Technische Universität München

Lehrstuhl für Regelungstechnik

Centralized and Distributed Moving Horizon Strategies for State Estimation of Networked Control Systems

Peter Philipp

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

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Florian Holzapfel
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. habil. Boris Lohmann
	2. Univ.-Prof. Dr.-Ing. Jan Lunze, Ruhr-Universität Bochum

Die Dissertation wurde am 16.10.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 09.01.2014 angenommen.

To my parents,
for making it possible to embark
on this journey and giving me love
and support to conclude it.

Acknowledgments

The results presented in this thesis are the outcome of the research I performed during my time as a research assistant at the Institute of Automatic Control, Technische Universität München, in the years 2007 to 2013. In this context, I am very grateful to the Priority Program 1305, Control Theory of Digitally Networked Dynamical Systems of the German Research Foundation (DFG) for their financial support.

My first thanks go to Prof. Dr.-Ing. habil. Boris Lohmann for offering me the unique opportunity to be part of an open-minded research group and endowing me with lots of freedom in my research. Moreover, he supported my trips to many international conferences in order to present my work, to see the most recent results of other researchers, and to get into contact with other scientists in my field.

Second, I am deeply indebted to Univ.-Prof. Dr.-Ing. Jan Lunze and Univ.-Prof. Dr.-Ing. Florian Holzapfel for their interest in my research and for accepting to be members of my doctoral exam committee. The dedication and engagement of Univ.-Prof. Dr.-Ing. Jan Lunze within the Priority Program 1305 have been remarkable and deserve a special mention.

My thanks also go to the members of the Institute of Automatic Control. They created an excellent atmosphere full of interesting discussions on and off the job. Especially, I would like to thank my office mate Klaus Diepold and my colleagues Rudy Eid, Tobias Kloiber, Simon Altmannshofer, Paul De Monte, Enrico Pellegrini, Sebastian Bürger and Michael Buhl. I am also grateful to my students Thomas Schmid-Zurek, Michael Schneider, Agnes Gabrys and Florian Peter. Moreover, I would like to express my sincere thanks to Sebastian Pieczona for his support during the final stages.

Last but not least, I want to thank my parents Isolde and Günter, my sister Martina, my brother-in-law Thomas and my niece Leonie for their love, care and support throughout all my years of education and for finding the right words to keep me on track.

Neufahrn b. Freising, February 2014
Peter Philipp

Contents

Notation

Conventions

Scalars are denoted by upper and lower case letters in italic type. *Vectors* are denoted by lower case letters in boldface type, as the vector $\boldsymbol{v}$ is composed of elements ${}_{i}v$. *Matrices* are denoted by upper case letters in boldface type, as the matrix $\boldsymbol{M}$ is composed of elements ${}_{i,j}M$ (i-th row, j-th column). *Sets* are denoted by upper case letters in blackboard bold type or calligraphic type.

$a, b, \phi, \omega, \Phi, \Omega$	Scalars
$a(\cdot), b(\cdot), \phi(\cdot), \omega(\cdot), \Phi(\cdot), \Omega(\cdot)$	Scalar functions
$\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\phi}, \boldsymbol{\omega}$	Vectors
$\boldsymbol{a}(\cdot), \boldsymbol{b}(\cdot), \boldsymbol{\phi}(\cdot), \boldsymbol{\omega}(\cdot)$	Vector functions
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Phi}, \boldsymbol{\Omega}$	Matrices
$\boldsymbol{A}(\cdot), \boldsymbol{B}(\cdot), \boldsymbol{\Phi}(\cdot), \boldsymbol{\Omega}(\cdot)$	Matrix functions
$\mathbb{A}, \mathbb{B}, \mathcal{A}, \mathcal{B}$	Sets

Sets

$\mathbb{N}$	Set of all nonnegative natural numbers (without zero)
$\mathbb{N}_i$	Set of all natural numbers greater or equal than i, i. e. $\mathbb{N}_i \triangleq \{n \in \mathbb{N} \vert n \geq i\}$
$\mathbb{R}$	Set of all real numbers
$\mathbb{R}_i$	Set of all real numbers great or equal than i, i. e. $\mathbb{R}_i \triangleq \{n \in \mathbb{R} \vert n \geq i\}$
$\mathbb{R}^n$	Set of all real vectors with n components
$\mathbb{R}^{n \times m}$	Set of all real $n \times m$ matrices

Arguments, Mathematical Accents, Subscripts and Superscripts

$(\cdot)[l]$	Iteration l of the scalar, vector or matrix $(\cdot)$

$\hat{(\cdot)}$	Estimated value of the scalar, vector, matrix or set $(\cdot)$
$\bar{(\cdot)}$	Predicted value of the scalar or vector $(\cdot)$
$\tilde{(\cdot)}$	Scalar, vector, matrix or set $(\cdot)$ belonging to the controller
$\underline{(\cdot)}$	Vector or matrix collecting similar scalars, vectors or matrices $(\cdot)$
${}_i(\cdot)$	The i-th element of the vector or set $(\cdot)$
$(\cdot)^T$	Transpose of the vector or matrix $(\cdot)$
$(\cdot)^\circ$	Initial value of the scalar or vector $(\cdot)$
$(\cdot)^*$	Optimal value of the scalar or vector $(\cdot)$
$(\cdot)^{(i)}$	Scalar, vector, matrix or set $(\cdot)$ belonging to the i-th actuator
$(\cdot)^{[i]}$	Scalar, vector, matrix or set $(\cdot)$ belonging to the i-th sensor

Operators and Special Symbols

$\triangleq$	Definition; $A \triangleq B$ defines A to be equal to B
$\mathbf{0}$	Column vector or matrix with all elements equal to zero
$\boldsymbol{I}$	Identity matrix
$\lvert\mathcal{A}\rvert$	Cardinality of the set $\mathcal{A}$, i. e. its number of elements
$\lvert x\rvert$	Magnitude or absolute value of scalar $x \in \mathbb{R}$
$\lVert\boldsymbol{x}\rVert$	2-norm or Euclidean norm of the vector $\boldsymbol{x} \in \mathbb{R}^n$, $\lVert\boldsymbol{x}\rVert \triangleq \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$
$\lVert\boldsymbol{x}\rVert_{\boldsymbol{A}}$	Weighted 2-norm or weighted Euclidean norm of the vector $\boldsymbol{x} \in \mathbb{R}^n$ with the positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\lVert\boldsymbol{x}\rVert_{\boldsymbol{A}} \triangleq \sqrt{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}$
$\boldsymbol{a} > \boldsymbol{b}$	${}_ia > {}_ib$ for all i
$\boldsymbol{a} \geq \boldsymbol{b}$	${}_ia > {}_ib$ for all i
$\boldsymbol{A} > \boldsymbol{B}$	The square matrix $\boldsymbol{A} - \boldsymbol{B}$ is positive definite
$\boldsymbol{A} \geq \boldsymbol{B}$	The square matrix $\boldsymbol{A} - \boldsymbol{B}$ is positive semi-definite
$\lambda_{\min}(\boldsymbol{A})$	Minimum eigenvalue of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$
$\mathrm{col}(\boldsymbol{x}, \boldsymbol{y})$	Stacked column vector of the vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$, $\mathrm{col}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \left[\boldsymbol{x}^T, \boldsymbol{y}^T\right]^T$
$\mathrm{col}(\boldsymbol{x}^{(i)}, i \in \mathcal{I})$	Stacked and ordered column vector of the vectors $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ whose index i belongs to the index set $i \in \mathcal{I} = \{1, 2, \ldots, I\}$, $\mathrm{col}(\boldsymbol{x}^{(i)}, i \in \mathcal{I}) \triangleq \mathrm{col}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(I)})$
$\mathrm{diag}(\boldsymbol{A}, \ldots, \boldsymbol{Z})$	Block diagonal matrix with the square matrix block $\boldsymbol{A}, \ldots, \boldsymbol{Z} \in \mathbb{R}^{n\times n}$
$\ker(\boldsymbol{A})$	Kernel of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$
$\mathrm{rank}(\boldsymbol{A})$	Rank of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$

Derivatives

$$\frac{\partial f}{\partial \boldsymbol{x}} \triangleq \begin{bmatrix} \frac{\partial f}{\partial_1 x} \\ \vdots \\ \frac{\partial f}{\partial_n x} \end{bmatrix}$$ Gradient/First-order derivative of $f \in \mathbb{R}$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \triangleq \begin{bmatrix} \frac{\partial_1 f}{\partial_1 x} & \cdots & \frac{\partial_1 f}{\partial_n x} \\ \vdots & & \vdots \\ \frac{\partial_m f}{\partial_1 x} & \cdots & \frac{\partial_m f}{\partial_n x} \end{bmatrix}$$ Jacobian/First-order derivative of $\boldsymbol{f} \in \mathbb{R}^m$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$

$$\dot{\boldsymbol{x}} \triangleq \begin{bmatrix} \frac{\partial_1 f}{\partial t} \\ \vdots \\ \frac{\partial_m f}{\partial t} \end{bmatrix}$$ First-order derivative of $\boldsymbol{f} \in \mathbb{R}^m$ with respect to $t \in \mathbb{R}$

$$\frac{\partial^2 f}{\partial \boldsymbol{x}^2} \triangleq \begin{bmatrix} \frac{\partial^2 f}{\partial_1 x \partial_1 x} & \cdots & \frac{\partial^2 f}{\partial_1 x \partial_n x} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial_n x \partial_1 x} & \cdots & \frac{\partial^2 f}{\partial_n x \partial_n x} \end{bmatrix}$$ Hessian/Second-order derivative of $f \in \mathbb{R}$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$

$$\ddot{\boldsymbol{x}} \triangleq \begin{bmatrix} \frac{\partial^2{}_1 f}{\partial t^2} \\ \vdots \\ \frac{\partial^2{}_m f}{\partial t^2} \end{bmatrix}$$ Second-order derivative of $\boldsymbol{f} \in \mathbb{R}^m$ with respect to $t \in \mathbb{R}$

Acronyms

BDA	Bilevel Decomposition Algorithm
BSE	Batch State Estimator
BVP	Boundary Value Problem
CAN	Control Area Network
CDEKF	Continuous-Discrete Extended Kalman Filter
CKF	Centralized Kalman Filter
CMHE	Centralized Moving Horizon Estimator
CMHO	Centralized Moving Horizon Observer
CMHS	Centralized Moving Horizon Strategies
CSTR	Continuously Stirred Tank Reactor
DKF	Distributed Kalman Filter
DMHE	Distributed Moving Horizon Estimator
DMHO	Distributed Moving Horizon Observer
DMHS	Distributed Moving Horizon Strategies
DSP	Digital Signal Processor

EKF	Extended Kalman Filter
FPGA	Field Programmable Gate Array
FTSP	Flooding Time Synchronization Protocol
HJB	Hamilton-Jacobi Bellman
IP	Interior Point
ISO	International Standards Organization
ISS	Input-to-State Stable
IVP	Initial Value Problem
KF	Kalman Filter
KKT	Karush-Kuhn-Tucker
LICQ	Linear Independence Constraint Qualification
LP	Linear Program
LQR	Linear Quadratic Regulator
MAC	Medium Access Control
MHE	Moving Horizon Estimator
MHO	Moving Horizon Observer
MHS	Moving Horizon Strategies
MPBVP	Multi-Point Boundary Value Problem
MPC	Model Predictive Control
NCS	Networked Control System
NLP	Nonlinear Program
NTP	Network Time Protocol
ODE	Ordinary Differential Equation
OSI	Open Systems Interconnection
PDE	Partial Differential Equation
PI	Performance Index
PVD	Parallel Variable Distribution
QP	Quadratic Program
RBS	Reference Broadcast Synchronization
RMSE	Root Mean Square Error
SISO	Single Input Single Output
SQP	Sequential Quadratic Programming
SVD	Singular Value Decomposition
TDMA	Time-Division Multiple Access

Part I

Preliminaries

Chapter 1

Introduction

The control of dynamical systems requires data transmission between the spatially distributed components of the control loop. In the most basic case, this comprises the transmission of sensor information to a controller and of control input from the controller to the actuators. For conventional control systems, the data is transferred via hardwired point-to-point connections. However, this type of connection causes high costs of wiring and provides less flexibility for introducing additional components into the control loop as the needs change. These shortcomings initiated the search for alternative concepts of data transmission in control systems. In the 1980's, the concept of data transmission over a digital network emerged as the sought after solution and the *networked control systems* (NCSs) were born. One of the earliest efforts along the lines of modern networked control systems was the study started in 1983 by the Bosch GmbH to investigate the feasibility of networked devices to control different functions in a passenger car. This study was very successful and in 1986 the innovative communication protocol of the control area network (CAN) was announced and corresponding hardware was available by 1987. Today, almost all cars manufactured in Europe include embedded systems integrated through CAN. The use of NCSs has not been restricted to the automobile industry but has also found application in a broad range of other areas such as manufacturing automation (Biegacki and VanGompel, 1996; Schickhuber and McCarthy, 1997), aircraft (Seiler, 2001), remote surgery (Meng et al., 2004; Yogesan et al., 2006), mobile sensor networks (Ogren et al., 2004) and smart grid (Berger and Iniewski, 2012), to name only a few. These examples illustrate the growing interest in NCSs which is motivated by the benefits they offer compared to conventional point-to-point wired control systems. Modern NCSs have surpassed by far the original intended and above mentioned objectives and offer a variety of additional advantages. Besides the reduced volume of wiring, NCSs are more reliable due to fewer physical potential points of failure such as connectors and cable harness. This results in an increased capability of troubleshooting and maintenance which significantly reduces the installation and operation costs. The enhanced interchangeability and interoperability of devices offers easy reconfigurability resulting in increased flexibility. Maybe one of the most significant advantage of NCSs is the feature of enabling the realization of new control directions which are impossible with conventional point-to-point connections. Traditionally, conventional control systems with multiple actuators and multiple sensors have

been studied within either the *centralized* or *decentralized* framework. In the former case, the measurements from the different sensors are collected by a central unit for processing before the resulting control inputs are sent to the different actuators (Kailath, 1980; Lunze, 1992; Goodwin et al., 2001; Khalil, 2002). In the latter case, the system is decomposed into a number of simpler but interconnected subsystems which are controlled by local isolated units (Sandell et al., 1978; Siljak, 2007; Bakule, 2008). While both frameworks are capable of stabilizing a dynamical system, the centralized framework possesses, in general, superior performance due to the explicit consideration of the full system dynamics in the centralized controller design. However, the installation and maintenance costs as well as the robustness are better for the decentralized framework due to the reduced wiring and the divide and conquer scheme of the decentralized controller. In contrast, the utilization of networks for data transmission enables the creation of novel and innovative distributed control systems realized not only horizontally, e. g. peer-to-peer coordinated control among sensors and actuators, but also vertically, e. g. control over different levels (machine to cell to system level), like in manufacturing automation (Biegacki and VanGompel, 1996; Schickhuber and McCarthy, 1997), smart grid (Berger and Iniewski, 2012) and building automation (Newman, 1996). The inclusion of the Internet in the control system achieves a global interconnection and creates new possibilities in a world wide fashion like in teleoperation (Hirche, 2005), remote surgery (Meng et al., 2004; Yogesan et al., 2006) and haptics collaboration over the Internet (Hespanha et al., 2000). Moreover, the recent progress in electronics opened up the possibility of including low-cost wireless devices. These devices facilitates the expansion of the application to mobile objects because information can now be transmitted over wireless connections from nearly every place and without the need of hard-wired connections. Some remarkable applications in this context are mobile sensor networks (Ogren et al., 2004), coverage control (Cortés et al., 2004) and environmental monitoring and surveillance (Akyildiz et al., 2002).

1.1 Challenges in Networked Control Systems

The core difference between data transmission over networks and conventional hardwired point-to-point connections is that in the former case, data is transmitted in atomic units called *packets*. The transmission of these packets is governed by *protocols*. The functionality of protocols is commonly described and differentiated utilizing the International Standards Organization - Open Systems Interconnection (ISO-OSI) seven layer reference model (ISO 7498; Zimmermann, 1980). The seven layers are physical, data link, network, transport, session, presentation, and application. Each layer is a collection of similar functions which provide services to the layer above it and receives service from the layer below it. For instance, the physical layer defines electrical and physical specifications for the devices and provides the service to establish, maintain and terminate connections to a communication medium and to transmit information. This could be in the form of electrical or optical signals for wired networks and electromagnetic waves for wireless networks. The choice of the different protocols

for each layer and the resulting interaction determines the properties of data transmission over networks. Compared to conventional data transmission over hardwired point-to-point connections, several new issues are introduced which make NCSs distinct from other control systems: In this thesis, we consider the following ones:

- *Packet Delays & Packet Reordering:* Every information transmitted over a network has to be encoded in a digital format first, then transmitted over the network and finally decoded at the receiver side. The execution time of these processes accumulate to the overall packet delay. This delay can be highly variable since the network access time, i. e. the time for a shared network to accept data, and the transmission delays, i. e. the time during which data are in transit inside the network, depend on highly variable network conditions such as the network status and traffic. Long transmission delays can sometimes result in packet reordering, i. e. the sequence of transmitted and arrived packets differs.
- *Packet Drops:* While the packets are in transit through the network, the possibility of losing a packet exists. Typical sources for packet drops are buffer overflows due to congestion or transmission errors in physical network links. The later is far more common in wireless than wired networks.
- *Synchronization:* Each device in a network is equipped with an individual clock which determines the time when an action is executed. However, these clocks will differ after some amount of time even when all clocks are initially perfectly tuned because of the imperfections of the clock oscillators. This has severe effects on the network since, for instance, the transmission scheduling in a network often requires a common notion of time like in the case of time division multiple access (TDMA) based protocols. Thus, clock synchronization is of fundamental importance for all networks since it enables the successful communication between the nodes on the network which leads to a smooth operation of the NCSs.
- *Limited Energy Supplies:* The energy supplies of some devices in a network can be limited. For instance, wireless sensor nodes consist of a micro-controller, sensors, a transceiver unit and a limited power source. The by far most energy consuming task for such a device is data transmission. The energy cost of transmitting a packet of size 1 kB a distance of 100 m is approximately the same as that for executing 3 million instruction by a 100 million instruction per second/W processor (Pottie and Kaiser, 2000).

The protocols of the second layer, or more precisely, the medium access control (MAC) sublayer of the second layer, determine essentially the packet delays and packet drops (Xia and Sun, 2008; Hristu-Varsakelis and Levine, 2005). This relation is investigated for some of the most common MAC protocols (Ethernet, ControlNet and DeviceNet) from a NCS perspective in Lian et al. (2001).

The common remedy to solve the synchronization problem is the utilization of clock synchronization protocols which establish a common notion of time for all devices in the network.

However, this task is anything but trivial and requires dedicated methods, like the network time protocol (NTP) (Mills, 1991), reference broadcast synchronization (RBS) (Elson et al., 2002) and flooding time synchronization protocol (FTSP) (Maroti et al., 2004). The higher the requirements on the synchronization precision are, the more and the more frequently packets have to be transmitted over the network solely for synchronization. This reduces both the available capacity of the network for the actual control or estimation task and the limited energy supplies. As a consequence, clock synchronization is one of the most critical components contributing to energy consumption for wireless devices due to the highly energy consuming radio transmissions for delivering timing information (Wu et al., 2011). Additional information about this topic including surveys about different clock synchronization techniques can be found in Elson (2003), Sundararaman et al. (2005) and Wu et al. (2011).

In the recent years, the theory of networked control systems has become one of the most active research areas in the control community. It is beyond the scope of this thesis to discuss all important topics in this area. Therefore, we refer the interested reader to the following surveys, special issues and books: Antsaklis and Baillieul (2004); Hespanha et al. (2007); Antsaklis and Baillieul (2007); Moyne and Tilbury (2007); Chiang et al. (2007); Zampieri (2008); Bemporad et al. (2010); Lunze (2014).

1.2 Moving Horizon Strategies for State Estimation

Both conventional and networked control systems often require knowledge about the full state of the system to solve the respective control problems. In most practical cases, however, measurements for all states are not available. On the one hand, this is often caused by the fact that suitable sensors to measure certain states do not exist. On the other, we can observe the trend of reducing the installation and maintenance costs simply by saving sensors. In any case, if knowledge about the full state is unavailable but required, the full state of the system needs to be estimated from the possibly noisy measurements of the sensors in combination with a system model gained by physical insight. From an engineering point of view, it is quiet natural not to be satisfied with any estimate but to ask for the optimal one. This problem can be tackled within either the deterministic or stochastic framework. In the former case, one has to minimize a residual function, while in the latter case one has to maximize a conditional density probability function. Interestingly, even though both problems originated from different perspectives, they are mathematically equivalent and result in the identical infinite-horizon optimal estimation problem. The term infinite-horizon indicates that the size of the problem grows without bound as the number of measurements increases. For this reason, it is impossible to find analytical or numerical solutions, in general, with the remarkable exception of linear unconstrained systems. In this case, a recursive solution method exists which is the well-known Kalman filter. In all other cases, one needs some form of data compression to handle the ever growing influx of measurements and thus to keep the problem size manageable.

One powerful method are *moving horizon strategies* (MHS) which come in two flavors: *moving horizon observers* (MHOs) and *moving horizon estimators* (MHEs). The main difference between both is that only the latter considers state disturbances. The key idea behind the MHS is to reformulate the infinite-horizon optimal estimation problem as a sequence of finite-horizon optimal estimation problems. The basic strategy is to estimate the state using a moving but fixed-size window of data and to compress the remaining data in an arrival cost. Whenever a new measurement becomes available, the oldest measurement is removed from the data window and the newest measurement is added. As a result, the problem size of the estimation problem is bounded since only a subset of the available information is used directly. The remaining ones are taken into account indirectly via the arrival cost. The MHS possess some significant features which makes their application appealing not only for conventional but also for networked control systems: First, the MHS approach an optimal performance for linear as well as nonlinear systems. This means that for the linear unconstrained case, the MHE is identical to a Kalman filter. For the nonlinear case, however, the MHE is superior to any traditional recursive estimation methods such as the extended Kalman filter. Second, because the MHS are formulated as an optimal estimation problem, it is straightforward to incorporate additional equality and inequality constraints in the estimation process. Third, MHS can explicitly incorporate more than one measurement in the estimation process as opposed to recursive estimation methods. However, the price to pay is an increased complexity since a fixed-size optimal estimation problem has to be solved whenever a new measurement is available.

For conventional control systems, the body of research around MHS for state estimation is deep and diverse. The paper of Bryson and Frazier (1963) is one of the first works where the connection between Kalman filtering and the equivalent formulation as an optimal infinite-horizon estimation problem is shown. In the sequel, several early formulations of linear unconstrained MHEs, sometimes referred to as limited memory filters, were presented in Schweppe (1964); Jazwinski (1968); Thomas (1975). Since the computational power was severely limited and expensive at that time, only recursive solution methods were regarded as computational feasible. This imposed severe restrictions for state estimation of general nonlinear systems since in this case the problem of calculating exact recursive solutions becomes infinite dimensional and thus impossible to compute (Kushner, 1964). The common remedy to this problem was to introduce some forms of approximation which led to several techniques for nonlinear systems. The extended Kalman filter has been derived by linearizing the nonlinear models through a first-order Taylor series around the current estimate (Cox, 1964). A first version of nonlinear MHS has been presented in Jang et al. (1986). In the following years, this work was extended by Tjoa and Biegler (1991); Liebman et al. (1992); Muske et al. (1993). Subsequent research focused on improving the optimality and stability properties which resulted in effective and stable MHE formulations. This was addressed for nonlinear MHOs in Zimmer (1994); Moraal and Grizzle (1995); Michalska and Mayne (1995); Alamir (1999); Alamir and Calvillo-Corona (2002), for linear MHEs in Rao et al. (2001); Alessandri et al. (2003) and

for nonlinear MHEs in Robertson et al. (1996); Rao (2000); Rao et al. (2003); Zavala (2008); Alessandri et al. (2008).

For networked control systems, the body of research around MHS for state estimation is less extensive. This is especially evident for nonlinear NCSs where only few results have been achieved. For *linear NCSs*, some literature is available. In Goodwin et al. (2004), a MHE is presented as part of a control system design strategy where the controller, sensors and actuators are connected via a *data-rate limited channel*. In order to minimize bandwidth utilization, a communication constraint is imposed which restricts all transmitted data to belong to a finite set. The resulting quantization issues are solved by means of the moving horizon technique. The problem of *packet delays* for the transmission of measurement data is considered in Zou and Li (2009); Valencia et al. (2011). In Zou and Li (2009), it is assumed that the delays are restricted to be less than one sample period and the problem is solved by a suitable MHE formulation within the framework of multirate control systems. In Valencia et al. (2011), the packet delays are assumed to be a multiple of the sampling period. The authors propose a MHE with variable structure to deal with this situation. However, the stability of the MHE is not investigated. The effect of *packet drops* is investigated in Xue et al. (2012a,b); Liu et al. (2012, 2013). In Xue et al. (2012a), packet drops in the sensor-to-controller channel are considered. Based on a linear model for the packet drops with stochastic parameters, a MHE combined with a local observer is proposed to deal with the uncertainty from the lossy network. This approach is extended in Xue et al. (2012a) to the case of multiple packet drops not only in the sensor-to-controller channel but also in the controller-to-actuator channel. The work of Liu et al. (2012) models the packet drops for the transmitted measurement data as a binary switching random sequence. This approach results in a MHE design for which a maximum packet dropout probability can be given such that convergence of the state estimation error is achieved. The authors extended their work in Liu et al. (2013) to tackle simultaneously the problem of packet drops and quantized measurements resulting from communication constraints. Thereby, the packet drops are modeled as before as a binary switching random sequence and the measurements are quantized by a logarithmic quantizer. The convergence of the resulting MHE depends on the packet dropout probability and on the quantization density. For both parameters maximum values are derived to ensure the convergence of the state estimation error. The problem of *restricted network access* is investigated in Xue et al. (2013) where the plant has multiple sensor nodes and only some of them are allowed to communicate with the remote estimator through a limited-bandwidth network at each time instant. The key to derive a suitable MHE is the presented communication logic sequence which enables the derivation of sufficient conditions for the boundedness of the square norm of the estimation error.

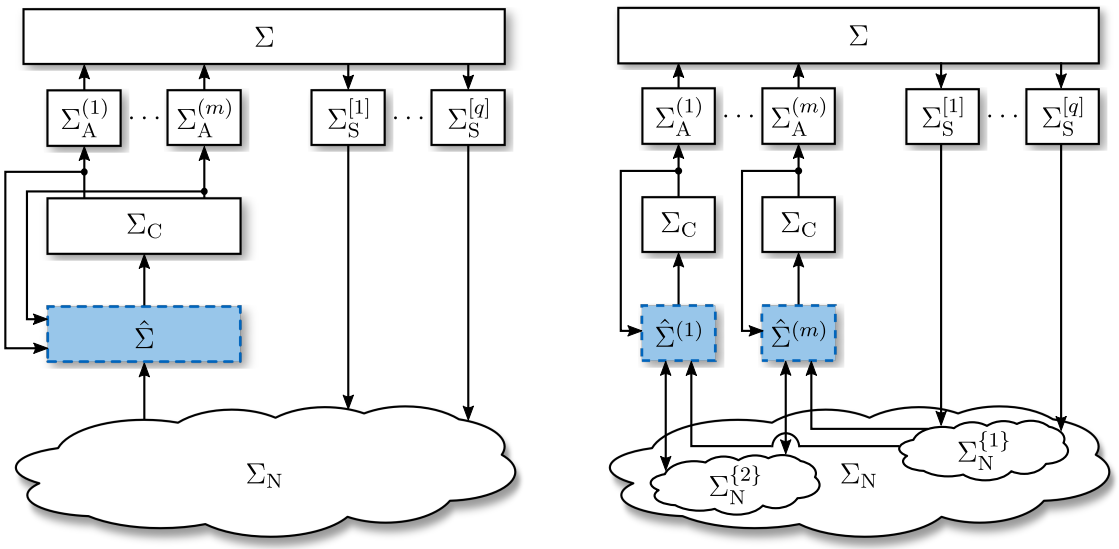
For *nonlinear NCSs*, only very few literature is available. In Jin et al. (2007), an extended Kalman filter (EKF) and a MHE are discussed for state estimation over *packet-dropping networks*. Sufficient conditions are presented for the EKF that guarantee a bounded EKF error covariance. The authors propose a scheme for organizing the moving horizon to handle

intermittent observations. However, the stability of the MHE is not investigated. The work of Ji'an et al. (2008) investigates the problem of *quantized measurements* and discuss a MHE from a practical point view, i. e. without providing a stability analysis. In the upcoming work of Johansen et al. (2013), a MHO is presented to deal with the problem of simultaneous *packet loss and delay* for the transmitted measurement data. This is achieved by deriving a suitable regularized formulation of the estimation problem within the moving horizon framework. This includes the addition of stabilizing terms in the cost function coupled with an SVD-based weight selection method which avoids drift of the estimates when the data in the current moving horizon is not sufficiently informative. Sufficient conditions for exponential stability of the observer error are provided.

To sum up, the body of research around MHS for state estimation of conventional control systems is deep and diverse and important results are available in the literature. However, this is not the case for networked control systems, especially for nonlinear NCSs. As a consequence, there are still many open issues which have to be resolved before all advantages of moving horizon strategies for state estimation of NCSs can be harvested.

1.3 Scope of the Thesis

The goal of this thesis is the development of moving horizon strategies for state estimation of networked control systems. To this end, we first consider the *centralized NCS architecture* depicted in Figure 1.1(a) where the measurements of the *nonlinear system* Σ generated by the different sensors $\Sigma_{\mathrm{S}}^{[i]}$ are transmitted in packets over the network Σ_{N} to a centralized observer/estimator $\hat{\Sigma}$. For this scenario, we develop two centralized moving horizon strate-



(a) Centralized NCS architecture.

(b) Distributed NCS architecture.

Figure 1.1: Considered NCS architectures with system Σ, sensors $\Sigma_{\mathrm{S}}^{[i]}$, networks $\Sigma_{\mathrm{N}}^{\{l\}}$, observers/estimators $\hat{\Sigma}^{(j)}$, controller Σ_{C} and actuators $\Sigma_{\mathrm{A}}^{(j)}$.

gies for state estimation, namely the *centralized moving horizon observer* (CMHO) and the *centralized moving horizon estimator* (CMHE). While the CMHO is constructed for the case of undisturbed system and sensors, the CMHE takes into account disturbances acting on the system and the sensors. Both strategies are explicitly designed to deal with the following simultaneously appearing imperfections induced by the data transmission over the network Σ_{N}:

- unknown and variable packet delays which include the possibility of packet reordering,
- unknown and variable packet drops,
- unsynchronized sensor clocks, and
- limited energy supplies of the sensors.

Moreover, we develop methods which facilitate the real-time implementation of both strategies, discuss the problem of observability of undisturbed and disturbed NCSs, analyze the stability of both strategies and validate their performance not only in simulations but also in experiments on a test-rig.

Next, we combine the experience gained from the centralized setting with the capability of NCSs to realize new control directions to propose a novel *distributed NCS architecture* which combines the benefits of the centralized and decentralized framework of conventional control systems. This architecture facilitates the *decentralized implementation of any centralized controller* but requires distributed knowledge about the full state of the system. For this task, distributed versions of the developed centralized moving horizon strategies for state estimation are predestined. To this end, we consider the novel distributed NCS architecture depicted in Figure 1.1(b) where the measurements of the *linear system* Σ generated by the different sensors $\Sigma_{\mathrm{S}}^{[i]}$ are transmitted in packets over the network $\Sigma_{\mathrm{N}}^{\{1\}}$ to different distributed observers/estimators $\hat{\Sigma}^{(i)}$ which are interconnected by the network $\Sigma_{\mathrm{N}}^{\{2\}}$. For this scenario, we develop two distributed moving horizon strategies for state estimation, namely the *distributed moving horizon observer* (DMHO) and the *distributed moving horizon estimator* (DMHE). Similar as before, the DMHO is constructed for the case of an undisturbed system, while the DMHE takes into account disturbances acting on the system and the sensors. Both strategies are explicitly designed to deal simultaneously with the above stated imperfections induced by the data transmission over the network $\Sigma_{\mathrm{N}}^{\{1\}}$ with the exception of synchronized instead of unsynchronized clocks. Moreover, for both strategies, we discuss the allocation of the measurements over the network $\Sigma_{\mathrm{N}}^{\{1\}}$ to the distributed observers/estimators, investigate the impact of the topology of the network $\Sigma_{\mathrm{N}}^{\{2\}}$, analyze the stability and validate the performance in simulations.

1.4 Outline and Contributions of the Thesis

The following overview reveals the outline of this thesis and briefly summarizes its contributions.

Part I: Preliminaries

Chapter 2: Preliminaries

In this chapter, we present the mathematical building blocks, on which this thesis is built. We start with reviewing moving horizon strategies for state estimation of nonlinear systems with discrete-time measurements which leads to nonlinear optimal estimation problems on moving horizons. An important topic in this context is the possibility of reconstructing the states. This issue is discussed under the aspect of observability of discrete-time nonlinear systems. Moreover, we give an overview of different solution methods for optimal state estimation problems. The most promising and efficient methods require the solution of nonlinear optimization problems. Hence, we introduce important notions for nonlinear optimization and review related and relevant standard solution techniques. These techniques require the calculation of certain derivatives which can be calculated according to one of the presented methods. Finally, we introduce some graph related notation which we will use for distributed state estimation. The main contribution of this chapter is:

- Presentation of the mathematical building blocks of this thesis.

Part II: Centralized Moving Horizon Strategies

Chapter 3: Centralized Moving Horizon Strategies

This chapter presents the centralized moving horizon observer (CMHO) and the centralized moving horizon estimator (CMHE) for state estimation of nonlinear systems within a common framework for the centralized NCS architecture. The CMHO is constructed for the case of undisturbed system and sensors, while the CMHE considers disturbances acting on the system and the sensors. Both strategies are explicitly designed to deal simultaneously with the network-induced imperfections of unknown and variable packet delays which include the possibility of packet reordering, unknown and variable packet drops, unsynchronized sensor clocks, and limited energy supplies of the sensors. To overcome these challenges, we consider event-based sampling, introduce time stamps for the measurements, propose an affine clock model for the sensor clocks and extend the moving horizon to a buffer logic. This buffer serves as the information basis for both centralized moving horizon strategies and enables us along with the other introduced steps to formulate the state estimation problems as suitable optimization problems where we additionally estimate the unknown clock parameters of the sensors. To achieve a practical feasible implementation of these optimization problems in real-time and thus of the CMHO and CMHE, we propose the following two steps which significantly contribute to this goal. First, we introduce a suboptimal approach which requires only suboptimal instead of optimal solutions. Second, we provide efficient methods for generating proper initial conditions for finding (sub)optimal solutions. The main contributions of this chapter are:

- Development of the CMHO and CMHE for state estimation of nonlinear systems within a common framework for the centralized NCS architecture.
- Development of two measures which contribute to the goal of realizing a real-time implementation of the CMHO and CMHE.

Parts of the results presented in this chapter have been published in Philipp and Lohmann (2009); Philipp (2009); Philipp and Lohmann (2011, 2014).

Chapter 4: Efficient Derivative Calculation

This chapter proposes new methods to efficiently calculate the derivatives required by the CMHO and CMHE for deriving (sub)optimal solutions to the optimization problems. More precisely, we present for the CMHO as well as the CMHE a method, which calculates the exact gradients and excellent approximations of the Hessians of the Lagrangians corresponding to the respective optimization problems. The key elements of these methods are the first-order state sensitivities which describe how sensitive the state acts on changes in the optimization variables. The proposed derivative calculation methods do not only significantly contribute to the goal of achieving a practical feasible implementation of the CMHO and CMHE but play also a central role in the observability and stability analysis of Chapter 5 and 6, respectively. The main contribution of this chapter is:

- Development of new methods for efficiently calculating the gradient and Hessian of the Lagrangian corresponding to the optimization problems of the CMHO and CMHE which exploits the structure not only in the derivatives but also in the first-order state sensitivities.

This chapter is mainly based on Philipp (2011a,b).

Chapter 5: Observability of Networked Control Systems

In this chapter, we deal with the question of observability of networked control systems. We investigate when it is possible to uniquely reconstruct all unknown states and parameters of the sensor clocks from the information stored in the buffer. This possibility depends for the given combined state and parameter estimation problem not only on the structure of the system but also on the information content in the control input. We formulate this relation as a sufficient excitation property of the control input. The main contributions of this chapter are:

- Introduction of an observability notion for undisturbed and disturbed NCSs which ensures the well-posedness of the respective observability map defined on the buffer.
- Derivation of necessary conditions for the control input of undisturbed and disturbed NCSs to be sufficiently exciting which ensures in both cases the possibility of uniquely reconstructing all unknown parameters from the information stored in the buffer.

Parts of this chapter are based on Philipp (2012).

Chapter 6: Stability Analysis

In this chapter, we analyze the stability of the CMHO and CMHE. To this end, we introduce a unifying representation for the optimization algorithms presented in Chapter 2 which utilize the derivatives derived in Chapter 4. The characteristic feature of this unifying representation is its formulation as a continuous-time system where the optimization variables are the states. This perception enables the treatment of the transition from one optimization problem to the next as a change in the vector field of the introduced continuous-time system along with an adjustment of its current state due to the choice of the initial conditions. As a consequence, the core of the CMHO and CMHE stability analysis is tantamount to investigating the stability of a nonlinear switched impulsive system. This is done by invoking arguments based on the multiple Lyapunov functions framework. The main contributions of this chapter are:

- Derivation of conditions for the CMHO under which we can proof asymptotic and finite-time convergence of the observation error to zero.
- Derivation of conditions for the CMHE under which we can proof boundedness of the estimation error.

Chapter 7: Simulation and Experimental Results

This chapter presents simulation as well as experimental results for both centralized moving horizon strategies (CMHS). A conventional continuous-discrete extended Kalman filter (CDEKF) serves in all cases as a comparison to the CMHS. The simulation results are derived for a networked version of a common nonlinear benchmark system. This system is characterized by the strong nonlinearities not only in the state equation but also in the sensing model. The experimental results are conducted on a networked pendulum test-rig. The transition between two stationary setpoints and the more challenging swing-up and stabilization problem serve as an open and closed loop benchmark, respectively. The latter represents an especially challenging benchmark due to the unstable and non-minimum phase system dynamics along with the unsynchronized sensor clock and the network-induced non-negligible packet delays and packet drops. The main contribution of this chapter is:

- Presentation of simulation and experimental results for the CMHO and CMHE which demonstrate the performance of both strategies, in general, and especially compared to the CDEKF and justify the assumptions made during the derivation and proofs of both CMHS.

The experimental results of this chapter are based on Philipp and Altmannshofer (2012).

Chapter 8: Conclusions

In this chapter, we provide some conclusive remarks summarizing Part II of the thesis and hint to possible future directions of research.

Part III: Distributed Moving Horizon Strategies

Chapter 9: Distributed Moving Horizon Strategies

This chapter presents the distributed moving horizon observer (DMHO) and the distributed moving horizon estimator (DMHE) for state estimation of linear systems within a common framework for the distributed NCS architecture. Both distributed moving horizon strategies (DMHS) provide distributed knowledge about the full state of the system. This enables the decentralized implementation of any centralized designed controller within the distributed NCS architecture. Both DMHS are extensions to their respective centralized counterparts. As a consequence, the DMHO is constructed for the case of an undisturbed system, while the DMHE considers disturbances acting on the system and the sensors. The derivation of the DMHS constitutes of the following key steps. First, we discuss the issue of how to allocate the measurements from the various sensors to the DMHO/DMHE. Then, we propose to model the communication topology among the distributed estimators as a directed graph and to extend the optimization problems of the centralized moving horizon strategies by additional consensus constraints which reflect the interconnection structure. To achieve a distributed algorithm, we apply a dual decomposition technique to reveal a separable dual problem which we solve by a suitable subgradient method. The main contributions of this chapter are:

- Development of the DMHO and DMHE for state estimation of linear systems within a common framework for the distributed NCS architecture based on the CMHO and CMHE, respectively.
- Development of a distributed Kalman filter as a special case of the DMHE.
- Derivation of conditions under which we can proof equivalence of the state estimates derived by centralized and decentralized moving horizon strategies.

Parts of the results presented in this chapter have been published in Philipp and Schmid-Zurek (2012); Philipp and Schneider (2013); Philipp and Lohmann (2014).

Chapter 10: Simulation Results

This chapter presents simulation results for both distributed moving horizon strategies. The simulation results are derived for a networked four tank system within the distributed NCS architecture. Thereby, the plant consists of four interconnected tanks. Three of these tanks are equipped with an actuator, a sensor and a distributed observer/estimator. This setup provides the possibility of investigating several strategies for the measurement allocation as well as the communication topology among the observers/estimators. The controller is a centralized designed two-degree-of-freedom controller which is implemented in a decentralized fashion. The main contribution of this chapter is:

- Presentation of simulation results for the DMHO and DMHE which demonstrate the functionality as well as the performance of both strategies.

Chapter 11: Conclusions

In this chapter, we provide some conclusive remarks summarizing Part III of the thesis and hint to possible future directions of research.

Appendix

Supplementary material is provided in several appendices, referenced at appropriate places, with the aim to make this thesis self-contained. Appendix A states several definitions, lemmas and theorems which are of central importance for the proofs in this thesis. Appendix B presents derivatives of a specific function which are required at several places in this thesis. Appendix C summarizes the continuous-extended Kalman filter (CDEKF).

Chapter 2

Preliminaries

In this chapter, we present the mathematical building blocks, on which this thesis is built. The presentation is at times rather brief, but there are references to the relevant literature. The detailed outline of the chapter is as follows. In Section 2.1, we review moving horizon strategies for state estimation of nonlinear systems with discrete-time measurements which leads to nonlinear optimal estimation problems on moving horizons. An important topic in this context is the possibility of reconstructing the states. This issue is discussed in Section 2.2 under the aspect of observability of discrete-time nonlinear systems. An overview of different solution methods for optimal state estimation problems is given in Section 2.3. The most promising and efficient methods require the solution of nonlinear optimization problems. Hence, we introduce in Section 2.4 important notions for nonlinear optimization and review related and relevant standard nonlinear optimization techniques. These techniques require the calculation of certain derivatives which can be calculated according to one of the methods reviewed in Section 2.5. Finally, we present some graph related notation in Section 2.6 which we will use for distributed state estimation.

2.1 Moving Horizon Strategies for State Estimation

In this section, we review moving horizon strategies for state estimation of nonlinear systems with discrete-time measurements. To this end, we start in Section 2.1.1 with introducing the dynamic models which we use to describe the dynamical behavior of systems and sensors. In Section 2.1.2, we derive the batch state estimator (BSE) which is an optimal state estimator. However, the BSE possesses a severe drawback: it is computational infeasible. To overcome this problem, moving horizon strategies for state estimation have been developed. In Section 2.1.3 and Section 2.1.4, we present the moving horizon estimator (MHE) and the moving horizon observer (MHO), respectively.

2.1.1 Dynamic Models

There exist a variety of methods to describe the dynamical behavior of systems. In this thesis, we assume that we can capture our physical insight of the plant Σ with a finite-dimensional

differential or difference equation. This leads to a state space representation of the plant Σ in continuous or discrete time. The actual behavior of the plant Σ is observed by measurements generated by the sensors Σ_{S}. Thus, the objective of the model of these sensors Σ_{S} is to couple our physical insight of the plant with the measurements.

Continuous-Time Models

The continuous-time nonlinear state space representation of the plant Σ is

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \tag{2.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the state with the initial value $\boldsymbol{x}(0) \in \mathbb{X}_0 \subseteq \mathbb{R}^{n_x}$, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control input, $\boldsymbol{w}(t) \in \mathbb{R}^{n_x}$ is the state disturbance and $t \in \mathbb{R}_0$ is the continuous time. The model for the sensors Σ_{S} is

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{v}(t), \tag{2.2}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ is the measurement and $\boldsymbol{v}(t) \in \mathbb{R}^{n_y}$ is the measurement disturbance.

If (2.1) and (2.2) can be written as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{w}(t), \tag{2.3a}$$
$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{v}(t), \tag{2.3b}$$

we have the special case of a continuous-time linear state space representation.

Discrete-time Models

Often measurements are only available at the times $t_i = i\Delta T$, where the integer $i \in \mathbb{N}_0$ represents the discrete time and ΔT denotes the sampling period. This situation requires a representation of the states at the times t_i. If we suppose that the state equation (2.1) is discretized with zero-order hold on the control input $\boldsymbol{u}(\cdot)$ and the state disturbance $\boldsymbol{w}(\cdot)$, then we obtain the following discrete-time nonlinear state space representation

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) + \boldsymbol{w}_i, \tag{2.4}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{n_x}$ is the state with the initial value $\boldsymbol{x}_0 \in \mathbb{X}_0 \subseteq \mathbb{R}^{n_x}$, $\boldsymbol{u}_i \in \mathbb{R}^{n_u}$ is the control input, $\boldsymbol{w}_i \in \mathbb{R}^{n_x}$ is the state disturbance and $i \in \mathbb{N}_0$ is the discrete time. Note that for the sake of avoiding an unnecessary and cumbersome notation, we have denoted the function $\boldsymbol{f}(\cdot)$ identically as in the continuous-time case (2.2) even though they are different. The model for the sensors Σ_{S} is

$$\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{x}_i) + \boldsymbol{v}_i, \tag{2.5}$$

where $\boldsymbol{y}_i \in \mathbb{R}^{n_y}$ is the measurement and $\boldsymbol{v}_i \in \mathbb{R}^{n_y}$ is the measurement disturbance.

In analogy to the continuous-time case, if (2.4) and (2.5) can be written as

$$\boldsymbol{x}_{i+1} = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{w}_i, \tag{2.6a}$$

$$\boldsymbol{y}_i = \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{v}_i, \tag{2.6b}$$

we have the special case of a discrete-time linear state space representation.

2.1.2 Batch State Estimator

Since we consider only the case of discrete-time measurements in this thesis, we review the batch state estimator (BSE) only for the discrete-time case and refer the interested reader to the literature for the continuous-time case (see e. g. Jazwinski, 1970). The objective of the BSE at time k can be stated as follows:

Problem 2.1 (Batch State Estimation)**.** *Let an initial estimate $\bar{\boldsymbol{x}}_0$ of $\boldsymbol{x}_0$, the measurement sequence $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$, the control input sequence $\{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}\}$ and the models (2.4) and (2.5) be given. The problem is to estimate the error in the initial estimate $\hat{\boldsymbol{e}}_0 \triangleq \hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0$ and the unknown state disturbance sequence $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$.*

Once the error $\hat{\boldsymbol{e}}_0$ (or equivalently the estimate $\hat{\boldsymbol{x}}_0$) and $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$ are found, the current state estimate $\hat{\boldsymbol{x}}_k$ is obtained by the state equation (2.4).

Usually there exists an infinite number of choices for the estimates $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$, $\{\hat{\boldsymbol{v}}_0, \ldots, \hat{\boldsymbol{v}}_k\}$ and $\hat{\boldsymbol{x}}_0$ that are consistent with the given measurement sequence $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_k\}$ and the control input sequence $\{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}\}$. It should be noted that $\{\hat{\boldsymbol{v}}_0, \ldots, \hat{\boldsymbol{v}}_k\}$ are not independent of $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$ and $\hat{\boldsymbol{x}}_0$. Therefore, it is necessary to establish a criterion for calculating the best combination of these unknown variables. If we consider $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$, $\{\hat{\boldsymbol{v}}_0, \ldots, \hat{\boldsymbol{v}}_k\}$ and $\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0$ to be errors in the state equation, measurement equation and initial estimate, respectively, then we can use classical least-squares theory to minimize these errors. This means that we are obtaining in a certain sense the best fit of the state trajectory to the measurements subject to the models and control inputs. The resulting estimates at time k can be obtained by the solution of the following weighted least-squares minimization problem:

$$\min_{\substack{\hat{\boldsymbol{x}}_0 \\ \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}}} \quad J_k(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}) \tag{2.7a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i, \qquad i = 0, \ldots, k-1 \tag{2.7b}$$

$$\boldsymbol{y}_i = \boldsymbol{h}(\hat{\boldsymbol{x}}_i) + \hat{\boldsymbol{v}}_i, \qquad i = 0, \ldots, k \tag{2.7c}$$

where

$$J_k(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}) = \|\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0\|^2_{\boldsymbol{P}_0^{-1}} + \sum_{i=0}^{k} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=0}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}}. \tag{2.7d}$$

The positive definite weighting matrices $\boldsymbol{P}_0^{-1}$, $\boldsymbol{R}^{-1}$ and $\boldsymbol{Q}^{-1}$ reflect our confidence in the initial estimate, the measurement model and the state model, respectively. Once the solution to the minimization problem is obtained, the current state estimate $\hat{\boldsymbol{x}}_k$ is calculated by the state equation (2.5) with the initial condition $\hat{\boldsymbol{x}}_0$ and the state disturbance sequence $\{\hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$.

Stochastic Interpretation

The preceding formulation has been derived from a purely deterministic viewpoint. Although this approach is adequate for the scope of this thesis, we nevertheless want to mention that there exists also a probabilistic motivated derivation of the minimization problem (2.7). Both viewpoints have their own advantages and disadvantages. Depending on the situation, sometimes one or the other interpretation is more beneficial for achieving the desired objective.

Recall that there are an infinite combination of the variables $\{\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}, \hat{\boldsymbol{v}}_0, \ldots, \hat{\boldsymbol{v}}_k\}$ that explain a given measurement sequence in combination with a certain control input sequence. Therefore, we have to provide additional information prior to the estimation. This knowledge can be included in a rigorous statistical manner by means of the Bayesian framework. All of the information about the states that is contained in the measurement sequence can be written in terms of the conditional joint density function $p(\hat{\boldsymbol{x}}_0, \ldots, \hat{\boldsymbol{x}}_k | \boldsymbol{y}_0, \ldots, \boldsymbol{y}_k)$. In general, this density is complex and cumbersome to work with. However, Bayes' theorem provides a convenient method of expressing the conditional density in terms of the often less complicated densities of $\hat{\boldsymbol{x}}_0$, $\{\hat{\boldsymbol{w}}_i\}$ and $\{\hat{\boldsymbol{v}}_i\}$. When $\hat{\boldsymbol{x}}_0$, $\{\hat{\boldsymbol{w}}_i\}$ and $\{\hat{\boldsymbol{v}}_i\}$ are uncorrelated Gaussian sequences with mean $\bar{\boldsymbol{x}}_0$, $\boldsymbol{0}$ and $\boldsymbol{0}$ and covariance $\boldsymbol{P}_0$, $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively, then we can write the joint conditional probability density function of the state trajectory as

$$p(\hat{\boldsymbol{x}}_0, \ldots, \hat{\boldsymbol{x}}_k | \boldsymbol{y}_0, \ldots, \boldsymbol{y}_k) = A \exp\left(-\frac{1}{2}\left(\|\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0\|^2_{\boldsymbol{P}_0^{-1}} + \sum_{i=0}^{k} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=0}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}}\right)\right), \tag{2.8}$$

where A is a constant independent of $\{\hat{\boldsymbol{x}}_0, \ldots, \hat{\boldsymbol{x}}_k\}$ (for a proof, see e. g. Cox, 1964 or Sage and Melsa, 1971). Maximizing (2.8) with respect to $\{\hat{\boldsymbol{x}}_0, \ldots, \hat{\boldsymbol{x}}_k\}$ is equivalent to minimizing the cost function (2.7d). Thus, the weighted least-squares estimates derived from solving (2.7) corresponds to the peak of the joint conditional density and are known as the maximum a posteriori (MAP) estimates.

