uniquely to one or the other. The result would be a close-to-singular Hessian matrix, as was discussed in section 3.6. This may happen due to an active automatic control system that e.g. moves control surfaces proportionally to body angular rates, or may be inherent to flight mechanics (control surface deflections usually lead to body angular rates of quite similar signal shape). Nevertheless, inputs should be designed such, that influencing quantities appear as independently as possible: by moving control surfaces in different manners; by exciting them one by one, where feasible; by turning an automatic control system off where possible, or reduce gains where it cannot be fully deactivated. All these aspects are true both for identification of large commercial or military aircraft and for RPAS projects.

5.1.2.1 Optimal Inputs

If exact, mathematical maximization of the input signal's information content is sought, some scalar function the Fisher information matrix $\Psi(\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}_{prior}))$ is commonly used. Possible choices are based on norm-like expression involving the trace or determinant of $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}_{prior})$ [WP1997, Ch. 6.1]. The different possible cost functions then have geometric interpretations: some minimize the volume of the confidence ellipsoid, defined via the corresponding parameter covariances; some minimize the sum of squares of its semi-major axes; others focus on minimizing the largest semi-major axis [WP1997, Ch. 6.1] [Jat2015, Ch. 2].

Depending on the model formulation and optimality criterion chosen, the arising optimal control problem may be solved semi-analytically [Meh1974], via Dynamic Programming [Mor1990], full discretization using the direct method of optimal control [HGH2019], or any other fitting approach.

However, for all cases the approach relies heavily on the prior parameter values, as was mentioned before. Ideas exist to alleviate this by considering the complete prior distribution and using expectations of the cost function or minimax-optimality, which however significantly complicates the solution process [WP1997, Ch. 6.4].

5.1.2.2 Standard Inputs

Some input shapes have proven their merit in the past, where the necessary prior knowledge is limited to a rough estimate of the dominant frequency of the eigenmotion to be excited. This is easier to come by compared to a complete set of aerodynamic parameters as is necessary for the strictly "optimal" inputs of the last section.

The basic idea is then to design the experiment such, that the majority of its frequency content is centered at the assumed natural frequency $f_n = \frac{1}{T_n}$. Further, the

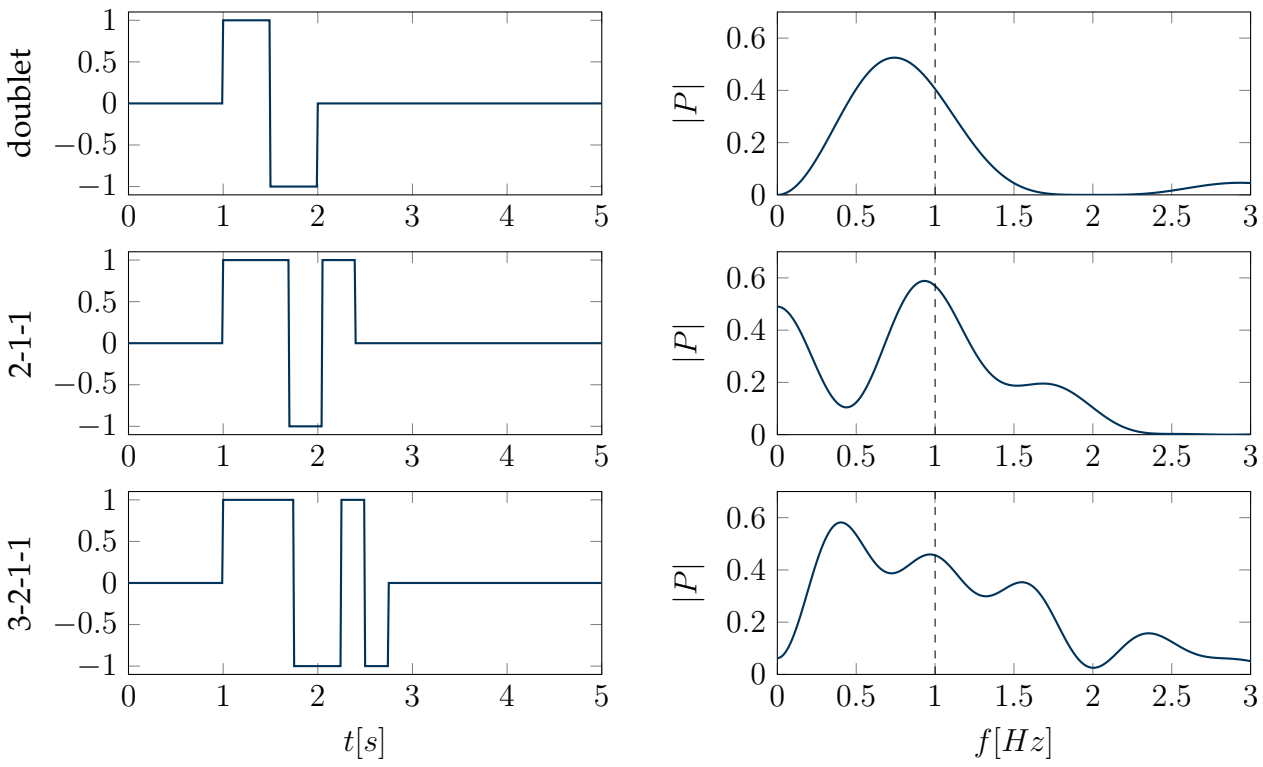


Figure 5.2: *Example step inputs, designed for a nominal frequency of 1Hz. The power spectral densities are computed using the Discrete Fourier Transform as approximation of the Fourier integral.*

inputs are close to symmetric about a trim condition, in order to minimize the deviation from the pre-defined trim point [MK2016, Ch. 9].

The resulting inputs are a combination of step functions, defined via the ratio of their durations. Some advice on how to choose the base step duration is given in [MK2016, Ch. 9] and [Jat2015, Ch. 2] and summed up next: A doublet, or 1-1 input consists of two steps in opposite directions of the same duration. Often the step time is chosen as

$$\Delta t_{1\,1} = \frac{1}{2} T_n = \frac{1}{2 f_n} \tag{5.1}$$

It has a distinct frequency peak just below the frequency of the corresponding square wave, but also contains the adjacent frequencies, which is favorable if the frequency to be excited is not perfectly known [MK2016, Ch. 9]. An example can be seen in the topmost subplot of figure 5.2.

To broaden the frequency spectrum, more pulses with different lengths can be included. In this way, the well-established 3-2-1-1 input arises, which is designed such, that the width of the 2 pulse corresponds to half the period of the expected natural frequency to excite

$$\Delta t_{3\,2\,1\,1} = \frac{1}{4} T_n = \frac{1}{4 f_n} \tag{5.2}$$

The other pulses then add spectral content to the left and right, which makes it even more robust to uncertainties compared to the doublet input [MK2016, Ch. 9]. It can be seen in the last subplot of figure 5.2.

A problem with the 3-2-1-1 input is the potentially long first pulse, which may drive the aircraft considerably from the trim condition. This may be alleviated by using a 2-1-1 sequence, where good results have been obtained with a pulse width of roughly 35% of the natural period.

$$\Delta t_{2\,1\,1} = \frac{0.7}{2} T_n = \frac{0.7}{2 f_n} \tag{5.3}$$

The price to pay is a somewhat poorer frequency content [MK2016, Ch. 9], see the center subplot of figure 5.2.

Other input shapes exist for special purposes, such as the push-over-pull-up, or roller coaster maneuver (which serves to excite lift and drag characteristics over a wide range of angle of attack), steady-heading sideslip (to investigate lateral stability), level-turn (to discern between q and $\dot{\alpha}$ influences), or the windup-turn (for data compatibility analysis). For more details, consider the respective sources [MK2016, Jat2006, Mul1986], which also give quite extensive practical advice on how to tune and conduct these especially tailored maneuvers.

5.1.2.3 No Prior Knowledge

If the methods of obtaining preliminary information are deemed unsatisfactory, inputs can be designed that are independent of the system characteristics. In that case, the approach entails exciting a broad range of input frequencies, in order to be sure to cover the characteristic ones.

Frequency sweeps, i.e. sine waves with linearly or exponentially increasing frequency, are used to this end, due to their broad, and adjustable frequency spectrum. Two examples for linear and exponential sweeps can be seen in the two top plots of figure 5.3, along with the corresponding power spectra. The disadvantage of frequency sweeps is their comparatively long duration of $30\,\mathrm{s}$ - $90\,\mathrm{s}$, which poses practical problems. Especially for RPAS applications, flying straight for around $30\,\mathrm{s}$ usually brings the aircraft out of view of the pilot, which complicates test execution and may be illegal under local BVLOS regulations.

Some of the downsides of a frequency sweep can be alleviated by using a multi-sine signal, i.e. the superposition of sine waves of different amplitude, frequency, and phase. They can be designed to cover a wide range of frequencies as well, and their execution time is usually shorter. However, they necessitate an automatic flight control system and an approach to determine the phase angles. This can be achieved by optimization for an optimum peak or crest-factor [MK2016, Ch. 9] [Mor2012b], or by using heuristic phase distributions like the “Schroeder sweep” [MK2016, Ch. 9].

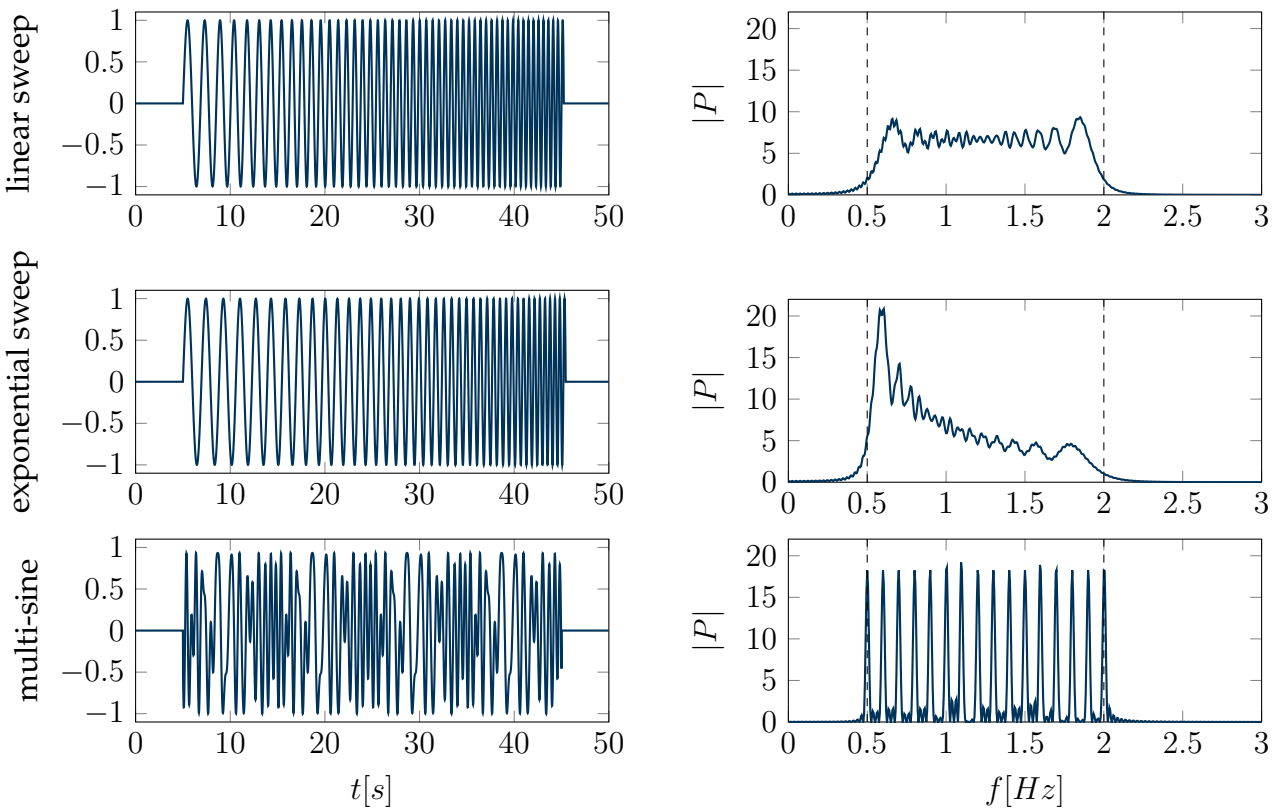


Figure 5.3: *Example sinusoidal inputs, designed for a nominal frequency range of 0.5Hz to 2Hz. The power spectral densities are computed using the Discrete Fourier Transform as approximation of the Fourier integral.*

An example, using the same overall test time as the sweep examples can be seen in figure 5.3. The power spectral density shows distinct spikes at the excitation frequencies, along with some other frequency content due to finite data-lengths and the resulting leakage.

5.1.3 Flight Testing for System Identification

MORELLI and KLEIN list a number of recommendations for practical flight testing for system identification [MK2016, Ch. 9], similar considerations can be found in [Jat2015, Ch. 2].

It has become evident in practice, that "first time right" is pretty much impossible, i.e. iterations will always occur. Mainly this is due to the fact, that for a new system, one has to gather some experience first, before meaningful experiments can be conducted. Thus MORELLI and KLEIN recommend to plan for these iterations from the start.

Also, to reduce the number of necessary maneuvers at a test point, certain control surface effects can be considered together. For example, if ailerons are always used asymmetrically, it may be possible to consider them as one control effector. The same is true for symmetric use of high-lift devices.

Maneuver repetition for each maneuver and at each test point is desirable, since random disturbances may make a data set unusable. A bare minimum of two repetitions (for identification and validation purposes) is recommended, where three or four repetitions allow for some robustness against random disturbances. Also, reversing the sign of the amplitude may be a good means to vary input shapes.

During the maneuver execution, it is advisable to keep the desired trim point for a short time before initiating the actual maneuver ($\sim 2\,\mathrm{s}$ for larger aircraft, less for RPAS), and to allow for some time after the actual excitation to record the free response of the system ($\sim 5\,\mathrm{s}$ for larger aircraft, less for RPAS) [Jat2015, Ch. 2].

For RPAS applications several aspects make manual flight testing for system identification purposes difficult: Most pilots are trained to keep the system airborne, and it goes against their nature to excite a characteristic motion in a pure feed-forward manner. Thus the actual input shapes will seldom correspond to what the analyst defined beforehand. Also, for small systems, visibly determining its state at some 100 meters away and precisely conducting a system identification maneuver is a challenging task at best. Thus these tests usually necessitate an automatic system, which, depending on its fidelity, may conduct a subset of the necessary steps (trimming the aircraft, pure feed-forward maneuvers, maneuvers with feedback control in the secondary inputs). An example system, co-developed by the author is illustrated in [KGH2018].

5.1.4 Data Acquisition System

For RPAS applications, the specification of dedicated flight test instrumentation for system identification is difficult: often no specific additional equipment can be installed. This is due to the usually tight cost structure and short development cycles on the commercial drone market. Thus the analyst will have to cope with the standard sensor suite installed on the aircraft. Fortunately, this is quite comprehensive in most cases due to the high degree of automation of currently available products.

Nevertheless, even if one cannot influence the choice of sensors, it is still crucial to know their characteristics. The main aspects to be aware of are illustrated next [MK2016, Ch. 9] [Jat2015, Ch. 2].

5.1.4.1 Sampling Rate

SHANNON's sampling theorem states a theoretical lower limit: if the sampling frequency f_s is at least twice as large as the highest frequency of interest, perfect reconstruction is theoretically possible [MK2016, Ch. 9] [Jat2015, Ch. 2]. However this result has seen to be over-confident in practice, which is why a factor of 25 is commonly advised [MK2016, Ch. 9]

$$f_s = 25 f_{max} \tag{5.4}$$

For aircraft, the characteristic frequencies additionally scale with the square root of the geometric model scale [MK2016, Ch. 9]

$$f_{max,RPAS} = \frac{1}{\sqrt{s}} f_{max} \tag{5.5}$$

Thus for a RPAS, which is a rough 1:10 scale model of a conventional aircraft with eigenfrequencies just below $2Hz$, a sampling frequency of at least $\sqrt{10} \cdot 25 \approx 80Hz$ would be advisable.

5.1.4.2 Signal Conditioning

When it comes to the actual data recording process, simultaneous sampling of all signals of interest is favorable. This can seldom be guaranteed in RPAS projects, which necessitates pre-processing of the digital data. On the one hand, signals with low sample rates can either be included directly, using the direct covariance scaling approach of section 3.10.2. Alternatively, they may be interpolated to higher sampling rates. Here, the problem with interpolation is twofold: it has a smoothing effect, which may be undesired; and it introduces dependent information that was not independently collected, thereby invalidating some of the assumptions in statistical estimation.

On the other hand, if sampling does not occur simultaneously, explicitly determining time-shifts between the signals becomes necessary. Otherwise the time delay may be attributed to a phase lag due to the characteristics of the dynamic system [MK2016, Ch. 9]. This reduces the requirement of simultaneous sampling at a common rate to at least accurately time-tagging all signals.

5.1.4.3 Sensor Installation Locations

For all sensors, it is not only in the interest of system identification, but also important for guidance and control, that they measure the quantity they were designed to measure. For air data sensors this means, that they should be located in the airflow where it is unobstructed by the aircraft, if possible. Accelerometers should be placed as close to the aircraft c.g. as possible in order to avoid measurement of virtual accelerations due to lever-arm effects [MK2016, Ch. 9]. If this is not possible, at least the installation location needs to be known accurately, to compensate for these lever arm effects. Also, the effect of structural vibration on measured accelerations should be minimized.

Further, it is important to measure the control surface deflections as close to the actual control surface as possible. Even though there is a kinematic link between the pilot controls and the control surface, this link includes many unknowns such as hysteresis elements, friction, flexible rods and cables etc.

The same is true for automatically controlled aircraft, which usually applies to RPAS: whereas the command signal to an actuator is easily obtained, this does not

necessarily correspond to the actual control surface deflection. In-between are the actuators dynamics and the physical link to the surface, which is why direct measurement of the control surface deflection should be preferred, where it is available. If it is not possible to measure control surface deflections, at least pre-processing the control commands using an assumed feedforward model of the actuator dynamics is advisable (see section 5.3.2.4 for an example). Otherwise, actuator dynamics will be lumped together with the aircraft's inherent damping and eigenfrequency, which may be undesired.

5.2 Flight Path Reconstruction

The first step after recording data should always be to perform a first sanity check by answering questions of the following type

- are inputs consistent with the modeling conventions in use?
- do positive control surface deflections lead to negative rotational rates?
- does a north velocity component lead to an increase in latitude?
- is the acceleration in body z-direction during steady, wing-level flight roughly $-9.81\,\mathrm{m/s^2}$?
- does a positive rate lead to a positive change in the corresponding attitude angle?
- are the orders of magnitude of the signals consistent with the assumed units?
- ...

Those checks involve plotting of many quantities, and usually do not take much time. However, they may save the analyst considerable trouble in later stages.

What is particular for flight vehicle system identification, is that the kinematic relationship between certain measurement data can be used to reconstruct a kinematically consistent trajectory, and even estimate some sensor error parameters. This is especially attractive for the estimation algorithms discussed so far: most of them inherently assume a minimum of corruption from systematic errors like biases, scale factors, or time lags, which can all be corrected based on kinematic relationships.

The basic idea in all cases is that measured accelerations, when integrated, should be consistent with measured velocities, which in turn, when integrated, should roughly match measured positions. The same is true for rotational rate, and possible attitude measurements. These relations can then be used to estimate systematic sensor errors, together with the aircraft's states by minimizing the differences between integrated and measured quantities [Jat2015, Ch. 10] [MK2016, Ch. 10]. The approach is applicable both for large aircraft and for RPAS, where in the latter case the larger random errors of the usual low-cost sensors need to be taken into account.

5.2.1 Commonly Used Kinematic Relationships

A short summary of the naming convention used at Institute of Flight System Dynamics (FSD) can be found in appendix F. More detailed derivations of the kinematic relationships will be found in section 5.3. Here, only the main results necessary for FPR are illustrated.

The velocity of an arbitrary point on the aircraft, given the velocity at the reference point, can be determined using the Euler derivative [Hol2018a]

$$\left(\vec{\boldsymbol{v}}_K^P\right)_B^E = \left(\vec{\boldsymbol{v}}_K^R\right)_B^E + \left(\vec{\boldsymbol{\omega}}_K^{IB}\right)_B \times \left(\vec{\boldsymbol{r}}^{RP}\right)_B \tag{5.6}$$

where it is assumed, that the aircraft is a rigid body.

Its acceleration can be determined based on the same rigid-body assumption, and repeated application of the Euler derivative [Hol2018a]. This is used to determine the acceleration at the Inertial Measurement Unit (IMU) installation location in the body fixed frame. Neglecting earth's rotational rate and the transport rate (to be discussed in section 5.3.1.1) this yields

$$\left(\vec{\boldsymbol{\omega}}_K^{IB}\right)_B \approx \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \approx \left(\vec{\boldsymbol{\omega}}_K^{EB}\right)_B \tag{5.7}$$

$$\begin{aligned}\left(\vec{\boldsymbol{a}}_K^{IMU}\right)_B^{II} &= \left(\dot{\vec{\boldsymbol{v}}}_K^R\right)_B^{EB} + \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{v}}_K^R\right)_B^E \\ &\quad + \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{r}}^{R,IMU}\right)_B\right) + \left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)_B^B \times \left(\vec{\boldsymbol{r}}^{R,IMU}\right)_B\end{aligned} \tag{5.8}$$

$$\begin{aligned}\Leftrightarrow \left(\dot{\vec{\boldsymbol{v}}}_K^R\right)_B^{EB} &= \left(\vec{\boldsymbol{a}}_K^{IMU}\right)_B^{II} - \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{v}}_K^R\right)_B^E \\ &\quad - \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{r}}^{R,IMU}\right)_B\right) - \left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)_B^B \times \left(\vec{\boldsymbol{r}}^{R,IMU}\right)_B\end{aligned} \tag{5.9}$$

In above equations, no model for the aircraft's forces and moments (aerodynamic, gravitational and thrust) are necessary, since the measured accelerations and rotational rates may be used directly [Jat2015, Ch. 10].

Also, the same attitude propagation equations, as will be detailed in section 5.3.1.2 may be used, assuming the pitch angle stays well below the singularity at $90°$

$$\dot{\boldsymbol{\Theta}}^{BO} = \begin{bmatrix} \dot{\phi}^{BO} \\ \dot{\theta}^{BO} \\ \dot{\psi}^{BO} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi^{BO}\tan\theta^{BO} & \cos\phi^{BO}\tan\theta^{BO} \\ 0 & \cos\phi^{BO} & -\sin\phi^{BO} \\ 0 & \frac{\sin\phi^{BO}}{\cos\theta^{BO}} & \frac{\cos\phi^{BO}}{\cos\theta^{BO}} \end{bmatrix} \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \tag{5.10}$$

Position propagation equations are also the same as will be presented in detail in section 5.3.1.3

$$\begin{bmatrix} \dot{\lambda}^R \\ \dot{\mu}^R \\ \dot{h}^R \end{bmatrix}_{WGS84} = \begin{bmatrix} \frac{\left(v_K^R\right)_O^E}{(N_\mu + h^R)\cos\mu^R} \\ \frac{\left(u_K^R\right)_O^E}{M_\mu + h^R} \\ -\left(w_K^R\right)_O^E \end{bmatrix} \tag{5.11}$$

Often the lateral position states are not directly used, since they do not influence the aircraft's dynamics. Only altitude has an influence on air density and thus dynamic pressure [MK2016, Ch. 10.1]. However, in order to check the correctness of the postulated model, it may make sense to include them, in order to avoid errors in the assumed coordinate systems.

The above are all purely kinematic, perfectly known relationships, i.e. there are no aspects that necessitate a model of the aircraft. Collecting them, a non-linear system of equations of the form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ may be built. Then, velocity $\left(\vec{\boldsymbol{v}}_K^R\right)_B^E$, attitude $\boldsymbol{\Theta}^{BO}$, and position $\left(\vec{\boldsymbol{r}}^R\right)_{WGS84}$ serve as states, whereas acceleration $\left(\vec{\boldsymbol{a}}_K^{IMU}\right)_B^{II}$, rotational rate $\left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B$, and rotational acceleration $\left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)_B^B$ are considered to be inputs. This model is then at the heart of a state estimation problem.

5.2.2 Sensor Models and Error Estimation

The relations in the last section are purely kinematic. However, some modeling effort cannot be avoided when solving the FPR problem. Sensor errors may be treated as deterministic and constant, or may be included as additional states in the problem, and thus be allowed to vary in time. The former is often a good approximation, if the considered time span is short (e.g. short flight time in RPAS applications; considering only individual maneuvers), whereas the latter might make more sense for longer time segments, but this choice is in general problem specific.

Here, sensor error terms will be considered to be constant, and deterministic. Additionally, random measurement noise will be included. Usually, an approach involving a bias term $\boldsymbol{\Delta}_{\boldsymbol{i}}$, scale factor error $K_{\boldsymbol{i},1} \dots K_{\boldsymbol{i},n_i}$ and white, Gaussian measurement noise $\boldsymbol{n_i}$ will be used to relate a measured, physical input quantity $\boldsymbol{i}$ to the measurement output $\boldsymbol{o}$ as [MK2016, Ch. 10.4]

$$\boldsymbol{o} = (\mathbf{I}_{n_i} + \operatorname{diag}(K_{\boldsymbol{i},1} \dots K_{\boldsymbol{i},n_i}))\, \boldsymbol{i} + \boldsymbol{\Delta_i} + \boldsymbol{n_i} \tag{5.12}$$

If the respective quantity serves as an input $\boldsymbol{u}$ to the system (e.g. acceleration, rotational rate), above sensor error model may be solved for $\boldsymbol{i}$

$$\begin{aligned} \boldsymbol{i} &= (\mathbf{I}_{n_i} + \operatorname{diag}(K_{\boldsymbol{i},1} \dots K_{\boldsymbol{i},n_i}))^{-1} (\boldsymbol{o} - \boldsymbol{\Delta_i} - \boldsymbol{w_i}) \\ &= \left(\mathbf{I}_{n_i} + \operatorname{diag}\left(\tilde{K}_{\boldsymbol{i},1} \dots \tilde{K}_{\boldsymbol{i},n_i}\right)\right) \boldsymbol{o} + \tilde{\boldsymbol{\Delta}}_{\boldsymbol{i}} + \tilde{\boldsymbol{n}}_{\boldsymbol{i}} \end{aligned} \tag{5.13}$$

where the tilde-quantities are introduced to facilitate the model.

It can be noted, that the random error terms will play two different roles: if the respective quantity enters the system as an input $\boldsymbol{u}$, the random error will affect the system as if it were *process noise* [MK2016, Ch. 10.4]. If, however, the respective quantity is to be found on the output side of the system, the random error will act as *measurement noise*.

Above equation (5.12) represents the most generic sensor error model to be used in this work. Despite its simplicity it has served well, as will be illustrated in chapter 6 and is mentioned in [MK2016, Ch. 10.4]. For some sensors it is not possible to consider bias and scale factor separately, then often the $K_{i,j}$ are set fixed at 0. This may arise in cases, when the physical quantity has rather large values, not centered around zero (e.g. airspeed); then, a small scale factor may have a similar effect as a bias term, which introduces large correlations between these two error sources and thus makes their independent determination difficult [Jat2015, Ch. 10].

The input quantities $\boldsymbol{i}$ represent the physical quantity to be measured, such as

- *Magnetometer*
  the three axis magnetic field vector in body fixed coordinates (using eq. (5.37) for the rotation matrix M_{BO})
  $$\left(\vec{\boldsymbol{b}}\right)_B\left(\mu^R, \lambda^R, h^R\right) = \mathrm{M}_{BO} \cdot \left(\vec{\boldsymbol{b}}\right)_O\left(\mu^R, \lambda^R, h^R\right) \tag{5.14}$$
  where the dependency of the latter on the position may be dropped if flights were conducted within a small area
- *Accelerometer*
  accelerometer measurements, i.e. combination of acceleration and gravity
  $$\left(\vec{\boldsymbol{a}}_K^{IMU}\right)_B^{II} + \mathrm{M}_{BO}(\vec{\boldsymbol{g}})_O \tag{5.15}$$
- *Global Navigation Satellite System (GNSS)*
  GNSS velocity, corrected for the installation location of the GNSS antenna
  $$\left(\vec{\boldsymbol{v}}_K^{GNSS}\right)_O^E = \mathrm{M}_{BO}^\intercal\left(\left(\vec{\boldsymbol{v}}_K^R\right)_B^E + \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{r}}^{R,GNSS}\right)_B\right) \tag{5.16}$$
  in this context, the second term, i.e. the lever arm influence is sometimes neglected. GNSS position is considered directly, without correcting for the lever arm.

Combining the kinematic relations of section 5.2.1 with above sensor error models, yields a system of nonlinear differential equations of the form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{w})$ with an output equation $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) + \boldsymbol{v}$. Although, generally the process noise term here enters non-linearly, the system may very well be approximated using additive $\boldsymbol{w}$, which then is in agreement with the problem formulations used here. All deterministic sensor errors to be estimated are collected in $\boldsymbol{\theta}$, all random errors on the input side in $\boldsymbol{w}$ and all random errors on the output side are collected in $\boldsymbol{v}$.

5.2.3 Wind and Flow Angle Estimation

For aircraft system identification, the knowledge of flow quantities (angle of attack, angle of sideslip, aerodynamic velocity) is paramount, since they are the main influencing factors for the acting aerodynamic forces. If the flight tests were conducted in

calm weather conditions, the influence of wind may be neglected, and the aerodynamic quantities may be computed from their kinematic counterparts.

However, due to the small inertia of the aircraft, and the comparatively slow cruise speeds, even small wind components may have considerable effects in RPAS applications. This is way it often makes sense to estimate at least some wind components. This can be achieved by extending the FPR model with wind states, and including them in the measurement models for aerodynamic quantities, such as dynamic pressure.

The most unreliable estimates are obtained for vertical wind components, so they are often set to 0. Then, assuming a random walk behavior for the wind estimates yields a the following propagation equation

$$\left(\dot{u}_W^R\right)_O^E = w_{W,u} \tag{5.17}$$

$$\left(\dot{v}_W^R\right)_O^E = w_{W,v} \tag{5.18}$$

$$\left(\dot{w}_W^R\right)_O^E = 0 \tag{5.19}$$

More complicated models, possibly related to the DRYDEN atmosphere may be included in a similar fashion, but above simple formulation has served well [GH2016]. Wind rotational rates, due to a changing wind profile over the extension of the aircraft are assumed to be zero.

Dynamic pressure at the installation location of the Pitot probe is then

$$\left(\vec{\boldsymbol{v}}_A^{PITOT}\right)_B^E = \left(\vec{\boldsymbol{v}}_K^R\right)_B^E - \mathbf{M}_{BO}\left(\vec{\boldsymbol{v}}_W^R\right)_O^E + \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{r}}^{R,PITOT}\right)_B \tag{5.20}$$

$$\bar{q} = \frac{\rho}{2}\left\|\left(\vec{\boldsymbol{v}}_A^{PITOT}\right)_B^E\right\|_2^2 \tag{5.21}$$

where the lever arm correction is small compared to the nominal values of the other quantities, thus often neglected [MK2016, Ch. 10.3].

If one is so fortunate as to have angle of attack and angle of sideslip sensors aboard the aircraft, the respective flow quantities at the installation locations are (ignoring possibly different installation locations of the two) [MK2016, Ch. 10.3]

$$\left(\vec{\boldsymbol{v}}_A^{FLOW}\right)_B^E = \left(\vec{\boldsymbol{v}}_K^R\right)_B^E - \mathbf{M}_{BO}\left(\vec{\boldsymbol{v}}_W^R\right)_O^E + \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \times \left(\vec{\boldsymbol{r}}^{R,FLOW}\right)_B \tag{5.22}$$

$$\alpha = \operatorname{atan2}\left(\left(w_A^{FLOW}\right)_B^E, \left(u_A^{FLOW}\right)_B^E\right) \tag{5.23}$$

$$\beta = \operatorname{atan2}\left(\left(v_A^{FLOW}\right)_B^E, \sqrt{(u_A^{FLOW})_B^{E\,2} + (w_A^{FLOW})_B^{E\,2}}\right) \tag{5.24}$$

Dynamic pressure, and flow quantities may be subjected to the same error model (5.12), involving bias and scale factor error as before.

In this setup, the horizontal wind components are part of an augmented state vector. The state estimation algorithm will then be able to reconstruct the wind data, based on the available measurements. The main information source for this is the difference between kinematic velocities and the measured aerodynamic velocities.

5.2.4 Solving the Flight Path Reconstruction Problem

The resulting FPR problem is then one of combined state and parameter estimation, and may be tackled with the approaches illustrated in chapters 2.2 and 2.3, respectively the problem formulations of section 4.3 and 4.4. Alternatively, non-linear smoothing solutions have been obtained using a cost function including measurement and process noise inputs, together with calculus of variations to solve this problem [Bac1982, BW1985]. Some authors note that if the stochastic part is not too dominant (e.g. because high quality sensors are used), it may be neglected and deterministic methods may be used [Jat2015, Ch. 10] [MK2016, Ch. 10.1]. In RPAS applications with often low-cost sensors, the random components tend to be significant, which makes a stochastic treatment almost inevitable.

Irrespective of the solution methodology, the results for the FPR problem formulated in this section will consist of

- a set of deterministic sensor error parameters, to correct the measurements with,
- a kinematically consistent state trajectory, possibly to be used as initial guess for the states when applying the full discretization methods of chapter 4.2
- an estimate for the horizontal wind components during the maneuvers

All these can be used in the subsequent parameter estimation. Sometimes these reconstructed measurements are even used instead of real data in the subsequent parameter estimation step, if the corresponding sensors are impractical, or not available [MK2016, Ch. 10.2]. However, using this kind of "artificial measurements" may introduce residual coloring, since they were not obtained independently. This violates some basic assumptions in the estimation approach. The severity of this, possibly rather academic issue, has to be investigated on a case to case basis.

As with all parameter estimation algorithms, a suitable initial guess for the deterministic parameters is necessary. Very often it is possible, to obtain bias estimates from steady state conditions before take-off, whereas scale factor errors are often small, so an initial value of 0 can be a good guess.

Three major difficulties remain, however. The first is the determination of the process and measurement noise covariance matrices. Physically, they represent the sensor noise characteristics, and may thus be extracted from data-sheets or laboratory calibrations. However, during real-life operations, other factors, such as vibration or flexibility of the aircraft may increase the noise level, which is why the original noise characteristics may offer a good starting point, but manual tuning of covariance matrices is often inevitable. Unfortunately, there are no hard guidelines, on how the results should look, rather some experience is necessary to assess them correctly.

Good results have been achieved, by setting the process noise terms to zero, i.e. only integrating the inertial quantities in time. There will be some drift in the solution (due to uncorrected biases and integration of noise), but in this way a rough comparison of

the resulting attitudes and velocities with measured data is possible. These integration results offer valuable clues as to the correctness of the postulated model: it will be directly evident if signs or axes in the measurements are wrongly considered. Then the noise covariances can be slowly increased, to create a better match between propagated and measured data.

The second problem is closely related to the first: the choice of process noise covariances for the wind states is not evident at all. Its value may be rather large to allow comparatively large changes per sampling instant, if flow angle measurements are available. Then the algorithm will be able to estimate relatively high frequency fluctuations of the wind vector. This is possible, since reliable information about its direction and magnitude is available at all times by considering the difference between kinematic and aerodynamic velocities.

If, on the other hand, no flow angle sensors are available, only very low frequency changes in the wind vector are possible to estimate. If the process noise covariance is then set too large, the estimation algorithm quickly starts to compensate for other estimation errors with wind components. This happens, since a Pitot probe only yields information about the magnitude of the aerodynamic velocity vector, and no reliable information about its direction is available.

The last problem is due to the rotational acceleration information, which is necessary to properly integrate the measured accelerations. Angular acceleration sensors are still very uncommon, and usually too big, heavy and too costly to be integrated in RPAS. Thus different means of obtaining $\left(\dot{\vec{\omega}}_K^{OB}\right)_B^B$ are needed. Some numerical approaches will be briefly discussed in section 5.4. The increased uncertainty, which is introduced via these approaches, needs to be included in the corresponding process noise covariance matrix. This can be carried so far as to completely ignore rotational rate measurements, and purely include their effect via an increase in the acceleration measurement covariance. An example application of this approach can be found in [GH2016].

5.3 Modeling

All of the estimation algorithms that have been presented here can be formulated independently of the model at hand. Nevertheless, successful application hinges on understanding the estimation approaches as well as a thorough understanding of the governing principles of the system to be identified [CJ2012, Ch. 1]. For this reason, the basic model equations to be used in the applications in chapter 6, will be illustrated next. They are mainly based on the application of the translational and rotational forms of NEWTON's second law [MK2016, Ch. 2]. Again, this is presented with RPAS applications in mind, thus the common simplifications and model assumptions are illustrated.

5.3.1 Rigid Body Aircraft Equations of Motion

Models will come in the form of a system of non-linear, first order, ordinary differential equations, together with an algebraic, non-linear output equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) + \boldsymbol{w} \tag{5.25}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \tag{5.26}$$

For aircraft applications, the rigid body states usually comprise of components related to linear and angular velocity, attitude and position. Further subsystem states may appear, too, depending on the fidelity of the model. Inputs are the control surface deflections, together with power settings for the propulsion system, and secondary inputs to change the aircraft configuration (landing gear, high-lift devices). The parameter vector mainly consists of aerodynamic parameters, together with nuisance variables for e.g. sensor errors. Above system can in general only be solved numerically, which is usually done with the aid of Runge-Kutta methods (see Appendix D.3 for a short summary).

The following subsections mainly follow the notation as is common at the FSD, see appendix F for details. Also, most of the following derivations are based on [Hol2018a, Hol2018b], further details can be found in [SL2003, ER1996, BAL2010] or the books dedicated to aircraft system identification [MK2016, Jat2006].

The following presentation is not intended as an in-depth illustration of flight mechanic modeling. Moreover it presents the necessary basics, which will be used in the applications in chapter 6. It will be based on the following assumptions, which are similar to those used in [MK2016, Ch. 3]

- the earth is considered inertial, i.e. Newton's law is eventually formulated in the Earth-Centered-Earth-Fixed (ECEF) frame
- the aircraft is a rigid body
- the aircraft mass and mass distribution is quasi-constant, i.e. it may be subject to temporal changes, however these are not explicitly considered in the conservation of momentum
- earth curvature is negligible ("flat-earth")
- gravitational acceleration is fixed, i.e. constant in direction and magnitude

5.3.1.1 Translational and Rotational Kinematics

The basis for all kinematic considerations in the aircraft context is the Euler derivative for time derivatives in rotating frames. The time derivative with respect to the A-frame of a vector $(\vec{\boldsymbol{v}})_B$ given in the B frame then consists of a direct part, and a term due to the rotation of the A frame versus the B frame [CJ2012, Ch. 3]

$$\left(\frac{d}{dt}\right)^A (\vec{\boldsymbol{v}})_B = \left(\dot{\vec{\boldsymbol{v}}}\right)_B^B + \left(\vec{\boldsymbol{\omega}}_K^{AB}\right)_B \times (\vec{\boldsymbol{v}})_B \tag{5.27}$$

Repeated application of above equation to first obtain the velocity, then the acceleration of an arbitrary point on the aircraft leads to an expression for the inertial acceleration $\left(\vec{\boldsymbol{a}}_K^P\right)^{II}$. Then, assuming a quasi-steady mass distribution, and considering the aircraft to be a rigid body, the conservation of linear momentum leads to the following equation for the velocity propagation based on the sum of all acting forces in the reference point R [Hol2018a] [CJ2012, Ch. 3]

$$\begin{aligned}\frac{\sum \vec{\boldsymbol{F}}^R}{m} = &\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{IE} \times \left(\vec{\boldsymbol{\omega}}_K^{IE} \times \vec{\boldsymbol{r}}^R\right) + \vec{\boldsymbol{\omega}}_K^{IE} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E \\ &+ \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E + \left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B \times \vec{\boldsymbol{r}}^{RG} + \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{\omega}}_K^{IB} \times \vec{\boldsymbol{r}}^{RG}\right)\end{aligned} \tag{5.28}$$

The following quantities are related in above equation

- $\sum \vec{\boldsymbol{F}}^R$: the sum of all forces acting at the aircraft (A/C) reference point
- m: the total mass of the A/C
- $\vec{\boldsymbol{r}}^{RG}$: position of the A/C c.g. relative to the A/C reference point
- $\vec{\boldsymbol{r}}^R$: position of the A/C reference point w.r.t the origin of the coordinate system
- $\left(\vec{\boldsymbol{v}}_K^R\right)^E$: linear velocity of the A/C reference point with respect to the ECEF frame
- $\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB}$: change in the velocity with respect to the ECEF frame; derivative taken in the body-fixed frame
- $\vec{\boldsymbol{\omega}}_K^{IB}$: rotational rate of the A/C versus inertial space
- $\vec{\boldsymbol{\omega}}_K^{IE}$: earth's rotational rate
- $\left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B$: change in rotational rate of the A/C versus inertial space; derivative taken in the body-fixed frame

For system identification purposes, a flat, non-rotating earth can be assumed (earth rotation $\vec{\boldsymbol{\omega}}_K^{IE} \approx \vec{\boldsymbol{0}}$; transport rate $\vec{\boldsymbol{\omega}}_K^{EO} \approx \vec{\boldsymbol{0}}$), since the travel distances usually stay small, especially for RPAS. Then the following simplifications arise [Hol2018a]

$$\vec{\boldsymbol{\omega}}_K^{IB} \approx \vec{\boldsymbol{\omega}}_K^{OB} \approx \vec{\boldsymbol{\omega}}_K^{EB} \tag{5.29}$$

$$\frac{\sum \vec{\boldsymbol{F}}^R}{m} = \left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E + \left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)^B \times \vec{\boldsymbol{r}}^{RG} + \vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{\omega}}_K^{OB} \times \vec{\boldsymbol{r}}^{RG}\right) \tag{5.30}$$

With the aid of the Euler derivative, the assumptions of quasi-steady mass distribution and a rigid body aircraft, the conservation of angular momentum leads to a differential equation for the propagation of the rotational rates, based on the sum of all acting moments in the reference point R [Hol2018a] [CJ2012, Ch. 3]

$$\begin{aligned}\sum \vec{\boldsymbol{M}}^R = &\left(\mathbf{I}^R\right)\left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B + \vec{\boldsymbol{\omega}}_K^{IB} \times \left[\left(\mathbf{I}^R\right)\vec{\boldsymbol{\omega}}_K^{IB}\right] \\ &+ m\vec{\boldsymbol{r}}^{RG} \times \left[\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{IE} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E + \vec{\boldsymbol{\omega}}_K^{IE} \times \left(\vec{\boldsymbol{\omega}}_K^{IE} \times \vec{\boldsymbol{r}}^R\right) + \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E\right]\end{aligned} \tag{5.31}$$

Again, with the assumption of a flat, non-rotating earth, above equation may be simplified

$$\begin{aligned}\sum \vec{\boldsymbol{M}}^R = &\left(\mathbf{I}^R\right)\left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)^B + \vec{\boldsymbol{\omega}}_K^{OB} \times \left[\left(\mathbf{I}^R\right)\vec{\boldsymbol{\omega}}_K^{OB}\right] \\ &+ m\vec{\boldsymbol{r}}^{RG} \times \left[\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E\right]\end{aligned} \tag{5.32}$$

$\left(\mathbf{I}^R\right)$ denotes the total tensor of inertia at the A/C reference point, see section 5.3.2.3 for details.

Here it is decided to formulate the equations of motion at a body-fixed reference point R. The equations would simplify, if the c.g. was to be used as reference point, however this might not be fixed (due to fuel consumption or different payload weights), which complicates the computations at other stages. Considering the equations of motion at the body fixed reference point R necessitates a simultaneous solution of the translational and rotational equations, since both depend linearly on the derivatives $\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB}$ and $\left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)^B$. For the solution, it is necessary to define the operator $[\vec{\boldsymbol{r}}\times]$, which translates a cross product into a matrix-vector multiplication according to

$$\vec{\boldsymbol{r}}_1 \times \vec{\boldsymbol{r}}_2 = \begin{bmatrix} y_1z_2 - y_2z_1 \\ x_2z_1 - x_1z_2 \\ x_1y_2 - x_2y_1 \end{bmatrix} = \begin{bmatrix} 0 & -z_1 & y_1 \\ z_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = [\vec{\boldsymbol{r}}_1\times]\,\vec{\boldsymbol{r}}_2 = -[\vec{\boldsymbol{r}}_2\times]\,\vec{\boldsymbol{r}}_1 \tag{5.33}$$

Then it is possible to assemble the following linear system of equations [Hol2018a]

$$\begin{bmatrix} \vec{\boldsymbol{\alpha}}_1 \\ \vec{\boldsymbol{\alpha}}_2 \end{bmatrix} = \begin{bmatrix} m\mathbf{I}_{n_3} & -m\left[\vec{\boldsymbol{r}}^{RG}\times\right] \\ m\left[\vec{\boldsymbol{r}}^{RG}\times\right] & \left(\mathbf{I}^R\right) \end{bmatrix} \begin{bmatrix} \left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} \\ \left(\dot{\vec{\boldsymbol{\omega}}}_K^{OB}\right)^B \end{bmatrix} \tag{5.34}$$

$$\vec{\boldsymbol{\alpha}}_1 = \sum \vec{\boldsymbol{F}}^R - m\left(\vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E + \vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{\omega}}_K^{OB} \times \vec{\boldsymbol{r}}^{RG}\right)\right) \tag{5.35}$$

$$\vec{\boldsymbol{\alpha}}_2 = \sum \vec{\boldsymbol{M}}^R - \vec{\boldsymbol{\omega}}_K^{OB} \times \left[\left(\mathbf{I}^R\right)\vec{\boldsymbol{\omega}}_K^{OB}\right] - m\vec{\boldsymbol{r}}^{RG} \times \vec{\boldsymbol{\omega}}_K^{OB} \times \left(\vec{\boldsymbol{V}}_K^R\right)^E \tag{5.36}$$

which can be solved for the time derivatives of velocity and rotational rate. Here, above equations will be formulated in the body fixed B-frame.

5.3.1.2 Attitude Kinematics

Commonly, aircraft attitude is represented by Euler angles $\boldsymbol{\Theta}^{BO}$. The transformation matrix from local North-East-Down (NED) to body fixed coordinates, $\mathbf{M}_{BO}$ is then parameterized by heading angle ψ^{BO}, pitch angle θ^{BO}, and bank angle ϕ^{BO} [Hol2018a]

$$\mathbf{M}_{BO}\left(\boldsymbol{\Theta}^{BO}\right) = \mathbf{M}_x\left(\phi^{BO}\right)\mathbf{M}_y\left(\theta^{BO}\right)\mathbf{M}_z\left(\psi^{BO}\right) \tag{5.37}$$

$$\boldsymbol{\Theta}^{BO} = \begin{bmatrix} \phi^{BO} & \theta^{BO} & \psi^{BO} \end{bmatrix}^\intercal \tag{5.38}$$

The time derivatives of the Euler angles depend on the body-fixed rotational rates $\left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B$ according to

$$\begin{bmatrix} p_K^{OB} \\ q_K^{OB} \\ r_K^{OB} \end{bmatrix}_B = \begin{bmatrix} \dot{\phi}_{BO} \\ 0 \\ 0 \end{bmatrix} + \mathbf{M}_x\left(\phi^{BO}\right) \begin{bmatrix} 0 \\ \dot{\theta}_{BO} \\ 0 \end{bmatrix} + \mathbf{M}_x\left(\phi^{BO}\right) \mathbf{M}_y\left(\theta^{BO}\right) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_{BO} \end{bmatrix} \tag{5.39}$$

$$\begin{aligned} \dot{\boldsymbol{\Theta}}^{BO} &= \begin{bmatrix} \dot{\phi}^{BO} \\ \dot{\theta}^{BO} \\ \dot{\psi}^{BO} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi^{BO}\tan\theta^{BO} & \cos\phi^{BO}\tan\theta^{BO} \\ 0 & \cos\phi^{BO} & -\sin\phi^{BO} \\ 0 & \frac{\sin\phi^{BO}}{\cos\theta^{BO}} & \frac{\cos\phi^{BO}}{\cos\theta^{BO}} \end{bmatrix} \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \\ &= \mathbf{M}\left(\boldsymbol{\Theta}^{BO}\right) \cdot \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \end{aligned} \tag{5.40}$$

Above equation illustrates the biggest downside of using Euler angles for attitude representation: the singularity at pitch angles of $\theta^{BO} = 90°$. This could be circumvented by using unit-quaternions for attitude representation, see e.g. [CJ2012, Ch. 3], which is however seldom done for system identification tasks, because

- Flight testing for system identification should not lead to pitch angles close to above mentioned singularity.
- Euler angles are more intuitive, which helps interpreting the results, detecting dependencies and separating the equations of motion in lateral and longitudinal subsystems.
- In order for quaternions to be a valid attitude representation, they need to be of unit length, necessitating additional effort.

5.3.1.3 Position Kinematics

To complete the necessary set of rigid body states, position information needs to be included. Common approaches for RPAS applications are to consider either a local, Cartesian navigation frame, or to propagate position based on World Geodetic System 1984 (WGS84) coordinates.

The first version can easily be achieved using the velocity in the local NED frame

$$\left(\dot{\vec{\boldsymbol{r}}}^R\right)_N^N = \begin{bmatrix} \dot{x}^R \\ \dot{y}^R \\ \dot{z}^R \end{bmatrix}_N^N = \mathbf{M}_{BO}^{\intercal}\left(\vec{\boldsymbol{v}}_K^R\right)_B^E \tag{5.41}$$

where the last component is the negative change in altitude $-\dot{h} = \left(\dot{z}_K^R\right)_N^N$.

Propagation of the WGS84 coordinates necessitates the computation of the radii of curvature of the meridian M_μ and prime vertical N_μ [Hol2018a]

$$N_\mu = \frac{a}{\sqrt{1 - e^2 \sin^2 \mu^R}} \tag{5.42}$$

$$M_\mu = N_\mu \frac{1 - e^2}{1 - e^2 \sin^2 \mu^R} \tag{5.43}$$

The change in WGS84 position is then described using the components of the velocity in NED coordinates $\left(\vec{\boldsymbol{v}}_K^R\right)_O^E = \mathrm{M}_{BO}^{\mathrm{T}}\left(\vec{\boldsymbol{v}}_K^R\right)_B^E$

$$\left(\dot{\vec{\boldsymbol{r}}}^R\right)_{WGS84} = \begin{bmatrix} \dot{\lambda}^R \\ \dot{\mu}^R \\ \dot{h}^R \end{bmatrix}_{WGS84} = \begin{bmatrix} \frac{\left(v_K^R\right)_O^E}{(N_\mu + h^R)\cos\mu^R} \\ \frac{\left(u_K^R\right)_O^E}{M_\mu + h^R} \\ -\left(w_K^R\right)_O^E \end{bmatrix} \tag{5.44}$$

Since only altitude has an influence on flight dynamics (via the corresponding change in density), which is significant enough to retain, the lateral position states are often discarded. At a maximum they may be used to check if the position propagation based on the estimated model velocity agrees with the measured position.

5.3.2 Subsystem Modelling

After having the kinematic equations of motion in place, the only thing remaining is to determine the acting forces $\sum \vec{\boldsymbol{F}}^R$ and moments $\sum \vec{\boldsymbol{M}}^R$, which drive the translational and rotational equations of motion. The main contributions usually considered in RPAS applications are due to aerodynamics, gravity, propulsion, and possibly ground reaction forces. Since the main goal in aircraft system identification, as presented here, is the determination of a suitable aerodynamic model, this aspects is treated in more detail in section 5.3.4, whereas the other subsystems will be briefly illustrated next.

Obtaining subsystem parameters is often difficult. Sometimes the manufacturer does not want to provide them, because they are proprietary, and sometimes the manufacturer does not know them himself. Thus, even if a detailed modeling was theoretically possible, the lack of information often makes it necessary to find suitable abstractions, with fewer, easy to estimate parameters. Common approximations then include first and second order lag elements, or mass-spring-damper combinations. A good rule of thumb is to use subsystem parameters, where available, and to make simple assumptions otherwise. Building high-fidelity models with a huge number of unknown parameters is of no use, if they cannot be reasonably determined.

5.3.2.1 Engine Modeling

Appropriately incorporating propulsion effects is crucial, since the aerodynamic drag cannot be estimated accurately otherwise [Jat2015, Ch. 2]. For many engines (combustion and electric), a first order lag behavior between the engine command and the rotational rate is a good approximation.

$$\dot{\omega} = -\frac{1}{T}\omega + \frac{1}{T}\omega_{cmd} \tag{5.45}$$

In contrast to that, e.g. a fully physical approach for an electrical propulsion necessitates values for three to five electrical constants for the actual Brushless DC (BLDC)

motor and a set of gains for the engine speed controller, resulting in many more parameters whose values are often not easy to come by. Accurately modeling a combustion engine is even more challenging.

5.3.2.2 Propeller Modeling

In propeller modeling reliable information is also difficult to obtain, especially for RPAS applications. Further, it is not possible to estimate propulsion and aerodynamic parameters simultaneously, since only their combined effect is visible in the system reaction. Thus, their parameters are highly correlated, making an independent determination impossible.

This also leads to the effect that deficiencies in the propulsion model will inevitably be compensated for by aerodynamic effects. Eventually, the overall system outputs then may match the measurements, which is a mixed blessing: on the one hand, the overall system performance may represent the measured values very well. On the other hand, however, the predictive capability of the model may be restricted, since propulsion model deficiencies can only be meaningfully compensated in domains, where data is available. For other domains, it cannot be guaranteed that the model is still reasonable.

A well-established way to formulate propeller effects is by considering non dimensional coefficients, similar to those used in fixed wing aerodynamics [DAS2014]

$$C_T = \frac{4\pi^2 T}{\rho\omega^2 d^4} \tag{5.46}$$

$$C_Q = \frac{4\pi^2 Q}{\rho\omega^2 d^5} \tag{5.47}$$

$$C_P = \frac{8\pi^3 P}{\rho\omega^3 d^5} = \frac{2\pi\omega}{\omega}\frac{4\pi^2 Q}{\rho\omega^2 d^5} = 2\pi C_Q \tag{5.48}$$

$$J = \frac{V_A 2\pi}{\omega d} \tag{5.49}$$

where C_T, C_Q, and C_P are thrust, torque and power coefficient respectively. In order to obtain a dimensionless thrust coefficient, similarly to aerodynamic coefficients, a representative velocity squared ($\omega^2 d^2$), reference area ($\sim d^2$) and density (ρ) are used. For the torque coefficient an additional representative length enters the equation. J is termed the *advance ratio* and as relation between axial and radial velocity is related to the local angle of attack at the propeller blade.

A large data-base, for many commercially available propellers is provided by the University of Illinois at Urbana-Champaign [BDAS2018], where Thrust and Power coefficients are logged as function of the advance ratio for different rotational speeds. The findings collected there have also been published [Bra2005, BS2011, Det2014, DAS2014].

In order to be compatible with the equations of motion, propulsion forces and moments need to be translated from the propeller's coordinate system P and installation

location P to the reference point R, and the body fixed frame B

$$\left(\vec{\boldsymbol{F}}_P^R\right)_B = \mathbf{M}_{BP} \cdot \left(\vec{\boldsymbol{F}}_P^P\right)_P = \mathbf{M}_{BP} \cdot \begin{bmatrix} C_T(J)\,\frac{\rho\omega^2 d^4}{4\pi^2} \\ 0 \\ 0 \end{bmatrix} \tag{5.50}$$

$$\begin{aligned} \left(\vec{\boldsymbol{M}}_P^R\right)_B &= \left(\vec{\boldsymbol{r}}^{RP}\right)_B \times \mathbf{M}_{BP}\left(\vec{\boldsymbol{F}}_P^P\right)_P + \mathbf{M}_{BP} \cdot \left(\vec{\boldsymbol{M}}_P^P\right)_P \\ &= \left(\vec{\boldsymbol{r}}^{RP}\right)_B \times \mathbf{M}_{BP}\left(\vec{\boldsymbol{F}}_P^P\right)_P + \mathbf{M}_{BP} \cdot \begin{bmatrix} C_Q(J)\,\frac{\rho\omega^2 d^5}{4\pi^2} \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{5.51}$$

The second equality in both cases showcases the use of thrust and torque coefficients, while neglecting radial components, i.e. the influence of non-axial inflow will not be considered.

5.3.2.3 Weight and Balance Modeling

Since most current RPAS systems rely on an electrical propulsion system, weight and balance considerations are comparatively easy: no fuel is consumed, which would change the aircraft weight, inertia and c.g. If so desired, only additional mass due to changes in payload may be considered.

The overall c.g. and inertia may then be computed as weighted sum, and using the parallel axis theorem. Considering n_m elements (one of which is the empty aircraft structure), the common c.g. in body fixed coordinates is

$$m = \sum_{i=1}^{n_m} m_i \tag{5.52}$$

$$\left(\vec{\boldsymbol{r}}^{RG}\right)_B = \frac{1}{m} \sum_{1=1}^{n_m} m_i \cdot \left(\vec{\boldsymbol{r}}^{RG_i}\right)_B \tag{5.53}$$

where the m_i are the masses of the components, and $\left(\vec{\boldsymbol{r}}^{RG_i}\right)_B$ are the position vectors from the body fixed reference point R to the respective i-th element c.g.

The overall inertia at the common c.g. is then [ER1996]

$$\left(\mathbf{I}^G\right)_B = \sum_{i=1}^{n_m} \mathbf{M}_{BB_i}\left(\mathbf{I}^{G_i}\right)_{B_i} \mathbf{M}_{BB_i}^{\intercal} - m_i\left[\left(\vec{\boldsymbol{r}}^{GG_i}\right)_B \times\right]\left[\left(\vec{\boldsymbol{r}}^{GG_i}\right)_B \times\right] \tag{5.54}$$

where

- the $\mathbf{M}_{BB_i}$ describe a rotation from the i-th element's body fixed coordinate frame to the aircraft's body-fixed frame,
- $\left(\mathbf{I}^{G_i}\right)_{B_i}$ is the i-th element's tensor of inertia at its c.g., noted in its own body fixed frame and
- $\left[\left(\vec{\boldsymbol{r}}^{GG_i}\right)_B \times\right]$ is the skew symmetric matrix that would result from a cross-product with the vector from the common c.g. to the element's c.g. according to (5.33).

Elements that are only modeled as point-masses simply have $\left(\mathbf{I}^{G_i}\right)_{B_i} = \mathbf{0}$.

To include the effects of inertia and weight in the equations of motion, the resulting forces and moments at the fix reference point R have to be considered as

$$\left(\vec{\boldsymbol{F}}_G^R\right)_B = \left(\vec{\boldsymbol{F}}_G^G\right)_B = \mathbf{M}_{BO}(\vec{\boldsymbol{g}})_O\, m = \mathbf{M}_{BO}\begin{bmatrix} 0 \\ 0 \\ -gm \end{bmatrix} \tag{5.55}$$

$$\left(\vec{\boldsymbol{M}}_G^R\right)_B = \left(\vec{\boldsymbol{r}}^{RG}\right)_B \times \left(\vec{\boldsymbol{F}}_G^G\right)_B + \left(\vec{\boldsymbol{M}}_G^G\right)_B = \left(\vec{\boldsymbol{r}}^{RG}\right)_B \times \left(\vec{\boldsymbol{F}}_G^G\right)_B \tag{5.56}$$

where the fact is used that the mass properties of a body do not result in a moment at the body's c.g., i.e. $\left(\vec{\boldsymbol{M}}_G^G\right)_B = \vec{\mathbf{0}}$.

