TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Flugsystemdynamik

A Heuristic Parameter Estimation Approach for FDM Data

Javensius Sembiring

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. phil. Klaus Bengler
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Florian Holzapfel
	2. Prof. Peter Chudy, Ph.D

Die Dissertation wurde am 09.09.2019 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 24.02.2020 angenommen.

Abstract

This dissertation introduces a heuristic system identification method and its application to Flight Data Monitoring (FDM) data to estimate unrecorded parameters required in safety-related analyses. This approach solves issues of the computational instability and the physically questionable estimates of the classical system identification methods in its application to FDM data. Both of these issues are due to the fact that FDM data do not have dedicated maneuver and control input excitation as in the case of flight test data, which consequently lead to the low information content of the parameters in the FDM data. Moreover, the data used for FDM analyses are generated by civilian aircraft during their daily operation, in which the flight condition cannot be assumed to be free of disturbances from the environment. This type of uncertainty as well as noise from sensor devices degrade the quality of the recorded data.

The approach presented in this dissertation solves the issues by developing a heuristic parameter estimation strategy where the noise statistics and system-related parameters are quantified in a separate process. The separation is meant to ensure the computational stability as the number of parameters to be estimated are reduced and inter-dependencies of parameters are decreased. In estimating the process and measurement noise parameters, an Extended Kalman Filter algorithm followed by a smoothing algorithm is employed in the first step. Once the cost function criteria is achieved, the estimated noise statistics are used together with the Maximum Likelihood principle to estimate system's parameters in the second step. To increase the information content in the data, several flights are processed simultaneously. Flights with the same aircraft type, configuration, and destination airport/runway are combined and fed to the algorithm. With the developed estimation method and strategy of processing several flights simultaneously introduced in this dissertation, it is now possible to estimate parameters from the FDM data. Furthermore, this approach also opens the door for FDM analysts to extend the capability of the FDM software by embedding this approach to estimate parameters that are not recorded in the FDM data but required in the analysis.

Zusammenfassung

Die vorliegende Arbeit führt ein heuristisches Verfahren der Systemidentifikation ein, um nicht aufgezeichnete aber für Sicherheitsanalysen notwendige Parameter zu bestimmen. Diese Methodik wird in der Arbeit für verschiedene Auswertungen von Flugdaten verwendet und analysiert. Das Konzept löst Probleme mit der Instabilität der Berechnungen sowie deren physikalisch fragwürdigen Schätzungen, die durch direkte Anwendung klassischer Methoden der Systemidentifikation in der Flugdatenauswertung generiert werden können. Beide Problematiken entstehen aufgrund der Tatsache, dass die vorhandenen Daten nicht die spezifischen Manöver und Steuereingaben aufweisen, wie sie in speziellen Testflügen hervorgerufen und verwendet werden. Dies führt zu einer reduzierten Qualität der Information über die Parameter in den Daten. Darüber hinaus werden die Daten im täglichen Betrieb von kommerziell eingesetzten Flugzeugen aufgezeichnet und sind dadurch verschiedenen Umgebungsstörungen ausgesetzt. Diese Einflüsse zusammen mit den Unsicherheiten der Sensoren mindern die Qualität der aufgezeichneten Daten weiter.

Der Ansatz der vorliegenden Arbeit überwindet diese Probleme mittels einer entwickelten heuristischen Parameterschätzung in der die Charakteristiken des Rauschens und systemrelevante Parameter in einem eigenständigen Verfahren geschätzt werden. Dieses zusätzliche Schätzverfahren stellt die numerische Stabilität sicher, da die Anzahl der zu schätzenden Parameter reduziert wird sowie die Abhängigkeiten der Parameter verringert werden. Für die Schätzung der Parameter des Prozess- und Messwertrauschens wird zunächst ein Extended Kalman Filter und ein Glättungsalgorithmus verwendet. Sobald eine bestimmte Bedingung an die zugehörige Kostenfunktion erfüllt ist werden die geschätzten Rauschcharakteristiken zusammen mit der Maximum Likelihood Methode verwendet um die Systemparameter in einem nächsten Schritt zu bestimmen. Um den Informationsgehalt der Daten zu erhöhen werden die Aufzeichnungen von mehreren Flügen aggregiert. Dabei werden Flüge des gleichen Flugzeugtyps, der gleichen Konfiguration und mit dem gleichen Zielflughafen bzw. Landebahn zusammengeführt um parallel durch den Algorithmus analysiert zu werden. Mithilfe der entwickelten Schätzmethode und dem Verfahren der gleichzeitigen Auswertung mehrerer Flüge ist es nun möglich, Parameterschätzungen für Flugdaten durchzuführen. Außerdem legt das Verfahren die Grundlage zur Erweiterung der Funktionalität von bestehenden Datenauswerteprogrammen durch Flugdatenanalysten, in dem die Methodik zur Schätzung nicht vorhandener aber notwendiger Parameter integriert wird.

Acknowledgement

First of all, I would like to express my sincere gratitude to my supervisor, Florian Holzapfel, for all of the opportunities and autonomy I was given to conduct my research and for his immense knowledge. I would also like to thank Peter Chudý for being my second evaluator and Klaus Bengler for chairing my Ph.D defense. My appreciation also goes to Indonesian-German Scholarship Program (IGSP) for sponsoring me during the first four years of my research.

Special thanks to my colleagues in the Flight Safety Group at the Institute of Flight System Dynamics of Technische Universität München: Ludwig Drees, Lukas Höhndorf, Chong Wang, Phillip Koppitz, Xiaolong Wang, Sercin Höhndorf, Florian Schwaiger, Lukas Beller, and Stefan Schiele for the inspiring and exciting discussions throughout the years. Also, I would like to thank Joachim Siegel, Changwu Liu, Niclas Bähr, Nikolaos Siamisiis, and Kurthan Kersch for their contributions as students. Moreover, I would like to thank Monica Kleinoth-Gross for all of the support, particularly at the beginning and the end of my study. I would also like to thank Idris E. Putro, Bharathkumar Balaji, Jun Shi, and Xiang Fang for the fun discussions during lunch time which make Germany feels like home.

From the bottom of my heart, I would like to say a big thank you to my family: my parents, my sisters, my brother, and my in-laws for their constant support and prayers. And a very special gratitude to my wife, Debora Situmorang, to whom I am forever indebted, thank you for everything. I also would like to thank God, Almighty, as without His help, I could never have completed this dissertation. Soli Deo Gloria.

Garching bei München, August 2019 Javensius Sembiring

Table of Contents

List of Figures

List of Tables

Acronyms

A/C	Aircraft
ACMS	Aircraft Condition Monitoring System
ADSB	Automatic Dependent Surveillance-Broadcast
AFDX	Avionics Full-Duplex Switched Ethernet
AGL	Above Ground Level
AGS	Analysis Ground Station
AMC	Acceptable Means of Compliance
ANN	Artificial Neural Network
ARINC	Aeronautical Radio, Incorporated
ARL	Aeronautical Research Laboratories
ATC	Air Traffic Controller
BCD	Binary Coded Decimal
BNR	Binary
BPRZ	Bipolar Return to Zero
BSCU	Brake Steering Control Unit
CAP	Civil Aviation Publication
CDAU	Centralized Data Acquisition Unit
CPR	Correlated Position Reports
CVR	Cockpit Voice Recorder
DAR	Digital ACMS Recorder
DAU	Data Acquisition Unit
DFDR	Digital Flight Data Recorder
DFL	Data Frame Layout
DITS	Digital Information Transfer System
DMC	Display Management Computer
DMU	Data Management Unit
EAFR	Enhanced Airborne Flight Recorder
EASA	European Aviation Safety Agency
ECEF	Earth-Centered-Earth-Fixed
ECI	Earth-Centered-Inertial
ED	EUROCAE Document
EEM	Equation Error Method

EGPWS	Enhanced Ground Proximity Warning System
ELT	Emergency Locator Transmitter
EMS	Event Measurement System
EU	European Union
EUROCAE	European Organisation for Civil Aviation Equipment
FAA	Federal Aviation Administration
FCDC	Flight Control Data Concentrator
FDAU	Flight Data Acquisition Unit
FDIMU	Flight Data Interface and Management Unit
FDIU	Flight Data Interface Unit
FDM	Flight Data Monitoring
FDR	Flight Data Recorder
FEM	Filter Error Method
FOQA	Flight Operational Quality Assurance
FPM	Flight Progress Messages
FSD	Institute of Flight System Dynamics (at TUM)
FWC	Flight Warning Computer
GPS	Global Positioning System
GPWS	Ground Proximity Warning System
ICAO	International Civil Aviation Organization
ID	Identifiers
iid	Independent and identically distributed
INS	Inertial Navigation System
IT	Information Technology
LRU	Line Replaceable Unit
LSB	Least Significant Bit
MDCS	Matlab Distributed Computing Server
MLE	Maximum Likelihood Estimation
MSB	Most Significant Bit
ODE	Ordinary Differential Equation
OEM	Output Error Method
OFDM	Operational Flight Data Monitoring
OLS	Ordinary Least Square
ORO	Organization Requirements for Air Operations
P	Parity
PCMCIA	Personal Computer Memory Card International Association
PDF	Probability Density Function
QAR	Quick Access Recorder
RDMS	Relational Database Management System
RTS	Rauch-Tung-Striebel (smoother)

SDAC	System Data Acquisition Concentrator
SDI	Source/Destination Identifiers
SFDR	Standard Flight Data Recorder
SI	International System of Units
SMS	Safety Management System
SOPs	Standard Operating Procedures
sps	sample per second
SSFDR	Solid State Flight Data Recorder
SSM	Sign/Status Matrix
SSVCR	Solid State Voice Cockpit Recorder
TCAS	Terrain Collision Avoidance System
TLS	Total Least Square
TSO	Technical Standard Order
TUM	Technische Universität München
ULB	Underwater Locator Beacon
VOR	VHF (Very High Frequency) Omni-directional Radio-range
WGS	World Geodetic System
WLS	Weighted Least Square
WPS	Word per Second

Symbols and Indices

Symbols

C_D	Drag force coefficient in the Aerodynamic frame
C_L	Lift force coefficient in the Aerodynamic frame
C_Q	Side aerodynamic force coefficient in the Aerodynamic frame
C_X	Axial aerodynamic force coefficient in the Body-Fixed frame
C_Y	Side aerodynamic force coefficient in the Body-Fixed frame
C_Z	Normal aerodynamic force coefficient in the Body-Fixed frame
C_l	Rolling moment coefficient
C_m	Pitching moment coefficient
C_n	Yawing moment coefficient
E	Expected value (operator)
J	Cost function
M	Mach number
N	Number of data points (measurement)
R^2	Coefficient of determination
R	Reynold number
SS_R	Regression sum of squares
SS_T	Total sum of squares
S	Wing planform area
T	Thrust
α	Angle of attack
$\bar{c}$	Mean aerodynamic chord
β	Sideslip angle
δ_T	Throttle position
δ_{Flap}	Flap deflection
η	Elevator deflection
γ	Flight path angle
λ	Geodetic longitude
$\mathbf{A}$	State matrix of the linearized system dynamics
$\mathbf{B}$	Input matrix of the linearized system dynamics
$\mathbf{C}$	Output matrix

$\mathbf{D}$	Feedthrough matrix
$\mathbf{F}$	Process noise distribution matrix
$\mathbf{G}$	Measurement noise distribution matrix
$\mathbf{M}$	Transformation matrix between two frames
$\mathbf{R}$	Measurement noise covariance matrix
$\boldsymbol{\Phi}$	State transition matrix
$\boldsymbol{\Psi}$	Integral of state transition matrix
$\hat{\mathbf{P}}$	Corrected state error covariance
$\tilde{\mathbf{P}}$	Predicted state error covariance
$\boldsymbol{A}$	Aerodynamic force vector
$\boldsymbol{A}$	Aerodynamic center
$\boldsymbol{F_P}$	Propulsive force vector
$\boldsymbol{G}$	Center of gravity
$\boldsymbol{G}$	Gravity force vector
$\boldsymbol{H}$	Angular momentum
$\boldsymbol{I}$	Inertia
$\boldsymbol{L}$	Rolling moment
$\boldsymbol{M}$	Pitching moment
$\boldsymbol{N}$	Yawing moment
$\boldsymbol{P}$	Arbitary point in the body of the aircraft
$\boldsymbol{R}$	Aircraft reference point
$\boldsymbol{\beta}$	System's parameters vector
$\hat{\boldsymbol{e}}$	Corrected state error
$\boldsymbol{\omega}$	Rotational rate vector
$\tilde{\boldsymbol{e}}$	Predicted state error
$\boldsymbol{r}$	Position vector
$\boldsymbol{u}$	Control vector
$\boldsymbol{w}$	Process noise vector
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Output model in vector
$\mathcal{F}$	Fisher information matrix
$\boldsymbol{\Theta}$	Unknown parameter vector
μ	Bank angle
ω	Rotational rate
ϕ	Roll angle
ψ	Yaw angle
σ	Thrust elevation angle
θ	Pitch angle
φ	Geodetic latitude
ξ	Aileron deflection

ζ	Rudder deflection
f_N	Nyquist frequency
f_s	Capture frequency
f	System's state model
g	System's output model
m	mass
p	Roll rate
q	Pitch rate
r	Yaw rate
t	Time variable
u	Speed component in longitudinal axis
v	Speed component in lateral axis
v	Speed component in directional axis
x	Position in x-axis
y	Position in y-axis
z	Position in z-axis

Indices

0	Initial condition
$\bar{\boldsymbol{K}}$	Kinematic frame rotated by kinematic bank angle about the kinematic x-axis
$\boldsymbol{A}$	Aerodynamic frame
$\boldsymbol{B}$	Body-fixed frame
$\boldsymbol{E}$	Earth-Centered-Earth-Fixed frame
$\boldsymbol{I}$	Earth-Centered Inertial frame
$\boldsymbol{K}$	Kinematic frame
$\boldsymbol{N}$	Navigation frame
$\boldsymbol{O}$	North-East-Down frame
a	Augmented state
i, j	General indices
k	Discrete-time index
m	Measurement variables

Chapter 1

Introduction

1.1 Background and Motivation

Safety is of paramount importance to aviation. Individual or collaborative efforts among aircraft manufacturers, regulators, operators, and research institutions have made significant contributions to the enhancement of aviation safety. In a regional context, for example, the European Union through the European Commission has set a high safety target of its aviation accident rate for 2050 that is less than or equal to one accident per ten million flights, or equivalently with an accident probability of 10^{-7} per flight [1]. In the operator level, Deutsche Lufthansa AG[1] defines an even stricter safety target than the European's, i.e., an accident probability of equal or less than 10^{-8} per flight hour [2]. Such a high safety target has motivated all related parties to work together for making aviation meets its standard safety level and keeping it as the safest mode of transportation.

However, safety is dynamic, meaning it evolves over time. The complexity of today's aviation cannot be handled with old technology. This principle also applies to safety measures. Precautions and preventive actions or any safety procedures developed and used in the past might or might not still be relevant with the current aviation system due to safety dynamics nature which evolves and follows the changes of the system. For example, the increasing number of commercial flights must be followed by the modernization of the air traffic controller, and the advanced technologies in today's airplanes, which enable them to fly longer and with higher capacity, also require new procedures that need to be learned and adapted to the system by the respective crews.

Consequently, such changes require adjustments in safety procedures. Therefore, in order to maintain its high safety level, flight safety enhancement has to be perceived as a continuous

[1] AG stands for *Aktiengesellschaft* is a German word for a corporation limited by share ownership.

effort which mandates the adoption of changes. In the context of commercial flights, airlines play a central role as they directly interface with the aircraft manufacturers, regulators, and passengers. Any system deficiency and procedures that lead to incidents or accidents will directly affect the other three parties. Thus, as aircraft operators, airlines are required by law to strictly follow regulations in their daily operations, including regular maintenance, procedures for flight crews, and other procedures related to flight operations.

As one of the preventive and proactive safety measures, airlines implement a program called Flight Data Monitoring (FDM) program. Through this program, airlines are continuously monitoring and evaluating their operational flight safety by analyzing flight data recorded onboard. Any exceedances or deviations with respect to airlines' Standard Operating Procedures (SOPs) during a flight will be evaluated and usually followed by appropriate preventive actions. In addition to that, the FDM program allows airlines to analyze not only a single flight, but also the whole flights within an airline. This feature enables airlines to monitor and evaluate their safety in a broader context. An FDM program is mandated by law, specifically in International Civil Aviation Organization (ICAO) Annex 6 "Operation of Aircraft", Part I "International Commercial Air Transport-Aeroplanes" [3]. Excerpt of this rule is quoted as,

> *"3.3.5 An operator of an aeroplane of a maximum certificated take-off mass in excess of 27,000 kg shall establish and maintain a flight data analysis programme as part of its safety management system."*

In a regional context such as the European Union (EU), this program is adopted by European Aviation Safety Agency (EASA) and documented in Part-ORO.AOC.130 "Flight Data Monitoring-Aeroplanes" [4]. An FDM program relies on data known as FDM data, which are basically a combination of Quick Access Recorder (QAR) data and Digital ACMS[2] Recorder (DAR) data. Both data are time-series which cover entire flight phases and contain more than 2,000 parameters, including flight states and conditions. Several techniques such as exceedance or event detection, all-flight measurements, and statistical analysis are commonly applied to the FDM data [5]. In this dissertation, the term FDM and QAR/DAR are used interchangeably in the texts.

However, none of such techniques takes a physical model of the aircraft into account. They basically rely on statistical and simple algebraic approaches. On the other hand, taking a physical model of the aircraft into account might bring new insights into how safety is monitored and improved in the context of FDM program. One case in point is detecting a touchdown point during the landing phase. In an FDM approach, this event is typically detected using a landing gear squat-switch signal, which is recorded in FDM data. However, detection by solely using this parameter usually provides a rough estimation result as the recorded signal contains delay. This delay results from the time required by the landing gear strut to be compressed enough

[2] ACMS stands for Aircraft Condition Monitoring System.

and to activate the squat-switch signal. At the time the squat-switch signal is activated, the actual touch down point has occurred before the activation time [6]. In addition to that, the landing gear squat-switch parameter is usually recorded at a low sampling rate (1 Hz), which further increases the inaccuracy of this signal. Another approach of solving this problem is by involving more parameters, especially parameters that are related to this event and recorded in a higher frequency, for example, the vertical acceleration parameter [7] which is recorded at 8 Hz. By involving this parameter into the analysis, the inaccuracy result due to the time delay that results from the landing gear switch might be decreased. As pointed out earlier, none of such methods uses aircraft physical models in the analysis. Internally, the Flight Safety Research Group at the Institute of Flight System Dynamics (FSD) of Technische Universität München (TUM) has developed an alternative approach for solving such a problem by taking an aircraft physical model into account [8]. In this approach, the flight physics model together with more parameters that are involved in the analysis and an advanced Rauch-Tung-Striebel (RTS) smoother [9] algorithm which is used for data smoothing, lead to a robust approach with a consistent and higher certainty of results for the touchdown detection problem.

Furthermore, the FDM system recognizes parameters in two different categories; one category belongs to parameters that are directly recorded in QAR devices and the other belongs to non-recorded parameters which are computed based on recorded parameters through simple algebraic formulations. However, in some specific conditions, for example, when the FDM system is used to find precursors of an incident, it may require more parameters in the analysis since the existing parameters could not provide the cause-effect relationship of the incident. Such parameters cannot be computed through the method mentioned previously since the employed model has more equations than the unknowns. This is called an over-determined system. This problem can only be approached through parameter estimation, which is well addressed in the flight vehicle system identification field. Such an approach basically estimates the unknown parameters by employing the flight physical model along with estimation theory. In this approach, the aircraft physical model is used in order to solve the problem in which the current FDM system cannot. Using this approach, a new category of parameters – known as estimated parameters – is introduced to the FDM system. With these estimated parameters, analysts are provided with more comprehensive information in analyzing a particular flight. Examples presented above are also supported by the fact that the aircraft physical model has well been established and used in building aircraft simulator for pilot training purposes [10], which means that such a physical model in some extent is able to resemble the real dynamic behavior of the aircraft. With the mentioned argument, using the physical model approach as an add-on tool in FDM system is valid and might enhance the current FDM system to some extent.

In parallel with those efforts from airlines, TUM Flight Safety Group has internally been developing an alternative method in enhancing flight safety, which is called predictive analysis.

In contrast to the techniques employed in the FDM system which are mostly driven by reactive and preventive approaches, predictive analysis is a data-driven approach which involves a variety of statistical techniques to analyze historical data to make a prediction about future events or behavior [11]. Through this approach, airlines are enabled to quantify their safety level and proactively identify and quantify the main drivers of the anticipated incident/accident risks along with their most likely causal chain [12, 13]. The core of the algorithm developed in this approach is based on an aircraft physical model which is tailored to a specific incident. The approach fully utilizes the FDM data with an example algorithm applied for the runway overrun case in which the algorithm can also be extended to and applied for other cases. Seven steps are defined through this approach to address and solve the incident systematically. Those seven steps are briefly introduced here. Interested readers may refer to [12].

① **Define**. In this first step, a metric is defined through an inequality constraint that describes occurrences of an incident that may result in an accident.

② **Model**. In the second step, an incident model that contains relationships between the incident metric and other physical parameters and causal factors is developed. The model is developed based on the aircraft physical model, which is tailored to a specific incident.

③ **Identify**. This step deals with identifying and quantifying model inputs or so-called contributing factors which basically utilize parameters from FDM data. This step is related to the work carried out in this dissertation by providing parameters required in the model. These parameters are not available in the FDM data nor could be computed from a simple algebraic formulation.

④ **Cumulate**. In this step, probability distributions of each contributing factors are identified through a so-called distribution fitting. The typical method used for this purpose includes the Maximum Likelihood estimator. To automate the distribution-fitting, several fitting measures might be used, such as the Negative Log-Likelihood, Kullback-Leibler divergence, Akaike information criteria, Bayesian information criteria, Integrated quadratic distance, and mean value divergence.

⑤ **Calibrate**. The step deals with the calibration of the incident model and its contributing factors by checking if the incident model and its inputs include all relevant functional relationships and contributing factors. The comparison is made between the incident model output and flight measurement. If there is a minor difference between the two quantities, a minimum adjustment is performed on the input distributions without falsifying their statistics. However, if there is a major difference which requires a large adjustment to the model inputs, the proposed incident model indicates that it is not suitable for the predictive analysis. If this case found at this step, proposing a new incident model with less or more contributing factors might be required.

⑥ **Revise**. As pointed out earlier in the Calibrate step, the Revise step is only necessary if there is a major mismatch between the incident model and the flight measurements. Therefore, in this step, the complexity of the model might be increased by including more variables as contributing factors; or it might also be decreased by excluding variables that do not significantly affect the model.

⑦ **Predict**. In this last step, the probability of the considered incident is quantified along with its contributing factors.

As presented above, the lack of additional parameters required for the analysis has motivated the work in this dissertation. As part of the seven steps of the predictive analysis approach being developed in the Flight Safety Group, the work carried out in this dissertation is expected to provide more parameters as required by the incident model. In the context of flight vehicle system identification method, this dissertation is driven by the fact that the existing methods are not tailored to work with FDM data due to the low information content in such data. Thus, the work in this dissertation introduces some modifications to the existing methods and a data processing technique that are compatible with FDM data in order to produce reasonable estimates with good statistical properties.

1.2 State of the Art

In this section, the state of the art related to the capabilities of existing FDM software and classical system identification methods are discussed. As for FDM programs, the description is focused on the limitations of the existing software to generate parameters other than the recorded ones, while in the system identification method, the description is emphasized on the limitations of the existing method when applied to the FDM data.

1.2.1 FDM Program and Existing Software

The Flight Data Monitoring program is a part of the airline Safety Management System (SMS) program aimed at proactively identifying risks along with appropriate remedial actions [5, 14]. The FDM program mainly utilizes FDM data as recorded onboard aircraft and other parameters that are computed through simple algebraic formulations. In implementing such a program, airlines might choose between different software vendors which are commercially available. Several common techniques such as all-flight measurements, event detections, and some statistical analyses are provided as built-in features in the software [15, 16]. These techniques rely only on the existing parameters [17]. However, in some cases, an analysis might require a new parameter other than parameters that are available in the FDM recorder, for example, the runway friction coefficient, engine thrust during specific flight phase, or aerodynamic-related parameters. These parameters are not recorded, nor could they be derived through simple algebraic formulations. The work carried out in this dissertation fills this gap by estimating such parameters through a system identification method with fully utilizing the recorded data along with the aircraft physics-based model.

1.2.2 Classical System Identification Methods

Flight vehicle system identification methods such as Equation Error Method (EEM), Output Error Method (OEM), Filter Error Method (FEM) and filtering-approach method such as Extended Kalman Filter for dual estimation problem are developed to be compatible with data recorded from flight testing programs [18, 19]. Such methods have less numerical instability issues and produce good estimates since the utilized data contain rich information contents. Flight test data are obtained from a designated experiment setup involving dedicated flight maneuvers, dedicated control input, and use of sensors with high quality and high sampling rates [18, 20, 21, 22]. In addition to that, flight testing is typically performed in a relatively calm air condition which results in data with low process noises. In contrast, FDM data are recorded mostly in opposite conditions, for example:

- no dedicated flight maneuver meant for data content enrichment but for operational purposes only;
- no designated input on the control surface, and;
- the flight is performed in a condition that is permissible within the safety boundary. In this case, one may expect data that is disturbed by the environmental condition such as gust and turbulence.

Applying the classical system identification methods on such data typically lead to numerical instabilities as well as poor estimates results. Based on the literature study, very few references or works utilize FDM data for parameter estimation purpose. Some of them are presented below.

Wang *et al.* in [23] present aerodynamic modeling and parameter estimation by utilizing the FDM data from an airplane approaching at a high-altitude airport. The Extended Kalman Filter followed by the Modified Bryson-Frazier smoother is used for state estimation, while the parameter estimation techniques (Neural-Network-based and Delta method) are applied for estimating aerodynamic parameters. This paper also presents a geometric method for reconstructing angle of attack (α) and sideslip angle (β) variables. However, no further explanation on how the process and measurement noises are treated as no values corresponding to these two parameters presented in the paper. Furthermore, only approach phase is considered in the paper.

In [24], Lan *et al.* also apply parameter estimation technique to the FDM data from an airplane with the same airport destination as in [23]. In this paper, the landing phase is investigated with a focus on analyzing the effect of horizontal aerodynamics and flight dynamics at a high-altitude airport. The compatibility analysis is conducted to remove bias in the data by minimizing weighted residuals. The aerodynamic stability is obtained through Fuzzy Logic modeling where the estimates are time-dependent, which are a nonlinear function of the state variables. As in the previous paper [23], no further explanation is found on how the process and measurement noises are treated in this paper. Even though the investigated phase is the landing phase, the data considered in the paper do not include the ground roll phase.

In addition, the two presented papers do not provide further explanations concerning the number of flights used in the analysis. In the context of flight vehicle system identification, particularly in the parameter estimation field, a complementary data is required in order to check the predictive capability of the developed model and the estimated parameters [18, 25, 26].

In [27, 28], Haverdings *et al.* present the Extended Kalman Filter combined with the smoothing algorithm for windshear and turbulence studies based on the FDM data. No system's parameters are estimated in this paper; thus, the Extended Kalman Filter is purely used only for state estimation. As in the previous paper, it also not clear how the process and measurement noises are treated in this paper.

The work conducted in this dissertation presents a systematic approach where process and measurement noise, as well as the system parameters, are estimated. The classical system identification methods are improved to be able to work with the FDM data. To increase the information content of the FDM data, simultaneous flights are processed by the developed algorithm. With the approach proposed in this dissertation, the estimates are expected to be physically reasonable with good statistics properties.

1.3 Mission Objective

The objective of this dissertation is to estimate unrecorded parameters based on the FDM data through a system identification method that is tailored specifically to low information content data. Being able to estimate the unrecorded parameters introduces a new set of parameters to the FDM system which enables airlines to analyze an incident using well-informed data. Moreover, any flight physics with FDM data-based researches, e.g., flight physics-based predictive analysis, may be provided with the estimation approach and unrecorded parameters through the work conducted in this dissertation.

1.4 Contribution

The following summarizes the contributions of this dissertation:

1. **Estimation of parameters which are required for incident analysis but cannot be measured nor computed directly.** Based on the author's involvements in German

and European projects with several airlines, the existing FDM programs that are utilized by the airlines do not have a capability to extract the parameters through the parameter estimation method. Thus, the daily analysis performed within the airlines is based on the recorded parameters with some complementary parameters through an algebraic formulation. In this context, this dissertation contributes to the airline industry or to any related institutions that utilize FDM data by providing unrecorded parameters through the parameter estimation method. In the context of FDM, parameters are broadly classified into two categories, i.e., 1) measured parameters, available from FDM data as recorded directly by the aircraft's sensors, and 2) derived parameters, computed from a simple closed form algebraic formulation or logical evaluation. However, in a specific incident type, an analysis requires additional parameters, e.g., thrust parameter, runway friction coefficients, or aerodynamic coefficients. These parameters are required in a specific event or incident analysis, but they could not be recorded nor computed through a simple algebraic formulation. This dissertation provides these parameters through a parameter estimation method tailored to FDM data.

2. **Development of system identification method that is capable of handling low information content data.** Flight vehicle system identification methods basically deal with data obtained from a dedicated flight testing program. In flight testing, control inputs are optimized in order to enrich the information content in the measured data, e.g., dedicated control inputs, specific flight maneuvers, selection of appropriate flight conditions, and the use of the dedicated sensors with high accuracy and high sampling rate. In contrast, a flight performed on a daily basis by an airline does not follow procedures aimed at data quality enrichment but to bring passengers from one point to the other point safely, economically, and comfortably. The lack of active excitation on a daily civil flight leads to less information content in the recorded data. Furthermore, the sampling rate of the data recorded in the FDM can be regarded as low compared to the flight testing data. Application of a classical system identification method to FDM data will produce physically questionable estimates as well as numerical instability issues. This dissertation introduces a novel method by modifying the Filter Error Method, which is the most advanced method in the classical system identification methods as it can handle data affected by process and measurement noises. The modification is mainly done by solving the instability issue of the Filter Error Method by separating the estimation of the noise statistics parameters and system parameters into two different stages. In the first stage, the algorithm is only dedicated to estimating noise-related parameters such as process noise and measurement noise covariances. While in the second stage, the algorithm is dedicated to estimating system parameters only. With this strategy, the stability of the algorithm is expected to increase as fewer parameters are required to be estimated during the estimation process. However, the method developed in this dissertation is different from the two-step-procedure approach. In the two-step-procedure approach, two different dynamic models are utilized. In the first step, a kinematic model,

along with filtering/smoothing algorithm, is employed to decrease the noise effect to the data. While, in the second step, a dynamic model that contains aerodynamic parameters or any related system parameters along with parameter estimation methods are used together to estimate the parameters of interest. Furthermore, in the two-step-procedure approach, there is no feedback link between the first and the second step. Any model deficiency or wrong selection of the parameter estimation method in the second step will not affect the quality of the result produced in the first step. However, this relation does not apply in the forward link. The result produced in the first step will affect the quality of estimates produced in the second step. In contrast, the developed method only utilizes one dynamic model both in the first and the second step, which leads to more straightforward and tractable procedures.

3. **Development of an FDM data processing technique for the enrichment of data information content.** Linked with properties of the FDM data as described in point 2, the application of the developed method in this dissertation alone is not enough to estimate parameters of interests with reasonable results. Applying the developed method to a single flight would produce questionable estimates in terms of physical value due to low information content in the data as well as numerical instability during the computation process. Therefore, a new data processing technique is required to enrich the information content in the data. This dissertation proposed a novel processing technique of FDM data by simultaneously processing several numbers of flights with similar configurations. This approach is supported by the fact that some parameters can be assumed the same for all flights while some parameters are flight-dependent. Thus, during the execution of the algorithm, the parameters that are independent of flight will be estimated once and kept the same for all flights while the parameter-dependent flight will be estimated for each flight. Combining several flights and processing them simultaneously ensure that each data complement each other in term of information content required in the estimation process. Combination of this data processing technique along with the method developed in point 2 above ensures the success of the parameter estimation process, both from numerical stability and reasonable estimates with good statistical properties.

Besides the main contributions presented above, there are also some further contributions that are worth mentioning. These contributions are not directly linked to the research topic but technically support the Flight Safety research group members in general and consortium members of several European projects where the author is involved.

1. **Development of a flight data decoder following the ARINC[3] 717 standard**. Flight Data Acquisition Unit (FDAU) records the flight data and stores them in binary files. The file is then downloaded on a daily basis or within a certain period by the airlines using FDM software. From the aircraft, the FDM software further decodes the binary data into engineering values so that the analyst can work on the data. However,

[3] ARINC stands for Aeronautical Radio, Incorporated.

FDM software is proprietary. As a research institution, data are required in a readable format to be analyzed, but receiving flight data in this engineering format from airlines is storage- and time-consuming. Storage-consuming is caused by all parameters, both recorded and computed, are decoded by the FDM software. For a long-haul flight, the decoded flight data of a full flight phase might contain around 2,000 parameters with 1 GBytes of file size in a .csv format. Besides storage-consuming, this big file also increases the loading time when the data is processed by the algorithm. In many cases, it is found that not all parameters are required in an analysis. Typically, not more than 150 parameters are required in a specific event/incident analysis. Therefore, the Flight Safety group preferred to have data in the binary format and developed its own decoding tool. With this decoding tool, the analyst has an option only to decode parameters that are relevant to the case at hand, which consequently reduce the file size and processing time. The decoding tool was initially developed by Möhr [29, 30] in Matlab. The involvement of the Flight Safety group in two European projects, Future Sky Safety[4] [31] and SafeClouds.eu[5] [32], also requires an FDM decoding tool to be used within the consortium. However, as a general agreement between consortium members, the software used in the project must be from an open source project which prevents the Flight Safety group decoding tool from being used in the projects. Therefore, the author ported the Matlab decoding tool by using open source software, Python. The author then added more features into the ported decoding tool, improved run-time performance, and furthermore deployed the algorithm in the protected source code to be used by all consortium members. Additionally, to protect the code from the legal point, the author registered the decoding tool at TUM ForTe under the name "Dekodierung von binären Daten im ARINC 717 Format mit Python" with 2017-10S01 as the reference number.

2. **Development of the Flight Safety group IT infrastructure in a scalable distributed storage and processing system that is capable of dealing with big-data environment**. The research activity of the Flight Safety group was started in 2010. During this early period, the number of FDM data received from partner airlines was very limited due to airlines FDM data protection which restricts FDM data distribution outside the company. With the very limited FDM data, the storage and processing system were performed conventionally. However, as the predictive analysis method became more attractive in the aviation community, specifically in airlines, it opened more opportunities for the Flight Safety group to cooperate with other airlines. From 2014 until 2018, the Flight Safety group has cooperated with five airlines, including airlines from European and Asian region. Consequently, the Flight Safety group received a huge number of FDM data. Storing and processing these huge number of

[4] Future Sky Safety is an EU-funded transport research program in the field of European aviation safety that brings together 33 European partners to develop new tools and new approaches to aeronautics safety. The project's period is over four years starting in January 2015.

[5] Safeclouds.eu is an EU-funded research project that aims at improving aviation safety by developing big data tools. The project is a three-year period starting in September 2016.

data in a conventional way would be very inefficient. Therefore, an Information Technology (IT) infrastructure that is capable of handling huge data in a scalable system is needed. In 2014, the author along with Lukas Höhndorf started to develop the IT system by utilizing an open source software, in this case Hadoop[6] for handling large data in a scalable distributed storage system and MySQL[7] for the database management system. Whereas for a parallel processing system, Matlab Distributed Computing Server (MDCS) was installed that enables data to be processed in parallel. The IT system was configured in a multi clusters framework, and the access is restricted only for members of the Flight Safety group through their computer. The developed IT infrastructure enables the Flight Safety group members to work in a parallel framework. Compared to typical FDM program used by airlines where data processed by a single computer, the IT framework developed in the Flight Safety group provides faster computation time in a distributed and redundancy storage and processing system. The description of the developed IT infrastructure is depicted in Appendix B.

1.5 Organization of Dissertation

Including this introductory chapter, this dissertation is organized in seven chapters (see Figure 1.1), and each chapter is briefly described as follows:

- Chapter 2 introduces aspects of flight recorder in general, including technological evolvement, regulatory aspect, and the usage of these data in the aviation field. Furthermore, Chapter 2 presents the Quick Access Recorder (QAR) data, which is the main data used in this dissertation. Aspects related to data acquisition and protocol are also presented in this chapter. Chapter 2 is enclosed by presenting the characteristics of the FDM data from the Flight Vehicle System Identification point of view.
- Chapter 3 presents a nonlinear mathematical model of the aircraft dynamic motion. Assumptions which serve as foundations for deriving the equation of motions are discussed, and the corresponding four sets of equation of motions are derived and presented afterward. This chapter also presents aerodynamic models that are commonly found in the Flight Mechanics field.
- Chapter 4 contains theory foundations behind the system identification in general, and some derivative methods used in Flight Vehicle System Identification, including Equation Error Method which basically based on the least-square principle, Output Error Method which is derived based on the Maximum Likelihood principle, Filter Error Method which is an extension of Output Error Method and Extended Kalman Filter (EKF) which is adapted to dual-estimation problem.
- Having been discussed properties of QAR data, aircraft mathematical model, and system

[6] An open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems [33].

[7] An open source Relational Database Management System based on Structured Query Language (SQL) [34].

identification method in general, Chapter 5 presents a system identification method tailored to QAR/FDM data. The method introduced in this chapter is an extension from a classical system identification method to a method that is capable of dealing with the FDM data.

- In Chapter 6, the application of the developed algorithm introduced in Chapter 5 is presented. Some application aspects, such as data selection and computational strategy, are covered in this chapter. The corresponding results are then presented in this chapter.
- Finally, Chapter 7 summarizes the final conclusion about what has been achieved together with some perspective of the work conducted in this dissertation.

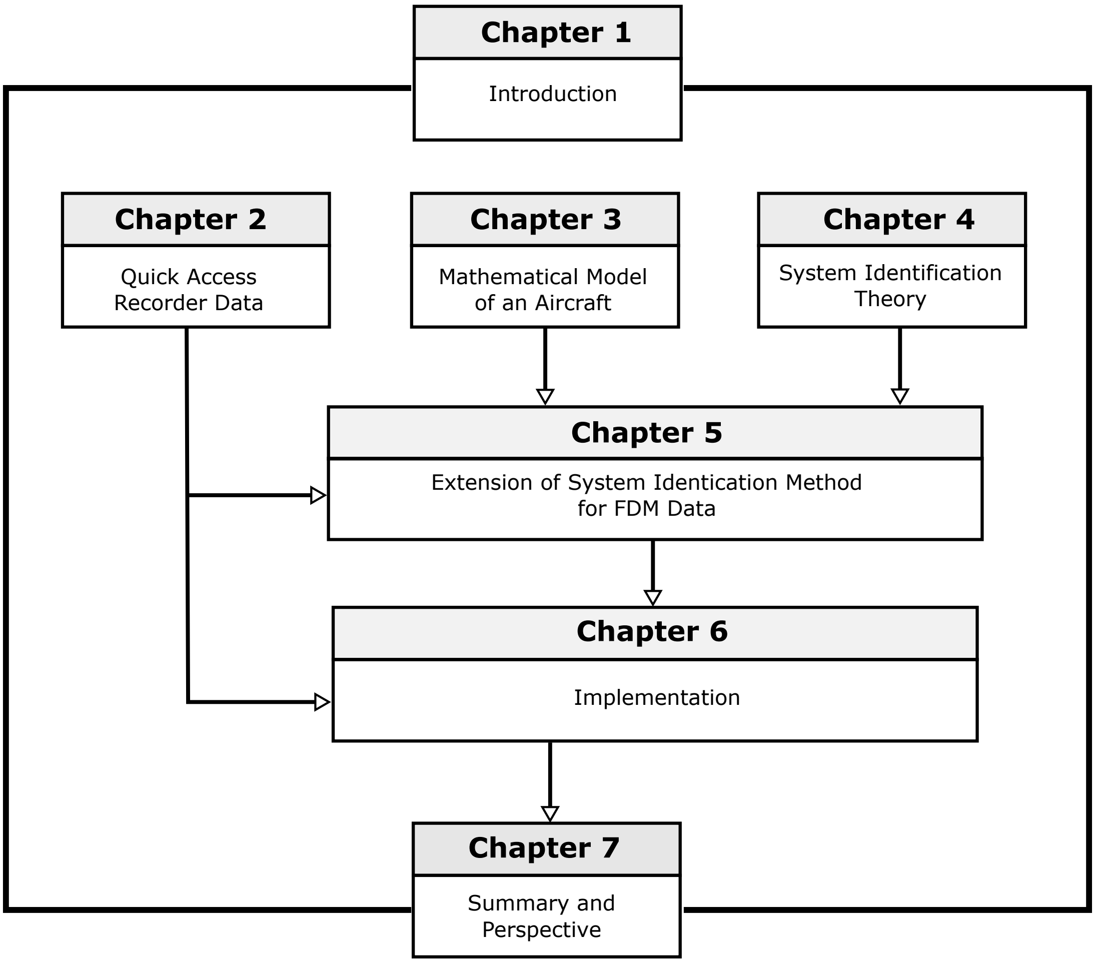

Chapter 1
Introduction
Chapter 2
Quick Access Recorder Data
Chapter 3
Mathematical Model of an Aircraft
Chapter 4
System Identification Theory
Chapter 5
Extension of System Identication Method for FDM Data
Chapter 6
Implementation
Chapter 7
Summary and Perspective

Figure 1.1: Structure of dissertation

Chapter 2

Quick Access Recorder Data

Today, there are wide varieties of ground-based and airborne aviation recorders which provide vital information required for accident prevention or investigation purposes. Airborne-based information sources include the mandatory crash-protected flight recorders and quick-access data recorders. In addition to these sources, avionics system such as the Enhanced Ground Proximity Warning System (EGPWS)[1] computer, whose main purpose serves as a warning system, can also store valuable parameters for incident investigation due to the advancements in digital computer technology [35]. The ground-based information sources may include radar track data, Automatic Dependent Surveillance-Broadcast (ADSB) data, and Air Traffic Controller (ATC) data such as Correlated Position Reports (CPR)[2] and Flight Progress Messages (FPM)[3] [36]. The analysis conducted in this dissertation mainly processes data from an airborne-based recorder called Quick Access Recorder (QAR) device. Thus, this chapter will focus on the explanation of this flight data type along with its usage in the aviation community. However, as the QAR device is originally derived from the Flight Data Recorder (FDR), this chapter begins with a discussion of the FDR technological evolution and its regulations. This discussion will be followed by a presentation on technical aspects related to QAR data, which include data acquisition, recording and transmission process in the aircraft, and data transfer to the ground facility. The Flight Data Monitoring program, which utilizes QAR data for incident/accident prevention, is also introduced in this chapter. Finally, QAR data characteristics related to system identification field are briefly discussed in the last section of this chapter.

2.1 Flight Recorder

A flight recorder popularly referred to as the black box, is a crash-protected electronic recording device which keeps track of specific flight states and performances and is at the core of accident

[1] Enhanced Ground Proximity Warning System is a warning system designed to alert pilots if the aircraft is in immediate danger of flying into terrain or an obstacle.

[2] Aircraft position data derived from Air Traffic Control (ATC) surveillance systems, normally updated every 1 to 3 minutes.

[3] Messages which inform the Notification Message (NM) about the progress of airborne or almost airborne flights.

investigations [37]. There are usually two individual devices inside the flight recorder which are a Flight Data Recorder (FDR) and a Cockpit Voice Recorder (CVR). Even though these two recorders are called ‘black boxes,’ they are actually painted in bright orange for ease in spotting it among the wreckage in case of an accident. A typical FDR is 12.7 cm in width, 16 cm in height, 50 cm in depth and 4.8 kg in weight. The FDR device is meant to preserve the time series of flight data through recording parameters such as airspeed, positions, engines, control surfaces, and environmental conditions from a variety of aircraft sensors. Typically, an FDR device fitted in a modern jet aircraft can record thousands of parameters. The recording retains the last 25 hours of aircraft operation in the endless-loop principle [38]. A CVR, on the other hand, records the history of the audio environment in the cockpit area through several microphones which are usually located on the overhead instrument panel between the two pilots. The audios recorded in CVR might include engine noises, radio transmissions, audios of the crew’s conversations, stall warnings, and other clicks and pops. While an older CVR only retains the last 30 minutes of aircraft operation, a modern CVR extends up to the last 2 hours. Similar to an FDR, a CVR also records the information in the endless-loop operation. A typical traditional CVR is 16 cm in height, 12.7 cm in width and 32 cm in depth with 4.5 kg in weight [38].

For spotting purpose following an accident, an aircraft is equipped with a radio beacon called Emergency Locator Transmitter (ELT) which is located at the top of the rear fuselage, see Figure 2.1. This device sends out a radio signal through an external fixed antenna on a dedicated frequency of 406 MHz (formerly 121.5 MHz). This signal is automatically activated 50 seconds after an accident happens [39]. The ELT frequency is monitored worldwide and can be located by triangulation method or by a GPS signal [40]. However, the ELT device is inoperable underwater, which necessitates a submersible spotting device also to be fitted onboard aircraft. This device is called Underwater Locator Beacon (ULB), commonly known as 'pinger' [41]. Unlike ELT, a ULB is directly attached to both FDR and CVR and is not equipped with an external antenna. Its purpose is solely for underwater relocation of the FDR and CVR devices by transmitting an emergency signal at a frequency of 37.5 kHz. The signal is transmitted every second and can only be detected with a special receiver. The signal is automatically broadcasted by the ULB when the recorder is immersed in water with a maximum depth operation of 6,000 m [42]. As this device is battery-powered, it can only transmit a signal for at least 30 days.

Generally, flight recorders are mounted in the tail section of an aircraft as this location is least likely to be severely damaged following an accident. Placing the flight recorders in this position makes the entire front of the aircraft as a 'crush zone' that reduces the impact on the flight recorders. All data recorded in the black-box comes from a device called a Centralized Data Acquisition Unit (CDAU). This device gathers all data from the sensors and transfers them to be recorded in the FDR device. Figure 2.1 depicts an indicative location of flight recorder system in an aircraft.

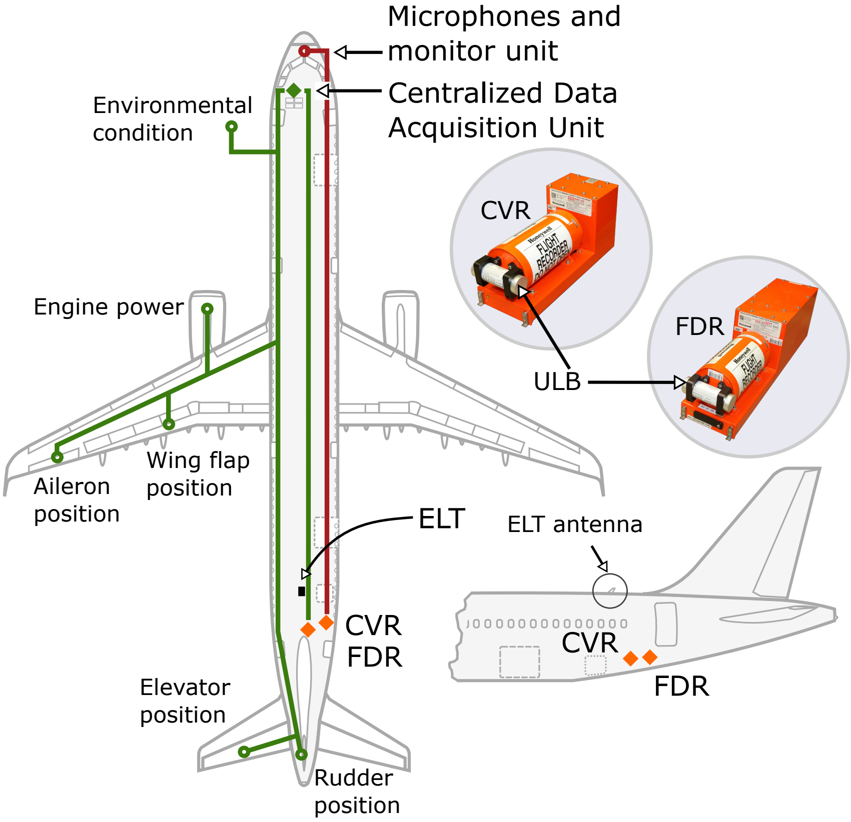


Figure 2.1: Typical FDR/CVR, ELT antenna and data acquisition unit location, adapted from [39]

ICAO Annex 6 "Operation of Aircraft" allows a commercial airplane to use a combination recorder [3]. In this combined recorder, the CVR and FDR are housed in a single box and commonly known as the digital voice and data recorder (DVDR). Some aircraft types, such as Embraer 170, are fitted with this type of recorder [43]. When the combined recorder (DVDR) is used on an aircraft, two units of such recorder are required to increase the redundancy and the likelihood of readable information surviving a crash [3]. One DVDR is located near the cockpit and the second DVDR is placed in the same position as the traditional FDR/CVR's location, which is in the tail section of the aircraft. Placing the recorder at the forward part of the fuselage section has the advantage of shorter cable distances from the cockpit area to the flight recorder device's location. This configuration leads to a lower risk of the wires being breached or broken up during an in-flight fire. On the other hand, mounting recorders in the rear of the plane provides a higher impact survivability.

2.1.1 Technological Evolution

Many important innovations in the development of the flight data recorder have been built in several simultaneous and separate streams of work. The history has not shown a single inventor but attributed several individuals and team efforts in the invention of this recording device. In the following paragraphs, three stages of flight recorder development, particularly from technological evolvement, are briefly discussed.

Early Development, 1930 - 1960s

The earliest attempt of the development of the flight recorder dated back to 1939 when the so-called 'HB' flight recorder was developed by Paul Beaudouin and François Hussenot at the Marignane Flight Test Center, France [44, 45]. This recorder was a photograph-based device which used a scrolling photographic film as the recording medium. The medium was 8 meters long and 88 millimeters wide. The parameters such as altitude and speed were recorded by shooting a thin ray of light on the photographic film. The light deviated through a mirror tilted according to the magnitude of the parameter. Even though the 'HB' flight recorder was used in service during this period, it was mainly employed during a planned flight test because this recorder could not be erased and recycled and it needed to be changed periodically.

Len Harrison and Vic Husband made another attempt on developing the flight recorder at Farnborough, United Kingdom. Copper foil was used as a recording medium which protected the data from fires and high impact loads. The parameters were recorded through various styli indicating aircraft instruments which indented the copper foil. The indented copper foil provides the records of the instruments' readings. Even though this invention was limited in terms of the amount of data recorded, it was the pioneer of today's flight data recorder, one which is able to withstand extreme conditions [46]. Although the two flight recorders were already operational and used in military aircraft, the recorders did not gain attention from commercial flights until David Warren from Australia introduced the flight data recorder along with a voice recorder into a single unit called 'ARL[4] Flight Memory Unit' in 1958 [47]. The involvement of Dr. Warren in aircraft accident investigation of the famous De Havilland DH-106 'Comet' in 1953 prompted the development of this device. The safety investigators at that time found only a few clues and possible causes of the accident, as there were no witnesses, nor survivors of the accident. David Warren, who was part of the investigatory committee, then realized the necessity of finding a way to record flight crews' conversations. The ARL recorder could store up to 4 hours of pilot's voices as well as instrument readings up to the moment of an accident in an endless-loop mechanism. The steel wire was used as the recording medium. The original of Warren's ARL Flight Memory Recorder shown in Figure 2.2 is now displayed in the Science Museum, Melbourne, Australia [48].

The Warren's flight recorder was then developed into its second generation to update it to a pre-production standard. Many improvements were made, such as increasing the accuracy of the recording system as well as an increasing number of parameters to be recorded. A

[4] ARL refers to the Aeronautical Research Laboratories which at the time of the invention was part of the Australian Commonwealth Government of Supply.

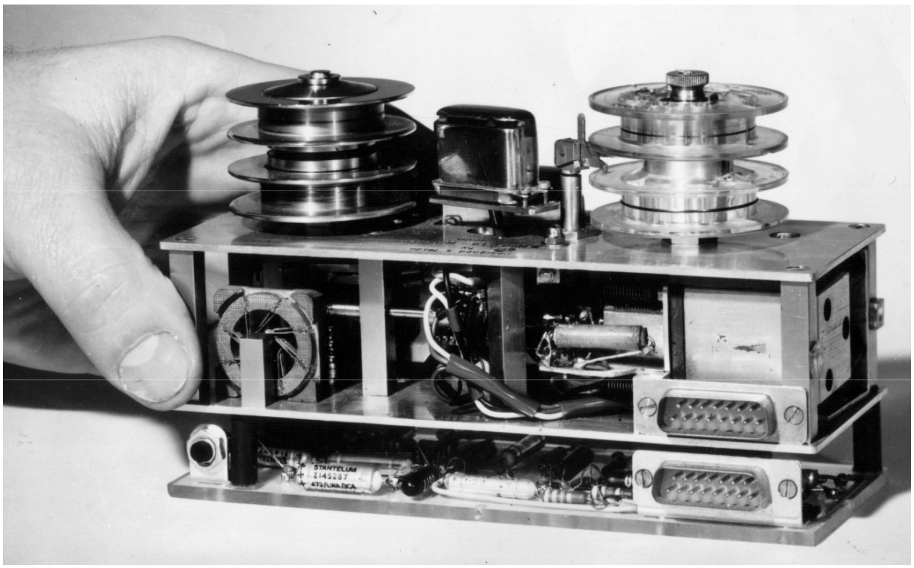


Figure 2.2: The original ARL Flight Memory Recorder [48]

British firm, S. Davall & Son, obtained the production rights and their well-known 'Red Egg' crash recorder was developed based on it. The 'Red Egg' recorder produced a large interest from British and overseas markets at that time [47]. In the early of 1960, Dr. Warren's flight recorder became mandatory for all civil aircraft in Australia, and it made Australia become the first country in the world to make flight data and voice recording compulsory [49].

During this period, another contribution to the development of the flight recorder came from the United States. The contribution was related to the survivability of flight data recorders. It was known at the time that all operable flight recorders shared a common issue, i.e., they could not survive a catastrophic crash which led to the loss of valuable information needed in the accident investigations. Prof. James R. Ryan, who was involved in national 'FDR'[5] project began the effort to design an anti-crash flight recorder. His approach was to construct a flight recorder as simple as possible in terms of design, installation, and service requirements. Prof. Ryan, along with his team, successfully built the first anti-crash flight recorder named the Ryan VGA recorder. The recorder was labeled as VGA as it can only record three parameters, which are velocity (V), g-force (G) and altitude (A). In principle, the design of the VGA recorder consisted of a tiny electric motor that moved a thin sheet of aluminum foil (2 inches in width) horizontally at an average speed of 4 inches/hour. The recordings were marked by a small pointed stylus scratched along the foil. The foil was able to survive a temperature of 1,000 $°C$ for 30 minutes and could withstand 20 g. Altitude parameter was measured through a precision aneroid that is a tube-connected to the aircraft's pitot-static system, while airspeed parameter of up to 500 mph was determined with a diaphragm that measured the difference between the static and dynamic pressure. The g-force parameter was measured by a small weight attached on a cantilever balance beam. This device could measure the acceleration from -3

[5] A project initiated by the U.S Army Air Corps and Civil Aeronautics Boards aiming at the development of anti-crash flight recorder.

to $+12\ g$. Each of the three instruments has its separated pointed stylus and moves along its track on the rolling aluminum foil. The instruments are housed in a 16-pound hatbox-size in two compartments with the recording equipment positioned above and measuring devices below. The container was sealed against humidity and moisture. Beside its survivability, the VGA recorder could also operate for 300 hours without maintenance, which makes it the state-of-the-art flight recorder in its time [50].

In summary, the early development of flight recorder stored data on a metal foil in an analog format, which limits the recorded number of parameters. Typically, around five parameters were recorded, including airspeed, heading, altitude, vertical acceleration, and time [6]. However, as more parameters were required in an aircraft accident investigation and the inability of the early flight recorders from the 1930s to the 1960s to provide these additional parameters, there was a need for a more advanced recorder with higher storage capacity. This led to the development of the second generation of flight recorders in the 1970s and 1980s.

Second Generation, 1970 - 1980s

The first generation of flight recorder was unable to record many parameters because each parameter was stored in an analog format. However, the change of FDR rule in this era (1970 – 1980s) required more parameters to be recorded. The introduction of the Flight Data Acquisition Unit (FDAU) and the replacement of metal foil with magnetic tape were the response to the rule change. FDAU enabled the process of digitizing and storing a large amount of incoming sensors data in the magnetic tape. The term 'Digital Flight Data Recorder' (DFDR) was introduced in this era since all data were stored in a digital format. The medium used for storing data was similar to an audio recording tape and mounted in a crash-protected box, see Figure 2.3. Typically, this recorder was able to record up to 25 hours data on a 300 to 500 ft long rolling-tape [51].

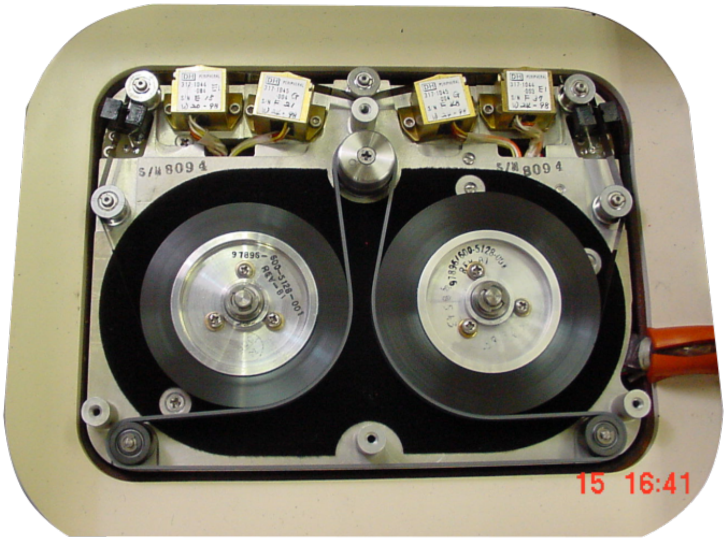


Figure 2.3: DFDR magnetic tape medium [6]

Third Generation, 1990 - now

The introduction of solid-state memory as a recording medium marked the beginning of the third generation of FDR device. This recorder is also called Solid State Flight Data Recorder (SSFDR) for recording flight data, and correspondingly Solid State Voice Cockpit Recorder (SSVCR) for recording sounds in the cockpit. This new technology was intended to replace the tape-recorder as a response to FAA FDR regulatory change in 1991 [51]. The new medium allows the recording of a large amount of data and improving data usability. Furthermore, solid-state technology avoids the use of moving parts as in a tape-recorder, which leads to the ease of maintenance and increases reliability of the recorder.

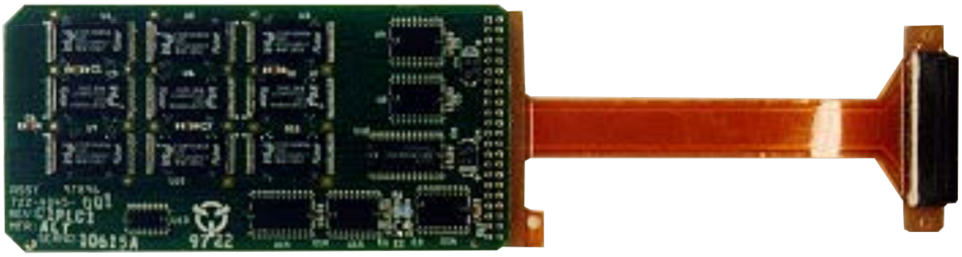


Figure 2.4: DFDR solid state medium [52]

The latest advancement of FDR technology is the so-called Enhanced Airborne Flight Recorder (EAFR) which was first introduced by the Federal Aviation Administration (FAA) to the International Civil Aviation Organization (ICAO) in 2007 [53]. Boeing was the first aircraft manufacturer who adopted this technology through its Boeing 787 Dreamliner in 2011 [54], [55]. A critical feature in the EAFR technology is its ability to receive and process data directly from sensors without passing the Data Acquisition Unit (DAU) [56]. A special software embedded in the EAFR system serves as virtual DAU which receives the analog data from the sensors, digitizes the data, and sends them through Avionics Full-Duplex Switched Ethernet (AFDX)[6] to a recorder device. This design significantly reduces the overall weight of the system. Furthermore, the EAFR can record 2 hours of video information and up to 2,000 parameters for 50 hours of recording time. The following Figure 2.5 summarizes the evolution of the FDR technology in terms of the storage medium and the corresponding capability of storing parameters in some commercial aircraft.

2.1.2 Operational and Regulatory Aspects

General regulations of flight recorder are extensively described in the ICAO document specifically in Annex 6 "Operational of Aircraft", Chapter 6 and Appendix 8 [3]. This document covers aspects of the FDR equipment regulations, installations, inspections, serviceability, calibrations as well as the use of FDR data for flight data monitoring (FDM) program. The latter

[6] A data network, patented by Airbus for safety-critical applications that utilizes dedicated bandwidth while providing deterministic quality of service.

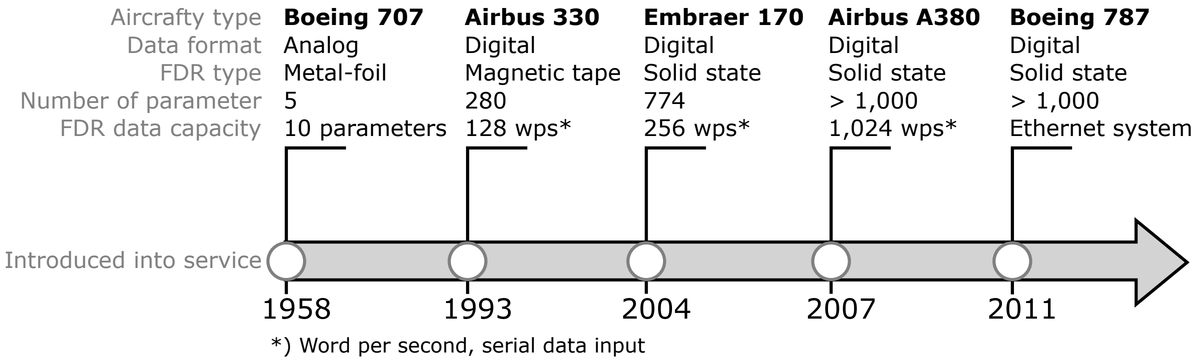


Figure 2.5: FDR technological evolution, adapted from [43]

utilizes data from a so-called Quick Access Recorder (QAR) data, which is essentially a copy of FDR data. This data type is further described in Section 2.2. Some general requirements for FDR device as documented in ICAO Annex 6 are listed below:

- It is painted a distinctive orange or yellow color;
- It carries reflective material to facilitate its location;
- It is placed in a location that the probability of damage to the recordings is minimized;
- It is provided with the underwater locating device, etc.

Pertaining to technical specifications of flight recorder, ICAO Annex 6 refers to ED-112 document produced by the European Organisation for Civil Aviation Equipment (EUROCAE). ED-112 specifies technical specifications of flight recorders (FDR and CVR) to comply with ICAO general regulation of flight recorders as a 'crash-protected' recorder. Specifications documented in ED-112 became standard and adopted by many regulators [52], for example FAA TSO[7] C-124b for flight data recorders [57], and TSO C-123c [58] for cockpit voice recorder which both refer to ED-112 document. Table 2.1 shows technical specifications for flight recorder as 'crash-protected' recorder as specified in ED-112 document. ICAO Annex 6 also regulates the minimum number of parameters to be recorded in the flight recorder. Furthermore, these parameters are called as mandatory parameters that every aircraft manufactures should implement in their flight recorders. However, the mandatory parameters are not the same for all aircraft but depending on the aircraft's date of certification, maximum take-off weight, and the maximum number of passenger seats. This regulation is adopted by the FAA and documented in Part 121.344 "Flight Data Recorders for Transport Airplanes," see Figure 2.6.

2.2 Quick Access Recorder Data

Quick Access Recorder (QAR) data are a copy of the Digital Flight Data Recorder (DFDR) data in a non-crash protected recorder which has the specificity to be quick and easy to download. A QAR device receives data from the same data acquisition unit as DFDR. The

[7] TSO, Technical Standard Order, an FAA document used for specifying a minimum performance standard for specified materials, parts, and appliances used on civil aircraft.

Items	Test Condition
Impact shock	Half sine wave shock with a duration of 6.5 miliseconds and a peak acceleration of 3,400 g
Penetration resistance	Penetration force produced by a 227 kg weight that is dropped from a height of 3 m to strike the most critical point.
Static crush	Static crush force of 22.25 kN applied continuously for 5 minutes
High temperature fire	Minimum thermal flux of 158 kW/m^2 for a continuous period of at least 60 minutes. The nominal flame temperature should be 100 $°C$
Low temperature fire	Temperature of 260 $°C$ for a duration of 10 hours
Deep-sea pressure and seawater immersion	Seawater at a pressure of 60 MPa (equivalent to a depth of 6,000 m) for a period of 30 days.
Fluid immersion	The recording medium may not be damaged by immersion in fluids that may be encountered

Table 2.1: Flight recorder resistance specifications

characteristic of the data in terms of resolution, sampling rate, precision, and range also applies to the QAR data. Contrary to DFDR, this recorder has a removable recording medium such as a tape, an optical disk cartridge, a Personal Computer Memory Card International Association (PCMCIA) or a modern medium solid state which provides much higher data capacity and reliability [5]. The QAR recorder is located in a suitable place in the aircraft where it can be easily collected. Locations such as the avionics bay or a cockpit area are the places where a QAR device is usually located [6]. Furthermore, as QAR is not designed to survive an accident, it is mainly utilized for improving operational efficiency and flight safety [14] by the aircraft operator. In the context of airlines, this activity is usually called as Flight Data Monitoring (FDM) or Flight Operational Quality Assurance (FOQA) which is covered in Section 2.3.

2.2.1 Data Acquisition

A Data Acquisition Unit (DAU) plays a significant role in collecting data from sensors and sending them to the recorder devices in an aircraft system. To perform data transmission between devices, the DAU uses protocols specified by the ARINC[8] standard. Airbus nomenclature calls the DAU as the Flight Data Interface and Management Unit (FDIMU)[9] which

[8] ARINC stands for Aeronautical Radio Incorporated, a company established in 1929 by four major airlines that develop and operates systems and services to ensure the efficiency and performance in the communication in the aviation industry.

[9] In Boeing context, this device is called Flight Data Acquisition Unit - FDAU.

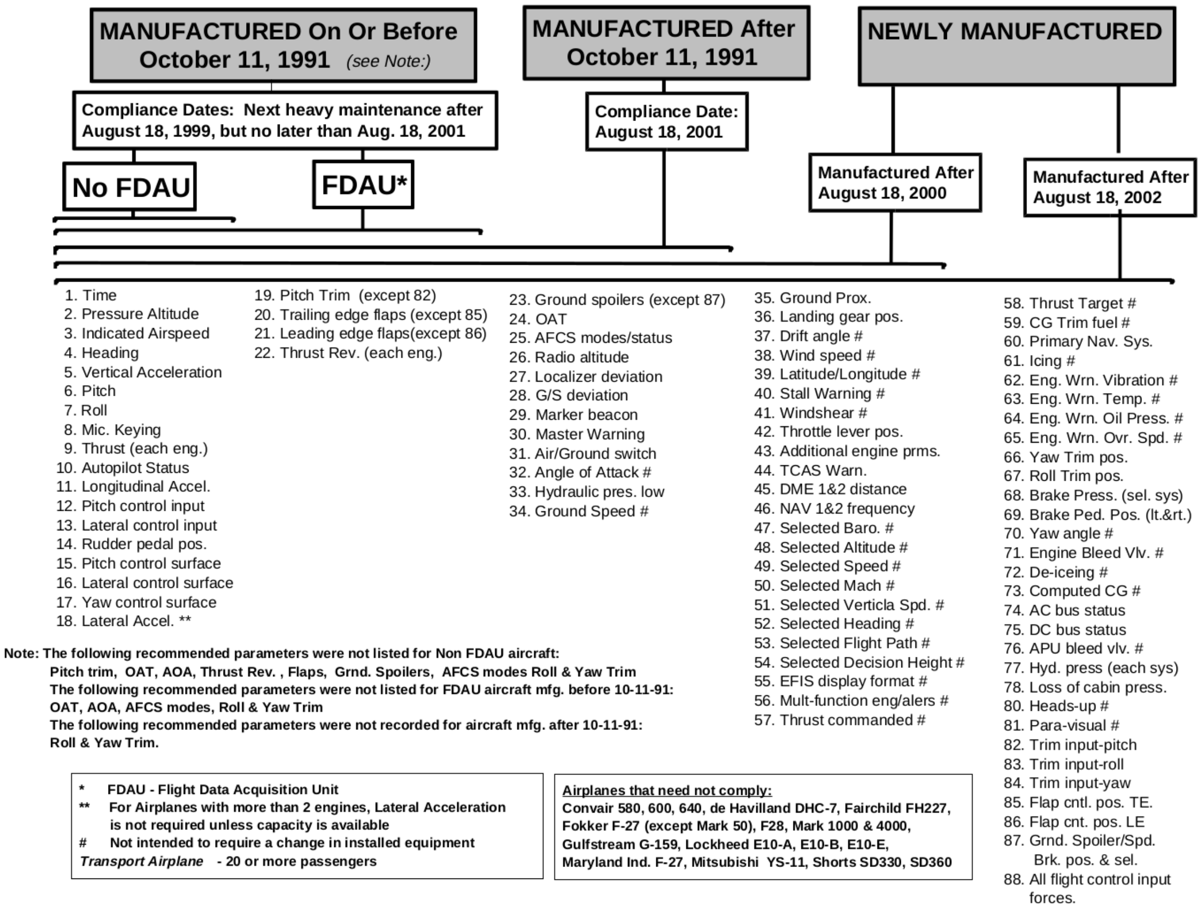

MANUFACTURED On Or Before October 11, 1991 (see Note:)
MANUFACTURED After October 11, 1991
NEWLY MANUFACTURED
Compliance Dates: Next heavy maintenance after August 18, 1999, but no later than Aug. 18, 2001
Compliance Date: August 18, 2001
Manufactured After August 18, 2000
Manufactured After August 18, 2002
No FDAU
FDAU*
1. Time
2. Pressure Altitude
3. Indicated Airspeed
4. Heading
5. Vertical Acceleration
6. Pitch
7. Roll
8. Mic. Keying
9. Thrust (each eng.)
10. Autopilot Status
11. Longitudinal Accel.
12. Pitch control input
13. Lateral control input
14. Rudder pedal pos.
15. Pitch control surface
16. Lateral control surface
17. Yaw control surface
18. Lateral Accel. **
19. Pitch Trim (except 82)
20. Trailing edge flaps (except 85)
21. Leading edge flaps(except 86)
22. Thrust Rev. (each eng.)
23. Ground spoilers (except 87)
24. OAT
25. AFCS modes/status
26. Radio altitude
27. Localizer deviation
28. G/S deviation
29. Marker beacon
30. Master Warning
31. Air/Ground switch
32. Angle of Attack #
33. Hydraulic pres. low
34. Ground Speed #
35. Ground Prox.
36. Landing gear pos.
37. Drift angle #
38. Wind speed #
39. Latitude/Longitude #
40. Stall Warning #
41. Windshear #
42. Throttle lever pos.
43. Additional engine prms.
44. TCAS Warn.
45. DME 1&2 distance
46. NAV 1&2 frequency
47. Selected Baro. #
48. Selected Altitude #
49. Selected Speed #
50. Selected Mach #
51. Selected Verticla Spd. #
52. Selected Heading #
53. Selected Flight Path #
54. Selected Decision Height #
55. EFIS display format #
56. Mult-function eng/alers #
57. Thrust commanded #
58. Thrust Target #
59. CG Trim fuel #
60. Primary Nav. Sys.
61. Icing #
62. Eng. Wrn. Vibration #
63. Eng. Wrn. Temp. #
64. Eng. Wrn. Oil Press. #
65. Eng. Wrn. Ovr. Spd. #
66. Yaw Trim pos.
67. Roll Trim pos.
68. Brake Press. (sel. sys)
69. Brake Ped. Pos. (lt.&rt.)
70. Yaw angle #
71. Engine Bleed Vlv. #
72. De-iceing #
73. Computed CG #
74. AC bus status
75. DC bus status
76. APU bleed vlv. #
77. Hyd. press (each sys)
78. Loss of cabin press.
80. Heads-up #
81. Para-visual #
82. Trim input-pitch
83. Trim input-roll
84. Trim input-yaw
85. Flap cntl. pos. TE.
86. Flap cnt. pos. LE
87. Grnd. Spoiler/Spd. Brk. pos. & sel.
88. All flight control input forces.
Note: The following recommended parameters were not listed for Non FDAU aircraft:
Pitch trim, OAT, AOA, Thrust Rev. , Flaps, Grnd. Spoilers, AFCS modes Roll & Yaw Trim
The following recommended parameters were not listed for FDAU aircraft mfg. before 10-11-91:
OAT, AOA, AFCS modes, Roll & Yaw Trim
The following recommended parameters were not recorded for aircraft mfg. after 10-11-91:
Roll & Yaw Trim.
* FDAU - Flight Data Acquisition Unit
** For Airplanes with more than 2 engines, Lateral Acceleration is not required unless capacity is available
# Not intended to require a change in installed equipment
Transport Airplane - 20 or more passengers
Airplanes that need not comply:
Convair 580, 600, 640, de Havilland DHC-7, Fairchild FH227, Fokker F-27 (except Mark 50), F28, Mark 1000 & 4000, Gulfstream G-159, Lockheed E10-A, E10-B, E10-E, Maryland Ind. F-27, Mitsubishi YS-11, Shorts SD330, SD360

Figure 2.6: Mandatory parameters as recorded in FDR data [59]

is a combination of two separated units from the previous generation of Airbus aircraft. The first unit is called Flight Data Interface Unit (FDIU), which serves as a data acquisition unit for mandatory parameters; the second unit is the Data Management Unit (DMU) for collecting non-mandatory parameters. The FDIMU device outputs the mandatory parameters into DFDR and QAR recorders while the non-mandatory parameters are stored in the Digital ACMS Recorder (DAR) data. The data streaming begins when the FDIMU receives parameters from sensors on the ARINC 429 protocol and condenses the data into a multiplexed digital data stream on the ARINC 573/717 protocol. The condensed data are then systematically sent to the recorder devices. However, the transmission is different between DFDR and QAR/DAR. In a DFDR, the transmission is in a bidirectional bus in which the data sent to DFDR device are played back for verification and synchronization, while the QAR/DAR transmission is in a unidirectional bus where data recording errors are checked internally in the recorder [6].

Furthermore, the device in the FDIMU which sends the mandatory data (FDIU)[10] cannot be programmed, so the number of parameters and data characteristics such as sampling rate is fixed. In comparison, the DMU is programmable, which allows the aircraft operator to extend the amount of recorded data. The DMU also provides a reporting system that can be

[10] FDIU stands for Flight Data Interface Unit.

generated during a flight and accessed during the flight or later on the ground. Both QAR and DAR data are used for Flight Data Monitoring program. The typical airborne data acquisitions and protocols of a commercial aircraft are shown in Figure 2.7.

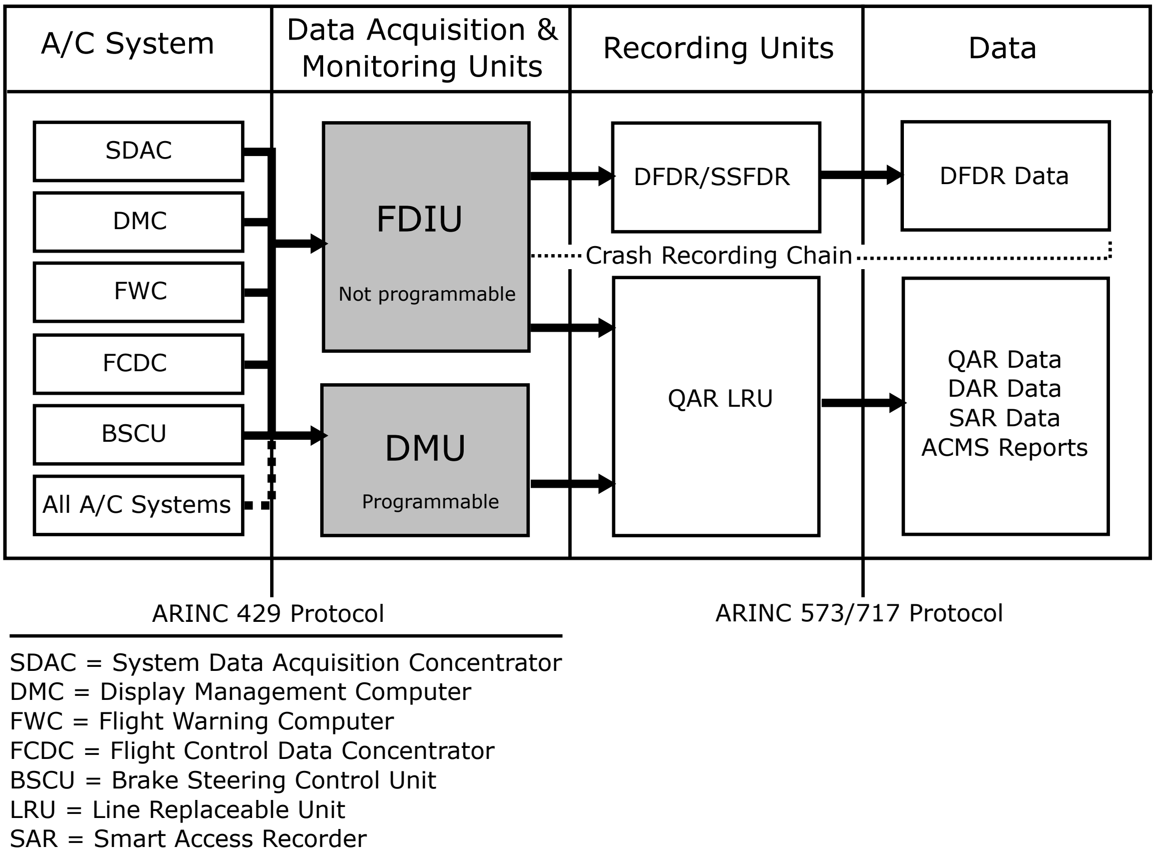

A/C System
Data Acquisition & Monitoring Units
Recording Units
Data
SDAC
DMC
FWC
FCDC
BSCU
All A/C Systems
FDIU
Not programmable
DMU
Programmable
DFDR/SSFDR
Crash Recording Chain
QAR LRU
DFDR Data
QAR Data
DAR Data
SAR Data
ACMS Reports
ARINC 429 Protocol
ARINC 573/717 Protocol
SDAC = System Data Acquisition Concentrator
DMC = Display Management Computer
FWC = Flight Warning Computer
FCDC = Flight Control Data Concentrator
BSCU = Brake Steering Control Unit
LRU = Line Replaceable Unit
SAR = Smart Access Recorder

Figure 2.7: Mandatory and non-mandatory parameters recording chain [60]

2.2.2 On-board Transmission and Recording Protocols

The FDIMU uses the ARINC standards for data transmission among avionics systems in the aircraft. The most common protocol and still widely found in many commercial aircraft is the ARINC 429 protocol [61]. It was first published in 1977 to cope with the issues found in Mil-Std 1553. The main hurdle was the stringent requirements in Mil-Std 1553 to be overly complex for used in commercial aviation, which leads to the high-cost development and difficulty in the certification process [62]. Then, ARINC 429 was introduced for commercial transport airplanes to reduce the complexity of the Mil-Std 1553 standard. Since then, ARINC 429 has become the industry standard that provides protocols which exhibit a high level of efficiency, ease of certification, and reliability. The ARINC 429 point-to-point protocol is the simplest one among the other existing aircraft data bus protocols such as ARINC 629 and Avionics Full-duplex Ethernet (AFDX). ARINC 629 is a shared data bus protocol which was introduced by Boeing and has been used on the B777 aircraft. The latest and the most advanced protocol is the Avionics Full-duplex Ethernet, which is also known as ARINC 664. This protocol is based on Ethernet technology which was introduced on the A380, A350 and

B787 aircraft. With the advancement in avionics technology, AFDX has proved to provide a better performance and flexibility without losing the compliance with the safety, redundancy, and the reliability of avionics requirements [63].

The format in which the data are stored in the onboard recording units is specified by different protocols. The most common one is the ARINC 573/717 protocol. The latest protocol which serves the same functionality as ARINC 573/717 but with enhanced features is the ARINC 767. Smiths Aerospace developed the ARINC 767 protocol based on the US Air Force's Standard Flight Data Recorder (SFDR). The ARINC 767 can record data at any sampling rate [64] as well as handle more data types such as image/video, voice, and normal raw data. The ARINC 767 is also known as the Enhanced Airborne Flight Recorder (EAFR) which was introduced in B787 aircraft [65]. However, ARINC 717 is the most commonly found protocol in current commercial transport airplanes. Furthermore, the data used in this dissertation are also decoded based on the ARINC 717 standard. As the ARINC 537 is superseded by ARINC 717 which is used to perform the same functionalities, the rest of the section only pertains to ARINC 717. For data transmission protocol, this dissertation only covers ARINC 429 with the same reason as explained for ARINC 717.