2.1.3 Moving Horizon Estimator

The batch state estimator described above is computationally infeasible because the size of the optimal estimation problem (2.7) grows without bound as the number of measurements in-

creases. Therefore, the actual implementation requires that we bound the optimal estimation problem (2.7). This is achieved by the so-called moving horizon strategy which compresses the data (Robertson et al., 1996).

Consider the cost function $J_k(\cdot)$ defined in (2.7d). Let us rearrange it by breaking the time interval into two pieces $T_1 = [0, k-N-1]$ and $T_2 = [k-N, k]$ as follows:

$$\begin{aligned} J_k(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{k-1}) = \|\hat{\boldsymbol{x}}_0 - \bar{\boldsymbol{x}}_0\|^2_{\boldsymbol{P}_0^{-1}} + \sum_{i=0}^{k-N-1} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=0}^{k-N-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}} \\ + \sum_{i=k-N}^{k} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}}. \end{aligned} \tag{2.9}$$

Because of the Markov property, which arises from the state space description of the system, the last two terms in (2.9) depend only on the state $\hat{\boldsymbol{x}}_{k-N}$, the disturbance sequence $\{\hat{\boldsymbol{w}}_{k-N}, \ldots \hat{\boldsymbol{w}}_{k-1}\}$, the control inputs $\{\boldsymbol{u}_{k-N}, \ldots \boldsymbol{u}_{k-1}\}$ and the measurements $\{\boldsymbol{y}_{k-N}, \ldots, \boldsymbol{y}_k\}$. Thus, the principle of optimality allows us to replace the estimation problem (2.7) with the following equivalent fixed-size estimation problem:

$$\min_{\substack{\hat{\boldsymbol{x}}_{k-N} \\ \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1}}} \quad \Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) + \sum_{i=k-N}^{k} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}} \tag{2.10a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i, \qquad i = k-N, \ldots, k-1 \tag{2.10b}$$

$$\boldsymbol{y}_i = \boldsymbol{h}(\hat{\boldsymbol{x}}_i) + \hat{\boldsymbol{v}}_i, \qquad i = k-N, \ldots, k \tag{2.10c}$$

where $N+1$ is the length of the moving horizon and $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$ is the arrival cost defined as

$$\Gamma_j(\boldsymbol{z}) \triangleq \min_{\substack{\hat{\boldsymbol{x}}_0 \\ \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{j-1}}} \quad J_j(\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{w}}_0, \ldots, \hat{\boldsymbol{w}}_{j-1}) \tag{2.11a}$$

subject to

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i, \qquad i = 0, \ldots, j-1 \tag{2.11b}$$

$$\hat{\boldsymbol{x}}_j = \boldsymbol{z}, \tag{2.11c}$$

$$\boldsymbol{y}_i = \boldsymbol{h}(\hat{\boldsymbol{x}}_i) + \hat{\boldsymbol{v}}_i, \qquad i = 0, \ldots, j. \tag{2.11d}$$

In contrast to the problem (2.7), where all of the available measurements were considered, we explicitly take into account in moving horizon estimation only the last $N+1$ measurements. The remaining ones are accounted for by the arrival cost $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$.

The basic strategy of moving horizon estimation is to solve a growing batch state estimation problem until $k = N$ and afterwards a fixed-size moving horizon estimation problem. This means that at every time step $k > N$, the oldest measurement $\boldsymbol{y}_{k-N-1}$ is discarded and the current measurement $\boldsymbol{y}_k$ is added to the moving horizon. The functionality of the MHE is schematically summarized and illustrated in Figure 2.1.

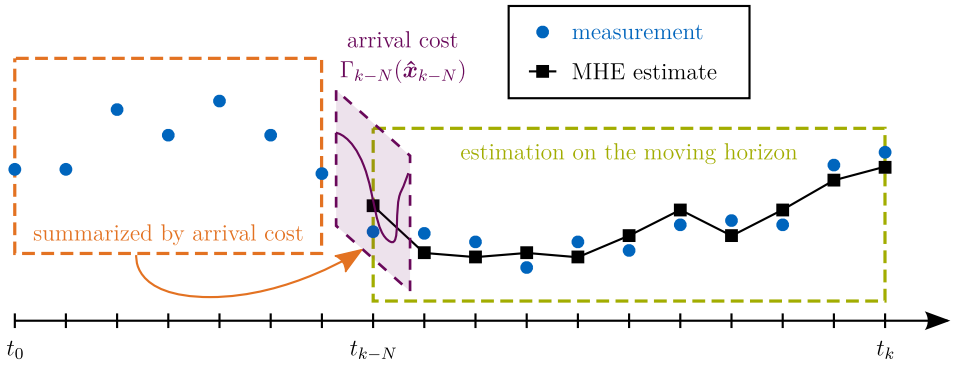


Figure 2.1: Schematic illustration of the moving horizon estimator approach.

The arrival cost $\Gamma_j(\boldsymbol{z})$ plays a fundamental role in moving horizon estimation because it provides means to compress the data and thus allows us to transform an unbounded estimation problem into an equivalent fixed-size one. For linear systems, an algebraic expression for the arrival cost exists. This does not only facilitates the exact compression of the data, but also provides the foundation for establishing a relation between the MHE and the Kalman filter (KF). The precise relation is given in the following Theorem which compactly summarizes the results of Robertson et al. (1996).

Theorem 2.1.1. *Let the initial value $\boldsymbol{x}_0$ and the disturbances $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ be uncorrelated Gaussian sequences with mean $\bar{\boldsymbol{x}}_0$, $\boldsymbol{0}$ and $\boldsymbol{0}$, and covariance $\boldsymbol{P}_0$, $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively, i. e. $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{P}_0)$, $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{v}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$. If the constraints (2.10b) and (2.10c) are the linear system state space representation (2.6a) and (2.6b), respectively, and if the arrival cost $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$ is chosen as*

$$\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) = \frac{1}{2}\|\hat{\boldsymbol{x}}_{k-N} - \bar{\boldsymbol{x}}_{k-N}\|^2_{\boldsymbol{P}^{-1}_{k-N}}, \tag{2.12a}$$

where $\bar{\boldsymbol{x}}_{k-N}$ denotes the optimal MHE estimate at time $k-N$ given the measurements $\{\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{k-N-1}\}$ and where the covariance matrix $\boldsymbol{P}_{k-N}$ is updated by the Riccati equation

$$\boldsymbol{P}_{k+1} = \boldsymbol{Q} + \boldsymbol{A}\boldsymbol{P}_k\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}_k\boldsymbol{C}^T(\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{C}^T + \boldsymbol{R})^{-1}\boldsymbol{C}\boldsymbol{P}_k\boldsymbol{A}^T, \tag{2.12b}$$

then the state estimate obtained by the moving horizon estimator is for all $N+1 \geq 1$ equivalent to the state estimate derived by a Kalman filter.

For nonlinear systems, however, algebraic expressions for the arrival cost do not exist. A common and often the only remedy to this problem is to generate approximate algebraic expressions for the arrival cost and to replace $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$ with its approximation $\hat{\Gamma}_{k-N}(\hat{\boldsymbol{x}}_{k-N})$.

A frequent strategy for calculating such an approximate arrival cost is to use a first-order Taylor expansion of the model around the estimated trajectory $\{\hat{\boldsymbol{x}}_0, \dots, \hat{\boldsymbol{x}}_{k-N}\}$. This is equivalent to applying an extended Kalman Filter recursion for the covariance update. The approximation scheme and its relation to the extended Kalman filter (EKF) are given in the following Theorem which compactly summarizes the results of Robertson et al. (1996).

Theorem 2.1.2. *Let the initial value $\boldsymbol{x}_0$ and the disturbances $\boldsymbol{w}_i$ and $\boldsymbol{v}_i$ be uncorrelated Gaussian sequences with mean $\bar{\boldsymbol{x}}_0$, $\mathbf{0}$ and $\mathbf{0}$, and covariance $\boldsymbol{P}_0$, $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively, i. e. $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{P}_0)$, $\boldsymbol{w}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$ and $\boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R})$. If the arrival cost $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$ is chosen as the approximation*

$$\hat{\Gamma}_{k-N}(\hat{\boldsymbol{x}}_{k-N}) = \frac{1}{2}\|\hat{\boldsymbol{x}}_{k-N} - \bar{\boldsymbol{x}}_{k-N}\|^2_{\boldsymbol{P}^{-1}_{k-N}}, \tag{2.13a}$$

where $\bar{\boldsymbol{x}}_{k-N}$ denotes the optimal MHE estimate at time $k-N$ given the measurements $\{\boldsymbol{y}_0, \dots, \boldsymbol{y}_{k-N-1}\}$ and where the covariance matrix $\boldsymbol{P}_{k-N}$ is updated by the Riccati equation

$$\boldsymbol{P}_{k+1} = \boldsymbol{Q} + \boldsymbol{A}_k\boldsymbol{P}_k\boldsymbol{A}_k^T - \boldsymbol{A}_k\boldsymbol{P}_k\boldsymbol{C}_k^T(\boldsymbol{C}_k\boldsymbol{P}_k\boldsymbol{C}_k^T + \boldsymbol{R})^{-1}\boldsymbol{C}_k\boldsymbol{P}_k\boldsymbol{A}_k^T, \tag{2.13b}$$

where

$$\boldsymbol{A}_k = \left.\frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i)}{\partial \hat{\boldsymbol{x}}_i}\right|_{i=k}, \quad \boldsymbol{C}_k = \left.\frac{\partial \boldsymbol{h}(\hat{\boldsymbol{x}}_i)}{\partial \hat{\boldsymbol{x}}_i}\right|_{i=k}, \tag{2.13c}$$

then the state estimate obtained by the moving horizon estimator is for $N+1=1$ equivalent to the state estimate derived by an iterated extended Kalman filter.

Both theorems show algebraic methods for compressing the data and reveal the relation of the estimates derived by the MHE to those of the Kalman filter. While this relation is independent of the moving horizon length in the linear case, it only holds for $N+1=1$ in the nonlinear case. In general, for moving horizon lengths $N+1>1$, the MHE is more efficient than the EKF (Robertson et al., 1996). This is due to the fact that since the system is nonlinear, the conditional density is non-Gaussian and some information will be lost by the EKF due to the linearization process. In contrast, the MHE utilizes the exact nonlinear model and thus none of the information contained in the last $N+1$ measurements is lost.

Another favorable property of the MHE is the possibility of directly handling constraints. This is useful from an engineering point of view since in practice, additional information about the system is often available in form of constraints. For instance, the range of some values in the system are known. Incorporating this prior knowledge into the estimation process typically improves the performance and convergence of the estimator (Haseltine and Rawlings, 2005). In general, the following inequalities can be considered as additional constraints in problem (2.10):

$$\boldsymbol{c}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{w}}_i) \geq \mathbf{0}, \quad i = k-N, \dots, k-1 \tag{2.14a}$$

$$\boldsymbol{c}_k(\hat{\boldsymbol{x}}_k) \geq \mathbf{0}, \tag{2.14b}$$

where $\boldsymbol{c}_i(\cdot)$ are arbitrary nonlinear functions. This includes as a special case bound inequality constraints of the form

$$\hat{\boldsymbol{x}}_{\min} \leq \hat{\boldsymbol{x}}_i \leq \hat{\boldsymbol{x}}_{\max}, \qquad i = k-N, \ldots, k \tag{2.15a}$$

$$\hat{\boldsymbol{w}}_{\min} \leq \hat{\boldsymbol{w}}_i \leq \hat{\boldsymbol{w}}_{\max}, \qquad i = k-N, \ldots, k-1 \tag{2.15b}$$

$$\hat{\boldsymbol{v}}_{\min} \leq \hat{\boldsymbol{v}}_i \leq \hat{\boldsymbol{v}}_{\max}, \qquad i = k-N, \ldots, k. \tag{2.15c}$$

Regardless of the presence of inequality constraints, we have to be aware of divergence, i. e. instability, when approximating the arrival cost. As long as this approximation satisfies certain technical conditions, non-divergence is guaranteed (Rao, 2000). For example, the choice of the arrival cost for linear system according to (2.12), with or without constraints, yields a stable estimator (Rao et al., 2001). However, when the system is nonlinear, the arrival cost choice according to (2.13) does not guarantee stability and additional measures are needed to ensure stability.

The most important advantages of the MHE compared to the KF/EKF with an eye towards the application in a networked environment are:

- possibility of incorporating more than one measurement for estimating the current state,
- superior handling of nonlinearities, and
- possibility of incorporating constraints.

The price to pay is an increased complexity since an optimal estimation problem has to be solved in every time step.

2.1.4 Moving Horizon Observer

The moving horizon observer (MHO) is another frequently used method for state estimation. Many published methods can be categorized into this framework, like Zimmer (1994); Moraal and Grizzle (1995); Michalska and Mayne (1995); Alamir (1999); Alamir and Calvillo-Corona (2002). The MHO is a simpler form of the MHE and estimates the initial state without including either information prior to the current moving horizon, i. e. $\boldsymbol{P}_i^{-1} = \boldsymbol{0}$, nor estimated state disturbances. Utilizing the above introduced notation as well as the moving horizon formulation, we can write the MHO problem as follows

$$\min_{\hat{\boldsymbol{x}}_{k-N}} \sum_{i=k-N}^{k} \|\hat{\boldsymbol{v}}_i\|^2_{\boldsymbol{R}^{-1}} \tag{2.16a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i), \qquad i = k-N, \ldots, k-1 \tag{2.16b}$$

$$\boldsymbol{y}_i = \boldsymbol{h}(\hat{\boldsymbol{x}}_i) + \hat{\boldsymbol{v}}_i, \qquad i = k-N, \ldots, k. \tag{2.16c}$$

This approach is well motivated if there are no state disturbances and if the measurements are corrupted by zero mean noise (Gelb, 1974). The stochastic interpretation of this approach is that the initial state $\hat{\boldsymbol{x}}_{k-N}$ completely describes the value of all future states and that the prior distribution of the initial states is uniform, i. e. $\boldsymbol{P}_i^{-1} = \boldsymbol{0}$. The advantage of the MHO over the MHE is on the one hand the lower numerical complexity due to the reduced variable space and on the other hand the non-requirement of choosing an (approximated) arrival cost. However, the MHO obtains suitable estimates of the state only if the measurement sequence $\{\boldsymbol{y}_{k-N}, \ldots, \boldsymbol{y}_k\}$ contains enough information. For instance, this is the case if the signal-to-noise-ratio is sufficiently high.

2.2 Observability of Discrete-Time Nonlinear Systems

Nonlinear observability has been studied extensively in the last decades. Different nonlinear observers require different notions of observability which led to different definitions of observability in the literature. For instance, a geometric approach has established significant results for continuous-time systems with continuous measurements (Krener and Respondek, 1985; Nijmeijer and Van Der Schaft, 1990; Isidori, 1995). Although considerable efforts have been made to obtain similar results for discrete-time systems (Lin and Byrnes, 1995; Califano et al., 2003; Rieger et al., 2008), a geometric-like approach has been less successful for discrete-time nonlinear systems. Hence, an alternative approach has been developed for nonlinear observers of receding horizon type. The basic idea of this approach is to ensure that the problem of inverting the nonlinear observation map is well-posed in the sense of Tikhonov and Arsenin (1977) for any state and input. This means that the state estimate solution exists, is unique and depends continuously on the measurement data. The resulting observability definitions guarantee that if the observer selects a sufficiently wide horizon, the state is uniquely reconstructible from the observed output sequence.

In order to state these different observability definitions, we consider the discrete-time nonlinear system

$$\boldsymbol{x}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) \tag{2.17a}$$

$$\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{x}_i), \tag{2.17b}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{n_x}$ is the state with the initial value $\boldsymbol{x}_0 \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$, $\boldsymbol{u}_i \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ is the control input and $\boldsymbol{y}_i \in \mathbb{R}^{n_y}$ is the measurement. The sets $\mathbb{X}$ and $\mathbb{U}$ are assumed to be compact and convex and the functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are assumed to be twice continuously differentiable with respect to their arguments.

Observability is the possibility of reconstructing $\boldsymbol{x}_{k-N}$ from an output sequence $\{\boldsymbol{y}_{k-N}$, $\boldsymbol{y}_{k-N+1}, \boldsymbol{y}_{k-N+2}, \ldots\}$ along with the corresponding input sequence $\{\boldsymbol{u}_{k-N}, \boldsymbol{u}_{k-N+1}, \ldots\}$. Since the time $k - N$ is immaterial for time-invariant systems, we restrict our attention to the case

where $k - N = 0$. Consequently, we define the $N+1$-length observation map as

$$\underline{\boldsymbol{h}}_N(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_N) \triangleq \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_o) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{\phi}_N(\boldsymbol{x}_0, \boldsymbol{u})) \end{bmatrix}, \tag{2.18}$$

where $\boldsymbol{\phi}_i(\boldsymbol{x}_0, \boldsymbol{u})$ denotes the solution of (2.17a) at the time i initialized with $\boldsymbol{x}_0$ at the time $i = 0$ and where $\underline{\boldsymbol{u}}_N = \mathrm{col}(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1})$ is the control input sequence. The observability definitions related to discrete-time nonlinear systems used in the literature aim at the common goal of guaranteeing the well-posedness of the problem of inverting the nonlinear observation map (2.18). Although different authors use slightly different names, conditions and contexts, typical observability definitions can be classified into three categories and are based on the injectivity of the observation map, full-rankness of the Jacobian of the observation map and on a class $\mathcal{K}$-function that determines the relation between the state error and the corresponding observation map error. In the following, we give for each category an exemplary definition.

Definition 2.2.1 (Moraal and Grizzle, 1995; Karafyllis and Kravaris, 2007; Besançon, 2007; Hanba, 2009)**.** *The system* (2.17) *is said to be observable in N+1 steps if* $\forall \underline{\boldsymbol{u}}_N \in \mathbb{U}^N$*, the map* $\underline{\boldsymbol{h}}_N(\boldsymbol{x}_0, \underline{\boldsymbol{u}}_N)$ *is injective as a function of* $\boldsymbol{x}_0$*.*

Definition 2.2.2 (Moraal and Grizzle, 1995; Califano et al., 2003; Besançon, 2007; Hanba, 2009)**.** *The system* (2.17) *is said to satisfy the observability rank condition in N+1 steps if* $\forall \boldsymbol{x}_0 \in \mathbb{X}$, $\forall \underline{\boldsymbol{u}}_N \in \mathbb{U}^N$,

$$\operatorname{rank} \frac{\partial \underline{\boldsymbol{h}}_N}{\partial \boldsymbol{x}_0}\bigg|_{\boldsymbol{x}_0, \underline{\boldsymbol{u}}_N} = n_x. \tag{2.19}$$

Definition 2.2.3 (Alamir and Calvillo-Corona, 2002; Rao et al., 2003; Kreisselmeier and Engel, 2003; Besançon, 2007; Alessandri et al., 2008)**.** *The system* (2.17) *is said to be* $\mathcal{K}$*-observable in N+1 steps if* $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{X}$, $\forall \underline{\boldsymbol{u}}_N \in \mathbb{U}^N$,

$$\varphi(\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|) \leq \|\underline{\boldsymbol{h}}_N(\boldsymbol{x}_1, \underline{\boldsymbol{u}}_N) - \underline{\boldsymbol{h}}_N(\boldsymbol{x}_2, \underline{\boldsymbol{u}}_N)\| \tag{2.20}$$

for some class $\mathcal{K}$*-function* $\varphi(\cdot)$*.*

Since all definitions follow the same common goal, one may wonder how these definitions are related to each other. This question has been tackled in Hanba (2010).

2.3 Overview of Solution Methods for Optimal State Estimation Problems

In this section, we discuss methods which are commonly used to solve the optimal estimation problems arising from the moving horizon techniques for state estimation presented in

Section 2.1. To this end, we consider the following general moving horizon optimal state estimation problem

$$\min_{\substack{\hat{\boldsymbol{x}}_{k-N},\dots,\hat{\boldsymbol{x}}_k \\ \hat{\boldsymbol{w}}_{k-N},\dots,\hat{\boldsymbol{w}}_{k-1}}} \Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) + \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{x}}_i) - \boldsymbol{y}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}} \tag{2.21a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} - \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) - \hat{\boldsymbol{w}}_i = \boldsymbol{0}, \qquad i = k-N,\dots,k-1 \tag{2.21b}$$

$$\boldsymbol{c}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{w}}_i) \geq \boldsymbol{0}, \qquad i = k-N,\dots,k-1 \tag{2.21c}$$

$$\boldsymbol{c}_k(\hat{\boldsymbol{x}}_k) \geq \boldsymbol{0}, \tag{2.21d}$$

where $\hat{\boldsymbol{x}}_i \in \mathbb{R}^{n_x}$ is the estimated state, $\boldsymbol{u}_i \in \mathbb{R}^{n_u}$ is the control input, $\hat{\boldsymbol{w}}_i \in \mathbb{R}^{n_x}$ is the estimated state disturbance and $\boldsymbol{y}_i \in \mathbb{R}^{n_y}$ is the measurement. The functions $\boldsymbol{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x}$ and $\boldsymbol{h} : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ describe the system and sensor dynamics, respectively, while $\boldsymbol{c}_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_{c,i}}$, $i = k-N,\dots,k-1$, and $\boldsymbol{c}_k : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_{c,k}}$ are general nonlinear inequality constraint functions. The objective function incorporates the arrival cost $\Gamma : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ and two sums on the moving horizon which weight on the one hand the difference between the actual measurements and the corresponding model responses and on the other hand the state disturbances. The functions $\boldsymbol{f}(\cdot)$, $\boldsymbol{h}(\cdot)$ and $\boldsymbol{c}_i(\cdot)$, $i = k-N,\dots,k$ are assumed to be twice continuously differentiable with respect to their arguments.

The optimal estimation problem (2.21) results from the MHE problem (2.10) by substituting the estimated measurement disturbances $\hat{\boldsymbol{v}}_i$ in the cost function (2.10a) with the sensor model constraint (2.10c) and adding the inequality constraints (2.14). Note that by setting $\Gamma(\hat{\boldsymbol{x}}_{k-N}) = 0$ and removing the estimated state disturbances, we can recover the MHO problem (2.16).

There exist three basic approaches for solving optimal state estimation problems of the form (2.21) which are depicted in Figure 2.2. We will briefly comment on the Hamilton-Jacobi-Bellman (HJB) equation and dynamic programming in Section 2.3.1 and on the indirect methods in Section 2.3.2. The direct methods will be presented in more detail in Section 2.3.3, because they have proven to be most successful in the context of real life applications. For a more thorough exposition, we refer to Binder et al. (2001). Note that these three basic solution approaches apply equally well to the optimization problems arising from model predictive control (MPC). The reason for this is the fact that MHE is the counterpart to MPC.

2.3.1 Hamilton-Jacobi-Bellman Equation & Dynamic Programming

The optimal solution to the estimation problem (2.21) can be obtained from the solution of a partial differential equation (PDE), the so called Hamilton-Jacobi-Bellman (HJB) equation. While theoretically appealing, this approach has two severe drawbacks in practice. First, a closed-form expression for the solution does not exist and numerical solutions can be calculated

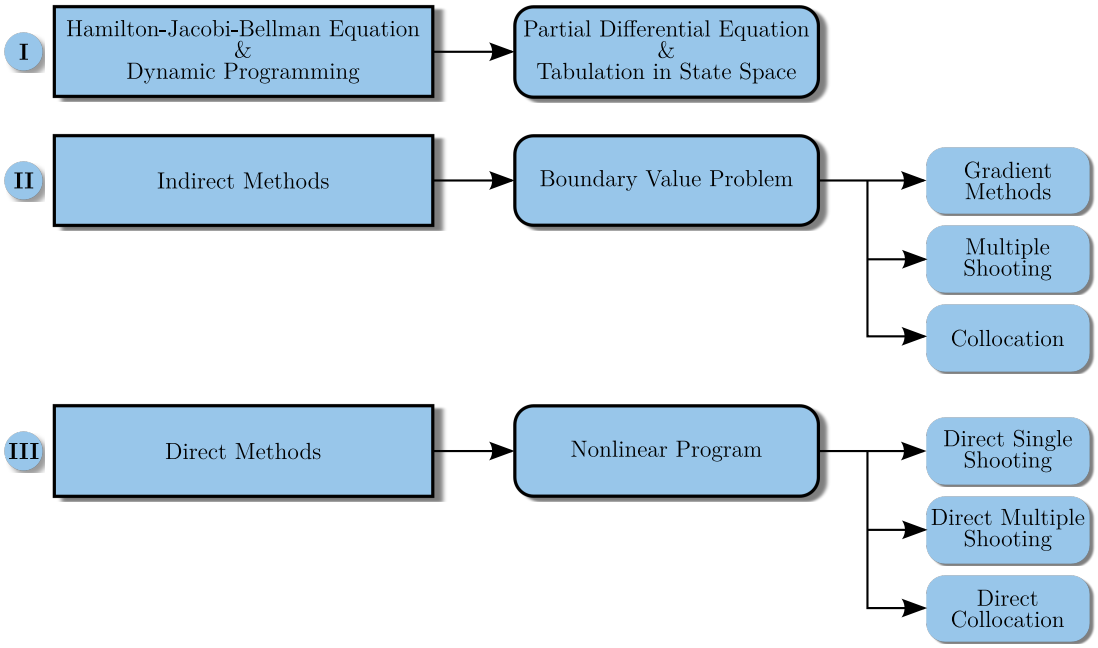


Figure 2.2: Overview of solution methods for optimal state estimation problems.

for very small state dimensions only. Second, inequality constraints on the state usually lead to discontinuous partial derivatives which cannot easily be included. However, it is worth mentioning that for the subclass of linear unconstrained systems with quadratic cost function, the HJB-PDE can be solved analytically or numerically by solving either an algebraic or dynamic matrix Riccati equation.

Dynamic programming is a similar solution methodology which provides the global optimal solution (Bellman, 1957). Unfortunately, its application is severely restricted to systems with small dimension due to the curse of dimensionality.

2.3.2 Indirect Methods

The indirect methods calculate the optimal solution to (2.21) based on the Minimum Principle (Pontryagin et al., 1962), which we will sketch for the case of optimal state estimation problems with only inequality constraints for the state disturbance, i. e. $\hat{\mathbb{W}}_i \triangleq \{\hat{\boldsymbol{w}}_i \in \mathbb{R}^{n_x} | \boldsymbol{c}_i(\hat{\boldsymbol{w}}_i) \geq \boldsymbol{0}\}$, $i = k-N, \ldots, k-1$.

First, we introduce the following abbreviation for the addends of the second and third term in the cost function (2.21a)

$$L_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{w}}_i) \triangleq \|\boldsymbol{h}(\hat{\boldsymbol{x}}_i) - \boldsymbol{y}_i\|^2_{\boldsymbol{R}^{-1}} + \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}}, \quad i = k-N, \ldots, k-1 \tag{2.22a}$$

$$L_k(\hat{\boldsymbol{x}}_i) \triangleq \|\boldsymbol{h}(\hat{\boldsymbol{x}}_k) - \boldsymbol{y}_k\|^2_{\boldsymbol{R}^{-1}}. \tag{2.22b}$$

Next, we define the Hamiltonians as

$$H_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{w}}_i, \boldsymbol{\lambda}_{i+1}) \triangleq L_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{w}}_i) + \boldsymbol{\lambda}^T_{i+1}\Big(\boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i\Big), \quad i = k-N, \ldots, k-1, \tag{2.23}$$

where $\boldsymbol{\lambda}_{i+1} \in \mathbb{R}^{n_x}$ denotes for $i = k-N, \ldots, k-1$, the so-called adjoint variables. Necessary conditions for optimality of the solution trajectories $\{\hat{\boldsymbol{x}}^*_{k-N}, \ldots, \hat{\boldsymbol{x}}^*_k\}$ and $\{\hat{\boldsymbol{w}}^*_{k-N}, \ldots, \hat{\boldsymbol{w}}^*_{k-1}\}$ are given by the following boundary value problem (BVP) in the states and in the adjoint variables:

$$\hat{\boldsymbol{x}}^*_{i+1} = \frac{\partial H_i(\hat{\boldsymbol{x}}^*_i, \hat{\boldsymbol{w}}^*_i, \boldsymbol{\lambda}^*_{i+1})}{\partial \boldsymbol{\lambda}^*_{i+1}}, \qquad i = k-N, \ldots, k-1 \tag{2.24a}$$

$$\boldsymbol{\lambda}^*_i = \frac{\partial H_i(\hat{\boldsymbol{x}}^*_i, \hat{\boldsymbol{w}}^*_i, \boldsymbol{\lambda}^*_{i+1})}{\partial \hat{\boldsymbol{x}}^*_i}, \qquad i = k-N+1, \ldots, k-1 \tag{2.24b}$$

$$\boldsymbol{\lambda}^*_k = \frac{\partial L_k(\hat{\boldsymbol{x}}^*_k)}{\partial \hat{\boldsymbol{x}}^*_k}, \tag{2.24c}$$

$$\mathbf{0} = \frac{\partial(\Gamma_{k-N}(\hat{\boldsymbol{x}}^*_{k-N}) + H_{k-N}(\hat{\boldsymbol{x}}^*_{k-N}, \hat{\boldsymbol{w}}^*_{k-N}, \boldsymbol{\lambda}^*_{k-N+1}))}{\partial \hat{\boldsymbol{x}}^*_{k-N}}. \tag{2.24d}$$

The optimal state disturbances are obtained by minimizing the Hamiltonians

$$\hat{\boldsymbol{w}}^*_i = \arg\min_{\hat{\boldsymbol{w}}_i \in \hat{\mathbb{W}}_i} H_i(\hat{\boldsymbol{x}}^*_i, \hat{\boldsymbol{w}}_i, \boldsymbol{\lambda}^*_{i+1}), \quad i = k-N, \ldots, k-1. \tag{2.25}$$

Note that (2.24d) is the transversality condition resulting from the free initial optimal state $\hat{\boldsymbol{x}}^*_{k-N}$. If $\hat{\boldsymbol{x}}^*_{k-N}$ is fixed, like in the optimal control case, then (2.24d) is replaced by $\hat{\boldsymbol{x}}^*_{k-N} = \hat{\boldsymbol{x}}_{k-N}$. The presented approach has been extended to handle general inequality constraints (2.21c) and (2.21d) on the state disturbances and the states itself. An overview about this topic can be found in Hartl et al. (1995). If both state disturbance and state constraints are present, the optimality conditions form an intricate multi-point boundary value problem (MPBVP) with a priori unknown interior switching points denoting the times when one of the constraints become active or inactive. Activation or deactivation of a state constraint generally leads to a jump in the adjoint variables.

There exist several families of numerical methods for solving the optimality conditions (2.24) and (2.25). Some of them are listed in Figure 2.2 and are briefly commented in the following.

Gradient methods iteratively improve an approximation of the optimal solution by minimizing the Hamiltonians (2.25) subject to the BVP (2.24) (see e. g. Chernousko and Lyubushin, 1982). Multiple shooting is a powerful numerical method for generating accurate solutions to the MPBVPs. An excellent discussion of multiple shooting in this context can be found in Ascher et al. (1995). Collocation methods have also been applied for solving the optimality conditions (Ascher et al., 1979), but they are more successful in the context of direct methods.

The practical drawbacks of indirect methods are:

- Difficulties to derive $\partial H_i / \partial \hat{\boldsymbol{x}}^*_i$ and $\partial H_i / \partial \boldsymbol{\lambda}^*_i$ and to formulate a numerically suitable problem formulation
- Non-intuitive to find proper initial guesses for the adjoint variables
- In order to handle active constraints properly, their switching structure must be guessed

➪ Changes in the problem formulation or low differentiability of the model functions are difficult to include in the solution procedure

2.3.3 Direct Methods

The basic idea of direct methods is to tackle the optimal estimation problem (2.21) directly by means of nonlinear optimization techniques which we will present in Section 2.4. Due to the progress in this area along with the increasing computational power, the direct methods are currently viewed as one of the most powerful approaches for treating real life large scale optimization problems (Binder et al., 2001). Thereby, two basic approaches can be distinguished regarding the treatment of the system equation constraint (2.21b). In the sequential approach, the two steps, system simulation and optimization are performed sequentially in each iteration. In the simultaneous approach, these two steps are performed simultaneously.

Direct Single Shooting (Sequential Approach)

In the direct single shooting method, we treat the system equation constraint (2.21b) as a single initial value problem (IVP) on the interval $i \in [k-N, k]$

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i, \quad i = k-N, \ldots, k-1 \tag{2.26a}$$

$$\hat{\boldsymbol{x}}_{k-N} = \hat{\boldsymbol{x}}_{k-N}. \tag{2.26b}$$

The solution of this problem is a trajectory which is a function of $\hat{\boldsymbol{x}}_{k-N}$ and $\{\hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1}\}$ only. To keep this dependency in mind, we will denote in the sequel this solution by $\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{x}}_{k-N}, \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1})$. Consequently, the constraint (2.21b) can be replaced in the optimization problem (2.21) by substituting the solution $\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{x}}_{k-N}, \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1})$ with $\hat{\boldsymbol{x}}_i$. Hence, the problem (2.21) can be reduced to

$$\min_{\substack{\hat{\boldsymbol{x}}_{k-N} \\ \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1}}} \Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) + \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{x}}_{k-N}, \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1})) - \boldsymbol{y}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}} \tag{2.27a}$$

subject to:

$$\boldsymbol{c}_i(\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{x}}_{k-N}, \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1}), \hat{\boldsymbol{w}}_i) \geq \boldsymbol{0}, \qquad i = k-N, \ldots, k-1 \tag{2.27b}$$

$$\boldsymbol{c}_k(\hat{\boldsymbol{\phi}}_k(\hat{\boldsymbol{x}}_{k-N}, \hat{\boldsymbol{w}}_{k-N}, \ldots, \hat{\boldsymbol{w}}_{k-1})) \geq \boldsymbol{0}. \tag{2.27c}$$

The direct single shooting method is a sequential approach and has the following advantages (✓) and disadvantages (✗):

✓ Small number of optimization variables

✓ State-of-the-art ordinary differential equations (ODE) solvers can be used

✓ Off-the-shelf nonlinear program (NLP) solvers can be used

- ✓ State trajectories are continuous for each iteration step
- ✓ Only initial guesses for the initial state value and the state disturbances are required
- ✗ Applicability to highly unstable systems difficult (i. e. initial value problems with strong dependence on the optimization variables)
- ✗ Rate of convergence depends on accurate derivatives which are generally costly to calculate

Direct Mutiple Shooting (Simultaneous Approach)

In the direct multiple shooting method, we treat the system equation constraint (2.21b) as N independent initial value problems on the N subintervals $[i, i+1]$, $i = k-N, \dots, k-1$. To this end, we introduce $N+1$ additional vectors $\boldsymbol{s}_{k-N}, \dots, \boldsymbol{s}_k$ of the same dimension n_x as the state which we will refer to as the multiple shooting node values. All but the last serve as initial values for N independent IVPs on the intervals $[i, i+1]$, $i = k-N, \dots, k-1$

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_i, \boldsymbol{u}_i) + \hat{\boldsymbol{w}}_i, \tag{2.28a}$$

$$\hat{\boldsymbol{x}}_i = \boldsymbol{s}_i. \tag{2.28b}$$

The solution to these problems are N independent trajectories which are a function of $\boldsymbol{s}_i$ and $\hat{\boldsymbol{w}}_i$ on the corresponding intervals only. To keep this dependency in mind, we will denote this solution by $\hat{\boldsymbol{\phi}}_j(\boldsymbol{s}_i, \hat{\boldsymbol{w}}_i)$, $j \in [i, i+1]$. The N decoupled IVPs are connected by matching conditions which require that each node value should equal the final value of the preceding trajectory, i. e.

$$\boldsymbol{s}_{i+1} = \hat{\boldsymbol{\phi}}_{i+1}(\boldsymbol{s}_i, \hat{\boldsymbol{w}}_i), \quad i = k-N, \dots, k-1. \tag{2.29}$$

These constraints remove the additional degrees of freedom introduced with the parameters $\boldsymbol{s}_i$. It is important to note that the matching conditions do not need to be satisfied during the optimization. In fact, it is a crucial feature of the direct multiple shooting method that it can handle infeasible initial guesses of the variables $\boldsymbol{s}_i$ and $\hat{\boldsymbol{w}}_i$. Consequently, the problem (2.21) can be written as

$$\min_{\substack{\boldsymbol{s}_{k-N}, \dots, \boldsymbol{s}_k \\ \hat{\boldsymbol{w}}_{k-N}, \dots, \hat{\boldsymbol{w}}_{k-1}}} \Gamma_{k-N}(\boldsymbol{s}_{k-N}) + \|\boldsymbol{h}(\boldsymbol{s}_{k-N}) - \boldsymbol{y}_{k-N}\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N+1}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}_i(\boldsymbol{s}_{i-1}, \hat{\boldsymbol{w}}_{i-1})) - \boldsymbol{y}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}} \tag{2.30a}$$

subject to:

$$\boldsymbol{s}_{i+1} - \hat{\boldsymbol{\phi}}_{i+1}(\boldsymbol{s}_i, \hat{\boldsymbol{w}}_i) = \boldsymbol{0}, \qquad i = k-N, \dots, k-1 \tag{2.30b}$$

$$\boldsymbol{c}_i(\boldsymbol{s}_i, \hat{\boldsymbol{w}}_i) \geq \boldsymbol{0}, \qquad i = k-N, \dots, k-1 \tag{2.30c}$$

$$\boldsymbol{c}_k(\boldsymbol{s}_k) \geq \boldsymbol{0}. \tag{2.30d}$$

The direct multiple shooting method is a simultaneous approach and has the following advantages (✓), properties (➪) and disadvantages (✗):

- ✓ State-of-the-art ODE solvers can be used
- ✓ Applicable to highly unstable and chaotic systems
- ➪ Initial guesses for all states and all state disturbances are required
- ➪ Medium number of optimization variables
- ✗ Specially tailored NLP solvers required to exploit the structure of the problem
- ✗ State trajectories are continuous only after successful termination of the solution procedure

Direct Collocation (Simultaneous Approach)

The collocation method starts with revisiting the continuous-time system equation (2.1). The basic idea is to approximate the continuous-time state $\hat{\boldsymbol{x}}(t)$ as well as the continuous-time state disturbance $\hat{\boldsymbol{w}}(t)$ by piecewise defined functions $\hat{\boldsymbol{\phi}}(t,\cdot)$ and $\hat{\boldsymbol{\psi}}(t,\cdot)$ on the time grid

$$t_{k-N} < t_{k-N+1} < \ldots < t_k, \tag{2.31}$$

corresponding to the discrete measurement times. Within each collocation interval $[t_i, t_{i+1}[$, $k-N \leq i \leq k-1$, these functions are chosen as parameter dependent polynomials of order $l \in \mathbb{N}$, i. e.

$$\hat{\boldsymbol{\phi}}(t,\boldsymbol{a})\Big|_{t\in[t_i,t_{i+1}[} \triangleq \hat{\boldsymbol{\phi}}_i(t,\boldsymbol{a}_i) \in \Pi_l^{n_x}, \tag{2.32a}$$

$$\hat{\boldsymbol{\psi}}(t,\boldsymbol{b})\Big|_{t\in[t_i,t_{i+1}[} \triangleq \hat{\boldsymbol{\psi}}_i(t,\boldsymbol{b}_i) \in \Pi_l^{n_x}, \tag{2.32b}$$

where $\Pi_l^{n_x}$ denotes the space of n_x-dimensional vectors of polynomials up to degree l. The coefficients of the polynomials are known as the shape parameters and are collected in the vectors

$$\underline{\boldsymbol{a}} \triangleq \operatorname{col}(\boldsymbol{a}_{k-N},\ldots,\boldsymbol{a}_{k-1}) \in \mathbb{R}^{N(l+1)n_x}, \quad \boldsymbol{a}_i \in \mathbb{R}^{(l+1)n_x}, \quad i = k-N,\ldots,k-1, \tag{2.33a}$$

$$\underline{\boldsymbol{b}} \triangleq \operatorname{col}(\boldsymbol{b}_{k-N},\ldots,\boldsymbol{b}_{k-1}) \in \mathbb{R}^{N(l+1)n_x}, \quad \boldsymbol{b}_i \in \mathbb{R}^{(l+1)n_x}, \quad i = k-N,\ldots,k-1. \tag{2.33b}$$

To guarantee continuity of the approximating functions $\hat{\boldsymbol{\phi}}(t,\cdot)$ and $\hat{\boldsymbol{\psi}}(t,\cdot)$ in the complete interval $[t_{k-N}, t_k]$, we have to impose the following matching conditions at the boundaries of the subintervals

$$\hat{\boldsymbol{\phi}}_i(t_{i+1}^-,\cdot) = \hat{\boldsymbol{\phi}}_{i+1}(t_{i+1}^+,\cdot), \qquad i = k-N,\ldots,k-1, \tag{2.34a}$$

$$\hat{\boldsymbol{\psi}}_i(t_{i+1}^-,\cdot) = \hat{\boldsymbol{\psi}}_{i+1}(t_{i+1}^+,\cdot), \qquad i = k-N,\ldots,k-1, \tag{2.34b}$$

where the superscripts $^-$ and $^+$ denote the left-hand and right-hand limits. Similar, we can ensure differentiability up to order d by imposing

$$\frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\phi}}_i(t_{i+1}^-,\cdot) = \frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\phi}}_{i+1}(t_{i+1}^+,\cdot), \quad i = k-N,\dots,k-1, \quad \kappa = 1,\dots,d \tag{2.34c}$$

$$\frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\psi}}_i(t_{i+1}^-,\cdot) = \frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\psi}}_{i+1}(t_{i+1}^+,\cdot), \quad i = k-N,\dots,k-1, \quad \kappa = 1,\dots,d. \tag{2.34d}$$

By requiring that (2.1) is only to be satisfied at the collocation points $t_{i\nu}$, $\nu = 1,\dots,M$, within each subinterval $[t_i, t_{i+1}[$, $i = k-N,\dots,k-1$, with

$$t_i \le t_{i0} < \dots < t_{iM} \le t_{i+1}, \tag{2.35}$$

we can replace the continuous-time system

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)) + \hat{\boldsymbol{w}}(t), \quad t \in [t_{k-N}, t_k] \tag{2.36}$$

by its discretization

$$\dot{\hat{\boldsymbol{\phi}}}(t_{i\nu}, \underline{\boldsymbol{a}}) = \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t_{i\nu}, \underline{\boldsymbol{a}}), \boldsymbol{u}(t_{i\nu})) + \hat{\boldsymbol{\psi}}(t_{i\nu}, \underline{\boldsymbol{b}}), \quad i = k-N,\dots,k-1,\ \nu = 1,\dots,M \tag{2.37}$$

along with the matching conditions (2.34) which form together a system of nonlinear equations for the shape parameters $\underline{\boldsymbol{a}}$ and $\underline{\boldsymbol{b}}$.

This leads to the following formulation of the MHE optimization problem derived by collocation

$$\min_{\underline{\boldsymbol{a}},\underline{\boldsymbol{b}}} \Gamma_{k-N}(\hat{\boldsymbol{\phi}}(t_{k-N}, \underline{\boldsymbol{a}})) + \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}(t_i, \underline{\boldsymbol{a}})) - \boldsymbol{y}_i\|^2_{\boldsymbol{R}^{-1}} + \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{\psi}}(t_i, \underline{\boldsymbol{b}})\|^2_{\boldsymbol{Q}^{-1}} \tag{2.38a}$$

subject to:

$$\dot{\hat{\boldsymbol{\phi}}}(t_{i\nu}, \underline{\boldsymbol{a}}) - \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t_{i\nu}, \underline{\boldsymbol{a}}), \boldsymbol{u}(t_{i\nu})) - \hat{\boldsymbol{\psi}}(t_{i\nu}, \underline{\boldsymbol{b}}) = \boldsymbol{0}, \quad i = k-N,\dots,k-1,\ \nu = 1,\dots,M \tag{2.38b}$$

$$\frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\phi}}_i(t_{i+1}^-, \boldsymbol{a}_i) - \frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\phi}}_{i+1}(t_{i+1}^+, \boldsymbol{a}_{i+1}) = \boldsymbol{0}, \quad i = k-N,\dots,k-1,\ \kappa = 0,\dots,d \tag{2.38c}$$

$$\frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\psi}}_i(t_{i+1}^-, \boldsymbol{b}_i) - \frac{\partial^\kappa}{\partial t^\kappa}\hat{\boldsymbol{\psi}}_{i+1}(t_{i+1}^+, \boldsymbol{b}_{i+1}) = \boldsymbol{0}, \quad i = k-N,\dots,k-1,\ \kappa = 0,\dots,d \tag{2.38d}$$

$$\boldsymbol{c}_i(\hat{\boldsymbol{\phi}}(t_\gamma, \underline{\boldsymbol{a}}), \hat{\boldsymbol{\psi}}(t_\gamma, \underline{\boldsymbol{b}})) \ge \boldsymbol{0}, \quad i = k-N,\dots,k-1,\ \gamma = 1,\dots,C \tag{2.38e}$$

$$\boldsymbol{c}_k(\hat{\boldsymbol{\phi}}(t_\gamma, \underline{\boldsymbol{a}})) \ge \boldsymbol{0}, \quad \gamma = 1,\dots,C. \tag{2.38f}$$

Note that the inequality constraints $\boldsymbol{c}_i(\cdot)$, $i = k-N,\dots,k$ do not have to be fulfilled on the first time grid but on a second time grid within $[t_{k-N}, t_k]$ with

$$t_{k-N} \le t_1^c \le \dots \le t_C^c \le t_k. \tag{2.39}$$

The collocation method is a simultaneous approach and has the following advantages (✓), properties (➪) and disadvantages (✗):

- ✓ Applicable to highly unstable and chaotic systems
- ✓ Reliable estimation of the adjoint variables is available which can severe as good start estimates for indirect methods (leads to a highly accurate hybrid approach, see Stryk and Bulirsch (1992))
- ➪ Initial guesses for all states and all state disturbances are required
- ✗ High number of optimization variables
- ✗ State-of-the-art ODE solvers cannot be used directly
- ✗ Specially tailored NLP solvers required to exploit the sparse structure of the problem
- ✗ State trajectories are continuous only after successful termination of the solution procedure

2.4 Optimization

In the previous section, we have seen that optimal state estimation problems can be formulated as either unconstrained or constrained optimization problems. A huge number of algorithms have been developed in the last years which can be applied for solving these type of problems. However, not every algorithm is equally well-suited. This is due to the so-called "No Free Lunch Theorem of Optimization" that tells us that a general-purpose universal optimization strategy is impossible, see e. g. the discussion in Ho and Pepyne (2002). Therefore, we will not attempt to provide a survey of all existing algorithms. Instead, we will focus on algorithms for efficiently solving constrained and unconstrained optimization problems arising in this thesis. These algorithms calculate beginning from a starting point a sequence of iterates that terminate when a solution point has been approximately found. Moreover, we will give a brief overview of decomposition methods which enables us to split up an original optimization into several distributively solvable subproblems. These methods provide the foundation for Part III of this thesis.

The remainder of this section is organized as follows. In Section 2.4.1, we introduce important notions for optimization. Section 2.4.2 and 2.4.3 present theory of and algorithms for solving constrained and unconstrained optimization problems, respectively. In Section 2.4.4, we describe relevant techniques that can be used to decompose an optimization problem.

2.4.1 Definitions

This subsection introduces some important definitions appearing in optimization. For a more thorough exposition, see e. g. Nocedal and Wright (2006), Boyd and Vandenberghe (2004) or Bertsekas et al. (2003).

Definition 2.4.1. *A set $\mathcal{X} \subseteq \mathbb{R}^n$ is convex if*

$$(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \in \mathcal{X}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\alpha \in [0,1]$.

Definition 2.4.2. *A function $\boldsymbol{f} : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}^m$ is affine if it is a sum of a linear function and vector, i. e.*

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$$

for all $\boldsymbol{x} \in \mathcal{X}$.

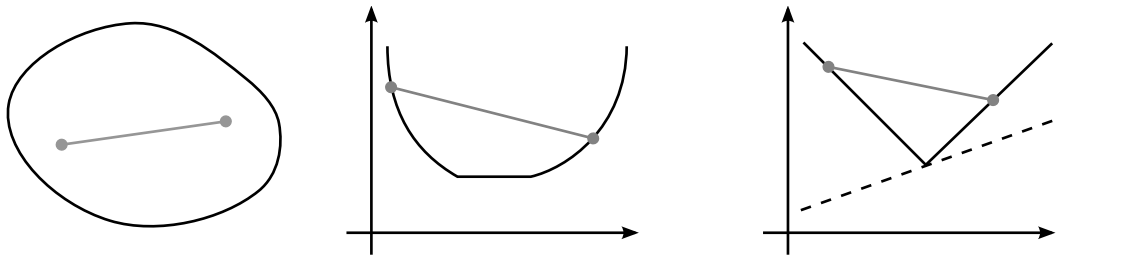
Definition 2.4.3. *A function $f : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ is convex if its domain $\mathcal{X}$ is convex and for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ and $\alpha \in [0,1]$ the following inequality holds*

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}). \tag{2.40}$$

If $-f$ is convex, then f is concave.

Definition 2.4.4. *A function $f : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ is strictly convex if its domain $\mathcal{X}$ is convex and strict inequality holds in (2.40) for $\boldsymbol{x} \neq \boldsymbol{y}$ and $\alpha \in [0,1]$. If $-f$ is strictly convex, then f is strictly concave.*

A convex set, convex function and strictly convex function is illustrated in Figure 2.3(a), 2.3(b) and 2.3(c), respectively. Convexity plays a role much the same as that of linearity in the study of dynamical systems. We will see in the upcoming subsections the effect of convexity.