5.3.2.4 Actuator Modeling

Actuators in use in RPAS projects are commonly electro-mechanical servo actuators. Based on a similar reasoning as was applied in section 5.3.2.1, considering the actuator as a second order lag element is often a good approximation.

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ K \cdot \omega_0^2 \end{bmatrix} x_{cmd} \tag{5.57}$$

In addition, most actuator controllers are tuned to have no steady-state error and to be well damped, resulting in $K \approx 1$ and $\zeta \approx \frac{1}{\sqrt{2}}$, thus reducing the actuator parameters to be determined to its natural frequency. As in the determination of propulsion models, determining ω_0 from flight data is bound to fail, since it will in general be strongly correlated with the respective damping and control effectiveness parameters.

Above model can be extended with limits on actuator velocity and position, if they are known. Those limits translate directly to limits on the optimization vector in the full discretization approach, since actuator states are directly available as its components. To incorporate further effects would in general necessitate models for aerodynamic hinge moments at the control surfaces, and knowledge of the internal workings of the actuator and its controller, which is again hard to come by.

5.3.2.5 Further External Forces

Depending on the RPAS configuration to be investigated, several other forces and moments may influence its behavior, such as ground reaction forces, or forces appearing during a characteristic start or landing. Although not necessarily important for flight dynamics identification, a complete simulation of the system for engineering purposes may necessitate modeling these aspects as well, in order to e.g. fully test the flight control system from start to landing.

Many of these effects can be modeled using simple mass-spring-damper systems, which exert a force upon the aircraft as soon as some condition is fulfilled. For example,

ground reaction forces for vertical take-off vehicles may be modeled as spring-damper combinations, that exert forces as soon as one (or several) characteristic points of the aircraft pass the local ground plane. In the same manner, a landing in a net, or a catapult-start may be included in the simulation. Spring and damper coefficients can then be tuned by hand to roughly reproduce the true behavior, since in these cases perfectly reproducing reality is often not necessary.

5.3.3 Environment Modeling

In order to fully describe an aircraft's behavior, models for the environment are necessary, too. Fortunately, for RPAS applications, these tend to be very simple.

5.3.3.1 Gravity

Gravity does change both with the location on the WGS84 ellipsoid and the altitude above it, for which there exist very sophisticated models. However, for RPAS applications, due to the short traveling distances and low altitudes, gravity may be assumed to be constant.

5.3.3.2 Atmosphere

Especially aerodynamic forces depend heavily on the atmospheric conditions in the forms of pressure, density, temperature, and wind. The first three may be computed based on the International Standard Atmosphere (ISA), which is a model for their change with altitude [Hol2018b]. Again, some simplifications for RPAS applications are in order: the difference between geopotential and geometric altitude is neglected, and only the lowest atmosphere layer, the troposphere, is considered due to the generally low altitudes of RPAS operations.

International Standard Atmosphere (ISA) temperature is assumed to vary linearly in the Troposphere

$$T(h) = (T_{MSL} + \Delta T) + \gamma_{Tr} h \tag{5.58}$$

$T_{MSL} = 288.15\,\mathrm{K}$ is the norm temperature at mean sea level, ΔT an offset to account for local conditions and $\gamma_{Tr} = -6.5 \times 10^{-3}\,\mathrm{K/m}$ the temperature decrease with altitude. Pressure varies according to

$$p_{stat}(h) = (p_{stat,MSL} + \Delta p_{stat}) \left(1 + \frac{\gamma_{Tr}}{T_{MSL}} h\right)^{\frac{n_{Tr}}{n_{Tr}-1}} \tag{5.59}$$

with pressure at mean sea level $p_{stat,MSL} = 1.013\,25 \times 10^5\,\mathrm{Pa}$, an offset to account for local conditions Δp_{stat}, and the polytropic exponent of the troposphere $n_{Tr} = 1.235$. Density may then be computed from the ideal gas law

$$\rho = \frac{p_{stat}(h)}{T(h) \cdot R} \tag{5.60}$$

with the gas constant for dry air $R = 287.05\,\mathrm{J/kg/K}$.

Dynamic atmosphere influences are combined in the influence of wind $\left(\vec{\boldsymbol{v}}_W^R\right)^E$, which is simply included in the aerodynamic velocity

$$\left(\vec{\boldsymbol{v}}_A^R\right)^E = \left(\vec{\boldsymbol{v}}_K^R\right)^E - \left(\vec{\boldsymbol{v}}_W^R\right)^E \tag{5.61}$$

5.3.4 Aerodynamics

The main goal in aircraft system identification is to find suitable models to describe aerodynamic effects. These models are based on non-dimensional derivatives for forces and moments in the aerodynamic reference frame, leading to expressions of the form

$$\left(\vec{\boldsymbol{F}}_A^A\right)_A = \bar{q}^A S_{ref} \begin{bmatrix} -\mathrm{C}_D \\ \mathrm{C}_Q \\ -\mathrm{C}_L \end{bmatrix} = \frac{\rho}{2} \left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2^2 S_{ref} \begin{bmatrix} -\mathrm{C}_D \\ \mathrm{C}_Q \\ -\mathrm{C}_L \end{bmatrix} \tag{5.62}$$

$$\left(\vec{\boldsymbol{M}}_A^A\right)_B = \frac{\rho}{2} \left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2^2 S_{ref} \begin{bmatrix} b \cdot \mathrm{C}_l \\ \bar{c} \cdot \mathrm{C}_m \\ b \cdot \mathrm{C}_n \end{bmatrix} \tag{5.63}$$

Attention has to be paid to the way how the non-dimensional coefficients are obtained, e.g. some authors use the half-span $\frac{b}{2}$ instead of b in the moment equations.

Above coefficients are then a function of a large number of independent variables. MORELLI and KLEIN give the result of a dimensional analysis, resulting in 13 influencing factors [MK2016, Ch. 3]. Since usually only a limited amount of information is "spread out" over all parameters to be estimated, reducing their number improves the information per parameter ratio. Thus a small number of parameters should always be the goal. This also lies at the heart of the *principle of parsimony*: "Given two models fitted to the same data with nearly equal residual variances, choose the model with the fewest parameters"[MK2016, p. 145].

Coming back to above thirteen influencing factors, common simplifications (A/C mass & inertia are larger than that of the surrounding air; fluid properties change slowly; quasi-steady flow) lead to a dependency mainly on flow conditions, control surface deflections, rotational rates as well as Mach and Reynolds numbers. For RPAS applications, those latter are often neglected, too, which eventually yields [MK2016, Ch. 3]

$$\mathrm{C}_i = \mathrm{C}_i \left(\frac{\left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2}{V_0}, \alpha, \beta, \delta_j, \left(\vec{\boldsymbol{\omega}}_A^{AB}\right)^* \right) \qquad i = D, Q, L, l, m, n \tag{5.64}$$

where δ_j represents control surface deflections and $\left(\vec{\boldsymbol{\omega}}_A^{AB}\right)^*$ are non-dimensional rota-

tional rates [MK2016, Ch. 3]

$$\left(\vec{\boldsymbol{\omega}}_A^{AB}\right)^* = \begin{bmatrix} \frac{b \cdot p_A^{AB}}{2 \cdot \left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2} \\ \frac{\bar{c} \cdot q_A^{AB}}{2 \cdot \left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2} \\ \frac{b \cdot r_A^{AB}}{2 \cdot \left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2} \end{bmatrix} = \begin{bmatrix} \left(p_A^{AB}\right)^* \\ \left(q_A^{AB}\right)^* \\ \left(r_A^{AB}\right)^* \end{bmatrix} \tag{5.65}$$

Some authors also include dependencies on temporal changes of angle of attack and angle of sideslip $\dot{\alpha}^*$ and $\dot{\beta}^*$, but the corresponding parameters are often strongly correlated with body rotational rates, and will be omitted here.

Estimating the full functional dependency of an aerodynamic coefficient on the remaining influencing factors is still close to impossible, which is why these dependencies are parameterized further. The probably most widely used approach is to consider a Taylor series expansion around a reference condition [MK2016, Ch. 3]

$$\begin{aligned} \mathrm{C}_i &= \mathrm{C}_{i0} + \mathrm{C}_{iV} \frac{\left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2}{V_0} + \mathrm{C}_{i\alpha}\alpha + \mathrm{C}_{i\beta}\beta \\ &\quad + \mathrm{C}_{ip}\left(p_A^{AB}\right)^* + \mathrm{C}_{iq}\left(q_A^{AB}\right)^* + \mathrm{C}_{ir}\left(r_A^{AB}\right)^* + \sum_j \mathrm{C}_{i\delta_j}\delta_j \\ &= \begin{bmatrix} 1 & \frac{\left\| \left(\vec{\boldsymbol{v}}_A^A\right)^E \right\|_2}{V_0} & \alpha & \beta & \left(p_A^{AB}\right)^* & \left(q_A^{AB}\right)^* & \left(r_A^{AB}\right)^* & \delta_1 & \cdots \end{bmatrix} \begin{bmatrix} \mathrm{C}_{i0} \\ \mathrm{C}_{iV} \\ \mathrm{C}_{i\alpha} \\ \mathrm{C}_{i\beta} \\ \mathrm{C}_{ip} \\ \mathrm{C}_{iq} \\ \mathrm{C}_{ir} \\ \mathrm{C}_{i\delta_1} \\ \vdots \end{bmatrix} \end{aligned} \tag{5.66}$$

where the $\mathrm{C}_{i\delta_j}\delta_j$ represent the effect of control surface deflections. For a conventional configuration, this would be elevator, aileron and rudder deflections. Power controls, like throttle setting, propeller rotational rate or thrust coefficient $C_T(J)$ may also be included, as they may change the flow condition on the wings, thus influencing aerodynamics. This approach is especially appealing, if the effects of e.g. propeller downwash cannot be modeled adequately [MK2016, Ch. 5.1]. The C_{i0} lumps together the actual coefficient at the reference condition, and the effect of all base values, such as α_0.

Above model can be extended by either strictly following the Taylor series argumentation, and extending the model with selected higher order polynomial terms. Alternatively, regressors may be chosen, which do not follow the strict Taylor series form, but which are intuitively appealing, such as "signed squares" of the form $\mathrm{sign}(\beta)\,\beta^2$. As long as these new regressors contribute with a significantly different shape compared

to the ones already in the model, i.e. as long as they do not introduce correlations, many effects can be modeled in this way.

If an aerodynamic model for the whole flight envelope is desired, several different test points (see section 5.1.1) may be investigated separately, and afterwards combined via interpolation. If the models obtained in this way still do not perform as desired, a next step can be to include even more non-linear models. Possible extensions are to use spline functions or generic lookup tables to make above coefficients of the linear model explicitly dependent on the flow conditions via α and/or β. However, this introduces a large number of new parameters, contradicting the principle of parsimony, and should only be done as a last resort [MK2016, Ch. 3]

Similar Considerations are true for multi-point aerodynamic models: it is possible to implement above model not only once, at the aerodynamic reference point, but several times, e.g. for the main wings, horizontal and vertical tail, and body. This may make the model more versatile, however it also introduces strong correlations between the parameters: for example a pitching motion $\left(q_A^{AB}\right)^* \neq 0$ will change the local angle of attack at the horizontal tail. Then the local lift coefficient at the tail $C_{L\alpha}^{tail}$ together with the respective lever arm, will have exactly the same effect on aircraft motion as the pitch damping coefficient at the wing C_{mq}^{wing} This makes their independent determination impossible.

The foregoing discussion only hints at the myriad of possibilities to formulate aerodynamic models. Finding the most suitable formulation for the system at hand is the most challenging task in aircraft system identification and usually comprises of a lengthy trial and error phase. In the application chapter 6, those final models will be presented in detail, however most of them are implemented on the basis illustrated here, together with as few non-linear extensions as possible.

As with all other forces and moments, the resulting aerodynamic effects have to be translated to the aircraft reference point in body fixed coordinates

$$\left(\vec{\boldsymbol{F}}_A^R\right)_B = \mathbf{M}_{BA}(\alpha,\beta)\left(\vec{\boldsymbol{F}}_A^A\right)_A \tag{5.67}$$

$$\left(\vec{\boldsymbol{M}}_A^R\right)_B = \left(\vec{\boldsymbol{M}}_A^A\right)_B + \left(\vec{\boldsymbol{r}}^{RA}\right)_B \times \mathbf{M}_{BA}(\alpha,\beta)\left(\vec{\boldsymbol{F}}_A^A\right)_A \tag{5.68}$$

5.3.5 Decoupling Longitudinal and Lateral Motion

In flight testing for aircraft system identification it is commonly made use of the fact that lateral and longitudinal aircraft motion are well decoupled. Thus maneuvers to excite characteristic longitudinal (short-period, phugoid) and lateral motion (roll-motion, spiral-motion, Dutch roll) are often conducted and analyzed independently of each other. However, due to slight control inputs in the respective other axis, it might be necessary to have an acceptable model for this other axis, too.

For example, maneuvers to excite the phugoid motion commonly last relatively

long. During this time, it is not uncommon that the pilot needs to use the aileron in order to keep the wings of the aircraft level. During parameter estimation for this maneuver, the lateral dynamics will need to be considered, in order to avoid divergence of the model and measurements.

This coupling may be removed, if instead of the model lateral states, the measured or reconstructed states are used, i.e. instead of propagating the lateral states, their measured counterpart is used as input to the system. The same may be applied for the longitudinal motion. An example separation may be achieved using the following state and input vectors, with the correspondingly separated system equations

$$\begin{aligned} \boldsymbol{x}_{long} &= \begin{bmatrix} \left(u_K^R\right)_B^E & \left(w_K^R\right)_B^E & \left(q_K^{OB}\right)_B & \theta^{BO} & \dot{h}^R \end{bmatrix}^\intercal \\ \boldsymbol{u}_{long} &= \begin{bmatrix} \boldsymbol{u}_\delta^\intercal & \left(v_K^R\right)_B^E & \left(p_K^{OB}\right)_B & \left(r_K^{OB}\right)_B & \phi^{BO} & \psi^{BO} \end{bmatrix}^\intercal \end{aligned} \tag{5.69}$$

$$\begin{aligned} \boldsymbol{x}_{lat} &= \begin{bmatrix} \left(v_K^R\right)_B^E & \left(p_K^{OB}\right)_B & \left(r_K^{OB}\right)_B & \phi^{BO} & \psi^{BO} \end{bmatrix}^\intercal \\ \boldsymbol{u}_{lat} &= \begin{bmatrix} \boldsymbol{u}_\delta^\intercal & \left(u_K^R\right)_B^E & \left(w_K^R\right)_B^E & \left(q_K^{OB}\right)_B & \theta^{BO} & \dot{h}^R \end{bmatrix}^\intercal \end{aligned} \tag{5.70}$$

where $\boldsymbol{u}_\delta$ lumps together the original control surface and power inputs. Above system formulations enable the analyst to treat lateral and longitudinal modeling completely separate [MK2016, Ch. 3] [Jat2015, Ch. 3]. For the final result consistency of the complete model has to be investigated, however. A downside of this is that depending on the quality of the measured/reconstructed quantities, a considerable amount of random error is fed to the system by considering possibly noisy state measurements as inputs.

5.4 Parameter Estimation

The actual parameter estimation step has been discussed in detail in the first chapters of this work. In summary, the cost function considerations in chapter 2.2, are combined with the problem formulations in chapter 4, a model based on the flight mechanic modeling of chapter 5.3, and the implementation considerations in chapter 3. The arising optimization problem may eventually be solved using the methods illustrated in chapter 2.1.

However, one major issue in the application has as of yet not been discussed: The initialization of the algorithm with a suitable initial model parameter guess $\boldsymbol{\theta}_0$. This may either be based on experience, a different data source, or non-iterative estimation methods such as linear least-squares. The focus here will lie on that last possibility, since it is the most versatile, and may easily be adapted to new model structures. Also, in RPAS application, other data-sources are often limited to preliminary design tools, which may not be compatible with the model structure at hand, since Computational

Fluid Dynamics (CFD) models or wind-tunnel campaigns are too costly or not available.

In order to apply linear least-squares methods, first an estimate of the aerodynamic forces and moments needs to be computed from flight test data. Then, the parameters of a *linear* model structure may be determined non-iteratively. Even though the restriction to linear models may seem like a drawback, this formulation for aerodynamic models is most of the time able to cover the most important flight dynamic characteristics of an aircraft. Even if non-linear terms turn out to be necessary, having good initial values for the linear parts available, and initializing the non-linear parameters with zero, is often enough to end up in the region of convergence of the iterative optimization algorithm.

To get started, the aerodynamic forces and moments may be derived from the translational and rotational propagation equations (5.30) and (5.32).

$$\frac{\sum \vec{\boldsymbol{F}}^R}{m} = \left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E + \left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B \times \vec{\boldsymbol{r}}^{RG} + \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{\omega}}_K^{IB} \times \vec{\boldsymbol{r}}^{RG}\right)$$

$$\sum \vec{\boldsymbol{M}}^R = \left(\mathbf{I}^R\right)\left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B + \vec{\boldsymbol{\omega}}_K^{IB} \times \left[\left(\mathbf{I}^R\right)\vec{\boldsymbol{\omega}}_K^{IB}\right] + m\vec{\boldsymbol{r}}^{RG} \times \left[\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB} + \vec{\boldsymbol{\omega}}_K^{IB} \times \left(\vec{\boldsymbol{v}}_K^R\right)^E\right]$$

The values on the right hand side are usually measured directly (velocity, attitude angles, rotational rates) or may be derived from measured quantities (temporal change in velocity $\left(\dot{\vec{\boldsymbol{v}}}_K^R\right)^{EB}$ may be determined from acceleration measurements and gravity).

The only quantity that is usually not measured is angular acceleration $\left(\dot{\vec{\boldsymbol{\omega}}}_K^{IB}\right)^B$, which needs to be determined numerically from body rotational rates $\left(\vec{\boldsymbol{\omega}}_K^{IB}\right)^B$. Since the measurement of those is quite noisy, direct application of finite differences often yields unusable results. Better estimates for rotational acceleration may be based on one of the following approaches [MK2016, Ch. 11]

- Fit a smoothing spline to the rotational rate measurements, to get rid of the noise; this smoothing spline may then be differentiated analytically.
- use (smoothing) local polynomial approximation to rid the data of noise; then use the derivative of the polynomial as estimate for the angular acceleration
- Translate the rotational rate measurements to the frequency domain; then high frequency content (dominated by noise) larger than a break frequency to be chosen manually may be ignored; The remaining Fourier coefficients can then be used to compute a smoothed estimate of the actual rotational rates, which essentially is a sum of sines and cosines of different frequency. Then an analytic time derivative may be computed, and used as smoothed rotational acceleration, see [MK2016, Ch. 11] for details on this "global Fourier smoothing".

Although above approaches provide superior results compared to direct finite differences, the inaccuracies introduced into the overall process via the numerical deter-

mination of angular accelerations often limit the obtainable quality of the parameter estimates via linear least-squares. Thus it makes sense to use these quantities only as initial values for iterative methods, especially in RPAS applications where data is collected using low-cost sensors with high noise levels.

After determining all quantities on the right hand sides of above propagation equations, together with the inertial characteristics of the aircraft under investigation, the sums of forces and moments are available. Then, known forces and moments originating in other subsystems (propulsion, weight and balance) need to be determined, using flight data and/or reconstruction results (see section 5.2 for details) and the corresponding subsystem models. Eventually, those subsystem contributions are subtracted from the total forces and moments to only retain the aerodynamic effects.

From these, together with geometric data and the dynamic pressure at the aerodynamic reference point, the aerodynamic coefficients C_D, C_Q, C_L, C_l, C_m, and C_n may be obtained from their definitions in equations (5.62) and (5.63). These can then be used as "measurements" in a linear least-squares problem, where the regressors are dependent on the respective model in use. Example 5.1 illustrates the resulting ordinary least-squares problem for the roll moment coefficient.

Since the regressors are built from measured, noisy data, the resulting estimates are biased, and inefficient [MK2016, Ch. 5.1]. The severity of this increases with the noise level of the data, however the estimates are often good enough as an initial guess.

Initial states at the beginning of each maneuver can easily be obtained from averaging the first few measured samples. In the approaches based on full discretization, as illustrated in chapter 4, an initial guess for the state and output trajectory needs to be provided, too. The simplest approach would be, to use the initial guess for the model parameters $\boldsymbol{\theta}_0$, together with the initial conditions $\boldsymbol{x}_0$, and propagate the dynamic system in time. To give the full discretization approach a "running start", one can also choose to use measured or reconstructed states as initial guess for the state trajectory. Thus the outputs will already be in close vicinity of the measurements at the beginning of the optimization.

Unfortunately, the initial model parameter guesses, determined in the above fashion, may sometimes turn out to be too inaccurate. Symptoms of this can be either divergence of the state propagation (if single-shooting is used), or divergence of the optimization algorithm as a whole. Possible remedies are then

- The use of Filter Error Methods (FEMs), which numerically stabilize the state estimate and may thus be used to overcome intermediate divergence of the state propagation. If the complexity of a complete stochastic treatment is to be avoided, this may only be applied initially, to improve upon the initial model parameter guess so far as to enable meaningful application of the single-shooting approach.
- The use of the full-discretization approach, which is numerically more stable. If additionally, measured states are used as initial guess in the optimization vector,

Example 5.1: Least-squares Roll Moment Estimation
Assuming values for the roll moment coefficient C_l at sampling instants $k = 0 \dots N-1$ are computed via the approach illustrated in section 5.4. Further assuming a model structure of the form

$$C_l = C_{l0} + C_{l\beta}\tilde{\beta} + C_{lp}\left(p_A^{AB}\right)^* + C_{lr}\left(r_A^{AB}\right)^* + C_{l\xi}\tilde{\xi} + C_{l\zeta}\tilde{\zeta}$$

The primary purpose of C_{l0} is not to cover possible aircraft asymmetry, but to cover biases introduced in the computation of C_l. Also, the constant (trim) parts are removed from all regressors (denoted by the $\tilde{\square}$), as already mentioned in section 2.2.4.8, to avoid correlation with the bias term C_{l0}. The maneuver is assumed to start in a trimmed condition, i.e. the rotational rate terms do not have a constant part.
The resulting ordinary least-squares problem can then be solved using

$$\underbrace{\begin{bmatrix} C_{l,0} \\ \vdots \\ C_{l,N-1} \end{bmatrix}}_{=\boldsymbol{z}} = \underbrace{\begin{bmatrix} 1 & \tilde{\beta}_0 & \left(p_A^{AB}\right)^*_0 & \left(r_A^{AB}\right)^*_0 & \tilde{\xi}_0 & \tilde{\zeta}_0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \tilde{\beta}_{N-1} & \left(p_A^{AB}\right)^*_{N-1} & \left(r_A^{AB}\right)^*_{N-1} & \tilde{\xi}_{N-1} & \tilde{\zeta}_{N-1} \end{bmatrix}}_{=\mathbf{X}} \cdot \underbrace{\begin{bmatrix} C_{l0} \\ C_{l\beta} \\ C_{lp} \\ C_{lr} \\ C_{l\xi} \\ C_{l\zeta} \end{bmatrix}}_{=\boldsymbol{\theta}_0} + \boldsymbol{v}$$

$$\hat{\boldsymbol{\theta}}_0 = \left(\mathbf{X}^\intercal \mathbf{X}\right)^{-1} \mathbf{X}^\intercal \boldsymbol{z}$$

the algorithm exhibits a relatively strong robustness against unfavorable initial model parameters $\boldsymbol{\theta}_0$.

- The approaches illustrated in section 3.8 may be used to improve upon the initial model parameter guess to enable meaningful state propagation in the first place. In the applications to be illustrated in chapter 6, those were used exclusively, together with initial model parameters set to zero; still meaningful results could be obtained. Unfortunately, as is often the case in system identification, this may not be enough in all possible applications. However, combining the improvement ideas of section 3.8 with the approach illustrated in this section, a very robust way to obtain initial parameter guesses for flight vehicle system identification may be constructed.

5.5 Model Validation

As WALTER puts it: "It would be naive to consider that [$\hat{\boldsymbol{\theta}}$] resulting from the optimization procedure corresponds to the only model worthy of consideration." [WP1997, p. 5]. No model will cover all of the effects present in reality, thus a thorough mod-

el validation should be performed after any estimation process. Usually, two main aspects are investigated: the parameter uncertainty as obtained from the estimation algorithm, and the performance of the model in reproducing measured outputs.

One of the biggest problems in model validation is nicely illustrated by WALTER, in whose book the respective chapter is called "Falsification" [WP1997, Ch. 7]: In general it is seldom the case, that the analyst can definitely confirm all of the aspects of the previously developed model. More often than not, the best thing one can do is to *not reject* the model at hand. Some of the possible approaches, which have found to work well in RPAS applications, are illustrated next.

5.5.1 Engineering Judgement

The simplest way of checking the validity of a model is to ensure, that the estimated parameter values are within reasonable bounds. In many cases, even though no hard boundaries for a parameter can be stated, some intuitive interpretation is possible. Thus, at least the order of magnitude or the sign of a parameter can be checked immediately after the estimation has finished [WP1997, Ch. 7].

A second aspect is the model's predictive capability, i.e. its performance when exposed to a data-set that was not part of the estimation process (different input shape, different operation condition, ...). This test is passed, if the outputs of the model for this complementary data-set are still "close enough" to the measurements [WP1997, Ch. 7]. According to WALTER this test can also be used to discriminate between different model complexities: very complex models, with a large number of parameters, often perform very badly for complementary data-sets [WP1997, Ch. 7]. This can happen due to interactions between the parameters and their consequences, that have not been excited in the original data-set. With a growing number of parameters, the number of possibly adverse interactions naturally grows, too.

Next, linear approximations in some representative operating points can give useful insights. Even if the exact frequencies, damping ratios, and modal shapes of the eigenmotions may not be known, it can at least be checked, that the identified model exhibits the usual aircraft characteristics (short period, dutch roll, roll motion, ...). If any of those modes do not appear, this might not be a reason to reject the model, but may highlight room for improvement.

Lastly, the benefit of simply "taking the model for a ride" should not be underestimated: especially for system identification for aerial vehicles, a lot can be learned by connecting a real-time version of the model with a joystick and a visualization system. After only a few attempts, the analyst usually gets a good "feeling" for the system, i.e. can immediately tell, if the identified model resembles an aircraft or not. Also, it is relatively easy to bring the model into extreme conditions, to check that the formulation, together with the estimated parameter values, does not break down. Plus, it is fun.

5.5.2 Residual Analysis

Investigating the properties of the remaining residuals can often lead to important insights into the validity of the assumptions made during the estimation process. Especially the assumption of "white, Gaussian" noise can be checked by plotting the residuals' auto-correlation functions and histograms. More mathematically rigorous testing is based on the following

- *Testing for Normality*
  Formal tests for normality of the residuals are based on histogram plots, or on an empirical density, which are then compared to the nominal values of a Gaussian probability density function (pdf) [MK2016, Ch. 5] [WP1997, Ch. 7.2].
- *Testing for Independence*
  Formal tests for independence are based on empirical auto-correlation functions, and comparing their values with thresholds to determine sufficient "whiteness" [WP1997, Ch. 7.2] [MK2016, Ch. 5]

A problem in RPAS estimation is that model quality is seldom good enough to capture all deterministic effects. Then, even for acceptable results, the remaining residuals often are neither perfectly "white" nor "Gaussian". Nevertheless, inspecting them is always a good idea, since they may contain hints on how to improve a model: by investigating any underlying, deterministic part, and visually comparing it to the available regressors, conclusions may be drawn on how to advantageously extend the model formulation.

5.5.3 Inverse Simulation

Another potential approach for model validation is based on the idea of having a feedback controller, together with the identified plant, track the *measured* trajectory. JATEGAONKAR states, that a proportional integral controller is oftentimes well suited for this task. The necessary control effort can then be used as indicator of model quality, i.e. if only slight additional control action is necessary, the model is able to reproduce the measurements sufficiently well [Jat2015, Ch. 11].

The illustrated optimal control view on system identification allows for an intuitive formulation of the above problem in the presented stochastic estimation framework. The above problem may be formulated using a system model of the form

$$\begin{bmatrix} \boldsymbol{x}_{k+1} \\ \Delta\boldsymbol{u}_{k+1} \end{bmatrix} = \begin{bmatrix} {}^{d}\mathbf{f}_k\left[\boldsymbol{x}_k, \boldsymbol{u}_k + \Delta\boldsymbol{u}_k, \hat{\boldsymbol{\theta}}\right] \\ \Delta\boldsymbol{u}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{w}_k^{\boldsymbol{u}} \end{bmatrix} \qquad k = 0, \ldots, \bar{N} \tag{5.71}$$

for the fixed, estimated $\hat{\boldsymbol{\theta}}$, together with the original output equation and the cost functions discussed at several places in this thesis. If the process noise covariance is then

chosen to be (together with suitable boundaries on $\Delta\boldsymbol{u}$)

$$\mathrm{Cov}\left[\left[\begin{array}{c}\boldsymbol{w}_k\\ \boldsymbol{w}_k^u\end{array}\right]\right] = \left[\begin{array}{cc}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{Q}^u\end{array}\right] \tag{5.72}$$

a problem arises that considers the model states as deterministic, but models the additional control effort, which is necessary to match the outputs, as random walk. This corresponds to a stochastic model formulation, which may be solved using full discretization as in the problem formulation "case IV" of section 4.4.

Thus choosing suitable gains for a feedback loop is replaced by finding a suitable $\mathbf{Q}^u$, which allows the system to track the outputs well. However, almost no additional implementation effort is necessary, since the tools for parameter estimation in a stochastic systems are available anyways. Furthermore, some more freedom may be provided to the estimation algorithm by using a non-zero process noise covariance for $\boldsymbol{w}$, which turns the task into a state estimation problem with fixed model parameters $\hat{\boldsymbol{\theta}}$. Also the random walk assumption for the additional control input may be revised, and more complicated approaches implemented. For example a low-pass filter may be used to limit the bandwidth of $\Delta\boldsymbol{u}$. Still the same reasoning may be applied: the "quantity" of additional control effort and process noise is used to determine, if a model structure and the parameter estimates $\hat{\boldsymbol{\theta}}$ are adequate. This shows, that using optimal control for parameter estimation, as is advertised in this thesis, has another big advantage: it naturally includes model validation formulations, which otherwise would need to be implemented separately.

Of course, the above problem may also be solved using an iterative state estimation algorithm together with a single shooting approach ("case III" in section 4.3). However, then it will be difficult to enforce boundaries on $\Delta\boldsymbol{u}$, which may result in unfeasible additional control effort $\Delta\boldsymbol{u}$.

How insidious Nature is when one is trying to get at it experimentally

Albert Einstein, 1915 [Föl2015]

6

Application Examples

The following sections show applications of the algorithms developed in this thesis. First, two examples illustrate the superior performance of the novel full discretization approach when applied to unstable aircraft models. Both examples are based on simulation results: the first is a de-stabilized short period approximation of the DeHavilland DHC-2. The second is a linear longitudinal model of the F16 with aft c.g.

The following two examples show applications in the presence of process noise. The first covers the lateral directional motion of the DeHavilland DHC-2. The second example is based on real flight test data of the DLR research aircraft HFB 320.

The last two examples are chosen to illustrate the full capabilities of the algorithms developed here. Real flight test data, collected with a model aircraft using solely low-cost sensors, is analyzed, and a full 6-degree of freedom (DOF) non-linear simulation model is built in a first step. The capability of the full discretization approach to treat stochastic systems is used to estimate wind conditions along with the aerodynamic parameters. In a second step, a slight reformulation of the model enables the presented algorithms to perform an inverse simulation for model validation purposes without further implementation efforts.

Usually, the Non-Linear Programming (NLP) solver IPOPT will be used to obtain a solution to the underlying optimization problem. Only in some cases SNOPT showed superior performance. If IPOPT was used, it is noteworthy that it seemed to yield better results, if the initial conditions $\boldsymbol{x}_0$ were estimated along with the model parameters. This "additional freedom" enabled smoother convergence to meaningful results, even in cases where the $\boldsymbol{x}_0$ were perfectly known.

All computations were performed on a consumer PC with an 8^{th} generation Intel i5 CPU using MATLAB 2016b.

6.1 Unstable Aircraft Identification

In [Jat2015] the complete ninth chapter is dedicated to unstable aircraft identification. The major problem with parameter estimation for unstable systems is twofold:

Firstly, in order to perform any sort of experiment, an automatic control system to artificially stabilize the plant is necessary. This inevitably introduces strong correlations between input and output quantities, which makes the problem of data-collinearity very prominent [Jat2015, Ch. 9]. Thus certain parameters cannot be estimated independently.

In order to obtain informative data suitable for parameter estimation, it is usually necessary to excite the characteristic motion of the system (see section 5.1.2 on input design). Again, this is prohibited by the automatic control system, which necessarily alters the system characteristics and may even hide parts of the dynamics.

The second major problem is of methodological nature: the standard approach as illustrated in section 4.1 uses a single shooting transcription method. Thus, even for small perturbations around the true trajectory, an unstable system will necessarily diverge. This divergence may pose serious problems for a classical single shooting formulation, especially if the divergence is fast enough as to result in numerical problems.

The simplest way to overcome this problem is to estimate the parameters of the aircraft in a closed loop with the control system. However, the exact internal workings of the controller need to be known for this approach. Furthermore, correlations between estimated parameters may be increased in this way.

In [Jat2015] different other approaches are illustrated, most of which use some sort of artificial stabilization (either by explicitly introducing an artificial stabilization or by using a Kalman filter), or are based on equation error methods, where no integration needs to be performed. Only on a side-note JATEGAONKAR mentions, that another approach of overcoming the issue of a divergent system is to use a multiple shooting method [Jat2015, Ch. 9]. Then, the problems can be kept at bay, if the segments are kept small enough.

Here, it will now be illustrated, how full discretization (as an extension to multiple shooting) essentially overcomes this problem without additional programming effort. Thus the original system needs not be altered or artificially stabilized. The standard algorithms as are applied for stable systems may directly be used.

Another advantage of using a full-discretization approach manifests itself in the computation of uncertainties: The problem of a divergent system is not only restricted to the state equations, but the sensitivity equations are unstable, too, since they are based on a linearization of the dynamic system under investigation. Thus solving the sensitivity equations in order to determine estimated parameter covariances suffers from the same problems of divergence and possible numerical inaccuracies. However, the novel covariance estimate of section 3.5.3 offers a solution here: the proposed ana-

lytic method of determining a null-space basis of the constraint Jacobian may already alleviate the problem, since it solves the sensitivity equations in one shot, rather than iteratively. Alternatively, the numerical approach via the Singular Value Decomposition (SVD) is independent of the stability of the system. Thus the analyst is able to compute uncertainty estimates in the same manner as for stable systems.

6.1.1 Unstable DeHavilland DHC-2 Short Period

The first example to be discussed is taken from [Jat2015, Ch. 9], where a short-period approximation of the de Havilland DHC-2 "Beaver" is artificially made unstable.

6.1.1.1 Model Formulation

The system formulation is

$$\begin{aligned} \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} Z_w & u_0 + Z_w \\ M_w & M_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} Z_\eta \\ M_\eta \end{bmatrix} \eta \\ \dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} \end{aligned} \tag{6.1}$$

where u_0 is the body-x component of the trim-velocity, w is the body-z velocity component, q describes pitch rate and η elevator deflection. The parameters Z_w, Z_q, and Z_η describe the influences of aforementioned signals on the acceleration in body-z direction, whereas M_w, M_q, and M_η quantify their influence on the pitching acceleration.

The outputs of the system are

$$\begin{aligned} \begin{bmatrix} w \\ q \\ a_z \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ Z_w & Z_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ Z_\eta \end{bmatrix} \eta \\ \boldsymbol{y} &= \mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u} \end{aligned} \tag{6.2}$$

where in addition to w and q the acceleration in body-z direction a_z is assumed to be measured.

6.1.1.2 Data Gathering

The simulated measurement data is directly taken from the supplementary material of [Jat2015] in order to enable a direct comparison. The data had been generated using a proportional controller in order to stabilize the system for simulated flight tests. This controller puts the eigenvalues of the closed loop system approximately where the original, stable short period approximation of the DHC-2 was. The resulting elevator deflection, when using a multi-sine as reference command, is shown in the bottom plot of figure 6.1. This is then used for parameter estimation of the (unstable) bare-airframe.

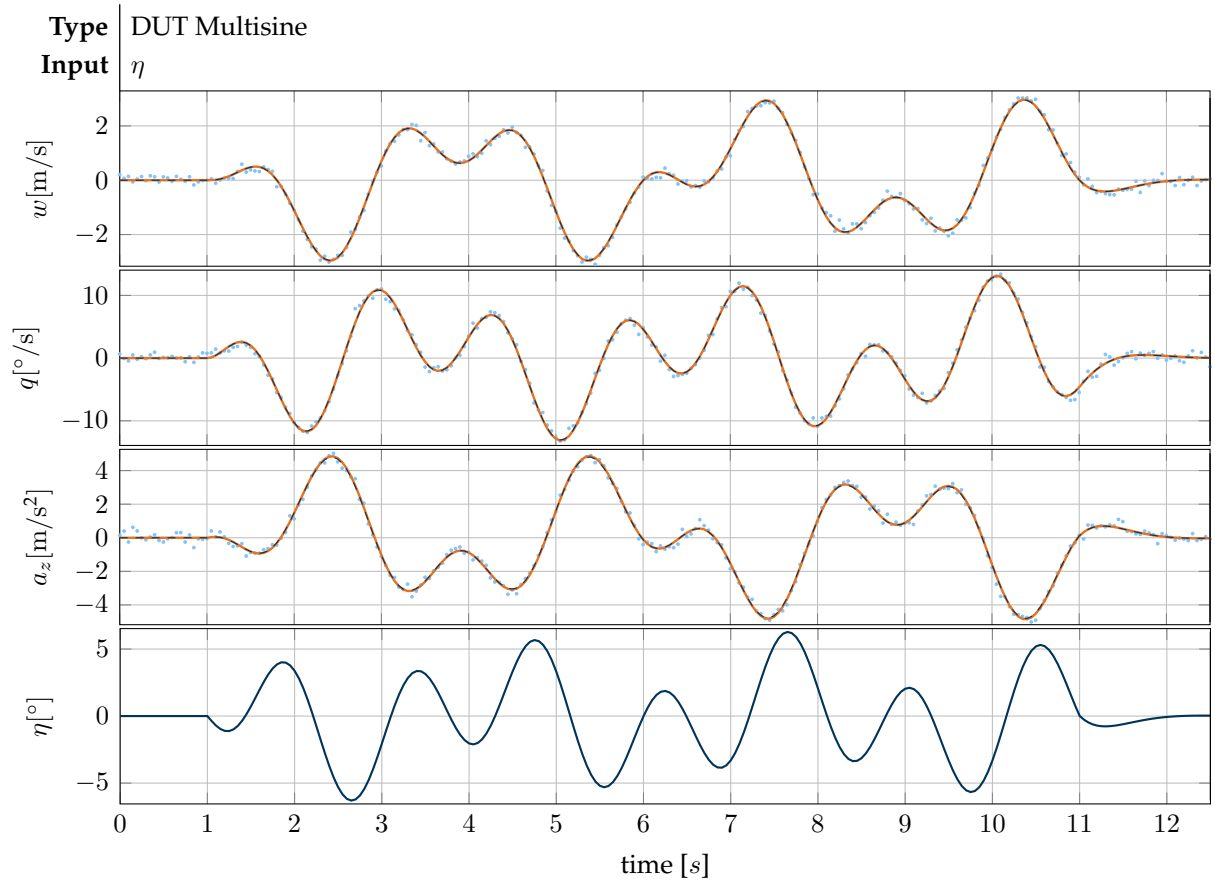


Figure 6.1: *DUT multisine for unstable DHC-2 short period estimation; measurements $\boldsymbol{z}$ (·), estimated $\hat{\boldsymbol{y}}$ (——) and true outputs $\boldsymbol{y}$ (- - -); inputs $\boldsymbol{u}$ (——) are shown in the bottom plot*

In contrast to the example in [Jat2015], a small amount of measurement noise is added to the outputs, since the algorithm as used here cannot cope with "perfect" data: The estimated residual covariance matrix would be zero and could not be inverted in the cost function formulation. For realistic cases, this does not pose a problem, since perfect outputs are never achieved. However, for academic examples as this one, a small amount of white, Gaussian noise needs to be added, which then also fulfills the statistical assumptions about the problem. Here, a signal to noise ratio of 1:50 was assumed, in order to keep close to the original, perfect data.

The resulting measurements, as used in the estimation are shown in the three first plots of figure 6.1.

6.1.1.3 Results

As in [Jat2015], the trim speed $u_0 = 44.57\mathrm{m/s}$ was assumed to be given and the value of Z_q was fixed at its nominal value, since otherwise too large correlations would prohibit a meaningful estimation. Especially the strong correlation with Z_η posed a problem. The initial conditions $\boldsymbol{x}_0$ were estimated along with the model parameters.

Measurements of all states were available, thus the initial parameter guesses could be set to zero, before applying the equation-error based method to improve upon them,

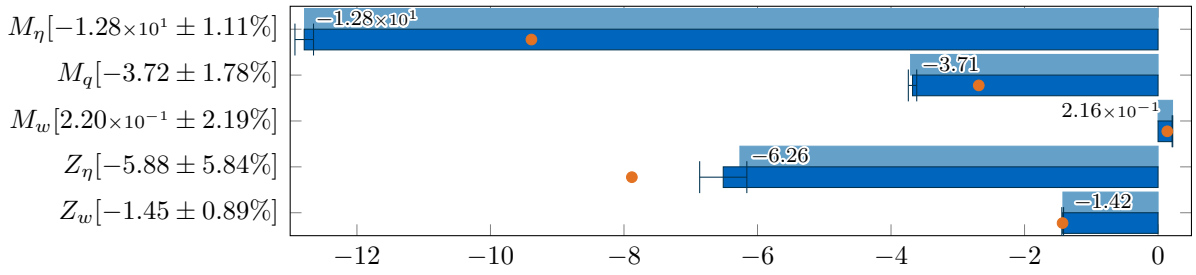


Figure 6.2: *true (* ▬ *) and estimated (* ▬ *) parameter values with estimated* 1σ *bound and initial guesses after refinement (* ● *); numerical values of the estimated parameters and standard deviations are given as lables, true parameter values are written on top of the respective bar*

as illustrated in section 3.8.

The output match that was obtained by estimating the remaining, free parameters using the full discretization approach (see section 4.2 for details) can be seen in figure 6.1. The true and estimated outputs show a perfect match: they cannot be visually discerned.

Figure 6.2 shows a comparison between estimated, and true parameter values. Additionally, the initial parameter guesses after applying the refinement approach are illustrated as dots. The refinement step already gives the right tendencies without any prior knowledge. The overall algorithm is then able to almost perfectly estimate the parameter values despite the original absence of a meaningful initial parameter guess. All estimated standard deviations are very small, below $6\,\%$. Since this is a simulated example, a comparison with the true parameter values is possible, which also shows an almost perfect agreement between the estimated values (y-labels in figure 6.2) and their true counterparts (atop the respective bars in figure 6.2). The eigenvalues of the estimated system (6.9360×10^{-1} and -5.7997) agree to within 0.5% with the true eigenvalues (6.9343×10^{-1} and -5.8250) further underlining the close to perfect match.

The only downside of the results obtained here, are the exceedingly large correlations between the parameters pertaining to the pitch rate equation, as shown in figure 6.3. However, this merely illustrates one of the big problems with unstable aircraft identification: the proportional controller that was used to generate the example dataset makes it difficult to independently estimate the pitching moment coefficients. Since this example is intended to illustrate the capability to treat unstable systems, no further measures are taken to try to reduce these correlations, as this would entail collecting more informative data.

Since fixing Z_q at its *nominal* value is not realistic, a test was performed with a value that was $20\,\%$ off. The output match and parameter characteristics do not change significantly, merely a different estimated value of Z_η results. The algorithm thus accounts for the altered Z_q through adjusting Z_η. This verifies the strong correlation between the two, which prohibited meaningful results in the first place: virtually the same results

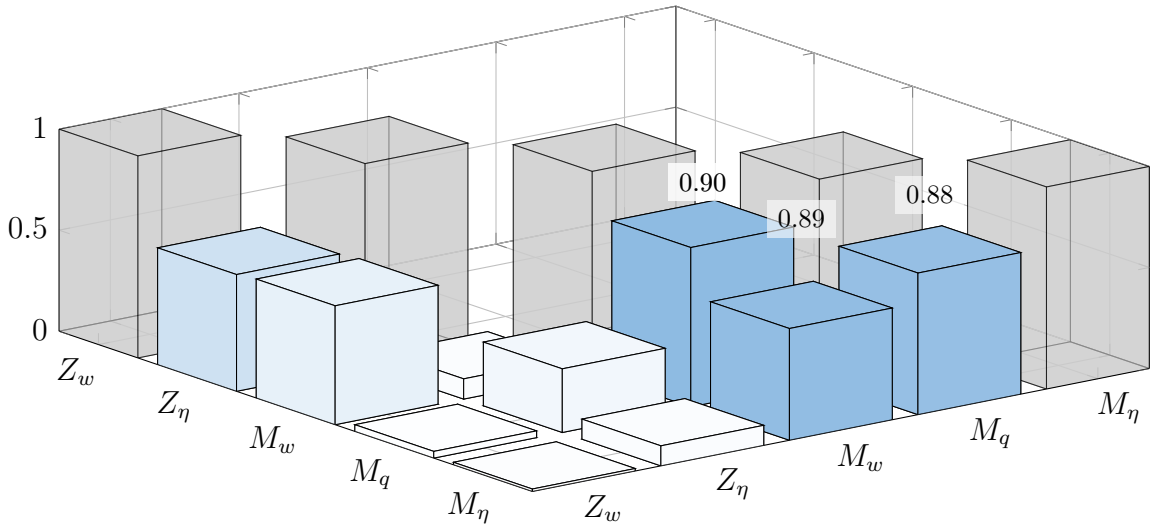


Figure 6.3: *Absolute values of the correlation matrix; largest absolute values are printed atop the respective bar*

are obtained although Z_q is fixed at a largely different value.

Above presented results are the same, as were obtained by JATEGAONKAR using approaches especially tailored to unstable system identification [Jat2015, Ch. 9]. The big difference here is that no special provisions were necessary to treat this system. Merely convergence of the algorithm took somewhat longer compared to simple examples using stable systems, where about $8\,\mathrm{s}$ is still considered acceptable. Also, a little more effort was necessary to find settings of the optimization algorithm that enabled this convergence (scaling, tolerances etc.). Other than that, no additional implementation of special methods was necessary.

6.1.2 F16 Longitudinal Motion

A second example is based on an unstable, linear model of the longitudinal motion of an F16. Depending on the location of the c.g., the F16 model exhibits stable, or unstable longitudinal characteristics.

6.1.2.1 Model Formulation

The model which is used here, is based on a linearization of a non-linear simulation model with a c.g. position at $1.2075\mathrm{m}$ from the aircraft reference point. It is taken from [Lei], which in turn is based on [SL2003, NOG+1979].

The model is formulated in the standard manner for linear, longitudinal systems

$$\begin{bmatrix} \Delta\dot{V} \\ \Delta\dot{\gamma} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \end{bmatrix} = \begin{bmatrix} X_V & -g\cos\gamma_0 & X_\alpha - g\cos\gamma_0 & X_q \\ -Z_V & \frac{g}{V_0}\sin\gamma_0 & \frac{g}{V_0}\sin\gamma_0 - Z_\alpha & -Z_q \\ Z_V & -\frac{g}{V_0}\sin\gamma_0 & Z_\alpha - \frac{g}{V_0}\sin\gamma_0 & Z_q + 1 \\ M_V & 0 & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta\gamma \\ \Delta\alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_\eta & X_{\delta_T} \\ -Z_\eta & -Z_{\delta_T} \\ Z_\eta & Z_{\delta_T} \\ M_\eta & M_{\delta_T} \end{bmatrix} \begin{bmatrix} \Delta\eta \\ \Delta\delta_T \end{bmatrix} \tag{6.3}$$

with states representing velocity V, climb angle γ, angle of attack α and pitch rate q. The inputs are elevator deflection η and throttle setting δ_T. The dimensional coefficients, relating the states to the acceleration in body-x and body-z direction X, and Z and the pitching acceleration M are treated as the parameters of the system. The system is then

$$\Delta\dot{\boldsymbol{x}} = \mathbf{A}\Delta\boldsymbol{x} + \mathbf{B}\Delta\boldsymbol{u} = \mathbf{A}(\boldsymbol{x} - \boldsymbol{x}_0) + \mathbf{B}(\boldsymbol{u} - \boldsymbol{u}_0) \tag{6.4}$$

$$\boldsymbol{y} = \mathbf{C}(\Delta\boldsymbol{x} + \boldsymbol{x}_0) \tag{6.5}$$

where Δ indicates deviations from the known trim values.The trim conditions $\gamma_0 = 0°$, $V_0 = 160.27\,\mathrm{m/s}$, $\alpha_0 = 4°$, trim inputs $\eta_0 = -0.0076$, $\delta_{T,0} = 0.2$ as well as the gravitational acceleration $g = 9.806\,65\,\mathrm{m/s^2}$ are given constants of the problem. Also, the pitching moment due to throttle inputs and due to velocity changes are zero, i.e. $M_V = M_{\delta_T} = 0$, and are thus not considered as parameters.

The matrix $\mathbf{C}$ is used for unit conversion to make interpretation easier.

6.1.2.2 Data Gathering

In order to conduct simulated flight tests, the model needs a stabilizing feedback controller. Based on the work conducted in [Lei], a Linear Quadratic Regulator (LQR) approach with full state feedback was chosen to achieve this. It is designed to follow γ and V commands. For stationary accuracy, the system is augmented with two integral error states e_γ and e_V and the reference command input $\boldsymbol{r}$

$$\begin{bmatrix} \dot{e}_V \\ \dot{e}_\gamma \\ \Delta\dot{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & & \mathbf{A} \end{bmatrix} \begin{bmatrix} e_V \\ e_\gamma \\ \Delta\boldsymbol{x} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix} \Delta\boldsymbol{u} + \begin{bmatrix} -\mathbf{I}_2 \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} V_{cmd} - V_0 \\ \gamma_{cmd} - \gamma_0 \end{bmatrix} \tag{6.6}$$

$$\Delta\dot{\boldsymbol{x}}_a = \mathbf{A}_a\Delta\boldsymbol{x}_a + \mathbf{B}_a\Delta\boldsymbol{u} + \mathbf{B}_r\Delta\boldsymbol{r} \tag{6.7}$$

Then the LQR gain matrix is computed based on the solution to the following quadratic Riccati equation [SL2003, Ch. 5]

$$\mathbf{0} = \mathbf{A}_a^\intercal\mathbf{P} + \mathbf{P}\mathbf{A}_a + \mathbf{Q} - \mathbf{P}\mathbf{B}_a\mathbf{R}^{-1}\mathbf{B}_a^\intercal\mathbf{P} \tag{6.8}$$

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}_a^\intercal\mathbf{P} \tag{6.9}$$

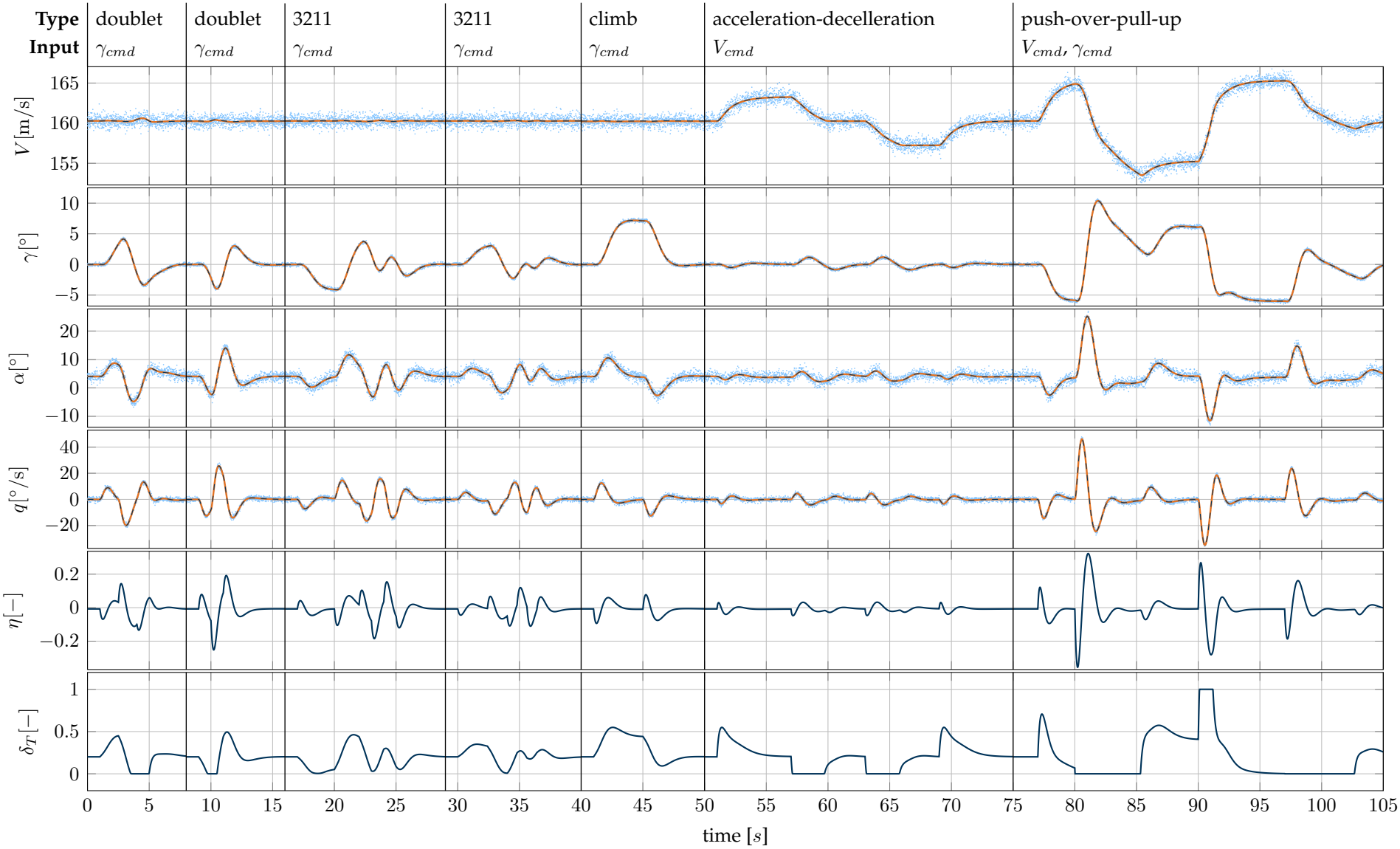


Figure 6.4: *Input-Output data for unstable F16 parameter estimation; measurements $\boldsymbol{z}$ ( ), estimated $\hat{\boldsymbol{y}}$ ( ) and true outputs $\boldsymbol{y}$ ( ); inputs $\boldsymbol{u}$ ( ) are shown in the bottom two plots*

The weighting matrices for the states $\mathbf{Q}$ and inputs $\mathbf{R}$ were chosen as

$$\mathbf{Q} = \begin{bmatrix} 0.8 & & & & & & \\ & 50 & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$$

The resulting stable, closed loop dynamics are then

$$\boldsymbol{u} = -\mathbf{K}\Delta\boldsymbol{x}_a + \boldsymbol{u}_0 \tag{6.10}$$

$$\Delta\dot{\boldsymbol{x}}_a = (\mathbf{A}_a - \mathbf{B}_a\mathbf{K})\,\Delta\boldsymbol{x}_a + \mathbf{B}_r\Delta\boldsymbol{r} \tag{6.11}$$

The focus, when designing this controller, was not laid on very good performance, or real-life applicability. It was merely intended to provide a stable plant, which may be used for simulated flight tests.

Outputs of the closed loop system consist of the original outputs, together with the control inputs to the original system

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{C} \\ \mathbf{0} & -\mathbf{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{e} \\ \Delta\boldsymbol{x} \end{bmatrix} + \begin{bmatrix} \mathbf{C}\boldsymbol{x}_0 \\ \boldsymbol{u}_0 \end{bmatrix} \tag{6.12}$$

the first part $\boldsymbol{y}$ of above output equation will be used as outputs for the unstable system identification, whereas the second part will serve as inputs to the plant whose parameters are to be estimated.

With this closed loop stable system, different simulated maneuvers are conducted, in order to gather data for parameter estimation. Although the controller was designed purely based on the linear system description, non-linear input limits were implemented during simulation, to enforce throttle settings in the range $[0, 1]$.

The resulting outputs $\boldsymbol{y}$ and inputs to the original system $\boldsymbol{u}$ can be seen in figure 6.4. The types of maneuvers are indicated at the top of the figure, together with the main reference command. They consist of four multi-step maneuvers in γ_{cmd} with different step widths and step sequences, one climb maneuver, one acceleration-deceleration and a push-over-pull-up maneuver. White, Gaussian noise with covariance matrix

$$\mathrm{Cov}[\boldsymbol{v}] = \begin{bmatrix} (0.5\,\mathrm{m/s})^2 & & & \\ & (0.15^\circ)^2 & & \\ & & (1.0^\circ)^2 & \\ & & & (1.0\,^\circ/\mathrm{s})^2 \end{bmatrix} \tag{6.13}$$

was added to the outputs to eventually obtain the simulated measurements $\boldsymbol{z}$.

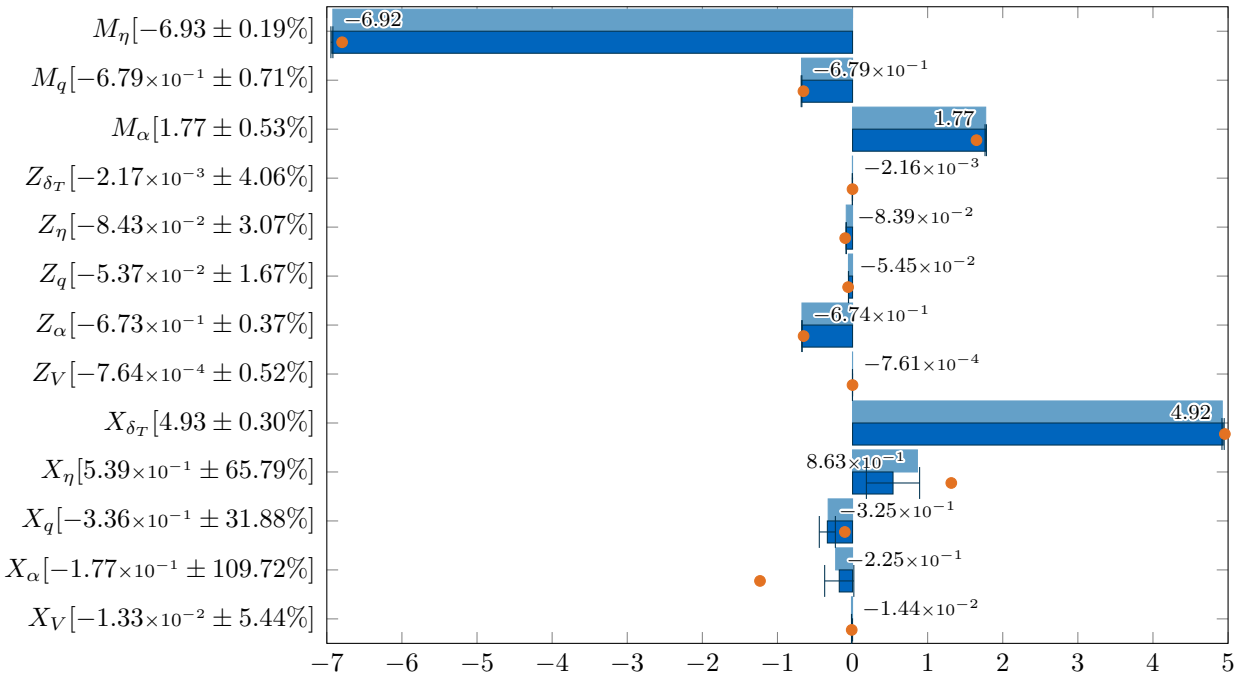


Figure 6.5: *True ( ) and estimated ( ) parameter values with estimated 1σ bound and initial guesses after refinement ( ); numerical values of the estimated parameters and standard deviations are given as lables, true parameter values are written on top of the respective bar*

6.1.2.3 Results

Then the full discretization estimation approach, summed up in section 4.2, was applied. Model parameter estimates and outputs have been obtained after roughly $50\,\mathrm{s}$, numerically determining the null space of the constraint Jacobian took another $10\,\mathrm{s}$. The resulting output match is also part of figure 6.4, and again no difference between true and estimated outputs can be seen.