ARINC 429 Protocol

ARINC 429 defines protocols, electrical, and data characteristics between various avionics equipment and aircraft systems. This protocol is interconnected with wires in twisted pairs in which the communication is performed through a unidirectional data bus standard. Such one-way flow of transmission locates a transmitter and a receiver on a separate port and leads to the integrity and low-cost installations [62]. This type of transmission is popularly known as the Mark 33 Digital Information Transfer System (DITS) [66]. ARINC 429 is the most popular bus standard used in commercial aircraft, including Airbus A310, A320 and A330/340; Boeing 737, 747, 757, and 767; and McDonnell Douglas MD-11 [61]. The following section presents three aspects related to the ARINC 429 protocol, including electrical characteristics, data word format, and data types.

ARINC 429 Electrical Characteristics

ARINC 429 uses two signal wires to transmit 32-bit words in two available speeds depending on the specification of the bus. A signal at low speed is transmitted at 12.5 $kbits/sec$ while at high speed, the signal is transmitted at 100 $kbits/sec$. The transmission of the sequential words is separated by at least four-bit times of zero (NULL) voltage eliminating the need for a separate clock signal which makes ARINC 429 has its own self-clocking system. The bits are transmitted by employing bipolar return to zero (BPRZ) consisting of three states, namely [61]:

- HI or 'A' or '+' is specified within the range of $+7.25V$ to $11V$
- NULL which should be within $+0.5V$ to -0.5V

Parameter	Value
Voltage level	$+5V$, $0V$, -$5V$ (each conductor with respect to ground)
Data encoding	Bi-Polar Return to Zero
Word size	32 bits
Bit rate (high)	100 $kbps$
Bit rate (low)	12.5 $kbps$
Slew rate (high)	1.5 ms ($\pm$0.5 ms)
Slew rate (low)	10 ms ($\pm$5 ms)

Table 2.2: ARINC 429 electrical characteristics [61]

- LO or 'B' or '-' is specified within the range of -7.25V to -11V

With this configuration, the information is received by measuring the voltage difference between the two wires, see Figure 2.8. The nominal transmission voltage is 10V ± 1V (differential) with either positive or negative polarity. Thus, each signal leg ranges between $+5V$ and -5V. When one leg is $+5V$, the other is -5V and vice-versa. The differential voltage at the receiver depends on the line length and the number of the receiver connected to each bus, which is limited to 20 receivers by the protocol. As each bus is a one-way flow (unidirectional), a system needs to have its own transmit bus if the system is required to respond to or to send messages. Thus, to achieve bidirectional data transfer, it is necessary to have two separate bus connections. With this configuration, implementing ARINC 429 in an aircraft that uses a sophisticated avionics system increases data bus complexity which at the end increases the overall weight as a large number of cables may be required by the avionics systems. Table 2.2 summarizes the electrical characteristics of the ARINC 429.

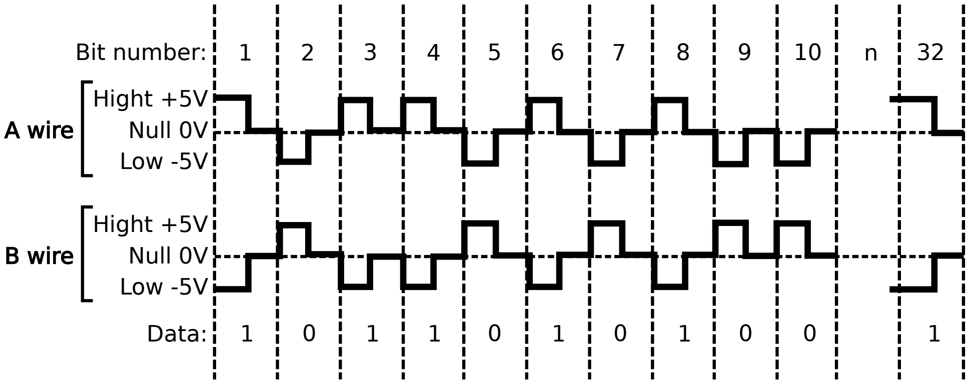


Figure 2.8: Signals presentation on the twisted pair conductors in the ARINC 429 [61]

ARINC 429 Data Word Format

The ARINC 429 message consists of a single 32-bit data word and typically uses the format shown in Figure 2.9. The 32-bit data word consists of 5 primary fields: Parity (P), Sign/Status Matrix (SSM), Data, Source/Destination Identifiers (SDI), and Label. The ARINC 429 data word is always in 32 bits, and any unused bits are padded with zeros. By convention, bit "1" is the least significant bit (LSB) and the bit "32" is the most significant bit (MSB). In the ARINC 429, the transmission order is started with the least significant bit (LSB) of each byte, except for the Label field where the most significant bit is transmitted first. The 32-bit word of ARINC 429 structure is shown in Figure 2.9. Each of the field is described as follows:

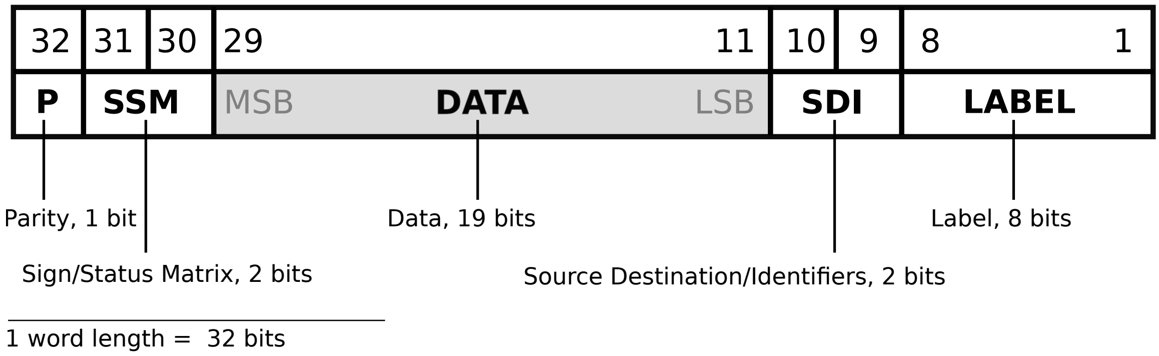


Figure 2.9: ARINC 429 data word format [61]

- Parity, 1 bit
  ARINC 429 uses an odd parity (P) bit which is located in bit 32 or the Most Significant Bit (MSB) position. Consequently, the parity bit is the last bit transmitted within the data word. ARINC uses odd parity as an error check to ensure accurate data reception by setting the parity bit to "0" when there is an odd number of 1 bit from bit 1 to 31 and to "1" when it is even. However, the parity is not intended for data error correction; it is used only for data error detection.
- Sign/Status Matrix (SSM), 2 bits
  Sign/Status Matrix (SSM) field is located in bits 30 and 31 (2 bits). This field provides different information depending on the type of data being transmitted as encoded in the Label field. Typically, the information provided in the SSM is represented in a code form. For Binary Coded Decimal (BCD) data, it may contain information such as direction and sign, see Table 2.3. When the type of the transmitted data is binary, the SSM field is used to indicate the status of the operated equipment and the information it may have including failure warning, no computed data, functional test, and normal operation.
- Data, 19 bits
  Bits 11-29 contains the actual data to be sent in a specific data format, e.g., binary, discrete, and other standard data formats supported in ARINC 429. These data formats are covered in the following section along with the corresponding examples. Usually, not all bits are used; only those necessary to cover the range and resolution of the information transmitted. ARINC 429 transmits data words in LSB-MSB sequence except for the Label field. The order of transmission follows Label (MSB first), SDI, Data, and the remaining of the bit fields (LSB first).

Bit no.		
31	**30**	**BCD data SSM sign coding**
0	0	Plus, North, Right, To, Above
0	1	No computed data
1	0	Functional test
1	1	Minus, South, West, Left, From, Below

Table 2.3: Example of SSM codes for BCD data

- Source/Destination Identifiers (SDI), 2 bits
  The Source/Destination Identifier utilizes bits 9-10 and is optional under the ARINC 429 protocol. The SDI field contains information of the data source and the target receiver to which the data is meant for. In some specific cases, the SDI field (2 bits) may be used when a datum requires more bits for storing high-resolution data. In a case when the SDI is used as an identifier, it is interpreted as an extension to the word Label [67].
- Label, 8 bits
  Field Label is registered in bits 1-8 and is used to identify the word's data type such as binary (BNR), binary coded decimal (BCD), discrete, and other supported data types. This field may also contain instruction or data reporting information. The Label is expressed as a three-digit octal number with receivers programmed to accept up to 255 Labels. The labels are listed in the ARINC specifications; for example, BNR Label 102 is corresponding to the Selected Altitude parameter. In contrast with the ARINC 429 data transmission order, Label data are transmitted with LSB first and followed by the rest of bit to the MSB bit.

ARINC 429 Data Types

The data transmitted in the ARINC 429 protocol can be classified into several types. For each type, there is a specific format of encoding the data in the 32-bit words. The data types and their formats are described as follows:

- BNR Data
  Binary, or BNR, stores data as a binary number. The sign bit is located in bit 29, with a "1" representing a negative number in the decoded format. It may also indicate South, West, Left, From or Below, depending on the data being decoded. The remaining bits (bit 28 - 11) contain the actual data of the respective parameter. The structure of the ARINC 429 in the BNR data format is shown in Figure 2.10.

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
P	**SSM**		MSB									**DATA**							LSB			**SDI**		**LABEL**							

Figure 2.10: ARINC 429 BNR data word structure

- BCD Data
  Binary Coded Decimal or BCD format divides bits into five groups; in the first four groups, each group contains 4 bits while the fifth group contains only 3 bits of data. The actual data are represented by each decimal digit in the 4-bit group, or if required, the 3-bit group can be utilized to provide the fifth binary value. The sign of the BCD is encoded in SSM field. The structure of this data format is depicted in Figure 2.11.

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
P	**SSM**		**Digit 1**			**Digit 2**				**Digit 3**				**Digit 4**				**Digit 5**				**SDI**		**LABEL**							

Figure 2.11: ARINC 429 BCD data word structure

- Discrete Data
  Discrete data format represents specific equipment conditions such as True/False, Up/-Down, Pass/Fail, Air/Ground, Activated/Not Activated, or other operational conditions of systems or subsystems. The discrete value can be made up of binary or binary coded decimal or as individual bits.
- Maintenance Data and Acknowledgement
  Two-way communication between the source and the sink is utilized in the Maintenance Data and Acknowledgement. However, as ARINC 429 only provides a unidirectional transmission, two ARINC channels are required for line replaceable units (LRU) for sending and receiving data. Other maintenance messages typically require exchanging a sequence of messages and often utilize a bit oriented protocol such as the Williamsburg/Buckhorn protocol [67].
- Williamsburg/Bukhorn Protocol
  Williamsburg/Buckhorn is a bit-oriented protocol. It is mainly used to transfer files across the ARINC buses. File transfer protocols are necessary when more than 21 bits of data are required to be transmitted [67].

ARINC 717 Protocol

The ARINC 717 protocol was created to specify the data format output for use in Flight Data Recorder and Quick Access Recorder data. The FDIMU processes the incoming data, formats them following the ARINC 717 standard, and stores them in FDR/QAR recorders. Generally, there are four types of input data processed by the FDIMU [52]:

- Discrete (logical status detection, indicators, switches, relays);
- Analog (potentiometer);
- Synchronization transmitters;
- Digital bus (ARINC 429).

These four data types are transmitted in a continuous data stream of Harvard Bi-Phase in 12-bit words and organized by following data structure specified in the ARINC 717 protocol [68]. ARINC 717 specifies data structure into a frame which is repeated every 4 seconds consisting of 4 sequential subframes. Furthermore, 16 frames are joined into one superframe and repeated every 64 seconds. Table 2.4 summarizes the ARINC 717 data structure.

Data Structure	**Description**
Bit	Smallest unit, can be 1 or 0
Word	12 bits are packed to 1 word
Subframe	Subframe is recorded in every 1 second. It contains 64, 128, 256, 512, or 1,024 words (2^{n+6}, with n from 0 to 4)
Frame	One frame is composed of 4 subframes and repeated every 4 seconds
Superframe	One superframe is composed of 16 frames. A superframe patterns repeats every 64 seconds.

Table 2.4: ARINC 717 data structure

A word contains 12 bits and sequentially numbered from 1 to 12. Bit "1" corresponds to the Least Significant Bit (LSB) and is transmitted first; bit "12" is the Most Significant Bit (MSB) and is the last bit to be transmitted. Using the 12-bit word means that each word potentially can have 4,096 ($=2^{12}$) possible states, having a range between 0 to 4,095 counts. Furthermore, a resolution, the smallest change in a parameter value that can be recorded, is computed by the range of a parameter (specified for each parameter) divided by the number of possible states. With the ARINC 717 data format, the sampling rate is defined by occurrences of parameter recorded in every subframe. For example, if one parameter is stored two times in every subframe, it corresponds to 2 Hz of sampling rate since every subframe is recorded in every second, see Table 2.4. Thus, for each data sent, the FDIMU reorganizes it into word, subframe, frame, and superframe correspondingly and stores it as binary data in the recorder units. To decode this binary data into engineering unit, the process mentioned above has to be reversed, and it involves two different steps. In the first step, the binary data needs to be reorganized back to the data structure specified in the ARINC 717. This step is followed by the second step of finding the parameter location in each datum structure by using a so-called data frame layout (DFL) document. Both of these steps are described below.

Finding Data Structure
ARINC 717 defines two types of words, one that contains information for data synchronization, and one which represents the actual data information. A sync word, as suggested by its name, is used to find the pattern of the binary data. The sync word is a constant and unique number stored in a 12-bit word. Each subframe contains its unique sync word which is located in the first word in each subframe. Thus, there are four unique sync words defined to find the data pattern. These four sync words are shown in Table 2.5.

Subframe	Binary (MSB-LSB)	Octal	Decimal	Hexadecimal
1	001001000111	1107	583	247
2	010110111000	2670	1464	5B8
3	101001000111	5107	2631	A47
4	000111011011	6670	3512	DB8

Table 2.5: Subframe sync words [69]

Constructing data pattern is started by finding the sync word in the binary data stream, once the 1st sync word found, the 2nd sync word is usually jumped to the n-th[11] word. The third and fourth sync words are also found in the same way until subframe 1 (sync word 1) is found again in the next frame. This process is repeated to construct the second frame until all bits are constructed into the data frame pattern. However, in most cases, not all bits can be organized into the data frame structure. Typically, some bits at the beginning and the end of the data are discarded as they usually contain data from the previous flight.

Up to this point, the data frame has been constructed and restored. The next step is to find a superframe pattern in the data. In contrast to the frame, the superframe structure is found by a continuous parameter called a frame counter. Table 2.4 exhibits that a superframe is composed of continuous 16 frames and repeated itself in the same pattern. Thus, a continuous frame counter is required to locate a superframe. The frame counter itself is located in a 12-bit word in one of the subframes (or 1 word in every frame), meaning that the frame counter ranges from 1 to 4,096 counts. By using the modulo (divided by 16) operator, the frame counter of the respective frame can be retrieved. The process of constructing the data frame structure from a binary data stream to the ARINC 717 structure is depicted in Figure 2.12.

Data Frame Layout (DFL)

Having constructed the binary streams into the data frame structure, the second step required for the engineering data conversion is locating each parameter in the data frame structure. The location of each parameter in the data frame structure can be retrieved through a data frame layout (DFL) document. ARINC 717 provides this information for all data recorded in the flight recorder. Typically, the DFL document describes [52]:

- Properties of a parameter in terms of the data frame structure, including bit location, number of bits used to encode parameters, scale factor, offset, word location, subframe location, and parameter type.
- Function to convert bits into engineering values.

[11] It can be 64, 128, 256, 512, 1024 depending on the specification used in the equipment and recording device.

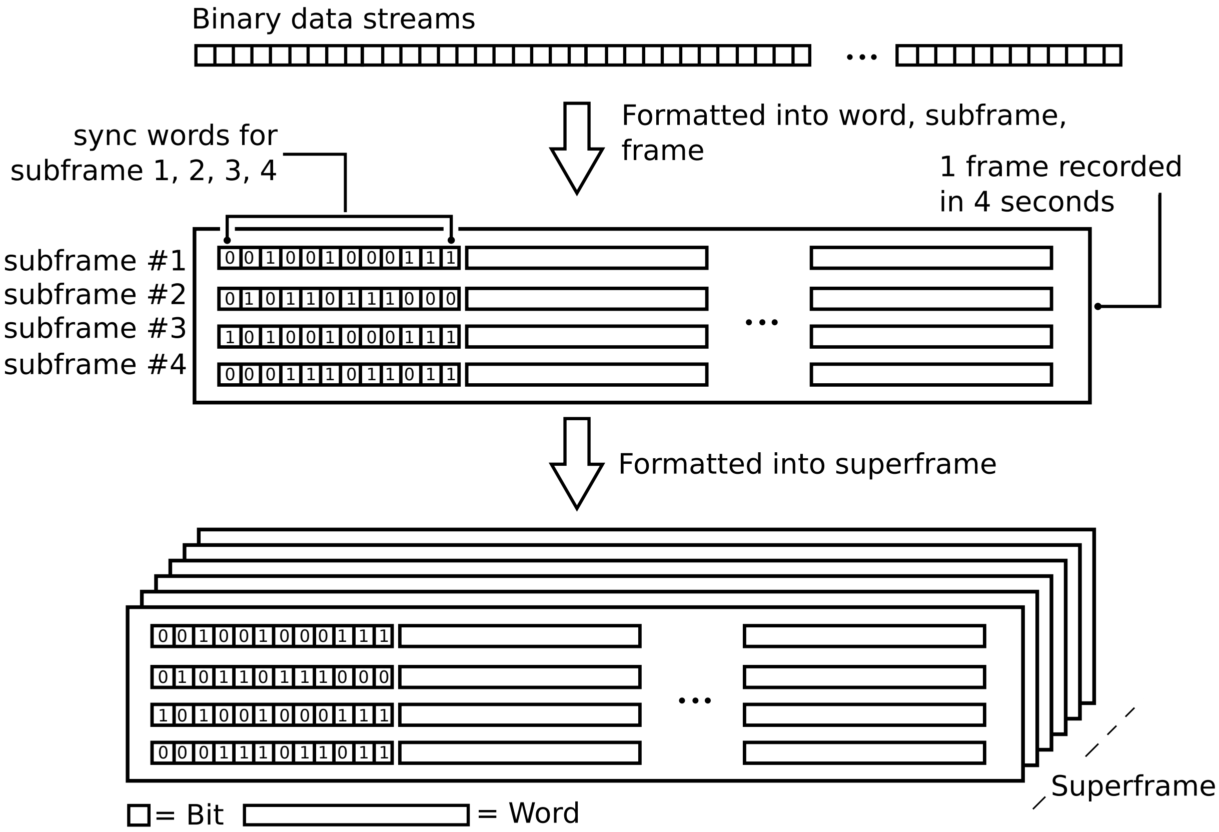


Figure 2.12: Binary streams to data frame structure

As mentioned in the previous section, ARINC 717 defines two type words, i.e., a sync word and a data word. A sync word provides information used for data frame construction, whereas a data word is used for storing the actual data. One example of a data frame layout for airspeed parameter is shown in Table 2.6. Furthermore, ARINC 717 classifies parameters into several different types, and each type defines how a parameter is decoded into the engineering unit, see Table 2.7. For example, the airspeed parameter shown in Table 2.6 belongs to LINEAR category and the formula used for decoding the parameter is a linear function in which the scaling factor and offset are provided in the Data Frame Layout (DFL) document.

2.2.3 Data Transfer

In a normal operation, airlines download the QAR data to the ground station once the aircraft lands at the destination airport or download them periodically depending on the capacity of the storage of the QAR device used in the aircraft. The QAR data are transferred to the airlines' hub through several ways depending on the technology used onboard the aircraft. Typically, there are three common ways that are widely practiced by airlines [14]:

- Ground-Based Transportation. QAR media is replaced at the end of each day or after several days, depending on the media capacity, data recovery strategy, or the urgency of using the data for analyzing a specific event or incident. The media are then sent to airlines' major hub by using a regular mail system or a company mail system. Typically,

Parameter: Airspeed			
Word(s): 19	Bits:12-1	Rate: 1 Hz	Subframe: 1, 2, 3, 4
Conversion Factor:			
Value	Decimal	Eng. Units	
Minimum	0	0 kt	
Mid point	2047	512 kt	
Maximum	4095	1024 kt	
Engineering unit conversion equation: LINEAR			
$Y = A_0 + A_1 X$	where,	Y: Output in engineering unit	
$A_0 =$ 0.0		X: input in decimal	
$A_1 =$ 0.25006			

Table 2.6: Example of data frame layout [52]

a tracking system is also used along with this classical transfer method. Such a tracking system ensures that the recording medium can be verified with respect to data storage ID, time, location, and aircraft flight number.

- Electronic Transmission. This method uses download equipment that interfaces with the aircraft or by removing the storage media from the onboard system as in the case of Ground-Based Transportation. In contrast to the Ground-Based Transportation method, Electronic Transmission sends the QAR data from the remote maintenance locations to the airlines' major hub station where the data will be processed. While this method is more efficient than the first method, it requires a larger capital outlay and sufficient data transmitting capability. As the data are transferred electronically, the data security protection must also be equipped with this transfer method.
- Wireless Transmission. The wireless transmission is an advanced method for QAR data transfer. Such a method enables QAR data to be downloaded from an onboard storage facility device (e.g., a wireless QAR) to an operator's file server. The system can either use a short range transmission to an airport-based local area network or a mobile phone technology. The latter allows airlines to collect the QAR data from their aircraft's landing on airports around the world. Typically, the system onboard transfers encrypted QAR data to an FDM data server ready for automated processing once the aircraft is parked and the exit door is opened. Since the download is accomplished automatically, it removes the requirements for maintenance involvement and reduces logistical problems related to the movement of media/physical downloading tasks. Typically, the QAR wireless transmission system consists of three main components: recording device + data sender, transmission protocol, and ground-based data receiver.

Parameter Type	Description
Packed bits	An array of bits and is processed without any conversion or changes.
Discrete	Contains only one bit per parameter and decoded as Boolean.
Linear	An array of bits or a composed of bits from different position. Formula to decode y = scale-factor $\cdot\, x +$ offset, where x is input in decimal and y is corresponding output (engineering value).
Polynomial	The same as linear parameter except the conversion is done through second order polynomial function as, $y = a \cdot x^2 + b \cdot x + c$, where a, b, c are constants, x is input in decimal and y is the decoded value (engineering value).
Segmented	Segmented parameter is joined and decoded in the same way as linear parameter, the only difference is precision and offset are varied.
Binary Decoded Decimal	A BCD parameter requires 4 bits in order to display one decimal with the range of $0 - 9$. Usually BCD parameter is used to record frequency of an instrument, for example VOR frequency recorded in QAR data is decoded as BCD parameter.
ASCII	An ASCII parameter is based on a 7-bit array and converted according to the ASCII code.
Time and Date	Time and date may be decoded through one of two options below: • Bits are constructed as linear parameter by separating the date into year, month, day, hour, minute, and second. The conversion into time/date is done separately for each item and each part is joined together to construct normal time format. • Bits are constructed as BCD parameter and conversion into time/date is done through BCD parameter rule.

Table 2.7: ARINC 717 parameter type

These three components are depicted in Figure 2.13.

2.3 Flight Data Monitoring

EASA ORO.AOC.130 document [4] defines a Flight Data Monitoring (FDM) or Operational Flight Data Monitoring (OFDM) program as a proactive and non-punitive program for gathering and analyzing data recorded during routine flights to improve aviation safety. The terminology of FDM or OFDM is commonly used in the European region. A different term in a different region for such a program may be used, e.g., Flight Operations Quality Assurance (FOQA) term is used in North America [71]. In the flight operational context, FDM can be seen as a part of a quality assurance process. While in a broader context, it provides a vital

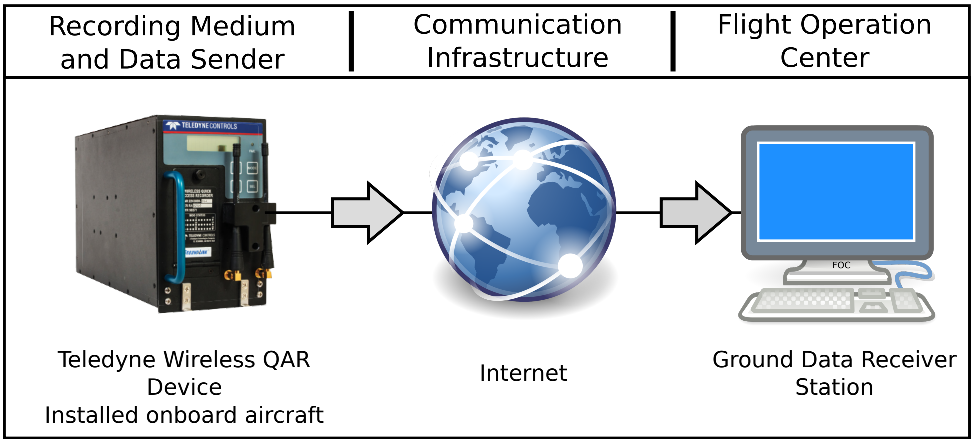

Recording Medium and Data Sender
Communication Infrastructure
Flight Operation Center
TELEDYNE CONTROLS
FOC
Teledyne Wireless QAR Device
Installed onboard aircraft
Internet
Ground Data Receiver Station

Figure 2.13: Wireless QAR system, adapted from [70]

contribution to airline safety management system. FDM is performed by individual airlines on a daily routine basis involving several activities such as downloading flight data recorded onboard the aircraft, analyzing the data, and providing corrective actions in the context of operational areas. Specifically, the FDM program enables the aircraft operator to track and evaluate its individual flight operation trends, identify risk precursors, and take the appropriate remedial action. In the 1990s, only a few countries, including France and India, mandated FDM as part of their airline Safety Management System (SMS). However, on January 1st, 2005, the ICAO introduced Amendment 26 to ICAO Annex 6 – "Operation of Aircraft," which mandated airlines to implement an FDM program as part of its accident prevention and flight safety program.

Since then, there has been significant adoption of the FDM as a mandatory requirement in most countries, with the notable exception of the USA, where the FDM has been introduced as one of the voluntary safety initiatives [14]. In the European level, this regulation is adopted by EASA in EU-OPS 1.037 "Accident Prevention and Flight Safety Programme." The flight data used within FDM - usually known as 'FDM data' - is a combination of Quick Access Recorder (QAR) and Digital ACMS Recorder (DAR) data. The combination provides more data compared to the crash-protected FDR data as the DAR allows airlines to program their own data frame in which a new parameter can be derived [72]. The implementation of the FDM program as part of the airlines Safety Management System has the following goals [4, 5]:

- To identify areas of operational risk and quantify current safety margins.
- To identify and quantify changing operational risks by highlighting when non-standard, unusual, or unsafe circumstances occur.
- To use the FDM information on the frequency of occurrence, combined with an estimation of the level of severity, to assess the risks and to determine which are or may become unacceptable if the discovered trend continues.
- To put in place appropriate risk mitigation actions to provide remedial action once an unacceptable risk, either actually present or predicted by trending, has been identified.

- To confirm the effectiveness of any remedial action by continued monitoring.

The FDM is a closed loop system in which flight operations' risks are continuously identified and quantified on a daily basis routine. This step is followed by a remedial action once the quantified risks are not acceptable within operational ranges. As this process is performed continuously, it allows airlines to evaluate the effectiveness of the preventive action. This closed loop FDM process is depicted in Figure 2.14. In short, an FDM system allows airlines to compare their Standard Operating Procedures (SOPs) with those actually achieved in daily flights and to provide feedback to be taken as a corrective action where safety may be compromised by significant deviations from the airlines' SOPs. The airlines use commercial FDM software to implement the FDM program in their system. Some of the FDM programs used by world-wide airlines are listed below.

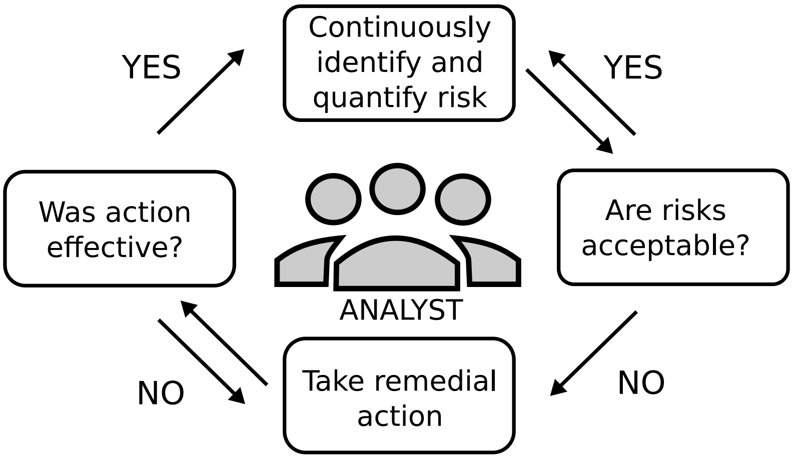


Figure 2.14: FDM as a close loop system

- Aerobytes from Aerobytes
- AirFase by Teledyne Controls
- Analysis Ground Station (AGS) by Safran
- Event Measurement System (EMS) by GE Aviation
- FlightScape by CAE
- POLARIS by Flight Data Services

Such FDM software provides many features, including automatic data retrieval through wireless technology [70], data decoding, data visualization, and built-in analysis techniques such as exceedance detection and statistical analysis. The FDM software also allows the aircraft operators to add additional features such as implementing an airline specific algorithm on flight data or modifying the existing algorithm to comply with the airlines' SOPs. FDM software becomes a standard and useful tool for airlines in implementing the FDM program as part of their safety management system. UK Civil Aviation Authority in CAP 739 [5] noted several techniques which airlines can use when analyzing the flight data.

- Exceedance or event detection

  Exceedance/event detection or also called as a threshold analysis, is the most common

technique used in an FDM system. This technique works by comparing the daily operational flight with airlines' Standard Operator Procedures (SOPs) or flight manual limits. Any deviation from SOPs or any references used is marked as an event. A set of core events over the years have been identified and collected by airlines and become the standard across the airlines. Typically, event detection is performed on a single flight. Thus, any detected event is usually addressed to parties involved in that flight, e.g., flight crew, maintenance crew.
Example events: Hard landing, low approach speed, high rate descent, low go-around, rejected takeoff, slow climb-out.

- All-flights measurements
  In contrast to event detection, all-flight measurements or routine data measurements are carried out on all flights within a certain period of time. The result of this technique usually can be used to get a broad picture of flight operations within an airline. Through this technique, an airline is able to monitor trends and tendencies before the trigger levels are reached.
  Example measurements: take-off weight, landing weight, touchdown distance, maximum braking used, landing gear retraction, and extension speeds.
- Statistics
  Statistical analysis in terms of a number of data used in the analysis is the same as All-flight measurement technique. The difference is that statistical analysis provides a summary of the occurrence of events within a certain period of time.
  Example of statistical analysis: top 10 events, top 10 events at each airport, top events trends.
- Investigation of incidents flight data
  An FDM program can also be used to investigate incidents of flight data. Records from previous flights or any relevant information collected in the FDM system may provide indications of system status and performance which may help in determining cause and effect relationships.
  Example of incidents where FDM data could be useful: vortex wake encounters, all flight control problems, Terrain Collision Avoidance System (TCAS) or Ground Proximity Warning System (GPWS) triggered maneuvers, unstabilized and rushed approaches.
- Continuing airworthiness
  The output from all-flight measurement and exceedance detection can be used to support the continuing airworthiness function. For example, engine-monitoring program looks at measures of engine performance to determine operating efficiency and predict impending failures [4].
  Examples of continuing airworthiness use: engine thrust level and airframe drag measurement, flying control performance, brake and landing gear usage.

2.4 QAR Data Characteristics

The characteristics of the QAR data presented in this section are related to the use of data in the system identification field. The properties of the QAR data such as sampling rate, sensor information, and data quality in connection with the flight maneuvers are described in the following sections.

2.4.1 Sampling Rate

A sampling rate is defined as the rate at which the parameters are measured or recorded, usually expressed in Hz or samples per second. Depending on the parameters, QAR data are recorded in a different sampling rate, which varies between 0.25 - 8 Hz. In some cases, some parameters might be recorded at 16 Hz. Parameters that change significantly over time are recorded at a high sampling rate while those that change slowly over time are recorded at less than 1 Hz. As an example, the QAR data utilized in this dissertation have sampling rates shown in Table 2.8.

Parameter	Sampling (Hz)
Vertical acceleration	8
Lateral acceleration	4
Radio altitude	4
Ground speed	1
True airspeed	1
Longitudinal acceleration	1

Parameter	Sampling (Hz)
Body pitch rate	1
Body roll rate	1
Body yaw rate	1
Barometric altitude	1
Angle of attack	1
Mass	0.25

Table 2.8: Selected QAR parameters and sampling rates

In contrast, flight test data are sampled at a higher sampling rate. The typical sampling rate for flight data is at a frequency of higher than 20 Hz. Such a high sampling rate is motivated by the Shannon sampling theorem, which states that the minimum sampling rate should be twice of the system's frequency [73]. In terms of Nyquist frequency, this statement is formulated as,

$$f_s = 2 \times f_N \tag{2.4.1}$$

where f_s is capture frequency and f_N is the Nyquist frequency or the upper limit of a system's frequency. This theoretical minimum sampling rate is further elaborated in Figure 2.15. As shown in the figure, the continuous signal has a frequency at 1 Hz. Based on Shannon's theorem, a minimum sampling rate that enables one to construct the signal is 2 × 1 = 2 Hz, indicated by the circle mark in the upper figure. However, if the signal is shifted in time or equivalent by the phase angle of 90 deg and the signal is sampled again with the same time as before, it would yield all to zero samples, see the central plot in Figure 2.15. This figure clearly shows that sampling data at the theoretical minimum rate is not sufficient. In the worst case, sampling data at a lower rate than the system's frequency might result in a so-called aliasing [74], in which the resulted sample falsely identifies the correct signal frequency, see the lower plot of Figure 2.15. In practice, data is usually recorded at a much higher rate than the theoretical minimum sampling rate. Specifically, for flight test data, Jategaonkar recommends that a minimum sampling rate for flight test data should be around 25 Hz [18]. While Morreli even suggests that the flight test data should be recorded at a minimum of 50 Hz [75]. Furthermore, the aircraft rigid-body dynamics are mostly around 2 Hz, which requires a minimum sampling of 4 Hz [25]. Thus, the sampling rate recommended by Jategaonkar and Morelli provides a sufficient rate for recording flight test data meant for flight vehicle system identification. Given this fact, utilizing the QAR data for parameter estimation might raise some problems as most of the parameters are recorded very close to and even lower than the frequency specified by Shannon's theorem.

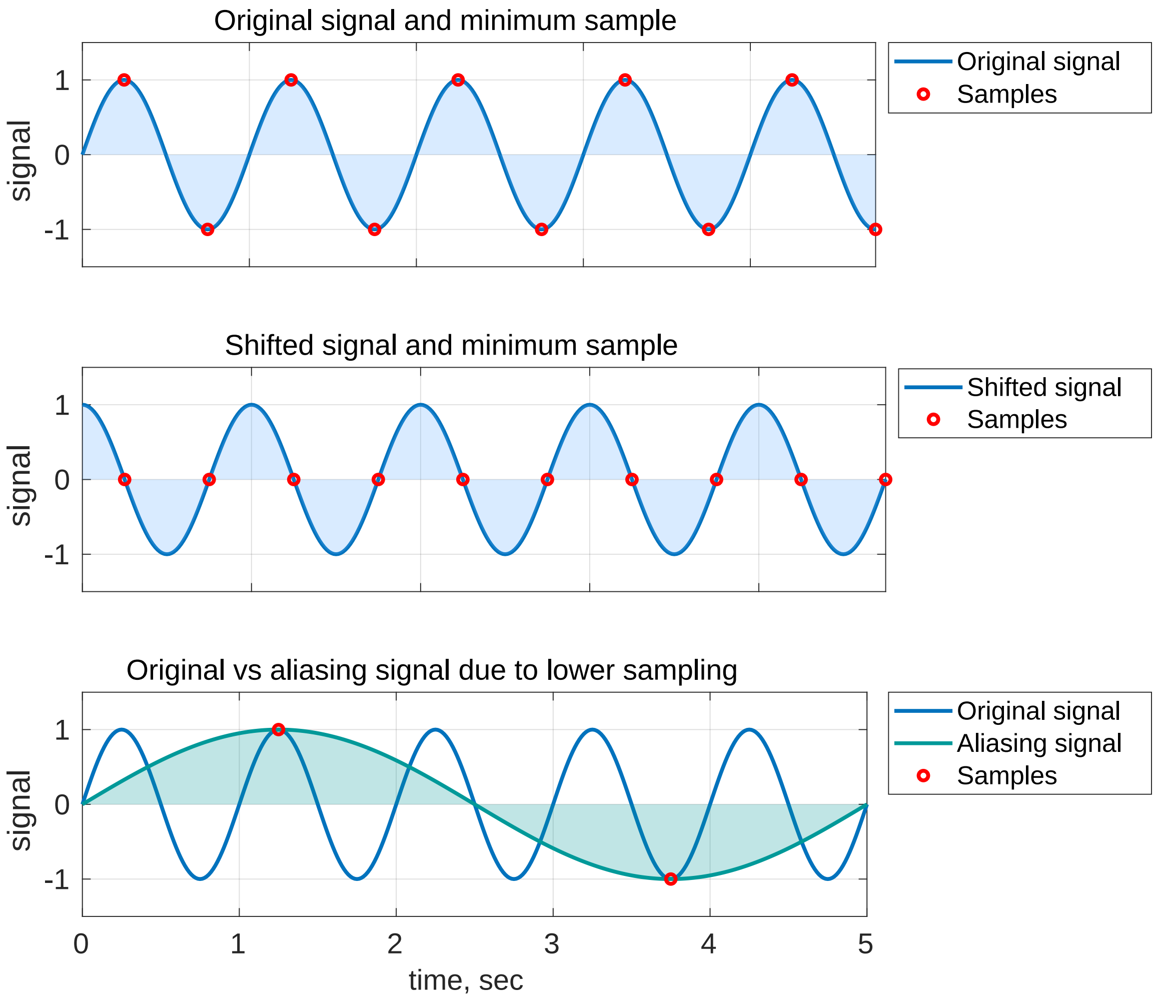


Figure 2.15: The Shannon's theorem of a theoretical minimum sampling rate

2.4.2 Low Information Content

The information content in the measured data depends highly on the treatments performed on the system during the data recording process. In the context of flight vehicle system identification, the treatments pertain to the designated flight maneuvers and control inputs. Typical flight maneuvers for system identification purpose including short period mode (short period oscillations), phugoid mode (longer period oscillations), level-turn, pushover-pullup, bank-to-bank roll, Dutch roll, thrust variation, and acceleration-deceleration are performed during flight test to excite a specific dynamic motion of the aircraft. For example, the pushover-pullups maneuver results in data which are suitable for determining the aerodynamic lift and drag characteristics. Along with the flight maneuvers, a set of dedicated control inputs is implemented during a flight test. Normally, the control inputs are optimally designed to increase the identifiability of certain parameters. Such optimal control inputs include 3-2-1-1, 2-1-1, and doublet [18, 25]. As will be presented in Chapter 4, the information content of the data can be quantified through the inverse of the Fisher information matrix ($\mathcal{F}$), which is derived from the Maximum Likelihood principle. The matrix is formulated as,

$$\mathcal{F}^{-1} = \mathrm{inv}\left[\sum_{k=1}^{N}\left[\frac{\partial \boldsymbol{y}_k}{\partial \Theta}\right]^T \mathbf{R}^{-1}\left[\frac{\partial \boldsymbol{y}_k}{\partial \Theta}\right]\right] \tag{2.4.2}$$

where y_k is the model output, $\mathbf{\Theta}$ is the system parameters, and $\mathbf{R}$ is the measurement noise covariance. The inverse of the Fisher information matrix ($\mathcal{F}$) formulated above basically contains the parameter error covariances. The higher the value of the error covariances indicates the less the information content in the data. A matrix $\mathbf{R}$ depends on the measurement noise while $\mathbf{\Theta}$ are assumed to be fixed for a certain period. Given these facts, the inverse of the Fisher matrix depends mainly on the response gradients of the system output. This dependency is mainly derived by the input excitation. Optimal control input design is derived by this formula, i.e., the $\mathcal{F}^{-1}$ is minimized with respect to the control input u which leads to a higher certainty of the parameters $\mathbf{\Theta}$ or an optimal control input that maximizes the information content in the data [21, 76].

In contrast, QAR data are recorded without such treatments as they are obtained from daily flight operations in which the flight is performed in normal flight condition (low control input excitation). Consequently, the information content in the QAR data is considered low compared to the flight testing data. This fact leads to significant limitations to the identifiability of certain system parameters.

2.4.3 Sensor Information

Sensors play an important role in the System Identification field as such devices collect all data required in the analysis. One aspect that brings success to the system identification is having a good quality sensor as it directly affects the quality of the recorded data. In system identification, the model output is used to describe the relationship between the measured values and the system's true quantity (system's states). In this model, the properties of the sensors, such as noise characteristics, biases, and scale factors, must be known and included in the model. Typically, in a flight system identification where data gathered through flight testing, such required sensor's information is known or can be retrieved through a laboratory test. In contrast, the QAR data utilized in this dissertation are processed without the knowledge of the sensor's properties. Some problems can be expected, such as identifiability and numerical instability due to too many unknown parameters that need to be estimated. These issues are addressed later in Chapter 5 and 6.

Chapter 3

Mathematical Model of an Aircraft

This chapter presents a mathematical model of the aircraft motion as well as its corresponding output model. The model of the aircraft motion provides a relationship among system states, while the output model describes a connection between system states and the measured variables. The chapter is started by defining some notations and reference frames along with frame transformation techniques which will be used to derive the equations of motion. Since the aircraft motion and its interaction with the environment is a complex process, no model can perfectly capture this behavior. However, through some assumptions, such complex behavior can be resembled by a mathematical model to some extent. Therefore, this chapter presents some assumptions employed in deriving the equations of motion so that the models are valid within the predefined conditions and configurations. For a stochastic and time-varying system, the aircraft motion is postulated in a nonlinear first-order differential equation as in [25],

$$\dot{\boldsymbol{x}}(t) = f\left[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t), t\right] \tag{3.0.1}$$

where the state vector ($\boldsymbol{x}$) may include airspeed (V), angle of attack (α), angle of sideslip (β), the Euler angles (ϕ, θ, ψ), and position (x_E, y_E, z_E). While the input vector ($\boldsymbol{u}$) may include throttle position (δ_T), rudder (ζ), elevator (η), aileron (ξ), and other control surfaces. Both state and input variables are a function of independent variable time (t). Vector $\boldsymbol{w}$ is a representation of a system affected by a stochastic environment. In the case of a deterministic, time-invariant system, Eq. (3.0.1) is simplified to,

$$\dot{\boldsymbol{x}}(t) = f\left[\boldsymbol{x}(t), \boldsymbol{u}(t)\right] \tag{3.0.2}$$

Four sets of state equations are derived, including translational, rotational, attitude, and position propagation equations. External forces such as gravity, propulsion, and aerodynamics are discussed in detail including the corresponding moments caused by the forces. Figure 3.1 summarizes the states, input variables, the equations of motion, as well as the external forces that will be elaborated in this chapter.

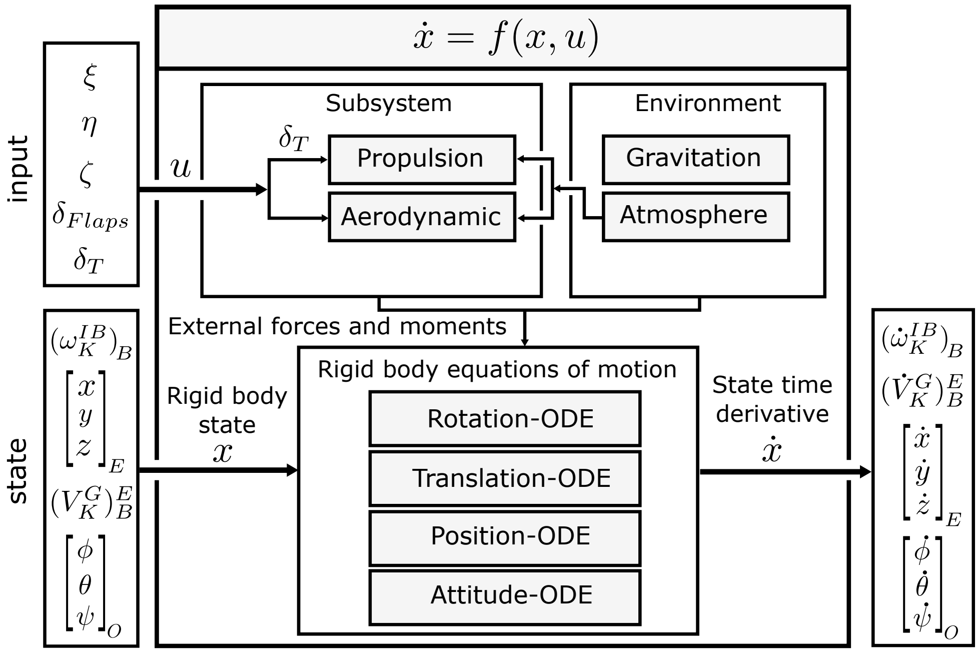

$\dot{x} = f(x,u)$
input
ξ
η
ζ
δ_{Flaps}
δ_T
u
Subsystem
δ_T
Propulsion
Aerodynamic
Environment
Gravitation
Atmosphere
External forces and moments
state
$(\omega_K^{IB})_B$
$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_E$
$(V_K^G)_B^E$
$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_O$
Rigid body state
x
Rigid body equations of motion
Rotation-ODE
Translation-ODE
Position-ODE
Attitude-ODE
State time derivative
$\dot{x}$
$(\dot{\omega}_K^{IB})_B$
$(\dot{V}_K^G)_B^E$
$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_E$
$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_O$

Figure 3.1: Summary of an aircraft mathematical model [77]

The output equation highly depends on how the state variables are modeled (the choice of reference frame) and sensors used in the aircraft. A typical output model takes form as algebraic equation which is also formulated as a function of state ($\boldsymbol{x}$) and input variables ($\boldsymbol{u}$). In general, the output models that correspond to the situation of Eq. (3.0.1) and Eq. (3.0.2) are formulated in Eqs. (3.0.3) and (3.0.4) respectively,

$$\boldsymbol{y} = h\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] \tag{3.0.3}$$

$$\boldsymbol{y} = h\left[\boldsymbol{x}(t), \boldsymbol{u}(t)\right] \tag{3.0.4}$$

3.1 Notations

Notations used in this chapter refer to the following nomenclatures. Scalar quantity of a vector component is denoted by italic font with a subscript represents the type of quantity or component's address in a reference frame, e.g., a_x where a is the scalar quantity, while index x denotes that the quantity lies on the x-axis of a reference frame. Angles and angular rates are denoted mainly by Greek letters, e.g., $\gamma_{\boldsymbol{K}}, \psi_{\boldsymbol{K}}$, where index $\boldsymbol{K}$ represents the type of quantity. All vectors will be printed in bold-face type fonts. For position vector, the superscript specifies two points which are used to indicate a position of one vector relative to the other. If only one superscript is specified in the vector, it indicates that the point is relative to an absolute point (the center of the earth). Parentheses with subscript are used to denote a vector in a specific frame. For velocity and acceleration vectors, the notation follows the position vector's nomenclature with additional superscript that represents the relative frame in which the derivative is taken. Similar to a scalar quantity, a vector may also have a subscript denoting the type of component's address in a reference frame. The description of the vectors' (position, velocity, and acceleration) nomenclatures are best described by the following examples:

$\left(\boldsymbol{r}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)_{\boldsymbol{B}} \equiv$ Position of point $\boldsymbol{P}$ relative to point $\boldsymbol{G}$ in the $\boldsymbol{B}$ frame. If only one superscript is used in the vector $\left(\boldsymbol{r}_{\boldsymbol{K}}^{\boldsymbol{P}}\right)_{\boldsymbol{B}}$, it indicates the point's position with respect to the center of the earth $\boldsymbol{O}$. Subscript $\boldsymbol{K}$ in this case denotes the type of quantity.

$\left(\dot{\boldsymbol{r}}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)_{\boldsymbol{B}}^{\boldsymbol{E}} \equiv \left(\boldsymbol{V}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)_{\boldsymbol{B}}^{\boldsymbol{E}} \equiv$ Velocity of vector $\left(\boldsymbol{r}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)$ with respect to the $\boldsymbol{E}$ frame denoted in the $\boldsymbol{B}$ frame.

$\left(\dot{\boldsymbol{V}}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)_{\boldsymbol{B}}^{\boldsymbol{EB}} \equiv \left(\boldsymbol{a}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)_{\boldsymbol{B}}^{\boldsymbol{EB}} \equiv$ $\left(\boldsymbol{V}_{\boldsymbol{K}}^{\boldsymbol{GP}}\right)$ relative to the $\boldsymbol{E}$ frame differentiated with respect to the $\boldsymbol{B}$ frame and denoted in the $\boldsymbol{B}$ frame.

Forces are notated by a bold-italic and capital letter, e.g., $\boldsymbol{A}, \boldsymbol{P}$, and $\boldsymbol{G}$ which denote total, aerodynamic, propulsive, and gravity forces respectively. Three main reference points are used throughout the chapter, and they are denoted as $\boldsymbol{G}$ for the center of gravity, $\boldsymbol{R}$ for the aircraft reference point, and $\boldsymbol{P}$ for an arbitrary point in the body of the aircraft. Matrices related to the reference frame are denoted by a capital and bold-face type font with one or two subscripts. If used with two subscripts, the inner index denotes output/target frame while the outer index specifies input/base frame. For example, a matrix $\mathbf{M}_{\boldsymbol{BA}}$ contains components used to transform a quantity on $\boldsymbol{A}$ frame (base frame) to $\boldsymbol{B}$ frame (target frame). Any other notation which is not covered in this section will be briefly introduced before being used in the formula. Notations presented in this chapter are mainly adapted from nomenclatures used at the Institute of Flight System Dynamics of TUM [77].

3.2 Reference Frames and Transformation

Formulating the aircraft equations of motion requires reference frames for specifying a relative component of the aircraft dynamic variables such as forces, velocities, attitudes, and other motion variables. Without a reference frame, modeling a flight vehicle's position or motion could not be made. Therefore, defining and knowing the reference frames is an essential step in modeling the aircraft motions. By definition, a frame or loosely referred to as a reference frame is an unbound continuous set of points over the Euclidean three-space with invariant distances and which possesses, as a subset, at least three non-collinear points [78]. In aerospace applications, there are numerous reference frames but only some of them are referred to in this dissertation, namely Earth-Centered Inertial - ECI ($\boldsymbol{I}$), Earth Centered Earth Fixed - ECEF ($\boldsymbol{E}$), which can be expressed in two forms, i.e., Word Geodetic System 1984 (WGS84 frame) and Cartesian ECEF frame, North-East-Down frame - NED ($\boldsymbol{O}$), Body-Fixed frame ($\boldsymbol{B}$), Aerodynamic frame ($\boldsymbol{A}$), Navigation frame ($\boldsymbol{N}$), and Kinematic frame ($\boldsymbol{K}$). Each of the reference frames is briefly described below. Further description of the reference frames can be found in Appendix A.1 [77].

- ECI frame, $\boldsymbol{I}$. The origin of this reference frame is at the center of the earth. The positive

x-axis is in the direction of the vernal equinox while the positive y-axis is forming a right-hand system with the x- and z-axis. The z-axis is the rotation axis of the earth. This frame is assumed to be fixed (non-rotating frame) with respect to the earth. Newton's laws apply to this frame.

- ECEF frame, $\boldsymbol{E}$. The origin is at the center of the earth with the positive x-axis is in the direction of the Greenwich Meridian while the y-axis is forming a right-hand axis system with the x- and z-axis. The positive z-axis lies on the earth's rotation axis (collinear with the ECI z-axis). This frame is rotating with respect to the ECI frame at earth's rotation rate (2π per 24 h). The ECEF frame can typically be presented in two different frames, either in ECEF cartesian frame (x, y, z) or in WGS-84 model where positions are represented by the geodetic latitude, longitude, and altitude [79].
- NED frame, $\boldsymbol{O}$. The origin point of this reference frame is at the aircraft reference point, usually at the aircraft's center of gravity G. Therefore, this reference frame moves and rotates along with the aircraft. The positive x-axis is parallel and pointing to the geographic North Pole while the y-axis pointing to east forming the right-hand system with the x- and z-axis. The positive z-axis is pointing downwards perpendicular to the local geoid surface.
- Body-Fixed frame, $\boldsymbol{B}$. The origin is at the aircraft reference point, usually at the aircraft's center of gravity, G. Therefore, this frame rotates and moves along with the aircraft. The positive x-axis is pointing forward through the nose of the aircraft; the positive y-axis points out to the right wing and the positive z-axis through the underside forming the right-hand system with the x- and y-axis.
- Aerodynamic frame, $\boldsymbol{A}$. The origin of this frame is at the reference point of the aircraft with the positive x-axis aligns with the aerodynamic velocity, pointing into the direction of the aerodynamic velocity while the y-axis points to the right perpendicular to the x- and z-axis. The positive z-axis points downwards on the symmetry plane of the aircraft and perpendicular to the xy plane. This frame moves and rotates along with the aircraft.
- Kinematic frame, $\boldsymbol{K}$. The origin of this frame is at the reference point of the aircraft. The positive x-axis aligns with the kinematic velocity and points to the direction of the kinematic velocity. The positive y-axis points to the right, perpendicular to the x- and z-axis. The positive z-axis is parallel to the projection of the local normal of the WGS-84 ellipsoid forming the right-hand system with the x- and y-axis.
- Navigation frame, $\boldsymbol{N}$. The origin of this frame is at an arbitrary point on the earth's surface and fixed with respect to the earth and rotates at the same speed as the earth's angular rate. The positive x-axis is pointing to a direction that deviates with the alignment of an angle from north direction while the positive y-axis is parallel to the local geoid surface and forming a right-hand system with the x- and z-axis. The z-axis is pointing downwards perpendicular to the local geoid surface.

Aircraft equations of motion may be written in any of the reference frames, and the choice is being a matter of application. For example, the Body-Fixed frame ($\boldsymbol{B}$) is preferred if the model includes the measurement variables as the sensors are aligned with the aircraft's body axis. However, if the model is used for simulating the aircraft's trajectory, then the Kinematic ($\boldsymbol{K}$) frame is preferred than the other frames. Since there are some choices of reference frames, the formula for transforming one reference frame to the others must be available. The following Figure 3.2 depicts an illustrative example of transforming one frame to the other, along with the corresponding angles.

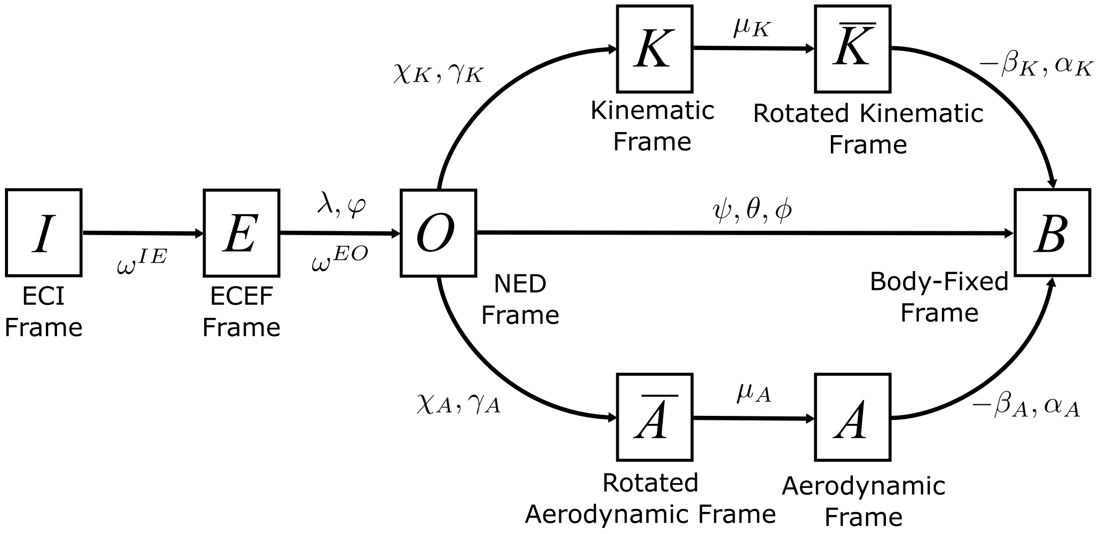


Figure 3.2: Frames references and corresponding angles for transformation [77]

The Inertial frame ($\boldsymbol{I}$), where Newton's laws apply, is taken as a starting point. Transformation begins by transforming the Inertial frame ($\boldsymbol{I}$) to the ECEF frame ($\boldsymbol{E}$) via the Earth's rotation rate ($\boldsymbol{\omega^{IE}}$) of approximately around 2π per 24h. The ECEF frame is then transformed into the NED frame ($\boldsymbol{O}$) through two corresponding angles, longitude (λ) and latitude (φ), with a transport rotation rate of $\boldsymbol{\omega^{EO}}$. The ECEF ($\boldsymbol{E}$) and NED ($\boldsymbol{O}$) frames serve for different purposes. The ECEF frame is mainly used for specifying the aircraft's position, while the NED frame is mainly used for describing the aircraft's attitude. To transform the NED frame into the Body-Fixed frame ($\boldsymbol{B}$), one can use the two steps procedure via the Kinematic Frame ($\boldsymbol{K}$) and Rotated Kinematic frame ($\overline{\boldsymbol{K}}$) on the top, or via the Rotated Aerodynamic frame ($\overline{\boldsymbol{A}}$) and Aerodynamic frame ($\boldsymbol{A}$) on the bottom, or the transformation can be done directly through the Euler angles (ψ, θ, ψ). When rotating one frame to another, it should be noted that successive rotation around frame axis is not commutative,

$$\mathbf{M}_{321} = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \neq \mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{M}_3 \tag{3.2.1}$$

For example, transforming a vector in the NED frame ($\boldsymbol{O}$) into the Body-Fixed frame ($\boldsymbol{B}$) must follow a sequence of rotation which is started from ψ, followed by θ, and ended with ϕ angle or in the axis rotation, it follows $z - y - x$ sequence. Each rotation requires a matrix which contains elements that project the NED frame ($\boldsymbol{O}$) to the element of the Body-Fixed frame ($\boldsymbol{B}$). The basic rule of getting the matrix transformation for each axis rotation is described in Appendix A.2 and is further detailed in [77]. Thus, the following rotation sequence and matrices are applied for transforming the NED frame ($\boldsymbol{O}$) to the Body-Fixed frame ($\boldsymbol{B}$),

$$\mathbf{M}_{\boldsymbol{BO}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}}_{3^{rd}\text{ rotation, } x\text{-axis}} \underbrace{\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}}_{2^{nd}\text{ rotation, } y\text{-axis}} \underbrace{\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{1^{st}\text{ rotation, } z\text{-axis}} \tag{3.2.2}$$

Multiplying all 3 matrices yields the matrix transformation from the NED frame ($\boldsymbol{O}$) to the Body-Fixed frame ($\boldsymbol{B}$),

$$\mathbf{M}_{\boldsymbol{BO}} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.2.3}$$

The same procedure can be used to transform one reference frame to the other. Since the matrix transformation is orthogonal, the reverse of the transformation matrix can be found through the relation,

$$\mathbf{M}_{\boldsymbol{OB}} = \mathbf{M}_{\boldsymbol{BO}}^{T} = \mathbf{M}_{\boldsymbol{BO}}^{-1} \tag{3.2.4}$$

While the matrix transformation of the angular rate between the NED frame ($\boldsymbol{O}$) and the Body-Fixed frame ($\boldsymbol{B}$) can be found by using the strap down equation. The angular rates for this transformation are represented by $\dot\psi, \dot\theta, \dot\phi$. The corresponding matrices transformation are shown in Eq. (3.2.5).

$$(\boldsymbol{\omega}^{\boldsymbol{OB}})_B = \begin{bmatrix} \dot\phi - \dot\psi\sin\theta \\ \dot\theta\cos\phi + \dot\psi\sin\phi\cos\theta \\ -\dot\theta\sin\phi + \dot\psi\cos\dot\phi\cos\theta \end{bmatrix}_B \qquad (\boldsymbol{\omega}^{\boldsymbol{OB}})_O = \begin{bmatrix} \dot\phi\cos\theta\cos\psi - \dot\theta\sin\psi \\ \dot\phi\cos\theta\sin\psi + \dot\theta\cos\psi \\ -\dot\phi\sin\theta + \dot\psi \end{bmatrix}_O \tag{3.2.5}$$

3.3 Modeling Requirements

In classical mechanics, an object might be modeled as individual particles or also known as point masses and assemblages of particles model called bodies. Bodies can further be modeled as a rigid-body model and an elastic-body model with varying mass [80]. Based on these object models, the formulation of the aircraft flight dynamics can be categorized into three general models, namely as a point mass model, a rigid-body model, and an elastic-body model. All these models are subject to aerodynamic, propulsive, and gravitational forces in non-stationary air. In the point mass model, the aircraft is assumed as an idealized body of an infinitesimal size where angular motions are not taken into account. While in the rigid-body model, both translational motion due to forces and angular motion due to moments are of concern. The third model describes the most complex formulation of the aircraft flight dynamics among others. However, when building a mathematical model of a system, one needs to follow the principle of system modeling: "as complex as necessary, as simple as possible." Thus, in building a mathematical model of the aircraft motion, several assumptions need to be defined in order to reduce the complexity of the system in the underlying model yet still capable of resembling the system behavior in the defined constraints. The following assumptions are employed throughout this dissertation for building aircraft equations of motion [25, 80]:

1. Aircraft is a rigid body with fixed mass distribution and constant mass. This assumption leads to several simplifications:
   - it allows the aircraft's translational and angular motion to be defined by the location of its center of mass,
   - it allows describing a model in the Body-Fixed frame $(\boldsymbol{B})$ relative to the Inertial frame $(\boldsymbol{I})$,
   - it limits the degree of freedom required in modeling the aircraft's motion as well as eliminates the dependence of aerodynamic and inertial effects on structural bending and torsion.
2. The earth is fixed in inertial space and serves as the Inertial frame of reference. Newton's laws are valid in this frame.
3. The rotation between the ECI frame and the ECEF frame, denoted as $\boldsymbol{\omega^{IE}}$, is assumed to be constant which leads to zero rotational acceleration,

$$\left(\dot{\boldsymbol{\omega}}^{\boldsymbol{IE}}\right)^{\boldsymbol{I}} = \left(\dot{\boldsymbol{\omega}}^{\boldsymbol{IE}}\right)^{\boldsymbol{E}} = 0 \tag{3.3.1}$$

4. Flight in the Earth's atmosphere is close to the earth's surface, so the Earth's surface can be approximated as flat. The flat-earth assumption neglects the variation of gravitational acceleration with distance from the center of the Earth, and it disregards relatively subtle Coriolis and centrifugal effects. As the gravitational acceleration is uniform, it allows the assumption that the aircraft's center of gravity coincides with the aircraft's center of mass.

With the mentioned assumptions above, aircraft equations of motion that will be derived in this chapter consist of 12 equations (or 13 in case of quaternions for the attitude propagation) coupled in a non-linear first order ordinary differential equations. Each model set will be explained and derived in Section 3.4.1 - 3.4.4.

3.4 Rigid-Body Equations of Motion

Deriving aircraft equations of motion requires a transformation and derivation of a reference frame relative to the other frame. Transformation is done through a matrix that transforms the orientation of a vector from its base frame to the target frame. For example, a matrix

$\mathbf{M}_{BO}$ in Eq. (3.2.3) transforms the NED frame to the Body-Fixed frame. On the other hand, a derivation requires special treatment as the derivative operator is applied to a vector that lies on a frame that rotates relative to the other frame. Visually, a vector on a rotating frame differentiated with respect to the other frame is depicted in Figure 3.3.

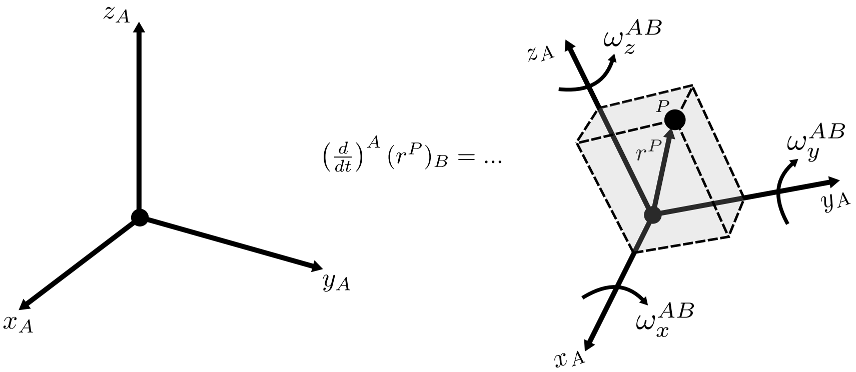


Figure 3.3: Visual representation of a rotating frame with respect to the other frame

Started by defining a vector $\boldsymbol{r^P}$ that points to point $\boldsymbol{P}$ in frame $\boldsymbol{B}$. The base frame (frame $\boldsymbol{B}$) rotates relative to frame $\boldsymbol{A}$. Taking the derivative operator on vector $\boldsymbol{r^P}$ involves two terms – one account for the time derivative of position $\boldsymbol{P}$ with respect to frame $\boldsymbol{A}$ and the other relates to the rotation of the reference frame or the so-called 'Coriolis effect' [81], expressed as:

$$\left(\frac{d}{dt}\right)^{\boldsymbol{A}} (\boldsymbol{r^P})_{\boldsymbol{B}} = \underbrace{(\dot{\boldsymbol{r}}^{\boldsymbol{P}})_{\boldsymbol{B}}^{\boldsymbol{B}}}_{\text{Direct term}} + \underbrace{(\boldsymbol{\omega^{AB}})_{\boldsymbol{B}} \times (\boldsymbol{r^P})_{\boldsymbol{B}}}_{\text{Coriolis term}} \tag{3.4.1}$$

Eq. (3.4.1) will serve as a basis for deriving the aircraft equations of motion through Newton's second law. As already mentioned in Section 3.3, the equations of motion will be derived in the Inertial frame since Newton's second law applies in this frame. The transformation from the Inertial frame ($\boldsymbol{I}$) to the other frame of interest is possible through the matrix transformations, see Section 3.2. Further explanation on the derivation of Eq. (3.4.1) can be found in [77].

3.4.1 Translational Equations of Motion

In the context of aircraft equations of motion, a force can be defined as a physical influence that changes the translational motion of the aircraft. Forces such as aerodynamics, propulsion, and gravitational cause the aircraft to accelerate or decelerate with respect to a reference frame. In this section, the translational equations of motion due to such forces will be briefly derived and discussed. For an extensive explanation, interested readers may refer to Stengel [80], Boiffier [82], Etkin [83], and Roskam [84]. As indicated in Eq. (3.4.1), two variables are involved in an object that rotates and moves with respect to a reference frame. One variable is related to the translational term and the other variable related to the rotational effect. In terms of the aircraft equations of motion, one frame of interest is the Body-Fixed frame where sensors such as accelerometer align with the body axis line. This frame rotates and moves with respect to the ECI frame. Therefore, two variables, i.e., absolute velocity $(\boldsymbol{V^P})^{\boldsymbol{I}}$ of a point in the Body-Fixed frame with respect to the ECI frame and its corresponding angular rate ($\boldsymbol{\omega^{IB}}$) will be derived. To derive these two variables, Figure 3.4 is used as a reference. Four reference frames are defined in Figure 3.4, i.e., Body-Fixed frame ($\boldsymbol{B}$), NED frame ($\boldsymbol{O}$), ECEF frame ($\boldsymbol{E}$) and ECI frame ($\boldsymbol{I}$). Each of the frames is rotating relative to others except for the ECI frame, which is defined as a non-rotating frame. The angular rate of the Body-Fixed frame relative to the ECI frame, denoted as $\boldsymbol{\omega^{IB}}$, is computed by summing all angular rates for each frame. In a vector notation, this angular rate is written as,

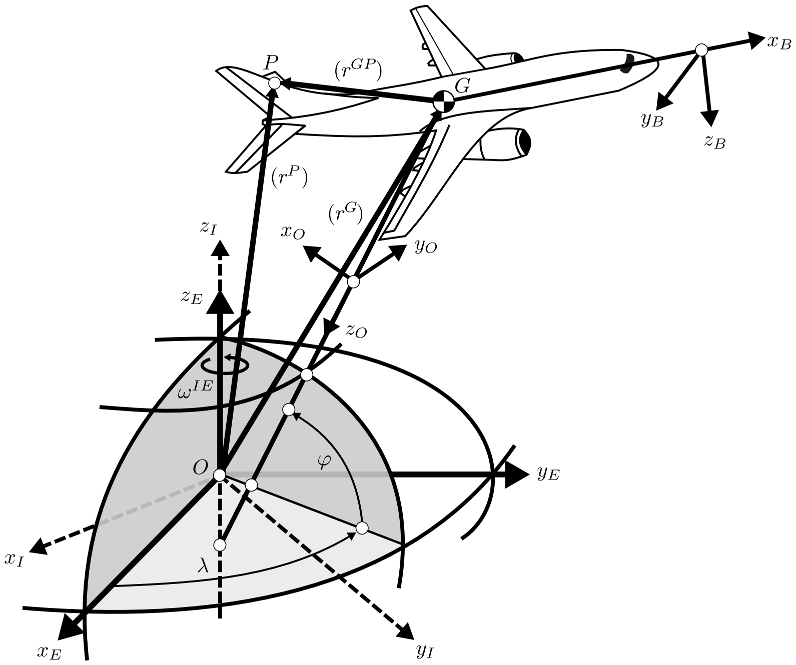


Figure 3.4: Variable definition for translation equation of motion [77]

$$\boldsymbol{\omega^{IB}} = \boldsymbol{\omega^{IE}} + \boldsymbol{\omega^{EO}} + \boldsymbol{\omega^{OB}} \tag{3.4.2}$$

or in a shorter form where variable $\boldsymbol{\omega^{EO}}$ and $\boldsymbol{\omega^{OB}}$ are combined in variable $\boldsymbol{\omega^{EB}}$ that represents the angular rate of the Body-Fixed frame relative to the ECEF frame. Thus, in a simplified form, Eq. (3.4.2) can be rewritten as,

$$\boldsymbol{\omega^{IB}} = \boldsymbol{\omega^{IE}} + \boldsymbol{\omega^{EB}} \tag{3.4.3}$$

where,

- $\boldsymbol{\omega^{IE}}$ is the angular velocity of the ECEF frame relative to the ECI frame (Earth's rotation rate).
- $\boldsymbol{\omega^{EO}}$ is the angular velocity of the NED frame relative to the ECEF frame (transport's rate).
- $\boldsymbol{\omega^{OB}}$ is the angular velocity of the Body-Fixed frame relative to the NED frame.

- $\boldsymbol{\omega}^{\boldsymbol{EB}}$ is the sum of $\boldsymbol{\omega}^{\boldsymbol{EO}}$ and $\boldsymbol{\omega}^{\boldsymbol{OB}}$ that represents the angular velocity of the Body-Fixed frame relative to the ECEF frame.

The second variable to be derived is the absolute velocity of an arbitrary point $\boldsymbol{P}$ in the Body-Fixed frame relative to the ECI frame. The velocity of this point is derived by first defining its position with respect to other frames. In the vector notation, variable $(\boldsymbol{r}^{\boldsymbol{P}})$ takes form as,

$$(\boldsymbol{r}^{\boldsymbol{P}}) = (\boldsymbol{r}^{\boldsymbol{G}}) + (\boldsymbol{r}^{\boldsymbol{GP}}) \tag{3.4.4}$$

Taking derivative of the vector $(\boldsymbol{r}^{\boldsymbol{P}})$ with respect to the ECI frame yields the absolute velocity $(\boldsymbol{V}^{\boldsymbol{P}})^{\boldsymbol{I}}$,

$$\left(\frac{d}{dt}\right)^{\boldsymbol{I}} (\boldsymbol{r}^{\boldsymbol{P}}) = \left(\frac{d}{dt}\right)^{\boldsymbol{I}} \Big[(\boldsymbol{r}^{\boldsymbol{G}}) + (\boldsymbol{r}^{\boldsymbol{GP}})\Big] \tag{3.4.5}$$

Applying Eq. (3.4.1) to each variable in Eq. (3.4.5) yields in,

$$(\boldsymbol{V}^{\boldsymbol{P}})^{\boldsymbol{I}} = \underbrace{(\dot{\boldsymbol{r}}^{\boldsymbol{G}})^{\boldsymbol{E}} + (\boldsymbol{\omega}^{\boldsymbol{IE}}) \times (\boldsymbol{r}^{\boldsymbol{G}})}_{(\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{I}}} + \underbrace{(\dot{\boldsymbol{r}}^{\boldsymbol{GP}})^{\boldsymbol{B}} + (\boldsymbol{\omega}^{\boldsymbol{IB}}) \times (\boldsymbol{r}^{\boldsymbol{GP}})}_{(\boldsymbol{V}^{\boldsymbol{GP}})^{\boldsymbol{I}}} \tag{3.4.6}$$

With the rigid body assumption is taken into account, the distance between point $\boldsymbol{G}$ and $\boldsymbol{P}$ is fixed. Consequently, the derivative of this distance is zero, $(\dot{\boldsymbol{r}}^{\boldsymbol{GP}})^{\boldsymbol{B}} = 0$. Furthermore, by defining $(\dot{\boldsymbol{r}}^{\boldsymbol{G}})^{\boldsymbol{E}} = (\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{E}}$, Eq. (3.4.6) is simplified into,

$$(\boldsymbol{V}^{\boldsymbol{P}})^{\boldsymbol{I}} = (\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{E}} + (\boldsymbol{\omega}^{\boldsymbol{IE}}) \times (\boldsymbol{r}^{\boldsymbol{G}}) + (\boldsymbol{\omega}^{\boldsymbol{IB}}) \times (\boldsymbol{r}^{\boldsymbol{GP}}) \tag{3.4.7}$$

Eqs. (3.4.2) and (3.4.6) summarize the two variables required in deriving the aircraft translational equations. Since Newton's second law requires acceleration variable, Eq. (3.4.6) is differentiated with respect to time relative to the ECI frame gives,

$$\left(\frac{d}{dt}\right)^{\boldsymbol{I}} (\boldsymbol{V}^{\boldsymbol{P}})^{\boldsymbol{I}} = \left(\frac{d}{dt}\right)^{I} \Big[(\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{E}} + (\boldsymbol{\omega}^{\boldsymbol{IE}}) \times (\boldsymbol{r}^{\boldsymbol{G}}) + (\boldsymbol{\omega}^{\boldsymbol{IB}}) \times (\boldsymbol{r}^{\boldsymbol{GP}})\Big] \tag{3.4.8}$$

Here, the principle of Eq. (3.4.1) is applied again, and the corresponding derivative of each variable is obtained as follow,

$$\begin{aligned}\left(\frac{d}{dt}\right)^{I} (\boldsymbol{V}^{P})^{I} = \underbrace{(\dot{\boldsymbol{V}}^{\boldsymbol{G}})^{\boldsymbol{EB}}}_{a} + \underbrace{(\boldsymbol{\omega}^{\boldsymbol{EB}}) \times (\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{E}}}_{b} + \underbrace{2 \cdot (\boldsymbol{\omega}^{\boldsymbol{IE}}) \times (\boldsymbol{V}^{\boldsymbol{G}})^{\boldsymbol{E}}}_{c} + \underbrace{(\boldsymbol{\omega}^{\boldsymbol{IE}}) \times \Big[(\boldsymbol{\omega}^{\boldsymbol{IE}}) \times (\boldsymbol{r}^{\boldsymbol{G}})\Big]}_{d} \\ + \underbrace{(\dot{\boldsymbol{\omega}}^{\boldsymbol{IB}})^{\boldsymbol{B}} \times (\boldsymbol{r}^{\boldsymbol{GP}})}_{e} + \underbrace{(\boldsymbol{\omega})^{\boldsymbol{IB}} \times \Big[(\boldsymbol{\omega}^{\boldsymbol{IB}}) \times (\boldsymbol{r}^{\boldsymbol{GP}})\Big]}_{f}\end{aligned} \tag{3.4.9}$$

where,

- a is the linear acceleration of the Body-Fixed frame with respect to the ECEF frame.
- b is the acceleration due to angular rates of transport rate $(\boldsymbol{\omega})^{\boldsymbol{EO}}$ and relative angular

rate $(\boldsymbol{\omega})^{\boldsymbol{OB}}$ (without the Earth's rotation).

- c is the Coriolis acceleration at a reference point due to the Earth's rotation.
- d is the centrifugal acceleration at a reference point due to the Earth's rotation.
- e is the relative acceleration at point $\boldsymbol{P}$ due to the rigid body angular acceleration.
- f is the relative centrifugal acceleration at point $\boldsymbol{P}$ due to the rigid body rotation.

A comprehensive derivation of Eq. (3.4.9) can be found in [77]. To derive the aircraft translational equations of motion, Newton's second law is applied to the ECI frame and formulated as,

$$\sum \boldsymbol{F} = \left(\frac{d}{dt}\right)^{\boldsymbol{I}} \int_m (\boldsymbol{V^P})^{\boldsymbol{I}} \cdot dm \tag{3.4.10}$$

By plugging Eq. (3.4.9) into Eq. (3.4.10) yields in,

$$\begin{aligned}\sum \boldsymbol{F} = \int_m dm \Bigg((\dot{\boldsymbol{V}}^{\boldsymbol{G}})^{\boldsymbol{EB}} + (\boldsymbol{\omega^{EB}}) \times (\boldsymbol{V^G})^{\boldsymbol{G}} + 2 \cdot (\boldsymbol{\omega^{IE}}) \times (\boldsymbol{V^G})^{\boldsymbol{E}} \\ + (\boldsymbol{\omega^{IE}}) \times \Big[(\boldsymbol{\omega^{IE}}) \times (\boldsymbol{r^G}) \Big] + (\dot{\boldsymbol{\omega}}^{\boldsymbol{IB}})^{\boldsymbol{B}} \times (\boldsymbol{r^{GP}}) \\ + (\boldsymbol{\omega})^{\boldsymbol{IB}} \times \Big[(\boldsymbol{\omega^{IB}}) \times (\boldsymbol{r^{GP}}) \Big] \Bigg)\end{aligned} \tag{3.4.11}$$

By taking the assumptions of the non-rotating earth $\boldsymbol{\omega^{IE}} = 0$ into account as well as point $\boldsymbol{G}$ is assumed as the center of gravity, which leads to static mass equal to zero $\int_m (\boldsymbol{r^{GP}}) dm = 0$, and $\int_m dm = m$, Eq. (3.4.11) is simplified into,

$$\sum \boldsymbol{F} = m \cdot \Big((\boldsymbol{V^G})^{\boldsymbol{EB}} + (\boldsymbol{\omega^{EB}}) \times (\boldsymbol{V^G})^{\boldsymbol{E}} \Big) \tag{3.4.12}$$

Furthermore, Eq. (3.4.12) is rearranged and denoted in the Body-Fixed frame yields,

$$(\dot{\boldsymbol{V}}^{\boldsymbol{G}}_{\boldsymbol{K}})^{\boldsymbol{EB}}_{\boldsymbol{B}} = \frac{\sum (\boldsymbol{F^G})_{\boldsymbol{B}}}{m} - (\boldsymbol{\omega^{EB}_K})_{\boldsymbol{B}} \times (\boldsymbol{V^G_K})^{\boldsymbol{E}}_{\boldsymbol{B}} \tag{3.4.13}$$

Note that $(\boldsymbol{\omega^{EB}}) = (\boldsymbol{\omega^{OB}}) = [p \quad q \quad r]^T$ due to the assumption of the flat Earth. Variable $\sum (\boldsymbol{F^G})_{\boldsymbol{B}}$ represents the sum of all external forces exerting on the aircraft denoted in the Body-Fixed frame. Eq. (3.4.13) can further be elaborated into each axis results in,

$$\begin{bmatrix} \dot{u}^{\boldsymbol{G}}_{\boldsymbol{K}} \\ \dot{v}^{\boldsymbol{G}}_{\boldsymbol{K}} \\ \dot{w}^{\boldsymbol{G}}_{\boldsymbol{K}} \end{bmatrix}^{\boldsymbol{EB}}_{\boldsymbol{B}} = \frac{1}{m} \cdot \begin{bmatrix} X^{\boldsymbol{G}} \\ Y^{\boldsymbol{G}} \\ Z^{\boldsymbol{G}} \end{bmatrix}_{\boldsymbol{B}} - \begin{bmatrix} q_{\boldsymbol{K}} \cdot w_{\boldsymbol{K}} - r_{\boldsymbol{K}} \cdot v_{\boldsymbol{K}} \\ r_{\boldsymbol{K}} \cdot u_{\boldsymbol{K}} - p_{\boldsymbol{K}} \cdot w_{\boldsymbol{K}} \\ p_{\boldsymbol{K}} \cdot v_{\boldsymbol{K}} - q_{\boldsymbol{K}} \cdot u_{\boldsymbol{K}} \end{bmatrix}_{\boldsymbol{B}} \tag{3.4.14}$$

Eq. (3.4.14) summarizes the translational equation of motion taking into account all assumptions defined in Section 3.3. The external forces such as gravity, aerodynamic, and propulsion are embodied in variables X, Y, Z which will further be explained in Section 3.5.