(a) Example of a convex set. (b) Example of a convex function. (c) Example of a strictly convex function.

Figure 2.3: Various examples of convexity.

Not all functions are differentiable. To handle this case, we make the following definition.

Definition 2.4.5. *A vector $\boldsymbol{s} \in \mathbb{R}^n$ is a subgradient of a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ at a point $\boldsymbol{x} \in \mathbb{R}^n$ if*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{s}^T(\boldsymbol{y} - \boldsymbol{x}) \tag{2.41}$$

holds for any $\boldsymbol{y} \in \mathbb{R}^n$.

An exemplary subgradient is depicted in Figure 2.3(c) by a dashed line. Note that if the convex function f is differentiable, then the subgradient coincides with the gradient.

2.4.2 Unconstrained Optimization

In this subsection, we consider the following unconstrained optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{2.42}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a real vector and $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth or non-smooth function.

Definition 2.4.6. *The problem (2.42) is a convex optimization problem, if f is convex.*

2.4.2.1 What is a solution?

Ideally, we want to find a global minimizer of (2.42), i. e. a point where the function attains its least value. The formal definition is as follows.

Definition 2.4.7. *A point $\boldsymbol{x}^*$ is a global minimizer if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n$.*

However, the global minimizer can be difficult to find since we usually have only local knowledge about f. This fact is taken into account by the following definitions.

Definition 2.4.8. *A point $\boldsymbol{x}^*$ is a local minimizer if there exists a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{N}$.*

Definition 2.4.9. *A point $\boldsymbol{x}^*$ is a strict local minimizer if there exists a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) < f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{N}$ with $\boldsymbol{x} \neq \boldsymbol{x}^*$.*

While the local minimizer is a point which achieves the smallest value of f in its neighborhood, a strict local minimizer is the outright winner in its neighborhood.

The following theorem shows the impact of convexity, see e. g. Nocedal and Wright (2006, Theorem 2.5).

Theorem 2.4.1. *Suppose that f is convex. Then any local minimizer $\boldsymbol{x}^*$ is a global minimizer of f.*

2.4.2.2 How to recognize a solution?

When the function f is twice continuously differentiable, then we are able to tell whether a point $\boldsymbol{x}^*$ is a (strict) local minimizer just by examining the gradient $(\partial f/\partial \boldsymbol{x})(\boldsymbol{x}^*)$ and Hessian $(\partial^2 f/\partial \boldsymbol{x}^2)(\boldsymbol{x}^*)$. A first-order necessary condition for optimality is given by the following Theorem, see e. g. Nocedal and Wright (2006, Theorem 2.2).

Theorem 2.4.2. *If $\boldsymbol{x}^*$ is a local minimizer and f is continuously differentiable in an open neighborhood of $\boldsymbol{x}^*$, then*

$$\frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}^*) = \mathbf{0}. \tag{2.43}$$

The following theorem is a direct consequence of Theorem 2.4.1 and shows the impact of convexity.

Theorem 2.4.3. *If problem (2.42) is a convex optimization problem, then any local minimizer $\boldsymbol{x}^*$ is a global minimizer and the first-order optimality condition (2.43) is a sufficient condition. If in addition f is strictly convex, then any local minimizer is unique.*

If the problem (2.42) is not convex, then the following theorem states a second-order sufficient condition for optimality, see e. g. Nocedal and Wright (2006, Theorem 2.4).

Theorem 2.4.4. *Suppose that for some point $\boldsymbol{x}^* \in \mathbb{R}^n$ the first-order optimality condition (2.43) holds. Suppose also that $\partial^2 f/\partial \boldsymbol{x}^2$ is continuous in an open neighborhood of $\boldsymbol{x}^*$ and that $\partial^2 f/\partial \boldsymbol{x}^2 > \mathbf{0}$. Then $\boldsymbol{x}^*$ is a strict local minimizer of problem (2.42).*

These results provide the foundations for unconstrained optimization algorithms. In one way or another, these algorithms seek a point where $\partial f/\partial \boldsymbol{x}$ vanishes.

2.4.2.3 Line Search Methods

Line search methods (see Algorithm 1) can be applied for finding a minimizer of smooth but not necessarily convex problems (2.42). In each iteration step, these methods choose a direction $\boldsymbol{s}[l]$ and search along this direction from the current iterate $\boldsymbol{x}[l]$ for a new iterate with lower function value. The distance to move along $\boldsymbol{s}[l]$ is found by solving the one-dimensional minimization problem

$$\min_{\gamma[l]} f(\boldsymbol{x}[l] + \gamma[l]\boldsymbol{s}[l]) \tag{2.44}$$

to find a step length $\gamma[l]$. Although the exact minimization would derive the maximum possible reduction of f in the direction $\boldsymbol{s}[l]$, its calculation is expensive and usually unnecessary. Instead, line search methods approximately solve (2.44) by generating trial step lengths until one satisfies some decreasing conditions. These guarantee a sufficient decrease in the objective functions. Some prominent examples are the Armijo conditions, Wolfe conditions and Goldstein conditions, see Nocedal and Wright (2006). The actual iteration step reads as

$$\boldsymbol{x}[l+1] = \boldsymbol{x}[l] + \gamma[l]\boldsymbol{s}[l]. \tag{2.45}$$

Algorithm 1 Line search method

Input: Initial value $\boldsymbol{x}^\circ$.

1: Set $l = 0$ and $\boldsymbol{x}[0] = \boldsymbol{x}^\circ$.
2: **loop**
3: Compute a search direction $\boldsymbol{s}[l]$.
4: Compute a step length $\gamma[l]$ by (approximately) solving (2.44).
5: Update the current point according to (2.45).
6: **end loop**

In the following, we present some prominent line search methods. These methods are able to find the global minimizer for convex problems and at least local minimizers for non-convex problems. Moreover, these algorithms differ among other things in their robustness, convergence speed, accuracy, computational as well as storage requirements.

Steepest Descent Method

The steepest descent method takes steps in the opposite direction of the gradient, i. e.

$$\boldsymbol{s}[l] = -\frac{\partial f}{\partial \boldsymbol{x}}[l]. \tag{2.46}$$

This method impresses with its simplicity and possesses the lowest computational load of all presented methods. However, the price to pay is the only linear convergence rate.

Newton's Method

In contrast to the steepest descent method, Newton's method possesses a quadratic convergence rate close to a minimizer. This is achieved by incorporating additionally curvature information in the search direction, i. e.

$$\boldsymbol{s}[l] = -\frac{\partial^2 f}{\partial \boldsymbol{x}^2}[l]^{-1} \frac{\partial f}{\partial \boldsymbol{x}}[l]. \tag{2.47}$$

However, this method possesses the highest computational load of all presented methods due to the exact Hessian calculation. Note that the Hessian may not always be positive definite which may result in a non-decreasing direction of (2.47). Therefore, modifications of Newton's method exist which replace the exact Hessian by a positive definite modification.

Quasi-Newton Methods

A compromise between the steepest descent and Newton's method are the quasi-Newton methods. The main idea of these methods is to increase the convergence rate of the steepest descent method but to avoid the Hessian calculation required by Newton's method. Their superlinear convergence rate is achieved by choosing the search direction as

$$\boldsymbol{s}[l] = -\boldsymbol{B}[l]^{-1} \frac{\partial f}{\partial \boldsymbol{x}}[l], \tag{2.48}$$

where $\boldsymbol{B}[l]$ is a symmetric positive definite matrix. This matrix approximates the true Hessian and is generated by measuring the changes in the gradients. The quasi-Newton methods differ in the way how this approximation is performed. The most popular algorithm is the BFGS method, named for its discoverers Boyden, Fletcher, Goldfarb and Shanno. It calculates the matrix $\boldsymbol{B}[l]$ as

$$\boldsymbol{B}[l+1] = \boldsymbol{B}[l] - \frac{\boldsymbol{B}[l]\boldsymbol{\Delta}[l]\boldsymbol{\Delta}[l]^T\boldsymbol{B}[l]}{\boldsymbol{\Delta}[l]^T\boldsymbol{B}[l]\boldsymbol{\Delta}[l]} + \frac{\boldsymbol{z}[l]\boldsymbol{z}[l]^T}{\boldsymbol{z}[l]^T\boldsymbol{\Delta}[l]}, \tag{2.49}$$

where $\boldsymbol{\Delta}[l] = \boldsymbol{x}[l+1] - \boldsymbol{x}[l] = \gamma[l]\boldsymbol{s}[l]$ and $\boldsymbol{z}[l] = (\partial f/\partial \boldsymbol{x})[l+1] - (\partial f/\partial \boldsymbol{x})[l]$. Other important methods are the DFP method, SR1 method or modifications of the aforementioned, see Nocedal and Wright (2006).

Gauss-Newton Method

If the objective function (2.42) is a least-squares problem, i. e. $f(\boldsymbol{x}) = \|\boldsymbol{r}(\boldsymbol{x})\|^2$ where $\boldsymbol{r} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is the residual vector, then the exact gradient and Hessian of f can be expressed as

$$\frac{\partial f}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}^T \boldsymbol{r} \tag{2.50a}$$

$$\frac{\partial^2 f}{\partial \boldsymbol{x}^2} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}} + \sum_{i=1}^{m} {}_i r \frac{\partial^2 {}_i r}{\partial \boldsymbol{x}^2}. \tag{2.50b}$$

The Gauss-Newton method utilizes the search direction (2.48) where $\boldsymbol{B}[l]$ is the first term of the exact Hessian, i. e. $\boldsymbol{B}[l] = (\partial \boldsymbol{r}/\partial \boldsymbol{x})^T \partial \boldsymbol{r}/\partial \boldsymbol{x}$. This term can be calculated for free, since it components are already required for the gradient calculation. The convergence rate of the Gauss-Newton method depends on the approximation quality of the exact Hessian. This means that the convergence of the Gauss-Newton method is similar to the one of Newton's method whenever the first term of (2.50b) dominates the second term of (2.50b). This is the case when either the residuals ${}_i r$ are small or when they are nearly affine.

2.4.2.4 Trust Region Methods

Trust region methods (see Algorithm 2) can be applied for finding a minimizer of smooth but not necessarily convex problems (2.42). In each iteration step, these methods construct a model function $m[l]$ whose behavior near the current point $\boldsymbol{x}[l]$ is similar to that of the actual objective function f. The model $m[l]$ is usually a quadratic function of the form

$$m[l](\boldsymbol{s}[l]) = f[l] + \boldsymbol{s}[l]^T \frac{\partial f}{\partial \boldsymbol{x}}[l] + \frac{1}{2}\boldsymbol{s}[l]^T \boldsymbol{B}[l]\boldsymbol{s}[l], \tag{2.51}$$

where $\boldsymbol{B}[l]$ is, similar to the line search scenario, either the Hessian of f or some approximation to it. Trust region methods take into account the fact that this model may not be a good approximation of f when $\boldsymbol{s}[l]$ is far away from $\boldsymbol{x}[l]$. Hence, they restrict the search for a

minimizer of $m[l]$ to a trust region around $\boldsymbol{x}[l]$ which is usually expressed as $\|\boldsymbol{s}[l]\| \leq \Delta[l]$, where $\Delta[l]$ is called the trust region radius. In other words, a candidate step $\boldsymbol{s}[l]$ is calculated by solving the following subproblem

$$\min_{\boldsymbol{s}[l]} m[l](\boldsymbol{s}[l]), \quad \text{subject to } \|\boldsymbol{s}[l]\| \leq \Delta[l]. \tag{2.52}$$

The trustworthiness of the trust region is judged by the ratio

$$\varrho[l] = \frac{f(\boldsymbol{x}[l]) - f(\boldsymbol{x}[l] + \boldsymbol{s}[l])}{m[l](\mathbf{0}) - m[l](\boldsymbol{s}[l])}, \tag{2.53}$$

where the numerator is called the actual reduction and the denominator is the predicted reduction. This ratio is the decision basis for either rejecting the candidate step $\boldsymbol{s}[l]$ and resolving (2.52) with a shrinked trust region or to accept the candidate step $\boldsymbol{s}[l]$, update the iterate by

$$\boldsymbol{x}[l+1] = \boldsymbol{x}[l] + \boldsymbol{s}[l] \tag{2.54}$$

and possibly expand the trust region.

Algorithm 2 Trust region method

Input: Initial value $\boldsymbol{x}^\circ$

1: Set $l = 0$ and $\boldsymbol{x}[0] = \boldsymbol{x}^\circ$
2: **loop**
3: Compute a model function $m[l]$ according to (2.51).
4: Compute a candidate step $\boldsymbol{s}[l]$ according to (2.52).
5: Compute the ratio $\varrho[l]$ according to (2.53).
6: **if** $\varrho[l] > \delta$ **then**
7: Update the current point according to (2.54) and possibly increase the trust region radius $\Delta[l]$.
8: **else**
9: Reduce the trust region radius $\Delta[l]$ and go to line 4.
10: **end if**
11: **end loop**

Note that besides spherical also elliptical and box-shaped trust-regions may be used. Moreover, all Hessian approximation methods presented for line search methods can be employed in the trust-region framework resulting in different methods. In the following, we present only one prominent representative, namely the Levenberg-Marquardt method.

Levenberg-Marquardt Method

This method is suitable for least-squares problems where the objective function (2.42) can be written as $f(\boldsymbol{x}) = \|\boldsymbol{r}(\boldsymbol{x})\|^2$ with the residual vector $\boldsymbol{r} : \mathbb{R}^n \mapsto \mathbb{R}^m$. It uses the same

approximation of the exact Hessian as the Gauss-Newton method, but in the trust-region framework. This avoids one weakness of the Gauss-Newton method, namely, its behavior when the Jacobian $\partial \boldsymbol{r}/\partial \boldsymbol{x}$ is rank-deficient or nearly so. Since the same approximations are used in each case, the local convergence properties of the two methods are similar. In fact, if we are near a minimizer $\boldsymbol{x}^*$, the trust-region becomes inactive and the algorithm takes Gauss-Newton steps.

2.4.2.5 Subgradient Method

This method can handle non-differentiable convex problems (2.42). The basic idea is similar to the steepest descent method. However, since the function f is non-differentiable, we use subgradients (see Definition 2.4.5) instead of gradients to proceed from one iterate to the next, i. e.

$$\boldsymbol{x}[l+1] = \boldsymbol{x}[l] - \gamma[l]\boldsymbol{s}[l], \tag{2.55}$$

where $\gamma[l]$ is the step length and $\boldsymbol{s}[l]$ is a subgradient of f at $\boldsymbol{x}[l]$. In contrast to line search and trust-region methods, the objective value does not necessarily decrease in each iteration of the subgradient method. Rather, the distance between the iterate $\boldsymbol{x}[l]$ and the optimal solution $\boldsymbol{x}^*$ will decrease. The following theorem shows that convergence of the subgradient method can be achieved for diminishing step lengths, see e. g. Bertsekas et al. (2003, Proposition 8.2.5).

Theorem 2.4.5. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function which has a bounded set of minimizers $\mathcal{M}^*$ and let the sequence of step lengths $\{\gamma[l]\}$, $l \in \mathbb{N}_0$, satisfy*

$$\gamma[l] > 0, \quad \lim_{l\to\infty} \gamma[l] = 0, \quad \sum_{i=0}^{\infty} \gamma[l] = \infty. \tag{2.56}$$

If there is a positive constant $c \in \mathbb{R}_0$ such that $\|\boldsymbol{s}[l]\| \leq c$, $\forall l \in \mathbb{N}_0$, then for any starting point $\boldsymbol{x}[0] \in \mathbb{R}^n$, the sequence $\{\boldsymbol{x}[l]\}_{l=0}^{\infty}$ generated according to (2.55) satisfies the relations

$$\lim_{l\to\infty} \min_{\boldsymbol{x}\in\mathcal{M}^*} \|\boldsymbol{x}[l] - \boldsymbol{x}\| = 0, \qquad \lim_{l\to\infty} f(\boldsymbol{x}[l]) = f^*. \tag{2.57}$$

This Theorem does not hold if the subgradients cannot be bounded. In this case, we can slightly modify the subgradient method (2.55) as stated in the following Theorem, see e. g. Shor (1985, Theorem 2.4).

Theorem 2.4.6. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function which has a bounded set of minimizers $\mathcal{M}^*$ and let the sequence of step lengths $\{\gamma[l]\}$, $l \in \mathbb{N}_0$, satisfy*

$$\gamma[l] > 0, \quad \lim_{l\to\infty} \gamma[l] = 0, \quad \sum_{i=0}^{\infty} \gamma[l] = \infty. \tag{2.58}$$

Then for any starting point $\boldsymbol{x}[0] \in \mathbb{R}^n$*, the sequence* $\{\boldsymbol{x}[l]\}_{l=0}^{\infty}$ *generated according to the formula*

$$\boldsymbol{x}[l+1] = \begin{cases} \boldsymbol{x}[l] - \gamma[l]\boldsymbol{s}[l], & \text{if } \gamma[l]\,\|\boldsymbol{s}[l]\| \leq c, \\ \boldsymbol{x}[0], & \text{otherwise}, \end{cases} \tag{2.59}$$

where $c \in \mathbb{R}_0$ *is a positive constant, satisfies the relations*

$$\lim_{l\to\infty} \min_{\boldsymbol{x}\in\mathcal{M}^*} \|\boldsymbol{x}[l] - \boldsymbol{x}\| = 0, \qquad \lim_{l\to\infty} f(\boldsymbol{x}[l]) = f^*. \tag{2.60}$$

The advantage of this modification is that a explicit bound for the subgradients is not required because if $\gamma[l] \to 0$, then the condition $\gamma[l]\,\|\boldsymbol{s}[l]\| \leq c$ holds for all sufficiently large l, regardless of $\boldsymbol{s}[l]$. However, resetting of the subgradient method whenever this condition is violated implicates wasted iteration steps. A typical step size which satisfies (2.56) and (2.58) is $\gamma[l] = a/\sqrt{l}$ where a is a positive constant. Additional details about subgradient methods can be found in Bertsekas et al. (2003); Shor (1985).

2.4.3 Constrained Optimization

In this subsection, we consider the following constrained optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{2.61a}$$

subject to:

$$\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}, \quad i \in \mathcal{E}, \tag{2.61b}$$

$$\boldsymbol{c}_i(\boldsymbol{x}) \geq \boldsymbol{0}, \quad i \in \mathcal{I}, \tag{2.61c}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a real vector, $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth objective function, $\boldsymbol{c}_i : \mathbb{R}^n \mapsto \mathbb{R}^{m_i}$, $i \in \mathcal{E}$ are smooth equality constraints and $\boldsymbol{c}_i : \mathbb{R}^n \mapsto \mathbb{R}^{m_i}$, $i \in \mathcal{I}$ are smooth inequality constraints.

Definition 2.4.10. *The feasible set* Ω *contains all points* $\boldsymbol{x}$ *which satisfies the constraints* (2.61b) *and* (2.61c)*, i. e.*

$$\Omega \triangleq \{\boldsymbol{x} | \boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}, i \in \mathcal{E}, \boldsymbol{c}_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}. \tag{2.62}$$

Definition 2.4.11. *The active set* $\mathcal{A}(\boldsymbol{x})$ *at any feasible* $\boldsymbol{x}$ *consists of the equality constraint indices from* $\mathcal{E}$ *together with the indices of the inequality constraints* i *for which* $\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}$*, i. e.*

$$\mathcal{A}(\boldsymbol{x}) \triangleq \mathcal{E} \cup \{i \in \mathcal{I} | \boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}\}. \tag{2.63}$$

Definition 2.4.12. *The problem (2.61) is a convex optimization problem, if f is convex, the negative equality constraints $-\boldsymbol{c}_i(\boldsymbol{x}), i \in \mathcal{E}$ are convex and the inequality constraints $\boldsymbol{c}_i(\boldsymbol{x}), i \in \mathcal{I}$ are affine.*

2.4.3.1 What is a solution?

The addition of constraints may improve the situation since the feasible set might exclude many local minimizers. However, constraints can also aggravate the situation by generating additional minimizers. The definition of the solution types are extensions of the corresponding definitions for the unconstrained case, except that now we restrict our consideration to the feasible points in the neighborhood of $\boldsymbol{x}^*$. This leads to the following definitions.

Definition 2.4.13. *A point $\boldsymbol{x}^*$ is a global solution of the problem (2.61) if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n$.*

Definition 2.4.14. *A point $\boldsymbol{x}^*$ is a local solution of the problem (2.61) if $\boldsymbol{x}^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{N} \cap \Omega$.*

Definition 2.4.15. *A point $\boldsymbol{x}^*$ is a strict local solution of the problem (2.61) if $\boldsymbol{x}^* \in \Omega$ and there is a neighborhood $\mathcal{N}$ of $\boldsymbol{x}^*$ such that $f(\boldsymbol{x}^*) < f(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{N} \cap \Omega$ with $\boldsymbol{x} \neq \boldsymbol{x}^*$.*

2.4.3.2 How to recognize a solution?

In order to state conditions for optimality, we need the following definitions.

Definition 2.4.16. *The Lagrange function (or just Lagrangian) for the optimization problem (2.61) is*

$$L(\boldsymbol{x}, \underline{\boldsymbol{\lambda}}) \triangleq f(\boldsymbol{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \boldsymbol{\lambda}_i^T \boldsymbol{c}_i(\boldsymbol{x}) \tag{2.64}$$

where $\underline{\boldsymbol{\lambda}} \triangleq \operatorname{col}(\boldsymbol{\lambda}_i, i \in \mathcal{E} \cup \mathcal{I})$ denotes the stacked Lagrange multiplier vector.

Definition 2.4.17. *Given a feasible point $\boldsymbol{x}$ and the active set $\mathcal{A}(\boldsymbol{x})$, the set of linearized feasible directions $\mathcal{F}(\boldsymbol{x})$ is*

$$\mathcal{F}(\boldsymbol{x}) \triangleq \left\{ \boldsymbol{d} \;\middle|\; \begin{array}{ll} \boldsymbol{d}^T \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{x}}(\boldsymbol{x}) = 0, & \forall i \in \mathcal{E} \\ \boldsymbol{d}^T \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{x}}(\boldsymbol{x}) = 0, & \forall i \in \mathcal{A}(\boldsymbol{x}) \cap \mathcal{I} \end{array} \right\}. \tag{2.65}$$

Definition 2.4.18. *Given the point $\boldsymbol{x}$ and the active set $\mathcal{A}(\boldsymbol{x})$, we say that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\partial \boldsymbol{c}_i / \partial \boldsymbol{x}, i \in \mathcal{A}(\boldsymbol{x})\}$ is linearly independent.*

In general, if LICQ holds, none of the active constraint gradients can be zero which guarantees regularity in some sense of the constraints. The first-order necessary conditions for optimality are given in the following Theorem, see e. g. Nocedal and Wright (2006, Theorem 12.1).

Theorem 2.4.7. *Suppose that $\boldsymbol{x}^*$ is a local solution of (2.61), that the functions f and $\boldsymbol{c}_i$ in (2.61) are continuously differentiable and that the LICQ holds at $\boldsymbol{x}^*$. Then there is a Lagrange multiplier vector $\underline{\boldsymbol{\lambda}}^* \triangleq \operatorname{col}(\boldsymbol{\lambda}_i^*, i \in \mathcal{E} \cup \mathcal{I})$, such that the following conditions are satisfied at $(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*)$*

$$\frac{\partial L}{\partial \boldsymbol{x}}(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*) = \mathbf{0} \tag{2.66a}$$

$$\boldsymbol{c}_i(\boldsymbol{x}^*) = \mathbf{0}, \quad \forall i \in \mathcal{E}, \tag{2.66b}$$

$$\boldsymbol{c}_i(\boldsymbol{x}^*) \geq \mathbf{0}, \quad \forall i \in \mathcal{I}, \tag{2.66c}$$

$$\boldsymbol{\lambda}_i^* \geq \mathbf{0}, \quad \forall i \in \mathcal{I}, \tag{2.66d}$$

$$\boldsymbol{\lambda}_i^* \boldsymbol{c}_i(\boldsymbol{x}^*) = \mathbf{0}, \quad \forall i \in \mathcal{E} \cup \mathcal{I}. \tag{2.66e}$$

The conditions (2.66) are known as the Karush-Kuhn-Tucker (KKT) conditions and form the foundation for many optimization algorithms. Similar to the unconstrained case, these algorithms search for a point which satisfies the KKT conditions. The impact of convexity is shown by the following Theorem.

Theorem 2.4.8. *If problem (2.61) is a convex optimization problem, then any local solution $\boldsymbol{x}^*$ is a global solution and the KKT conditions (2.66) are a sufficient condition. If in addition f is strictly convex, then any local solution is unique.*

If the problem (2.61) is not convex, then the following Theorem states second-order sufficient conditions for optimality, see e. g. Nocedal and Wright (2006, Theorem 12.6).

Theorem 2.4.9. *Suppose that for some feasible point $\boldsymbol{x}^* \in \mathbb{R}^n$ there is a Lagrange multiplier vector $\underline{\boldsymbol{\lambda}}^*$ such that the KKT conditions (2.66) are satisfied. Suppose also that*

$$\boldsymbol{w}^T \frac{\partial^2 L}{\partial \boldsymbol{x}^2}(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*) \boldsymbol{w} > 0, \quad \forall \boldsymbol{w} \in \mathcal{C}(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*), \tag{2.67}$$

where $\mathcal{C}(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*)$ is the critical cone defined as

$$\mathcal{C}(\boldsymbol{x}^*, \underline{\boldsymbol{\lambda}}^*) \triangleq \left\{ \boldsymbol{w} \in \mathcal{F}(\boldsymbol{x}) \,\middle|\, \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{x}}^T(\boldsymbol{x}^*)\, \boldsymbol{w} = \mathbf{0} \text{ , all } i \in \mathcal{A}(\boldsymbol{x}^*) \cap \mathcal{I} \text{ with } \boldsymbol{\lambda}_i^* > 0 \right\}. \tag{2.68}$$

Then $\boldsymbol{x}^*$ is a strict local solution of (2.61).

2.4.3.3 Duality

In this subsection, we present some elements of the duality theory. This theory shows how we can construct an alternative problem to the constrained problem (2.61). In this context, the original problem (2.61) is referred to as the primal problem while the alternative problem is known as the dual problem. The primal and dual problem are related in many fascinating ways. The relation we have in mind enables us to calculate the solution to the primal problem

by solving the dual problem. This is favorable if the dual problem possesses properties which the primal problem doesn't have. In order to state this result, we begin with the following definition.

Definition 2.4.19. *The Lagrange dual function (or just dual function)* $q : \mathbb{R}^n \mapsto \mathbb{R}$ *of the problem* (2.61) *is defined as*

$$q(\underline{\boldsymbol{\lambda}}) \triangleq \min_{\boldsymbol{x}} L(\boldsymbol{x}, \underline{\boldsymbol{\lambda}}) = \min_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \boldsymbol{\lambda}_i^T \boldsymbol{c}_i(\boldsymbol{x}) \right). \tag{2.69}$$

When the Lagrangian is unbounded below in $\boldsymbol{x}$, the dual function takes on the value $-\infty$. Since the dual function is the pointwise infimum of a family of affine functions of $\underline{\boldsymbol{\lambda}}$, it is concave, even when the problem (2.61) is not convex. An important property of the dual function is stated in the following Theorem, see e. g. Bertsekas et al. (2003, Proposition 6.2.2).

Lemma 2.4.10. *For any* $\boldsymbol{\lambda}_i \geq 0, i \in \mathcal{I}$ *and any* $\boldsymbol{\lambda}_i, i \in \mathcal{E}$ *we have*

$$q(\underline{\boldsymbol{\lambda}}) \leq f(\boldsymbol{x}^*). \tag{2.70}$$

This means that the dual function yields always lower bounds on the optimal value $f(\boldsymbol{x}^*)$ of the problem (2.61). The question for the best lower bound leads to the Lagrange dual problem.

Definition 2.4.20. *The Lagrange dual problem (or just dual problem) associated to the problem* (2.61) *is defined as*

$$\max_{\underline{\boldsymbol{\lambda}}} q(\underline{\boldsymbol{\lambda}}), \quad \textit{subject to } \boldsymbol{\lambda}_i \geq \boldsymbol{0}, i \in \mathcal{I}. \tag{2.71}$$

The Lagrange dual problem (2.71) is always a convex optimization problem, since the dual function is concave and the constraint is convex. This is the case whether or not the primal problem (2.61) is convex. The quality of the lower bound can be judged by means of the duality gap.

Definition 2.4.21. *The duality gap is defined as*

$$p^* - d^* \geq 0, \tag{2.72}$$

where $p^* \triangleq f(\boldsymbol{x}^*)$ *and* $d^* \triangleq q(\underline{\boldsymbol{\lambda}}^*)$. *If* $d^* = p^*$, *strong duality holds, otherwise weak duality.*

Conditions which assure that strong duality holds are given in the following Theorem, see e. g. Boyd and Vandenberghe (2004, Subsection 5.2.3).

Theorem 2.4.11. *If the problem* (2.61) *is convex and strictly feasible, i. e. there exists* $\boldsymbol{x} \in \mathbb{R}^n$ *such that* $\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}, i \in \mathcal{E}$ *and* $\boldsymbol{c}_i(\boldsymbol{x}) > \boldsymbol{0}, i \in \mathcal{I}$, *then strong duality holds.*

Strong duality can sometimes be used to compute a primal optimal solution from a dual optimal solution. Suppose that strong duality holds and a dual optimal solution $\underline{\boldsymbol{\lambda}}^*$ exists. Then the primal optimal solutions can be obtained by minimizing the Lagrangian $L(\boldsymbol{x}, \underline{\boldsymbol{\lambda}}^*)$ over $\boldsymbol{x}$. If in addition $L(\boldsymbol{x}, \underline{\boldsymbol{\lambda}}^*)$ is a strictly convex function of $\boldsymbol{x}$, then the minimizer of the Lagrangian $L(\boldsymbol{x}, \underline{\boldsymbol{\lambda}}^*)$ is the unique primal optimal solution if it is primal feasible. This fact is interesting when the dual problem is easier to solve than the primal problem, for example, because it can be solved analytically, or has some special structure that can be exploited. This idea is key for the dual decomposition method presented in Section 2.4.4.

2.4.3.4 Linear Programming

A linear program (LP) is a special case of the constrained optimization problem (2.61) and can be written as

$$\min_{\boldsymbol{x}} \boldsymbol{c}^T\boldsymbol{x}, \quad \text{subject to} \begin{cases} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b}, \\ \boldsymbol{x} &\geq \boldsymbol{0}, \end{cases} \tag{2.73}$$

where $\boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The algorithms for solving this type of problem are either simplex methods or interior point (IP) methods, see e. g. Nocedal and Wright (2006, Chapter 13 and 14).

2.4.3.5 Quadratic Programming

A quadratic program (QP) is another special case of the constrained optimization problem (2.61) and consists of a quadratic objective function and affine constraints. It can be written as

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^T\boldsymbol{G}\boldsymbol{x}^T + \boldsymbol{x}^T\boldsymbol{c}, \quad \text{subject to} \begin{cases} \boldsymbol{a}_i^T\boldsymbol{x} &= b_i, \quad i \in \mathcal{E}, \\ \boldsymbol{a}_i^T\boldsymbol{x} &\geq b_i, \quad i \in \mathcal{I}, \end{cases} \tag{2.74}$$

where $\boldsymbol{G} \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $\boldsymbol{c}, \boldsymbol{a}_i \in \mathbb{R}^n$, $i \in \mathcal{E} \cup \mathcal{I}$ are vectors and b_i, $i \in \mathcal{E} \cup \mathcal{I}$ are scalars. If $\boldsymbol{G}$ is positive definite, semi-definite or indefinite, then (2.74) is a strictly convex, convex or nonconvex problem, respectively. Typical solution methods are active set methods and interior point (IP) methods, see e. g. Nocedal and Wright (2006, Chapter 16). The basic idea of these methods is to tackle appropriately the affine KKT conditions (2.66) corresponding to (2.74). For instance, suppose that there are no inequality constraints. Then the KKT conditions reduce to a simple system of linear equations which can be readily solved.

2.4.3.6 Nonlinear Programming

The general constrained optimization problem (2.61) involves nonlinear functions and is referred to as a nonlinear program (NLP). In the following, we outline the key idea of the main two algorithmic concepts for actually solving a NLP, namely the sequential quadratic pro-

gramming (SQP) method and the interior point (IP) method. Both methods have in common that they try to find iteratively a point which satisfies the KKT conditions (2.66).

Sequential Quadratic Programming

The idea of SQP methods consists in linearizing in every iteration step all nonlinear functions appearing in the KKT conditions. It turns out that the resulting equations can be interpreted as the KKT conditions of the following QP

$$\min_{\Delta \boldsymbol{x}[l]} \frac{1}{2} \Delta \boldsymbol{x}[l]^T \frac{\partial^2 L(\boldsymbol{x}[l], \underline{\boldsymbol{\lambda}}[l])}{\partial \boldsymbol{x}^2} \Delta \boldsymbol{x}[l] + \Delta \boldsymbol{x}[l]^T \frac{\partial f(\boldsymbol{x}[l])}{\partial \boldsymbol{x}}, \tag{2.75a}$$

subject to:

$$\boldsymbol{c}_i(\boldsymbol{x}[l]) + \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{x}}^T(\boldsymbol{x}[l])\, \Delta \boldsymbol{x}[l] = \boldsymbol{0}, \quad i \in \mathcal{E}, \tag{2.75b}$$

$$\boldsymbol{c}_i(\boldsymbol{x}[l]) + \frac{\partial \boldsymbol{c}_i}{\partial \boldsymbol{x}}^T(\boldsymbol{x}[l])\, \Delta \boldsymbol{x}[l] \geq \boldsymbol{0}, \quad i \in \mathcal{I}, \tag{2.75c}$$

where $L(\cdot)$ is the Lagrangian to the NLP (2.61). The next iteration step of the SQP is given by $\boldsymbol{x}[l+1] = \boldsymbol{x}[l] + \Delta \boldsymbol{x}[l]$. Consequently, SQP methods compute a solution to the NLP (2.61) by solving a sequence of QPs.

Interior Point Methods

The idea of IP methods is to introduce slack variables $\boldsymbol{s}_i \geq \boldsymbol{0}, i \in \mathcal{I}$ and to replace the inequality (2.66c) of the KKT conditions by the equality

$$\boldsymbol{c}_i(\boldsymbol{x}^*) - \boldsymbol{s}_i = \boldsymbol{1}\mu, \quad i \in \mathcal{I}, \tag{2.76}$$

where $\boldsymbol{1}$ is a vector with unit elements and μ is a positive parameter. The resulting perturbed KKT conditions form a system of nonlinear equations and are solved by Newton's method for a sequence of positive parameters $\{\mu[l]\}$ that converges to zero. It can be shown that in a neighborhood of a solution $(\boldsymbol{x}^*, \boldsymbol{s}_i^*, i \in \mathcal{I}, \underline{\boldsymbol{\lambda}}^*)$ to (2.61) that satisfies the LICQ conditions (Definition 2.4.18) and the second-order sufficient conditions (Theorem 2.4.9), the perturbed KKT conditions have a unique solution for sufficiently small positive values of μ which converges to the optimal solution as $\mu \to 0$. It is important to note that Newton's method for solving the perturbed KKT conditions requires the same gradient and Hessian of the Lagrangian $L(\cdot)$ to the NLP (2.61) as the SQP method in (2.75). Moreover, the perturbed KKT conditions can be interpreted as the KKT conditions of the barrier problem

$$\min_{\boldsymbol{x}[l], \{\boldsymbol{s}_i[l]\}_{i \in \mathcal{I}}} f(\boldsymbol{x}[l]) - \mu[l] \sum_{i \in \mathcal{I}} \sum_{j=1}^{\dim(\boldsymbol{s}_i)} \ln({}_j s_i[l]) \tag{2.77a}$$

subject to:

$$\boldsymbol{c}_i(\boldsymbol{x}[l]) = \mathbf{0}, \quad i \in \mathcal{E} \tag{2.77b}$$

$$\boldsymbol{c}_i(\boldsymbol{x}[l]) - \boldsymbol{s}_i[l] = \mathbf{0}, \quad i \in \mathcal{I}. \tag{2.77c}$$

This interpretation explains why the inequalities $\boldsymbol{s}_i \geq \mathbf{0}, i \in \mathcal{I}$ do not need to be included. The barrier term in the objective function prevents the components of $\boldsymbol{s}_i[l]$ from being close to zero. Moreover, we can see from (2.77) that the combinatorial aspect of NLPs is avoided, i. e. the determination of the active set at the solution.

2.4.4 Decomposition Methods

The basic idea of decomposition methods is to split up an original optimization problem into several distributively solvable subproblems. These subproblems are then coordinated by a high-level master problem to reach an optimal solution to the original problem, see Figure 2.4.

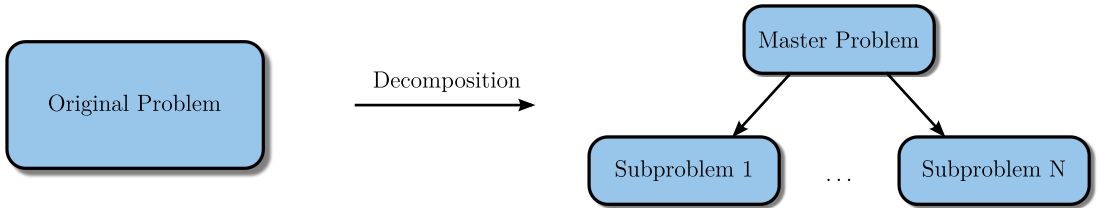


Figure 2.4: Decomposition of an optimization problem into several subproblems coordinated by a master problem.

Decomposition in optimization is an old idea and appears in early work on linear programs from the 1960s (Dantzig and Wolfe, 1960). The original primary motivation was to solve very large problems, which were beyond the reach of standard techniques, by possibly using multiple processors (Bertsekas and Tsitsiklis, 1997). A prominent method, which follows this paradigm, is parallel variable distribution (PVD) (Ferris and Mangasarian, 1994; Sagastizábal and Solodov, 2002). This method distributes the optimization variable among several parallel processors. Thereby, each processor has the primary responsibility for updating its block of variables while the remaining variables are allowed to change in a restricted fashion along some computable search directions. Another class of decomposition methods are bilevel decomposition algorithms (BDA) (DeMiguel and Murray, 2006; Colson et al., 2005). They are applicable if the optimization problem is partially separable. This enables the exploitation of the structure by breaking up the original problem into several subproblems and a master problem. Both methods are reasonable if the objective is to enable the solvability of large optimization problems or to increase the efficiency of the solution procedure. However, in this thesis, our focus is different. Our intention is to use decomposition techniques for dividing the optimization problem resulting from the moving horizon framework into distributed

subproblems. These subproblems fit the structure of the network and yield a distributed moving horizon algorithm. To this end, we will review two classes of suitable decomposition principles: primal and dual, see Palomar and Chiang (2006) and Bertsekas (1999, Chapter 6). Note that primal and dual has the meaning introduced in Section 2.4.3.3 which indicates that the optimization problem is decoupled by investigating either the original or the alternative problem formulation.

2.4.4.1 Primal Decomposition

Primal decomposition is appropriate when the optimization problem has a coupling variable such that, when fixed to some value, the rest of the optimization problem decouples into several subproblems. For instance, consider the following problem

$$\min_{\boldsymbol{y},\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \quad \text{subject to} \begin{cases} \boldsymbol{c}_i(\boldsymbol{x}_i) &= \boldsymbol{0}, \quad i=1,\ldots,N, \\ \boldsymbol{c}_{N+i}(\boldsymbol{x}_i) &\geq \boldsymbol{y}, \quad i=1,\ldots,N. \end{cases} \tag{2.78}$$

Clearly, if we fix the variable $\boldsymbol{y}$, then this problem would decouple. Thus, we can split up the problem (2.78) into two levels of optimization. At the lower level, we have for a fixed $\boldsymbol{y}$ the N decoupled subproblems (one for each i)

$$f_i^*(\boldsymbol{y}) \triangleq \min_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i), \quad \text{subject to} \begin{cases} \boldsymbol{c}_i(\boldsymbol{x}_i) &= \boldsymbol{0}, \\ \boldsymbol{c}_{N+i}(\boldsymbol{x}_i) &\geq \boldsymbol{y}. \end{cases} \tag{2.79}$$

At the higher level, we have the master problem responsible for updating the coupling variable $\boldsymbol{y}$ by solving

$$\min_{\boldsymbol{y}} \sum_{i=1}^{N} f_i^*(\boldsymbol{y}), \tag{2.80}$$

where $f_i^*(\boldsymbol{y})$ is the optimal objective function value of problem (2.79) for a given $\boldsymbol{y}$. If the original problem (2.78) is convex, then the subproblems as well as the master problem are convex. If the function $\sum_{i=1}^{N} f_i^*(\boldsymbol{y})$ is differentiable, then the master problem (2.80) can be solved with one of the presented gradient-based methods. In general, however, the function $\sum_{i=1}^{N} f_i^*(\boldsymbol{y})$ may be non-differentiable and the subgradient method becomes a convenient approach.

2.4.4.2 Dual Decomposition

Dual decomposition is appropriate when the optimization problem has a coupling constraint such that the dual problem decouples into several subproblems. For instance, consider the following problem

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \quad \text{subject to} \begin{cases} \boldsymbol{c}_i(\boldsymbol{x}_i) &= \boldsymbol{0}, \quad i=1,\ldots,N, \\ \sum_{i=1}^{N} \boldsymbol{c}_{N+i}(\boldsymbol{x}_i) &\geq \boldsymbol{0}. \end{cases} \tag{2.81}$$

Clearly, if the constraint $\sum_{i=1}^{N} \boldsymbol{c}_{N+i}(\boldsymbol{x}_i) \geq \mathbf{0}$ was absent, then this problem would decouple. Thus, we investigate the dual function

$$q(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{N+1}) \triangleq \inf_{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) - \sum_{i=1}^{N} \boldsymbol{\lambda}_i^T \boldsymbol{c}_i(\boldsymbol{x}_i) - \boldsymbol{\lambda}_{N+1}^T \sum_{i=1}^{N} \boldsymbol{c}_{N+i}(\boldsymbol{x}_i), \tag{2.82}$$

which can be decoupled such that the problem (2.81) can be split up into two levels of optimization. At the lower level, we have the N decoupled subproblems (one for each i)

$$q_i(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{N+1}) \triangleq \inf_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) - \boldsymbol{\lambda}_i^T \boldsymbol{c}_i(\boldsymbol{x}_i) - \boldsymbol{\lambda}_{N+1}^T \boldsymbol{c}_{N+i}(\boldsymbol{x}_i). \tag{2.83}$$

At the higher level, we have the master problem responsible for updating the dual variables $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{N+1}$ by solving

$$\max_{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{N+1}} \sum_{i=1}^{N} q_i(\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{N+1}), \quad \text{subject to } \boldsymbol{\lambda}_{N+1} \geq \mathbf{0}. \tag{2.84}$$

In fact, this approach solves the dual problem instead of the primal one. Hence, it will only give appropriate results if strong duality holds, see Theorem 2.4.11. The subproblems and the master problem can be solved with one of the presented algorithms for solving constrained and unconstrained optimization problems. Note that in general, the dual function q might be non-differentiable.

2.5 Derivative Calculation

Gradient-based optimization algorithms require knowledge of derivatives. Sometimes the derivatives are easy to calculate by hand. However, this is generally not the case for the derivatives needed for solving the optimization problems arising in the moving horizon framework. As an alternative, there are a number of approaches available which calculate the required derivatives in a systematic way. Note that in this thesis, as introduced in the Notation, the derivative symbol is uniformly ∂ for all types of derivatives. For instance, this means that there is no difference between the total and partial derivative in notation. Keeping this convention in mind, we review in the following various suitable derivative calculation methods. To this end, we consider the following smooth optimization problem which reflects the essential characteristics of the moving horizon framework:

$$\min_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{x}}_{k-N}, \ldots, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}) \tag{2.85a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1} - \boldsymbol{f}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{p}}) = \mathbf{0}, \quad i = k-N, \ldots, k-1, \tag{2.85b}$$

where $J : \mathbb{R}^{n_x} \times \ldots \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}$ is the objective function, $\hat{\boldsymbol{x}}_i \in \mathbb{R}^{n_x}$ is the estimated state and $\hat{\boldsymbol{p}} \in \mathbb{R}^{n_p}$ is the parameter vector. This parameter vector can e. g. be the initial estimated state $\hat{\boldsymbol{x}}_{k-N}$, all estimated states $\mathrm{col}(\hat{\boldsymbol{x}}_i, i \in \{k-N, \ldots, k\})$, unknown parameters of the system, all estimated state disturbances $\mathrm{col}(\hat{\boldsymbol{w}}_i, i \in \{k-N, \ldots, k-1\})$ or a combination of these values. The objective is now to calculate the gradient of the objective function, i. e. $\partial J / \partial \hat{\boldsymbol{p}}$.

Finite-Difference Approximation

This technique has its roots in Taylor's theorem. The derivatives are estimated by observing the change in function values in response to infinitesimal perturbations of the unknowns near a given point $\hat{\boldsymbol{p}}$. For instance, the elements of $\partial J / \partial \hat{\boldsymbol{p}}$ can be approximated by the central-difference formula

$$\frac{\partial J}{\partial_i p} \approx \frac{J(\hat{\boldsymbol{p}} + \epsilon \boldsymbol{e}_i) - J(\hat{\boldsymbol{p}} - \epsilon \boldsymbol{e}_i)}{2\epsilon}, \quad i = 1, \ldots, n_p, \tag{2.86}$$

where ϵ is a small positive scalar and $\boldsymbol{e}_i$ is the i-th unit vector. The evaluation of all elements in $\partial J / \partial \hat{\boldsymbol{p}}$ is as costly as solving $2 n_x n_p$ difference equations from $i = k - N$ to k. This computational load is highly inefficient compared to other existing methods. Moreover, the resulting derivatives are often significantly inaccurate due to truncation and round-off errors. The primary advantage of the finite-difference approaches is that they require almost no extra effort to implement. Further details about the finite-difference approximation can be found in Nocedal and Wright (2006).

Sensitivity Method

The basic principle of the sensitivity method is the chain rule. Keeping the dependence of $\hat{\boldsymbol{x}}_i$ on $\hat{\boldsymbol{p}}$ due to (2.85b) in mind, the application of the chain rule to the cost function J leads to

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}} = \frac{\partial J}{\partial \hat{\boldsymbol{p}}} + \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial J}{\partial \hat{\boldsymbol{x}}_i}, \tag{2.87}$$

where $\partial \hat{\boldsymbol{x}}_i / \partial \hat{\boldsymbol{p}}$ are the first-order state sensitivities. These sensitivities can be interpreted as a measure of how sensitive the state $\hat{\boldsymbol{x}}_i$ reacts upon changes in the parameter $\hat{\boldsymbol{p}}$ and can be calculated by solving the sensitivity matrix difference equation. Application of the chain rule to the state difference equation (2.85b) yields the sensitivity matrix difference equation

$$\frac{\partial \hat{\boldsymbol{x}}_{i+1}}{\partial \hat{\boldsymbol{p}}} = \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{x}}_i} \frac{\partial \hat{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{p}}} + \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{p}}}, \quad i = k - N, \ldots, k - 1, \tag{2.88}$$

with the initial value $\partial \hat{\boldsymbol{x}}_{k-N} / \partial \hat{\boldsymbol{p}}$. The resulting algorithm for computing $\partial J / \partial \hat{\boldsymbol{p}}$ follows:

Algorithm 3 Sensitivity method for the calculation of the gradient $\partial J/\partial \hat{\boldsymbol{p}}$

1: Solve $\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{p}})$ from $i = k-N$ to $k-1$ with the initial condition $\hat{\boldsymbol{x}}_{k-N}$.

2: Solve $\dfrac{\partial \hat{\boldsymbol{x}}_{i+1}}{\partial \hat{\boldsymbol{p}}} = \dfrac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{x}}_i}\dfrac{\partial \hat{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{p}}} + \dfrac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{p}}}$ from $i = k-N$ to $k-1$ with the initial condition $\dfrac{\partial \hat{\boldsymbol{x}}_{k-N}}{\partial \hat{\boldsymbol{p}}}$.

3: Set $\dfrac{\partial J}{\partial \hat{\boldsymbol{p}}} = \dfrac{\partial J}{\partial \hat{\boldsymbol{p}}} + \displaystyle\sum_{i=k-N}^{k} \dfrac{\partial \hat{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{p}}}^T \dfrac{\partial J}{\partial \hat{\boldsymbol{x}}_i}$.

The evaluation of this algorithm involves solving $n_x(n_p + 1)$ difference equations from $i = k - N$ to k. Further details about the sensitivity method can be found in Rosenwasser and Yusupov (1999); Sandu et al. (2003).