As in the foregoing example, measurements of all states were available. Thus all initial model parameter guesses could be set to zero, before using the equation error based approach of section 3.8 for initial guess refinement. The actual state measurements used in this step were first smoothed using a moving average filter. Figure 6.5 shows a comparison between true parameters, estimated parameters with their estimated standard deviation, and initial parameter guesses after the refinement step. The initial refinement already finds parameter values, that are in close proximity to the fi-

Table 6.1: *Comparison of estimated and true eigenvalues for the unstable F16 example*

λ_{true}	λ_{est}	rel. difference [%]
-1.972	-1.969	0.1450
6.313×10^{-1}	6.313×10^{-1}	0.0003
$-1.301\times10^{-2} \pm i\ 1.023\times10^{-1}$	$-1.310\times10^{-2} \pm i\ 1.024\times10^{-1}$	0.1656

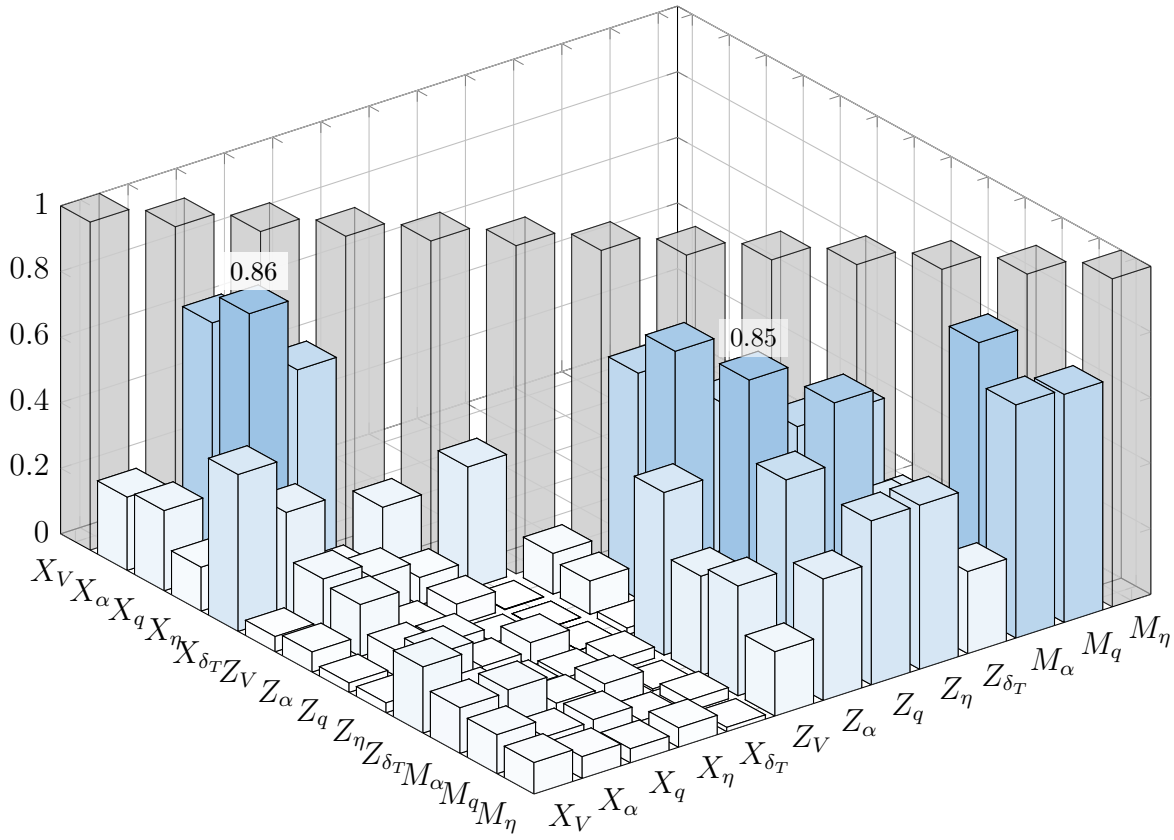


Figure 6.6: *Absolute values of the correlation matrix; largest absolute values are printed atop the respective bar*

nal/true values. Also, most of the estimated standard deviations are very good, well below $5\,\%$. Table 6.1 shows a comparison between true and estimated eigenvalues, whose magnitude agrees to within 0.2%, further underlining the almost perfect match.

Only those parameters pertaining to the body-x direction X, and thus to the velocity differential equation cannot be well estimated: their values are comparatively far away from the truth, and their estimated standard deviations are considerably larger than for the rest of the parameters. This can be attributed to the automatic controller in place: It implements a velocity control, which effectively hides those dynamics (namely the phugoid motion), and thus makes the estimation of the respective parameters more difficult.

Overall, the results are consistent in the sense that large estimated standard deviations appear in estimates where the differences between estimated and true parameters are large, too. Additionally, the initial guess refinement in those cases does not work as well as for the other parameters. Thus, at least for this academic simulation example, it can be concluded that the uncertainty estimates in the form of the parameter covariance matrix are quite reliable.

The estimated absolute values of the correlation coefficients between the parameter estimates are shown in figure 6.6. The relatively similar signal shapes for angle of

attack α and pitch rate q lead to correlations between the parameters describing their influence on the evolution of the states (Z_α, Z_q, X_α, X_q, M_α, M_q). The feedback control then also correlates the elevator input with these signals, leading to somewhat higher correlations of the control derivatives (X_η, Z_η, M_η). However, overall only two correlation coefficients exceed 0.85.

6.1.3 Conclusion – Unstable Aircraft Identification

The last two examples illustrated the application of the methods developed in this thesis to parameter estimation and uncertainty quantification for unstable systems. Classic approaches as illustrated in [Jat2015, Ch. 9] necessitate additional implementation effort or approximations. Here, meaningful results could be obtained by merely using the tools that were implemented for standard, stable aircraft descriptions. The additional robustness of full discretization, together with the novel, especially tailored uncertainty computation approach may readily be used directly with unstable systems. Merely convergence of the NLP solver was not as monotonous and took longer compared to stable examples of similar complexity.

For the example involving the DHC-2, quite similar results could actually be obtained using the implementation of the standard, single shooting method. A version of it was developed during the work on this thesis, as alternative to the full discretization transcription method. Apparently, the instability was not too severe to prohibit computing estimates and their covariances using the classical single shooting approach.

However, for the F16 example, this is not true anymore. Integration of both the state and sensitivity equations quickly resulted in deviations of several orders of magnitude, making the calculation of reasonable results impossible. However, as illustrated in this section, the application of the estimation approach using full discretization (see section 4.2), together with the null-space method to obtain parameter covariances (see section 3.5) does not suffer from those same problems.

A minor drawback of the novel approach, to be kept in mind when interpreting results, is due to the nature of unstable systems: small changes (e.g. due to round-off errors or model re-formulations) may have considerable influence on the results, both with respect to parameter and uncertainty estimates. This can hardly be avoided because it is inherent to the *system* at hand, not due to the estimation algorithm. A symptom of this is that uncertainty estimates that were computed using the numeric null space approach of section 3.5.4.1 may differ notably from their counterpart computed with the analytic approach of section 3.5.4.2. Although the results in both cases are qualitatively the same, the differences in the numerical values cannot solely be explained by round-off errors.

6.2 Parameter Estimation in Stochastic Systems

The following two subsection take up two example cases, that JATEGAONKAR presents to illustrate the application of the FEM [Jat2015]. They will be solved using the novel stochastic problem formulation transcribed via full discretization, as was summarized in section 4.4.

6.2.1 DHC-2 Lateral Directional Motion with Process Noise

The first case, which is illustrated in [Jat2015, Ch. 5.11.1], treats the lateral directional motion of a DeHavilland DHC-2. Since this is a simulated example, additional information such as the true outputs and parameters are available for comparison.

6.2.1.1 Model Formulation

The model is the following

$$\begin{bmatrix} \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L_p & L_r \\ N_p & N_r \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} L_\xi & L_\zeta & L_v \\ N_\xi & N_\zeta & N_v \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \\ v \end{bmatrix} + \begin{bmatrix} b_{\dot{p}} \\ b_{\dot{r}} \end{bmatrix} \tag{6.14}$$

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u} + \boldsymbol{b}_{\boldsymbol{x}} \tag{6.15}$$

with states $\boldsymbol{x}$ representing roll rate p and yaw rate r. The inputs to the system are aileron deflection ξ, rudder deflection ζ and the side component of the velocity in body fixed coordinates v. The latter is assumed to be known from data post-processing. Parameters to be estimated are again the dimensional derivatives of above system equation. The output equation is

$$\begin{bmatrix} \dot{p} \\ \dot{r} \\ a_y \\ p \\ r \end{bmatrix} = \begin{bmatrix} L_p & L_r \\ N_p & N_r \\ Y_p & Y_r \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} L_\xi & L_\zeta & L_v \\ N_\xi & N_\zeta & N_v \\ Y_\xi & Y_\zeta & Y_v \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \\ v \end{bmatrix} + \begin{bmatrix} b_{\dot{p}} \\ b_{\dot{r}} \\ b_{ay} \\ b_p \\ b_r \end{bmatrix} \tag{6.16}$$

$$\boldsymbol{y} = \mathbf{C}\boldsymbol{x} + \mathbf{D}\boldsymbol{u} + \boldsymbol{b}_{\boldsymbol{y}} \tag{6.17}$$

where in addition to the states, and the lateral acceleration a_y, measurements of the state derivatives $\dot{p}$, $\dot{r}$ are assumed to be available.

One major difference in comparison with the original example is, that the same bias parameters $b_{\dot{p}}$ and $b_{\dot{r}}$ are used on the input (eq. (6.15)) *and* output side (eq. (6.17)) of the model, where JATEGAONKAR introduces two different sets of bias terms. However, it was found that only extremely correlated estimates of the two could be obtained, which is why the model was simplified in this manner.

6.2.1.2 Data Gathering

Since the accompanying material of [Jat2015] does not provide the true process noise sequence, data was generated instead of using the provided record, in order to have the true $\boldsymbol{w}$ available for comparison. This was done using the same inputs $\boldsymbol{u}$ as in [Jat2015], which can be seen in the three bottom plots of figure 6.7. In addition to that, a white noise sequence was generated using MATLAB's `randn` function and the following covariance matrix

$$\mathrm{Cov}[\boldsymbol{w}_k] = \mathbf{Q} = \begin{bmatrix} \left(2.83\,\mathrm{rad/s^2}\right)^2 & \\ & \left(2.83\,\mathrm{rad/s^2}\right)^2 \end{bmatrix} \tag{6.18}$$

which is used as process noise input $\boldsymbol{w}$ and may be seen in figure 6.8. Since the exact inner workings of JATEGAONKAR's data-generation process are not known, above process noise covariance was tuned manually, to obtain a result that is visually comparable to his original data-set.

Similarly, the measurement noise covariance matrix was chosen to achieve an effect that seemed realistic for the output values at hand

$$\mathbf{R} = \begin{bmatrix} \left(4.79\,^\circ/\mathrm{s^2}\right)^2 & & & & \\ & \left(4.79\,^\circ/\mathrm{s^2}\right)^2 & & & \\ & & \left(0.20\,\mathrm{m/s^2}\right)^2 & & \\ & & & \left(0.57\,^\circ/\mathrm{s}\right)^2 & \\ & & & & \left(0.57\,^\circ/\mathrm{s}\right)^2 \end{bmatrix} \tag{6.19}$$

The resulting measurements are shown in the top plots of figure 6.7.

6.2.1.3 Results

It was found, that the close to quadratic nature of the problem agrees well with SNOPT as solver for the resulting NLP problem. IPOPT was also able to find the same solution, but more iterations and thus longer computational times were necessary.

In order to reduce the number of optimization variables, the process noise samples were thinned out to $80\,\%$ of their original number. FALCON.m [RBG+2018] then interpolated them onto the state time grid to correctly incorporate their effect at every integration time step. This effectively reduces the possible process noise bandwidth, but reduced optimization time considerably, and the adverse effects of the limited bandwidth were not noticeable.

Since state measurements were available, the initial model parameters could be obtained using the improvement approach of section 3.8. In order to obtain reasonable results in this step, a moving average filter was applied to the states and an initial guess of $\boldsymbol{\theta}_0 = \mathbf{0}$ was used. Figure 6.9 then shows the true parameter values, the final estimates, and the initially improved guess. Again, the improvement approach is

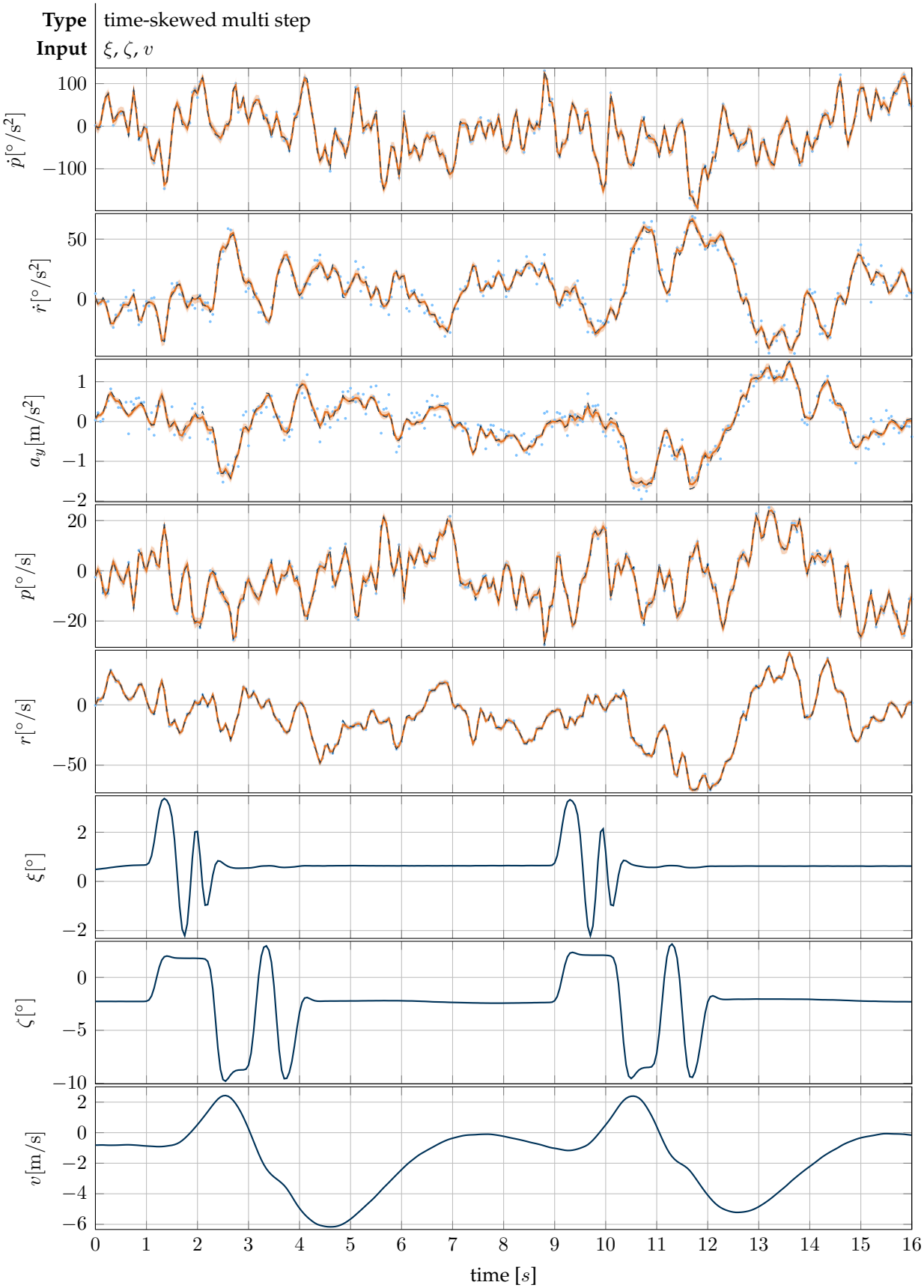


Figure 6.7: *Measurements $\boldsymbol{z}$ (·), estimated $\hat{\boldsymbol{y}}$ (——) and true outputs $\boldsymbol{y}$ (- - -); Control inputs $\boldsymbol{u}$ (——) are shown in the three bottom plots, where the side component of body fixed kinematic velocity vector v is used as additional pseudo-control; the shaded areas indicate the 3σ output bounds, but are almost too small to be visible*

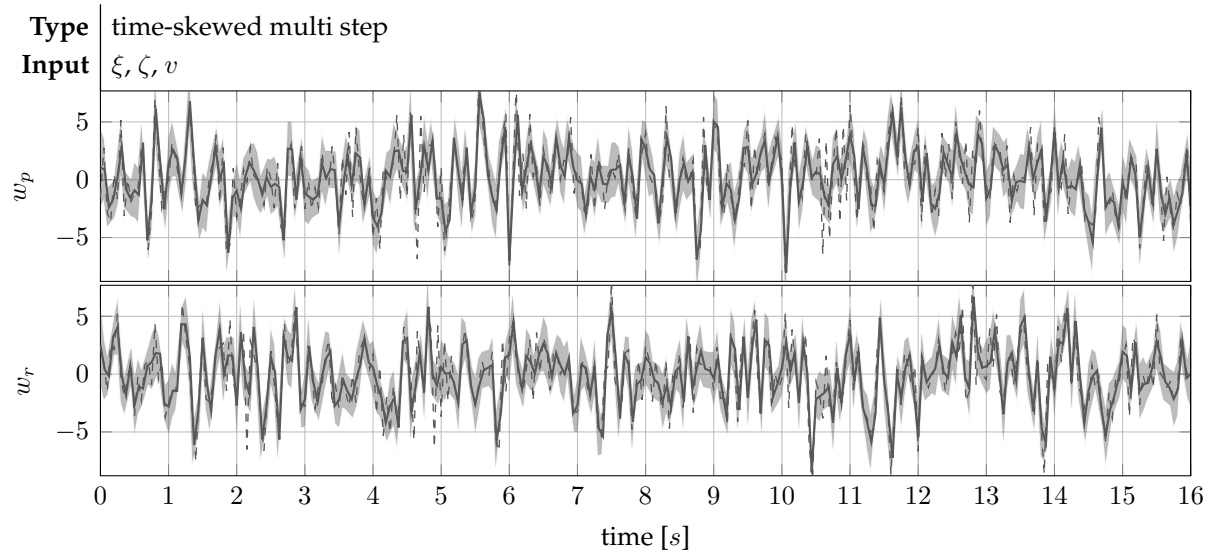


Figure 6.8: *Estimated $\hat{\boldsymbol{w}}$ (——) and true process noise sequence $\boldsymbol{w}$ (- - -) with estimated 3σ bound*

able to provide initial parameter values that are already close to the solution, giving the NLP solver a good chance of finding valid estimates. In general, the agreement between estimated and true values is very good, considering the significant amount of process noise acting on the system. In some cases, the difference between true and estimated parameters is larger, however in those cases the uncertainty estimates indicate this nicely. The worst parameter estimates are those of N_ξ and Y_ξ, where in the latter case even the sign is incorrect. However, for both the estimated relative standard deviation is very high, thus they could be identified as problematic also if their true values were unknown. Correlations, which can be seen in figure 6.10, are very small overall, thus independence of the parameter estimates is ensured.

The estimates of the bias parameters are not shown. They are far off from the true values, and have immense estimated uncertainties. This is accepted, since they represent only nuisance parameters, whose value will not be used further.

Overall, the model performs very well, which can also be seen when comparing the eigenvalues of the system, see table 6.2. This underlines that the characteristic motion is correctly reproduced, which can also be seen when comparing the true and estimated trajectory in figure 6.7. Visually, there is no difference to be seen. Further, it is noteworthy, how well the algorithm is able to reproduce the random process noise

Table 6.2: *Comparison of estimated and true eigenvalues for lateral directional motion of DHC2 with process noise*

λ_{true}	λ_{est}	rel. difference [%]
-5.5764	-5.5833	0.1243
-0.9556	-0.9652	0.9989

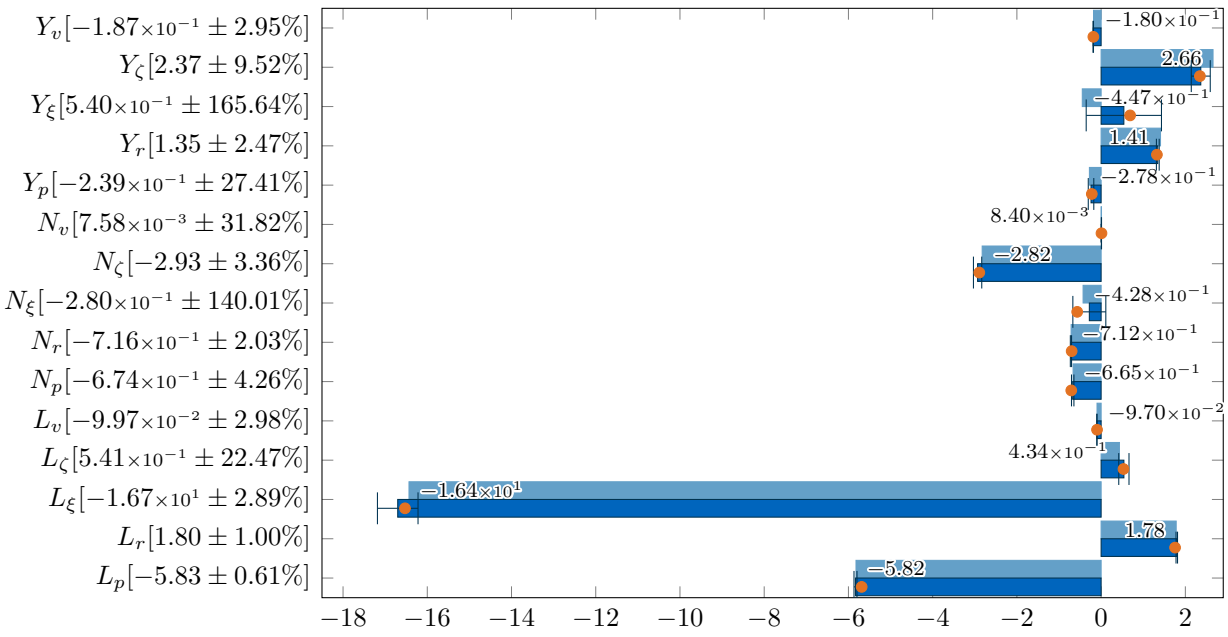


Figure 6.9: *true (* ▬ *) and estimated (* ▬ *) parameter values with estimated* 1σ *bound and initial guesses (* ● *)*

sequence, which can be seen in figure 6.8: all major trends are replicated very well, and only minor differences compared to the true sequence remain. It was found that thinning out the number of process noise samples had a favorable influence on the number of optimization iterations. Probably this is because the same amount of "information" is used to determine fewer parameters, yielding a better defined optimum.

Uncertainty estimates are computed, both for model outputs (figure 6.7) and process noise (figure 6.8). They are relatively small, which is probably due to the model structure: since derivative information was assumed to be available, a lot of information can be presented to the algorithm, which in turn results in "sure" estimates.

Overall the algorithm yields very good results: the time-series of figure 6.7 are strongly corrupted by random inputs. So much so that it is difficult to discern reactions to the deterministic inputs from consequences of the acting process noise. However, in this idealized setting, the algorithm is still able to yield meaningful results. The standard approach would be to use iterative state estimation together with a single shooting formulation for the parameter estimation problem (see section 4.3). Although computations are more involved when using the novel full discretization transcription, and take longer compared to this standard approach, the results are very good. Despite this downside, the novel approach provides valuable additional possibilities, such as the straight forward inclusion of further constraints. These additional algorithmic possibilities can definitely be worth the computational cost. Admittedly, the circumstances in this example are idealized (perfectly white noise; known process noise covariance; perfectly known model structure). However, the very good results give reason to hope that acceptable estimates may still be obtained under less ideal conditions.

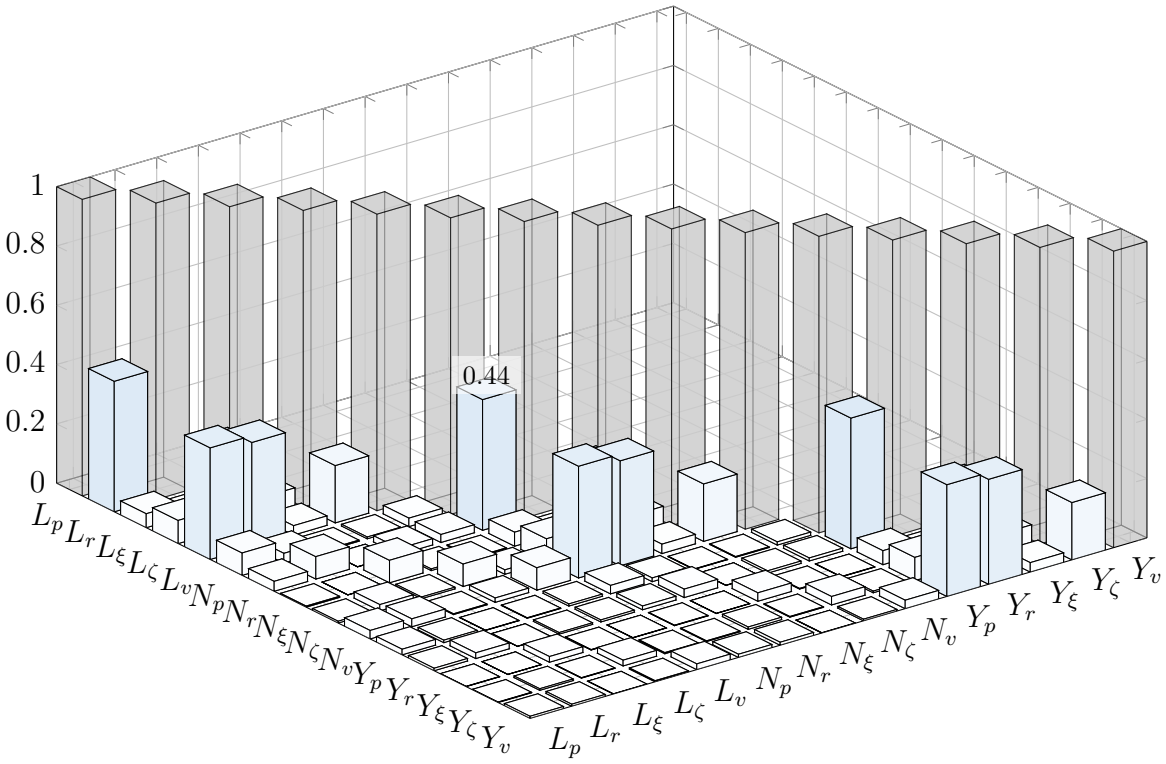


Figure 6.10: *absolute values of the correlation matrix for DHC-2 lateral motion estimation with process noise; largest absolute values are printed atop the respective bar*

6.2.2 HFB 320 Longitudinal Motion with Process Noise

The second case, which is illustrated in [Jat2015, Ch. 5.11.2], treats the longitudinal motion of the HFB 320 research aircraft. A non-liner model for the longitudinal motion of this twin-engine, ten-seat business jet is formulated, and subsequently used for parameter estimation based on real flight test data.

6.2.2.1 Model Formulation

The non-linear model formulation is directly taken from [Jat2015, Ch. 5.11.2]. Since the model itself is rather simple, it is not necessary to use the full extent of the notation introduced in section 5.3. Rather, the model is noted in a simpler fashion, borrowing from [Jat2015, Ch. 5.11.2].

Table 6.3: *constants for HFB320 longitudinal estimation with process noise*

S_{ref}	$\bar{c}$	σ_T	$\frac{\left(x_P \sin\sigma_T + z_P \cos\sigma_T\right)}{I_y}$
$30.0\,\mathrm{m}^2$	$2.43\,\mathrm{m}$	$3.00°$	$-7.0153 \times 10^{-6}\,\mathrm{N}^{-1}\,\mathrm{s}^{-2}$

g	I_y	V_0	m	ρ
$9.806\,65\,\mathrm{m\,s}^{-2}$	$9.1389 \times 10^4\,\mathrm{kg\,m}^2$	$104.67\,\mathrm{m\,s}^{-1}$	$7472\,\mathrm{kg}$	$0.7920\,\mathrm{kg\,m}^{-3}$

The following differential equations are used to connect the longitudinal states V (true airspeed), α (angle of attack), θ (pitch angle), q (pitch rate), with their derivatives and the inputs η (elevator deflection) and T (thrust force)

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{V} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\frac{\bar{q}S_{ref}}{m}\mathsf{C}_D + g\sin(\alpha-\theta) + \frac{T}{m}\cos(\alpha+\sigma_T) \\ -\frac{\bar{q}S_{ref}}{mV}\mathsf{C}_L + q + \frac{g}{V}\cos(\alpha-\theta) - \frac{T}{mV}\sin(\alpha+\sigma_T) \\ q \\ \frac{\bar{c}\bar{q}S_{ref}}{I_y}\mathsf{C}_m + T\frac{x_P\sin\sigma_T + z_P\cos\sigma_T}{I_y} \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta}) \tag{6.20}$$

The other quantities are dynamic pressure $\bar{q} = \frac{\rho}{2}V^2$, air density ρ, wing reference area S_{ref}, gravitational acceleration g, mass m, inertia around the body y-axis I_y, engine installation angle σ_T, mean aerodynamic chord $\bar{c}$ and the coordinates of the propulsion reference point w.r.t. the c.g. (x_P, z_P). The aerodynamic coefficients are modeled as

$$\mathsf{C}_D = \mathsf{C}_{D0} + \mathsf{C}_{DV}\frac{V}{V_0} + \mathsf{C}_{D\alpha}\alpha \tag{6.21}$$

$$\mathsf{C}_L = \mathsf{C}_{L0} + \mathsf{C}_{LV}\frac{V}{V_0} + \mathsf{C}_{L\alpha}\alpha \tag{6.22}$$

$$\mathsf{C}_m = \mathsf{C}_{m0} + \mathsf{C}_{mV}\frac{V}{V_0} + \mathsf{C}_{m\alpha}\alpha + \mathsf{C}_{mq}\frac{\bar{c}q}{2V_0} + \mathsf{C}_{m\eta}\eta \tag{6.23}$$

with reference velocity V_0.

The output equation is

$$\boldsymbol{y} = \begin{bmatrix} V \\ \alpha \\ \theta \\ q \\ \dot{q} \\ a_{x,K} \\ a_{z,K} \end{bmatrix} = \begin{bmatrix} V \\ \alpha \\ \theta \\ q \\ \frac{\bar{c}\bar{q}S_{ref}}{I_y}\mathsf{C}_m + T\frac{x_P\sin\sigma_T + z_P\cos\sigma_T}{I_y} \\ \frac{\bar{q}S_{ref}}{m}\mathsf{C}_X + \frac{T}{m}\cos\sigma_T \\ \frac{\bar{q}S_{ref}}{m}\mathsf{C}_Z - \frac{T}{m}\sin\sigma_T \end{bmatrix} = \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta}) \tag{6.24}$$

where the force coefficients in the body fixed frame relate to the aerodynamic coefficients in the aerodynamic frame according to

$$\mathsf{C}_X = \mathsf{C}_L\sin\alpha - \mathsf{C}_D\cos\alpha \tag{6.25}$$

$$\mathsf{C}_Z = -\mathsf{C}_L\cos\alpha - \mathsf{C}_D\sin\alpha \tag{6.26}$$

The geometric data may be found in table 6.3. The parameters to be estimated are the aerodynamic derivatives

$$\boldsymbol{\theta}_{aero} = \begin{bmatrix} \mathsf{C}_{D0} & \mathsf{C}_{DV} & \mathsf{C}_{D\alpha} & \dots \\ \mathsf{C}_{L0} & \mathsf{C}_{LV} & \mathsf{C}_{L\alpha} & \dots \\ \mathsf{C}_{m0} & \mathsf{C}_{mV} & \mathsf{C}_{m\alpha} & \mathsf{C}_{mq} & \mathsf{C}_{m\eta} \end{bmatrix} \tag{6.27}$$

6.2.2.2 Data Gathering

According to [Jat2015, Ch. 5.11.2] the data was collected during a flight test campaign with only small input perturbations. Since the changes in altitude were very small, density ρ was considered to be constant, see table 6.3.

The inputs were designed to first investigate the short period by using a 3-2-1-1 input, after which the phugoid is excited via a step change. Both are executed using the elevator as primary control, which may be seen in the next to last subplot of figure 6.11. The acting thrust force, probably computed via a propulsion model during the data post-processing step, serves as second input and may be seen in the last subplot of figure 6.11. It further shows the aircraft's reaction in the states, and rotational and linear accelerations.

6.2.2.3 Results

The data was then analyzed using the approach illustrated in section 4.4 on full discretization for stochastic system descriptions. The true value of the process noise covariance matrix was unknown, since the example is based on real flight test data. It was assumed to be

$$\mathbf{Q} = \begin{bmatrix} \left(1.13\,\mathrm{m/s^2}\right)^2 & & & \\ & \left(0.440\,^\circ/\mathrm{s}\right)^2 & & \\ & & \left(0\,^\circ/\mathrm{s}\right)^2 & \\ & & & \left(0.315\,^\circ/\mathrm{s^2}\right)^2 \end{bmatrix} \tag{6.28}$$

The numerical values are roughly based on results presented in [Jat2015, Ch. 5.11.2], where elements of $\mathbf{Q}$ are estimated along with the model parameters. Those are then divided by the sample time $0.1\,\mathrm{s}$ to obtain the above values.

A major difference arises in the covariance associated with the pitch angle θ: in contrast to the original example it is assumed to be deterministic here, i.e. the corresponding process noise covariance element is zero. This not only reduces the number of optimization variables (since $w_{\theta,k} = 0\ \forall k$), but also improves convergence in this special case. The reason is probably that the information in the measurements of both θ and q are used to determine a meaningful time series for q (and consequently w_q). Otherwise, additional uncertainty would be introduced, while not having enough information at ones disposal to determine the corresponding w_θ signal.

Results were obtained after roughly one minute of computations. The arising output estimates can be seen in figure 6.11 along with the inputs and measurements. The record contains some turbulence influence, which is most obvious in the first $4\,\mathrm{s}$, where considerable changes in the states are present, while the inputs are relatively constant. Also, the velocity measurement shows fluctuations over the complete time history that are not directly related to other states or inputs. However, the algorithm is able to track

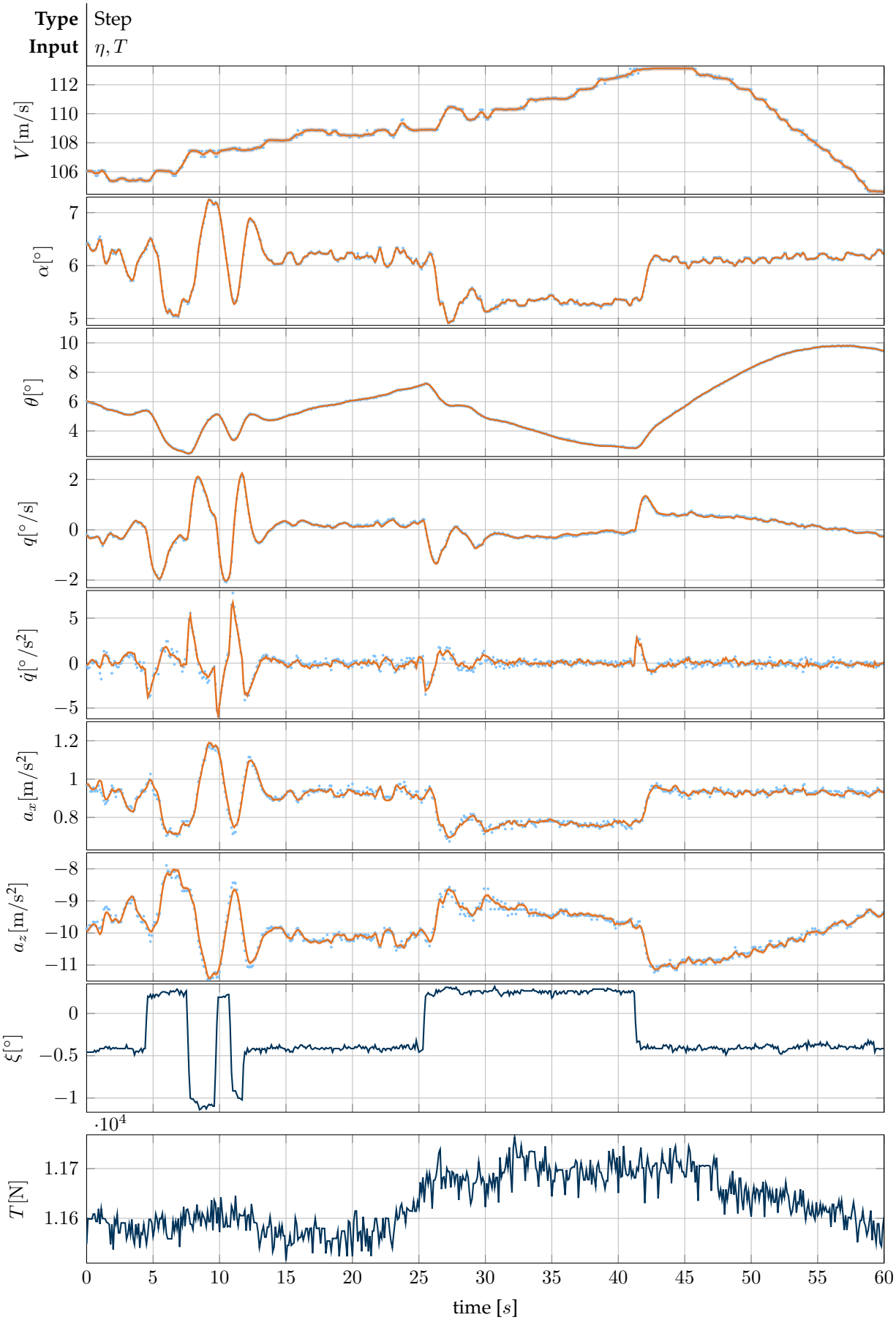


Figure 6.11: *Measurements $\boldsymbol{z}$ (·), estimated outputs $\hat{\boldsymbol{y}}$ (—) and control inputs $\boldsymbol{u}$ (—); 3σ bounds of the outputs are indicated as shaded areas but are barely visible*

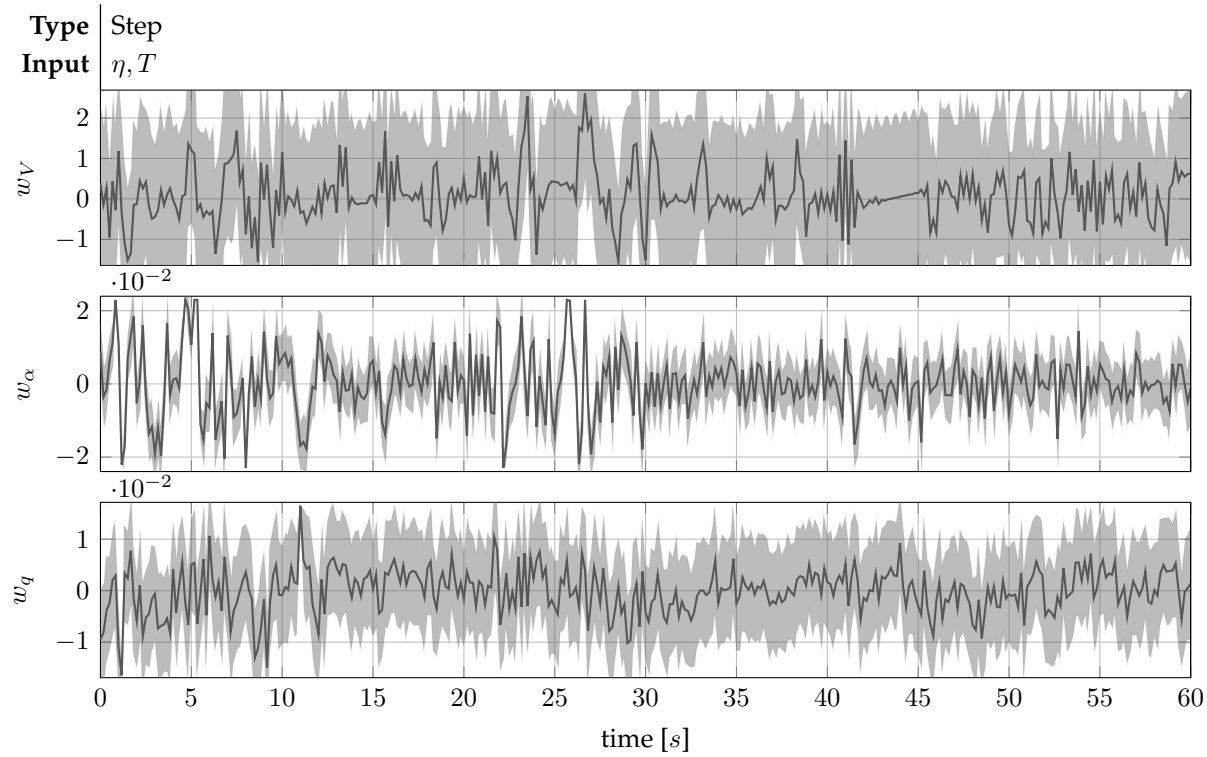


Figure 6.12: *Estimated process noise for HFB 320 longitudinal motion example*

these turbulence influences very well, yielding the overall nearly perfect fit. Figure 6.11 also contains estimated 3σ bounds for the outputs, however they are very small and nearly indiscernible here.

Figure 6.12 shows the estimated process noise signals, together with their estimated 3σ bounds. The process noise signals have again been thinned out to only retain $60\,\%$ of the original number of samples. A consequence of this is the same as before: more information is used per process noise sample, which improves convergence. Although the estimated $\boldsymbol{w}$ sequences are not perfectly white, and still contain some minor deterministic influences, the overall shape of the $\boldsymbol{w}$ signals is deemed satisfactory for the case at hand.

Actual model parameter estimates are shown as bar plot in figure 6.13, together with the estimated standard deviations. These can overall be considered to be very good, most of them lie below $4\,\%$, where only C_{mV}, C_{LV}, and C_{L0} have estimated standard deviations that are somewhat larger. This may hint at identifiability problems. Figure 6.13 further shows the results as obtained by JATEGAONKAR, where a close agreement between the two sets can be stated. The differences are mostly below $7\,\%$ of JATEGAONKAR's values. Only the C_{mV} estimates differ largely, further confirming above suspicion for this parameter.

The last aspect illustrated by figure 6.13 are the improved initial guesses shown as orange dots. Again, they were obtained using the novel equation error approach illustrated in section 3.8 together with $\boldsymbol{\theta}_0 = \mathbf{0}$. They already give the right tendencies and are thus perfectly suited as starting values for an iterative procedure.

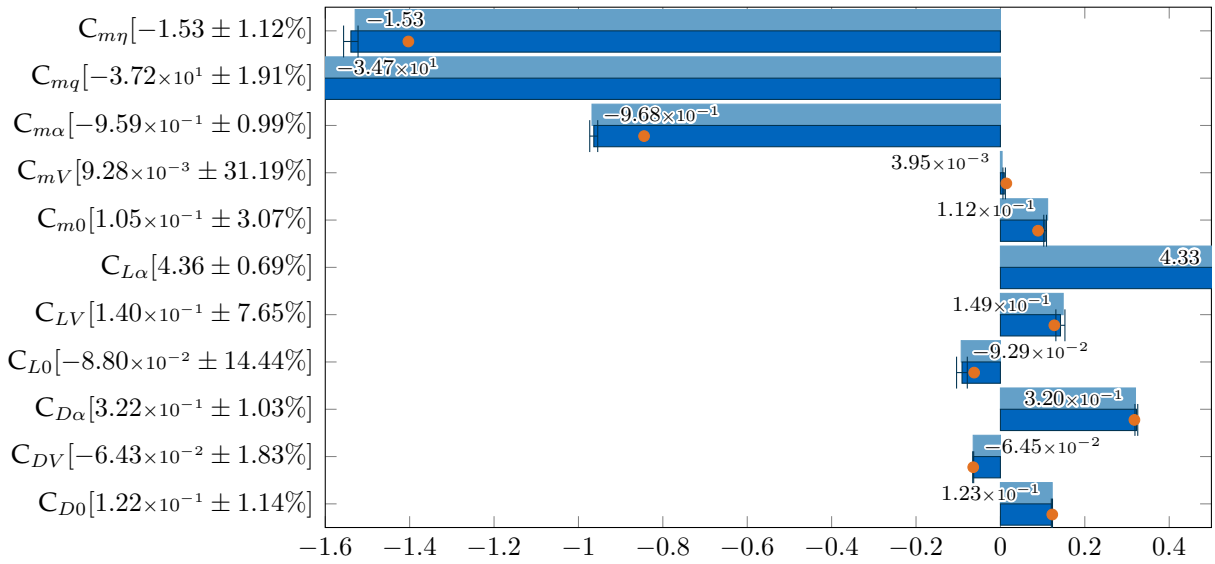


Figure 6.13: JATEGAONKAR's *result (* ▬ *) and parameters values estimated here (* ▬ *) with estimated* 1σ *bound and improved initial guesses (* ● *); numerical values of the estimated parameters and standard deviations are given as lables,* JATEGAONKAR's *parameter values are written on top of the respective bar*

To conclude this example, figure 6.14 illustrates the correlation coefficients between the model parameter estimates. It further underlines the suspected identifiability issues mentioned above: considerable linear dependencies between the 0- and V- derivatives exist, i.e. the estimation algorithm is not able to properly discern between bias and velocity influences. This is probably due to insufficient variability in the V measurement. However, a very similar result was originally obtained by JATEGAONKAR. Thus the problem may be attributed to the information content and model structure, rather than the estimation algorithm.

6.2.3 Conclusion – Parameter Estimation in Stochastic Systems

The two foregoing examples illustrated the use of the full discretization approach in connection with a stochastic system formulation. First, the two obvious disadvantages of using full discretization for stochastic systems have to be mentioned: The first is related to the choice of the process noise covariance matrix $\mathbf{Q}$, which has to be set a priori. So far, no clear guidelines could be found on how to determine a meaningful value. In contrast, the standard FEM offers an automatic way to compute $\mathbf{Q}$. Unfortunately it is not easily possible to integrate this automatic $\mathbf{Q}$ determination in the novel approach, since it actively alters some of the optimization steps (heuristic F-correction; adjustment of $\boldsymbol{\theta}_i$ to keep $(\mathbf{I} - \mathbf{KC})$ positive definite [Jat2015, Ch. 5]). Here this would mean to interfere with the internal workings of the NLP in use, which is not practical. Future research into the applicability of "adaptive filtering" [CJ2012, Ch. 5.7] [Jaz1970, Ch. 8.11] may be more promising in solving this issue. The second disadvantage is the

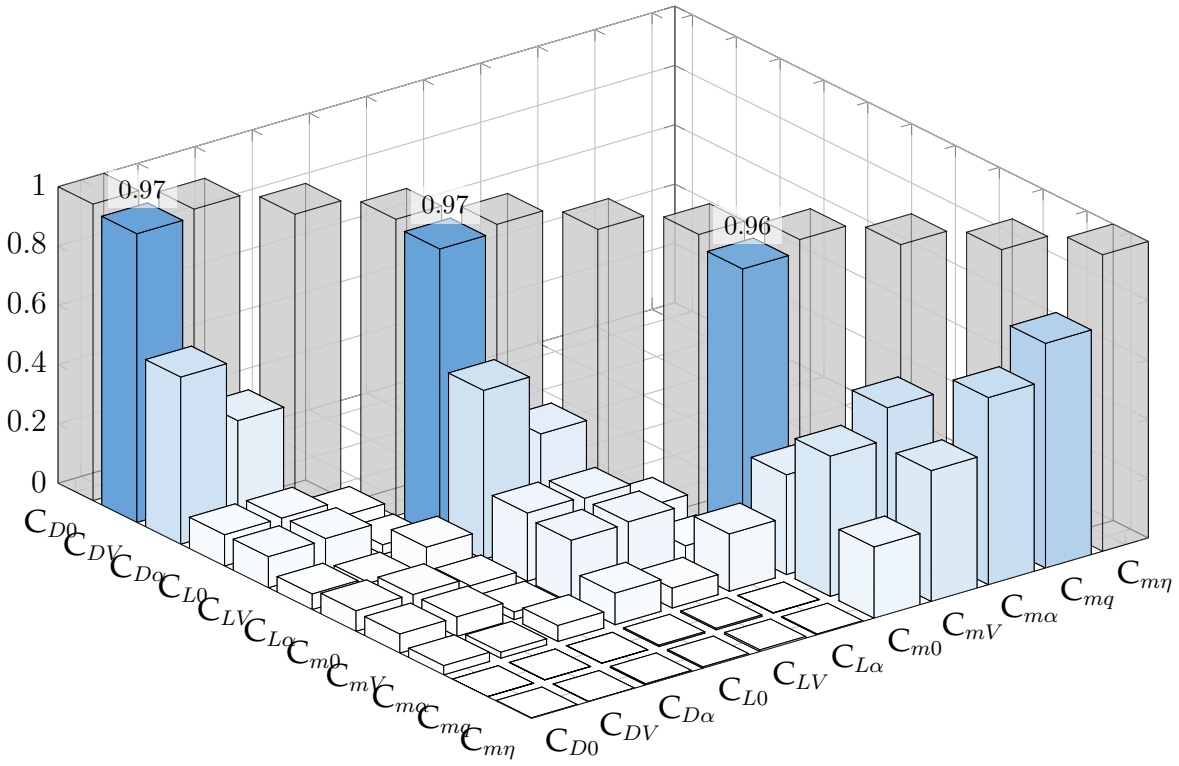
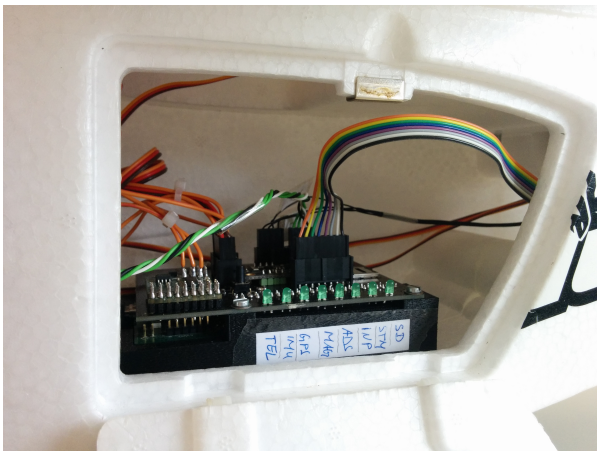


Figure 6.14: *absolute values of the correlation matrix for HFB 320 longitudinal motion estimation with process noise; largest absolute values are printed atop the respective bar*

significantly longer computational time necessary to obtain a result, which is due to the increased size of the problem.

However, these are counterbalanced by the advantages of the method, such as explicit availability of a process noise estimate $\boldsymbol{w}$. This may be used for analysis after the estimation, or to enforce additional constraints during the estimation (e.g. enforce energy dissipation, enforce whiteness, ...). The easy incorporation of arbitrary constraints, not only in the estimation but also in the uncertainty determination process is another big advantage over classical methods. In some cases, they may even make convergence to meaningful results possible, which is otherwise impossible: closeness to measurements may be enforced by using constraints at early optimization stages, when model parameters are still far from their final values. Also, using full discretization tends to be more robust, which is a big advantage if no meaningful initial model parameter guesses may be obtained. These aspects have to be weighed carefully, when deciding which of the two methods to use.

(a) *Airframe with RC controller and telemetry base station*



(b) *Flight Control System (FCC) based on STM-discovery board and custom sensor shield*

Figure 6.15: *The* SKYMULE *testbed*

6.3 Skymule

The following subsections are dedicated to full, non-linear system identification for a small, low-cost RPAS, which is based on the commercially available model aircraft SKYMULE. They are intended to show the full potential of the advertised parameter estimation techniques using optimal control methods. The airframe was chosen because of its comparatively large interior, which was used to install a full sensor suite mainly using low-cost MEMS components:

- MEMS IMU with 3-DOF gyroscopes, accelerometer, and magnetometer
- auxiliary MEMS magnetometer (not used here)
- static pressure sensor
- GNSS receiver
- dynamic pressure probe
- 2.4 GHz transceiver for telemetry link

A first prototype, based on a different airframe, is described in [GH2016], where it was used for FPR experiments by the author. The testbed at hand (figure 6.15) is described in detail in [Bar2016], a semester thesis which was supervised by the author.

6.3.1 SkyMule Non-Linear Parameter and Wind Estimation

The first application, based on flight data collected with the SKYMULE, is a parameter estimation task for a full 6-DOF, non-linear simulation model of the aircraft. To do so, first the model will be postulated, then the data gathering and preprocessing steps will be illustrated, before estimation results are shown and discussed.

6.3.1.1 Model Formulation

The non-linear model is largely based on what was discussed in section 5.3:

- the equations of motion are formulated for a flat, non-rotating earth, with respect to a fixed reference point (section 5.3.1.1)
- attitude kinematics are formulated in terms of Euler angles (section 5.3.1.2)
- position kinematics are formulated in WGS84 coordinates (section 5.3.1.3)
- the engine dynamics are approximated as first order lag (section 5.3.2.1)
- actuator dynamics are approximated as second order lag (section 5.3.2.4)
- propeller data is taken from [BDAS2018], by choosing one that is similar (three blades, comparable diameter and pitch), and implementing thrust and torque coefficient as lookup table over the advance ratio (section 5.3.2.2)
- weight and balance modeling is based on assumed mass and inertia properties (section 5.3.2.3)
- environmental conditions are implemented with constant gravity, and ISA (section 5.3.3)

After a lengthy trial and error phase, the following aerodynamic model parameters were found to perform well, while still being adequately identifiable

$$\boldsymbol{\theta}_{aero} = \begin{bmatrix} \mathrm{C}_{D0} & \mathrm{C}_{D\alpha} & \mathrm{C}_{D\alpha 2} & \dots & \\ \mathrm{C}_{Q\beta} & \mathrm{C}_{Q\zeta} & \dots & & \\ \mathrm{C}_{L0} & \mathrm{C}_{L\alpha} & \mathrm{C}_{Lq} & \dots & \\ \mathrm{C}_{l\beta} & \mathrm{C}_{lp} & \mathrm{C}_{lr} & \mathrm{C}_{l\xi} & \dots \\ \mathrm{C}_{m0} & \mathrm{C}_{m\alpha} & \mathrm{C}_{mq} & \mathrm{C}_{m\eta} & \dots \\ \mathrm{C}_{n0} & \mathrm{C}_{n\beta} & \mathrm{C}_{np} & \mathrm{C}_{nr} & \mathrm{C}_{n\zeta} \end{bmatrix} \tag{6.29}$$

Further, according to the scheme presented in section 5.2 on sensor modeling and flight path reconstruction, the following sensor error parameters were introduced

$$\boldsymbol{\theta}_{sensor} = \begin{bmatrix} \Delta\left(p_K^{OB}\right)_B & \Delta\left(q_K^{OB}\right)_B & \Delta\left(r_K^{OB}\right)_B \\ \Delta\left(f_x^R\right)_B^{II} & \Delta\left(f_y^R\right)_B^{II} & \Delta\left(f_z^R\right)_B^{II} \\ \Delta\left(b_x^R\right)_B & \Delta\left(b_y^R\right)_B & \Delta\left(b_z^R\right)_B \\ \Delta T & \Delta p_{stat} & \end{bmatrix} \tag{6.30}$$

with gyroscope biases ($\Delta\left(p_K^{OB}\right)_B$, $\Delta\left(q_K^{OB}\right)_B$, $\Delta\left(r_K^{OB}\right)_B$), accelerometer bias ($\Delta\left(f_x^R\right)_B^{II}$, $\Delta\left(f_y^R\right)_B^{II}$, $\Delta\left(f_z^R\right)_B^{II}$), magnetometer bias ($\Delta\left(b_x^R\right)_B$, $\Delta\left(b_y^R\right)_B$, $\Delta\left(b_z^R\right)_B$) and a temperature and pressure offset to account for non-standard ISA conditions (ΔT, Δp_{stat}).

During data-preprocessing it was noted that wind effects cannot be neglected: flight tests were conducted at velocities between $15\,\mathrm{m/s}$ and $20\,\mathrm{m/s}$, whereas comparing kinematic and aerodynamic velocities indicated wind velocities of about $4\,\mathrm{m/s}$. Thus the

model was augmented with a simple random-walk description for wind velocity and direction

$$\begin{bmatrix} \dot{V}_W \\ \dot{\chi}_W \end{bmatrix} = \begin{bmatrix} w_{V_W} \\ w_{\chi W} \end{bmatrix} \tag{6.31}$$

where the corresponding process noise covariance matrix was tuned manually to

$$\mathbf{Q} = \begin{bmatrix} (1\,\mathrm{m/s})^2 & \\ & (10^\circ)^2 \end{bmatrix} \tag{6.32}$$

The resulting model then relates the following states, inputs, and outputs

$$\begin{aligned} \boldsymbol{x} = \big[\; & \left(u_K^R\right)_B^E \quad \left(v_K^R\right)_B^E \quad \left(w_K^R\right)_B^E \quad \left(p_K^{OB}\right)_B \quad \left(q_K^{OB}\right)_B \quad \left(r_K^{OB}\right)_B \quad \dots \\ & \phi^{BO} \quad \theta^{BO} \quad \psi^{BO} \quad \lambda^R \quad \mu^R \quad h^R \quad \dots \\ & V_W \quad \chi_W \; \big]^\intercal \end{aligned} \tag{6.33}$$

$$\boldsymbol{u} = \big[\; \xi_l \quad \xi_r \quad \eta \quad \zeta \quad \omega_l \quad \omega_r \; \big]^\intercal \tag{6.34}$$

$$\begin{aligned} \boldsymbol{y} = \big[\; & \left(p_K^{OB}\right)_B \quad \left(q_K^{OB}\right)_B \quad \left(r_K^{OB}\right)_B \quad \left(f_x^R\right)_B^{II} \quad \left(f_y^R\right)_B^{II} \quad \left(f_z^R\right)_B^{II} \quad \dots \\ & \left(b_x^R\right)_B \quad \left(b_y^R\right)_B \quad \left(b_z^R\right)_B \quad \lambda^R \quad \mu^R \quad h^R \quad \dots \\ & \left(u_K^R\right)_O^E \quad \left(v_K^R\right)_O^E \quad \left(w_K^R\right)_O^E \quad \bar{q} \quad T \quad p_{stat} \quad \dots \\ & \phi^{BO} \quad \theta^{BO} \quad \psi^{BO} \quad V_W \quad \chi_W \; \big]^\intercal \end{aligned} \tag{6.35}$$

6.3.1.2 Data Gathering

The flight data stems from a test performed in August 2017, using a semi-automatic maneuver injection system: the pilot trimmed the aircraft, before a pre-defined automatic maneuver was started via the telemetry link. The software in use was an early prototype of what was presented in [KGH2018]. Since automatically generated signals were added to the pilot's inputs, he could still influence the execution. Further, the fail-safe system allowed for an immediate overtake, in case something unexpected happened.