3.4.2 Rotational Equations of Motion

The second set of the aircraft equations of motion is related to the angular momentum quantity that represents a product of the aircraft's body rotational inertial and rotational velocity about a particular axis. This quantity can be considered as a rotational motion analogue of the linear momentum. Therefore, as in the translational equations of motion, Newton's second law is also applied in the derivation of the rotational equations. Figure 3.4 is referred for deriving the equations. The angular momentum of the body with respect to the ECEF frame is given as,

$$\boldsymbol{H^O} = \int_m \left(\boldsymbol{r^P}\right) \times \left(\boldsymbol{V^P}\right)^{\boldsymbol{I}} dm \tag{3.4.15}$$

where, terms $(\boldsymbol{r^P})$ and $(\boldsymbol{V^P})^{\boldsymbol{I}}$ can be constructed through their companion vectors as in Eqs. (3.4.4) and (3.4.6). These two equations are plugged back into Eq. (3.4.15) yields in,

$$\boldsymbol{H^O} = \int_m \left[\left(\boldsymbol{r^G}\right) + \left(\boldsymbol{r^{GP}}\right)\right] \times \left[\left(\boldsymbol{V^G}\right)^{\boldsymbol{I}} + \left(\boldsymbol{V^{GP}}\right)^{\boldsymbol{I}}\right] dm \tag{3.4.16}$$

Eq. (3.4.16) is further elaborated by introducing the $\left(\boldsymbol{V^{GP}}\right)^{\boldsymbol{I}}$ model as defined in Eq. (3.4.6). With the rigid body assumption $\left(\dot{\boldsymbol{r}}^{\boldsymbol{GP}}\right)^{\boldsymbol{B}} = 0$ as well as the assumptions that the mass distribution is constant ($\int_m dm = m$) and the reference point is at the center of gravity $\boldsymbol{G}$ ($\int_m \left(\boldsymbol{r^{GP}}\right) dm = 0$), Eq. (3.4.16) is detailed into,

$$\begin{aligned}
H^O &= \int_m \left[\left(\boldsymbol{r^G}\right) + \left(\boldsymbol{r^{GP}}\right)\right] \times \left[\left(\boldsymbol{V^G}\right)^{\boldsymbol{I}} + \left(\boldsymbol{\omega^{IB}}\right) \times \left(\boldsymbol{r^{GP}}\right)\right] dm \\
&= \underbrace{\int_{=m} dm}_{m} \left(\boldsymbol{r^G}\right) \times \left(\boldsymbol{V^G}\right)^{\boldsymbol{I}} + \left(\boldsymbol{r^G}\right) \times \left(\boldsymbol{\omega^{IB}}\right) \times \underbrace{\int_m \left(\boldsymbol{r^{GP}}\right) dm}_{=0} \\
&+ \underbrace{\int_m \left(\boldsymbol{r^G}\right) dm}_{=0} \times \left(\boldsymbol{V^G}\right)^{\boldsymbol{I}} + \int_m \left(\boldsymbol{r^{GP}}\right) \times \left[\left(\boldsymbol{\omega^{IB}}\right) \times \left(\boldsymbol{r^{GP}}\right)\right] dm \\
&= m \cdot \left[\left(\boldsymbol{r^G}\right) \times \left(\boldsymbol{V^G}\right)\right] + \int_m \left(\boldsymbol{r^G}\right) \times \left[\left(\boldsymbol{\omega^{IB}}\right) \times \left(\boldsymbol{r^{GP}}\right)\right] dm
\end{aligned} \tag{3.4.17}$$

The last term of Eq. (3.4.17) can further be simplified through the anti-commutative property of the cross product, i.e., $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$. Where the cross product operation can be replaced by the matrix-vector product as in,

$$\left(\boldsymbol{r^{GP}}\right) \cdot \left(\boldsymbol{\Omega^{IB}}\right) = \left(\boldsymbol{\Omega^{GP}}\right) \cdot \left(\boldsymbol{\omega^{IB}}\right) \tag{3.4.18}$$

whereas variable $\boldsymbol{\Omega}^{\boldsymbol{GP}}$ represents the skew-symmetric matrix that contains the components of position vector $\boldsymbol{r}^{\boldsymbol{GP}}$ in the off-diagonal elements. Both of these variable are written as,

$$\left(\boldsymbol{r}^{\boldsymbol{GP}}\right) = \begin{bmatrix} x^{\boldsymbol{GP}} \\ y^{\boldsymbol{GP}} \\ z^{\boldsymbol{GP}} \end{bmatrix} \qquad \left(\boldsymbol{\Omega}^{\boldsymbol{GP}}\right) = \begin{bmatrix} 0 & -z^{\boldsymbol{GP}} & y^{\boldsymbol{GP}} \\ z^{\boldsymbol{GP}} & 0 & -x^{\boldsymbol{GP}} \\ -y^{\boldsymbol{GP}} & x^{\boldsymbol{GP}} & 0 \end{bmatrix} \tag{3.4.19}$$

Taking these two operations into account, the last term of Eq. (3.4.17) is simplified into,

$$\begin{aligned}\int_m \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left[\left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \left(\boldsymbol{r}^{\boldsymbol{GP}}\right)\right] dm &= -\int_m \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\boldsymbol{r}^{\boldsymbol{GP}}\right) \times \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) dm \\ &= -\underbrace{\int_m \left(\boldsymbol{\Omega}^{\boldsymbol{GP}}\right) \cdot \left(\boldsymbol{\Omega}^{\boldsymbol{GP}}\right) dm}_{\boldsymbol{I}^{\boldsymbol{G}}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)\end{aligned} \tag{3.4.20}$$

Element $\int_m (\boldsymbol{\Omega}^{\boldsymbol{GP}}) \cdot (\boldsymbol{\Omega}^{\boldsymbol{GP}}) dm$ represents the symmetric moment of the inertia $\boldsymbol{I}^{\boldsymbol{G}}$ with respect to the point $\boldsymbol{G}$, or known as inertia tensor of the aircraft. With the simplified model introduced in Eq. (3.4.20) and the definition of the linear momentum with respect to the center of gravity $\boldsymbol{p}^{\boldsymbol{G}}$, the final model of angular momentum in the ECEF frame can be written as,

$$\boldsymbol{H}^{O} = \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\boldsymbol{p}^{\boldsymbol{G}}\right) + \boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \tag{3.4.21}$$

The next step is to apply Newton's second law to the angular momentum equation as formulated in Eq. (3.4.21). This yields in,

$$\left(\frac{d}{dt}\right)^{\boldsymbol{I}} \boldsymbol{H}^{\boldsymbol{O}} = \sum \boldsymbol{M}^{\boldsymbol{O}} \tag{3.4.22}$$

Where $\sum \boldsymbol{M}^{\boldsymbol{O}}$ represents the total moment exerting on the aircraft. With the assumption that moment of inertia does not change over time, the derivative of Eq. (3.4.22) with respect to the ECI frame leads to,

$$\begin{aligned}\left(\frac{d}{dt}\right)^{\boldsymbol{I}} \boldsymbol{H}^{\boldsymbol{O}} &= \left(\boldsymbol{V}_{\boldsymbol{K}}^{\boldsymbol{G}}\right) \times \left(\boldsymbol{p}^{\boldsymbol{G}}\right) + \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\dot{\boldsymbol{p}}^{\boldsymbol{G}}\right)^{\boldsymbol{I}} \\ &\quad + \left(\frac{d}{dt}\right)^{\boldsymbol{B}} \left(\boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)\right) + \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \left(\boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)\right) \\ &= \underbrace{\left(\boldsymbol{V}_{\boldsymbol{K}}^{\boldsymbol{G}}\right) \times \left(\boldsymbol{V}_{\boldsymbol{K}}^{\boldsymbol{G}}\right) \cdot m}_{=0} + \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\dot{\boldsymbol{p}}^{\boldsymbol{G}}\right)^{\boldsymbol{I}} + \boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\dot{\boldsymbol{\omega}}^{\boldsymbol{IB}}\right)^{\boldsymbol{B}} + \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \left(\boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)\right) \\ &= \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\dot{\boldsymbol{p}}^{\boldsymbol{G}}\right)^{\boldsymbol{I}} + \boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\dot{\boldsymbol{\omega}}^{\boldsymbol{IB}}\right)^{\boldsymbol{B}} + \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \left(\boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)\right)\end{aligned} \tag{3.4.23}$$

The right side of Eq. (3.4.22) is further modified by shifting the reference point to the aircraft's center of gravity $\boldsymbol{G}$ yields in,

$$\sum \boldsymbol{M}^{\boldsymbol{O}} = \sum \boldsymbol{M}^{\boldsymbol{G}} + \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \sum \boldsymbol{F}^{\boldsymbol{G}} \tag{3.4.24}$$

By combining Eqs. (3.4.23) and (3.4.24) and the fact that the linear momentum $\left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \left(\frac{d}{dt}\right)^{\boldsymbol{I}} \boldsymbol{p}^{\boldsymbol{G}} = \left(\boldsymbol{r}^{\boldsymbol{G}}\right) \times \sum \boldsymbol{F}^{\boldsymbol{G}}$, the rotational equation of motion for the aircraft can be formulated as,

$$\sum \boldsymbol{M}^{\boldsymbol{G}} = \boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\dot{\boldsymbol{\omega}}^{\boldsymbol{IB}}\right)^{\boldsymbol{B}} + \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \times \boldsymbol{I}^{\boldsymbol{G}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right) \tag{3.4.25}$$

Furthermore, $\left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)$ can be obtained by taking the non-rotational Earth and flat-Earth assumptions into account yields in $\left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)$ equals to $\left(\boldsymbol{\omega}^{\boldsymbol{OB}}\right)$. In scalar notation, the equation can be modeled as,

$$\left(\boldsymbol{\omega}^{\boldsymbol{IB}}\right)_{\boldsymbol{B}} = \left(\boldsymbol{\omega}^{\boldsymbol{OB}}\right)_{\boldsymbol{B}} = \begin{bmatrix} p_{\boldsymbol{K}} \\ q_{\boldsymbol{K}} \\ r_{\boldsymbol{K}} \end{bmatrix} \tag{3.4.26}$$

Eq. (3.4.25) is rearranged in terms of its states which leads to the common notation used for modeling the aircraft rotational equation of motion,

$$\left(\dot{\boldsymbol{\omega}}^{\boldsymbol{OB}}\right)_{\boldsymbol{B}}^{\boldsymbol{B}} = \left(\boldsymbol{I}^{\boldsymbol{G}}\right)_{\boldsymbol{BB}}^{-1} \cdot \left[\sum \left(\boldsymbol{M}^{\boldsymbol{G}}\right)_{\boldsymbol{B}} - \left(\boldsymbol{\omega}^{\boldsymbol{OB}}\right)_{\boldsymbol{B}} \times \left(\boldsymbol{I}^{\boldsymbol{G}}\right)_{\boldsymbol{BB}} \cdot \left(\boldsymbol{\omega}^{\boldsymbol{OB}}\right)\right] \tag{3.4.27}$$

The inertial matrix $(\boldsymbol{I}^{\boldsymbol{G}})_{\boldsymbol{BB}}$ is referred with respect to the Body-Fixed frame and can be assumed as constant [25]. This intertia matrix takes form as,

$$(\boldsymbol{I}^{\boldsymbol{G}})_{\boldsymbol{BB}} = \begin{bmatrix} \boldsymbol{I}_{xx}^{\boldsymbol{G}} & -\boldsymbol{I}_{xy}^{\boldsymbol{G}} & -\boldsymbol{I}_{xz}^{\boldsymbol{G}} \\ -\boldsymbol{I}_{xy}^{\boldsymbol{G}} & \boldsymbol{I}_{yy}^{\boldsymbol{G}} & -\boldsymbol{I}_{yz}^{\boldsymbol{G}} \\ -\boldsymbol{I}_{xz}^{\boldsymbol{G}} & -\boldsymbol{I}_{yz}^{\boldsymbol{G}} & \boldsymbol{I}_{zz}^{\boldsymbol{G}} \end{bmatrix}_{\boldsymbol{BB}} \tag{3.4.28}$$

where,

$$\begin{aligned} &\boldsymbol{I}_{xx}^{\boldsymbol{G}} \equiv \int_{\text{volume}} x^2 dm \quad \boldsymbol{I}_{yy}^{\boldsymbol{G}} \equiv \int_{\text{volume}} y^2 dm \quad \boldsymbol{I}_{zz} \equiv \int_{\text{volume}} z^2 dm \\ &\boldsymbol{I}_{xy}^{\boldsymbol{G}} = I_{yx}^{\boldsymbol{G}} \equiv \int_{\text{volume}} xydm \\ &\boldsymbol{I}_{yz}^{\boldsymbol{G}} = I_{zy}^{\boldsymbol{G}} \equiv \int_{\text{volume}} yzdm \\ &\boldsymbol{I}_{xz}^{\boldsymbol{G}} = I_{zx}^{\boldsymbol{G}} \equiv \int_{\text{volume}} xzdm \end{aligned} \tag{3.4.29}$$

As indicated in Eq. (3.4.29), the inertial matrix $(\boldsymbol{I}^{\boldsymbol{G}})_{\boldsymbol{BB}}$ is symmetric in the rigid body of the Body-Fixed Frame, where $\boldsymbol{O}_{xz}$ being the symmetric plane, which then leads to $\boldsymbol{I}_{xy}^{\boldsymbol{G}} = \boldsymbol{I}_{yx}^{\boldsymbol{G}} = \boldsymbol{I}_{yz}^{\boldsymbol{G}} = \boldsymbol{I}_{zy}^{\boldsymbol{G}} = 0$. Eq. (3.4.28) is simplified into,

$$(\boldsymbol{I}^{\boldsymbol{G}})_{\boldsymbol{BB}} = \begin{bmatrix} \boldsymbol{I}_{xx}^{\boldsymbol{G}} & 0 & -\boldsymbol{I}_{xz}^{\boldsymbol{G}} \\ 0 & \boldsymbol{I}_{yy}^{\boldsymbol{G}} & 0 \\ -\boldsymbol{I}_{xz}^{\boldsymbol{G}} & 0 & \boldsymbol{I}_{zz}^{\boldsymbol{G}} \end{bmatrix}_{\boldsymbol{BB}} \tag{3.4.30}$$

Taking Eq. (3.4.30) into account, Eq. (3.4.27) can be rewritten in scalar form as,

$$\begin{bmatrix}\dot{p}_{\boldsymbol{K}}\\ \dot{q}_{\boldsymbol{K}}\\ \dot{r}_{\boldsymbol{K}}\end{bmatrix}_{\boldsymbol{B}}^{\boldsymbol{B}} = \begin{bmatrix}\boldsymbol{I}_{xx}^{\boldsymbol{G}} & 0 & -\boldsymbol{I}_{xz}^{\boldsymbol{G}}\\ 0 & \boldsymbol{I}_{yy}^{\boldsymbol{G}} & 0\\ -\boldsymbol{I}_{xz}^{\boldsymbol{G}} & 0 & \boldsymbol{I}_{zz}^{\boldsymbol{G}}\end{bmatrix}_{\boldsymbol{BB}}^{-1} \left(\begin{bmatrix}\boldsymbol{L}^{\boldsymbol{G}}\\ \boldsymbol{M}^{\boldsymbol{G}}\\ \boldsymbol{N}^{\boldsymbol{G}}\end{bmatrix} - \begin{bmatrix}p_{\boldsymbol{K}}\\ q_{\boldsymbol{K}}\\ r_{\boldsymbol{K}}\end{bmatrix}_{\boldsymbol{B}}^{\boldsymbol{B}} \times \begin{bmatrix}\boldsymbol{I}_{xx}^{\boldsymbol{G}} & 0 & -\boldsymbol{I}_{xz}^{\boldsymbol{G}}\\ 0 & \boldsymbol{I}_{yy}^{\boldsymbol{G}} & 0\\ -\boldsymbol{I}_{xz}^{\boldsymbol{G}} & 0 & \boldsymbol{I}_{zz}^{\boldsymbol{G}}\end{bmatrix}_{\boldsymbol{BB}} \times \begin{bmatrix}p_{\boldsymbol{K}}\\ q_{\boldsymbol{K}}\\ r_{\boldsymbol{K}}\end{bmatrix}_{\boldsymbol{B}}^{\boldsymbol{B}}\right) \tag{3.4.31}$$

Eq. (3.4.27) summarizes the second set model used in simulating the rotational dynamics of the aircraft.

3.4.3 Position Propagation Equations

The third set of equations is related to the position propagation due to translational speed relative to the earth. These equations are usually referred to as kinematic equations as they only describe the relationship between the state of the aircraft and its change without considering the cause of the change. Such a position propagation model can be described in different frames such as the NED frame, the Navigation frame, or the ECEF (WGS-84) frame. If the velocity of the aircraft in the Body-Fixed frame is known and the position propagation model is preferred in the NED frame, the following model is used,

$$\begin{aligned}(\boldsymbol{V_K})_{\boldsymbol{O}}^{\boldsymbol{E}} &= \mathbf{M}_{\boldsymbol{OB}}(\phi,\theta,\psi)\cdot(\boldsymbol{V_K})_{\boldsymbol{B}}^{\boldsymbol{E}}\\ &= \mathbf{M}_{\boldsymbol{OB}}\cdot\begin{bmatrix}u_k\\ v_k\\ w_k\end{bmatrix}_{\boldsymbol{B}}^{\boldsymbol{E}}\end{aligned} \tag{3.4.32}$$

where $\mathbf{M}_{\boldsymbol{OB}}$ represents the transformation matrix from the Body-Fixed frame to the NED frame as in,

$$\mathbf{M}_{\boldsymbol{OB}} = \begin{bmatrix}\cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta\\ \cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi+\cos\psi\cos\phi & \cos\theta\sin\phi\\ \cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi & \cos\theta\cos\phi\end{bmatrix} \tag{3.4.33}$$

The translational propagation given in Eq. (3.4.32) assumes the flat and non-rotating Earth. The WGS-84 model provides an alternative form of position propagation equation in which the Earth is considered as round and rotating. The form of this model is given as,

$$\begin{bmatrix}\dot{\lambda}\\ \dot{\varphi}\\ \dot{h}\end{bmatrix} = \begin{bmatrix}\dfrac{v_{\boldsymbol{K}}^{\boldsymbol{G}}}{(N_\varphi+h)\cos\varphi}\\ \dfrac{u_{\boldsymbol{K}}^{\boldsymbol{G}}}{M_\varphi+h}\\ -w_{\boldsymbol{K}}^{\boldsymbol{G}}\end{bmatrix}_{\boldsymbol{O}}^{\boldsymbol{E}} \tag{3.4.34}$$

where variables M_φ and N_φ are given in Eqs. (3.4.35) and (3.4.36) below,

$$M_\varphi = a\frac{1-e^2}{(1-e^2\sin^2\varphi)^{\frac{3}{2}}} \tag{3.4.35}$$

$$N_\varphi = \frac{a}{\sqrt{(1-e^2\sin^2\varphi)}} \tag{3.4.36}$$

Variables $a = 6,378,137.0\ m$, $f = \frac{1}{298.257223563}$ and $e = \sqrt{(2f - f^2)}$ represent the Earth's semi major axis length, flattening, and eccentricity coefficients respectively. Eqs. (3.4.32) and (3.4.34) summarize the aircraft translational propagation equations.

3.4.4 Attitude Propagation Equations

The last set of the aircraft equations of motion is the attitude differential equation. Similar to the position propagation model, the attitude propagation model also represents the aircraft kinematics in terms of rotation that leads to the change of the aircraft's attitude. The Euler angles (ϕ, θ, ψ) are used as the angular measures to describe the aircraft orientation in the Body-Fixed frame with respect to the NED frame. The change of the aircraft orientation due to the rotation of the Body-Fixed frame relative to the NED frame denoted in the NED frame yields in,

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\phi\tan\theta & \cos\phi\tan\theta\\ 0 & \cos\phi & -\sin\phi\\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta}\end{bmatrix} \cdot \begin{bmatrix}p_{\boldsymbol{K}}\\ q_{\boldsymbol{K}}\\ r_{\boldsymbol{K}}\end{bmatrix} \tag{3.4.37}$$

Eq. (3.4.37) is derived based on the strap-down equation and commonly used to represent the aircraft rotational propagation equation. However, it suffers from singularity when the pitch angle, $\theta = \pm\frac{\pi}{2}$ deg as shown in the component $\frac{\sin\phi}{\cos\theta}$ and $\frac{\cos\phi}{\cos\theta}$, as $\cos\pm\frac{\pi}{2} = 0$. This problem is solved by converting the Euler angles into quaternion representation as q_0, q_1, q_2 and q_3 variables instead of ϕ, θ, and ψ. The relation between quaternions and the Euler angles are described in the following,

$$q_0 = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} \tag{3.4.38}$$

$$q_1 = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} \tag{3.4.39}$$

$$q_2 = \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} \tag{3.4.40}$$

$$q_3 = \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} - \cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2} \tag{3.4.41}$$

To compute the Euler angles back from the quaternion model, the following relation is used,

$$\phi = \tan^{-1} \frac{2\left(q_2 q_3 + q_0 q_1\right)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \tag{3.4.42}$$

$$\theta = \sin^{-1}\left(-2\left[q_1 q_3 - q_0 q_2\right]\right) \tag{3.4.43}$$

$$\psi = \tan^{-1} \frac{2\left(q_1 q_2 + q_0 q_3\right)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \tag{3.4.44}$$

Therefore, the aircraft rotational propagation equation in terms of quaternion variables can be modeled as,

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}_{\boldsymbol{B}} \cdot \begin{bmatrix} p_{\boldsymbol{K}} \\ q_{\boldsymbol{K}} \\ r_{\boldsymbol{K}} \end{bmatrix}_{\boldsymbol{B}} \tag{3.4.45}$$

The quaternion equation in Eq. (3.4.37) is commonly found when simulating the aircraft with high maneuverability, e.g., fighter aircraft. However, for civil aircraft where the degree of maneuverability is low, Eq. (3.4.37) is found in practice for representing the aircraft attitude propagation motion. Eqs. (3.4.37) and (3.4.45) conclude the last set of the aircraft equations of motion.

3.5 External Forces and Moments

External forces and moments influence aircraft's dynamic behavior both in translational and rotational motions. Aerodynamics, propulsion, and gravitational are three forces that exert on an aircraft in-flight. While the fourth force occurs due to the interaction between the landing gear and the runway surface when the aircraft moves on the ground, e.g., during the take-off or landing phase. Aerodynamic and propulsion forces are basically the same in nature as these two forces result from the interaction between the aircraft and the atmosphere. It is, therefore, the atmosphere's characteristics such as density, temperature, and pressure that influence the value of these two forces and consequently affect the aircraft's performance in general. The four forces and the corresponding moments need to be taken into account when building the aircraft equations of motion. For the sake of simplicity, only three quantities will be included when formulating external forces and moments, namely aerodynamics, propulsion, and gravitational. The friction force and moment will be specified as a special case, i.e., only when the aircraft moves on the ground. As indicated in Eqs. (3.4.13) and (3.4.27), the external forces and moments are represented by variable $\sum(\boldsymbol{F}^{\boldsymbol{G}})_{\boldsymbol{B}}$ and $\sum(\boldsymbol{M}^{\boldsymbol{G}})_{\boldsymbol{B}}$ consecutively. Taking into account all the external factors mentioned previously, variable $\sum(\boldsymbol{F}^{\boldsymbol{G}})_{\boldsymbol{B}}$ can therefore be written as,

$$\sum \left(\boldsymbol{F^G}\right)_{\boldsymbol{B}} = \left(\boldsymbol{F_A^G}\right)_{\boldsymbol{B}} + \left(\boldsymbol{F_P^G}\right)_{\boldsymbol{B}} + \left(\boldsymbol{F_G^G}\right)_{\boldsymbol{B}} \tag{3.5.1}$$

While the corresponding moment results from the external forces is written as,

$$\sum \left(\boldsymbol{M^G}\right)_{\boldsymbol{B}} = \left(\boldsymbol{M_A^G}\right)_{\boldsymbol{B}} + \left(\boldsymbol{M_P^G}\right)_{\boldsymbol{B}} + \underbrace{\left(\boldsymbol{M_G^G}\right)_{\boldsymbol{B}}}_{=0} \tag{3.5.2}$$

Note that, all forces and moments are denoted in the Body-Fixed frame. The sub-index $\boldsymbol{A}, \boldsymbol{P}, \boldsymbol{G}$ represent the aerodynamics, propulsion, and gravity respectively. There is no gravity moment acting on the aircraft, $(\boldsymbol{M_G^G})_{\boldsymbol{B}} = 0$, as the gravity force acts through the aircraft's center of gravity and the gravity field is assumed uniform.

3.5.1 Aerodynamics

Aerodynamics force and moment acting through the aerodynamic center $\boldsymbol{A}$ of the aircraft are generated by the relative motion of the air and the aircraft. In the Body-Fixed frame, the aerodynamic force is formulated as,

$$\left(\boldsymbol{F_A^A}\right)_{\boldsymbol{B}} = \begin{bmatrix} X_{\boldsymbol{A}}^{\boldsymbol{A}} \\ Y_{\boldsymbol{A}}^{\boldsymbol{A}} \\ Z_{\boldsymbol{A}}^{\boldsymbol{A}} \end{bmatrix}_{\boldsymbol{B}} \tag{3.5.3}$$

Components $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ consecutively denote axial, side, and normal aerodynamic forces. Each component can further be elaborated in terms of aerodynamic coefficients,

$$\begin{bmatrix} \boldsymbol{X_A^A} \\ \mathbf{Y}_{\boldsymbol{A}}^{\boldsymbol{A}} \\ \boldsymbol{Z_A^A} \end{bmatrix}_{\boldsymbol{B}} = \bar{q} \cdot S \cdot \begin{bmatrix} C_X \\ C_Y \\ C_Z \end{bmatrix} \tag{3.5.4}$$

Variable $\bar{q} = \frac{1}{2} \cdot \rho (V_{\boldsymbol{A}}^{\boldsymbol{A}})^2$ denotes the dynamic pressure due to the relative motion of the aircraft and the air, ρ is the air density, S is the wing reference area, $V_{\boldsymbol{A}}^{\boldsymbol{A}}$ is the airspeed and C_X, C_Y, and C_Z represent the aerodynamic coefficients of each force components in the Body-Fixed frame. The same as the force components, the aerodynamic moments can be modeled in the Body-Fixed frame as,

$$\left(\boldsymbol{M_A^A}\right)_{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{L_A^A} \\ \boldsymbol{M_A^A} \\ \boldsymbol{N_A^A} \end{bmatrix}_{\boldsymbol{B}} = \bar{q} \cdot S \cdot \begin{bmatrix} s \cdot C_l \\ \bar{c} \cdot C_m \\ s \cdot C_n \end{bmatrix} \tag{3.5.5}$$

Each component in the aerodynamic moments is required to be multiplied with the arm moment. In this case, wing span length (s) for moment is acting on the axial and normal direction, while mean aerodynamic chord ($\bar{c}$) for moment is acting on lateral side. Moment coefficients in this frame are denoted as C_l, C_m, and C_n which express the rolling, pitching, and yawing moment coefficients consecutively. The other common notations used to represent the aerodynamic forces and moments can also be denoted in the Aerodynamic frame. Eqs. (3.5.4) and (3.5.5) in the Aerodynamic frame are written as,

$$\left(\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{A}} = \begin{bmatrix} -\boldsymbol{D} \\ \boldsymbol{Q} \\ -\boldsymbol{L} \end{bmatrix}_{\boldsymbol{A}} = \bar{q} \cdot S \cdot \begin{bmatrix} -C_D \\ C_Q \\ -C_L \end{bmatrix} \tag{3.5.6}$$

$$\left(\boldsymbol{M}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{L}_{\boldsymbol{A}}^{\boldsymbol{A}} \\ \boldsymbol{M}_{\boldsymbol{A}}^{\boldsymbol{A}} \\ \boldsymbol{N}_{\boldsymbol{A}}^{\boldsymbol{A}} \end{bmatrix}_{\boldsymbol{A}} = \bar{q} \cdot S \cdot \begin{bmatrix} s \cdot C_{l\boldsymbol{A}} \\ \bar{c} \cdot C_{m\boldsymbol{A}} \\ s \cdot C_{n\boldsymbol{A}} \end{bmatrix} \tag{3.5.7}$$

Where variable $\boldsymbol{D}$, drag force, is the force component that is parallel to the aerodynamic velocity $\boldsymbol{V_A}$. Lift force, denoted as $\boldsymbol{L}$, is the force component perpendicular to the aerodynamic velocity vector $\boldsymbol{V_A}$, while variable $\boldsymbol{Q}$ or so-called side force is pointing to the right side from the drag-lift plane. In the non-dimensional form, these three forces are denoted by variables C_D, C_Q, and C_L consecutively. The negative sign in C_D, C_L, and the positive sign in C_Q account for normal conventions, i.e., drag is positive forward, side force is positive to the right and lift is positive up. The moment coefficients $C_{l\boldsymbol{A}}, C_{m\boldsymbol{A}}, C_{n\boldsymbol{A}}$ have the same naming as in the Body-Fixed frame but projected onto the Aerodynamic frame. The forces and moments transformation between the Body-Fixed to the Aerodynamic frame and vice versa is also possible through the matrix transformation. For example, a matrix $\mathbf{M}_{\boldsymbol{BA}}$ will transform any quantity in the Aerodynamic frame to the Body-Fixed frame as in,

$$\left(\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{B}} = \mathbf{M}_{\boldsymbol{BA}} \cdot \left(\boldsymbol{F}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{A}} \tag{3.5.8}$$

$$\left(\boldsymbol{M}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{B}} = \mathbf{M}_{\boldsymbol{BA}} \cdot \left(\boldsymbol{M}_{\boldsymbol{A}}^{\boldsymbol{A}}\right)_{\boldsymbol{A}} \tag{3.5.9}$$

where $\mathbf{M}_{\boldsymbol{BA}}$ matrix contains orientation angles $\alpha_{\boldsymbol{A}}$ and $\beta_{\boldsymbol{A}}$ between the Body-Fixed frame and the Aerodynamic frame. This matrix is expressed as,

$$\mathbf{M}_{\boldsymbol{BA}} = \begin{bmatrix} \cos\alpha_{\boldsymbol{A}} \cdot \cos\beta_{\boldsymbol{A}} & -\cos\alpha_{\boldsymbol{A}} \cdot \sin\beta_{\boldsymbol{A}} & -\sin\alpha_{\boldsymbol{A}} \\ \sin\beta_{\boldsymbol{A}} & \cos\beta_{\boldsymbol{A}} & 0 \\ \sin\alpha_{\boldsymbol{A}} \cdot \beta_{\boldsymbol{A}} & -\sin\alpha_{\boldsymbol{A}} \cdot \sin\beta_{\boldsymbol{A}} & \cos\alpha_{\boldsymbol{A}} \end{bmatrix} \tag{3.5.10}$$

The forces and moments of the aerodynamic coefficients are dimensionless and not constant during the flight. Typically, the factors affecting the aerodynamic coefficients are coming from the flight condition and configuration, such as the angle of attack, airspeed, and control surfaces deflection. These parameters cannot be measured using sensors equipped onboard but are possible to be estimated through other measured variables. This estimation process is one of the central themes in the field of aircraft system identification. The other theme in this field is determining the model structure of the investigated system. In the context of aircraft equations of motion, this activity related to the determination of the aerodynamic model, which will be discussed in details in the following section.

Aerodynamic Model

Formulation of the aerodynamic model along with the estimation of parameters involved in the model based on measured variables is one of the goals in the aircraft system identification. Modeling and estimation is an iterative process until the estimated values fulfill certain conditions such as having good statistical properties as well as having adequate values with respect to references. Typically, aerodynamic force and moment coefficients are influenced by factors such as the motion of the aircraft and control surfaces. Motion-induced factors include:

- Translational motion: angle of attack $(\alpha_{\boldsymbol{A}})$ and sideslip angle $(\beta_{\boldsymbol{A}})$ variables are used when the aerodynamic coefficient is modeled in the Aerodynamic frame, while variables $u_{\boldsymbol{B}}, v_{\boldsymbol{B}}, w_{\boldsymbol{B}}$ are used as replacement of $(\alpha_{\boldsymbol{A}}, \beta_{\boldsymbol{A}})$ when the aerodynamic coefficients are modeled in the Body-Fixed frame.
- Rotational motion: roll rate (p), pitch rate (q), yaw rate (r), angle of attack rate $(\dot{\alpha}_{\boldsymbol{A}})$, and sideslip rate $(\dot{\beta}_{\boldsymbol{A}})$.

While the control-induced factors may include aileron deflection (ξ), elevator deflection (η), rudder deflection (ζ), spoiler deflection $(\delta_{\text{spoiler}})$, and flap deflection (δ_{flaps}). The last factor that also influences the aerodynamic force and moment coefficients is categorized as non-dimensional quantities which includes Mach number (M) and Reynold number (R). Thus, the general formulation of the aerodynamic coefficient model can be modeled as a function of all the contributing factors and formulated as,

$$C_a = f(\alpha_{\boldsymbol{A}}, \beta_{\boldsymbol{A}}, p, q, r, \dot{\alpha}_{\boldsymbol{A}}, \dot{\beta}_{\boldsymbol{A}}, \xi, \eta, \zeta, \delta_{\text{spoiler}}, \delta_{\text{flaps}}, M, R) \tag{3.5.11}$$

where $a = \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}$ or $\boldsymbol{L}, \boldsymbol{Q}, \boldsymbol{D}$ and $\boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n}$.

Furthermore, the effect of the explanatory variables on the aerodynamic model depends on the flow type assumptions. Klein and Morelli [25] classify the aerodynamic model into quasi-steady flow and unsteady flow categories. In the quasi-steady flow, the past values of the explanatory variables $(u_{\boldsymbol{B}}, v_{\boldsymbol{B}}, w_{\boldsymbol{B}}, p, q, r, \dots)$ or $(\alpha_{\boldsymbol{A}}, \beta_{\boldsymbol{A}}, p, q, r, \dots)$ do not affect the aerodynamic forces and moments. This assumption presumes that the air flow reaches a steady state instantaneously. The aerodynamic model that belongs to this category is furthermore classified into the linear and nonlinear aerodynamic model. For the unsteady flow category, it is assumed that the past values of the explanatory variables influence the aerodynamic forces and moments, and therefore, their corresponding values must be restored. Each of the two aerodynamic model categories is discussed in the following section.

Quasi-Steady Flow

In general, the non-dimensional aerodynamic force and moment coefficients are a function of instantaneous values of the explanatory variables as well as their past values [25]. In the quasi-steady flow category, dependency on the past values of the explanatory variables is neglected by assuming that the steady state is reached instantaneously by the air flow. Consequently, the aerodynamic model is simplified from a function that depends on the entire history of the explanatory variables into a function that depends only on the current values, e.g., $\alpha(t), \eta(t)$. In this category, two aerodynamic models are considered, linear and nonlinear model. In the following, the linear aerodynamic model for quasi-steady flow will be considered first. While the nonlinear aerodynamic model, which will be modeled as polynomial or spline, is explained briefly afterward. For interested readers may refer to Klein [25] for further description of this aerodynamic model category.

Linear Aerodynamic Model

Modeling the aerodynamic force and moment coefficients that depends linearly with respect to the explanatory variables leads to the concept of stability and control derivatives [85, 86]. These variables are commonly referred to as the aerodynamic derivatives. The values that are represented by these aerodynamic derivatives only cover a certain point of the entire flight envelope and are assumed constant during the selected time window. A visual representation of these aerodynamic derivatives is depicted in Figure 3.5. As indicated by the figure, linearization of the nonlinear curve of the aerodynamic model leads to the linear dependency of the aerodynamic forces/moments coefficients with respect to the explanatory variables.

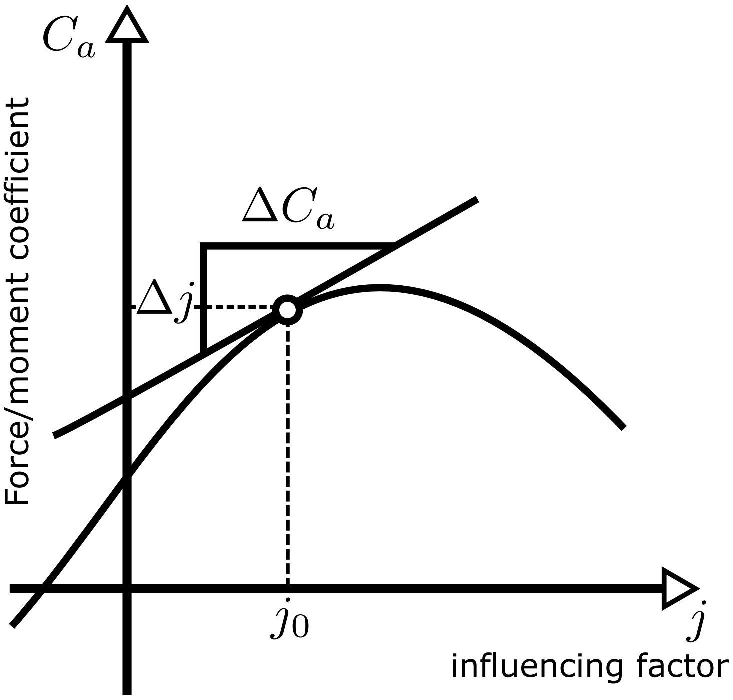


Figure 3.5: Visual representation of the aerodynamic derivative [77]

Furthermore, the stability derivative can be classified into static and dynamic stability categories. Static stability derivatives are associated to the air-relative velocity quantity such as $u_{\boldsymbol{B}}, v_{\boldsymbol{B}}, w_{\boldsymbol{B}},\ M, \alpha_{\boldsymbol{A}}, \beta_{\boldsymbol{A}}$. While the dynamic stability derivatives are derived with respect to angular rates such as p, q, r, and with additional variables $\dot{w}_{\boldsymbol{B}}, \dot{\alpha_{\boldsymbol{A}}}$ in the case of unsteady aerodynamics [25]. Control derivatives, as expressed by its name, are derivatives associated with the control devices such as aileron (ξ), elevator (η), rudder (ζ), and spoiler (δ_{spoiler}). In the linear aerodynamic model, dependency of coefficients on the explanatory variables as in Eq. (3.5.11) can be simplified by neglecting dependency between the symmetric (longitudinal) forces and moments on asymmetric (lateral) variables v, p, r and vice versa [25]. This assumption is commonly found in the flight dynamics books which is also adopted in this dissertation.

To compensate the dependency of the longitudinal forces and moments on the lateral motion, Greenberg [87] and Klein & Morelli [25] suggest to augment the model by adding two terms $C_{Z_{\dot{w}_{\boldsymbol{B}}}} \Delta \dot{w}_{\boldsymbol{B}}^*$ and $C_{m_{\dot{w}_{\boldsymbol{B}}}} \Delta \dot{w}_{\boldsymbol{B}}^*$ in the Body-Fixed frame which is equivalent by formulating them in terms of aerodynamic angle of attack as $C_{Z_{\dot{\alpha}_{\boldsymbol{A}}}} \Delta \dot{\alpha}_{\boldsymbol{A}}^*$ and $C_{m_{\dot{\alpha}_{\boldsymbol{A}}}} \Delta \dot{\alpha}_{\boldsymbol{A}}^*$. The necessity of adding these two variables into the model is to increase the correlation of the predicted model output with the observed aircraft longitudinal motion [87]. Consequently, the augmented terms should also be introduced into the aerodynamic model in the lateral motion, i.e., $C_{Y_{\dot{v}}} \Delta \dot{v}^*, C_{l_{\dot{v}}} \Delta \dot{v}^*$ and $C_{n_{\dot{v}}} \delta \dot{v}^*$ or equivalently in terms of sideslip angle β, the terms can be modeled as $C_{Y_{\dot{\beta}}} \Delta \dot{\beta}^*, C_{l_{\dot{\beta}}} \Delta \dot{\beta}^*$ and $C_{n_{\dot{\beta}}} \Delta \dot{\beta}^*$. However, Klein *et al.* stated that the terms affected by $\dot{v}$ and $\dot{\beta}$ can only be determined if the underlying flow is unsteady. Therefore, the augmentation by variables $\dot{v}$ and $\dot{\beta}$ will not be considered in the quasi-steady flow-based model. Thus, the linear aerodynamic model decouples in the longitudinal and lateral motion in the Body-Fixed frame are written as follows.
Longitudinal:

$$C_X = C_{X_0} + C_{X_u} \Delta u + C_{X_w} \Delta w + C_{X_q} q^* + C_{X_\eta} \Delta \eta \tag{3.5.12}$$

$$C_Z = C_{Z_0} + C_{Z_u} \Delta u + C_{Z_w} \Delta w + C_{Z_{\dot{w}}} \Delta \dot{w}^* + C_{Z_q} q^* + C_{Z_\eta} \Delta \eta \tag{3.5.13}$$

$$C_m = C_{m_0} + C_{m_u} \Delta u + C_{m_w} \Delta w + C_{m_{\dot{w}}} \Delta \dot{w}^* + C_{m_q} q^* + C_{m_\eta} \Delta \eta \tag{3.5.14}$$

Lateral:

$$C_Y = C_{Y_0} + C_{Y_v} \Delta v + C_{Y_p} p^* + C_{Y_r} r^* + C_{Y_\xi} \Delta \xi + C_{Y_\zeta} \Delta \zeta \tag{3.5.15}$$

$$C_l = C_{l_0} + C_{l_v} \Delta v + C_{l_p} p^* + C_{l_r} r^* + C_{l_\xi} \Delta \xi + C_{l_\zeta} \Delta \zeta \tag{3.5.16}$$

$$C_n = C_{n_0} + C_{n_v} \Delta v + C_{n_p} p^* + C_{n_r} r^* + C_{n_\xi} \Delta \xi + C_{n_\zeta} \Delta \zeta \tag{3.5.17}$$

Similarly, the aerodynamic models with different explanatory variables in the Aerodynamic frame are expressed as,

Longitudinal:

$$C_D = C_{D_0} + C_{D_M}\Delta M + C_{D_\alpha}\Delta\alpha + C_{D_q}q^* + C_{D_\eta}\Delta\eta \tag{3.5.18}$$
$$C_L = C_{L_0} + C_{L_M}\Delta M + C_{L_\alpha}\Delta\alpha + C_{L_{\dot\alpha}}\Delta\dot\alpha^* + C_{L_q}q^* + C_{L_\eta}\Delta\eta \tag{3.5.19}$$
$$C_m = C_{m_0} + C_{m_M}\Delta M + C_{m_\alpha}\Delta\alpha + C_{m_{\dot\alpha}}\Delta\dot\alpha^* + C_{m_q}q^* + C_{m_\eta}\Delta\eta \tag{3.5.20}$$

Lateral:

$$C_Q = C_{Q_0} + C_{Q_\beta}\Delta\beta + C_{Q_p}p^* + C_{Q_r}r^* + C_{Q_\xi}\Delta\xi + C_{Q_\zeta}\Delta\zeta \tag{3.5.21}$$
$$C_l = C_{l_0} + C_{l_\beta}\Delta\beta + C_{l_p}p^* + C_{l_r}r^* + C_{l_\xi}\Delta\xi + C_{l_\zeta}\Delta\zeta \tag{3.5.22}$$
$$C_n = C_{n_0} + C_{n_\beta}\Delta\beta + C_{n_p}p^* + C_{n_r}r^* + C_{n_\xi}\Delta\xi + C_{n_\zeta}\Delta\zeta \tag{3.5.23}$$

Nonlinear Aerodynamic Model

The linear aerodynamic model discussed previously is a good representation for a system with a small deviation from the reference (small perturbation). For a system with rapid excursion and large amplitudes, the linear model should be extended to a nonlinear model to cover larger condition. As an example, the lift coefficient in the nonlinear model can be extended up to the second series in the Taylor Series expansion, see Eq. (3.5.24). Only two dependent variables are included to demonstrate the expansion of the model. Other dependent variables can also be added into the model [25].

$$\begin{aligned} C_L = {} & C_{L_0} + C_{L_\alpha}\Delta\alpha + C_{L_q}\frac{q\bar{c}}{2V_0} \\ & + \frac{1}{2}\left[C_{L_{\alpha^2}}(\Delta\alpha)^2 + 2C_{L_{\alpha q}}\left(\Delta\alpha\frac{q\bar{c}}{2V_0}\right) + C_{L_{q^2}}\left(\frac{q\bar{c}}{2V_0}\right)^2\right] \end{aligned} \tag{3.5.24}$$

Another model representation of the nonlinear aerodynamic model is modeled by combining the static terms and treating the dynamic stabilities and control derivatives as function of the explanatory variables. Taking the example from the previous model, the nonlinear model would be,

$$C_L = C_{L_0}(\alpha) + C_{L_q}(\alpha)\frac{q\bar{c}}{2V_0} \tag{3.5.25}$$

The dependency of parameters on the explanatory variables can be formulated in a polynomial or spline model. For further investigation of this model, interested readers may refer to [25].

Unsteady Flow

In the quasi-steady flow aerodynamic model, it is assumed that the parameters are time-invariant and the dependency of the parameters with respect to the past explanatory variables are ignored. In the unsteady flow, model formulation takes these two assumptions into account by assuming that the aerodynamic parameters are time-varying and depending on the current and past explanatory variables. This unsteady flow aerodynamic linear model was first introduced by Tobak [88] and extended later by Tobak and Schiff [89] by introducing dependency of parameters with respect to state and input variables. The general formulation of this approach is expressed as [25, 89],

$$
\begin{aligned}
C_a(t) = C_a(\infty) &+ \int_0^t C_{a_{\xi_1}}\left[t-\tau;\xi(\tau)\right]^T \frac{d}{d\tau}\xi_1(\tau)d\tau \\
&+ \frac{l}{V}\int_0^t C_{a_{\xi_2}}\left[t-\tau;\xi(\tau)\right]^T \frac{d}{d\tau}\xi_2(\tau)d\tau
\end{aligned} \tag{3.5.26}
$$

where,

a: $\boldsymbol{D}, \boldsymbol{L}, \boldsymbol{Q}, \boldsymbol{m}, \boldsymbol{l}$, or $\boldsymbol{n}$

$C_a(t)$: aerodynamic force and moment coefficient

$C_a(\infty)$: steady-state value of the aerodynamic force or moment coefficient

$C_{a_\xi}(t)$: vector of indicial functions, each element of the vector is the response of C_a to a unit step in an element of ξ

ξ_1: $[\alpha, \beta]^T$

ξ_2: p, q, r

ξ : $[\xi_1^T \;\; \xi_2^T]^T = [\alpha \;\; \beta \;\; p \;\; q \;\; r]^T$

l : characteristics length, $l = \bar{c}/2$ or $l = s/2$

Explanation of the unsteady aerodynamic model presented in Eq. (3.5.26) can be further found in [25, 89].

3.5.2 Gravity

The gravitational force is assumed to be constant in both magnitude and direction relative to the NED frame. This force is also assumed to be acting only along the z_O-axis of the NED frame. However, in the Body-Fixed frame, the effect of the gravitational force varies depending on the orientation angles of the Body-Fixed frame with respect to the NED frame. The orientation between the Body-Fixed frame relative to the NED frame can be described by the Euler angles ϕ, θ, ψ formulated in matrix transformation $\mathbf{M}_{\boldsymbol{BO}}$. Thus, the transformation of gravitational force from the NED frame to the Body-Fixed frame can be written as,

$$
\begin{aligned}
\left(\boldsymbol{F}_{\boldsymbol{G}}^{\boldsymbol{G}}\right)_{\boldsymbol{B}} &= \mathbf{M}_{\boldsymbol{BO}} \cdot \left(\boldsymbol{F}_{\boldsymbol{G}}^{\boldsymbol{G}}\right)_{\boldsymbol{O}} \\
&= \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_{\boldsymbol{O}} \\
&= \begin{bmatrix} -g\sin\theta \\ mg\sin\phi\cos\theta \\ mg\cos\phi\cos\theta \end{bmatrix}_{\boldsymbol{B}}
\end{aligned} \tag{3.5.27}
$$

3.5.3 Propulsion

A simple formulation of the propulsion system is done by assuming its thrust line aligns with the x-axis of the Body-Fixed frame. With this assumption, the thrust produced by the propulsion system only affects the equation of motion in the longitudinal axis of the Body-Fixed frame. Therefore, the force model of the propulsion system denoted in the Body-Fixed frame can be written as,

$$\left(\boldsymbol{F_P^G}\right)_{\boldsymbol{B}} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}_{\boldsymbol{B}} \tag{3.5.28}$$

Thrust variable T in Eq. (3.5.28) depends on the propulsion system type employed in the aircraft such as propeller, turbofan, or jet based propulsion system. Since the reference point of the propulsion system is not from aircraft's center of gravity $\boldsymbol{G}$ but from the propulsion system reference point $\boldsymbol{P}$, it will generate moment with respect to point $\boldsymbol{G}$, denoted as,

$$\left(\boldsymbol{M_P^G}\right)_{\boldsymbol{B}} = \left(\boldsymbol{r^{GP}}\right) \times \left(\boldsymbol{F_P^G}\right)_{\boldsymbol{B}} \tag{3.5.29}$$

Variable $(\boldsymbol{r^{GP}})$ denotes the position between propulsion system reference point $\boldsymbol{P}$ relative to aircraft's center of gravity $\boldsymbol{G}$ in vector notation. If the propulsion is assumed not acting along the x-axis of the Body-Fixed frame, Eq. (3.5.28) needs to be modified by taking its orientation into account with respect to the other frame. In this case, a new reference frame called Propulsion frame is introduced. This reference frame with respect to the Body-Fixed frame is described by two orientation angles, i.e., thrust elevation angle σ and engine mounting angle κ. With these two orientation angles, the matrix transformation from the Propulsion frame to the Body-Fixed frame is defined as,

$$\mathbf{M}_{\boldsymbol{BP}} = \begin{bmatrix} \cos\kappa \cdot \cos\sigma & -\sin\kappa & \cos\kappa \cdot \sin\sigma \\ \sin\kappa \cdot \cos\sigma & \cos\kappa & \sin\kappa \cdot \sin\sigma \\ -\sin\sigma & 0 & \cos\sigma \end{bmatrix} \tag{3.5.30}$$

Consequently, the thrust model that acts aligned with the x-axis of the Propulsion frame is transformed to the Body-Fixed frame by multiplying it with Eq. (3.5.30) yields in,

$$\left(\boldsymbol{F_P^G}\right)_{\boldsymbol{B}} = \mathbf{M}_{\boldsymbol{BP}} \cdot \left(\boldsymbol{F_P^G}\right)_{\boldsymbol{P}} \tag{3.5.31}$$

The moment equation is also changed accordingly by plugging Eq. (3.5.31) into the thrust variable in Eq. (3.5.29).

3.6 Collected Equations of Motion

Based on the assumptions used to derive aircraft equations of motion in Section 3.4.1 - 3.5, the following summarizes the equations of motion both in the Body-Fixed frame and the Aerodynamic frame. As indicated in Eq. (3.4.14), the external forces $(\boldsymbol{X^G}, \boldsymbol{Y^G}, \boldsymbol{Z^G})$ in the Body-Fixed frame consist of aerodynamic, gravity, and propulsion forces denoted by,

$$(\boldsymbol{X^G})_{\boldsymbol{B}} = \bar{q}SC_X - mg\sin\theta + T \tag{3.6.1}$$

$$(\boldsymbol{Y^G})_{\boldsymbol{B}} = \bar{q}SC_Y + mg\cos\theta\sin\phi \tag{3.6.2}$$

$$(\boldsymbol{Z^G})_{\boldsymbol{B}} = \bar{q}SC_Z + mg\cos\theta\cos\phi \tag{3.6.3}$$

Plugging Eqs. (3.6.1) - (3.6.3) back into Eq. (3.4.14) and taking the assumptions defined in Section (3.3) into account leads to a complete set of aircraft translational equation of motions in the Body-Fixed frame.
Translational equations:

$$\dot{u} = rv - qw + \frac{\bar{q}S}{m}C_X - g\sin\theta + \frac{T}{m} \tag{3.6.4}$$

$$\dot{v} = pw - ru + \frac{\bar{q}S}{m}C_Y + g\cos\theta\sin\phi \tag{3.6.5}$$

$$\dot{w} = qu - pv + \frac{\bar{q}S}{m}C_Z + g\cos\theta\cos\phi \tag{3.6.6}$$

The same principles also apply to the rotational equations of motion. The external moments $(\boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})_B$ on the Body-Fixed frame as formulated in Eqs. (3.6.7 - 3.6.9),

$$(\boldsymbol{l^G})_{\boldsymbol{B}} = \bar{q} \cdot S \cdot C_l \cdot s \tag{3.6.7}$$

$$(\boldsymbol{m^G})_{\boldsymbol{B}} = \bar{q} \cdot S \cdot C_m \cdot \bar{c} \tag{3.6.8}$$

$$(\boldsymbol{n^G})_{\boldsymbol{B}} = \bar{q} \cdot S \cdot C_n \cdot s \tag{3.6.9}$$

are plugged back into Eq. (3.4.27). Note that, the rotational effect from engine is assumed to be small and ignored from the model. The final form of the aircraft rotational equations of motion in the Body-Fixed frame are formulated as [25],
Rotational equations:

$$\dot{p} = (c_1 r + c_2 p) \cdot q + s\bar{q}S \cdot (c_3 C_l + c_4 C_n) \tag{3.6.10}$$

$$\dot{q} = (c_5 p) \cdot r - c_6 \cdot (p^2 - r^2) + c_7 \bar{q}S\bar{c}C_m \tag{3.6.11}$$

$$\dot{r} = (c_8 p - c_2 r) \cdot q + s\bar{q}S \cdot (c_9 C_n + c_4 C_l) \tag{3.6.12}$$

where $c_1 \cdots c_9$ represent the inertia constants dependent on the body-axis moments of inertia,

$$\begin{aligned} c_1 &= [(I_y - I_z)I_z - I_{xz}^2]/\mathit{\Gamma} & c_6 &= I_{xz}/I_y \\ c_2 &= [(I_x - I_y + I_z) \cdot I_{xz}]/\mathit{\Gamma} & c_7 &= 1/I_y \\ c_3 &= I_z/\mathit{\Gamma} & c_8 &= [(I_x - I_y)I_x - I_{xz}^2]/\mathit{\Gamma} \\ c_4 &= I_{xz}/\mathit{\Gamma} & c_9 &= I_x/\mathit{\Gamma} \\ c_5 &= (I_z - I_x)/I_y & \mathit{\Gamma} &= I_x I_z - I_{xz}^2 \end{aligned} \tag{3.6.13}$$

Kinematic/attitude propagation equations:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{3.6.14}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{3.6.15}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{cos\theta} \tag{3.6.16}$$

Position propagation equations:

$$(V_{\boldsymbol{K}})_{\boldsymbol{O}}^{\boldsymbol{E}} = \mathbf{M}_{\boldsymbol{OB}} \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}_{\boldsymbol{B}}^{\boldsymbol{E}} \tag{3.6.17}$$

Formulating the translational equations in the Aerodynamics frame are done by replacing state variables (u, v, w) with (V, α, β) and transforming the forces from the Body-Fixed frame to the Aerodynamic frame, yields in [25]
Translational equations:

$$\begin{aligned} \dot{V} &= -\frac{\bar{q}S}{m}C_{D_W} + \frac{T}{m}\cos\alpha\cos\beta + g(\cos\phi\cos\theta\sin\alpha\cos\beta) \\ &\quad + \sin\phi\cos\theta\sin\beta - \sin\theta\cos\alpha\cos\beta) \end{aligned} \tag{3.6.18}$$

$$\begin{aligned} \dot{\alpha} &= -\frac{\bar{q}S}{mV\cos\beta}C_L + q - \tan\beta(p\cos\alpha + r\sin\alpha) - \frac{T\sin\alpha}{mV\cos\beta} \\ &\quad + \frac{g}{V\cos\beta}(\cos\phi\cos\theta\cos\alpha + \sin\theta\sin\alpha) \end{aligned} \tag{3.6.19}$$

$$\begin{aligned} \dot{\beta} &= \frac{\bar{q}S}{mV}C_{Y_W} + p\sin\alpha - r\cos\alpha + \frac{g}{V}\cos\beta\sin\phi\cos\theta \\ &\quad + \frac{\sin\beta}{V}\left(g\cos\alpha\sin\theta - g\sin\alpha\cos\phi\cos\theta + \frac{T\cos\alpha}{m}\right) \end{aligned} \tag{3.6.20}$$

where [25],

$$C_{D_W} = C_D\cos\beta - C_Q\sin\beta \tag{3.6.21}$$

$$C_{Y_W} = C_Q\cos\beta + C_D\sin\beta \tag{3.6.22}$$

Aerodynamic coefficient C_L and C_D can be computed through the aerodynamic model as formulated in Eqs. (3.5.18) and (3.5.19) or through transforming the aerodynamic model in the Body-Fixed frame to the Aerodynamic frame as [25],

$$C_L = -C_Z \cos\alpha + C_X \sin\alpha \tag{3.6.23}$$

$$C_D = -C_X \cos\alpha - C_Z \sin\beta \tag{3.6.24}$$

The rotational and attitude propagation model take the same form as in the Body-Fixed frame while the position propagation model is computed through states V, α, β instead of u, v, w as in the Body-Fixed frame case [25].
Position propagation:

$$\begin{aligned}
\dot{x}_O &= V\cos\alpha\cos\beta\cos\phi\cos\theta + V\sin\beta(\cos\phi\sin\theta\sin\phi - \sin\phi\cos\phi) \\
&\quad + V\sin\alpha\cos\beta(\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi) && (3.6.25) \\
\dot{y}_O &= V\cos\alpha\cos\beta\sin\psi\cos\theta + V\sin\beta(\sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi) \\
&\quad + V\sin\alpha\cos\beta(\sin\psi\sin\theta\sin\phi - \cos\psi\sin\phi) && (3.6.26) \\
\dot{z}_O &= V\cos\alpha\cos\beta\sin\theta - V\sin\beta\cos\theta\sin\phi - V\sin\alpha\cos\beta\cos\theta\cos\phi && (3.6.27)
\end{aligned}$$

3.7 Output Equation

In the context of aircraft system identification, the output model provides an analytical connection between the measured variables and the aircraft states, state derivatives, and controls variables. The output variables obtained from the output model are directly related to the measured variables such as airspeed (V), angle of attack (α), angle of sideslip β, Euler angles (ϕ, θ, ψ), position (x_O, y_O, z_O), accelerations (a_x, a_y, a_z), and body angular rate ($\dot{p}, \dot{q}, \dot{r}$). In general, the output model can be formulated as the function of states, states derivatives and control variables,

$$\boldsymbol{y} = h(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{u}) \tag{3.7.1}$$

However, the exact form of the output model for each variable depends on the aircraft's sensors and how the state variables are formulated. For example, if the state variables are modeled in the Aerodynamic frame, then the output model takes a simple form as the sensors used in the measurements are aligned with the Aerodynamic frame which is the reference frame of the state variables. The output models are simply modeled as [18, 25],

$$V = V \tag{3.7.2}$$

$$\alpha = \alpha \tag{3.7.3}$$

$$\beta = \beta \tag{3.7.4}$$

If the force equations are formulated in the Body-Fixed frame, the output model is adapted to accommodate the variables provided from the Body-Fixed frame. The output model is transformed into [18, 25],

$$V = \sqrt{u^2 + v^2 + w^2} \tag{3.7.5}$$

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \tag{3.7.6}$$

$$\beta = \sin^{-1}\left(\frac{v}{V}\right) \tag{3.7.7}$$

The same principle also applies when modeling the output model for states and state derivatives $p, q, r, \phi, \theta, \psi, \dot{p}, \dot{q}, \dot{r}, x_O, y_O$ and z_O since all these variables are available directly from the sensors as well as from the aircraft equations of motion. Modeling the accelerometers' output requires a special treatment since these variables are not directly modeled in the aircraft equations of motion. In addition to that, the accelerometers only measure the translational acceleration due to the external forces, excluding gravity. In vector notation, the output model for the translational acceleration can be formulated as,

$$a = \dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \frac{\boldsymbol{F_G}}{m} = \frac{\boldsymbol{F_A} + \boldsymbol{F_P}}{m} \tag{3.7.8}$$

or in each component of the Body-Fixed frame take form as,

$$a_x = \dot{u} - rv + qw + g\sin\theta \tag{3.7.9}$$

$$a_y = \dot{v} - pw + ru - g\cos\theta\sin\phi \tag{3.7.10}$$

$$a_z = \dot{w} - qu + pv - g\cos\theta\cos\phi \tag{3.7.11}$$

The accelerometers output can also be modeled in terms of the external forces as these sensors measure the change of speed due to the applied forces. In the Body-Fixed frame, the output model for the acceleration variables take form as [25],

$$a_x = \frac{\bar{q}SC_x + T}{m} \tag{3.7.12}$$

$$a_y = \frac{\bar{q}SC_Y}{m} \tag{3.7.13}$$

$$a_z = \frac{\bar{q}SC_Z}{m} \tag{3.7.14}$$

Chapter 4

System Identification Theory

The following chapter presents the foundation theory of the system identification method, which serves as a basis for analysis in this dissertation. This chapter is started by introducing the background of the System Identification in general, including definitions, the goals, and the needs. This general introduction will then be extended to flight vehicle field, which is the main subject of this dissertation. After briefly going into System Identification, two estimation theories namely, Maximum Likelihood for parameter estimation and Kalman Filter for state estimation will be elaborated. This chapter is concluded by introducing several common techniques used in the flight vehicle system identification such as Equation Error Method, Output Error Method, Filter Error Method and the extension of Kalman Filter for state and parameter estimation.

4.1 System Identification

The importance of modeling and system analysis has significantly increased in many fields such as ecology, biology, economy, engineering, and certainly in the field related to the flight vehicle. Design of a mathematical model of a system is necessary to fathom its dynamic behavior and to enable an analyst to carry out system analysis. In general, modeling is defined as an abstraction process of a real system. The abstracted model may be logical or mathematical. In the context of system identification, the abstracted model refers to a mathematical model which describes system behavior using variables and parameters in a set of functions [90]. The variables and parameters involved in the model represent properties of the system under investigation. Meanwhile, the functions describe the relationship between the variables and parameters, usually in the form of algebraic or differential equations.

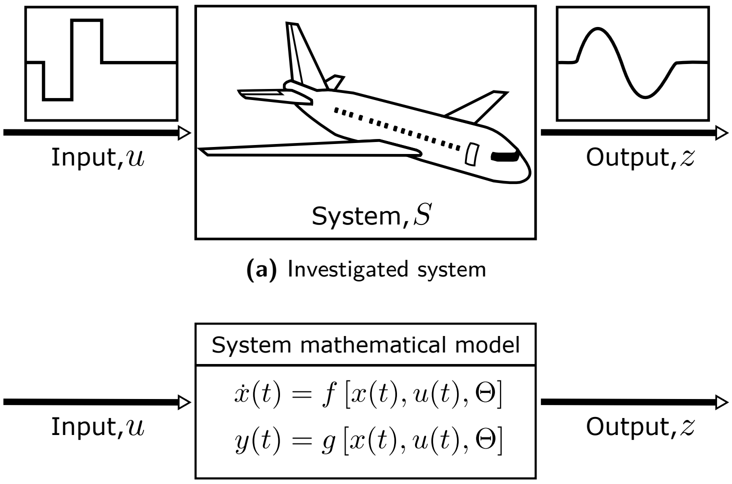
In general, there are three approaches to develop a mathematical model of a system. The first approach is related to a purely theoretical approach, which is driven by the physical relationships of the system. The second approach is related to a purely empirical approach based on experiments on the already existing system. The last approach of a model building is through a sensible combination of the theoretical and empirical approach. The theoretical model building comes into play if an experiment of the respective system cannot or must not be carried out. If the plant to be modeled does not yet exist, theoretical modeling is the only possible way to obtain a mathematical model of the plant. However, if an experimental analysis of a process is performed, the input and output signals are measured. The measurements are then evaluated in an identification procedure yielding a mathematical model of the investigated process. Typically, the developed mathematical model from this identification procedure only covers the specific behavior of the system since the development of the model is purely based on the limited measurement data. Based on the mentioned process, system identification – regardless of the field and plant to be modeled – can be defined as an activity of developing a mathematical model for physical systems based on imperfect observations or measurements [25]. It is also commonly defined as a scientific discipline that provides answers to the age-old "inverse problem" because it is the opposite of the problem of computing the response of a system with known characteristics [91]. Therefore, based on its definition, system identification mainly deals with building a mathematical model of a system. A more practical definition of system identification is coined by Lofti A. Zadeh [92] as,

> *"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."*

This definition implicitly stated that the developed mathematical model of a physical system is not unique, and the observations are corrupted by noise. The non-uniqueness of the mathematical model developed from an experiment leads to the question of the model adequateness. The guiding principle for model selection is known as parsimony principle or Ockham's razor, which states that of all models in a specified class that exhibit the desired characteristics, the simplest one should be preferred [18]. However, the most important requirement for a mathematical model is that it should be useful in some way, meaning that sometimes the model is used to predict some aspects of a physical system's behavior, while at other times the values of parameters in the model are sufficient to provide the desired insight. In other cases, a synthesized model must be simple and useful, yet at the same time complex enough to capture the essential dependencies and features embodied in the measurements.

The latter definition, as quoted from Zadeh [92], also stated that system identification is carried out based on the observations of the system's input and output under the investigation. Data gathered from measurement activities contain errors which add complexity in modeling the system. To deal with such situation, estimation theory that is commonly used to extract system parameters under noisy data is required. Thus, estimation theory in general and its corresponding derivative methods are introduced in Section 4.4 and 4.5. In the context of system theory, system identification is one of the three types of general problems encountered in this field. The other two problems being simulation and control. All these types of problems involve three main components that describe a system, they are input $(\boldsymbol{u})$, output $(\boldsymbol{y})$ or $(\boldsymbol{z})$ and system model (S), see Figure 4.1. The three problems in system theory are described as follows [25, 93]:

1. **Simulation**. The problem addressed in simulation by finding the output ($\boldsymbol{y}$) given the input ($\boldsymbol{u}$) and the system (S).
2. **Control**. The control problem deals with finding the input ($\boldsymbol{u}$) given the system model (S) and the output ($\boldsymbol{y}$).
3. **System identification**. In system identification problem, the goal is to find the system model (S) given the system input ($\boldsymbol{u}$) and output/measurement ($\boldsymbol{z}$).


Input,u
System,S
Output,z
System mathematical model
$\dot{x}(t) = f\left[x(t), u(t), \Theta\right]$
$y(t) = g\left[x(t), u(t), \Theta\right]$
Input,u
Output,z

(a) Investigated system

(b) System representation in a mathematical model

Figure 4.1: System and its general abstraction in a mathematical model [93]

In this dissertation, nomenclature f is used to represent a general state model of the system while g is used to denote the system's output model in which both models are functions of internal state variables ($\boldsymbol{x}$), input ($\boldsymbol{u}$) and system parameters ($\boldsymbol{\Theta}$). As already pointed out, a mathematical model developed through system identification is not unique, meaning that different model forms may have to be investigated. For a specified model, parameters involved in the system model are quantified by an activity called "parameter estimation." A statistical procedure is involved in the estimation of unknown parameters by matching the model response and system measurements. The system response is computed through a numerical procedure called "simulation" for given inputs and the model candidate. In this simulation step, the model structure and the related parameters are known a priori and held fixed in the span of simulation time. Based on the mentioned processes, it is apparent that system identification, parameter estimation, and simulation are three interrelated aspects where the last two procedures are an integral part of the overall system identification. This coupled-process is visually depicted in Figure 4.2. To assess model fidelity, a procedure called "model validation" is conducted to the identified model and the estimated parameters. If the estimated parameters and the identified model do not meet the requirements, e.g., the model response does not match the measurement or the estimated parameters values are unrealistic physically, then the whole process needs to be repeated. This process indicates that system identification is not a one-shot procedure but requires an iterative step to find an adequate model of the investigated system [94]. The iterative procedure may involve a major change to the postulated model, for example, by introducing more parameters or discarding low response parameters from the model. In a worst-case scenario, such an iterative procedure may require new experimental data with different input strategy. In flight vehicle system identification, experimental setup or termed as flight testing in a flight vehicle development is an integral part of system identification which one of its goals is to record high-quality data from the in-flight measurement. Commonly, a specific flight maneuver and dedicated control input are designed to be carried out during a flight test to increase information content in the data that brings to the success of system identification activity [20, 95, 96, 97].

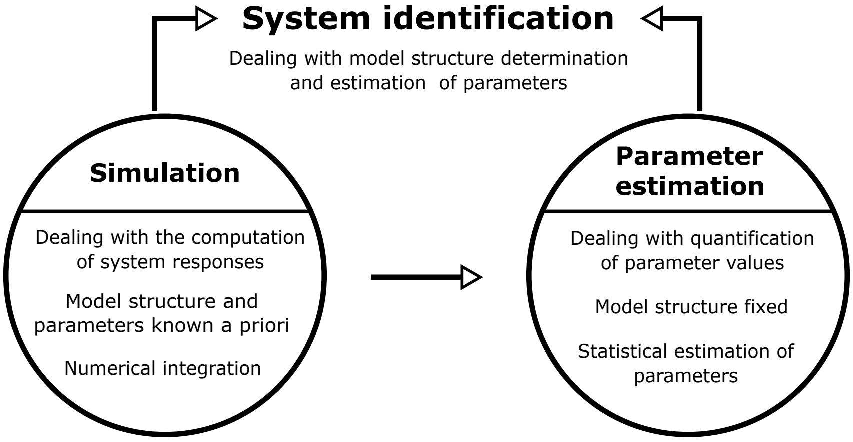

System identification
Dealing with model structure determination and estimation of parameters
Simulation
Dealing with the computation of system responses
Model structure and parameters known a priori
Numerical integration
Parameter estimation
Dealing with quantification of parameter values
Model structure fixed
Statistical estimation of parameters

Figure 4.2: Parameter estimation and simulation are an integral part of system identification process

In general, a mathematical model developed based on system identification procedure can be applied for the following purposes [98]:

- To obtain a better knowledge of the process
- Verification of theoretical models
- Synthesis of control systems
- Prediction of signals
- Optimization of process behavior
- Computation of variables which are not directly measurable

4.2 Model Classification

"All models are wrong, but some are useful" is an aphorism that was first coined by statistician George E. P. Box [99]. The idea behind this aphorism is that there is no model that can perfectly capture the behavior of a system since the model is only an approximation of the system behavior. However, a good approximation of the model may provide a useful application. Models, based on the process deriving a model, can be categorized into two types [100]:

- Phenomenological model
- Behavioral model

Phenomenological model, also known as a knowledge-based model, is built based on fundamental principles and theoretical formulation that describes the empirical relationship of phenomena involved in the system. In other words, a phenomenological model tries to explain why and how variables interact in a system. The fact that this model is formulated from physical considerations leads to parameters having physical interpretations. Therefore, the postulated model is usually represented in ordinary or partial differential equations. Each part in the model is a representation of a system's sub-process. Such models might consist of many equations and turn out to be highly complex models as they try to capture the overall system behavior.

However, from a practicality point of view, a highly complex model is rarely used in a real application as this model generally time-consuming. Furthermore, in system analysis, only some parts of the system at some specific conditions are required. Thus, a reduced model with shorter simulation time is preferred in practice as it can resemble some parts of the system's characteristics. This approach is commonly used in system modeling of engineering problem where the system under investigation is treated as simple as possible by limiting inputs to the system and selecting test environment which leads to a simple model representation. Behavioral models, on the other hand, provide a more straightforward approach to capture system behavior through a cause-effect relationship. In this model type, no physical interpretation of the parameters involved in the model. During the development of the model, no prior knowledge required, and it is even not necessary to know what the inputs and outputs stand for or in what units they are expressed. In principle, a model that captures the system behavior is fully developed based on the information from the data. The more data used to develop the model, the more spectrum of a system is covered by the model. A behavioral model is suitable for a complex system in which input-output is the only concern without considering how the internal system processes the input to generate the output. Usually, an artificial neural network (ANN) is commonly used to develop this type of model. Table 4.1 summarizes the distinctions between these two types of models [100].

	Phenomenological models	Behavioral models
Parameters	physical meaning	no concrete meaning
Simulation	long and difficult	quick and easy
Prior information	included	neglected
Validity domain	large (if structure is correct)	restricted

Table 4.1: White-box vs black-box model

Furthermore, the phenomenological model can be classified into two categories: parametric models and non-parametric models. Parametric models assume some finite set of parameters in a suitable model structure and order, given observed data. That is, the finite set of parameters in a model tries to capture everything there is to know about the data. The complexity of the model is bounded even if the amount of data is unbounded, which makes the parametric models not quite flexible since a different test condition might require a different model structure. However, this model type is very often used in practice because it provides a simple model structure and acceptable model predictions, and it is also fast to develop. The parametric models might be represented in a different form such as linear or nonlinear, continuous or discontinuous, time-invariant or time-variant, and deterministic or stochastic. In non-parametric models, data distribution cannot be defined in terms of a finite set of parameters but usually defined by assuming infinite dimensional parameters, e.g., a parameter is assumed as a function. In this way, the amount of information that the parameters can capture on data can grow as the amount of data grows. This property makes the non-parametric models more flexible than the parametric models. Impulse or step response, frequency response, or power spectral density models belong to non-parametric models which do not have any specific model structure or order. Besides model classification which is previously explained, models can also be categorized into three types [18, 93]:

- White-box model
- Black-box model
- Grey-box model

White-box model related to the phenomenological models, while black-model corresponds to the behavioral models. The last type of models in this category, grey-box model, is a combination of the two variants. In the case of aircraft system identification and specifically on this dissertation, we will be dealing with a parametric model which belongs to the phenomenological/white-box model category. Therefore, the aircraft equations of motion introduced in Chapter 3 belongs to this category. The following Figure 4.3 summarizes the model classification. Note that, the bold black boxes borders indicate the flow of the model types used in this dissertation.

4.3 A Flight Vehicle System Identification

The ultimate goal of system identification is to postulate an adequate model of a system in a specific test condition. In the context of a flight vehicle, it is the model that captures the dynamic behavior of an aircraft in an adequate and validated mathematical model representation. Postulating a model in flight vehicle development cycle requires a lot of efforts and resources. Sensors with high accuracy, dedicated flight maneuvers, and control input for the sake of having data with good quality/rich information contents are some of the efforts meant for system identification. Furthermore, data with high information content should also be supported with a suitable selection of estimation methods in order to have a good estimate. For example, implementing an estimation method that can only deal with measurement noise on data which contain process and measurement noises might lead to inaccurate or high uncertainty in the estimates. Therefore, knowing the quality of the data used in the analysis and implementing the correct method are essential in the aircraft system identification.

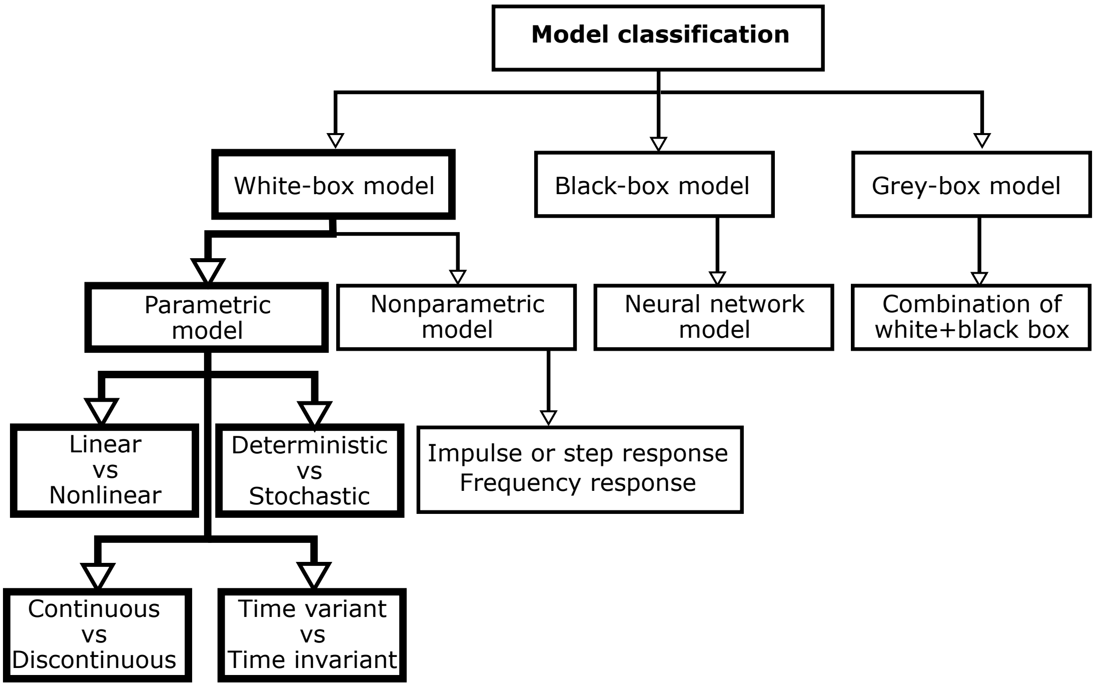


Figure 4.3: Model classification [93]

Having postulated the model as well as estimated the unknown parameters, the next process is to validate the result. In this step, some complementary data might be required in order to validate the estimates from the data used in the identification process. All these processes are strongly interdependent and must be treated carefully. It is also not guaranteed that the analyst will produce a good model and parameters in a single run, but it is also possible that the model would need to be modified and the estimation process needs to be started again from the beginning which is a very common case found in this field. In a worst-case scenario where the system identification method fails to get a good result due to bad data quality, the data collection step might need to be performed. In aircraft system identification, this complex and iterative process may be summarized into four essential aspects, namely [19]:

- **Maneuvers** - related to designing optimal control inputs (aileron, elevator, rudder, throttle) that excite all modes of aircraft dynamic motion.
- **Measurement** - related to the selection of sensors' equipment that can record data in high accuracy and high sampling rate.
- **Models** - related to defining possibility structures of the system's mathematical model.
- **Method** - related to the selection of a suitable estimator and optimization algorithm with respect to data quality.

All four aspects should be treated carefully for the success of aircraft system identification. Hamel *et al.* [19] summarize these four essential aspects as "Quad-M" along with procedures commonly found in flight vehicle system identification. This "Quad-M" is depicted in Figure 4.4. Once the model developed, it can serve many purposes, which in many cases, e.g., during aircraft development, contributes to budget efficiency. Not only in flight vehicle development but also in other fields, such as flight incident/accident investigation, system identification

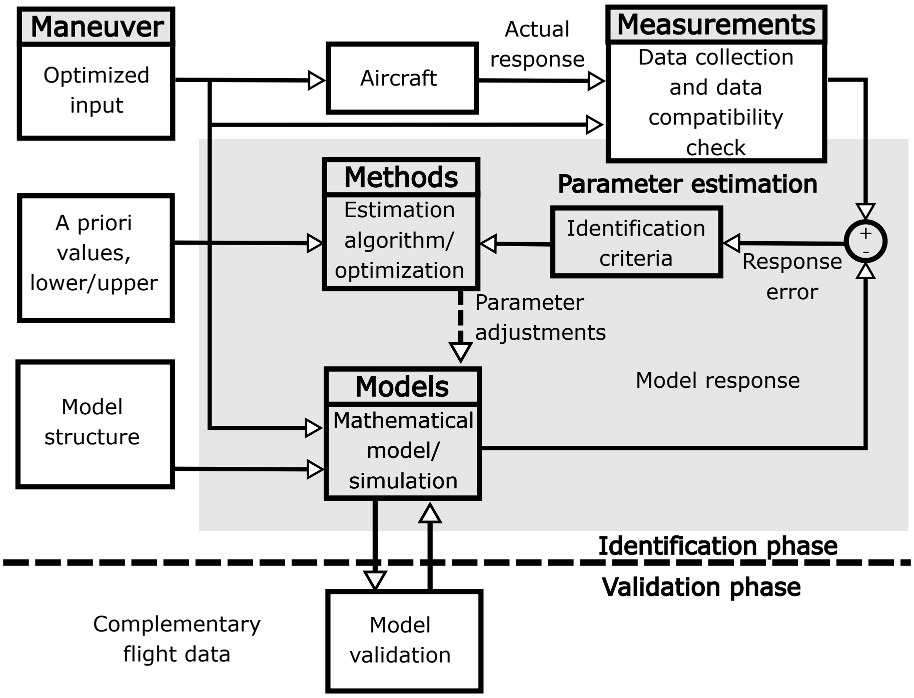


Figure 4.4: Quad-M procedures in aircraft system identification [18, 19]

plays a vital role, for instance, in data compatibility check/flight path reconstruction which mainly aims at enhancing FDR data quality by utilizing a combined aircraft kinematics model and filtering algorithm.

Furthermore, Jategaonkar *et al.* [101] summarize some important applications of system identification on flight vehicle, i.e., to produce a model that is used to:

- understand the cause-effect relationship that underlies a physical phenomenon,
- design flight control laws including stability augmentation systems,
- investigate system performance and characteristics,
- verify wind-tunnel and analytical predictions,
- support flight envelope expansion during prototype testing,
- derive high-fidelity and high-bandwidth models for in-flight simulators,
- develop a high-fidelity aerodynamic database for flight simulators which meets FAA fidelity requirements,
- reconstruct the flight path trajectory, including wind estimation and incident analysis,
- perform fault-diagnosis and adaptive control or reconfiguration,
- analyze handling qualities specification compliance.