Adjoint Method

In contrast to the sensitivity method, the objective of the adjoint method is to avoid the computation of the sensitivities $\partial \hat{\boldsymbol{x}}_i/\partial \hat{\boldsymbol{p}}$ for all $i > k-N$. The first step in achieving this goal is to introduce the Lagrangian corresponding to the optimization problem

$$L(\cdot) = J(\hat{\boldsymbol{x}}_{k-N}, \ldots, \hat{\boldsymbol{x}}_k, \hat{\boldsymbol{p}}) - \sum_{i=k-N}^{k-1} \boldsymbol{\lambda}_i^T \left(\hat{\boldsymbol{x}}_{i+1} - \boldsymbol{f}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{p}})\right), \tag{2.89}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}^{n_x}$ are the Lagrangian multipliers. Because the constraint (2.85b) is always satisfied by construction, we are free to set the values of $\boldsymbol{\lambda}_i$ and $\partial L/\partial \hat{\boldsymbol{p}} = \partial J/\partial \hat{\boldsymbol{p}}$. Next, we omit for ease of presentation the argument of all functions and change the subscript of $\hat{\boldsymbol{x}}_{i+1}$ from $i+1$ to i by writing

$$L = J - \sum_{i=k-N+1}^{k} \boldsymbol{\lambda}_{i-1}^T \hat{\boldsymbol{x}}_i + \sum_{i=k-N}^{k-1} \boldsymbol{\lambda}_i^T \boldsymbol{f}_i. \tag{2.90}$$

Then, we take the derivative with respect to $\hat{\boldsymbol{p}}$ and collect terms in $\partial \hat{\boldsymbol{x}}_i/\partial \hat{\boldsymbol{p}}$ to yield

$$\begin{aligned} \frac{\partial J}{\partial \hat{\boldsymbol{p}}} = \frac{\partial J}{\partial \hat{\boldsymbol{p}}} &+ \sum_{i=k-N}^{k-1} \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{p}}}^T \boldsymbol{\lambda}_i + \frac{\partial \hat{\boldsymbol{x}}_{k-N}}{\partial \hat{\boldsymbol{p}}}^T \left(\frac{\partial J}{\partial \hat{\boldsymbol{x}}_{k-N}} + \frac{\partial \boldsymbol{f}_{k-N}}{\partial \hat{\boldsymbol{x}}_{k-N}}^T \boldsymbol{\lambda}_{k-N}\right) \\ &+ \sum_{i=k-N+1}^{k-1} \frac{\partial \hat{\boldsymbol{x}}_i}{\partial \hat{\boldsymbol{p}}}^T \left(\frac{\partial J}{\partial \hat{\boldsymbol{x}}_i} - \boldsymbol{\lambda}_{i-1} + \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{x}}_i}^T \boldsymbol{\lambda}_i\right) + \frac{\partial \hat{\boldsymbol{x}}_k}{\partial \hat{\boldsymbol{p}}}^T \left(\frac{\partial J}{\partial \hat{\boldsymbol{x}}_k} - \boldsymbol{\lambda}_{k-1}\right). \end{aligned} \tag{2.91}$$

To avoid the calculation of $\partial \hat{\boldsymbol{x}}_k/\partial \hat{\boldsymbol{p}}$, we set

$$\boldsymbol{\lambda}_{k-1} = \frac{\partial J}{\partial \hat{\boldsymbol{x}}_k}. \tag{2.92}$$

Similarly, we set

$$\boldsymbol{\lambda}_{i-1} = \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{x}}_i}^T \boldsymbol{\lambda}_i + \frac{\partial J}{\partial \hat{\boldsymbol{x}}_i} \tag{2.93}$$

to cancel the second-to-last term. The resulting algorithm for computing $\partial J/\partial \hat{\boldsymbol{p}}$ follows:

Algorithm 4 Adjoint method for the calculation of the gradient $\partial J/\partial \hat{\boldsymbol{p}}$

1: Solve $\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{f}_i(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{p}})$ from $i = k-N$ to $k-1$ with the initial condition $\hat{\boldsymbol{x}}_{k-N}$.

2: Solve $\boldsymbol{\lambda}_{i-1} = \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{x}}_i}^T \boldsymbol{\lambda}_i + \frac{\partial J}{\partial \hat{\boldsymbol{x}}_i}$ from $i = k-1$ to $k-N$ with the final condition $\boldsymbol{\lambda}_{k-1} = \frac{\partial J}{\partial \hat{\boldsymbol{x}}_k}$.

3: Set $\frac{\partial J}{\partial \hat{\boldsymbol{p}}} = \frac{\partial J}{\partial \hat{\boldsymbol{p}}} + \sum_{i=k-N}^{k-1} \frac{\partial \boldsymbol{f}_i}{\partial \hat{\boldsymbol{p}}}^T \boldsymbol{\lambda}_i + \frac{\partial \hat{\boldsymbol{x}}_{k-N}}{\partial \hat{\boldsymbol{p}}}^T \left(\frac{\partial J}{\partial \hat{\boldsymbol{x}}_{k-N}} + \frac{\partial \boldsymbol{f}_{k-N}}{\partial \hat{\boldsymbol{x}}_{k-N}} \boldsymbol{\lambda}_{k-N} \right)$.

The evaluation of this algorithm involves solving $2n_x$ difference equations from $i = k-N$ to k. Further details about the adjoint method can be found in Cao et al. (2002); Sandu et al. (2003).

Automatic Differentiation

This technique evaluates numerically the derivative of a function specified by a computer program. Automatic differentiation takes the view that every computer code for evaluating a function, no matter how complicated, can be broken down into a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the chain rule repeatedly to these operations, derivatives of arbitrary order can be computed automatically up to the working precision accuracy. An excellent textbook regarding automatic differentiation is the one of Griewank and Walther (2008). The software tools for automatic differentiation can be categorized by their implementation strategy which uses either source code transformation or operator overloading. Tools using the former strategy (such as ADIFOR (Bischof et al., 1996), OpenAD (Utke et al., 2008) and TAMC (Giering and Kaminski, 1998)) produce new code which calculates both function and derivative values. Tools using the latter strategy (such as ADOL-C (Griewank et al., 1996) and ADF (Straka, 2005)) keep a record of the elementary computations that takes place while the code for evaluating the function is executed. This information is processed to produce the derivative values.

Symbolic Differentiation

The idea in symbolic differentiation is to manipulate the algebraic specification of a function by symbolic manipulation tools (such as Maple, Mathematica and Macsyma) to produce new algebraic expressions for the derivatives.

2.6 Graphs

For the development of the distributed moving horizon strategies presented in Part III of this thesis, we will use some graph notation and a small part of the available graph theory. More details on graphs can be found e. g. in Diestel (2012) and Godsil and Royle (2001).

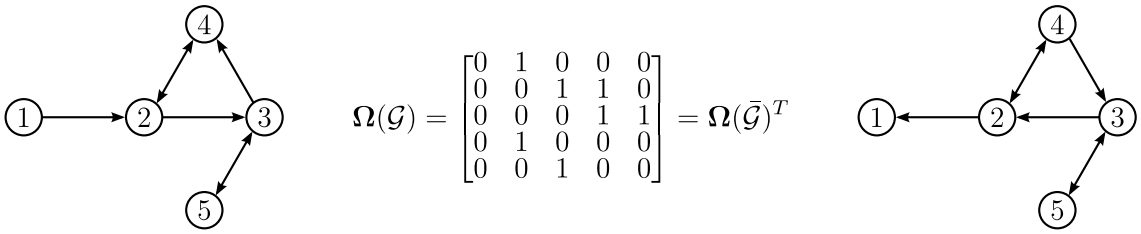
Definition 2.6.1. *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices or nodes $\mathcal{V} = \{v_1, \dots, v_N\}$, $|\mathcal{V}| = N > 0$ and a set of edges or links $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. If $v_i, v_j \in \mathcal{V}$ and $e_{ij} = (v_i, v_j) \in \mathcal{E}$, then there is an edge from node v_i to node v_j.*

Definition 2.6.2. *A self-loop is an edge with two identical vertices, i. e. $e_{ii} = (v_i, v_i)$.*

Definition 2.6.3. *A graph $\mathcal{G}$ is called undirected if $e_{ij} \in \mathcal{E}$ implies $e_{ji} \in \mathcal{E}$. Otherwise the graph is called directed.*

Definition 2.6.4. *The adjacency matrix $\mathbf{\Omega}(\mathcal{G}) = [\omega_{ij}] \in \mathbb{R}^{N \times N}$ of a graph $\mathcal{G}$ is such that $\omega_{ij} = 1$ if $e_{ij} \in \mathcal{E}$ and $\omega_{ij} = 0$ if $e_{ij} \notin \mathcal{E}$.*

In this thesis, we will only use directed graphs without self-loops. Note that the diagonal elements of the adjacency matrix of such graphs is always zero. An example of such a graph with 5 vertices and 7 edges and its adjacency matrix is depicted in Figure 2.5.



directed graph $\mathcal{G}$ reverse directed graph $\bar{\mathcal{G}}$

Figure 2.5: Example of a weakly connected directed graph $\mathcal{G}$ with 5 nodes and 7 vertices, the reverse directed graph $\bar{\mathcal{G}}$ as well as both adjacency matrices $\mathbf{\Omega}(\mathcal{G})$ and $\mathbf{\Omega}(\bar{\mathcal{G}})$.

Moreover, we will need the following two special type of graphs.

Definition 2.6.5. *A complete directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph where each pair of distinct vertices is connected by a pair of unique edges (one in each direction), i. e. if $v_i, v_j \in \mathcal{V}$ and $v_i \neq v_j$, then $e_{ij}, e_{ji} \in \mathcal{E}$.*

The adjacency matrix of a complete directed graph is all 1's except for 0's on the diagonal.

Definition 2.6.6. *The reverse directed graph $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of the same set of vertices $\mathcal{V}$ as the original graph $\mathcal{G}$ but the set of edges $\bar{\mathcal{E}}$ is reversed, i. e. if $e_{ij} = (v_i, v_j) \in \mathcal{E}$, then $e_{ji} = (v_j, v_i) \in \bar{\mathcal{E}}$.*

The adjacency matrix of the reverse directed graph is the transpose of the adjacency matrix of the original graph, i. e. $\mathbf{\Omega}(\bar{\mathcal{G}}) = \mathbf{\Omega}(\mathcal{G})^T$, see the example given in Figure 2.5.

In order to introduce different definitions of connectivity for directed graphs, we first need to define directed and undirected paths.

Definition 2.6.7. *A directed path from node v_{l_1} to node v_{l_r} in a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a sequence $\{v_{l_1}, \dots, v_{l_r}\}$ of $r > 1$ distinct nodes such that $(v_{l_i}, v_{l_{i+1}}) \in \mathcal{E}$, $i = 1, \dots, r-1$.*

Definition 2.6.8. *An undirected path from node v_{l_1} to node v_{l_r} in a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a sequence $\{v_{l_1}, \ldots, v_{l_r}\}$ of $r > 1$ distinct nodes such that $(v_{l_i}, v_{l_{i+1}}) \in \mathcal{E} \cup \bar{\mathcal{E}}$, $i = 1, \ldots, r-1$.*

Consequently, we can define different types of connectivity as follows.

Definition 2.6.9. *A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is weakly connected if there exists an undirected path between any pair of vertices.*

Definition 2.6.10. *A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is strongly connected if there exists a directed path between any pair of vertices.*

Note that if a directed graph is weakly/strongly connected, then the reverse directed graph is also weakly/strongly connected, see the example given in Figure 2.5.

Part II

Centralized Moving Horizon Strategies

Chapter 3

Centralized Moving Horizon Strategies

In this chapter, we develop the centralized moving horizon observer (CMHO) and the centralized moving horizon estimator (CMHE) within a common framework for the undisturbed and the disturbed centralized NCS architecture, respectively.

We start with introducing the undisturbed and the disturbed centralized NCS architectures including the resulting problem definitions in Section 3.1. In Section 3.2, we design a buffer which extends the moving horizon to the networked scenario. The clock model introduced in Section 3.3 enables the extraction of information stored in the buffer by formulating a suitable optimization problem in Section 3.4 for the undisturbed case and in Section 3.5 for the disturbed case. For both optimization problems, we present in Section 3.6 efficient methods for choosing proper initial conditions. The actual state estimates are derived in Section 3.7 based on suboptimal solutions to the optimization problems. Section 3.8 provides the extension to the parameter estimation case. The resulting overall algorithm of the CMHO and CMHE is presented in Section 3.9. Finally, we conclude this chapter with a summary given in Section 3.10.

3.1 Problem Formulation

Consider the centralized NCS architecture depicted in Figure 3.1. The plant Σ is described by the continuous-time nonlinear time-invariant system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}^{(1)}(t), \ldots, \boldsymbol{u}^{(m)}(t)) + \boldsymbol{w}(t), \tag{3.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the state with the initial value $\boldsymbol{x}(0) \in \mathbb{X}_0 \subseteq \mathbb{R}^{n_x}$, $\boldsymbol{u}^{(l)}(t) \in \mathbb{U}^{(l)} \subseteq \mathbb{R}^{n_u,l}$, $l \in \mathcal{A} \triangleq \{1, 2, \ldots, m\}$ is one of the m control inputs, $\boldsymbol{w}(t) \in \mathbb{W} \subseteq \mathbb{R}^{n_x}$ is the state disturbance and $t \in \mathbb{R}_0$ is the global time. By introducing the control input $\boldsymbol{u}(t) \triangleq \operatorname{col}(\boldsymbol{u}^{(l)}(t), l \in \mathcal{A}) \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$, we can write (3.1) compactly as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t). \tag{3.2}$$

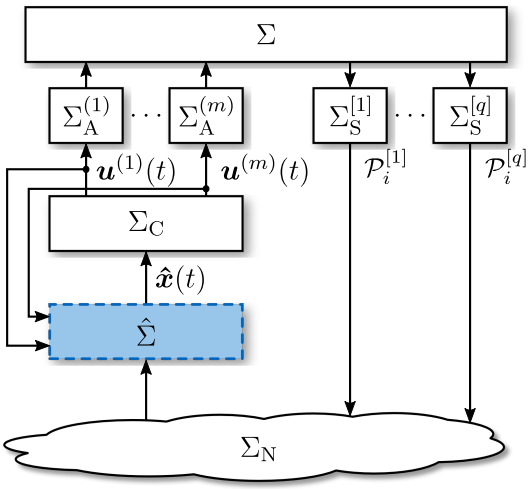


Figure 3.1: Centralized NCS architecture with system Σ, sensors $\Sigma_{\mathrm{S}}^{[i]}$, network Σ_{N}, observer/estimator $\hat{\Sigma}$, controller Σ_{C} and actuators $\Sigma_{\mathrm{A}}^{(i)}$.

Each of the q sensors $\Sigma_{\mathrm{S}}^{[j]}$ with $j \in \mathcal{S} \triangleq \{1, 2, \ldots, q\}$ is equipped with an individual clock which possesses the local sensor time $\bar{t}^{[j]} \in \mathbb{R}_0$. Note that the relation between the global time t and the local sensor times $\bar{t}^{[j]}$ is unknown due to the unsynchronized clocks. Moreover, each sensor generates a sequence of packets $\mathcal{P}_i^{[j]} = \{\boldsymbol{y}^{[j]}(\bar{t}_i^{[j]}), \bar{t}_i^{[j]}\}$ consisting of measurements $\boldsymbol{y}^{[j]}(\bar{t}_i^{[j]}) \in \mathbb{R}^{n_{y,j}}$ and corresponding sensor time stamps $\bar{t}_i^{[j]} \in \mathbb{R}_0$. These measurements are derived by non-uniformly sampling the sensing model

$$\boldsymbol{y}^{[j]}(\bar{t}^{[j]}) = \boldsymbol{h}^{[j]}(\boldsymbol{x}(\bar{t}^{[j]})) + \boldsymbol{v}^{[j]}(\bar{t}^{[j]}), \quad j \in \mathcal{S} \tag{3.3}$$

at the sensor times $\bar{t}_i^{[j]}$ yielding the measurement model

$$\boldsymbol{y}^{[j]}(\bar{t}_i^{[j]}) = \boldsymbol{h}^{[j]}(\boldsymbol{x}(\bar{t}_i^{[j]})) + \boldsymbol{v}^{[j]}(\bar{t}_i^{[j]}), \quad j \in \mathcal{S}, \quad i \in \mathbb{N}, \tag{3.4}$$

where $\boldsymbol{v}^{[j]}(\bar{t}_i^{[j]}) \in \mathbb{V}^{[j]} \subseteq \mathbb{R}^{n_{y,j}}$ are the measurement disturbances. Each packet $\mathcal{P}_i^{[j]}$ is transmitted to the observer/estimator $\hat{\Sigma}$ over the packet-delaying, packet-dropping and unidirectional network Σ_{N}.

The controller Σ_{C} is designed for the nominal case, i. e. without the network Σ_{N}, and is described as

$$\dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{r}(t)), \tag{3.5a}$$

$$\boldsymbol{u}(t) = \begin{bmatrix} \boldsymbol{u}^{(1)}(t) \\ \vdots \\ \boldsymbol{u}^{(m)}(t) \end{bmatrix} = \tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{r}(t)) = \begin{bmatrix} \tilde{\boldsymbol{h}}^{(1)}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{r}(t)) \\ \vdots \\ \tilde{\boldsymbol{h}}^{(m)}(\tilde{\boldsymbol{x}}(t), \boldsymbol{x}(t), \boldsymbol{r}(t)) \end{bmatrix}, \tag{3.5b}$$

where $\tilde{\boldsymbol{x}}(t) \in \mathbb{R}^{n_{\tilde{x}}}$ is the control state with the initial value $\tilde{\boldsymbol{x}}(0) \in \mathbb{R}^{n_{\tilde{x}}}$ and $\boldsymbol{r}(t) \in \mathbb{R}^{n_r}$ is the reference input. The control input $\boldsymbol{u}^{(l)}(t)$ of the l-th actuator results from the corresponding

rows of (3.5b). The successful implementation of the controller Σ_C requires knowledge about the full state of the system.

In order to set up reasonable problem formulations, we make the following assumptions:

Assumption 1. The statistics of $\boldsymbol{x}(0)$, $\boldsymbol{w}(t)$ and $\boldsymbol{v}^{[j]}(t)$, $j \in \mathcal{S}$, are unknown. Consequently, $\boldsymbol{x}(0)$, $\boldsymbol{w}(t)$ and $\boldsymbol{v}^{[j]}(t)$, $j \in \mathcal{S}$ are considered as deterministic variables of unknown character which take their values from the known compact sets $\mathbb{X}_0$, $\mathbb{W}$ and $\mathbb{V}^{[j]}$, $j \in \mathcal{S}$, respectively, with $\mathbf{0} \in \mathbb{W}$ and $\mathbf{0} \in \mathbb{V}^{[j]}$, $j \in \mathcal{S}$.

Assumption 2. The initial value $\boldsymbol{x}(0)$ and the control input $\boldsymbol{u}(t)$, which take their values from the compact sets $\mathbb{X}_0$ and $\mathbb{U}$, respectively, are such that, for any possible disturbance $\boldsymbol{w}(t) \in \mathbb{W}$, the system trajectory $\boldsymbol{x}(t)$ lies in the compact set $\mathbb{X}$.

Assumption 3. The time between two consecutive sensor time stamps is bounded by $\delta_{\bar{t}}$, i. e.

$$\bar{t}_{i+1}^{[j]} - \bar{t}_i^{[j]} \leq \delta_{\bar{t}}, \quad j \in \mathcal{S}, \quad i \in \mathbb{N}. \tag{3.6}$$

Assumption 4. The functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are $\mathcal{C}^2$-functions, i. e. at least twice continuously differentiable, on the closed sets $\mathbb{X} \times \mathbb{U}$ and $\mathbb{X}$, respectively.

Assumption 5. The statistics of the time delays $\tau_i^{[j]}$ of the packets $\mathcal{P}_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}$ unknown. Consequently, the time delays $\tau_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}$ considered as deterministic variables of unknown character which take their values from the bounded set

$$\tau_i^{[j]} \in [0, \tau_{\max}^{[j]}] \subset \mathbb{R}_0, \quad j \in \mathcal{S}, \quad i \in \mathbb{N}. \tag{3.7}$$

Assumption 6. The statistics of the packet drop probability of the packets $\mathcal{P}_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}$ unknown. Consequently, the maximum number of consecutive packet drops for the j-th sensor is considered to be bounded by $N_{\max,\text{drop}}^{[j]} \in \mathbb{N}$, $j \in \mathcal{S}$.

Assumption 7. The network Σ_N is unidirectional, i. e. packet transmission is only possible from the sensors $\Sigma_S^{[j]}$ to the observer/estimator $\hat{\Sigma}$.

Note that the Assumptions 1-7 are quite reasonable from a practical point of view. It is very typical that the states and disturbances of a physical system are bounded in some way and that the characteristics of the disturbances are unknown. For instance, if Assumption 1 is satisfied, then Assumption 2 automatically holds whenever the system (3.2) is input-to-state stable (ISS) with respect to the control input and the state disturbances. Moreover, Assumption 3 only bounds the time between two consecutive sampling times of a sensor but facilitates time-varying and thus event-based sampling. This enables the application of smart sampling strategies. Compared to conventional equidistant time-based sampling, these strategies can be designed such that the overall number of measurements is reduced while the information content gained about the system is maintained or even increased. As a consequence, the limited energy supplies of wireless sensor nodes are saved and the overall network load is

reduced. Note that Assumption 3 together with the Assumptions 5 and 6 guarantee that the time between two consecutive packet arrivals at the observer/estimator is bounded. The Assumptions 5 and 6 reflect the fact that, in general, only bounds for the network statistics are known in advance rather than the precise network statistics. This is due to the fact that the network statistics are essentially the result of the functionality and interaction of the protocols defining the MAC sublayer (see Section 1.1). For instance in wireless networks, the property of the transmission medium and therefore the network statistics can change rapidly. Another consequence of wireless networks is the possibility of utilizing so-called motes, i. e. ultra low power wireless sensor modules. Thereby, a major challenge represents their limited energy supplies. The durability of these motes depend especially on the time the radio is on and on the number of received and transmitted packets. Although Assumption 7 is not required for the subsequent results to hold, it provides the following two advantages. First, it enables the efficient operation of those motes by significantly increasing their durability caused by strictly prohibiting packet receiving. Second, it decreases considerably the overall network load which generally results in either reduced network requirements or in increased network performance.

Problem 3.1 (Centralized Observer Design)**.** *Let the presented undisturbed centralized NCS architecture be given consisting of the undisturbed system* Σ*, i. e.* $\mathbb{W} = \{\mathbf{0}\}$*, undisturbed sensors* $\Sigma_S^{[j]}$*, i. e.* $\mathbb{V}^{[j]} = \{\mathbf{0}\}$*,* $j \in \mathcal{S}$*, controller* Σ_C*, actuators* $\Sigma_A^{(l)}$*,* $l \in \mathcal{A}$*, and network* Σ_N*.*

The problem is to design under the given circumstances an observer $\hat{\Sigma}$ *which reconstructs the current state of the system* Σ *given the sequence of arrived packets* $\{\mathcal{P}_i^{[j]}\}$ *and the control input* $\boldsymbol{u}(t)$*.*

Problem 3.2 (Centralized Estimator Design)**.** *Let the presented disturbed centralized NCS architecture be given consisting of the disturbed system* Σ*, i. e.* $\mathbb{W} \subseteq \mathbb{R}^{n_x}$*, disturbed sensors* $\Sigma_S^{[j]}$*, i. e.* $\mathbb{V}^{[j]} \subseteq \mathbb{R}^{n_{y,j}}$*,* $j \in \mathcal{S}$*, controller* Σ_C*, actuators* $\Sigma_A^{(l)}$*,* $l \in \mathcal{A}$*, and network* Σ_N*.*

The problem is to design under the given circumstances an estimator $\hat{\Sigma}$ *which reconstructs the current state of the system* Σ *given the sequence of arrived packets* $\{\mathcal{P}_i^{[j]}\}$ *and the control input* $\boldsymbol{u}(t)$*.*

Note that for both problems, the difference between the multiple sensor case and the single sensor case is more notational than theoretical. In other words, the presentation of the multiple sensor case involves only an increased notational burden. Otherwise, the extension of the single to the multiple sensor case is straightforward. Therefore, we consider in the following only one sensor for both problems, i. e. $\mathcal{S} = \{1\}$. Consequently, we omit the superscript $[j]$ indicating the sensor affiliation from notation to ease the presentation.

Moreover, it is important to note that throughout Part II of this thesis including the current chapter, the *Assumptions 1-7 are supposed to hold without explicitly stating them.*

3.2 Buffer Design

The transmitted packets $\mathcal{P}_i$ are subject to different time delays τ_i which may cause the packets to arrive out-of-order at the estimator site. This constitutes a problem for conventional observer and estimator concepts which suffer in this case from their design of generating the current estimates based on information stored in only one packet. A common remedy is to simply ignore packets which contain older information in the estimation process. However, this approach is not optimal since not all available information is incorporated. To overcome this problem, the information basis for the CMHO as well as for the CMHE is the buffer Σ_B, which stores $N+1$ augmented packets $\mathcal{P}_{i,j}$, rather than a single packet. These augmented packets $\mathcal{P}_{i,j}$ are derived by extending each successful received packet $\mathcal{P}_i = \{\boldsymbol{y}(\bar{t}_i), \bar{t}_i\}$ by the arrival time stamp t_j to yield the augmented packet $\mathcal{P}_{i,j} = \{\boldsymbol{y}(\bar{t}_i), \bar{t}_i, t_j\}$. The decision logic that generates the ordered sequence $\{\mathcal{B}_k\}$ consisting of ordered $N+1$-tuples with the augmented packets $\mathcal{P}_{i,j}$ as elements can be seen in Algorithm 5. Whenever a new packet arrives (line 3), it is decided (line 6) based on the relation between the sensor time stamp of the new packet, i. e. ${}_2\mathcal{P}_{\cdot,j}$, and the sensor time stamps stored in the buffer, i. e. ${}_2({}_i\mathcal{B}_k)$, whether to incorporate the new packet in the appropriate position in a new buffer (line 8) or to discard the packet. Recall that the left subscript indicates the corresponding element of the associated set.

This buffer design has two main advantages. First, it is optimal in the sense that the latest packets according to the actual sampling order are stored subject to the condition of limited buffer size. Second, the impact of packet reordering is countered successfully, as long as the time stamp of the latest arrived packet is not too old. In this case, the packet is discarded.

Algorithm 5 Buffer Σ_B

Input: buffer size $N+1$

1: Initialization: -global time $t = 0$
-buffer index $k = 0$
-buffer $\mathcal{B}_0 = \{\{\mathbf{0}, 0, 0\}, \ldots, \{\mathbf{0}, 0, 0\}\}$, $|\mathcal{B}_0| = N+1$
2: **for all** $t \in \mathbb{R}_0$ **do**
3: **if** $t = t_j$ **then**
4: Set: termination condition $\leftarrow$ false **and** $i \leftarrow N+1$
5: **while** termination condition = false **and** $i > 0$ **do**
6: **if** ${}_2\mathcal{P}_{\cdot,j} > {}_2({}_i\mathcal{B}_k)$ **then**
7: Increase buffer index: $k \leftarrow k+1$
8: Update buffer by inserting packet $\mathcal{P}_{\cdot,j}$ at the i-th position:
$$\mathcal{B}_k = \{{}_2\mathcal{B}_{k-1}, \ldots, {}_i\mathcal{B}_{k-1}, \mathcal{P}_{\cdot,j}, {}_{i+1}\mathcal{B}_{k-1}, \ldots, {}_{N+1}\mathcal{B}_{k-1}\}$$
9: Set: termination condition $\leftarrow$ true
10: **end if**
11: Decrease position index: $i \leftarrow i-1$
12: **end while**
13: **end if**
14: **end for**

To reduce the occurrence of this effect, the buffer size $N+1$ can be adjusted accordingly.

For consistency with the established notation in the estimation framework, we number the elements, i. e. the packets, and subscript the corresponding quantities, i. e. the sensor and arrival time stamps, in the buffer $\mathcal{B}_k$ for $k \in \mathbb{N}_{N+1}$ from $k-N$ to k. Therefore, we introduce the following notations.

Notation 3.2.1. The sets $\mathcal{I}_k \triangleq \{k-N, k-N+1, \ldots, k\}$, $\underline{\mathcal{I}}_k \triangleq \mathcal{I}_k \backslash \{k-N\}$ and $\overline{\mathcal{I}}_k \triangleq \mathcal{I}_k \backslash \{k\}$ denote sets of indices corresponding to the buffer $\mathcal{B}_k$. The abbreviations $\bar{t}_{i|k}$ and $t_{i|k}$ denote for $i \in \mathcal{I}_k$ and $k \in \mathbb{N}_{N+1}$ the sensor time stamp and the arrival time stamp stored in the $(i-k+N+1)$-th packet in the buffer $\mathcal{B}_k$, respectively, i. e. $\bar{t}_{i|k} \triangleq {}_2({}_{i-k+N+1}\mathcal{B}_k)$ and $t_{i|k} \triangleq {}_3({}_{i-k+N+1}\mathcal{B}_k)$.

Figure 3.2 illustrates the functionality of the presented buffer decision logic. A cross on the upper timescale represents an unknown sensor time stamp in global time t of the corresponding packet $\mathcal{P}_i$. These packets are transmitted over the network Σ_{N} and arrive after different unknown time delays τ_i at the estimator site, provided no packet drop occurs. A cross on the lower timescale depicts the arrival time stamp in global time t of the corresponding augmented packet $\mathcal{P}_{i,j}$. Note that packet overtaking occurs whenever two arrows intersect. The resulting buffer sequence $\{\mathcal{B}_i\}_{i=0}^{3}$ for a buffer size of $N+1=2$ is depicted underneath the augmented packets. To illustrate the introduced notation, the sensor and arrival time stamp

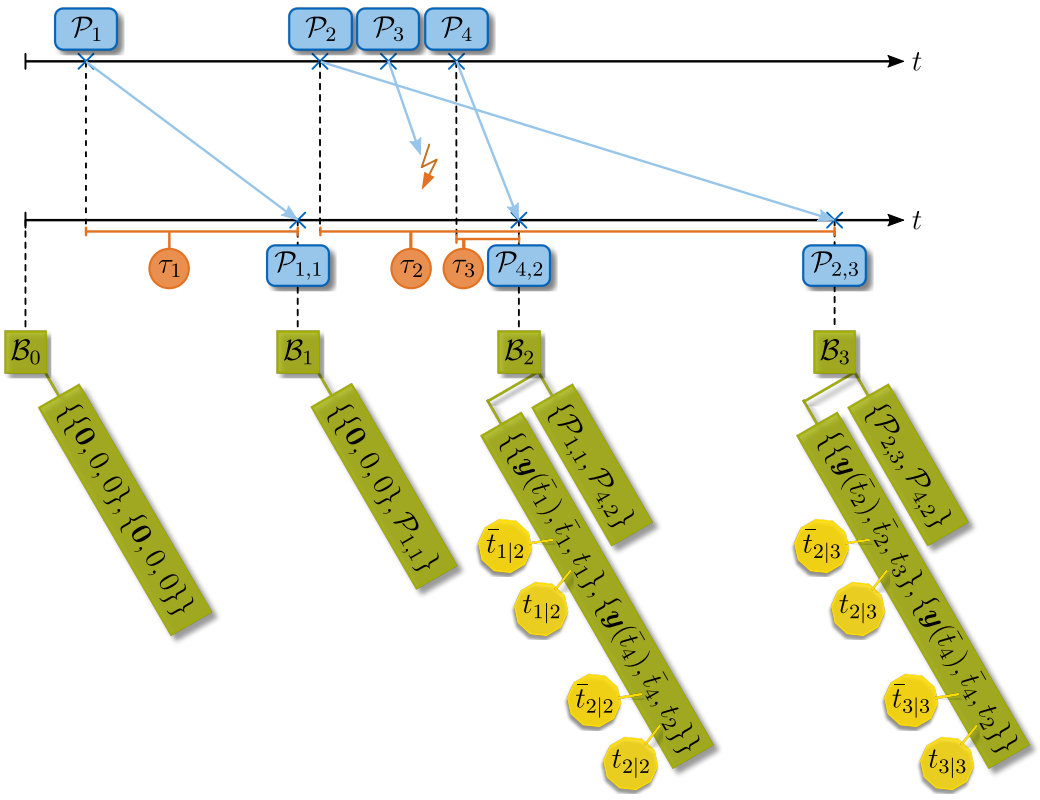


Figure 3.2: Illustration of the buffer decision logic and the introduced notation.

abbreviations are highlighted for $k \geq N+1 = 2$. For instance, the following relations hold

$$\bar{t}_{2|2} = {}_2({}_2\mathcal{B}_2) = \bar{t}_{3|3} = {}_2({}_2\mathcal{B}_3).$$

3.3 Clock Model

The buffer provides among other things the information when a measurement has been taken according to the local sensor time $\bar{t}$. However, without any knowledge about the relation between this local sensor time $\bar{t}$ and the global time t, it is impossible to know when a measurement was taken in global time. The common remedy to this problem is to utilize clock synchronization protocols to establish a common notion of time. However, these protocols have two severe drawbacks. First, most protocols require bidirectional communication which is prohibited by Assumption 7. Second, these protocols consume bandwidth which should not be underestimated and reduce the possibly limited energy supplies of sensor nodes. Thus, we seek for an alternative solution. Although we do not utilize clock synchronization protocols, we can benefit from the a basic idea of these protocols, namely, to model the clock dynamics as a linear system. In general, the clock function $C(t)$ representing the time of a clock is modeled as

$$C(t) = \alpha\, t + \beta, \tag{3.8}$$

where $\alpha \in \mathbb{R}_0$ is the *clock skew* (frequency difference) and $\beta \in \mathbb{R}$ is the *clock offset* (phase difference) (Wu et al., 2011). In the long term, clock parameters are subject to changes due to environmental or other external effects such as temperature, atmospheric pressure, voltage changes, and hardware aging (Vig, 1992). But for sufficiently short periods of time, we can view the clock parameters as constant. This leads directly to the following assumption.

Assumption 8. The relation between the global time t and the local sensor time $\bar{t}$ is given by the clock model

$$t = \alpha_k \bar{t} + \beta_k, \quad \bar{t} \in \bar{\mathcal{T}}_k = \left[\bar{t}_{k-N|k}, \bar{t}_{k|k}\right] \subset \mathbb{R}_0, \quad k \in \mathbb{N}_{N+1}, \tag{3.9}$$

where $\alpha_k \in \mathbb{R}_0$ and $\beta_k \in \mathbb{R}$ represent the clock skew and the clock offset, respectively.

A graphical representation of Assumption 8 is illustrated in Figure 3.3. Although both clock parameters are in general unknown, Assumption 8 enables us not only to recast (3.4) in global time

$$\boldsymbol{y}(\alpha_k\, \bar{t}_{i|k} + \beta_k) = \boldsymbol{h}(\boldsymbol{x}(\alpha_k\, \bar{t}_{i|k} + \beta_k)) + \boldsymbol{v}(\alpha_k\, \bar{t}_{i|k} + \beta_k), \quad i \in \mathbb{N}, \tag{3.10}$$

but also to express the time delay $\tau_{i|k}$ of the i-th packet in the buffer $\mathcal{B}_k$ as a function of the

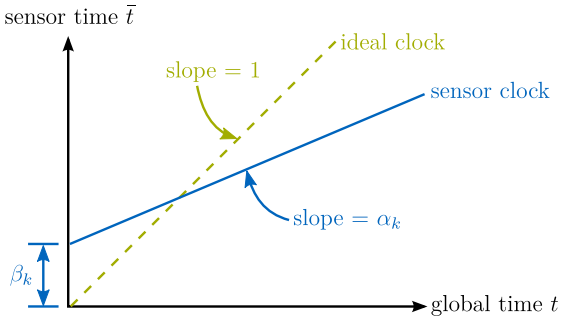


Figure 3.3: Illustration of the relation between the global time t and the local sensor time $\bar{t}$ in the time interval $\bar{t} \in \overline{\mathcal{T}}_k$.

clock model parameters

$$\tau_{i|k} = t_{i|k} - \alpha_k \bar{t}_{i|k} - \beta_k, \quad i \in \mathcal{I}_k, \quad k \in \mathbb{N}_{N+1}. \tag{3.11}$$

This approach will be key for the presented centralized moving horizon strategies as well as for the subsequent analysis. Instead of considering the time delays explicitly like in the theory of time-delay systems (Richard, 2003), we can deal with the time delays implicitly by investigating the two unknown parameters of the clock model.

For consistency with the established notation in the estimation framework, the following notations are introduced.

Notation 3.3.1. The abbreviation $\boldsymbol{x}_{i|k}$ of a time-depending vector $\boldsymbol{x}(t)$ denotes for $i \in \mathcal{I}_k$ and $k \in \mathbb{N}_{N+1}$ its value at the global time $\alpha_k \bar{t}_{i|k} + \beta_k$, i. e. $\boldsymbol{x}_{i|k} \triangleq \boldsymbol{x}(\alpha_k \bar{t}_{i|k} + \beta_k)$. Similarly, the abbreviation $\hat{\boldsymbol{x}}_{i|k}$ of an estimated time-depending vector $\hat{\boldsymbol{x}}(t)$ denotes for $i \in \mathcal{I}_k$ and $k \in \mathbb{N}_{N+1}$ its value at the estimated global time $\hat{\alpha}_k \bar{t}_{i|k} + \hat{\beta}_k$, i. e. $\hat{\boldsymbol{x}}_{i|k} \triangleq \hat{\boldsymbol{x}}(\hat{\alpha}_k \bar{t}_{i|k} + \hat{\beta}_k)$.

Consequently, the measurements $\boldsymbol{y}(\bar{t}_i)$ in the buffer $\mathcal{B}_k$ can be written in global time as

$$\boldsymbol{y}_{i|k} = \boldsymbol{h}(\boldsymbol{x}_{i|k}) + \boldsymbol{v}_{i|k}, \quad i \in \mathcal{I}_k, \quad k \in \mathbb{N}_{N+1}. \tag{3.12}$$

Note that this notation as well as the associated meaning are extensions of the usual ones in the estimation framework to the networked scenario. In fact, both notations are identical for the nominal case, i. e. without the network Σ_N and with synchronized clocks.

For the later analysis, we additionally introduce the following notations.

Notation 3.3.2. The sets $\mathcal{T}_{i|k} \triangleq \{t \in \mathbb{R}_0 | \alpha_k \bar{t}_{i|k} + \beta_k < t \leq \alpha_k \bar{t}_{i+1|k} + \beta_k\}$ define for $i \in \overline{\mathcal{I}}_k$ and $k \in \mathbb{N}_{N+1}$ all admissible global times between two consecutive measurements in the buffer $\mathcal{B}_k$. The union of these sets $\mathcal{T}_k \triangleq \bigcup_{i \in \overline{\mathcal{I}}_k} \mathcal{T}_{i|k}$ denotes for $k \in \mathbb{N}_{N+1}$ all admissible global times between the first and the last measurement in the buffer $\mathcal{B}_k$. Similarly, the sets $\hat{\mathcal{T}}_{i|k} \triangleq \{t \in \mathbb{R}_0 | \hat{\alpha}_k \bar{t}_{i|k} + \hat{\beta}_k < t \leq \hat{\alpha}_k \bar{t}_{i+1|k} + \hat{\beta}_k\}$ define for $i \in \overline{\mathcal{I}}_k$ and $k \in \mathbb{N}_{N+1}$ all admissible estimated global times between two consecutive measurements in the buffer $\mathcal{B}_k$. The union of these sets $\hat{\mathcal{T}}_k \triangleq \bigcup_{i \in \overline{\mathcal{I}}_k} \hat{\mathcal{T}}_{i|k}$ denotes for $k \in \mathbb{N}_{N+1}$ all admissible estimated global times between the first and the last measurement in the buffer $\mathcal{B}_k$.

3.4 Update Step of the Centralized Moving Horizon Observer

Before presenting the update step of the CMHO, we need a suitable mathematical formulation for the discrete-time states $\boldsymbol{x}_{i|k}$ as well as for the estimated counterparts $\hat{\boldsymbol{x}}_{i|k}$. These are required for expressing the measurements $\boldsymbol{y}_{i|k}$ and the estimated counterparts $\hat{\boldsymbol{y}}_{i|k}$ corresponding to the buffer $\mathcal{B}_k$ which are described by the measurement model (3.12).

The continuous-time nonlinear system defined in (3.2) without state disturbance and with the initial value corresponding to the buffer $\mathcal{B}_k$ can be seen in the upper left of Figure 3.4. The discrete-time states $\boldsymbol{x}_{i|k}$ are derived by integrating (3.2) over the sets $\mathcal{T}_{i|k}$ for $i \in \overline{\mathcal{I}}_k$ and

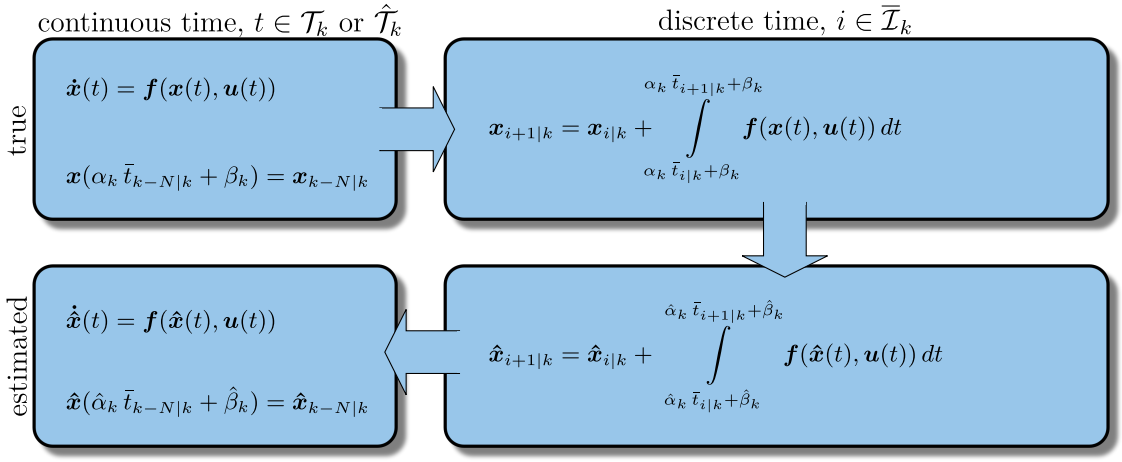


Figure 3.4: Relation between the mathematical formulation of the true and estimated system in continuous and discrete time for the CMHO.

are depicted in the upper right of Figure 3.4. The estimated discrete-time states $\hat{\boldsymbol{x}}_{i|k}$ are derived by copying the equation for $\boldsymbol{x}_{i|k}$ and then replacing all unknowns by their estimated counterparts, see the lower right of Figure 3.4. In order to evaluate the estimated states $\hat{\boldsymbol{x}}_{i|k}$, however, $\hat{\boldsymbol{x}}(t)$ is required which is the solution of the associated estimated continuous-time nonlinear system depicted in the lower left of Figure 3.4. Note that the estimated dynamics are able to reflect exactly the true dynamics.

Now we can define the update step of the CMHO within the presented moving horizon framework (cf. Section 2.1) as follows:

Definition 3.4.1. *The update step of the centralized moving horizon observer is a constrained optimization problem of the form*

$$\min_{\substack{\hat{\alpha}_k, \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k}, \ldots, \hat{\boldsymbol{x}}_{k|k}}} \sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k})) \tag{3.13a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1|k} - \hat{\boldsymbol{x}}_{i|k} - \int_{\hat{\alpha}_k \bar{t}_{i|k}+\hat{\beta}_k}^{\hat{\alpha}_k \bar{t}_{i+1|k}+\hat{\beta}_k} \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))\, dt = \boldsymbol{0}, \quad i \in \overline{\mathcal{I}}_k \tag{3.13b}$$

$$\boldsymbol{c}_{i|k}(\hat{\boldsymbol{x}}_{i|k}) \geq \boldsymbol{0}, \quad i \in \mathcal{I}_k \tag{3.13c}$$

$$\boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k) \geq \boldsymbol{0}. \tag{3.13d}$$

The different components of the problem are given by

- ***cost function*** (3.13a)*:*

  Although the cost function is formulated in a general manner for consistency with the CMHE, it is defined as a least-squares approach

$$J_k(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \ldots, \hat{\boldsymbol{x}}_{k|k}) = \sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k})) = \frac{1}{2} \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k}) - \boldsymbol{y}_{i|k}\|^2. \tag{3.14}$$

- ***state constraints*** (3.13b)*:*

  These equality constraints combine the system dynamics (3.2) with the clock model (3.9) by expressing (3.2) in integral form. It is important to note that the estimated clock parameters appear in the integral bounds. This leads to the fact that not only the estimated measurement values are, as usual, subject to the optimization formulation but also the estimated measurement times.

- ***state constraints*** (3.13c)*:*

  These inequality constraints are optional and enable the incorporation of restrictions for the states and disturbances, see Section 2.1. Special care must be taken to guarantee non-emptiness of the feasible set.

- ***clock parameter constraints*** (3.13d)*:*

  This inequality constraint is optional and enables the incorporation of additional information about the clock parameters. This can be useful, e. g., if some properties about the quality of the clock hardware is known, like its precision.

The optimal solution to this NLP can be derived by means of the methods presented in Section 2.3. However, calculating the optimal solution in a few milliseconds, which is necessary for a real-time application, is a challenging and difficult task and still an active area of research (Diehl et al., 2009). To relax this problem, we propose to compute a suboptimal solution instead of the optimal one. This suboptimal solution has to satisfy the decreasing condition

$$J_k \leq \xi_k J_{k-1}, \quad k \in \mathbb{N}_{N+1}, \tag{3.15}$$

where $\xi_k \in [0, 1[$ are the *decreasing factors* and where the initial cost function value J_N is the cost function value of the $N+1$-th step before the optimization, i. e. $J_N \triangleq J^\circ_{N+1}$. Note that the cost function values itself are used as a measure of optimality. The sequence of the decreasing factors $\{\xi_i\}$ determines on the one hand the solution time and on the other hand the convergence speed. We will show in Chapter 6, that the suboptimal approach is sufficient for the stability of the CMHO.

A consequence of the suboptimal approach for the NLP (3.13) is that its formulation as a simultaneous approach is no longer suitable. As discussed in Section 2.3, this method has the property that the trajectories are only continuous for the optimal solution. This is problematic for the suboptimal approach since the estimation quality of the CMHO depends only on the estimation quality of $\hat{\alpha}_k$,$\hat{\beta}_k$ and $\hat{\boldsymbol{x}}_{k|k}$, see Section 3.7. If the trajectories are discontinuous, the cost function and therefore the decreasing condition (3.15) is no longer a reliable measure for the estimation quality of $\hat{\alpha}_k$,$\hat{\beta}_k$ and $\hat{\boldsymbol{x}}_{k|k}$ and thus for the CMHO. To resolve this issue, we propose to transform the NLP (3.13) into sequential form where the trajectories are always continuous. To this end, we define the overall optimization variable corresponding to the buffer $\mathcal{B}_k$ as

$$\hat{\boldsymbol{p}}_k \triangleq \operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}. \tag{3.16}$$

The constraint (3.13b) uniquely determines $\hat{\boldsymbol{x}}_{i|k}$ for $i \in \mathcal{I}_k$ if $\hat{\boldsymbol{p}}_k$ is fixed. Thus, a function $\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$ can be defined that satisfies (3.13b) for all $\hat{\boldsymbol{p}}_k$.

Definition 3.4.2. *The function $\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$ is a mapping $\hat{\boldsymbol{\phi}} : \hat{\mathcal{T}}_k \times \mathbb{P} \times \mathbb{U} \mapsto \mathbb{R}^n$ which satisfies*

$$i) \qquad \frac{\partial \hat{\boldsymbol{\phi}}}{\partial t}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u}), \boldsymbol{u}(t)), \tag{3.17a}$$

$$ii) \qquad \hat{\boldsymbol{\phi}}(\hat{\alpha}_k\, \bar{t}_{k-N|k} + \hat{\beta}_k, \hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \hat{\boldsymbol{x}}_{k-N|k}. \tag{3.17b}$$

In accordance with the introduced notation, $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ denotes $\hat{\boldsymbol{\phi}}(\hat{\alpha}_k\, \bar{t}_{i|k} + \hat{\beta}_k, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$.

Consequently, we can remove the constraint (3.13b) in the optimization problem (3.13) by substituting the function $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ with $\hat{\boldsymbol{x}}_{i|k}$.

Definition 3.4.3. *The update step of the centralized moving horizon observer in sequential form is*

$$\min_{\hat{\boldsymbol{p}}_k} \sum_{i=k-N}^{k} \Upsilon_{i|k}\left(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u}))\right) \tag{3.18a}$$

subject to:

$$\boldsymbol{c}_{i|k}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \geq \boldsymbol{0}, \quad i \in \mathcal{I}_k \tag{3.18b}$$

$$\boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k) \geq \boldsymbol{0}. \tag{3.18c}$$

This leads to the sequential approach, where in each optimization iteration, the two steps, system simulation and optimization, are performed sequentially, one after the other. Besides the suitability for the suboptimal approach, the advantage of this method is the strongly reduced variable space compared to the problem (3.13). However, the computation of the derivatives is more costly, but there is a certain structure in the problem which will be fully exploited in Chapter 4.

3.5 Update Step of the Centralized Moving Horizon Estimator

The update step of the CMHE is based on the same ideas as the update step of the CMHO. However, the situation is significantly aggravated by the measurement and state disturbances. For instance, the latter has to be considered in the derivation of a suitable mathematical formulation for the discrete-time states $\boldsymbol{x}_{i|k}$ and for the estimated counterparts $\hat{\boldsymbol{x}}_{i|k}$ corresponding to the buffer $\mathcal{B}_k$.

The continuous-time nonlinear system defined in (3.2) with the initial value corresponding to the buffer $\mathcal{B}_k$ can be seen in the upper left of Figure 3.5. The discrete-time states $\boldsymbol{x}_{i|k}$ are

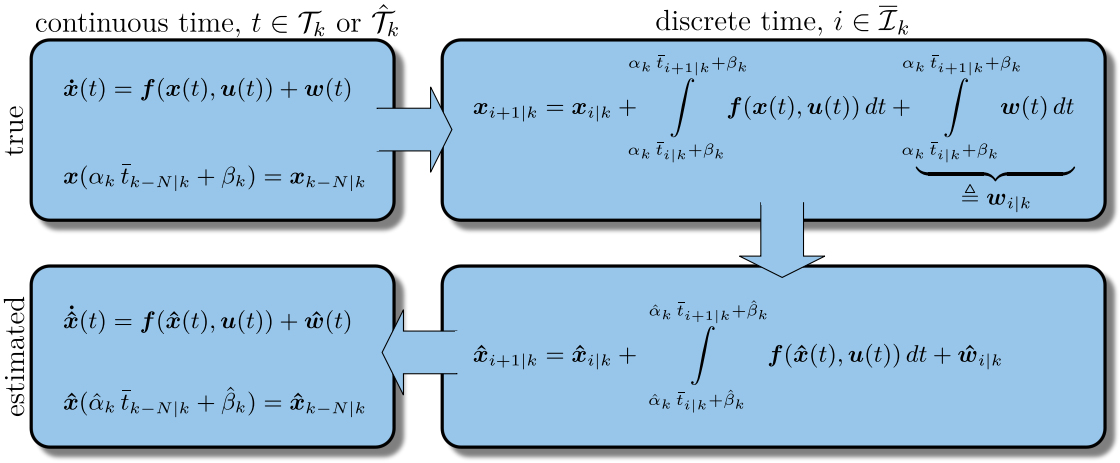


Figure 3.5: Relation between the mathematical formulation of the true and estimated system in continuous and discrete time for the CMHE.

derived by integrating (3.2) over the sets $\mathcal{T}_{i|k}$ for $i \in \overline{\mathcal{I}}_k$ and are depicted in the upper right of Figure 3.5. Note that $\boldsymbol{w}_{i|k}$ are the integral values of $\boldsymbol{w}(t)$ over the sets $\mathcal{T}_{i|k}$ for $i \in \overline{\mathcal{I}}_k$. The estimated discrete-time states $\hat{\boldsymbol{x}}_{i|k}$ are derived by copying the equation for $\boldsymbol{x}_{i|k}$ and then replacing all unknowns by their estimated counterparts, see the lower right of Figure 3.5. Note that $\hat{\boldsymbol{w}}_{i|k}$ are for $i \in \overline{\mathcal{I}}_k$ the estimates of $\boldsymbol{w}_{i|k}$. In order to evaluate the estimated states $\hat{\boldsymbol{x}}_{i|k}$, however, $\hat{\boldsymbol{x}}(t)$ is required which is the solution of the associated estimated continuous-time nonlinear system. This system is depicted in the lower left of Figure 3.5, where $\hat{\boldsymbol{w}}(t)$ is an estimate of $\boldsymbol{w}(t)$. It is important to note, that, in general, $\hat{\boldsymbol{w}}(t)$ and thus the estimated system itself cannot be defined uniquely although their integrated representation is unique.