Several maneuvers were conducted to excite both the longitudinal and lateral directional motion of the aircraft. During data pre-processing, actuator positions and engine rotational rates were precomputed based on the resulting control inputs and the respective models. They then served as inputs to the non-linear simulation model. This was possible, since these subsystem models do not depend on flight mechanic quantities: they are implemented as simple low-pass filters. The result is a size reduction of the state vector, since actuator and engine states are eliminated.

Figure 6.16 shows the main signals of the longitudinal motion for the five respective maneuvers. They were designed to excite the short period motion ("3211") and phugoid ("PULSE"), mainly using the elevator input. The off-axis response, i.e. the

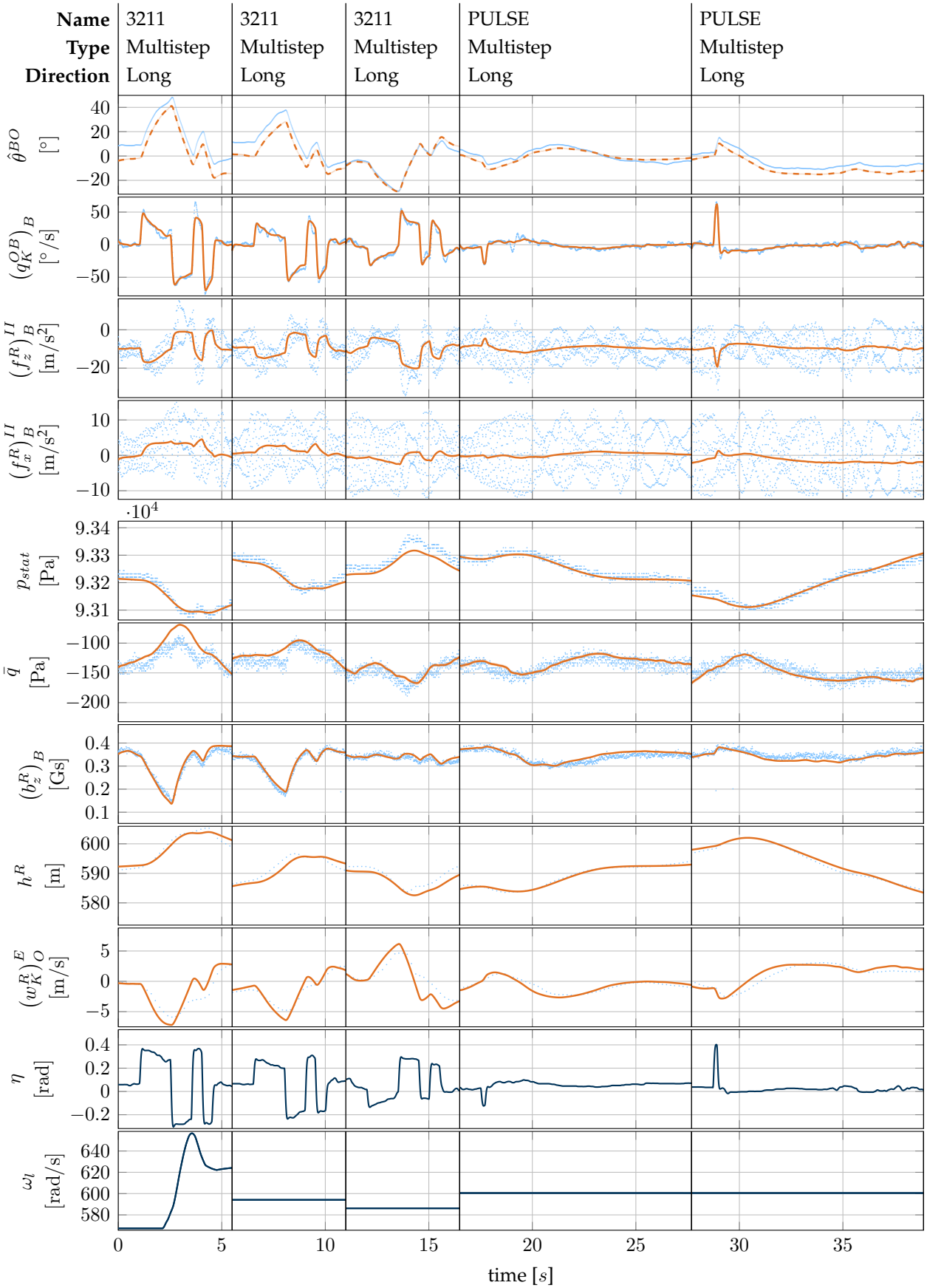


Figure 6.16: *Longitudinal motion stemming from maneuvers in the longitudinal plane; measurements $\boldsymbol{z}$ (·), model outputs $\boldsymbol{y}$ (——), outputs not considered in the estimation (- - -), and control inputs $\boldsymbol{u}$ (——)*

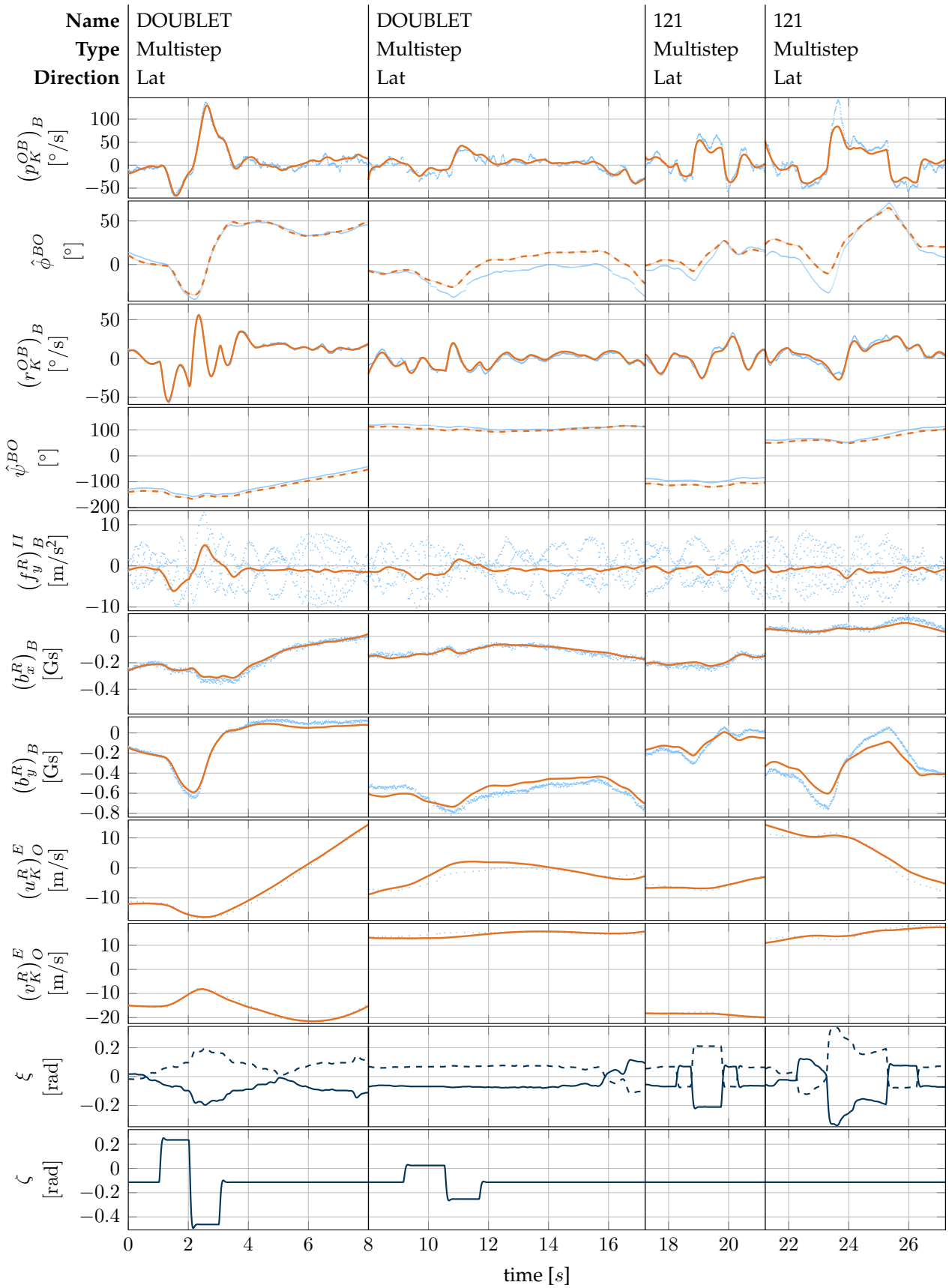


Figure 6.17: *Lateral motion stemming from maneuvers in the lateral plane; measurements* $\boldsymbol{z}$ (·), *model outputs* $\boldsymbol{y}$ (——), *outputs not considered in the estimation* (- - -), *and control inputs* $\boldsymbol{u}$ (- - -)

aircraft's reaction in the lateral plane, is shown for completeness in figure G.1 in the appendix.

Figure 6.17 illustrates the lateral signals resulting from maneuvers that were intended to excite the lateral directional motion. They were designed to excite the dutch-roll (rudder "DOUBLET") and roll motion (aileron "121"). Again, the off-axis response is shown in figure G.2 in the appendix. In all figures one can see the effect of the safety pilot still having some control: the automatic part of the control signals are perfect square waves, however they are added to the pilot's control inputs, which distorts them to a certain extent.

The collected data consists almost exclusively of raw measurement data from magnetometer, GNSS receiver as well as inertial and air data sensors. The different sampling rates may be seen in the density of the respective signals: the GNSS experimental data ($\sim 5\,\mathrm{Hz}$) is a lot less dense when compared to inertial, magnetometer and dynamic pressure data ($\sim 100\,\mathrm{Hz}$); static pressure data is in between ($\sim 70\,\mathrm{Hz}$).

Overall, the data quality is acceptable for low-cost sensors: clear trends are visible during the maneuvers, but the noise level is comparatively high. Only gyroscope measurements seem to be very good, with hardly any random measurement noise. In contrast, accelerometer readings are of poor quality to the point of being useless. The reason for is not only sensor noise, but also significant contributions due to structural vibrations: at certain time instants clear patterns from high frequency oscillations are visible. This hints at a very unfavorable choice of installation location for the accelerometer. Even though the quality of its measurements might not be fit for control purposes, the basic trends in the signals are still observable. This is why the measured accelerations still provide additional information that was considered in the estimation process.

The only signals in the figures that do not correspond to raw data are those of the estimated attitude angles $\hat{\phi}^{BO}$, $\hat{\theta}^{BO}$, and $\hat{\psi}^{BO}$. They stem from a complementary filter using inertial and magnetometer data only, which is illustrated in appendix G.1. However, these attitude estimates are not used in the actual estimation process (indicated by the dashed lines in the figures), they are merely presented for comparison.

6.3.1.3 Results

This example case is too complex to find meaningful initial parameter guesses solely based on the initial guess improvement method of section 3.8. Next to the obvious increase in complexity when treating full 6-DOF non-linear models, two other probable reasons have been identified: Firstly, the approach relies heavily on the availability of meaningful state measurements. However, in this example, attitude estimates stem from a complementary filter, which was not fine-tuned. The remaining estimation errors in the Euler angles complicate the application of the initial guess improvement.

Secondly, especially the initial guess for the force derivatives relies heavily on mea-

sured accelerations and correctly computed forces of the aircraft's subsystems. However, as was already noted, acceleration measurements are of very low quality, and the propulsion model is quite basic. These aspects make a meaningful determination of initial force coefficients very difficult.

Nevertheless, a set of initial values that allowed for good final results was found, after a lengthy trial and error period. In this process, the model was split in lateral and longitudinal models (employing the approach detailed in section 5.3.5), and each was treated separately. Then different model formulations were tested, some parameters were tuned manually, and the initial guess improvement method was employed for the sub-models. Especially matching the lateral directional motion turned out to be difficult. Eventually a good initial model parameter guess was obtained, both for aerodynamic and sensor error parameters. The latter were then kept constant during the actual estimation process.

The final output match for the dominant directions of the longitudinal and lateral maneuvers can be seen in figures 6.16 and 6.17. The off-axis results are shown in figures G.1 and G.2 in the appendix. These results were obtained after roughly $30\,\mathrm{min}$ of computations, and another $5\,\mathrm{min}$ were spent on uncertainty quantification. These may seem rather long, however, after a few minutes it was usually apparent, if the optimization had a chance of converging to meaningful results, or if it should be aborted and started anew with different initial guesses or another model structure.

Overall, the output match can be considered to be very good, especially considering the low-cost sensors in use. Especially the match of the rotational rates is very promising: both in the lateral and longitudinal plane, all major influences are captured very well, with only minor residuals remaining. These may be due to unmodeled dynamics, or atmospheric disturbances that could not be captured by the wind estimate. Two major spikes, one in the roll rate $\left(p_K^{OB}\right)_B$ of the last aileron 1-2-1 maneuver, and the other in the pitch rate $\left(q_K^{OB}\right)_B$ of the first 3-2-1-1 maneuver remain, which cannot be properly explained using the linear model introduced here. Also, the overall match of the roll rate $\left(p_K^{OB}\right)_B$, although being acceptable, is worse compared to the fit that was achieved for yaw and pitch rate. However, these aspects are considered a minor drawback that mainly arises with relatively large, abrupt inputs and is thus tolerated.

Resulting model attitudes confirm this picture: The magnetometer readings $\left(b_x^R\right)_B$, $\left(b_y^R\right)_B$, and $\left(b_z^R\right)_B$, which are the main source of direct attitude information, are matched very well. Only during those maneuvers, where there is some unexplained rotational rate remaining, the magnetometer match tends to degenerate, e.g. during the lateral maneuvers, where not all of the roll rate $\left(p_K^{OB}\right)_B$ may be replicated. When comparing the model attitudes with the estimates of the complementary filter no clear conclusion can be drawn: the main trends are captured, and often the picture is consistent with that of the magnetometer data. However, especially the pitch angle θ^{BO} shows a significant offset during the 3-2-1-1 maneuvers, although the pitch rate $\left(q_K^{OB}\right)_B$ is matched

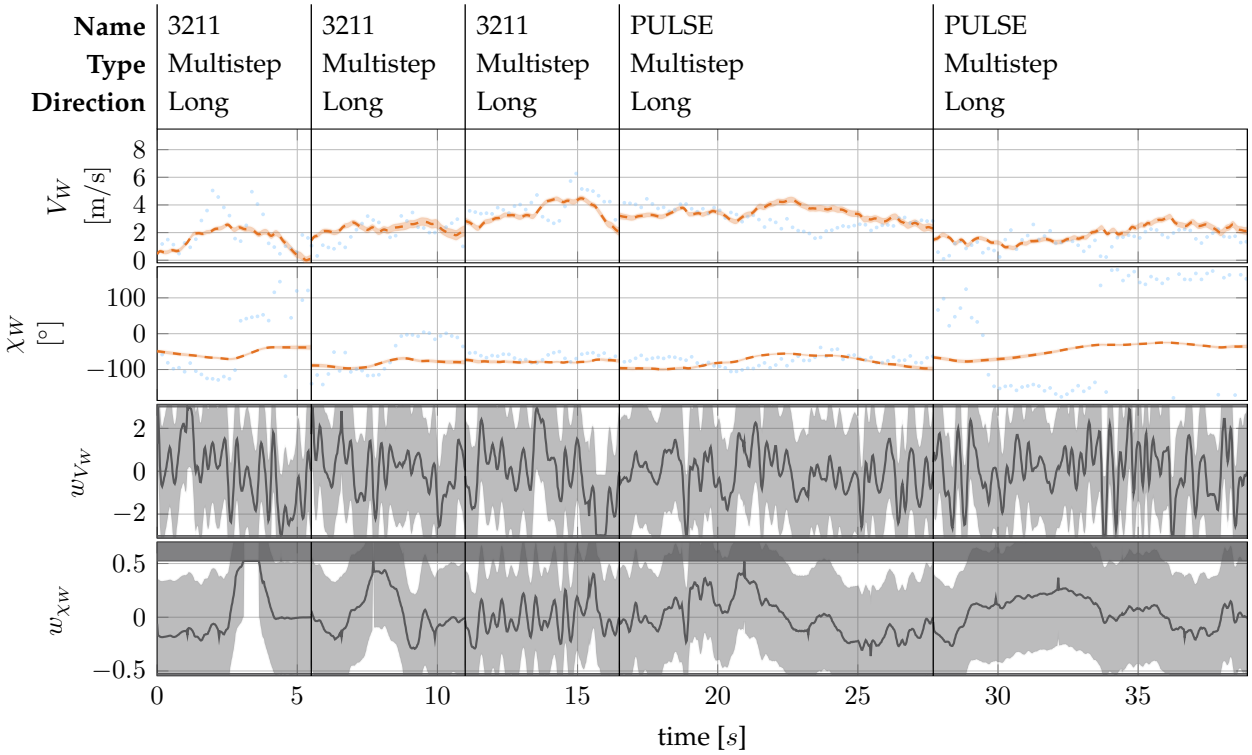


Figure 6.18: *Wind estimate for longitudinal maneuvers in* SKYMULE *example; "measurements" $\boldsymbol{z}$ (·) stem from simple difference between kinematic and aerodynamic velocity, and are not considered in estimation, i.e. are not compared with outputs $\hat{\boldsymbol{y}}$ (- - -); process noise driving the random walk wind model (——) is shown in the bottom plot; 3σ bounds are given as shaded areas, boundaries are a darker shade of grey*

almost perfectly. Since the fit with the $\left(\vec{\boldsymbol{b}}_x^R\right)_B$ is acceptable here, this may hint at slight inconsistencies of the complementary filter rather than model deficiencies.

As was mentioned before, accelerometer data is quite unreliable. Nevertheless, whenever there is significant excitation, this is visible in $\left(\vec{\boldsymbol{f}}^R\right)_B^{II}$, and matched well by the model.

Velocity information is mainly due to GNSS velocity $\left(\vec{\boldsymbol{v}}_K^R\right)_O^E$ and dynamic pressure data $\bar{q}$. For the former the results are good in the lateral directions $\left(u_K^R\right)_O^E$, $\left(v_K^R\right)_O^E$, and they remain acceptable in the vertical direction $\left(w_K^R\right)_O^E$. Some deficiencies may be seen in the vertical velocity, which manifest themselves also to a certain extend in the altitude h^R and static pressure p_{stat} during the longitudinal maneuvers. Probably, this is due to shortcomings in the propulsion model, which adversely influence the longitudinal parameters of the lift and drag equations. However, without more data at further trim speeds, this reaction will be difficult to capture. The velocity graphs also show the effect of direct covariance scaling (section 3.10.2): despite the fact that there is 20 times fewer samples of the GNSS signals, they are still equally well matched compared to the inertial information and air data.

As was mentioned in the first section, wind influences needed to be considered, too.

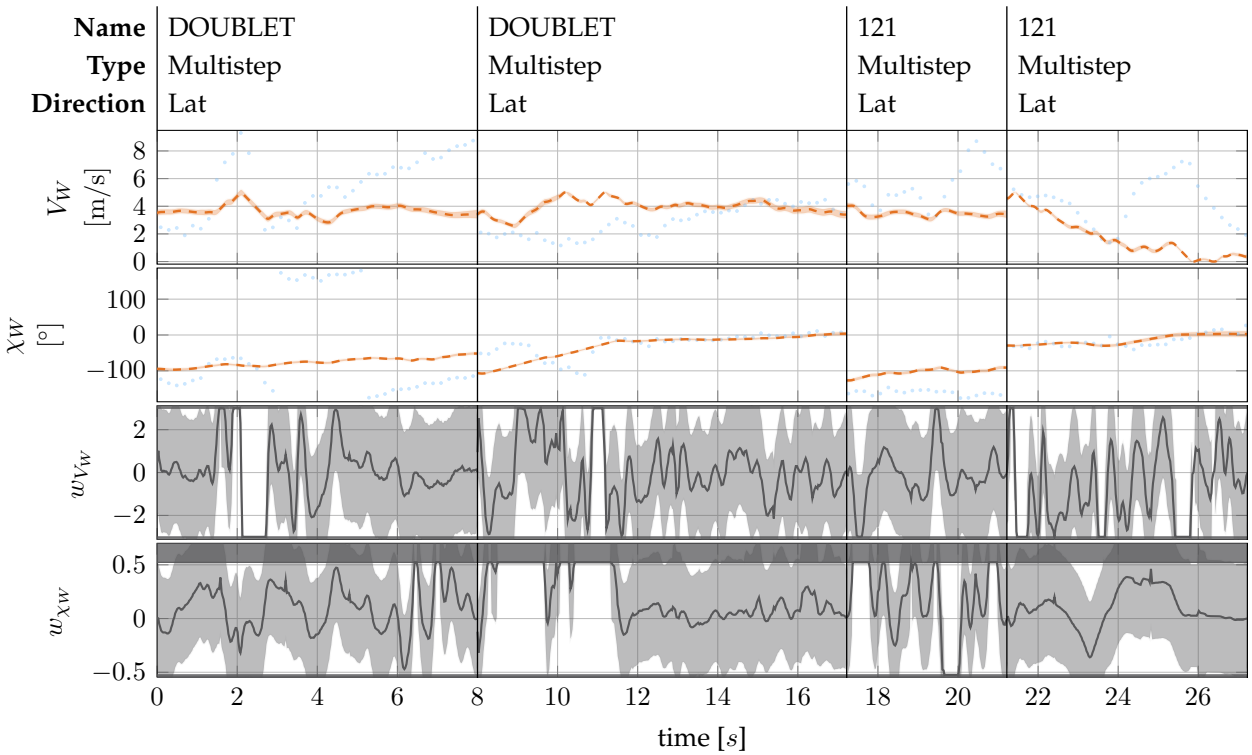


Figure 6.19: *Wind estimate for lateral maneuvers in* SKYMULE *example; "measurements" $\boldsymbol{z}$ (·) stem from simple difference between kinematic and aerodynamic velocity, and are not considered in estimation, i.e. are not compared with outputs $\hat{\boldsymbol{y}}$ (- - -); process noise driving the random walk wind model (——) is shown in the bottom plot; 3σ bounds are given as shaded areas, boundaries are a darker shade of grey*

In order to achieve this, a stochastic, random-walk model for the wind was included, which necessitated the use of the stochastic full discretization approach of section 4.4. However, the problem size did not increase drastically, since only two process noise terms (w_{V_W}, w_{χ_W}) were considered. Additionally, those were thinned out to contain $40\,\%$ of the number of measurement samples, using the same idea as illustrated for the two examples in section 6.2. Using fewer samples for the wind effectively reduces the bandwidth of the signals. This is advantageous here, since we would like to avoid compensating for high frequency model deficiencies with the wind estimate.

Figures 6.18 and 6.19 show the resulting noise sequences and wind estimates. The "measurements" here are approximations based on the difference between kinematic and aerodynamic velocity

$$\left(\vec{\boldsymbol{v}}_W^R\right)_O^E = \left(\vec{\boldsymbol{v}}_K^R\right)_O^E - \mathbf{M}_{BO}^{\mathsf{T}} \begin{bmatrix} \sqrt{\frac{-2\bar{q}}{\rho}} \\ 0 \\ 0 \end{bmatrix} \tag{6.36}$$

Wind magnitude and course are then extracted from the above result. Many errors are contained in this approximation: attitude estimates stem from a complementary filter; flow angle are disregarded; no sensor errors are considered in the air data signals;

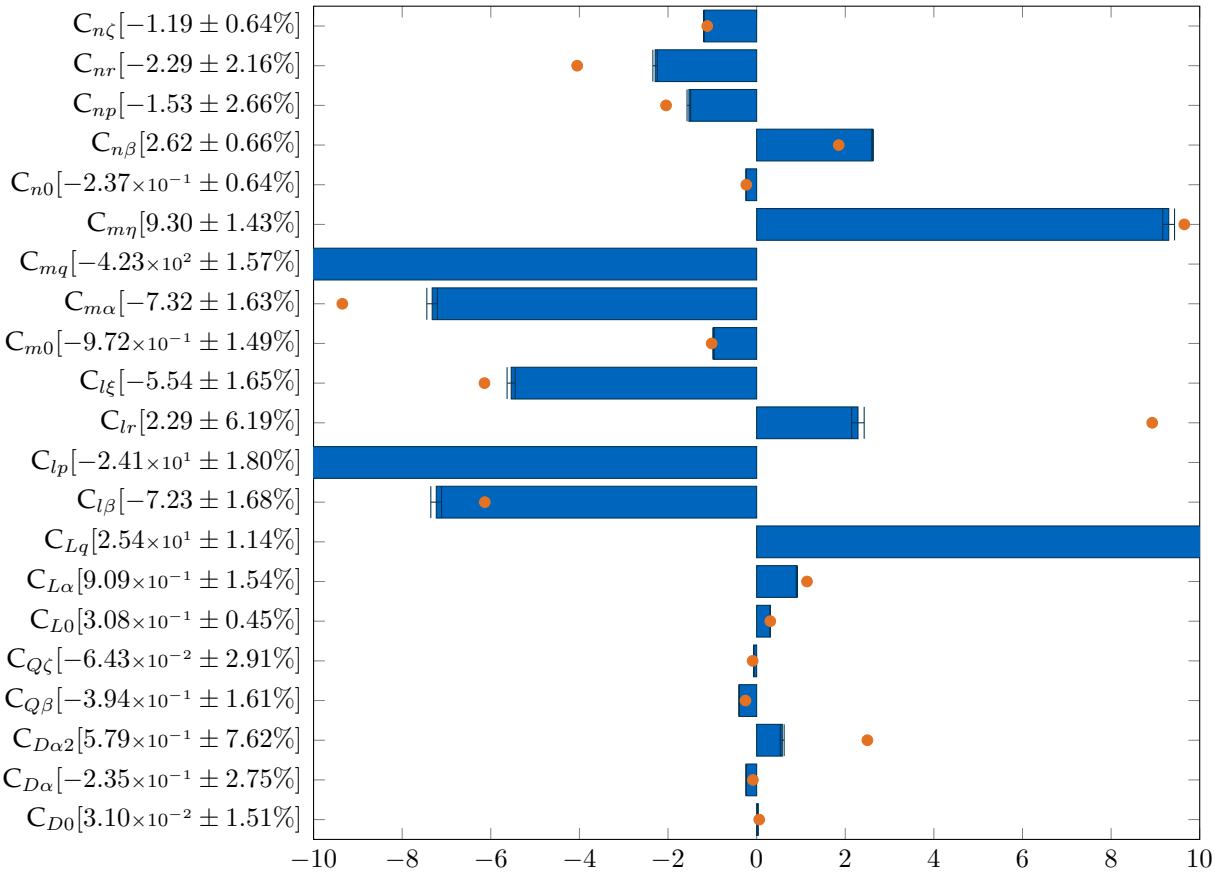


Figure 6.20: *Parameter estimates (▬) with estimated 1σ bound and initial guesses (●); numerical values of the estimated parameters and standard deviations are given as labels;*

density is computed from static pressure and temperature, whose quality is hard to assess. Thus the above $\left(\vec{v}_W^R\right)_O^E$ may only indicate rough trends.

Nevertheless, the resulting estimates lie at least in the same regions. Considering the noise in the measurements, this match is more than acceptable. The corresponding noise sequences cannot be described as "white", but together with the resulting wind estimates, the final result is good enough: The overall goal was not to perfectly reproduce the acting wind sequence, but to consider atmospheric influences in order to help the algorithm determine better parameter estimates.

The driving process noise was limited to its assumed 3σ bound, which was actually attained at several time instants. Even though statistical considerations here are more than questionable, the result seems to justify this choice of boundary. This behavior may be used as indication for possible model deficiencies: at those times when a process noise sequence attains its limit, the algorithm might try to compensate for effects that cannot be reproduced with the model by driving the wind signals to an extreme. The model behavior at those instants should then be scrutinized further. Here, the estimate is accepted in the light of the overall quality of the sensor data and the performance of the model.

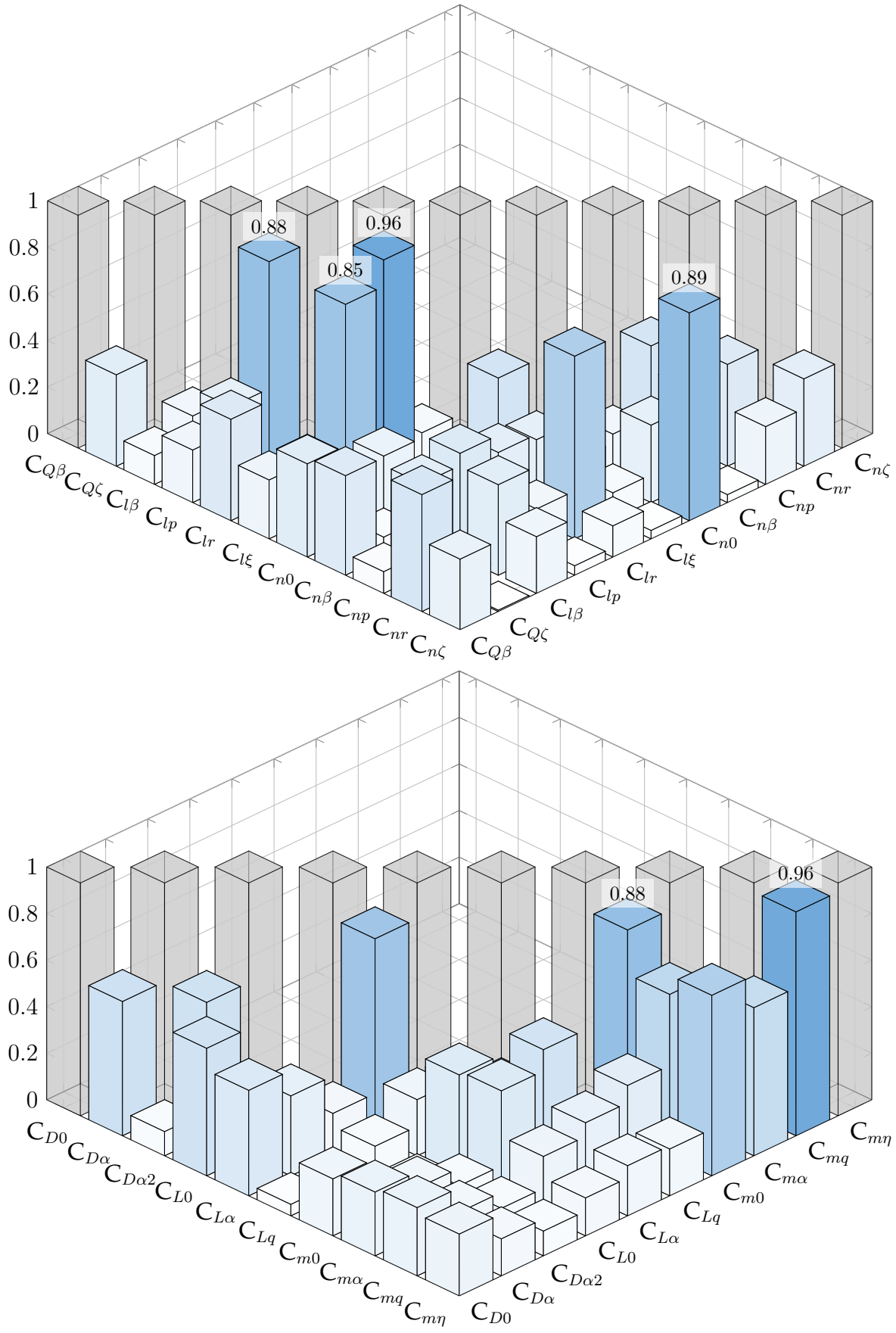


Figure 6.21: *Absolute values of the correlation matrices in the lateral and longitudinal plane for the* SKYMULE *example; largest absolute values are printed atop the respective bar*

Another detail that the figures 6.18 and 6.19 illustrate is the treatment of constraints in the uncertainty determination process. At those times, when the process noise terms are at their boundaries, their uncertainty is zero, as illustrated in example 3.1 of section 3.4. By using the novel null space uncertainty quantification approach of section 3.5, this information is then used for the covariance computations of all other optimization parameter elements.

Figure 6.20 shows the resulting parameter values, initial guesses as well as estimated standard deviations. Especially the overall very low standard deviations, with a maximum of just below $8\,\%$, are noteworthy here. With few exceptions, also the parameter correlations of figure 6.21 underline the quality of the result. Only the correlations $\rho[\mathrm{C}_{l\xi}, \mathrm{C}_{lp}]$ and $\rho[\mathrm{C}_{m\eta}, \mathrm{C}_{mq}]$ are very large, which is unfortunately often the case for RPAS models: their small inertia leads to a very quick reaction to a control surface deflection. Then the signal shapes for the control input, and the resulting rotational rate are too similar, introducing rather strong correlations between the two. However, both parameters are necessary for a meaningful model, which is why this exceedingly large correlation is accepted here.

6.3.2 Model Validation Through Inverse Simulation

The algorithms presented in this thesis may naturally be extended to model validation through inverse simulation. This section formulates and solves the corresponding problem for the SKYMULE results.

6.3.2.1 Model Formulation

The model to be used is very similar to that, which was used for parameter estimation, with two major differences: The wind signals, which have been determined together with the aerodynamic parameters, are now treated as inputs to the system. Secondly, the model is augmented with four Δ-states, each modeled as random walk, to implement the additional control effort, which is necessary to track the trajectory.

$$\begin{bmatrix} \Delta\dot{\xi} \\ \Delta\dot{\eta} \\ \Delta\dot{\zeta} \\ \Delta\dot{\omega} \end{bmatrix} = \begin{bmatrix} w_{\Delta\xi} \\ w_{\Delta\eta} \\ w_{\Delta\zeta} \\ 0 \end{bmatrix} \tag{6.37}$$

It was found that the additional $\Delta\omega$ only varied very little, thus the corresponding process noise covariance element was set to zero. The effect is, that only its initial condition will be estimated, resulting in one constant $\Delta\omega$ per maneuver. In order to keep a good balance between following the trajectory, and minimizing control effort, above Δ-states and process noise elements were limited. This corresponds to limits on the additional control effort (state limits) and limits on its first derivative (process

noise limits). If not bounded, the algorithm might try to chase every last measurement noise sample, which is undesired.

The resulting states are

$$\boldsymbol{x} = \begin{bmatrix} \left(u_K^R\right)_B^E & \left(v_K^R\right)_B^E & \left(w_K^R\right)_B^E & \left(p_K^{OB}\right)_B & \left(q_K^{OB}\right)_B & \left(r_K^{OB}\right)_B & \dots \\ \phi^{BO} & \theta^{BO} & \psi^{BO} & \lambda^R & \mu^R & h^R & \dots \\ \Delta\xi & \Delta\eta & \Delta\zeta & \Delta\omega & & & \end{bmatrix}^\intercal \tag{6.38}$$

and the nominal inputs to the system are

$$\boldsymbol{u} = \begin{bmatrix} \xi_l & \xi_r & \eta & \zeta & \omega_l & \omega_r & V_W & \chi_W \end{bmatrix}^\intercal \tag{6.39}$$

The inputs that actually act on the aircraft's dynamics incorporate the effect of the delta states

$$\boldsymbol{u}^{act} = \begin{bmatrix} \xi_l - K_\xi \cdot \Delta\xi & \xi_r + K_\xi \cdot \Delta\xi & \eta + K_\eta \cdot \Delta\eta & \zeta + K_\zeta \cdot \Delta\zeta & \dots \\ \omega_l + \Delta\omega & \omega_r + \Delta\omega & V_W & \chi_W & \end{bmatrix}^\intercal \tag{6.40}$$

The factors $K_\xi = K_\eta = K_\zeta = 0.70\,\mathrm{rad}$ are used to map the normalized delta states to actual control surface deflections. This effectively bypasses the actuator dynamics, by adding the additional control effort directly to the control surface position signals. However, for model validation purposes, this was accepted, especially since incorporating actuator dynamics would drastically increase the model (and thus problem) size. At the same time, the additional insight to be gained by considering actuator dynamics is expected to be minor.

The outputs and measurements are exactly the same as were considered in the estimation step (equation (6.35)), and all model parameters $\boldsymbol{\theta}$ are fixed at the resulting values of the last section. Thus the only remaining free elements of the optimization vector are the states and process noise controls. This essentially transforms the problem into an optimal control problem without model parameters that has a cost function rooted in statistical considerations.

6.3.2.2 Results

The problem has been solved with the methods as illustrated in case IV of section 4.4. The initial conditions $\boldsymbol{x}_0$ were estimated to provide more freedom to the algorithm. To reduce problem size and bandwidth in the additional control efforts, the number of process noise samples was again reduced to $60\,\%$, still the computation took comparatively long with just over $80\,\mathrm{min}$. This may partly be attributed to the rather strict convergence criteria in use.

Figures 6.22 and 6.23 show a comparison of the measurements, original trajectory of the last section, and inverse simulation result. Figures 6.24 and 6.25 illustrate the Δ-states and the associated process noise signals. The off-axis response, i.e. lateral

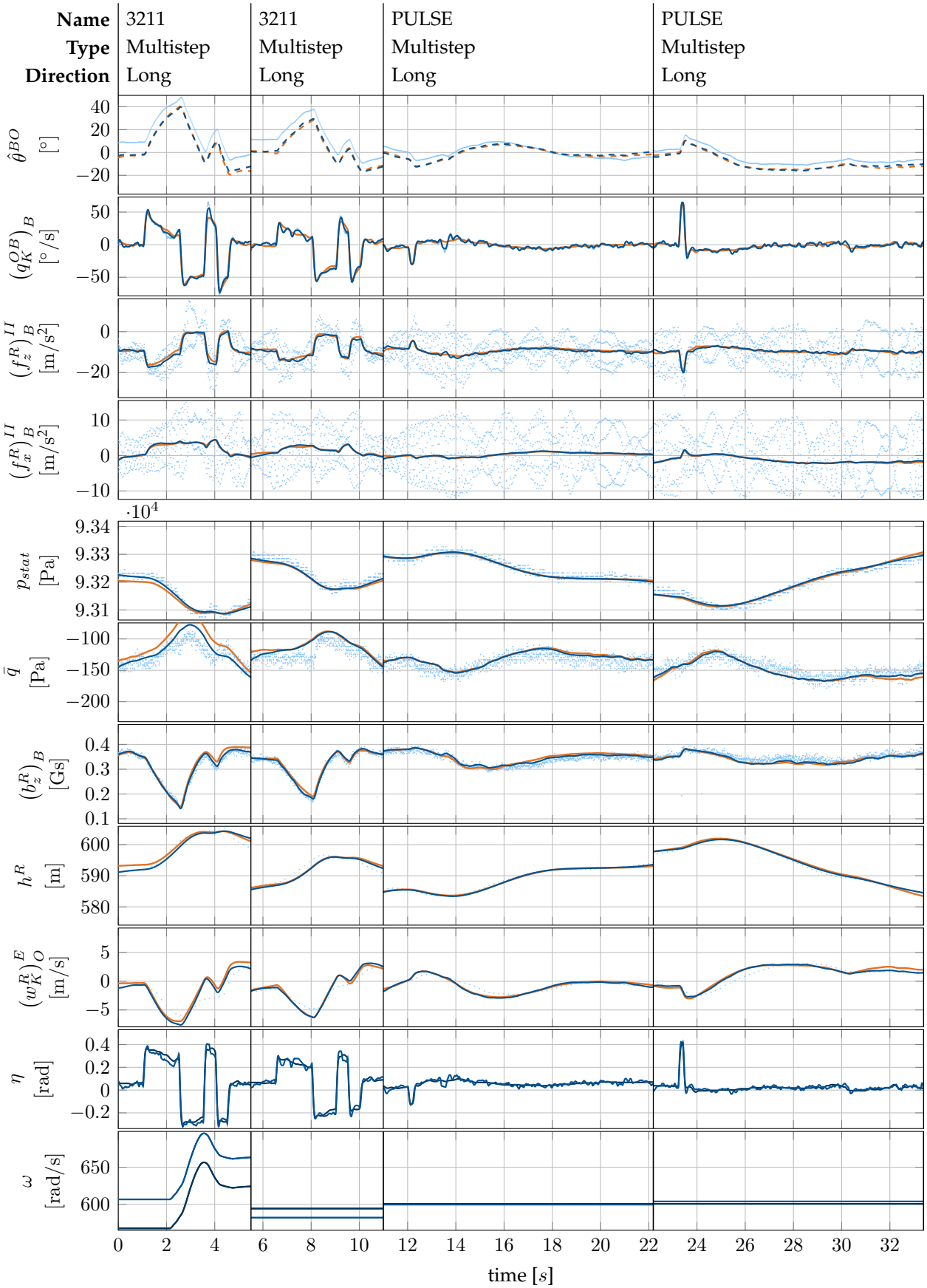


Figure 6.22: *Inverse simulation results for longitudinal plane; measurements $\boldsymbol{z}$ (·), estimated model outputs $\boldsymbol{y}$ (considered —— / unconsidered - - -), and control inputs $\boldsymbol{u}$ (——); Corresponding inverse simulation results (——) are shown on top*

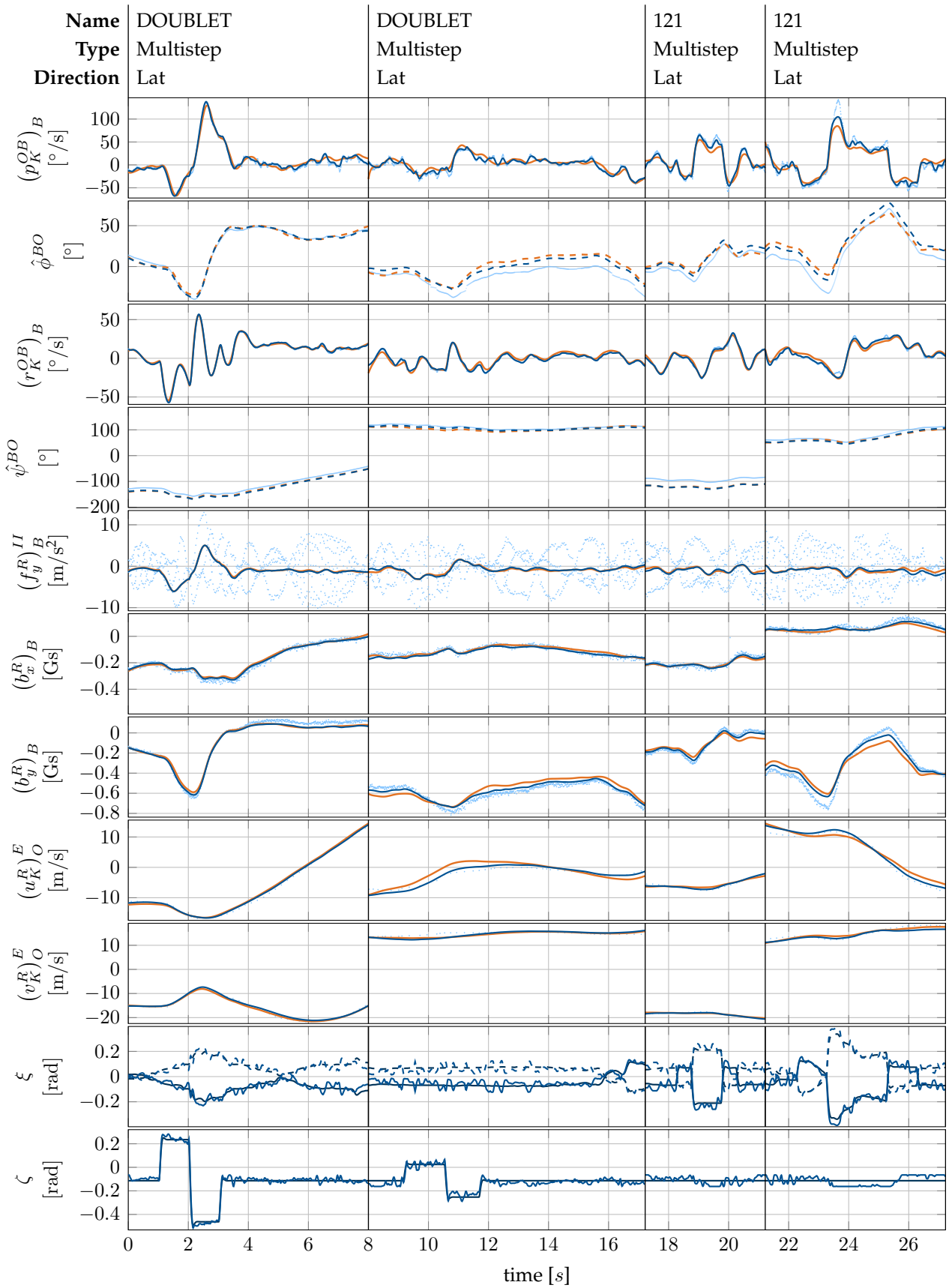


Figure 6.23: *Inverse simulation results for lateral plane; measurements $\boldsymbol{z}$ (·), estimated model outputs $\boldsymbol{y}$ (considered —— / unconsidered - - -), and control inputs $\boldsymbol{u}$ (——); Corresponding inverse simulation results (——) are shown on top*

motion data for maneuvers conducted to excite the longitudinal plane and vice versa, may be found in figures G.3 and G.4 in the appendix.

The output fit was already very good after the original estimation. Now, the additional control effort is mainly used to make the trajectory follow smaller oscillations. These are probably rooted in unsteady air, which, together with the small inertia of the SKYMULE, lead to notable effects. This tracking of smaller oscillations is most notable in the longitudinal motion (figure 6.22) during the two "PULSE" maneuvers: the elevator, which originally only showed small variation throughout large parts of the maneuver, is now used to almost perfectly track the pitch rate signal. The additional control effort, which is necessary to achieve this, may be seen in the two bottom plots of figure 6.22, or magnified in the $\Delta\eta$ plots of figure 6.24. For the pulse maneuvers, its magnitude is small enough to be acceptable. The situation is slightly different for the short-period motion, i.e. the 3-2-1-1 maneuvers. Here the tracking of the pitch rate is not as perfect, and it comes at a somewhat higher elevator activity. Sometimes the $\Delta\eta$ signal is even at its boundary, indicating that the algorithm has trouble to perfectly follow the measured trajectory. Nevertheless, this "optimal tracking" of the pitch rate, and consequently the attitude via the magnetometer readings $\left(\vec{\boldsymbol{b}}^R\right)_B$, is considered good enough to accept the longitudinal rotational dynamics part of the model.

When it comes to the vertical velocity and altitude match, the model validation step confirms the suspicions that were expressed in the parameter estimation section: The model is not capable, not even with optimized additional control effort, to track neither the GNSS measured $\left(w_K^R\right)_O^E$ and h^R nor their air data counterparts $\bar{q}$ and p_{stat}. Again, this is probably due to the rather crude propulsion model, which then disadvantageously influences drag and lift characteristics.

In the lateral plane, almost perfect tracking can be achieved for the yaw rate, with acceptable additional rudder deflections, even if they are somewhat larger than the corresponding elevator inputs. This cannot be said for the roll rate, where tracking is still acceptable, but notably worse compared to $\left(r_K^{OB}\right)_B$ and $\left(q_K^{OB}\right)_B$. This again confirms the suspicion that the roll motion is not as well modeled as the other parts of the rotational dynamics. The inverse simulation experiment actually shows that this is not only the case for the obvious deficiency when it comes to the large amplitude spike in the 1-2-1 maneuver (figure 6.23). Also for the remainder of the maneuvers the quality of the tracking of $\left(p_K^{OB}\right)_B$ is not as high as for the other rotational rates.

The match of the magnetometer readings for the lateral maneuvers is improved through inverse simulation: more minor oscillations can be replicated, leading to smaller errors in the rotational rates and consequently smaller attitude errors. Nevertheless, the roll rate deficiencies also show their effect here, their overall match is not as good as for the longitudinal case. North and east velocities are mapped very well, although some discrepancies remain. This is probably due to errors in heading and bank, and to a somewhat lesser extend due to lift and drag characteristics that are slightly off.

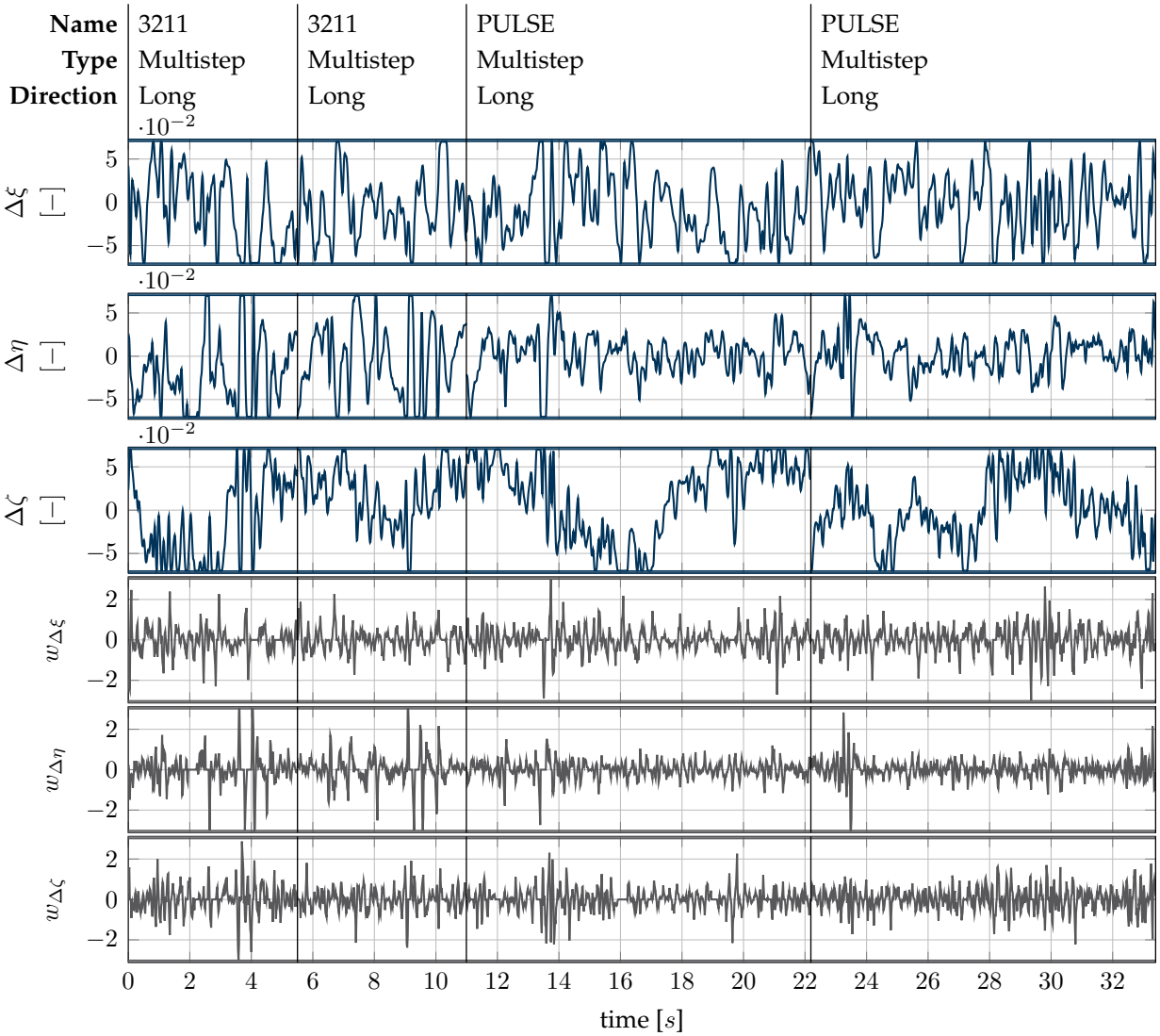


Figure 6.24: *Additional control inputs for maneuvers in longitudinal plane; Δ-states (——), representing additional control inputs and process noise $\boldsymbol{w}$ (——), representing their derivatives*

6.3.3 Conclusion – Skymule

The last two sections illustrated, how the algorithms detailed in this thesis may be applied to full, non-linear 6-DOF RPAS identification. They illustrated most of the novel ideas:

- both problems used the stochastic full-discretization formulation of section 4.4; in the first formulation, the stochastic part represented atmospheric influences, whereas in the second example it was used to account for additional control inputs in an inverse simulation setting.
- Initial model parameter guesses were partly computed based on the improvement approaches of section 3.8;
- the residual covariance estimate (section 3.1) was scaled to account for different numbers of samples using the approach of section 3.10.2;

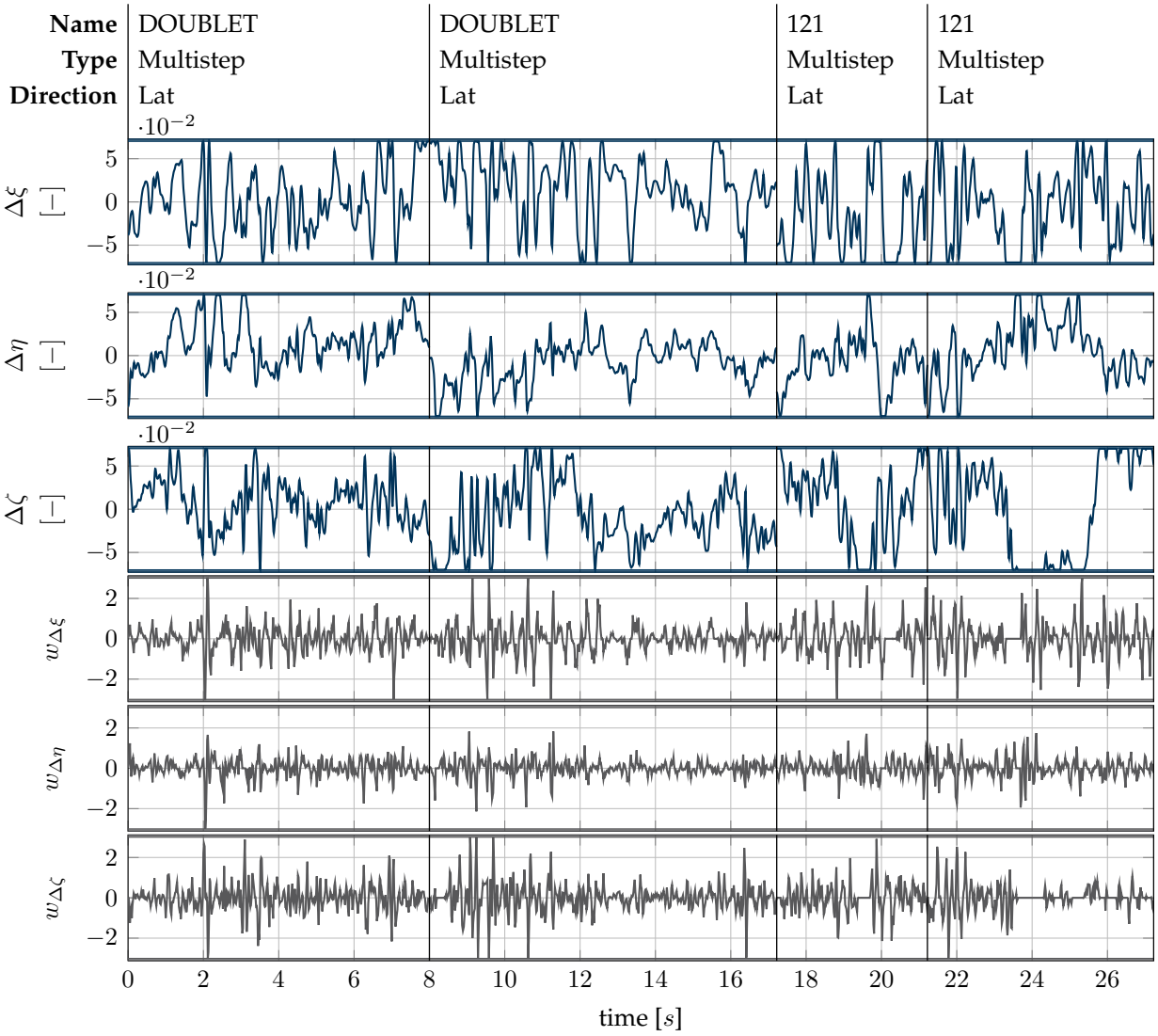


Figure 6.25: *Additional control inputs for maneuvers in lateral plane; Δ-states (——), representing additional control inputs and process noise $\boldsymbol{w}$ (——), representing their derivatives*

- when testing different model formulations, the interpolation approach of section 3.9 helped to keep computation times short
- uncertainty quantification was only possible by using the novel approach of section 3.5

The very good estimation results show, how the novel approaches developed in this thesis may be applied in a very challenging, realistic scenario. Especially the increased robustness due to using full discretization is of great importance. Together with the initial guess improvement step it enabled convergence to meaningful results in the first place, although this part of the process was not as straight forward as for simpler examples.

Further, the implementation that was developed facilitated the analysis greatly: merely few lines of coded needed to be altered in order to test different model formulations, initial guesses, versions of the cost function and estimation algorithms. With

this powerful tool at hand, the analysis could focus on the actual data, model and estimation process, rather than considering low-level implementation aspects.

Admittedly, the model still shows room for improvement, especially in the parameters of the longitudinal plane. However, this is to be considered a problem of the data, rather than of the estimation algorithm in use. A first step could be to collect more flight data by exciting the longitudinal motion at other trim speeds. If this does not lead to the desired improvements, further investigations of the propulsion system (e.g. laboratory and/or wind-tunnel experiments to characterize its behavior) will definitely be beneficial. Lastly, approximated mass and inertia properties could be measured directly. All these inaccuracies are currently compensated for by the estimated aerodynamic parameters, a problem which would be alleviated if the respective aspects were better known. The roll motion also shows some deficiencies, but they are minor compared to the effects of an improved thrust, drag and lift model, which is why these aspects should be addressed first.

Although a discussion of the deficiencies of the model has to be included for a serious presentation of the results, its overall quality is still very good: The overall output fit is absolutely satisfactory, model parameters have favorable statistical properties, and the inverse simulation showed a generally good agreement with only few aspects that cannot be replicated properly with the model at hand. Merely the match of the estimated and complementary-filtered attitudes is observed not to be adequate at times. However, since this happens both in the estimation and in the inverse simulation case, the reason for this is probably insufficient tuning of the complementary filter, rather than an inadequate model.

To summarize, improving the model's quality has to entail the collection of further data, rather than adjusting the model structure or improving the estimation algorithm. The latter performed remarkably well considering the problem size, model complexity and high noise levels in the data.

The most probable value of the unknown quantities will be that in which the sum of the squares of the differences between the actually observed and the computed values multiplied by numbers that measure the degree of precision is a minimum.

Carl Friedrich Gauss, 1857 [Gau1857]

7

Conclusion and Outlook

This last chapter sums up the main findings of this thesis, and lists possible topics for future research.

7.1 Summary and Conclusion

The thesis at hand was intended to bring the two fields of optimal control and parameter estimation closer together. The successful application of the newly developed approaches to several examples of increasing complexity shows that this original goal was successfully met.

In order to be able to utilize any arising synergies, chapter 2 first gathered the necessary theoretical basics from optimal control theory, statistical estimation, and state estimation. In doing so, a common notational framework was set up that allowed the consideration of optimal control as well as parameter estimation aspects. This collection of fundamentals then enables the identification of those aspects of modern optimal control theory that may be used advantageously for parameter estimation. First and foremost, using full discretization to transcribe the resulting problem into a Non-Linear Programming (NLP) problem was found to be most valuable in terms of robustness and straight forward incorporation of constraints.

After having the basics in place, chapter 3 continued to discuss those challenges that appear in real-life application of optimal control and parameter estimation. Many well established routines were collected that help to overcome these difficulties, namely an approach to estimate the residual covariance matrix from the measured data (section 3.1); expressions for the cost function's first and second derivatives (section 3.2);

and methods to compute analytic output sensitivities (section 3.3); Next to those well-established approaches, some of this thesis' main contributions are assembled in chapter 3.