With the broad spectrum of model applications and its importance in the aviation field, system modeling or identification becomes an integral and important part in the flight vehicle development cycle [18].

Mathematical Model Extensions

Aircraft dynamic model is developed based on Newtonian mechanics where ordinary differential equations and parameters that have physical meaning are parts of the model. In Chapter 3, aircraft equations of motion have been introduced. Four sets of models, in which models that represent translational, rotational, position, and attitude propagations have been derived by assuming aircraft as a rigid-body model. Through these models, it is now possible to obtain the propagation of models' states. Usually, states propagation with respect to time is carried out by solving initial value problem through numerical integration procedures. This simulation can be done as the underlying model is assumed to be free of noise or so-called deterministic system. However, in the case of models developed through the system identification procedure, this assumption is not valid anymore since the data used for building the model contain errors. Errors such as deterministic error (bias) or stochastic error (noise) which can further be classified into process noise and measurement noise may corrupt data obtained from the measurement. The same case happens for data obtained from a flight test where errors that come from atmospheric turbulence (process noise) and sensor errors (measurement noise) along with biases corrupt and decrease the data quality, see Figure 4.5. Thus, in order to fully utilize flight-testing data for system modeling, all these issues should be addressed by modeling each of the errors in the system and output model.

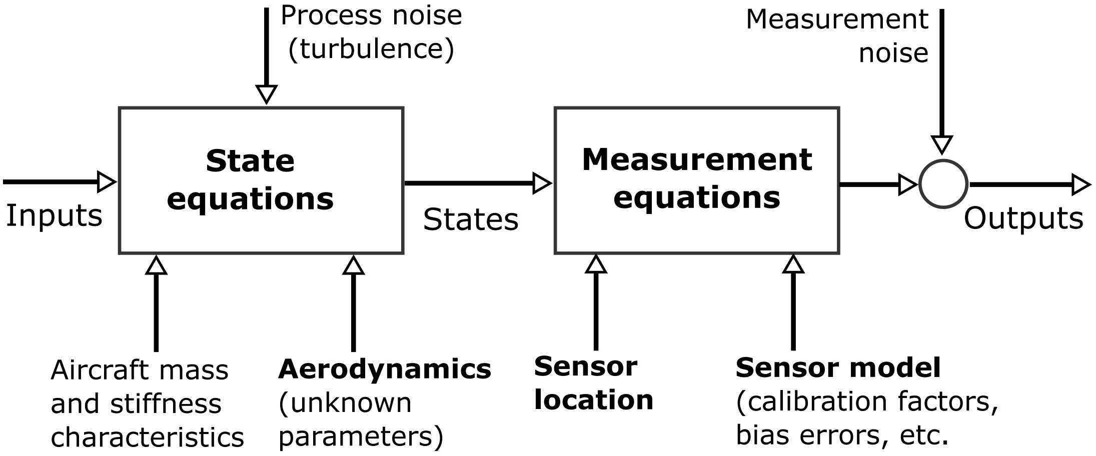


Figure 4.5: Error corrupting state and output model [18]

Therefore, the general equation of motion introduced in Chapter 3 is now modified by taking these errors into account,

$$\dot{\boldsymbol{x}} = f\left[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}\right] + \mathbf{F}(\lambda)w(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_\mathbf{o} \tag{4.3.1}$$

As in the deterministic model, $\boldsymbol{x}$ is the state variables in column vector $n_x \times 1$, $\boldsymbol{x}_\mathbf{o}$ is the initial state vector which has the same size as $\boldsymbol{x}$, $\boldsymbol{u}$ is the control input vector $n_u \times 1$ which is assumed as deterministic, and $\boldsymbol{\beta}$ is the system parameters vector $n_q \times 1$ which is the main focus of the aircraft parameter estimation. The unknown parameters ($\boldsymbol{\beta}$) will be briefly described in the next section. As shown in Eq. (4.3.1), the system is now excited by stochastic input $\boldsymbol{w}(t)$, a column vector $n_w \times 1$, also called as process noise which is non-measurable. In this dissertation, the process noise is assumed as a zero-mean white Gaussian noise with an identity power spectral density, while $\mathbf{F}(\lambda)$ denotes the additive process noise distribution matrix. Having postulated models tailored for system identification case, the unknown parameters that need to be estimated are formulated as,

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\beta}^T & \boldsymbol{\lambda}^T & \boldsymbol{x_\mathrm{o}}^T \end{bmatrix} \tag{4.3.2}$$

Correspondingly, the output model is now adapted to take the measurement noise and bias into account as given by,

$$\boldsymbol{z}_k = \boldsymbol{y}_k + \Delta \boldsymbol{z} + \mathbf{G}\boldsymbol{v}_k \tag{4.3.3}$$

where $\boldsymbol{z}_k$ is now called the measurement model as this model represents the discrete measurement of the model outputs $\boldsymbol{y}_k$, where biases and measurement noise are taken into account. Vector $\Delta \boldsymbol{z}$ is the measurement bias, $\mathbf{G}$ is the additive measurement noise distribution matrix, which is unknown. This matrix can also be represented by a covariance matrix as $\mathbf{R} = \mathbf{G}\mathbf{G}^T$ and is also being a part of the parameters to be estimated along with other unknown parameters as defined in Eq. (4.3.2). However, this covariance matrix is estimated differently and is further discussed in Section 4.5.2. Thus, the total unknown parameter vector both from state and output model is given by,

$$\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{\beta}^T & \boldsymbol{\lambda}^T & \boldsymbol{x_\mathrm{o}}^T & \Delta \boldsymbol{z} \end{bmatrix} \tag{4.3.4}$$

Unknown System Parameters, $\boldsymbol{\beta}$

As pointed out in Section 3.5, external forces and moments are commonly represented in non-dimensional form. This form is more convenient to work with as both forces and moments depend on dynamic pressure. In the Body-Fixed frame, these force and moment coefficients are commonly denoted as C_X, C_Y, C_Z and C_l, C_m, C_n. In general, these coefficients depend on flight condition and configuration, modeled as (see Chapter 3),

$$C_a = f(\alpha, \beta, p, q, r, ..., \delta_{spoiler}, \delta_{flaps}) \tag{4.3.5}$$

C_a represents force or moment coefficient, which is a function of quantities such as the angle of attack, sideslip angle, control input, and other influencing factors. The change of the explanatory variables will also change the overall force coefficient. The model can further be expanded in the Taylor expansion where the changes are expressed as coefficients in partial derivatives,

$$C_a = \frac{\partial C_a}{\partial \alpha}\Delta\alpha + \frac{\partial C_a}{\partial \beta}\Delta\beta + \frac{\partial C_a}{\partial \delta_{spoiler}}\Delta\delta_{spoiler} + \frac{\partial C_a}{\partial \delta_{flaps}}\Delta\delta_{flaps} + ... \tag{4.3.6}$$

Each of partial derivatives appearing in the model above is defined as the stability and control derivatives or in the context of system identification, they are called as system parameters. For example, applying Taylor expansion on lift coefficient in the Aerodynamic frame yields in,

$$C_L = C_{L_0} + C_{L_\alpha}\Delta\alpha + C_{L_\eta}\Delta\eta + ... \tag{4.3.7}$$

$C_{L_0}, C_{L_\alpha}, C_{L_\eta}, \cdots$ are the parameters that have physical meaning concerning aircraft dynamic behavior. In this context, system identification is dealing with the determination of the model structure, while system parameters involved in the model are estimated through parameter estimation procedure.

4.4 Maximum Likelihood and Kalman Filter

In the following sections, two foundation theories used in this dissertation will be discussed in details; they are Maximum Likelihood and Kalman Filter. In the system identification field, the Maximum Likelihood principle is commonly used for parameter estimation purpose while Kalman Filter is used for state estimation. Both of these theories will serve as a foundation for deriving system identification methods such as Output Error Method and Filter Error Method.

4.4.1 Maximum Likelihood

One of the most important developments in the 20th-century statistics is the making of Maximum Likelihood. It is well known as one of the most used estimation methods as it has properties of 'good estimator.' The method was mainly founded by R. A. Fisher of which he formalized and completed in three different papers from 1912 until 1925 [102, 103, 104]. He started by introducing the concept of ‘absolute criterion.’ This concept was introduced in his paper titled ‘An absolute criterion for fitting frequency curves’ (1912) in which the concept was derived from the ‘principle of inverse probability.’ In 1921, Fisher introduced the second concept of ‘optimum’ which was associated with the notion of ‘likelihood’ as a quantifying process for appraising hypothetical quantities by given data. R. A. Fisher (1922) completed the concept of Maximum Likelihood by introducing three "Criteria of Estimation," in which two of them linked to Maximum Likelihood, i.e., estimates given by Maximum Likelihood satisfy the criteria of 'sufficiency' and 'efficiency.' The third criteria, 'consistency' linked to the method of moments, taking it for granted that Maximum Likelihood satisfies it [105].

Generally, Maximum Likelihood Estimation (abbreviated as MLE) is an estimation method to find the most likely density function that would have generated the data [106]. It requires an assumption of the distribution type of the data sample and provides estimates of the respected chosen distribution. In the context of system modeling where data is fixed, MLE will give estimates of the model's parameters. Intuitively, MLE can be depicted in Figure 4.6. The general idea behind Maximum Likelihood can be illustrated as follows. In Figure

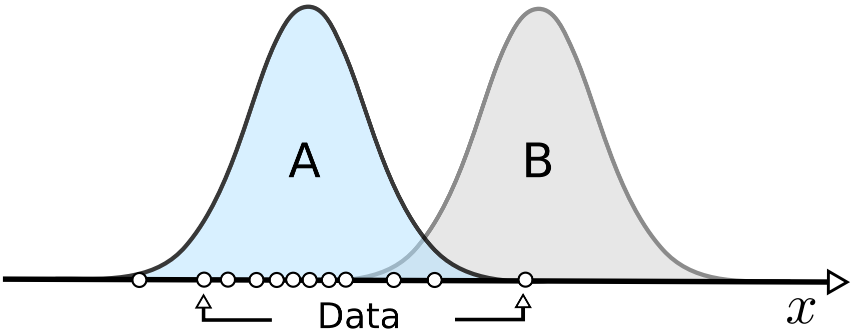


Figure 4.6: The Maximum Likelihood principle visually depicted

4.6, the data is plotted along x-axis in which some of the data clustered in some part on the x-axis. In this particular case, the data is assumed fixed, and we are asked to find a model and parameters which more likely to represent the data. This process can be done by fitting some distributions to the data. The distribution that fits the data well is the distribution/model we are looking for. As indicated in Figure 4.6, visually we can choose that distribution A is more likely to have generated the data as the data is clustered around the center of distribution A. What the example above is trying to explain is that by looking at the data, it is possible to find the distribution that is most likely to have generated the data. In principle, MLE is also working the same way. The distribution is determined by utilizing all available data through estimation of parameters that are most likely to produce the sample data. This procedure is an inverse probability problem where the main focus is more to the 'process' rather than producing the 'output.' Mathematically, this inverse problem is solved through a function known as the likelihood function. The likelihood function is algebraically the same as the probability density function (PDF) of observed data but different in terms of roles between parameters and observation. To begin with the formulation of likelihood, we will first define the probability density function of observed data given parameters, denoted as $f(\mathbf{Z}|\mathbf{\Theta})$. If each observation z_i is independent and identically distributed (*iid*) then according to the theory of probability, the joint distribution of the PDF is the multiplication of PDFs for each of individual observations,

$$\begin{aligned} f(\mathbf{Z}|\mathbf{\Theta}) &= f(z_1|\mathbf{\Theta}) \cdot f(z_2|\mathbf{\Theta}) \cdot \cdots \cdot f(z_k|\mathbf{\Theta}) \\ &= \prod_{k=1}^{N} f(z_k|\mathbf{\Theta}) \end{aligned} \tag{4.4.1}$$

In the likelihood function, Eq. (4.4.1) is modified slightly by reversing the role of parameters with observation in $f(\mathbf{Z}|\mathbf{\Theta})$ [107], i.e.,

$$L(\mathbf{Z}|\Theta) = f(\mathbf{Z}|\Theta) \tag{4.4.2}$$

The notation of the likelihood function in Eq. (4.4.2) is formulated based on a hypothesis defined in the likelihood that if a parent population has a known determinate mathematical form which only applies for the numerical values of some parameter vectors ($\mathbf{\Theta}$), it is required to deduce those unknown parameters using a set of observations (sample) taken from the parent population. Thus, $L(\mathbf{\Theta}|\mathbf{Z})$ represents the likelihood of the parameter $\mathbf{\Theta}$ given the observed data $z_1, z_2, \cdots z_N$. Though the two equations, Eq. (4.4.1) and Eq. (4.4.2), are the same, it is to be emphasized that the likelihood is written in this fashion to indicate that the focus is in the parameters ($\mathbf{\Theta}$) and the information about them is contained in the observed data ($\mathbf{Z}$).

Furthermore, it is understood that the likelihood function does not represent the probability distribution of the unknown parameters but of the measurement. Thus, in this classical estimation framework, the parameters are assumed to be fixed and constant, which are expected to be inferred from the data. As depicted in Figure 4.6, Maximum Likelihood aims at estimating the distribution that most likely has produced the data. The estimation of distribution is performed by applying the parameter values within the admissible range into the model and choose the one that most likely produced the data sample. In the context of the likelihood function, this procedure is the same as maximizing Eq. (4.4.2) with respect to the unknown parameters $\mathbf{\Theta}$. Due to the exponential nature of many density functions and the fact that the log function is monotonically increasing and easier to work with, the logarithm of the likelihood function is generally preferable. Thus, the Maximum Likelihood estimates are obtained as,

$$\hat{\mathbf{\Theta}}_{ML} = \text{arg}\{\min_{\mathbf{\Theta}} \text{ln } L(\mathbf{\Theta}|\mathbf{Z})\} \tag{4.4.3}$$

The values of $\mathbf{\Theta}$ that maximizes the $L(\mathbf{\Theta}|\mathbf{Z})$ or in its logarithmic form is denoted as $\hat{\mathbf{\Theta}}_{ML}$. The necessary condition for maximizing $\ln L(\mathbf{\Theta}|\mathbf{Z})$ given by,

$$\frac{\partial \ln L(\mathbf{\Theta}|\mathbf{Z})}{\partial \mathbf{\Theta}} = 0 \tag{4.4.4}$$

Eq. (4.4.4) leads to an optimization problem where the log of likelihood is minimized with respect to $\mathbf{\Theta}$. There are generally three options to deal with this which are:

1. Analytic. In this approach, the problem is solved analytically by setting the first derivative of the likelihood function to zero and solving the system's equations to find extrema points. Then identify if the extrema points are maximum or minimum by taking the second derivative of the model. Even though this method is straightforward, it only works if there is an existing analytical solution.

2. Grid Search. In the Grid Search method, the solution is found by a brute-force search or an exhaustive search method in a particular subspace in which $\mathbf{\Theta}$ lies. The algorithm will check for every possibility of $\mathbf{\Theta}$ and test whether it produces the maximum value of the likelihood function. It is a very straightforward method and easy to implement. However, this approach is not practical when the number of parameters to be estimated is more than one.

3. Numerical. It is probably the most common method used to solve the optimization problem of the likelihood function. From the computation point of view, this method is also very practical and does not have a strict limitation as with the Grid Search in terms of the number of parameters to be estimated. There are some algorithms that can be used, such as Newton-Raphson, Levenberg-Marquardt, or other optimization algorithms [108]. This dissertation employs this numerical approach to solve the optimization problem.

Linear expansion of Eq. (4.4.4) about the first approximation $\mathbf{\Theta}_0$ of $\mathbf{\Theta}$ yields,

$$\frac{\partial \ln L(\mathbf{\Theta}_1|\mathbf{Z})}{\partial \mathbf{\Theta}} = \frac{\partial \ln L(\mathbf{\Theta}_0|\mathbf{Z})}{\partial \mathbf{\Theta}} + \frac{\partial^2 \ln L(\mathbf{\Theta}_0|\mathbf{Z})}{\partial \mathbf{\Theta}^2} \Delta\mathbf{\Theta} \tag{4.4.5}$$

where $\mathbf{\Theta}_1 = \mathbf{\Theta}_0 + \mathbf{\Delta\Theta}$ is the improved approximation to $\mathbf{\Theta}$. Equating Eq. (4.4.5) to zero yields a linear system of equation:

$$\frac{\partial^2 \ln L(\mathbf{\Theta}_0|\mathbf{Z})}{\partial \mathbf{\Theta}^2} \mathbf{\Delta\Theta} = -\frac{\partial \ln L(\mathbf{\Theta}_0|\mathbf{Z})}{\partial \mathbf{\Theta}} \tag{4.4.6}$$

The expected value of the second gradient matrix, $\frac{\partial^2 \ln L(\mathbf{\Theta}_0|\mathbf{Z})}{\partial \mathbf{\Theta}^2}$, is called the Fisher information matrix which indicates the amount of information content of the unknown parameters $\mathbf{\Theta}$ in the observed data. Maximum Likelihood Estimator (MLE) has no optimum properties for finite samples, in the sense that when evaluating on finite samples, other estimators may have greater concentration around the true parameter value [109]. However, similar to other estimators whose size increases to infinity when being applied to a sample data, Maximum Likelihood possesses a number of attractive limiting properties [18, 25]:

1. Estimates of the Maximum Likelihood are asymptotically unbiased,

   $$\lim_{N\to\infty} \mathsf{E}(\hat{\mathbf{\Theta}}_{ML}) = \mathbf{\Theta} \tag{4.4.7}$$

   $\mathbf{\Theta}$ represents the true values of the parameters. This property states that as the number of data points increases to infinity, the mean of the distribution for the random vector $\hat{\mathbf{\Theta}}$ approaches the true parameter ($\mathbf{\Theta}$).

2. The Maximum Likelihood estimates $\hat{\mathbf{\Theta}}_{ML}$, obtained from different sets of data samples, are asymptotically normally distributed around the true value $\mathbf{\Theta}$,

   $$\sqrt{N} \cdot (\hat{\mathbf{\Theta}}_{ML} - \mathbf{\Theta}) \to rv \sim \mathcal{N}(0, \mathcal{F}^{-1}) \tag{4.4.8}$$

   where rv is a random variable, $\mathcal{F}$ is the average Fisher information matrix per sample, and $\mathcal{N}(0, \mathcal{F}^{-1})$ is the normal (Gaussian) distribution with zero mean and variance of $\mathcal{F}^{-1}$. The property of asymptotic normality implies that the estimates, obtained from different sets of data samples corresponding to different experiments, are clustered around the true value with a normal distribution. The Fisher information matrix is defined as,

$$\mathcal{F} \equiv \mathsf{E}\left\{\left[\frac{\partial \ln \mathcal{L}(\mathbf{Z}_N;\mathbf{\Theta})}{\partial \mathbf{\Theta}}\right]\left[\frac{\partial \ln \mathcal{L}(\mathbf{Z}_N;\mathbf{\Theta})}{\partial \mathbf{\Theta}}\right]^T\right\} = -\mathsf{E}\left[\frac{\partial^2 \ln \mathcal{L}(\mathbf{Z}_N;\mathbf{\Theta})}{\partial \mathbf{\Theta}\partial \mathbf{\Theta}^T}\right] \tag{4.4.9}$$

3. The Maximum Likelihood estimates $\hat{\mathbf{\Theta}}_{ML}$ are asymptotically efficient in the sense that they attain the Cramer-Rao lower bounds,

$$\text{Cov}(\hat{\mathbf{\Theta}}) \rightarrow \mathcal{F}^{-1} \text{ for } N \rightarrow \infty \tag{4.4.10}$$

   The Cramer-Rao bounds are the diagonal components of the inverse information matrix $\mathcal{F}^{-1}$, which indicates the parameter variances or the achievable accuracy for the estimated parameters. In practice, the number of data points N, and sampling rates are the main contributing factors for the achievable accuracy of the estimated parameters.

4. The Maximum Likelihood estimates $\hat{\mathbf{\Theta}}_{ML}$ are asymptotically consistent, that is, $\hat{\mathbf{\Theta}}_{ML}$ converges in probability to the true value $\mathbf{\Theta}$ as the number of data points increases, given as,

$$\hat{\mathbf{\Theta}} \rightarrow \mathbf{\Theta} \text{ as } N \rightarrow \infty \tag{4.4.11}$$

4.4.2 Kalman Filter

In terms of the availability of the measurement data, estimation can be classified into three different problems, namely filtering, smoothing, and prediction [110]. Filtering occurs when the time at which an estimate is desired coincides with the last measurement point. When the time of interest falls within the span of available measurement data, the problem is referred to as smoothing. Prediction, on the other hand, is defined when the time of interest occurs after the last available measurement. These three problems are depicted in Figure 4.7.

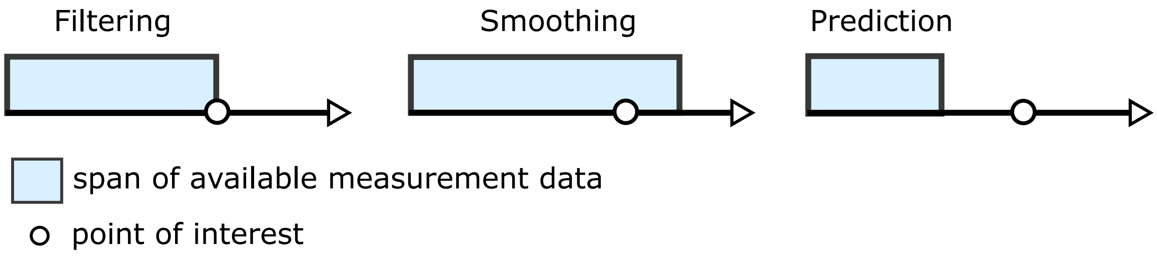


Figure 4.7: Measurement vs filtering, smoothing and prediction [110]

The concept of filtering became more known after R. E. Kalman introduced his paper titled 'A new approach to linear filtering and prediction problems' in 1960 [111]. This paper provides a foundation of a recursive solution to a discrete-data linear filtering problem. Since then, this filtering algorithm became the subject of extensive researches and applied in many different areas. Some of the Kalman Filter applications include objects tracking, economics, navigation, computer vision such as cluster tracking, image processing, and many more. These widespread implementations of the Kalman Filter algorithm were also supported by the facts that this algorithm is very practical, memory-less algorithm (minimal storage when implementing in a computer system), and suitable for online estimation problem. With the advancement in the digital computing, the Kalman Filter algorithm becomes one of the filtering algorithms found as an embedded system in today's electronic devices, e.g., in smartphone device, in aircraft Inertial Navigation System (INS), and other navigation-related devices. Kalman Filter can simply be defined as an optimal recursive data processing algorithm [112]. This definition introduces the three principles in Kalman Filter, i.e., optimal, recursive, and data processing algorithm. Each of these principles is elaborated in the following.

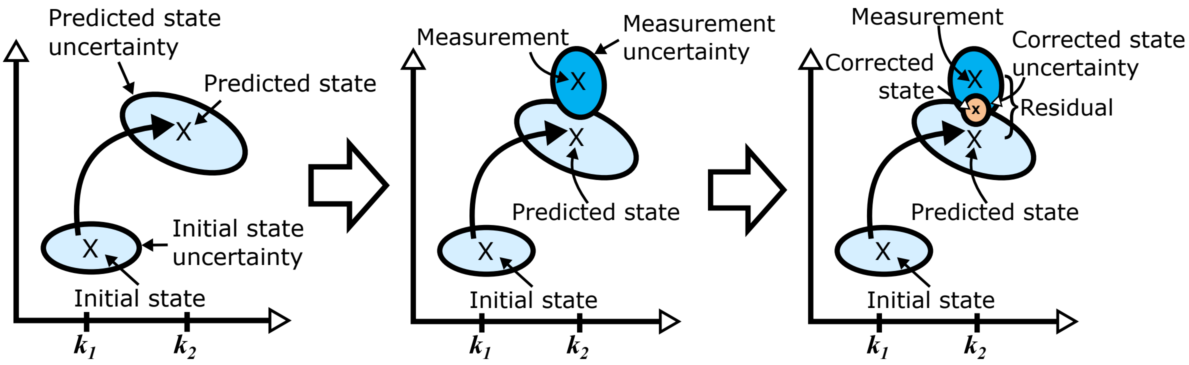
- Optimal

  In the sense of optimal aspect, Kalman Filter produces minimum variance estimates of the system states [110]. Furthermore, this algorithm also processes all available measurements by incorporating knowledge of the system, measurement of the dynamic system, the statistical description of measurement and process noise, dynamic models uncertainties, and information about initial conditions of the variables.
- Recursive

  The second aspect of the Kalman Filter is 'recursive.' It is the opposite of batch processing, which indicates that the Kalman Filter does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken. This property makes Kalman Filter an efficient algorithm and suitable for real-time implementation.
- Data processing algorithm

  The third aspect, 'data processing algorithm,' is related to a filtering process where a computer program processes the discrete-time measurement samples.

Fundamentally, Kalman Filter involves two main steps: (1) prediction, which is also called the propagation phase, and followed by (2) correction or also called the update phase. The dynamic model of the system is employed in the prediction phase between the two discrete points, k_1 to k_2. The prediction step is then followed by updating the phase using measurements data of time point k_2. The estimates obtained in the last step is optimal in the sense that it produces minimum variance error. The process of prediction and updating runs recursively. The mentioned process is depicted in Figure 4.8, which also summarizes the basic principle of the Kalman Filter algorithm. A typical application of Kalman Filter is implemented on a system disturbed by stochastic non-measurable inputs as illustrated in Figure 4.9. The block 'system' can be represented by any physical system such as an aircraft trajectory, a chemical process, or a mobile robot which is driven by a set of external controls both being deterministic and stochastic inputs. The output of the system is evaluated by measuring devices or sensors in such that the knowledge of the system's behavior is solely given by the inputs and the observed outputs. The observations convey the errors and uncertainties in the process, namely sensor noise, and system errors. Based on the available information, it is required to obtain an



(a) Propagation: The initial state, along with the model of a system, is used to propagate the state. Both the initial state and the propagated state have uncertainty, which is depicted as a circular area.

(b) Measurement: Besides a model, Kalman Filter also requires a measurement/observation in which every measurement has uncertainty in their value.

(c) Updating: The predicted state is corrected using the measurement. The Kalman gain is used to determine the contribution of the measurement to the corrected state.

Figure 4.8: Kalman Filter principle [113]

estimate of the system's states that optimizes the given criteria. Under such condition, the propagation of the system's states cannot be solved by using the initial-value problem through simple numerical integration as in the case of a deterministic system. This is the role played by the Kalman Filter by filtering noise from the corrupted system's states and providing the states with minimum errors. In the following section, the general formulation of the Kalman Filter is derived. First, the Kalman Filter algorithm for the linear model is discussed as this model representation is easier to understand and is tractable mathematically. Second, the Kalman Filter algorithm, which is introduced for the linear model, is then extended to cope with the nonlinear model.

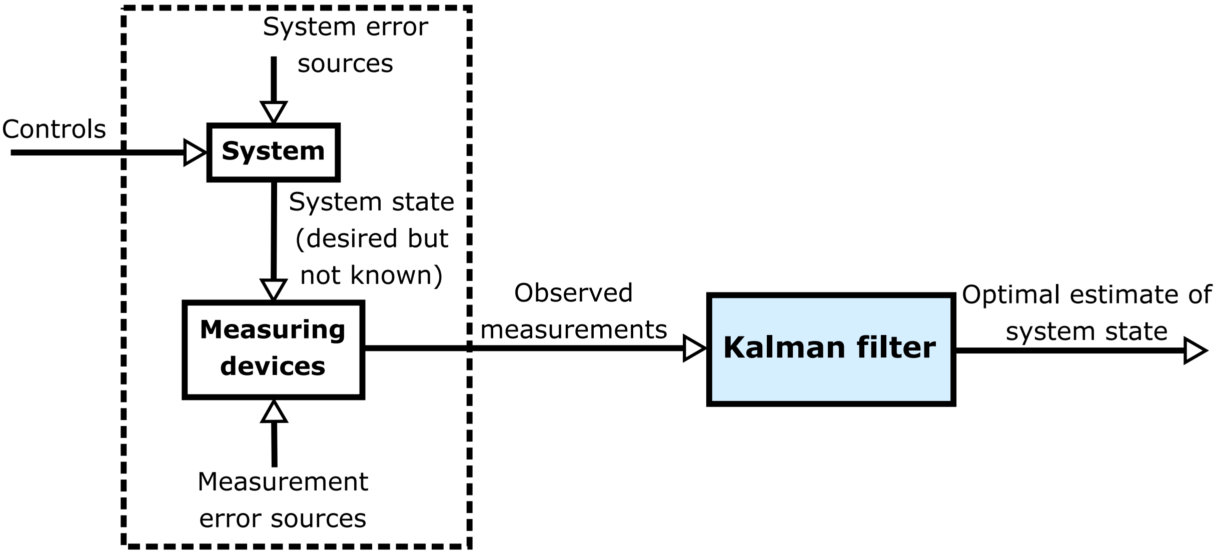


Figure 4.9: Kalman Filter, system, and measurement

4.4.2.1 Linear Model

In this dissertation, the Kalman Filter is formulated in a continuous-discrete representation. Other formulations such as purely continuous-time or purely discrete-time can also be used to model the Kalman Filter algorithm. Continuous in the sense that the system dynamics is represented by a dynamic-continuous model, while 'discrete' in the sense that the measurement is formulated in a discrete model which aligns with the fact that sensors measure the system output discretely. As depicted in Figure 4.8, two steps are involved in the Kalman Filter algorithm. The first step deals with the state propagation between time step k to $k+1$, which is carried out through the system model. In other literature, this step is also known as prediction step or time update. The second step performs a correction to the updated step through the measurement data $(\boldsymbol{z})$ at a specific time point $(k+1)$. Therefore, this step is also known as the measurement update. There are various notations used in deriving the Kalman Filter algorithm such as using (˜) for prediction step and (ˆ) symbol for updating step or negative $(-)$/positive $(+)$ signs for prediction and updating steps respectively. This dissertation employs the following convention. Variables related to the predicted step will be denoted by superscript ˜ (tilde) symbol, while superscript ˆ (hat) is used to denote variables in the corrected step. The other assumptions and variable definitions used in this section are summarized as follows,

$$\begin{aligned} \boldsymbol{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) & \quad \text{measurement noise} \\ E\left[\boldsymbol{v}_k \boldsymbol{v}_j^T\right] = \mathbf{R}_k \delta_{k-j} & \quad \text{measurement noise covariance} \\ \tilde{\boldsymbol{e}}_k = \tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k & \quad \text{predicted state error} \\ \tilde{\mathbf{P}}_k = E\left[\tilde{\boldsymbol{e}}_k \tilde{\boldsymbol{e}}_k^T\right] & \quad \text{predicted state error covariance} \\ \hat{\boldsymbol{e}}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k & \quad \text{corrected state error} \\ \hat{\mathbf{P}}_k = E\left[\hat{\boldsymbol{e}}_k \hat{\boldsymbol{e}}_k^T\right] & \quad \text{corrected state error covariance} \end{aligned}$$

Process noise definition, $\boldsymbol{w}$, is discussed separately in this section since the transition from continuous formulation $\boldsymbol{w}(t)$ to the discrete formulation $\boldsymbol{w}_k$, should be addressed by this variable. System disturbed by stochastic process formulated in a linear dynamic model takes form as,

$$\dot{\boldsymbol{x}}(t) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{F}\boldsymbol{w}(t) \tag{4.4.12}$$

where $\mathbf{A}$ denotes the state matrix of the linearized system dynamics, $\mathbf{B}$ denotes input matrix, and $\boldsymbol{u}$ denotes control input. Based on the discrete-time theory of linear system, the solution of Eq. (4.4.12) is given by,

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{\Psi}\mathbf{B}\boldsymbol{u}_k + \boldsymbol{\Psi}\frac{\mathbf{F}}{\sqrt{\Delta t}}\boldsymbol{w}_k \tag{4.4.13}$$

While the output model is represented by,

$$\boldsymbol{y}(t) = \mathbf{C}\boldsymbol{x}(t) + \mathbf{D}\boldsymbol{u}(t) \tag{4.4.14}$$

In the system model of Eq. (4.4.12), $\boldsymbol{w}(t)$ represents the process noise and is assumed to be zero-mean white Gaussian-noise with an identity power spectral density matrix, while $\mathbf{F}$ denotes the process noise distribution matrix. The corresponding discrete-time white-noise process $\boldsymbol{w}_k$ has to have the covariance $1/\sqrt{\Delta t}$ times the identity matrix [18]. This is represented in Eq. (4.4.13) by the factor $1/\sqrt{\Delta t}$ corresponding with $\mathbf{F}$. Such variable $(1/\sqrt{\Delta t})$ is required to account for the transition between the continuous-time and the discrete-time model.

Furthermore, it is also critical to get the limit between the discrete-time system and continuous-time; otherwise, in the limit case, the system will not respond to noise, which is physically unrealistic. Maine and Iliff [114] provide a detailed treatment of this subtle issue. It is specifically emphasized here that $\boldsymbol{w}(t)$ and $\boldsymbol{w}_k$ are not the same, although the same symbol is used, $\boldsymbol{w}$, for the two processes. Variables $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ represent state transition matrix and its integral, given as,

$$\boldsymbol{\Phi} = e^{\mathbf{A}\Delta t} \approx \mathbf{I} + \mathbf{A}\Delta t + \mathbf{A}^2\frac{\Delta t^2}{2!} + \cdots \tag{4.4.15}$$

$$\boldsymbol{\Psi} = \int_0^{\Delta t} e^{\mathbf{A}\tau} d\tau \approx I\Delta t + \mathbf{A}\frac{\Delta t^2}{2!} + \mathbf{A}^2\frac{\Delta t^3}{3!} + \cdots \tag{4.4.16}$$

In terms of stochastic input, the linear model provides an attractive property which transfers the input's distribution type to its output. For example, feeding the linear model with Gaussian distribution input will also produce output with a Gaussian distribution. This property makes the linear model fits perfectly with the Kalman Filter algorithm since Kalman Filter expects that the input distribution and its propagated output should have the same distribution type. This property is visually illustrated in Figure 4.10.

The derivation of the Kalman Filter algorithm is started by formulating the predicted state $\tilde{\boldsymbol{x}}$ and its corresponding uncertainty as given by the covariance matrix $\tilde{\mathbf{P}}$. The predicted state or the propagation of the state is simply computed through the system model as in,

$$\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{\Phi}\hat{\boldsymbol{x}}_k + \boldsymbol{\Psi}\mathbf{B}\boldsymbol{u}_k \tag{4.4.17}$$

where $\hat{\boldsymbol{x}}$ denotes the corrected state. Predicted state $\tilde{\boldsymbol{x}}$ is not precise as this variable is corrupted by the process noise $\boldsymbol{w}(t)$ in which the error from this state can be quantified by subtracting the predicted state $\tilde{\boldsymbol{x}}$ with the true state $\boldsymbol{x}$ as given by,

$$\tilde{\boldsymbol{e}}_{k+1} = \tilde{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1} \tag{4.4.18}$$

The uncertainty in the predicted state $\tilde{\boldsymbol{x}}$ can further be modeled in terms of a covariance

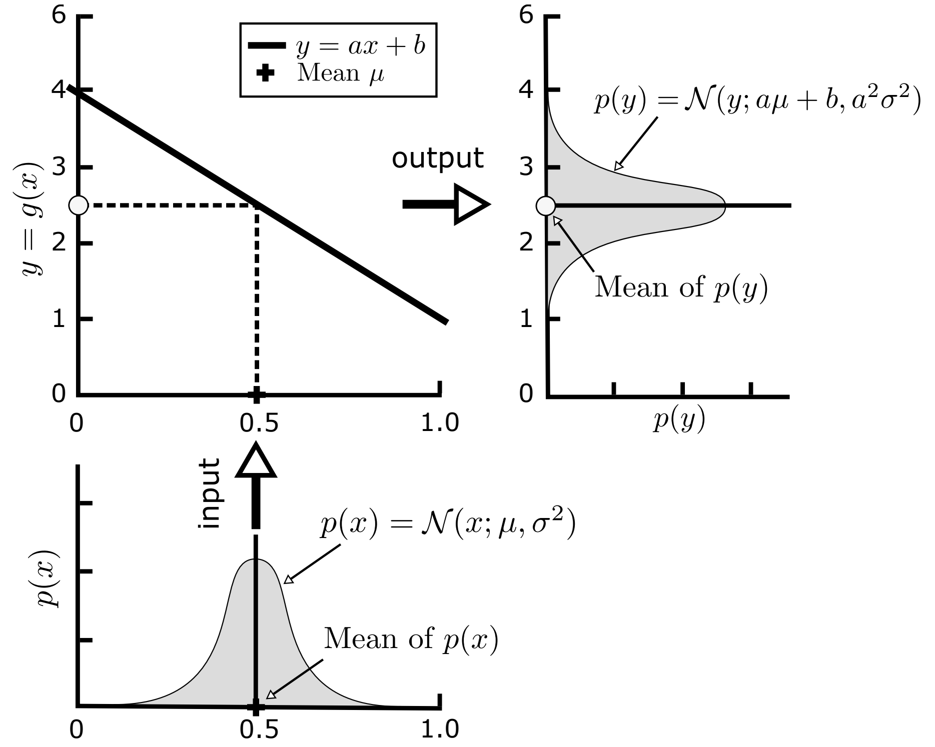


Figure 4.10: Gaussian input fed to linear model produces Gaussian output

matrix,

$$\begin{aligned}\tilde{\mathbf{P}}_{k+1} &= E\{\tilde{\boldsymbol{e}}_{k+1}\tilde{\boldsymbol{e}}_{k+1}^T\} \\ &= E\{[\tilde{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}]\,[\tilde{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}]^T\}\end{aligned} \tag{4.4.19}$$

where E denotes the expected value. Eq. (4.4.19) is further simplified by plugging Eq. (4.4.12) into $\boldsymbol{x}_{k+1}$ and Eq. (4.4.17) into $\tilde{\boldsymbol{x}}_{k+1}$ as given by,

$$\tilde{\mathbf{P}}_{k+1} = E\{[\boldsymbol{\Phi}\hat{\boldsymbol{e}}_k - \boldsymbol{\Psi}\mathbf{F}_{\mathbf{d}}\boldsymbol{w}_k]\,[\boldsymbol{\Phi}\hat{\boldsymbol{e}}_k - \boldsymbol{\Psi}\mathbf{F}_{\mathbf{d}}\boldsymbol{w}_k]^T\} \tag{4.4.20}$$

where $\hat{\boldsymbol{e}} = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$ denotes the error between the updated state $\hat{\boldsymbol{x}}_k$ and true state $\boldsymbol{x}_k$, $\mathbf{F}_{\mathbf{d}} = (\Delta t)^{-1/2}\mathbf{F}$. By using the assumptions that noise $\boldsymbol{w}_k$ and errors $\hat{\boldsymbol{e}}_k$ are uncorrelated, Eq. (4.4.20) is further simplified as,

$$\tilde{\mathbf{P}}_{k+1} = \boldsymbol{\Phi}E\{\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\}\boldsymbol{\Phi}^T + \boldsymbol{\Psi}\mathbf{F}_{\mathbf{d}}E\{\boldsymbol{w}_k\boldsymbol{w}_k^T\}\mathbf{F}_d^T\boldsymbol{\Psi}^T \tag{4.4.21}$$

$E[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k] = \hat{\mathbf{P}}_k$ is the corrected state covariance matrix which is modeled in the same way as in the predicted state covariance matrix. $E[\boldsymbol{w}_k\boldsymbol{w}_k^T] = 1$ as defined in the beginning of this section. For small sampling time Δt, Eq. (4.4.16) can be approximated as $\boldsymbol{\Psi} \approx \mathbf{I}\Delta t$. Taking all these terms into account, the predicted state covariance matrix $\tilde{\mathbf{P}}_{k+1}$ is now modeled in the final form as,

$$\tilde{\mathbf{P}}_{k+1} = \boldsymbol{\Phi}\hat{\mathbf{P}}_k\boldsymbol{\Phi}^T + \Delta t\mathbf{F}\mathbf{F}^T \tag{4.4.22}$$

with initial condition given by $\tilde{\mathbf{P}}_0 = E\{\tilde{\boldsymbol{x}}_0\tilde{\boldsymbol{x}}_0^T\}$. The next step is to develop an optimal expression for the corrected state $\hat{\boldsymbol{x}}_k$ along with its uncertainty $\hat{\mathbf{P}}_k$. In this formulation, at which the measurement becomes available is denoted by k instead of $k+1$ for notational simplicity without loss of generality as the computation is done recursively. In the formulation developed by Kalman, the corrected state $\hat{\boldsymbol{x}}$ is computed through a linear combination of predicted state $\tilde{\boldsymbol{x}}_k$ and the residual $\boldsymbol{v}_k$ between measurement $\boldsymbol{z}_k$ and model output $\boldsymbol{y}_k$, given by,

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}_k\boldsymbol{v}_k \tag{4.4.23}$$

where $\mathbf{K}$ is the so-called Kalman gain matrix that determines the quality of the estimates $\hat{\boldsymbol{x}}$. Residual $\boldsymbol{v}_k$ is computed by subtracting the measurement data with the predicted output. The predicted output is computed by utilizing Eq. (4.4.14) in discrete formulation with the predicted state used as a state variable in the model leads to,

$$\boldsymbol{y}_k = \mathbf{C}\tilde{\boldsymbol{x}}_k - \mathbf{D}\boldsymbol{u}_k \tag{4.4.24}$$

Substitution of Eq.(4.4.24) into $\boldsymbol{v}_k = \boldsymbol{z}_k - \boldsymbol{y}_k$ and plugging this equation back into Eq. (4.4.23) leads to the final form of the corrected state model as in,

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}_k\left[\boldsymbol{z}_k - \mathbf{C}\tilde{\boldsymbol{x}}_k\right] \tag{4.4.25}$$

Note that, in Eq. (4.4.25) the control observation matrix $\mathbf{D}$ is dropped from Eq. (4.4.14) for convenience without loss of generality. The next step is to formulate the optimal Kalman gain $\mathbf{K}$ in such that the corrected state $\hat{\boldsymbol{x}}$ is efficient. The same way as in the predicted state, $\tilde{\boldsymbol{x}}$, the uncertainty of the corrected state is quantified through its error covariance matrix $\hat{\mathbf{P}}_k = E\{\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\}$. The corrected state error $\hat{\boldsymbol{e}}_k$ is formulated by plugging the discrete form of Eq. (4.4.14) and Eq. (4.4.25) into corrected state error $\hat{\boldsymbol{e}}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$, yields in,

$$\hat{\boldsymbol{e}}_k = \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\boldsymbol{e}}_k + \mathbf{K}\boldsymbol{v}_k \tag{4.4.26}$$

Substitution of Eq. (4.4.26) into $\hat{\mathbf{P}}_k$ and with the assumptions that measurement noise $\boldsymbol{v}_k$ and state error $\tilde{\boldsymbol{e}}_k$ are uncorrelated, $E\{\boldsymbol{v}_k\boldsymbol{v}_k^T\} = \mathbf{R}_k$, the corrected state error covariance matrix is formulated as,

$$\hat{\mathbf{P}}_k = \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\mathbf{P}}_k\left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{4.4.27}$$

The estimate $\hat{\boldsymbol{x}}_k$ is efficient if the variance of the estimation error, $\hat{\boldsymbol{e}}_k$, which is the same as the trace of the covariance matrix $\hat{\mathbf{P}}_k$, is minimized. This leads to the cost function given by,

$$J_k = \text{trace}\left[\hat{\mathbf{P}}_k\right] \tag{4.4.28}$$

Taking the partial derivative of Eq. (4.4.28) with respect to $\mathbf{K}$ and applying the partial differentiation operator as in $\frac{\partial}{\partial \mathbf{A}}\{\text{trace}(\mathbf{A}\mathbf{B}\mathbf{A}^T)\} = 2\mathbf{A}\mathbf{B}$, Eq. (4.4.28) is further expanded as,

$$\frac{\partial J_k}{\partial \mathbf{K}_k} = -2\left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\mathbf{P}}_k\mathbf{C}^T + 2\mathbf{K}_k\mathbf{R}_k \tag{4.4.29}$$

The Kalman gain matrix is obtained by equating Eq. (4.4.29) to zero and solving for $\mathbf{K}$ yields in,

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k\mathbf{C}^T\left[\mathbf{C}\tilde{\mathbf{P}}_k\mathbf{C}^T + \mathbf{R}_k\right]^{-1} \tag{4.4.30}$$

Eq. (4.4.30) is the last variable required for the implementation of the Kalman Filter on a linear dynamic system. Intuitively, the Kalman gain can be seen as a relative weight given to the measurement and current state estimate whose value depends on the ratio of uncertainty in the estimate ($\tilde{\mathbf{P}}$) and measurement covariance $\mathbf{R}$. To better observe the Kalman gain formula as expressed in Eq. (4.4.30), $\mathbf{C}$ can be assumed as an identity matrix. In this case, both $\tilde{\mathbf{P}}$ and $\mathbf{R}$ are matrices with size of $n_y \times n_y$. Furthermore, if $\mathbf{R}$ takes form as a diagonal matrix, then the expression for the Kalman gain is just a multiplication of each column of the state error covariance matrix with the corresponding inverse of mean square measurement noise. Thus, each element in $\mathbf{K}$ is essentially the ratio between the statistical measure of the uncertainty in the estimate and the uncertainty in the measurement. Consequently, a high gain means that the filter places more weight on the recent measurements, which causes the estimates to behave more responsively. On the other hand, with low gain, the filter follows the model predictions more closely, which also decreases the responsiveness of the estimates.

An alternative form of the corrected state error covariance matrix, $\hat{\mathbf{P}}_k$, can be derived by substituting the Kalman gain matrix as in Eq. (4.4.30) into Eq. (4.4.27), leads to a short form of the corrected state error covariance [115],

$$\begin{aligned}
\hat{\mathbf{P}}_k &= \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\mathbf{P}}_k\left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\
&= \left[\tilde{\mathbf{P}}_k - \mathbf{K}_k\mathbf{C}\tilde{\mathbf{P}}_k\right]\left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\
&= \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\mathbf{P}}_k - \tilde{\mathbf{P}}_k\mathbf{C}^T\mathbf{K}^T + \underbrace{\mathbf{K}_k\left[\mathbf{C}\tilde{\mathbf{P}}_k\mathbf{C}^T + \mathbf{R}_k\right]}_{=\tilde{\mathbf{P}}_k\mathbf{C}^T}\mathbf{K}_k^T \\
&= \left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]\tilde{\mathbf{P}}_k
\end{aligned} \tag{4.4.31}$$

Eq. (4.4.31) is often used in practice as it is less complex and computationally faster than Eq. (4.4.27). However, due to round-off errors, Eq. (4.4.27) might produce non-positive diagonal elements of the covariance matrix, which is physically unrealistic. On the other hand, Eq. 4.4.27, also called 'Joseph' form, always produce a positive definite matrix. Thus, Eq. (4.4.27) is the preferred option in order to avoid numerical instability. Table 4.2 summarizes the computational steps of the Kalman Filter algorithm for a linear dynamic system.

Steps	Description and formula
①	**Set initial values:** $\tilde{\boldsymbol{x}}_0, \tilde{\mathbf{P}}_0, \mathbf{Q}$, and $\mathbf{R}_0$
②	**Kalman gain computation:** $\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{C}_k^T \left[\mathbf{C}_k \tilde{\mathbf{P}}_k^T \mathbf{C}_k^T + \mathbf{R}_k \right]^{-1}$
③	**Correction step (measurement update)** $\tilde{\boldsymbol{y}}_k = \mathbf{C}\tilde{\boldsymbol{x}}_k + \mathbf{D}\boldsymbol{u}_k$ $\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}_k \left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k \right]$ $\hat{\mathbf{P}}_k = \left[\mathbf{I} - \mathbf{K}_k \mathbf{C} \right] \tilde{\mathbf{P}}_k \left[\mathbf{I} - \mathbf{K}_k \mathbf{C} \right]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$ or using the short form, $\hat{\mathbf{P}}_k = \left[\mathbf{I} - \mathbf{K}_k \mathbf{C} \right] \tilde{\mathbf{P}}_k$
④	**Prediction step (time update)** $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{\Phi} \hat{\boldsymbol{x}}_k + \boldsymbol{\Psi} \mathbf{B} \boldsymbol{u}_k$ $\tilde{\mathbf{P}}_{k+1} = \boldsymbol{\Phi} \hat{\mathbf{P}}_k \boldsymbol{\Phi}^T + \Delta t \mathbf{F}\mathbf{F}^T$ where, $\Delta t \mathbf{F}\mathbf{F}^T = \mathbf{Q}$ $\boldsymbol{\Phi} = e^{\mathbf{A}\Delta t} \approx \mathbf{I} + \mathbf{A}\Delta t + \mathbf{A}^2 \frac{\Delta t^2}{2!} + \cdots$ $\boldsymbol{\Psi} = \int_0^{\Delta t} e^{\mathbf{A}\tau} d\tau \approx \mathbf{I}\Delta t + \mathbf{A}\frac{\Delta t^2}{2!} + \mathbf{A}^2 \frac{\Delta t^3}{3!} + \cdots$

Table 4.2: Kalman Filter computational steps on a linear system

4.4.2.2 Nonlinear Model

Propagation of a Gaussian input in a linear model will also produce a Gaussian output. This property makes the linear model fits perfectly for the Kalman Filter algorithm. However, for a nonlinear model, this fact does not apply. Feeding Gaussian into a nonlinear model will produce a non-Gaussian output, see Figure 4.11. Gaussian's property is 'destroyed' by the nonlinearity model on which Kalman Filter can no longer be applied. Extended Kalman Filter (EKF) solves this issue by linearizing the current state and covariance and use the linear Kalman Filter framework.

Since EKF is obtained using a linear approximation of a nonlinear system, it offers no guarantees of optimality in a mean squared error sense. However, for many systems, EKF has proven to be a useful method of obtaining good estimates of the system states and applied widely in many fields, particularly in applications related to the flight vehicle. The EKF procedures are slightly different as in the case of the linear Kalman Filter. Steps in the linear Kalman Filter framework can be directly implemented on the nonlinear model once the model linearized at a

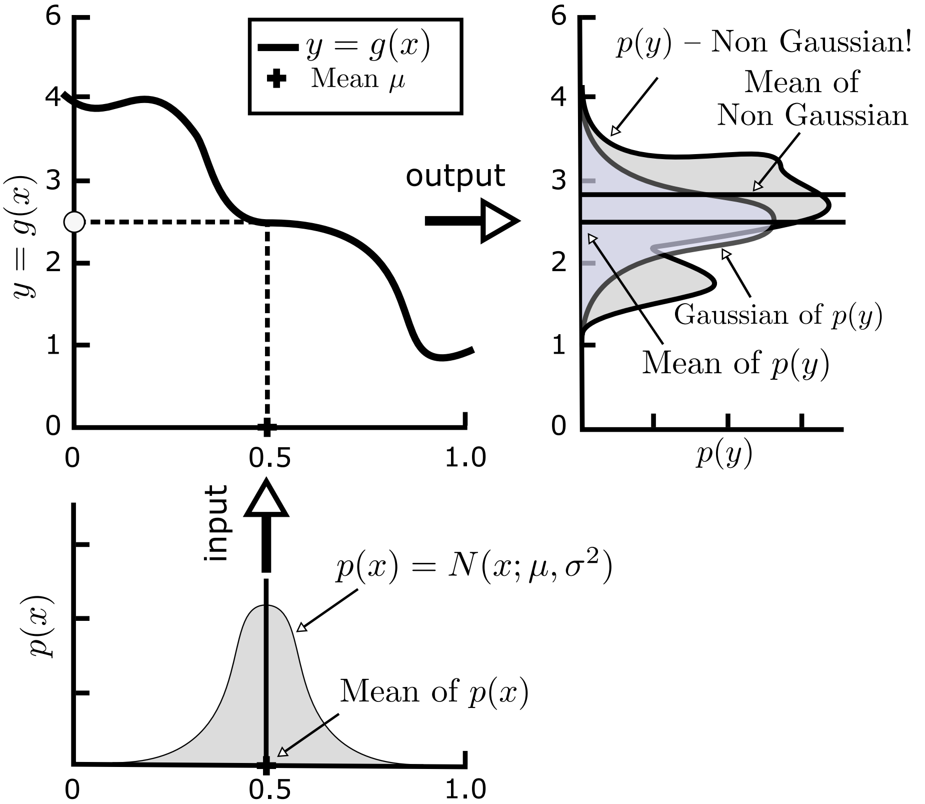


Figure 4.11: Gaussian fed into nonlinear model produces non Gaussian

specific time point. The nonlinear model disturbed by a stochastic model and its corresponding output model are rewritten here for clarity,

$$\dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] + \mathbf{F}\boldsymbol{w}(t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{4.4.32}$$

$$\boldsymbol{y}(t) = g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] \tag{4.4.33}$$

where $\boldsymbol{\beta}$ represents the system parameters both in state model f and output model g. The process noise $\boldsymbol{w}$ is assumed to be additive so that the assumption used in the Kalman Filter linear model is still valid to be used in the nonlinear model. The linearization of the nonlinear model of Eqs. (4.4.32) and (4.4.33) at any discrete time point k can be done through analytically or numerically. The analytical approach provides the most accurate result but not very convenient, especially when dealing with a complex system. Numerical approaches are more convenient and commonly found in practices, especially for a complex system, for example, central difference formula might be used for this purpose. Linearization of the nonlinear model, Eq. (4.4.32), leads to the linearized system matrices as given by,

$$\begin{aligned} \mathbf{A} &= \frac{\partial f[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta}]}{\partial \boldsymbol{x}} \qquad & \mathbf{B} &= \frac{\partial f[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta}]}{\partial \boldsymbol{u}} \\ \mathbf{C} &= \frac{\partial g[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta}]}{\partial \boldsymbol{x}} \qquad & \mathbf{D} &= \frac{\partial g[\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\beta}]}{\partial \boldsymbol{u}} \end{aligned} \tag{4.4.34}$$

Having linearized the nonlinear model, the steps defined in Table 4.2 are implemented in the same way as in the nonlinear model case. The state transition matrix $\mathbf{\Phi}$ and its integral $\mathbf{\Psi}$ are computed through Eqs. (4.4.15) and (4.4.16) respectively. Table 4.3 summarizes the procedures of the Extended Kalman Filter.

Steps	Description and formula
①	**Set initial values:** $\tilde{\boldsymbol{x}}_0, \tilde{\mathbf{P}}_0, \mathbf{Q}$, and $\mathbf{R}_0$
②	**Kalman gain computation:** $\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{C}_k^T \left[\mathbf{C}_k \tilde{\mathbf{P}}_k^T \mathbf{C}_k^T + \mathbf{R}_k\right]^{-1}$
③	**Correction step (measurement update)** $\tilde{\boldsymbol{y}}_k = g\left[\tilde{\boldsymbol{x}}(k), \boldsymbol{u}(k), \boldsymbol{\beta}\right]$ $\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}_k \left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k\right]$ $\hat{\mathbf{P}}_k = \left[\mathbf{I} - \mathbf{K}_k \mathbf{C}\right] \tilde{\mathbf{P}}_k \left[\mathbf{I} - \mathbf{K}_k \mathbf{C}\right]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$ or using the short form, $\hat{\mathbf{P}}_k = \left[\mathbf{I} - \mathbf{K}_k \mathbf{C}\right] \tilde{\mathbf{P}}_k$ where, $\mathbf{C}_k = \left.\dfrac{\partial g\left[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}\right]}{\partial \boldsymbol{x}}\right\|_{\boldsymbol{x}=\tilde{\boldsymbol{x}}}$
④	**Prediction step (time update)** $\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \int_{t_k}^{t_{k+1}} f\left[\boldsymbol{x}(t), \boldsymbol{u}(t_k), \boldsymbol{\beta}\right] \mathrm{d}t$ $\tilde{\mathbf{P}}_{k+1} = \mathbf{\Phi} \hat{\mathbf{P}}_k \mathbf{\Phi}^T + \mathit{\Delta} t \mathbf{F}\mathbf{F}^T$ where, $\Delta t \mathbf{F}\mathbf{F}^T = \mathbf{Q}$ $\mathbf{\Phi}_{k+1} = e^{\mathbf{A}\Delta t} \approx I + \mathbf{A}\Delta t + \mathbf{A}^2 \frac{\Delta t^2}{2!} + \cdots$ $\mathbf{A}_k = \left.\dfrac{\partial g\left[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}\right]}{\partial \boldsymbol{x}}\right\|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}$

Table 4.3: EKF computational steps on a nonlinear system [18]

4.5 Methods Used in Aircraft System Identification

The two estimation theories described in Sections 4.4.1 and 4.4.2 provide a foundation for deriving methods in aircraft system identification field. Four methods commonly used in flight vehicle system identification are briefly introduced in this section, namely Least Square (Equation Error Method), Output Error Method, Filter Error Method, and Extended Kalman Filter. The latter method was initially developed for state estimation; however, with a minor change on the algorithm, it can be used for parameter estimation purpose. The modification of the Kalman Filter algorithm for dual estimation is covered in section 4.5.4.

4.5.1 Equation Error Method

Equation Error Method is based on the principle of least squares, which is relatively fast and applicable to linear as well as to the nonlinear system [116]. This method is the simplest approach among other methods in aircraft system identification. Its simplicity based on the assumptions used pertaining to the model parameters $\mathbf{\Theta}$ and measurement noise $\boldsymbol{v}$. In this approach, both $\mathbf{\Theta}$ and $\boldsymbol{v}$ do not hold any probability assumptions. Variable $\mathbf{\Theta}$ is assumed as a vector of unknown constant parameters, while $\boldsymbol{v}$ is a random vector of measurement noise. This assumption leads to a general statement of the least-square estimator as "*given measurement data $\boldsymbol{z}$, the "best" estimate of the unknown parameters $\mathbf{\Theta}$ come from minimizing the weighted sum of squared differences between the measured outputs $\boldsymbol{z}$ and the model outputs $\boldsymbol{y}$*", mathematically expressed by [25],

$$J(\mathbf{\Theta}) = \frac{1}{2}(\boldsymbol{z} - \mathbf{H\Theta})^T \mathbf{R}^{-1}(\boldsymbol{z} - \mathbf{H\Theta}) \tag{4.5.1}$$

where J denotes the cost function to be minimized with respect to $\mathbf{\Theta}$, $\mathbf{R}^{-1}$ is a weighting matrix which is chosen through an engineering judgment, and $\mathbf{H\Theta}$ represents the linear output model $\boldsymbol{y}$. In general, the optimization of cost function formulated in Eq. (4.5.1) leads to the Weighted Least Squares (WLS) estimator in which the weighting is quantified in $\mathbf{R}^{-1}$. In the case where $\mathbf{R}$ is assumed as the identity matrix $\mathbf{I}$, the well-known Ordinary Least Squares (OLS) estimator is obtained, leads to the reduced cost function given by,

$$J(\theta) = \frac{1}{2}(\boldsymbol{z} - \mathbf{H\Theta})^T(\boldsymbol{z} - \mathbf{H\Theta}) \tag{4.5.2}$$

Both methods mentioned previously assume that the stochastic input/measurement noise only affects the dependent variables while the independent variables or regressor variables are assumed to be noise free. These approaches lead to biased estimates if the investigated data contain noise, both in independent and dependent variables. The other derivative of the least squares approaches called Total Least Squares (TLS). TLS approach solves the OLS issue by taking the noise, both from the independent and dependent variables, into account. Each of these techniques is discussed in details in the following section.

4.5.1.1 Ordinary Least Squares Method

The formulation of the Ordinary Least Squares method is focused on the linear model for mathematical tractability. The least-square model can also be applied to the nonlinear model as the basic principle is only based on the minimization of the errors between the measurement and the model response. Mathematically, the linear model, in this case, takes form as,

$$\boldsymbol{y} = \mathbf{H\Theta} \tag{4.5.3}$$

Correspondingly the measurement affected by noise $\boldsymbol{v}$ takes form,

$$\boldsymbol{z} = \mathbf{H}\boldsymbol{\Theta} + \boldsymbol{v} \tag{4.5.4}$$

where

$$\begin{aligned}
\mathbf{H} &= [1 \quad h_1 \quad \cdots h_n] = \text{matrix with dimension of } N \times n_p \\
&= \begin{bmatrix} 1 & h_{1_1} & h_{2_1} & \cdots & h_{n_1} \\ 1 & h_{1_2} & h_{2_2} & \cdots & h_{n_2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & h_{1_n} & h_{2_n} & \cdots & h_{n_n} \end{bmatrix} \\
&= \text{denotes independent variable or input variable} \\
\boldsymbol{\Theta} &= \begin{bmatrix} \Theta_0, & \Theta_1, \cdots & \Theta_{n_p} \end{bmatrix}^T = \text{vector with dimension of } n_p \times 1 \\
&= \text{denotes vector of unknown parameters (bias + system parameters)} \\
\boldsymbol{v} &= [v_1 \quad v_2 \quad \cdots v_n]^T = \text{vector with dimension of } N \times 1 \\
&= \text{denotes vector of measurement noise} \\
\boldsymbol{z} &= [z_1 \quad z_2 \quad \cdots z_n] = \text{vector with dimension of } N \times 1 \\
&= \text{denotes vector of measurement data}
\end{aligned}$$

As already discussed briefly at the beginning of this section, there is no probability density assumption on measurement noise $\boldsymbol{v}$ and the unknown parameters $\boldsymbol{\Theta}$. Therefore, the 'best' estimates of parameters $\boldsymbol{\Theta}$ are obtained through the minimization of the error between the measurement and the model output as given by,

$$J = \frac{1}{2}(\boldsymbol{z} - \mathbf{H}\boldsymbol{\Theta})^T(\boldsymbol{z} - \mathbf{H}\boldsymbol{\Theta}) \tag{4.5.5}$$

Taking the first derivative of Eq. (4.5.5) with respect to $\boldsymbol{\Theta}$ and equating to zero leads to,

$$\frac{\partial J}{\partial \boldsymbol{\Theta}} = -\mathbf{H}^T\boldsymbol{z} + \mathbf{H}^T\mathbf{H}\hat{\boldsymbol{\Theta}} = 0 \tag{4.5.6}$$

which is further simplified into the so-called ordinary least squares estimator [25, 117],

$$\hat{\boldsymbol{\Theta}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\boldsymbol{z} \tag{4.5.7}$$

The uncertainty in the estimates can be computed through the matrix covariance of the estimation error, given by [25],

$$\begin{aligned}
\text{Cov}(\hat{\mathbf{\Theta}}) &\equiv E\left[(\hat{\mathbf{\Theta}}-\mathbf{\Theta})(\hat{\mathbf{\Theta}}-\mathbf{\Theta})^T\right] \\
&= E\left[(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T(\boldsymbol{z}-\boldsymbol{y})^T(\boldsymbol{z}-\boldsymbol{y})^T\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\right] \\
&= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T E[\boldsymbol{v}\boldsymbol{v}^T]\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} \qquad (4.5.8)
\end{aligned}$$

The only unknown variable in Eq. (4.5.8) is the expected value of noise covariance, i.e., $E[\boldsymbol{v}\boldsymbol{v}^T]$. By assuming that the measurement errors are uncorrelated and have constant variance σ^2, $E[\boldsymbol{v}\boldsymbol{v}^T]$ reduces to $\sigma^2 I$. In a real application, the measurement error variance σ^2 is usually not known but can be estimated through residuals,

$$\begin{aligned}
\hat{\sigma}^2 &= \frac{\boldsymbol{v}^T\boldsymbol{v}}{N-n_p} \\
&= \frac{\sum_{i=1}^{N}\left[\boldsymbol{z}_i-\hat{\boldsymbol{y}}_i\right]^2}{N-n_p} \equiv s^2 \qquad (4.5.9)
\end{aligned}$$

Where N denotes the number of measurement (data point), n_p denotes the number of the unknown parameter, and $\hat{\boldsymbol{y}}_i$ denotes the predicted output. With Eq. (4.5.9) plugged in back into Eq. (4.5.8), the covariance of the estimation error can be obtained. The computed covariance is a square matrix with a dimension of $n_p \times n_p$. The diagonal elements of this matrix represent the variance of the estimates, while the off-diagonal elements represent the correlation coefficients of the estimates.

4.5.1.2 Weighted Least Squares

The formulation of the Ordinary Least Squares (OLS) introduced in Section 4.5.1.1 produces unbiased estimates in the case where noises are distributed uniformly. However, in the case where the noises are distributed non-uniformly, the estimates are biased. To tackle this issue, the Weighted Least Squares (WLS) which is also known as Generalized Least Squares (GLS) is chosen, see Figure 4.12. In WLS, the non-constant variability of the noise is addressed by introducing a weighting variable into the OLS model [26]. The cost function is then modified

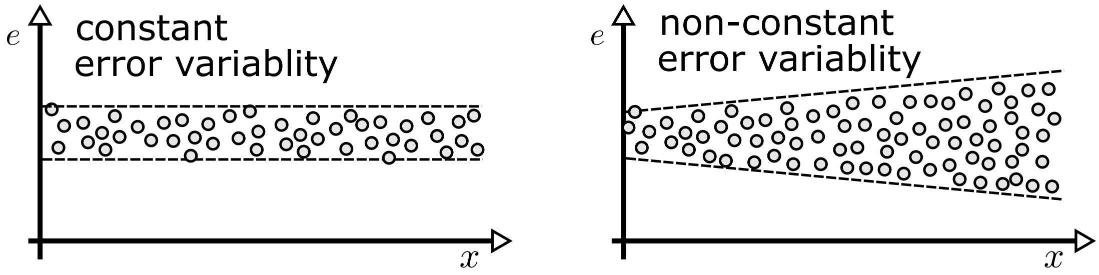


Figure 4.12: Constant and non-constant error variabilities

as given by,

$$J = \frac{1}{2}(\boldsymbol{z} - \mathbf{H\Theta})^T \mathbf{R}(\boldsymbol{z} - \mathbf{H\Theta}) \tag{4.5.10}$$

where $\mathbf{R}$ is the weighting matrix, assumed symmetric positive definite. Taking the first derivative of the cost function with respect to the unknown parameters $\mathbf{\Theta}$ leads to,

$$\frac{\partial J}{\partial \mathbf{\Theta}} = -\boldsymbol{z}^T \mathbf{RH} + \mathbf{\Theta}^T \mathbf{H}^T \mathbf{RH} \tag{4.5.11}$$

Setting Eq. (4.5.10) equal to zero and solving for $\mathbf{\Theta}$, lead to a general formulation of the Weighted Least Squares (WLS) [18],

$$\hat{\mathbf{\Theta}} = (\mathbf{H}^T \mathbf{RH})^{-1} \mathbf{H}^T \mathbf{Rz} \tag{4.5.12}$$

In practice, the weighting matrix is given, which is hard to meet in reality. In the case of the absence of weighting matrix $\mathbf{R}$, a common estimation procedure is performed, i.e., by choosing the weights inversely proportional to the variance at each level of the regressors [18].

4.5.1.3 Total Least Squares

In the formulation of the OLS method as presented in Section 4.5.1.1, it is assumed that noise only affects the independent variables. Therefore, applying this approach to data contain noise both in the independent and dependent variables will yield in biased estimates. The Total Least Squares (TLS) method caters this issue by including the noise in the dependent and independent variables into the formulation. In OLS, the error is formulated as a function of independent variables, which is also the same as minimizing error along one axis (y-axis), see Figure 4.13. On the other hand, TLS reformulates error as a function of independent and

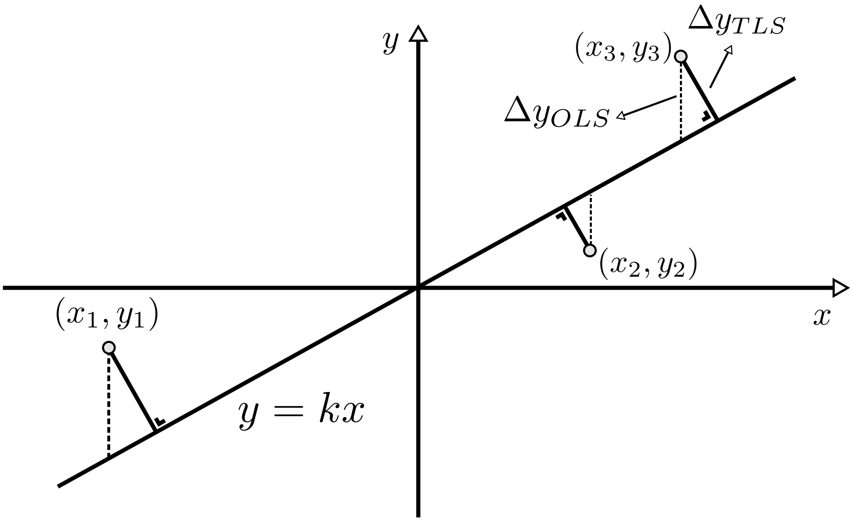


Figure 4.13: Error in the context of OLS and TLS

dependent variables which is the same as minimizing error in both axis (x, y-axis), see Figure

4.13. The model is then modified to take the noise in the independent variables into account as,

$$\boldsymbol{z} = (\mathbf{H} - \boldsymbol{\mu})\boldsymbol{\Theta} + \boldsymbol{v} \tag{4.5.13}$$

where $\boldsymbol{\mu}$ represents noise in the independent variable $\mathbf{H}$. Eq. (4.5.13) is further modified by multiplying out the terms on the right-hand side and bringing all variables to one side yields in [18],

$$[[\mathbf{H} \quad \boldsymbol{z}] - [\boldsymbol{\mu} \quad \boldsymbol{v}]] \begin{bmatrix} \boldsymbol{\Theta} \\ -1 \end{bmatrix} = \left[\tilde{\mathbf{H}} - \tilde{\Delta}\right] \begin{bmatrix} \boldsymbol{\Theta} \\ -1 \end{bmatrix} = 0 \tag{4.5.14}$$

where $\tilde{\mathbf{H}} = [\mathbf{H} \quad \boldsymbol{z}]$ is the compounded data matrix of size $(N \times n_{h+1})$, while $\tilde{\Delta} = [\boldsymbol{\mu} \quad \boldsymbol{v}]$ is the compounded noise vector. The TLS problem is commonly solved through Singular Value Decomposition (SVD). One solution proposed by Golub and van Loan [118] is carried out by writing the compounded matrix $[\mathbf{H} \quad \boldsymbol{z}]$ as SVD given by [18],

$$[\mathbf{H} \quad \boldsymbol{z}] = USV^T \tag{4.5.15}$$

where $S = [\text{diag}(\sigma_1, \sigma_1, \cdots, \sigma_{n_{h+1}})]$, while U and V are the left and right singular matrices. Without going into details, the estimates based on TLS are found to be,

$$\hat{\boldsymbol{\Theta}}_{TLS} = (\mathbf{H}^T\mathbf{H} - \sigma^2_{n_{h+1}})^{-1}\mathbf{H}\boldsymbol{z} \tag{4.5.16}$$

where $\sigma_{n_{h+1}}$ represents the smallest singular value. Expression of Eq. (4.5.16) clearly shows that the noise in the independent variable is now addressed by involving a correction term in the model, i.e, variable $\sigma^2_{n_{h+1}}$.

4.5.2 Output Error Method

Output Error Method (OEM) is derived based on the Maximum Likelihood principle. The OEM provides a general approach in flight vehicle system identification as this method can handle both linear and nonlinear model naturally. Even though the Maximum Likelihood principle can be applied to a system affected by process and measurement noise, OEM is limited for a deterministic system. In this case, the system is not affected by the process noise, but only affected by the measurement noise. A more advanced method, Filter Error Method (FEM), which caters process and measurement noise is discussed separately in Section 4.5.3. In the formulation of the Output Error Method, a nonlinear dynamic system is modeled in a continuous form, while the measurement is formulated in a discrete model. They are expressed as,

$$\dot{\boldsymbol{x}} = f[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}], \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{4.5.17}$$

$$\boldsymbol{y}(t) = g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] \tag{4.5.18}$$

$$\boldsymbol{z}_k = \boldsymbol{y}_k + \mathbf{G}\boldsymbol{v}_k \tag{4.5.19}$$

The state model derivative $\dot{\boldsymbol{x}}$ has been introduced in Chapter 3 as well as the output model $\boldsymbol{y}$ and the measurement model $\boldsymbol{z}$ which are briefly discussed in Section 4.4.2. Thus, no further explanation presented here of all variables defined in the equations. However, as the measurement model is affected by noise, which makes the measurement data nondeterministic, it is required to capture this stochastic behavior by modeling it through a probability density function. One method is through the Maximum Likelihood principle in which the measurement is expressed by the conditional probability density function $p(\boldsymbol{z}|\boldsymbol{\Theta})$. Although the model can be represented in any distribution types, mathematical tractability of the Gaussian distribution is used in the model to capture the stochastic part as introduced by the measurement noise. This assumption is also valid and commonly found in the engineering field, which deals with the system affected by measurement noise. Another advantage of using Gaussian is that the behavior of the stochastic system can completely be described by two variables, i.e., the mean and the covariance variables. Thus, the error between model output and measurement can be formulated as,

$$\boldsymbol{v}_k = \boldsymbol{z}_k - \boldsymbol{y}_k \tag{4.5.20}$$

at every time points k to be statistically independent. With the assumption that the error is distributed as Gaussian with zero mean, the covariance $\mathbf{R}$ can be expressed as,

$$E[\boldsymbol{v}_k] = 0 \tag{4.5.21}$$

$$E[\boldsymbol{v}_{k_i}\boldsymbol{v}_{k_j}^T] = \mathbf{R}\delta_{ij} \tag{4.5.22}$$

where δ_{ij} is the Kronecker delta symbol; $\delta_{ij} = 1$ for $i = j$, and $\delta_{ij} = 0$ for $i \neq j$. For such assumptions, the expression for $p(\boldsymbol{z}|\boldsymbol{\Theta})$ can further be extended to [18, 26],

$$\begin{aligned} p(z_1, z_2, \cdots, z_n|\boldsymbol{\Theta}, \mathbf{R}) &= \prod_{k=1}^{N} p(\boldsymbol{z}_k|\boldsymbol{\Theta}, \mathbf{R}) \\ &= \frac{1}{\sqrt[N]{(2\pi)^{n_y}|\mathbf{R}|}} e^{-0.5\sum_{k=1}^{N} \boldsymbol{v}_k^T \times \mathbf{R}^{-1}\boldsymbol{v}_k} \end{aligned} \tag{4.5.23}$$

The likelihood function $p(\boldsymbol{z}|\boldsymbol{\Theta}, \mathbf{R})$ does not represent the probability distribution of the unknown parameters $\boldsymbol{\Theta}$, but of the measurements, $\boldsymbol{z}$. In the Maximum Likelihood principle, a set of unknown parameters $\boldsymbol{\Theta}$ and covariance matrix $\mathbf{R}$ are given, then Maximum Likelihood will search vector $\boldsymbol{\Theta}$ which gives the highest probability to the measurement [18]. As already introduced in Section 4.4.1, finding the unknown parameters $\boldsymbol{\Theta}$ which maximize the density function $p(\boldsymbol{z}|\boldsymbol{\Theta},\mathbf{R})$ leads to an optimization problem. A more convenient form of the likelihood density function to work with is done by taking its negative logarithmic form and changing the optimization problem to the function minimization which yields the same result as maximizing the density function. Thus, the density function is modified into [18],

$$L(\boldsymbol{z}|\boldsymbol{\Theta},\mathbf{R}) = \frac{1}{2}\sum_{k=1}^{N}[\boldsymbol{z}_k - \boldsymbol{y}_k]^T\mathbf{R}^{-1}[\boldsymbol{z}_k - \boldsymbol{y}_k] + \frac{N}{2}\ln|\mathbf{R}| + \frac{Nn_y}{2}\ln(2\pi) \tag{4.5.24}$$

Where N is the number of data points (measurements) and n_y is the number of the model outputs. Note that, in Eq. (4.5.24), term $\boldsymbol{v}_k$ is expanded as $\boldsymbol{z}_k - \boldsymbol{y}_k$ as defined by Eq. (4.5.20). Minimizing Eq. (4.5.24) with respect to the unknown parameters $\boldsymbol{\Theta}$, requires covariance matrix $\mathbf{R}$ to be known. For such case, two possibilities pertaining to this covariance matrix are considered.

The first case is where the covariance matrix $\mathbf{R}$ is known, which usually can be obtained from the sensor information or from a laboratory testing. In this case, Eq. (4.5.24) is simplified to,

$$J(\boldsymbol{\Theta}) = \frac{1}{2}\sum_{k=1}^{N}[\boldsymbol{z}_k - \boldsymbol{y}_k]^T\mathbf{R}^{-1}[\boldsymbol{z}_k - \boldsymbol{y}_k] \tag{4.5.25}$$

For convenience reason, in Eq. (4.5.25), variable $L(\boldsymbol{z}|\boldsymbol{\Theta},\mathbf{R})$ is renamed to $J(\boldsymbol{\Theta})$. The last two terms in Eq. (4.5.24) are constant and can be discarded from the equation. The cost function of Eq. (4.5.25) can simply be solved through any optimization methods such as Gauss-Newton, Levenberg-Marquardt, or other advanced optimization algorithms where parameters constraints are taken into account [108]. Optimization of the cost function of Eq. (4.5.25) also leads to the same result as in the case of Weighted Least Square (WLS). Thus, for the linear model, the WLS formulation as presented in Eq. (4.5.12) can be applied in this particular case.

In the second case, the covariance matrix $\mathbf{R}$ is unknown. For such a case, the optimization of Eq. (4.5.24) with respect to the unknown parameters $\boldsymbol{\Theta}$ is a bit more complex since there are two sets of unknown variables, $\boldsymbol{\Theta}$ and $\mathbf{R}$. One method to solve this issue is performed through a so-called relaxation strategy in which the optimization of the cost function is carried out through two steps [18, 25]. In the first step, Eq. (4.5.24) is optimized by taking its derivative with respect to the measurement noise covariance $\mathbf{R}$,

$$\frac{\partial J}{\partial \mathbf{R}} = -\frac{1}{2}\mathbf{R}^{-1}\sum_{k=1}^{N}[\boldsymbol{z}_k - \boldsymbol{y}_k][\boldsymbol{z}_k - \boldsymbol{y}_k]\mathbf{R}^{-1} + \frac{N}{2}\mathbf{R}^{-1} \tag{4.5.26}$$

Setting the gradient of the cost function equals to zero and solving for $\mathbf{R}$ yields in an estimator for the measurement noise covariance,

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N}[\boldsymbol{z}_k - \boldsymbol{y}_k][\boldsymbol{z}_k - \boldsymbol{y}_k]^T \tag{4.5.27}$$

Thus, for a given estimate $\hat{\mathbf{R}}$, the cost function is further modified by plugging Eq. (4.5.27) into $\mathbf{R}^{-1}$ variable in Eq. (4.5.24), this leads to,

$$J(\boldsymbol{\Theta}) = \frac{1}{2}n_y N + \frac{N}{2}\ln(|\hat{\mathbf{R}}|) + \frac{Nn_y}{2}\ln(2\pi) \tag{4.5.28}$$

The first and the third term in Eq. (4.5.28) are constant and can be discarded from the equation without affecting the minimization result. Thus, the cost function for unknown measurement noise covariance matrix $\mathbf{R}$ takes form as,

$$J(\boldsymbol{\Theta}) = |\hat{\mathbf{R}}| \tag{4.5.29}$$

Another assumption commonly used in practice is that the estimated covariance matrix $\hat{\mathbf{R}}$ is assumed to be a diagonal matrix. With this simplified form, the cost function in Eq. (4.5.29) is just a product of the variances.