In other words, this means that, in general, $\hat{\boldsymbol{w}}(t)$ cannot be an exact copy of $\boldsymbol{w}(t)$. On the one hand, this is due to the problem of measuring only the accumulated impact of the continuous disturbance $\boldsymbol{w}(t)$ at discrete times. On the other hand, this is due to the absence of knowledge about the characteristic of $\boldsymbol{w}(t)$, see Assumption 1. Therefore, we make the following assumption to relate uniquely $\hat{\boldsymbol{w}}(t)$ with $\hat{\boldsymbol{w}}_{i|k}$.

Assumption 9. The estimated state disturbance is a piecewise constant function $\hat{\boldsymbol{w}} : \hat{\mathcal{T}}_k \mapsto \mathbb{W} \subset \mathbb{R}^{n_x}$ defined as

$$\hat{\boldsymbol{w}}(t) \triangleq \frac{1}{\hat{\alpha}_k T_{i|k}} \hat{\boldsymbol{w}}_{i|k}, \quad t \in \hat{\mathcal{T}}_{i|k}, \quad i \in \overline{\mathcal{I}}_k, \quad k \in \mathbb{N}_{N+1},$$

where $T_{i|k} \triangleq \bar{t}_{i+1|k} - \bar{t}_{i|k}$ is the difference between two consecutive measurement time stamps in the buffer $\mathcal{B}_k$.

This choice expresses $\hat{\boldsymbol{w}}(t)$ as a zero-order-hold approximation of the determinable integral values $\hat{\boldsymbol{w}}_{i|k}$ and satisfies the integral relation

$$\int_{\hat{\alpha}_k \bar{t}_{i|k}+\hat{\beta}_k}^{\hat{\alpha}_k \bar{t}_{i+1|k}+\hat{\beta}_k} \hat{\boldsymbol{w}}(t)\, dt = \hat{\boldsymbol{w}}_{i|k}, \quad i \in \overline{\mathcal{I}}_k, \quad k \in \mathbb{N}_{N+1}.$$

Now we can define the update step of the CMHE within the presented moving horizon framework (cf. Section 2.1) as follows:

Definition 3.5.1. *The update step of the centralized moving horizon estimator is a constrained optimization problem of the form*

$$\min_{\substack{\hat{\alpha}_k, \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k}, \ldots, \hat{\boldsymbol{x}}_{k|k} \\ \hat{\boldsymbol{w}}_{k-N|k}, \ldots, \hat{\boldsymbol{w}}_{k-1|k}}} \Gamma_k(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N|k}) + \sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k})) + \sum_{i=k-N}^{k-1} \Psi_{i|k}(\hat{\boldsymbol{w}}_{i|k}) \tag{3.19a}$$

subject to:

$$\hat{\boldsymbol{x}}_{i+1|k} - \hat{\boldsymbol{x}}_{i|k} - \int_{\hat{\alpha}_k \bar{t}_{i|k}+\hat{\beta}_k}^{\hat{\alpha}_k \bar{t}_{i+1|k}+\hat{\beta}_k} \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))\, dt - \hat{\boldsymbol{w}}_{i|k} = \boldsymbol{0}, \quad i \in \overline{\mathcal{I}}_k \tag{3.19b}$$

$$\boldsymbol{c}_{i|k}(\hat{\boldsymbol{x}}_{i|k}, \hat{\boldsymbol{w}}_{i|k}) \geq \boldsymbol{0}, \quad i \in \overline{\mathcal{I}}_k \tag{3.19c}$$

$$\boldsymbol{c}_{k|k}(\hat{\boldsymbol{x}}_{k|k}) \geq \boldsymbol{0}, \tag{3.19d}$$

$$\boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k) \geq \boldsymbol{0}. \tag{3.19e}$$

The different components of the problem are given by

➪ ***cost function*** (3.19a)*:*

Depending on the characteristics of $\boldsymbol{x}(0)$, $\boldsymbol{w}_{i|k}$ and $\boldsymbol{v}_{i|k}$, different formulations based on e. g. the l_1-norm, the l_2-norm or the Huber loss-function are appropriate for the cost function. To retain this possibility, the cost function is formulated in a general manner.

However, due to Assumption 1, a natural criterion to define the terms of the cost function consists in resorting to a least-squares approach.

① ***first term:***

This term penalizes the distance between the predictions $\bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N|k}$ *derived in the previous buffer* $\mathcal{B}_{k-1}$ *and the corresponding optimization variables* $\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}$ *of the current buffer* $\mathcal{B}_k$*. Usually, these predictions are the (sub)optimal values derived in the previous buffer, i. e.* $\bar{\alpha}_k \triangleq \hat{\alpha}_{k-1}$, $\bar{\beta}_k \triangleq \hat{\beta}_{k-1}$ *and* $\bar{\boldsymbol{x}}_{k-N|k} \triangleq \hat{\boldsymbol{\phi}}(\hat{\alpha}_{k-1}\, \bar{t}_{k-N|k} + \hat{\beta}_{k-1}, \hat{\boldsymbol{p}}_{k-1}, \boldsymbol{u})$*. The purpose of the first term is twofold. First, it enables the incorporation of information which is not explicitly accounted for in the current buffer, see Section 2.1. Second, it guarantees uniqueness of the solution to the optimization problem even if the current problem is unobservable, see Chapter 5. The nominal least-squares choice for the first term is*

$$\Gamma_k(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N|k}) = \frac{1}{2} \left\| \begin{bmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k} \end{bmatrix} - \begin{bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \\ \bar{\boldsymbol{x}}_{k-N|k} \end{bmatrix} \right\|^2_{\boldsymbol{P}_k^{-1}}, \tag{3.20a}$$

where $\boldsymbol{P}_k^{-1} \geq \boldsymbol{0}$ *determines the weight that is given to the predictions relative to the other terms in the cost function. If we have high (low) confidence in the predictions then* $\boldsymbol{P}_k^{-1}$ *is chosen large (small).*

② ***second term:***

This term penalizes the distance between the measurements $\boldsymbol{y}_{i|k}$ *stored in the buffer* $\mathcal{B}_k$ *and the corresponding estimated measurements* $\boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k})$*. The nominal least-squares choice for the second term is*

$$\sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k})) = \frac{1}{2} \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{x}}_{i|k}) - \boldsymbol{y}_{i|k}\|^2_{\boldsymbol{R}^{-1}}, \tag{3.20b}$$

where $\boldsymbol{R}^{-1} > \boldsymbol{0}$ *determines the weight that is given to the measurement model relative to the other terms in the cost function. If we have high (low) confidence in the measurement model then* $\boldsymbol{R}^{-1}$ *is chosen large (small).*

③ ***third term:***

This term penalizes the state disturbances corresponding to the buffer $\mathcal{B}_k$*. The purpose of the third term is to guarantee uniqueness of the solution to the optimization problem, see Chapter 5. The nominal least-squares choice for the third term is*

$$\sum_{i=k-N}^{k-1} \Psi_{i|k}(\hat{\boldsymbol{w}}_{i|k}) = \frac{1}{2} \sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_{i|k}\|^2_{\boldsymbol{Q}^{-1}}, \tag{3.20c}$$

where $\boldsymbol{Q}^{-1} > \boldsymbol{0}$ *determines the weight that is given to the system model relative to*

the other terms in the cost function. If we have high (low) confidence in the system model then $\boldsymbol{Q}^{-1}$ is chosen large (small).

➪ ***state constraints*** (3.19b)*:*

These equality constraints combine the system dynamics (3.2) *with the clock model* (3.9) *by expressing* (3.2) *in integral form. It is important to note that the estimated clock parameters appear in the integral bounds. This leads to the fact that not only the estimated measurement values are, as usual, subject to the optimization formulation but also the estimated measurement times.*

➪ ***state*** (3.19c) ***and disturbance constraints*** (3.19d)*:*

These inequality constraints are optional and enable the incorporation of restrictions for the states and disturbances, see Section 2.1. Special care must be taken to guarantee non-emptiness of the feasible set.

➪ ***clock parameter constraints*** (3.19e)*:*

This inequality constraint is optional and enables the incorporation of additional information about the clock parameters. This can be useful, e. g., if some properties about the quality of the clock hardware is known, like its precision.

The optimal solution to this NLP can be derived once again by means of the methods presented in Section 2.3. However, with a real-time application in mind, just like in the CMHO case, we propose to compute a suboptimal solution instead of the optimal one. This suboptimal solution has to satisfy the decreasing condition

$$J_k \leq \max\left\{\xi_k J_{k-1}, \delta_J\right\}, \quad k \in \mathbb{N}_{N+1}, \tag{3.21}$$

where $\xi_k \in [0, 1[$ are the *decreasing factors*, $\delta_J \in \mathbb{R}_0$ is a bound for the optimal cost function values and where the initial cost function value J_N is the cost function value of the $N+1$-th step before the optimization, i. e. $J_N \triangleq J^\circ_{N+1}$. Note that the cost function values itself are used as a measure of optimality. The sequence of the decreasing factors $\{\xi_i\}$ determines on the one hand the solution time and on the other hand the convergence speed until the cost function values reach the bound δ_J. This bound is necessary because the cost functions of the CMHE cannot be made arbitrarily small as contrary to cost functions of the CMHO, see Chapter 6.

As in the case of the CMHO, the consequence of the suboptimal approach is that the formulation of the NLP (3.19) as a simultaneous approach is no longer suitable. To resolve this issue, we propose to transform the NLP (3.19) into sequential form. To this end, we define the overall optimization variable corresponding to the buffer $\mathcal{B}_k$ as

$$\hat{\boldsymbol{p}}_k \triangleq \operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \hat{\boldsymbol{w}}_{k-N|k}, \ldots, \hat{\boldsymbol{w}}_{k-1|k}) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}. \tag{3.22}$$

The constraint (3.19b) uniquely determines $\hat{\boldsymbol{x}}_{i|k}$ for $i \in \mathcal{I}_k$ if $\hat{\boldsymbol{p}}_k$ is fixed. Thus, a function $\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$ can be defined that satisfies (3.19b) for all $\hat{\boldsymbol{p}}_k$.

Definition 3.5.2. *The function $\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$ is a mapping $\hat{\boldsymbol{\phi}} : \hat{\mathcal{T}}_k \times \mathbb{P} \times \mathbb{U} \mapsto \mathbb{R}^{n_x}$ which satisfies*

$$i) \qquad \frac{\partial \hat{\boldsymbol{\phi}}}{\partial t}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u}), \boldsymbol{u}(t)) + \hat{\boldsymbol{w}}(t), \tag{3.23a}$$

$$ii) \qquad \hat{\boldsymbol{\phi}}(\hat{\alpha}_k \bar{t}_{k-N|k} + \hat{\beta}_k, \hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \hat{\boldsymbol{x}}_{k-N|k}. \tag{3.23b}$$

In accordance with the introduced notation, $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ denotes $\hat{\boldsymbol{\phi}}(\hat{\alpha}_k \bar{t}_{i|k} + \hat{\beta}_k, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$.

Note that in contrast to the CMHO, the function $\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}_k, \boldsymbol{u})$ of the CMHE also depends on the estimated state disturbance sequence $\{\hat{\boldsymbol{w}}_{i|k}\}_{i \in \overline{\mathcal{I}}}$. By utilizing Definition 3.5.2, we can remove the constraint (3.19b) in the optimization problem (3.19) by substituting the function $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ with $\hat{\boldsymbol{x}}_{i|k}$.

Definition 3.5.3. *The update step of the centralized moving horizon estimator in sequential form is*

$$\min_{\hat{\boldsymbol{p}}_k} \Gamma_k(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N|k}) + \sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) + \sum_{i=k-N}^{k-1} \Psi_{i|k}(\hat{\boldsymbol{w}}_{i|k}) \tag{3.24a}$$

subject to:

$$\boldsymbol{c}_{i|k}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u}), \hat{\boldsymbol{w}}_{i|k}) \geq \boldsymbol{0}, \quad i \in \overline{\mathcal{I}}_k \tag{3.24b}$$

$$\boldsymbol{c}_{k|k}(\hat{\boldsymbol{\phi}}_{k|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \geq \boldsymbol{0}, \tag{3.24c}$$

$$\boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k) \geq \boldsymbol{0}. \tag{3.24d}$$

This leads to the sequential approach, where in each optimization iteration, the two steps, system simulation and optimization, are performed sequentially, one after the other. Besides the superior suitability for the suboptimal approach, the advantage of this method is the strongly reduced variable space compared to the problem (3.19). However, the computation of the derivatives is more costly, but there is a certain structure in the problem which will be fully exploited in Chapter 4.

3.6 Choice of the Initial Conditions

The time until a (sub)optimal solution to the optimization problem of the CMHO (3.18) and the CMHE (3.24) are found, depends mainly on two facts: an efficient optimization algorithm and a proper choice of the initial conditions. The better the latter are, the faster the solution is found. Two consecutive optimization problems of both, the CMHO (3.18) and the CMHE (3.24), are similar in the sense that two consecutive buffers are identical

except one packet. Thus, we use a so-called "warm-start-strategy" to initialize the current optimization problem. This means that the initial conditions of the current optimization problem are taken from the solution of the previous one. Since the optimization problems are almost identical, the expectation is that the (sub)optimal solution of the previous problem is also close to a (sub)optimal solution of the current problem. Consequently, for all but the very first optimization, i. e. $k \in \mathbb{N}_{N+2}$, we choose the initial values as follows. The clock parameters are initialized as

$$\hat{\alpha}_k^\circ = \hat{\alpha}_{k-1}, \tag{3.25a}$$

$$\hat{\beta}_k^\circ = \hat{\beta}_{k-1}, \tag{3.25b}$$

and the initial value for the state is

$$\hat{\boldsymbol{x}}_{k-N|k}^\circ = \hat{\boldsymbol{\phi}}(\hat{\alpha}_{k-1}\,\bar{t}_{k-N|k} + \hat{\beta}_{k-1}, \hat{\boldsymbol{p}}_{k-1}, \boldsymbol{u}), \tag{3.25c}$$

where the function $\hat{\boldsymbol{\phi}}(\cdot)$ has been defined for the CMHO and the CMHE in Definition 3.4.2 and 3.5.2, respectively. The initial values of the state disturbance sequence $\{\hat{\boldsymbol{w}}_{i|k}^\circ\}_{i\in\overline{\mathcal{I}}}$ required by the CMHE depends on the position $j \in [k-N, k]$ of the latest arrived packet in the buffer $\mathcal{B}_k$

$$\text{if } j = k : \hat{\boldsymbol{w}}_{i|k}^\circ = \begin{cases} \hat{\boldsymbol{w}}_{i|k-1}, & i = k-N, \dots, k-2 \\ \boldsymbol{0}, & i = k-1 \end{cases}$$

$$\text{if } k-N < j < k : \hat{\boldsymbol{w}}_{i|k}^\circ = \begin{cases} \hat{\boldsymbol{w}}_{i|k-1}, & i = k-N, \dots, j-2 \\ \frac{\bar{t}_{j|k}-\bar{t}_{j-1|k}}{\bar{t}_{j|k-1}-\bar{t}_{j-1|k-1}}\hat{\boldsymbol{w}}_{j-1|k-1}, & i = j-1 \\ \frac{\bar{t}_{j+1|k}-\bar{t}_{j|k}}{\bar{t}_{j|k-1}-\bar{t}_{j-1|k-1}}\hat{\boldsymbol{w}}_{j-1|k-1}, & i = j \\ \hat{\boldsymbol{w}}_{i-1|k-1}, & i = j+1, \dots, k-1 \end{cases} \tag{3.25d}$$

$$\text{if } j = k-N : \hat{\boldsymbol{w}}_{i|k}^\circ = \begin{cases} \frac{\bar{t}_{k-N+1|k}-\bar{t}_{k-N|k}}{\bar{t}_{k-N|k-1}-\bar{t}_{k-1+N|k-1}}\hat{\boldsymbol{w}}_{k-1+N|k-1}, & i = k-N \\ \hat{\boldsymbol{w}}_{i-1|k-1}, & i = k-N+1, \dots, k-1. \end{cases}$$

The case $j = k$ reflects the nominal case without packet reordering, i. e. the latest arrived packet in the current buffer is the one with the latest information. In this case, no information about the latest disturbance is available in the previous solution. Thus, $\hat{\boldsymbol{w}}_{k-1|k}$ is initialized by $\boldsymbol{0}$. In the remaining two cases, packet reordering occurred, i. e. the latest arrived packet in the current buffer is not the one with the latest information. In this case, information about the new disturbances $\hat{\boldsymbol{w}}_{j-1|k}$ and $\hat{\boldsymbol{w}}_{j|k}$, if $j > k-N$, and $\hat{\boldsymbol{w}}_{k-N|k}$, if $j = k-N$, is available in the previous solution. Thus, these new disturbances receive their initial values by linearly dividing the respective state disturbance from the previous solution $\hat{\boldsymbol{w}}_{j-1|k-1}$, if $j > k-N$, and $\hat{\boldsymbol{w}}_{k-1+N|k-1}$, if $j = k-N$.

However, for the very first optimization, some proper initial guesses have to be provided.

This is especially difficult for the clock parameters $\hat{\alpha}_{N+1}$ and $\hat{\beta}_{N+1}$. Fortunately, these two values are not required right away from the start for the prediction but only when the $N+1$-th packet arrives. Therefore, the information stored in buffer $\mathcal{B}_{N+1}$ can be used to derive proper initial guesses.

In the following, we present two methods for finding proper initial guesses $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ which differ in their complexity, computational load and the achieved accuracy. The idea of both methods is first to find an admissible region for $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ which is then used to generate proper initial guesses. Both methods are afterwards illustrated and compared with each other by an example.

3.6.1 Initial Conditions for the Clock Parameters: Method 1

The idea of the first method is to upper and lower bound the time delays $\tau_{i|N+1}$ of each packet in the buffer $\mathcal{B}_{N+1}$ by some provided guesses $\hat{\tau}_{\min}$ and $\hat{\tau}_{\max}$, respectively, which yields with (3.11) the inequalities

$$t_{i|N+1} - \hat{\alpha}^\circ_{N+1}\bar{t}_{i|N+1} - \hat{\beta}^\circ_{N+1} \geq \hat{\tau}_{\min}, \quad i \in \mathcal{I}_{N+1} \tag{3.26a}$$

$$t_{i|N+1} - \hat{\alpha}^\circ_{N+1}\bar{t}_{i|N+1} - \hat{\beta}^\circ_{N+1} \leq \hat{\tau}_{\max}, \quad i \in \mathcal{I}_{N+1}. \tag{3.26b}$$

Note that the bounds $\hat{\tau}_{\min}$ and $\hat{\tau}_{\max}$ do not have to be tight or even true in order to yield some proper initial guesses for the optimization. The inequalities (3.26) define the polytope

$$\mathcal{M}_{\text{polytope}} = \left\{ \begin{bmatrix} \hat{\alpha}^\circ_{N+1} \\ \hat{\beta}^\circ_{N+1} \end{bmatrix} \in \mathbb{R}^2 \,\middle|\, \begin{bmatrix} \bar{t}_{i|N+1} & 1 \\ -\bar{t}_{i|N+1} & -1 \end{bmatrix} \begin{bmatrix} \hat{\alpha}^\circ_{N+1} \\ \hat{\beta}^\circ_{N+1} \end{bmatrix} \leq \begin{bmatrix} -\hat{\tau}_{\min} + t_{i|N+1} \\ \hat{\tau}_{\max} - t_{i|N+1} \end{bmatrix}, \; i \in \mathcal{I}_{N+1} \right\}. \tag{3.27}$$

A reasonable choice for the initial conditions $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ would be the center of $\mathcal{M}_{\text{polytope}}$. However, its determination involves costly numerical calculations, like finding the inequalities in $\mathcal{M}_{\text{polytope}}$ which are actually necessary for describing the domain of $\mathcal{M}_{\text{polytope}}$ and the subsequent computation of its vertexes, which are inappropriate for real-time application. An alternative choice is the center of an easy to compute subset of $\mathcal{M}_{\text{polytope}}$. To this end, n_s search directions $\boldsymbol{s}_i \in \mathbb{R}^2$ have to be provided, e. g. the unit vectors. Then this subset can be formulated as the polygon $\mathcal{M}_{\text{polygon}} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{n_s}\}$ with the vertices $\boldsymbol{g}_i \in \mathbb{R}^2$ resulting from the solution of the linear programs

$$\boldsymbol{g}_i \triangleq \underset{[\hat{\alpha}^\circ_{N+1}, \hat{\beta}^\circ_{N+1}]^T \in \mathcal{M}_{\text{polytope}}}{\arg\max} \boldsymbol{s}_i^T \begin{bmatrix} \hat{\alpha}^\circ_{N+1} \\ \hat{\beta}^\circ_{N+1} \end{bmatrix}, \quad i = 1, \ldots, n_s. \tag{3.28}$$

A proper choice for the initial conditions $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ is the vertex center of $\mathcal{M}_{\text{polygon}}$

$$\begin{bmatrix} \hat{\alpha}^\circ_{N+1} \\ \hat{\beta}^\circ_{N+1} \end{bmatrix} = \frac{1}{n_s} \sum_{i=1}^{n_s} \boldsymbol{g}_i. \tag{3.29}$$

3.6.2 Initial Conditions for the Clock Parameters: Method 2

The idea of the second method is based on the causality of events. This means that the first packet in $\mathcal{B}_{N+1}$ cannot be transmitted in negative time, i. e.

$$\hat{\alpha}^{\circ}_{N+1}\bar{t}_{1|N+1} + \hat{\beta}^{\circ}_{N+1} \geq 0, \tag{3.30a}$$

and that the last packet in $\mathcal{B}_{N+1}$ cannot be received before being transmitted, i. e.

$$\hat{\alpha}^{\circ}_{N+1}\bar{t}_{N+1|N+1} + \hat{\beta}^{\circ}_{N+1} \leq t_{N+1|N+1}. \tag{3.30b}$$

Moreover, it is reasonable to assume that the clock frequency error is bounded around the nominal value 1, i. e.

$$|\hat{\alpha}^{\circ}_{N+1} - 1| \leq \delta_{\hat{\alpha}}. \tag{3.30c}$$

These inequalities define the trapezoid $\mathcal{M}_{\text{trapezoid}} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \boldsymbol{g}_4\}$ with the vertices $\boldsymbol{g}_i \in \mathbb{R}^2$ given in Table 3.1. A proper choice for the initial conditions $\hat{\alpha}^{\circ}_{N+1}$ and $\hat{\beta}^{\circ}_{N+1}$ is the vertex

Table 3.1: Vertex coordinates of the trapezoid $\mathcal{M}_{\text{trapezoid}} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \boldsymbol{g}_4\}$.

vertex	$\hat{\alpha}$-coordinate	$\hat{\beta}$-coordinate
$\boldsymbol{g}_1$	$1 - \delta_{\hat{\alpha}}$	$-(1 - \delta_{\hat{\alpha}})\bar{t}_{N+1\|N+1} + t_{N+1\|N+1}$
$\boldsymbol{g}_2$	$1 + \delta_{\hat{\alpha}}$	$-(1 + \delta_{\hat{\alpha}})\bar{t}_{N+1\|N+1} + t_{N+1\|N+1}$
$\boldsymbol{g}_3$	$1 + \delta_{\hat{\alpha}}$	$-(1 + \delta_{\hat{\alpha}})\bar{t}_{1\|N+1}$
$\boldsymbol{g}_4$	$1 - \delta_{\hat{\alpha}}$	$-(1 - \delta_{\hat{\alpha}})\bar{t}_{1\|N+1}$

center of $\mathcal{M}_{\text{trapezoid}}$ calculated as

$$\begin{bmatrix} \hat{\alpha}^{\circ}_{N+1} \\ \hat{\beta}^{\circ}_{N+1} \end{bmatrix} = \frac{1}{4}\sum_{i=1}^{4} \boldsymbol{g}_i = \begin{bmatrix} 1 \\ \frac{1}{2}(t_{N+1|N+1} - \bar{t}_{N+1|N+1} - \bar{t}_{1|N+1}) \end{bmatrix}. \tag{3.31}$$

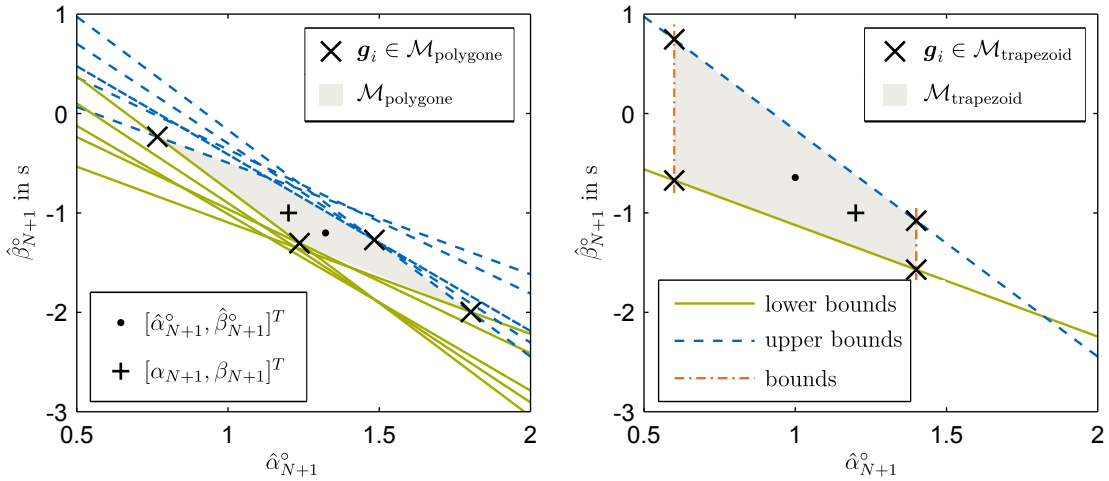
Note that this result is independent of $\delta_{\hat{\alpha}}$.

3.6.3 Illustration and Comparison of Method 1 and Method 2

Both proposed methods are applied to a generic buffer $\mathcal{B}_5$, for which the corresponding values are given in Table 3.2. For method 1, the bounds are chosen as $\hat{\tau}_{\min} = 0\,\text{s}$ and $\hat{\tau}_{\max} = 0.6\,\text{s}$ and the search directions as $\boldsymbol{s}_1 = [2, 1]^T$, $\boldsymbol{s}_2 = [-2, -1]^T$, $\boldsymbol{s}_3 = [-1, 1]^T$ and $\boldsymbol{s}_4 = [1, -1]^T$. The strategy of method 1 and method 2 for finding proper initial guesses are illustrated in Figure 3.6(a) and 3.6(b), respectively. The lower bounds (3.26b), (3.30a) for $\hat{\beta}^{\circ}_{N+1}$, the upper bounds (3.26a), (3.30b) for $\hat{\beta}^{\circ}_{N+1}$ and the bound (3.30c) for $\hat{\alpha}^{\circ}_{N+1}$ are shown as solid green, dashed blue and dash dotted red lines, respectively. The true value

Table 3.2: Relevant values corresponding to the generic buffer $\mathcal{B}_5$.

description	symbol	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
sensor time stamp	$\bar{t}_{i\|5}$	1.122 s	1.450 s	1.776 s	2.005 s	2.280 s
time delay	$\tau_{i\|5}$	0.280 s	0.348 s	0.233 s	0.298 s	0.378 s
arrival time stamp	$t_{i\|5}$	0.626 s	1.088 s	1.364 s	1.704 s	2.114 s



(a) Method 1. (b) Method 2.

Figure 3.6: Illustration of different methods for finding proper initial guesses $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$.

$[\alpha_{N+1}, \beta_{N+1}] = [1.2, -1\,\text{s}]$, the vertices $\boldsymbol{g}_i$ and the resulting choices of the initial values $[\hat{\alpha}^\circ_{N+1}, \hat{\beta}^\circ_{N+1}]_{\text{method 1}} = [1.322, -1.201\,\text{s}]$, $[\hat{\alpha}^\circ_{N+1}, \hat{\beta}^\circ_{N+1}]_{\text{method 2}} = [1, -0.644\,\text{s}]$ are depicted as a vertical cross, as diagonal crosses and as point markers, respectively.

Here and in general, the advantage of method 1 is the more accurate choice of the initial values. This is especially true if the clock skew of the sensor α_{N+1} significantly differs from 1. However, the price to pay for this advantage is the increased complexity, the higher computational load and the in advance determination of the parameters $\hat{\tau}_{\min}$, $\hat{\tau}_{\max}$ and $\boldsymbol{s}_i$. These facts hamper a real-time application of the centralized moving horizon strategies. Therefore, the recommendation is to apply method 2 whenever the sensors are equipped with accurate clocks. For low-cost sensors which possess low-accuracy clocks, however, method 1 might be necessary to get proper initial values.

3.7 Prediction Step

The update step provides information about the system at the discrete times when a new packet arrives and the buffer changes. However, due to the time delays affecting the packet transmissions, the time at which the latest information about the system is gained in the update step lies in the past and is not the current time t where the actual state estimate is required. Moreover, the state estimates have to be provided continuously and not only at dis-

crete times. To address both issues, we define the state prediction function $\hat{\boldsymbol{\phi}}_{\text{pre}}(t, t_{\text{ini}}, \hat{\boldsymbol{x}}_{\text{ini}}, \boldsymbol{u})$ as follows.

Definition 3.7.1. *The state prediction function* $\hat{\boldsymbol{\phi}}_{pre}(t, t_{ini}, \hat{\boldsymbol{x}}_{ini}, \boldsymbol{u})$ *is a mapping* $\hat{\boldsymbol{\phi}}_{pre} : [t_{ini}, \infty] \times \mathbb{R}_0 \times \mathbb{R}^{n_x} \times \mathbb{U} \mapsto \mathbb{R}^{n_x}$ *which satisfies*

$$i) \qquad \frac{\partial \hat{\boldsymbol{\phi}}_{pre}}{\partial t}(t, t_{ini}, \hat{\boldsymbol{x}}_{ini}, \boldsymbol{u}) = \boldsymbol{f}(\hat{\boldsymbol{\phi}}_{pre}(t, t_{ini}, \hat{\boldsymbol{x}}_{ini}, \boldsymbol{u}), \boldsymbol{u}(t)), \tag{3.32a}$$

$$ii) \qquad \hat{\boldsymbol{\phi}}_{pre}(t_{ini}, t_{ini}, \hat{\boldsymbol{x}}_{ini}, \boldsymbol{u}) = \hat{\boldsymbol{x}}_{ini}. \tag{3.32b}$$

Based on the state prediction function, we can express the actual estimated continuous-time state $\hat{\boldsymbol{x}}(t)$ as

$$\hat{\boldsymbol{x}}(t) = \begin{cases} \hat{\boldsymbol{\phi}}_{\text{pre}}(t, 0, \hat{\boldsymbol{x}}^{\circ}, \boldsymbol{u}), & t \in [0, t_{N+1}[\\ \hat{\boldsymbol{\phi}}_{\text{pre}}(t, \hat{\alpha}_i\, \bar{t}_{i|i} + \hat{\beta}_i, \hat{\boldsymbol{x}}_{i|i}, \boldsymbol{u}), & t \in \left[\max\limits_{j \in \mathcal{I}_i} t_{j|i}, \max\limits_{j \in \mathcal{I}_{i+1}} t_{j|i+1}\right[, \quad i \in \mathbb{N}_{N+1}. \end{cases} \tag{3.33}$$

This means that the actual estimated state at time t of both, the CMHO and the CMHE, is the solution to the undisturbed nonlinear system, however, with changing initial times and initial values. Until the $N+1$-th packet has been arrived, the initial time and the initial value is 0 and $\hat{\boldsymbol{x}}^{\circ}$, respectively. Afterwards, the latest information gained in the latest update step of the respective CMHS is used to update the initial time and the initial value to $\hat{\alpha}_i\, \bar{t}_{i|i} + \hat{\beta}_i$ and $\hat{\boldsymbol{x}}_{i|i}$, respectively. Note that the cumbersome formulation of the prediction intervals is necessary since not every arrived packet is incorporated in the buffer and thus the arrival time stamps t_j cannot be directly used to define the prediction intervals.

3.8 Parameter Estimation

Often, not all parameters of the nonlinear system Σ_{S} are known in advance, i. e. the function $\boldsymbol{f}(\cdot)$ in (3.2) depends also on an unknown parameter vector $\boldsymbol{a} \in \mathbb{R}^{n_a}$. This problem can be tackled within the presented framework by imposing a model on the parameter variation

$$\dot{\boldsymbol{a}}(t) = \boldsymbol{f}_{\text{parameter}}(\boldsymbol{a}(t), \boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3.34}$$

and treating the parameter vector $\boldsymbol{a}$ as additional states of the system. If no explicit model for the parameter variation is available, the parameter vector can be assumed constant, i. e. $\dot{\boldsymbol{a}}(t) = \boldsymbol{0}$. In this case, the resulting estimated parameter sequence $\{\hat{\boldsymbol{a}}_i\}_{i=N+1}^{\infty}$ can reflect constant as well as slowly time-varying parameters. This is reasonable since parameters typically vary much slower than states.

One may wonder if the unknown clock parameters $\hat{\alpha}_k$ and $\hat{\beta}_k$ could be treated the same way. The answer to this question is negative since the function $\boldsymbol{f}(\cdot)$ in (3.2) is independent of the clock parameters. The clock parameters come into play only in the respective integral formulation (3.13b) and (3.19b) of the system instead of its differential form (3.2).

3.9 Overall Algorithm

The presented components are assembled in Algorithm 6 to yield the CMHO and the CMHE which solve the Problem 3.1 and 3.2, respectively. The algorithm is identical for the CMHO and the CMHE except in the update step (line 16) and in the required input which is summarized in Table 3.3. Whenever a new packet arrives (line 3) with suitable information (line 6), the buffer is updated (line 8). If enough packets have been arrived (line 10), the information in the buffer is extracted (line 16) to update the actual state estimation (line 23).

Algorithm 6 Centralized moving horizon observer/estimator $\hat{\Sigma}$ (Black, green and red text color valid for CMHO & CMHE, CMHO only, CMHE only, respectively.)

Input: Initial parameters according to Table 3.3

1: Initialization: -global time $t = 0$
-buffer index $k = 0$
-buffer $\mathcal{B}_0 = \{\{\mathbf{0}, 0, 0\}, \ldots, \{\mathbf{0}, 0, 0\}\}$, $|\mathcal{B}_0| = N + 1$
2: **for all** $t \in \mathbb{R}_0$ **do**
3: **if** $t = t_j$ **then**
4: Set: termination condition $\leftarrow$ false **and** position index $i \leftarrow N + 1$
5: **while** termination condition = false **and** $i > 0$ **do**
6: **if** ${}_2\mathcal{P}_{\cdot,j} > {}_2({}_i\mathcal{B}_k)$ **then**
7: Increase buffer index: $k \leftarrow k + 1$
8: Update buffer by inserting packet $\mathcal{P}_{\cdot,j}$ at the i-th position:
$$\mathcal{B}_k = \{{}_2\mathcal{B}_{k-1}, \ldots, {}_i\mathcal{B}_{k-1}, \mathcal{P}_{\cdot,j}, {}_{i+1}\mathcal{B}_{k-1}, \ldots, {}_{N+1}\mathcal{B}_{k-1}\}$$
9: Set: termination condition $\leftarrow$ true
10: **if** $k \geq N + 1$ **then**
11: **if** $k = N + 1$ **then**
12: Set initial conditions $\hat{\alpha}^{\circ}_{N+1}$ and $\hat{\beta}^{\circ}_{N+1}$ by either Method 1 (see Section 3.6.1) or Method 2 (see Section 3.6.2).
13: **else**
14: Set initial conditions by the warm-start-strategy (3.25).
15: **end if**
16: Find a suboptimal solution $\hat{\boldsymbol{p}}_k$ to (3.18)/(3.24) by the gradient based optimization methods presented in Chapter 2 with the derivatives calculated according to the methods derived in Chapter 4 such that the decreasing condition (3.15)/(3.21) is satisfied.
17: Use the suboptimal solution $\hat{\boldsymbol{p}}_k$ to update the initial time and the initial value of the prediction step (3.33) to $\hat{\alpha}_k\, \bar{t}_{k|k} + \hat{\beta}_k$ and $\hat{\boldsymbol{x}}_{k|k}$, respectively.
18: **end if**
19: **end if**
20: Decrease position index: $i \leftarrow i - 1$
21: **end while**
22: **end if**
23: Predict state $\hat{\boldsymbol{x}}(t)$ according to (3.33).
24: **end for**

Table 3.3: Initial parameters for the nominal CMHO and CMHE.

description	CMHO+CMHE	CMHE only	restrictions
buffer size	$N+1$	–	$N+1 \geq (n_x+2)/n_y$
decreasing factors	$\xi_k, k \in \mathbb{N}_{N+1}$	–	$1 > \xi_k \geq 0$
bound	–	δ_J	$\delta_J > 0$
initial state	$\hat{\boldsymbol{x}}^\circ$	–	–
initial disturbances	–	$\hat{\boldsymbol{w}}^\circ_{i\|N+1}, i \in \overline{\mathcal{I}}_{N+1}$	–
initial clock parameters:			
-Method 1	$\hat{\tau}_{\min}, \hat{\tau}_{\max}$	–	$\hat{\tau}_{\max} > \hat{\tau}_{\min} \geq 0$
	$\boldsymbol{s}_i, i \in \{1, \ldots, n_s\}$	–	–
-Method 2	–	–	–
weighting matrices	–	$\boldsymbol{P}_k^{-1}, k \in \mathbb{N}_{N+1}$	$\boldsymbol{P}_k^{-1} \geq \boldsymbol{0}$
	–	$\boldsymbol{R}^{-1}$	$\boldsymbol{R}^{-1} > \boldsymbol{0}$
	–	$\boldsymbol{Q}^{-1}$	$\boldsymbol{Q}^{-1} > \boldsymbol{0}$

3.10 Summary

In this chapter, we have presented the CMHO and the CMHE within a common framework for the undisturbed and the disturbed centralized NCS architecture, respectively. Both strategies have been explicitly designed to deal simultaneously with the network-induced imperfections of unknown and variable packet delays which include the possibility of packet reordering, unknown and variable packet drops, unsynchronized sensor clocks, and limited energy supplies of the sensors. To overcome these challenges, we have introduced event-based sampling along with time stamping of the resulting measurements. Moreover, we have proposed an affine clock model for the sensor clocks and have extended the moving horizon to a buffer logic. This buffer serves as the information basis for both centralized moving horizon strategies and has enabled us along with the other introduced steps to formulate the state estimation problems as suitable optimization problems where we additionally estimate the unknown clock parameters of the sensors. To achieve a practical feasible implementation of these optimization problems in real-time and thus of the CMHO and CMHE, we have proposed the following two steps which significantly contribute to this goal. First, we have introduced a suboptimal approach which requires only suboptimal instead of optimal solutions. Second, we have provided efficient methods for generating proper initial conditions for finding (sub)optimal solutions.

Chapter 4

Efficient Derivative Calculation

In Chapter 3, the centralized moving horizon observer (CMHO) and the centralized moving horizon estimator (CMHE) have been presented. The update step of both strategies consists of an optimization problem. When solving these problems with the optimization algorithms presented in Section 2.4, the derivatives of the Lagrangian to each optimization problem is required. In this chapter, we investigate how the gradient and the Hessian of these Lagrangians can be efficiently calculated. It is important to note that these derivatives do not only play a crucial role in the efficient solution of the optimization problems but are also of key importance for the observability and stability analysis of Chapter 5 and 6, respectively.

After recalling the different derivative calculation methods presented in Section 2.5, the adjoint method does seem at a first glance like the appropriate and straightforward choice for computing the gradient. However, the networked situation is far more complex and we have to consider the whole picture. First of all, the special nature of the optimization problem of the update steps is reflected in the derivative calculation. The state constraint for both centralized moving horizon strategies (CMHS) is not anymore a conventional difference equation but rather an integral equation resulting from the underlying differential equation (cf. Section 3.4 and 3.5) where the clock parameters appear in the integral bounds. This causes the following severe effects on the sensitivity and adjoint method summarized in Algorithm 3 (page 52) and 4 (page 53), respectively. The difference equations in line 1 and 2 of the former and in line 1 of the latter are replaced by their differential equations counterpart. Moreover, the calculation of the derivatives $\partial \boldsymbol{f}_i / \partial \hat{\boldsymbol{p}}$ in line 3 of the adjoint algorithm is not anymore a simple derivative calculation but requires the solution of sensitivity matrix differential equations which are similar to the sensitivity case. The calculation of the solution to these differential equations is now the dominating computational load for both methods. The number of ODEs that have to be solved for evaluating the gradient is depicted in Table 4.1. While the sensitivity and the adjoint method show the best efficiency in the CMHO case, the CMHE case is a clear vote in favor of the latter. However, the performance of the optimization algorithms does not only depend on the gradient but also on the Hessian. Law and Sharma (1997) have found by numerical comparison that for a sum-of-squares cost function, a Gauss-Newton strategy with the sensitivity method is superior to a quasi-Newton approach with the adjoint method. This is mainly due to the fact that the Gauss-Newton method provides an excellent approximation

Table 4.1: Number of ODEs required for calculating the gradient of the Lagrangian.

	ODEs for CMHO	ODEs for CMHE
finite-difference approximation	$2n_x^2 + 4n_x$	$2(N+1)n_x^2 + 4n_x$
sensitivity method	$n_x^2 + \ n_x$	$(N+1)n_x^2 + \ n_x$
adjoint method	$n_x^2 + \ n_x$	$2n_x^2 + \ n_x$

of the Hessian almost for free.

Finally, one may wonder why automatic and symbolic differentiation are inappropriate. The application of the former is hampered by the problem formulation which requires the manual implementation of a specially tailored ODE-solver for incorporating the influence of the clock parameters on the state differential equation. The general lack of a closed-form representation of the solution to this state differential equation renders the utilization of the latter infeasible.

Based on these considerations, the contribution of this chapter is three-fold. First, we present for both CMHS a sensitivity method for calculating the gradient of the Lagrangian for the optimization problem of the respective update step. Second, we exploit the structure of the state sensitivities resulting from the CMHE case such that the number of ODEs required for calculating the gradient is independent of the number of optimization variables and equal to the adjoint case, namely $2n_x^2 + n_x$. Third, we derive for both CMHS an efficient method for calculating an excellent approximation of the Hessian of the respective Lagrangian almost for free for arbitrarily cost functions based on the already available state sensitivities.

The remainder of this chapter is organized as follows. In Section 4.1, we state the problem formulation. The sensitivity method and the structure exploiting version for calculating the gradient of the Lagrangian to the optimization problem resulting from the update step of the CMHO and the CMHE are presented in Section 4.2 and 4.3, respectively. The corresponding Hessian approximation method is shown in Section 4.4 and 4.5 for the CMHO and the CMHE, respectively. In Section 4.6, we verify the efficiency of the presented derivative calculation strategies by investigating a networked continuously-stirred tank reactor. Finally, we conclude this chapter with a summary given in Section 4.7.

4.1 Problem Formulation

The problem formulations are stated as follows.

Problem 4.1 (CMHO Derivative Calculation)**.** *Let the Lagrangian to the optimization problem of the CMHO update step* (3.18) *be given by*

$$L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k) = \sum_{i=k-N}^{k} \Upsilon_{i|k}\left(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u}))\right) - \sum_{i=k-N}^{k} \boldsymbol{\lambda}_{i|k}^T \boldsymbol{c}_{i|k}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) - \boldsymbol{\mu}_k^T \boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k), \quad (4.1)$$

where $\underline{\boldsymbol{\lambda}}_k = \mathrm{col}(\boldsymbol{\lambda}_{i|k}, i \in \mathcal{I}_k, \boldsymbol{\mu}_k)$ *are the overall Lagrange multipliers and where* $\hat{\boldsymbol{\phi}}(\cdot)$ *has been defined in Definition 3.4.2.*

The problem is to calculate the exact gradient $\partial L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k)/\partial \hat{\boldsymbol{p}}_k$ *and an approximation of the Hessian* $\partial^2 L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k)/\partial \hat{\boldsymbol{p}}_k^2$ *in an efficient way.*

Problem 4.2 (CMHE Derivative Calculation)**.** *Let the Lagrangian to the optimization problem of the CMHE update step (3.24) be given by*

$$\begin{aligned} L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k) = \Gamma_k(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k}, \bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N|k}) + \sum_{i=k-N}^{k} \Upsilon_{i|k}(\boldsymbol{y}_{i|k}, \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \\ + \sum_{i=k-N}^{k-1} \Psi_{i|k}(\hat{\boldsymbol{w}}_{i|k}) - \sum_{i=k-N}^{k-1} \boldsymbol{\lambda}_i^T \boldsymbol{c}_{i|k}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u}), \hat{\boldsymbol{w}}_{i|k}) - \boldsymbol{\lambda}_k^T \boldsymbol{c}_{k|k}(\hat{\boldsymbol{\phi}}_{k|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \\ - \boldsymbol{\mu}_k^T \boldsymbol{d}_k(\hat{\alpha}_k, \hat{\beta}_k), \end{aligned} \tag{4.2}$$

where $\underline{\boldsymbol{\lambda}}_k = \mathrm{col}(\boldsymbol{\lambda}_{i|k}, i \in \mathcal{I}_k, \boldsymbol{\mu}_k)$ *are the overall Lagrange multipliers and where* $\hat{\boldsymbol{\phi}}(\cdot)$ *has been defined in Definition 3.5.2.*

The problem is to calculate the exact gradient $\partial L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k)/\partial \hat{\boldsymbol{p}}_k$ *and an approximation of the Hessian* $\partial^2 L_k(\hat{\boldsymbol{p}}_k, \underline{\boldsymbol{\lambda}}_k)/\partial \hat{\boldsymbol{p}}_k^2$ *in an efficient way.*

Both problems can be tackled within a fixed buffer $\mathcal{B}_k$. Thus, to increase the understandability, we ease the presentation by omitting the subscripts indicating the buffer affiliation. Moreover, we suppress the arguments of all functions from the notation when the meaning is clear. It is important to recall that throughout Part II of this thesis including the current chapter, the *Assumptions 1-9 are supposed to hold without explicitly stating them.*

4.2 Gradient of the Lagrangian for the CMHO

Following the basic principle of the sensitivity method, the gradient of the Lagrangian (4.1) can be stated as follows.

Lemma 4.2.1. *The exact gradient of the Lagrangian L defined in (4.1) with respect to* $\hat{\boldsymbol{p}}$ *is*

$$\frac{\partial L}{\partial \hat{\boldsymbol{p}}} = \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i} - \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial \boldsymbol{c}_i}{\partial \hat{\boldsymbol{\phi}}_i}^T \boldsymbol{\lambda}_i - \frac{\partial \boldsymbol{d}}{\partial \hat{\boldsymbol{p}}}^T \boldsymbol{\mu}, \tag{4.3}$$

where $\partial \hat{\boldsymbol{\phi}}_i/\partial \hat{\boldsymbol{p}}$ *are the first-order state sensitivities.*

Proof. Equation (4.3) results from differentiating (4.1) with respect to $\hat{\boldsymbol{p}}$ where the dependence of $\hat{\boldsymbol{\phi}}$ on $\hat{\boldsymbol{p}}$ is taken into account by Lemma B.1. □

The straightforward idea to calculate the first-order state sensitivities would be to directly differentiate the function $\hat{\boldsymbol{\phi}}_i$ defined in Definition 3.4.2 with respect to $\hat{\boldsymbol{p}}$. However, this procedure is not viable as no general closed-form representation of the function $\hat{\boldsymbol{\phi}}_i$ can be

given. Due to the fact that $\hat{\boldsymbol{\phi}}_i$ is described as a solution to an ODE, a natural approach is to describe the first-order state sensitivities as (possibly appropriately combined) solutions to first-order sensitivity ODEs. To this end, we classify the first-order state sensitivities into the following categories 1 and 2

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} = \Bigg[\underbrace{\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}}, \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}}}_{\text{category 2}}, \quad \underbrace{\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}}}_{\text{category 1}}\Bigg], \quad i \in \mathcal{I}, \tag{4.4}$$

depending on the required mathematical representation level to fully describe the influence of the different parameters in $\hat{\boldsymbol{p}}$ on $\hat{\boldsymbol{\phi}}_i$. In other words, this classification answers the question which level is sufficient to investigate in order to calculate the first-order state sensitivities. In category 1, the influence of the parameters on $\hat{\boldsymbol{\phi}}_i$ can be fully covered on the ODE level, i. e. it is sufficient to investigate only the ODE (3.17). In category 2, its integral equation

$$\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{p}}, \boldsymbol{u}) = \hat{\boldsymbol{x}}_{k-N} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}, \boldsymbol{u}), \boldsymbol{u}(t))\, dt, \quad i \in \mathcal{I} \tag{4.5}$$

is additionally required to fully cover the influence of the corresponding parameters (cf. Section 3.4).

First, we tackle the problem of calculating the first-order state sensitivities in category 1. To this end, the following notation is introduced.

Notation 4.2.1. The abbreviation $\boldsymbol{X}(t)$ denotes the first-order state sensitivity $\partial \hat{\boldsymbol{\phi}}(t)/\partial \hat{\boldsymbol{x}}_{k-N}$.

Lemma 4.2.2. *The first-order state sensitivity $\boldsymbol{X}(t) \in \mathbb{R}^{n_x \times n_x}$ satisfies for $t \in \hat{\mathcal{T}}$ the following first-order sensitivity matrix differential equation*

$$\dot{\boldsymbol{X}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, \boldsymbol{X}(t), \tag{4.6a}$$

with the initial value

$$\boldsymbol{X}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) = \boldsymbol{I}. \tag{4.6b}$$

The unique solution is

$$\boldsymbol{X}(t) = \boldsymbol{\Phi}(t, \hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta})\, \boldsymbol{X}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}), \tag{4.6c}$$

where $\boldsymbol{\Phi}(\cdot) \in \mathbb{R}^{n_x \times n_x}$ denotes the time-varying transition matrix.