The first to be mentioned is an approach to reduce problem size and thus computation time. It is based on spline interpolation and the numeric, variable step-size solution to non-linear differential equations (section 3.9). The novel idea is not only applicable in the context of full discretization but also when using the standard single shooting approach. In both cases computation times can be drastically reduced, which is of great benefit in the early stages of the parameter estimation process when many different model formulations are tested and compared.

Further, a new scaling scheme is devised, which enables weighting of different outputs. This in turn may be used to simultaneously consider data sources with different sampling frequencies (section 3.10), both in fully discretized and single shooting problems. The scaling approach may thus be used to integrate the flight path reconstruction step more tightly with the actual model parameter estimation by directly using raw measurement data without lengthy preprocessing. This had not been possible before.

The combination of optimal control methods and parameter estimation does not only offer many advantages, such as increased robustness, direct applicability to unstable systems, easy incorporation of constraints or the applicability of mature NLP solver implementations. It also poses some challenges, where the most important ones for practical application have however been overcome in this work. The first is related to uncertainty quantification when post-processing parameter estimation results. No system identification project can be complete without a measure for the uncertainty involved, but the standard methods are not able to incorporate constraints in the determination of covariances. However, in full discretization the complete system dynamics are *purely* enforced via integration defects that are implemented as equality constraints. To solve this issue, a novel uncertainty quantification scheme for parameter estimation using full discretization has been developed (section 3.5). It is based on general considerations of parameter bounds and prior work on constrained parameter estimation, which is then applied to fully discretized parameter estimation problems (section 3.4). It is further shown, that this novel scheme actually contains the standard textbook approach for specific problem formulations on the one hand. On the other hand it may be applied to unstable systems without further modifications, which is not the case for standard methods. Only this new framework allows for a meaningful application of the methods presented, since now a complete discussion of the results involving parameter values and their uncertainty is possible.

Next, a novel approach to improve upon initial model parameter guesses is shown, which is closely related to the idea of full discretization. This helps to make the overall process more robust: the improved initial model parameter guesses were often found to lie close enough to the final result to enable smooth convergence, even if the original

initial guesses were set to zero (section 3.8). The idea may again be used for standard single shooting as well as fully discretized problem formulations. It greatly facilitates practical parameter estimation, since the need for initial model parameters that are already located close to the final result is alleviated significantly.

In chapter 4 four different possibilities were discussed on how to formulate a state and parameter estimation problem in an optimal control framework. They may be arranged in the following pattern

		System	
		deterministic	stochastic
Transcription	single shooting	case I	case III
Method	full discretization	case II	case IV

The cases in the first row "single shooting" describe the well-established standard approaches, whereas those in the second row "full discretization" have been developed throughout this thesis. For all formulations, the main characteristics, governing ideas and the applicability of the approaches of chapter 3 are collected in order to fully define the respective problem. This completes the discussion of parameter estimation tasks in an optimal control context.

For a complete discussion of the topic of system identification for aerial vehicles using optimal control methods, a brief overview over the actual process is given in chapter 5. It lists the most important aspects of experiment design, flight testing, data acquisition, flight path reconstruction, modeling, parameter estimation, and model validation. All of those topics are discussed with Remotely Piloted Aerial System (RPAS) applications in mind.

Everything that had been discussed in this work was implemented in an add-on to the optimal control toolbox FALCON.m [RBG+2018]. It was realized that although the optimal control based methods offer many advantages, they are not necessarily the best choice in all possible applications. Thus the focus of this add-on was on modularity in order to be able to quickly change between solution methodologies. The user is then able to solve the same parameter estimation problem with different transcription methods, NLP solvers, different settings, and different data sets by altering few lines of code. Additionally, most of the novel features (residual covariance estimation, interpolation, initial guess improvement, ...) may easily be de-/activated.

Eventually, three sets of application examples show the applicability of the novel ideas in chapter 6, where parameter estimates together with the respective uncertainty quantification have been computed in different settings. The first two cases show the superior results to be obtained if the system under consideration is unstable. The next set illustrates the methods when treating stochastic systems. The last two examples treat the most complex case: a 6-DOF simulation model of the SKYMULE is constructed, purely based on low-cost sensor data. Most of the novel ideas of chapter 3 come into

play in order to eventually determine good model parameter values. In both SKYMULE cases, the model is formulated as being stochastic: during the actual parameter estimation, wind components are considered as random inputs; in the inverse simulation example, the additional control effort necessary to perfectly track the recorded data is modeled as random walk. Although in both cases the process noise vector is made as small as possible in order to reduce the problem size, the resulting optimization vectors still have around 1×10^5 elements. Nevertheless, this did not pose a problem: the implemented parameter estimation add-on, together with the base implementation of FALCON.m [RBG+2018] found meaningful results.

Overall, this thesis showed the successful application of optimal control methods to aircraft system identification problems. The biggest advantages of this are the increased robustness of full discretization, together with the availability of many mathematical and numerical tools for the solution of large scale problems, such as FALCON.m [RBG+2018]. Further, when including constraints both in the estimation and uncertainty quantification many new possibilities arise, such as explicitly enforcing energy dissipation or keeping attitude quaternions at unit length. This is bought with somewhat longer computational times compared to the standard single shooting algorithms. However, often it can already be seen during early iterations, if the current problem formulation is likely to yield good results, or if the optimization should be aborted and the model reformulated. Thus the longer computational times only really matter when having to wait for convergence of the NLP solver in generating final estimation results.

The classic textbook approaches on the other hand, have been successfully applied for several decades now. The computation times in these cases tend to be shorter, but the algorithms rely heavily on good initial model parameters. In the implemented add-on, developed during the work on this thesis, these two approaches are combined in order to obtain the best of both worlds.

7.2 Outlook

Although the fundamental challenges that arise when applying optimal control methods to parameter estimation problems have been solved in this thesis, some aspects for future research remain. Further, new possibilities are opened up by combining the two fields. Possible topics for future work may be separated in two classes: performance improvement in the solution of the resulting NLP problem on the one hand, and broadening the spectrum of possible applications on the other hand.

Some performance issues that have been identified are

- Although very good approximations to the cost function Hessian are available, they are currently only used in the implementation of the single shooting method

and in the uncertainty quantification step. Memory limitations have so far prohibited its use when transcribing the problem using full-discretization, although the sparsity of all influencing factors had been exploited. Future work could try to find ways to incorporate this knowledge, either exact or by providing meaningful limited memory approximations.

- Although the methods developed in this thesis allow to properly incorporate constraints in a parameter estimation problem, the potential of this aspect is definitely not exhausted yet. Here the focus was on incorporating integration defects, but other applications may be imagined: for example, constraints may be used to force the state and/or output trajectory to lie close to the measured values. This may improve robustness especially in the early stages of the optimization process, when model parameters are still far from their optimum and integration defect equations are not yet fully satisfied. Together with the proposed improvement step (section 3.8), this may even render meaningful initial model parameters completely unnecessary.
- The actual solution of the resulting NLP problem often followed the same three-stage pattern: at first the solver tried to make the trajectory feasible, which usually deteriorated the output fit. Then model parameters and states were adjusted in order to fit the outputs as good as possible, where the output plots showed very good agreement after about $50\,\%$ to $60\,\%$ of the overall optimization time. During the last stage, the solver only took small steps, which resulted in no visible change of the outputs, before finally declaring the problem solved.

  Intuitively, the two aspects to be improved here would be the initial phase, during which the original good output fit should be conserved as far as possible. Maybe the previously discussed additional constraints could help here. The second aspect would be the rather long time in the end, when no visual changes take place anymore. This behavior may be improved through advantageous scaling of the resulting NLP problem to arrive at the final solution more quickly.
- The last aspect of performance improvement would be a thorough investigation into the quantities that influence the number of necessary iterations. Sometimes seemingly small changes in the problem definition (e.g. adding prior information of the state initial guesses) resulted in hugely different computation times. Fully understanding these dependencies may help to streamline the process in the future, by giving proper guidelines on how to set up a parameter estimation problem to enable an efficient solution.

To enable more diverse applications, the following methodological aspects may be investigated further

- Although the assumption of normally distributed residuals is most widely spread, and very attractive due to its close relationship to least squares estimation, it may

be interesting to investigate other shapes of the governing probability density function (pdf). This may be considered for parts of the residuals only, e.g. using a different pdf for different signal sources such as Global Navigation Satellite System (GNSS) versus Inertial Measurement Unit (IMU) data.

- The second aspect to be re-evaluated is the basic assumption that residuals are white. Different courses of action may be envisioned
  - An additional term could be introduced in the cost function, which punishes non-white residuals, i.e. which compares a sample autocorrelation function of the residuals with its desired, white version $\mathrm{Cov}[\boldsymbol{r}_j, \boldsymbol{r}_k] = \boldsymbol{\mathcal{B}} \cdot \delta_{jk}$ enforcing small covariances between sampling instants.
  - The idea of "adaptive filtering" [CJ2012, Ch. 5.7] [Jaz1970, Ch. 8.11] may be incorporated to tune the process noise covariance matrix to eventually obtain white residuals. This would also solve the problem of finding a meaningful $\mathbf{Q}$ in the first place.
  - Residual coloring may be explicitly included in the estimation process by not only estimating the residual covariance matrix, but an approximation to its complete autocorrelation. Although this significantly complicates the cost function formulation and derivative computation, it would be closer to reality, since usually modeling errors manifest themselves as low-frequency residual content.
- the novel uncertainty determination scheme presented here yields additional information that is not used yet, mainly by providing covariances for all elements of the optimization vector. Currently, the main focus lies on the main diagonal elements of sub-matrices, and possibly cross variances between parameter estimates. It has to be investigated if meaningful interpretations for the off-diagonal elements may be devised and used in the model validation step.

Part IV

Appendix

Appendix A

Mathematical Background

A.1 Calculus

A.1.1 Layout Convention

Neither a consistent numerator nor denominator layout convention for matrix and vector derivatives is strictly kept throughout this thesis. Moreover, derivatives are always applied along the "last" dimension, with rows being the first, columns being the second and possibly higher dimensions. This results in the following form for the derivatives, with $x, y \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^{n_x \times 1}$, $\boldsymbol{y} \in \mathbb{R}^{n_y \times 1}$, $\mathbf{X} \in \mathbb{R}^{n_{x1} \times n_{x2}}$ and $\mathbf{Y} \in \mathbb{R}^{n_{y1} \times n_{y2}}$

scalar by scalar

$$\frac{\partial x}{\partial y} \in \mathbb{R} \tag{A.1}$$

scalar by vector often called the *gradient*, the derivatives appear as *rows* in the coloumn vector

$$\frac{\partial x}{\partial \boldsymbol{y}} = \begin{bmatrix} \frac{\partial x}{\partial [\boldsymbol{y}]_{(1)}} \\ \vdots \\ \frac{\partial x}{\partial [\boldsymbol{y}]_{(n_y)}} \end{bmatrix} \in \mathbb{R}^{n_y \times 1} \tag{A.2}$$

vector by scalar the derivatives appear in the same order as in the original vector

$$\frac{\partial \boldsymbol{x}}{\partial y} = \begin{bmatrix} \frac{\partial [\boldsymbol{x}]_{(1)}}{\partial y} \\ \vdots \\ \frac{\partial [\boldsymbol{x}]_{(n_x)}}{\partial y} \end{bmatrix} \in \mathbb{R}^{n_x \times 1} \tag{A.3}$$

$$\frac{\partial \boldsymbol{x}^\intercal}{\partial y} = \begin{bmatrix} \frac{\partial [\boldsymbol{x}]_{(1)}}{\partial y} & \dots & \frac{\partial [\boldsymbol{x}]_{(n_x)}}{\partial y} \end{bmatrix} \in \mathbb{R}^{1 \times n_x} \tag{A.4}$$

vector by vector the rows contain the elements of $\boldsymbol{x}$, and the columns the derivatives with respect to the elements in $\boldsymbol{y}$

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}} = \begin{bmatrix} \frac{\partial [\boldsymbol{x}]_{(1)}}{\partial [\boldsymbol{y}]_{(1)}} & \cdots & \frac{\partial [\boldsymbol{x}]_{(1)}}{\partial [\boldsymbol{y}]_{(n_y)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial [\boldsymbol{x}]_{(n_x)}}{\partial [\boldsymbol{y}]_{(1)}} & \cdots & \frac{\partial [\boldsymbol{x}]_{(n_x)}}{\partial [\boldsymbol{y}]_{(n_y)}} \end{bmatrix} \in \mathbb{R}^{n_x \times n_y} \tag{A.5}$$

This introduces a slight inconsistency, if the vector $\boldsymbol{x}$ degrades to a scalar, since the resulting Jacobian would be a row vector. However, as in [CJ2012,] it is considered unproblematic, since the dimensionality will be apparent from the context.

The matrix of second derivatives of a scalar function, i.e. the Hessian, is constructed in the same way

$$\frac{\partial}{\partial \boldsymbol{y}} \frac{\partial x}{\partial \boldsymbol{y}} = \frac{\partial^2 x}{\partial \boldsymbol{y}^2} = \begin{bmatrix} \frac{\partial^2 x}{\partial [\boldsymbol{y}]_{(1)}{}^2} & \cdots & \frac{\partial}{\partial [\boldsymbol{y}]_{(n_y)}} \frac{\partial x}{\partial [\boldsymbol{y}]_{(1)}} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial [\boldsymbol{y}]_{(1)}} \frac{\partial x}{\partial [\boldsymbol{y}]_{(n_y)}} & \cdots & \frac{\partial^2 x}{\partial [\boldsymbol{y}]_{(n_y)}{}^2} \end{bmatrix} \in \mathbb{R}^{n_y \times n_y} \tag{A.6}$$

matrix by scalar the layout is the same as the original matrix $\mathbf{X}$

$$\frac{\partial \mathbf{X}}{\partial y} = \begin{bmatrix} \frac{\partial [\mathbf{X}]_{(1,1)}}{\partial y} & \cdots & \frac{\partial [\mathbf{X}]_{(1,n_{x2})}}{\partial y} \\ \vdots & \ddots & \vdots \\ \frac{\partial [\mathbf{X}]_{(n_{x1},1)}}{\partial y} & \cdots & \frac{\partial [\mathbf{X}]_{(n_{x1},n_{x2})}}{\partial y} \end{bmatrix} \in \mathbb{R}^{n_{x1} \times n_{x2}} \tag{A.7}$$

scalar by matrix the layout is the same as the original matrix $\mathbf{Y}$

$$\frac{\partial x}{\partial \mathbf{Y}} = \begin{bmatrix} \frac{\partial x}{\partial [\mathbf{Y}]_{(1,1)}} & \cdots & \frac{\partial x}{\partial [\mathbf{Y}]_{(1,n_{y2})}} \\ \vdots & \ddots & \vdots \\ \frac{\partial x}{\partial [\mathbf{Y}]_{(n_{y1},1)}} & \cdot & \frac{\partial x}{\partial [\mathbf{Y}]_{(n_{y1},n_{y2})}} \end{bmatrix} \in \mathbb{R}^{n_{y1} \times n_{y2}} \tag{A.8}$$

A.1.2 Matrix Derivatives

The following rules are taken from [BS2012] and adapted to fit the layout conventions used here. A similar set of derivative rules can be found in [CJ2012, App. A.5]. The derivative of the trace of a matrix inverse is

$$\frac{\partial \mathrm{tr}\left[\mathbf{X}^{-1}\mathbf{A}\right]}{\partial \mathbf{X}} = -\mathbf{X}^{-1}\mathbf{A}\mathbf{X}^{-1} \tag{A.9}$$

The derivative of the logarithm of the determinant of a matrix is

$$\frac{\partial \ln|a\mathbf{X}|}{\partial \mathbf{X}} = \mathbf{X}^{-1} \tag{A.10}$$

The derivative of the trace of a linear matrix expression is

$$\frac{\partial \text{tr}[\mathbf{X}\mathbf{A}]}{\partial \mathbf{X}} = \mathbf{A}^\intercal \tag{A.11}$$

$$\frac{\partial \text{tr}[\mathbf{A}\mathbf{X}^\intercal]}{\partial \mathbf{X}} = \mathbf{A} \tag{A.12}$$

The derivative of the trace of a quadratic matrix expression is

$$\frac{\partial \text{tr}[\mathbf{X}\mathbf{A}\mathbf{X}^\intercal]}{\partial \mathbf{X}} = \begin{cases} 2\mathbf{X}\mathbf{A} & \text{if } \mathbf{A} = \mathbf{A}^\intercal \\ \mathbf{X}\mathbf{A}^\intercal + \mathbf{X}\mathbf{A} & \text{otherwise} \end{cases} \tag{A.13}$$

A.1.3 Inverse Function Theorem

The inverse Function Theorem in higher dimensions states that for a function $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ with $\boldsymbol{f} : \mathbb{R}^{n_x} \longrightarrow \mathbb{R}^{n_y}$, $\boldsymbol{f} \in \mathcal{C}^1$, if $\left|\frac{\partial \boldsymbol{f}(\boldsymbol{x}_0)}{\partial \boldsymbol{x}}\right| \neq 0$ then $\boldsymbol{f}$ is invertible near $\boldsymbol{x}_0$, i.e. the system of equations

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{A.14}$$

has a unique solution $\boldsymbol{x} = \boldsymbol{f}^{-1}(\boldsymbol{y})$.

Finally, the theorem states further, that $\boldsymbol{f}^{-1} \in \mathcal{C}^1$ and the Jacobians of the function and its inverse follow the relation

$$\frac{\partial \boldsymbol{f}^{-1}(\boldsymbol{y})}{\partial \boldsymbol{y}} = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^{-1} \tag{A.15}$$

Consequently, their determinants are [MV2003, Ch. 5]

$$\left|\frac{\partial \boldsymbol{f}^{-1}}{\partial \boldsymbol{y}}\right| = \frac{1}{\left|\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|} \tag{A.16}$$

A.2 Linear Algebra

A.2.1 Singular Value Decomposition

For general rectangular matrices $\mathbf{A} \in \mathbb{R}^{n_m \times n_n}$ the following is called a *singular value decomposition*

$$\mathbf{A} = \begin{cases} \mathbf{U}\begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} \end{bmatrix}\mathbf{V}^\intercal & \text{if } m < n \\ \mathbf{U}\begin{bmatrix} \boldsymbol{\Sigma} \\ \mathbf{0} \end{bmatrix}\mathbf{V}^\intercal & \text{if } m > n \end{cases} \tag{A.17}$$

if the left, and right singular matrices $\mathbf{U} \in \mathbb{R}^{n_m \times n_m}$ and $\mathbf{V} \in \mathbb{R}^{n_n \times n_n}$ are ortho-normal

$$\mathbf{U}\mathbf{U}^\intercal = \mathbf{I}_m \tag{A.18}$$

$$\mathbf{V}\mathbf{V}^\intercal = \mathbf{I}_n \tag{A.19}$$

and the matrix of singular values $\mathbf{\Sigma} \in \mathbb{R}^{n_m \times n_n}$ has only non-negative values on the main diagonal.

The Singular Value Decomposition (SVD) is related to the eigendecomposition via (only the case $m < n$ is considered, analogous facts hold for $m > n$)

$$\mathbf{A}\mathbf{A}^\intercal = \mathbf{T}\mathbf{D}\mathbf{T}^\intercal = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^\intercal \tag{A.20}$$

$$\mathbf{A}^\intercal\mathbf{A} = \tilde{\mathbf{T}}\tilde{\mathbf{D}}\tilde{\mathbf{T}}^\intercal = \mathbf{V}\begin{bmatrix} \mathbf{\Sigma}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{V}^\intercal \tag{A.21}$$

Thus the left singular matrix $\mathbf{U}$ is made up of the eigenvectors of $\mathbf{A}\mathbf{A}^\intercal$, whereas $\mathbf{V}$ is made up of the eigenvectors of $\mathbf{A}^\intercal\mathbf{A}$. The matrix of singular values $\mathbf{\Sigma}$ then contains the square roots of the eigenvalues of $\mathbf{A}\mathbf{A}^\intercal$, which are non-negative since $\mathbf{A}\mathbf{A}^\intercal$ is symmetric.

Also, a null-space basis of $\mathbf{A}$ may be determined based on the SVD. If there are zero-submatrices in the decomposition (either because some singular vales are zero, or because $m < n$), the SVD may be partitioned into

$$\mathbf{A} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}\begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{V}_1^\intercal \\ \mathbf{Z}^\intercal \end{bmatrix} \tag{A.22}$$

Any vector that is located within the range of $\mathbf{Z}$, i.e. that may be expressed as $\mathbf{Z}\boldsymbol{v}$ is mapped onto zero

$$\mathbf{A}\cdot\mathbf{Z}\boldsymbol{v} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}\begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{V}_1^\intercal\mathbf{Z}\boldsymbol{v} \\ \mathbf{Z}^\intercal\mathbf{Z}\boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}\begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{0} \\ \boldsymbol{v} \end{bmatrix} = \mathbf{0} \tag{A.23}$$

the multiplication $\mathbf{V}_1\mathbf{Z} = \mathbf{0}$, since the right singular matrix is ortho-normal. Further, $\mathbf{Z}^\intercal\mathbf{Z}\cdot\boldsymbol{v} = \boldsymbol{v}$ for the same reason. Thus, $\mathbf{Z}$ constitutes a null-space basis of $\mathbf{A}$.

A.2.2 Generalized Matrix Inverse

A generalized inverse is defined in terms of the PENROSE equations [BIG2003, Ch. 1]. For every real, finite matrix $\mathbf{A}$ there is a unique, real matrix $\mathbf{X}$ satisfying

$$\mathbf{A}\mathbf{X}\mathbf{A} = \mathbf{A} \tag{A.24}$$

$$\mathbf{X}\mathbf{A}\mathbf{X} = \mathbf{X} \tag{A.25}$$

$$(\mathbf{A}\mathbf{X})^\intercal = \mathbf{A}\mathbf{X} \tag{A.26}$$

$$(\mathbf{X}\mathbf{A})^\intercal = \mathbf{X}\mathbf{A} \tag{A.27}$$

in this work, this unique generalized inverse, which is often called the MOORE-PENROSE inverse, is denoted as $\mathbf{X} = \mathbf{A}^\dagger$.

For square, non-singular matrices, above simplifies to the conventional inverse $\mathbf{A}^\dagger = \mathbf{A}^{-1}$. For matrices with full column rank the pseudo inverse can be expressed

via

$$\mathbf{A}^{\dagger} = (\mathbf{A}^{\intercal}\mathbf{A})^{-1}\,\mathbf{A}^{\intercal} \tag{A.28}$$

whereas for matrices with full row rank the pseudo inverse becomes

$$\mathbf{A}^{\dagger} = \mathbf{A}^{\intercal}(\mathbf{A}\mathbf{A}^{\intercal})^{-1} \tag{A.29}$$

For general, rank deficient, real, rectangular matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m < n$, the pseudo inverse can be computed, using the SVD (see Appendix A.2.1) [BIG2003, Ch. 6]

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\intercal} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^{\intercal} \\ \mathbf{V}_2^{\intercal} \\ \mathbf{V}_3^{\intercal} \end{bmatrix} \tag{A.30}$$

The pseudo inverse $\mathbf{A}^{\dagger}$ is then [BIG2003, Ch. 6]

$$\mathbf{A}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{\dagger}\mathbf{U}^{\intercal} = \mathbf{V}_1\mathbf{\Sigma}_1^{\dagger}\mathbf{U}_1^{\intercal} \tag{A.31}$$

$$\mathbf{\Sigma}^{\dagger} = \begin{bmatrix} \frac{1}{\sigma_1} & & & \\ & \ddots & & \mathbf{0} \\ & & \frac{1}{\sigma_p} & \\ & \mathbf{0} & & \mathbf{0} \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_1^{\dagger} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.32}$$

which can be verified by plugging this into the PENROSE equations (A.24) - (A.27).

A.3 Spline Interpolation

The following presentation of basic properties of B-Splines is taken from [Boo1991]. Using a non-decreasing vector of time instants t_j as knot sequence

$$\mathbf{t} = \{t_j : j = 0, \ldots, N_{knots} - 1\} \tag{A.33}$$

the B-Spline basis functions of order 1 are

$${}_{\mathbf{t}}B_j^1(t) = \begin{cases} 1 & \text{if } t_j \le t < t_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{A.34}$$

Higher order basis functions are then constructed by the recursion

$${}_{\mathbf{t}}B_j^n(t) = \tau_{j,n}(t)\,{}_{\mathbf{t}}B_j^{n-1}(t) + (1 - \tau_{j+1,n}(t))\,{}_{\mathbf{t}}B_{j+1}^{n-1}(t) \tag{A.35}$$

$$\tau_{j,n}(t) = \begin{cases} \frac{t-t_j}{t_{j+n-1}-t_j} & \text{if } t_j \neq t_{j+n-1} \\ 0 & \text{otherwise} \end{cases} \tag{A.36}$$

One of the advantages of B-splines is their local support, i.e. they are only non-zero locally

$$ {}_{\mathbf{t}}B_j^n(t) = 0 \text{ for } t \notin [t_j, t_{j+n}] \tag{A.37} $$

Non-distinct knots t_j lead to cases, where the B-Spline looses smoothness, i.e. full continuous differentiability is not given anymore, it can only be obtained up to "$n -$ # non-distinct t_j". For interior knots, this is in our cases physically impossible, however this fact is used at the beginning and the end of a time-series: by setting $t_0 = t_1 = t_2 = \ldots$ and $\cdots = t_{N_{knots}-2} = t_{N_{knots}-1} = t_{N_{knots}}$, i.e. using the first and last value with a multiplicity of n in the knot series, a jump at the beginning and the end of the time-series is possible. Values outside of the considered time vector are then assumed to be zero, and no derivatives have to be specified at its boundaries. Also, unwanted oscillations can be avoided this way.

Furthermore, the ${}_{\mathbf{t}}B_j^n(t)$ are completely determined by the knot sequence $\mathbf{t}$. A spline of n-th order is then a linear combination of the B-splines ${}_{\mathbf{t}}B_j^n(t)$

$$ {}_{\mathbf{t}}S^n(t) = \sum_j \alpha_j \, {}_{\mathbf{t}}B_j^n(t) \qquad \alpha_j \in \mathbb{R} \tag{A.38} $$

A.4 Statistical Basics

A.4.1 Random Variables

Most of the following presentations can be found in any textbook on probability theory, e.g. [CS2011, CB2001, Sha2003, HH2015]. The selected topics regarding random variables follow those presented in [Jaz1970], since many of the basic ideas regarding estimation theory are based on this source.

The basis for probability theory is the *probability space*, or *basic sample space* Ω, with elements (samples, experimental outcomes) ω. Combinations of the ω make up *events*, and the probability function $Pr\{\omega\}$ assigns a probability to the event ω. The nature of $Pr\{\omega\}$ is defined via KOLMOGOROV's axioms [Jaz1970, p. 11]. For more details on these definitions, any statistics textbook can be consulted, e.g. [Jaz1970, CS2011]. The latter underline the "relative frequency" interpretation of probabilities, i.e. if the assigned probability of event ω_1 is larger than that of ω_2, the event ω_1 will, in average, occur more often when conducting the underlying experiment.

Let the function $\mathrm{x}(\omega)$ be real, and defined on the basic sample space Ω. Then it is called a *random variable*, if, for every real number x the inequality

$$ \mathrm{x}(\omega) \leq x \tag{A.39} $$

defines an ω set, whose probability is defined [Jaz1970, p. 13]. Intuitively, the random variable maps any random event to the real number line. Notation-wise x denotes a random variable, and x is its *realizations*.

The probability, which is assigned to above relation is the *cumulative distribution function* of the random variable x

$$F_{\mathrm{x}}(x) = Pr\{\mathrm{x} \leq x\} \tag{A.40}$$

where x in the notation of the cumulative distribution serves to identify the relevant random variable [Jaz1970]. Wherever it is clear from the context, the dependency on the respective random variable will not be noted explicitly. Here, random variables are always considered to be defined on the whole of $\mathbb{R}$: $-\infty < x < \infty$, i.e. no complicated set geometries are considered.

The cumulative distribution function completely describes the properties of a random variable. Some properties are

$$\lim_{x\to\infty} F_{\mathrm{x}}(x) = 1 \tag{A.41}$$

$$\lim_{x\to-\infty} F_{\mathrm{x}}(x) = 0 \tag{A.42}$$

$$Pr\{x_1 < \mathrm{x}(\omega) \leq x_2\} = F_{\mathrm{x}}(x_2) - F_{\mathrm{x}}(x_1) \tag{A.43}$$

$$Pr\{x_1 < \mathrm{x}(\omega)\} = 1 - F_{\mathrm{x}}(x_1) \tag{A.44}$$

$$\begin{aligned} Pr\{x = \mathrm{x}(\omega)\} &= \lim_{\epsilon\to 0} Pr\{x - \epsilon < \mathrm{x}(\omega) \leq x\} \\ &= F_{\mathrm{x}}(x) - \lim_{\epsilon\to 0} F_{\mathrm{x}}(x-\epsilon) = F_{\mathrm{x}}(x) - F_{\mathrm{x}}\left(x^{-}\right) \end{aligned} \tag{A.45}$$

If the random variable x can only assume a countable number of values, it is called discrete. It can then equivalently be characterized by its probability mass function $m_{\mathrm{x}}(x)$, which, in the light of the foregoing equations, has the following properties [Jaz1970]

$$m_{\mathrm{x}}(x) = Pr\{\mathrm{x}(\omega) = x\} = F_{\mathrm{x}}(x) - F_{\mathrm{x}}\left(x^{-}\right) \tag{A.46}$$

$$0 \leq m_{\mathrm{x}}(x) \leq 1 \tag{A.47}$$

$$F_{\mathrm{x}}(x) = \sum_{\xi\leq x} m_{\mathrm{x}}(\xi) \tag{A.48}$$

$$\sum_{\xi} m_{\mathrm{x}}(\xi) = 1 \tag{A.49}$$

Thus, for discrete random variables, the cumulative distribution function has discontinuous jumps at a countable number of values.

If there exists a probability density function $p_{\mathrm{x}}(x)$, s.t.

$$F_{\mathrm{x}}(x) = \int_{-\infty}^{x} p_{\mathrm{x}}(\xi)\, d\xi \qquad\qquad -\infty \leq x \leq \infty \tag{A.50}$$

the random variable is called continuous. If the distribution function is absolutely continuous, the inverse holds, too [Jaz1970]

$$p_{\mathrm{x}}(x) = \frac{d}{dx} F_{\mathrm{x}}(x) \tag{A.51}$$

For continuous random variables, the following hold [Jaz1970]

$$\begin{aligned} Pr\{x_1 < \mathrm{x}(\omega) \leq x_2\} &= F_{\mathrm{x}}(x_2) - F_{\mathrm{x}}(x_1) = \int_{-\infty}^{x_2} p_{\mathrm{x}}(\xi)\, d\xi - \int_{-\infty}^{x_1} p_{\mathrm{x}}(\xi)\, d\xi \\ &= \int_{x_1}^{x_2} p_{\mathrm{x}}(\xi)\, d\xi \end{aligned} \tag{A.52}$$

$$Pr\{\mathrm{x}(\omega) = x\} = \lim_{\epsilon \to 0} \int_{x-\epsilon}^{x} p_{\mathrm{x}}(\xi)\, d\xi = 0 \tag{A.53}$$

$$\Rightarrow Pr\{x_1 \leq \mathrm{x}(\omega) \leq x_2\} = Pr\{x_1 < \mathrm{x}(\omega) \leq x_2\} \tag{A.54}$$

$$0 \leq p_{\mathrm{x}}(x) \not\leq 1 \tag{A.55}$$

$$F_{\mathrm{x}}(\infty) = \int_{-\infty}^{\infty} p_{\mathrm{x}}(\xi)\, d\xi = 1 \tag{A.56}$$

A function, satisfying the last two equations is a density function, since it defines a valid cumulative distribution function and thus a valid random variable [Jaz1970, p. 17]. It is tempting to extend this to the Dirac delta function

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \tag{A.57}$$

$$\int_{-\infty}^{\infty} \delta(\tau)\, d\tau = 1 \tag{A.58}$$

which fulfills the requirements. Furthermore, it can formally be used to translate a discrete probability mass function $m_{\mathrm{x}}(x)$ into a quasi-continuous probability density function. Using the sifting property of the Dirac function, the equivalence of the two distribution functions may be shown

$$p_{\mathrm{x}}(x) = \sum_{\xi} \delta(x - \xi)\, m_{\mathrm{x}}(\xi) \tag{A.59}$$

$$F_{\mathrm{x}}(x) = \int_{-\infty}^{x} \sum_{\xi} \delta(x - \xi)\, m_{\mathrm{x}}(\xi)\, d\xi = \sum_{\xi} \int_{-\infty}^{x} \delta(x - \xi)\, m_{\mathrm{x}}(\xi)\, d\xi = \sum_{\xi} m_{\mathrm{x}}(\xi) \tag{A.60}$$

This, however, is an overly simplified illustration. The rigorous treatment of this aspect necessitates advanced mathematical tools (measure theory, generalized functions,...), which are not to be covered here. Nevertheless, this aspect allows for some interesting formal illustrations: e.g. it also allows for the definition of a probability density function for constants c [Jaz1970]

$$p_{\mathrm{c}}(x) = \delta(x - c) \tag{A.61}$$

which translates into the following probability mass function

$$m_{\mathrm{c}}(x) = \begin{cases} 1 & x = c \\ 0 & x \neq c \end{cases} \tag{A.62}$$

A.4.2 Moments

The following illustrations regarding moments are mainly based on [Jaz1970]. The first moment (mean, average) of a continuous random variable is defined as

$$\mathrm{E}[\mathrm{x}] = \int_{-\infty}^{\infty} x \cdot p_{\mathrm{x}}(x)\, dx \tag{A.63}$$

whereas the discrete counterpart is

$$\mathrm{E}[\mathrm{x}] = \sum_{x} x \cdot m_{\mathrm{x}}(x) \tag{A.64}$$

It defines the *mean* of the distribution, and can also be interpreted as its center of gravity. The expectation operator is linear, and the mean of a constant is the constant itself

$$\mathrm{E}[a\mathrm{x} + b\mathrm{y}] = a\mathrm{E}[\mathrm{x}] + b\mathrm{E}[\mathrm{y}] \tag{A.65}$$

The n-th moment of a random variable is defined as

$$\begin{aligned} \mathrm{E}[\mathrm{x}^n] &= \int_{-\infty}^{\infty} x^n \cdot p_{\mathrm{x}}(x)\, dx && \text{(continuous case)} \\ &= \sum_{x} x^n \cdot m_{\mathrm{x}}(x) && \text{(discrete case)} \end{aligned} \tag{A.66}$$

From the last equation, it is obvious, that it does not matter if the random variable is discrete or continuous, when using the proper definition of the expected value. Thus, the distinction between the two will not be carried further. The second moment $n = 2$ is called the *mean square value*.

Some *central* moments (i.e. moments that are shifted by the mean) play an important role; their general definition is

$$\mathrm{E}[(\mathrm{x} - \mathrm{E}[\mathrm{x}])^n] \tag{A.67}$$

The most important higher, central moment to be considered here is the variance $n = 2$, together with its square root, the standard deviation

$$\begin{aligned} \mathrm{Var}[\mathrm{x}] &= \mathrm{E}\Big[(\mathrm{x} - \mathrm{E}[\mathrm{x}])^2\Big] = \mathrm{E}[(\mathrm{x} - \mathrm{E}[\mathrm{x}])(\mathrm{x} - \mathrm{E}[\mathrm{x}])] = \mathrm{E}\Big[\mathrm{x}^2 - 2\mathrm{x}\mathrm{E}[\mathrm{x}] + \mathrm{E}[\mathrm{x}]^2\Big] \\ &= \mathrm{E}\Big[\mathrm{x}^2\Big] - 2\mathrm{E}[\mathrm{x}]\,\mathrm{E}[\mathrm{x}] + \mathrm{E}[\mathrm{x}]^2 \\ &= \mathrm{E}\Big[\mathrm{x}^2\Big] - \mathrm{E}[\mathrm{x}]^2 \end{aligned} \tag{A.68}$$

$$\sigma[\mathrm{x}] = \sqrt{\mathrm{Var}[\mathrm{x}]} \tag{A.69}$$

The variance is computed from the squared deviation of x from its mean, i.e. it can be used as an indicator of how dispersed the random variable is. The standard deviation contains the same information, however it has the advantage, that it has the same units as the original random variable, and is thus easier interpreted.

A.4.3 Multivariate Case

If several random variables $\mathrm{x}_1 \dots \mathrm{x}_{\mathrm{n_x}}$ are defined on the same probability space Ω, they may be treated together. They can be characterized by their joint distribution function [Jaz1970]

$$F_{\mathrm{x}_1,\dots\mathrm{x}_{\mathrm{n_x}}}(x_1, \dots x_{n_x}) = Pr\{\mathrm{x}_1(\omega) \leq x_1 \dots \mathrm{x}_{\mathrm{n_x}}(\omega) \leq x_{n_x}\} \tag{A.70}$$

$$F_{\mathbf{x}}(\boldsymbol{x}) = Pr\{\mathbf{x}(\omega) \leq \boldsymbol{x}\} \tag{A.71}$$

where the second equation uses a shorthand vector notation

$$\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_{n_x} \end{bmatrix}^\mathsf{T} \tag{A.72}$$

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}_1 & \cdots & \mathrm{x}_{\mathrm{n_x}} \end{bmatrix}^\mathsf{T} \tag{A.73}$$

and the inequalities are to be read element-wise. For a continuous random variable, the joint density function is then

$$F_{\mathbf{x}}(\boldsymbol{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_{n_x}} p_{\mathbf{x}}(\xi_1 \dots \xi_{n_x})\, d\xi_1 \dots d\xi_{n_x} \tag{A.74}$$

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{\partial^{n_x}}{\partial x_1 \dots \partial x_{n_x}} F_{\mathbf{x}}(\boldsymbol{x}) \tag{A.75}$$

If one is only interested in some of the random variables, the *marginal* distribution and density function are of interest. They can be obtained by "integrating out", e.g. the k-th random variable

$$F_{\mathrm{x}_1\dots\mathrm{x}_{\mathrm{k-1}},\mathrm{x}_{\mathrm{k+1}}\dots\mathrm{x}_{\mathrm{n_x}}}(x_1 \dots x_{k-1}, x_{k+1} \dots x_{n_x}) = F_{\mathbf{x}}(x_1 \dots x_{k-1}, \infty, x_{k+1} \dots x_{n_x}) \tag{A.76}$$

$$p_{\mathrm{x}_1\dots\mathrm{x}_{\mathrm{k-1}},\mathrm{x}_{\mathrm{k+1}}\dots\mathrm{x}_{\mathrm{n_x}}}(x_1 \dots x_{k-1}, x_{k+1} \dots x_{n_x}) = \int_{-\infty}^{\infty} p_{\mathbf{x}}(\boldsymbol{x})\, dx_k \tag{A.77}$$

The approach can be applied repeatedly to obtain other "marginal" distributions.

A.4.3.1 Moments

The extension to multi-dimensional moments is straightforward. For the mean of a continuous, multi-dimensional random variable one gets [Jaz1970]

$$\begin{aligned} \mathrm{E}[\mathbf{x}] &= \begin{bmatrix} \mathrm{E}[\mathrm{x}_1] \\ \vdots \\ \mathrm{E}[\mathrm{x}_{\mathrm{n_x}}] \end{bmatrix} \\ \text{with } \mathrm{E}[\mathrm{x_i}] &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} x_i p_{\mathbf{x}}(\boldsymbol{x})\, dx_1 \dots dx_{n_x} = \int_{-\infty}^{\infty} x_i p_{\mathbf{x}}(\boldsymbol{x})\, d\boldsymbol{x} \end{aligned} \tag{A.78}$$

The last equality shows a shorthand notation of how the integral is performed over the entire sample space. Extension to the variance is

$$\mathrm{Var}[\mathrm{x_i}] = \mathrm{E}\Big[(\mathrm{x_i} - \mathrm{E}[\mathrm{x_i}])^2\Big] \tag{A.79}$$

where also mixed terms become possible. The joint, second, central moment the so-called cross- or co-variances, play again an important role

$$\mathrm{Cov}[\mathrm{x_i}, \mathrm{x_j}] = \mathrm{E}[(\mathrm{x_i} - \mathrm{E}[\mathrm{x_i}])(\mathrm{x_j} - \mathrm{E}[\mathrm{x_j}])] \tag{A.80}$$

$$\mathrm{Cov}[\mathrm{x_i}, \mathrm{x_i}] = \mathrm{Var}[\mathrm{x_i}] = \mathrm{Cov}[\mathrm{x_i}] \tag{A.81}$$

The second equation shows the relation of the covariance and variance. Since they are closely related, from here on only the name covariance will be used. Whenever only one input argument is listed, this implies the variance of the considered random variable.

Arranging all covariances in a matrix yields the covariance matrix of the vector of random variables

$$\begin{aligned} \mathrm{Cov}[\mathbf{x}] &= \begin{bmatrix} \mathrm{Cov}[\mathrm{x_1}, \mathrm{x_1}] & \cdots & \mathrm{Cov}[\mathrm{x_1}, \mathrm{x_{n_x}}] \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}[\mathrm{x_{n_x}}, \mathrm{x_1}] & \cdots & \mathrm{Cov}[\mathrm{x_{n_x}}, \mathrm{x_{n_x}}] \end{bmatrix} \\ &= \mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])(\mathbf{x} - \mathrm{E}[\mathbf{x}])^\intercal] \\ &= \mathrm{E}[\mathbf{x}\mathbf{x}^\intercal] - \mathrm{E}[\mathbf{x}]\,\mathrm{E}[\mathbf{x}]^\intercal \end{aligned} \tag{A.82}$$

The covariance $\mathrm{Cov}[\mathbf{x}]$ is symmetric and positive definite by construction. The variances of the elements of $\mathbf{x}$ can be found on the main diagonal; standard deviations are computed as the respective square roots

$$\mathrm{Var}[\mathrm{x_i}] = [\mathrm{Cov}[\mathbf{x}]]_{(i,i)} \qquad i = 1 \dots n_x \tag{A.83}$$

$$\sigma[\mathrm{x_i}] = \sqrt{[\mathrm{Cov}[\mathbf{x}]]_{(i,i)}} \tag{A.84}$$

Normalizing the covariance with the two respective standard deviations yields the correlation coefficient

$$\rho[\mathrm{x_i}, \mathrm{x_j}] = \frac{\mathrm{Cov}[\mathrm{x_i}, \mathrm{x_j}]}{\sigma[\mathrm{x_i}]\,\sigma[\mathrm{x_j}]} \tag{A.85}$$

which is in the range -1 to 1 per construction. It can indicate linear dependence between variables, when its absolute value is close to 1; independence when it is close to 0. For this definition to be valid, the two random variables must be non-degenerate, i.e. their variances must not be 0.

Analogously to the covariance, the correlation coefficients can be arranged in a correlation matrix

$$\begin{aligned} \mathrm{Corr}[\mathbf{x}] &= \begin{bmatrix} \rho[\mathrm{x_1}, \mathrm{x_1}] & \cdots & \rho[\mathrm{x_1}, \mathrm{x_{n_x}}] \\ \vdots & \ddots & \vdots \\ \rho[\mathrm{x_{n_x}}, \mathrm{x_1}] & \cdots & \rho[\mathrm{x_{n_x}}, \mathrm{x_{n_x}}] \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sigma[\mathrm{x_1}]} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma[\mathrm{x_{n_x}}]} \end{bmatrix} \mathrm{Cov}[\mathbf{x}] \begin{bmatrix} \frac{1}{\sigma[\mathrm{x_1}]} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma[\mathrm{x_{n_x}}]} \end{bmatrix} \end{aligned} \tag{A.86}$$

It has 1's on the main diagonal per construction.

For two random vectors, the covariance matrix $\mathrm{Cov}[\mathbf{x}, \mathbf{y}]$ is

$$\begin{aligned} \mathrm{Cov}[\mathbf{x}, \mathbf{y}] &= \mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])(\mathbf{y} - \mathrm{E}[\mathbf{y}])^\intercal] \\ \mathrm{Cov}[\mathbf{x}, \mathbf{y}] &\in \mathbb{R}^{n_{\boldsymbol{x}} \times n_{\boldsymbol{y}}} \end{aligned} \tag{A.87}$$

which is neither necessarily symmetric, nor quadratic.

For computations involving covariance matrices, the following rules can be applied [HH2015, Ch. 4.2, p 125], with constants $\mathbf{A}, \mathbf{B}$, and $\boldsymbol{b}, \boldsymbol{c}$ and the random vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$; all of appropriate Dimension

$$\mathrm{Cov}[\mathbf{A}\mathbf{x} + \boldsymbol{b}] = \mathbf{A}\mathrm{Cov}[\mathbf{x}]\,\mathbf{A}^\intercal \tag{A.88}$$

$$\mathrm{Cov}[\mathbf{A}\mathbf{x} + \boldsymbol{b}, \mathbf{B}\mathbf{y} + \boldsymbol{c}] = \mathbf{A}\mathrm{Cov}[\mathbf{x}, \mathbf{y}]\,\mathbf{B}^\intercal \tag{A.89}$$

$$\mathrm{Cov}[\mathbf{x}, \mathbf{y} + \mathbf{z}] = \mathrm{Cov}[\mathbf{x}, \mathbf{y}] + \mathrm{Cov}[\mathbf{x}, \mathbf{z}] \tag{A.90}$$

$$\mathrm{Cov}[\mathbf{x} + \mathbf{y}] = \mathrm{Cov}[\mathbf{x}] + \mathrm{Cov}[\mathbf{y}, \mathbf{x}] + \mathrm{Cov}[\mathbf{x}, \mathbf{y}] + \mathrm{Cov}[\mathbf{y}] \tag{A.91}$$

$$\begin{aligned} \mathrm{Cov}\left[\sum_{i=1}^{N} \mathbf{x}_i\right] &= \sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{Cov}[\mathbf{x}_i, \mathbf{x}_j] \\ &= \sum_{i=1}^{N} \mathrm{Cov}[\mathbf{x}_i] + \sum_{i=1}^{N}\sum_{\substack{j=1 \\ i \neq j}}^{N} \mathrm{Cov}[\mathbf{x}_i, \mathbf{x}_j] \end{aligned} \tag{A.92}$$

A.4.3.2 Transformations

Consider the transformation $\mathbf{y} = \boldsymbol{f}(\mathbf{x})$ of a random variable, with $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n_x}$, where $p_{\mathbf{x}}(\boldsymbol{x})$ is given. Suppose, $\boldsymbol{f}^{-1}$ exists, and both $\boldsymbol{f}$ and $\boldsymbol{f}^{-1}$ are continuously differentiable. Then the transformed probability density $p_{\mathbf{y}}(\boldsymbol{y})$ is [Jaz1970, Theorem 2.7]

$$p_{\mathbf{y}}(\boldsymbol{y}) = p_{\mathbf{x}}\Big(\boldsymbol{f}^{-1}(\boldsymbol{y})\Big)\left|\left|\frac{\partial \boldsymbol{f}^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right|\right| \tag{A.93}$$

where $\left|\left|\frac{\partial \boldsymbol{f}^{-1}(\mathbf{y})}{\partial \mathbf{y}}\right|\right|$ indicates the absolute value of the determinant of the Jacobian of the inverse transformation.

A.4.4 Independence

Two random variables are said to be *independent*, if any of the two following conditions hold [Jaz1970]

$$F_{\mathrm{x_i},\mathrm{x_j}}(x_i, x_j) = F_{\mathrm{x_i}}(x_i) \cdot F_{\mathrm{x_j}}(x_j) \tag{A.94}$$

$$p_{\mathrm{x_i},\mathrm{x_j}}(x_i, x_j) = p_{\mathrm{x_i}}(x_i) \cdot p_{\mathrm{x_j}}(x_j) \tag{A.95}$$

which extends to the concept of *mututal independence*, if all $\mathrm{x_i}$, $\mathrm{x_j}$ are independent for $i \neq j$

$$p_{\mathbf{x}}(\boldsymbol{x}) = \prod_i p_{\mathrm{x_i}}(x_i) \tag{A.96}$$

and "joint independence", if all $\mathrm{x_i}$, $\mathrm{y_j}$ are independent

$$p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y}) = p_{\mathbf{x}}(\boldsymbol{x}) \cdot p_{\mathbf{y}}(\boldsymbol{y}) \tag{A.97}$$

Two random variables $\mathrm{x_i}$, $\mathrm{x_j}$ are said to be "uncorrelated" if

$$\mathrm{Cov}[\mathrm{x_i}, \mathrm{x_j}] = 0 \tag{A.98}$$

otherwise they are correlated. For uncorrelated random variables, their correlation coefficient is also 0. Furthermore, it holds

$$\mathrm{E}[\mathbf{x}\mathbf{y}^\mathsf{T}] = \mathrm{E}[\mathbf{x}]\,\mathrm{E}[\mathbf{y}]^\mathsf{T} \tag{A.99}$$

$$\mathrm{Cov}[\mathbf{x}+\mathbf{y}] = \mathrm{Cov}[\mathbf{x}] + \mathrm{Cov}[\mathbf{y}] \tag{A.100}$$

If two random variables are independent, they are also uncorrelated. However, the inverse is in general not true [Jaz1970]. One important special case, where correlation does however imply dependence, is for Gaussian random variables, see section A.4.6 Gaussian Random Variables.

A.4.5 Conditional Probabilities

The definition of a conditional probability is [Jaz1970]

$$Pr\{\omega_1|\omega_2\} = \frac{Pr\{\omega_1,\omega_2\}}{Pr\{\omega_2\}} \tag{A.101}$$

Intuitively, this can be read as the (modified) probability of ω_1, given that an occurrence of ω_2 has been observed. For this to be meaningful, there has to be a possibility that ω_2 occurs, which translates to $Pr\{\omega_2\} \neq 0$, thus the division above does not pose a problem for meaningful tasks.

Employing a limiting argument, JAZWINSKI then arrives at a definition for the conditional, continuous probability density using the marginal distribution $p_{\mathrm{y}}(y)$ [Jaz1970, p. 38]

$$p_{\mathrm{x}|\mathrm{y}}(x|y) = \frac{p_{\mathrm{x},\mathrm{y}}(x,y)}{p_{\mathrm{y}}(y)} \tag{A.102}$$

$$p_{\mathrm{y}}(y) = \int_{-\infty}^{\infty} p_{\mathrm{x},\mathrm{y}}(x,y)\,dx \tag{A.103}$$

and a conditional cumulative function

$$F_{\mathrm{x}|\mathrm{y}}(x|y) = \frac{\int_{-\infty}^{x} p_{\mathrm{x},\mathrm{y}}(\xi,y)\,d\xi}{p_{\mathrm{y}}(y)} \tag{A.104}$$

For independent x and y it holds

$$p_{\mathrm{x}|\mathrm{y}}(x|y) = \frac{p_{\mathrm{x},\mathrm{y}}(x,y)}{p_{\mathrm{y}}(y)} = \frac{p_{\mathrm{x}}(x)\,p_{\mathrm{y}}(y)}{p_{\mathrm{y}}(y)} = p_{\mathrm{x}}(x) \tag{A.105}$$

The extension to the vector case is straight forward

$$\begin{aligned} p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y}) &= \frac{p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})}{p_{\mathbf{y}}(\boldsymbol{y})} \\ p_{\mathrm{x}_1\ldots\mathrm{x}_{\mathrm{n_x}}|\mathrm{y}_1\ldots\mathrm{y}_{\mathrm{n_y}}}\Big(x_1\ldots x_{n_x}\Big|y_1\ldots y_{n_y}\Big) &= \frac{p_{\mathrm{x}_1\ldots\mathrm{x}_{\mathrm{n_x}},\mathrm{y}_1\ldots\mathrm{y}_{\mathrm{n_y}}}\Big(x_1\ldots x_{n_x},y_1\ldots y_{n_y}\Big)}{p_{\mathrm{y}_1\ldots\mathrm{y}_{\mathrm{n_y}}}\Big(y_1\ldots y_{n_y}\Big)} \end{aligned} \tag{A.106}$$

$$\begin{aligned} p_{\mathbf{y}}(\boldsymbol{y}) &= \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})\,d\boldsymbol{x} \\ p_{\mathrm{y}_1\ldots\mathrm{y}_{\mathrm{n_y}}}\Big(y_1\ldots y_{n_y}\Big) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\xi_1\ldots\xi_{n_x},\boldsymbol{y})\,d\xi_1 d\xi_{n_x} \end{aligned} \tag{A.107}$$

Combining above definitions for $p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})$ and $p_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x})$ and substituting $p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y})$ eventually yields Baye's Theorem

$$p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{p_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x})\cdot p_{\mathbf{x}}(\boldsymbol{x})}{p_{\mathbf{y}}(\boldsymbol{y})} \tag{A.108}$$

for continuous random vectors.

Since it holds [Jaz1970]

$$p_{\mathrm{x}|\mathrm{y}}(x|y) \geq 0 \tag{A.109}$$

$$\int_{-\infty}^{\infty} p_{\mathrm{x}|\mathrm{y}}(x|y)\,dx = 1 \tag{A.110}$$

conditional probability densities are themselves valid density functions. However, since they depend on the realizations of y, they are at the same time random variables.

The mean of a continuous conditional probability $p_{\mathrm{x}|\mathrm{y}}(x|y)$ is the marginal distribution $p_{\mathrm{x}}(x)$

$$\mathrm{E}\Big[p_{\mathrm{x}|\mathrm{y}}(x|y)\Big] = \int_{-\infty}^{\infty} p_{\mathrm{x}|\mathrm{y}}(x|y)\cdot p_{\mathrm{y}}(y)\,dy = \int_{-\infty}^{\infty} p_{\mathrm{x},\mathrm{y}}(x,y)\,dy = p_{\mathrm{x}}(x) \tag{A.111}$$

A.4.5.1 Moments

Statistical moments can also be defined based on conditional probabilities [Jaz1970]

$$\mathrm{E}[\mathbf{x}|\mathbf{y}] = \int_{-\infty}^{\infty} x p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})\,dx \tag{A.112}$$

Since $p_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x}|\boldsymbol{y})$ depends on the realizations of $\mathbf{y}$, the conditional mean $\mathrm{E}[\mathbf{x}|\mathbf{y}]$ is a function of $\mathbf{y}$, and thus a random vector.

Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be jointly distributed random variables, and $\mathbf{A}$, $\mathbf{B}$, and $\boldsymbol{c}$ be fixed constants of appropriate size. Then the following equalities hold for conditional means [Jaz1970, Theorem 2.9]

$$\mathrm{E}[\mathbf{x}|\mathbf{y}] = \mathrm{E}[\mathbf{x}] \qquad \text{if } p_{\mathbf{x},\mathbf{y}}(\boldsymbol{x},\boldsymbol{y}) = p_{\mathbf{x}}(\boldsymbol{x})\,p_{\mathbf{y}}(\boldsymbol{y}) \tag{A.113}$$

$$\mathrm{E}[\mathrm{E}[\mathbf{x}|\mathbf{y}]] = \mathrm{E}[\mathbf{x}] \tag{A.114}$$

$$\mathrm{E}[\boldsymbol{c}|\mathbf{y}] = \boldsymbol{c} \tag{A.115}$$

$$\mathrm{E}[\boldsymbol{g}(\mathbf{y})|\mathbf{y}] = \boldsymbol{g}(\mathbf{y}) \tag{A.116}$$

$$\mathrm{E}[\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{z}|\mathbf{y}] = \mathbf{A}\mathrm{E}[\mathbf{x}|\mathbf{y}] + \mathbf{B}\mathrm{E}[\mathbf{z}|\mathbf{y}] \tag{A.117}$$

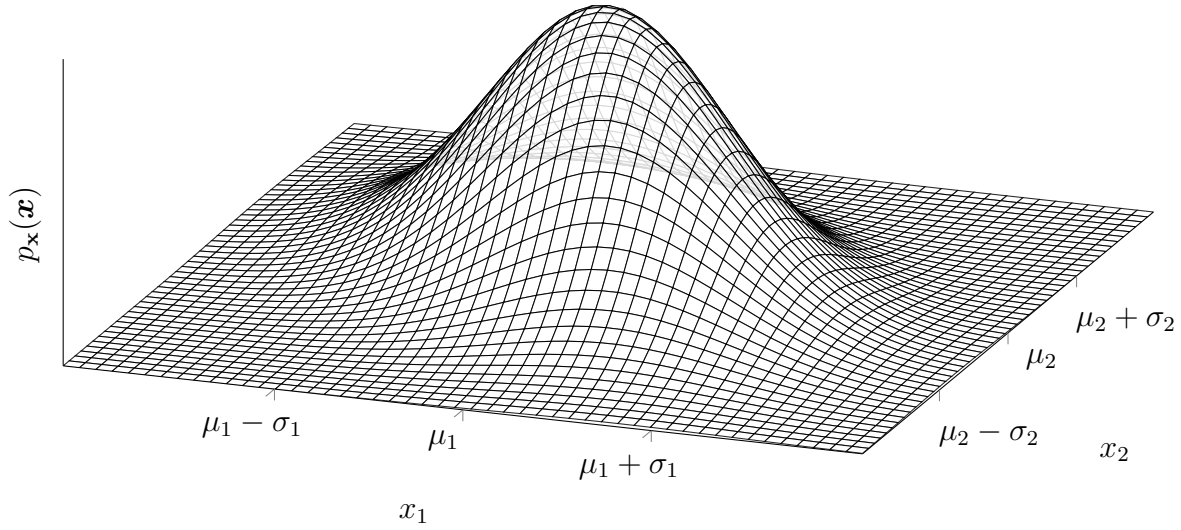


Figure A.1: *Example of a two-dimensional Normal probability density function*

Analogously, conditional covariances can be defined, which are again a random variable dependent on y

$$\mathrm{Cov}[\mathbf{x}|\mathbf{y}] = \mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}|\mathbf{y}])(\mathbf{x} - \mathrm{E}[\mathbf{x}|\mathbf{y}])^{\intercal}|\mathbf{y}] \tag{A.118}$$

Using the computational rules of the conditional mean and the definition of the conditional covariance, the following "rule of total variances" can be derived [Jaz1970, p. 41]

$$\mathrm{Cov}[\mathbf{x}] = \mathrm{E}[\mathrm{Cov}[\mathbf{x}|\mathbf{y}]] + \mathrm{Cov}[\mathrm{E}[\mathbf{x}|\mathbf{y}]] \tag{A.119}$$

Thus, the covariance of the random variable x can be split in two parts using the information that might be contained in a realization of y: One part pertains to the mean of the conditional variance, and the second part is the covariance of the conditional mean.

A.4.6 Gaussian Random Variables

The probability density function of a "normally" or "Gaussian" distributed random vector x is

$$p_{\mathbf{x}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{n_x}|\boldsymbol{\Sigma}|}} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\intercal}\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\Big) \tag{A.120}$$

where $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$. Figure A.1 shows an example for a two dimensional Gaussian probability density function. The Gaussian distribution is characterized by two parameters only, namely its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

$$\mathrm{E}[\mathbf{x}] = \mu \tag{A.121}$$

$$\mathrm{Cov}[\mathbf{x}] = \boldsymbol{\Sigma} \tag{A.122}$$

Sometimes a normally distributed random vector will be indicated as

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{A.123}$$

Separating the parts of $\mathbf{x}$ in two, yields the following partitioning of the mean and covariance

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \tag{A.124}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} = \begin{bmatrix} \mathrm{E}[\mathbf{x_1}] \\ \mathrm{E}[\mathbf{x_2}] \end{bmatrix} \tag{A.125}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} = \begin{bmatrix} \mathrm{Cov}[\mathbf{x_1}] & \mathrm{Cov}[\mathbf{x_1}, \mathbf{x_2}] \\ \mathrm{Cov}[\mathbf{x_1}, \mathbf{x_2}]^\intercal & \mathrm{Cov}[\mathbf{x_2}] \end{bmatrix} \tag{A.126}$$

Based on this partitioning, some very appealing properties of the Normal distribution can be shown:

Independence of Gaussian Random Variables If the two parts are uncorrelated, i.e. $\mathrm{Cov}[\mathbf{x_1}, \mathbf{x_2}] = \mathbf{0}$, after some algebraic reformulations, the joint distribution function can be expressed as

$$p_{\mathbf{x}}(\boldsymbol{x}) = p_{\mathbf{x}_1}(\boldsymbol{x}_1) \cdot p_{\mathbf{x}_2}(\boldsymbol{x}_2) \tag{A.127}$$

That is, for Gaussian random variables, no correlation implies independence [Jaz1970, Theorem 2.10].