The next step is minimizing the cost function of Eq. (4.5.28) with respect to unknown parameters $\boldsymbol{\Theta}$, given the 'estimated' measurement noise covariance $\hat{\mathbf{R}}$. Optimization of Eq. (4.5.29) can be carried out by any optimization algorithm as in the case of 'known $\mathbf{R}$.' One example presented here is based on the Gauss-Newton algorithm, which is found to work well in flight vehicle system identification. Taking a partial derivative of Eq. (4.5.29) with respect to unknown parameters and equating it to zero yields,

$$\frac{\partial J(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = 0 \tag{4.5.30}$$

This partial derivative can be approached through Taylor series expansion and truncated after two terms as in,

$$\left(\frac{\partial J}{\partial \boldsymbol{\Theta}}\right)_{i+1} \approx \left(\frac{\partial J}{\partial \boldsymbol{\Theta}}\right)_{i} + \left(\frac{\partial^2 J}{\partial \boldsymbol{\Theta}^2}\right)_{i}\Delta\boldsymbol{\Theta} \tag{4.5.31}$$

Plugging Eq. (4.5.31) back into Eq. (4.5.30) and solving for $\Delta\boldsymbol{\Theta}$ yields in,

$$\Delta\boldsymbol{\Theta} = -\left[\left(\frac{\partial^2 J}{\partial \boldsymbol{\Theta}^2}\right)_{i}\right]^{-1}\left(\frac{\partial J}{\partial \boldsymbol{\Theta}}\right)_{i} \tag{4.5.32}$$

where $\Delta\boldsymbol{\Theta} = \boldsymbol{\Theta}_{i+1} - \boldsymbol{\Theta}_i$ is the parameter change and $(\partial^2 J/\partial\boldsymbol{\Theta}^2)$ denotes the second gradient of the cost function, also called Hessian or information matrix. Furthermore, partial differentiation of the likelihood function with respect to unknown parameters $\boldsymbol{\Theta}$ is given by,

$$\frac{\partial J}{\partial \boldsymbol{\Theta}} = -\sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1}[\boldsymbol{z}_k - \boldsymbol{y}_k] \tag{4.5.33}$$

Correspondingly the partial differentiation of $\dfrac{\partial J}{\partial \boldsymbol{\Theta}}$ with respect to $\boldsymbol{\Theta}$ is given by,

$$\frac{\partial^2 J}{\partial \boldsymbol{\Theta}^2} = \sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right] + \sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1}[\boldsymbol{z}_k - \boldsymbol{y}_k] \tag{4.5.34}$$

Eq. (4.5.34) is further simplified by taking the zero mean and independent noise assumptions into account which lead the second term to zero. Thus Eq. (4.5.34) is approximated by,

$$\frac{\partial^2 J}{\partial \boldsymbol{\Theta}^2} \approx \sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right] \tag{4.5.35}$$

Up to this point, all variables required in estimating the unknown parameters have been defined except for response gradient, $\partial \boldsymbol{y}_k/\partial \boldsymbol{\Theta}$, as involved in Eqs.(4.5.33) and (4.5.35). This variable, also called the sensitivity matrix, can be approximated through a numerical approach. One example is forward difference approximation which yields each element of the response gradient matrix, given by [18],

$$\begin{aligned}\left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]_{ij} &\approx \frac{\boldsymbol{y}^p_{i_k} - \boldsymbol{y}_{i_k}}{\delta \boldsymbol{\Theta}}; \qquad i = 1, 2, \cdots, n_y; \quad j = 1, 2, \cdots, n_p \\ &= \frac{g_i[\boldsymbol{x}^p_k, \boldsymbol{u}_k, \boldsymbol{\Theta} + \boldsymbol{\Theta}_j e^j] - g_i[\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\Theta}]}{\delta \boldsymbol{\Theta}_j}\end{aligned} \tag{4.5.36}$$

In practical application, especially pertaining to parameter estimation based on flight measured data, Jategaonkar proposed that parameter perturbation ($\delta\Theta_j$) [18],

$$\delta\Theta_j = 10^{-6}\Theta_j \tag{4.5.37}$$

is found to be a good choice. By denoting the gradient of the cost function as $\mathcal{G}$, called gradient matrix, and the second derivative as $\mathcal{F}$, called information matrix, the solution of the optimization problem of Maximum Likelihood on a system affected by measurement noise is summarized as follows,

- Estimator for measurement noise covariance, $\hat{\mathbf{R}}$

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N}[\boldsymbol{z}_k - \boldsymbol{y}_k][\boldsymbol{z}_k - \boldsymbol{y}_k]^T \tag{4.5.38}$$

- Estimator for unknown parameters, $\boldsymbol{\Theta}$

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i + \Delta\boldsymbol{\Theta} \tag{4.5.39}$$

where $\Delta\boldsymbol{\Theta}$ is approximated by,

$$\mathcal{F}\Delta\boldsymbol{\Theta} = -\mathcal{G} \tag{4.5.40}$$

and $\mathcal{F}$ and $\mathcal{G}$ are given by [18],

$$\mathcal{F} = \sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]; \quad \mathcal{G} = -\sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1}[\boldsymbol{z}_k - \boldsymbol{y}_k] \tag{4.5.41}$$

As in the case of Equation Error Method, the statistical accuracy estimates of the Output Error Method can also be quantified by taking the inverse of the Fisher information matrix $\mathcal{F}$ of Eq. (4.5.35) which leads to,

$$\mathbf{P} = \text{inv}\left[\sum_{k=1}^{N} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} \left[\frac{\partial \boldsymbol{y}_k}{\partial \boldsymbol{\Theta}}\right]\right] \tag{4.5.42}$$

$\mathbf{P}$ is also called the parameter error covariance square matrix in which the diagonal elements denote the standard deviation of the estimates, while the off-diagonal components quantify the correlation coefficients between parameters.

4.5.3 Filter Error Method

Filter Error Method (FEM) provides the most general stochastic approach to aircraft parameter estimation. This approach can handle system affected by process and measurement noise as well as system represented by the linear and nonlinear model [119]. These capabilities make the Filter Error Method become the most advanced and complex system identification method, particularly in the classical flight vehicle system identification field. FEM can be seen as an extension of the Output Error Method since both approaches stand on the foundation of the Maximum Likelihood principle [120]. The difference is that Filter Error Method is designed to be capable of handling process and measurement noises while Output Error Method is restricted on a system with only measurement noise. Precisely, when the Output Error Method is implemented on data in the presence of process and measurement noise, it yields poor estimation results in terms of numerical stability and estimates. However, the Filter Error Method provides a better result and is numerically more stable when applied to such data as both noise types are catered specifically by the algorithm. The stability property of FEM also makes this algorithm suitable for open-loop parameter estimation of an unstable aircraft [18]. FEM was originally introduced by Balakrishnan [121] and adapted by Mehra and Illif to work on estimating aircraft parameters from flight measured data [114, 122] on a linear model.

Even though the extension from linear to nonlinear model is a natural step, no attempt has been made until Jategaonkar and Plaetschke [120, 123] developed such a workable extension. The extension is basically developed so that the original FEM algorithm can handle general model structure through the numerical approach and by involving a mixed state estimator on the prediction step of the nonlinear model as well as correction step using a first-order linear approximation. Filter Error Method may be formulated in different forms depending on model type (linear vs. nonlinear) as well as assumptions regarding the noise affecting the system behavior. For flight vehicle system identification, FEM formulated in a nonlinear with steady-state filter is commonly found in practice. The usage of such filter in this field is supported by the fact that the system parameters are time-invariant and deviations from nominal trajectory are not large due to the selection of the observation time which is usually short. In this formulation, both process and measurement noise are assumed as the additive to the system and distributed as Gaussian with zero mean and identity covariance. The system with such conditions are modeled as [18],

$$\dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] + \mathbf{F}(\lambda)\boldsymbol{w}(t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{4.5.43}$$

$$\boldsymbol{y}(t) = g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] \tag{4.5.44}$$

$$\boldsymbol{z}_k = \boldsymbol{y}_k + \mathbf{G}\boldsymbol{v}_k \tag{4.5.45}$$

where f denotes a nonlinear of the system state model, g represents output model while $\boldsymbol{z}$ is the discrete measurement model. The state model is affected by process noise as denoted by $\boldsymbol{w}(t)$ with its corresponding noise matrix $\mathbf{F}$, which makes the system stochastic. In this form, the propagation of system state cannot be carried out through model integration as in the deterministic case. Therefore, a state estimator is required to deal with this stochastic system in which the Extended Kalman Filter is one of the alternatives for such purpose. While the principle of the Maximum Likelihood is applied for parameter estimation purpose. Based on the formulation of the system affected by process noise and the presence of measurement noise in the observation, the unknown parameter vector $\boldsymbol{\Theta}$ is then given by,

$$\boldsymbol{\Theta} = [\boldsymbol{\beta} \quad \boldsymbol{\lambda}] \tag{4.5.46}$$

which clearly shows that in a stochastic system, more parameters are required to be estimated. Parameter $\boldsymbol{\lambda}$ related to the diagonal element of the process noise distribution matrix $(\mathbf{F})$, while the covariance matrix of the measurement noise is estimated through a relaxation strategy as in the Output Error Method case. The cost function as formulated in Eq. (4.5.24) is used in this method with a minor change due to the disturbance from the process noise [18].

$$J(\boldsymbol{\Theta}, \mathbf{R}) = \frac{1}{2}\sum_{k=1}^{N} [\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k]^T \mathbf{R}^{-1} [\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k] + \frac{1}{2}\ln|\mathbf{R}| + \frac{Nn_y}{2}\ln(2\pi) \tag{4.5.47}$$

The output model $\boldsymbol{y}_k$ is now replaced by the predicted output $\tilde{\boldsymbol{y}}_k$ to incorporate the process noise that affects the system's states. Minimization of this cost function with respect to unknown parameters $\boldsymbol{\Theta}$ is carried out through a two-step relaxation strategy which leads to the same estimator as in the case of the Output Error Method,

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i + \Delta\boldsymbol{\Theta}, \qquad \text{and} \qquad \mathcal{F}\Delta\boldsymbol{\Theta} = -\mathcal{G} \tag{4.5.48}$$

Sub-index i denotes the iteration index, while $\mathcal{F}$ and $\mathcal{G}$ are given by [120],

$$\mathcal{F} = \sum_{k=1}^{N} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right] \tag{4.5.49}$$

$$\mathcal{G} = -\sum_{k=1}^{N} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]^T \mathbf{R}^{-1} [\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k] \tag{4.5.50}$$

The sensitivity coefficient is approximated through numerical difference as in [120],

$$\left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]_{ij} = \frac{\tilde{\boldsymbol{y}}_{ki}^p - \tilde{\boldsymbol{y}}_{ki}}{\delta \boldsymbol{\Theta}_j} \tag{4.5.51}$$

The perturbed response variable $\tilde{\boldsymbol{y}}^p$ is obtained by replacing $\boldsymbol{\Theta}$ with $\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_j e^j$ in the Eqs. (4.5.43) - (4.5.44). Furthermore, the predicted output is computed by employing the nonlinear Kalman Filter in a two-step procedure,

Prediction step

$$\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \int_k^{k+1} f\left[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}\right] \mathrm{d}t, \qquad \hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0 \tag{4.5.52}$$

$$\tilde{\boldsymbol{y}}_k = g\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{u}_k, \boldsymbol{\beta}\right) \tag{4.5.53}$$

Correction step

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}\left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k\right] \tag{4.5.54}$$

The steady state Kalman gain $\mathbf{K}$ is computed through the steady-state of $\mathbf{R}$, $\mathbf{P}$, and $\mathbf{C}$, given by [18],

$$\mathbf{K} = \mathbf{P}\mathbf{C}^T\mathbf{R}^{-1} \tag{4.5.55}$$

The steady-state covariance matrix $\mathbf{P}$ is computed by solving the Riccati equation, while variable $\mathbf{C}$ is approximated through [18],

$$\mathbf{C} = \left[\frac{\partial g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}]}{\partial x}\right]_{t=t_0} \tag{4.5.56}$$

In the Filter Error Method, the exact formulation of $\mathbf{R}$ takes a different form as in the case of the Output Error Method. In the Output Error Method, the covariance of measurement noise

$\mathbf{R}$ can be directly linked to being pure noise from the sensors, $\mathbf{R} = \mathbf{G}\mathbf{G}^T$ [18]. However, in the Filter Error Method, this model is not valid anymore as the sensors also pick the noise that comes from the environment. Therefore, the covariance of the measurement noise is modified to include this noise. Furthermore, this noise covariance is called residual noise covariance to differentiate with measurement noise covariance, which is purely affected by noise from the sensor [18]. Accordingly, this residual covariance matrix is modeled as [18],

$$\mathbf{R} = \mathbf{G}\mathbf{G}^T + \mathbf{C}\mathbf{P}\mathbf{C}^T \tag{4.5.57}$$

However, as the cost function formulated in Eq. (4.5.47) depends on total $\mathbf{R}$, the computation of this residual covariance matrix ($\mathbf{R}$) through Eq. (4.5.57) can be avoided, instead Eq. (4.5.38) which is obtained from the relaxation strategy can be used for this purpose,

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N}[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k][\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k]^T \tag{4.5.58}$$

Computational details of the Filter Error Method will be provided in Chapter 5 since the method proposed in this dissertation is developed based on the Filter Error algorithm.

4.5.4 Extended Kalman Filter

In Section 4.4.2, Kalman Filter for state estimation has been introduced. The linear Kalman Filter framework is extended to deal with the nonlinear model by linearizing the system states at a particular time step yields in a so-called Extended Kalman Filter (EKF). This approach has been widely used in practice as it provides a quick and general approximation for the nonlinear model. Besides being implemented for state estimation, EKF also provides a solution for the combined state and parameter estimation problem [124]. This dual-estimation problem is commonly found when dealing with system identification where both states and parameters of the investigated system are required to be estimated simultaneously. In aerospace application, particularly related to the estimation of aerodynamic parameters, the EKF-based dual estimation is commonly used in practice [18, 101, 125]. The same as the FEM, the EKF algorithm is also meant for dealing with systems affected by the process and measurement noises. However, applying this method to such data requires specification of process and measurement noise statistics. EKF for dual-estimation is carried out through the inclusion of parameters along with states into 'system states.' Both parameters and states are stacked together in the so-called augmented state vector, $\boldsymbol{x}_a$.

$$\boldsymbol{x}_a = [\boldsymbol{x} \quad \boldsymbol{\Theta}]^T \tag{4.5.59}$$

The unknown parameters $\boldsymbol{\Theta}$ are assumed to be constant over time. This leads to,

$$\dot{\boldsymbol{\Theta}} = 0 \tag{4.5.60}$$

Consequently, the general formulation of the nonlinear model for the dual-estimation purpose is reformulated as [101],

$$\begin{aligned}\dot{\boldsymbol{x}}_a(t) &= \begin{bmatrix} f[\boldsymbol{x}_a(t), \boldsymbol{u}(t)] \\ 0 \end{bmatrix} + \left[\begin{array}{c|c} \mathbf{F} & 0 \\ \hline 0 & 0 \end{array}\right] \begin{bmatrix} \boldsymbol{w}(t) \\ 0 \end{bmatrix} \\ &= f_a\left[\boldsymbol{x}_a(t), \boldsymbol{u}(t)\right] + \mathbf{F}_a \boldsymbol{w}_a(t) \end{aligned} \tag{4.5.61}$$

Correspondingly, the output and measurement models are formulated as,

$$\boldsymbol{y}(t) = g_a\left[\boldsymbol{x}_a(t), \boldsymbol{u}(t)\right] \tag{4.5.62}$$

$$\boldsymbol{z}_k = \boldsymbol{y}_k + \mathbf{G}\boldsymbol{v}_k \tag{4.5.63}$$

The computational procedures for EKF as defined in Table 4.3 remains the same except for details computations. The EKF procedures for dual-estimation problem are summarized in Table 4.4 [18].

Steps	Description and formula
①	**Set initial values** $\tilde{\boldsymbol{x}}_{a_0}, \tilde{\mathbf{P}}_{a_0}, \mathbf{Q}_a,$ and $\mathbf{R}$
②	**Kalman gain computation** $\mathbf{K}_{a_k} = \tilde{\mathbf{P}}_{a_k}\mathbf{C}_a^T\left[\mathbf{C}\tilde{\mathbf{P}}_{a_k}^T\mathbf{C}^T + \mathbf{R}\right]^{-1}$ where, $\mathbf{C}_{a_k} = \left.\frac{\partial g_a\left[\boldsymbol{x}_a(t), \boldsymbol{u}(t), \boldsymbol{\beta}\right]}{\partial x_a}\right\vert_{\boldsymbol{x}_a=\tilde{\boldsymbol{x}}_{a_k}} = \left[\frac{\partial g}{\partial x} \middle\vert \frac{\partial g}{\partial \Theta}\right]_{\boldsymbol{x}_a=\tilde{\boldsymbol{x}}_{a_k}}$
③	**Correction step (measurement update)** $\tilde{\boldsymbol{y}}_k = g_a\left[\tilde{\boldsymbol{x}}_{a_k}, \boldsymbol{u}_k, \boldsymbol{\beta}\right]$ $\hat{\boldsymbol{x}}_{a_k} = \tilde{\boldsymbol{x}}_{a_k} + \mathbf{K}_{a_k}\left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k\right]$ $\hat{\mathbf{P}}_{a_k} = \left[\mathbf{I} - \mathbf{K}_{a_k}\mathbf{C}_a\right]\tilde{\mathbf{P}}_{a_k}\left[\mathbf{I} - \mathbf{K}_{a_k}\mathbf{C}_a\right]^T + \mathbf{K}_{a_k}\mathbf{R}\mathbf{K}_{a_k}^T$ or using the short form, $\hat{\mathbf{P}}_{a_k} = \left[\mathbf{I} - \mathbf{K}_{a_k}\mathbf{C}_a\right]\tilde{\mathbf{P}}_{a_k}$
④	**Prediction step (time update)** $\tilde{\boldsymbol{x}}_{a_{k+1}} = \hat{\boldsymbol{x}}_{a_k} + \int_{t_k}^{t_{k+1}} f\left[\hat{\boldsymbol{x}}_a(t), \boldsymbol{u}(t_k), \boldsymbol{\beta}\right]\mathrm{d}t$ $\tilde{\mathbf{P}}_{a_k+1} = \boldsymbol{\Phi}_a\hat{\mathbf{P}}_{a_k}\boldsymbol{\Phi}_a^T + \Delta t\mathbf{F}_a\mathbf{F}_a^T$ where, $\Delta t\mathbf{F}_a\mathbf{F}_a^T = \mathbf{Q}_a$ $\boldsymbol{\Phi}_{k+1} = e^{\mathbf{A}\Delta t} \approx \mathbf{I} + \mathbf{A}\Delta t + \mathbf{A}^2\frac{\Delta t^2}{2!} + \cdots$ $\mathbf{A}_{a_k} = \left.\frac{\partial g_a\left[x_a(t), u(t)\right]}{\partial x}\right\vert_{x_a=\hat{x}_a} = \left[\begin{array}{ccc} \partial f/\partial x & \vert & \partial f/\partial \Theta \\ 0 & \vert & 0 \end{array}\right]_{x_a=\hat{x}_a}$

Table 4.4: EKF computational steps for dual-estimation problem

As indicated in Table 4.4, variables affected due to the inclusion of the unknown parameters into the augmented state vector are now denoted by subscript a. Some points to be stressed out are related to noise covariances $\mathbf{R}$ and $\mathbf{Q}_a$. The covariance matrix $\mathbf{R}$ represents the measurement noise covariance. The statistics of this noise can be obtained through the characteristics of sensors or can also be computed through the Fourier decomposition approach by utilizing the measurement data [75].

The augmented process noise covariance, $\mathbf{Q}_a$, is a non-measurable quantity which requires a different treatment. However, this matrix, which takes form as a diagonal matrix, is commonly found in practice. The first n_x elements represent the state process noise, and the remaining n_q elements reflect the nature of the parameters. If the unknown parameters are constant during the selected time window, their corresponding process noise can be assumed as zero. However, if parameters change during the period of observation, the covariance elements should be set large enough to allow the variation tracking of the corresponding parameters. A more advanced method for obtaining both of these noise statistics are addressed as an adaptive filtering [110, 126]. In this approach, the measurement noise and process noise covariances are estimated through an iterative process within the EKF algorithm.

Chapter 5

Extension of System Identification Method for FDM Data

This chapter presents a flight vehicle system identification method tailored to FDM data. The chapter begins by revisiting the Filter Error Method with emphasizing on the computational details. This is followed by introducing a new method that is tailored specifically to FDM data. Derivation of the formula, as well as computational details of the proposed method, are also given in this chapter. Implementation aspects of the proposed method are given afterward. To validate the proposed method, Section 5.4 provides a step in which the method is applied to simulated data. In this simulated data, all noise statistics and system parameters are known and re-estimated using the proposed method. The evaluation of the proposed method from aspects related to computation complexity and running time, the convergence properties, statistical properties of the parameters and their respective accuracies are covered in the last section of this chapter.

5.1 Filter Error Method-Revisited

In this section, the Filter Error Method (FEM) is presented again with the emphasis on the algorithm details and computational procedures. The FEM may take different forms depending on the model used to represent the underlying system, either a system takes a linear or nonlinear model [18]. As already pointed out in Chapter 4, FEM employs Extended Kalman Filter for state estimation and maximum likelihood principle for updating the unknown parameters. These two process are carried out iteratively until the convergence criteria is achieved. Figure

5.1 depicts a schematic diagram of FEM along with a system affected by noises. The figure shows a typical application of the FEM algorithm on the system that is disturbed by process and measurement noises.

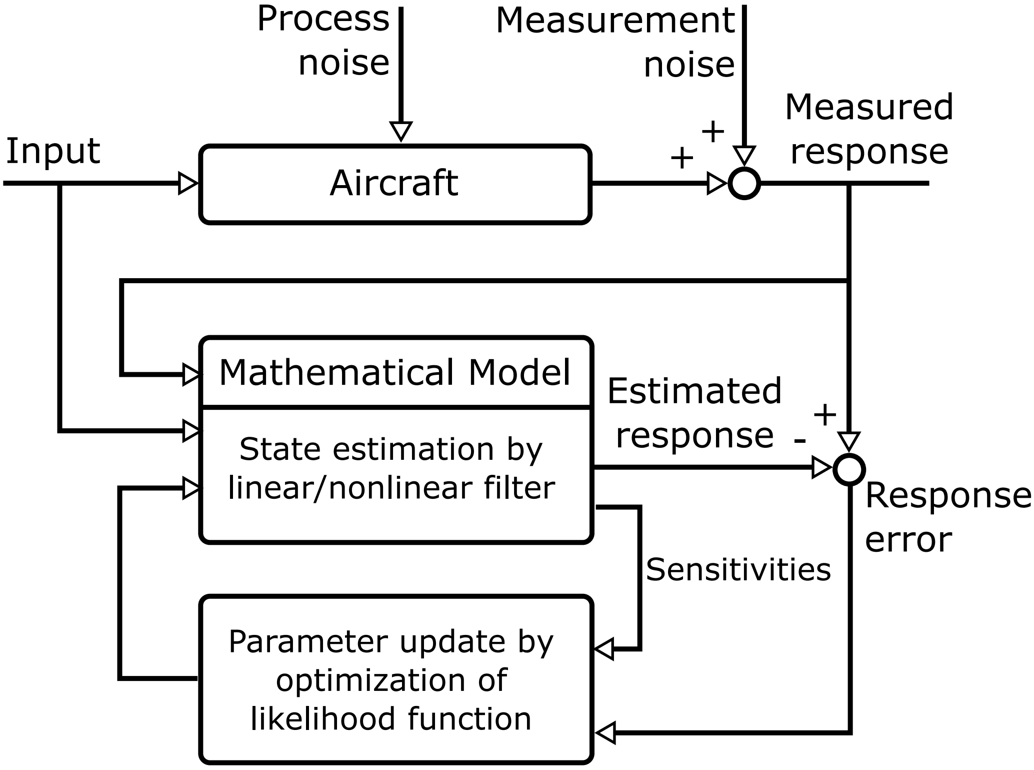


Figure 5.1: Disturbed system and FEM application [120]

For showing the computational procedures of the FEM algorithm, a nonlinear model will be employed. Precisely, a nonlinear model affected by additive process noise will be used to represent the system dynamics. While, for representing the output model ($\boldsymbol{y}$), an algebraic model will be used. The corresponding measurement model ($\boldsymbol{z}$) is represented by adding the additive noise to the output model. Eqs. (4.5.43), (4.5.44), and (4.5.45) respectively denote state model, output model, and observation model. Each of the equations are rewritten here for clarity [120].

$$\dot{\boldsymbol{x}}(t) = f[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] + \mathbf{F}(\lambda)\boldsymbol{w}(t), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{5.1.1}$$

$$\boldsymbol{y}(t) = g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}] \tag{5.1.2}$$

$$\boldsymbol{z}_k = \boldsymbol{y}_k + \mathbf{G}\boldsymbol{v}_k \tag{5.1.3}$$

The unknown parameters given in a vector notation is written as,

$$\boldsymbol{\Theta} = [\boldsymbol{\beta} \quad \boldsymbol{\lambda}]^T \tag{5.1.4}$$

Where $\boldsymbol{\beta}$ represents system parameters and $\boldsymbol{\lambda}$ denotes the diagonal components of matrix $\mathbf{F}$. Each of the components in ($\boldsymbol{\lambda}$) represents the scale factor of the process noise distribution matrix. In some cases, biases that result from the unknown initial state conditions ($\boldsymbol{x}_0$) and sensors' misalignment affect the quality of the measurement. Thus, in modeling the underlying system, these biases need to take into account. With biases included in the model, the unknown parameters are extended to,

$$\boldsymbol{\Theta} = [\boldsymbol{\beta} \quad \boldsymbol{\lambda} \quad \Delta]^T \tag{5.1.5}$$

The process noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$ are assumed to be characterized by Gaussian distribution with zero-mean with an identity spectral density matrix [120]. These two noises are also assumed to be independent. The noise distribution matrix $\mathbf{F}$ and $\mathbf{G}$ are assumed to be time-invariant. Furthermore, the measurement noise covariance $\mathbf{GG}^T$ can be modeled in terms of the residual covariance $(\mathbf{R})$ and the uncertainty in the states $(\mathbf{P})$, expressed as [120, 123],

$$\mathbf{R} = \mathbf{GG}^T + \mathbf{CPC}^T \tag{5.1.6}$$

Residual covariance matrix $(\mathbf{R})$ is also unknown and required to be estimated. However, in the FEM algorithm, the covariance matrix $(\mathbf{R})$ is treated differently and estimated in a separated stage within the algorithm. It should also be noted here that the covariance matrix $(\mathbf{R})$ in the FEM algorithm is not the same as in the OEM algorithm. This is because the covariance matrix $(\mathbf{R})$ in the FEM algorithm contains not only noise from the measurement but also uncertainty in the model that might result from the process noise or model deficiency. In the OEM algorithm, the system dynamics is deterministic and assumed to be free of noise. The difference formulation of $\mathbf{R}$ in the FEM algorithm compared to the OEM algorithm can be directly seen by an additional term $\mathbf{CPC}^T$ as presented in Eq. (5.1.6). By assuming the system is time-invariant, FEM is carried out through the following procedures [18]:

1. Suitable starting values for system parameters $\boldsymbol{\beta}$, initial conditions $\boldsymbol{x}_0$, diagonal elements of the process noise distribution matrix $\mathbf{F}$ and diagonal elements of the residual covariance matrix $\mathbf{R}$ are specified.
2. Applying the Extended Kalman Filter (EKF) for state estimation. Predicted state $(\tilde{\boldsymbol{x}}_k)$ and corrected state $(\hat{\boldsymbol{x}}_k)$ are obtained from this step and the corresponding predicted output $(\tilde{\boldsymbol{y}}_k)$ is computed. In this step, the steady-state Kalman gain $(\mathbf{K})$, steady state error covariance $\mathbf{P}$, steady-state $\mathbf{C}$ of the linearized system, and steady-state residual noise covariance $\mathbf{R}$ are used instead of the time-varying form. For the predicted state estimation, the following Eq. (5.1.7) is used [123],

$$\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \int_k^{k+1} f\left[\boldsymbol{x}_k, \bar{\boldsymbol{u}}_k, \boldsymbol{\beta}\right] \mathrm{d}t, \quad \hat{\boldsymbol{x}}(t_0) = \boldsymbol{x}_0 \tag{5.1.7}$$

Where $\bar{\boldsymbol{u}}_k$ is the interpolated input from k to $k+1$. Correspondingly, the predicted output is computed through Eq. (5.1.8),

$$\tilde{\boldsymbol{y}}_k = g\left[\tilde{\boldsymbol{x}}_k, \boldsymbol{u}_k, \boldsymbol{\beta}\right] \tag{5.1.8}$$

The corrected state is computed using Eq. (5.1.9),

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}(\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k) \tag{5.1.9}$$

While the steady-state Kalman gain is computed through Eq. (5.1.10) [18]

$$\mathbf{K} = \mathbf{P}\mathbf{C}^T\mathbf{R}^{-1} \tag{5.1.10}$$

where $\mathbf{P}$ is the state error covariance in the steady-state form. This variable is computed by solving the steady-state Ricatti equation [114].

$$\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T - \frac{1}{\Delta t}\mathbf{P}\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\mathbf{P} + \mathbf{F}\mathbf{F}^T = 0 \tag{5.1.11}$$

The linearized system matrix $\mathbf{A}$ and $\mathbf{C}$ are formulated as,

$$\mathbf{A} = \left[\frac{\partial f[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}]}{\partial x}\right]_{t=t_0} \tag{5.1.12}$$

$$\mathbf{C} = \left[\frac{\partial g[\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\beta}]}{\partial x}\right]_{t=t_0} \tag{5.1.13}$$

in which both matrices can be numerically approximated through the central-difference formula as,

$$\mathbf{A}_{ij} \approx \left.\frac{f_i\left[\boldsymbol{x} + \delta_j e^j, \boldsymbol{u}, \boldsymbol{\beta}\right] - f_i\left[\boldsymbol{x} - \delta_j e^j, \boldsymbol{u}, \boldsymbol{\beta}\right]}{2\delta x_j}\right|_{x=x_0} \tag{5.1.14}$$

$$\mathbf{C}_{ij} \approx \left.\frac{g_i\left[\boldsymbol{x} + \delta_j e^j, \boldsymbol{u}, \boldsymbol{\beta}\right] - g_i\left[\boldsymbol{x} - \delta_j e^j, \boldsymbol{u}, \boldsymbol{\beta}\right]}{2\delta x_j}\right|_{x=x_0} \tag{5.1.15}$$

where e^j is a column vector with one in the jth row and zeros elsewhere.

3. Computing the residual covariance matrix through Eq. (5.1.16). In this formulation, the computed residual covariance matrix can be seen as a representation of the noise covariance matrix since the uncertainty in the model is included in the predicted output ($\tilde{\boldsymbol{y}}$). Note that, the model used for computing the residual covariance matrix $\mathbf{R}$ takes the same form as in the case of the Output Error Method case. The different is that, in the FEM the predicted output $\tilde{\boldsymbol{y}}$ is used instead of $\boldsymbol{y}$ as the system cannot be assumed as a deterministic system since the system is corrupted by the process noise.

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{k=1}^{N}\left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k\right]\left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k\right]^T \tag{5.1.16}$$

4. Applying the Gauss-Newton method to update $\boldsymbol{\Theta}$.

$$\boldsymbol{\Theta}_{i+1} = \boldsymbol{\Theta}_i + \boldsymbol{\Delta\Theta} \quad \text{where} \quad \mathcal{F}\Delta\boldsymbol{\Theta} = -\mathcal{G} \tag{5.1.17}$$

where i is the iteration index. $\mathcal{F}$ and $\mathcal{G}$ are given by,

$$\mathcal{F} = \sum_{k=1}^{N} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]^T \hat{\mathbf{R}}^{-1} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right] \tag{5.1.18}$$

$$\mathcal{G} = -\sum_{k=1}^{N} \left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]^T \hat{\mathbf{R}}^{-1} [\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k] \tag{5.1.19}$$

Computation of $\mathcal{F}$ and $\mathcal{G}$ requires the sensitivities output which can be computed through,

$$\left[\frac{\partial \tilde{\boldsymbol{y}}_k}{\partial \boldsymbol{\Theta}}\right]_{ij} \approx \frac{\tilde{\boldsymbol{y}}_k^{pi} - \tilde{\boldsymbol{y}}_k^{i}}{\delta \boldsymbol{\Theta}_j} \tag{5.1.20}$$

Variable $\tilde{\boldsymbol{y}}^p$ denotes the perturbed system output which is computed based on Eqs. (5.1.7), (5.1.8), and (5.1.9) by replacing $\boldsymbol{\Theta}$ with $\boldsymbol{\Theta} + \delta\boldsymbol{\Theta}_j e^j$. Thus, for each elements of $\boldsymbol{\Theta}$, the perturbed states and the corresponding outputs are computed through the EKF procedures, that is, prediction and correction steps, given by,
Prediction step:

$$\tilde{\boldsymbol{x}}_{k+1}^{p\_j} = \hat{\boldsymbol{x}}_k^{p\_j} + \int_k^{k+1} f\left[\boldsymbol{x}_k^{p\_j}, \bar{\boldsymbol{u}}_k, \boldsymbol{\beta} + \delta\boldsymbol{\beta}_j e^j\right] \mathsf{d}t \tag{5.1.21}$$

$$\tilde{\boldsymbol{y}}_k^{p\_j} = g\left[\tilde{\boldsymbol{x}}_k^{p\_j}, \boldsymbol{u}_k, \boldsymbol{\beta} + \delta\boldsymbol{\beta}_j e^j\right] \tag{5.1.22}$$

Correction step:

$$\hat{\boldsymbol{x}}_k^{p\_j} = \tilde{\boldsymbol{x}}_k^{p\_j} + \mathbf{K}^{p\_j}\left[\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k^{p\_j}\right] \tag{5.1.23}$$

Since the states are perturbed, the corresponding perturbed Kalman gain $\mathbf{K}$ must also be computed as in,

$$\mathbf{K}^{p\_j} = \mathbf{P}^{p\_j}[\mathbf{C}^{p\_j}]^T \hat{\mathbf{R}}^{-1} \tag{5.1.24}$$

where p represents the perturbed variables and j the index for the parameter being varied. Again, Eq. (5.1.11) along with Eqs. (5.1.12) and (5.1.13) are used to compute $\mathbf{P}^p$. Since $\hat{\mathbf{R}}$ is affected by $\mathbf{F}$ and $\mathbf{G}$, it is necessary to ensure that $\mathbf{G}\mathbf{G}^T$ covariance matrix is to be physically meaningful, i.e., its value is positive semi-definite. The covariance matrix $\mathbf{G}\mathbf{G}^T$ can be computed indirectly from Eq. (5.1.6) Thus, in order to meet the required condition for the measurement noise covariance $\mathbf{G}\mathbf{G}^T$, the eigenvalues of $\mathbf{KC}$ must be less than 1 [114]. Furthermore, since estimations of $\hat{\mathbf{R}}$ and $\boldsymbol{\Theta}$ are carried out independently through the two-step procedure, it may yield strongly correlated matrices between $\mathbf{R}$ and $\mathbf{F}$. Thus, to account for this correlation, Maine and Illif [114] suggest to compensate matrix $\mathbf{F}$ whenever matrix $\hat{\mathbf{R}}$ is revised. This procedure is formulated as,

$$\mathbf{F}_{ij}^{new} = \mathbf{F}_{ij}^{old} \left(\frac{\sum_{j=1}^{n_y} \mathbf{C}_{ij}^2 r_j^{old} \sqrt{r_j^{old}/r_j^{new}}}{\sum_{j=1}^{n_y} \mathbf{C}_{ij}^2 r_j^{old}} \right) \tag{5.1.25}$$

where r_j is the jth diagonal element of $\hat{\mathbf{R}}^{-1}$ and superscripts "old" and "new" represent the previous and revised estimates respectively.

5. Steps 2 - 4 are iterated until it converges. This last step is terminated once the relative cost function is less than the defined tolerance in the last two consecutive iterations. Typically, a relative cost function value less than **1E-04** is sufficiently enough and commonly found in the flight vehicle system identification field [18]. The relative cost function is computed through,

$$\|J_{\text{relative}}\| = \frac{J_k - J_{k-1}}{J_{k-1}} \tag{5.1.26}$$

where J is the cost function, given as,

$$J = \det(\hat{\mathbf{R}}) \tag{5.1.27}$$

As indicated at the beginning of this section, the procedures presented above are based on the assumption of the time-invariant filter in which the deviations from the nominal trajectory are small. However, if the nonlinearities dominate the system response, the time-varying filter should be used. In this time-varying filter, the FEM algorithm takes different formulations as in the case of time-invariant filter. It also adds more complexity into the algorithm as noise covariances are required to be updated in every time-steps. Even though the time-varying filter leads to more complex algorithm than the time-invariant filter; they share the same principle, i.e., the EKF algorithm is employed to filter the state noise while all unknown parameters except the residual covariance matrix $(\mathbf{R})$ are updated through the maximum likelihood principle as in the case of the Output Error Method. To conclude, the following Fig. 5.2 depicts the FEM algorithm for the nonlinear model.

5.2 Developed Method

The new method presented here is developed to overcome the instability issue of the Filter Error Method due to too many unknown parameters that are required to be estimated. In the general case, it is found that too many parameters to be estimated may lead to identifiability as well as divergence issues, particularly when a parameter estimation method is applied to data with low information content [120]. Furthermore, a model with highly correlated and linear dependency parameters also contribute to the deterioration of the quality of the es-

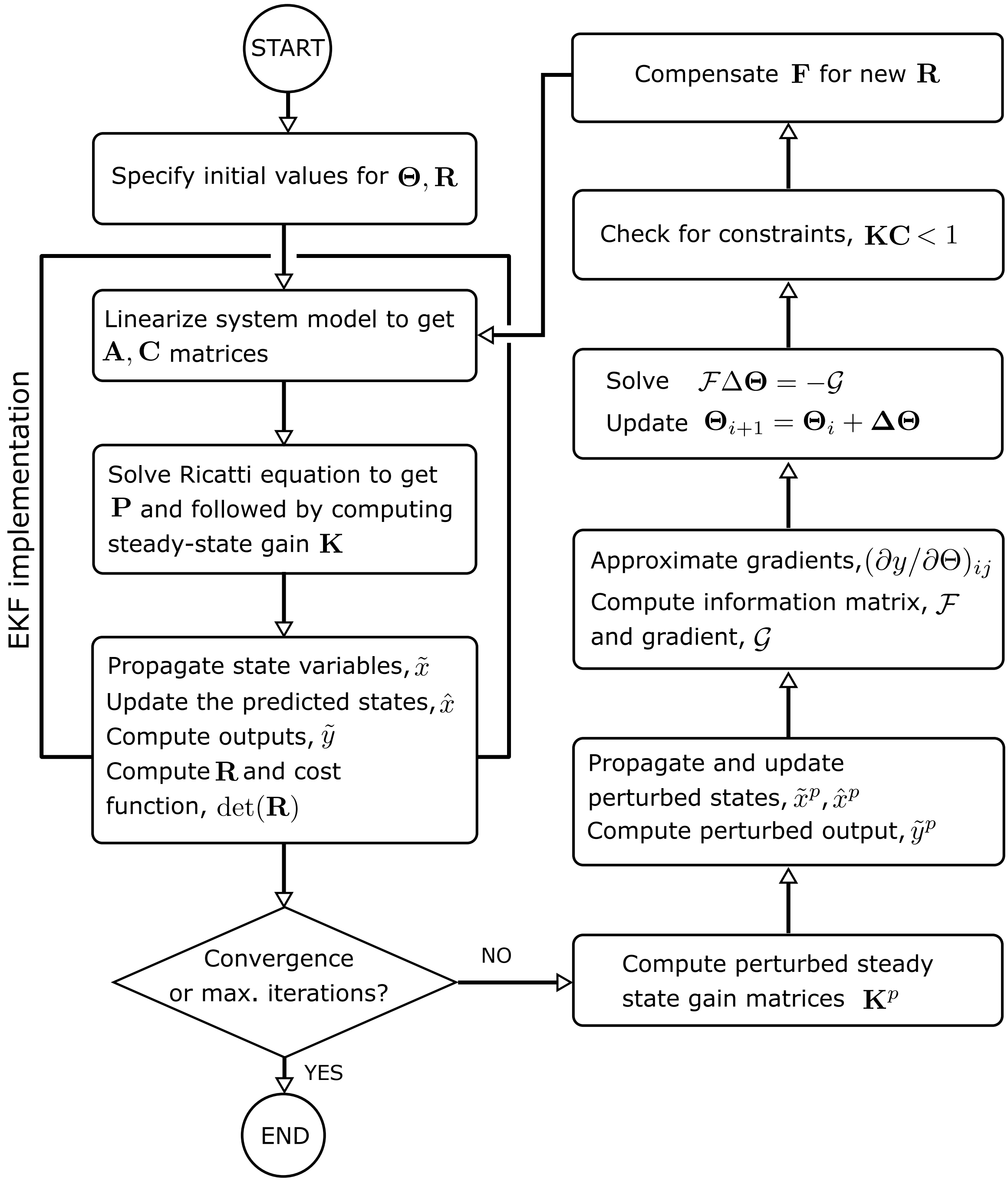

START
Specify initial values for $\mathbf{\Theta}, \mathbf{R}$
EKF implementation
Linearize system model to get $\mathbf{A}, \mathbf{C}$ matrices
Solve Ricatti equation to get $\mathbf{P}$ and followed by computing steady-state gain $\mathbf{K}$
Propagate state variables, $\tilde{x}$
Update the predicted states, $\hat{x}$
Compute outputs, $\tilde{y}$
Compute $\mathbf{R}$ and cost function, $\det(\mathbf{R})$
Convergence or max. iterations?
NO
YES
END
Compute perturbed steady state gain matrices $\mathbf{K}^p$
Propagate and update perturbed states, $\tilde{x}^p, \hat{x}^p$
Compute perturbed output, $\tilde{y}^p$
Approximate gradients, $(\partial y / \partial \Theta)_{ij}$
Compute information matrix, $\mathcal{F}$ and gradient, $\mathcal{G}$
Solve $\mathcal{F}\Delta\mathbf{\Theta} = -\mathcal{G}$
Update $\mathbf{\Theta}_{i+1} = \mathbf{\Theta}_i + \mathbf{\Delta\Theta}$
Check for constraints, $\mathbf{KC} < 1$
Compensate $\mathbf{F}$ for new $\mathbf{R}$

Figure 5.2: FEM computational procedures [18]

timates. Applying FEM to FDM data leads to both issues mentioned previously. The low information content in FDM data, as well as too many parameters that are required to be estimated simultaneously, tend to fail the FEM algorithm in estimating the parameters of interest. To solve these two issues, particularly the issue related to the number of parameters to be estimated, this dissertation proposes a novel strategy which separately estimates the noise statistics and system related parameters in two different stages. A strategy for enrichment of FDM data content information is discussed separately in Chapter 6 Section 6.4.3 where a strategy of simultaneously processing several flights will be introduced. The approach of separately estimating noise related parameters and system related parameters have been published by the author in a Scopus listed IEEE[1] publication [127]. The strategy of separating the estimation of noise statistics and system parameters ensures the stability of the computation. This constitutes the second contribution of this dissertation as addressed in Chapter 1, Section 1.4.

In this developed method, the statistics of the process and measurement noises are estimated based on the filtering approach while the system parameters (β) are estimated through the output-error approach. Thus, two phases of the estimation steps are involved in this proposed parameter estimation technique. In the first step, the Extended Kalman Filter followed by smoothing algorithm is employed only for estimating the process noise covariance $\mathbf{Q}$ and the residual noise covariance $\mathbf{R}$ matrices. The process of estimating both noise statistics are performed iteratively until the convergence criteria of the defined cost function are achieved. In the second loop, the modified Filter Error Method is used for estimating system parameters by utilizing the estimated noise statistics obtained from the first step. The principle of the Maximum Likelihood is applied to only estimate the system parameters. With the noise statistics which are given from the first step, a 'relaxation strategy' which is required in the Filter Error Method is skipped from the steps. The number of parameters to be estimated is also reduced as the algorithm only deals with estimating the system parameters. A fewer number of estimates leads to the convergence of the computation.

In contrast, the Filter Error Method simultaneously estimates the process and system parameters through the Maximum Likelihood principle while the measurement noise covariance matrix is estimated through a relaxation strategy [18, 120]. This approach, when applied to FDM data with low information content, leads to the instability of the computation due to the simultaneous estimation of the process and system parameters. This is the inadequacy of the FEM making its application to the FDM data unattractive. The new method proposed in this dissertation shall remove this problem.

Furthermore, in the flight vehicle system identification field, there is a two-step approach, or called as estimation-before-modeling [25], which works the same way as the method proposed in this dissertation. The two methods are categorized as the same approach since both methods employ the two-step procedure. However, both methods are different in many aspects. In the estimation-before-modeling approach, the noise statistics are estimated in the first loop through the kinematic model of the aircraft in which the filtering/smoothing algorithm is applied to enhance the quality of the measurement. While in the second loop, a different model is utilized. This model usually related to aircraft dynamics in terms of forces and moments that are required to be estimated. Typically, a least-square or Equation Error Method is employed in the second step. With the two different models utilized in this method, it is less transparent

[1] Institute of Electrical and Electronic Engineers

and a bit difficult to assess the performance of the algorithm if the quality of the estimates are poor. This is because the link between the first and second step is not strong. The quality of estimates from the first step affects the quality of the second step but not in the backward direction. Thus, any poor estimates obtained from the second loop can be affected by the model or filtering algorithm used in the first step or the model and algorithm used in the second step.

In contrast, the developed algorithm only utilize one dynamic model in which the system parameters, along with the noise statistics, are embedded in the model. Then, noise related parameters and system parameters are estimated in different stages through two different estimation approaches. Since only one dynamic model is utilized in the developed method, it consists of simpler and more tractable procedures compared to the estimation-before-modeling approach. In the following Sections 5.2.1 - 5.2.3, the proposed estimation method is elaborated in details.

5.2.1 Noise Statistics Estimation

Noise covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ play an important role in the Kalman Filter-based approaches, including the implementation of the EKF for state or dual-estimation problems. As presented in Section 4.4.2 for general Kalman Filter and Section 4.5.4 for dual estimation problem, the two covariance matrices are integral part of the algorithm and if they are not correctly specified, one might expect the biased estimates [128]. This might happen even when the system parameters (β) are known but incorrect noise statistics are employed in the algorithm, the EKF may result in poor estimates. In the worst case when the algorithm is fed with incorrect noise statistics, it might lead to divergence issue which consequently produces unreliable estimates [129]. Thus, providing the correct values of $\mathbf{R}$ and $\mathbf{Q}$ to the filter is of a paramount importance in the EKF algorithm. Normally, components in the noise covariance matrix $\mathbf{R}$ are obtained through a laboratory test or from a sensor specification [18], while the process noise covariance ($\mathbf{Q}$) cannot be measured through such a test nor from the sensor's specification. The matrix covariance ($\mathbf{Q}$) depends on the environment condition in which the corresponding noise statistics are only valid at a particular condition. In practical implementation, this covariance matrix is typically adjusted in an ad hoc manner or through an engineering judgment [130].

In the present technique, since both $\mathbf{R}$ and $\mathbf{Q}$ information are not available, the EKF algorithm is employed iteratively over the whole data points to produce the estimation of the two covariance matrices. This technique is commonly known as adaptive filtering. There are several common approaches that can be employed to estimate these noise statistics. Ananthasayanam *et al.* [129] noted 4 different approaches to compute such covariance matrices, namely Bayesian approach [128], maximum likelihood approach [131, 132], covariance matching [129, 130, 133, 134, 135, 136, 137], and correlation approach [138, 139, 140, 141]. In the present work, the covariance matching approach is adopted. The algorithm proposed by Tapley and Myers [133] with the improvement from Leathrum[136] is embedded in the EKF algorithm for estimation of the noise statistics $(\mathbf{R}, \mathbf{Q})$. Furthermore, algorithm from Gemson [142] is employed for sequential estimation of $\mathbf{Q}$ and $\mathbf{R}$. Combining approaches proposed by Tapley *et al.* and Gemson yields an heuristic sequential estimation of $\mathbf{Q}$ and $\mathbf{R}$ covariances. This approach does not guarantee to be optimal or perfect but provide a solution in an efficient way with good estimates.

In addition to the estimation of the noise statistics, the initial state error covariance $\mathbf{P}_0$ also plays a significant role in the EKF implementation, especially when the algorithm is used for dual-estimation purpose [142]. Several data points at the beginning of the measurement data fed to the inverse of the Fisher information matrix [112] yield in the initial state error covariance matrix $\mathbf{P}_0$. The following sections 5.2.1.1 - 5.2.1.3 present the estimation of $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}_0$ in details.

5.2.1.1 Estimation of Measurement Noise Covariance, $\mathbf{R}$

The observation model formulated in Eq. (5.1.3) is affected by the uncertainty in the state variable $(\boldsymbol{x})$ and measurement noise $(\boldsymbol{v})$. The latter variable is assumed to be distributed as Gaussian with mean $\bar{\boldsymbol{v}}$ and covariance $\mathbf{R}$, written as,

$$\boldsymbol{v}_k \sim \mathcal{N}(\bar{\boldsymbol{v}}, \mathbf{R}) \tag{5.2.1}$$

The measurement noise covariance matrix $\mathbf{R}$ is a parameter of interest and derived in this section. Typically, zero mean noise $\bar{\boldsymbol{v}} = 0$ is commonly found in practice [18, 25]. Thus, Eq. (5.2.1) is rewritten as,

$$\boldsymbol{v}_k \sim \mathcal{N}(0, \mathbf{R}) \tag{5.2.2}$$

This assumption is also used in this dissertation. Since the true value of $\boldsymbol{x}$ is not known, $\boldsymbol{v}_k$ cannot be determined. However, variable $\boldsymbol{v}_k$ can be approached through a residual parameter $(\boldsymbol{r})$ which is characterized by the difference between the predicted output $(\tilde{\boldsymbol{y}})$ and the observation $(\boldsymbol{z})$. If the residual at any discrete-time point $(\boldsymbol{r}_k)$ is assumed as a representative of the discrete measurement noise $\boldsymbol{v}_k$, it may also be considered as Gaussian. Taking the assumption of the residual is characterized by the Gaussian distribution, an estimator for the residual can be constructed through its mean and covariance [143]. In other reference, the residual is also termed as innovation [25, 129]. Based on its definition, a residual in every time-step k is formulated as,

$$\boldsymbol{r}_k \equiv \boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - g(\tilde{\boldsymbol{x}}_k, \boldsymbol{u}_k) \tag{5.2.3}$$

where $\boldsymbol{z}_k$ is the observation at time-step k and $\tilde{\boldsymbol{y}}_k$ is the discrete output model as computed by utilizing the predicted state $\tilde{\boldsymbol{x}}_k$. Furthermore, for a given number of samples n_r and by assuming that the residual in each time-step is independent, the residual mean $\bar{\boldsymbol{r}}$ and its covariance $\mathbf{C}_r$ are constant. Thus, the unbiased estimator for $\boldsymbol{r}$ can be formulated as,

$$\bar{\boldsymbol{r}} = \frac{1}{n_r} \sum_{k=1}^{n_r} \boldsymbol{r}_k \tag{5.2.4}$$

and the corresponding covariance matrix $\mathbf{C}_r$ is modeled as,

$$\mathbf{C}_r = \frac{1}{n_r} \sum_{k=1}^{n_r} (\boldsymbol{r}_k - \bar{\boldsymbol{r}})(\boldsymbol{r}_k - \bar{\boldsymbol{r}})^T \tag{5.2.5}$$

However, $\mathbf{C}_r$ in Eq. (5.2.5) represents the covariance of the sample data and not the population data since the true mean $\bar{\boldsymbol{r}}$ (population mean) is unknown. In order to infer the covariance of the population data through the sample data, Bessel's correction factor is introduced into Eq. (5.2.5). Such a correction factor basically rectifies bias of the sample data's variance [144]. Eq. (5.2.5) with the correction factor is rewritten as,

$$\mathbf{C}_r = \frac{1}{n_r - 1} \sum_{k=1}^{n_r} (\boldsymbol{r}_k - \bar{\boldsymbol{r}})(\boldsymbol{r}_k - \bar{\boldsymbol{r}})^T \tag{5.2.6}$$

In order to construct the measurement noise covariance $\mathbf{R}$, the expected value of Eq. (5.2.6) is computed as,

$$\begin{aligned} E\left[\mathbf{C}_r\right] &= E\left[\frac{1}{n_r - 1} \sum_{k=1}^{n_r} (\boldsymbol{r}_k - \bar{\boldsymbol{r}})(\boldsymbol{r}_k - \bar{\boldsymbol{r}})^T\right] \\ &= E\left[\frac{1}{n_r - 1} \sum_{k=1}^{n_r} (\boldsymbol{r}_k \boldsymbol{r}_k^T - \boldsymbol{r}_k \bar{\boldsymbol{r}}^T - \bar{\boldsymbol{r}} \boldsymbol{r}_k^T + \bar{\boldsymbol{r}} \bar{\boldsymbol{r}}^T\right] \\ &= \frac{1}{n_r - 1} \sum_{k=1}^{n_r} E\left[\boldsymbol{r}_k \boldsymbol{r}_k^T\right] - E\left[\boldsymbol{r}_k \bar{\boldsymbol{r}}^T\right] - E\left[\bar{\boldsymbol{r}} \boldsymbol{r}_k^T\right] + E\left[\bar{\boldsymbol{r}} \bar{\boldsymbol{r}}^T\right] \end{aligned} \tag{5.2.7}$$

Eq. (5.2.7) assumes a significant error in the model which is indicated by the non-zeros residual mean ($\bar{\boldsymbol{r}} \neq 0$). However, in the present case, such an assumption might be ignored as the $\mathbf{R}$ estimator is a part of the modified Filter Error Method. In this method, the estimation of the system error/bias due to model deficiency is addressed separately by the algorithm. Thus, in order to avoid interference with the modified Filter Error algorithm, the residual mean can be assumed zero, $\bar{\boldsymbol{r}} = 0$. Furthermore, the flight physical model is assumed to be accurate and unbiased which allows the zero-mean residual assumption. Thus, by taking the expected value of $E\left[\bar{\boldsymbol{r}} \bar{\boldsymbol{r}}^T\right] = 0$, and $E\left[\bar{\boldsymbol{r}}_k^T \boldsymbol{r}_k\right] = E\left[\boldsymbol{r}_k \bar{\boldsymbol{r}}^T\right] = 0$, Eq. (5.2.7) is simplified to,

$$E\left[\mathbf{C}_r\right] = \frac{1}{n_r} \sum_{k=1}^{n_r} E\left[\boldsymbol{r}_k \boldsymbol{r}_k^T\right] \tag{5.2.8}$$

Bessel's correction factor $(n_r - 1)$ is now replaced by n_r since the residual mean $\bar{\boldsymbol{r}} = 0$ represents the 'true' mean (population mean). Furthermore, term $E\left[\boldsymbol{r}_k\boldsymbol{r}_k^T\right]$ in Eq. (5.2.7) is elaborated by plugging Eq. (5.2.3) and introducing the linear form of the output model as $\tilde{\boldsymbol{y}} = \mathbf{C}_k\tilde{\boldsymbol{x}}_k$ yields in,

$$\begin{aligned}
E\left[\boldsymbol{r}_k\boldsymbol{r}_k^T\right] &= E\left[(\boldsymbol{z}_k - \mathbf{C}_k\tilde{\boldsymbol{x}}_k)(\boldsymbol{z}_k - \mathbf{C}_k\tilde{\boldsymbol{x}}_k)^T\right] \\
&= E\left[(\mathbf{C}_k\boldsymbol{x}_k + \boldsymbol{v}_k - \mathbf{C}_k\tilde{\boldsymbol{x}}_k)(\mathbf{C}_k\boldsymbol{x}_k + \boldsymbol{v}_k - \mathbf{C}_k\tilde{\boldsymbol{x}}_k)^T\right] \\
&= E\left[(\mathbf{C}_k(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}) + \boldsymbol{v}_k)(\mathbf{C}_k(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k) + \boldsymbol{v}_k)^T\right] \\
&= E\left[\mathbf{C}_k(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)^T\mathbf{C}_k^T\right] + E\left[\boldsymbol{v}_k\boldsymbol{v}_k^T\right] \\
&= \mathbf{C}_k\tilde{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k
\end{aligned} \tag{5.2.9}$$

Component $E\left[\boldsymbol{x}_k - \tilde{\boldsymbol{x}})(\boldsymbol{x}_k - \tilde{\boldsymbol{x}})^T\right]$ is the predicted state error covariance $\tilde{\mathbf{P}}_k$ which is computed in the Kalman filter. As shown in Eq. (5.2.9), the residual is constructed by the measurement noise $\mathbf{R}_k$ and uncertainty due to the model deficiency. The latter is indicated by term $\mathbf{C}_k\tilde{\mathbf{P}}_k\mathbf{C}_k^T$. This fact supports the statement made at the beginning of this section that the observation noise can be approximated through its corresponding residual. Since $\mathbf{R}$ is the parameter of interest, Eq. (5.2.9) is further simplified by solving for $\mathbf{R}_k$ yields in,

$$\mathbf{R}_k = E\left[\mathbf{C}_r\right] - \mathbf{C}_k\tilde{\mathbf{P}}_k\mathbf{C}_k^T \tag{5.2.10}$$

By plugging Eq. (5.2.5) into Eq. (5.2.10) and solving $\mathbf{R}$ for the whole sample data (n_r), the unbiased estimator for $\mathbf{R}$ is obtained as follows [133],

$$\hat{\mathbf{R}} = \frac{1}{n_r - 1}\sum_{k=1}^{n_r}\left[(\boldsymbol{r}_k - \hat{\boldsymbol{r}})(\boldsymbol{r}_k - \hat{\boldsymbol{r}}) - \frac{n_r - 1}{n_r}\mathbf{C}_k\tilde{\mathbf{P}}_k\mathbf{C}_k^T\right] \tag{5.2.11}$$

In Eq. (5.2.11), Bessel's correction factor is used again as the residual mean is represented by the residual mean $(\hat{\boldsymbol{r}})$ of the sample data.

In the original formulation of $\hat{\mathbf{R}}$, as presented in Eq. (5.2.11), Tapley and Myers utilize the residual which results from the predicted output $\tilde{\boldsymbol{y}}_k$. This predicted output is computed based on the predicted state $\tilde{\boldsymbol{x}}_k$ and its corresponding predicted state error covariance $\tilde{\mathbf{P}}_k$. In the context of filtering and smoothing, the residual can also be estimated based on the corrected output $\hat{\boldsymbol{y}}$ and smoothed output $\boldsymbol{y}^S$. Thus, three options are available for computing residual that later can be used for inferring the measurement noise covariance $\hat{\mathbf{R}}$. These three options are listed here for clarity [129, 145]:

1. By using the predicted output $\tilde{\boldsymbol{y}}_k$. This leads to $\boldsymbol{r}_k = \boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k$ and $\mathbf{P}_k = \tilde{\mathbf{P}}_k$
2. By using the corrected output $\hat{\boldsymbol{y}}_k$. This leads to $\boldsymbol{r}_k = \boldsymbol{z}_k - \hat{\boldsymbol{y}}_k$ and $\mathbf{P}_k = \hat{\mathbf{P}}_k$
3. By using the smoothed output $\boldsymbol{y}^S$. This leads to $\boldsymbol{r}_k = \boldsymbol{z}_k - \boldsymbol{y}^S$ and $\mathbf{P}_k = \mathbf{P}_k^S$

Among the three options for computing the residual, Ananthasayanam [129] suggests using the smoothed output as it provides the best statistics for estimating the matrix covariance $\mathbf{R}$.

This dissertation adopts this recommendation. Furthermore, Eq. (5.2.11) is modified slightly by computing $\boldsymbol{r}_k$ through the smoothed output $\boldsymbol{y}^S$ and replacing $\tilde{\mathbf{P}}_k$ with $\mathbf{P}_k^S$. One of the smoothing algorithms that can be used is the Rauch-Tung-Striebel (RTS) smoother algorithm [9]. This algorithm is briefly discussed in Section 5.2.2.

5.2.1.2 Estimation of Process Noise Covariance, $\mathbf{Q}$

Estimation of $\mathbf{Q}$ takes a similar approach as in the case of the measurement noise covariance $\mathbf{R}$. Again, a residual parameter is used to derive the expression for the process noise covariance. A limited amount of data is taken as sample in which the residual mean and its corresponding covariance are assumed constant. In the present case, a residual is defined as a difference between the updated and the predicted state, written as,

$$\boldsymbol{q}_k = \hat{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k \tag{5.2.12}$$

The residual mean and its corresponding covariance can be computed as in,

$$\bar{\boldsymbol{q}}_k = \frac{1}{n_q}\sum_{k=1}^{n_q} \boldsymbol{q}_k \tag{5.2.13}$$

$$\mathbf{C}_q = \frac{1}{n_q - 1}\sum_{k=1}^{n_q} (\boldsymbol{q}_k - \bar{\boldsymbol{q}})(\boldsymbol{q}_k - \bar{\boldsymbol{q}}) \tag{5.2.14}$$

where n_q denotes the number of the sample data. Estimates of $\mathbf{Q}$ is derived by taking the expected value of the residual covariance $\mathbf{C}_q$. In this case, the true mean of the residual is assumed to be known and set as zero. This assumption leads the residual to behave like the process noise which is assumed to be distributed as Gaussian with zero mean and covariance $\mathbf{Q}$. Thus, by applying the expected operator $E[\,]$, Eq. (5.2.14) is further simplified into,

$$\begin{aligned} E\left[\mathbf{C}_q\right] &= E\left[\frac{1}{n_q}\sum_{k=1}^{n_q} (\boldsymbol{q}_k - 0)(\boldsymbol{q}_k - 0)^T\right] \\ &= \frac{1}{n_q}\sum_{k=1}^{n_q} E\left[\boldsymbol{q}_k \boldsymbol{q}_k^T\right] \end{aligned} \tag{5.2.15}$$

Term $E\left[\boldsymbol{q}_k \boldsymbol{q}_k^T\right]$ in Eq. (5.2.15) can be solved through the predicted state error covariance, $\tilde{\mathbf{P}}$. By definition, this covariance matrix is computed through the expected value of the error between true state ($\boldsymbol{x}_k$) and the predicted state ($\tilde{\boldsymbol{x}}_k$). In the Kalman filter algorithm, variable $\tilde{\mathbf{P}}_k$ takes form as,

$$\tilde{\mathbf{P}}_k = E\left[(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)^T\right] \tag{5.2.16}$$

$$= \mathbf{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{\Phi}_{k-1} + \mathbf{Q}_k \tag{5.2.17}$$

Eq. (5.2.17) is further elaborated by including term $\boldsymbol{q}_k$ into the equation. This process is carried out by rearranging Eq. (5.2.12) as $\tilde{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{q}_k$ and plugging it back into Eq. (5.2.17) yields in,

$$\begin{aligned} E\left[(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k)\right] &= E\left[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k + \boldsymbol{q}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k + \boldsymbol{q}_k)\right] \\ &= E\left[(\boldsymbol{q}_k - (\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))(\boldsymbol{q}_k - (\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k))\right] \end{aligned} \tag{5.2.18}$$

Furthermore, term $\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k = \hat{\boldsymbol{e}}_k$ represents the error between the updated state and the true state. Substituting this variable into Eq. (5.2.18) and applying the E operator on each term results in,

$$E\left[(\boldsymbol{q}_k - \hat{\boldsymbol{e}}_k)(\boldsymbol{q}_k - \hat{\boldsymbol{e}}_k)^T\right] = E\left[\boldsymbol{q}_k\boldsymbol{q}_k^T\right] + E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - E\left[\boldsymbol{q}_k\hat{\boldsymbol{e}}_k^T\right] - E\left[\hat{\boldsymbol{e}}_k\boldsymbol{q}_k^T\right] \tag{5.2.19}$$

Replacing the left term in Eq. (5.2.19) with the right term of Eq. (5.2.17) and solving for $E\left[\boldsymbol{q}_k\boldsymbol{q}_k^T\right]$ yields in,

$$E\left[\boldsymbol{q}_k\boldsymbol{q}_k^T\right] = \mathbf{Q}_k - E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] + E\left[\boldsymbol{q}_k\hat{\boldsymbol{e}}_k^T\right] + E\left[\hat{\boldsymbol{e}}_k\boldsymbol{q}_k^T\right] + \boldsymbol{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\Phi}_{\mathbf{k-1}} \tag{5.2.20}$$

By definition, term $E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right]$ represents the updated state error covariance ($\hat{\mathbf{P}}_k$) that is computed recursively in the Kalman filter algorithm. Thus, the remaining term that needs to be solved in order to get the expression for $\mathbf{Q}_k$ is $E\left[\boldsymbol{q}_k\boldsymbol{e}_k^T\right]$ or $E\left[\hat{\boldsymbol{e}}_k\boldsymbol{q}_k^T\right]$. These two terms can be derived by plugging Eq. (5.2.12) and expression $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k + \hat{\boldsymbol{e}}_k$ in $E\left[\boldsymbol{q}_k\hat{\boldsymbol{e}}_k^T\right]$ as,

$$\begin{aligned} E\left[\boldsymbol{q}_k\hat{\boldsymbol{e}}_k^T\right] &= E\left[(\hat{\boldsymbol{x}}_k - \tilde{\boldsymbol{x}}_k)\hat{\boldsymbol{e}}_k^T\right] \\ &= E\left[(\boldsymbol{x}_k + \hat{\boldsymbol{e}}_k - \tilde{\boldsymbol{x}}_k)\hat{\boldsymbol{e}}_k^T\right] \\ &= E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - E\left[(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)\hat{\boldsymbol{e}}_k^T\right] \end{aligned} \tag{5.2.21}$$

Furthermore, term $\hat{\boldsymbol{e}}_k$ in Eq. (5.2.21) is expressed in terms of $\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k$ to make the calculation of $E\left[\boldsymbol{q}_k\hat{\boldsymbol{e}}_k^T\right]$ possible. To get a such expression of $\hat{\boldsymbol{e}}_k$, the Kalman filter updated step is used.

$$\hat{\boldsymbol{x}}_k = \tilde{\boldsymbol{x}}_k + \mathbf{K}_k(\boldsymbol{z}_k - \mathbf{C}_k\tilde{\boldsymbol{x}}_k) \tag{5.2.22}$$

The observation variable $\boldsymbol{z}_k$ is replaced with the observation model $\mathbf{C}_k\boldsymbol{x}_k + \boldsymbol{v}_k$ and plugged in back into Eq. (5.2.22) yields in,

$$\hat{\boldsymbol{x}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\tilde{\boldsymbol{x}}_k + \mathbf{K}_k(\mathbf{C}_k\boldsymbol{x}_k + \boldsymbol{v}_k) \tag{5.2.23}$$

$$= (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k) + \mathbf{K}_k\boldsymbol{v}_k + \boldsymbol{x}_k \tag{5.2.24}$$

By rearranging $\boldsymbol{x}_k$ to the left side of Eq (5.2.23), yields the expression for $\hat{\boldsymbol{e}}$ as follows,

$$\hat{\boldsymbol{e}}_k = \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k) + \mathbf{K}_k\boldsymbol{v}_k \tag{5.2.25}$$

Eq. (5.2.25) is plugged in back into Eq. (5.2.21) yields in,

$$\begin{aligned} E\left[\boldsymbol{q}_k\boldsymbol{e}_k^T\right] &= E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - E\left[(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)\left((\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)^T + \boldsymbol{v}_k\mathbf{K}_k^T\right)\right] \\ &= E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - E\left(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)^T\right] \\ &\quad - E\left[(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)\boldsymbol{v}_k^T\mathbf{K}_k^T\right] \end{aligned} \quad (5.2.26)$$

By taking assumption $E\left[\boldsymbol{v}_k\right] = 0$ into account, it leads to $E\left[(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)\boldsymbol{v}_k^T\mathbf{K}_k^T\right] = 0$. Furthermore, by replacing $E\left[(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)(\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k)^T\right]$ with $\tilde{\mathbf{P}}_k$, the final expression for $E\left[\hat{\boldsymbol{e}}_k\boldsymbol{q}_k^T\right]$ is obtained as,

$$E\left[\hat{\boldsymbol{e}}_k\boldsymbol{q}_k^T\right] = E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - \tilde{\mathbf{P}}_k(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)^T \quad (5.2.27)$$

The same procedures can be done to derive the expression for $E\left[\boldsymbol{e}_k\boldsymbol{q}_k^T\right]$,

$$E\left[\boldsymbol{e}_k\boldsymbol{q}_k^T\right] = E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k^T\right] - (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\tilde{\mathbf{P}}_k^T \quad (5.2.28)$$

By substituting Eqs. (5.2.27), (5.2.28), $E\left[\hat{\boldsymbol{e}}_k\hat{\boldsymbol{e}}_k\right] = \hat{\mathbf{P}}_k$ into Eq. (5.2.20) and solving for $\mathbf{Q}_k$ gives,

$$\mathbf{Q}_k = E\left[\boldsymbol{q}_k\boldsymbol{q}_k^T\right] - \hat{\mathbf{P}}_k + \tilde{\mathbf{P}}_k(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)^T + (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\tilde{\mathbf{P}}_k^T - \boldsymbol{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\Phi}_{\mathbf{k-1}} \quad (5.2.29)$$

For the whole data points of the sample data (n_q), the estimate for $\mathbf{Q}$ is assumed constant. The expression for this estimate is derived based on Eq. (5.2.29) in which the $E\left[\boldsymbol{q}_k\boldsymbol{q}_k^T\right]$ term is replaced by Eq. (5.2.15) yields an unbiased estimator for the process noise covariance [136],

$$\begin{aligned} \hat{\mathbf{Q}} = \frac{1}{n_q - 1}\sum_{k=1}^{n_q}\Bigg[&(\boldsymbol{q}_k - \bar{\boldsymbol{q}})(\boldsymbol{q}_k - \bar{\boldsymbol{q}})^T \\ &- \frac{n_q - 1}{n_q}\left(\hat{\mathbf{P}}_k - \tilde{\mathbf{P}}_k(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)^T - (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\tilde{\mathbf{P}}_k^T + \boldsymbol{\Phi}_{k-1}\hat{\mathbf{P}}_{k-1}\boldsymbol{\Phi}_{k-1}^T\right)\Bigg] \end{aligned} \quad (5.2.30)$$

The computational of $\mathbf{Q}$ is very straightforward since all involved parameters are provided by the Kalman filter algorithm. The only variables that need to be computed are $\boldsymbol{q}_k$ and $\bar{\boldsymbol{q}}$. However, these two variables can be computed with a minimum effort since $\tilde{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{x}}_k$, which are required by the two variables, are also provided by the Kalman filter algorithm.

5.2.1.3 Estimation of Initial State Error Covariance, $\mathbf{P}_0$

In the filtering approach-based parameter estimation problem, estimation of the initial state error covariance ($\mathbf{P}_0$) is equally important as the measurement noise covariance ($\mathbf{R}$) and process noise covariance ($\mathbf{Q}$) [129, 142]. In general, the state error covariance represents the uncertainty of the estimated state which is recursively computed in the Kalman filter algorithm. However, the Kalman filter algorithm requires the initialization of this covariance matrix in order to start the computation. In the absence of the state error covariance, which is a typical case in the parameter estimation problem, the Kalman filter algorithm recommends forming such a parameter as a diagonal matrix whose values are large enough to ensure the estimate converges quickly and thus, the influence of the initial guess soon will be negligible. This fact is supported by the natural property of the Kalman filter where the predicted state is updated with the measurement. A large value of the state error covariance leads to a high corresponding gain which indicates that the measurement is trusted more than the current predicted state and vice versa. Thus, by setting a large value, the state error covariance matrix will make the Kalman filter to force the updated state following the measurement quickly.

However, for the present case where the algorithm is applied to measurements with low sampling rates, choosing a high value for the initial state error covariance is not considered desirable. Furthermore, as the algorithm is adopted based on Tapley and Myers, it requires that the initial state error covariance matrix $(\mathbf{P}_0)$ should be selected so that the state converges quickly to its true value. Motivated by these facts, the estimation of $\mathbf{P}_0$ is necessary for the present case. Gemson *et al.* [142] proposes a systematic approach to compute the state error covariance matrix. The form of the initial state error covariance proposed by Gemson *et al.* is adopted in this dissertation. The estimation of this matrix is basically based on the assumption that good measurements from sensors can be used as initial state values x_0. In an ideal case, where the states are measured the usual recommendation for $\mathbf{P}_0$ is by setting $\mathbf{P}_0 = \mathbf{R}$ [139]. Motivated by this fact, it seems intuitive to infer the initial state error covariance matrix $\mathbf{P}_0$ from the best available estimate of the measurement covariance matrix $\mathbf{R}$. Based on [112, 142], $\mathbf{P}_0$ can be computed through the accumulation of the information matrices for several data points and inverted at the end of data point. This variable takes form as [142],

$$\hat{\mathbf{P}}_0 = \left[\frac{1}{n_p}\sum_{k=1}^{n_p} \mathcal{I}_k\right]^{-1} \tag{5.2.31}$$

where $\mathcal{I}$ is the information matrix and defined as,

$$\mathcal{I}_k = \mathbf{\Phi}_k^T \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \mathbf{\Phi}_k \tag{5.2.32}$$

While the remaining variables are parts of the Kalman filter algorithm and computed within the algorithm, the computation of $\mathbf{P}_0$ is initialized by providing its initial value, and along with measurement data, the information matrix $\mathcal{I}$ is computed for n_p data points. The state error covariance matrix is obtained by inverting the information matrix at the end of n_p. Typical value for n_p around 15-30 provides a good estimate of $\mathbf{P}_0$. For practical purpose, Eq. (5.2.31) is only used for estimating the state variables variance (diagonal element of $\mathbf{P}_0$). Thus, the utilization of matrix $\mathbf{R}$ only involves the diagonal elements of this matrix to ensure the accuracy of the matrix inversion.

5.2.1.4 Heuristic Adaptive Estimation of $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$

Kalman filter algorithm requires $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ to be specified in order to run the algorithm. However, as already presented in the previous section, the estimators for $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ have been established in which the form that each estimator takes, is suitable with the Kalman filter algorithm. Thus, embedding those estimators in the Kalman filter will provide an algorithm that is suitable for the present case, where the noise statistics are unknown. Gemson *et al.* [142] propose an iterative scheme where the proposed estimators for $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ are embedded in the Extended Kalman Filter algorithm. To check the convergence during the iterative process, the cost function is introduced which basically aims at minimizing the residual/innovation based on the maximum likelihood criterion. For residuals that are distributed as Gaussian, the cost function takes form as [142],

$$J = \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{z}_k - \tilde{\boldsymbol{y}}_k)\left(\mathbf{C}_k\tilde{\mathbf{P}}_k\mathbf{C}_k + \mathbf{R}^{-1}\right)^{-1}(\boldsymbol{z}_k - \tilde{\mathbf{y}}_k)^T \tag{5.2.33}$$

The tuning for $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ are visually depicted in Figure 5.3 which is further elaborated as follows:

① The algorithm is started by providing initial guesses for $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$. The value for $\mathbf{P}_0$ might be chosen as a unity matrix while $\mathbf{Q}$ might be set to zero. The initial value for $\mathbf{R}$ is set as a diagonal matrix with variances of the measurement noise represented by the diagonal term(s).

② Make scouting first pass with the initial $\mathbf{P}_0$ and obtain $\mathbf{P}_0$ from the inverse of the information matrix, see Eq. (5.2.31)

③ Using the estimated $\mathbf{P}_0$, iterate over $\mathbf{Q}$ using Mayers and Tapley's algorithm embedded in the Extended Kalman Filter and followed by smoothing process through the RTS smoother algorithm. The iteration is terminated once the relative absolute difference of the cost function in the two consecutive iterations, $\Delta J < 10^{-4}$, is achieved. The employed cost function is defined in Eq. (5.2.33).

④ With the convergence of $\mathbf{Q}$, the covariance matrix $\mathbf{R}$ is estimated in the same way as $\mathbf{Q}$. When $\mathbf{R}$ has not yet converged, the iteration is repeated from the second step by setting $\mathbf{Q} = 0$, $\mathbf{P}_0 = \mathbf{P}_G$, subscript G means given. Once the cost function constraint achieved, the iteration is terminated, and the estimates for $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ are obtained which will be used later in the parameter estimation step. In this loop, the convergence criteria for $\mathbf{R}$ is achieved when the relative change of `det`$(\mathbf{R})$ in the two consecutive loops is less than **1E-4**. Operator `det` is an operator for computing determinant of a matrix.

During the estimation of the noise statistics using the above procedures, system parameters (β) are kept constant and only be updated in the parameter estimation step. The parameter estimation step is elaborated in the following section.

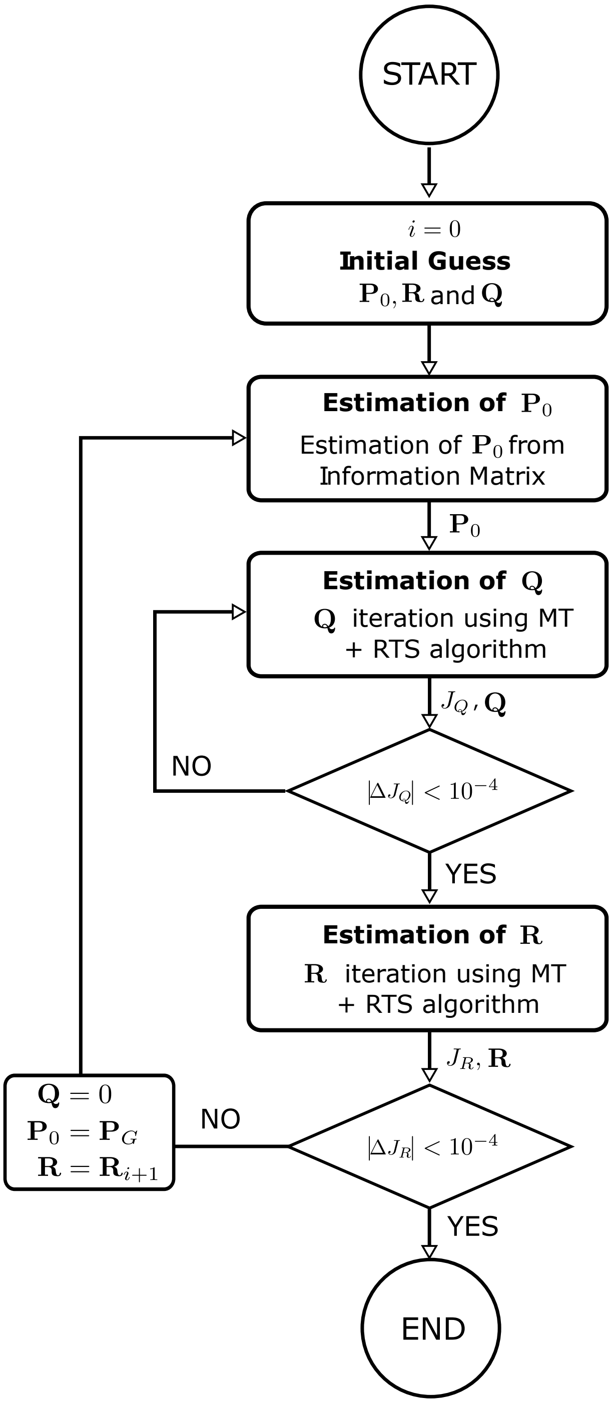

START
$i = 0$
Initial Guess
$\mathbf{P}_0, \mathbf{R}$ and $\mathbf{Q}$
Estimation of $\mathbf{P}_0$
Estimation of $\mathbf{P}_0$ from Information Matrix
$\mathbf{P}_0$
Estimation of $\mathbf{Q}$
$\mathbf{Q}$ iteration using MT + RTS algorithm
$J_Q, \mathbf{Q}$
$|\Delta J_Q| < 10^{-4}$
NO
YES
Estimation of $\mathbf{R}$
$\mathbf{R}$ iteration using MT + RTS algorithm
$J_R, \mathbf{R}$
$|\Delta J_R| < 10^{-4}$
NO
$\mathbf{Q} = 0$
$\mathbf{P}_0 = \mathbf{P}_G$
$\mathbf{R} = \mathbf{R}_{i+1}$
YES
END

Figure 5.3: Flow chart of $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{P}_0$ tuning

5.2.2 Parameter Estimation

The Filter Error Method utilized in this section is modified to adapt to the estimation of the noise statistics and initial state error covariance provided in the previous section. In the original formulation of the Filter Error Method, the process noise covariance matrix $\mathbf{Q}$ is estimated along with the system's parameters. However, as this noise covariance along with $\mathbf{R}$ and $\mathbf{P}_0$ have been estimated in a separate step, the modified Filter Error Method is dedicated only for the system's parameter estimation. With fewer unknown parameters to be estimated, the modified Filter Error Method provides a stable computation. Furthermore, the state estimation in FEM using the Extended Kalman Filter is replaced by the Rauch-Tung-Striebel (RTS) smoother that provides lower state error covariance [9].

From the three types of smoothing techniques, i.e., fixed-lag, fixed-point, and fixed-interval smoothing, the RTS smoother belongs to the latter type. This technique is commonly used in the aircraft state estimation because the state estimation is performed in a fixed window-time corresponding to a specific flight maneuver. In principle, the RTS smoother is a two-pass procedure in which the Kalman filter algorithm is running forward from $k = 1, 2, \cdots N$ while at the second pass, the smoothing algorithm is running backward from $k = N, N-1, \cdots 1$. Thus, the smoothed estimate is a weighted linear combination of the Kalman filter forward and backward smoothing [18]. Visually, the RTS smoother is depicted in Figure 5.4, During forward pass, the Kalman filter computational procedures are applied to the data, and the corresponding variables such as $\hat{\boldsymbol{x}}, \tilde{\boldsymbol{x}}, \hat{\mathbf{P}}, \tilde{\mathbf{P}}$, and $\boldsymbol{\Phi}$ are stored temporarily to be used later in the second pass. In the second pass, three variables are computed, namely the smoothed gain $(\mathbf{K}^S)$, the smoothed state error covariance $(\mathbf{P}^S)$, and the smoothed state $(\boldsymbol{x}^S)$. These three variables are computed through Eqs. (5.2.34) - (5.2.36) [146].