Proof. Equation (4.6a) and (4.6b) results from differentiating (3.17a) and (3.17b) with respect to $\hat{\boldsymbol{x}}_{k-N}$, respectively. The solution approach for linear time-varying matrix differential equations and linear time-varying systems (see Kailath (1980, pg. 595-601) or Ludyk (1995, pg. 40-55)) are identical. Its application to (4.6a) combined with (4.6b) leads to (4.6c). □

Next, we introduce the following notation which facilitates the compact description of the first-order state sensitivities.

Notation 4.2.2. The abbreviation $\boldsymbol{X}_b^a$ denotes the solution of (4.6a) at the time $\hat{\alpha}\,\bar{t}_b + \hat{\beta}$ with the initial value $\boldsymbol{I}$ at the initial time $\hat{\alpha}\,\bar{t}_a + \hat{\beta}$.

As previously discussed, we need additionally the integral equation (4.5) for the first-order state sensitivities in category 2. The idea now is to interpret the integral equation (4.5) as a parameter integral in terms of $\hat{\alpha}$ and $\hat{\beta}$ and to apply the Leibniz integral rule (Flanders, 1973). The outcome is combined with Lemma 4.2.2 to yield the desired sensitivities in category 2. The resulting first-order state sensitivities for both categories are stated in the following Theorem.

Theorem 4.2.3. *The first-order state sensitivities in* (4.4) *are for* $i \in \mathcal{I}$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \boldsymbol{X}_i^{k-N} \tag{4.7a}$$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} \tag{4.7b}$$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}} = \boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N}. \tag{4.7c}$$

Proof. Equation (4.7a) is a direct result of Lemma 4.2.2 and Notation 4.2.2. Application of the Leibniz integral rule (Flanders, 1973) to the integral equation (4.5) reveals

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \boldsymbol{0} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\alpha}}\, dt + \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{f}_{k-N},$$

which can be further rearranged by the chain rule to yield

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{\phi}}|_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\alpha}}\bigg|_{k-N} dt + \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{f}_{k-N}.$$

Note that $\partial \hat{\boldsymbol{\phi}}/\partial \hat{\alpha}\big|_{k-N}$ and $\partial \hat{\boldsymbol{\phi}}_{k-N}/\partial \hat{\alpha}$ are not identical. The difference between both lies in the interchanged order of calculating the derivative and evaluating the expression at the corresponding time. While the latter term vanishes, the former term is

$$\frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\alpha}}\bigg|_{k-N} = \lim_{t \to \hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}} \frac{\partial}{\partial \hat{\alpha}} \left(\hat{\boldsymbol{x}}_{k-N} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{t} \boldsymbol{f}\, dt \right) = -\bar{t}_{k-N} \boldsymbol{f}_{k-N}$$

and can be taken out of the integral to yield

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \bar{t}_i \boldsymbol{f}_i - \left(\boldsymbol{I} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{x}}_{k-N}}\, dt \right) \bar{t}_{k-N} \boldsymbol{f}_{k-N}.$$

The term in brackets is the right-hand side of the integral equation corresponding to (4.6a) and thus identical to $\boldsymbol{X}_i^{k-N}$ which proves (4.7b). By noting that $\partial\hat{\boldsymbol{\phi}}/\partial\hat{\beta}\big|_{k-N}$ is equivalent to $-\boldsymbol{f}_{k-N}$, the prove of (4.7c) goes along the very same lines as the prove of (4.7b) and is therefore omitted. □

The overall number of ODEs that have to be solved for evaluating the gradient of the Lagrangian is $n_x^2 + n_x$ since no additional ODEs have to be solved for the first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$. Note that the difference between both results from the different derivative of the clock model (3.9) with respect to the corresponding clock parameter.

4.3 Gradient of the Lagrangian for the CMHE

The main problem of calculating the gradient of the Lagrangian in the CMHE case stems from the assumption that the estimated state disturbance is a piecewise constant function (cf. Assumption 9) which significantly aggravates the sensitivity calculation of $\hat{\boldsymbol{\phi}}_i$. Nevertheless, we follow again the basic principle of the sensitivity method and derive the following Lemma.

Lemma 4.3.1. *The exact gradient of the Lagrangian L defined in (4.2) with respect to $\hat{\boldsymbol{p}}$ is*

$$\frac{\partial L}{\partial \hat{\boldsymbol{p}}} = \frac{\partial \Gamma}{\partial \hat{\boldsymbol{p}}} + \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i} + \sum_{i=k-N}^{k-1} \frac{\partial \Psi_i}{\partial \hat{\boldsymbol{p}}} - \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial \boldsymbol{c}_i}{\partial \hat{\boldsymbol{\phi}}_i}^T \boldsymbol{\lambda}_i - \frac{\partial \boldsymbol{d}}{\partial \hat{\boldsymbol{p}}}^T \boldsymbol{\mu}, \tag{4.8}$$

where $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\boldsymbol{p}}$ are the first-order state sensitivities.

Proof. Equation (4.8) results from differentiating (4.2) with respect to $\hat{\boldsymbol{p}}$ where the dependence of $\hat{\boldsymbol{\phi}}$ on $\hat{\boldsymbol{p}}$ is taken into account by Lemma B.1. □

As in the CMHO case, the straightforward idea to calculate the first-order state sensitivities by directly differentiating the function $\hat{\boldsymbol{\phi}}_i$ defined in Definition 3.5.2 with respect to $\hat{\boldsymbol{p}}$ is not viable due to the lack of a general closed-form expression of $\hat{\boldsymbol{\phi}}$. Thus, we describe the first-order state sensitivities as (possibly appropriately combined) solutions to first-order sensitivity ODEs. To this end, we classify the first-order state sensitivities once again into the following categories 1 and 2

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} = \Bigg[\underbrace{\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}}, \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}}}_{\text{category 2}}, \underbrace{\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}}, \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{w}}_{k-N}}, \ldots, \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{w}}_{k-1}}}_{\text{category 1}}\Bigg], \quad i \in \mathcal{I}, \tag{4.9}$$

depending on the required mathematical representation level to fully describe the influence of the different parameters in $\hat{\boldsymbol{p}}$ on $\hat{\boldsymbol{\phi}}_i$. Recall that this classification answers the question which level is sufficient to investigate in order to calculate the first-order state sensitivities. In category 1, the influence of the parameters on $\hat{\boldsymbol{\phi}}_i$ can be fully covered on the ODE level,

i. e. it is sufficient to investigate the ODE (3.23). In category 2, the integral equation

$$\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{p}}, \boldsymbol{u}) = \hat{\boldsymbol{x}}_{k-N} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \boldsymbol{f}(\hat{\boldsymbol{\phi}}(t, \hat{\boldsymbol{p}}, \boldsymbol{u}), \boldsymbol{u}(t)) + \hat{\boldsymbol{w}}(t)\, dt, \quad i \in \mathcal{I} \tag{4.10}$$

is additionally required to fully cover the influence of the corresponding parameters.

First, we tackle the problem of calculating the first-order state sensitivities in category 1. To this end, the following notation is introduced.

Notation 4.3.1. The abbreviations $\boldsymbol{X}(t)$ and ${}^{j}\boldsymbol{Z}(t)$ denote for $j \in \overline{\mathcal{I}}$ the first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}(t)/\partial\hat{\boldsymbol{x}}_{k-N}$ and $\partial\hat{\boldsymbol{\phi}}(t)/\partial\hat{\boldsymbol{w}}_j$, respectively.

Lemma 4.3.2. *The first-order state sensitivity $\boldsymbol{X}(t) \in \mathbb{R}^{n_x \times n_x}$ satisfies for $t \in \hat{\mathcal{T}}$ the following first-order sensitivity matrix differential equation*

$$\dot{\boldsymbol{X}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, \boldsymbol{X}(t), \tag{4.11a}$$

with the initial value

$$\boldsymbol{X}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) = \boldsymbol{I}. \tag{4.11b}$$

The unique solution is

$$\boldsymbol{X}(t) = \boldsymbol{\Phi}(t, \hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta})\, \boldsymbol{X}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}), \tag{4.11c}$$

where $\boldsymbol{\Phi}(\cdot) \in \mathbb{R}^{n_x \times n_x}$ denotes the time-varying transition matrix.

Proof. Equation (4.11a) and (4.11b) results from differentiating (3.23a) and (3.23b) with respect to $\hat{\boldsymbol{x}}_{k-N}$, respectively. The solution approach for linear time-varying matrix differential equations and linear time-varying systems (see Kailath (1980, pg. 595-601) or Ludyk (1995, pg. 40-55)) are identical. Its application to (4.11a) combined with (4.11b) leads to (4.11c). □

Note that Lemma 4.3.2 and Lemma 4.2.2 are identical except for the different definition of the function $\hat{\boldsymbol{\phi}}$. The remaining first-order state sensitivities in category 1 are derived in the following Lemma.

Lemma 4.3.3. *The first-order state sensitivities ${}^{j}\boldsymbol{Z}(t) \in \mathbb{R}^{n_x \times n_x}$ satisfy for $j \in \overline{\mathcal{I}}$ and $t \in \hat{\mathcal{T}}$ the following first-order sensitivity matrix differential equation*

$${}^{j}\dot{\boldsymbol{Z}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, {}^{j}\boldsymbol{Z}(t) + \begin{cases} \boldsymbol{0}, & t \in \bigcup_{i=k-N}^{j-1} \hat{\mathcal{T}}_i \\ \dfrac{1}{\hat{\alpha}\, T_j}\boldsymbol{I}, & t \in \hat{\mathcal{T}}_j \\ \boldsymbol{0}, & t \in \bigcup_{i=j+1}^{k-1} \hat{\mathcal{T}}_i, \end{cases} \tag{4.12a}$$

with the initial value

$$ {}^{j}\boldsymbol{Z}(\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}) = \mathbf{0}. \tag{4.12b} $$

The unique solution is

$$ {}^{j}\boldsymbol{Z}(t) = \begin{cases} \mathbf{0}, & t \in \bigcup_{i=k-N}^{j-1} \hat{\mathcal{T}}_i \\ \dfrac{1}{\hat{\alpha}\,T_j} \displaystyle\int_{\hat{\alpha}\,\bar{t}_j+\hat{\beta}}^{t} \boldsymbol{\Phi}(t,\tau)\,d\tau, & t \in \hat{\mathcal{T}}_j \\ \dfrac{1}{\hat{\alpha}\,T_j} \displaystyle\int_{\hat{\alpha}\,\bar{t}_j+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{j+1}+\hat{\beta}} \boldsymbol{\Phi}(t,\tau)\,d\tau, & t \in \bigcup_{i=j+1}^{k-1} \hat{\mathcal{T}}_i, \end{cases} \tag{4.12c} $$

where $\boldsymbol{\Phi}(\cdot) \in \mathbb{R}^{n_x \times n_x}$ *denotes the time-varying transition matrix.*

Proof. Equation (4.12a) and (4.12b) results from differentiating (3.23a) and (3.23b) with respect to $\hat{\boldsymbol{w}}_j$, respectively. Note that the piecewise character of (4.12a) is due to the piecewise definition of $\hat{\boldsymbol{w}}(t)$ in Assumption 9. The piecewise matrix differential equation (4.12a) is for each of the three time intervals an ordinary linear matrix differential equation with corresponding initial value. While the initial condition for the first ordinary matrix differential equation is given in (4.12b), the initial condition for the second and third one is the final condition of the first and second one, respectively. The solution approach for ordinary linear time-varying matrix differential equations is identical to the one for linear time-varying systems (see Kailath (1980, pg. 595-601) or Ludyk (1995, pg. 40-55)). Its application to the three cases with the corresponding time intervals and initial values leads to three solutions which are the three cases in (4.12c). □

Evaluating only the non-zero elements in the first-order sensitivities is as costly as solving $(\frac{N}{2}+\frac{3}{2})n_x^2+n_x$ ODEs over $\hat{\mathcal{T}}$. To further reduce this complexity, we exploit the common structure of the first-order sensitivity matrix differential equations. This means that we break down the problem of determining the sensitivities $\boldsymbol{X}(\hat{\alpha}\,\bar{t}_i+\hat{\beta})$ and ${}^{j}\boldsymbol{Z}(\hat{\alpha}\,\bar{t}_i+\hat{\beta})$ on the set $\hat{\mathcal{T}}$ to independent problems on the sets $\hat{\mathcal{T}}_i$, $i \in \overline{\mathcal{I}}$. Afterwards, the solutions to these subproblems are assembled in a suitable manner to yield the desired sensitivities. The advantage of this procedure is two-fold. First, several of these subproblems are identical due to the common underlying structure and thus need to be solved only once. Second, the solutions to these subproblems can be used to calculate the first-order sensitivities in category 2. To this end, the following notation is introduced.

Notation 4.3.2. The abbreviations $\boldsymbol{X}_b^a$ and ${}^{j}\boldsymbol{Z}_b^a$ denote the solutions of (4.11c) and (4.12c) at the time $\hat{\alpha}\,\bar{t}_b+\hat{\beta}$ with the initial value $\boldsymbol{I}$ and $\mathbf{0}$ at the initial time $\hat{\alpha}\,\bar{t}_a+\hat{\beta}$, respectively.

Lemma 4.3.4. *The solution $\boldsymbol{X}_b^a$ and ${}^d\boldsymbol{Z}_b^a$ satisfy for $a, b, c \in \mathcal{I}$ and $d \in \overline{\mathcal{I}}$ with $a \leq d < b < c$ the following properties*

$$\begin{aligned} (i) \quad \boldsymbol{X}_c^a &= \boldsymbol{X}_c^b \boldsymbol{X}_b^a & (iv) \quad {}^d\boldsymbol{Z}_b^a &= {}^d\boldsymbol{Z}_b^d \\ (ii) \quad (\boldsymbol{X}_b^a)^{-1} &= \boldsymbol{X}_a^b & (v) \quad {}^d\boldsymbol{Z}_b^d &= \boldsymbol{X}_b^{d+1d} \boldsymbol{Z}_{d+1}^d \\ (iii) \quad \boldsymbol{X}_a^a &= \boldsymbol{I} & (vi) \quad {}^d\boldsymbol{Z}_a^a &= \boldsymbol{0}. \end{aligned}$$

Proof. The proof is based on the properties of the state transition matrix $\boldsymbol{\Phi}(\cdot)$ which are stated in Kailath (1980, pg. 599) or Ludyk (1995, pg. 47): (i) $\boldsymbol{\Phi}(t_2, t_0) = \boldsymbol{\Phi}(t_2, t_1)\,\boldsymbol{\Phi}(t_1, t_0)$, (ii) $\boldsymbol{\Phi}(t_1, t_0)^{-1} = \boldsymbol{\Phi}(t_0, t_1)$ and (iii) $\boldsymbol{\Phi}(t_0, t_0) = \boldsymbol{I}$. Furthermore, the Notation 4.3.2 implies together with (4.11c) that $\boldsymbol{X}_b^a = \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_b + \hat{\beta}, \hat{\alpha}\,\bar{t}_a + \hat{\beta})$. The combination of both facts proofs the properties (i)-(iii). Writing the solution (4.12c) in terms of the Notation 4.3.2 and noting that the solution (4.12c) is identical to $\boldsymbol{0}$ in the first case proofs the property (iv). For $i > j$, the solution (4.12c) is transformed by using the properties of the state transition matrix

$${}^d\boldsymbol{Z}_b^d = \int_{\hat{\alpha}\,\bar{t}_d+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{d+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_b + \hat{\beta}, \tau)\,\frac{1}{\hat{\alpha}\,T_d}\,d\tau = \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_b + \hat{\beta}, \hat{\alpha}\,\bar{t}_{d+1} + \hat{\beta})\,{}^d\boldsymbol{Z}_{d+1}^d = \boldsymbol{X}_b^{d+1d} \boldsymbol{Z}_{d+1}^d$$

which proofs the property (v). The property (vi) is just the initial value $\boldsymbol{0}$. □

These properties are key to decompose the first-order state sensitivities as follows.

Theorem 4.3.5. *The first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\boldsymbol{x}}_{k-N}$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\boldsymbol{w}}_j$ in (4.9) are for $i \in \mathcal{I}$*

$$\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\boldsymbol{x}}_{k-N}} = \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \ldots \boldsymbol{X}_{k-N+1}^{k-N} \tag{4.13a}$$

$$\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\boldsymbol{w}}_j} = \begin{cases} \boldsymbol{0}, & i < j+1 \\ \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \ldots \boldsymbol{X}_{j+2}^{j+1j} \boldsymbol{Z}_{j+1}^j, & i \geq j+1 \end{cases}, \quad j \in \overline{\mathcal{I}}. \tag{4.13b}$$

Proof. Equation (4.13a) and (4.13b) are derived by applying the properties stated in Lemma 4.3.4 to the solutions (4.11c) and (4.12c), respectively. □

The advantage of this Theorem over the finite-difference approximation and the approach described in Lemma 4.3.2 and 4.3.3 is two-fold. First, the number of ODEs that have to be solved over $\hat{\mathcal{T}}$ is independent of N, namely $2n_x^2 + n_x$. In other words, the complexity of determining the first-order state sensitivities is independent of the number of unknowns $\hat{\boldsymbol{w}}_i$ and independent of the number of first-order state sensitivities ${}^j\boldsymbol{Z}$. Second, each subproblem can be solved independently and thus in parallel. It is important to note that these advantages does not only hold for the networked scenario but also for conventional MHEs, see Philipp (2011a).

Now, the first-order sensitivities in category 2 are considered. As discussed previously, we need additionally the integral equation (4.10) for the first-order state sensitivities in category 2. Just like in the CMHO case, the idea is to interpret the integral equation (4.10) as a

parameter integral in terms of $\hat{\alpha}$ and $\hat{\beta}$ and to apply the Leibniz integral rule (Flanders, 1973). The outcome is combined with Theorem 4.3.5 to yield the desired sensitivities in category 2. In contrast to the CMHO case, however, the Leibniz integral rule cannot be applied directly to the integral equation (4.10). Due to the piecewise estimated state disturbance $\hat{\boldsymbol{w}}(t)$, the integrand $\boldsymbol{f}(\hat{\boldsymbol{\phi}}(t,\hat{\boldsymbol{p}},\boldsymbol{u}),\boldsymbol{u}(t))+\hat{\boldsymbol{w}}(t)$ is only continuous in $\hat{\mathcal{T}}_i$ but not in $\hat{\mathcal{T}}$. Consequently, the Leibniz integral rule can be applied only to

$$\hat{\boldsymbol{\phi}}_{i+1}(\hat{\boldsymbol{p}},\boldsymbol{u})=\hat{\boldsymbol{\phi}}_i(\hat{\boldsymbol{p}},\boldsymbol{u})+\int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}}\boldsymbol{f}(\hat{\boldsymbol{\phi}}(t,\hat{\boldsymbol{p}},\boldsymbol{u}),\boldsymbol{u}(t))\,dt+\hat{\boldsymbol{w}}_i,\quad i\in\overline{\mathcal{I}}. \tag{4.14}$$

The result is given in the following Lemma.

Lemma 4.3.6. *Two consecutive first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ and $\partial\hat{\boldsymbol{\phi}}_{i+1}/\partial\hat{\alpha}$ respective $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$ and $\partial\hat{\boldsymbol{\phi}}_{i+1}/\partial\hat{\beta}$ satisfy for $i\in\overline{\mathcal{I}}$*

$$\frac{\partial\hat{\boldsymbol{\phi}}_{i+1}}{\partial\hat{\alpha}}=\boldsymbol{X}_{i+1}^{i}\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\alpha}}+\bar{t}_{i+1}\left(\boldsymbol{f}_{i+1}+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right)-\bar{t}_i\boldsymbol{X}_{i+1}^{i}\left(\boldsymbol{f}_i+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right)-\frac{1}{\hat{\alpha}}{}^{i}\boldsymbol{Z}_{i+1}^{i}\hat{\boldsymbol{w}}_i \tag{4.15a}$$

$$\frac{\partial\hat{\boldsymbol{\phi}}_{i+1}}{\partial\hat{\beta}}=\boldsymbol{X}_{i+1}^{i}\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\beta}}+\left(\boldsymbol{f}_{i+1}+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right)-\boldsymbol{X}_{i+1}^{i}\left(\boldsymbol{f}_i+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right). \tag{4.15b}$$

Proof. Differentiating (4.14) with respect to $\hat{\alpha}$ by means of the Leibniz integral rule (Flanders, 1973) reveals

$$\frac{\partial\hat{\boldsymbol{\phi}}_{i+1}}{\partial\hat{\alpha}}=\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\alpha}}+\int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}}\underbrace{\frac{\partial}{\partial\hat{\alpha}}(\boldsymbol{f}+\hat{\boldsymbol{w}})}_{\triangleq\boldsymbol{a}}\,dt+\bar{t}_{i+1}\left(\boldsymbol{f}_{i+1}+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right)-\bar{t}_i\left(\boldsymbol{f}_i+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right). \tag{4.16}$$

Performing the derivation in the intermediate term $\boldsymbol{a}$ by repetitive application of the chain-rule and recalling the Assumption 9 about $\hat{\boldsymbol{w}}$ leads to

$$\boldsymbol{a}=\frac{\partial\boldsymbol{f}}{\partial\hat{\boldsymbol{\phi}}}\left(\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\alpha}}+\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\boldsymbol{w}}}\frac{\partial\hat{\boldsymbol{w}}}{\partial\hat{\alpha}}\right)+\frac{\partial\hat{\boldsymbol{w}}}{\partial\hat{\alpha}}=\frac{\partial\boldsymbol{f}}{\partial\hat{\boldsymbol{\phi}}}\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\boldsymbol{\phi}}}\Big|_i\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\alpha}}\Big|_i-\left(\frac{\partial\boldsymbol{f}}{\partial\hat{\boldsymbol{\phi}}}\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\boldsymbol{w}}}+\boldsymbol{I}\right)\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}^2\,T_i}.$$

Note that $\partial\hat{\boldsymbol{\phi}}/\partial\hat{\alpha}\big|_i$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ are not identical. The difference between both lies in the interchanged order of calculating the derivative and evaluating the expression at the corresponding time. The former term is

$$\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\alpha}}\Big|_i=\lim_{t\to\hat{\alpha}\,\bar{t}_i+\hat{\beta}}\frac{\partial}{\partial\hat{\alpha}}\left(\hat{\boldsymbol{\phi}}_i+\int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{t}\boldsymbol{f}\,dt\right)=\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\alpha}}-\bar{t}_i\left(\boldsymbol{f}_i+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right),$$

which is used for the intermediate term $\boldsymbol{a}$ to yield

$$\boldsymbol{a}=\frac{\partial\boldsymbol{f}}{\partial\hat{\boldsymbol{\phi}}}\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\boldsymbol{\phi}}}\Big|_i\left[\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\alpha}}-\bar{t}_i\left(\boldsymbol{f}_i+\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right)\right]-\left(\frac{\partial\boldsymbol{f}}{\partial\hat{\boldsymbol{\phi}}}\frac{\partial\hat{\boldsymbol{\phi}}}{\partial\hat{\boldsymbol{w}}}+\boldsymbol{I}\right)\frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}^2\,T_i}.$$

Noting that $\hat{\boldsymbol{\phi}}\big|_i = \hat{\boldsymbol{\phi}}_i$, the desired derivative (4.16) is obtained as

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{\phi}}_{i+1}}{\partial \hat{\alpha}} = \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} + \underbrace{\int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{\phi}}_i} dt}_{\triangleq \boldsymbol{B}} \left[\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\, T_i}\right)\right] \\ - \underbrace{\int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{w}}} + \boldsymbol{I} dt}_{\triangleq \boldsymbol{C}} \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}^2\, T_i} + \bar{t}_{i+1}\left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\, T_i}\right) - \bar{t}_i \left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\, T_i}\right),\end{aligned} \tag{4.17}$$

where the intermediate terms abbreviated with $\boldsymbol{B}$ and $\boldsymbol{C}$ have to be determined. The term $\boldsymbol{B}$ results from the integral equation of the corresponding first-order sensitivity matrix differential equation with respect to $\hat{\boldsymbol{\phi}}_i$ for $t \in \hat{\mathcal{T}}_i$, i. e.

$$\boldsymbol{B} = \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{\phi}}_i} dt = \boldsymbol{X}_{i+1}^i - \boldsymbol{I}.$$

For the determination of $\boldsymbol{C}$, the matrix differential equation derived by differentiating (3.23a) with respect to $\hat{\boldsymbol{w}}$ for $t \in \hat{\mathcal{T}}_i$

$$\frac{\partial \dot{\hat{\boldsymbol{\phi}}}}{\partial \hat{\boldsymbol{w}}} = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{w}}} + \boldsymbol{I}, \qquad \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{w}}}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}) = \boldsymbol{0},$$

is compared to the first-order sensitivity matrix differential equation (4.12a) multiplied by $\hat{\alpha}\, T_i$ on the same interval $\hat{\mathcal{T}}_i$

$$\hat{\alpha}\, T_i{}^i \dot{\boldsymbol{Z}} = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \hat{\alpha}\, T_i{}^i \boldsymbol{Z} + \boldsymbol{I}, \quad \hat{\alpha}\, T_i{}^i \boldsymbol{Z}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}) = \boldsymbol{0}.$$

Both matrix differential equations are identical if $\partial \hat{\boldsymbol{\phi}} / \partial \hat{\boldsymbol{w}} = \hat{\alpha}\, T_i{}^i \boldsymbol{Z}$ is chosen. This fact is exploited to calculate $\boldsymbol{C}$ which is the integral equation of these differential equations over $\hat{\mathcal{T}}_i$

$$\boldsymbol{C} = \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \frac{\partial \hat{\boldsymbol{\phi}}}{\partial \hat{\boldsymbol{w}}} + \boldsymbol{I}\, dt = \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}} \hat{\alpha}\, T_i{}^i \boldsymbol{Z} + \boldsymbol{I}\, dt = \hat{\alpha}\, T_i{}^i \boldsymbol{Z}_{i+1}^i - \boldsymbol{0}.$$

Substituting $\boldsymbol{B}$ and $\boldsymbol{C}$ in (4.17) concludes the proof of (4.15a). The proof of (4.15b) goes along the very same lines as the proof of (4.15a) and is therefore omitted. The major difference is that $\hat{\boldsymbol{w}}$ does not depend on $\hat{\beta}$, i. e. $\partial \hat{\boldsymbol{w}} / \partial \hat{\beta} = \boldsymbol{0}$, and thus the corresponding term $\boldsymbol{C}$ does not exist. □

Based on this result, the overall first-order state sensitivities of category 2 can be derived as follows.

Theorem 4.3.7. *The first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$ are for $i \in \underline{\mathcal{I}}$*

$$\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\alpha}} = \bar{t}_i\boldsymbol{f}_i - \bar{t}_{k-N}\boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1}\left(\bar{t}_{j+1}\boldsymbol{X}_i^{j+1} - \bar{t}_j\boldsymbol{X}_i^j\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j} - \frac{1}{\hat{\alpha}}\sum_{j=k-N}^{i-1}{}^j\boldsymbol{Z}_i^{k-N}\hat{\boldsymbol{w}}_j \tag{4.18a}$$

$$\frac{\partial\hat{\boldsymbol{\phi}}_i}{\partial\hat{\beta}} = \boldsymbol{f}_i - \boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1}\left(\boldsymbol{X}_i^{j+1} - \boldsymbol{X}_i^j\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j}. \tag{4.18b}$$

Proof. The proof is done by induction. Let $S(i)$ be the statement for $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ in the above Theorem. For $i = k-N$, equation (4.15a) of Lemma 4.3.6 leads together with $\partial\hat{\boldsymbol{\phi}}_{k-N}/\partial\hat{\alpha} = \boldsymbol{0}$ to

$$\begin{aligned}\frac{\partial\hat{\boldsymbol{\phi}}_{k-N+1}}{\partial\hat{\alpha}} = \boldsymbol{X}_{k-N+1}^{k-N}\boldsymbol{0} + \bar{t}_{k-N+1}\left(\boldsymbol{f}_{k-N+1} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{\alpha}\,T_{k-N}}\right) - \bar{t}_{k-N}\boldsymbol{X}_{k-N+1}^{k-N}\left(\boldsymbol{f}_{k-N} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{\alpha}\,T_{k-N}}\right)\\ - \frac{1}{\hat{\alpha}}{}^{k-N}\boldsymbol{Z}_{k-N+1}^{k-N}\hat{\boldsymbol{w}}_{k-N},\end{aligned}$$

which is equivalent to $S(k-N+1)$, i. e. $S(k-N+1)$ is true. This starts the induction. Now assume that $S(i)$ is true for some $i \in \underline{\mathcal{I}}$. Replacing this assumption with $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ in (4.15a) of Lemma 4.3.6 and applying the properties stated in Lemma 4.3.4 yields

$$\begin{aligned}\frac{\partial\hat{\boldsymbol{\phi}}_{i+1}}{\partial\hat{\alpha}} &= \bar{t}_i\boldsymbol{X}_{i+1}^i\boldsymbol{f}_i - \bar{t}_{k-N}\boldsymbol{X}_{i+1}^i\boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1}\left(\bar{t}_{j+1}\boldsymbol{X}_{i+1}^i\boldsymbol{X}_i^{j+1} - \bar{t}_j\boldsymbol{X}_{i+1}^i\boldsymbol{X}_i^j\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j}\\ &\quad - \frac{1}{\hat{\alpha}}\sum_{j=k-N}^{i-1}\boldsymbol{X}_{i+1}^i{}^j\boldsymbol{Z}_i^{k-N}\hat{\boldsymbol{w}}_j + \bar{t}_{i+1}\left(\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right) - \bar{t}_i\boldsymbol{X}_{i+1}^i\left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}\,T_i}\right) - \frac{1}{\hat{\alpha}}{}^i\boldsymbol{Z}_{i+1}^i\hat{\boldsymbol{w}}_i\\ &= \bar{t}_{i+1}\boldsymbol{f}_{i+1} - \bar{t}_{k-N}\boldsymbol{X}_{i+1}^{k-N}\boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i}\left(\bar{t}_{j+1}\boldsymbol{X}_{i+1}^{j+1} - \bar{t}_j\boldsymbol{X}_{i+1}^j\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j} - \frac{1}{\hat{\alpha}}\sum_{j=k-N}^{i}{}^j\boldsymbol{Z}_{i+1}^{k-N}\hat{\boldsymbol{w}}_j,\end{aligned}$$

which is $S(i+1)$, completing the inductive step. Thus, by the principle of induction, $S(i)$ and, accordingly, (4.18a) is true for all $i \in \underline{\mathcal{I}}$. The proof of (4.18b) goes along the very same lines as the proof of (4.18a) and is therefore omitted. □

The overall number of ODEs that have to solved for evaluating the gradient of the Lagrangian is $2n_x^2 + n_x$ since no additional ODEs have to be solved for the first-order state sensitivities $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$. Note that the difference between both results from the different derivative with respect to the corresponding clock parameter of the clock model in Assumption 8 and of the estimated state disturbance in Assumption 9. Moreover, if $\hat{\boldsymbol{w}}(t) = \boldsymbol{0}$, then Theorem 4.3.7 recovers the result of the CMHO derived in Theorem 4.2.3.

The following corollary summarizes the overall approach for calculating the first-order state sensitivities defined in (4.9) and is a direct consequence of the Theorems 4.3.5 and 4.3.7 combined with Lemma 4.3.4.

Corollary 4.3.8. *The first-order state sensitivities in (4.9) are for* $i \in \underline{\mathcal{I}}$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \boldsymbol{f}_{k-N} - \frac{1}{\hat{\alpha}} \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} {}_j\boldsymbol{Z}_{j+1}^{j} \hat{\boldsymbol{w}}_j + \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} \left(\bar{t}_{j+1} \boldsymbol{I} - \bar{t}_j \boldsymbol{X}_{j+1}^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j} \tag{4.19a}$$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}} = \boldsymbol{f}_i - \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} \left(\boldsymbol{I} - \boldsymbol{X}_{j+1}^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j} \tag{4.19b}$$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{k-N+1}^{k-N} \tag{4.19c}$$

$$\frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{w}}_j} = \begin{cases} \boldsymbol{0}, & i < j+1 \\ \boldsymbol{X}_i^{i-1} \boldsymbol{X}_{i-1}^{i-2} \dots \boldsymbol{X}_{j+2}^{j+1} {}_j\boldsymbol{Z}_{j+1}^{j}, & i \geq j+1 \end{cases}, \quad j \in \overline{\mathcal{I}} \tag{4.19d}$$

and for $i = k - N$ *all identical to* $\boldsymbol{0}$*, except* $\partial \hat{\boldsymbol{\phi}}_{k-N} / \partial \hat{\boldsymbol{x}}_{k-N}$ *which is the identity matrix* $\boldsymbol{I}$*.*

4.4 Hessian of the Lagrangian for the CMHO

The finite-difference method could once again be applied to approximate the Hessian of L. However, the computational complexity is even higher than in the gradient case. An alternative and more commonly used method is to estimate the Hessian by applying a quasi-Newton approximation which measures only the changes in the gradients, see Section 2.4. The drawback of this approximation method is that it does not explicitly consider the true structure of the Hessian which may result in significant deviations from the real one. As a consequence, the quality of the overall optimization algorithm can be poor, and many costly iteration steps are required to converge to a local optimum. For this reason, the following approach is proposed to improve the convergence speed and to reduce the computational load by exploiting the structure of the Hessian.

Applying the chain rule to (4.3) yields the Hessian of L stated in the following Lemma.

Lemma 4.4.1. *The exact Hessian of the Lagrangian* L *defined in (4.2) with respect to* $\hat{\boldsymbol{p}}$ *is*

$$\frac{\partial^2 L}{\partial \hat{\boldsymbol{p}}^2} = \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} - \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} {}_j\lambda_i \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 {}_jc_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} - \sum_{j=1}^{\dim(\boldsymbol{\mu})} {}_j\mu \frac{\partial^2 {}_jd}{\partial \hat{\boldsymbol{p}}^2} + \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} \left(\frac{\partial \Upsilon_i}{\partial {}_j\hat{\phi}_i} - \left(\boldsymbol{\lambda}_i^T \frac{\partial \boldsymbol{c}_i}{\partial {}_j\hat{\phi}_i} \right) \right) \frac{\partial^2 {}_j\hat{\phi}_i}{\partial \hat{\boldsymbol{p}}^2}, \tag{4.20}$$

where ${}_j\mu$*,* ${}_jd$*,* ${}_jc_i$*,* ${}_j\lambda_i$ *and* ${}_j\hat{\phi}_i$ *denote the* j*-th element of* $\boldsymbol{\mu}$*,* $\boldsymbol{d}$*,* $\boldsymbol{c}_i$*,* $\boldsymbol{\lambda}_i$ *and* $\hat{\boldsymbol{\phi}}_i$*, respectively, and where* $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\boldsymbol{p}}$ *and* $\partial^2 {}_j\hat{\phi}_i / \partial \hat{\boldsymbol{p}}^2$ *are the first-order and second-order state sensitivities, respectively.*

Proof. Equation (4.20) results from differentiating (4.3) with respect to $\hat{\boldsymbol{p}}$ where the dependence of $\hat{\boldsymbol{\phi}}$ on $\hat{\boldsymbol{p}}$ is taken into account by Lemma B.1. □

Based on this structure of the Hessian, we partition it into two parts

$$\frac{\partial^2 L}{\partial \hat{\boldsymbol{p}}^2} = \boldsymbol{H}_1 + \boldsymbol{H}_2, \tag{4.21a}$$

where $\boldsymbol{H}_1$ contains the second-order derivatives of the cost function and the terms involving the first-order state sensitivities

$$\boldsymbol{H}_1 = \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} - \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} {}_j\lambda_i \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 {}_j c_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} - \sum_{j=1}^{\dim(\boldsymbol{\mu})} {}_j\mu \frac{\partial^2 {}_j d}{\partial \hat{\boldsymbol{p}}^2} \tag{4.21b}$$

and where $\boldsymbol{H}_2$ contains the terms involving the second-order state sensitivities

$$\boldsymbol{H}_2 = \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} \left(\frac{\partial \Upsilon_i}{\partial {}_j\hat{\phi}_i} - \left(\boldsymbol{\lambda}_i^T \frac{\partial \boldsymbol{c}_i}{\partial {}_j\hat{\phi}_i} \right) \right) \frac{\partial^2 {}_j\hat{\phi}_i}{\partial \hat{\boldsymbol{p}}^2}. \tag{4.21c}$$

The first part $\boldsymbol{H}_1$ can be easily calculated due to the already available first-order sensitivities that were necessary for the computation of the gradient $\partial L / \partial \hat{\boldsymbol{p}}$. A crucial role in the computation of the second part $\boldsymbol{H}_2$ plays the second-order state sensitivities. For a discussion of this topic, we restrict ourselves to the conventional MHE scenario without the network. Then, three cases can be differentiated for $\boldsymbol{H}_2$ depending on the characteristic of the system dynamic:

① If $\partial^2 \boldsymbol{f} / \partial \hat{\boldsymbol{\phi}}^2 = \boldsymbol{0}$ holds, like for linear systems, then $\boldsymbol{H}_2 = \boldsymbol{0}$, because the second-order state sensitivities are exactly zero. This fact directly results from investigating the second-order sensitivity matrix differential equations.

② If the system has only "weak nonlinearities", then $\boldsymbol{H}_2 \approx \boldsymbol{0}$. Note that this approach shows strong similarities to the Gauss-Newton approach for solving nonlinear least-squares problems.

③ In the general case, neglecting $\boldsymbol{H}_2$ may result in a reduced performance of the optimization algorithm which motivates the inclusion of $\boldsymbol{H}_2$. Ideally, the exact second-order sensitivities should be calculated. However, this is not even reasonable for small-sized systems due to the high complexity involved for solving the n_x^3 ODEs for the second-order sensitivity matrix differential equations.

Based on these considerations and with a real-time application in mind, we do not further investigate the second-order sensitivities. Instead, we propose to approximate the Hessian of the Lagrangian L as follows.

Proposition 4.4.2. *The Hessian of the Lagrangian L defined in (4.1) with respect to $\hat{\boldsymbol{p}}$ can be approximated by*

$$\frac{\partial^2 L}{\partial \hat{\boldsymbol{p}}^2}[l], = \boldsymbol{H}_1[l] + \hat{\boldsymbol{H}}_2[l], \tag{4.22a}$$

where the argument $[l]$ denotes the l-th iteration step of the optimization. Thereby, the matrix $\boldsymbol{H}_1[l]$ is

$$\boldsymbol{H}_1[l] = \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T[l] \frac{\partial^2 \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i^2}[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}[l] - \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} {}_j\lambda_i[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T[l] \frac{\partial^2 {}_jc_i}{\partial \hat{\boldsymbol{\phi}}_i^2}[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}[l] - \sum_{j=1}^{\dim(\boldsymbol{\mu})} {}_j\mu[l] \frac{\partial^2 {}_jd}{\partial \hat{\boldsymbol{p}}^2}[l], \tag{4.22b}$$

where $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\boldsymbol{p}}$ are the first-order state sensitivities derived in Theorem 4.2.3. The matrix $\hat{\boldsymbol{H}}_2[l]$ is either neglected, i. e. $\hat{\boldsymbol{H}}_2[l] = \boldsymbol{0}$, or updated according to

$$\hat{\boldsymbol{H}}_2[l+1] = \hat{\boldsymbol{H}}_2[l] - \frac{\hat{\boldsymbol{H}}_2[l]\, \Delta\hat{\boldsymbol{p}}[l]\, \Delta\hat{\boldsymbol{p}}[l]^T\, \hat{\boldsymbol{H}}_2[l]}{\Delta\hat{\boldsymbol{p}}[l]^T\, \hat{\boldsymbol{H}}_2[l]\, \Delta\hat{\boldsymbol{p}}[l]} + \frac{\boldsymbol{z}[l]\, \boldsymbol{z}[l]^T}{\boldsymbol{z}[l]^T\, \Delta\hat{\boldsymbol{p}}[l]}, \tag{4.22c}$$

with $\Delta\hat{\boldsymbol{p}}[l] = \hat{\boldsymbol{p}}[l+1] - \hat{\boldsymbol{p}}[l]$, $\boldsymbol{z}[l] = (\partial L / \partial \hat{\boldsymbol{p}})[l+1] - (\partial L / \partial \hat{\boldsymbol{p}})[l] - \boldsymbol{H}_1[l+1] \Delta\hat{\boldsymbol{p}}[l]$ and $\hat{\boldsymbol{H}}_2[0] = \boldsymbol{0}$.

This approximation schema follows the strategy of calculating only the parts of the exact Hessian which can be derived without solving any additional ODEs, i. e. $\boldsymbol{H}_1$, and to approximate the rest, i. e. $\boldsymbol{H}_2$. This approximation is either $\boldsymbol{0}$ or a modified BFGS update. The decision which one to choose depends on the choice of the decreasing factor ξ_k in the decreasing condition (3.15) and on the nonlinearities of the optimization problem (3.18). For instance, a choice of ξ_k close to 1 results in only few iteration steps of the optimization algorithm. Consequently, $\boldsymbol{H}_2$ can be neglected since the influence of the modified BFGS update on the Hessian is noticeable only after a few iteration steps. Note that the proposed Hessian approximation schema is not restricted to the presented modified BFGS approach. Modifications, like damped BFGS or limited-memory BFGS, or other update schemes, like SR1 updating, can be used when deemed appropriate.

4.5 Hessian of the Lagrangian for the CMHE

The idea for calculating the Hessian of the Lagrangian (4.2) is identical to the CMHO case. This means that all what have been said for calculating the Hessian of the Lagrangian in the previous section is now also valid in the figurative sense. Thus, we only present the key aspects and refer the interested reader for details to the previous section.

Applying the chain rule to (4.8) yields the following exact Hessian of the Lagrangian (4.2).

Lemma 4.5.1. *The exact Hessian of the Lagrangian L defined in (4.2) with respect to $\hat{\boldsymbol{p}}$ is*

$$\begin{aligned}\frac{\partial^2 L}{\partial \hat{\boldsymbol{p}}^2} = \frac{\partial^2 \Gamma}{\partial \hat{\boldsymbol{p}}^2} + \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} + \sum_{i=k-N}^{k-1} \frac{\partial^2 \Psi_i}{\partial \hat{\boldsymbol{p}}^2} - \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} {}_j\lambda_i \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T \frac{\partial^2 {}_jc_i}{\partial \hat{\boldsymbol{\phi}}_i^2} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}} \\ - \sum_{j=1}^{\dim(\boldsymbol{\mu})} {}_j\mu \frac{\partial^2 {}_jd}{\partial \hat{\boldsymbol{p}}^2} + \sum_{i=k-N}^{k} \sum_{j=1}^{n} \left(\frac{\partial \Upsilon_i}{\partial {}_j\hat{\phi}_i} - \left(\boldsymbol{\lambda}_i^T \frac{\partial \boldsymbol{c}_i}{\partial {}_j\hat{\phi}_i} \right) \right) \frac{\partial^2 {}_j\hat{\phi}_i}{\partial \hat{\boldsymbol{p}}^2},\end{aligned} \tag{4.23}$$

where ${}_j\mu$, ${}_jd$, ${}_jc_i$, ${}_j\lambda_i$ and ${}_j\hat{\phi}_i$ denote the j-th element of $\boldsymbol{\mu}$, $\boldsymbol{d}$, $\boldsymbol{c}_i$, $\boldsymbol{\lambda}_i$ and $\hat{\boldsymbol{\phi}}_i$, respectively, and where $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\boldsymbol{p}}$ and $\partial^2 {}_j\hat{\phi}_i / \partial \hat{\boldsymbol{p}}^2$ are the first-order and second-order state sensitivities, respectively.

Proof. Equation (4.23) results from differentiating (4.8) with respect to $\hat{\boldsymbol{p}}$ where the dependence of $\hat{\boldsymbol{\phi}}$ on $\hat{\boldsymbol{p}}$ is taken into account by Lemma B.1. □

Based on this structure of the Hessian, we partition it again into an easy to compute part $\boldsymbol{H}_1$ and in an hard to compute part $\boldsymbol{H}_2$ which is approximated. The resulting approximation schema is identical to the one in Proposition 4.4.2 except that $\boldsymbol{H}_1$ contains now two more terms. For sake of completeness, we state the approximation schema in the following proposition.

Proposition 4.5.2. *The Hessian of the Lagrangian L defined in (4.2) with respect to $\hat{\boldsymbol{p}}$ can be approximated by*

$$\frac{\partial^2 L}{\partial \hat{\boldsymbol{p}}^2}[l], = \boldsymbol{H}_1[l] + \hat{\boldsymbol{H}}_2[l], \tag{4.24a}$$

where the argument $[l]$ denotes the l-th iteration step of the optimization. Thereby, the matrix $\boldsymbol{H}_1[l]$ is

$$\begin{aligned}\boldsymbol{H}_1[l] = \frac{\partial^2 \Gamma}{\partial \hat{\boldsymbol{p}}^2}[l] + \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T[l] \frac{\partial^2 \Upsilon_i}{\partial \hat{\boldsymbol{\phi}}_i^2}[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}[l] + \sum_{i=k-N}^{k-1} \frac{\partial^2 \Psi_i}{\partial \hat{\boldsymbol{p}}^2}[l] \\ - \sum_{i=k-N}^{k} \sum_{j=1}^{\dim(\boldsymbol{\lambda}_i)} {}_j\lambda_i[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}^T[l] \frac{\partial^2 {}_jc_i}{\partial \hat{\boldsymbol{\phi}}_i^2}[l] \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{p}}}[l] - \sum_{j=1}^{\dim(\boldsymbol{\mu})} {}_j\mu[l] \frac{\partial^2 {}_jd}{\partial \hat{\boldsymbol{p}}^2}[l],\end{aligned} \tag{4.24b}$$

where $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\boldsymbol{p}}$ are the first-order state sensitivities derived in Corollary 4.3.8. The matrix $\hat{\boldsymbol{H}}_2[l]$ is either neglected, i. e. $\hat{\boldsymbol{H}}_2[l] = \boldsymbol{0}$, or updated according to

$$\hat{\boldsymbol{H}}_2[l+1] = \hat{\boldsymbol{H}}_2[l] - \frac{\hat{\boldsymbol{H}}_2[l]\, \Delta\hat{\boldsymbol{p}}[l]\, \Delta\hat{\boldsymbol{p}}[l]^T\, \hat{\boldsymbol{H}}_2[l]}{\Delta\hat{\boldsymbol{p}}[l]^T\, \hat{\boldsymbol{H}}_2[l]\, \Delta\hat{\boldsymbol{p}}[l]} + \frac{\boldsymbol{z}[l]\, \boldsymbol{z}[l]^T}{\boldsymbol{z}[l]^T\, \Delta\hat{\boldsymbol{p}}[l]}, \tag{4.24c}$$

with $\Delta\hat{\boldsymbol{p}}[l] = \hat{\boldsymbol{p}}[l+1] - \hat{\boldsymbol{p}}[l]$, $\boldsymbol{z}[l] = (\partial L / \partial \hat{\boldsymbol{p}})[l+1] - (\partial L / \partial \hat{\boldsymbol{p}})[l] - \boldsymbol{H}_1[l+1] \Delta\hat{\boldsymbol{p}}[l]$ and $\hat{\boldsymbol{H}}_2[0] = \boldsymbol{0}$.

4.6 Numerical Case Study

In this section, we verify the efficiency of the presented derivative calculation strategies. To this end, we consider the centralized NCS architecture depicted in Figure 3.1 where the system Σ consists of a continuously stirred tank reactor (CSTR) (Hicks and Ray, 1971). The CSTR can be described by the nonlinear system (Uppal et al., 1974)

$$ {}_1\dot{x}(t) = {}_1a({}_2a - {}_1x(t)) - {}_3a\,{}_1x(t)\,e^{-\frac{{}_4a}{{}_2x(t)}} + {}_2w(t) \tag{4.25a} $$

$$ {}_2\dot{x}(t) = {}_1a({}_5a - {}_2x(t)) + {}_6a\,{}_1x(t)\,e^{-\frac{{}_4a}{{}_2x(t)}} + {}_7a(u(t) - {}_2x(t)) + {}_2w(t). \tag{4.25b} $$

The system involves two states $\boldsymbol{x}(t) = [{}_1x(t), {}_2x(t)]^T$ corresponding to the concentration and the temperature, respectively, the control input $u(t)$ corresponding to the cooling water temperature and two state disturbances $\boldsymbol{w}(t) = [{}_1w(t), {}_2w(t)]^T$. The initial condition is $\boldsymbol{x}(0) = [0.005, 445]^T$ and the model parameters are ${}_1a = 1$, ${}_2a = 0.02$, ${}_3a = 10^6$, ${}_4a = 5665$, ${}_5a = 340$, ${}_6a = 4.25 \cdot 10^9$ and ${}_7a = 2$. The sensor Σ_S shall be able to move freely in the surrounding reaction mixture and transmits the measured temperatures $y(\bar{t}_i)$ together with the corresponding time stamps $\bar{t}_i$ as packets $\mathcal{P}_i = \{y(\bar{t}_i), \bar{t}_i\}$ over the network Σ_N to the CMHO/CMHE. These measurements are described by the measurement model

$$ y(\bar{t}_i) = {}_2x(\bar{t}_i) + v(\bar{t}_i), \quad i \in \mathbb{N}, \tag{4.26} $$

where $v(\bar{t})$ is the zero-mean measurement disturbance with variance $R = 1$. To compare the results for different derivative calculation strategies and various settings for the CMHS, we provide an identical initial situation for all experiments. This is achieved by generating the measurement $y(\bar{t})$ from a simulated closed-loop feedback control scenario with direct access to the full state vector. More precisely, we use a two-degree-of-freedom control scheme consisting of a flatness-based feedforward controller (Fliess et al., 1995) and a linear quadratic state feedback controller (Kwakernaak and Sivan, 1972). The resulting trajectories are shown in Figure 4.1 where the covariance of the zero-mean state disturbance is $\boldsymbol{Q} = \mathrm{diag}(0.002^2, 250)$. To enable a meaningful comparison of different experiments, we set the clock parameters to $\alpha_k = 1$ and $\beta_k = 0$ and use an ideal network without packet delays nor packet drops. The cost functions of the CMHS are chosen as the nominal least-squares approaches defined in (3.14) respective (3.20). The CMHE weights are $\boldsymbol{P}_k^{-1} = \mathrm{diag}(10, 10, 10, 0.1)$, R^{-1} and $\boldsymbol{Q}^{-1}$. Furthermore, both CMHS should satisfy the inequalities $0 \leq {}_1\hat{x}(t) \leq 0.03$, $300 \leq {}_2\hat{x}(t) \leq 500$.