Linear Operations on Gaussian Random Variables Consider the linear transformation $\mathbf{y} = \mathbf{C}\mathbf{x} + \boldsymbol{b}$. Then applying the rule about transforming random variables (A.93) yields

$$p_{\mathbf{y}}(\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^{n_x}|\boldsymbol{\Sigma}|}} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{b} - \mathbf{C}\boldsymbol{\mu})^\intercal (\mathbf{C}\boldsymbol{\Sigma}\mathbf{C}^\intercal)^{-1}(\boldsymbol{x} - \boldsymbol{b} - \mathbf{C}\boldsymbol{\mu})\Big) \cdot \frac{1}{|\mathbf{C}|} \tag{A.128}$$

$$\Rightarrow \mathbf{y} \sim \mathcal{N}(\boldsymbol{b} + \mathbf{C}\boldsymbol{\mu}, \mathbf{C}\boldsymbol{\Sigma}\mathbf{C}^\intercal) \tag{A.129}$$

Thus, linear transformations of Gaussian random variables yield again Gaussian random variables, where the mean and covariance can easily be computed.

Conditional Gaussian Densities Using above partitioning, and the results w.r.t. independence and linear operations on Gaussian variables, it can be shown that the conditional pdf $p_{\mathbf{x}_1|\mathbf{x}_2}(\boldsymbol{x}_1|\boldsymbol{x}_2)$ is again Gaussian with

$$\mathrm{E}[\mathbf{x_1}|\mathbf{x_2}] = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \tag{A.130}$$

$$\mathrm{Cov}[\mathbf{x_1}|\mathbf{x_2}] = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \tag{A.131}$$

$$\Rightarrow \mathbf{x_1}|\mathbf{x_2} \sim \mathcal{N}\Big(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\Big) \tag{A.132}$$

The derivation is lengthy and can e.g. be found in [Jaz1970, Ch. 2.6].

These characteristics make the Gaussian distribution especially appealing, since computations simplify greatly. Furthermore, its application is often well justified, due to the "Central Limit Theorem", see A.4.9.2.

A.4.7 Optimal Linear Combination of Unbiased and Uncorrelated Estimates

Assume that two uncorrelated, unbiased estimates $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ (together with their respective covariances) of a parameter $\boldsymbol{\theta}$ are available

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}_1\right] = \boldsymbol{\theta} \tag{A.133}$$

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}_2\right] = \boldsymbol{\theta} \tag{A.134}$$

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2\right] = \mathbf{0} \tag{A.135}$$

CRASSIDIS and JUNKINS use similar arguments as the ones to follow, in order to arrive at an optimal combination of a forward and a backward filter for fixed interval smoothing [CJ2012, Ch. 6.1]. Their results can be generalized for any estimate, not necessarily originating in a state estimation problem.

Consider a linear combination of $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ as an estimator for $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_c = \mathbf{M}_1\hat{\boldsymbol{\theta}}_1 + \mathbf{M}_2\hat{\boldsymbol{\theta}}_2 \tag{A.136}$$

In order for this estimate to be unbiased, the following has to hold

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}_c\right] = \mathrm{E}\left[\mathbf{M}_1\hat{\boldsymbol{\theta}}_1 + \mathbf{M}_2\hat{\boldsymbol{\theta}}_2\right] = \mathbf{M}_1\mathrm{E}\left[\hat{\boldsymbol{\theta}}_1\right] + \mathbf{M}_2\mathrm{E}\left[\hat{\boldsymbol{\theta}}_2\right] = (\mathbf{M}_1 + \mathbf{M}_2)\,\boldsymbol{\theta} \stackrel{!}{=} \boldsymbol{\theta} \tag{A.137}$$

$$\Rightarrow \mathbf{I}_{n_{\boldsymbol{\theta}}} = \mathbf{M}_1 + \mathbf{M}_2 \tag{A.138}$$

Now an optimal estimate can be obtained by minimizing the trace of the covariance matrix $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right]$, i.e. by minimizing the sum of the parameter variances. Since the estimates are assumed to be uncorrelated, this becomes

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right] = \mathbf{M}_1\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]\mathbf{M}_1^{\intercal} + (\mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{M}_1)\,\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right](\mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{M}_1)^{\intercal} \tag{A.139}$$

$$\frac{\partial \mathrm{tr}\left[\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right]\right]}{\partial \mathbf{M}_1} = 2\mathbf{M}_1\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] - 2(\mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{M}_1)\,\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right] \stackrel{!}{=} \mathbf{0} \tag{A.140}$$

$$\Rightarrow \mathbf{M}_1 = \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]\right)^{-1} \tag{A.141}$$

$$\mathbf{M}_2 = \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]\right)^{-1} \tag{A.142}$$

Using the following identity

$$\begin{aligned}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]\right)^{-1} &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1}\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \\ &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1}\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1}\end{aligned} \tag{A.143}$$

the weighting matrices become

$$\mathbf{M}_1 = \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} \tag{A.144}$$

$$\mathbf{M}_2 = \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \tag{A.145}$$

The covariance of the optimal combination is

$$\begin{aligned}
&\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right] = \mathbf{M}_1 \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] \mathbf{M}_1^\intercal + \mathbf{M}_2 \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right] \mathbf{M}_2^\intercal \\
&= \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right] \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \\
&+ \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right] \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \\
&= \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right) \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \\
&= \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1}
\end{aligned} \tag{A.146}$$

The optimal estimate is then

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_c &= \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1}\right)^{-1} \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} \hat{\boldsymbol{\theta}}_1 + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \hat{\boldsymbol{\theta}}_2\right) \\
&= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c\right] \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_1\right]^{-1} \hat{\boldsymbol{\theta}}_1 + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_2\right]^{-1} \hat{\boldsymbol{\theta}}_2\right)
\end{aligned} \tag{A.147}$$

I.e. the optimal, linear combination of two uncorrelated, unbiased estimates of the same quantity is a weighted sum, where the weights are proportional to the inverse covariance matrices of the two estimates. MORELLI and KLEIN show similar results for including prior knowledge in linear least-squares estimation [MK2016, Ch. 5.5], i.e. for combining the least-squares solution with an uncorrelated, unbiased prior estimate.

This can be extended to m uncorrelated, unbiased estimates

$$\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^m\right] = \left(\sum_{i=1}^{m} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1}\right)^{-1} \tag{A.148}$$

$$\hat{\boldsymbol{\theta}}_c^m = \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^m\right] \cdot \sum_{i=1}^{m} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1} \hat{\boldsymbol{\theta}}_i \tag{A.149}$$

The proof can be based on the optimal combination of the first m estimates $\hat{\boldsymbol{\theta}}_c^m$, with the $m+1$ estimate $\hat{\boldsymbol{\theta}}_{m+1}$ and using the induction step for the covariance

$$\begin{aligned}
\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m+1}\right] &= \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^m\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{m+1}\right]^{-1}\right)^{-1} = \left(\sum_{i=1}^{m} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{m+1}\right]^{-1}\right)^{-1} \\
&= \left(\sum_{i=1}^{m+1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1}\right)^{-1}
\end{aligned} \tag{A.150}$$

and the parameter estimate

$$\begin{aligned}\hat{\boldsymbol{\theta}}_c &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m+1}\right] \cdot \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m}\right]^{-1} \hat{\boldsymbol{\theta}}_c^{m} + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{m+1}\right]^{-1} \hat{\boldsymbol{\theta}}_{m+1}\right) \\ &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m+1}\right] \cdot \left(\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m}\right]^{-1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m}\right] \cdot \sum_{i=1}^{m} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1} \hat{\boldsymbol{\theta}}_i + \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{m+1}\right]^{-1} \hat{\boldsymbol{\theta}}_{m+1}\right) \\ &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_c^{m+1}\right] \cdot \sum_{i=1}^{m+1} \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_i\right]^{-1} \hat{\boldsymbol{\theta}}_i \end{aligned} \tag{A.151}$$

A.4.8 Stochastic Convergence

Convergence for random variables is not as straightforward as in the deterministic case, since even close to the limit, there may remain a certain probability, that the difference between a random variable and its limiting value is suddenly large. This is the reason, why different types of convergences (some stronger, some weaker) are available for random variables: this "chance of departing from the limiting value" may be incorporated differently into the analysis. All types of convergence describe the behavior of the sequence of random variables $\mathrm{x}_1, \mathrm{x}_2, \ldots$

A.4.8.1 Almost Sure Convergence

Almost sure convergence (alternatively "convergence with probability 1") of x_N towards x is defined as [CS2011, Definition 1.26]

$$Pr\left\{\lim_{N\to\infty} \mathrm{x}_N = \mathrm{x}\right\} = 1 \tag{A.152}$$

i.e. those events, for which the sequence x_N does not have the limit x have to have probability 0 [Sor1980, App C.5]. Intuitively one could say that "There is a point in the series x_N after which the two distributions are practically equal" [HGH+2017, Ch. 4, slide on "Stochastic Convergence"]. Almost sure convergence is abbreviated as

$$\mathrm{x}_N \xrightarrow[N\to\infty]{a.s.} \mathrm{x} \tag{A.153}$$

A.4.8.2 Convergence in Probability

For convergence in probability it has to hold that for every $\epsilon > 0$ [CS2011, Definition 1.26]

$$Pr\{|\mathrm{x}_N - \mathrm{x}| > \epsilon\} \xrightarrow[N\to\infty]{} 0 \tag{A.154}$$

An intuitive interpretation is: "As the sequence progresses, the probability of an unusual outcome becomes smaller and smaller" [HGH+2017, Ch. 4, slide on "Stochastic Convergence"]. It is abbreviated as

$$\mathrm{x}_N \xrightarrow[N\to\infty]{Pr} \mathrm{x} \tag{A.155}$$

A.4.8.3 Convergence in Distribution

Convergence in distribution is based on the distribution functions of the series and its limit [Sor1980, App C.5]: If at every point of continuity of $F_{\mathrm{x}}(x)$, it holds that

$$\lim_{N\to\infty} F_{\mathrm{x_N}}(x_N) = F_{\mathrm{x}}(x) \tag{A.156}$$

Convergence in distribution is given. This can be interpreted as: "if an experiment with outcome x_N is only executed often enough, the results will eventually be distributed as x" [HGH+2017, Ch. 4, slide on "Stochastic Convergence"]. Convergence in distribution is abbreviated as

$$\mathrm{x}_N \xrightarrow[N\to\infty]{D} \mathrm{x} \tag{A.157}$$

A.4.8.4 Connections between the Convergence types

The three foregoing types of convergence were presented in ascending order when it comes to their strength: almost sure convergence can be considered to be the strongest type, since no outliers are tolerated, whereas for convergence in distribution, no statement about the realizations can be made. Also, almost sure convergence implies convergence in probability, which in turn implies convergence in distribution.

A.4.9 Statistical Theorems

Two statistical theorems of significant importance are listed next. Using them, convergence statements about the sample average and sample mean are possible.

The sample average of a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ of the random variable $\mathbf{x}$ is

$$\bar{\boldsymbol{x}}_N = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}_k \tag{A.158}$$

The sample variance is, if $\boldsymbol{\mu} = \mathrm{E}[\mathbf{x}]$

$$s^2_{\boldsymbol{x},N} = \frac{1}{N}\sum_{k=1}^{N} (\boldsymbol{x}_k - \boldsymbol{\mu})(\boldsymbol{x}_k - \boldsymbol{\mu})^{\mathsf{T}} \tag{A.159}$$

if the mean is unknown, an unbiased estimate of the sample variance is

$$s^2_{\boldsymbol{x},N} = \frac{1}{N-1}\sum_{k=1}^{N} (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_N)(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_N)^{\mathsf{T}} \tag{A.160}$$

A.4.9.1 The Law of Large Numbers

Two versions of the law of large numbers are presented, with different requirements

Weak Law of Large Numbers Let $\mathrm{x}_1, \mathrm{x}_2, \ldots$ be pair-wise uncorrelated with common mean $\mathrm{E}[\mathrm{x}_k] = \mathrm{E}[\mathrm{x}]$ ("all RV's have the same mean") and $\mathrm{Var}[\mathrm{x}_k] \leq M \leq \infty$ for all $i \geq 1$ and some $M \in \mathbb{R}$ ("all RV's have finite variance"), then the sample average of the series converges in probability towards the expected value [CS2011, Satz 1.29]

$$\bar{\mathrm{x}}_N = \frac{1}{N} \sum_{k=1}^{N} \mathrm{x}_k \xrightarrow[N\to\infty]{Pr} \mathrm{E}[\mathrm{x}] \tag{A.161}$$

Strong Law of Large Numbers Stronger requirements, namely that the x_k be independent and identically distributed (i.i.d.) and have finite mean $\mathrm{E}[\mathrm{x}_i] = \mathrm{E}[\mathrm{x}] < \infty$, lead to almost sure convergence of the sample mean towards the expected value [CS2011, Satz 1.30]

$$\bar{\mathrm{x}}_N = \frac{1}{N} \sum_{k=1}^{N} \mathrm{x}_k \xrightarrow[N\to\infty]{a.s.} \mathrm{E}[\mathrm{x}] \tag{A.162}$$

A.4.9.2 The Central Limit Theorem

Let $\mathbf{x} = [\mathrm{x}_1, \ldots, \mathrm{x}_{n_x}]^\intercal$ be a n_x-dimensional, random vector with mean $\mathrm{E}[\mathbf{x}]$ and covariance matrix $\mathrm{Cov}[\mathbf{x}]$, where all elements of the covariance matrix are finite. Further, let $\mathbf{x}_1 \ldots \mathbf{x}_N$ be a series of i.i.d. samples of $\mathbf{x}$. Then the scaled distance of the sample average from the mean converges in distribution towards a multivariate normal distribution with zero mean and covariance $\mathrm{Cov}[\mathbf{x}]$ [CS2011, Satz 1.33]

$$\sqrt{N}(\bar{\mathbf{x}}_N - \mathrm{E}[\mathbf{x}]) \xrightarrow[N\to\infty]{D} \mathcal{N}(\mathbf{0}, \mathrm{Cov}[\mathbf{x}]) \tag{A.163}$$

A.5 Stochastic Processes

A family of random variables, which is indexed by a parameter set T, i.e. $\{\mathbf{x}_t, t \in T\}$ is called stochastic process. In the applications here, T will always refer to time. Both, the random variable and the index set can either be discrete or continuous, resulting in four possible scenarios. Due to the sampling process involved when measuring physical processes, the main focus will be on discrete time, continuous state space processes. Additionally, they can be analyzed using the tools developed so far. Especially for continuous time stochastic processes, the necessary mathematical tools are quite advanced and will be omitted here. A complete, and rigorous treatment of the topic, can e.g. be found in [Jaz1970]. Here only the main results and definitions are listed.

A complete stochastic characterization of the process is possible, if the joint density function for the time instants t_k $k = 0, \ldots, N-1$ was known [Jaz1970, MIM1985]

$$\begin{aligned} & p_{\mathbf{X}}(\mathbf{X}) \\ \mathbf{X} = & \begin{bmatrix} \boldsymbol{x}_0 & \cdots & \boldsymbol{x}_{N-1} \end{bmatrix} \\ = & \begin{bmatrix} \boldsymbol{x}_{t_0} & \cdots & \boldsymbol{x}_{t_{N-1}} \end{bmatrix} \end{aligned} \tag{A.164}$$

A.5.1 Moments

In analogy to the previous sections, moments can be defined for stochastic processes. The first moment is known as the mean value vector, whereas the second, central moment is termed the auto-correlation matrix [Jaz1970, MIM1985]

$$\boldsymbol{m}_k^{\mathbf{x}} = \mathrm{E}[\mathbf{x}_k] \tag{A.165}$$

$$\mathbf{R}_{k,i}^{\mathbf{x}} = \mathrm{E}[(\mathbf{x}_k - \boldsymbol{m}_k^{\mathbf{x}})(\mathbf{x}_i - \boldsymbol{m}_i^{\mathbf{x}})^{\mathsf{T}}] \tag{A.166}$$

The latter is an indicator for the correlation between states at different points in time. An important special case is the covariance function, which arises for $k = i$

$$\mathbf{P}_k^{\mathbf{x}} = \mathbf{R}_{k,k}^{\mathbf{x}} \tag{A.167}$$

Often, it is easier to come by the mean value vector and covariance matrix, compared to formulating the full probability law of the process [Jaz1970]. This is also the reason, why often stochastic processes are only discussed in terms of their first two moments, since a full treatment becomes too complicated [MIM1985].

A.5.2 Stationary Processes

A stochastic process is called *strictly stationary*, if it holds that [Jaz1970, MIM1985]

$$p_{\mathbf{x}_{\mathrm{t}_0}\ldots\mathbf{x}_{\mathrm{t}_{\mathrm{N}-1}}}\left(\boldsymbol{x}_{t_0}\ldots\boldsymbol{x}_{t_{N-1}}\right) = p_{\mathbf{x}_{\mathrm{t}_0+\tau}\ldots\mathbf{x}_{\mathrm{t}_{\mathrm{N}-1}+\tau}}\left(\boldsymbol{x}_{t_0+\tau}\ldots\boldsymbol{x}_{t_{N-1}+\tau}\right) \tag{A.168}$$

i.e. if the probability law of the process is not affected by a shift in time τ. This implies, that the probability law of a strictly stationary process must not be a function of time.

A weaker form of stationarity is *weak* stationarity, or *wide sense stationarity*, where independence of a time-shift is only required for the first two moments (which have to exist). Thus for weak stationary processes it holds [Jaz1970, MIM1985]

$$\boldsymbol{m}_{k+\tau}^{\mathbf{x}} = \boldsymbol{m}_k^{\mathbf{x}} = const \tag{A.169}$$

$$\mathbf{R}_{k+\tau,k}^{\mathbf{x}} = \mathbf{R}_{\tau}^{\mathbf{x}} \tag{A.170}$$

i.e. its mean value function needs to be constant, whereas its autocorrelation function only depends on the time shift.

A.5.3 Gaussian Processes

If additionally, the probability law (A.164) is Gaussian, the process is called normal, and thus completely characterized by its mean value and auto-correlation function. Furthermore, for Gaussian processes, wide sense stationarity implies strict stationarity

[Jaz1970, MIM1985]. Using the property of Gaussian distributions, that linear operations on them result again in Gaussian distributions (A.128), gives rise to a very important class of stochastic processes, namely those, who can be described by a linear difference equation of the form

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{F}_k \boldsymbol{w}_k \tag{A.171}$$

Then, if the $\boldsymbol{w}_k$ are independent Gaussian variables, the complete stochastic process is Gaussian.

A.5.4 Markov Process

A process is said to be *Markov* if it shows the following "Markov Property" [Jaz1970, MIM1985]

$$p_{\mathbf{x}_{\mathrm{k}+1}|\mathbf{x}_{\mathrm{k}}\ldots\mathbf{x}_0}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k \ldots \boldsymbol{x}_0) = p_{\mathbf{x}_{\mathrm{k}+1}|\mathbf{x}_{\mathrm{k}}}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \tag{A.172}$$

Intuitively, this means that the probability of the next state $\boldsymbol{x}_{k+1}$ is only dependent on the current state $\boldsymbol{x}_k$, not on past states, i.e. it is only important, where the process is at time k, not how it arrived there. Thus, if a process is Markov, recursively applying the definition of conditional probabilities (A.106) yields [MIM1985]

$$\begin{aligned} p_{\mathbf{X}_{\mathrm{N}-1}}(\mathbf{X}_{N-1}) &= p_{\mathbf{x}_{\mathrm{N}-1}|\mathbf{X}_{\mathrm{N}-2}}(\boldsymbol{x}_{N-1}|\mathbf{X}_{N-2})\, p_{\mathbf{X}_{\mathrm{N}-2}}(\mathbf{X}_{N-2}) = \ldots \\ &= p_{\mathbf{x}_0}(\boldsymbol{x}_0) \cdot \prod_{k=0}^{N-2} p_{\mathbf{x}_{\mathrm{k}+1}|\mathbf{x}_{\mathrm{k}}}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \end{aligned} \tag{A.173}$$

where the *transition probabilities* $p_{\mathbf{x}_{\mathrm{k}+1}|\mathbf{x}_{\mathrm{k}}}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$ may still depend on time, but not on past state values. This property can even hold for non-linear difference equations with additive, independent process noise terms $\boldsymbol{w}_k$

$$\mathbf{x}_{k+1} = {}^{d}\mathbf{f}_d[\mathbf{x}_k, \boldsymbol{u}_k] + \mathbf{F}_k \mathbf{w}_k \tag{A.174}$$

The extension to non-additive process noise can be found in [Jaz1970]. Thus future states only depend on the current state, current realization of the process noise, and possibly time. If the probability density of the process noise term $\mathbf{F}_k \boldsymbol{w}_k$ can be stated, the (time dependent) transition probability becomes [Jaz1970]

$$p_{\mathbf{x}_{\mathrm{k}+1}|\mathbf{x}_{\mathrm{k}}}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = p_{\mathbf{F}_{\mathrm{k}}\mathbf{w}_{\mathrm{k}}}\Big(\boldsymbol{x}_{k+1} - {}^{d}\mathbf{f}_d[\boldsymbol{x}_k, \boldsymbol{u}_k]\Big) \tag{A.175}$$

If the process in addition is stationary, all transition probabilities are equal, thus defining them, and the probability of the initial condition fully characterizes the process. Eventually, a *Gauss-Markov* process is a stochastic process, which, in addition to the Markov property, also has Gaussian transition probabilities.

A.5.5 White Processes

Another important special case arises, if the transition probabilities do not even depend on the current state anymore

$$p_{\mathbf{x}_{k+1}|\mathbf{x}_k}(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = p_{\mathbf{x}_{k+1}}(\boldsymbol{x}_{k+1}) \tag{A.176}$$

Thus, no statement about the evolution of the states, based on current observations are possible. The states at all points in time are stochastically independent of each other, and thus totally unpredictable. These sequences are commonly called *white*.

The first two moments of a white, stochastic process are [Jaz1970, MIM1985]

$$\boldsymbol{m}_k^{\mathbf{x}} = \mathrm{E}[\mathbf{x}_k] \tag{A.177}$$

$$\mathbf{R}_{k,i}^{\mathbf{x}} = \delta_{ki}\mathbf{P}_k^{\mathbf{x}} \tag{A.178}$$

i.e. the autocorrelation function only has non-zero values for $i = j$. If a white, Gaussian process is considered, the process is fully characterized by above two equations. In many technical applications, white, Gaussian stochastic processes are used as a model for noise processes in the system.

The adjective *white* stems from an analogy with white light, which contains light of all frequencies. For continuous time white noise, it can be shown, that it has a constant power spectral density function, i.e. it also contains all frequencies. However, this would necessitate infinite power, which is why white noise processes are not physically realizable. Nevertheless, they have proven to be a valid mathematical model in many applications [Jaz1970]. Its application can also be justified, by assuming that the correlation times of the noise process are short, compared to the characteristic times of the process of interest [BF1963].

Appendix B

Derivation of Descent Direction Condition

If it holds that (proof taken from [Ger2017, Ch. 3.5])

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}^{\mathsf{T}} \boldsymbol{d}_i < 0 \tag{B.1}$$

the following definition of the directional derivative of J in the direction $\boldsymbol{d}_i$ is also smaller than zero

$$\left.\frac{\partial J}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}^{\mathsf{T}} \boldsymbol{d}_i = \lim_{\alpha \to 0^+} \frac{J(\boldsymbol{\theta}_i + \alpha \boldsymbol{d}_i) - J(\boldsymbol{\theta}_i)}{\alpha} < 0 \tag{B.2}$$

Then there exists an $\bar{\alpha}$ such that

$$\begin{aligned} J(\boldsymbol{\theta}_i + \alpha \boldsymbol{d}_i) - J(\boldsymbol{\theta}_i) &< 0 \\ J(\boldsymbol{\theta}_i + \alpha \boldsymbol{d}_i) &< J(\boldsymbol{\theta}_i) \qquad \forall 0 < \alpha < \bar{\alpha} \end{aligned} \tag{B.3}$$

i.e. a step of at max $\bar{\alpha}$ reduces the cost function, thus $\boldsymbol{d}_i$ is a descent direction.

Furthermore, a geometric interpretation of this is that the gradient and search direction enclose an angle between $90°$ and $270°$, which can be derived from the following property of scalar products [Ger2017, Ch. 3.5.]

$$\cos \angle(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^{\mathsf{T}} \boldsymbol{b}}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2} \tag{B.4}$$

Appendix C

Statistical Proofs

The following relations are used in several of the proofs of this Appendix, for further details see [MK2016, Appendix B]. Let $p(\mathbf{Z}|\boldsymbol{\theta})$ be the probability density describing the probability of obtaining the measurements $\mathbf{Z}$ given the parameters $\boldsymbol{\theta}$. Assuming sufficient smoothness of $\ln p(\mathbf{Z}|\boldsymbol{\theta})$, its gradient with respect to $\boldsymbol{\theta}$ is called the *Score* $\boldsymbol{s}(\mathbf{Z}, \boldsymbol{\theta})$

$$\boldsymbol{s}(\mathbf{Z}, \boldsymbol{\theta}) \coloneqq \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{C.1}$$

executing the differentiation using the chain rule, it can be expressed as

$$\boldsymbol{s}(\mathbf{Z}, \boldsymbol{\theta}) = \frac{1}{p(\mathbf{Z}|\boldsymbol{\theta})} \cdot \frac{\partial p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{C.2}$$

Rearranging above equation yields

$$\frac{\partial p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = p(\mathbf{Z}|\boldsymbol{\theta}) \, \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = p(\mathbf{Z}|\boldsymbol{\theta}) \, \boldsymbol{s}(\mathbf{Z}, \boldsymbol{\theta}) \tag{C.3}$$

Furthermore, $p(\mathbf{Z}|\boldsymbol{\theta})$ has to obey the basic statistical properties illustrated in section A.4

$$\int p(\mathbf{Z}|\boldsymbol{\theta}) \, d\mathbf{Z} = 1 \tag{C.4}$$

Differentiating equation (C.4) with respect to $\boldsymbol{\theta}$, using equation (C.3) and assuming sufficient smoothness such that integration and differentiation can be exchanged yields

$$\int \frac{\partial p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} d\mathbf{Z} = \int \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot p(\mathbf{Z}|\boldsymbol{\theta}) \, d\mathbf{Z} = \mathbf{0} \tag{C.5}$$

Together with the definition of the expected value (A.78) this yields

$$\mathrm{E}[\boldsymbol{s}(\mathbf{Z}, \boldsymbol{\theta})] = \mathrm{E}\left[\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] = \mathbf{0} \tag{C.6}$$

which holds for any sample size N, especially also for $N = 1$

$$\mathrm{E}[\boldsymbol{s}(\boldsymbol{z}_k, \boldsymbol{\theta})] = \mathbf{0} \tag{C.7}$$

Since the score has zero-mean, its covariance is the Fisher information matrix as defined in equation (2.76) and (2.83)

$$\begin{aligned}\mathrm{Cov}[\boldsymbol{s}(\mathbf{Z},\boldsymbol{\theta})] &= \mathrm{E}[(\boldsymbol{s}(\mathbf{Z},\boldsymbol{\theta})-\mathbf{0})(\boldsymbol{s}(\mathbf{Z},\boldsymbol{\theta})-\mathbf{0})^{\intercal}] \\ &= \mathrm{E}\left[\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal}\right] = \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})\end{aligned} \tag{C.8}$$

It is noticeable that, since the expectation involves an integration over all possible $\mathbf{Z}$, $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})$ is no function of $\mathbf{Z}$, in contrast to $\boldsymbol{s}(\mathbf{Z},\boldsymbol{\theta})$.

If the Markov Criterion (2.86) holds, i.e. if the conditional expectation can be expressed as $p(\mathbf{Z}|\boldsymbol{\theta}) = \prod_{k=0}^{N-1} p(\boldsymbol{z}_k|\boldsymbol{\theta})$, the following expressions for one sample arise

$$\boldsymbol{s}(\mathbf{Z},\boldsymbol{\theta}) = \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{k=0}^{N-1} \frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{k=0}^{N-1} \boldsymbol{s}(\boldsymbol{z}_k,\boldsymbol{\theta}) = N\bar{\boldsymbol{s}}_N(\boldsymbol{\theta}) \tag{C.9}$$

with the score per sample $\boldsymbol{s}(\boldsymbol{z}_k,\boldsymbol{\theta})$ and the average score for N samples $\bar{\boldsymbol{s}}_N$. The corresponding Fisher information matrix is based on the alternate expression in Appendix C.1

$$\mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{z}_1|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = \mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{z}_2|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = \cdots = \boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta}) \tag{C.10}$$

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) = -\sum_{k=0}^{N-1} \mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = N\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta}) \tag{C.11}$$

with the Fisher information matrix per sample $\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})$ [HH2015, Remark 6.1].

C.1 Alternate Form of the Fisher Information Matrix

Differentiating equation (C.5) a second time with respect to $\boldsymbol{\theta}$ yields

$$\int \frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} + \int \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \cdot \left(\frac{\partial p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal} d\mathbf{Z} = \mathbf{0} \tag{C.12}$$

which can be modified using equation (C.3)

$$\begin{aligned}&\int \left(\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} + \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal}\right) \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} = \\ &\qquad = \mathrm{E}\left[\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} + \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal}\right] = \mathbf{0}\end{aligned} \tag{C.13}$$

With the linearity of the expectation operator, and the definition of the likelihood function (2.83) this eventually results in the alternative formulation of the Fisher information matrix

$$\boldsymbol{\mathcal{F}}(\boldsymbol{\theta}) = \mathrm{E}\left[\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal}\right] = -\mathrm{E}\left[\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \tag{C.14}$$

which is again also valid for only one sample

$$\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta}) = \mathrm{E}\left[\left(\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\left(\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\intercal}\right] = -\mathrm{E}\left[\frac{\partial^2 \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \tag{C.15}$$

C.2 Cramér-Rao Inequality

The following proof is a slightly altered formulation of the proof in [MIM1985], further details can be found in [MK2016] and [HGH+2017], and one of the original publications [Cra1946, Ch. 33]. The basic idea for the proof is to first derive an expression for the cross-covariance matrix $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\right]$. Then an advantageously chosen covariance matrix is considered. The property of positive definiteness of the latter, together with the derived expression for $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\right]$ can then be used to conclude the proof of the Cramér-Rao bounds. An alternative proof, based on the Schwarz inequality can be found in [Sor1980, Ch. 3], and [CJ2012, Ch. 2].

Assuming a biased estimator according to equation (2.73) the following is true, when using the linearity of the expectation operator

$$\mathrm{E}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z}) - \boldsymbol{\theta} - \boldsymbol{b}(\boldsymbol{\theta})\right] = \int \left(\hat{\boldsymbol{\theta}}(\mathbf{Z}) - \boldsymbol{\theta} - \boldsymbol{b}(\boldsymbol{\theta})\right) \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} = \mathbf{0} \tag{C.16}$$

Now, assuming sufficient smoothness in $p(\mathbf{Z}|\boldsymbol{\theta})$ such that differentiation and integration can be interchanged, the derivative with respect to $\boldsymbol{\theta}$ can be taken

$$\int \underbrace{\left(-\mathbf{I}_{n_\theta} - \frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)}_{\neq f(\mathbf{Z})} \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} + \int \left(\hat{\boldsymbol{\theta}}(\mathbf{Z}) - \boldsymbol{\theta} - \boldsymbol{b}(\boldsymbol{\theta})\right) \cdot \left(\frac{\partial p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\mathsf{T}} d\mathbf{Z} = \mathbf{0} \tag{C.17}$$

Using equation (C.3) for the second part of above equation it can be further modified to read

$$\left(\mathbf{I}_{n_\theta} + \frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right) \underbrace{\int p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z}}_{=1} = \int \left(\hat{\boldsymbol{\theta}}(\mathbf{Z}) - \boldsymbol{\theta} - \boldsymbol{b}(\boldsymbol{\theta})\right) \cdot \boldsymbol{s}(\boldsymbol{\theta})^{\mathsf{T}} \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} \tag{C.18}$$

Together with equations (2.73) and (C.6) it can be realized, that the right hand side of above equation equals the covariance between the parameter estimate and the score

$$\begin{aligned} \int \left(\hat{\boldsymbol{\theta}}(\mathbf{Z}) - \underbrace{(\boldsymbol{\theta} + \boldsymbol{b}(\boldsymbol{\theta}))}_{=\mathrm{E}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z})\right]}\right) \left(\boldsymbol{s}(\boldsymbol{\theta}) - \underbrace{\mathbf{0}}_{=\mathrm{E}[\boldsymbol{s}]}\right)^{\mathsf{T}} \cdot p(\mathbf{Z}|\boldsymbol{\theta})\, d\mathbf{Z} &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\right] \\ \left(\mathbf{I}_{n_\theta} + \frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right) &= \mathrm{Cov}\left[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\right] \end{aligned} \tag{C.19}$$

Now, consider the covariance matrix of two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ together with a non-random matrix $\mathbf{C}$ of appropriate size, which is positive definite per definition.

$$\mathrm{Cov}[\boldsymbol{X} - \mathbf{C}\boldsymbol{Y}] \geq \mathbf{0} \tag{C.20}$$

$$\begin{aligned} \mathrm{Cov}[\boldsymbol{X} - \mathbf{C}\boldsymbol{Y}] = \mathrm{Cov}[\boldsymbol{X}] &+ \mathbf{C} \cdot \mathrm{Cov}[\boldsymbol{Y}] \cdot \mathbf{C}^{\mathsf{T}} \\ &- \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}] \cdot \mathbf{C}^{\mathsf{T}} - \mathbf{C} \cdot \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}]^{\mathsf{T}} \geq \mathbf{0} \end{aligned} \tag{C.21}$$

$$\mathrm{Cov}[\boldsymbol{X}] \geq \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}] \cdot \mathbf{C}^{\mathsf{T}} + \mathbf{C} \cdot \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}]^{\mathsf{T}} - \mathbf{C} \cdot \mathrm{Cov}[\boldsymbol{Y}] \cdot \mathbf{C}^{\mathsf{T}} \tag{C.22}$$

Above relation holds for any random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ and deterministic $\mathbf{C}$. Now consider

$$\mathbf{C} \coloneqq \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}]\,\mathrm{Cov}[\boldsymbol{Y}]^{-1} \tag{C.23}$$

$$\Rightarrow \mathrm{Cov}[\boldsymbol{X}] \geq \mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}]\,\mathrm{Cov}[\boldsymbol{Y}]^{-1}\,\mathrm{Cov}[\boldsymbol{X}, \boldsymbol{Y}]^{\mathsf{T}} \tag{C.24}$$

Substituting $\boldsymbol{X} = \hat{\boldsymbol{\theta}}(\mathbf{Z})$ and $\boldsymbol{Y} = \boldsymbol{s}(\boldsymbol{\theta})$ in above equation yields

$$\mathrm{Cov}\Big[\hat{\boldsymbol{\theta}}(\mathbf{Z})\Big] \geq \mathrm{Cov}\Big[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\Big]\,\mathrm{Cov}[\boldsymbol{s}(\boldsymbol{\theta})]^{-1}\,\mathrm{Cov}\Big[\hat{\boldsymbol{\theta}}(\mathbf{Z}), \boldsymbol{s}(\boldsymbol{\theta})\Big]^{\mathsf{T}} \tag{C.25}$$

Then, together with equations (C.8) and (C.19) the Cramér-Rao bounds for a biased estimator result

$$\mathrm{Cov}\Big[\hat{\boldsymbol{\theta}}(\mathbf{Z})\Big] \geq \left(\mathbf{I}_{n_\theta} + \frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right) \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1} \left(\mathbf{I}_{n_\theta} + \frac{\partial \boldsymbol{b}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^{\mathsf{T}} \tag{C.26}$$

C.3 Properties of Maximum Likelihood Estimates

The derivations for the properties of the maximum likelihood Estimates, which are to follow in the next three subsections, are mainly based on [Jat2006, App. D] and [GP1977, Ch. 3]. Additional Details can be found in [Sor1980, Ch. 5]

C.3.1 Asymptotic Consistency

In order to prove asymptotic consistency, first an expression similar to the score, evaluated at the maximum likelihood estimates $\hat{\boldsymbol{\theta}}_{ML}$, is expressed in a Taylor series around the true parameter value $\boldsymbol{\theta}$

$$\frac{\partial \ln p\Big(\mathbf{Z}\Big|\hat{\boldsymbol{\theta}}_{ML}\Big)}{\partial \boldsymbol{\theta}} = \frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial^2 \ln p\Big(\mathbf{Z}\Big|\tilde{\boldsymbol{\theta}}\Big)}{\partial \boldsymbol{\theta}^2}\Big(\hat{\boldsymbol{\theta}}_{ML} - \boldsymbol{\theta}\Big) \tag{C.27}$$

$$\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_{ML} + \lambda\Big(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{ML}\Big) \qquad 0 \leq \lambda \leq 1 \tag{C.28}$$

which is no approximation since the explicit LAGRANGE form of the remainder is used. For a maximum likelihood estimate, the first order necessary condition (2.6) has to hold, i.e. above equation has to be equal to zero. Again, if the Markov Criterion (2.86) holds, a similar reasoning as in (C.9) and (C.11) can be applied, together with some rearranging one can arrive at

$$\sum_{k=0}^{N-1} \frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\sum_{k=0}^{N-1} \frac{\partial^2 \ln p\Big(\boldsymbol{z}_k\Big|\tilde{\boldsymbol{\theta}}\Big)}{\partial \boldsymbol{\theta}^2}\Big(\hat{\boldsymbol{\theta}}_{ML} - \boldsymbol{\theta}\Big) \tag{C.29}$$

Dividing both sides by N and applying the strong law of large numbers (A.162) yields for the left hand side

$$\frac{1}{N}\sum_{k=0}^{N-1} \frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \xrightarrow[a.s.]{N\to\infty} \mathrm{E}\left[\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right] = \mathrm{E}[\bar{\boldsymbol{s}}_N(\boldsymbol{\theta})] = \mathbf{0} \tag{C.30}$$

and for the right hand side

$$-\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial^2 \ln p\left(\boldsymbol{z}_k\middle|\tilde{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}^2}\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)\xrightarrow[a.s.]{N\to\infty}\mathrm{E}\left[-\frac{\partial^2 \ln p\left(\boldsymbol{z}_k\middle|\tilde{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}^2}\right]\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right) \tag{C.31}$$

According to JATEGAONKAR it can be shown, that the likelihood function is concave in $\boldsymbol{\theta}$ and thus it can be appropriately assumed that $\mathrm{E}\left[-\frac{\partial^2 \ln p\left(\boldsymbol{z}_k\middle|\tilde{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}^2}\right]$ is positive definite [Jat2006, App. D-I]. Since the left hand side of equation (C.29) converges to zero, whereas the matrix on its right converges to a positive definite value, the only way how the equation can hold is if

$$\hat{\boldsymbol{\theta}}_{ML}\xrightarrow[a.s.]{N\to\infty}\boldsymbol{\theta} \tag{C.32}$$

Thus the maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{ML}$ is consistent.

C.3.2 Asymptotic Normality

Again, starting from a TAYLOR series expansion

$$\frac{\partial \ln p\left(\mathbf{Z}\middle|\hat{\boldsymbol{\theta}}_{ML}\right)}{\partial\boldsymbol{\theta}}=\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}+\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)+\mathcal{O}\left(\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)^2\right) \tag{C.33}$$

the higher order terms are neglected, since it has already been shown, that the maximum likelihood estimates $\hat{\boldsymbol{\theta}}_{ML}$ are consistent. Again using the fact that the left hand side of above equation has to equal zero for maximum likelihood estimates one can arrive at

$$-\frac{1}{N}\frac{\partial^2 \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)=\frac{1}{N}\frac{\partial \ln p(\mathbf{Z}|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \tag{C.34}$$

$$-\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial^2 \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)=\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \tag{C.35}$$

The difference between this expression and the Taylor series of the last section lies in the evaluation of the Hessian. Here, it is evaluated at the true parameter vector $\boldsymbol{\theta}$, whereas before *some* $\tilde{\boldsymbol{\theta}}$ between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}_{ML}$ was used for the explicit LAGRANGE form of the residual.

Both sides of above equation are now considered separately. On the left, using the strong Law of Large Numbers (A.162), the expression involving the Hessian converges to

$$\begin{aligned}-\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial^2 \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)\xrightarrow[a.s.]{N\to\infty}&\mathrm{E}\left[-\frac{\partial^2 \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2}\right]\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)=\\&=\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right)\end{aligned} \tag{C.36}$$

For the right hand side of equation (C.35), equations (C.6), (C.8) and (C.11) provide values for the mean and covariance of $\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}$, namely zero and $\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})$. With this information, and the Central Limit Theorem (A.163) one can see that the sample average

$\frac{1}{N}\sum_{k=1}^{N}\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ converges in distribution to a random variable with zero mean and covariance $\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})$

$$\sqrt{N}\frac{1}{N}\sum_{k=0}^{N-1}\frac{\partial \ln p(\boldsymbol{z}_k|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \xrightarrow[D]{N\to\infty} \boldsymbol{r}_2 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})) \tag{C.37}$$

Replacing the left hand side expression by the result obtained in (C.36) and (C.35) and using (C.11) yields

$$\sqrt{N}\boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right) \xrightarrow[D]{N\to\infty} \boldsymbol{r}_2 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{F}}_k(\boldsymbol{\theta})) \tag{C.38}$$

$$\left(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}\right) \xrightarrow[D]{N\to\infty} \boldsymbol{r}_1 \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1}\right) \tag{C.39}$$

Since $\boldsymbol{\mathcal{F}}(\boldsymbol{\theta})$ tends to grow with larger sample sizes N, i.e. the parameter covariance tends to shrink, this again underlines the asymptotic consistency of the maximum likelihood estimator [Sor1980].

C.3.3 Asymptotic Efficiency

In the foregoing sections, it has been shown that the maximum likelihood estimates $\hat{\boldsymbol{\theta}}_{ML}$ are asymptotically consistent and normally distributed with covariance matrix $\mathrm{Cov}\left[\hat{\boldsymbol{\theta}}_{ML}\right] = \boldsymbol{\mathcal{F}}(\boldsymbol{\theta})^{-1}$, which already proves that they attain the Cramér-Rao bound and are thus asymptotically efficient.

C.4 Constrained Fisher information matrix

Much of the following section will be based on the theory of generalized inverses, see Appendix A.2.2. The overall goal of this section is to proof the following equality for the Fisher information matrix under equality constraints

$$\mathbf{P}_{\tilde{c}} = \mathbf{I}_{n_{\boldsymbol{\theta}}} - \frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{\theta}}\bigg|_{\hat{\boldsymbol{\theta}}}^{\intercal}\left(\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{\theta}}\bigg|_{\hat{\boldsymbol{\theta}}}\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{\theta}}\bigg|_{\hat{\boldsymbol{\theta}}}^{\intercal}\right)^{\dagger}\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{\theta}}\bigg|_{\hat{\boldsymbol{\theta}}} \tag{C.40}$$

$$\boldsymbol{\mathcal{F}}_{constr.}^{-1} = \mathbf{P}_{\tilde{c}}(\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^{\dagger}\mathbf{P}_{\tilde{c}}^{\intercal} = \mathbf{Z}(\mathbf{Z}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{Z})^{-1}\mathbf{Z}^{\intercal} \tag{C.41}$$

where $\mathbf{Z}$ is a basis for the null-space of $\frac{\partial \tilde{\boldsymbol{c}}}{\partial \boldsymbol{\theta}}\big|_{\hat{\boldsymbol{\theta}}}$.

The proof will take four steps

1. first show that it holds

$$\boldsymbol{\mathcal{F}}_{constr.}^{-1} = \mathbf{P}_{\tilde{c}}(\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^{\dagger}\mathbf{P}_{\tilde{c}}^{\intercal} = (\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^{\dagger} \tag{C.42}$$

2. show that the projection matrix can be expressed in terms of an orthonormal null space as

$$\mathbf{P}_{\tilde{c}} = \tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^{\intercal} \tag{C.43}$$

3. then show the equality

$$\boldsymbol{\mathcal{F}}_{constr.}^{-1} = (\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^{\dagger} = \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^{\intercal} \tag{C.44}$$

   for the special case of an orthonormal null-space basis $\tilde{\mathbf{Z}}$ with $\tilde{\mathbf{Z}}^{\intercal}\tilde{\mathbf{Z}} = \mathbf{I}$

4. lastly extend the result to arbitrary null-space bases

$$\boldsymbol{\mathcal{F}}_{constr.}^{-1} = \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^{\intercal} = \mathbf{Z}(\mathbf{Z}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{Z})^{-1}\mathbf{Z}^{\intercal} \tag{C.45}$$

The basic idea behind the first step is to show that $\mathbf{P}_{\tilde{c}}(\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^{\dagger}\,\mathbf{P}_{\tilde{c}}^{\intercal}$ is a valid pseudo inverse for $\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}$. This can be achieved by verifying the four PENROSE equations, together with the characteristics of idempotence and symmetry of orthogonal projectors. Since the proof itself is merely cumbersome but straight forward, it is left out here; also, the actual proof is very similar to what will be done in step three.

The second step makes use of the SVD of the constraint Jacobian

$$\left.\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^{\intercal}\\ \tilde{\mathbf{Z}}^{\intercal}\end{bmatrix} \tag{C.46}$$

The orthogonal projector is then

$$\begin{aligned}
\mathbf{P}_{\tilde{c}} &= \mathbf{I}_{n_{\boldsymbol{\theta}}} - \left.\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}^{\intercal}\left(\left.\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\left.\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}^{\intercal}\right)^{\dagger}\left.\frac{\partial\tilde{\boldsymbol{c}}}{\partial\boldsymbol{\theta}}\right|_{\hat{\boldsymbol{\theta}}}\\
&= \mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{V}\boldsymbol{\Sigma}^{\intercal}\mathbf{U}^{\intercal}(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}\mathbf{V}\boldsymbol{\Sigma}^{\intercal}\mathbf{U}^{\intercal})^{\dagger}\,\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}\\
&= \mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{V}\boldsymbol{\Sigma}^{\intercal}\mathbf{U}^{\intercal}\cdot\mathbf{U}(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\intercal})^{\dagger}\,\mathbf{U}^{\intercal}\cdot\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}\\
&= \mathbf{I}_{n_{\boldsymbol{\theta}}} - \mathbf{V}\boldsymbol{\Sigma}^{\intercal}(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\intercal})^{\dagger}\,\boldsymbol{\Sigma}\mathbf{V}^{\intercal}\\
&= \begin{bmatrix}\mathbf{V}_1 & \tilde{\mathbf{Z}}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^{\intercal}\\ \tilde{\mathbf{Z}}^{\intercal}\end{bmatrix} - \begin{bmatrix}\mathbf{V}_1 & \tilde{\mathbf{Z}}\end{bmatrix}\begin{bmatrix}\mathbf{I}_p & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^{\intercal}\\ \tilde{\mathbf{Z}}^{\intercal}\end{bmatrix}\\
&= \tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^{\intercal}
\end{aligned} \tag{C.47}$$

where equation (A.31) has been used in the third and equation (A.32) in the fifth equality.

To proof the third step, the four PENROSE equations are verified in order to show that $\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^{\intercal}$ is a valid pseudo-inverse for $\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}$

Penrose I (A.24): "$\mathbf{AXA} = \mathbf{A}$"

$$\begin{aligned}
\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\cdot\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^{\intercal}\cdot\mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}} &= \mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\cdot\tilde{\mathbf{Z}}\overbrace{\left(\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\underbrace{\tilde{\mathbf{Z}}^{\intercal}\cdot\tilde{\mathbf{Z}}}_{=\mathbf{I}}\tilde{\mathbf{Z}}^{\intercal}\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}}^{=\mathbf{I}}\tilde{\mathbf{Z}}^{\intercal}\\
&= \mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^{\intercal} = \mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\mathbf{P}_{\tilde{c}}\\
&= \mathbf{P}_{\tilde{c}}^{\intercal}\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}
\end{aligned} \tag{C.48}$$

Penrose II (A.25): "$\mathbf{XAX} = \mathbf{X}$"

$$\begin{aligned}\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\mathbf{P}_{\tilde{c}}^\intercal\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\tilde{\mathbf{Z}}\overbrace{\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\underbrace{\tilde{\mathbf{Z}}^\intercal\tilde{\mathbf{Z}}}_{=\mathbf{I}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}}^{=\mathbf{I}}\tilde{\mathbf{Z}}^\intercal \\ &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal \end{aligned} \tag{C.49}$$

Penrose III (A.26): "$(\mathbf{AX})^\intercal = \mathbf{AX}$"

$$\begin{aligned}\left(\mathbf{P}_{\tilde{c}}^\intercal\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}\cdot\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\right)^\intercal &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\mathbf{P}_{\tilde{c}}^\intercal\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}} \\ &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^\intercal \\ &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)\tilde{\mathbf{Z}}^\intercal \\ &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)\tilde{\mathbf{Z}}^\intercal\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal \\ &= \mathbf{P}_{\tilde{c}}^\intercal\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}}^\intercal\cdot\tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal\end{aligned} \tag{C.50}$$

Penrose IV (A.27): "$(\mathbf{XA})^\intercal = \mathbf{XA}$" uses exactly the same means as those above for Penrose III, so the details are omitted. This proves the third step

$$(\mathbf{P}_{\tilde{c}}^\intercal\boldsymbol{\mathcal{F}}\mathbf{P}_{\tilde{c}})^\dagger = \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal = \boldsymbol{\mathcal{F}}_{constr.}^{-1} \tag{C.51}$$

The last step then consists of showing, that the null-space basis does not necessarily have to be orthonormal. Every real null-space basis $\mathbf{Z}$ with full column-rank can be made orthonormal by using the Cholesky decomposition of $\mathbf{Z}^\intercal\mathbf{Z}$ to transform its input space

$$\mathbf{S}\mathbf{S}^\intercal = \mathbf{Z}^\intercal\mathbf{Z} \tag{C.52}$$

$$\tilde{\mathbf{Z}} = \mathbf{Z}\mathbf{S}^{-\intercal} \tag{C.53}$$

where $\mathbf{S}$ is a lower triangular matrix with positive diagonal elements. The resulting matrix $\tilde{\mathbf{Z}}$ is orthonormal

$$\tilde{\mathbf{Z}}^\intercal\tilde{\mathbf{Z}} = \mathbf{S}^{-1}\mathbf{Z}^\intercal\mathbf{Z}\mathbf{S}^{-\intercal} = \mathbf{S}^{-1}\mathbf{S}\mathbf{S}^\intercal\mathbf{S}^{-\intercal} = \mathbf{I} \tag{C.54}$$

Then the final expression for the constrained inverse Fisher information matrix is

$$\begin{aligned}\boldsymbol{\mathcal{F}}_{constr.}^{-1} &= \tilde{\mathbf{Z}}\left(\tilde{\mathbf{Z}}^\intercal\boldsymbol{\mathcal{F}}\tilde{\mathbf{Z}}\right)^{-1}\tilde{\mathbf{Z}}^\intercal = \mathbf{Z}\mathbf{S}^{-\intercal}\left(\mathbf{S}^{-1}\mathbf{Z}^\intercal\boldsymbol{\mathcal{F}}\mathbf{Z}\mathbf{S}^{-\intercal}\right)^{-1}\mathbf{S}^{-1}\mathbf{Z}^\intercal \\ &= \mathbf{Z}\left(\mathbf{S}\mathbf{S}^{-1}\mathbf{Z}^\intercal\boldsymbol{\mathcal{F}}\mathbf{Z}\mathbf{S}^{-\intercal}\mathbf{S}^\intercal\right)^{-1}\mathbf{Z}^\intercal \\ &= \mathbf{Z}(\mathbf{Z}^\intercal\boldsymbol{\mathcal{F}}\mathbf{Z})^{-1}\mathbf{Z}^\intercal\end{aligned} \tag{C.55}$$

Appendix D

Additional Kalman Filter Derivations

D.1 Discrete Time Approximation to Continuous Time Stochastic System

A linear, continuous time system, which is excited by a white process noise term may be described by [Sim2006, Ch. 4.2]

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}(t)\,\boldsymbol{x}(t) + \mathbf{B}(t)\,\boldsymbol{u}(t) + {}^{c}\mathbf{F}(t)\,{}^{c}\boldsymbol{w}(t) \tag{D.1}$$

$$\mathrm{E}[{}^{c}\boldsymbol{w}(t)] = \mathbf{0} \tag{D.2}$$

$$\mathrm{Cov}[{}^{c}\boldsymbol{w}(t)\,,{}^{c}\boldsymbol{w}(t+\tau)] = {}^{c}\mathbf{Q}(t)\,\delta(\tau) \tag{D.3}$$

with the continuous time, white noise process $\boldsymbol{w}(t)$ with covariance ${}^{c}\mathbf{Q}(t)$.

For a rigorous treatment of stochastic differential equations advanced mathematical tools would be necessary: actually this system model "has no mathematical meaning" [Jaz1970, p. 94], since the continuous time process noise term is delta correlated and thus not mean square Riemann integrable. Advanced integral definitions as e.g. in the ITÔ or STRATONOVICH sense are necessary to cope with this [Jaz1970], which is beyond the scope of this work.

The naive solution of above system via a variation of constants approach lacks a sound mathematical basis. Nevertheless, it still provides some useful insight, and an easy way of approximating a continuous with a discrete time systems, which is why it is found very often in the respective literature [Sim2006, Jat2006]. That said, the ideas behind that approximation are illustrated next.

If the time step $\Delta t_k = t_{k+1} - t_k$ is sufficiently small, changes in $\mathbf{A}(t)$, $\mathbf{B}(t)$, ${}^{c}\mathbf{F}(t)$, ${}^{c}\mathbf{Q}(t)$ and $\boldsymbol{u}(t)$ may be neglected, and the respective quantities considered constant during this time step ("zero order hold" assumption for all time-varying quantities). Then the state value at the next time step $k+1$ can be obtained based on the state value at the current time step k by applying the solution due to variation of constants, see [Sim2006, Ch. 1][CJ2012, Ch. 3.1]. Replacing the argument (t_k) by the subscript $\square_k$

yields

$$\begin{aligned}\boldsymbol{x}_{k+1} = \exp(\mathbf{A}_k(t_{k+1}-t_k))\,\boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,d\tau \mathbf{B}_k \boldsymbol{u}_{k+1} \\ + \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,{}^c\mathbf{F}_k\,{}^c\boldsymbol{w}(\tau)\,d\tau\end{aligned} \tag{D.4}$$

The three parts of above equation can then be related to a standard, discrete time system description, as e.g. (2.180). The first part is [CJ2012, Ch. 3.5] [Jat2015, Ch. 3]

$$\boldsymbol{\Phi}_k = \exp(\mathbf{A}_k(t_{k+1}-t_k)) = \sum_{j=0}^{\infty} \frac{(\mathbf{A}_k \Delta t_k)^j}{j!} \tag{D.5}$$

The second part yields, after substitution of $\alpha = t_{k+1} - \tau$ [CJ2012, Ch. 3.5] [Jat2015, Ch. 3]

$$\begin{aligned}\boldsymbol{\Gamma}_k &= \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,d\tau\mathbf{B}_k = -\int_{\Delta t_k}^{0} \exp(\mathbf{A}_k\alpha)\,d\alpha\mathbf{B}_k \\ &= \int_0^{\Delta t_k} \sum_{j=0}^{\infty} \frac{(\mathbf{A}_k\alpha)^j}{j!} d\alpha\mathbf{B}_k = \left[\sum_{j=0}^{\infty} \frac{\mathbf{A}_k^j \alpha^{j+1}}{(j+1)!}\right]_0^{\Delta t_k} \mathbf{B}_k \\ &= \sum_{j=0}^{\infty} \frac{\mathbf{A}_k^j \Delta t_k^{j+1}}{(j+1)!}\mathbf{B}_k\end{aligned} \tag{D.6}$$

Since the contribution of the terms in above sums shrink with $\mathcal{O}\left(\Delta t_k^j\right)$, it is usually enough to consider the first few [CJ2012, Ch. 3.5]. Also, above expression for $\boldsymbol{\Gamma}_k$ is computationally more stable, compared to the direct solution of

$$\begin{aligned}\boldsymbol{\Gamma}_k &= \int_0^{\Delta t_k} \exp(\mathbf{A}_k\alpha)\,d\alpha\mathbf{B}_k = \left[\mathbf{A}_k^{-1}\exp(\mathbf{A}_k\alpha)\right]_0^{\Delta t_k}\mathbf{B}_k \\ &= \mathbf{A}_k^{-1}(\exp(\mathbf{A}_k\Delta t_k) - \mathbf{I}_{n_x})\,\mathbf{B}_k\end{aligned} \tag{D.7}$$

Since this formulation shows problems for rank-deficient $\mathbf{A}_k$.