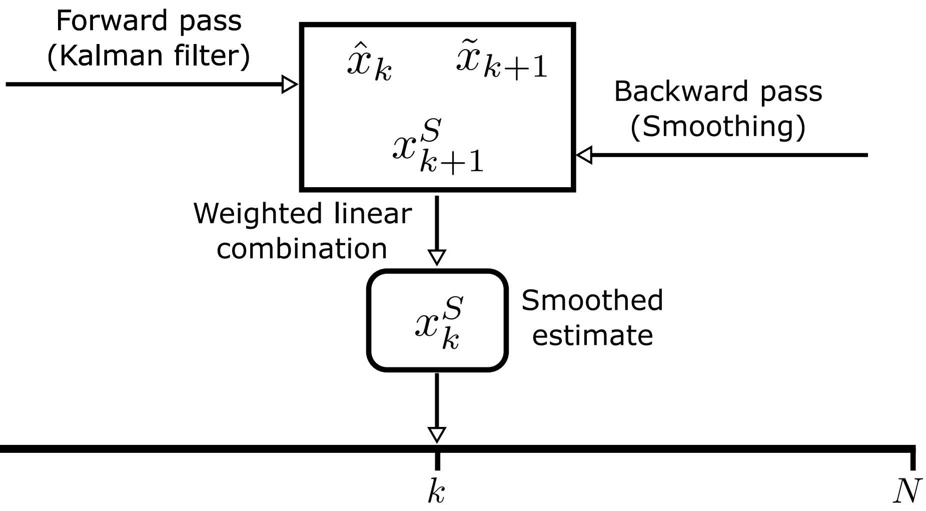


Figure 5.4: RTS smoother as a two-pass procedure [18]

$$\mathbf{K}^S_k = \hat{\mathbf{P}}_k \boldsymbol{\Phi}^T_{k+1} \tilde{\mathbf{P}}^{-1}_{k+1} \tag{5.2.34}$$

$$\boldsymbol{x}^S_k = \hat{\boldsymbol{x}}_{a_k} + \mathbf{K}^S_k \left[\boldsymbol{x}^S_{k+1} - \tilde{\boldsymbol{x}}_{k+1}\right] \tag{5.2.35}$$

$$\mathbf{P}^S_k = \hat{\mathbf{P}}_k + \mathbf{K}^S_k \left[\mathbf{P}^S_{k+1} - \tilde{\mathbf{P}}_{k+1}\right] \left(\mathbf{K}^S_k\right)^T \tag{5.2.36}$$

where $k = N, N-1, \cdots 1 \ldots$ as the computation is done backward. Furthermore, the smoothed estimates are used for parameter estimation through the maximum likelihood principle. Since the noise statistics $\mathbf{R}, \mathbf{Q}$ have been estimated, the steps in the parameter estimation stage are reduced by excluding the estimation of $\mathbf{R}$ covariance. The unknown parameters are updated through Eqs. (5.1.17) - (5.1.19). Figure 5.5 shows the schematic diagram of the parameter estimation stage.

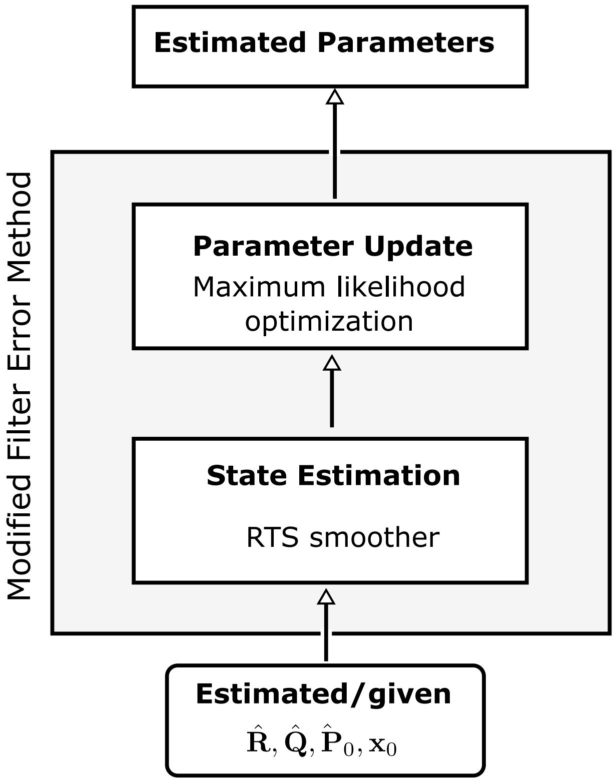

Estimated Parameters
Modified Filter Error Method
Parameter Update
Maximum likelihood optimization
State Estimation
RTS smoother
Estimated/given
$\hat{\mathbf{R}}, \hat{\mathbf{Q}}, \hat{\mathbf{P}}_0, \mathbf{x}_0$

Figure 5.5: Parameter estimation: maximum likelihood + RTS smoother

5.2.3 Combined Noise Statistics and Parameter Estimation

Section 5.2.1 and 5.2.2 have presented the estimation of noise statistics and the system's parameters respectively. In the following section, both steps are combined for a complete parameter estimation method that is capable of dealing with FDM data. In the first stage, noise statistics are estimated through procedures described in section 5.2.1. In this stage, an iterative process is carried out to estimate the process noise $\mathbf{Q}$ and the measurement noise $\mathbf{R}$ covariances. Along with that, the initial state error covariance ($\mathbf{P}_0$) is also estimated through Eq. (5.2.31). The iterative process is terminated when the constraint for the cost function defined in (5.2.33) is achieved. The estimated noise statistics are passed to the second stage for parameter estimation. In this second stage, the RTS smoother is used for state estimation by utilizing the noise statistics and the initial state error covariance estimated in the previous step. The parameters are estimated through the maximum likelihood principle. Each parameter is updated by using the Gauss-Newton algorithm as defined in Eqs. (5.1.17) - (5.1.19). At the final stage, the relative change of each parameters is checked, if the relative change is less than **1E-4**, the loop is terminated, and the corresponding estimated parameters are obtained. However, if the relative change of the parameters is not achieved, the process is started again from the first step where the noise statistics and state error covariance are estimated. The following Fig. 5.6 shows the schematic diagram of the parameter estimation technique proposed in this work.

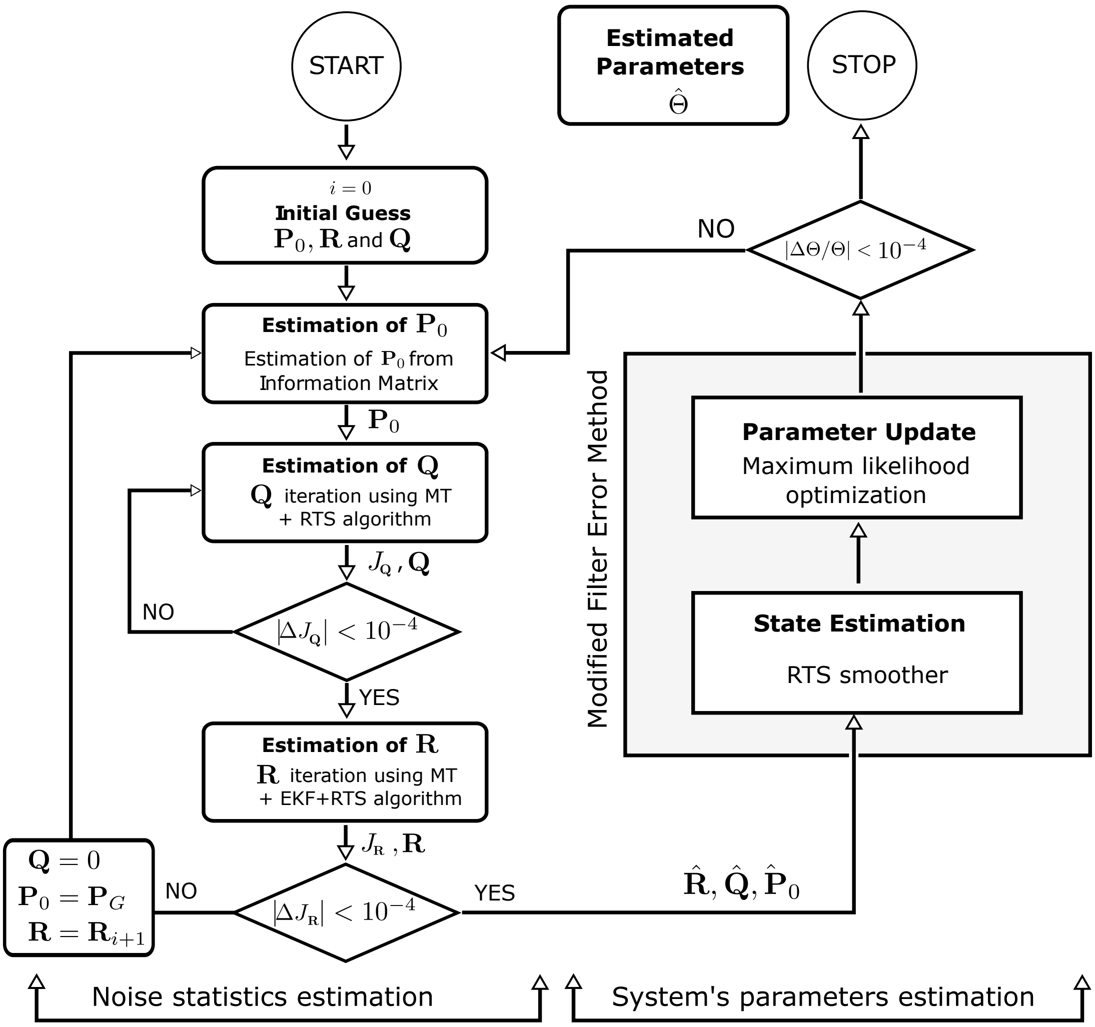

START
Estimated Parameters
$\hat{\Theta}$
STOP
$i = 0$
Initial Guess
$\mathbf{P}_0, \mathbf{R}$ and $\mathbf{Q}$
NO
$|\Delta\Theta/\Theta| < 10^{-4}$
Estimation of $\mathbf{P}_0$
Estimation of $\mathbf{P}_0$ from Information Matrix
$\mathbf{P}_0$
Modified Filter Error Method
Parameter Update
Maximum likelihood optimization
Estimation of $\mathbf{Q}$
$\mathbf{Q}$ iteration using MT + RTS algorithm
J_Q, $\mathbf{Q}$
NO
$|\Delta J_Q| < 10^{-4}$
YES
State Estimation
RTS smoother
Estimation of $\mathbf{R}$
$\mathbf{R}$ iteration using MT + EKF+RTS algorithm
J_R, $\mathbf{R}$
$\mathbf{Q} = 0$
$\mathbf{P}_0 = \mathbf{P}_G$
$\mathbf{R} = \mathbf{R}_{i+1}$
NO
$|\Delta J_R| < 10^{-4}$
YES
$\hat{\mathbf{R}}, \hat{\mathbf{Q}}, \hat{\mathbf{P}}_0$
Noise statistics estimation
System's parameters estimation

Figure 5.6: Schematic diagram of the developed parameter estimation technique

Compared to the Filter Error Method, the developed algorithm provides a shorter computational step, specifically during parameter estimation step. In the developed algorithm, computation of the perturbed state gain matrix ($\mathbf{K}^p$), constraints checking ($\mathbf{KC} < 1$) and compensating $\mathbf{F}$ for new $\mathbf{R}$ are discarded from the steps. The latter is discarded from the developed algorithm since the noise statistics are estimated in the separated loop.

5.3 Implementation Aspects

When applying the developed algorithm either for simulated data or real flight data, the following simplifications are required by the algorithm:

1. Measurement noise covariance matrix $\mathbf{R}$ that results from Eq. (5.2.11) contains the correlation coefficient in the off-diagonal components. However, as the assumption that the noise is uncorrelated, the off-diagonal of matrix $\mathbf{R}$ is set to zero every time the

matrix updated. Thus, the covariance matrix $\mathbf{R}$ is reformatted through,

$$\mathbf{R} = \texttt{diag}(\texttt{diag}(\mathbf{R}_{\text{computed}})) \tag{5.3.1}$$

where $\mathbf{R}_{\text{computed}}$ is computed through Eq. (5.2.11) and `diag` is a Matlab's function to take the diagonal element of a matrix or to create a diagonal matrix [147].

2. The same case as the covariance matrix ($\mathbf{R}$), only diagonal elements of the process noise matrix ($\mathbf{Q}$) are involved in the computation. Thus, the same operation is applied on matrix $\mathbf{Q}$. This operation will also improve the stability of the algorithm as in many cases, the covariance matrix computed through Eq. (5.2.30) has a non-positive definite correlation components.
3. The same operation is also applied on $\mathbf{P}_0$, i.e. only the diagonal elements of the computed covariance matrix are used as $\mathbf{P}_0$. In some cases, the elements of matrix $\mathbf{P}_0$ tend toward zero. To remedy this behavior, it is required to check at the end of the computation of matrix $\mathbf{P}_0$; if the covariance elements are close to zero, each element is scaled up by multiplying it with n_p, where n_p is the number of data points passed in to the algorithm for computing $\mathbf{P}_0$.

5.4 Method Validation

In the following section, the developed algorithm is applied to a simple mass-damper system. In this model, the true values of the parameters are known and all noise statistics are specified. With these all known variables, the 'measurement' data are generated and used later to estimate the parameters involved in the systems. The mass-damper system takes the following state model [129],

$$\dot{x}_1(t) = x_2(t) \tag{5.4.1}$$

$$\dot{x}_2(t) = -a \cdot x_1(t) - b \cdot x_2(t) - c \cdot x_1^3(t) \tag{5.4.2}$$

Where $\Theta = [a,\, b,\, c]^T$ denotes the unknown parameters and x_1, x_2 consecutively represent the displacement and velocity. The dot operator ($\dot{x}$) denotes the differentiation of the system state with respect to time. The measurement models of the above system states are represented by discrete models, given as,

$$z_{1,k} = x_{1,k} + v_{1,k} \tag{5.4.3}$$

$$z_{2,k} = x_{2,k} + v_{2,k} \tag{5.4.4}$$

The system states are simulated by specifying the unknown parameters $a = 3.75, b = 0.47, c = 0.75$. Variable c is a weak parameter which does not significantly affect the system. In the context of system identification field, particularly in parameter estimation step, a weak parameter is usually estimated with low accuracy. This is derived by the factor that any input given to the system, the weak parameters will not respond or change the system behavior. This factor consequently leads to the low information content in the observed data corresponding to the weak parameter. This fact will be also demonstrated through the example presented in this section.

The process noises are assumed as a Gaussian distribution with zero-mean and variances as specified by $\mathbf{Q}$ = [1.563E-04, 1.225E-03], where the first and second components of the $\mathbf{Q}$ covariance denote the variance of the process noise of x_1 and x_2 respectively. The process noise is injected as additive noise to the system states. The same as the process noise, the measurement noise is also assumed as a Gaussian distribution with zero-mean and corresponding variances for each output are specified as $\mathbf{R}$ = [0.0100, 0.0081]. The measurement noise is also assumed as additive noise where every data points are added with noise sampled from the Gaussian distribution. The initial conditions of the states are specified as $x_1 = 1.0$ and $x_2 = 0.5$. The system states are simulated by setting d$t = 0.1$ second and duration $t = 15$ seconds. Correspondingly, 150 data points of the measurement data are obtained through the measurement models. These measurement data are visually depicted in Figure 5.7.

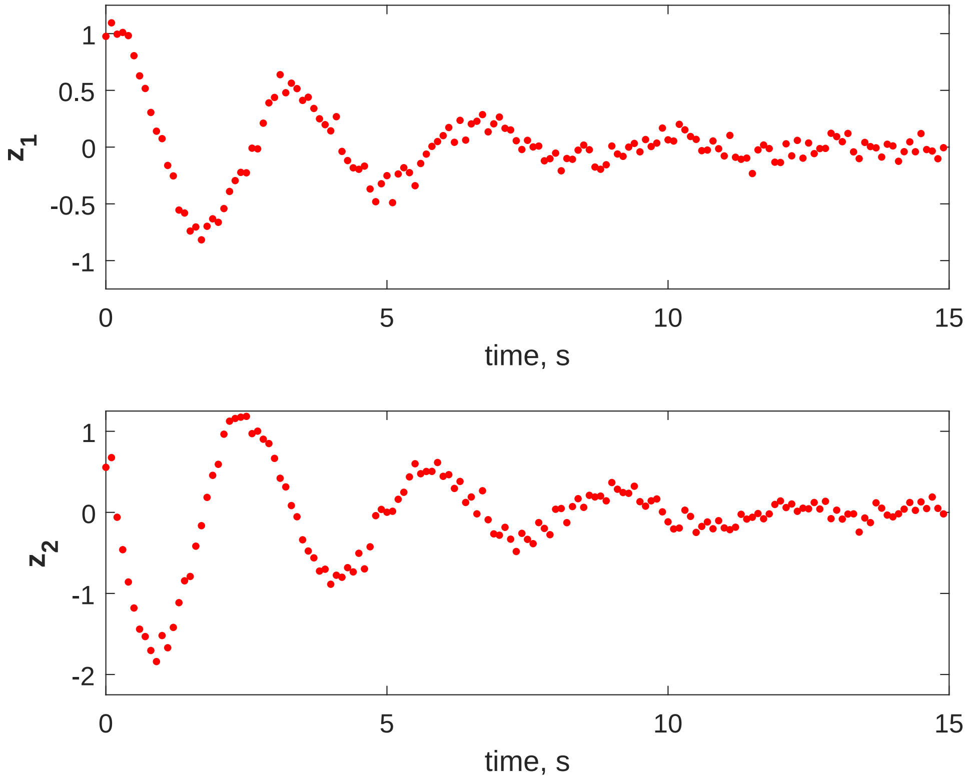


Figure 5.7: Simulated measurement data

With the generated measurement data and the system states model presented in Eqs. (5.4.1) - (5.4.2) and measurement models presented in Eqs. (5.4.3) - (5.4.4), the developed algorithm is applied to estimate the unknown parameters Θ as well as the noise statistics $\mathbf{Q}$ and $\mathbf{R}$. To start the algorithm, some initial values are required to be specified. The initial unknown parameters $\mathbf{\Theta}_0$ are specified to be $\pm 25\%$ errors with respect to the true parameters $\mathbf{\Theta}$. The initial states x_0 are also specified randomly within $\pm 25\%$ errors. The same procedures are also applied for the starting values of the process noise $\mathbf{Q}_0$ and measurement noise $\mathbf{R}_0$ covariances. As presented earlier in this chapter, the choice of the initial state error covariance matrix $\mathbf{P}_0$ also contributes to the quality of the estimates. Thus, in this simulation two scenarios are specified, one with constant $\mathbf{P}_0$ and the other one with updated $\mathbf{P}_0$ by employing Eq. (5.2.31). The remaining setups required by the algorithm are related to the maximum number of iteration in three different loops, the first loop is related to the estimation of the unknown parameters which is specified to 20 iterations. The second loop is for the estimation of the process noise $\mathbf{Q}$ while the last loop related to the estimation of the measurement noise covariance $\mathbf{R}$ which are specified to 100 and 20 iterations consecutively. Having specified the setups, the algorithm is executed, and the following quantities are analyzed.

1. Convergence of the estimated parameters and their corresponding uncertainties. The estimated parameters are also compared to the true parameters.
2. Convergence of the estimated noise statistics ($\mathbf{Q}$ and $\mathbf{R}$ matrix covariances) and its corresponding ratio with respect to the true value.
3. Predicted output and residual analysis.
4. Comparison of the estimated parameters that result from constant $\mathbf{P}_0$ and computed $\mathbf{P}_0$.

5.4.1 Convergence of the Estimated Parameters

The algorithm needs seven iterations until the convergence criterion is achieved, i.e. the relative change of the estimated parameters in the two consecutive iterations is less than **1E-4**. The parameters start to converge at the third iteration and slightly change until the end of the iteration. The convergence of the estimated parameters is also followed by the decrease of the standard deviations. The decrease of the standard deviation indicates the increase certainties of the estimated parameters over the iterations. Since the true parameters are known in this simulated case, they can be used to check the quality of the estimates. Table 5.1 shows the final estimated parameters along with the standard deviations and compared to the true parameters' values. Comparisons between estimates and true parameters values are computed through a parameter factor which is quantified by taking a ratio between the estimate and the true parameter. This quantity is shown in the last column of Table 5.1. A parameter ratio equals to 1 indicates that the estimate is the same as the true parameter value. As shown in Table 5.1, the accuracy of the estimated parameters with respect to the true parameters is high ($\geq 95\%$), except for the last parameter (parameter c) which is around 60%. However, as briefly discussed at the beginning of this section, the parameter c is addressed as a weak parameter in which the contribution of this parameter does not significantly change the system's behavior in general. Correspondingly, the generated data obtained through the simulation have a limited

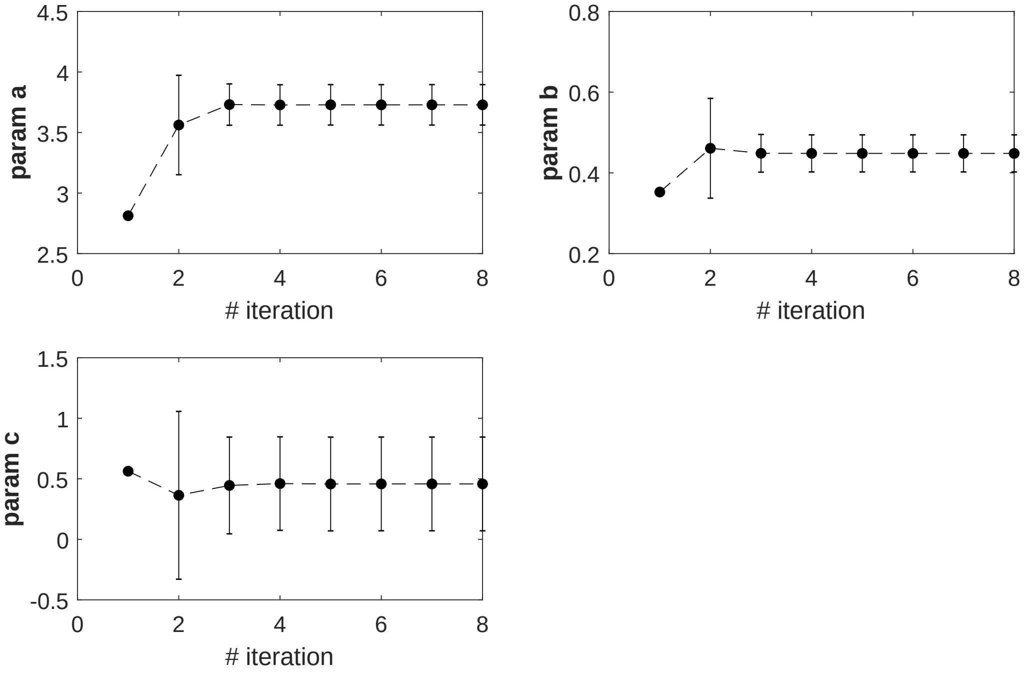


Figure 5.8: Convergence of the estimated parameters vs iteration

information content of parameter c. In this case, parameter c is less observable than parameter a or b. This is the reason behind the poor estimates of parameter c, i.e., not because of the capability of the algorithm but due to the limited information of the respective parameter in the 'measured' data.

Parameter	Θ_{true}	**Starting value**, Θ_0	Θ_{est}	Θ factor
a	3.750	2.8125	3.7286 (0.45)	0.994
b	0.470	0.3525	0.4482 (0.10)	0.954
c	0.750	0.5625	0.4575 (1.54)	0.610

Table 5.1: True parameter values vs estimated parameters

A sensitivity analysis is also conducted on the algorithm with respect to the starting values of the estimated parameters. In this analysis, the initial values of the parameters are varied with errors between $\pm(10-90)$ % with respect to the parameters true values. The results show that the algorithm can still robustly estimates the parameters close to their corresponding true values except for the weak parameter.

5.4.2 Convergence of the Estimated Noise Statistics

Both standard deviations of the process noise and the measurement noise are part of the parameters to be estimated by the algorithm. These noise statistics are estimated in two separate loops in which standard deviation of matrix covariance $\mathbf{Q}$ is estimated first, and once it achieves the cost function criteria, the resulted estimates are used to estimate statistics of $\mathbf{R}$ covariance. This process is iteratively carried out until the convergence criteria is achieved by the process and measurement noise covariances. Figure 5.9 depicts the convergence of the process noise (upper part) and the measurement noise (lower part) standard deviations.

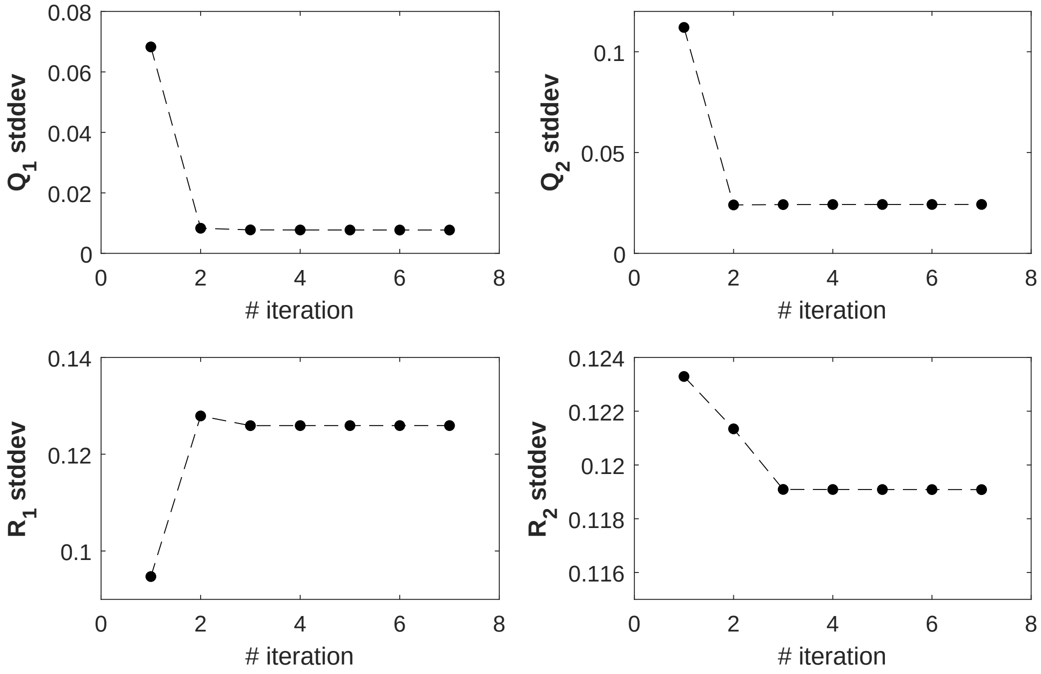


Figure 5.9: Convergence of the estimated noise statistics vs iteration

The same as the system's parameters estimates, the convergence of the process noise and measurement noise starts at the third iteration and changes slightly until the end of the iteration. To compare with the noise statistics true values, the noise ratio is computed by dividing the estimates with the noise statistics true values. Even though, the process and measurement noise converge, the corresponding estimates are poorly estimated compared to the system's parameters estimates. This is shown by the noise ratio between the estimates and the true values of the noise statistics, see Table 5.2. The noise factors for covariance matrix $\mathbf{Q}$ is higher than 60%, and for covariance matrix $\mathbf{R}$ is higher than 80% which is better than the result of the estimated covariance $\mathbf{Q}$.

These poor estimates might result from several sources. When the algorithm is applied, the initial state values are assumed unknown and specified to $\pm 25\%$ error with respect to the true initial state values. This error condenses in the noise which consequently affects the noise

Parameter	True Value	Starting value	Estimated	Noise ratio
$\mathbf{Q}_1$ stddev	0.0125	0.0219	0.0077	0.6167
$\mathbf{Q}_2$ stddev	0.0350	0.0613	0.0243	0.6931
$\mathbf{R}_1$ stddev	0.1000	0.1750	0.1259	1.2592
$\mathbf{R}_2$ stddev	0.0900	0.1575	0.1191	1.3231

Table 5.2: True noise statistics vs estimates

behavior. This affected noise is used by the algorithm to estimate their corresponding statistics which consequently lead to different estimates compared to their original noise statistics. The second source that might contribute to the decrease in the noise estimate's quality is the assumption that the noise is uncorrelated. This assumption is true during the sampling process. However, the state x_1 and x_2 has a high correlation as presented in Eqs. (5.4.1) and (5.4.2). Thus, adding noise in x_1 consequently affects the noise in the x_2 state which makes the two noises correlated. However, during the noise statistics estimation, the algorithm forces the noises to be uncorrelated by setting the off-diagonal elements of matrices $\mathbf{R}$ and $\mathbf{Q}$ equal to zero (see Section 5.3), which consequently decrease the estimates' quality. Even though this treatment affects the estimated parameters, but it ensures the stability of the algorithm which is preferred to be used in the implementation.

5.4.3 Model Output and Residual Analysis

The last analysis carried out to validate the developed algorithm is by computing the model output using the estimated parameters and comparing the results to the measurement. Two output variables y_1 and y_2 are computed and compared to their corresponding measurement z_1 and z_2. Coefficient of determination R^2 for each predicted model is computed. Coefficient of determination R^2 represents the proportion of the variation in the measured output that is explained by the model [25]. In another term, this coefficient quantifies the closeness between the measurement and the predicted model. R^2 ranges between 0 to 1, where 1 indicates a perfect fit between the model output and the measurement. This coefficient is formulated as [25],

$$R^2 = \frac{SS_R}{SS_T} \tag{5.4.5}$$

Where,

$$SS_R = \frac{1}{N}\sum_{k=1}^{N}[\tilde{y}_k - \bar{z}_k] \tag{5.4.6}$$

$$SS_T = \sum_{k=1}^{N}[z_k - \bar{z}]^2 = z^T z - N\bar{z}^2 \tag{5.4.7}$$

$$\bar{z} = \frac{1}{N}\sum_{k=1}^{N} z_k \tag{5.4.8}$$

Variable SS_R denotes the regression sum of squares, while SS_T denotes the total sum of squares. Figure 5.10 depicts the computed outputs and their corresponding measurements. As indicated by both figures, the model output provides high coefficient of determinations (R^2), i.e. greater than 90%. $R^2 = 1$ means that the model perfectly fits with the measurement which is not expected for system identification as this value means that the model fit with the noise of the measurement. Thus, the estimated parameters that give R^2 greater than 90% or even greater than 80% is acceptable in the system identification purpose. Correspondingly, the residual analysis can also be used to check the whiteness of the residual. The more residual resembles the Gaussian distribution, the better the model output that results from the algorithm. As briefly discussed at the beginning of section (5.2.1.1) and (5.2.1.2), the assumption of the noise used when deriving the algorithm is that it follows the Gaussian distribution with zero-mean. Thus, it is also expected that the result of the residual also follows this assumption. There are several residual plots that can be used to test the whiteness of

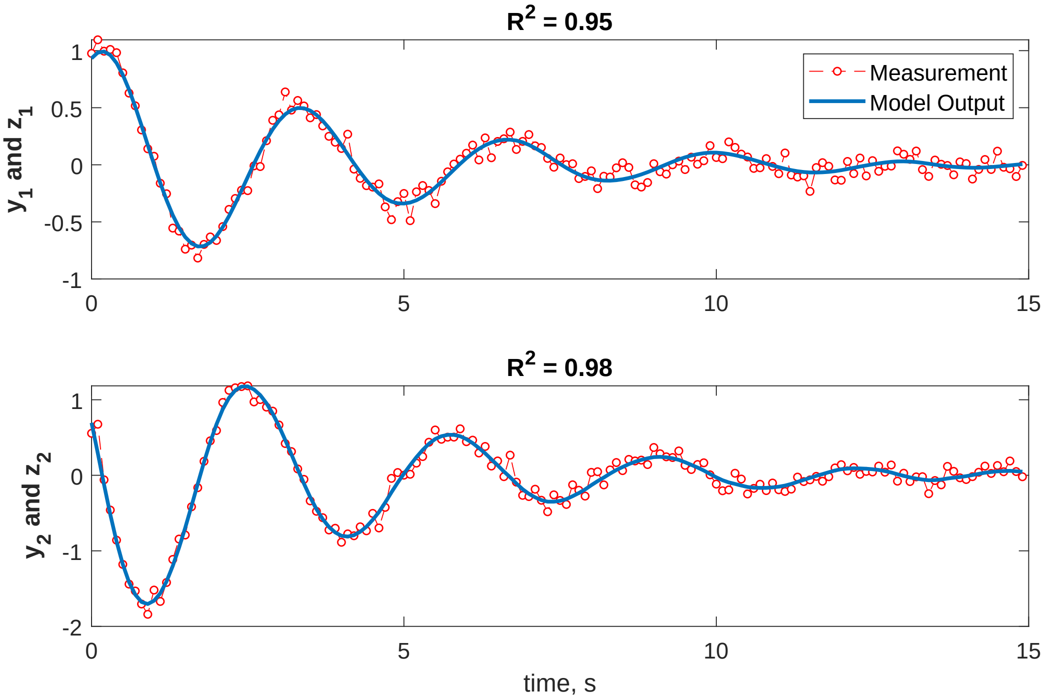


Figure 5.10: Proof of match: model output and measurement

the residual; two of them are depicted in Figure 5.11. In the upper part (called as Q-Q plot), the figure shows the comparison between the theoretical white noise (Gaussian distribution) and the residual plots that result from the developed algorithm. As indicated by the figure, the residual plots (depicted by '+' sign) follow the theoretical Gaussian distribution with only minor curvature. The lower part of Figure 5.11 shows the same findings that the residuals are distributed randomly around zero and there is no deterministic behavior captured by the residual. From both residuals plots, it may be concluded the residuals that result from the developed algorithm provide a good fit with the Gaussian distribution.

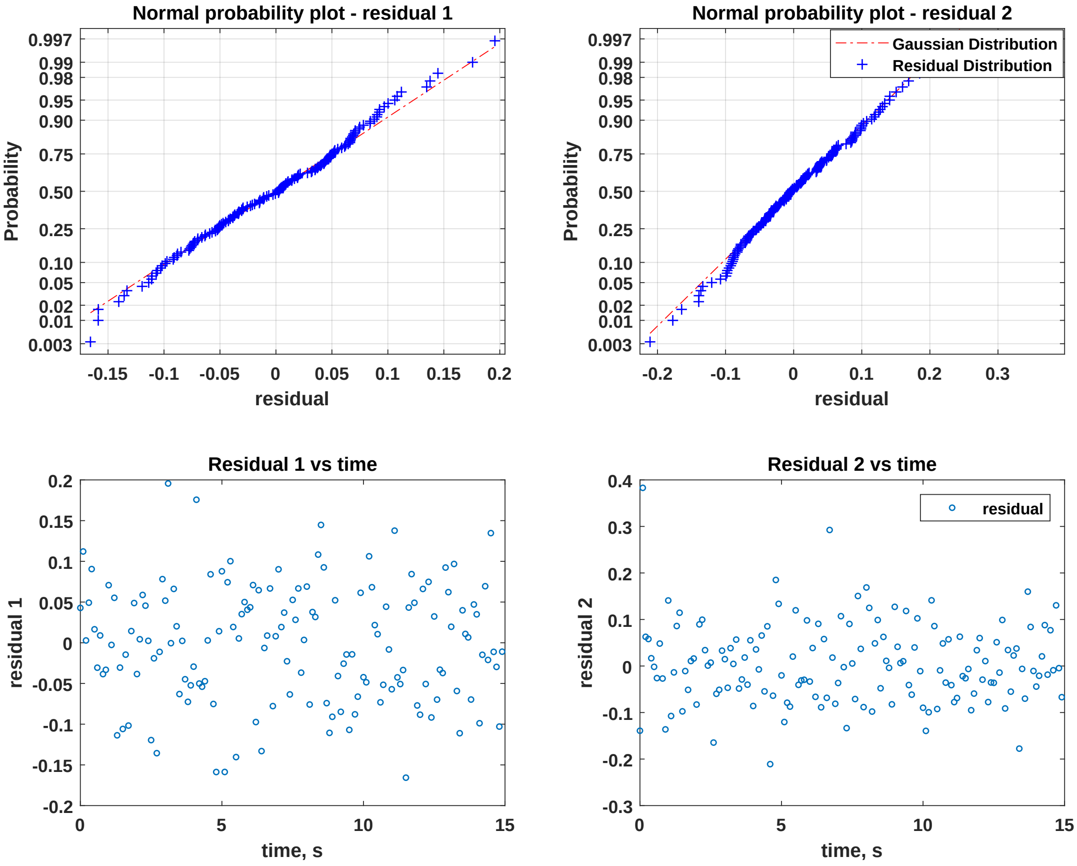


Figure 5.11: Residual plot for whiteness test

5.4.4 Estimated Parameters with Constant and Computed $\mathbf{P}_0$

The previously presented results are obtained by updating initial state error covariance $\mathbf{P}_0$ every time the loop is started. As discussed in Section 5.2.1.3, the choice of the initial state error covariance also contributes to the quality of the estimates. In order to prove this hypothesis, the developed algorithm is executed by updating $\mathbf{P}_0$ and with constant $\mathbf{P}_0$. All setups such as initial state value $\boldsymbol{x}_0$, initial parameter $\boldsymbol{\Theta}_0$, and initial noise statistics $\mathbf{Q}_0$ and $\mathbf{R}_0$ are specified at the same values as in the updated $\mathbf{P}_0$. The estimates are then compared between the two scenarios. Table 5.3 shows the results obtained from the constant and updated $\mathbf{P}_0$. The values inside the brackets in the last two columns of Table 5.3 indicate the parameter ratio which is the ratio between the estimates and the true parameters' values.

Parameter	Θ_{true}	**Starting value, Θ_0**	Θ_{est} **(updated $\mathbf{P}_0$)**	Θ_{est} **(constant $\mathbf{P}_0$)**
a	3.750	2.8125	3.7286 (0.994)	3.7382 (0.997)
b	0.470	0.3525	0.4482 (0.954)	0.4428 (0.942)
c	0.750	0.5625	0.4575 (0.610)	0.4050 (0.540)

Table 5.3: Estimates with updated and constant $\mathbf{P}_0$

Parameter ratio equal to 1 indicates that the estimates are the same as the parameter's true value. As indicated in the table, both the scenarios produce the estimates with a parameter ratio close to 1 of the first parameters a and b with only a slight difference. In parameter a, the estimates that result from constant $\mathbf{P}_0$ is slightly better than the estimate from constant $\mathbf{P}_0$. However, for parameters b and c, the estimates that results from updated P_0 provide better estimates. Besides the estimates' quality, the updated $\mathbf{P}_0$ also provides a shorter iteration compared to the constant $\mathbf{P}_0$. In this simulated case, the updated $\mathbf{P}_0$ needs seven iterations until it converges while the constant $\mathbf{P}_0$ needs ten iterations. This comparison is depicted in Figure 5.12. Thus, from the example presented in this section, the algorithm with updated $\mathbf{P}_0$ is preferred and adopted for the application to real flight data.

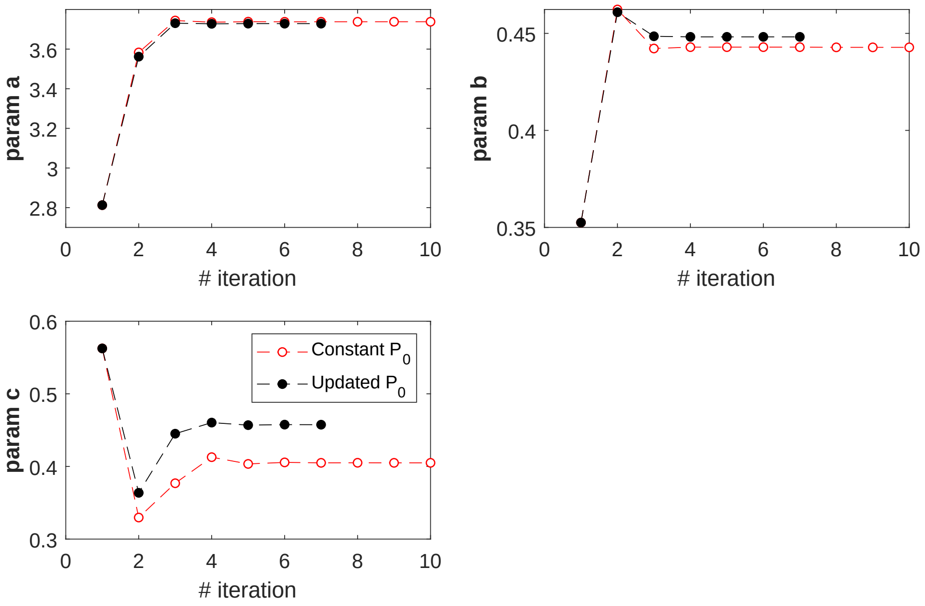


Figure 5.12: Estimated parameters with constant and updated P_0

Chapter 6

Implementation

This chapter presents the application of the developed system identification method elaborated in Chapter 5 to the FDM data. The chapter starts by providing an overview of the raw FDM data utilized in this dissertation. This is followed by presenting the decoding process of the FDM data into engineering unit. Afterward, data preprocessing steps including data and flight phase selection will be described. After the data preprocessing step, application of the developed method for parameter estimation purpose on the selected data will be elaborated. This chapter is concluded by presenting results and computational performance aspects of the developed algorithm.

6.1 Data Overview

The flight data received from the airline partner belong to different aircraft types that depart and land at different airports around the world. Since the estimated parameters depend on aircraft type, airport/runway and weather condition, the flight data are selected and grouped based on these categories. However, flights based on weather condition cannot be selected as the received data are de-identified by the airline partner in which the time and date of the flight are removed from the data. Thus, for demonstrating the applicability of the developed algorithm, flight data belong to a specific aircraft type that land at an airport and on different runways are selected. A total of 9.325 flights are processed and further clustered based on the runway and only flights that land on the same runway will be used for the analysis. Due to the data protection agreement, some meta information of the utilized flights will be discarded from the text. Table 6.1 summarizes the flight data used in this work. The flight data are received in a binary format with each filename indicates the aircraft type.

Along with the flight data, the dataframes required for decoding the data are also provided by the airline partner in a `*.txt` format. For ease of access to the information inside the dataframe, these data will be reformatted into a database format. Since each filename indicates the aircraft type, the corresponding dataframe can be directly used for decoding the data without checking the dataframe structure of each file. Furthermore, each file represents one full flight, thus no splitting algorithm required for the received data. Normally, one binary file might contain several flights as the current data storage is big enough to store several flights. In addition to that, an airline usually downloads the flight data not in every landing-basis but in a certain period that results in several flights recorded in one binary file [17]. Besides the flight data and dataframe, no other information received from the airline partner. Thus, data decoding from binary data into engineering unit, departure and arrival airports/runways detection are carried out in this work. Section 6.2 and 6.3 present these steps in details.

Aircraft Type:	Airbus series with 4 turbofan engines configuration
Airport:	Within EU region
Runway	**# Flight**
A	1,480
B	810
C	649
D	537

Table 6.1: Flight data summary

Meanwhile, the selection of flight phase is focused on final approach to landing phase, particularly 1,000 - 100 ft above ground level (AGL) during the final approach phase and 100 ft AGL until the end of the landing phase. In the first phase (1,000 - 100 ft AGL), the analysis will be focused on the longitudinal dynamics only, and the corresponding aerodynamics and thrust parameters will be estimated during this phase. While in the second phase, 100 ft until the end of the landing phase, the parameters related to deceleration forces (aerodynamics and friction forces coefficients) will be estimated. Figure 6.1 depicts the flight phase selected for the analysis. The selection of these two windows of interest is presented in Section 6.3.2.

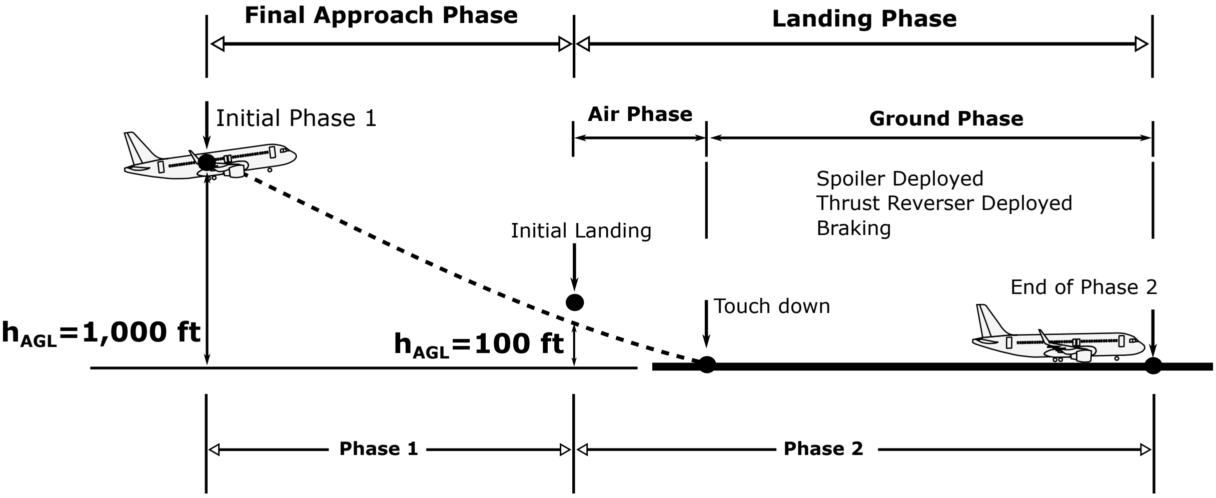


Figure 6.1: Selected flight phase

6.2 Data Decoding

Decoding the binary data into an engineering unit is started by reading a respective dataframe from a database. This step is followed by reading a binary file from storage and reformatting the 1D-bits stream into a 3D array format. The last step is to decode each parameter by using information retrieved from the dataframe. The decoded data is stored in `*.mat`[1] format which provides an easy access in the Matlab environment. For the data at hand, there are around 1,535 parameters which are stored in a 512 words per second (wps) dataframe. However, for the current parameter estimation problem, not all parameters are required but only around 200+ including backup parameters which might be needed in case the main parameters cannot be used due to large data errors or no data recorded in the parameter. The following steps describe the decoding process in details and visually depicted in Figure 6.2.

1. Reading the dataframe information. Dataframe information is stored in a MySQL database. The dataframe contains bit structure, e.g. 256, 512, 1,024 wps, parameter information such as bit location, scale factor, offset, parameter type, and other information. Once the dataframe are read from the database, they will be stored temporarily in the Matlab workspace for ease of access. Along with this dataframe information, the decoding algorithm also reads the meta information of the binary file from the database. This meta information such as file id, dataframe type, and other information are required later in step 2.
2. Reading the binary file. The corresponding binary file is read from storage. The algorithm identifies which file to be read by taking the file's meta information as provided in step 1. Matlab's built-in function `fread` is used to read the binary file. By providing the binary file's address and function's option set to `'ubit1'`, `fread` function will read every bit in the data and store each bit as a double format in Matlab's workspace for further process.
3. Reformatting 1D array of bits stream into the dataframe structure. The binary streams are reformatted following the dataframe structure in which the bits are formatted into subframe, frame, and superframe format. This process is started by finding the sync words for each subframe. Once the sync words found, the bits stream are formatted into subframe, frame and if required to decode a superframe parameter, the bits are then further formatted into superframe structure. In this step, the algorithm does not check every bit to find the sync words of each subframe but only check one sync word (either sync word #1, #2, #3 or #4). Once the sync word detected, the algorithm directly jumps to the next sync word to find the next subframe. The step of this jump depends on the file dataframe structure. For example, the 512 wps dataframe requires $512 \times 12 = 6,144$ bits jump. For data integrity check, each sync words is checked in every jump, if the pattern does not match with the sync word, the process is repeated by reading bit by bit until the next sync word found again.

[1] MAT-files are binary Matlab files that store workspace variables

4. Decoding parameter. In this step, the dataframe retrieved from step 1 is used to decode each parameter. Decoding of parameters is carried out in a serial stream by picking one respective parameter information from the dataframe at each time.
5. Unit conversion and data storing. In this last step, the unit of each parameter is converted into SI units, for example radio altitude and barometric altitude units are typically in foot which is changed into meter. After the unit conversion, the decoded parameters are stored into a `.mat` file format.

All decoding algorithm is developed in the Matlab environment. The running time is around 24.6 seconds for decoding one file that contains 267 parameters. In total, the algorithm takes around 63.7 hours to decode 9,325 flights. For speeding up the decoding process, the Matlab parallel processing toolbox is used in which the decoding process is carried out by utilizing all existed cores in the utilized computer. With eight cores, the decoding process reduces to only eight hours for all flight data.

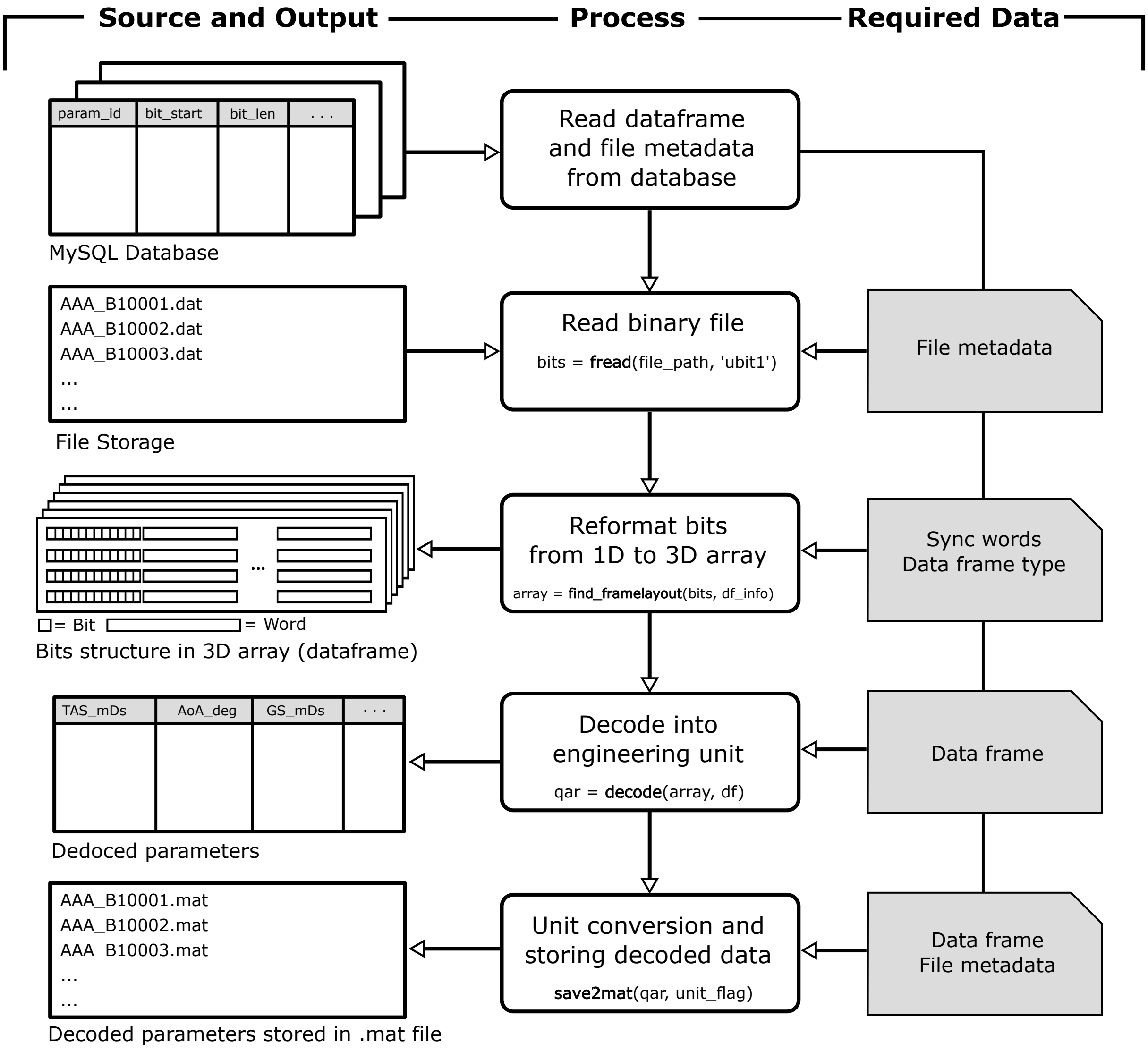


Figure 6.2: Raw FDM data decoding diagram

6.3 Data Preprocessing

Two steps are conducted in the data preprocessing, the first one related to flight selection based on categories such as aircraft type and airport/runway. The second step is concerned with flight phase selection which will split the full flight into a window of interest, for example, flight during final approach or flight during the landing phase. These two data preprocessing are discussed in details in the following sections.

6.3.1 Data Selection

One of the steps conducted in the data preprocessing is data selection where each flight will be classified into categories such as aircraft type, airport, and runway. This step is required by the developed algorithm so that the estimated parameters are relevant with respect to the selected categories. Based on flight data received from the airlines, the classification based on aircraft type category is a very straightforward step since this information is part of the filename of the data. In this case, the classification is done through a string manipulation on the filename. The results of this process are stored in a database for used later in the data selection based on aircraft type categories without repeating the string manipulation. However, the airport and runway identifier code for both departure and arrival flights cannot be inferred from the filename. Thus, this information needs to be inferred from other parameters that are recorded in the data. For example, departure or arrival airport code can be inferred through the latitude and longitude parameters along with information retrieved from an airport database. If the point of interest is the arrival airport, then one point of latitude and longitude parameters are retrieved from the data during the landing phase or at touchdown point. Then, the distance is computed between the observed point and the point retrieved from the airport database. The closest distance corresponds to the airport of interest. However, performing this computation on the whole data and comparing them to each airport in the database is very inefficient since every observed point needs to be evaluated to every airport available in the database. Therefore, a more systematic and vectorized approach is preferred.

One approach available that is borrowed from the Machine Learning field is the k-nearest neighbors (k-nn) algorithm. This algorithm belongs to the supervised algorithm which requires input (features) and output (known label) data to train the Machine Learning model. Afterward, this model will be used to label or to classify the new sample data [148]. Basically the k-nn algorithm works by memorizing the training data and then predicting the label of any new sample data based on the labels of its closest neighbors in the training data. The closest neighbors are quantified in term of 'distance' of input (feature) used in the training data and sample data to be predicted. The computed distances in terms of the k-nn algorithm can be seen as a measure of similarity between the training data and the new sample data. The closest the distance, the closest the similarity of the sample data to the class defined in the training data. There are several distance measures which can be used in this algorithm, e.g.

Euclidian, Manhattan, and Minkowski. They are formulated in Eqs. (6.3.1) - (6.3.3) below [149].

$$\text{Euclidean distance} \quad d(x_j, x_k) = \sqrt{\sum_i (x_{j,i} - x_{k,i})^2} \tag{6.3.1}$$

$$\text{Manhattan distance} \quad d(x_j, x_k) = \sum_i \mid x_{j,i} - x_{k,i} \mid \tag{6.3.2}$$

$$\text{Minkowski distance} \quad d(x_j, x_k) = \left(\sum_i^n \mid x_{j,i} - x_{k,i} \mid^p \right)^{1/p} \tag{6.3.3}$$

Where x_j denotes data to be classified and x_k represents the training data. In this dissertation, the algorithm is set to use Euclidean distance as it provides the real distance between the observed data point and the training data. Besides distance, the k-nn algorithm classifies data based on majority vote which is defined through 'k' as input to the algorithm [150]. Visually, the k-nn algorithm is depicted in Figure 6.3.

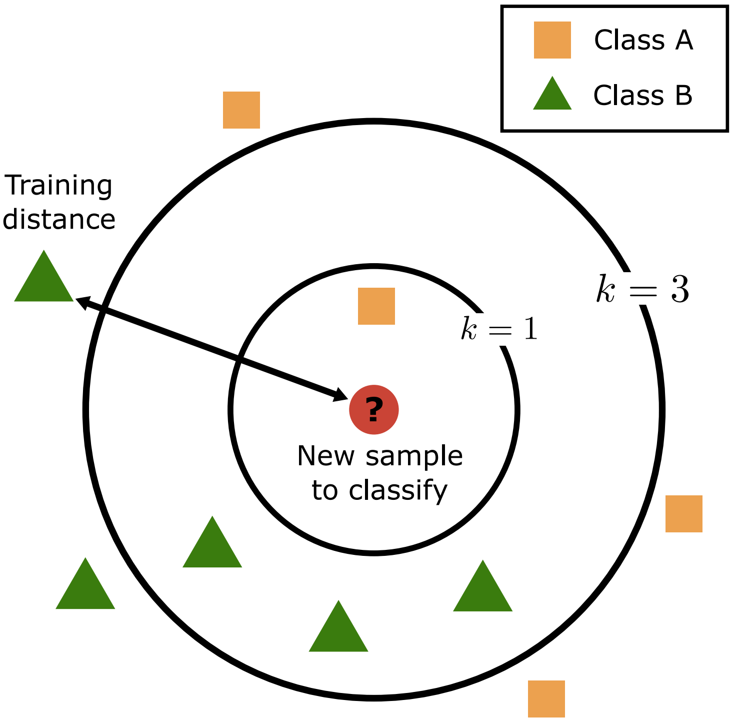


Figure 6.3: Illustration of the k-nn algorithm

As indicated in Figure 6.3, a new sample data (the circle symbol) will be classified into Class A if $k = 1$ since this point has the closest distance to Class A (the square symbol) and the algorithm is set to only select one vote (the closest neighbor). However, the sample data will be labeled to be Class B (the rectangle symbol) if $k = 3$ because there are two rectangles and only one square in the class of the selected training data. In this case, Class B dominates Class A in terms of a number of votes. Thus, the choice of k in this algorithm is critical and typically depends on the data analyzed. As a general rule of thumb, k is selected as an odd integer number to avoid tie condition. The larger values of k reduce the effect of the noise in the classification but make boundaries between classes less distinct. For airport or runway detection, the choice of k is simply 1 as the goal is to find the closest distance of the data sample to one of the airport or runway in the database. In Matlab, this algorithm is well implemented in `ClassificationKNN` class as part of the Statistics and Machine Learning Toolbox [149]. For airport detection, the training data are obtained from airport databases that are publicly available from [151] and [152]. The training data contain airport latitude and longitude along with its corresponding name in the ICAO format. These training data are used to construct the training model which will be used for predicting (labeling) the airport name of the observed data. The following Table 6.2 shows a snapshot of the training data sample as obtained from [151] and [152] databases.

airport_icao	**lat_deg**	**lon_deg**
EDDF	50.026401519775	8.543129920959
EDDM	48.353801727295	11.786100387573
ZBAA	40.080101013184	116.584999084473
KEWR	40.692501068115	-74.168701171875
CYYZ	43.677200317383	-79.630599975586
. . .	. . .	. . .

Table 6.2: Training data sample for airport detection

Meanwhile, the features of the data sample are obtained by measuring a snapshot of the latitude and longitude parameters at touchdown point or snapshot of the latitude and longitude values at the beginning of the take-off phase. The observations of the positions during touchdown are used for arrival airport detection, while the observations during the take-off phase are used for departure airport detection. As an example, Table 6.3 shows the observations used for predicting their corresponding airport name. These data points are obtained during touchdown at the arrival airport. The touchdown point is roughly obtained from the Landing Gear Switch parameters which are recorded in the FDM data. By taking these touchdown points, the corresponding latitude and longitude can be retrieved.

The implementation in Matlab is started by creating a `ClassificationKNN` a Machine Learning model through `fitcknn` function by inputting the training data (Table 6.2), in which the data will be stored and reused for prediction. After constructing the training model, the next step is to predict the airport name of the data sample. This step is carried out through the `predict` function which is part of the `fitcknn` class. The following pseudo-code shows how the airport detection problem is implemented in the Matlab environment.

```
%% Data Preparation
% Construct training data by using latitude
% and longitude parameter as obtained from online
```

lat_deg	lon_deg	airport_icao (predicted)
40.69580098	-74.16201231	To be labeled
50.03840721	8.581583805	To be labeled
35.7471865	140.3864018	To be labeled
28.55911712	77.11808334	To be labeled
40.06269744	116.6158477	To be labeled
43.67727769	-79.65830802	To be labeled
. . .	. . .	. . .

Table 6.3: Observed data sample for airport detection

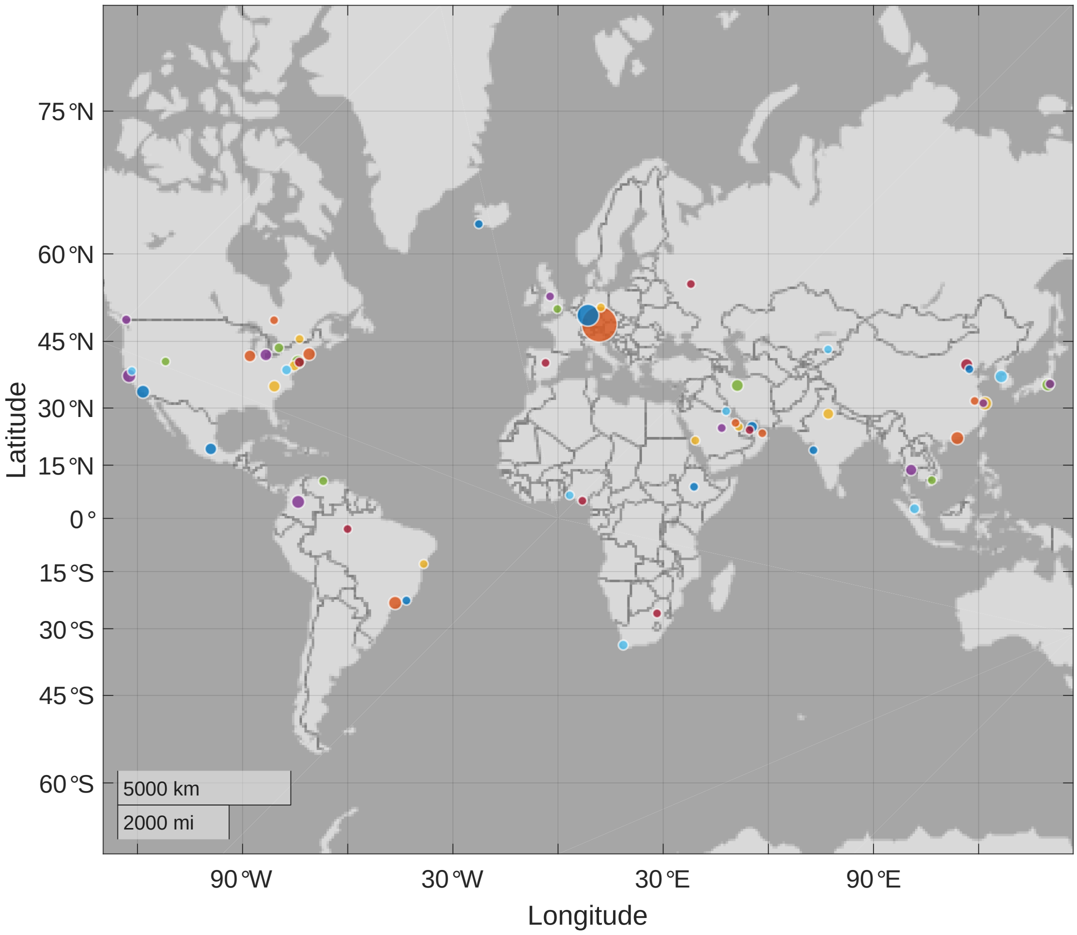
```
% airport database.
% For target model, the airport ICAO code are used.
xtrain = table(lon_deg, lat_deg);
ytarget = table(airport_icao);

% Construct training model
% Training model is constructed by inputting training data
% and target data.
trainmodel = fitcknn(xtrain, ytarget);

% Construct observed data - to be predicted
xmeas = table(lon_deg, lat_deg);

%% Do prediction
predicted_airport = trainmodel.predict(xmeas);
```

The above algorithm is applied to 9,325 flights with 6,735 training data. In total, the algorithm took less than 10 seconds to predict the arrival airports of the flight which is very efficient compared to the traditional approach. For the data sample shown in Table 6.3, the corresponding predicted airport codes are shown in Table 6.4 and for the whole flight data, the predicted airports are depicted in Figure 6.4. The size of the bubble indicates a relative number of flights landing at that particular airport compared to all flights in the other airports. The airport IDs are de-identified from the figure for data protection purpose.

The same workflow is also apllied for runway detection but with one additional feature on both training and observed data, i.e., runway bearing angle for training data and track angle on the observed data. The runway bearing angle can be obtained from the airport database while the track angle is a snapshot of aircraft track angle parameters as recorded in the FDM data. This additional feature leads the k-nn algorithm to robustly detect the correct runway as there might be multiple and overlap runways in an airport. In the training data, the latitude and longitude are corresponding to the runway threshold position. Meanwhile, the latitude and longitude parameters in the observed data remain the same as constructed in the airport

lat_deg	lon_deg	airport_icao (predicted)
40.69580098	-74.16201231	KEWR
50.03840721	8.581583805	EDDF
35.7471865	140.3864018	RJAA
28.55911712	77.11808334	VIDP
40.06269744	116.6158477	ZBAA
43.67727769	-79.65830802	CZZY

Table 6.4: Airport ICAO name as labeled by the k-nn algorithm

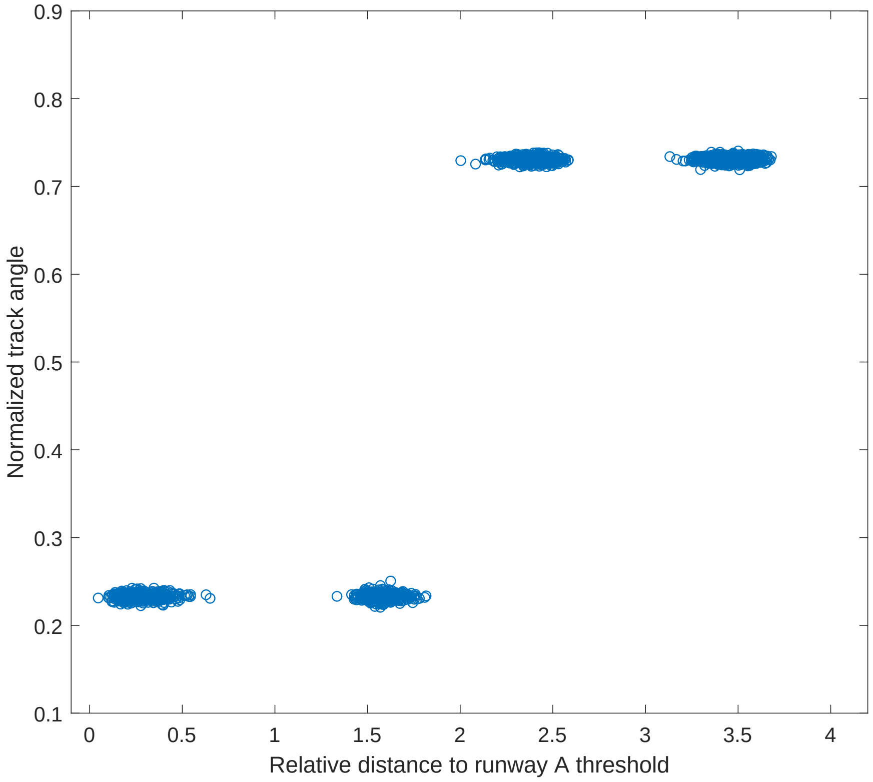


Legends are hidden from plot for data protection purpose. The bubble location is an indicative of the detected airport. The bubble size indicates the relative number of flights between one airport with respect to the other airports. The colors are meant for differentiating among the airports

Figure 6.4: Indicative locations of the detected arrival airports

detection case. As presented at the beginning of this chapter that not all flights will be used for the analysis but some of them that have the same airport/runway location will be selected. For this purpose, one airport is selected.

Furthermore, flights that land on this selected airport are further classified into runway category. The observed data in this airport are visually depicted in Figure 6.5. As indicated in the figure, no outliers in the data as the observed data are distinctively clustered into four groups. The selection of latitude, longitude, and track angle as features fit well for runway detection purpose. With this distinctive cluster, the k-nn algorithm provides an accurate result of the runway detection. The summary of the implemented algorithm on the selected airport with 3,470 flights is presented in Table 6.1 at the beginning of Section 6.1.



Each points represents landing point in each runway. For data protection purpose, the x-axis in the plot is represented by a relative distance instead of latitude or longitude parameters.

Figure 6.5: Features for runway detection

6.3.2 Flight Phase Selection

For demonstrating the applicability of the developed algorithm, the final approach to landing phase is selected as window of interest. During the final approach phase, only parameters related to aerodynamics and thrust are estimated, while during the landing phase, parameters related to the deceleration forces will be estimated. The final approach phase is selected from 1,000 ft until 100 ft above ground level (AGL). This selection is justified by the fact that during this phase, the aircraft already established a final approach track in which the flight profile is typically represented by a straight trajectory. With this profile, only the longitudinal dynamics is of most concern. Thus, the aerodynamic model can be simplified by taking the parameters related only to the longitudinal dynamics. The second phase is selected from 100 ft until the aircraft reaches a minimum speed or a track angle change that less than 5 degrees. These constraints ensure that the aircraft has not vacated the runway. Thus, the respective estimated parameters are still relevant to the runway condition. A simple algorithm is developed for the flight selection window.

In the FDM context, the points that define a specific window or event is known as timepoint. Basically, this timepoint is an index of the data where a specific event or any point of interest occurs. The data index is started from 1 at the beginning of the data point and ended at the last data point. Usually, this timepoint will be referred to observe other parameters. For example, if the point of interest is at touchdown point during landing, then one can roughly use the aircraft AIR/GND switch parameter. This parameter typically represents 0 when the aircraft is in the air and 1 when it is on the ground. Then, the touchdown point during landing is detected when there is a change between 0 to 1 in this parameter. The index of this change which is started from the beginning of the data is called the timepoint. Furthermore, this timepoint is used to observe other parameters such as the value of the vertical acceleration at this timepoint, the ground speed value at this timepoint, and other related parameters. With this approach, one may expand the analysis of a specific event by observing other parameters. In the FDM context, a snapshot of a parameter value in a specific timepoint is referred to as 'measurement.' The measurement activity can also be used to verify the accurateness of the timepoint which will be demonstrated in this section.

For phase 1 (1,000 - 100 ft AGL during the final approach phase), there are two timepoints that need to be observed, i.e., timepoint at 1,000 ft and 100 ft, both altitudes are referred to ground level during the final approach phase. The radio altitude parameter is the only parameter used for the window selection. The following pseudo-code presents how this window selection is implemented in Matlab.

```
%% A pseudo-code for a radio altitude timepoint detection
% Set constant - reference value
radio_altitude_ref = 1000;

% Set nan to data from k = 1 until the middle of flight
% This meant for avoiding outliers which is usually occurs
% at the beginning of data.
radio_altitude(1:floor(length(radio_altitude)/2)) = nan;

% Find the index.
% With the find function, the radi altitude timepoint is found
% by comparing the data with the reference and yielding the
% index that first fullfills the inequality constraint.
index = find(radio_altitude <= radio_altitude_ref, 1);
```

By changing the `radio_altitude_ref` variable to 100, the above algorithm is applied for the second timepoint (100 ft AGL) detection.

For phase 2 (100 ft AGL until the end of landing phase), there is only one timepoint that needs to be observed, i.e. timepoint at the end of the landing phase since the first timepoint is provided by the last timepoint of phase 1. Two parameters are involved in this timepoint detection, i.e. track angle and ground speed. Some constraints are set to ensure that the timepoint is the last point in which the aircraft starts to vacate the runway. For ground speed, the minimum value is set to 5 m/s while for the track angle parameter, the constraint is set when the change between 2 seconds time interval is greater than 5 degrees. The index of the last timepoint (the end of phase 2) is obtained by whichever comes first of the two defined constraints. The following pseudo-code presents how this algorithm implemented in Matlab.

```
%% A pseudo-code for final landing timepoint detection
% Define some constants
SAMPLING_RATE = 8;
MIN_GS = 5;
DELTA_TRACK = 5;

% Initialize index by using index from 100 ft AGL
index = index_100ft;
while true
    index = index + 1;
    % Check the change of track angle during 2 sec interval
    dtrack = abs(track_deg(index + 2 * SAMPLING_RATE) - track_deg(index));
    if or(dtrack >= DELTA_TRACK, gs_mDs <= MIN_GS)
        break;
    end
end
```

Furthermore, to verify the selected flight phases, measurement is carried out on time parameter, that is by computing the time difference (duration) in each flight phases. These time differences are then inspected by presenting them in the histogram plots. Through these plots, any outliers can be visually detected or can also be detected by computing the statistics of the time differences, and any anomalies can be discarded from the data. Figure 6.6 shows the time required in each selected flight phase. As indicated in each figure, there are no outliers in the data. The max/min values are 53/79 seconds for phase 1 and 44.5/57 seconds for phase 2, while the average of time required in each flight phase is 66 and 50 seconds respectively.

Furthermore, plots of some parameters within the selected windows can also be used to verify that the detected timepoints are accurate. For this purpose, plots of latitude, longitude, and radio altitude of each phase are presented. The numerical values of the parameters are hidden from the figures for data protection purpose. Both of the plots are depicted in Figure 6.7 and 6.8.

As indicated by both figures, profiles of the selected windows of interest are the straight line profile as depicted by the latitude and longitude plots. Furthermore, flight selection based on radio altitude also provides an accurate result. This is shown in Figure 6.7, where the upper limit is around 300 m (1,000 ft) AGL, and the lower limit is around 30 m AGL (100 ft). While

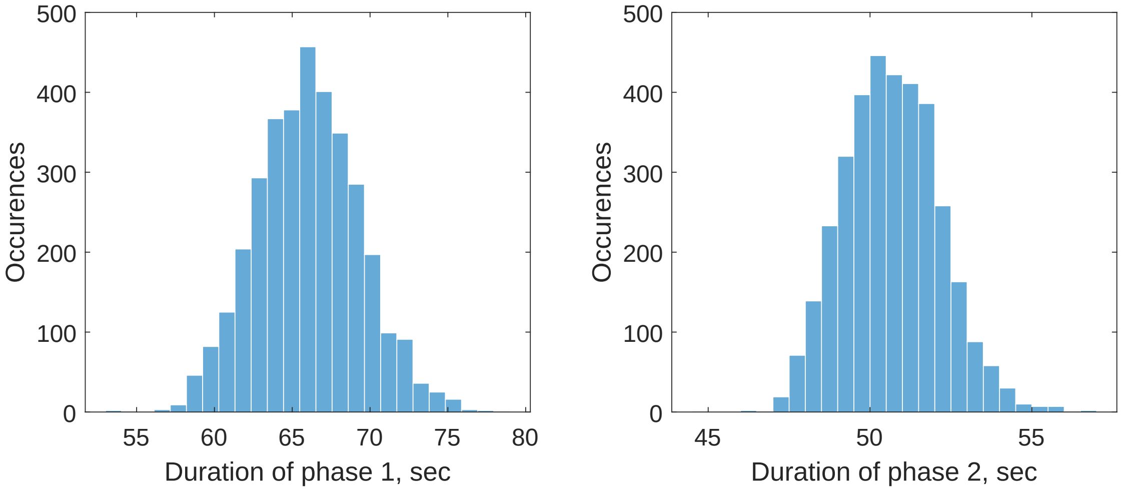


Figure 6.6: Time required in the selected windows

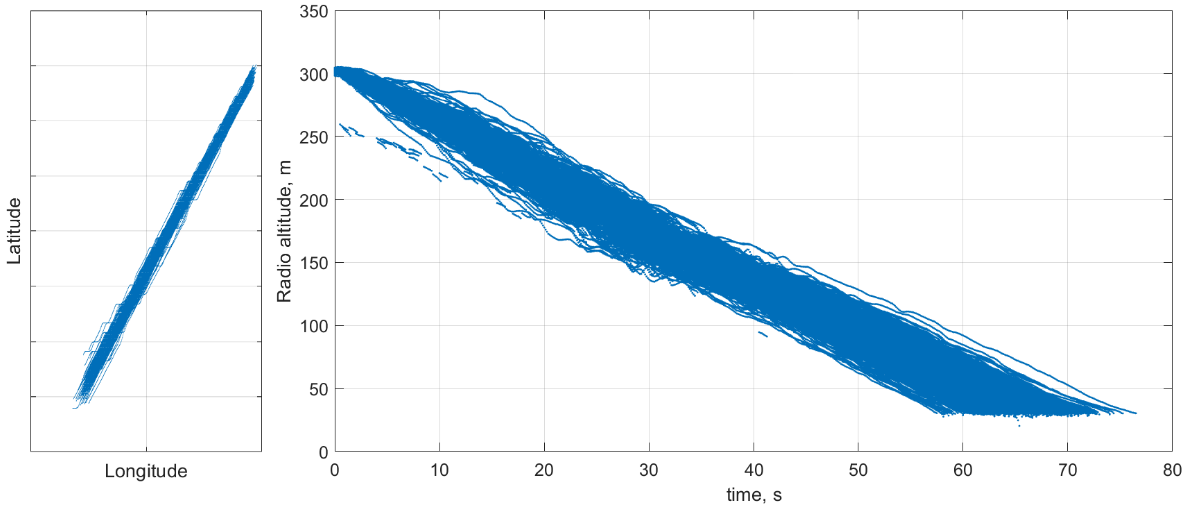


Figure 6.7: Phase 1 flight trajectory

for phase 2, the upper limit is 30 m (100 ft) AGL which is obtained from the last timepoint of phase 1, see Figure 6.8. To verify that the aircraft has not vacated the runway during phase 2 window, it can be observed that no large curvatures found from the latitude and longitude plot in left side plot of Figure 6.8.

6.4 Implementation: Final Approach to Landing Phase

Section 6.1 has briefly presented the selected flight phases in which the proposed algorithm will be applied. Besides demonstrating the applicability of the developed algorithm, the selected flight phases are closely related to the incident model used for predictive analysis in which the algorithm is being developed in the Flight Safety Group of the Institute of Flight System Dynamics of TUM. Some parameters in the incident model can be directly obtained from the FDM data as they are available as recorded parameters, and some parameters have to be

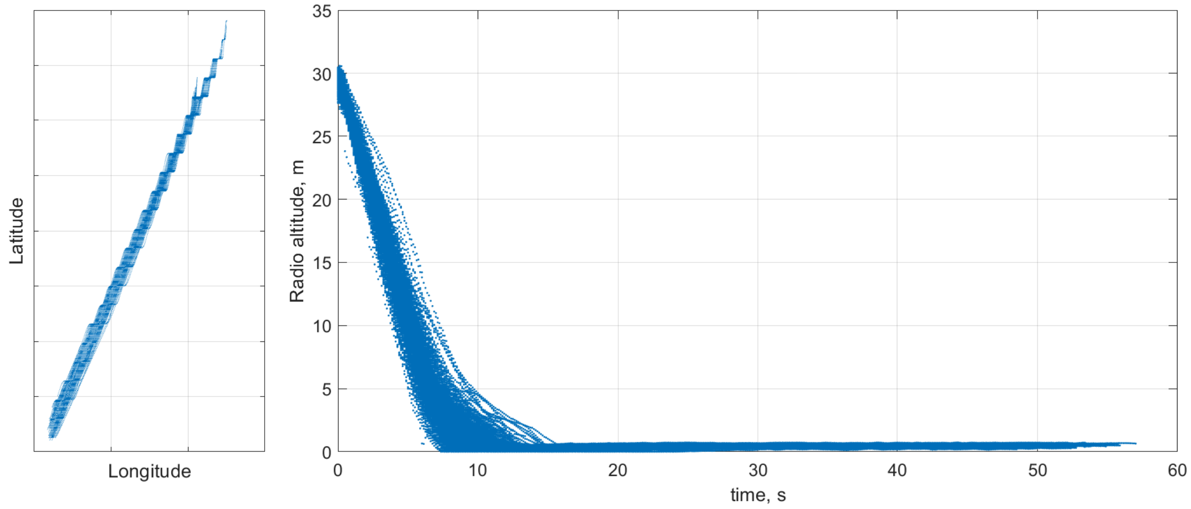


Figure 6.8: Phase 2 flight trajectory

derived from other parameters through algebraic computation or estimation approach. The parameters provided in this work are related to the latter approach.

In flight phase 1, 1,000 ft until 100 ft AGL, two main forces are involved, i.e. thrust and aerodynamics. Therefore, parameters related to the aerodynamic forces such as drag and lift coefficients are estimated. During this phase, derivation of a valid thrust model will also be presented as well as the estimation of the respective parameters. While, in flight phase 2, 100 ft AGL until the end of the landing phase, one additional force contributes to the model. This additional force which is known as friction force that contributes to the deceleration of the aircraft motion. Thus, parameters related to the friction force and the other two forces mentioned previously will be estimated. Section 6.4.1 and 6.4.2 present the model structure of each phases as well as the involved parameters to be estimated.

6.4.1 Phase 1 Model Structure

During phase 1, it is assumed that the aircraft has already established a final approach track. Thus, there are no significant movements in the lateral direction. Consequently, these assumptions lead to a simplified aircraft equation of motions that only concerns with the longitudinal dynamics only. Translational propagation equations in the Aerodynamic frame as presented in Chapter 3, specifically as formulated in Eqs. (3.6.18) - (3.6.20), are reduced to only involve two state variables, i.e. airspeed (V) and angle of attack (α). Other variables related to lateral and directional motions are ignored, and their values are set to zero. The attitude propagation equations as presented in Eqs.(3.6.14) - (3.6.16) are reduced to only include one state variable, that is, pitch angle (θ). These simplified state equations for phase 1 window are given in Eqs. (6.4.1) - (6.4.3) [18].

$$\dot{V}_{TAS} = -\frac{\bar{q} \cdot S}{m} \cdot C_D + \frac{F_P}{m} \cdot \cos(\alpha) + g \cdot \sin(\alpha - \theta) \tag{6.4.1}$$

$$\dot{\alpha} = -\frac{\bar{q} \cdot S}{m \cdot V_{TAS}} \cdot C_L + q - \frac{F_P}{m \cdot V_{TAS}} \cdot \sin(\alpha) + \frac{g}{V_{TAS}} \cdot \cos(\alpha - \theta) \tag{6.4.2}$$

$$\dot{\theta} = q \tag{6.4.3}$$

Variables C_L and C_D represent the aerodynamic lift and drag coefficients respectively. While the thrust is represented by variable F_P. The last term in Eqs. (6.4.1) and (6.4.2) denotes the gravitational force. Each of these forces is visually depicted in Figure 6.9. As indicated in the figure, all forces are referred to the Body-Fixed frame. However, the state models utilized in phase 1 are expressed in the Aerodynamic frame. Thus, the transformation of each force from Body to the Aerodynamic frame is required and described in Section 6.4.1.1 - 6.4.1.3.