Figure 4.2 compares the complexity, i. e. the normalized calculation time, for computing the gradient of the Lagrangian L of the CMHO and the CMHE for the following three scenarios:

① CMHO: SM: sensitivity method according to Theorem 4.2.3,

② CMHE: SM: structure exploiting sensitivity method according to Corollary 4.3.8, and

③ CMHE: CDF: central-difference formula according to equation (2.86).

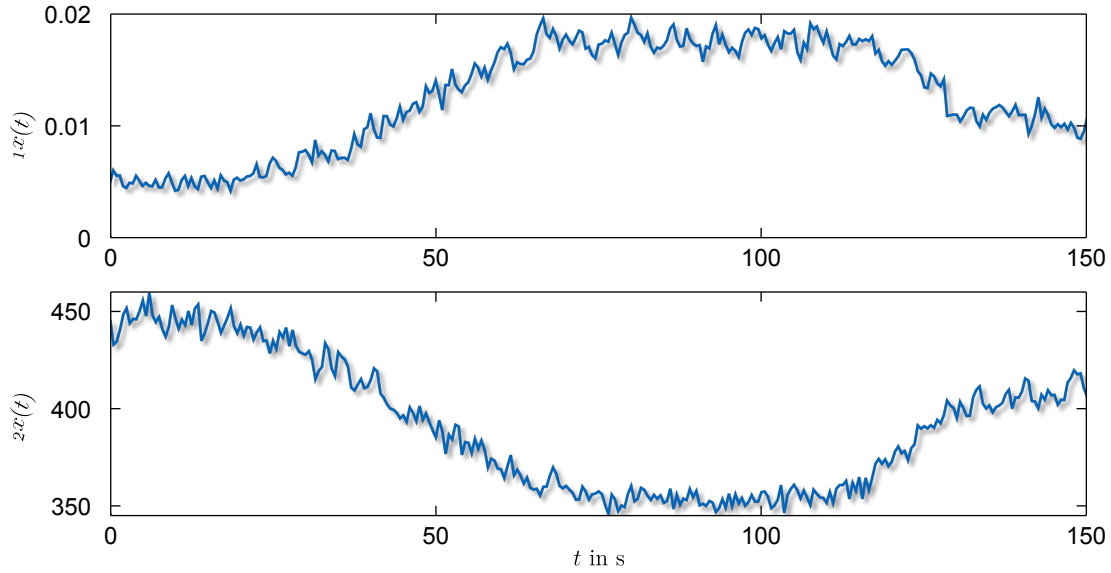


Figure 4.1: Trajectories of the CSTR resulting from a closed-loop feedback control scenario with direct access to the full state vector.

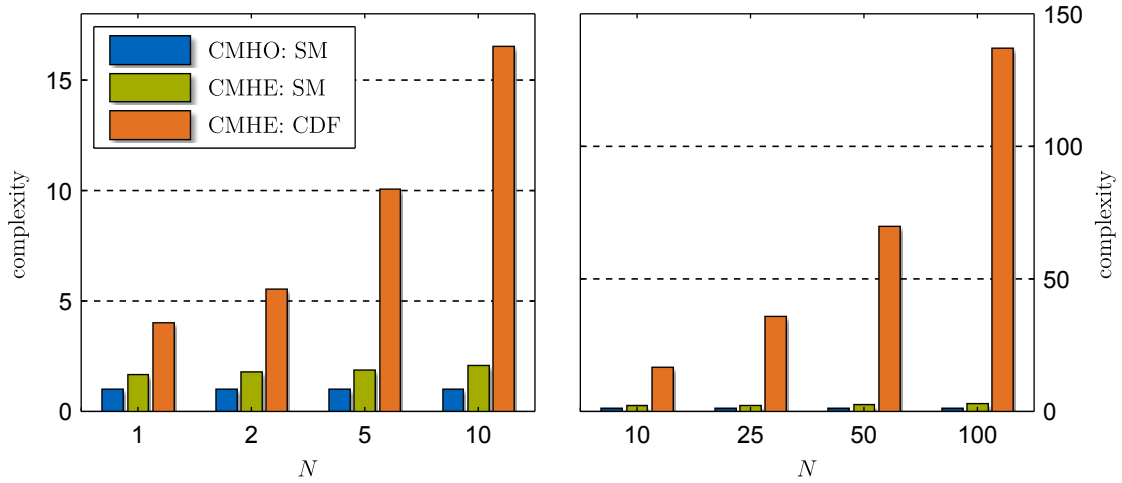


Figure 4.2: Comparison of the numerical complexity for different methods of computing $\partial L/\partial \hat{\boldsymbol{p}}$.

Thereby, we consider the constant time interval $\hat{\mathcal{T}} = [25\,\mathrm{s}, 125\,\mathrm{s}]$ and measure the time for calculating $\partial L/\partial \hat{\boldsymbol{p}}$ for various buffer sizes $N+1$. The sampling times in (4.26) are chosen as a function of the buffer size N to $\bar{t}_i = 25\,\mathrm{s} + \frac{i}{N}100\,\mathrm{s}$, $i = 0, \ldots, N$, $N = 1, 2, 5, 10, 25, 50, 100$. This means that the first and the last measurement are always taken at 25 s and 125 s, respectively, while all other sampling times are equally spaced in between. This enables in combination with the ideal network a reasonable comparison for different choices of N since $\hat{\mathcal{T}}$ is constant and independent of N. The time needed for the case CMHO: SM with $N = 1$ corresponds to the complexity 1. All calculations have been performed in `MATLAB(R2011a)` on an Intel Core 2 Duo E8400 CPU with 3 GHz. The ODE solver used in all simulations is a classical 4-step Runge-Kutta. The complexity of the case CMHO: SM is independent of N

and identical to 1 since the number of ODEs which have to be solved over $\hat{\mathcal{T}}$ is constantly 6. Although the number of ODEs which have to be solved over $\hat{\mathcal{T}}$ is constantly 10 for the scenario CMHE: SM, there is a slight increase in the complexity due to the increasing number of matrix multiplications for forming the derivatives in Corollary 4.3.8. It is important to note that even though the dimension of the optimization variable $\hat{\boldsymbol{p}}$ in the CMHE grows linearly with N, the complexity for calculating the gradient for the case CMHE: SM is almost kept constant. The linear growth of the $8N + 16$ ODEs which have to be solved over $\hat{\mathcal{T}}$ for the scenario CMHE: CDF is reflected in the linear growth in complexity.

The iteration numbers of the CMHE corresponding to the previous scenario for various values of N and three different Hessian approximation methods is shown in Table 4.1. The optimization has been performed with the `fmincon` function of `MATLAB`. The results provide preliminary evidence that the sensitivity method presented in Proposition 4.5.2 is more efficient than the conventional BFGS method. Moreover, in the considered scenario, the results justify the negligence of the second part of the true Hessian because its BFGS approximation yields almost the identical performance.

Table 4.2: Iteration numbers of the CMHE for three different Hessian approximation methods.

Hessian method		Iterations						
$\boldsymbol{H}_1$	$\boldsymbol{H}_2$	$N = 1$	$N = 2$	$N = 5$	$N = 10$	$N = 25$	$N = 50$	$N = 100$
BFGS of $\boldsymbol{H}_1 + \boldsymbol{H}_2$		9	11	12	14	15	13	14
SM	-	4	5	6	9	7	7	8
SM	BFGS	4	5	5	8	7	6	7

4.7 Summary

In this chapter, we have presented an efficient, parallelizable and sensitivity-based method to calculate the gradient and Hessian of the Lagrangian to the optimization problem of the CMHO and CMHE. The gradient computation method has been derived by applying the chain rule to the Lagrangian and depends on the respective first-order state sensitivities. These sensitivities have been calculated based on the solution of first-order sensitivity matrix differential equations as an answer to the difficulties stemming from the special nature of the optimization problem formulation where the clock parameters arise in the integral bounds. This perception facilitates the accomplishment of exploiting the structure of the state sensitivities such that the number of ODEs required for calculating the gradient in the CMHE case is independent of the number of optimization variables and equal to the adjoint case, namely $2n_x^2 + n_x$. The Hessian computation method is founded on the idea of partitioning the exact Hessian into two parts. The first part can be readily calculated due to the already available first-order state sensitivities while the second part can either be approximated by

a modified BFGS method or completely neglected. In contrast to the adjoint method, the proposed sensitivity-based method provides an excellent approximation of the Hessian for arbitrary cost functions almost for free. The theoretical benefits have been substantiated by a numerical case study of a continuously-stirred tank reactor where the proposed method has been compared to the finite-difference approach. Finally, it should be noted that the proposed derivative calculation methods also hold for conventional moving horizon estimators, cf. Philipp (2011a).

Chapter 5

Observability of Networked Control Systems

The update step of the centralized moving horizon observer (CMHO) and the centralized moving horizon estimator (CMHE) presented in Chapter 3 answer the question *how* the information stored in the buffer can be extracted to determine the parameter $\hat{\boldsymbol{p}}_k$. In this chapter, we answer the question *when* this procedure is feasible, i. e. *when* it is possible to determine uniquely the parameter $\hat{\boldsymbol{p}}_k$. In other words, we investigate the observability of undisturbed and disturbed NCSs.

The basic idea behind the observability definitions for discrete-time nonlinear systems presented in Section 2.2 is to guarantee the well-posedness of the nonlinear observation map. This perception will act as the model for the observability notion of NCSs. Even though every observability definition presented in Section 2.2 for discrete-time nonlinear systems can be extended to the networked scenario, not every observability definition is equally well-suited. This is due to the fact that the observation map for NCSs depends not only on the state but additionally on the clock parameters. For combined state and parameter estimation problems, the possibility of reconstructing the parameter often depends on the information content in the input data. This is typically formulated as a condition for persistently or sufficiently exciting input data appearing, inter alia, in identification (Ljung, 1999), estimation (Kreisselmeier, 1977) and adaptive control (Krstic et al., 1995).

Based on these considerations, the contribution of this chapter is three-fold. First, we present the observation map for undisturbed and disturbed NCSs which is formulated based on the buffer $\mathcal{B}_k$. Second, we define observability of undisturbed and disturbed NCSs based on the full-rankness of a certain derivative of the respective observation map. The resulting observability notion guarantees (at least local) well-posedness of the respective observation map and is key for the stability analysis presented in Chapter 6. Third, we derive for the undisturbed as well as for the disturbed NCSs a necessary condition for the control input to be sufficiently exciting such that all unknown parameters can be estimated. This is achieved by establishing a relation between the first-order state sensitivities (cf. Chapter 4) appearing in the elements of the derivative of the observation map and the control input.

The remainder of this chapter is organized as follows. In Section 5.1, we state the problem formulation. The observation map for NCSs, the observability definitions and the necessary condition for the control input to be sufficiently exciting are derived for the undisturbed and the disturbed NCSs in Section 5.2 and 5.3, respectively. Section 5.4 reveals the relation between the presented notion of observability to other quantities. The introduced terminology and the derived conditions are clarified by several examples in Section 5.5 before the Chapter is ended with a summary in Section 5.6.

5.1 Problem Formulation

The problem formulations are stated as follows.

Problem 5.1 (Observability of undisturbed NCSs)**.** *Let the buffer $\mathcal{B}_k$ and the corresponding control input $\boldsymbol{u}(t) \in \mathbb{U}$ be given for the undisturbed centralized NCS architecture described in Section 3.1.*

The problem is to answer the question when it is possible for the CMHO to determine uniquely the parameter $\hat{\boldsymbol{p}}_k$ defined in (3.16)?

Problem 5.2 (Observability of disturbed NCSs)**.** *Let the buffer $\mathcal{B}_k$ and the corresponding control input $\boldsymbol{u}(t) \in \mathbb{U}$ be given for the disturbed centralized NCS architecture described in Section 3.1.*

The problem is to answer the question when it is possible for the CMHE to determine uniquely the parameter $\hat{\boldsymbol{p}}_k$ defined in (3.22)?

It is important to recall that throughout Part II of this thesis including the current chapter, the *Assumptions 1-9 are supposed to hold without explicitly stating them.*

5.2 Observability of Undisturbed NCSs

5.2.1 Definitions

Following the lines of the observability notion presented in Section 2.2, we define a $N+1$-length observation map for NCSs associated to the buffer $\mathcal{B}_k$

$$\underline{\boldsymbol{h}}_k(\hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{h}(\hat{\boldsymbol{x}}_{k-N|k}) \\ \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{k-N+1|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \\ \vdots \\ \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{k|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \end{bmatrix}, \tag{5.1}$$

where $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ is defined in Definition 3.4.2. Note that in the undisturbed scenario, the mappings $\boldsymbol{\phi}(\cdot)$ and $\hat{\boldsymbol{\phi}}(\cdot)$ are identical. This means that $\underline{\boldsymbol{y}}_k = \mathrm{col}(\boldsymbol{y}_{i|k}, i \in \mathcal{I}_k)$ is in the image of $\mathbb{P}$ under $\underline{\boldsymbol{h}}_k$, i. e. $\underline{\boldsymbol{y}}_k \in \underline{\boldsymbol{h}}_k(\mathbb{P})$, $k \in \mathbb{N}_{N+1}$. Consequently, we formulate observability in

the context of undisturbed NCSs in terms of full-rankness of the Jacobian of the observation map (5.1). The idea behind is that $\underline{\boldsymbol{h}}_k(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ is at least locally injective. The precise observability definition reads as follows.

Definition 5.2.1. *The undisturbed NCS described in Section 3.1 is said to satisfy the extended observability rank condition in $N+1$ steps if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$, $\forall \hat{\boldsymbol{p}}_k \in \mathbb{P}$, $\forall k \in \mathbb{N}_{N+1}$,*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k} \right|_{\hat{\boldsymbol{p}}_k, \boldsymbol{u}} = n_x + 2. \tag{5.2}$$

This definition takes account for the fact that in contrast to state estimation of linear systems, the combined state and parameter estimation of nonlinear systems depends on the information content in the input data. This dependence is reflected in the following definition as a property of the control input $\boldsymbol{u}(t)$.

Definition 5.2.2. *The control input $\boldsymbol{u}(t) \in \mathbb{U}$ of the undisturbed NCS described in Section 3.1 is said to be extended $N+1$-exciting at the sampling instance k if*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k} \right|_{\hat{\boldsymbol{p}}_k, \boldsymbol{u}} = n_x + 2. \tag{5.3}$$

The word *extended* in Definition 5.2.1 and 5.2.2 indicates the fact that both Definitions are extensions to the corresponding network-free scenario where the output $\boldsymbol{y}(t)$ of a nonlinear continuous-time system is sampled non-uniformly. In this case, Definition 5.2.1 and 5.2.2 reduces to the following Definition 5.2.3 and 5.2.4, respectively.

Definition 5.2.3. *The undisturbed system described in Section 3.1 is said to satisfy the observability rank condition in $N+1$ steps if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$, $\forall \hat{\boldsymbol{x}}_{k-N|k} \in \mathbb{X}$, $\forall k \in \mathbb{N}_{N+1}$,*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N|k}} \right|_{\hat{\boldsymbol{x}}_{k-N|k}, \boldsymbol{u}} = n_x. \tag{5.4}$$

Definition 5.2.4. *The control input $\boldsymbol{u}(t) \in \mathbb{U}$ of the undisturbed system described in Section 3.1 is said to be $N+1$-exciting at the sampling instance k if*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N|k}} \right|_{\hat{\boldsymbol{x}}_{k-N|k}, \boldsymbol{u}} = n_x. \tag{5.5}$$

Definition 5.2.3 is, in turn, an extension of Definition 2.2.2 to the non-uniformly sampled output scenario. Suppose that the output $\boldsymbol{y}(t)$ of a nonlinear continuous-time system is uniformly sampled and the nonlinear continuous-time system itself is discretized with zero-order-hold for the control input and the state disturbance. Then the sampling instance k becomes immaterial for the observability investigation and Definition 5.2.3 coincides with Definition 2.2.2. Moreover, note that Definition 5.2.1 and 5.2.2 implies Definition 5.2.3 and 5.2.4, respectively.

5.2.2 A Necessary Condition for Extended N+1 Excitation

The goal of this subsection is to derive a necessary condition for the control input $\boldsymbol{u}(t)$ to be extended $N+1$-exciting for undisturbed NCSs at the sampling instance k. Since we will consider only a fixed buffer $\mathcal{B}_k$, we ease the presentation by omitting the subscripts indicating the buffer affiliation. Moreover, we suppress the arguments of all functions from the notation when the meaning is clear.

By applying the chain rule, the Jacobian of the observation map (5.1) can be written as

$$\frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{p}}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\alpha}} & \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\beta}} & \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\boldsymbol{x}}_{k-N}} \\ \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\alpha}} & \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\beta}} & \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\boldsymbol{x}}_{k-N}} \end{bmatrix}, \tag{5.6}$$

where the derivatives $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\alpha}$, $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\beta}$ and $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\boldsymbol{x}}_{k-N}$ are for $i \in \mathcal{I}$ the already known first-order state sensitivities, cf. Chapter 4.2. Recall that, in general, there does not exist a closed-form expression for these sensitivities due to the absence of an analytical expression for $\hat{\boldsymbol{\phi}}_i$. Nevertheless, the knowledge about the first-order sensitivities derived in Theorem 4.2.3 can be applied to (5.6). The elements of the resulting Jacobian are for $i \in \mathcal{I}$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} \right) \tag{5.7a}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} \right) \tag{5.7b}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \boldsymbol{X}_i^{k-N}. \tag{5.7c}$$

The next step is to establish a relation between the expression for the first-order state sensitivities $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\beta}$ and the control input. This is achieved by the following Lemma.

Lemma 5.2.1. *The following equivalence holds* $\forall i \in \mathcal{I}$, $\forall \hat{\boldsymbol{p}} \in \mathbb{P}$ *and* $\forall \boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$

$$\boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} = \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}, \tau) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\, \dot{\boldsymbol{u}}(\tau)\, d\tau.$$

Proof. Since $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$, differentiating (3.17) with respect to t leads to the second-order differential equation

$$\ddot{\hat{\boldsymbol{\phi}}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, \dot{\hat{\boldsymbol{\phi}}}(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t)\, \dot{\boldsymbol{u}}(t),$$

with the initial values $\hat{\boldsymbol{\phi}}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) = \hat{\boldsymbol{x}}_{k-N}$ and $\dot{\hat{\boldsymbol{\phi}}}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) = \boldsymbol{f}_{k-N}$. By introducing new coordinates $\boldsymbol{\zeta}(t) \triangleq \dot{\hat{\boldsymbol{\phi}}}(t)$ and $\dot{\boldsymbol{\zeta}}(t) \triangleq \ddot{\hat{\boldsymbol{\phi}}}(t)$, the second-order differential equation can be

rewritten as the following linear time-varying first-order differential equation

$$\dot{\boldsymbol{\zeta}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\,\boldsymbol{\zeta}(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t)\,\dot{\boldsymbol{u}}(t),$$

with the initial value $\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) = \boldsymbol{f}_{k-N}$. The solution $\boldsymbol{\zeta}(t)$ is given by the superposition of the homogeneous solution $\boldsymbol{\zeta}_h(t)$ and the particular solution $\boldsymbol{\zeta}_p(t)$ (see Kailath (1980, pg. 595-601) or Ludyk (1995, pg. 40-55))

$$\begin{aligned}\boldsymbol{\zeta}(t) &= \boldsymbol{\zeta}_h(t) + \boldsymbol{\zeta}_p(t)\\ &= \boldsymbol{\Phi}(t, \hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta})\,\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}) + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{t} \boldsymbol{\Phi}(t,\tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau,\end{aligned}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the time-varying transition matrix. Consequently, the overall solution evaluated at the sampling times $\hat{\alpha}\,\bar{t}_i + \hat{\beta}$ becomes

$$\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}) = \boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N} + \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}, \tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau, \quad \forall i \in \mathcal{I}.$$

Noting that $\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}) = \boldsymbol{f}_i$ and subtracting $\boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N}$ on both sides yields the equivalence stated in Lemma 5.2.1 which concludes the prove. □

The equivalence stated in Lemma 5.2.1 provides the basis for deriving necessary and sufficient conditions for the elements of the first and second column of (5.6) to be non-vanishing.

Lemma 5.2.2. *The i-th element of the second column of (5.6), i. e. $\partial \boldsymbol{h}_i/\partial\hat{\beta}$, is non-vanishing if and only if $\exists\,\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$,*

$$\int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i}\,\boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}, \tau)\,\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\,\dot{\boldsymbol{u}}(\tau)\,d\tau \neq \boldsymbol{0}. \tag{5.8}$$

Proof. Equation (5.8) directly results from applying Lemma 5.2.1 to the elements (5.7b) of the Jacobian of the observation map (5.1). □

The terms in front of $\dot{\boldsymbol{u}}(t)$ in Lemma 5.2.2 describe how the temporal change of $\boldsymbol{u}(t)$, i. e. $\dot{\boldsymbol{u}}(t)$, is mapped to the output $\partial \boldsymbol{h}_i/\partial\hat{\beta}$. The terms $\partial \boldsymbol{f}/\partial \boldsymbol{u}$, $\boldsymbol{\Phi}$ and $\partial \boldsymbol{h}_i/\partial\hat{\boldsymbol{\phi}}_i$ can be interpreted as maps from the input space to the state space, within the state space reflecting the system dynamics and from the state space to the output space, respectively. For instance, if we consider the continuous-time linear system $\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t)$ with the linear measurement model $\hat{\boldsymbol{y}}(\bar{t}_i) = \boldsymbol{C}\boldsymbol{x}(\bar{t}_i)$, then $\partial \boldsymbol{f}/\partial \boldsymbol{u} = \boldsymbol{B}$, $\boldsymbol{\Phi} = e^{\boldsymbol{A}(\hat{\alpha}\,\bar{t}_i+\hat{\beta}-\tau)}$ and $\partial \boldsymbol{h}_i/\partial\hat{\boldsymbol{\phi}}_i = \boldsymbol{C}$. Consequently, $\partial \boldsymbol{h}_i/\partial\hat{\beta}$ is only vanishing if the integral of the mapped temporal change of $\boldsymbol{u}(t)$ over the time interval $t \in [\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}, \hat{\alpha}\,\bar{t}_i + \hat{\beta}]$ is vanishing. For instance, this is the case if there does not exist a temporal change of $\boldsymbol{u}(t)$.

Lemma 5.2.3. *The i-th element of the first column of (5.6), i. e. $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$, is non-vanishing if and only if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$,*

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} \neq \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} \left(\left(\frac{\bar{t}_{k-N}}{\bar{t}_i} - 1 \right) \boldsymbol{f}_{k-N} \right). \tag{5.9}$$

Proof. After recalling the first-order state sensitivities derived in Theorem 4.2.3, the derivative $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$ can be rearranged in the following way

$$\begin{aligned} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\alpha}} &= \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} \right) \\ &= \left(\bar{t}_i - \bar{t}_{k-N} \right) \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \bar{t}_i \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} \right) \\ &= \left(\bar{t}_i - \bar{t}_{k-N} \right) \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} \boldsymbol{f}_{k-N} + \bar{t}_i \frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}}. \end{aligned}$$

The requirement for this expression to be non-vanishing leads to the desired condition (5.9) which concludes the proof. □

Lemma 5.2.3 connects the three derivatives $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$, $\partial \boldsymbol{h}_i / \partial \hat{\beta}$ and $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{x}}_{k-N}$. If $\partial \boldsymbol{h}_i / \partial \hat{\beta}$ cannot be generated by a certain linear combination of the columns of $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{x}}_{k-N}$, then $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$ is non-vanishing. This Lemma also includes some interesting special cases. If, on the one hand, the temporal change of $\boldsymbol{u}(t)$ is vanishing for $t \in [\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}, \hat{\alpha}\, \bar{t}_i + \hat{\beta}]$, then $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$ is non-vanishing if and only if the vector field $\boldsymbol{f}_{k-N}$ satisfies $(\bar{t}_{k-N} / \bar{t}_i - 1) \boldsymbol{f}_{k-N} \notin \ker(\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{x}}_{k-N})$. A counterexample would be if the system remains in an equilibrium in the interval $t \in [\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}, \hat{\alpha}\, \bar{t}_i + \hat{\beta}]$. If, on the other hand, the temporal change of $\boldsymbol{u}(t)$ is non-vanishing for $t \in [\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}, \hat{\alpha}\, \bar{t}_i + \hat{\beta}]$, then $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$ is non-vanishing if the vector field $\boldsymbol{f}_{k-N}$ satisfies $\boldsymbol{f}_{k-N} = \boldsymbol{0}$.

Based on Lemma 5.2.2 and 5.2.3, we derive in the following Theorem a necessary condition for $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$ to be extended $N+1$-exciting at the sampling instance k.

Theorem 5.2.4. *A necessary condition for the control input $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$ of the undisturbed NCS described in Section 3.1 to be extended $N+1$-exciting at the sampling instance k is that the following conditions (C0), (C1), (C2) and (C3) are fulfilled:*

(C0) $N + 1 \geq \dfrac{n_x + 2}{n_y}$.

(C1) $\boldsymbol{u}(t)$ *is* $N+1$*-exciting at the sampling instance* k.

(C2) $\exists\, j \in \mathcal{I}, \quad \displaystyle\int_{\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}}^{\hat{\alpha}\, \bar{t}_j + \hat{\beta}} \frac{\partial \boldsymbol{h}_j}{\partial \hat{\boldsymbol{\phi}}_j} \boldsymbol{\Phi}(\hat{\alpha}\, \bar{t}_j + \hat{\beta}, \tau) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\, \dot{\boldsymbol{u}}(\tau)\, d\tau \neq \boldsymbol{0}.$

(C3) $\exists\, i \in \mathcal{I}, \quad \dfrac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} \neq \dfrac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} \left(\left(\dfrac{\bar{t}_{k-N}}{\bar{t}_i} - 1 \right) \boldsymbol{f}_{k-N} \right).$

Proof. The condition (C0) ensures that (5.6) has at least $n_x + 2$ rows. The condition (C1) guarantees that the last n_x columns of (5.6) have full column rank. The conditions (C2) and (C3) make sure according to the Lemmas 5.2.2 and 5.2.3 that the first and second column of (5.6) are not $\mathbf{0}$, respectively. All conditions together are a necessary condition for the full rankness of (5.6). □

5.3 Observability of Disturbed Networked Control Systems

5.3.1 Definitions

As in the case of undisturbed NCSs, we define a $N+1$-length observation map associated to the buffer $\mathcal{B}_k$

$$\underline{\boldsymbol{h}}_k(\hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \begin{bmatrix} \boldsymbol{h}(\hat{\boldsymbol{x}}_{k-N|k}) \\ \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{k-N+1|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \\ \vdots \\ \boldsymbol{h}(\hat{\boldsymbol{\phi}}_{k|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) \end{bmatrix}, \tag{5.10}$$

where $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ is defined in Definition 3.5.2. In contrast to undisturbed NCSs, disturbed NCSs are characterized, as the name suggests, by the presence of disturbances. This fact is expressed without limitation in Definition 3.5.2 of $\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})$ and in (3.22) defining $\hat{\boldsymbol{p}}_k$. As a consequence, we cannot equate observability of disturbed NCSs with the requirement that the rank of the Jacobian of the observation map (5.10) is identical with the dimension of $\hat{\boldsymbol{p}}_k$ since the dimension of $\underline{\hat{\boldsymbol{y}}}_k = \operatorname{col}(\hat{\boldsymbol{y}}_{i|k}, i \in \mathcal{I}_k)$ is smaller than the dimension of $\hat{\boldsymbol{p}}_k$. This means that the observation map (5.10) is not injective which becomes clear based on the following consideration. Suppose that $\underline{\hat{\boldsymbol{y}}}_k$ is fixed and generated without measurement disturbances, i. e. $\hat{\boldsymbol{v}}_{i|k} = \mathbf{0}$. Then for every $\operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})$, there exists at least one disturbance sequence $\{\hat{\boldsymbol{w}}_{i|k}\}_{i=k-N}^{k-1}$ such that $\underline{\boldsymbol{h}}_k(\hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \underline{\hat{\boldsymbol{y}}}_k$ holds. Consequently, we formulate observability in the context of disturbed NCSs in terms of full-rankness of the partial derivative of the observation map (5.10) with respect to $\operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})$. The idea behind is that there exists for a fixed $\underline{\hat{\boldsymbol{y}}}_k$ which is generated without measurement disturbances, i. e. $\hat{\boldsymbol{v}}_{i|k} = \mathbf{0}$, and a fixed $\operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})$, at least locally, only one disturbance sequence $\{\hat{\boldsymbol{w}}_{i|k}\}_{i=k-N}^{k-1}$ such that $\underline{\boldsymbol{h}}_k(\hat{\boldsymbol{p}}_k, \boldsymbol{u}) = \underline{\hat{\boldsymbol{y}}}_k$ holds. The resulting observability definition reads as follows.

Definition 5.3.1. *The disturbed NCS described in Section 3.1 is said to satisfy the extended observability rank condition in $N+1$ steps if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$, $\forall \hat{\boldsymbol{p}}_k \in \mathbb{P}$, $\forall k \in \mathbb{N}_{N+1}$,*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})} \right|_{\hat{\boldsymbol{p}}_k, \boldsymbol{u}} = n_x + 2. \tag{5.11}$$

As for undisturbed NCSs, the dependence of the combined state and parameter estimation problem on the information content in the input data is reflected in the following definition as a property of the control input $\boldsymbol{u}(t)$.

Definition 5.3.2. *The control input $\boldsymbol{u}(t) \in \mathbb{U}$ of the disturbed NCS described in Section 3.1 is said to be extended $N+1$-exciting at the sampling instance k if*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})} \right|_{\hat{\boldsymbol{p}}_k, \boldsymbol{u}} = n_x + 2. \tag{5.12}$$

Similar to undisturbed NCSs, the word "extended" in Definition 5.3.1 and 5.3.2 indicates the fact that both Definitions are extensions to the corresponding network-free scenario where the output $\hat{\boldsymbol{y}}(t)$ of a nonlinear continuous-time system is sampled non-uniformly. In this case, Definition 5.3.1 and 5.3.2 reduces to the following Definition 5.3.3 and 5.3.4, respectively.

Definition 5.3.3. *The disturbed system described in Section 3.1 is said to satisfy the observability rank condition in $N+1$ steps if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$, $\forall \hat{\boldsymbol{x}}_{k-N|k} \in \mathbb{X}$, $\hat{\boldsymbol{w}}_{i|k} \in \mathbb{W}$, $\forall i \in \overline{\mathcal{I}}_k$, $\forall k \in \mathbb{N}_{N+1}$,*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N|k}} \right|_{\hat{\boldsymbol{x}}_{k-N|k}, \hat{\boldsymbol{w}}_{k-N|k}, \ldots, \hat{\boldsymbol{w}}_{k-1|k}, \boldsymbol{u}} = n_x. \tag{5.13}$$

Definition 5.3.4. *The control input $\boldsymbol{u}(t) \in \mathbb{U}$ of the disturbed system described in Section 3.1 is said to be $N+1$-exciting at the sampling instance k if*

$$\operatorname{rank} \left. \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N|k}} \right|_{\hat{\boldsymbol{x}}_{k-N|k}, \hat{\boldsymbol{w}}_{k-N|k}, \ldots, \hat{\boldsymbol{w}}_{k-1|k}, \boldsymbol{u}} = n_x. \tag{5.14}$$

Note that Definition 5.3.1 and 5.3.2 implies Definition 5.3.3 and 5.3.4, respectively.

5.3.2 A Necessary Condition for Extended N+1 Excitation

The goal of this subsection is to derive a necessary condition for the control input $\boldsymbol{u}(t)$ of the disturbed NCSs to be extended $N+1$-exciting at the sampling instance k. Since we will consider only a fixed buffer $\mathcal{B}_k$, we ease the presentation by omitting the subscripts indicating the buffer affiliation. Moreover, we suppress the arguments of all functions from the notation when the meaning is clear.

The procedure is identical to the case of undisturbed NCSs. This means, we derive the partial derivative of the observation map (5.10) with respect to $\operatorname{col}(\hat{\alpha}, \hat{\beta}, \hat{\boldsymbol{x}}_{k-N})$ by applying

the chain rule

$$\frac{\partial \underline{\boldsymbol{h}}}{\partial \operatorname{col}(\hat{\alpha}, \hat{\beta}, \hat{\boldsymbol{x}}_{k-N})} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\alpha}} & \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\beta}} & \frac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \frac{\partial \hat{\boldsymbol{\phi}}_{k-N}}{\partial \hat{\boldsymbol{x}}_{k-N}} \\ \vdots & \vdots & \vdots \\ \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\alpha}} & \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\beta}} & \frac{\partial \boldsymbol{h}_{k}}{\partial \hat{\boldsymbol{\phi}}_{k}} \frac{\partial \hat{\boldsymbol{\phi}}_{k}}{\partial \hat{\boldsymbol{x}}_{k-N}} \end{bmatrix}, \tag{5.15}$$

where the derivatives $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\alpha}$, $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$ and $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\boldsymbol{x}}_{k-N}$ are for $i \in \mathcal{I}$ the already known first-order state sensitivities, cf. Chapter 4.3. By applying Theorem 4.3.5 and 4.3.7, the elements of (5.15) are for $i \in \mathcal{I}$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\bar{t}_i \boldsymbol{f}_i - \bar{t}_{k-N} \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\bar{t}_{j+1} \boldsymbol{X}_i^{j+1} - \bar{t}_j \boldsymbol{X}_i^j \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j} - \frac{1}{\hat{\alpha}} \sum_{j=k-N}^{i-1} {}^{j}\boldsymbol{Z}_i^{k-N} \hat{\boldsymbol{w}}_j \right) \tag{5.16a}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left(\boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\boldsymbol{X}_i^{j+1} - \boldsymbol{X}_i^j \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j} \right) \tag{5.16b}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \boldsymbol{X}_i^{k-N}. \tag{5.16c}$$

The next step is to establish a relation between the expression for the first-order state sensitivity $\partial\hat{\boldsymbol{\phi}}_i/\partial\hat{\beta}$ and the control input. This is achieved by the following Lemma.

Lemma 5.3.1. *The following equivalence holds* $\forall i \in \underline{\mathcal{I}}$, $\forall \hat{\boldsymbol{p}} \in \mathbb{P}$ *and* $\forall \boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$

$$\boldsymbol{f}_i - \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i-1} \left(\boldsymbol{X}_i^{j+1} - \boldsymbol{X}_i^j \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j} = \int_{\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}}^{\hat{\alpha}\, \bar{t}_i + \hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\, \bar{t}_i + \hat{\beta}, \tau) \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\, \dot{\boldsymbol{u}}(\tau)\, d\tau. \tag{5.17}$$

Proof. Since $\hat{\boldsymbol{w}}(t)$ is only continuous in $t \in \hat{\mathcal{T}}_i$, $i \in \underline{\mathcal{I}}$, and not in $t \in \hat{\mathcal{T}}$ and $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$, we differentiate (3.23) with respect to t on the intervals $\hat{\mathcal{T}}_i$, $i \in \underline{\mathcal{I}}$, which leads for all $i \in \overline{\mathcal{I}}$ to the second-order differential equation

$$\ddot{\hat{\boldsymbol{\phi}}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, \dot{\hat{\boldsymbol{\phi}}}(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t)\, \dot{\boldsymbol{u}}(t), \tag{5.18}$$

with the initial values $\hat{\boldsymbol{\phi}}(\hat{\alpha}\, \bar{t}_i + \hat{\beta}) = \hat{\boldsymbol{x}}_i$ and $\dot{\hat{\boldsymbol{\phi}}}(\hat{\alpha}\, \bar{t}_i + \hat{\beta}) = \boldsymbol{f}_i + \hat{\boldsymbol{w}}_i/\hat{\alpha} T_i$. By introducing new coordinates $\boldsymbol{\zeta}(t) \triangleq \dot{\hat{\boldsymbol{\phi}}}(t)$ and $\dot{\boldsymbol{\zeta}}(t) \triangleq \ddot{\hat{\boldsymbol{\phi}}}(t)$, the second-order differential equation (5.18) can be rewritten as the following linear time-varying first-order differential equation

$$\dot{\boldsymbol{\zeta}}(t) = \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{\phi}}}(t)\, \boldsymbol{\zeta}(t) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(t)\, \dot{\boldsymbol{u}}(t),$$

with the initial value $\boldsymbol{\zeta}(\hat{\alpha}\, \bar{t}_i + \hat{\beta}) = \boldsymbol{f}_i + \hat{\boldsymbol{w}}_i/\hat{\alpha} T_i$. The solution $\boldsymbol{\zeta}(t)$ is given by the superposition of the homogeneous solution $\boldsymbol{\zeta}_h(t)$ and the particular solution $\boldsymbol{\zeta}_p(t)$ (see Kailath (1980, pg. 595-

601) or Ludyk (1995, pg. 40-55))

$$\begin{aligned}\boldsymbol{\zeta}(t) &= \boldsymbol{\zeta}_h(t) + \boldsymbol{\zeta}_p(t) \\ &= \boldsymbol{\Phi}(t, \hat{\alpha}\,\bar{t}_i + \hat{\beta})\,\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}) + \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{t} \boldsymbol{\Phi}(t,\tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau,\end{aligned}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the time-varying transition matrix. Consequently, the overall solution evaluated at the sampling time $\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}$ becomes

$$\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}) = \boldsymbol{X}_{i+1}^{i}\left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}T_i}\right) + \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}, \tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau.$$

By noting that $\boldsymbol{\zeta}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}) = \boldsymbol{f}_{i+1} + \hat{\boldsymbol{w}}_i/\hat{\alpha}T_i$, we get $\forall i \in \underline{\mathcal{I}}$

$$\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}T_i} - \boldsymbol{X}_{i+1}^{i}\left(\boldsymbol{f}_i + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}T_i}\right) = \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}, \tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau. \tag{5.19}$$

This result is now key for proofing Lemma 5.3.1 by means of induction. Let $S(i)$ be the statement (5.17). For $i = k - N$, equation (5.19) reads as

$$\begin{aligned}\boldsymbol{f}_{k-N+1} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{\alpha}T_{k-N}} - \boldsymbol{X}_{k-N+1}^{k-N}\left(\boldsymbol{f}_{k-N} + \frac{\hat{\boldsymbol{w}}_{k-N}}{\hat{\alpha}T_{k-N}}\right) \\ = \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{k-N+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_{k-N+1} + \hat{\beta}, \tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau,\end{aligned}$$

and is equivalent to $S(k - N + 1)$, i. e. $S(k - N + 1)$ is true. This starts the induction. Now assume that $S(i)$ is true for some $i \in \underline{\mathcal{I}}$. Solving this assumption for $\boldsymbol{f}_i$ and inserting it into (5.19) yields

$$\begin{aligned}&\boldsymbol{f}_{i+1} + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}T_i} - \boldsymbol{X}_{i+1}^{i}\left(\boldsymbol{X}_i^{k-N}\boldsymbol{f}_{k-N} - \sum_{j=k-N}^{i-1}\left(\boldsymbol{X}_i^{j+1} - \boldsymbol{X}_i^{j}\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j} + \frac{\hat{\boldsymbol{w}}_i}{\hat{\alpha}T_i}\right) \\ &= \boldsymbol{X}_{i+1}^{i}\int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}, \tau)\,\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\,\dot{\boldsymbol{u}}(\tau)\,d\tau + \int_{\hat{\alpha}\,\bar{t}_i+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}, \tau)\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\dot{\boldsymbol{u}}(\tau)\,d\tau.\end{aligned}$$

By applying the properties stated in Lemma 4.3.4, we get

$$\boldsymbol{f}_{i+1} - \boldsymbol{X}_{i+1}^{k-N}\boldsymbol{f}_{k-N} + \sum_{j=k-N}^{i}\left(\boldsymbol{X}_{i+1}^{j+1} - \boldsymbol{X}_{i+1}^{j}\right)\frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\,T_j} = \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_{i+1}+\hat{\beta}} \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_{i+1} + \hat{\beta}, \tau)\,\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\,\dot{\boldsymbol{u}}(\tau)\,d\tau,$$

which is $S(i + 1)$, completing the inductive step. Thus, by the principle of induction, $S(i)$ is true for all $i \in \underline{\mathcal{I}}$ and, accordingly, Lemma 5.3.1 holds. □

The equivalence stated in Lemma 5.3.1 provides the basis for deriving necessary and sufficient conditions for the elements of the first and second column of (5.15) to be non-vanishing.

Lemma 5.3.2. *The i-th element of the second column of (5.15), i. e. $\partial \boldsymbol{h}_i / \partial \hat{\beta}$, is non-vanishing if and only if $\exists\, \boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$,*

$$\int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i}\, \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_i + \hat{\beta}, \tau)\, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\, \dot{\boldsymbol{u}}(\tau)\, d\tau \neq \boldsymbol{0}. \tag{5.20}$$

Proof. Equation (5.20) directly results from applying Lemma 5.3.1 to the elements (5.16b) of the derivative of the observation map (5.10). □

Note that the condition derived in this Lemma is identical to the one in Lemma 5.2.2. Consequently, the aforementioned does apply universally.

Lemma 5.3.3. *The i-th element of the first column of (5.15), i. e. $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$, is non-vanishing if and only if $\exists\, \boldsymbol{u}(t) \in \mathbb{U}$,*

$$\begin{aligned} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} \neq \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} \left(\left(\frac{\bar{t}_{k-N}}{\bar{t}_i} - 1 \right) \boldsymbol{f}_{k-N} \right) + \sum_{j=k-N}^{i-1} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{w}}_j} \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\bar{t}_i} \\ + \sum_{j=k-N}^{i-1} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left[\left(1 - \frac{\bar{t}_{j+1}}{\bar{t}_i} \right) \boldsymbol{X}_i^{j+1} - \left(1 - \frac{\bar{t}_j}{\bar{t}_i} \right) \boldsymbol{X}_i^{j} \right] \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha} T_j}. \end{aligned} \tag{5.21}$$

Proof. After recalling the first-order state sensitivities derived in Theorem 4.3.7, $\partial \hat{\boldsymbol{\phi}}_i / \partial \hat{\alpha}$ can be rearranged in the following way

$$\begin{aligned} \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\alpha}} = \bar{t}_i \frac{\partial \hat{\boldsymbol{\phi}}_i}{\partial \hat{\beta}} + \left(\bar{t}_i - \bar{t}_{k-N} \right) \boldsymbol{X}_i^{k-N} \boldsymbol{f}_{k-N} - \frac{1}{\hat{\alpha}} \sum_{j=k-N}^{i-1} {}^{j}\boldsymbol{Z}_i^{k-N} \hat{\boldsymbol{w}}_j \\ + \sum_{j=k-N}^{i-1} \left(\left(\bar{t}_{j+1} - \bar{t}_i \right) \boldsymbol{X}_i^{j+1} - \left(\bar{t}_j - \bar{t}_i \right) \boldsymbol{X}_i^{j} \right) \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\, T_j}. \end{aligned}$$

A subsequent left multiplication with $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{\phi}}_i$ in combination with the requirement for the resulting expression to be non-vanishing leads to the desired condition (5.21) which concludes the proof. □

Lemma 5.3.3 connects the derivatives $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$, $\partial \boldsymbol{h}_i / \partial \hat{\beta}$, $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{x}}_{k-N}$ and $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{w}}_i$, $i \in \overline{\mathcal{I}}$. If $\partial \boldsymbol{h}_i / \partial \hat{\beta}$ cannot be generated by a certain combination, then $\partial \boldsymbol{h}_i / \partial \hat{\alpha}$ is non-vanishing.

Based on Lemma 5.3.2 and 5.3.3, we derive in the following Theorem a necessary condition for the control input $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$ of the disturbed NCSs to be extended $N+1$-exciting at the sampling instance k.

Theorem 5.3.4. *A necessary condition for the control input $\boldsymbol{u}(t) \in \mathcal{C}^1(\mathbb{U})$ of the disturbed NCS described in Section 3.1 to be extended $N+1$-exciting at the sampling instance k is that*

the following conditions (C0), (C1), (C2) and (C3) are fulfilled:

$$\text{(C0)}\ N+1 \geq \frac{n_x+2}{n_y}.$$

(C1) $\boldsymbol{u}(t)$ *is* $N+1$*-exciting at the sampling instance* k.

$$\text{(C2)}\ \exists\, j \in \mathcal{I}, \quad \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_j+\hat{\beta}} \frac{\partial \boldsymbol{h}_j}{\partial \hat{\boldsymbol{\phi}}_j}\, \boldsymbol{\Phi}(\hat{\alpha}\,\bar{t}_j+\hat{\beta},\tau)\, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\tau)\, \dot{\boldsymbol{u}}(\tau)\, d\tau \neq \boldsymbol{0}.$$

$$\begin{aligned}\text{(C3)}\ \exists\, i \in \mathcal{I}, \quad \frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} \neq{}& \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} \left(\left(\frac{\bar{t}_{k-N}}{\bar{t}_i} - 1 \right) \boldsymbol{f}_{k-N} \right) + \sum_{j=k-N}^{i-1} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{w}}_j} \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha}\bar{t}_i} \\ &+ \sum_{j=k-N}^{i-1} \frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} \left[\left(1 - \frac{\bar{t}_{j+1}}{\bar{t}_i} \right) \boldsymbol{X}_i^{j+1} - \left(1 - \frac{\bar{t}_j}{\bar{t}_i} \right) \boldsymbol{X}_i^{j} \right] \frac{\hat{\boldsymbol{w}}_j}{\hat{\alpha} T_j}.\end{aligned}$$

Proof. The condition (C0) ensures that (5.15) has at least n_x+2 rows. The condition (C1) guarantees that the last n_x columns of (5.15) have full column rank. The conditions (C2) and (C3) make sure according to the Lemmas 5.3.2 and 5.3.3 that the first and second column of (5.15) are not $\boldsymbol{0}$, respectively. All conditions together are a necessary condition for the full rankness of (5.15). □

It is important to note that if we compare the necessary condition of the undisturbed NCSs, i. e. Lemma 5.2.4, to the one of the disturbed NCSs, i. e. Lemma 5.2.4, we see that the conditions (C0), (C1) and (C2) are identical and (C3) differs by two additional terms. The reason for this stems from the different definitions of the function $\hat{\boldsymbol{\phi}}$ for undisturbed and disturbed NCSs, cf. Definition 3.4.2 and 3.5.2, respectively. The difference between both is the estimated state disturbance $\hat{\boldsymbol{w}}(t)$ defined in Assumption 9. The fact that $\hat{\boldsymbol{w}}(t)$ is piecewise constant and independent of $\hat{\boldsymbol{x}}_{k-N}$ and $\hat{\beta}$ is reflected in the aforementioned finding that the respective conditions (C1) and (C2) are identical for undisturbed and disturbed NCSs. However, the dependence of $\hat{\boldsymbol{w}}(t)$ on $\hat{\alpha}$ forces the appearance of two additional terms in the condition (C3) for the disturbed NCSs.

5.4 Relation of Observability of NCSs to Other Quantities

In this section, we reveal the relation between the introduced notion of observability to the conventional Kalman observability matrix for linear systems, to the update step of the CMHO and CMHE explained in Section 3.4 and 3.5, respectively, and to the gradient of the Lagrangian for the CMHO and CMHE derived in Section 4.2 and 4.3, respectively.

5.4.1 Relation to the Kalman Observability Matrix

The condition (5.5) of Definition 5.2.4 for undisturbed systems as well as the condition (5.14) of Definition 5.3.4 for disturbed systems can be reformulated by means of the transition property stated in Lemma 4.3.4 to yield a nonlinear analogon to the well-known Kalman

observability matrix

$$\operatorname{rank} \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N}} = \operatorname{rank} \begin{bmatrix} \dfrac{\partial \boldsymbol{h}_{k-N}}{\partial \hat{\boldsymbol{\phi}}_{k-N}} \\ \dfrac{\partial \boldsymbol{h}_{k-N+1}}{\partial \hat{\boldsymbol{\phi}}_{k-N+1}} \boldsymbol{X}_{k-N+1}^{k-N} \\ \dfrac{\partial \boldsymbol{h}_{k-N+2}}{\partial \hat{\boldsymbol{\phi}}_{k-N+2}} \boldsymbol{X}_{k-N+2}^{k-N+1} \boldsymbol{X}_{k-N+1}^{k-N} \\ \vdots \\ \dfrac{\partial \boldsymbol{h}_k}{\partial \hat{\boldsymbol{\phi}}_k} \boldsymbol{X}_k^{k-1} \dots \boldsymbol{X}_{k-N+1}^{k-N} \end{bmatrix} = n_x. \tag{5.22}$$

The relation between $\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N}$ and the Kalman observability matrix is clarified by the following example. Consider the undisturbed or disturbed continuous-time linear system $\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \hat{\boldsymbol{w}}(t)$ where the output $\hat{\boldsymbol{y}}(t) = \boldsymbol{C}\hat{\boldsymbol{x}}(t)$ is uniformly sampled with the sampling time T. Then, $\partial \boldsymbol{h}_i / \partial \hat{\boldsymbol{\phi}}_i = \boldsymbol{C}$, $i \in \mathcal{I}$, and $\boldsymbol{X}_{i+1}^{i} = e^{\boldsymbol{A}T} = \boldsymbol{A}_d$, $i \in \overline{\mathcal{I}}$, and (5.22) becomes

$$\operatorname{rank} \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{x}}_{k-N}} = \operatorname{rank} \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A}_d \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}_d^N \end{bmatrix} = n_x \tag{5.23}$$

and coincides for $N + 1 = n_x$ with the conventional observability definition based on the Kalman observability matrix. Consequently, the introduced notion of observability for NCSs can be understood as an extension of the observability definition for linear systems based on the Kalman observability matrix to the case of nonlinear NCSs.

5.4.2 Relation to the Update Step of the CMHS

In this subsection, we explain the relation between the introduced notion of observability of NCSs and the update step of the CMHO and the CMHE by a graphical illustration. To this end, we consider an exemplary situation for which the values of the defining quantities are given in Table 5.1. Although the dimension of the optimization variable $\hat{\boldsymbol{p}}_k$ is unrealistic, the considered situation reflects the dominating characteristics and enables a graphical visualization. The variable $\hat{p}_k$ and $\hat{w}_k$ reflects the optimization variables $\operatorname{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N})$ and $\operatorname{col}(\hat{\boldsymbol{w}}_{k-N}, \dots, \hat{\boldsymbol{w}}_{k-1})$, respectively. The main two differences between the considered undisturbed and disturbed NCS are as follows. First, the dimension of $\hat{\boldsymbol{p}}_k$ is in the latter case two while it is one in the former case. Second, in contrast to the former case, the measurements $\underline{y}_k$ are in the latter case not always in the image of $\mathbb{P}$ under $\underline{h}_k$ due to the measurement disturbances. Although the difference between the undisturbed and disturbed case doesn't look much at a first glance, its impact on the observation problem is significant.