The treatment of the third term of equation (D.4), which describes the influence of process noise on the system's evolution is not as straightforward anymore. One approach is to define a new random variable

$$\mathbf{F}_k\boldsymbol{w}_k = \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,{}^c\mathbf{F}_k\,{}^c\boldsymbol{w}(\tau)\,d\tau \tag{D.8}$$

and investigate its moments further. It has zero mean [Sim2006, Ch. 4.2]

$$\begin{aligned}\mathrm{E}[\mathbf{F}_k\boldsymbol{w}_k] &= \mathrm{E}\left[\int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,{}^c\mathbf{F}_k\,{}^c\boldsymbol{w}(\tau)\,d\tau\right] \\ &= \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\tau))\,{}^c\mathbf{F}_k \underbrace{\mathrm{E}[{}^c\boldsymbol{w}(\tau)]}_{=\mathbf{0}}\,d\tau = \mathbf{0}\end{aligned} \tag{D.9}$$

and its covariance can be computed as follows (using the sifting property of the Dirac function) [Sim2006, Ch. 4.2] [CJ2012, Ch. 5.4]

$$\begin{aligned}
&\mathbf{F}_k \mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_l]\, \mathbf{F}_l^\intercal = \\
&= \mathrm{E}\left[\left(\int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\alpha))\, {}^c\mathbf{F}_k\, {}^c\boldsymbol{w}(\alpha)\, d\alpha\right)\left(\int_{t_l}^{t_{l+1}} \exp(\mathbf{A}_l(t_{l+1}-\beta))\, {}^c\mathbf{F}_l\, {}^c\boldsymbol{w}(\beta)\, d\beta\right)^\intercal\right] \\
&= \int_{t_k}^{t_{k+1}} \int_{t_l}^{t_{l+1}} \exp(\mathbf{A}_k(t_{k+1}-\alpha))\, {}^c\mathbf{F}_k \mathrm{E}[{}^c\boldsymbol{w}(\alpha)\, {}^c\boldsymbol{w}(\beta)^\intercal]\, {}^c\mathbf{F}_l^\intercal \exp(\mathbf{A}_l(t_{l+1}-\beta))^\intercal\, d\beta d\alpha \\
&= \int_{t_k}^{t_{k+1}} \int_{t_l}^{t_{l+1}} \exp(\mathbf{A}_k(t_{k+1}-\alpha))\, {}^c\mathbf{F}_k\, {}^c\mathbf{Q}(\alpha)\, \delta(\beta-\alpha)\, {}^c\mathbf{F}_l^\intercal \exp(\mathbf{A}_l(t_{l+1}-\beta))^\intercal\, d\beta d\alpha \\
&= \delta_{k-l} \int_{t_k}^{t_{k+1}} \exp(\mathbf{A}_k(t_{k+1}-\alpha))\, {}^c\mathbf{F}_k\, {}^c\mathbf{Q}(\alpha)\, {}^c\mathbf{F}_k^\intercal \exp(\mathbf{A}(t_{k+1}-\alpha))^\intercal\, d\alpha
\end{aligned} \tag{D.10}$$

SIMON states, that above integral is difficult to solve in general, but proposes the following approximation for sufficiently small Δt_k [Sim2006, Ch. 4.2] [CJ2012, Ch. 5.4]

$$\exp(\mathbf{A}_k(t_{k+1}-\alpha)) \approx \mathbf{I}_{n_{\boldsymbol{x}}} \qquad \forall \alpha \in [t_k, t_{k+1}] \tag{D.11}$$

The covariance then becomes

$$\mathbf{F}_k \mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_l]\, \mathbf{F}_l^\intercal = \delta_{k-l} \int_{t_k}^{t_{k+1}} {}^c\mathbf{F}_k\, {}^c\mathbf{Q}(\alpha)\, {}^c\mathbf{F}_k^\intercal d\alpha \approx \delta_{k-l}\, {}^c\mathbf{F}_k\, {}^c\mathbf{Q}(t_k)\, {}^c\mathbf{F}_k^\intercal \Delta t_k \tag{D.12}$$

where the approximation is based on the assumption that the integrand is constant during the small time period Δt_k. From this the two following may be identified

$$\begin{aligned}
&\Rightarrow \mathbf{F}_k = {}^c\mathbf{F}_k \\
&\Rightarrow \mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_l] \approx \delta_{k-l}\, {}^c\mathbf{Q}(t_k)\, \Delta t_k
\end{aligned} \tag{D.13}$$

Thus the zero-mean, continuous time white noise process can be approximated up to the second moments by a zero-mean, discrete time white noise process, whose covariance is scaled by the sampling time ${}^c\mathbf{Q}(t_k)\, \Delta t_k$.

Some authors prefer to scale the process noise distribution matrix, and use the same values for the covariance matrices in discrete and continuous time

$$\begin{aligned}
&\Rightarrow \mathbf{F}_k = {}^c\mathbf{F}_k \sqrt{\Delta t_k} \\
&\Rightarrow \mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_l] \approx \delta_{k-l}\, {}^c\mathbf{Q}(t_k)
\end{aligned} \tag{D.14}$$

Here, the first approach will be used.

Above approximation becomes true, for a pure random walk process with $\mathbf{A} = \mathbf{0}$, i.e. in the case of a random walk model, and constant system matrices. Then the effect (up to second order) of a discrete white noise process with

$$\mathrm{Cov}[\boldsymbol{w}_k, \boldsymbol{w}_{k+l}] = \delta_l \Delta t_k\, {}^c\mathbf{Q}(t_k) \tag{D.15}$$

is the same as the influence of a continuous white noise process with

$$\mathrm{Cov}[{}^c\boldsymbol{w}(t)\,, {}^c\boldsymbol{w}(t+\tau)] = {}^c\mathbf{Q}\delta(\tau) \tag{D.16}$$

D.2 Kalman Filter Residuals

The following section is based on [Sim2006]. In order to keep the equations more readable, and indexing more straight forward, the correction step is considered at time k as opposed to $k+1$ as in the main text. Then, propagated quantities read $\square_{k|k-1}$ and corrected quantities become $\square_{k|k}$.

According to equation (2.219), the Kalman filter innovations can be expressed as

$$\begin{aligned} \boldsymbol{r}_{k|k-1} &= \boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1} = \mathbf{C}_k \boldsymbol{x}_k + \mathbf{G}_k \boldsymbol{v}_k + \mathbf{D}_k \boldsymbol{u}_k - \mathbf{C}_k \hat{\boldsymbol{x}}_{k|k-1} - \mathbf{D}_k \boldsymbol{u}_k \\ &= -\mathbf{C}_k \tilde{\boldsymbol{x}}_{k|k-1} + \mathbf{G}_k \boldsymbol{v}_k \end{aligned} \tag{D.17}$$

Both the measurement noise $\boldsymbol{v}_k$ and the a priori estimation error $\tilde{\boldsymbol{x}}_{k|k-1}$ are zero mean, thus the residuals $\boldsymbol{r}_{k|k-1}$ also are zero-mean.

The autocovariance of the residuals is

$$\begin{aligned} \mathrm{Cov}\left[\boldsymbol{r}_{k|k-1}, \boldsymbol{r}_{j|j-1}\right] &= \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \tilde{\boldsymbol{x}}_{j|j-1}\right] \mathbf{C}_j^\intercal - \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_j\right] \mathbf{G}_j^\intercal \\ &\quad - \mathbf{G}_k \mathrm{Cov}\left[\boldsymbol{v}_k, \tilde{\boldsymbol{x}}_{j|j-1}\right] \mathbf{C}_j^\intercal + \mathbf{G}_k \mathrm{Cov}[\boldsymbol{v}_k, \boldsymbol{v}_j] \mathbf{G}_j^\intercal \end{aligned} \tag{D.18}$$

The two separate cases to be treated consider the covariance at zero-lag ($k = l$) and the covariance between sampling instants ($k \neq l$).

For $k = j$ the cross terms vanish $\mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_k\right] = \mathrm{Cov}\left[\boldsymbol{v}_k, \tilde{\boldsymbol{x}}_{k|k-1}\right]^\intercal = \mathbf{0}$, since the measurements at time k were not yet processed and the white measurement noise $\boldsymbol{v}_k$ cannot yet be part of the estimation error $\tilde{\boldsymbol{x}}_{k|k-1}$ if $k = l$. Thus the covariance is

$$\mathrm{Cov}\left[\boldsymbol{r}_{k|k-1}, \boldsymbol{r}_{k|k-1}\right] = \mathbf{C}_k \mathbf{P}_{k|k-1}^{\tilde{\mathbf{x}}} \mathbf{C}_k^\intercal + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^\intercal \tag{D.19}$$

For $k \neq j$, without loss of generality it can be assumed that $k > j$. Then, due to the whiteness of $\boldsymbol{v}_k$ and the fact that $\tilde{\boldsymbol{x}}_{j|j-1}$ is independent of $\boldsymbol{v}_k$ for $k > j$, the autocovariance is

$$\begin{aligned} \mathrm{Cov}\left[\boldsymbol{r}_{k|k-1}, \boldsymbol{r}_{j|j-1}\right] &= \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \tilde{\boldsymbol{x}}_{j|j-1}\right] \mathbf{C}_j^\intercal - \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_j\right] \mathbf{G}_j^\intercal \\ &\quad - \underbrace{\mathbf{G}_k \mathrm{Cov}\left[\boldsymbol{v}_k, \tilde{\boldsymbol{x}}_{j|j-1}\right] \mathbf{C}_j^\intercal}_{=\mathbf{0} \text{ if } k>j} + \underbrace{\mathbf{G}_k \mathrm{Cov}[\boldsymbol{v}_k, \boldsymbol{v}_j] \mathbf{G}_j^\intercal}_{=\mathbf{0} \text{ if } k>j} \\ &= \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \tilde{\boldsymbol{x}}_{j|j-1}^\intercal\right] \mathbf{C}_j^\intercal - \mathbf{C}_k \mathrm{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_j\right] \mathbf{G}_j^\intercal \end{aligned} \tag{D.20}$$

In order to determine the covariances in above equation, first a propagation equation for the a priori estimate has to be obtained and solved. From equation (2.214) the propagation for the a priori estimation error can be obtained and slightly modified

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k|k-1} &= \boldsymbol{\Phi}_{k-1}(\mathbf{I}_{n_x} - \mathbf{K}_{k-1}\mathbf{C}_{k-1})\, \tilde{\boldsymbol{x}}_{k-1|k-2} - \mathbf{F}_{k-1}\boldsymbol{w}_{k-1} + \boldsymbol{\Phi}_{k-1}\mathbf{K}_{k-1}\mathbf{G}_{k-1}\boldsymbol{v}_{k-1} \\ &= \boldsymbol{\Phi}'_{k-1}\tilde{\boldsymbol{x}}_{k-1|k-2} + \boldsymbol{v}'_{k-1} \end{aligned} \tag{D.21}$$

$$\boldsymbol{\Phi}'_{k-1} = \boldsymbol{\Phi}_{k-1}(\mathbf{I}_{n_x} - \mathbf{K}_{k-1}\mathbf{C}_{k-1}) \tag{D.22}$$

$$\boldsymbol{v}'_{k-1} = \boldsymbol{\Phi}_{k-1}\mathbf{K}_{k-1}\mathbf{G}_{k-1}\boldsymbol{v}_{k-1} - \mathbf{F}_{k-1}\boldsymbol{w}_{k-1} \tag{D.23}$$

The solution from an initial condition $\tilde{\boldsymbol{x}}_{j|j-1}$ is

$$\boldsymbol{\Phi}'_{k,j} = \begin{cases} \boldsymbol{\Phi}'_{k-1} \cdot \boldsymbol{\Phi}'_{k-2} \cdot \dots \cdot \boldsymbol{\Phi}'_{j} & k > j \\ \mathbf{I}_{n_x} & k = j \end{cases} \tag{D.24}$$

$$\tilde{\boldsymbol{x}}_{k|k-1} = \boldsymbol{\Phi}'_{k,j}\tilde{\boldsymbol{x}}_{j|j-1} + \sum_{l=j}^{k-1} \boldsymbol{\Phi}'_{k,l+1}\boldsymbol{v}'_l \tag{D.25}$$

Now the first desired covariance for the solution of (D.20) can be determined

$$\begin{aligned} \operatorname{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \tilde{\boldsymbol{x}}_{j|j-1}\right] &= \mathrm{E}\left[\left(\boldsymbol{\Phi}'_{k,j}\tilde{\boldsymbol{x}}_{j|j-1} + \sum_{l=j}^{k-1} \boldsymbol{\Phi}'_{k,l+1}\boldsymbol{v}'_l\right)\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}\right] \\ &= \mathrm{E}\left[\boldsymbol{\Phi}'_{k,j}\tilde{\boldsymbol{x}}_{j|j-1}\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}\right] + \mathrm{E}\left[\left(\sum_{l=j}^{k-1} \boldsymbol{\Phi}'_{k,l+1}\boldsymbol{v}'_l\right)\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}\right] \end{aligned} \tag{D.26}$$

Multiplying out the second part of above equation yields terms of the form $\boldsymbol{v}'_l\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}$, which are zero-mean $\mathrm{E}\left[\boldsymbol{v}'_l\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}\right] = \mathbf{0}$ $l \geq j$ because the noise with $l \geq j$ cannot yet have influenced the estimation error $\tilde{\boldsymbol{x}}_{j|j-1}$, see (D.21).

$$\operatorname{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \tilde{\boldsymbol{x}}_{j|j-1}\right] = \mathrm{E}\left[\boldsymbol{\Phi}'_{k,j}\tilde{\boldsymbol{x}}_{j|j-1}\tilde{\boldsymbol{x}}^{\mathsf{T}}_{j|j-1}\right] = \boldsymbol{\Phi}'_{k,j}\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1} \tag{D.27}$$

The second covariance for the solution of (D.20) is

$$\operatorname{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_j\right] = \mathrm{E}\left[\left(\boldsymbol{\Phi}'_{k,j}\tilde{\boldsymbol{x}}_{j|j-1} + \sum_{l=j}^{k-1} \boldsymbol{\Phi}'_{k,l+1}\boldsymbol{v}'_l\right)\boldsymbol{v}^{\mathsf{T}}_j\right] \tag{D.28}$$

Using similar arguments as before, the only remaining, non-zero-mean term results for $l = j$

$$\begin{aligned} \operatorname{Cov}\left[\tilde{\boldsymbol{x}}_{k|k-1}, \boldsymbol{v}_j\right] &= \mathrm{E}\left[\boldsymbol{\Phi}'_{k,j+1}\boldsymbol{v}'_j\boldsymbol{v}^{\mathsf{T}}_j\right] \overset{\text{(D.23)}}{=} \mathrm{E}\left[\boldsymbol{\Phi}'_{k,j+1}(\boldsymbol{\Phi}_j\mathbf{K}_j\mathbf{G}_j\boldsymbol{v}_j - \mathbf{F}_j\boldsymbol{w}_j)\,\boldsymbol{v}^{\mathsf{T}}_j\right] \\ &= \boldsymbol{\Phi}'_{k,j+1}\boldsymbol{\Phi}_j\mathbf{K}_j\mathbf{G}_j\mathbf{R}_j \end{aligned} \tag{D.29}$$

Then the result can be combined to obtain an expression for the residual's autocovariance in equation (D.20)

$$\begin{aligned} \operatorname{Cov}\left[\boldsymbol{r}_{k|k-1}, \boldsymbol{r}_{j|j-1}\right] &= \mathbf{C}_k\boldsymbol{\Phi}'_{k,j}\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j - \mathbf{C}_k\boldsymbol{\Phi}'_{k,j+1}\boldsymbol{\Phi}_j\mathbf{K}_j\underbrace{\mathbf{G}_j\mathbf{R}_j\mathbf{G}^{\mathsf{T}}_j}_{=\tilde{\mathbf{R}}_j} \\ &= \mathbf{C}_k\boldsymbol{\Phi}'_{k,j+1}\left(\boldsymbol{\Phi}'_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j - \boldsymbol{\Phi}_j\mathbf{K}_j\tilde{\mathbf{R}}_j\right) \\ &= \mathbf{C}_k\boldsymbol{\Phi}'_{k,j+1}\Big(\underbrace{\boldsymbol{\Phi}_j(\mathbf{I}_{n_x} - \mathbf{K}_j\mathbf{C}_j)}_{\overset{\text{(D.22)}}{=}\boldsymbol{\Phi}'_j}\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j - \boldsymbol{\Phi}_j\mathbf{K}_j\tilde{\mathbf{R}}_j\Big) \\ &= \mathbf{C}_k\boldsymbol{\Phi}'_{k,j+1}\Big(\boldsymbol{\Phi}_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j - \boldsymbol{\Phi}_j\mathbf{K}_j\Big(\mathbf{C}_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j + \tilde{\mathbf{R}}_j\Big)\Big) \end{aligned} \tag{D.30}$$

The last step is to use the Kalman filter gain equation (2.212) to arrive at

$$\mathbf{K}_j = \mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j\Big(\mathbf{C}_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j + \tilde{\mathbf{R}}_j\Big)^{-1} \tag{D.31}$$

$$\operatorname{Cov}\left[\boldsymbol{r}_{k|k-1}, \boldsymbol{r}_{j|j-1}\right] = \mathbf{C}_k\boldsymbol{\Phi}'_{k,j+1}\Big(\boldsymbol{\Phi}_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j - \boldsymbol{\Phi}_j\mathbf{P}^{\tilde{\mathbf{x}}}_{j|j-1}\mathbf{C}^{\mathsf{T}}_j\Big) = \mathbf{0} \tag{D.32}$$

Thus it is shown that the residuals $\boldsymbol{r}_{k|k-1}$ are white and, zero-mean

$$\mathrm{E}\Big[\boldsymbol{r}_{k|k-1}\Big] = \mathbf{0} \tag{D.33}$$

$$\mathrm{Cov}\Big[\boldsymbol{r}_{k|k-1}\,,\boldsymbol{r}_{j|j-1}\Big] = \delta_{kj}\Big(\mathbf{C}_k\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\,\mathbf{C}_k^{\intercal} + \tilde{\mathbf{R}}_k\Big) \tag{D.34}$$

D.3 Relating Continuous and Discrete Non-Linear System Descriptions

In many applications of the extended Kalman filter to system identification problems, the system description is in the form of a system of continuous time differential equations

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)\,,\boldsymbol{u}(t)) \tag{D.35}$$

This can be transformed into a non-linear, discrete difference equation via an explicit s-stage Runge-Kutta integration scheme [Ger2018, Ch. 5]

$$\boldsymbol{k}_j(\boldsymbol{x}_k;\Delta t_k) = \boldsymbol{f}\left(\boldsymbol{x}_k + \Delta t_k \sum_{l=1}^{j-1} a_{jl}\boldsymbol{k}_l(\boldsymbol{x}_k;\Delta t_k)\,,\boldsymbol{u}(t_k + c_j\Delta t_k)\right) \tag{D.36}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t_k \sum_{j=1}^{s} b_j \cdot \boldsymbol{k}_j(\boldsymbol{x}_k;\Delta t_k) \tag{D.37}$$

The input term may be approximated to simplify computations

$$\text{zero-order hold:}\quad \boldsymbol{u}(t_k + c_j\Delta t_k) \approx \boldsymbol{u}_k \tag{D.38}$$

$$\text{linear interpolation:}\quad \boldsymbol{u}(t_k + c_j\Delta t_k) \approx \boldsymbol{u}_k(1 - c_j) + \boldsymbol{u}_{k+1}c_j \tag{D.39}$$

The coefficients a_{jl}, b_j, and c_l determine the characteristics of the integration scheme. They are usually chosen by comparing Taylor series expansions of the true solution with Taylor series expansions of the numerical solution and matching the coefficients of terms up to a certain order of the derivatives. The coefficients can be arranged

Table D.1: *Examples for different Butcher-Tableaux*

c_1	0			
$\vdots$	a_{21}	0		
$\vdots$	$\vdots$	$\ddots$	$\ddots$	
c_s	a_{s1}	$\cdots$	$a_{s,s-1}$	0
	b_1	$\cdots$	$\cdots$	b_s

(a) *a generic Butcher Tableau for an explicit s-stage integration*

0	0
	1

(b) *the Butcher Tableau for Euler Forward integration*

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

(c) *the Butcher Tableau fourth order Runge-Kutta integration*

in so-called Butcher Tableaux, see table D.1. Next to the generic layout for explicit schemes D.1a, two widely used integration schemes are shown as examples: simple Euler forward integration in D.1b and a fourth order Runge-Kutta integration in D.1c.

For *explicit* integration schemes, the upper right triangular part of the a_{jl} coefficients has to be equal to zero, otherwise above equation would be implicit in $\boldsymbol{k}_j$ and iterative solvers would be necessary. In the state estimation cases treated here, only explicit integration schemes are used, allowing for the following approximation of the solution of a continuous time, non-linear system of differential equations

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t_k \sum_{j=1}^{s} b_j \cdot \boldsymbol{k}_j(\boldsymbol{x}_k; \Delta t_k) = {}^{d}\mathbf{f}_k[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k] \tag{D.40}$$

D.4 Equality of Maximum Likelihood and Minimum Variance Filtering

Again, the correction step is considered to be taken at k to keep indexing more straightforward, as was done in appendix D.2.

The update step based on maximum likelihood considerations is

$$\hat{\boldsymbol{x}}_{k|k} = \left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right)^{-1} + \mathbf{C}_k^\intercal \tilde{\mathbf{R}}_k^{-1} \mathbf{C}_k\right)^{-1} \left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right)^{-1} \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{C}_k^\intercal \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k)\right) \tag{D.41}$$

Using the following matrix identity [Jaz1970, Appendix 7B] [Ho1963]

$$\left(\mathbf{P}^{-1} + \mathbf{C}^\intercal \mathbf{R}^{-1} \mathbf{C}\right)^{-1} = \mathbf{P} - \mathbf{P}\mathbf{C}^\intercal (\mathbf{C}\mathbf{P}\mathbf{C}^\intercal + \mathbf{R})^{-1} \mathbf{C}\mathbf{P} \tag{D.42}$$

the update becomes

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} = &\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right) \\ &\left(\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right)^{-1} \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{C}_k^\intercal \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k)\right) \end{aligned} \tag{D.43}$$

Now, the second factor in braces is multiplied out

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} = &\left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right) \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right)^{-1} \hat{\boldsymbol{x}}_{k|k-1} \\ &+ \left(\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1}\right) \mathbf{C}_k^\intercal \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k) \\ = &\Big(\mathbf{I}_{n_{\boldsymbol{x}}} - \underbrace{\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1}}_{=\mathbf{K}_k} \mathbf{C}_k\Big) \hat{\boldsymbol{x}}_{k|k-1} \\ &+ \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{I}_{n_{\boldsymbol{y}}} - \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal\right) \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k) \end{aligned} \tag{D.44}$$

Adding an advantageously chosen zero term in the second part allows to simplify further

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k} =& \left(\mathbf{I}_{n_{\boldsymbol{x}}} - \mathbf{K}_k \mathbf{C}_k\right) \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \Big(\mathbf{I}_{n_{\boldsymbol{y}}} - \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \\
& \underbrace{- \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \tilde{\mathbf{R}}_k + \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \tilde{\mathbf{R}}_k}_{=0} \Big) \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k) \\
=& \left(\mathbf{I}_{n_{\boldsymbol{x}}} - \mathbf{K}_k \mathbf{C}_k\right) \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \Big(\mathbf{I}_{n_{\boldsymbol{y}}} + \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \tilde{\mathbf{R}}_k \\
& \underbrace{- \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1} \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)}_{=\mathbf{I}_{n_{\boldsymbol{y}}}} \Big) \tilde{\mathbf{R}}_k^{-1} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k) \\
=& \left(\mathbf{I}_{n_{\boldsymbol{x}}} - \mathbf{K}_k \mathbf{C}_k\right) \hat{\boldsymbol{x}}_{k|k-1} + \underbrace{\mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal \left(\mathbf{C}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k-1} \mathbf{C}_k^\intercal + \tilde{\mathbf{R}}_k\right)^{-1}}_{=\mathbf{K}_k} (\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k) \\
=& \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k \left(\boldsymbol{z}_k - \mathbf{D}_k \boldsymbol{u}_k - \mathbf{C}_k \hat{\boldsymbol{x}}_{k|k-1}\right) = \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{K}_k \left(\boldsymbol{z}_k - \hat{\boldsymbol{y}}_{k|k-1}\right)
\end{aligned} \tag{D.45}$$

This last step shows the equality to the update step (2.207) using a gain obtained via a minimum trace approach (2.212).

D.5 Smoothed State Error Covariance Estimate

The backward smoothing recursion is

$$\hat{\boldsymbol{x}}_{k\big|\bar{N}} = \hat{\boldsymbol{x}}_{k|k} + \mathbf{M}_k \left(\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} - \hat{\boldsymbol{x}}_{k+1|k}\right) \tag{D.46}$$

In order to compute the covariance of the smoothing state estimate, the true state vector $\boldsymbol{x}_k$ is subtracted from both sides of the backwards recursion equation, and the terms are re-arranged [RTS1965]

$$\tilde{\boldsymbol{x}}_{k\big|\bar{N}} - \mathbf{M}_k \hat{\boldsymbol{x}}_{k+1\big|\bar{N}} = \tilde{\boldsymbol{x}}_{k|k} - \mathbf{M}_k \hat{\boldsymbol{x}}_{k+1|k} \tag{D.47}$$

Now, as presented in [Jaz1970, Ch 7], both sides are squared and the expectation is taken

$$\begin{aligned}
& \mathrm{E}\left[\tilde{\boldsymbol{x}}_{k\big|\bar{N}} \tilde{\boldsymbol{x}}_{k\big|\bar{N}}^\intercal\right] + \mathbf{M}_k \mathrm{E}\left[\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} \hat{\boldsymbol{x}}_{k+1\big|\bar{N}}^\intercal\right] \mathbf{M}_k^\intercal \\
& - \mathrm{E}\left[\tilde{\boldsymbol{x}}_{k\big|\bar{N}} \hat{\boldsymbol{x}}_{k+1\big|\bar{N}}^\intercal\right] \mathbf{M}_k^\intercal - \mathbf{M}_k \mathrm{E}\left[\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} \tilde{\boldsymbol{x}}_{k\big|\bar{N}}^\intercal\right] = \\
& = \mathrm{E}\left[\tilde{\boldsymbol{x}}_{k|k} \tilde{\boldsymbol{x}}_{k|k}^\intercal\right] + \mathbf{M}_k \mathrm{E}\left[\hat{\boldsymbol{x}}_{k+1|k} \hat{\boldsymbol{x}}_{k+1|k}^\intercal\right] \mathbf{M}_k^\intercal \\
& - \mathrm{E}\left[\tilde{\boldsymbol{x}}_{k|k} \left(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right)^\intercal\right] \mathbf{M}_k^\intercal - \mathrm{E}\left[\left(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k\right) \tilde{\boldsymbol{x}}_{k|k}^\intercal\right] \mathbf{M}_k
\end{aligned} \tag{D.48}$$

On the right hand side of above equation, the propagation equation (2.202) was used to replace $\hat{\boldsymbol{x}}_{k+1|k}$ in the cross terms. This can be simplified by using the following equalities from [Jaz1970, Ch. 7 eq. (7.89)] or [RTS1965] respectively, which give a statement

about the orthogonality of the estimate and accompanying estimation errors

$$\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] = \mathbf{0} \tag{D.49}$$

$$\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|k}\,\hat{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] = \mathbf{0} \tag{D.50}$$

Furthermore, assuming that the conditions for an unbiased state estimate are met, it holds

$$\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|\bar{N}}\Big] = \mathbf{0} \quad \Rightarrow \quad \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|\bar{N}}\Big] = \mathrm{E}[\boldsymbol{x}_k] \tag{D.51}$$

$$\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|k}\Big] = \mathbf{0} \quad \Rightarrow \quad \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k}\Big] = \mathrm{E}[\boldsymbol{x}_k] \tag{D.52}$$

Then all cross terms are equal to $\mathbf{0}$ and above equation simplifies to

$$\begin{aligned}\mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|\bar{N}}\,\tilde{\boldsymbol{x}}_{k|\bar{N}}^{\mathsf{T}}\Big] + \mathbf{M}_k\mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big]\,\mathbf{M}_k^{\mathsf{T}} = \\ = \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|k}\,\tilde{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] + \mathbf{M}_k\mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|k}\,\hat{\boldsymbol{x}}_{k+1|k}^{\mathsf{T}}\Big]\,\mathbf{M}_k^{\mathsf{T}}\end{aligned} \tag{D.53}$$

Here, the corrected and smoothed state error covariance matrices may be identified

$$\begin{aligned}\mathbf{P}_{k|\bar{N}}^{\tilde{\mathbf{x}}} + \mathbf{M}_k\mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big]\,\mathbf{M}_k^{\mathsf{T}} = \\ = \mathbf{P}_{k|k}^{\tilde{\mathbf{x}}} + \mathbf{M}_k\mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|k}\,\hat{\boldsymbol{x}}_{k+1|k}^{\mathsf{T}}\Big]\,\mathbf{M}_k^{\mathsf{T}}\end{aligned} \tag{D.54}$$

Solving for $\mathbf{P}_{k|\bar{N}}^{\tilde{\mathbf{x}}}$ yields

$$\begin{aligned}\mathbf{P}_{k|\bar{N}}^{\tilde{\mathbf{x}}} &= \mathbf{P}_{k|k}^{\tilde{\mathbf{x}}} + \mathbf{M}_k\Big(\mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|k}\,\hat{\boldsymbol{x}}_{k+1|k}^{\mathsf{T}}\Big] - \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big]\Big)\,\mathbf{M}_k^{\mathsf{T}} \\ &= \mathbf{P}_{k|k}^{\tilde{\mathbf{x}}} + \mathbf{M}_k\Big(\mathrm{E}\Big[\big(\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k\boldsymbol{u}_k\big)\big(\boldsymbol{\Phi}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k\boldsymbol{u}_k\big)^{\mathsf{T}}\Big] - \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big]\Big)\,\mathbf{M}_k^{\mathsf{T}}\end{aligned} \tag{D.55}$$

The last step is to identify and cancel the common parts of the terms in brackets on the right. Using equations (D.49) and (D.50), as well as the model equation (2.180), the remaining expectations can be expressed as

$$\begin{aligned}\mathrm{E}[\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}] &= \mathrm{E}\Big[\big(\hat{\boldsymbol{x}}_{k|k} + \tilde{\boldsymbol{x}}_{k|k}\big)\big(\hat{\boldsymbol{x}}_{k|k} + \tilde{\boldsymbol{x}}_{k|k}\big)^{\mathsf{T}}\Big] \\ &= \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k}\,\hat{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] + \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|k}\,\tilde{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] + \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k}\,\tilde{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] + \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k|k}\,\hat{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] \\ &= \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k}\,\hat{\boldsymbol{x}}_{k|k}^{\mathsf{T}}\Big] + \mathbf{P}_{k|k}^{\tilde{\mathbf{x}}}\end{aligned} \tag{D.56}$$

$$\begin{aligned}\mathrm{E}\Big[\boldsymbol{x}_{k+1}\boldsymbol{x}_{k+1}^{\mathsf{T}}\Big] &= \mathrm{E}\Big[\Big(\hat{\boldsymbol{x}}_{k+1|\bar{N}} + \tilde{\boldsymbol{x}}_{k+1|\bar{N}}\Big)\Big(\hat{\boldsymbol{x}}_{k+1|\bar{N}} + \tilde{\boldsymbol{x}}_{k+1|\bar{N}}\Big)^{\mathsf{T}}\Big] \\ &= \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] + \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k+1|\bar{N}}\,\tilde{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] \\ &\qquad + \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\tilde{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] + \mathrm{E}\Big[\tilde{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] \\ &= \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1|\bar{N}}\,\hat{\boldsymbol{x}}_{k+1|\bar{N}}^{\mathsf{T}}\Big] + \mathbf{P}_{k+1|\bar{N}}^{\tilde{\mathbf{x}}}\end{aligned} \tag{D.57}$$

$$\begin{aligned}\mathrm{E}\Big[\boldsymbol{x}_{k+1}\boldsymbol{x}_{k+1}^{\mathsf{T}}\Big] &= \mathrm{E}[(\boldsymbol{\Phi}_k\boldsymbol{x}_k + \boldsymbol{\Gamma}_k\boldsymbol{u}_k + \mathbf{F}_k\boldsymbol{w}_k)(\boldsymbol{\Phi}_k\boldsymbol{x}_k + \boldsymbol{\Gamma}_k\boldsymbol{u}_k + \mathbf{F}_k\boldsymbol{w}_k)^{\mathsf{T}}] \\ &= \boldsymbol{\Phi}_k\mathrm{E}[\boldsymbol{x}_k\boldsymbol{x}_k^{\mathsf{T}}]\,\boldsymbol{\Phi}_k^{\mathsf{T}} + \boldsymbol{\Gamma}_k\boldsymbol{u}_k\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{\Gamma}_k^{\mathsf{T}} + \mathbf{F}_k\mathbf{Q}_k\mathbf{F}_k^{\mathsf{T}} \\ &\qquad + \boldsymbol{\Phi}_k\mathrm{E}[\boldsymbol{x}_k]\,\boldsymbol{u}_k^{\mathsf{T}}\boldsymbol{\Gamma}_k^{\mathsf{T}} + \boldsymbol{\Gamma}_k\boldsymbol{u}_k\mathrm{E}[\boldsymbol{x}_k^{\mathsf{T}}]\,\boldsymbol{\Phi}_k^{\mathsf{T}}\end{aligned} \tag{D.58}$$

Collecting all this together, the final expression for the error covariance of the smooth state estimate is

$$
\begin{aligned}
\mathbf{P}^{\tilde{\mathbf{x}}}_{k\big|\bar{N}} =& \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k \Big(\mathrm{E}\Big[\Big(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \Big) \Big(\boldsymbol{\Phi}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \Big)^{\mathsf{T}} \Big] - \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} \hat{\boldsymbol{x}}^{\mathsf{T}}_{k+1\big|\bar{N}} \Big] \Big) \mathbf{M}_k^{\mathsf{T}} \\
=& \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k \Big(\boldsymbol{\Phi}_k \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k} \hat{\boldsymbol{x}}^{\mathsf{T}}_{k|k} \Big] \boldsymbol{\Phi}_k^{\mathsf{T}} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \boldsymbol{u}_k^{\mathsf{T}} \boldsymbol{\Gamma}_k^{\mathsf{T}} + \boldsymbol{\Phi}_k \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k|k} \Big] \boldsymbol{u}_k^{\mathsf{T}} \boldsymbol{\Gamma}_k^{\mathsf{T}} \\
& + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \mathrm{E}\Big[\hat{\boldsymbol{x}}^{\mathsf{T}}_{k|k} \Big] \boldsymbol{\Phi}_k^{\mathsf{T}} - \mathrm{E}\Big[\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} \hat{\boldsymbol{x}}^{\mathsf{T}}_{k+1\big|\bar{N}} \Big] \Big) \mathbf{M}_k^{\mathsf{T}} \\
=& \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k \Bigg(\boldsymbol{\Phi}_k \underbrace{\Big(\mathrm{E}[\boldsymbol{x}_k \boldsymbol{x}_k^{\mathsf{T}}] - \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \Big)}_{\overset{(D.56)}{=} \mathrm{E}\left[\hat{\boldsymbol{x}}_{k|k} \hat{\boldsymbol{x}}^{\mathsf{T}}_{k|k} \right]} \boldsymbol{\Phi}_k^{\mathsf{T}} + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \boldsymbol{u}_k^{\mathsf{T}} \boldsymbol{\Gamma}_k^{\mathsf{T}} + \boldsymbol{\Phi}_k \mathrm{E}[\boldsymbol{x}_k] \boldsymbol{u}_k^{\mathsf{T}} \boldsymbol{\Gamma}_k^{\mathsf{T}} \\
& + \boldsymbol{\Gamma}_k \boldsymbol{u}_k \mathrm{E}[\boldsymbol{x}_k^{\mathsf{T}}] \boldsymbol{\Phi}_k^{\mathsf{T}} - \underbrace{\Big(\mathrm{E}\Big[\boldsymbol{x}_{k+1} \boldsymbol{x}_{k+1}^{\mathsf{T}} \Big] - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1\big|\bar{N}} \Big)}_{\overset{(D.57)}{=} \mathrm{E}\left[\hat{\boldsymbol{x}}_{k+1\big|\bar{N}} \hat{\boldsymbol{x}}^{\mathsf{T}}_{k+1\big|\bar{N}} \right]} \Bigg) \mathbf{M}_k^{\mathsf{T}} \\
\overset{(D.58)}{=}& \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k \Big(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1\big|\bar{N}} - \boldsymbol{\Phi}_k \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} \boldsymbol{\Phi}_k^{\mathsf{T}} - \mathbf{F}_k \mathbf{Q}_k \mathbf{F}_k^{\mathsf{T}} \Big) \mathbf{M}_k^{\mathsf{T}} \\
\Rightarrow \mathbf{P}^{\tilde{\mathbf{x}}}_{k\big|\bar{N}} =& \mathbf{P}^{\tilde{\mathbf{x}}}_{k|k} + \mathbf{M}_k \Big(\mathbf{P}^{\tilde{\mathbf{x}}}_{k+1\big|\bar{N}} - \mathbf{P}^{\tilde{\mathbf{x}}}_{k+1|k} \Big) \mathbf{M}_k^{\mathsf{T}} \qquad \text{(D.59)}
\end{aligned}
$$

Appendix E

Cost Function Derivatives

All of the following derivatives have been double-checked using imaginary step finite differences.

E.1 Gradient of Modified Maximum Likelihood Cost Function

A slightly modified version of the following derivation has already been published in [GGBH2016]. The basis for the computation of the gradient of $\hat{J}$ is a result from matrix calculus [BS2012]

$$\frac{\partial \ln|\mathbf{X}|}{\partial a} = \mathrm{tr}\left[\mathbf{X}^{-1}\frac{\partial \mathbf{X}}{\partial a}\right] \tag{E.1}$$

Then, the derivative of $\hat{J}$ w.r.t. the j-th element of the k-th sample of the model output vector is

$$\frac{\partial \hat{J}}{\partial [\boldsymbol{y}_k]_{(j)}} = \frac{\partial}{\partial [\boldsymbol{y}_k]_{(j)}}\left(\frac{N}{2}\ln\left|\hat{\boldsymbol{\mathcal{B}}}\right|\right) = \frac{N}{2}\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}\right] \tag{E.2}$$

With the definition of $\hat{\boldsymbol{\mathcal{B}}}$ in equation (3.6) and keeping in mind that the residuals are defined as $\boldsymbol{r}_k = \boldsymbol{z}_k - \boldsymbol{y}_k$ above partial derivative can be expressed as

$$\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}} = -\frac{1}{N}\left((\boldsymbol{z}_k - \boldsymbol{y}_k)\,\boldsymbol{e}_j^\intercal + \boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\right) \tag{E.3}$$

This can then be plugged back into the expression for $\frac{\partial \hat{J}}{\partial [\boldsymbol{y}_k]_{(j)}}$, which yields, after factoring out and using the trace's linearity

$$\frac{\partial \hat{J}}{\partial [\boldsymbol{y}_k]_{(j)}} = -\frac{1}{N}\frac{N}{2}\left(\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k)\,\boldsymbol{e}_j^\intercal\right] + \mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\right]\right) \tag{E.4}$$

Using the fact that $\mathrm{tr}[\mathbf{A}] = \mathrm{tr}[\mathbf{A}^\intercal]$, the cyclic property of the trace, and the symmetry of $\hat{\boldsymbol{\mathcal{B}}}$, it can be shown, that above two trace terms are equal

$$\mathrm{tr}\Big[\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k)\,\boldsymbol{e}_j^\intercal\Big] = \mathrm{tr}\Big[\boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\,\hat{\boldsymbol{\mathcal{B}}}^{-1}\Big] = \mathrm{tr}\Big[\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\Big] \tag{E.5}$$

$$\frac{\partial \hat{J}}{\partial [\boldsymbol{y}_k]_{(j)}} = -\mathrm{tr}\Big[\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\Big] \tag{E.6}$$

If $\hat{\boldsymbol{\mathcal{B}}}^{-1}$ is partitioned according to

$$\hat{\boldsymbol{\mathcal{B}}}^{-1} = \begin{bmatrix} \boldsymbol{b}_1^\intercal \\ \vdots \\ \boldsymbol{b}_{n_y}^\intercal \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_{n_y} \end{bmatrix} \qquad \boldsymbol{b}_j \in \mathbb{R}^{n_y} \tag{E.7}$$

above expression can be simplified (again, using the cyclic property of the trace and its invariance w.r.t. transposing the argument)

$$\begin{aligned} \frac{\partial \hat{J}}{\partial [\boldsymbol{y}_k]_{(j)}} &= -\mathrm{tr}\Big[\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal\Big] = -\mathrm{tr}[\boldsymbol{b}_j(\boldsymbol{z}_k - \boldsymbol{y}_k)^\intercal] = \\ &= -\mathrm{tr}\Big[\boldsymbol{b}_j^\intercal(\boldsymbol{z}_k - \boldsymbol{y}_k)\Big] = -\boldsymbol{b}_j^\intercal(\boldsymbol{z}_k - \boldsymbol{y}_k) \end{aligned} \tag{E.8}$$

Finally, all derivatives w.r.t. the j output vector elements can be stacked vertically, to obtain

$$\frac{\partial \hat{J}}{\partial \boldsymbol{y}_k} = \frac{\partial \ln\big|\hat{\boldsymbol{\mathcal{B}}}\big|}{\partial \boldsymbol{y}_k} = -\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k) \tag{E.9}$$

The same results was obtained in [GP1977, Ch. 5.4].

E.2 Gradient of Modified Maximum Likelihood Cost Function for Scaled Residual Covariance

In the case of a scaled residual covariance matrix, the derivatives have to be adapted, too

$$\begin{aligned} \frac{\partial\, {}_{scaled}J}{\partial \boldsymbol{y}_k} &= \frac{\partial}{\partial \boldsymbol{y}_k}\left(\frac{1}{2}\sum_{k=0}^{N-1} \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k + \frac{N}{2}\ln\big|\hat{\boldsymbol{\mathcal{B}}}\big| + N\ln|\mathbf{W}|\right) \\ &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{y}_k}\left(\sum_{k=0}^{N-1} \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k\right) + \frac{\partial}{\partial \boldsymbol{y}_k}\frac{N}{2}\ln\big|\hat{\boldsymbol{\mathcal{B}}}\big| \end{aligned} \tag{E.10}$$

The second term can be identified as the derivative of $\hat{J}$ w.r.t. the model outputs

$$\frac{\partial}{\partial \boldsymbol{y}_k}\frac{N}{2}\ln\big|\hat{\boldsymbol{\mathcal{B}}}\big| = \frac{\partial \hat{J}}{\partial \boldsymbol{y}_k} = -\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k - \boldsymbol{y}_k) \tag{E.11}$$

For the first, a similar approach as above is used. Some identities from matrix calculus are important here, namely the derivative of a quadratic form, the chain rule [BS2012], and the derivative of the trace of an inverse

$$\frac{\partial \boldsymbol{x}^\intercal \mathbf{B} \boldsymbol{x}}{\partial \boldsymbol{x}} = \left(\mathbf{B} + \mathbf{B}^\intercal\right) \boldsymbol{x} \tag{E.12}$$

$$\frac{\partial g(\mathbf{U}(\mathbf{X}))}{\partial [\mathbf{X}]_{(i,j)}} = \mathrm{tr}\left[\left(\frac{\partial g(\mathbf{U}(\mathbf{X}))}{\partial \mathbf{U}}\right)^\intercal \frac{\partial \mathbf{U}}{\partial [\mathbf{X}]_{(i,j)}}\right] \tag{E.13}$$

$$\frac{\partial \mathrm{tr}\left[\mathbf{A}\mathbf{X}^{-1}\mathbf{B}\right]}{\partial \mathbf{X}} = -\mathbf{X}^{-\intercal}\mathbf{A}^\intercal\mathbf{B}^\intercal\mathbf{X}^{-\intercal} \tag{E.14}$$

Then, applying the product rule, an expression for the derivative w.r.t. the j-th element of the k-th sample of the model output vector can be determined

$$\begin{aligned}
&\frac{1}{2}\frac{\partial}{\partial [\boldsymbol{y}_k]_{(j)}}\left(\sum_{k=0}^{N-1} \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k\right) \\
&\overset{\text{(E.13)}}{\underset{\text{(E.14)}}{=}} -\boldsymbol{e}_j^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k + \frac{1}{2}\mathrm{tr}\left[\left(\frac{\partial}{\partial \hat{\boldsymbol{\mathcal{B}}}}\mathrm{tr}\left[\sum_{k=0}^{N-1} \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k\right]\right)^\intercal \frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}\right] \\
&\overset{\text{(E.14)}}{=} -\boldsymbol{e}_j^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k - \frac{1}{2}\mathrm{tr}\left[\left(\sum_{k=0}^{N-1} \hat{\boldsymbol{\mathcal{B}}}^{-\intercal}\mathbf{W}^{-1}\boldsymbol{r}_k\boldsymbol{r}_k^\intercal\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-\intercal}\right)^\intercal \frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}\right] \\
&= -\boldsymbol{e}_j^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k - \frac{1}{2}\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}N\frac{1}{N}\left(\sum_{k=0}^{N-1} \boldsymbol{r}_k\boldsymbol{r}_k^\intercal\right)\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}\right] \\
&\overset{(3.6)}{=} -\boldsymbol{e}_j^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k - \frac{N}{2}\mathrm{tr}\left[\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}\right]
\end{aligned} \tag{E.15}$$

The second part has now the same structure as equation (E.2), i.e. the same strategy can be applied: insert the expression for $\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_k]_{(j)}}$, combine the two parts of the resulting trace, partition the matrix $\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}$ and finally stack the derivatives vertically to arrive at

$$\frac{1}{2}\frac{\partial}{\partial \boldsymbol{y}_k}\left(\sum_{k=0}^{N-1} \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\boldsymbol{r}_k\right) = \left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\boldsymbol{r}_k \tag{E.16}$$

The complete cost function derivative then becomes

$$\frac{\partial\, {}_{scaled}J}{\partial \boldsymbol{y}_k} = \left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\boldsymbol{r}_k \tag{E.17}$$

E.3 Hessian of Maximum Likelihood Cost Function

In order to derive the second order derivatives of the cost function with covariance estimation, at first derivatives with respect to the j-th component of the output vector

at time instant l is considered

$$\begin{aligned}\frac{\partial}{\partial[\boldsymbol{y}_l]_{(j)}}\frac{\partial J}{\partial \boldsymbol{y}_k} &= \frac{\partial}{\partial[\boldsymbol{y}_l]_{(j)}}\left(-\hat{\boldsymbol{\mathcal{B}}}^{-1}(\boldsymbol{z}_k-\boldsymbol{y}_k)\right)\\ &= -\frac{\partial\hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial[\boldsymbol{y}_l]_{(j)}}(\boldsymbol{z}_k-\boldsymbol{y}_k)+\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j\delta_{kl}\overset{(E.7)}{=}-\frac{\partial\hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial[\boldsymbol{y}_l]_{(j)}}\boldsymbol{r}_k+\boldsymbol{b}_j\delta_{kl}\end{aligned} \tag{E.18}$$

In order to find an expression for the first term, another result from matrix algebra is used [BS2012]

$$\frac{\partial\mathbf{X}^{-1}}{\partial a}=-\mathbf{X}^{-1}\frac{\partial\mathbf{X}}{\partial a}\mathbf{X}^{-1} \tag{E.19}$$

Applied to the problem at hand, this yields firstly

$$\begin{aligned}-\frac{\partial\hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial[\boldsymbol{y}_l]_{(j)}}\boldsymbol{r}_k =&\hat{\boldsymbol{\mathcal{B}}}^{-1}\frac{\partial\hat{\boldsymbol{\mathcal{B}}}}{\partial[\boldsymbol{y}_l]_{(j)}}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\\ \overset{(E.3)}{=}&-\hat{\boldsymbol{\mathcal{B}}}^{-1}\frac{1}{N}\left(\boldsymbol{r}_l\boldsymbol{e}_j^\intercal+\boldsymbol{e}_j\boldsymbol{r}_l^\intercal\right)\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\\ =&-\frac{1}{N}\Bigg(\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\underbrace{\boldsymbol{e}_j^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k}_{=\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j\in\mathbb{R}}+\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\Bigg)\\ =&-\frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j+\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{e}_j\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\right)\\ \overset{(E.7)}{=}&-\frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\boldsymbol{b}_j+\boldsymbol{b}_j\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\right)\end{aligned} \tag{E.20}$$

Now, the complete matrix of second derivatives can be assembled, by writing the elements next to each other

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{y}_l}\frac{\partial J}{\partial\boldsymbol{y}_k} &= \begin{bmatrix}\frac{\partial}{\partial[\boldsymbol{y}_l]_{(1)}}\frac{\partial J}{\partial\boldsymbol{y}_k} & \cdots & \frac{\partial}{\partial[\boldsymbol{y}_l]_{(n_y)}}\frac{\partial J}{\partial\boldsymbol{y}_k}\end{bmatrix}\\ &= -\frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l\boldsymbol{r}_k^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}+\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_k\right)+\hat{\boldsymbol{\mathcal{B}}}^{-1}\delta_{kj}\end{aligned} \tag{E.21}$$

With large N the second term dominates, thus the following approximation is valid in practice

$$\frac{\partial}{\partial\boldsymbol{y}_l}\frac{\partial J}{\partial\boldsymbol{y}_k}\approx\hat{\boldsymbol{\mathcal{B}}}^{-1}\delta_{kj} \tag{E.22}$$

E.4 Hessian of Maximum Likelihood Cost Function for Scaled Residual Covariance

In the case of a scaled residual covariance matrix, the Hessian has to be adapted in a similar manner as the gradient before, it then reads

$$\frac{\partial}{\partial\boldsymbol{y}_l}\frac{\partial_{\,scaled}J}{\partial\boldsymbol{y}_k}=\frac{\partial}{\partial\boldsymbol{y}_l}\left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}+\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}-\hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\boldsymbol{r}_k \tag{E.23}$$

A similar approach as before is used, i.e. first the derivative with respect to the j-th element at time instant l is computed, before the results are stacked to obtain the derivative $\frac{\partial^2 {}_{scaled}J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\intercal}$. In order to do so, two auxiliary functions $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are defined as follows

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_{n_y} \end{bmatrix} \quad \mathbf{A} \in \mathbb{R}^{n_{n_y} \times n_{n_y}} \tag{E.24}$$

$$\mathbf{C}^\intercal = \begin{bmatrix} \boldsymbol{c}_1 & \cdots & \boldsymbol{c}_{n_y} \end{bmatrix} \quad \mathbf{C} \in \mathbb{R}^{n_{n_y} \times n_{n_y}} \tag{E.25}$$

$$\begin{aligned}
[\boldsymbol{f}_1]_{(j)}(\mathbf{A}, \mathbf{C}) &= \mathbf{A} \cdot \frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_l]_{(j)}} \cdot \mathbf{C}\boldsymbol{r}_k \\
&\overset{\text{(E.3)}}{=} -\frac{1}{N}\mathbf{A}\left(\boldsymbol{r}_l \boldsymbol{e}_j^\intercal + \boldsymbol{e}_j \boldsymbol{r}_l^\intercal\right)\mathbf{C}\boldsymbol{r}_k \\
&= -\frac{1}{N}\Bigg(\mathbf{A}\boldsymbol{r}_l \underbrace{\boldsymbol{e}_j^\intercal \mathbf{C}\boldsymbol{r}_k}_{=\boldsymbol{r}_k^\intercal \mathbf{C}^\intercal \boldsymbol{e}_j \in \mathbb{R}} + \mathbf{A}\boldsymbol{e}_j \boldsymbol{r}_l^\intercal \mathbf{C}\boldsymbol{r}_k\Bigg) \\
&= -\frac{1}{N}(\mathbf{A}\boldsymbol{r}_l \boldsymbol{r}_k^\intercal \mathbf{C}^\intercal \boldsymbol{e}_j + \mathbf{A}\boldsymbol{e}_j \boldsymbol{r}_l^\intercal \mathbf{C}\boldsymbol{r}_k) \\
&= -\frac{1}{N}(\mathbf{A}\boldsymbol{r}_l \boldsymbol{r}_k^\intercal \boldsymbol{c}_j + \boldsymbol{a}_j \boldsymbol{r}_l^\intercal \mathbf{C}\boldsymbol{r}_k)
\end{aligned} \tag{E.26}$$

$$\begin{aligned}
\boldsymbol{f}_1(\mathbf{A}, \mathbf{C}) &= \begin{bmatrix} [\boldsymbol{f}_1]_{(1)} & \cdots & [\boldsymbol{f}_1]_{(n_y)} \end{bmatrix} \\
&= -\frac{1}{N}(\mathbf{A}\boldsymbol{r}_l \boldsymbol{r}_k^\intercal \mathbf{C}^\intercal + \mathbf{A}\boldsymbol{r}_l^\intercal \mathbf{C}\boldsymbol{r}_k)
\end{aligned} \tag{E.27}$$

$$\begin{aligned}
[\boldsymbol{f}_2]_{(j)}(\mathbf{A}, \mathbf{C}) &= \mathbf{A} \cdot \frac{\partial \hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial [\boldsymbol{y}_l]_{(j)}} \cdot \mathbf{C}\boldsymbol{r}_k \\
&\overset{\text{(E.19)}}{=} -\mathbf{A}\hat{\boldsymbol{\mathcal{B}}}^{-1} \cdot \frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_l]_{(j)}} \cdot \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{C}\boldsymbol{r}_k \\
&= [\boldsymbol{f}_1]_{(j)}\left(-\mathbf{A}\hat{\boldsymbol{\mathcal{B}}}^{-1}, \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{C}\right)
\end{aligned} \tag{E.28}$$

$$\begin{aligned}
\boldsymbol{f}_2(\mathbf{A}, \mathbf{C}) &= \begin{bmatrix} [\boldsymbol{f}_2]_{(1)} & \cdots & [\boldsymbol{f}_2]_{(n_y)} \end{bmatrix} \\
&= \frac{1}{N}\left(\mathbf{A}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l \boldsymbol{r}_k^\intercal \mathbf{C}^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} + \mathbf{A}\hat{\boldsymbol{\mathcal{B}}}^{-1}\boldsymbol{r}_l^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{C}\boldsymbol{r}_k\right)
\end{aligned} \tag{E.29}$$

Now, computing the partial derivative with respect to $[\boldsymbol{y}_l]_{(j)}$ of the scaled gradient using the product rule yields

$$\begin{aligned}
\frac{\partial^2 {}_{scaled}J}{\partial \boldsymbol{y}_k \partial [\boldsymbol{y}_l]_{(j)}^\intercal} &= \frac{\partial}{\partial [\boldsymbol{y}_l]_{(j)}}\left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)\boldsymbol{r}_k \\
&= \Bigg(-\mathbf{W}^{-\intercal}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial [\boldsymbol{y}_l]_{(j)}}\mathbf{W}^{-1} + \frac{\partial \hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial [\boldsymbol{y}_l]_{(j)}}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} \\
&\quad + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}}{\partial [\boldsymbol{y}_l]_{(j)}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\frac{\partial \hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial [\boldsymbol{y}_l]_{(j)}} - \frac{\partial \hat{\boldsymbol{\mathcal{B}}}^{-1}}{\partial [\boldsymbol{y}_l]_{(j)}}\Bigg)\boldsymbol{r}_k \\
&\quad + \left(-\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\mathbf{W}^{-1}\hat{\boldsymbol{\mathcal{B}}}\mathbf{W}^{-\intercal}\hat{\boldsymbol{\mathcal{B}}}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1}\right)(-\boldsymbol{e}_j)\,\delta_{kl}
\end{aligned} \tag{E.30}$$

This can now be expressed using above auxiliary function definitions

$$\begin{aligned}\frac{\partial^2 {}_{scaled}J}{\partial \boldsymbol{y}_k \partial [\boldsymbol{y}_l]_{(j)}^\intercal} =& [\boldsymbol{f}_2]_{(j)} \left(-\mathbf{W}^{-\intercal}, \mathbf{W}^{-1}\right) + [\boldsymbol{f}_2]_{(j)} \left(\mathbf{I}_{n_{\boldsymbol{y}}}, \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1}\right) \\ &+ [\boldsymbol{f}_1]_{(j)} \left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1}, \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1}\right) + [\boldsymbol{f}_2]_{(j)} \left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal}, \mathbf{I}_{n_{\boldsymbol{y}}}\right) \\ &+ [\boldsymbol{f}_2]_{(j)} \left(-\mathbf{I}_{n_{\boldsymbol{y}}}, \mathbf{I}_{n_{\boldsymbol{y}}}\right) \\ &+ \left(\mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\right) \boldsymbol{e}_j \delta_{kl}\end{aligned} \tag{E.31}$$

Sorting all $j = 1, \ldots, n_{\boldsymbol{y}}$ elements next to each other, yields eventually

$$\begin{aligned}\frac{\partial^2 {}_{scaled}J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\intercal} =& -\frac{1}{N}\left(\mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l \boldsymbol{r}_k^\intercal \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} + \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \boldsymbol{r}_k\right) \\ &+ \frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l \boldsymbol{r}_k^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_k\right) \\ &- \frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \boldsymbol{r}_l \boldsymbol{r}_k^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \boldsymbol{r}_l^\intercal \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_k\right) \\ &+ \frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l \boldsymbol{r}_k^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_k\right) \\ &- \frac{1}{N}\left(\hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l \boldsymbol{r}_k^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_l^\intercal \hat{\boldsymbol{\mathcal{B}}}^{-1} \boldsymbol{r}_k\right) \\ &+ \left(\mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\right) \delta_{kl}\end{aligned} \tag{E.32}$$

Using the same argument as in the last section, this can be approximated as

$$\frac{\partial^2 {}_{scaled}J}{\partial \boldsymbol{y}_k \partial \boldsymbol{y}_l^\intercal} \approx \left(\mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} - \hat{\boldsymbol{\mathcal{B}}}^{-1} \mathbf{W}^{-1} \hat{\boldsymbol{\mathcal{B}}} \mathbf{W}^{-\intercal} \hat{\boldsymbol{\mathcal{B}}}^{-1} + \hat{\boldsymbol{\mathcal{B}}}^{-1}\right) \delta_{kl} \tag{E.33}$$

Appendix F

Flight Mechanics Nomenclature

F.1 Reference Frames and Transformations

Reference frames and the transformations between them are done using standard approaches in flight mechanics. To familiarize the reader with the nomenclature in use, they are briefly summed up next, further details can be found in [Hol2018b, Hol2018a].

The reference frames in use are all right-handed with mutually orthogonal axes, where the World Geodetic System 1984 (WGS84) 84 coordinates are the only exception.

- *Earth-Centered-Earth-Fixed [ECEF]* - index E
  Cartesian frame for determining positions; fixed at the earth's center with z-axis towards the north pole, x-axis in the equatorial plane towards the Greenwich Meridian, and y-axis to complement the right hand system; For system identification purposes, this is considered to be the "inertial" frame, its movement is thus neglected in formulating equations of motion
- *North-East-Down [NED]* - index O
  Cartesian frame for determining the aircraft's attitude; fixed at the aircraft reference point, with z-axis pointing down, perpendicular to the local geoid surface, x-axis pointing north and y-axis pointing east, both parallel to the local geoid surface;
- *Navigation Frame* - index N
  Local, Cartesian navigation frame, fixed at an arbitrary point; axis follow North-East-Down (NED) convention at the origin
- *Body-Fixed* - index B
  Fixed at the aircraft reference point R, used to denote locations of subsystems relative to the aircraft; x-axis points toward the nose, z-axis points downward in the aircraft's plane of symmetry, and y-axis completes the right hand system by pointing toward the starboard wing.
- *Aerodynamic* - index A
  System do denote aerodynamic quantities; x axis is aligned with the aerodynamic

velocity vector, z-axis points downward in the plane of symmetry of the aircraft, y-axis points to the right and completes the right hand system

- *World Geodetic System 1984 [WGS]*
  Coordinate system to represent a position on the WGS84 reference ellipsoid using latitude μ, longitude λ and altitude h over the reference ellipsoids' surface.

Transformations between the reference frames are achieved by using orthonormal rotation matrices $\mathbf{M}_{AB}$, which indicate a rotation from B-frame to the A-frame. Most of the rotation matrices can be parameterized using angles, and the following elementary rotation matrices about the three coordinate axis

$$\mathbf{M}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \tag{F.1}$$

$$\mathbf{M}_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{F.2}$$

$$\mathbf{M}_z(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{F.3}$$

The orthonormal transformation matrices between the systems can then be defined via a (non-commuting!) rotation sequence.

F.2 Nomenclature for Flight Mechanic Quantities

The general form for denoting flight mechanic quantities at the institute of flight system dynamics follows the following convention

$$\left(\text{Quantity}_{\text{Type}}^{\text{Reference}}\right)_{\text{Notation Frame}}^{\text{Derivative Frame}} \tag{F.4}$$

Three dimensional, Cartesian vectors are noted as $\vec{\boldsymbol{v}}$ as opposed to general vectors $\boldsymbol{x}$. The derivative and notation frames use the indices as illustrated in the last section, whereas the reference can be one of the following special points:

- G center of gravity
- R aircraft reference point
- A aerodynamic reference point
- P propulsion reference point

The "Type" of a force or moment quantity can be described, using those same indices (G gravity, A aerodynamic, ...), whereas for movement quantities the three possible types are

- K kinematic, i.e. with respect to the earth's surface

- A aerodynamic, i.e. with respect to the surrounding air
- W wind, i.e. the movement of the surrounding air with respect to the earth's surface

The same principles apply to the components of vector quantities. These components are

- $\vec{\boldsymbol{r}} = \begin{bmatrix} x, y, z \end{bmatrix}^\intercal$ for positions
- $\vec{\boldsymbol{v}} = \begin{bmatrix} u, u, w \end{bmatrix}^\intercal$ for velocities
- $\vec{\boldsymbol{a}} = \begin{bmatrix} a_x, a_y, a_z \end{bmatrix}^\intercal$ for accelerations
- $\vec{\boldsymbol{\omega}} = \begin{bmatrix} \omega_x, \omega_y, \omega_z \end{bmatrix}^\intercal$ for rotational rates
- $\vec{\boldsymbol{\omega}}^{OB} = \begin{bmatrix} p^{OB}, q^{OB}, r^{OB} \end{bmatrix}^\intercal$ for the special case of the rotational rate w.r.t. the earth's surface
- $\dot{\vec{\boldsymbol{\omega}}} = \begin{bmatrix} \dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z \end{bmatrix}^\intercal$ for rotational rates

F.2.1 Translational Examples

For positions, the reference can either be single, implying a position vector from the origin of the coordinate system, or double, implying a position vector between two points: $\left(\vec{\boldsymbol{r}}^{R}\right)_{E}$ is the absolute position of the aircraft reference point, noted in Earth-Centered-Earth-Fixed (ECEF) coordinates; $\left(\vec{\boldsymbol{r}}^{RG}\right)_{B}$ is the position of the center of gravity (c.g.) relative to the aircraft reference point, noted in body-fixed coordinates.