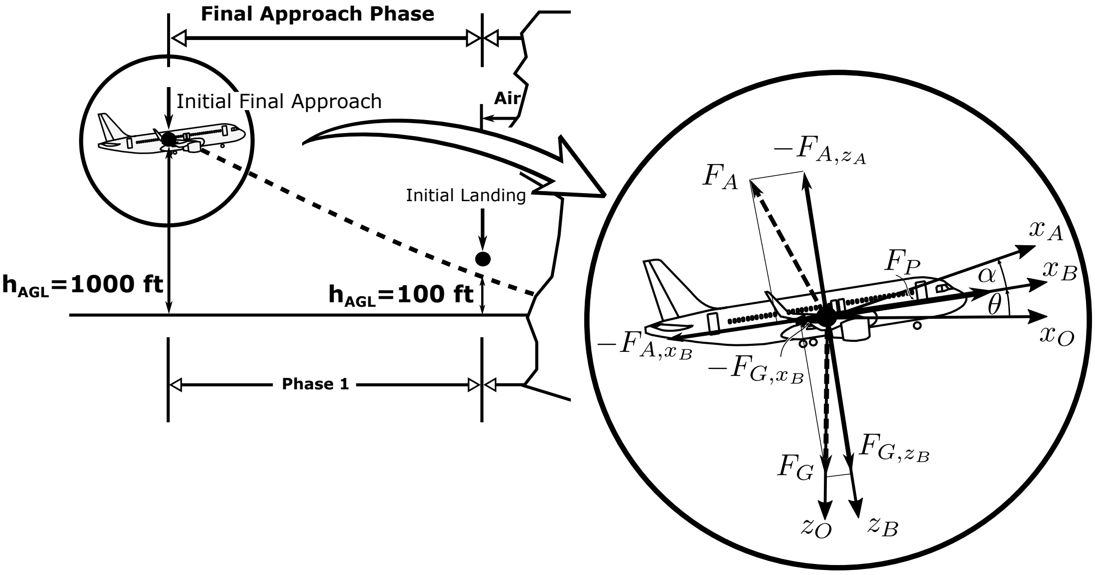


Figure 6.9: Schematic diagram of forces in phase 1

6.4.1.1 Aerodynamic Force Model

The aerodynamic forces in the Body-Fixed frame take form as,

$$F_{A,x_B} = \bar{q} \cdot S \cdot C_X \tag{6.4.4}$$

$$F_{A,z_B} = \bar{q} \cdot S \cdot C_Z \tag{6.4.5}$$

where S denotes wing reference area, $\bar{q}$ denotes dynamic pressure, and C_X and C_Z represent the aerodynamic force coefficients in x and z-axis of the Body-Fixed frame respectively. These two aerodynamic coefficients can be derived from the aerodynamics coefficient projected from the Aerodynamic frame. This projection is expressed as [25],

$$C_X = -C_D \cdot \cos\alpha + C_L \cdot \sin\alpha \tag{6.4.6}$$

$$C_Z = -C_D \cdot \sin\alpha - C_L \cdot \cos\alpha \tag{6.4.7}$$

Furthermore, variable C_L and C_D can be expanded by taking the contributing factors into account. For example, the aerodynamic coefficient can be expanded in terms of its static stability derivatives (u, v, w, α, β), dynamic stability derivatives (p, q, r) and control stability derivatives $(\xi, \eta, \zeta, \delta_{\text{spoiler}})$, see Chapter 3 for details. However, since the investigated model considers longitudinal dynamics only, some of the explanatory variables can be discarded as they have an insignificant influence on the model. Moreover, to keep the number of parameters tractable, a simple aerodynamic model yet capable of capturing the aerodynamic behavior into some extents, is preferred to be employed.

In this work, the utilized aerodynamic model of the lift coefficient takes form as a linear dynamic model with dependency only on the angle of attack (α) variable. This model is used by taking the fact that during flight phase 1, the angle of attack (α) variable dominates most of the aerodynamic behavior as it is shown by its large variabilities during this phase. The variabilities of the angle of attack (α) will be shown later in Section 6.6. While the drag coefficient C_D is modeled by a quadratic polar with a constant parasite drag coefficient $(C_{D_{min}})$ where the cambered factor is taken into account, see Figure 6.10.

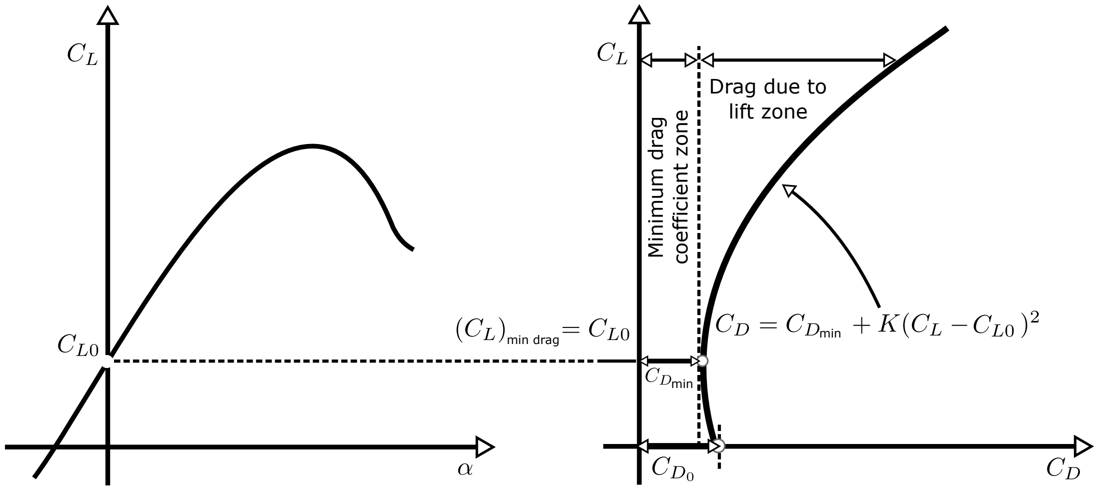

C_L
C_{L0}
α
C_L
Drag due to lift zone
Minimum drag coefficient zone
$(C_L)_{\text{min drag}} = C_{L0}$
$C_D = C_{D_{\text{min}}} + K(C_L - C_{L0})^2$
$C_{D_{\text{min}}}$
C_{D_0}
C_D

Figure 6.10: Asymmetric drag polar [153, 154]

In the asymmetric drag polar model, the minimum drag coefficient $C_{D_{min}}$ is used instead the zero-drag-lift coefficient (C_{D_0}) since in the cambered case, $C_{D_0} \neq C_{D_{min}}$. Furthermore, during the final approach phase, the flaps are in deployed state which consequently increase the cambered effect to the wing. Thus, in order to take this effect into account, this dissertation adopt the asymmetric drag polar model. Variable C_L and C_D are respectively expressed in

Eqs. (6.4.8) and (6.4.9) [153, 154].

$$C_L = C_{L_0} + C_{L_\alpha} \cdot \alpha \tag{6.4.8}$$

$$C_D = C_{D_{min}} + \frac{1}{e \cdot \pi \cdot \Lambda} \cdot (C_L - C_{L_0})^2 \tag{6.4.9}$$

Where, C_{L_0} represents the lift coefficient at zero angle of attack (α), C_{L_α} represents the gradient of the lift coefficient over angle of attack, $C_{D_{min}}$ represents minimum drag coefficient (see Figure 6.10), e denotes Oswald's factor and π, Λ respectively denote the pi number (3,1416) and wing aspect ratio. Eqs. (6.4.8) and (6.4.9) can be directly employed in the state equations presented earlier in Section 6.4.1. The unknown parameters $C_{L_0}, C_{D_{min}}, C_{L_\alpha}$ and Oswald's factor (e) are the parameters with the most concern and will be estimated based on the FDM data through the developed algorithm. However, with the aerodynamic model structure presented in Eq. (6.4.9), the correlation between C_{L_0} and e is high. Applying the parameter estimation method on these high correlation parameters leads to poor estimates since the involved parameters contain redundant information. Thus, to solve this particular issue, it is decided to compute the Oswald's efficiency factor (e) analytically while C_{L_0} is estimated through the developed method.

An algebraic model for computing the Oswald's efficiency factor is provided in many aircraft design books, namely Howe *et al.* [155], Raymer [156], and Torenbeek [157]. In this dissertation, the model from Howe's method is adopted for the computation of the Oswald's efficiency factor. This adoption is supported by the work of Zöld [158] in which he compares the Oswald's efficiency factor from three different methods, including Raymer, Howe, and Frost-Rutherford method. From these three methods, Howe's method provides a reasonable and consistent value. Since the aircraft type used in Zöld's work is also the same as used in this dissertation, for practical reason it is decided to adopt the value as computed by Zöld. The computed Oswald's efficiency factor as obtained from Zöld's work is 0.7527.

6.4.1.2 Thrust Model

Propulsive force may be represented in a very complex model by taking all relevant parameters into account and condition experienced by the engine at a particular time. However, the utilized FDM data lacks the parameters required in modeling such a complex thrust model. Thus, in this work, a simple yet still useful thrust model is used to capture thrust behavior especially in low thrust range. Based on [159], a thrust produced by the engine depends on engine RPM N_1 exponentially. This relation is expressed as [127, 160],

$$F_P = T_0 \cdot N_1^\tau \tag{6.4.10}$$

Where T_0 is the engine's maximum thrust, and τ is the exponent of the thrust model. These two parameters are unknown and will be estimated in this work. Eq. (6.4.10) is roughly derived from the data presented in [159] and adopted in this work. Furthermore, during phase 1 window, variabilities of most of the flight conditions can be regarded as small and insignificant to the generated thrust. From the FDM data, the engine parameter N_1 shows a large variability, and thus it contributes significantly to the thrust generated by the engine. With this assumption, the thrust model presented in Eq. $(6.4.10)$ can be assumed valid in this selected flight phase. The thrust model expressed in Eq. $(6.4.10)$ is referred to the Body-Fixed frame. Thus, to utilize this model in the state equations (6.4.1) - (6.4.3), it must be transformed into the Aerodynamic frame through the angle of attack (α). Since the state model only concerns on the longitudinal dynamics, the thrust model is projected only to the x-axis of the Aerodynamic frame. F_P in the x-axis of the Aerodynamic frame is formulated as,

$$\begin{aligned} F_{P,x_A} &= F_P \cdot \cos(\alpha) \\ &= T_0 \cdot N_1^{\tau} \cdot \cos(\alpha) \end{aligned} \tag{6.4.11}$$

6.4.1.3 Gravitational Force Model

The gravity force model does not involve any parameters that are required to be estimated. This force is originally referred to the NED frame. Thus, it needs to be transformed to the Aerodynamic frame through the Body-Fixed frame. Two orientation angles are required for this transformation, i.e. pitch angle (θ) that is required to transform the model from the NED frame to the Body-Fixed frame and angle of attack (α) that is required for the transformation from the Body-Fixed frame to the Aerodynamic frame. The gravitational force expressed in x-axis of the Aerodynamic frame given as,

$$F_{G,x_A} = m \cdot g \cdot \cos(\alpha - \theta) \tag{6.4.12}$$

Where m is mass of the aircraft and g is the gravitational acceleration. Eq. (6.4.12) appears in the last term of Eqs. (6.4.1) and (6.4.2).

Having discussed the state models and their corresponding forces for phase 1, the last model required for parameter estimation is discussed in the present section. This last equation is the output model which relates the state model and the measurement parameters as recorded in the FDM data. Five equations are used to represent the output models in this phase. These five models are given as,

$$a_{x,m} = \frac{1}{m}(F_{A,x_B} + F_P) + b_x \tag{6.4.13}$$

$$a_{z,m} = \frac{1}{m}F_{A,z_B} + b_z \tag{6.4.14}$$

$$V_{TAS,m} = V_{TAS} \tag{6.4.15}$$

$$\alpha_m = \alpha \tag{6.4.16}$$

$$\theta_m = \theta \tag{6.4.17}$$

6.4.2 Phase 2 Model Structure

Phase 2 is started at 100 ft AGL and ended at the end of the landing phase. Thus, there are air phase and ground phase during this selected phase. The same as in phase 1, phase 2 also considers the equation of motion only on the longitudinal dynamics. In contrast with phase 1, the forces in phase 2 are referred to runway frame (R) in order to accommodate the friction and normal force into the model. Basically, this frame is a modification of the NED frame by changing the orientation of x and y-axis but keeping the z-axis the same as in the NED frame, i.e. perpendicular to the Earth's surface. The x-axis is defined to be aligned with the runway length while y-axis points to the right side and perpendicular to xz plane. The orientation angle between x_R and x_B is defined by a pitch angle θ which is the same angle as used between x_O and x_B. Runway's slope angle is assumed to be small and is ignored in the model. Visually, the schematic diagram of forces acting on aircraft during phase 2 is depicted in Figure 6.11. Each of these forces will be detailed in Section 6.4.2.1 - 6.4.2.3. State

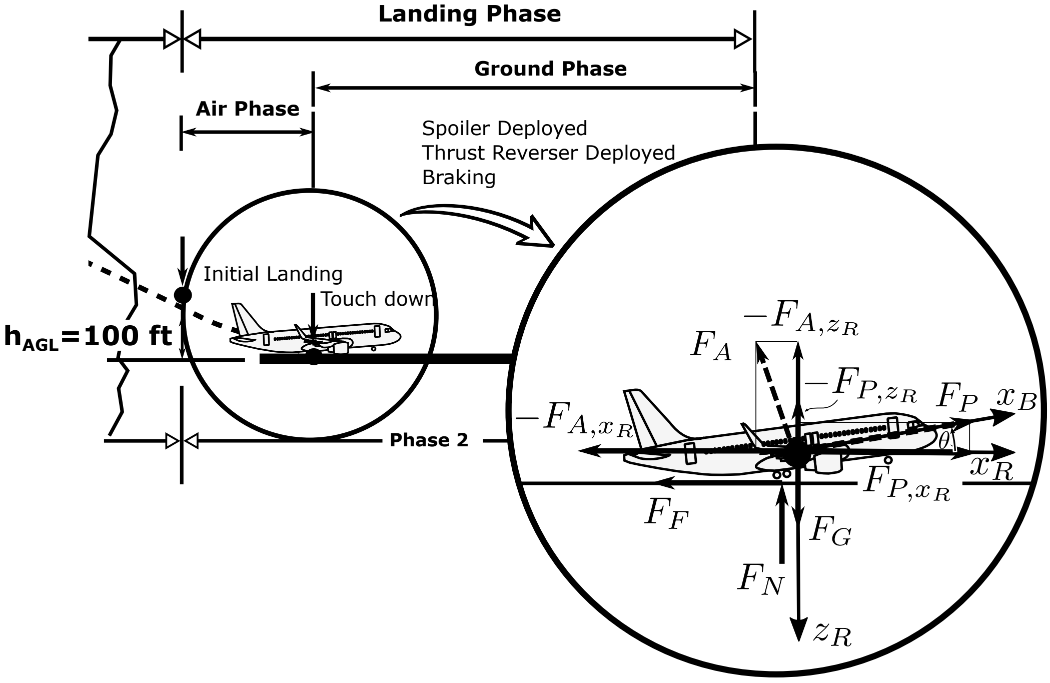


Figure 6.11: Schematic diagram of forces in phase 2

equations during phase 2 are expressed by two state variables including ground speed (V_{GND}) and pitch angle (θ). In this phase, ground speed (V_{GND}) is used instead of the true airspeed (V_{TAS}). The reason behind this replacement is due to the lack of reliability of the recorded airspeed, particularly during the landing phase where speed range is low. Thus, the employed state model for phase 2 are given as,

$$\dot{V}_{GND} = \frac{1}{m}(F_{P,x_R} + F_{A,x_R} - F_F) \tag{6.4.18}$$

$$\dot{\theta} = q \tag{6.4.19}$$

6.4.2.1 Aerodynamic Force Model

The aerodynamic model in phase 2 involves more parameters compared to phase 1's model due to the transition from air to ground and spoiler deflection. An aircraft flying close to ground experiences a phenomenon called ground effect due to the developed air cushion between the aircraft and the ground. This ground effect contributes to the increase of the lift and decrease of the induced drag which consequently leads to the increment of the lift to drag ratio (L/D) [18]. This phenomenon must be taken into account to produce a model with good predictive capability. A simple approach to address this phenomenon is by adding one parameter that accommodates the increase of the lift due to the ground effect. This parameter is addressed in C_{L_0}. Thus, two different C_{L_0} are defined in the aerodynamic model for phase 2. The first C_{L_0} is the same as formulated in phase 1, while the second C_{L_0}, written as $C_{L_0,GND}$, is used when the aircraft flies close to the ground. To ensure a smooth transition from the air phase to ground phase, a weighting factor w is introduced in the aerodynamic model. This weighting factor is the ratio between the current radio altitude (h) with respect to reference radio altitude h_0. The reference altitude is set to $r_0 \approx 20$ m with respect to the runway surface. This value is obtained by inspecting the comparison plot of the longitudinal acceleration (a_x) of the model output and its corresponding measurement ($a_{x,m}$) as recorded in the FDM data. It is found that during phase 2, specifically shortly before the touchdown, the measured longitudinal acceleration has some peaks (large fluctuation) compared to other vicinity data points. These peaks might result from the decrease of the induced drag that leads to additional force to accelerate the aircraft. The utilized aerodynamic model cannot predict these peaks without incorporation contribution from ground effect. By visual judgment, it is found that by setting the radio altitude reference value to 20 provides the best agreement of the model output (longitudinal acceleration) and its corresponding measured parameters. The model for parameter C_{L_α} is assumed still valid in phase 2, and the estimated value in phase 1 will be used in phase 2. Ground spoiler deployment affects the decrease of the lift coefficient and will be modeled as a degenerative factor to the total lift coefficient.

Aerodynamic drag coefficient model is slightly modified by incorporating the weighting factor (w) to ensure a smooth transient phase (from out of ground effect to in ground effect condition). The other change to the drag model is by adding the spoiler deployment effect. Eqs. (6.4.20) and (6.4.21) formulate the C_L and C_D coefficients during phase 2.

$$C_L = w \cdot C_{L_0} + (1-w) \cdot C_{L0,GND} + C_{L_\alpha} \cdot \alpha - \Delta C_{L,S} \cdot \frac{\delta_S}{\delta_{S,max}} \tag{6.4.20}$$

$$C_D = C_{D_{min}} + \frac{w}{e \cdot \pi \cdot \Lambda} \cdot (C_L - C_{L_0})^2 + \Delta C_{D,S} \cdot \frac{\delta_S}{\delta_{S,max}} \tag{6.4.21}$$

where,

$$w = \begin{cases} 1, & h \geq h_0(\approx 20\,\text{m}) \\ \sqrt{h/h_0}, & h \leq h_0(\approx 20\,\text{m}) \end{cases} \tag{6.4.22}$$

In the Runway frame, the horizontal x_R and vertical z_R of the aerodynamic forces take form as,

$$F_{\boldsymbol{A},x_{\boldsymbol{R}}} = \quad F_{\boldsymbol{A},x_{\boldsymbol{B}}} \cdot \cos\theta + F_{\boldsymbol{A},z_{\boldsymbol{B}}} \cdot \sin\theta \tag{6.4.23}$$

$$F_{\boldsymbol{A},z_{\boldsymbol{R}}} = -F_{\boldsymbol{A},x_{\boldsymbol{B}}} \cdot \sin\theta + F_{\boldsymbol{A},z_{\boldsymbol{B}}} \cdot \cos\theta \tag{6.4.24}$$

Where $F_{\boldsymbol{A},x_{\boldsymbol{B}}}$ and $F_{\boldsymbol{A},z_{\boldsymbol{B}}}$ can be computed through Eqs. (6.4.4) and (6.4.5) and the corresponding aerodynamic coefficients C_X and C_Z are obtained through Eqs. (6.4.6) and (6.4.7).

6.4.2.2 Thrust Model

The thrust model in phase 1 is employed in phase 2 with a minor change, i.e. by incorporating the thrust reverse factor into the model. This thrust reverse ratio is used when a pilot applies a reverse thrust for decelerating the aircraft during the landing phase. Eq. (6.4.25) presents the thrust model utilized in phase 2 [127, 160].

$$F_{\boldsymbol{P}} = \begin{cases} T_0 \cdot N_1^{\tau}, & N_1 \geq 0 \\ r \cdot T_0 \cdot N_1^{\tau}, & N_1 < 0 \end{cases} \tag{6.4.25}$$

The thrust reverse ratio (r) is the parameter to be estimated along with N_1's exponent (τ). Correspondingly, the horizontal and vertical components of the thrust model with respect to the Runway frame are computed through,

$$F_{\boldsymbol{P},x_{\boldsymbol{R}}} = \quad F_{\boldsymbol{P}} \cdot \cos\theta \tag{6.4.26}$$

$$F_{\boldsymbol{P},z_{\boldsymbol{R}}} = -F_{\boldsymbol{P}} \cdot \sin\theta \tag{6.4.27}$$

6.4.2.3 Friction Force Model

The third force in phase 2 is related to the decelerating force that results from friction between tire and runway surface. Thus, this force only contributes when the aircraft is on the ground;

otherwise, its value is set to zero, i.e., flight phase between 100 ft AGL until the aircraft touches the ground. During the ground phase, the friction force can be divided into two types. One type is related to the pure friction between tire and runway in which the aircraft tires freely roll on runway's surface. The friction force results from runway surface condition and small deformations of the tire. The friction coefficient during this rolling phase is attributed as the rolling friction coefficient and denoted as μ_{roll}. Since the value of the rolling friction coefficient depends mainly on the tire and runway characteristics, its value is assumed constant during this phase.

The second type of friction force is related to the applied braking pressure. This braking pressure generates decelerating force, and its corresponding coefficient is called the braking friction coefficient (μ_{brk}). Thus, the friction force during phase 2 is expressed as,

$$F_F = \begin{cases} 0, & \text{no ground contact} \\ (\mu_{roll} + \mu_{brk}) \cdot F_N, & \text{with ground contact} \end{cases} \tag{6.4.28}$$

As observed from the FDM data, hydraulic pressure is used to actuate the brakes, and intuitively, this parameter can be used to infer the value of the braking friction coefficient. The model for the braking friction coefficient is expressed as,

$$\mu_{brk} = \mu_{brk,max} \cdot \frac{\tilde{p}_{brk}}{\tilde{p}_{brk,max}} \tag{6.4.29}$$

$$\tilde{p}_{brk} = p_{brk} - p_{brk,0} \tag{6.4.30}$$

Parameter $p_{brk,0}$ has to be inferred manually from the measurement since even without the application of braking, the parameters show non-zero values. Thus, the $p_{brk,0}$ varies in every flight and only relevant to that respective flight. To fully utilize Eq. (6.4.28), one more variable that needs to be modeled is the normal force that results from the aircraft weight and the runway surface. As indicated in Figure 6.11, this force can be modeled through vertical aerodynamic force, gravity and vertical propulsive force. However, it is found that modeling the normal force this way is hardly predictable as this force also depends on the dynamics of the runway slope and gear strut mechanism. Thus, it is decided to compute this force using the recorded vertical and horizontal accelerations ($a_{z_{\boldsymbol{B}},m}, a_{x_{\boldsymbol{B}},m}$). By applying Newton's second law on $z_{\boldsymbol{R}}$ and solving for normal force ($F_{\boldsymbol{N}}$) yields,

$$F_{\boldsymbol{N}} = -m \cdot a_{z_{\boldsymbol{R}}} + F_{\boldsymbol{A},z_{\boldsymbol{R}}} + F_{\boldsymbol{P},z_{\boldsymbol{R}}} + F_{\boldsymbol{G}} \tag{6.4.31}$$

The vertical acceleration $a_{z_{\boldsymbol{R}}}$ can be replaced by the measured accelerations in the Body-fixed frame, given as,

$$a_{z_{\boldsymbol{R}}} = a_{z_{\boldsymbol{B}},m} \cdot \cos\theta - a_{x_{\boldsymbol{B}},m} \cdot \sin\theta + \frac{F_{\boldsymbol{G}}}{m} \tag{6.4.32}$$

$$\tag{6.4.33}$$

Furthermore, Eq. 6.4.32 is substituted to Eq. 6.4.31 to get the normal force $F_{\boldsymbol{N}}$ model in terms of the measured accelerations. The final model of $F_{\boldsymbol{N}}$ is given by,

$$F_{\boldsymbol{N}} = -m \cdot [a_{z_{\boldsymbol{B}},m} \cdot \cos\theta - a_{x_{\boldsymbol{B}},m} \cdot \sin\theta] + F_{\boldsymbol{A},z_{\boldsymbol{R}}} + F_{\boldsymbol{P},z_{\boldsymbol{R}}} \tag{6.4.34}$$

The same as in phase 1, the output model is also needed to be modeled for phase 2. In this phase, four models are used, and their corresponding output will be compared later with the measurements. These four output models for phase 2 are given as,

$$a_{x_{\boldsymbol{B}},m} = -\frac{F_{\boldsymbol{F}}}{m} \cdot \cos\theta + \frac{F_{\boldsymbol{N}}}{m} \cdot \sin\theta + \frac{1}{m}(F_{\boldsymbol{A},x_{\boldsymbol{B}}} + F_{\boldsymbol{P}}) + b_x \tag{6.4.35}$$

$$a_{z_{\boldsymbol{B}},m} = -\frac{F_{\boldsymbol{F}}}{m} \cdot \sin\theta - \frac{F_{\boldsymbol{N}}}{m} \cdot \cos\theta + \frac{1}{m} \cdot F_{\boldsymbol{A},z_{\boldsymbol{B}}} + b_z \tag{6.4.36}$$

$$V_{GND,m} = V_{GND} \tag{6.4.37}$$

$$\theta_m = \theta \tag{6.4.38}$$

It should be noted that the output models are referred to the Body-fixed frame since all the respective measurements are also given in the Body-fixed frame.

6.4.3 Simultaneous Processing Several Flights

Applying the above-proposed estimation approach to the flight data at hand leads to the physically questionable estimates. It is due to the fact that the information content in a single flight is low compared to the number of parameters to be estimated. Thus, to increase the information content of the flight data, a combination of several flights to be processed simultaneously by the algorithm is proposed. Visually, the simultaneous processing of several flights adopted in this work is presented in Figure 6.12.

Generally, parameters involved in the model depends on various influencing factors such as aircraft configuration, weight, and environment condition. However, by selecting flights with the same configurations leads to a negligible contribution to the estimated parameters. In addition to that, aerodynamic parameters such as $C_{L_0}, C_{D_{min}}, C_{L_\alpha}$ are independent of the aircraft's weight. Thus, these parameters can be assumed to be the same for all utilized flights in the same group. However, parameter T_0 and τ cannot be assumed the same for all utilized flights as these parameters depend on the ambient condition. However, simultaneous estimation of the maximum thrust and exponent component for each flight has shown to lead to the high linear dependencies of the two parameters. This high dependency leads to inconsistent and physically questionable results. Thus, to reduce the effect of the dependency between the two parameters during the estimation process, the thrust exponent (τ) is assumed to be the same for all flights in the same group and only estimated once. The effect of the ambient condition is assumed to be condensed in the estimated T_0.

In phase 2, more parameters are required to be estimated as the aircraft is influenced by the landing gear force that results from the interaction between the tire and the runway surface. Phase 2 is also affected by the increment of the lift force due to the ground effect. All these contributions increase the complexity of the model utilized in phase 2. Thus, some simplifications are made for phase 2, i.e., during the air phase within phase 2, all relevant parameters that are estimated in phase 1 will be used as input parameters in phase 2. In this scenario, the number of parameters to be estimated in phase 2 is reduced, which consequently reduces the identifiability issue and increases the quality of the estimates, see Table 6.10.

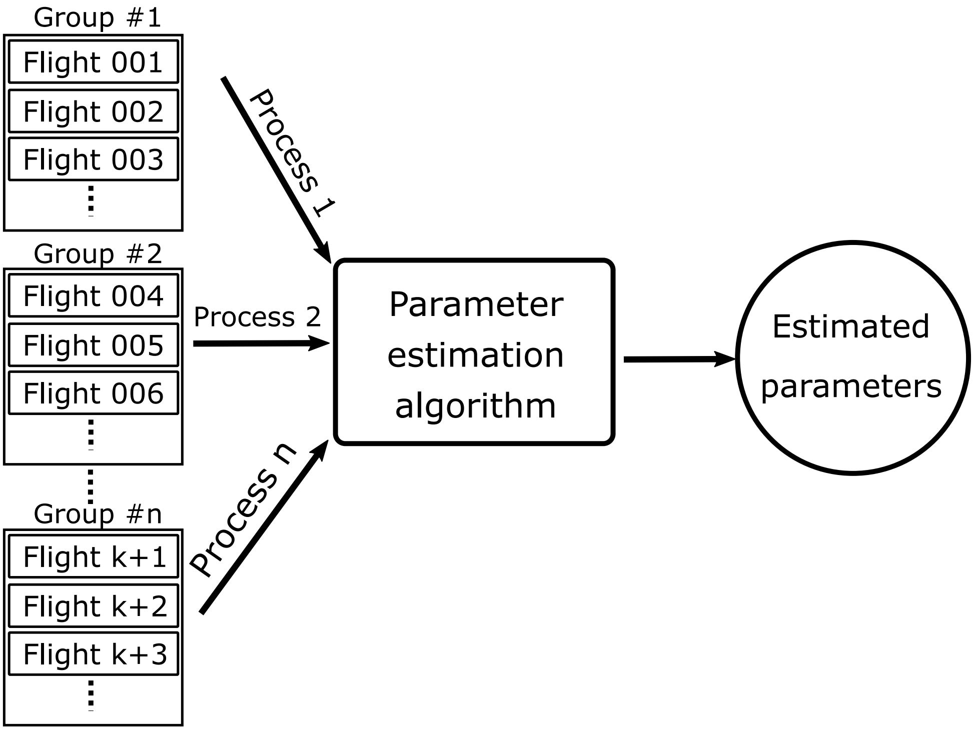


Figure 6.12: Simultaneous processing several flight

Furthermore, in connection with contribution in this dissertation, the strategy of enrichment of information content of FDM data through combining and processing several flight addresses the third contribution as presented in Chapter 1.

6.5 Algorithm Setups

In order to run the developed algorithm, some initial setups are required to be specified. These setups include the initial state variable ($\boldsymbol{x}_0$) of the model used in each phase, diagonal element of the process noise covariance matrix ($\mathbf{Q}_0$), diagonal element of the measurement noise covariance matrix ($\mathbf{R}_0$), diagonal element of the state error covariance matrix ($\mathbf{P}_0$), and parameters to be estimated ($\mathbf{\Theta}_0$). All these setups are specified in the following Sections 6.5.1 and 6.5.2.

6.5.1 Phase 1 Model Setup (1,000 - 100 ft AGL)

Based on the state model presented in Eqs. (6.4.1) - (6.4.3), there are three number of state variables including V_{TAS}, α, and θ. Since the state variables are measured, the initial values for the state variables can be approximated by computing the average of several points of the measured data [25]. In the present case, 10 data points of the measured data are used to compute the initial state values, while the values for diagonal elements of measurement noise covariance matrix $\mathbf{Q_0}$ and state error covariance matrix $\mathbf{P}_0$ are set with arbitrary number based on engineering judgment. Their corresponding initial state value, state error variances, and process noise variances are shown in Table 6.5. Note that, five values are computed for each state since the algorithm is fed by five flights. For $\mathbf{Q_0}$ and $\mathbf{P}_0$ covariance matrices, only only one values is given for each of the diagonal element of the corresponding matrix since for the same state variable, they are assumed to have the same value.

States	**Initial state value for each File ID**					**Diagonal Element**	
	B10675	**B10821**	**B10825**	**B10833**	**B10896**	$\mathbf{Q}_0$	$\mathbf{P}_0$
$V_{TAS}, m/s$	78.7064	79.7203	81.5070	76.8272	80.5394	2E-2	1E-2
α, rad	0.0856	0.0742	0.0683	0.0927	0.0781	1E-5	1E-5
θ, rad	0.0482	0.0328	0.0255	0.0579	0.0384	1E-5	1E-5

Table 6.5: Phase 1 - initial state value, process noise, and state error variances

Correspondingly, the input variables utilized in phase 1 include aircraft weight (m), engine rpm (N_1), and pitch rate (q). Since all input variables are recorded at a different rate, the parameter whose sampling is low is up-sampled by using cubic interpolation [25]. While, the output variables include the true airspeed (V_{TAS}), angle of attack (α), pitch angle (θ), acceleration in x-Body axis (a_x), and acceleration in z-Body axis (z_B). Each of these variables is presented in Table 6.6 along with their corresponding initial measurement noise variances. Note that, the values presented in Table 6.6 represents the diagonal element of the initial measurement noise covariance matrix $\mathbf{R}_0$ for file id `B10675` which is also used for the other four flights.

The remaining setup required in phase 1 is the initial unknown parameters. These unknown parameters are presented in Table 6.7 along with their corresponding initial values.

As discussed in Section 6.4.3, simultaneous data are processed by the algorithm in order to increase the information content in the data. Several flight data are selected, and respective

Output	Description	Unit	$\mathbf{R}_0$
V_{TAS}	True airspeed	m/s	1E-4
α	Angle of attack	rad	1E-6
θ	Pitch angle	rad	1E-6
a_x	Acceleration in x-Body axis	m/s^2	1E-3
a_z	Acceleration in z-Body axis	m/s^2	1E-3

Table 6.6: Phase 1 - output variables

Parameter	Description	Unit	Initial value Θ_0
C_{L_0}	Lift coefficient at zero angle of attack	-	1.2000
C_{L_α}	Gradient of lift coefficient over angle of attack	rad^{-1}	4.2000
$C_{D_{min}}$	Minimum drag coefficient	-	0.0358
T_0	Maximum thrust	N	2.000 × 1E6
τ	Thrust model exponent	-	0.9000
b_x	Accelerometer bias in x-Body axis	ms^{-2}	0.0000
b_z	Accelerometer bias in z-Body axis	ms^{-2}	0.0000

Table 6.7: Phase 1 - unknown parameters initialization

parameters (input and measurement variables) are horizontally stacked. These data sets are fed to the algorithm to be simultaneously processed. The combination of multiple data from 2-7 flights is applied to the algorithm. From the experiment, it is found that the combination of five flights provides better performance in terms of stability and the execution time. Thus, grouping files into a group consisting of five files are adopted in the implementation. The selection of the files is performed sequentially, i.e. group 1 (flight 1-5), group 2 (flight 6-10), etc. Among the seven unknown parameters as presented in Table 6.7, three of them are flight-dependent which are estimated for each flight. These three parameters include T_0, b_x, and b_z, see Section 6.4.3. The remaining parameters are assumed to be constant for each flight in the same group. All setups are already specified for phase 1. The corresponding results are presented in Section 6.6.1.

6.5.2 Phase 2 Model Setup (100 ft AGL - end of Landing Phase)

The same setups are required by the phase 2 model. These setups include initialization of the initial state value $\boldsymbol{x}_0$, initial state error covariance $\mathbf{P}_0$, measurement noise covariance $\mathbf{R}_0$, process noise covariance $\mathbf{Q}_0$, and initial unknown parameters $\boldsymbol{\Theta}_0$. All these setups are presented in the following Table 6.8 - 6.10.

State variables for phase 2 include ground speed (V_{GND}) and pitch angle (θ) and the respective initial values for V_{GND_0} and θ_0 are computed by taking the average of several measurement data points. In the present work, 10 data points are used to compute the initial state variables. The initial state error and process noise covariances are specified as presented in Table 6.8.

States	Initial state value for each File ID					Diagonal Element	
	B10675	B10821	B10825	B10833	B10896	$\mathbf{Q}_0$	$\mathbf{P}_0$
$V_{GND}, m/s$	76.4610	72.0266	74.1692	74.7918	86.7342	4E-2	2E-1
θ, rad	0.0482	0.0393	0.0613	0.0526	0.0391	1E-5	1E-5

Table 6.8: Phase 2 - initial state value, process noise, and state error variances

While, the initial measurement noise covariance $\mathbf{R}_0$ is given in Tabel 6.9. The last setup for phase 2 is related to the initial values of the unknown parameters $\mathbf{\Theta}_0$. However, as discussed in the previous section, some parameters estimated in phase 1 are used in phase 2, particularly when the aircraft is still in the air. In this case, these parameters are treated as input instead of parameters to be estimated. The values of these parameters are constant during the selected time window. Furthermore, some parameters that are flight-dependent, e.g. friction coefficients (rolling and braking friction coefficients) are estimated in each flight, while the remaining parameters are assumed the same for all flights within the same group. The same as in phase 1, five flights are used simultaneously for increasing the information content in the data. Table 6.10 summarizes the unknown parameter for phase 2 and their respective initial values.

State	Description	Unit	$\mathbf{R}_0$
V_{GND}	Ground speed	ms^{-1}	1E-4
θ	Pitch angle	rad	1E-6
a_x	Acceleration in x-Body axis	ms^{-2}	1E-3
a_z	Acceleration in z-Body axis	ms^{-2}	1E-3

Table 6.9: Phase 2 - measurement variables and corresponding initial noise variances

All the required parameters for phase 2 have been specified. The corresponding results are presented in Section 6.6.2.

6.6 Results

In this section, results obtained from the application of the developed algorithm to the FDM data on phase 1 and phase 2 are presented. The estimated parameters along with statistical

State	Description	Unit	Θ_0
$C_{L_0,GND}$	Lift coefficient at zero AoA during on ground	-	2.000
$\Delta C_{L,S}$	Lift coefficient change due to spoiler deployment	-	1.000
$\Delta C_{D,S}$	Drag coefficient change due to spoiler deployment	-	0.050
r	Reverse thrust ratio	-	0.500
μ_{roll}	Rolling friction coefficient	-	0.050
$\mu_{brk,max}$	Maximum realized braking friction coefficient	-	0.100
$C_{L_0,AIR}$	Lift coefficient at zero AoA during air phase	-	given from phase 1
C_{L_α}	Gradient of lift coefficient over angle of attack	rad^{-1}	given from phase 1
$C_{D_{min},AIR}$	Minimum drag coefficient during air phase	-	given from phase 1
e	Oswald number	-	given [158]
T_0	Maximum thrust	N	given from phase 1
τ	Thrust exponent model	-	given from phase 1

Table 6.10: Phase 2 - unknown parameters and their corresponding initial values

analysis will be elaborated for each flight phases. For demonstrating the capability of the developed algorithm, one flight group consisting of five different flights will be used. At the end of each section of phase 1 and phase 2, the results for the whole flight investigated in this work are presented in the histogram plot along with the reference values of the estimated parameters.

Linked with the contributions presented in Chapter 1, the estimated parameters presented in this section addresses the first contribution in which the lack of the existing FDM programs in providing parameters other than the recorded or the computed parameters is solved through applying the developed method to FDM data. Section 6.6.1 - 6.6.2 present the estimated parameters in the two selected flight phases.

6.6.1 Phase 1 (1,000 - 100 ft AGL)

The algorithm requires five iterations until the estimated parameters reach the convergence values. Overall, all parameters start to converge at the third iteration and slightly change until the end of the iterations. The history plot of the estimated parameters over the iterations, along with their standard deviations, is depicted in Figure 6.13. As a final result, the numerical values of the estimated parameters are presented in Table 6.11. Note that, the value inside the bracket symbol represents the standard deviation of the respective parameter.

As discussed previously in Section 6.4.3, only parameters related to engine and biases are estimated for each flight. These estimates are shown in each column in Table 6.11, while the other four parameters are flight-independent and assumed to be constant for each flight.

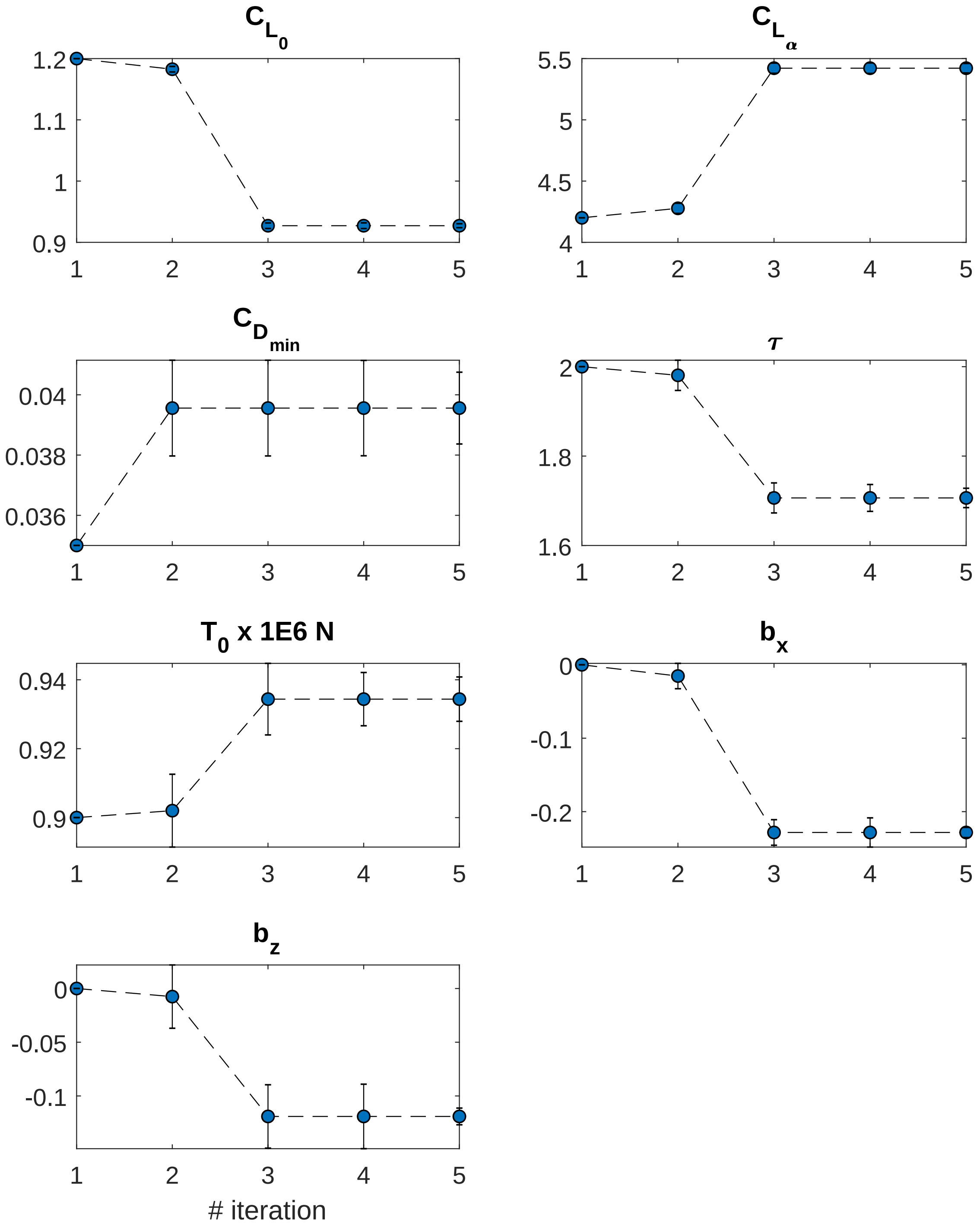


Figure 6.13: Phase 1 - estimated parameter vs iteration, flight B10675

Thus, the blank columns in Table 6.11 indicates that the respective parameters refer to flight 'B10675' (second column). Furthermore, parameters corresponding to noise statistics and initial state error variances are also estimated. There are three parameters to be estimated

Parameter	B10675	B10821	B10825	B10833	B10896
C_{L_0}	0.92711 (0.0032)	-	-	-	-
C_{L_α}	5.42160 (0.0389)	-	-	-	-
$C_{D_{min}}$	0.03956 (0.0012)	-	-	-	-
τ	1.70630 (0.0217)	-	-	-	-
T_0	0.93438 (0.0064)	0.82717 (0.0061)	0.78314 (0.0085)	0.86805 (0.0080)	0.72524 (0.0071)
b_x	-0.22836 (0.0069)	-0.23572 (0.0053)	-0.24811 (0.0069)	-0.21286 (0.0084)	-0.23247 (0.0064)
b_z	-0.11906 (0.0078)	0.04674 (0.0079)	0.12953 (0.0075)	-0.01059 (0.0090)	0.18514 (0.0085)

Table 6.11: Phase 1 - estimated parameters

for each $\mathbf{P}_0$ and $\mathbf{Q}$ in which each estimate related to state variables V_{TAS}, α, and θ, while five parameters related to the measurement variables corresponding to the diagonal element of the measurement noise covariance matrix $\mathbf{R}$. The final estimates of each initial state error variance, process noise and measurement noise variance parameters are presented in Table 6.12 - 6.14.

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{P}_0(1,1)$	m^2/s^2	8.059E-07	1.708E-06	2.924E-06	8.154E-06	4.956E-06
$\mathbf{P}_0(2,2)$	rad^2	1.027E-09	1.448E-09	4.852E-10	1.119E-09	3.014E-09
$\mathbf{P}_0(3,3)$	rad^2	3.428E-08	5.006E-09	7.485E-08	2.880E-08	8.445E-09

Table 6.12: Phase 1 - diagonal elements of $\mathbf{P}_0$ estimation

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{Q}(1,1)$	m^2/s^2	2.203E-03	3.474E-03	4.701E-03	8.274E-03	9.041E-03
$\mathbf{Q}(2,2)$	rad^2	8.882E-07	3.573E-06	3.315E-06	3.422E-06	4.540E-06
$\mathbf{Q}(3,3)$	rad^2	5.824E-07	3.602E-06	3.793E-06	2.557E-06	5.807E-06

Table 6.13: Phase 1 - diagonal elements of $\mathbf{Q}$ estimation

Correspondingly, the model outputs are simulated by utilizing all the estimated parameters and noise statistics. The model outputs are compared with the measured variables, and the corresponding coefficient of determination R^2 is computed. The results are depicted in Figure 6.14. As indicated in Figure 6.14, the model output fit the measurement well with a coefficient of determination greater than 90%, except for the vertical acceleration variable where the coefficient of determination is around 37%. By visually looking on the vertical acceleration plot (at the bottom part of Figure 6.14), the fluctuations of the measurements data are dominated by noise, in which the contribution of aircraft dynamics to the vertical acceleration is very low during this selected phase. This behavior can be seen from the low

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{R}(1,1)$	m^2/s^2	7.615E-07	1.614E-06	2.772E-06	7.742E-06	4.686E-06
$\mathbf{R}(2,2)$	rad^2	7.673E-10	1.077E-09	3.561E-10	8.319E-10	2.241E-09
$\mathbf{R}(3,3)$	rad^2	4.371E-08	4.841E-09	8.407E-08	2.795E-08	8.106E-09
$\mathbf{R}(4,4)$	m^2/s^4	8.294E-05	1.008E-05	1.410E-04	2.175E-04	1.173E-04
$\mathbf{R}(5,5)$	m^2/s^4	3.342E-03	3.509E-03	2.601E-03	8.812E-03	6.755E-03

Table 6.14: Phase 1 - diagonal elements of $\mathbf{R}$ estimation

range of the fluctuation values of the measured vertical acceleration. Since the goal of the curve-fitting is not to fit the model output with the noise but with the dynamic motion of the system as observed in the data, having a low coefficient of determination in this particular case is physically reasonable.

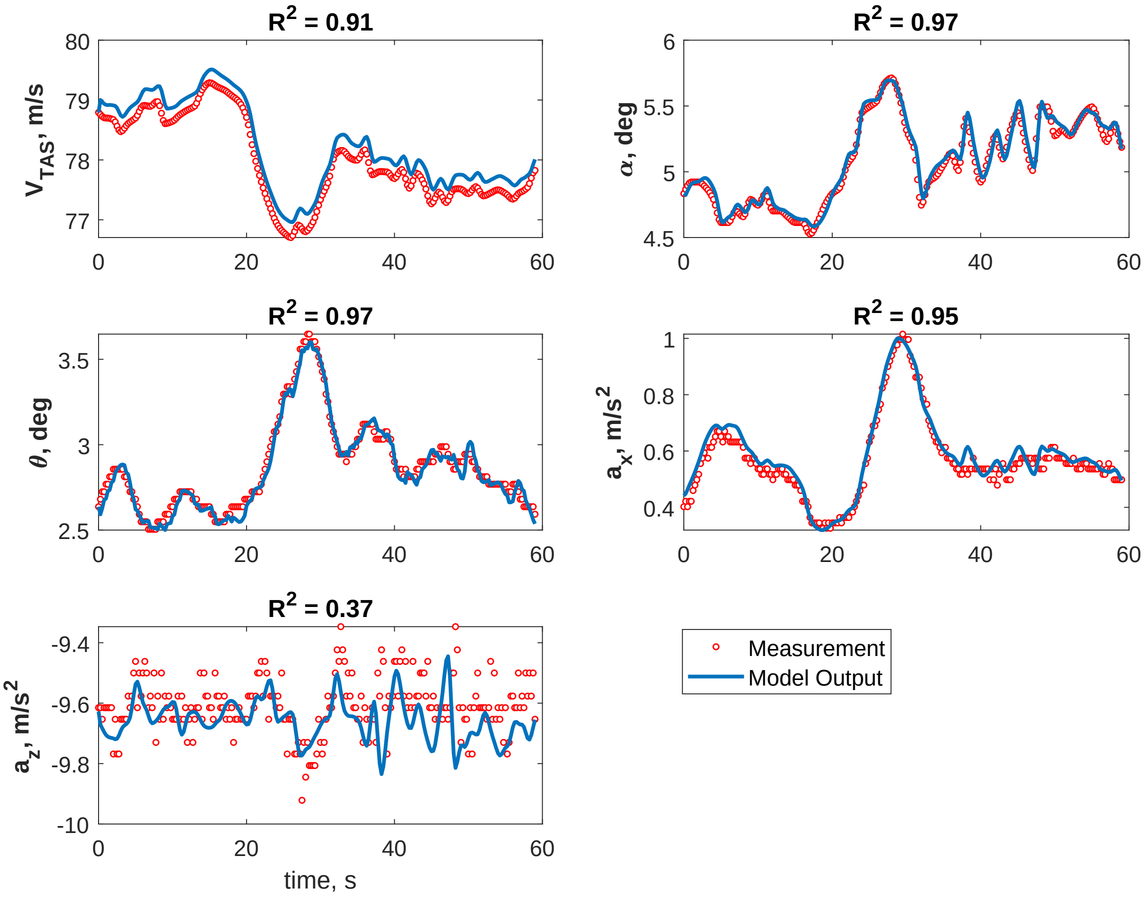


Figure 6.14: Phase 1 proof of match - flight B10675

Furthermore, as discussed in Section 6.4.3, the estimated parameters obtained from a combination of several flights should also have a good predictive capability for the other flights within a group. In phase 1, four parameters are assumed to be flight-independent, while the other three parameters are flight-dependent. With the same four and three different parameters, the model is simulated again by using different flight (B10821), and the respective result is depicted in Figure 6.15. As shown in the figure, a good predictive capability of the model and the corresponding estimated parameters is achieved. This is demonstrated by the high coefficient of determinations, which also resemble the results obtained from flight B10675.

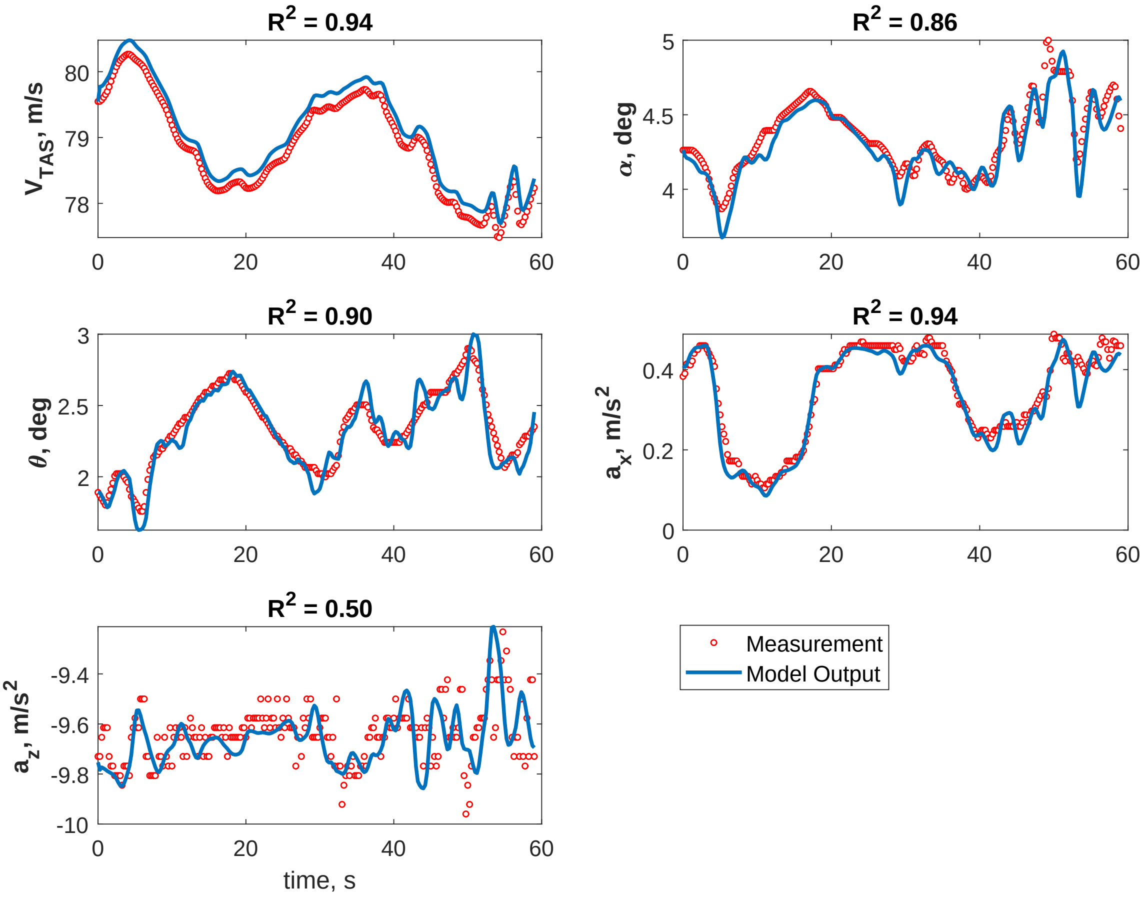


Figure 6.15: Phase 1 proof of match - flight B10821

Overall, with the high coefficient of determination values, the proposed model and the respective estimated parameters are able to capture the behavior of the aircraft dynamics within the selected phase. The other proof of match plots corresponding to the remaining three flights are presented in Appendix C. In the same way, the developed algorithm is applied to the whole data set (1,480 flights), and the estimated parameters are stored in a database for later use in other analysis [12]. To summarize, the estimated parameters are visually plotted in the histogram plots in Figure 6.16.

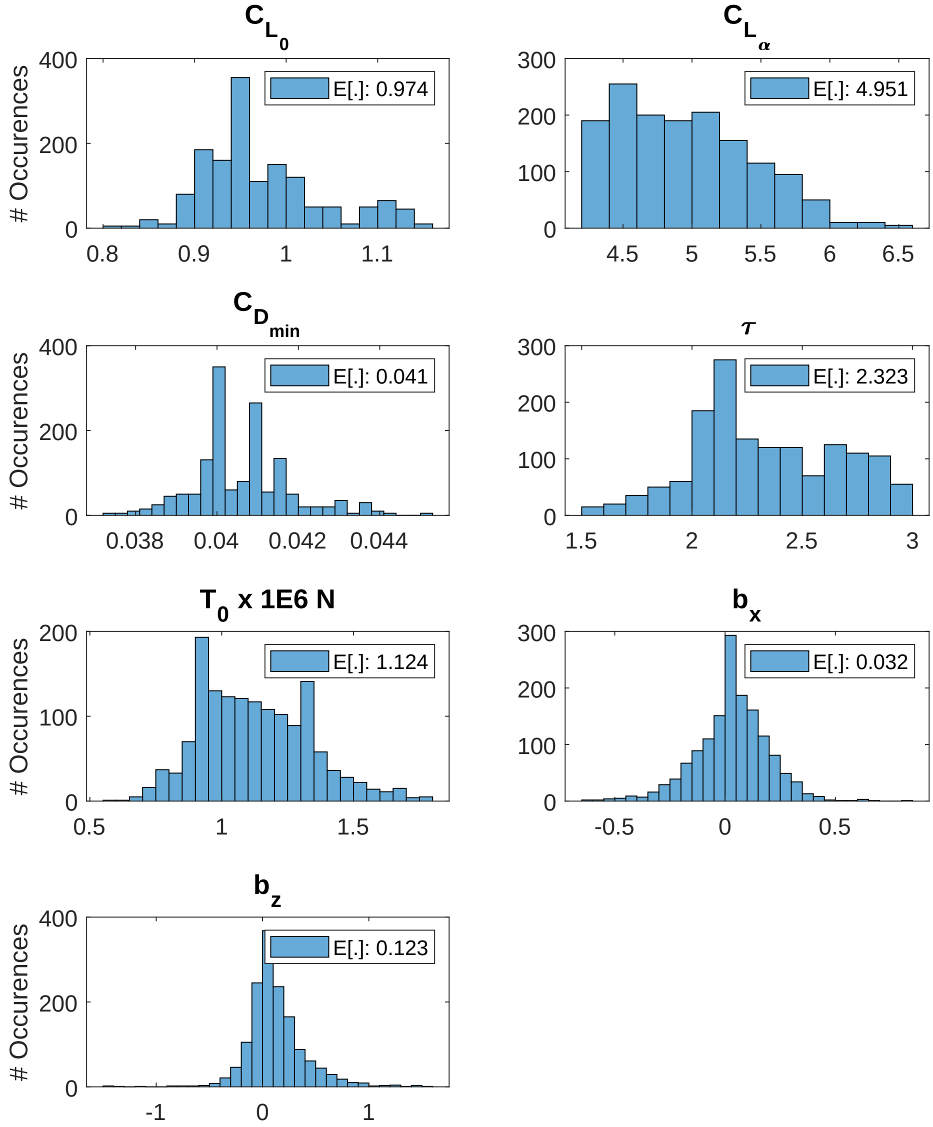


Figure 6.16: Phase 1 - estimated parameters for 1,480 flights, runway A

For verification, the estimated parameters are compared with some reference values, which can be found in some sources [161]. These comparison are presented in Table 6.15. As indicated, the estimated parameters, e.g., T_0 provides a small deviation with respect to the reference value ($\pm$ 8%). Unfortunately, not all of the estimated parameters can be compared with the reference values as some of them are not publicly available.

Parameter	Unit	Θ_{ref}	E[Θ]	Deviation	Source
C_{L_0}	-	N/A	0.974	N/A	N/A
C_{L_α}	rad^{-1}	N/A	4.951	N/A	N/A
$C_{D_{min}}$	-	N/A	0.041	N/A	N/A
τ	-	N/A	2.323	N/A	N/A
T_0	×1E6 N	1.040	1.124	8.08 %	[161]

Table 6.15: Phase 1 - expected value of the estimated parameters vs reference values

6.6.2 Phase 2 (100 ft AGL - end of Landing Phase)

In phase 2, the developed algorithm needs more iteration than in phase 1. In total, the algorithm takes six iterations until the estimated parameters converged. Each parameter starts to converge at the fifth iteration and change slightly until the end of the iterations. Figure 6.17 depicts the history of the estimated parameters for one flight over a number of iterations. Note that, the value for $\Delta C_{L,S}$ in Figure 6.17 is presented in the positive sign $(+)$ as this parameter in Eq. (6.4.20) has been modeled as a decreasing factor by directly putting the negative sign (-) in front of the parameter. As a final result, Table 6.16 presents all the estimated parameters for all five flights within the same group in which parameters related to friction coefficients are estimated in each flight. In estimating the rolling friction coefficient, the estimated values show values close to zeros in some flights, which is caused by a limited amount of data available during the ground roll phase.

Parameter	B10675	B10821	B10825	B10833	B10896
$C_{L_0,GND}$	1.9029 (4.717E-03)	-	-	-	-
$\Delta C_{L,S}$	1.4348 (8.316E-03)	-	-	-	-
$\Delta C_{D,S}$	0.0671 (7.410E-04)	-	-	-	-
r	0.6910 (3.746E-03)	-	-	-	-
μ_{roll}	∼ 0.00 (5.757E-04)	∼ 0.00 (9.437E-04)	0.0126 (9.437E-04)	∼ 0.00 (3.272E-04)	0.01196 (8.388E-04)
μ_{brake}	0.1997 (7.387E-04)	0.2017 (1.225E-03)	0.1966 (6.306E-04)	0.23235 (6.341E-04)	0.20833 (1.559E-03)

Table 6.16: Phase 2 - estimated parameters

Furthermore, the initial state error covariance $\mathbf{P}_0$, process noise covariance matrix $\mathbf{Q}$, and measurement noise covariance matrix $\mathbf{R}$ are also estimated. Their corresponding estimates for the five flights are presented in Table 6.17 - 6.19.

With all the estimated parameters, the model outputs are generated and compared with the measurement variables (flight B10675) as depicted in Figure 6.18. Coefficient of determination (R^2) for each variable is computed. The resulted values which are higher than 80% indicate that the model utilized in this phase is able to capture the dynamic behavior of the aircraft. As shown in Figure 6.18, the model outputs fit well with the measurement variables. Some

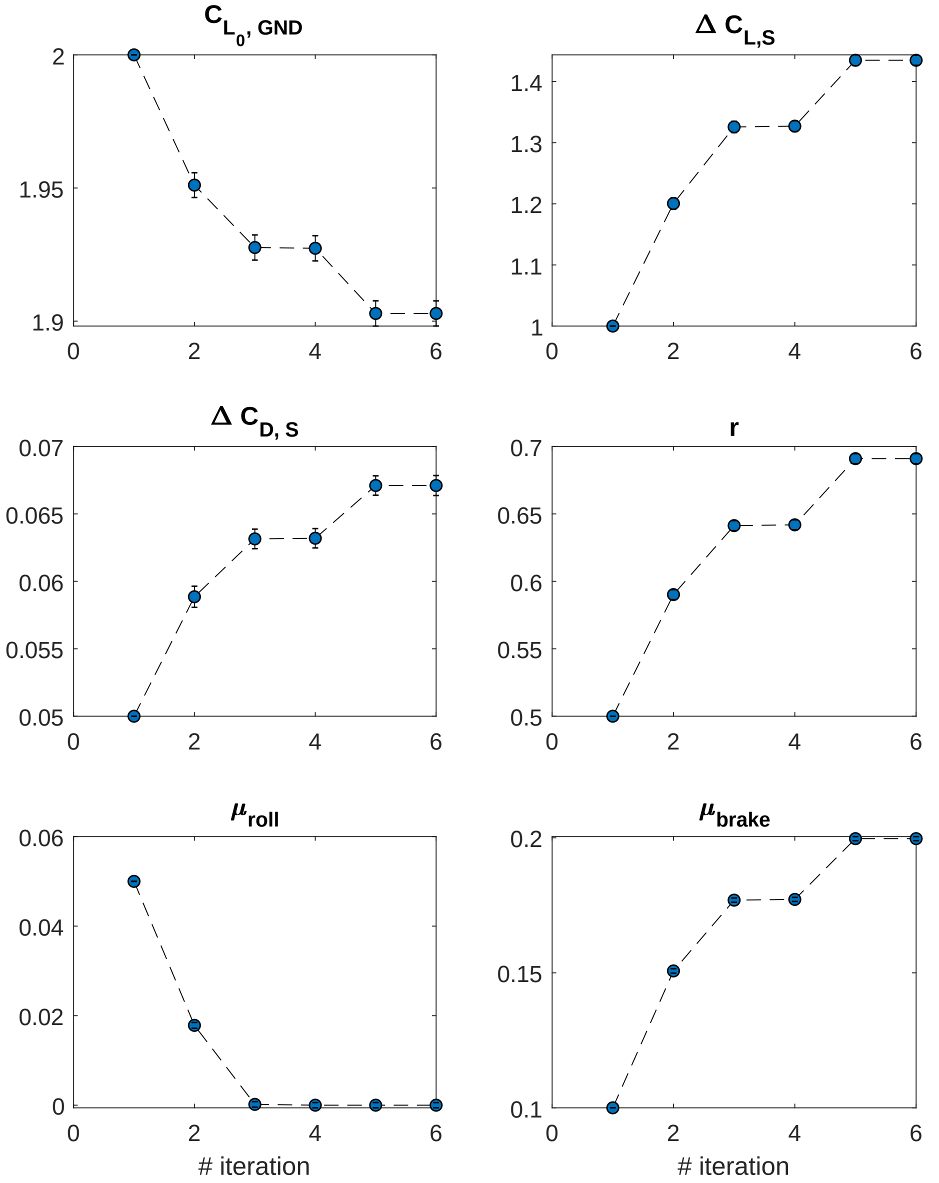


Figure 6.17: Phase 2 - estimated parameter vs iteration, flight B10675

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{P}_0(1,1)$	m^2/s^2	1.576E-04	3.0613e-03	7.010E-04	5.026E-03	8.642E-03
$\mathbf{P}_0(2,2)$	rad^2	1.964E-07	1.5480e-06	1.610E-07	1.598E-06	7.026E-07

Table 6.17: Phase 2 - diagonal elements of initial $\mathbf{P}_0$ estimation

peaks in the measurement data that might result from the noise are not tracked by the model output; instead, the model generates its trajectories based on the estimated parameters and the corresponding noise statistics. For the vertical acceleration's (a_z) proof of match, its

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{Q}(1,1)$	m^2/s^2	1.837E-03	3.568E-03	8.171E-03	5.858E-03	1.007E-02
$\mathbf{Q}(2,2)$	rad^2	2.289E-06	1.804E-06	1.877E-06	1.862E-06	8.188E-07

Table 6.18: Phase 2 - diagonal elements of $\mathbf{Q}$ estimation

Parameter	Unit	B10675	B10821	B10825	B10833	B10896
$\mathbf{R}(1,1)$	m^2/s^2	5.064E-07	1.560E-07	7.993E-07	4.385E-07	2.362E-06
$\mathbf{R}(2,2)$	rad^2	1.083E-04	1.456E-04	1.292E-04	2.414E-04	2.170E-03
$\mathbf{R}(3,3)$	m^2/s^4	3.630E-02	5.256E-02	4.837E-02	4.741E-02	6.035E-02
$\mathbf{R}(4,4)$	m^2/s^4	3.543E-09	5.568E-09	5.613E-09	4.760E-09	7.842E-09

Table 6.19: Phase 2 - diagonal elements of $\mathbf{R}$ estimation

corresponding coefficient of determination is a bit higher than in phase 1. One possible reason behind this R^2 increase is due to the dynamic motion of the aircraft during this phase which is captured in the measured data as indicated by a high fluctuation at around 18 - 22 seconds.

As in phase 1, the quality of the model predictive is also tested by simulating the developed model for the other flights with the same group. The corresponding result is shown in Figure 6.19 and the other three flights' results are shown in Appendix C. As shown in Figure 6.19, the developed model, along with the estimated parameters, have a good model predictive capability, as shown by the good fit between the predicted output and the measurement data.

The same procedures are applied to the whole selected flights (1,480 flights) in which the flights are clustered into groups consisting of five flights. The algorithm is then applied to each group, and the corresponding results are presented in histogram plots in Figure 6.20. In phase 2, the estimated parameters cannot be compared with a reference value as no source is publicly available for the respective parameters. Thus, the only qualitative analysis provided here is considered from the physical meaning point of view. The C_{L_0} parameter estimated during phase 2 is higher than the C_{L_0} value in phase 1. This high value might results from the ground effect since, in phase 2, the aircraft is exposed to the ground effect phenomena, which contributes to the increment of the lift coefficient. In this regard, the values indicated by both situations align with physical reality.

Furthermore, the friction coefficient parameter also represents an acceptable-physical value as the rolling friction coefficient (μ_{roll}) is lower than the friction coefficient that results from the braking pressure (μ_{brake}). The change of the lift coefficient due to spoiler deployment ($\Delta C_{L,S}$)

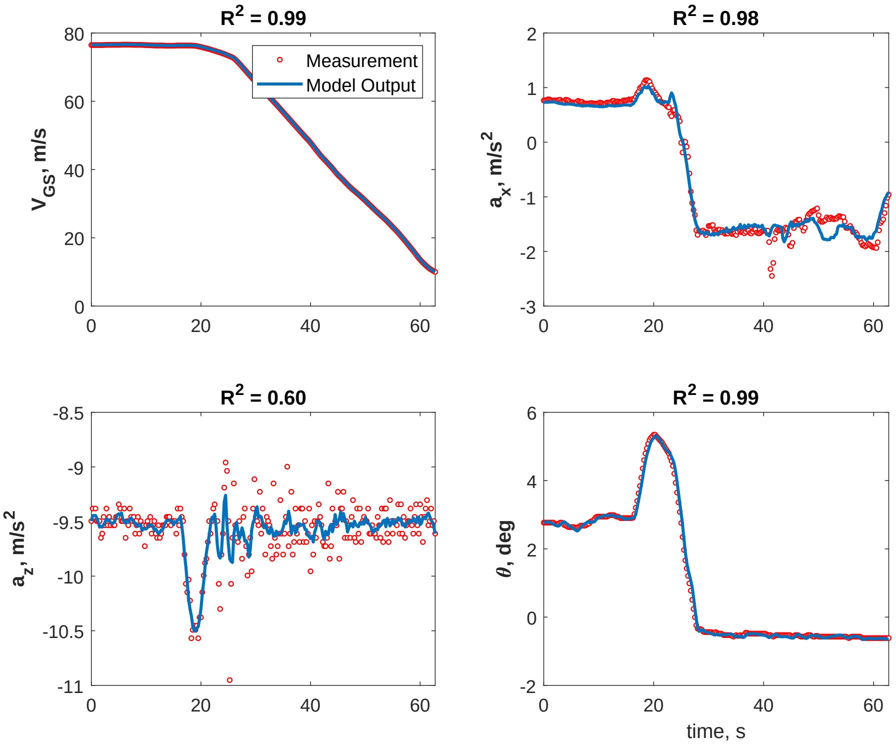


Figure 6.18: Phase 2 proof of match - flight B10675

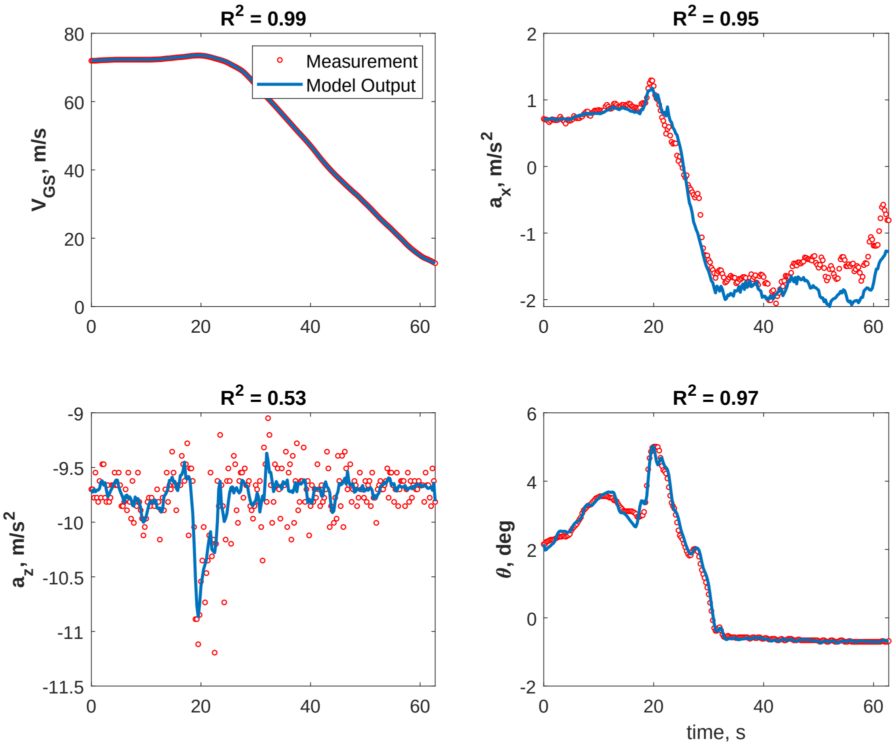


Figure 6.19: Phase 2 proof of match - flight B10821

parameter also indicates a decreasing factor to the total lift with an acceptable value. This is supported by the fact that the decrease of the total lift due to $\Delta C_{L,S}$ does not make the total lift coefficient negative, but some lift is still available as the aircraft is still in the air.

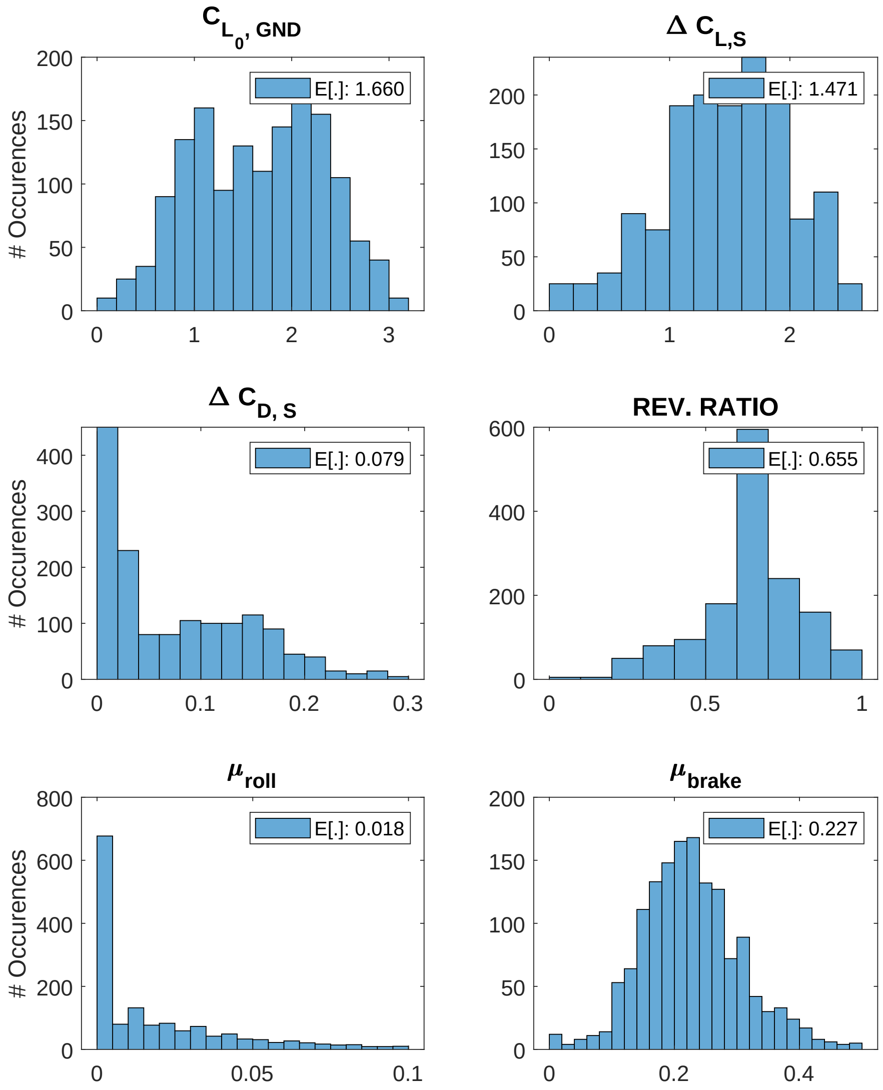


Figure 6.20: Phase 2 - estimated parameters for 1,480 flights, runway A

6.7 Computational Performance Aspects

Several performance aspects of the developed algorithm are noted during the application on the FDM data. They are listed and elaborated as follows.

- Execution time for a group consisting of five files during the final approach phase takes around 1-1.5 minutes. Thus, for 1,480 files clustered into 296 groups require around 7.4 hours to process all the data. However, by employing multi-core processing, e.g., 8-core processors, the execution time reduces to just around an hour. For phase 2 window, the algorithm requires less time than in phase 1. With the same configuration as phase 1, a group consisting five files is executed in less than a minute. Thus, in total, the algorithm takes around five hours in a single core processor or around 0.65 hour for 8-core processors.

- The initial values of the estimated parameters determine the time needed by the algorithm to process the data. By setting the initial values with random numbers or a bit far away from the convergence values, the algorithm needs around 22-25 iterations with an execution time around 2-2.5 minutes per group. Even though the parameters will achieve the same convergence values as with the designated initial values, it is not recommended for practice as the number of data to be processed is large. Thus, for practical purpose, once the convergence values are known and tested for several groups, it is recommended to set the initial values not to be too far away from the convergence values in order to reduce the execution time.
- In some data, it is found that the measurement noise covariance matrix $\mathbf{R}$ produced by the Myers-Tapley algorithm is a singular matrix which terminates the algorithm before the convergence criteria of the parameters achieved. To ensure the stability of the algorithm, the singular matrix is replaced with the estimates obtained from Fourier Smoothing [75]. Estimating the measurement noise covariance through the Fourier smoothing technique is proposed by Morelli and widely applied in the flight vehicle system identification field [25, 75]. This technique is based on the signal properties and independent of the system model. Thus, this technique is very practical and yet yield accurate noise covariance information. This dissertation adopts the Matlab function developed by Morelli and embedded in the developed algorithm to produce the measurement noise covariance in case the Myers-Tapley algorithm fails to produce a non-singular covariance matrix.

Chapter 7

Summary and Perspective

In this dissertation, a novel system identification method for estimating unrecorded parameters based on FDM data is introduced. The developed method is based on a heuristic approach which combines the classical output error and filtering approaches. Applying the classical system identification method to FDM data tends to bring the algorithm into instability issue as well as poor estimates due to the low information content of FDM data. Moreover, FDM data contains noises, both from process and measurement, as the data are recorded on a daily operational flight. In the system identification field, the statistics of the process and measurement noises are also required to be quantified by the algorithm. The noise statistics parameters, along with the system parameters, lead to an increase in the number of parameters to be estimated. Consequently, too many parameters being estimated may lead to poor estimates and instability computation that might result from linear dependency and low information content data. To solve these issues, the new algorithm contributed by this dissertation divides the estimation process into two stages. The first stage is dedicated only for estimating noise related parameters $(\mathbf{Q}, \mathbf{R})$ through the Rauch-Tung-Striebel (RTS) smoother algorithm, while the second stage is dedicated for estimating the system-related parameters (β) by applying the modified Filter Error Method and using the noise statistics parameters which are provided by the first stage. In this second stage, the RTS smoother is also used instead of the EKF algorithm as the RTS smoother provides states with lower variances. With the steps separation, the algorithm runs smoothly with less instability issue.

Furthermore, to increase the information content in FDM data, this dissertation demonstrates a new strategy of simultaneously processing several flights. A group consisting of five flights are fed to the algorithm and processed simultaneously to estimate the unknown parameters. With this strategy, the information content in the data is expected to increase and consequently leads to physically reasonable estimates with good statistical properties.

The parameters that result from the application of the developed algorithm to FDM data can serve to many purposes. For example, in an event or incident/accident investigation, the estimated parameters may provide complementary data for analysis. With the current trend in airline safety, many airlines are now interested in the data-driven predictive analysis approach [12, 13]. Such analysis usually involves a flight-physics model which requires a quantification of the involved parameters. Some of the parameters cannot be obtained directly from the recorded data, nor can they be algebraically computed since no existing FDM programs utilized parameter estimation method. In this case, the developed algorithm fills the gaps by providing the required parameters to the model. As a summary of the work conducted in this dissertation, the following paragraphs briefly summarize and present the main ideas from every chapter.

Chapter 1 starts by introducing some general aspects of the works, covering the motivation, state of the art, the mission objective, and its contributions. Chapter 2 then elaborates FDM data utilized in this work, including the history and evolution of the civil aircraft flight recorder, operational and regulatory aspects, data acquisitions and transfer, and the utilization of the FDM data for enhancing safety through the FDM program performed by the airlines. Furthermore, the FDM program utilized by the airlines is also briefly discussed concerning the lack of such a program to produce some parameters that might be required in a specific incident/accident analysis or flight safety-related monitoring. One important point discussed in this chapter is related to the raw flight data in which the flight parameters are stored in a binary format and the technique required for decoding the raw data into an engineering format. Thus, several aspects related to the decoding process are elaborated covering the introduction of a dataframe, and its utilization for decoding the raw data. This description concerning the dataframe serves as a basis for the development of the decoding program that is employed to decode all the utilized data used in this work. The decoding process is given in Chapter 6 as part of the data preprocessing step. At the end of Chapter 2, the characteristics of the FDM data, particularly the characteristics related to the technical aspects such as sampling rate, information content, and sensor information, are presented.

The 6-degree of freedom of the nonlinear aircraft rigid body equation of motion is introduced in Chapter 3 in which the underlying model will serve as a basis for developing a simplified model used in Chapter 6. This dynamic motion model takes form as the first ordinary differential equation, which is derived based on Newton's second law. The resulted model is further called the state model of the aircraft that represents the aircraft's dynamic motions with respect to the time domain. All forces involved in the model, including propulsive, aerodynamics, and gravity forces are also presented in Chapter 3. The aerodynamic force is further elaborated in details by introducing the aerodynamic model in a non-dimensional form. This model is expanded through the Taylor series expansion by taking all contributing factors affecting the forces such as the angle of attack, Mach number, and control surfaces of the aircraft.

Furthermore, since this dissertation deals with flight data measured onboard civil aircraft, the so-called measurement models are presented. These measurement models relate between the state variable of the aircraft's motion and the state variable measured by the sensors equipped in the aircraft. This model allows the application of the estimation techniques to infer the aircraft's characteristic by estimating parameters involved in the model.

Chapter 4 introduces the system identification theory, which underlies the method developed in this dissertation. Two estimation theories are elaborated, which are the Maximum Likelihood principle and the Kalman Filter. Based on these two theories, several classical parameter estimation techniques used in the aircraft system identification field are briefly introduced. These techniques include the Equation Error Method, Output Error Method, Filter Error Method, and the Kalman Filter method tailored for a dual-estimation problem. The derivation of each technique and their applicability on different systems are also elaborated in this chapter. Among all techniques elaborated in this chapter, the Filter Error Method provides a natural approach and the capability of dealing with linear or nonlinear system model affected by the process and measurement noises. Even though the filtering approach, e.g., the Extended Kalman Filter algorithm, is also able to deal with such a system, but it is not initially designed for parameter estimation purpose. Thus, the application of this technique for parameter estimation problem requires some adjustments.

Between these two advanced classical estimation techniques, it seems natural to adopt the Filter Error Method for the data at hand. However, with the Filter Error Method, more parameters are required to be estimated by the algorithm. These parameters include the system parameters and the process and measurement noise statistics. The increase of the number of parameters to be estimated leads to the increase of the linear dependencies among the parameters which consequently deteriorates the identifiability of their values. Furthermore, with the data quality at hand, applying the Filter Error Method to such data leads to the instability of the algorithm. Motivated by these facts, a new method that is capable of dealing with such a system as well as producing reasonable estimates is required to be developed.

In Chapter 5, a heuristic approach for the parameter estimation problem is presented. This method is tailored to be capable of dealing with FDM data. Basically, it is a modification of the Filter Error Method where parameters related to the noise statistics are estimated separately from the system parameters. Two main loops are defined in this approach; one loop is dedicated for estimating the noise statistics through the RTS smoother, while the second loop is designed for estimating the system parameters through the modified Filter Error Method. In estimating the noise statistics, the algorithm iteratively adjusts the noise statistics parameters until the convergence criteria of the cost function are achieved. This process is done sequentially by estimating the process noise covariance until the convergence criterion is achieved and then followed by estimating the measurement noise covariance. When the convergence criteria are achieved for both noise statistics, the algorithm jumps to the second loop to estimate the system parameters. In the first loop, it should be noted that the initial state error covariance $\mathbf{P}_0$ also contributes to the quality of the estimates. Thus, in this first loop, the algorithm computes the initial state error covariance matrix by using several points of the measured variables.

With the computed $\mathbf{P}_0$ and the estimation of the noise statistics $\mathbf{Q}$ and $\mathbf{R}$, the data are further processed in the second loop for system parameter estimation. In the second loop, the modified Filter Error Method is used to estimate the system parameters. The state is estimated through the RTS smoother, while the system parameters are adjusted iteratively through the gradient-based optimization approach until the convergence criterion of the defined cost function is achieved. Since the noise statistics are provided through the first loop, the burden of the algorithm in the second loop is highly reduced, which leads to the stability of the algorithm. Compensating process and measurement noise step which is incorporated in the original Filter Error Method is discarded in the present method as both noise statistics are estimated in the separate step. This reduction simplifies the steps conducted in the developed method.

At the end of Chapter 5, the developed algorithm is verified by applying it to simulated data where all parameters and noise statistics involved in the model are known. A simple mass-spring-damper system is taken as a system to be modeled in which the process and measurement noise are sampled from the Gaussian distribution and added to the model. With these setups, the simulated model results in data are affected by process and measurement noise. The simulated data are then used by the algorithm to estimate the involved parameters. Evaluation of the algorithm in terms of its performance is then conducted. For the simulated data, the developed algorithm requires eight iterations to achieve the convergence criteria. The resulted estimates show that the algorithm is able to produce the estimates with a parameter ratio higher than 90% for strong parameters and around 60% for weak parameter. The goodness of fit of the model output and the measurement is evaluated in terms of coefficient of determination (R^2). The resulted outputs fit the measurement in a relatively good fit with a coefficient of determination of higher than 95%. Residual analysis is conducted to check the whiteness of the resulted residuals by plotting them in the Q-Q plot. The resulted plots show that the residual results from the algorithm fit the Gaussian distribution with small curvatures. At the end of Chapter 5, the algorithm is tested on scenarios with constant $\mathbf{P}_0$ and computed $\mathbf{P}_0$. The corresponding results show that the parameters estimated with computed $\mathbf{P}_0$ show slightly better estimates as well as fewer iteration compared to the constant $\mathbf{P}_0$.

Chapter 6 demonstrates the applicability of the developed algorithm to the FDM data. Since the FDM data received from the partner airlines are in a binary format and mixed between different aircraft types that depart and land at different airports around the world, a data preprocessing step is required before the application of the algorithm. The data preprocessing step is started by decoding the raw flight data into an engineering unit by using a dataframe that maps the parameters' location in the binary data. This step is followed by a data selection process in which the flights are clustered based on the aircraft type, departure/arrival airport, and runway location categories. For the data selection process, the k-nn algorithm adopted from the Machine Learning field is used. Along with airport database that contains airport location, airport ICAO name, runway identifiers and their corresponding position, and parameters obtained from the decoded flight; the k-nn algorithm is able to robustly identify the arrival airport and runway identifiers.

Furthermore, a flight phase selection is conducted on the selected data. For the present case, two flight phases are defined during the final approach to landing phase. Phase 1 is defined from 1,000 ft until 100 ft AGL during the final approach phase, and phase 2 is defined from 100 ft AGL until the end of the landing phase where the aircraft has not vacated the runway. There are two reasons for the application of the developed algorithm on these two phases. First, these two phases are selected due to the relatively large variability of the aircraft states. With a relatively large variation, it is expected that the respective parameters can be observed and estimated from the data. Second, the parameters estimated from the algorithm are used in the predictive analysis algorithm, particularly for the runway overrun incident analysis, which is being developed at the Flight Safety Group of the Institute of Flight System Dynamics of TUM. Even though the runway overrun model requires only parameters relevant to the landing phase, but from the system identification point of view - especially for the present case - estimating the parameters by only using the landing phase leads to the instability of the algorithm and a questionable estimate. These two issues are driven by the fact that during the landing phase, more parameters are required to be estimated since one additional force that results from the interaction between the aircraft and the runway surface affects the aircraft's motion. The addressing of this force and the corresponding parameters in the model leads to more parameters to be estimated and consequently increase the linear dependency between the parameters. To solve this problem, some parameters that are assumed to be the same during phase 1 and phase 2 are estimated only in phase 1, and the resulted estimates are then used as input parameters in the phase 2 model. Through this strategy, the number of parameters to be estimated in phase 2 decrease, which leads to the stability of the algorithm.

Even though this strategy ensures the stability of the algorithm but the resulted estimates are physically questionable as the utilized data contain low information content. Thus, in order to increase the information content in the utilized data, one additional technique is proposed in this work. It is carried out by processing several flights simultaneously. The combination of the number of flights has been tried out from two to seven flights. Based on several testings, the combination of five flights ensures the stability of the algorithm, acceptable execution time, and reasonable estimates. With this combination, the algorithm is applied to the whole data set (1,480 flights) that land at the same airport, and on the same runway.

The estimated parameters are evaluated by comparing them with the reference values. However, not all parameters can be compared with their corresponding reference values as they are not publicly available. Thus, for parameters with the lack of reference values, only qualitative analysis is given. Parameters that are compared with the reference values show a reasonable deviation, for example, the estimated thrust (T_0) deviates around 8% with respect to the reference value. The proof of match step is also carried out in this chapter. The model outputs fit the measurements with a relatively high coefficient of determination (R^2), around 80-99%. With this high coefficient of determination values, the utilized model is expected to have a good predictive capability.

To conclude, the work presented in this dissertation provides an estimation technique that is capable of dealing with FDM data. From this perspective, this developed method provides an alternative approach for extending the capability of the FDM program by providing the unrecorded parameters. Embedding the algorithm presented in this work to the FDM program is going to provide a broader perspective for flight safety analyst since more information will be available to be investigated. This use is not only limited to the airlines' related industry, but also to other parties that require estimations of the unrecorded parameters based on FDM data.

Bibliography

[1] European Commision. *Flightpath 2050 Europe's Vision for Aviation*. Publication Office of the European Union, Luxembourg, 2011. ISBN: 978-92-79-19724-6.

[2] Jürgen Raps. Sicherheit muss jeden tag neu produziert werden, 9 2009. Risk Management bei Lufthansa.

[3] International Civil Aviation Organization (ICAO). *Annex 6 Operation of Aircraft, Part I, International Commercial Air Transport-Aeroplanes*. International Civil Aviation Organization, 999 University Street, Montreal, Quebec, Canada, H3C 5H7, 2010. ISBN: 978-92-9231-536-8.

[4] European Aviation Safety Agency (EASA). *Acceptable Means of Compliance and Guidance Material to Part-ORO*. European Aviation Safety Agency, Cologne, Germany, 2 2016.

[5] Civil Aviation Authority (CAA). *CAP 739: Flight Data Monitoring*. The Stationary Office (TSO), CAA Aviation House, Gatwick Airport South, West Sussex, RH6 0YR, 8 2013. ISBN: 0-86039-930-0.

[6] Helder Mendes. Study of mathematical algorithms to identify abnormal patterns in aircraft flight data. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2012.

[7] Gérau de Rivals. Landing trajectory computation. The 3^{rd} European Operators Flight Data Monitoring Conference, Köln, Germany, 2014.

[8] Phillip Koppitz, Joachim Siegel, Nikolaus Romanow, Lukas Höhndorf, and Florian Holzapfel. Touchdown point detection for operational flight data using quality measures and a model based approach. In *2018 AIAA Atmospheric Flight Mechanics Conference*, page 1018, 2018.

[9] Herbert E. Rauch, C.T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[10] Ravindra V. Jategaonkar and F. Thielecke. Aircraft parameter estimation—a tool for development of aerodynamic databases. *Sadhana*, 25(2):119–135, 2000.

[11] Charles Nyce and A. Cpcu. Predictive analytics white paper. *American Institute for CPCU. Insurance Institute of America*, pages 9–10, 2007.

[12] Ludwig Drees. *Predictive Analysis: Quantifying Operational Airline Risks*. PhD thesis, Technische Universität München, 2016.

[13] International Air Transport Association (IATA). *Safety Report*. International Air Transport Association, 800 Place Victoria, P.O. Box 113, Montreal, Quebec, Canada H4Z 1M1, 4 2014. ISBN: 978-92-9252-349-7. Chapter 9 Predictive Analysis.

[14] Federal Aviation Administration FAA. Flight operational quality assurance. Technical report, U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., 4 2004. Advisory Circular (AC) AC No:120-82.

[15] Teledyne Controls. AirFase Flight Data Monitoring System. *Teledyne Product Brochure*, pages 6–9, 2019. Accessed: 21-02-2019.

[16] Safran S.A. Cassiopée AGS Data Management Software. *Safran Product Brochure*, pages 1–3, 2019. Accessed: 21-02-2019.

[17] Serçin Höhndorf. Personal communication - Pegasus Airlines former employee.

[18] Ravindra V. Jategaonkar. *Flight vehicle system identification: A time-domain methodology*. American Institute of Aeronautics and Astronautics, Inc., 2015.

[19] Peter G. Hamel and Ravindra V. Jategaonkar. Evolution of flight vehicle system identification. *Journal of aircraft*, 33(1):9–28, 1996.

[20] Ralph D. Kimberlin. *Flight testing of fixed-wing aircraft*. American Institute of Aeronautics and Astronautics, 2003.

[21] Narendra K Gupta and W. Earl Hall Jr. Input design for identification of aircraft stability and control derivatives. *NASA CR-2493*, 1975.

[22] Eugene Morelli. Flight test validation of optimal input design and comparison to conventional inputs. In *22nd Atmospheric Flight Mechanics Conference*, page 3711, 1997.

[23] WANG Qing, WU Kaiyuan, Tianjiao ZHANG, KONG Yi'nan, and QIAN Weiqi. Aerodynamic modeling and parameter estimation from qar data of an airplane approaching a high-altitude airport. *Chinese Journal of Aeronautics*, 25(3):361–371, 2012.

[24] Edward C. Lan, WU Kaiyuan, and YU Jiang. Flight characteristics analysis based on qar data of a jet transport during landing at a high-altitude airport. *Chinese Journal of Aeronautics*, 25(1):13–24, 2012.

[25] Vladislav Klein and Eugene A. Morelli. *Aircraft system identification: theory and practice*. American Institute of Aeronautics and Astronautics Reston, VA, 2006.

[26] Jitendra R. Raol, Gopalrathnam Girija, and Jatinder Singh. *Modelling and parameter estimation of dynamic systems*, volume 65. IET, 2004.

[27] Henk Haverdings and Pak Wai Chan. Quick access recorder data analysis software for windshear and turbulence studies. *Journal of Aircraft*, 47(4):1443–1447, 2010.

[28] Henk Haverdings and Pak Wai Chan. Quick access recorder (qar) data analysis software for studies of windshear, turbulence, and wake vortex. *Executive Summary National Aerospace Laboratory (NLR)*, 2010.

[29] Nils Möhr. Flight data decoding used for generating en-route information based on binary quick access recorder data. Master's thesis, Technische Universität München, 2016.

[30] Chong Wang, Nils Möhr, and Florian Holzapfel. Decoding of binary flight data in arinc 717 format. In *1^{st} International Conference in Aerospace For Young Scientists (ICAYS)*, 2016.

[31] Future Sky Safety. `https://futuresky-safety.eu/`. Accessed: 2018-08-05.

[32] SafeClouds.eu. `https://safeclouds.eu/`. Accessed: 2018-05-04.

[33] Apache Foundation. `https://hadoop.apache.org/`. Accessed: 2018-09-05.

[34] MySQL. `https://www.mysql.com/`. Accessed: 2018-09-05.

[35] Neil A. H. Campbell. The use of enhanced ground proximity warning system (egpws) data for aviation safety investigation. Australian New Zealand Society of Air Safety Investigators Conference, ANZSASI, Queenstown, New Zealand, 2005.

[36] SafeClouds.eu. D2.2 data acquisition and computing requirements. SafeClouds.eu Project Report, 5 2017.

[37] SKYbrary Aviation Safety. `https://www.skybrary.aero/index.php/Flight_Data_Recorder_(FDR)`. Accessed: 2018-03-12.

[38] Australian Transport Safety Bureau (ATSB). `https://www.atsb.gov.au/publications/2014/black-box-flight-recorders/`. Accessed: 2018-03-01.

[39] Pascal Andrei, Uwe Bartels, and Volkmar Neeb. Flight data recovery: Time for evolutions. *Airbus FAST Magazine*, 48, 2011.

[40] Australian Transport Safety Bureau (ATSB). A review of the effectiveness of emergency locator transmitter in aviation accidents. Technical report, Australian Government of Australian Transport Safety Bureau, 62 Northbourne Avenue Canberra, Australian Capital Territory 2601, 5 2013. ATSB Transport Safety Report AR-2012-128.

[41] SKYbrary Aviation Safety. https://www.skybrary.aero/index.php/Underwater_Locator_Beacon_(ULB). Accessed: 2018-02-11.

[42] RJE International Inc. https://www.rjeint.com/wp-content/uploads/2017/01/ULB-362-ULB-362PL-Manual.pdf. Accessed: 2018-02-11.

[43] Neil A. H. Campbell. The evolution of flight data analysis. Australian New Zealand Society of Air Safety Investigators Conference, ANZSASI, Wellington, New Zealand, 2007.

[44] Nicolas Chrysanthos. *Kernel Methods for Flight Data Monitoring*. PhD thesis, Université de Technologie Troyes, 2014.

[45] Peter Ashford. http://aea.net/AvionicsNews/ANArchives/TechSpeakFeb10.pdf. Accessed: 2018-03-21.

[46] Project Gutenberg. http://self.gutenberg.org/articles/eng/Flight_data_recorder. Accessed: 2018-06-21.

[47] Jeremy Sear. The ARL 'black box' flight recorder - invention and memory. Bachelor's thesis, Department of History, Faculty of Arts, The University of Melbourne, 2001.

[48] David Warren and Ken Fraser. The black box: An australian contribution to air safety. Unpublished Note by Dr. David Warrend and Ken Fraser, 7 1998.

[49] Australian Government Department of Defence. https://www.dst.defence.gov.au/innovation/black-box-flight-recorder/david-warren-inventor-black-box-flight-recorder. Accessed: 2018-06-21.

[50] Scott M. Fisher. http://www.historynet.com/father-black-box.htm. Accessed: 2018-03-21.

[51] Boeing. https://www.boeing.com/commercial/aeromagazine/aero_02/textonly/s01txt.html. Accessed: 2018-03-21.

[52] Bureau d'Enquêtes et d'Analyses (BEA). Flight data recorder read-out: Technical and regulatory aspects. Technical report, Bureau d'Enquêtes et d'Analyses pour la Sécurité de l'Aviation Civile, 5 2005.

[53] Jianye Zhang and Peng Zhang. *Time series analysis methods and applications for flight data*. Springer, 2017.

[54] Gury Norris. https://www.flightglobal.com/news/articles/boeing-787s-to-be-fitted-with-enhanced-data-recorder-207283. Accessed: 2018-02-2.

[55] Tom Dodt. Introducing the 787. International Society of Air Safety Investigators, Salt Lake City, Utah, U.S., 2011.

[56] Wayne Rosenkrans. Fade-free memory. *FSF AeroSafetyWorld Magazine*, 2008.

[57] Federal Aviation Administration (FAA). Flight data recorder systems. Technical report, U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., 4 2007. TSO-C124b.

[58] Federal Aviation Administration (FAA). Cockpit voice recorder equipment. Technical report, U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C., 12 2013. TSO-C123c.

[59] Dennis R. Grossi. Aviation recorder overview. In *International Symposium On Transportation Recorders, Arlington, Virginia*, 1999.

[60] Paul Dubois. Flight data analysis, 9 2014. Lima FDA Seminar.

[61] Mike Tooley. *Aircraft digital electronic and computer systems*. Routledge, 2013.

[62] Cary Spitzer, Uma Ferrell, and Thomas Ferrell. *Digital avionics handbook*. CRC press, 2014.

[63] Nour El-Din Safwat, MA El-Dakroury, and Abdelhalim Zekry. The evolution of aircraft data networks. *International Journal of Computer Applications*, 94(11), 2014.

[64] M Sudolsky. ARINC 573/717, 767 and 647a: The logical choice for maintenance recording and IVHM interface control or frame updates. In *Proc. of Prognostics and Health Management Society*, 2009.

[65] Aeronautical Radio Inc. *Enhanced Airborne Flight Recorder: ARINC Characteristics 767-1*. Aeronautical Radio Inc., Annapolis, MA, U.S., 2009.

[66] Actel Corporation. ARINC 429 bus interface. Technical report, Actel Corporation, 9 2006.

[67] Avionics Interface Technologies (Aviftech). ARINC 429 protocol tutorial. Technical report, Avionics Interface Technologies, 3703 N. 200th Street, Omaha, NE 68022, U.S, —.

[68] João Pedro Rego Freitas. Study and implementation of algorithms for in flight performance analysis of the pw4000-100 turbofan engine for the purpose of engine condition monitoring. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, 2014.

[69] Aeronatical Radio Inc. *Flight Data Acquisition and Recording System: ARINC Characteristics 717-15*. Aeronautical Radio Inc., Annapolis, MA, U.S., 2011.

[70] Teledyne Controls. User's guide wireless groundlink - quick access recorder. Technical report, Teledye Controls, 501 Continental Blvd. El Segundo, CA 90245 USA, 11 2013.

[71] Skybrary. `https://www.skybrary.aero/index.php/Flight_Data_Monitoring_(FDM)`. Accessed: 2018-11-5.

[72] Flight Data Services. `https://www.flightdataservices.com/assets/pdf/2017-11-safety-seminar/Flight%20Data%20Recording%20%26%20Data%20Transfer.pdf`. Accessed: 2019-1-5.

[73] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.

[74] John G Proakis. *Digital signal processing: principles algorithms and applications*. Pearson Education India, 2001.

[75] Eugene A Morelli. Estimating noise characteristics from flight test data using optimal fourier smoothing. *Journal of Aircraft*, 32(4):689–695, 1995.

[76] Graham Clifford Goodwin and Robert L Payne. *Dynamic System Identification: Experiment Design and Data Analysis*, volume 136. Academic press New York, 1977.

[77] Florian Holzapfel. Lecture notes in Flight System Dynamics 2, February 2015.

[78] Peter H. Zipfel. *Modeling and Simulation of Aerospace Vehicle Dynamics*. American Institute of Aeronautics and Astronautics, 2007.

[79] National Imagery and Mapping Agency. Department of defense world geodetic system 1984: its definition and relationship with local geodetic system. Technical report, Geodesy and Geophysics Department, National Imagery and Mapping Agency, USA, 2000.

[80] Robert F. Stengel. *Flight Dynamics*. Princeton University Press, 2015.

[81] Brian L. Stevens, Frank L. Lewis, and Eric N. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. John Wiley & Sons, 2015.

[82] Jean-Luc Boiffier. *The Dynamics of Flight*. Wiley, 1998.

[83] Bernard Etkin. *Dynamics of atmospheric flight*. Dover Publications, Inc., 2012.

[84] Jan Roskam. Airplane flight dynamics and automatic flight controls, part i. *DARcorporation Design, Analysis, Research*, 120, 2001.

[85] Michael V. Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.

[86] Jitendra R. Raol and Jatinder Singh. *Flight mechanics modeling and analysis*. CRC Press, 2008.

[87] Greenberg H. Determination of stability derivatives from flight data. *Journal of the Aeronautical Sciences*, 16, 1949.

[88] Murray Tobak. On the use of the indicial-function concept in the analysis of unsteady motions of wings and wing-tail combinations. Technical report, NACA Rep. 1188, 1954.

[89] Murray Tobak and Lewis B. Schiff. On the formulation of the aerodynamic characteristics in aircraft dynamics. Technical report, NASA TR R-456, 1976.

[90] Devendra K. Chaturvedi. *Modeling and simulation of systems using MATLAB and Simulink*. CRC press, 2009.

[91] Richard E. Maine and Kenneth W. Iliff. *Identification of dynamic systems, theory and formulation*. AGARD AG-300, Vol. 2, 1985.

[92] Lotfi A. Zadeh. From circuit theory to system theory. *Proceedings of the IRE*, 50(5):856–865, 1962.

[93] Florian Holzapfel. Lecture notes in System Identification, February 2015.

[94] Vladislav Klein. On the adequate model for aircraft parameter estimation. Technical report, Cranfield Institute of Technology, 1975.

[95] Raman Mehra. Optimal inputs for linear system identification. *IEEE Transactions on Automatic Control*, 19(3):192–200, 1974.

[96] J. Mulder, J. Sridhar, and J. Breeman. Identification of dynamic system-application to aircraft nonlinear analysis and manoeuvre design. *Technical Report AG 300, AGARD, Tech. Report*, 1994.

[97] Eugene A. Morelli. Flight test of optimal inputs and comparison with conventional inputs. *Journal of aircraft*, 36(2):389–397, 1999.

[98] Karl Heinz Fasol and H. Peter Jörgl. Principles of model building and identification. In *System Identification*, pages 505–518. Elsevier, 1981.

[99] George E.P. Box. Robustness in the strategy of scientific model building. In *Robustness in statistics*, pages 201–236. Elsevier, 1979.

[100] Eric Walter and Luc Pronzato. *Identification of parametric models from experimental data*. Springer Verlag, 1997.

[101] Ravindra V. Jategaonkar and Ermin Plaetschke. Estimation of aircraft parameters using filter error methods and extended kalman filter. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[102] Ronald A. Fisher. On an absolute criterion for fitting frequency curves. *Statistical Science*, 12(1):39–41, 1912.

[103] Ronald A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922.

[104] Ronald A. Fisher. Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 22, pages 700–725. Cambridge University Press, 1925.

[105] John Aldrich. Ra fisher and the making of maximum likelihood 1912-1922. *Statistical science*, 12(3):162–176, 1997.

[106] B.L. van der Waerden. Mathematical statistics. *Proceedings of the Edinburgh Mathematical Society*, 17(3):284–285, 1971.

[107] Karl Johan Åström. Maximum likelihood and prediction error methods. In *System Identification*, pages 551–574. Elsevier, 1981.

[108] Florian Holzapfel, M. Rieck, M. Bittner, M. Richter, and J. Diepolder. Practical course on optimal control, February 2015.

[109] Johann Pfanzagl. *Parametric statistical theory*. Walter de Gruyter, 2011.

[110] Arthur Gelb. *Applied optimal estimation*. MIT Press, 1974.

[111] Rudolph Emil. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[112] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 3. Academic press, 1982.

[113] Christopher D'Souza. Webcast: Fundamentals of kalman filtering and estimation in aerospace engineering, February 2013.

[114] Richard E. Maine and Kenneth W. Iliff. Formulation and implementation of a practical algorithm for parameter estimation with process and measurement noise. *SIAM journal on applied mathematics*, 41(3):558–579, 1981.

[115] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[116] Jerry M. Mendel. *Discrete techniques of parameter estimation: the equation error formulation*. Marcel Dekker Inc., 1973.

[117] Eugene Morelli. Practical aspects of the equation-error method for aircraft parameter estimation. In *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, page 6144, 2006.

[118] Gene H. Golub and Charles F. van Loan. An analysis of the total least squares problem. *SIAM journal on numerical analysis*, 17(6):883–893, 1980.

[119] Jared A. Grauer and Eugene A. Morelli. A new formulation of the filter-error method for aerodynamic parameter estimation in turbulence. In *AIAA Atmospheric Flight Mechanics Conference*, page 2704, 2015.

[120] Ravindra V. Jategaonkar and E. Plaetschke. Algorithms for aircraft parameter estimation accounting for process and measurement noise. *Journal of Aircraft*, 26(4):360–372, 1989.

[121] AV Balakrishnan. Stochastic system identification techniques. *Stochastic Optimization and Control, edited by HF Karreman, John Wiley and sons, London*, 1968.

[122] Raman K. Mehra. Maximum likelihood identification of aircraft parameters. In *Joint Automatic Control Conference*, pages 442–444, 1970.

[123] Ravindra V. Jategaonkar and E. Plaetschke. Identification of moderately nonlinear flight mechanics systems with additive process and measurement noise. *Journal of Guidance, Control, and Dynamics*, 13(2):277–285, 1990.

[124] R. Kashyap. A new method of recursive estimation in discrete linear systems. *IEEE Transactions on Automatic Control*, 15(1):18–24, 1970.

[125] Juan Garcia-Velo and Bruce K. Walker. Aerodynamic parameter estimation for high-performance aircraft using extended kalman filtering. *Journal of Guidance, Control, and Dynamics*, 20(6):1257–1260, 1997.

[126] Andrew H. Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[127] Javensius Sembiring, Joachim Siegel, and Florian Holzapfel. Parameter estimation on low observability data. In *2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 1–7. IEEE, 2018.

[128] D. Alspach. A parallel filtering algorithm for linear systems with unknown time varying noise statistics. *IEEE Transactions on Automatic Control*, 19(5):552–556, 1974.

[129] M.R Ananthasayanam, Naren Naik M. Shyam Mohan, and R.M.O Gemson. A heuristic reference recursive recipe for adaptively tuning the kalman filter statistics part-1: formulation and simulation studies. *Sādhanā Indian Academy of Sciences*, 41(12):1473–1490, 2016.

[130] Vinay A. Bavdekar, Anjali P. Deshpande, and Sachin C. Patwardhan. Identification of process and measurement noise covariance for state and parameter estimation using extended kalman filter. *Journal of Process control*, 21(4):585–601, 2011.

[131] T. Bohlin. Four cases of identification of changing systems. In *Mathematics in Science and Engineering*, volume 126, pages 441–518. Elsevier, 1976.

[132] R. Kashyap. Maximum likelihood identification of stochastic linear systems. *IEEE Transactions on Automatic Control*, 15(1):25–34, 1970.

[133] Kenneth Myers and B.D. Tapley. Adaptive sequential estimation with unknown noise statistics. *IEEE Transactions on Automatic Control*, 21(4):520–523, 1976.

[134] R.M.O. Gemson. Estimation of aircraft aerodynamic derivatives accounting for measurement and process noise by ekf through adaptive filter tuning. *Bangalore, India: Department of Aerospace Engineering, Indian Institute of Science*, 1991.

[135] A.H. Mohamed and K.P. Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.

[136] J. Leathrum. On sequential estimation of state noise variances. *IEEE Transactions on Automatic Control*, 26(3):745–746, 1981.

[137] M. Shyam Mohan, Naren Naik, R.M.O. Gemson, and M.R. Ananthasayanam. A heuristic reference recursive recipe for adaptively tuning the kalman filter statistics part-2: real data studies. *Sādhanā Indian Academy of Sciences*, 41(12):1491–1507, 2016.

[138] Thomas Kailath. An innovations approach to least-squares estimation–part i: Linear filtering in additive white noise. *IEEE transactions on automatic control*, 13(6):646–655, 1968.

[139] R. Mehra. On the identification of variances and adaptive kalman filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1974.

[140] R.Mehra. Approaches to adaptive filtering. *IEEE Transactions on Automatic Control*, 17(5):693–698, 1972.

[141] Pierre R. Bélanger. Estimation of noise covariance matrices for a linear time-varying stochastic process. *Automatica*, 10(3):267–275, 1974.

[142] R. Gemson and M. Anathasayanam. Importance of initial state covariance matrix for the parameter estimation using an adaptive extended kalman filter. In *23rd Atmospheric Flight Mechanics Conference*, page 4153, 1998.

[143] Maria Isabel Ribeiro. Gaussian probability density functions: Properties and error characterization. *Institute for Systems and Robotics, Lisboa, Portugal*, 2004.

[144] Eric W. Weisstein. Bessel's correction. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/BesselsCorrection.html. Accessed: 2019-02-1.

[145] Shyam Mohan M. An iterative tuning strategy for achieving cramer rao bound using exended kalman filter for a parameter estimation problem. Master's thesis, Indian Institute of Technology Kanpur, 2014.

[146] Bruce P. Gibbs. *Advanced Kalman filtering, least-squares and modeling: a practical handbook*. John Wiley & Sons, 2011.

[147] Inc. MathWorks. Mathematics. Technical report, MathWorks, Inc., 3 Apple Hill Drive, Natic, MA 01760-2098, 3 2019. .

[148] Andreas C. Müller, Sarah Guido, et al. *Introduction to machine learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

[149] MathWorks Inc. Statistics and machine learning toolbox user's guide. Technical report, MathWorks, Inc., 3 Apple Hill Drive, Natic, MA 01760-2098, 9 2018.

[150] Wendy L. Martinez and Angel R. Martinez. *Computational statistics handbook with MATLAB*. Chapman and Hall/CRC, 2016.

[151] OurAirports. http://ourairports.com/data/. Accessed: 2018-11-1.

[152] OpenFlights. https://openflights.org/data.html. Accessed: 2018-11-1.

[153] Florian Holzapfel. Lecture notes in Flight System Dynamics 1, November 2015.

[154] Yongki Go. Lecture notes in MA6641 flight performance and dynamics, October 2015.

[155] Denis Howe and George Rorie. *Aircraft conceptual design synthesis*. Professional Engineering Publishing London, UK, 2000.

[156] Daniel Raymer. *Aircraft Design: A Conceptual Approach 5e*. American Institute of Aeronautics and Astronautics, Inc., 2006.

[157] Egbert Torenbeek. *Synthesis of subsonic airplane design: an introduction to the preliminary design of subsonic general aviation and transport aircraft, with emphasis on layout, aerodynamic design, propulsion and performance*. Springer Science & Business Media, 2013.

[158] Thomas Zöld. Performance assessment of a hybrid electric-powered long-range commercial airliner, 2014. Diploma Thesis.

[159] CFM. CFM56 engine. http://www.smartcockpit.com/docs/CFM_Flight_Ops_Support_B737.pdf, 12 2005. Slide presentation by CFM Flight Operations Support.

[160] Joachim Siegel. Quick acces recorder flight data analysis using extended kalman filter-based methods, 2014. Semester Thesis.

[161] European Aviation Safety Agency. EASA.E.060 type certificate data sheet. Technical report, European Aviation Safety Agency, Cologne, Germany, 2007.

[162] Javensius Sembiring, Ludwig Drees, and Florian Holzapfel. Extracting unmeasured parameters based on quick access recorder data using parameter-estimation method. In *AIAA Atmospheric Flight Mechanics (AFM) Conference*, page 4848, 2013.

[163] Ludwig Drees, Heiko Haselhofer, Javensius Sembiring, and Florian Holzapfel. Modeling the flare maneuver performed by airline pilots using flight operation data. In *AIAA Modeling and Simulation Technologies (MST) Conference*, page 4912, 2013.

[164] Javensius Sembiring, Lukas Höhndorf, and Florian Holzapfel. Bayesian approach implementation on quick access recorder data for estimating parameters and model validation. *Probabilistic Safety Assessment and Management PSAM*, 12, 2014.

[165] Lukas Höhndorf, Javensius Sembiring, and Florian Holzapfel. Copulas applied to flight data analysis. *Probabilistic Safety Assessment and Management PSAM*, 12, 2014.

[166] Chong Wang, Ludwig Drees, Nadine Gissibl, Lukas Höhndorf, Javensius Sembiring, and Florian Holzapfel. Quantification of incident probabilities using physical and statistical approaches. In *6th International Conference on Research in Air Transportation. Istanbul, Turkey*, 2014.

[167] Lukas Höhndorf, Joachim Siegel, Javensius Sembiring, Phillip Koppitz, and Florian Holzapfel. Reconstruction of aircraft trajectories during landing using a rauch-tung-striebel smoother, instrument landing system deviation information, and taxiway locations. In *AIAA Atmospheric Flight Mechanics Conference*, page 3705, 2016.

[168] Lukas Höhndorf, Joachim Siegel, Javensius Sembiring, Phillip Koppitz, and Florian Holzapfel. Reconstruction of aircraft states during landing based on quick access recorder data. *Journal of Guidance, Control, and Dynamics*, 40(9):2393–2398, 2017.

[169] Javensius Sembiring, Changwu Liu, Phillip Koppitz, and Florian Holzapfel. Energy management for unstable approach detection. In *2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 1–6. IEEE, 2018.

[170] Phillip Koppitz, Chong Wang, Lukas Höhndorf, Javensius Sembiring, Xiaolong Wang, and Florian Holzapfel. From raw operational flight data to incident probabilities using subset simulation and a complex thrust model. In *AIAA Scitech 2019 Forum*, page 2233, 2019.

[171] Xiaolong Wang, Javensius Sembiring, Phillip Koppitz, Lukas Höhndorf, Chong Wang, and Florian Holzapfel. Modeling of the aircraft's low energy state during the final approach phase using operational flight data. In *AIAA Scitech 2019 Forum*, page 0989, 2019.

Appendix A

Reference Frames and Transformation

Reference and transformation frames have been briefly given in Chapter 3. In this appendix a further explanation of the reference frames and the procedure used for frame transformation will be given in details. Each reference frame will be provided with a diagram to support the explanation visually. The material presented here mainly adapted from Flight System Dynamics 2 lecture at Institute of Flight System Dynamics, TUM [77]

A.1 Reference Frames

A reference frame plays an important role in modeling of the aircraft dynamic and motions. Without a reference frame, the positions or motions of the airplane vehicle cannot be modeled. A reference frame is constructed by location and orientation with respect to other frame. Location property represents the displacement of an object relative to the other frame. While the orientation property represent the attitude of an object with respect to the other frame. Both location and orientation determine the position of the frame. There are several reference frames commonly known in the aerospace application but only some them are referred in this dissertation. Each of them is discussed in the following section A.1.1 - A.1.7.

A.1.1 Earth Centered Inertial - ECI

Index:	$\boldsymbol{I}$
Role:	Euclidean frame (inertial axis system - Newton's Laws may be used)
Origin:	Center of the Earth
Translation:	Elliptical path around the sun with the solar system
Rotation:	none
x axis:	In the equatorial plane in the direction of the vernal equinox
y axis:	In the equatorial plane, forming a right hand system with x axis and z axis
z axis:	Rotation axis of the Earth

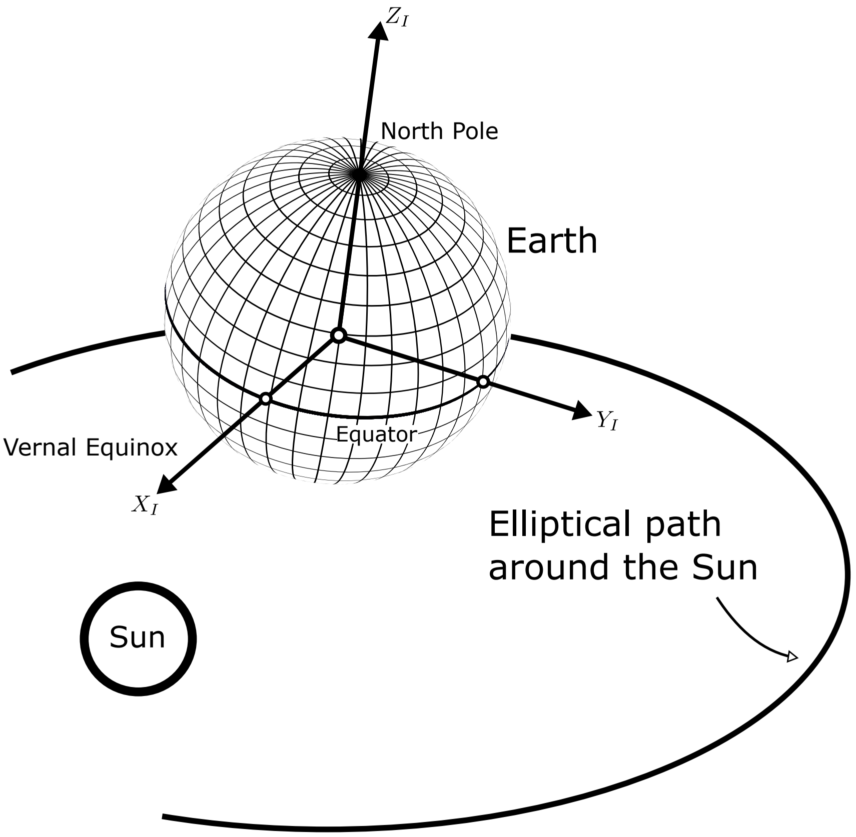


Figure A.1: ECI frame, $\boldsymbol{I}$

A.1.2 Earth Centered Earth Fixed Frame - ECEF

Index:	$\boldsymbol{E}$
Role:	Navigation frame for specifying a position
Origin:	Center of the Earth
Translation:	Moves with the ECI-frame
Rotation:	Around the z axis with Earth rotation rate, i.e. $\approx 2\pi$ per $24h$, ω^{IE}
x axis:	In the equatorial plane in the direction of the Greenwich Meridian
y axis:	In the equatorial plane, forming a right hand system with x axis and z axis
z axis:	Rotation axis of the Earth (collinear with the ECI's z axis)

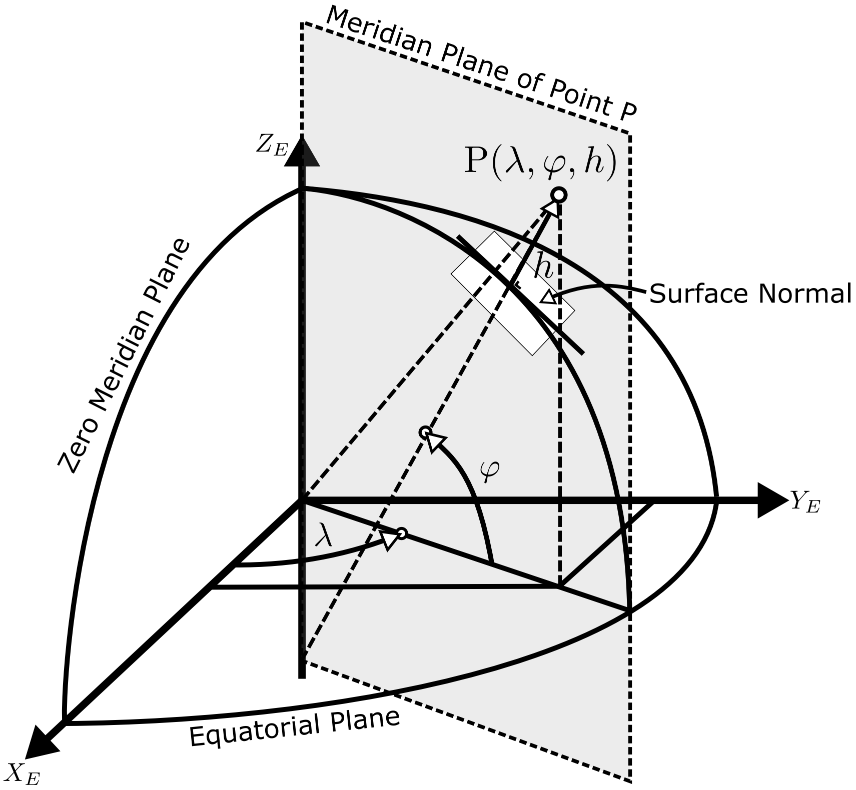


Figure A.2: ECEF frame, $\boldsymbol{E}$

A.1.3 North East Down Frame - NED

Index:	$\boldsymbol{O}$
Role:	Attitude/orientation frame, used to specify the attitude of the plane
Origin:	Reference point of the aircraft, $\boldsymbol{R}$
Translation:	Moves with the aircraft's reference point
Rotation:	Rotates with the transport's rate to keep the NED's alignment, ω^{EO}
x axis:	Parallel to the local geoid surface, pointing to the geographic north pole
y axis:	Parallel to the local geoid surface, pointing east to form a right hand system with x axis and z axis
z axis:	Pointing downwards, perpendicular to the local geoid surface

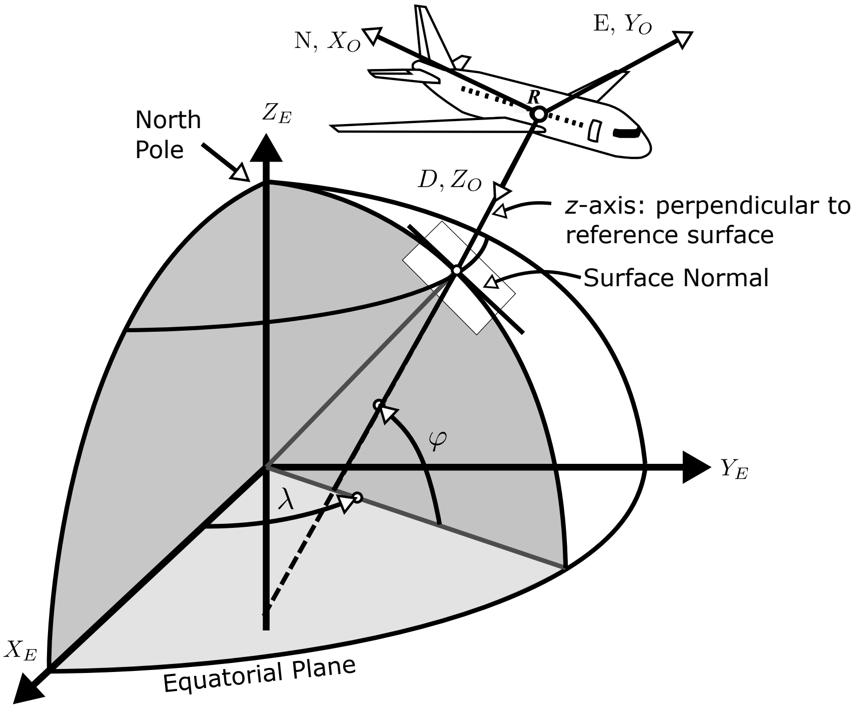


Figure A.3: NED frame, $\boldsymbol{O}$

A.1.4 Body Fixed Frame

Index:	$\boldsymbol{B}$
Role:	Notation frame, used for forces, moments, ...
Origin:	Reference point of the aircraft, $\boldsymbol{R}$
Translation:	Moves with the aircraft reference point
Rotation:	Rotates with the rigid body aircraft
x axis:	Pointing towards the aircraft nose
y axis:	Pointing to the right (starboard) wing to form an orthogonal right hand system
z axis:	Pointing downwards in the symmetry pane of the aircraft, perpendicular to the x and y axes

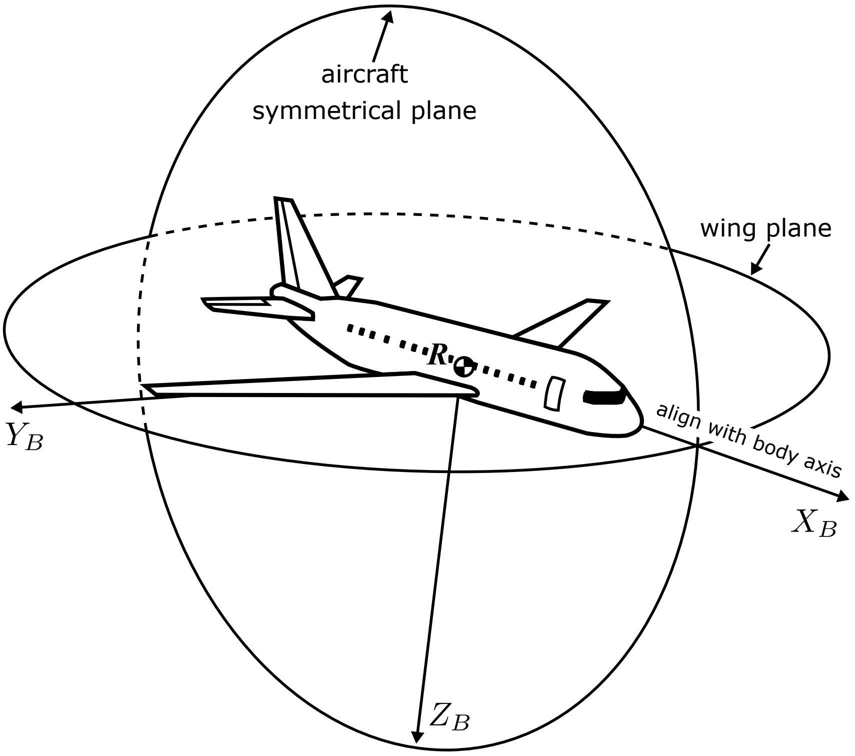


Figure A.4: Body-Fixed frame, $\boldsymbol{B}$

A.1.5 Aerodynamic Frame

Index:	$\boldsymbol{A}$
Role:	Notation frame for aerodynamic flow
Origin:	Reference point of the aircraft, $\boldsymbol{R}$
Translation:	Moves with the aircraft reference point
Rotation:	Rotates with the direction of the airflow
x axis:	Aligned with the aerodynamic velocity, pointing into the direct of the aerodynamic velocity
y axis:	Pointing to the right perpendicular to the x and z axes
z axis:	Pointing downwards in the symmetry plane of the aircraft perpendicular to the xy-plane

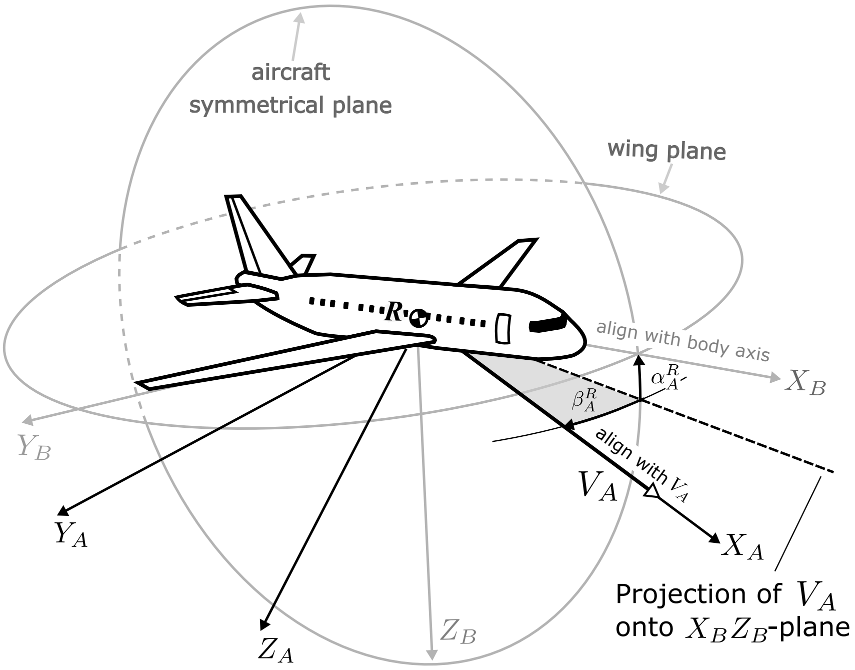


Figure A.5: Aerodynamic reference frame, ***A***

A.1.6 Kinematic Frame

Index:	$\boldsymbol{K}$
Role:	Flight-path axis frame
Origin:	Reference point of the aircraft, $\boldsymbol{R}$
Translation:	Moves with the aircraft reference point
Rotation:	Rotates with the direction of the kinematic velocity
x axis:	Aligned with the kinematic velocity, pointing into the direction of the kinematic velocity
y axis:	Pointing to the right perpendicular to the x and z axes
z axis:	Pointing downwards, parallel to the projection of the local normal to the WGS-84 ellipsoid into a plane perpendicular to the x axis (i.e. to the kinematic velocity)

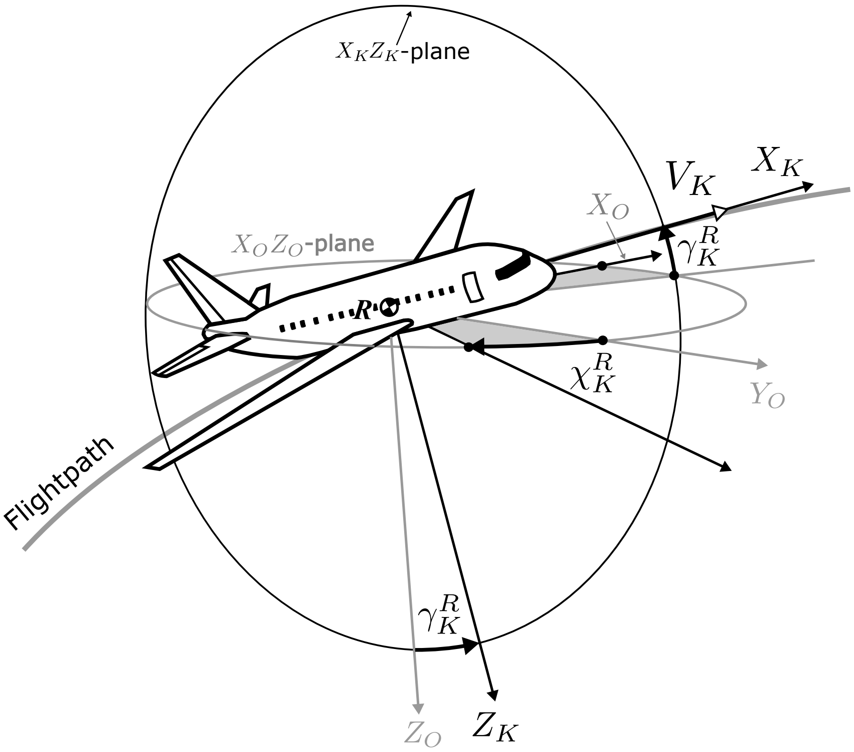


Figure A.6: Kinematic frame, $\boldsymbol{K}$

A.1.7 Navigation Frame

Index:	$\boldsymbol{N}$
Role:	Navigation frame, derived from the NED-frame to specify the position
Origin:	Point on the Earth's surface
Translation:	None
Rotation:	Only angular rate of earth, because fixed to the earth (does not rotate with transport rate)
x axis:	Parallel to the local geoid surface, pointing to a direction that deviates with alignment angle χ_N from north direction
y axis:	Parallel to the local geoid surface, to form a right hand system with x and z axes
z axis:	Pointing downwards, perpendicular to the local geoid surface

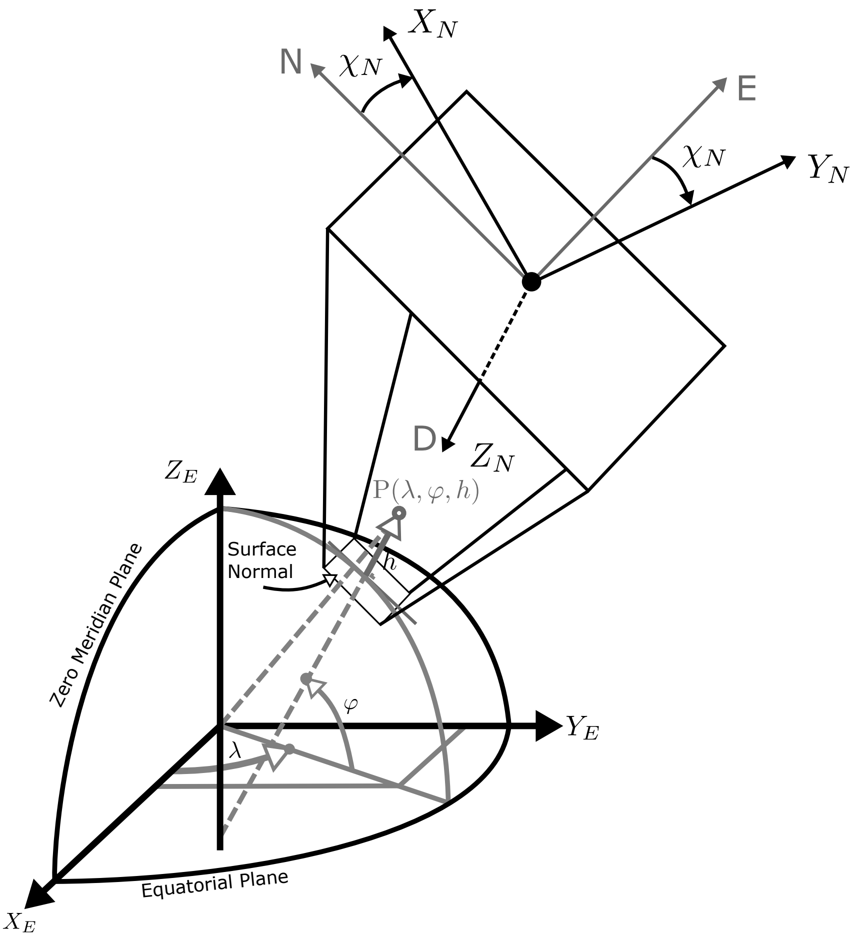


Figure A.7: Navigation frame, $\boldsymbol{N}$

A.2 Frame Transformation

Frame transformation is performed through a transformation matrix. A transformation matrix contains angles that represent attitude between one frame to the other frame. This section will present the derivation of a 3D transformation matrix. The derivation will be started through a 2D rotation matrix which can be easily extended to the 3D rotation matrix. The following Figure A.8 depicts a frame Ox_by_b (System-b) is obtained by rotating Ox_ay_a through a θ angle counterclockwise.

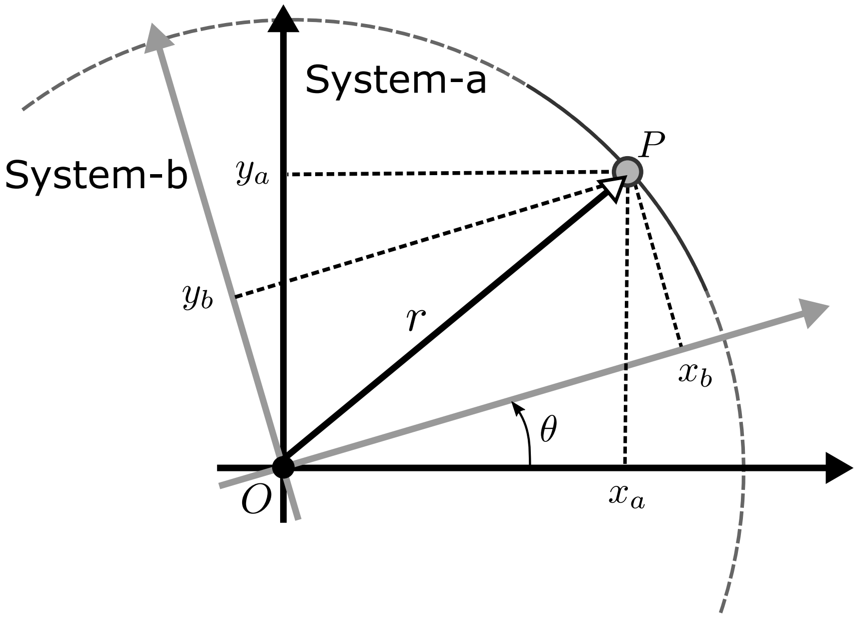


Figure A.8: 2D transformation matrix procedure

In the Ox_ay_a frame, every polar coordinates can be represented as function of length and angle (r, α). Thus, for point P yields,

$$x_a = r \cdot \cos\alpha \tag{A.2.1}$$

$$y_a = r \cdot \sin\alpha \tag{A.2.2}$$

In the second frame Ox_by_b, every polar coordinates can be also represented as function of length and angle as this frame is obtained from Ox_ay_a through the rotation angle θ. Thus, every polar coordinates in this frame is a function of $(r, \alpha - \theta)$. Consequently, point P in System-b is written as,

$$x_b = r \cdot cos(\alpha - \theta) = r \cdot \cos\alpha \cdot \cos\theta + r \cdot \sin\alpha \cdot \sin\theta \tag{A.2.3}$$

$$y_b = r \cdot sin(\alpha - \theta) = r \cdot \sin\alpha \cdot \cos\theta - r \cdot \cos\alpha \cdot \sin\theta \tag{A.2.4}$$

Substituting equations (A.2.1) and (A.2.2) into equations (A.2.3) and (A.2.4), yields

$$x_b = x_a \cdot \cos\theta + y_a \cdot \sin\theta \tag{A.2.5}$$

$$y_b = -x_a \sin\theta + y_a \cdot \cos\theta \tag{A.2.6}$$

In a matrix form, equation (A.2.5) and (A.2.6) can be represented as,

$$\begin{bmatrix} x_b \\ \\ y_b \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta \\ \\ -\sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{M}} \cdot \begin{bmatrix} x_a \\ \\ y_a \end{bmatrix} \tag{A.2.7}$$

The matrix $\mathbf{M}$ is so called the standard rotation matrix in two dimensions. To extend this rotation matrix into 3 dimensions, it can be assumed that 2D rotations is done in axis which is not used for rotation. For example, if z_a axis is assumed as rotation axis , then the respective rotation matrix is obtained as,
Rotation around z_a axis,

$$\mathbf{M}_{z_a} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & & 1 \end{bmatrix} \tag{A.2.8}$$

The same principle also applies when x_a and y_a axes are used as rotation axes,
Rotation around x_a axis,

$$\mathbf{M}_{x_a} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{A.2.9}$$

Rotation around y_a axis,

$$\mathbf{M}_{y_a} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{A.2.10}$$

To get the complete rotation matrix, all matrices are multiplied together. However, it should be noted that successive rotations around coordinate axis are not commutative.

$$\mathbf{M}_{321} = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \neq \mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{M}_3 \tag{A.2.11}$$

For aerospace application, the rotation matrix is obtained through sequence $z - y - x$ axis. For example, the rotation matrix from the NED frame to the Body-Fixed frame is constructed through the following sequence,

Rotation sequence	$\mathbf{M}_3$	$\mathbf{M}_2$	$\mathbf{M}_1$
Rotation angle	ϕ	θ	ψ
Rotation axis	z	y	x
Matrix	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & & 1 \end{bmatrix}$	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & & 1 \end{bmatrix}$	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & & 1 \end{bmatrix}$

Therefore, the transformation matrix from the NED frame to the Body-Fixed is obtained as,

$$\begin{aligned} \mathbf{M}_{BO} &= \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \\ &= \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \end{aligned} \tag{A.2.12}$$

Appendix B

Flight Safety Group IT Infrastructure

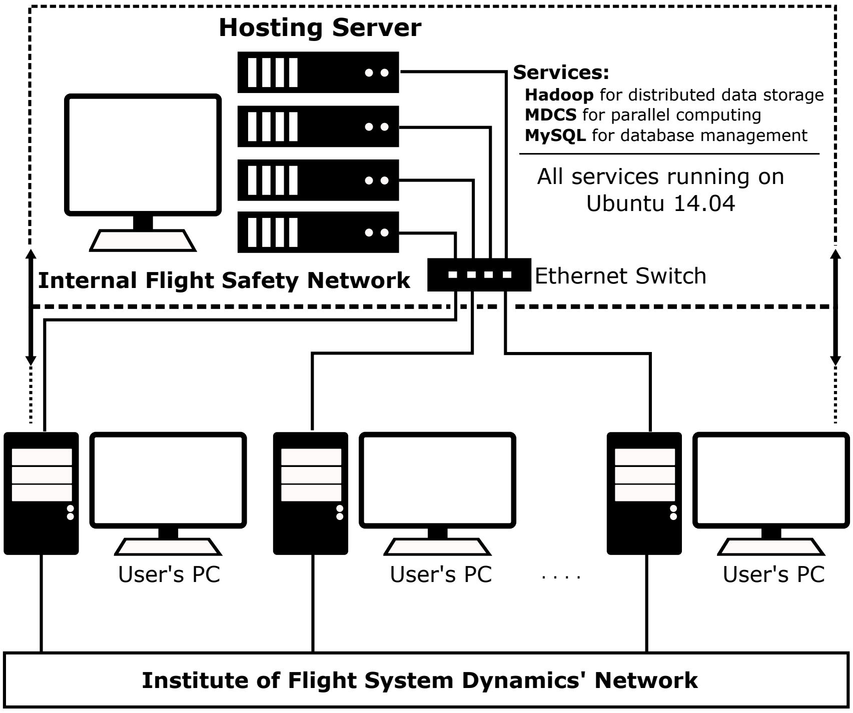

Hosting Server
Services:
Hadoop for distributed data storage
MDCS for parallel computing
MySQL for database management
All services running on Ubuntu 14.04
Internal Flight Safety Network
Ethernet Switch
User's PC
User's PC
....
User's PC
Institute of Flight System Dynamics' Network

Figure B.1: Flight Safety Group IT Infrastructure

Descriptions:

- Flight data are stored in the hosting server and distributed across all nodes. The distribution of the flight data is managed by Hadoop[1] framework.
- For parallel data processing, Matlab Distributed Computing Server (MDCS) is employed. Every user connected to the internal flight safety network can activate this service and process data in a distributed processing.
- MySQL[2] is used to store meta data (flight information, airport, runway, etc.), metar data, and results.

[1] A framework that allows the distributed processing of large data set across nodes of computers using programming models (map-reduce). This framework is an open-source software managed by Apache Software Foundation (ASF)

[2] An open-source relational database management system (RDMS)

Appendix C

Model Output and Corresponding Measurements for Other Flights

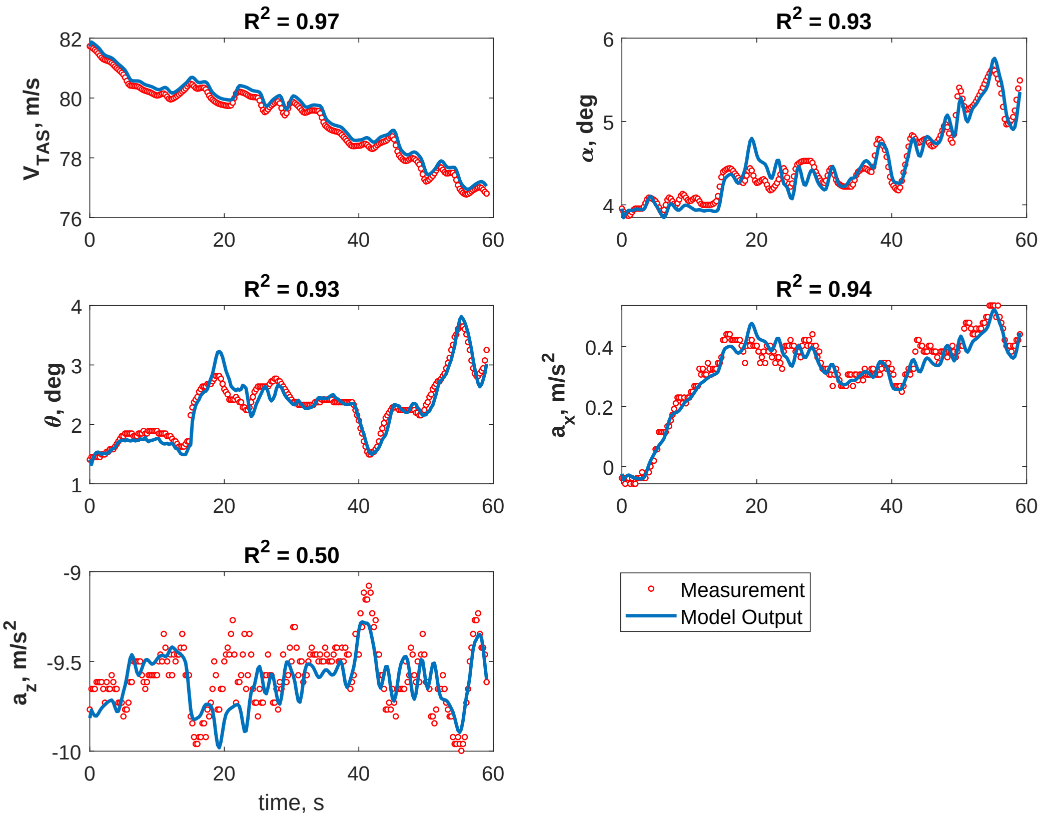


Figure C.1: Phase 1 proof of match - flight B10825

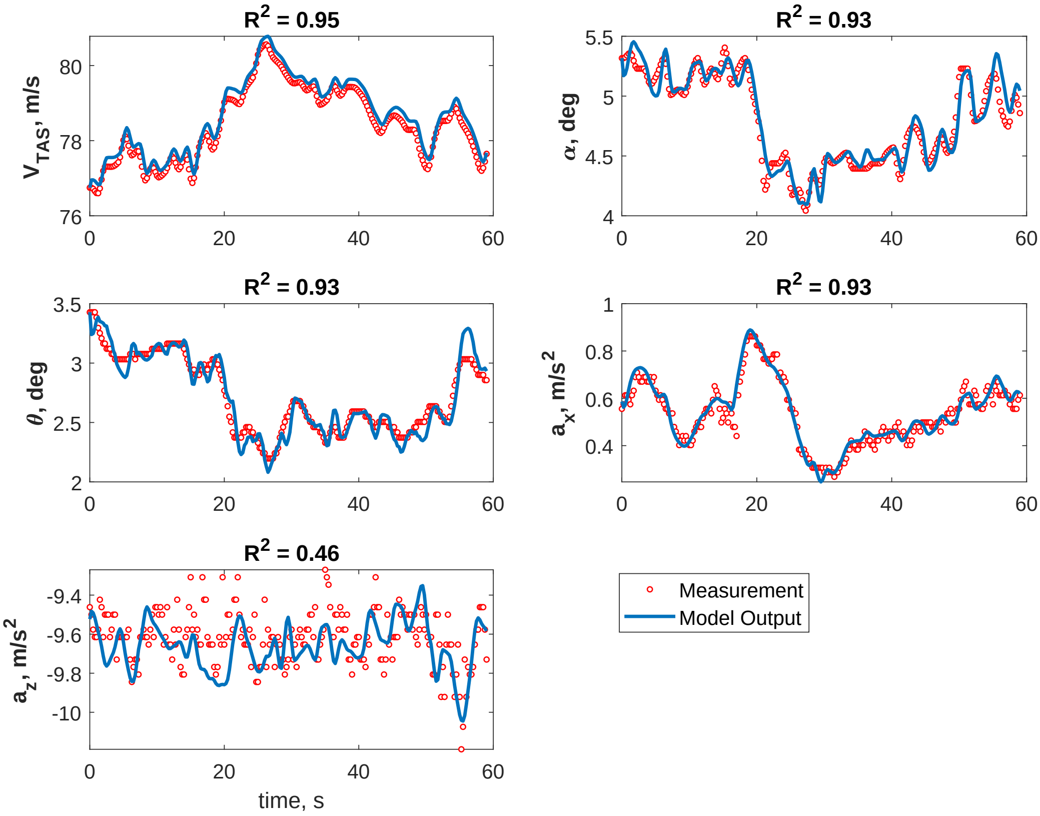


Figure C.2: Phase 1 proof of match - flight B10833

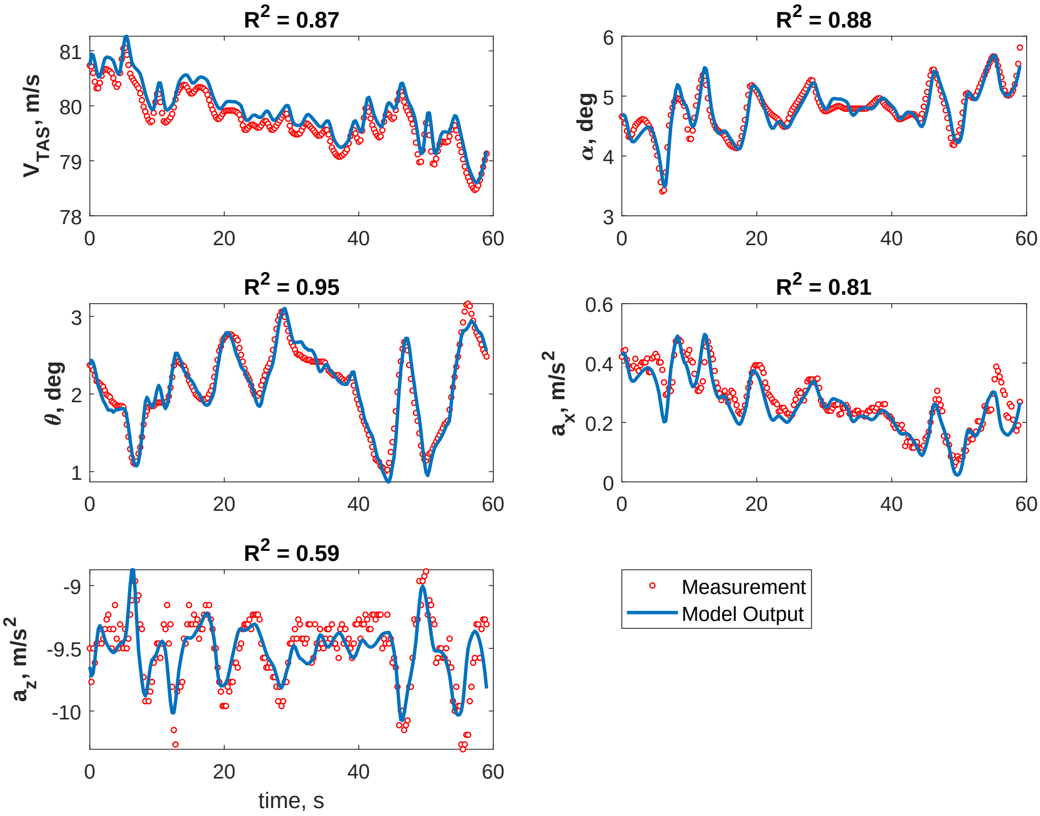


Figure C.3: Phase 1 proof of match - flight B10896

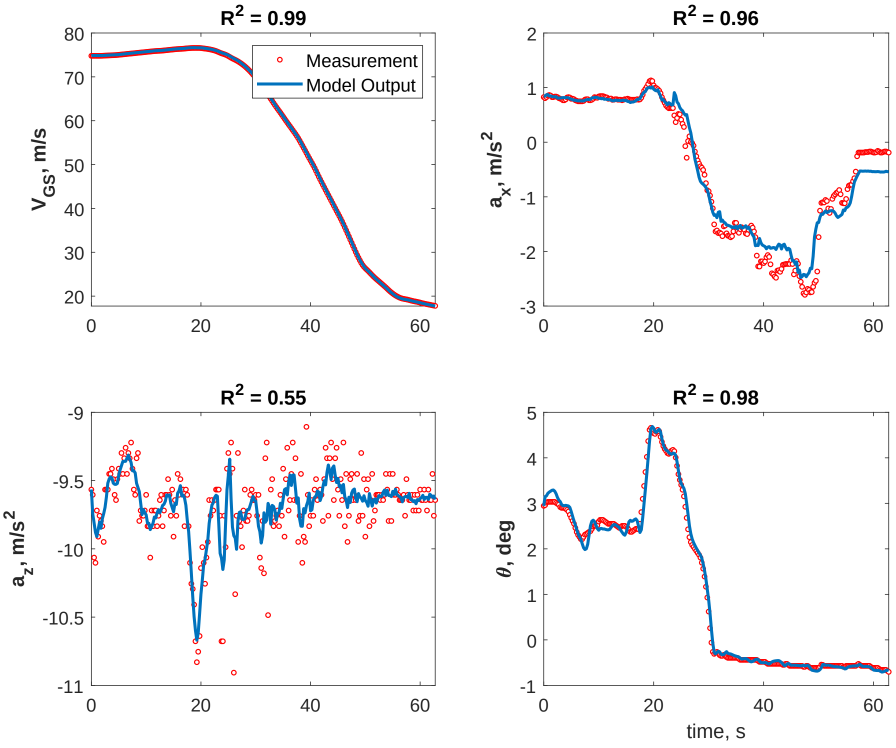


Figure C.4: Phase 2 proof of match - flight B10825

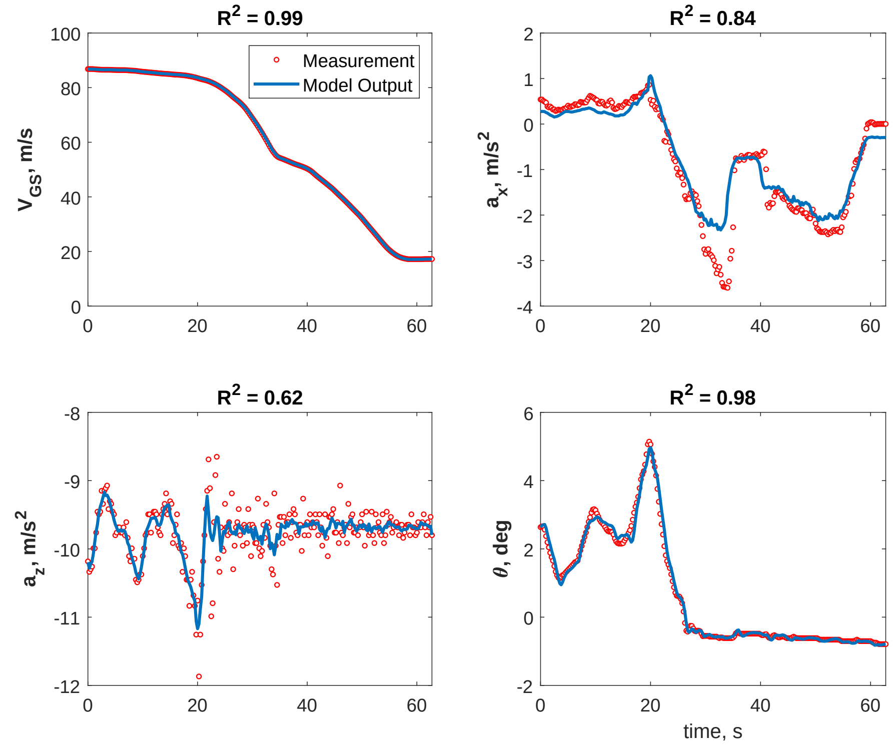


Figure C.5: Phase 2 proof of match - flight B10833

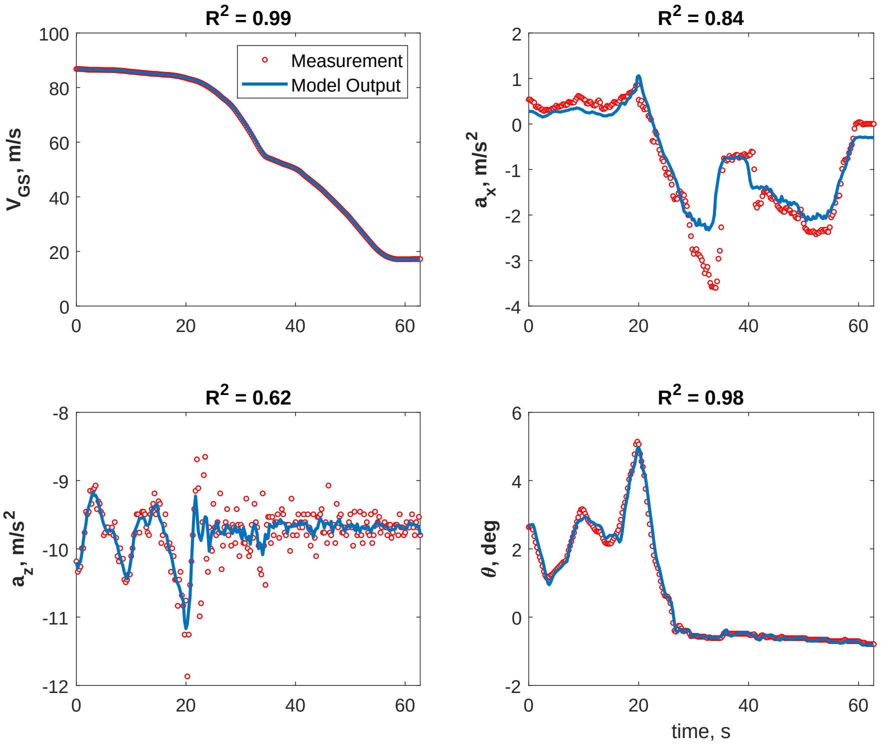


Figure C.6: Phase 2 proof of match - flight B10896

Appendix D

List of Publications

1. Javensius Sembiring, Ludwig Drees, and Florian Holzapfel. Extracting unmeasured parameters based on quick access recorder data using parameter-estimation method. In *AIAA Atmospheric Flight Mechanics (AFM) Conference*, page 4848, 2013

2. Ludwig Drees, Heiko Haselhofer, Javensius Sembiring, and Florian Holzapfel. Modeling the flare maneuver performed by airline pilots using flight operation data. In *AIAA Modeling and Simulation Technologies (MST) Conference*, page 4912, 2013

3. International Air Transport Association (IATA). *Safety Report*. International Air Transport Association, 800 Place Victoria, P.O. Box 113, Montreal, Quebec, Canada H4Z 1M1, 4 2014. ISBN: 978-92-9252-349-7. Chapter 9 Predictive Analysis

4. Javensius Sembiring, Lukas Höhndorf, and Florian Holzapfel. Bayesian approach implementation on quick access recorder data for estimating parameters and model validation. *Probabilistic Safety Assessment and Management PSAM*, 12, 2014

5. Lukas Höhndorf, Javensius Sembiring, and Florian Holzapfel. Copulas applied to flight data analysis. *Probabilistic Safety Assessment and Management PSAM*, 12, 2014

6. Chong Wang, Ludwig Drees, Nadine Gissibl, Lukas Höhndorf, Javensius Sembiring, and Florian Holzapfel. Quantification of incident probabilities using physical and statistical approaches. In *6th International Conference on Research in Air Transportation. Istanbul, Turkey*, 2014

7. Lukas Höhndorf, Joachim Siegel, Javensius Sembiring, Phillip Koppitz, and Florian Holzapfel. Reconstruction of aircraft trajectories during landing using a rauch-tung-striebel smoother, instrument landing system deviation information, and taxiway locations. In *AIAA Atmospheric Flight Mechanics Conference*, page 3705, 2016

8. Lukas Höhndorf, Joachim Siegel, Javensius Sembiring, Phillip Koppitz, and Florian Holzapfel. Reconstruction of aircraft states during landing based on quick access recorder data. *Journal of Guidance, Control, and Dynamics*, 40(9):2393–2398, 2017

9. Javensius Sembiring, Changwu Liu, Phillip Koppitz, and Florian Holzapfel. Energy management for unstable approach detection. In *2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 1–6. IEEE, 2018

10. Javensius Sembiring, Joachim Siegel, and Florian Holzapfel. Parameter estimation on low observability data. In *2018 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 1–7. IEEE, 2018

11. Phillip Koppitz, Chong Wang, Lukas Höhndorf, Javensius Sembiring, Xiaolong Wang, and Florian Holzapfel. From raw operational flight data to incident probabilities using subset simulation and a complex thrust model. In *AIAA Scitech 2019 Forum*, page 2233, 2019

12. Xiaolong Wang, Javensius Sembiring, Phillip Koppitz, Lukas Höhndorf, Chong Wang, and Florian Holzapfel. Modeling of the aircraft's low energy state during the final approach phase using operational flight data. In *AIAA Scitech 2019 Forum*, page 0989, 2019