Velocities, as time derivative of positions, need to indicate, in which coordinate frame the time derivative is taken, and what type of velocity is meant: $\left(\vec{\boldsymbol{v}}_K^R\right)_B^E$ is the kinematic velocity of the aircraft reference point with respect to the ECEF frame, noted in body fixed coordinates; $\left(\vec{\boldsymbol{v}}_A^A\right)_A^E$ is the aerodynamic velocity at the aerodynamic reference point with respect to the ECEF frame, noted in the aerodynamic frame;

The same reasoning applies to accelerations: $\left(\vec{\boldsymbol{a}}_K^R\right)_B^{EE}$ is the kinematic acceleration at the aircraft reference point, differentiated twice with respect to the ECEF frame, noted in body fixed coordinates;

F.2.2 Rotational Examples

Angles usually describe the relation between two reference frames, i.e. some different rules apply: α_A^A indicates the aerodynamic angle of attack (angle between aerodynamic and body-fixed frame) at the aerodynamic reference point; ϕ^{BO} is the bank angle of the aircraft (which is independent of the reference point for a rigid body), i.e. one of the angles between the NED (O) and body-fixed frame (B).

Rotational rates describe the rotation between two reference frames, and need to indicate their type and the notation frame: $\left(\vec{\boldsymbol{\omega}}_K^{EB}\right)_B$ is the kinematic rotational rate of

the body-fixed frame versus the ECEF frame, noted in body-fixed coordinates; $\left(\vec{\omega}_A^{OA}\right)_O$ is the aerodynamic rotational rate of the aerodynamic frame, noted in components of the NED frame, which may be used to describe turbulence.

Rotational accelerations need to additionally indicate the derivative frame: $\left(\vec{\omega}_K^{EB}\right)_B^B$ are the kinematic rotational accelerations, where differentiation took place with respect to the body-fixed frame, and it is expressed in body-fixed coordinates. The frame, in which the derivative is taken is only of secondary importance here, since it is any of the two frames involved in the rotation, the result is the same

$$\begin{aligned}\left(\frac{d}{dt}\right)^A \left(\vec{\omega}^{AB}\right)_B &= \left(\dot{\vec{\omega}}^{AB}\right)_B^A = \left(\dot{\vec{\omega}}^{AB}\right)_B^B + \underbrace{\left(\vec{\omega}^{AB}\right)_B \times \left(\vec{\omega}^{AB}\right)_B}_{=\vec{\mathbf{0}}} \\ &= \left(\dot{\vec{\omega}}^{AB}\right)_B^B = \left(\frac{d}{dt}\right)^B \left(\vec{\omega}^{AB}\right)_B\end{aligned} \tag{F.5}$$

Appendix G

Additional Data Skymule

G.1 Complementary Filter for Attitude Estimation

The basic idea behind the complementary filter for attitude estimation is the following equality in the Euler angles $\boldsymbol{\Theta}$ in the Laplace domain

$$\boldsymbol{\Theta}^{BO} = \frac{Ts+1}{Ts+1}\boldsymbol{\Theta}^{BO} = \frac{T}{Ts+1}\dot{\boldsymbol{\Theta}}^{BO} + \frac{1}{Ts+1}\boldsymbol{\Theta}^{BO} \tag{G.1}$$

Now the left hand side can be considered as the filtered combination ${}^{f}\boldsymbol{\Theta}^{BO}$ of the two signal sources on the right $\dot{\boldsymbol{\Theta}}^{BO}$, $\boldsymbol{\Theta}^{BO}$. Solving for $s\,{}^{f}\boldsymbol{\Theta}^{BO} = {}^{f}\dot{\boldsymbol{\Theta}}^{BO}$ yields

$${}^{f}\dot{\boldsymbol{\Theta}}^{BO} = \frac{1}{T}\left(\boldsymbol{\Theta}^{BO} - {}^{f}\boldsymbol{\Theta}^{BO}\right) + \dot{\boldsymbol{\Theta}}^{BO} \tag{G.2}$$

which is a simple first order ordinary differential equation (ODE) with state vector ${}^{f}\boldsymbol{\Theta}^{BO}$. The remaining quantities on the right are an attitude estimate from inertial and magnetometer measurements

$$\boldsymbol{\Theta}^{BO} = \begin{bmatrix} \phi^{BO} \\ \theta^{BO} \\ \psi^{BO} \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(-\left(f_y^R\right)_B^{II}, -\left(f_z^R\right)_B^{II}\right) \\ \operatorname{atan2}\left(\left(f_x^R\right)_B^{II}, \sqrt{\left(f_y^R\right)_B^{II\,2} + \left(f_z^R\right)_B^{II\,2}}\right) \\ \psi_{mag}^{BO} \end{bmatrix} \tag{G.3}$$

The magnetic course measurement ψ_{mag}^{BO} arises from

$$\mathbf{M}_x\left(\phi^{BO}\right)\mathbf{M}_y\left(\theta^{BO}\right)\mathbf{M}_z\left(\psi^{BO}\right)\left(\vec{\boldsymbol{b}}^R\right)_O = \left(\vec{\boldsymbol{b}}^R\right)_B - \left(\Delta\vec{\boldsymbol{b}}^R\right)_B \tag{G.4}$$

$$\mathbf{M}_z\left(\psi^{BO}\right)\left(\vec{\boldsymbol{b}}^R\right)_O = \mathbf{M}_y\left(\theta^{BO}\right)^{\intercal}\mathbf{M}_x\left(\phi^{BO}\right)^{\intercal}\left(\left(\vec{\boldsymbol{b}}^R\right)_B - \left(\Delta\vec{\boldsymbol{b}}^R\right)_B\right) = \tilde{\vec{\boldsymbol{b}}}^R \tag{G.5}$$

where $\left(\Delta\vec{\boldsymbol{b}}^R\right)_B$ is a bias estimate to be illustrated further down, and $\tilde{\vec{\boldsymbol{b}}}^R$ is the measurement, transformed to a horizontal plane. The first two elements of above equation may

be re-arranged to finally obtain ψ_{mag}^{BO}

$$\begin{bmatrix} \cos\psi^{BO} & \sin\psi^{BO} \\ -\sin\psi^{BO} & \cos\psi^{BO} \end{bmatrix} \begin{bmatrix} \left(b_x^R\right)_O \\ \left(b_y^R\right)_O \end{bmatrix} = \begin{bmatrix} \left(b_x^R\right)_O & \left(b_y^R\right)_O \\ \left(b_y^R\right)_O & -\left(b_x^R\right)_O \end{bmatrix} \begin{bmatrix} \cos\psi^{BO} \\ \sin\psi^{BO} \end{bmatrix} = \begin{bmatrix} \tilde{b}_x^R \\ \tilde{b}_y^R \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \cos\psi^{BO} \\ \sin\psi^{BO} \end{bmatrix} = \begin{bmatrix} \left(b_x^R\right)_O & \left(b_y^R\right)_O \\ \left(b_y^R\right)_O & -\left(b_x^R\right)_O \end{bmatrix}^{-1} \begin{bmatrix} \tilde{b}_x^R \\ \tilde{b}_y^R \end{bmatrix} \tag{G.6}$$

$$\Rightarrow \psi_{mag}^{BO} = \operatorname{atan2}\left(\sin\psi^{BO}, \cos\psi^{BO}\right) \tag{G.7}$$

The magnetometer bias estimate $\left(\Delta\vec{\boldsymbol{b}}^R\right)_B$ is simply the low-pass filtered difference between measured and estimated magnetic field vector

$$\left(\Delta\dot{\vec{\boldsymbol{b}}}^R\right)_B = -\frac{1}{T_{\boldsymbol{b}}}\left(\Delta\vec{\boldsymbol{b}}^R\right)_B + \frac{1}{T_{\boldsymbol{b}}}\left(\left(\vec{\boldsymbol{b}}^R\right)_B - \mathbf{M}_{BO}\left({}^f\boldsymbol{\Theta}^{BO}\right)\left(\vec{\boldsymbol{b}}^R\right)_O\right) \tag{G.8}$$

with a comparably large time-constant $T_{\boldsymbol{b}}$. This augments the overall state vector to include this bias estimate.

The time derivative of the Euler angles in above filter equation (G.1) are obtained via the attitude propagation using rotational rate measurements

$$\dot{\boldsymbol{\Theta}}^{BO} = \begin{bmatrix} 1 & \sin\phi^{BO}\tan\theta^{BO} & \cos\phi^{BO}\tan\theta^{BO} \\ 0 & \cos\phi^{BO} & -\sin\phi^{BO} \\ 0 & \frac{\sin\phi^{BO}}{\cos\theta^{BO}} & \frac{\cos\phi^{BO}}{\cos\theta^{BO}} \end{bmatrix} \left(\vec{\boldsymbol{\omega}}_K^{OB}\right)_B \tag{G.9}$$

The only remaining part is to determine the time constant T: since the attitude estimates from the accelerometer are in general not very reliable, and are further distorted by centripetal, and Coriolis effects during turning flight, their "weight" should not be too large. This is why the time constant was chosen to be scaled based upon the norm of the measured acceleration vector

$$n_{\vec{\boldsymbol{f}}} = n_{\vec{\boldsymbol{f}}}\left(\left(\vec{\boldsymbol{f}}^R\right)_B^{II}\right) = \begin{cases} \frac{\left(\left\|\left(\vec{\boldsymbol{f}}^R\right)_B^{II}\right\| - g\right)^2}{g^2}(n_{max} - n_{min}) + n_{min} & \text{if } \left\|\left(\vec{\boldsymbol{f}}^R\right)_B^{II}\right\| < 2g \\ n_{max} & \text{otherwise} \end{cases} \tag{G.10}$$

$$T = 10^{n_{\vec{\boldsymbol{f}}}} \tag{G.11}$$

Above increases the exponent $n_{\vec{\boldsymbol{f}}}$ quadratically, the further away the measured acceleration is from its nominal value g, i.e. then the filter relies more heavily on the integrated rotational rates. In contrast, whenever $\left\|\left(\vec{\boldsymbol{f}}^R\right)_B^{II}\right\|$ is close to g, the exponent, and thus the time constant is decreased and more weight is put on the attitude estimates stemming from accelerometer measurements.

G.2 Additional Figures Skymule

Figures G.1 and G.2 show the off-axis responses, i.e. lateral motion for longitudinal maneuvers and vice versa

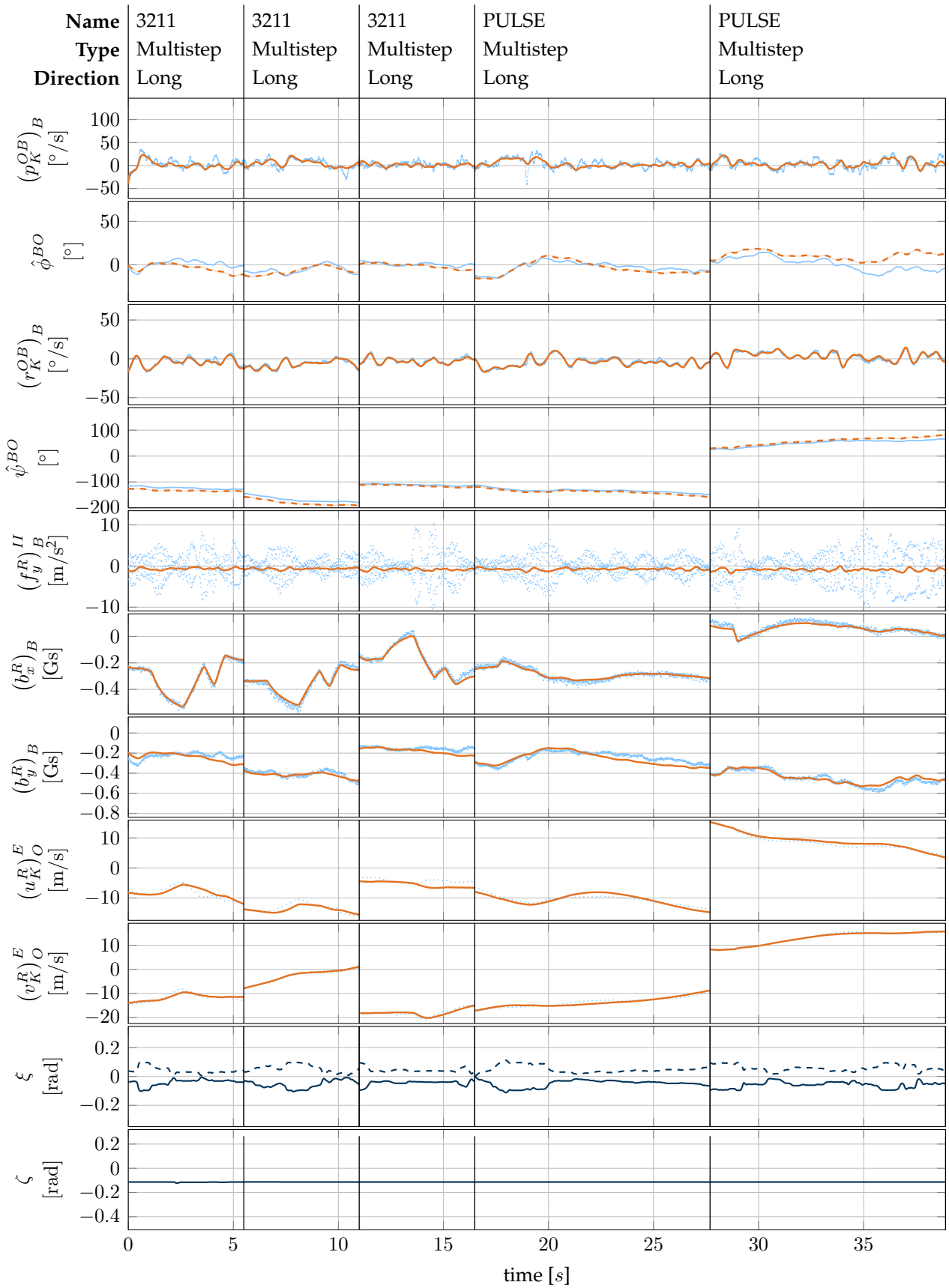


Figure G.1: *Lateral motion stemming from maneuvers in the longitudinal plane; measurements $\boldsymbol{z}$ (·), model outputs $\boldsymbol{y}$ (——), outputs not considered in the estimation (- - -), and control inputs $\boldsymbol{u}$ (- - -)*

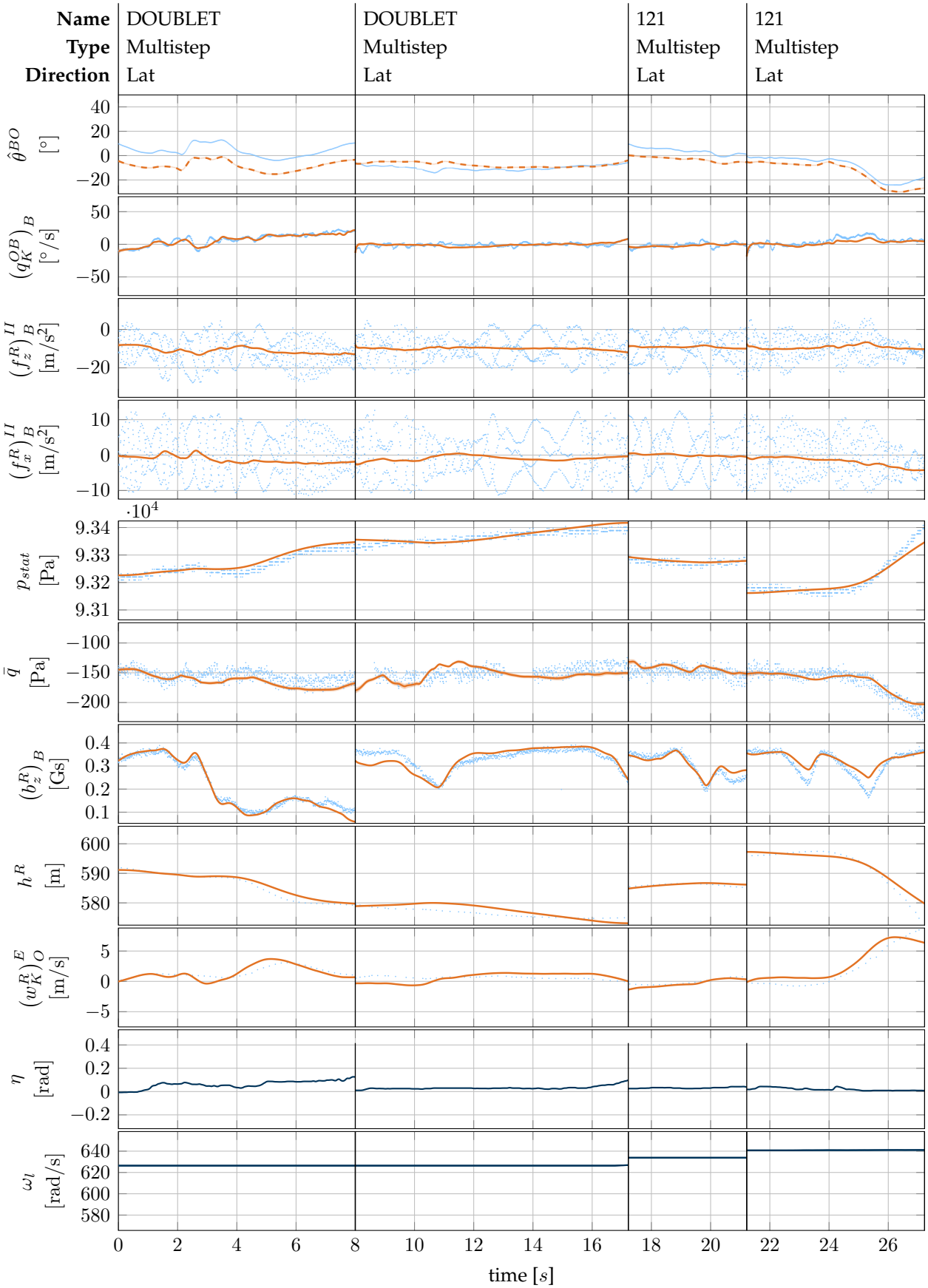


Figure G.2: *Longitudinal motion stemming from maneuvers in the lateral plane; measurements $\boldsymbol{z}$ (·), model outputs $\boldsymbol{y}$ (——), outputs not considered in the estimation (- - -), and control inputs $\boldsymbol{u}$ (——)*

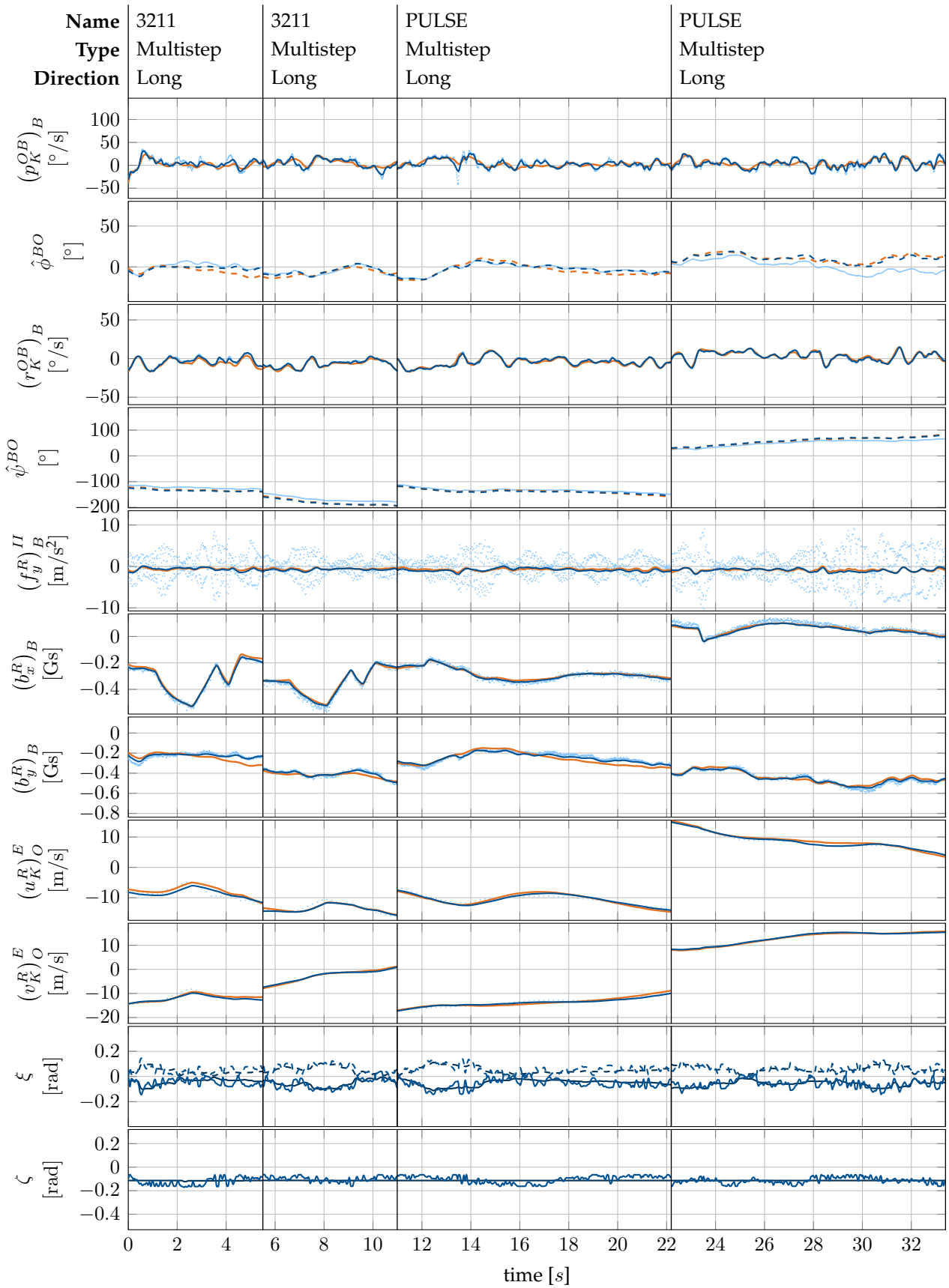


Figure G.3: *Inverse simulation results for lateral plane; measurements $\boldsymbol{z}$ (·), estimated model outputs $\boldsymbol{y}$ (considered —— / unconsidered - - -), and control inputs $\boldsymbol{u}$ (——); Corresponding inverse simulation results (——) are shown on top*

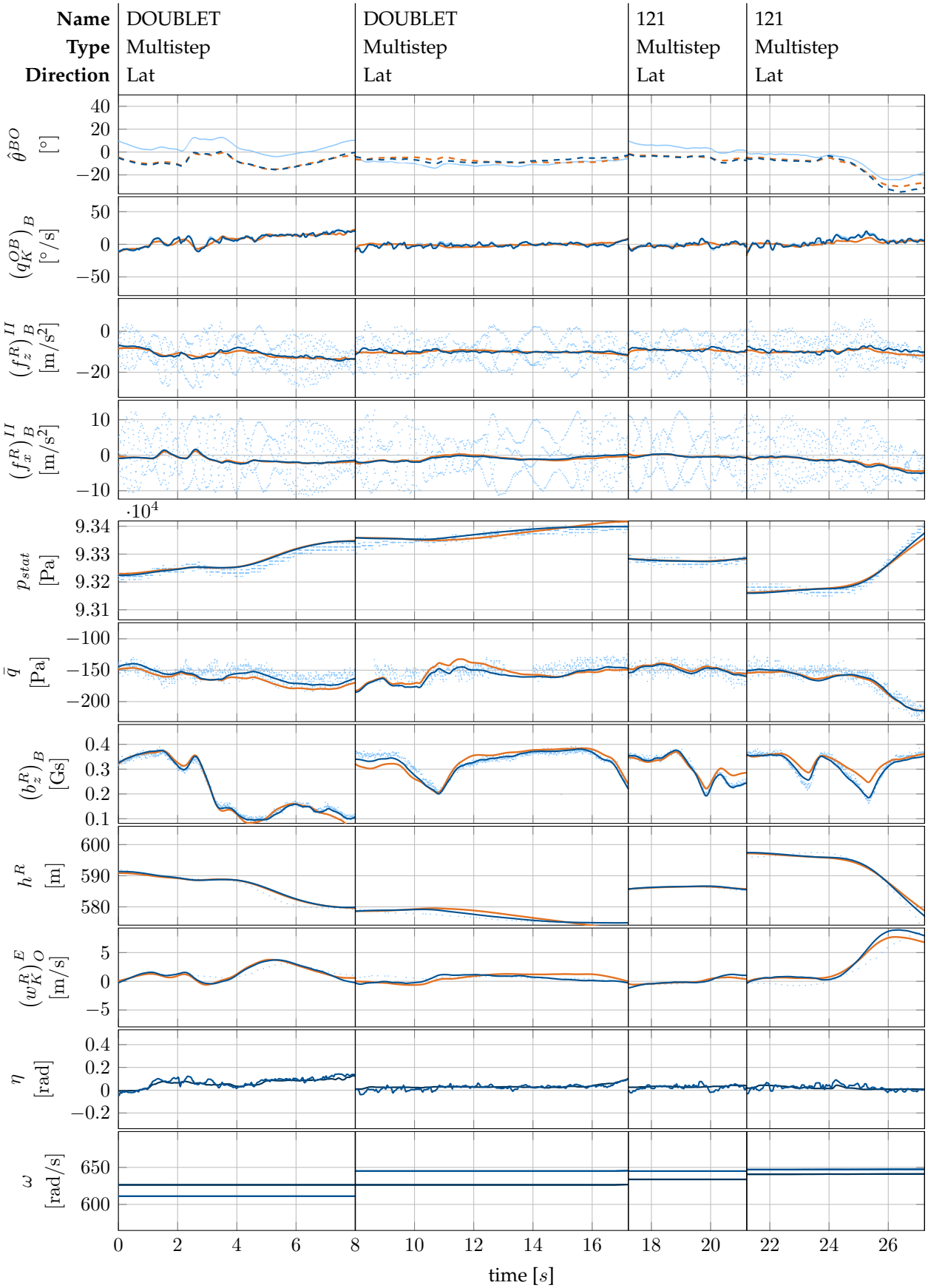


Figure G.4: *Inverse simulation results for longitudinal plane; measurements $\boldsymbol{z}$ (·), estimated model outputs $\boldsymbol{y}$ (considered —— / unconsidered - - -), and control inputs $\boldsymbol{u}$ (——); Corresponding inverse simulation results (——) are shown on top*

Publications

GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Flight Path Reconstruction for an Unmanned Aerial Vehicle Using Low-Cost Sensors. In: *ICAS 30th International Congress of the International Council of the Aeronautical Sciences*, 2016

GÖTTLICHER, Christoph; GNOTH, Marcus; BITTNER, Matthias; HOLZAPFEL, Florian: Aircraft Parameter Estimation Using Optimal Control Methods. In: *AIAA Atmospheric Flight Mechanics Conference*, 2016

HOSSEINI, Seyedbarzin; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Optimal input design for flight vehicle system identification via dynamic programming and the direct method for optimal control. In: *AIAA Scitech 2019 Forum*, 2019

KRAUSE, Christoph; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Development of a generic Flight Test Maneuver Injection Module. In: *ICAS 31st Congress of the International Council of the Aeronautical Science*, 2018

FANG, Xiang; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Attitude Estimation of Skis in Ski Jumping Using Low-Cost Inertial Measurement Units. In: *12th ISEA Conference on the Engineering of Sport*, 2018

DIEPOLDER, Johannes; GÖTTLICHER, Christoph; GRÜTER, Benedikt; AKMAN, Tuğba; HOLZAPFEL, Florian; BEN-ASHER, Joseph Z.: Optimal Control Based Flight Control Law Testing with Parameter Uncertainties. In: *IEEE Conference on Control Technology and Applications (CCTA)*, 2017

YU, Shanshan; WANG, Zhengjie; GOTTLICHER, Christoph: System identification for an unmanned aerial vehicle using the maximum likelihood method. In: *Proceedings of 2017 9th International Conference on Modelling, Identification and Control*, 2017

KRAUSE, Christoph; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Designing a low cost fixed wing flying testbed. In: *Aerospace Electronics and Remote Sensing Technology (ICARES)*, 2015

Bibliography

[Bab1864] BABBAGE, Charles: *The Life of a Philosopher*. London, United Kingdom: Longman, Green, Longman, Roberts, & Green, 1864

[Bac1982] BACH, Ralph E.: A Variational Technique for Smoothing Flight-Test and Accident Data. In: *Journal of Aircraft* 19 (1982), Nr. 7, p. 546–552. – ISSN 0021–8669

[BW1985] BACH, Ralph E.; WINGROVE, R. C.: Applications of state estimation in aircraft flight-data analysis. In: *Journal of Aircraft* 22 (1985), Nr. 7, p. 547–554. – ISSN 0021–8669

[Bar2016] BARTHEL, Tobias: *Designing a low-cost flight test platform*. Garching, Technische Universität München, Semester Thesis, 15.11.2016

[Bay1736] BAYES, Thomas: *An introduction to the doctrine of fluxions: and defence of the mathematicans against the objections of the author of the Analyst*. 1736

[Bay1763] BAYES, Thomas: An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. In: *Philosophical Transactions* 53 (1763), p. 370–418. – ISSN 0261–0523

[BA2010] BEN-ASHER, Joseph Z.: *Optimal control theory with aerospace applications*. Reston, Virginia, USA.: American Institute of Aeronautics and Astronautics, 2010 (AIAA education series). – ISBN 978–1–60086–732–3

[BIG2003] BEN-ISRAEL, Adi; GREVILLE, Thomas N. E.: *Generalized Inverses: Theory and Applications*. Second Edition. New York, NY, USA: Springer New York Inc., 2003 (CMS Books in Mathematics). – ISBN 9780387002934

[Ber1777] BERNOULLI, D.: The most probable choice between several discrepant observations and the formation therefrom of the most likely induction. In: *Biometrika* 48 (1961 (original 1777)), Nr. 1-2, p. 3–18. – ISSN 0006–3444

[Bet2010] BETTS, John T.: *Practical methods for optimal control and estimation using nonlinear programming*. 2nd ed. Philadelphia, PA, USA: Society for In-

dustrial and Applied Mathematics, 2010 (Advances in design and control). – ISBN 978–0–898716–88–7

[Bit2017] BITTNER, Matthias: *Utilization of Problem and Dynamic Characteristics for Solving Large Scale Optimal Control Problems*. Garching, Technische Universität München, Dissertation, 31.05.2017

[Boc1987] BOCK, Hans G.: *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*. Bonn, Universität Bonn, Dissertation, 1987

[Boo1991] BOOR, Carl de: *A practical guide to splines*. 5th print. New York, NY, USA: Springer, 1991 (Applied mathematical sciences 27). – ISBN 3–540–90356–9

[BLM2009a] BOTTASSO, Carlo; LURAGHI, Fabio; MAISANO, Giorgio: *Time-domain parameter estimation for first-principle rotorcraft models using recursive and batch procedures: Formulation and preliminary results*. Milano, Italy: Politecnico di Milano, 2009 (Scientific Report DIA-SR 09-05)

[BLM2009b] BOTTASSO, Carlo; LURAGHI, Fabio; MAISANO, Giorgio: A Unified Approach to Trajectory Optimization and Parameter Estimation in Vehicle Dynamics. In: *International Symposium on Coupled Methods in Numerical Dynamics*. Split, Croatia, 2009

[BM2009] BOTTASSO, Carlo; MAISANO, Giorgio: A New Single-Multiple Shooting Method for Trajectory Optimization of Complex Rotorcraft Models. In: *XX AIDAA National Conference*. Milano, Italy, 2009

[BLMM2010] BOTTASSO, Carlo L.; LURAGHI, Fabio; MAFFEZZOLI, Andrea; MAISANO, Giorgio: Parameter Estimation of Multibody Models of Unstable Systems From Experimental Data, With Application to Rotorcraft Vehicles. In: *Journal of Computational and Nonlinear Dynamics* 5 (2010), Nr. 3, p. 031010. – ISSN 15551423

[BM2009] BOTTASSO, Carlo L.; MAISANO, Giorgio: Efficient Rotorcraft Trajectory Optimization using Comprehensive Vehicle Models by Improved Shooting Methods. In: *35th European Rotorcraft Forum*. Hamburg, Germany, 2009

[BS2011] BRANDT, John; SELIG, Michael: Propeller Performance Data at Low Reynolds Numbers. In: *49th AIAA Aerospace Sciences Meeting*. Reston, VA, 2011

[Bra2005] BRANDT, John B.: *Small-Scale Propeller Performance at Low Speeds*. Urbana-Champaign, IL, USA, University of Illinois at Urbana-Champaign, M.Sc. Thesis, 2005

[BDAS2018] BRANDT, John B.; DETERS, Robert W.; ANANDA, Gavin K.; SELIG, Michael S.: *UIUC Propeller Data Site*. `http://m-selig.ae.illinois.edu/props/propDB.html`. Retrieved: June 8, 2018

[BS2012] BRANDT PETERSEN, Kaare; SYSKIND PEDERSEN, Michael: *The Matrix Cookbook*. `http://www2.imm.dtu.dk/pubdb/p.php?3274`. Retrieved: Jan. 2017 (Version Nov. 2012)

[BAL2010] BROCKHAUS, Rudolf; ALLES, Wolfgang; LUCKNER, Robert: *Flugregelung*. 3., neu bearbeitete Aufl. Berlin: Springer Berlin, 2010. – ISBN 3642014437

[BF1963] BRYSON, Arthur E.; FRAZIER, Malcom: Smoothing for linear and nonlinear dynamic Systems. In: *Proceedings of the Optimum System Synthesis Conference*, Aeronautical Systems Division, 1963, p. 353–364

[CB2001] CASELLA, George; BERGER, Roger L.: *Statistical Inference*. 2. edition. Belmont, CA, USA: Brooks/COle Cengage Learning, 2001. – ISBN 0534243126

[Cox1963] COX, Henry: *Estimation of State Variables for Noisy Dynamic Systems*. Boston, MA, USA, Massachusetts Institute of Technology, Dissertation, 1963

[Cox1964] COX, Henry: On the estimation of state variables and parameters for noisy dynamic systems. In: *IEEE Transactions on Automatic Control* 9 (1964), Nr. 1, p. 5–12. – ISSN 0018–9286

[Cra1946] CRAMÉR, Harald: *Mathematical Methods of Statistics*. Princeton, NJ, USA: Princeton University Press, 1946

[CJ2012] CRASSIDIS, John L.; JUNKINS, John L.: *Optimal estimation of dynamic systems*. 2nd ed. Boca Raton, FL, USA: CRC Press, 2012. – ISBN 1439839867

[CS2011] CZADO, Claudia; SCHMIDT, Thorsten: *Mathematische Statistik*. Berlin, Germany: Springer, 2011. – ISBN 978–3–642–17260–1

[Det2014] DETERS, Robert W.: *Performance and Slipstream Characteristic of Small-Scale Propellers at Low Reynolds Numbers*. Urbana, IL, USA, University of Illinois at Urbana-Champaign, Dissertation, 2014

[DAS2014] DETERS, Robert W.; ANANDA KRISHNAN, Gavin K.; SELIG, Michael S.: Reynolds Number Effects on the Performance of Small-Scale Propellers. In: *32nd AIAA Applied Aerodynamics Conference*. Reston, VA, USA, 2014. – ISBN 978–1–62410–288–2

[ER1996] ETKIN, Bernard; REID, Lloyd D.: *Dynamics of flight: Stability and control*. 3. ed. New York, NY, USA: Wiley, 1996. – ISBN 0471034185

[Eyk1987] EYKHOFF, P.: *System identification: Parameter and state estimation*. Chichester, United Kingdom: John Wiley and Sons, 1987. – ISBN 0–471–24980–7

[Fin1978] FINCK, R. D.: *USAF (United States Air Force) Stability and Control DATCOM (Data Compendium): Final rept. Sep 1975-Sep 1977*. St. Louis, MO, USA: McDonnell Aircraft Co., 1978

[Fis2011] FISCH, Florian: *Development of a framework for the solution of high-fidelity trajectory opitimization problems and bilevel optimal control problems*. Garching, Technische Universität München, Dissertation, 04.02.2011

[Fis1912] FISHER, R. A.: On an Absolute Criterion for Fitting Frequency Curves. In: *Messenger of Mathmatics* 41 (1912), p. 155–160

[Fis1922] FISHER, R. A.: On the Mathematical Foundations of Theoretical Statistics. In: *Philosophical Transactions of the Royal Society of London: Mathematical, Physical and Engineering Sciences* 222 (1922), Nr. 594-604, p. 309–368. – ISSN 0264–3952

[Fis1925] FISHER, R. A.: Theory of Statistical Estimation. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 22 (1925), Nr. 5, p. 700–725

[Föl2015] FÖLSING, Albrecht: *Albert Einstein: Eine Biographie*. 6. Aufl. Frankfurt am Main: Suhrkamp, 2015. – ISBN 978–3–518–38990–4

[FB1966] FRIEDLAND, Bernard; BERNSTEIN, Irwin: Estimation of the state of a nonlinear process in the presence of nongaussian noise and disturbances. In: *Journal of the Franklin Institute* 281 (1966), Nr. 6, p. 455–480. – ISSN 0016–0032

[Gau1857] GAUSS, C. F.: *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections: a translation of Gauss's 'Theoria motus.'*. Boston, Little, Brown and company, 1857

[Ger2017] GERDTS, M.: *Optimierung - Vorlesungsskript*. `https://www.unibw.de/lrt1/gerdts/lehre/optimierung.pdf`. Retrieved: 01/06/2017

[Ger2018] GERDTS, M.: *Optimale Steuerung - Vorlesungsskript*. `https://www.unibw.de/lrt1/gerdts/lehre/optimale_steuerung.pdf`. Retrieved: 19/03/2018

[Ger2012] GERDTS, M.: *Optimal Control of ODEs and DAEs*. 1st ed. Berlin, Germany and Boston, USA: de Gruyter, 2012. – ISBN 978–3–11–024999–6

[Ger1971] GERLACH, O. H.: *The determination of stability derivatives and performance characteristics from dynamic manoeuvres*. Delft, Netherlands: Delft University of Technology, 1971

[GMW1982] GILL, Philip E.; MURRAY, Walter; WRIGHT, Margaret H.: *Practical optimization*. Bingley, United Kingdom: Emerald Group Publishing Limited, 1982. – ISBN 978–0122839528

[GWMS2018] GILL, Philip E.; WONG, Elizabeth; MURRAY, Walter; SAUNDERS, Michael A.: *User's Guide for SNOPT Version 7.6: Software for Large-Scale Nonlinear Programming*. 13.05.2018

[Gla1919] GLAUERT, Hermann: Analysis of Phugoids Obtained by a Recording Airspeed Indicator. In: *Aeronautical Research Council, A.R.C.* R&M No. 576 (1919)

[Gno2016] GNOTH, Marcus: *Refinement of an Optimal Control Approach to System Identification and Application to a Model Aerobatic Aircraft*. Garching, Technische Universität München, MA, 13.05.2016

[GP1977] GOODWIN, Graham C.; PAYNE, Robert L.: *Dynamic system identification: Experiment design and data analysis*. 2nd ed. New York, NY, USA: Academic Press, 1977 (136). – ISBN 978–0–12–289750–4

[GH1990] GORMAN, J. D.; HERO, A. O.: Lower bounds for parametric estimation with constraints. In: *IEEE Transactions on Information Theory* 36 (1990), Nr. 6, p. 1285–1301. – ISSN 00189448

[GGBH2016] GÖTTLICHER, Christoph; GNOTH, Marcus; BITTNER, Matthias; HOLZAPFEL, Florian: Aircraft Parameter Estimation Using Optimal Control Methods. In: *AIAA Atmospheric Flight Mechanics Conference*, 2016

[GH2016] GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Flight Path Reconstruction for an Unmanned Aerial Vehicle Using Low-Cost Sensors. In: *ICAS 30th International Congress of the International Council of the Aeronautical Sciences*, 2016

[Gre1951] GREENBERG, Harry: *A survey of methods for determining stability parameters of an airplane from dynamic flight measurements*. NASA Ames Research Center, 1951 (NACA-TN-2220)

[Hal2008] HALD, A.: *A History of Parametric Statistical Inference from Bernoulli to Fischer, 1713-1935*. Dordrecht: Springer, 2008. – ISBN 978–0–387–46409–1

[HJ1996] HAMEL, Peter G.; JATEGAONKAR, Ravindra V.: Evolution of flight vehicle system identification. In: *Journal of Aircraft* 33 (1996), Nr. 1, p. 9–28. – ISSN 0021–8669

[HH2015] HÄRDLE, Wolfgang; HLÁVKA, Z.: *Multivariate statistics: Exercises and solutions*. 2nd edition. Heidelberg, Germany: Springer, 2015. – ISBN 9783642360046

[Ho1963] HO, Yu C.: On the stochastic approximation method and optimal filtering theory. In: *Journal of Mathematical Analysis and Applications* 6 (1963), Nr. 1, p. 152–154

[Hol2018a] HOLZAPFEL, Florian: *Flight System Dynamics II: Lecture Notes*. Department of Mechanical Engineering, Technical University of Munich, Summer Term 2018

[Hol2018b] HOLZAPFEL, Florian: *Flight System Dynamics I: Lecture Notes*. Department of Mechanical Engineering, Technical University of Munich, Winter Term 2017/2018

[HGH+2017] HOLZAPFEL, Florian; GÖTTLICHER, Christoph; HAGER, Stefan; SEMBIRING, Javensius; MERKL, Christian: *Systemidentifikation: Lecture Notes*. Munich, 2017

[HGH2019] HOSSEINI, Seyedbarzin; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Optimal input design for flight vehicle system identification via dynamic programming and the direct method for optimal control. In: *AIAA Scitech 2019 Forum*, 2019

[IM1977] ILIFF, K. W.; MAINE, R. E.: Further observations on maximum likelihood estimates of stability and control characteristics obtained from flight data. In: *AIAA Guidance and Control Conference*. Reston, VA, USA, 1977

[Ili1989] ILIFF, Kenneth W.: Parameter estimation for flight vehicles. In: *Journal of Guidance, Control, and Dynamics* 12 (1989), Nr. 5, p. 609–622. – ISSN 0731–5090

[IM1985] ILIFF, Kenneth W.; MAINE, Richard E.: *More than you may want to know about maximum likelihood estimation*. Edwards, CA, USA: NASA Ames Research Center, 1985 (NTM 85905)

[IM1986] ILIFF, Kenneth W.; MAINE, Richard E.: *Bibliography for aircraft parameter estimation*. Moffett Field, CA, USA: NASA Ames Research Center, 1986

[IP1969] ILIFF, Kenneth W.; POWERS, B.: A comparison of Newton-Raphson and other methods for determining stability derivatives from flight data. In: *3rd Flight Test, Simulation, and Support Conference*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 1969, p. 127

[Jat2004] JATEGAONKAR, Ravindra V.: *Journal of Aircraft - Special Section: Flight Vehice System ID: Part 1*. American Institute of Aeronautics and Astronautics, 2004

[Jat2005] JATEGAONKAR, Ravindra V.: *Journal of Aircraft - Special Section: Flight Vehicle System ID: Part 2*. American Institute of Aeronautics and Astronautics, 2005

[Jat2006] JATEGAONKAR, Ravindra V.: *Flight vehicle system identification: A time domain methodology*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2006. – ISBN 161583074X

[Jat2015] JATEGAONKAR, Ravindra V.: *Flight Vehicle System Identification: A time domain methodology*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2015. – ISBN 9781624102783

[JP1989] JATEGAONKAR, Ravindra V.; PLAETSCHKE, Ermin: Algorithms for aircraft parameter estimation accounting for process and measurement noise. In: *Journal of Aircraft* 26 (1989), Nr. 4, p. 360–372. – ISSN 0021–8669

[JP1990] JATEGAONKAR, Ravindra V.; PLAETSCHKE, Ermin: Identification of moderately nonlinear flight mechanics systems with additive process and measurement noise. In: *Journal of Guidance, Control, and Dynamics* 13 (1990), Nr. 2, p. 277–285. – ISSN 0731–5090

[Jaz1970] JAZWINSKI, Andrew H.: *Stochastic processes and filtering theory*. New York, NY, USA: Academic Press, 1970. – ISBN 9780080960906

[Kal1960] KALMAN, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Journal of Basic Engineering* 82 (1960), Nr. 1, p. 35. – ISSN 00219223

[KB1961] KALMAN, R. E.; BUCY, R. S.: New Results in Linear Filtering and Prediction Theory. In: *Journal of Basic Engineering* 83 (1961), Nr. 1, p. 95. – ISSN 00219223

[Kar1939] KARUSH, William: *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Chicago, IL, USA, University of Chicago, M.Sc. Thesis, 1939

[KML+2015] KAWAJIR, Yoshiaki; MARGOT, François; LAIRD, Carl; VIGERSKE, Stefan; WÄCHTER, Andreas: *Introduction to Ipopt: A tutorial for downloading, installing, and using Ipopt*. 2015

[Ken1961] KENDALL, M. G.: Studies in the history of probability and statistics: XI. Daniel Bernoulli on maximum likelihood. In: *Biometrika* 48 (1961), Nr. 1-2, p. 1–2. – ISSN 0006–3444

[KM2006] KLEIN, Vladislav; MORELLI, Eugene A.: *Aircraft System Identification: Theory and Practice*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2006 (AIAA education series). – ISBN 9781563478321

[KM2018] KLEIN, Vladislav; MORELLI, Eugene A.: *Aircraft System Identification - Theory and Practice: Errata*. `http://sunflyte.com/SIDBook_Errata_1e.pdf`. Retrieved: June 16, 2018

[KO1963] KOPP, Richard E.; ORFORD, Richard J.: Linear Regression Applied to System Identification for Adaptive Control Systems. In: *AIAA Journal* 1 (1963), Nr. 10, p. 2300–2306. – ISSN 0001–1452

[KGH2018] KRAUSE, Christoph; GÖTTLICHER, Christoph; HOLZAPFEL, Florian: Development of a generic Flight Test Maneuver Injection Module. In: *ICAS 31st Congress of the International Council of the Aeronautical Science*, 2018

[KT1951] KUHN, Harold W.; TUCKER, Albert W.: Nonlinear Programming. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, CA, USA, 1951, 481–492

[Lai1974] LAINIOTIS, D. G.: Estimation: A brief survey. In: *Information Sciences* 7 (1974), p. 191–202. – ISSN 0020–0255

[Lap1814] LAPLACE, Pierre-Simon: *Théorie Analytique des Probabilités*. Paris, France: Courcier, 1814

[Lei] LEITÃO, Miguel: *Performance Metrics for Adaptive Flight Control [working title, unpublished]*. Munich, Technical University of Munich, Dissertation

[Lev1944] LEVENBERG, Kenneth: A method for the solution of certain non-linear problems in least squares. In: *Quarterly of Applied Mathematics* 2 (1944), Nr. 2, p. 164–168. – ISSN 0033–569X

[Lju2009] LJUNG, Lennart: *System identification: Theory for the user*. 2nd Edition. Upper Saddle River, NJ, USA: Prentice-Hall, 2009. – ISBN 9780136566953

[Lju2011] LJUNG, Lennart: *System Identification Toolbox for use with MATLAB*. The MathWorks, Inc., 2011

[MI1981a] MAINE, R. E.; ILIFF, K. W.: Use of Cramer-Rao Bounds on Flight Data with Colored Residuals. In: *Journal of Guidance, Control, and Dynamics* 4 (1981), Nr. 2, p. 207–213. – ISSN 0731–5090

[MI1981b] MAINE, Richard E.; ILIFF, Kenneth W.: Formulation and implementation of a practical algorithm for parameter estimation with process and measurement noise. In: *SIAM Journal on Applied Mathematics* 41 (1981), Nr. 3, p. 558–579

[MI1986] MAINE, Richard E.; ILIFF, Kenneth W.: *Application of Parameter Estimation to Aircraft Stability and Control: The Output-error Approach*. NASA Scientific and Technical Information Branch, 1986

[MIM1985] MAINE, Richard E.; ILIFF, Kenneth W.; MAINE, R. E.: *Identification of dynamic systems: Theory and Formulation*. North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, 1985 (AGARD flight test techniques series). – ISBN 9789283514886

[Mar1963] MARQUARDT, Donald W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters. In: *Journal of the Society for Industrial and Applied Mathematics* 11 (1963), Nr. 2, p. 431–441. – ISSN 0368–4245

[Meh1971] MEHRA, R. K.: Identification of stochastic linear dynamic systems using Kalman filter representation. In: *AIAA Journal* 9 (1971), Nr. 1, p. 28–31. – ISSN 0001–1452

[Meh1970] MEHRA, Raman K.: Maximum likelihood identification of aircraft parameters. In: *Joint Automatic Control Conference*, 1970, p. 442–444

[Meh1974] MEHRA, Raman K.: Optimal inputs for linear system identification. In: *IEEE Transactions on Automatic Control* 19 (1974), Nr. 3, p. 192–200. – ISSN 0018–9286

[MV2003] MEYBERG, Kurt; VACHENAUER, Peter: *Höhere Mathematik 1: Differential- und Integralrechnung Vektor- und Matrizenrechnung*. 6., korr. Aufl. Berlin and Heidelberg: Springer, 2003. – ISBN 9783540418504

[Mil1947] MILLIKEN, William F.: Progress in Dynamic Stability and Control Research. In: *Journal of the Aeronautical Sciences* 14 (1947), Nr. 9, p. 493–519

[Mor1990] MORELLI, Eugene A.: *Practical Input Optimization for Aircraft Parameter Estimation Experiments*. Hampton, VA, USA, The George Washington University, Dissertation, 1990

[Mor2012a] MORELLI, Eugene A.: Flight Test Maneuvers for Efficient Aerodynamic Modeling. In: *Journal of Aircraft* 49 (2012), Nr. 6, p. 1857–1867. – ISSN 0021–8669

[Mor2012b] MORELLI, Eugene A.: Real-Time Aerodynamic Parameter Estimation Without Air Flow Angle Measurements. In: *Journal of Aircraft* 49 (2012), Nr. 4, p. 1064–1074. – ISSN 0021–8669

[MK1997] MORELLI, Eugene A.; KLEIN, Vladislav: Accuracy of Aerodynamic Model Parameters Estimated from Flight Test Data. In: *Journal of Guidance, Control, and Dynamics* 20 (1997), Nr. 1, p. 74–80. – ISSN 0731–5090

[MK2005] MORELLI, Eugene A.; KLEIN, Vladislav: Application of System Identification to Aircraft at NASA Langley Research Center. In: *Journal of Aircraft* 42 (2005), Nr. 1, p. 12–25. – ISSN 0021–8669

[MK2016] MORELLI, Eugene A.; KLEIN, Vladislav: *Aircraft System Identification - Theory and Practice*. 2nd Edition. Williamsburg, VA, USA: Sunflyte Enterprises, 2016. – ISBN 978–0997430615

[MS2009] MORELLI, Eugene A.; SMITH, Mark S.: Real-Time Dynamic Modeling: Data Information Requirements and Flight-Test Results. In: *Journal of Aircraft* 46 (2009), Nr. 6, p. 1894–1905. – ISSN 0021–8669

[Mul1986] MULDER, J. A.: *Design and evaluation of dynamic flight test manoeuvres*. Delft: Delft University of Technology, 1986

[MBI+1994] MULDER, Jan A.; BREEMAN, J. H.; ILIFF, Kenneth W.; MAINE, Richard E.; SRIDHAR, J. K.: *Identification of dynamic systems: Applications to Aircraft, Part 2: Nonlinear Analysis and Manoeuvre Design*. North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, 1994 (AGARD flight test techniques series 300). – ISBN 92–835–0748–7

[NOG+1979] NGUYEN, L. T.; OGBURN, M. E.; GILBERT, W. P.; KIBLER, K. S.; BROWN, P. W.; DEAL, P. L.: *Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability*. Hampton, VA, USA: National Aeronautics and Space Administration, 1979 (NASA technical paper 1538)

[Nor1924a] NORTON, F. H.: *The Measurement of the Damping in Roll on a Jn4h in Flight*. Langley Field, VA, USA, National Advisory Committee for Aeronautics, Diss., 1924

[Nor1924b] NORTON, F. H.: *A study of longitudinal dynamic stability in flight*. Langley Field, VA, United States, National Advisory Committee for Aeronautics., Diss., 1924

[PS2012] PINTELON, Rik; SCHOUKENS, Johan: *System identification: A frequency domain approach*. 2nd ed. Hoboken, NJ, USA: Wiley IEEE Press, 2012. – ISBN 9780470640371

[PBGM1962] PONTRYAGIN, L. S.; BOLTYANSKII, V. G.; GAMKRELIDZE, R. V.; MISHCHENKO E.F.: *The Mathematical Theory of Optimal Processes: English Translation*. New York, London: Interscience Publishers, 1962

[Pop2002] POPPER, Karl R.: *Conjectures and refutations: The growth of scientific knowledge*. London, United Kingdom: Routledge, 2002. – ISBN 0415285933

[RTS1965] RAUCH, H. E.; TUNG, F.; STRIEBEL, C. T.: Maximum likelihood estimates of linear dynamic systems. In: *AIAA Journal* 3 (1965), Nr. 8, p. 1445–1450. – ISSN 0001–1452

[Rie2017] RIECK, Matthias: *Discrete Controls and Constraints in Optimal Control Problems*. Garching, Technische Universität München, Dissertation, 29.01.2017

[RBG+2018] RIECK, Matthias; BITTNER, Matthias; GRÜTER, Benedikt; DIEPOLDER, Johannes; PIPREK, Patrick: *FALCON.m User Guide*. `www.falcon-m.com`. Retrieved: May 2018

[Ros2006] ROSKAM, Jan: *Determination of stability, control and performance characteristics: FAR and military requirements*. Lawrence, KS, USA: DARcorporation, 2006. – ISBN 9781884885549

[Ros2008] ROSKAM, Jan: *Preliminary calculation of aerodynamic, thrust and power characteristics*. Lawrence, KS, USA: DARcorporation, 2008. – ISBN 9781884885525

[SS2011] SASANE, Amol; SVANBERG, Krister: *Optimization: lecture notes*. Stockholm, Sweden: Royal Institute of Technology Stockholm, 2011

[Sch2000] SCHITTKOWSKI, Klaus: Parameter Estimation in Dynamic Systems. In: YANG, Xiaoqi; MEES, Alistair I.; FISHER, Mike; JENNINGS, Les: *Progress in Optimization: Contributions from Australasia*. Boston, MA: Springer US, 2000. – ISBN 978–1–4613–0301–5, p. 183–204

[SPR2012] SCHOUKENS, Johan; PINTELON, Rik; ROLAIN, Yves: *Mastering system identification in 100 exercises*. Hoboken, NJ, USA: Wiley, 2012. – ISBN 978–0–470–93698–6

[Sha2003] SHAO, Jun: *Mathematical statistics*. 2nd ed. New York: Springer, 2003. – ISBN 978–0–387–95382–3

[Shi1951] SHINBROT, Marvin: *A least squares curve fitting method with applications to the calculation of stability coefficients from transient-response data*. Moffett Field, CA, USA: National Advisory Committee for Aeronautics. Ames Aeronautical Lab., 1951 (NACA-TN-2341)

[Sim2006] SIMON, Dan: *Optimal state estimation: Kalman, H [infinity] and nonlinear approaches*. Hoboken, NJ, USA: Wiley-Interscience, 2006. – ISBN 978–0–471–70858–2

[SSM1962] SMITH, G. L.; SCHMIDT, S. F.; MCGEE, L. A.: *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*. NASA technical report, 1962

[SS1989] SÖDERSTRÖM, Torsten; STOICA, Petre G.: *System identification*. Hemel Hempstead, United Kingdom: Prentice-Hall, 1989. – ISBN 0138812365

[Sor1970] SORENSON, H. W.: Least-squares estimation: from Gauss to Kalman. In: *IEEE Spectrum* 7 (1970), Nr. 7, p. 63–68. – ISSN 0018–9235

[Sor1980] SORENSON, Harold W.: *Parameter Estimation: Principles and Problems*. New York: M. Dekker, 1980 (Control and systems theory 9). – ISBN 0824769872

[Ste2018] STEINBEIS-FORSCHUNGSZENTRUM OPTIMIERUNG, STEUERUNG UND REGELUNG: *Users' Guide to WORHP 1.12*. `https://worhp.de/latest/download/user_manual.pdf`. Retrieved: 2018

[SL2003] STEVENS, Brian L.; LEWIS, Frank L.: *Aircraft control and simulation*. 2nd ed. Hoboken, NJ, USA: J. Wiley, 2003. – ISBN 978–0–471–37145–8

[SBB+2010] STOER, Josef; BULIRSCH, Roland; BARTELS, Richard H.; GAUTSCHI, Walter; WITZGALL, Christoph: *Introduction to numerical analysis*. 3rd ed. New York, NY, USA: Springer, 2010. – ISBN 978–1–4419–3006–4

[SN1998] STOICA, P.; NG, Boon C.: On the Cramer-Rao bound under parametric constraints. In: *IEEE Signal Processing Letters* 5 (1998), Nr. 7, p. 177–179. – ISSN 1070–9908

[TI1969] TAYLOR, L. W.; ILIFF, K. W.: A Modified Newton-Raphson Method for Determining Stability Derivatives from Flight Data. In: *Computing Methods in Optimization Problems* (1969), p. 353–364

[TR2012] TISCHLER, Mark B.; REMPLE, Robert K.: *Aircraft and rotorcraft system identification: Engineering methods with flight test examples*. 2nd ed. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2012. – ISBN 1600868207

[TI1972] TYLOR, L. W.; ILIFF, Kenneth W.: *Systems identification using a modified Newton-Raphson method: A FORTRAN program*. Washington, USA, 1972

[WP1997] WALTER, Eric; PRONZATO, Luc: *Identification of parametric models from experimental data*. Berlin and New York and Paris: Springer and Masson, 1997. – ISBN 978–3–540–76119–8

[WI2004] WANG, K. C.; ILIFF, Kenneth W.: Retrospective and Recent Examples of Aircraft Parameter Identification at NASA Dryden Flight Research Center. In: *Journal of Aircraft* 41 (2004), Nr. 4, p. 752–764. – ISSN 0021–8669

[WV1979] WILLIAMS, J. E.; VUKELICH, S. R.: *The USAF Stability and Control Digital DATCOM. Volume I. Users Manual.* Wright-Patterson AFB, OH, USA, McDonnell Douglas Astronautics Co.-East, Diss., 1979

[Zad1962] ZADEH, L. A.: From Circuit Theory to System Theory. In: *Proceedings of the IRE* 50 (1962), Nr. 5, p. 856–865. – ISSN 0096–8390