Table 5.1: Illustrative example: values for the defining quantities.

NCS	description	source	value
undisturbed	optimization variable $\hat{\boldsymbol{p}}_k$	Eq. (3.16)	$\hat{\boldsymbol{p}}_k = \hat{p}_k \in \mathbb{P} \subset \mathbb{R}$
	observation map $\underline{\boldsymbol{h}}_k$	Eq. (5.1)	$\underline{h}_k(\mathbb{P}) = [\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}] \subset \mathbb{R}$
	observability	Def. 5.2.1 true Def. 5.2.2 true	$\Rightarrow \operatorname{rank} \dfrac{\partial \underline{h}_k(\hat{p}_k)}{\partial \hat{p}_k} = 1$
	measurements		$\underline{y}_k$ always in $[\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$
disturbed	optimization variable $\hat{\boldsymbol{p}}_k$	Eq. (3.22)	$\hat{\boldsymbol{p}}_k = [\hat{p}_k, \hat{w}_k]^T \in \mathbb{P} \subset \mathbb{R}^2$
	observation map $\underline{\boldsymbol{h}}_k$	Eq. (5.10)	$\underline{h}_k(\mathbb{P}) = [\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}] \subset \mathbb{R}$
	observability	Def. 5.3.1 true Def. 5.3.2 true	$\Rightarrow \operatorname{rank} \dfrac{\partial \underline{h}_k(\hat{p}_k, \hat{w}_k)}{\partial \hat{p}_k} = 1$
	measurements		$\underline{y}_k$ not always in $[\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$

Undisturbed Networked Control System

Figure 5.1 illustrates the relation between the introduced notion of observability of undisturbed NCSs and the functionality of the update step of the CMHO.Since the conditions in Definition 5.2.1 and 5.2.2 are satisfied, we can invoke the implicit function theorem (Fleming, 1977) to investigate the equation $\underline{h}_k(\hat{p}_k) - \underline{\hat{y}}_k = 0$. This theorem states that there exists a neighborhood $\hat{\mathcal{P}}$ of $\hat{p}_k$ and a neighborhood $\hat{\mathcal{Y}}$ of $\underline{\hat{y}}_k$ such that for every $\underline{\hat{y}}_k$ in $\hat{\mathcal{Y}}$ there exists precisely one $\hat{p}_k$ in $\hat{\mathcal{P}}$ such that $\underline{h}_k(\hat{p}_k) - \underline{\hat{y}}_k = 0$. In other words, this guarantees (at least local) injectivity of $\underline{h}_k(\hat{p}_k)$. This can be seen in Figure 5.1 by the fact that every intersection between the line $\underline{\hat{y}}_k = c$ with $c \in [\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$ and $\underline{h}_k(\hat{p}_k)$ is (at least locally) a single point.

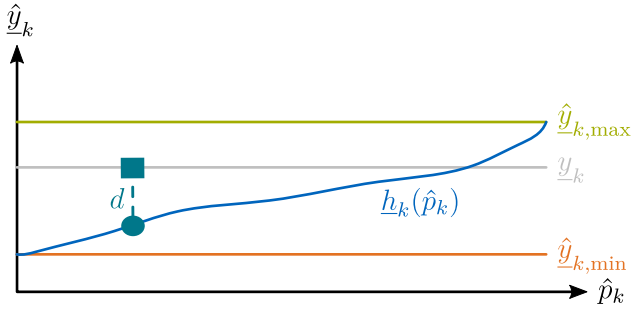


Figure 5.1: Illustration of the relation between observability of undisturbed NCSs and the functionality of the update step of the CMHO.

The CMHO tackles this observation problem by trying to find a point $\hat{p}_k$ on the blue curve $\underline{h}_k(\hat{p}_k)$ such that the distances d between this point and the gray line $\underline{\hat{y}}_k = \underline{y}_k$ is minimized. Since the true measurement $\underline{y}_k$ is always inside $[\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$, the minimal distance is always zero. In other words, the optimal solution of the update step of the CMHO is the true variable.

Disturbed Networked Control System

Figure 5.2 illustrates the relation between the introduced notion of observability of disturbed NCSs and the functionality of the update step of the CMHE.Since the conditions in Definition 5.3.1 and 5.3.2 are satisfied, we can invoke the implicit function theorem (Fleming, 1977) to investigate the equation $\underline{h}_k(\hat{p}_k, \hat{w}_k) - \underline{\hat{y}}_k = 0$. Suppose that $\underline{\hat{y}}_k$ is fixed, then there exists a neighborhood $\hat{\mathcal{P}}$ of $\hat{p}_k$ and a neighborhood $\hat{\mathcal{W}}$ of $\hat{w}_k$ such that for every $\hat{p}_k$ in $\hat{\mathcal{P}}$ there exists precisely one $\hat{w}_k$ in $\hat{\mathcal{W}}$ such that $\underline{h}_k(\hat{p}_k, \hat{w}_k) - \underline{\hat{y}}_k = 0$. In other words, there exists infinite unique combinations $(\hat{p}_k, \hat{w}_k)$ which are mapped to the same $\underline{\hat{y}}_k$. It is important to stress that these combinations are (at least locally) unique. This can be seen in Figure 5.2 by the fact that every intersection between the planes $\underline{\hat{y}}_k = c$ with $c \in [\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$ and $\underline{h}_k(\hat{p}_k, \hat{w}_k)$ are (at least locally) non-intersecting curves. Moreover, the observation problem is aggravated by the fact that the true measurements $\underline{y}_k$ need not to be inside $[\underline{\hat{y}}_{k,\min}, \underline{\hat{y}}_{k,\max}]$ due to the measurement disturbances. This can be seen in Figure 5.2 by the fact that the gray plane $\underline{\hat{y}}_k = \underline{y}_k$ is beneath the red plane $\underline{\hat{y}}_k = \underline{y}_{k,\min}$.

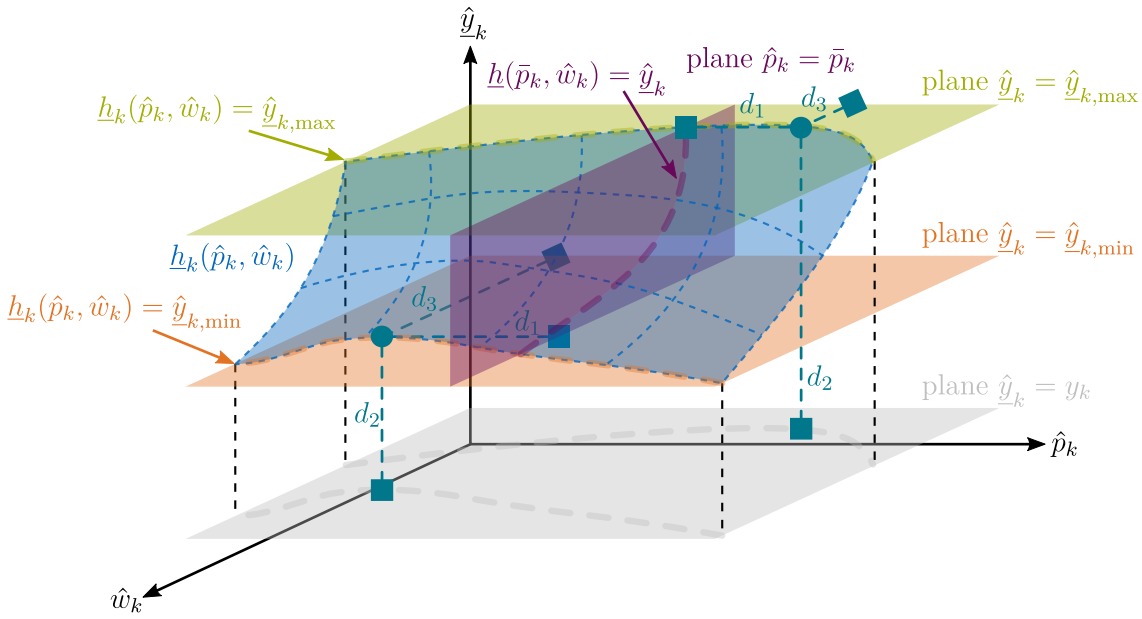


Figure 5.2: Illustration of the relation between observability for disturbed NCSs and the functionality of the update step of the CMHE.

The CMHE tackles this estimation problem by trying to find a point $(\hat{p}_k, \hat{w}_k)$ on the blue surface $\underline{h}_k(\hat{p}_k, \hat{w}_k)$ such that the weighted sum of the three distances d_1, d_2 and d_3 are minimized. The distances d_1, d_2 and d_3 are the distances between a point $(\hat{p}_k, \hat{w}_k)$ on the blue surface $\underline{h}_k(\hat{p}_k, \hat{w}_k)$ and the purple plane $\hat{p}_k = \bar{p}_k$, gray plane $\underline{\hat{y}}_k = \underline{y}_k$, and the plane $\hat{w}_k = 0$, respectively. Moreover, the distances d_1, d_2 and d_3 correspond to the first term (3.20a), second term (3.20b) and third term (3.20c) of the cost function (3.20) in the update step, respectively. These distances are exemplary depicted for two points marked with a filled circle. The optimal point depends on the chosen weightings. It is important to note that a simple minimization of the distance d_2, like in the undisturbed case, is ill-posed since there are infi-

nite points minimizing d_2, namely all points on the curve in the plane $\hat{\underline{y}}_k = \hat{\underline{y}}_{k,\min}$. For this reason, the weighted distance d_3 is considered in the cost function as the third term (3.20c). On the one hand, this addition guarantees well-posedness of the minimization problem. On the other hand, this addition can be interpreted as a model for the state disturbances. This means that the estimation quality depends on the relation between the characteristic of the true state disturbance and the estimated state disturbance model. Consequently, if we know some characteristics of the true state disturbances, then we can incorporate this information by formulating a suitable expression for the third term (3.20c). For instance, if the state disturbance is $\boldsymbol{w}(t) = \mathcal{N}(\boldsymbol{a}, \boldsymbol{Q})$, then a good choice for the third term is $\Psi_{i|k} = \|\hat{\boldsymbol{w}}_{i|k} - \boldsymbol{a}\|^2_{\boldsymbol{Q}^{-1}}$. Another consequence is that, in contrast to the undisturbed scenario, the state estimation error in the disturbed case won't vanish and the best thing to hope for is boundedness. The remaining first term (3.20a) serves two goals. First, it enables the incorporation of past information which is not explicitly accounted for in the current buffer. Second, it guaranties well-posedness of the minimization problem even if the condition of Definition 5.3.2 is not satisfied in the current buffer. In this case, there exists points $(\hat{p}_k, \hat{w}_k)$ on the blue surface which have the same distances d_2 and d_3. However, these points have different distances d_1.

5.4.3 Relation to the Gradient and Hessian of the Lagrangian for the CMHS

Having the relation between observability of NCSs and the update steps of the CMHS presented in the previous subsection in mind, it is not surprising that there exists a relation between the Jacobian $\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k$ of the observation map and the derivatives of the Lagrangian for the CMHS.

The first and second term of the exact gradient of the Lagrangian L stated in Lemma 4.2.1 and 4.3.1 for the CMHO and CMHE are of identical structure, respectively. This term can be reformulated to yield

$$\sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \frac{\partial \Upsilon_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}} = \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \frac{\partial \boldsymbol{h}_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}}^T \frac{\partial \Upsilon_{i|k}}{\partial \boldsymbol{h}_{i|k}} = \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k}^T \frac{\partial \operatorname{col}(\Upsilon_{i|k}, i \in \mathcal{I}_k)}{\partial \underline{\boldsymbol{h}}_k}, \tag{5.24}$$

which reveals the aforementioned relation. This opens up the possibility of checking observability online at the price of additionally calculating the rank of a matrix.

Similar, the first and second term of the first part $\boldsymbol{H}_1$ of the Hessian of the Lagrangian L stated in (4.22b) and (4.24b) for the CMHO and CMHE are of identical structure, respectively. By applying Lemma B.1 to $\partial^2 \Upsilon_{i|k} / \partial \hat{\boldsymbol{\phi}}^2_{i|k}$, this term can be reformulated as follows

$$\sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \frac{\partial^2 \Upsilon_{i|k}}{\partial \hat{\boldsymbol{\phi}}^2_{i|k}} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k} = \sum_{i=k-N}^{k} \left(\frac{\partial \boldsymbol{h}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \frac{\partial^2 \Upsilon_{i|k}}{\partial \boldsymbol{h}^2_{i|k}} \frac{\partial \boldsymbol{h}_{i|k}}{\partial \hat{\boldsymbol{p}}_k} + \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \left(\sum_{j=1}^{\dim(\boldsymbol{h}_{i|k})} \frac{\partial \Upsilon_{i|k}}{\partial {}_j h_{i|k}} \frac{\partial^2 {}_j h_{i|k}}{\partial \hat{\boldsymbol{\phi}}^2_{i|k}} \right) \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k} \right)$$

$$= \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k}^T \operatorname{diag}\left(\frac{\partial^2 \Upsilon_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}^2}, i \in \mathcal{I}_k\right) \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k} + \sum_{i=k-N}^{k} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}^T \left(\sum_{j=1}^{\dim(\boldsymbol{h}_{i|k})} \frac{\partial \Upsilon_{i|k}}{\partial_j h_{i|k}} \frac{\partial^2{}_j h_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}^2}\right) \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k}, \tag{5.25}$$

which reveals the aforementioned relation. Note that both relations will be of key importance for the stability analysis in Chapter 6.

5.5 Examples

In order to clarify the introduced terminology and the derived conditions, we first consider two cases of undisturbed linear NCSs which can be investigated analytically. For the first case, we derive the elements of the Jacobian of the observation map in closed form and verify analytically the equivalence presented in Lemma 5.2.1. This insight is used for the second case to clarify the introduced notion of observability and the derived conditions in Theorem 5.2.4. Afterwards, we consider a networked pendulum which falls into the category of disturbed nonlinear NCSs. We examine the extended excitation property of the control input to a real networked pendulum test-rig and compare them to the simulated counterpart.

5.5.1 Linear Networked Control System: The Undisturbed General Case

Consider the centralized NCS architecture depicted in Figure 3.1 where the system Σ consists of the linear model

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{5.26}$$

and where the sensor Σ_{S} possesses the measurement model $\boldsymbol{y}(\bar{t}_i) = \boldsymbol{C}\boldsymbol{x}(\bar{t}_i)$. Since we consider in the following only the buffer $\mathcal{B}_k$, we ease the presentation by omitting the subscripts indicating the buffer affiliation. By introducing the abbreviations

$$\boldsymbol{\Phi}_i \triangleq e^{\boldsymbol{A}\hat{\alpha}(\bar{t}_i - \bar{t}_{k-N})}, \quad \boldsymbol{\eta}_i \triangleq \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} e^{\boldsymbol{A}(\hat{\alpha}\,\bar{t}_i+\hat{\beta}-\tau)} \boldsymbol{B}\boldsymbol{u}(\tau) d\tau, \tag{5.27}$$

we can analytically express the states $\hat{\boldsymbol{x}}_i$, $i \in \mathcal{I}$, corresponding to the buffer $\mathcal{B}_k$ as (Kailath, 1980)

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\Phi}_i \hat{\boldsymbol{x}}_{k-N} + \boldsymbol{\eta}_i. \tag{5.28}$$

Due to the existence of a closed-form solution, the equivalence presented in Lemma 5.2.1 can be analytically verified. To this end, we need to calculate the first-order state sensitivities which can be calculated either by Theorem 4.2.3, or, in this case, by direct differentiation

of $\hat{\boldsymbol{x}}_i$. In the former case, by noting that for $i \in \mathcal{I}$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{\phi}}_i} = \boldsymbol{C}, \tag{5.29a}$$

$$\boldsymbol{X}_i^{k-N} = e^{\boldsymbol{A}\hat{\alpha}(\bar{t}_i - \bar{t}_{k-N})}, \tag{5.29b}$$

$$\boldsymbol{f}_i = \boldsymbol{A}\left(\boldsymbol{\Phi}_i \hat{\boldsymbol{x}}_{k-N} + \boldsymbol{\eta}_i\right) + \boldsymbol{B}\boldsymbol{u}_i, \tag{5.29c}$$

$$\boldsymbol{f}_{k-N} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-N} + \boldsymbol{B}\boldsymbol{u}_{k-N}, \tag{5.29d}$$

we can express the elements of (5.6) for $i \in \mathcal{I}$ as

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\alpha}} = \boldsymbol{C}\left((\bar{t}_i - \bar{t}_{k-N})\boldsymbol{A}\boldsymbol{\Phi}_i + \bar{t}_i\boldsymbol{A}\boldsymbol{\eta}_i + \bar{t}_i\boldsymbol{B}\boldsymbol{u}_i - \bar{t}_{k-N}\boldsymbol{\Phi}_i\boldsymbol{B}\boldsymbol{u}_{k-N}\right), \tag{5.30a}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} = \boldsymbol{C}\left(\boldsymbol{A}\boldsymbol{\eta}_i + \boldsymbol{B}\boldsymbol{u}_i - \boldsymbol{\Phi}_i\boldsymbol{B}\boldsymbol{u}_{k-N}\right), \tag{5.30b}$$

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \boldsymbol{C}\boldsymbol{\Phi}_i. \tag{5.30c}$$

Note that if $u(t) \in \mathcal{C}^1(\mathbb{U})$, (5.30b) can be integrated by parts to yield

$$\frac{\partial \boldsymbol{h}_i}{\partial \hat{\beta}} = \boldsymbol{C}\left(\boldsymbol{B}\boldsymbol{u}_i - \boldsymbol{\Phi}_i\boldsymbol{B}\boldsymbol{u}_{k-N} - (-\boldsymbol{A}\boldsymbol{\eta}_i)\right) = \boldsymbol{C}\int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} e^{\boldsymbol{A}(\hat{\alpha}\,\bar{t}_i+\hat{\beta}-\tau)}\boldsymbol{B}\dot{\boldsymbol{u}}(\tau)d\tau, \tag{5.30d}$$

which is equivalent to the aforementioned latter case of direct differentiation. This is a straight consequence of the equivalence derived in Lemma 5.2.1.

5.5.2 Linear Networked Control System: An Undisturbed Example

Consider the identical NCS setup as before where the system Σ described in (5.26) is

$$\begin{bmatrix} {}_1\dot{x}(t) \\ {}_2\dot{x}(t) \end{bmatrix} = \begin{bmatrix} -1 & \gamma \\ 0 & -1 \end{bmatrix} \begin{bmatrix} {}_1x(t) \\ {}_2x(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \tag{5.31}$$

with $\gamma \in \mathbb{R}$. The sensor Σ_{S} possesses the measurement model $y(\bar{t}_i) = {}_1x(\bar{t}_i)$. Then the elements of the Jacobian of the observation map corresponding to the buffer $\mathcal{B}_k$ can be calculated by means of (5.30) for $i \in \mathcal{I}$ as

$$\begin{aligned}\frac{\partial h_i}{\partial \hat{\alpha}} = {} & (\bar{t}_i - \bar{t}_{k-N})e^{-\hat{\alpha}(\bar{t}_i - \bar{t}_{k-N})}\Big(-{}_1\hat{x}_{k-N} + \gamma\Big(1 - \hat{\alpha}\left(\bar{t}_i - \bar{t}_{k-N}\right)\Big){}_2\hat{x}_{k-N} + u(\hat{\alpha}\,\bar{t}_{k-N} + \hat{\beta}\Big) \\ & + \bar{t}_i\int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} e^{-(\hat{\alpha}\,\bar{t}_i+\hat{\beta})+\tau}\dot{u}(\tau)\,d\tau,\end{aligned} \tag{5.32a}$$

$$\frac{\partial h_i}{\partial \hat{\beta}} = \int_{\hat{\alpha}\,\bar{t}_{k-N}+\hat{\beta}}^{\hat{\alpha}\,\bar{t}_i+\hat{\beta}} e^{-(\hat{\alpha}\,\bar{t}_i+\hat{\beta})+\tau}\dot{u}(\tau)\,d\tau, \tag{5.32b}$$

$$\frac{\partial h_i}{\partial \hat{\boldsymbol{x}}_{k-N}} = \begin{bmatrix} 1 \\ \gamma\hat{\alpha}(\bar{t}_i - \bar{t}_{k-N}) \end{bmatrix} e^{-\hat{\alpha}(\bar{t}_i - \bar{t}_{k-N})}. \tag{5.32c}$$

If $\gamma \neq 0$, we see from (5.32c) that the system satisfies for any $N+1 \geq 2$ the $N+1$-observability rank condition and any $u(t) \in \mathbb{R}$ is $N+1$-exciting at any sampling instance k. The inspection of (5.32b) reveals that the choice of γ is immaterial for the mapping from $\dot{u}$ to $\partial h_i / \partial \hat{\beta}$. In contrast to (5.32c) and (5.32b), (5.32a) depends explicitly on the condition of the state and the control input at the time $\hat{\alpha}\, \bar{t}_{k-N} + \hat{\beta}$. Moreover, if $\gamma \neq 0$, the NCS satisfies for any $N+1 \geq 4$ the extended $N+1$-observability rank condition. However, not every $u(t)$ is extended $N+1$-exciting, like the trivial choice of a constant control input which violates condition (C2) of Theorem 5.2.4. Another possibility is if $u(t)$ is such that (5.32b) and/or (5.32a) vanishes if the system is sampled at certain sensor times $\bar{t}_i$. As a consequence, the conditions (C2) and/or (C3) of Theorem 5.2.4 are not fulfilled and $u(t)$ is not extended $N+1$-exciting. To investigate such a situation, we consider the case where $N+1=4$ and examine the buffer $\mathcal{B}_4$. This results in the index set $\mathcal{I}_4 = \{1,2,3,4\}$. Since we consider in the following only the buffer $\mathcal{B}_4$, we ease the presentation by omitting the subscripts indicating the buffer affiliation. Moreover, the sensor possesses the clock parameters $\alpha = 1$ and $\beta = -0.5\,\mathrm{s}$ and the sensor times $\bar{t}_i = (i\,T - 0.5)\,\mathrm{s}$, $i \in \mathbb{N}$, where $T = 1\,\mathrm{s}$ is the sampling time. Then a control input which renders (5.32b) zero at the times $\hat{\alpha}\, \bar{t}_i + \hat{\beta}$, $i \in \mathcal{I}$, with $\hat{\alpha} = 1$ and $\hat{\beta} = -0.5\,\mathrm{s}$, for any γ, any initial state $\hat{\boldsymbol{x}}_1$ and any packet delay τ_i, $i \in \mathcal{I}$, is

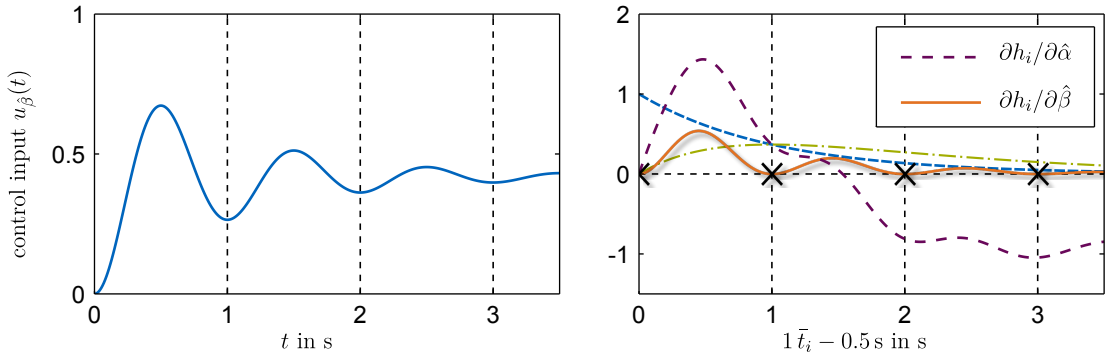
$$u_{\hat{\beta}}(t) = \tilde{u}\left(2\pi - \left(2\pi\cos(2\pi t) + \sin(2\pi t)\right) e^{-t}\right),$$

where $\tilde{u} \in \mathbb{R}$ is an arbitrary amplitude. Similarly, a control input which renders (5.32a) zero at the times $\hat{\alpha}\, \bar{t}_i + \hat{\beta}$, $i \in \mathcal{I}$, with $\hat{\alpha} = 1$ and $\hat{\beta} = -0.5\,\mathrm{s}$, for any γ, any initial state $\boldsymbol{x}_1$ and any packet delay τ_i, $i \in \mathcal{I}$, is

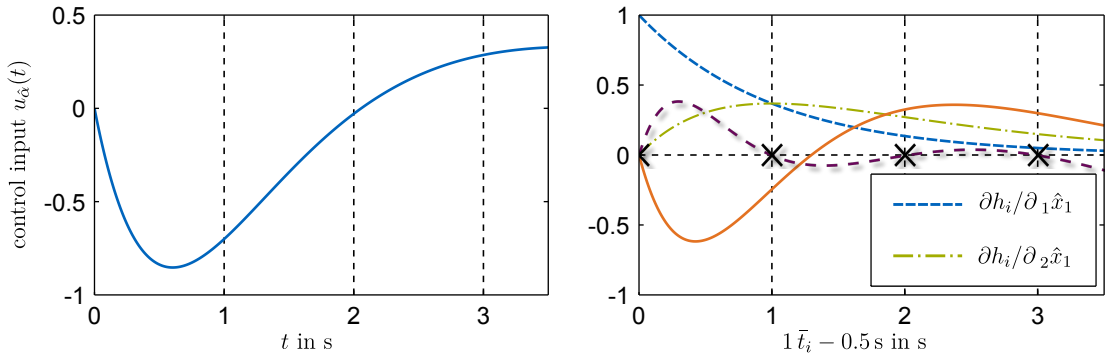
$$u_{\hat{\alpha}}(t) = \frac{2}{105}\left[\left(29 + \left(-29 + 28t - 12t^2\right) e^{-t}\right)\, {}_1\hat{x}_1 + \left(9 + \left(-9 - 42t + 18t^2\right) e^{-t}\right)\, {}_2\hat{x}_1\right].$$

Figure 5.3 shows these control inputs and the corresponding derivatives (5.32) as a function of the sensor time $\bar{t}_i$ expressed in global time, i. e. $t = 1\,\bar{t} + 0.5\,\mathrm{s}$. The sampling times $\bar{t}_i \in \{0.5\,\mathrm{s}, 1.5\,\mathrm{s}, 2.5\,\mathrm{s}, 3.5\,\mathrm{s}\}$, which are $\{0\,\mathrm{s}, 1\,\mathrm{s}, 2\,\mathrm{s}, 3\,\mathrm{s}\}$ in global time, and where the derivatives $\partial h_i / \partial \hat{\beta}$ and $\partial h_i / \partial \hat{\alpha}$ vanish, are marked with a cross in Figure 5.3(a) and 5.3(b), respectively. Note that in both figures, the derivative $\partial h_i / \partial \hat{\boldsymbol{x}}_1$ is identical since it is independent of the control input.

The condition number ϱ of the Jacobian of the observation map is the ratio of the largest to the smallest singular value and can be used as a measure of observability. If the observability rank condition is not satisfied, i. e. the Jacobian is rank-deficient, the smallest singular value is zero and the condition number tends to infinity. Hence, the lower is the condition number, the better is the observability.



(a) Control input $u_{\hat{\beta}}(t)$ with $\tilde{u} = \frac{1}{15}$.



(b) Control input $u_{\hat{\alpha}}(t)$.

Figure 5.3: Comparison between different control inputs and the resulting derivatives $\partial h_i/\partial\hat{\alpha}$, $\partial h_i/\partial\hat{\beta}$, $\partial h_i/\partial\hat{\boldsymbol{x}}_1$ as a function of the sensor time $\bar{t}_i$ expressed in global time, i. e. $t = 1\,\bar{t} + 0.5\,\mathrm{s}$, for the parameters $\hat{\alpha} = 1$, $\hat{\beta} = -0.5\,\mathrm{s}$, $\bar{t}_{k-N} = 0.5\,\mathrm{s}$, $\hat{\boldsymbol{x}}_1 = [-1, 4]^T$ and $\gamma = 1$.

In Figure 5.4, the condition numbers of $\partial\underline{\boldsymbol{h}}_2/\partial\hat{\boldsymbol{x}}_1$ and $\partial\underline{\boldsymbol{h}}_4/\partial\hat{\boldsymbol{p}}_4$ are depicted as a function of the sampling time T for the scenario shown in Figure 5.3(b). We can observe two important things. First, the control input $u_{\hat{\alpha}}(t)$ is 2-exciting for all depicted sampling times and extended 4-exciting for all depicted sampling times except $T = 1\,\mathrm{s}$. In this case, the condition number

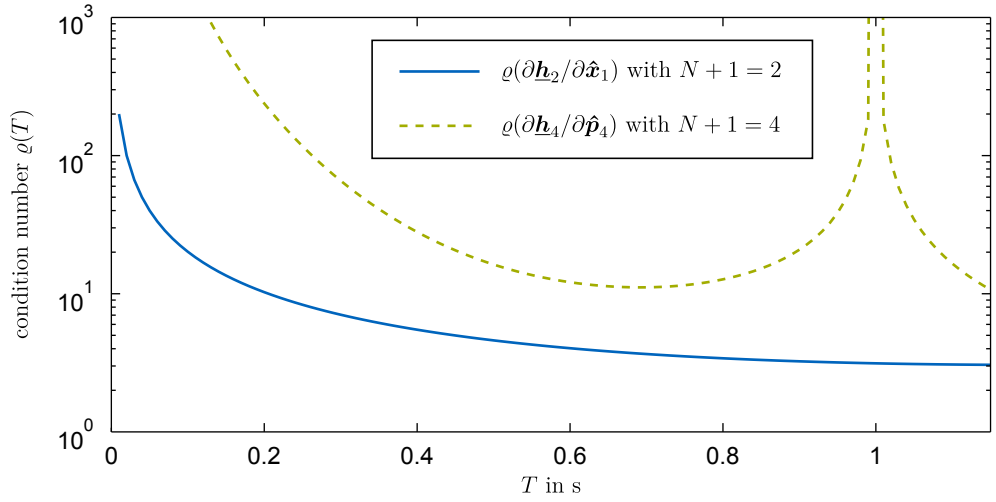


Figure 5.4: Comparison between the condition numbers $\varrho(\partial\underline{\boldsymbol{h}}_2/\partial\hat{\boldsymbol{x}}_1)$ and $\varrho(\partial\underline{\boldsymbol{h}}_4/\partial\hat{\boldsymbol{p}}_4)$ as a function of the sampling time T.

turns to infinity due to the zero column in $\partial \underline{\boldsymbol{h}}_4 / \partial \hat{\boldsymbol{p}}_4$ caused by the vanishing derivatives $\partial h_i / \partial \hat{\alpha}$, $i \in \mathcal{I}$, which can be seen in Fig. 5.3(b) marked with crosses. Second, the slightly higher condition number of $\partial \underline{\boldsymbol{h}}_4 / \partial \hat{\boldsymbol{p}}_4$ indicates that the combined state and clock parameter estimation problem is slightly more demanding compared to the sole state estimation problem.

5.5.3 Nonlinear Networked Control System: The Networked Pendulum

We use the networked pendulum described in Section 7.2 with the parameters and settings corresponding to the closed-loop benchmark with the network scenario A shown in Figure 7.11(a,b).

The upper plot in Figure 5.5 shows the simulated and experimental closed-loop control inputs $u_{\text{sim}}(t)$ and $u_{\text{exp}}(t)$, respectively, for the three phases: ① swing-up, ② stabilization and ③ set-point change. Note that the smoother course of $u_{\text{sim}}(t)$ is due to the undisturbed simulation.

The lower plot of Figure 5.5 depicts the related condition numbers $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$, $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$, $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$ and $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$ for a buffer size of $N+1 = 25$. The condition numbers $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$ and $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$ are bounded and almost identical which implies that the networked pendulum satisfies the 25-observability rank condition and that the control inputs $u_{\text{sim}}(t)$ and $u_{\text{exp}}(t)$ are 25-exciting. In fact, the networked pendulum system satisfies for any $N+1 \geq 2$ the $N+1$-observability rank condition and for any $N+1 \geq 3$ the extended $N+1$-observability rank condition. However, not every control input is extended $N+1$-exciting. This can be seen by the condition numbers $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$ and $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$.

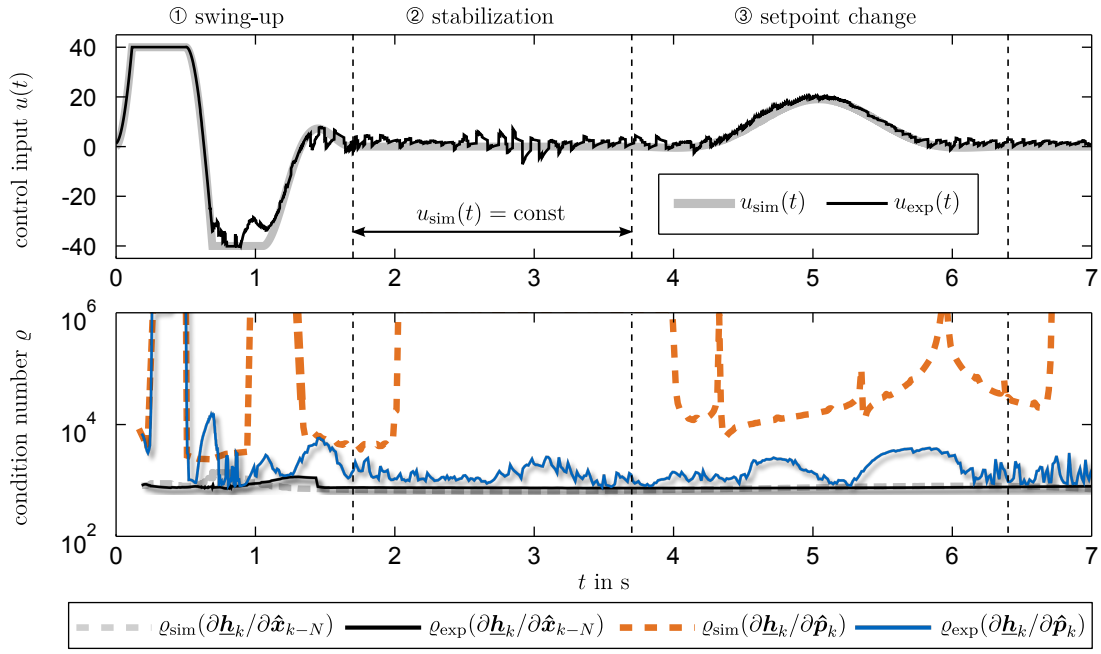


Figure 5.5: Comparison between the simulated and experimental control inputs $u_{\text{sim}}(t)$ and $u_{\text{exp}}(t)$ and the resulting condition numbers $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$, $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{x}}_{k-N})$, $\varrho_{\text{sim}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$ and $\varrho_{\text{exp}}(\partial \underline{\boldsymbol{h}}_k / \partial \hat{\boldsymbol{p}}_k)$.

Both tend to infinity whenever the control input corresponding to the buffer $\mathcal{B}_k$ is constant, like in the swing-up phase where both control inputs reach the maximum allowed value and thus violets condition (C2) of Theorem 5.2.4 respective Theorem 5.3.4. Note that the time between the moment where the control input remains constant and the moment where observability is lost corresponds to the time between the first and the last measurement in the buffer and scales with the buffer size N.

It is important to note that in contrast to $u_{\text{sim}}(t)$, the control input $u_{\text{exp}}(t)$ is extended 25-exciting during the stabilization phase. This means that slight perturbations, which are characteristic for real world applications, are enough to satisfy the extended excitation property and thus to enable observability of NCSs. Moreover, the combined state and clock parameter estimation problem of the real networked pendulum is better conditioned than the simulated counterpart and possesses almost the same condition numbers as the sole state estimation problem itself.

5.6 Summary

In this chapter, we have introduced the observability notion for undisturbed and disturbed NCSs. This notion ensures (at least locally) the well-posedness of the respective observation map by guaranteeing the full-rankness of its derivative with respect to $\text{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})$. The observability of NCSs does not only depend on the structure of the system but also on the information content of the control input. To this end, we have derived for the undisturbed as well as for the disturbed NCSs a necessary condition for the control input to be sufficiently exciting. The key idea for achieving this condition was to establishing a relation between the control input and the first-order state sensitivities appearing in the elements of the derivative of the observation map. The resulting conditions are almost identical for the undisturbed and disturbed NCSs and the only difference stems from the dependence of the estimated state disturbance $\hat{\boldsymbol{w}}(t)$ on the clock parameter $\hat{\alpha}$. Moreover, we have revealed the relation between the introduced notion of observability to the conventional Kalman observability matrix for linear systems and to the update step as well as to the gradient and Hessian of the Lagrangian of the CMHO and CMHE. Finally, we have clarified the introduced terminology and the derived conditions by a linear example system and pointed out their practical relevance by investigating a real networked pendulum test-rig.

Chapter 6

Stability Analysis

In this chapter, we analyze the stability of the nominal centralized moving horizon observer (CMHO) and the nominal centralized moving horizon estimator (CMHE) presented in Chapter 3. The arising difficulties and challenges stem on the one hand from the complexity of the centralized moving horizon strategies (CMHS) and on the other hand from the general framework expressed by the non-restrictive assumptions made in Chapter 3. Although these assumptions aim to capture realistically the actual circumstances, they do significantly aggravate the stability analysis. More precisely, the major six sources of difficulties are: ① the nonlinearities induced by the system Σ and the sensor Σ_{S}, ② the special nature of the problem, i. e. the combination of a continuous-time system with non-uniformly sampled discrete-time measurements, ③ the unsynchronized sensor clocks, ④ the bounded but unknown disturbance statistics, ⑤ the bounded but unknown network statistics and ⑥ the usage of suboptimal solutions to the optimization problems of the update steps instead of the optimal ones which can be derived by all admissible gradient based optimization algorithms.

The approach to tackle the CMHS stability analysis consists of two key ideas. The first one is to treat the gradient-based optimization algorithms used for finding (sub)optimal solutions to the optimization problems stemming from the nominal update step of the k-th buffer as a nonlinear continuous-time dynamical system with the optimization variable $\hat{\boldsymbol{p}}_k$ as the state vector. This facilitates the investigation of the stability properties of this system by means of Lyapunov-type arguments. The second one is to understand the transition from one optimization problem to the next, combined with the choice of the initial conditions according to Section 3.6, as a change in the vector field of the gradient-based optimization system along with an impulsive adjustment of its current state. From this point of view, the core of the CMHS stability analysis is tantamount to investigating the stability of a nonlinear continuous-time switched impulsive system. This is done by invoking arguments based on the multiple Lyapunov functions framework (Branicky, 1998).

Based on these considerations, the contribution of this chapter is three-fold: First, we introduce the perception of the gradient-based optimization algorithms presented in Section 2.4.2, which use the efficient gradient calculation method derived in Chapter 4 for generating (sub)-optimal solutions to the optimization problems stemming from the update steps, as nonlinear continuous-time dynamical systems. Second, we derive conditions under which we can prove

that the observation error of the CMHO converges asymptotically as well as in finite-time to zero. Third, we derive conditions under which we can prove boundedness of the CMHE estimation error, discuss the influence of various parameters on this bound and deduce actions to increase the estimation performance by decreasing this bound.

The remainder of this chapter is organized as follows. In Section 6.1, we state the problem formulation. The optimization algorithms are revisited in Section 6.2 and their treatment as dynamical systems is introduced. In Section 6.3, we give an outline of the stability analysis performed in Section 6.4 and Section 6.5 for the CMHO and CMHE, respectively. Finally, we conclude the chapter with a summary given in Section 6.6.

6.1 Problem Formulation

The problem formulations are stated as follows.

Problem 6.1 (Nominal CMHO Stability Analysis)**.** *Let the CMHO according to Chapter 3 be given for the undisturbed NCS architecture presented in Section 3.1.*

The problem is to analyze the stability of the nominal CMHO, i. e. to investigate the convergence behavior of the observation error $\boldsymbol{e}(t) \triangleq \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$.

Problem 6.2 (Nominal CMHE Stability Analysis)**.** *Let the CMHE according to Chapter 3 be given for the disturbed NCS architecture presented in Section 3.1.*

The problem is to analyze the stability of the nominal CMHE, i. e. to investigate the boundedness of the estimation error $\boldsymbol{e}(t) \triangleq \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$.

To ease the stability analysis and to increase the understandability, we suppress the arguments of all functions from the notation when the meaning is clear. It is important to recall that throughout Part II of this thesis including the current chapter, the *Assumptions 1-9 are supposed to hold without explicitly stating them.*

6.2 Gradient Based Optimization Algorithms Revisited

The centralized moving horizon strategies calculate their (sub)optimal solutions to the optimization problems stemming from the update steps by means of gradient-based optimization algorithms. According to Definition 3.4.3 respective 3.5.3, these optimization problems can be compactly written for the nominal choice, i. e. without inequality constraints, as

$$\min_{\hat{\boldsymbol{p}}_k} J_k(\hat{\boldsymbol{p}}_k). \tag{6.1}$$

As discussed in Section 2.4.2, there exists a variety of algorithms to tackle these type of unconstrained and, in general, non-convex problems. Taking into account each algorithm separately in the stability analysis of the CMHS would be possible but cumbersome and hence

unrewarding. Instead, we tread a different path by proposing a unifying formulation which includes all presented gradient-based optimization algorithms as special cases. As a result, only this unifying representation has to be considered in the overall stability analysis. To this end, we introduce the following unifying optimization algorithm representation formulated as the continuous-time dynamical system

$$\dot{\hat{\boldsymbol{p}}}_k(\kappa) \triangleq \frac{\partial \hat{\boldsymbol{p}}_k}{\partial \kappa}(\kappa) = -\boldsymbol{M}_k(\hat{\boldsymbol{p}}_k(\kappa))\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}(\hat{\boldsymbol{p}}_k(\kappa)), \tag{6.2}$$

where $\hat{\boldsymbol{p}}_k(\kappa) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$ is the state with the initial value $\hat{\boldsymbol{p}}_k(0) \in \mathbb{P} \subseteq \mathbb{R}^{n_p}$ abbreviated as $\hat{\boldsymbol{p}}_k^\circ$, $\boldsymbol{M}_k(\hat{\boldsymbol{p}}_k) : \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_p \times n_p}$ is a positive definite matrix function, $J_k(\hat{\boldsymbol{p}}_k(\kappa))$ is the cost function defined in (3.18a) and (3.24a) for the CMHO and CMHE, respectively, and $\kappa \in \mathbb{R}_0$ is the optimization time. Note that the optimization time is not a time in the proper sense and, hence, in no way related to either the global time t or the local sensor time $\bar{t}$. Instead, it should be seen as the counterpart to the iteration step l occurring in the gradient-based optimization algorithms. To see this fact and reveal the relation between (6.2) and the algorithms for unconstrained optimization problems presented in Section 2.4.2, we consider in a first step the discretization of (6.2) derived by the Euler forward method. The resulting discrete-time system can be expressed as

$$\hat{\boldsymbol{p}}_k[l+1] = \hat{\boldsymbol{p}}_k[l] - \Delta_k[l]\boldsymbol{M}_k[l]\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}[l], \tag{6.3}$$

where $\Delta_k[l] \in \mathbb{R}_0$ is the discretization step size and $l \in \mathbb{N}_0$ is the discrete-time. As the notation has already suggested, we understand the state $\hat{\boldsymbol{p}}_k$ as the optimization variable corresponding to the k-th buffer defined in (3.16) and (3.22) for the CMHO and CMHE, respectively.

Adapting the line search methods presented in Section 2.4.2.3 for solving (6.1) yields the iteration rule

$$\hat{\boldsymbol{p}}_k[l+1] = \hat{\boldsymbol{p}}_k[l] + \gamma_k[l]\boldsymbol{s}_k[l], \tag{6.4}$$

where $\gamma_k[l] \in \mathbb{R}_0$ is the step length, $\boldsymbol{s}_k[l] \in \mathbb{R}^{n_p}$ is the search direction and $l \in \mathbb{N}_0$ is the iteration step. Consequently, (6.3) coincides with (6.4), if we identify the discrete time l, the discretization step size $\Delta_k[l]$ and the matrix $\boldsymbol{M}_k[l]$ as the iteration step l, the step length $\gamma_k[l]$ and the inverse of the exact or approximated Hessian $\boldsymbol{B}_k[l]$ of the cost function J_k. The latter identification becomes clear if we recall that the search direction $\boldsymbol{s}_k[l]$ can be expressed as

$$\boldsymbol{s}_k[l] = -\boldsymbol{B}_k[l]^{-1}\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}[l]. \tag{6.5}$$

To reveal the relation between (6.4) and the trust region algorithms presented in Section 2.4.2.4, we need to derive an explicit expression for the update equation (2.54). By invoking the theory of constrained optimization, i. e. the KKT conditions of Theorem 2.4.7,

we can derive an explicit expression for the candidate step solving the constrained problem (2.52). Thus we can express the update equation of trust region methods as

$$\hat{\boldsymbol{p}}_k[l+1] = \hat{\boldsymbol{p}}_k[l] - (\boldsymbol{B}_k[l] + \lambda_k[l]\boldsymbol{I})^{-1}\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}[l], \tag{6.6}$$

where $\lambda_k \in \mathbb{R}_0$ is the Lagrange multiplier resulting from the trust region inequality and $\boldsymbol{B}_k[l]$ is the Hessian of the cost function J_k or some approximation to it. Consequently, (6.3) coincides with (6.6), if we set the discretization step size $\Delta_k[l]$ to 1 and identify the discrete time l and the matrix $\boldsymbol{M}_k[l]$ as the iteration step l and the expression $(\boldsymbol{B}_k[l] + \lambda_k[l]\boldsymbol{I})^{-1}$, respectively.

Figure 6.1 illustrates the above obtained relations.Note that the stability properties derived for (6.2) can be carried over to (6.3) to some extent. Furthermore, the introduced unifying formulation contains all the presented unconstrained optimization algorithms as special cases including the steepest descent, Newton's, quasi-Newton's, Gauss-Newton and the Levenberg Marquardt method. Moreover, the unifying representation opens up the possibility of incorporating the approximate Hessian calculation methods derived in Proposition 4.4.2 and 4.5.2 for the CMHO and CMHE, respectively, in the line search as well as the trust region framework. Note that according to (5.25), these approximations are guaranteed to be positive definite for the nominal choice of the update steps if the NCS is observable in the sense derived in Chapter 5 and the sensing model (3.3) is affine.

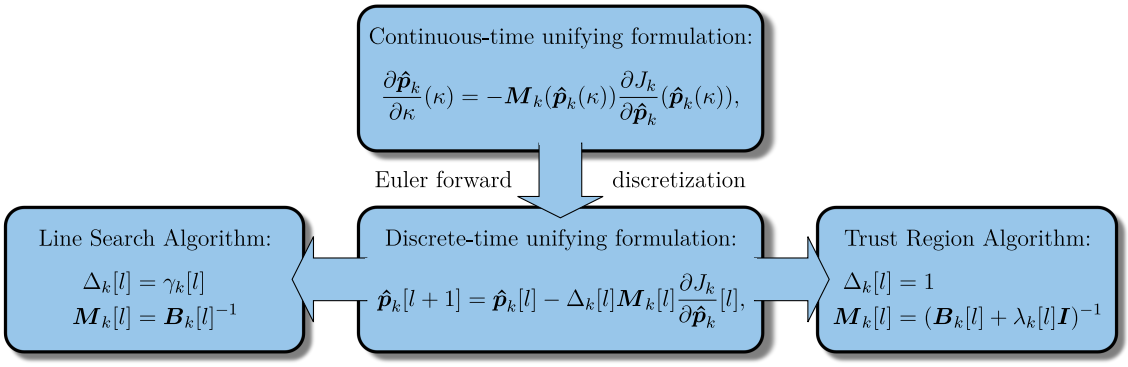


Figure 6.1: Illustration of the relation between the unifying formulation for the optimization algorithms, line search algorithms and trust region algorithms.

For the subsequent stability analysis, we define the function $\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$ as follows.

Definition 6.2.1. *The function* $\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$ *is a mapping* $\hat{\boldsymbol{\vartheta}} : \mathbb{R}_0 \times \mathbb{P} \times \mathbb{U} \mapsto \mathbb{R}^{n_p}$ *which satisfies*

$$i) \qquad \frac{\partial \hat{\boldsymbol{\vartheta}}}{\partial \kappa}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u}) = -\boldsymbol{M}_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})), \tag{6.7a}$$

$$ii) \qquad \hat{\boldsymbol{\vartheta}}(0, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u}) = \hat{\boldsymbol{p}}_k^\circ. \tag{6.7b}$$

6.3 Outline of the Stability Analysis

The CMHS stability analysis consists of three major steps which are outlined in the following. The first step is to investigate the equilibria of the system (6.2). We show that if the NCS is observable in the sense derived in Chapter 5, then the optimal solution $\hat{\boldsymbol{p}}_k^*$ of the optimization problem stemming from the update step of the k-th buffer is an isolated asymptotically stable equilibrium of the system (6.2). The corresponding proof is based on Lyapunov-type arguments utilizing a Lyapunov function V_k which is constructed by a suitable shift of the cost function J_k. The second step is to show that if the suboptimal approach satisfies the proposed decreasing condition, then the suboptimal solutions $\hat{\boldsymbol{p}}_k$ approach to the optimal solutions $\hat{\boldsymbol{p}}_k^*$ as k increases. Recall that this condition principally states that we switch to next system of the form (6.2), which triggers an impulsive change of its current state, whenever the cost function value J_k is less or equal than the previous cost function value J_{k-1} multiplied with the decreasing factor $\xi_k \in [0, 1[$. We illustrate the corresponding proof idea by means of the schematic illustration given in Figure 6.2. Thereby, we consider exemplary three systems of the form (6.2) corresponding to the $i-1$-th, i-th and $i+1$-th buffer. The equilibrium of each system is marked as a bottom-up triangle and the associated regions of attractions $\mathcal{R}_{i-1}$, $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$ are depicted as red, green and blue areas, respectively. The level sets of the cost functions J_{i-1}, J_i and J_{i+1} are plotted as red, green and blue dashed lines, respectively. Note that the stroke width of these lines correlates with the value of the level curves. The trajectory of each system is represented as an arrowed red, green and blue solid line starting from the corresponding initial values marked with a square. The proposed decreasing condition, which results in a switching procedure with impulsive state change, is depicted as yellow dashed arrows. This condition ensures that the suboptimal solutions $\hat{\boldsymbol{p}}_k$

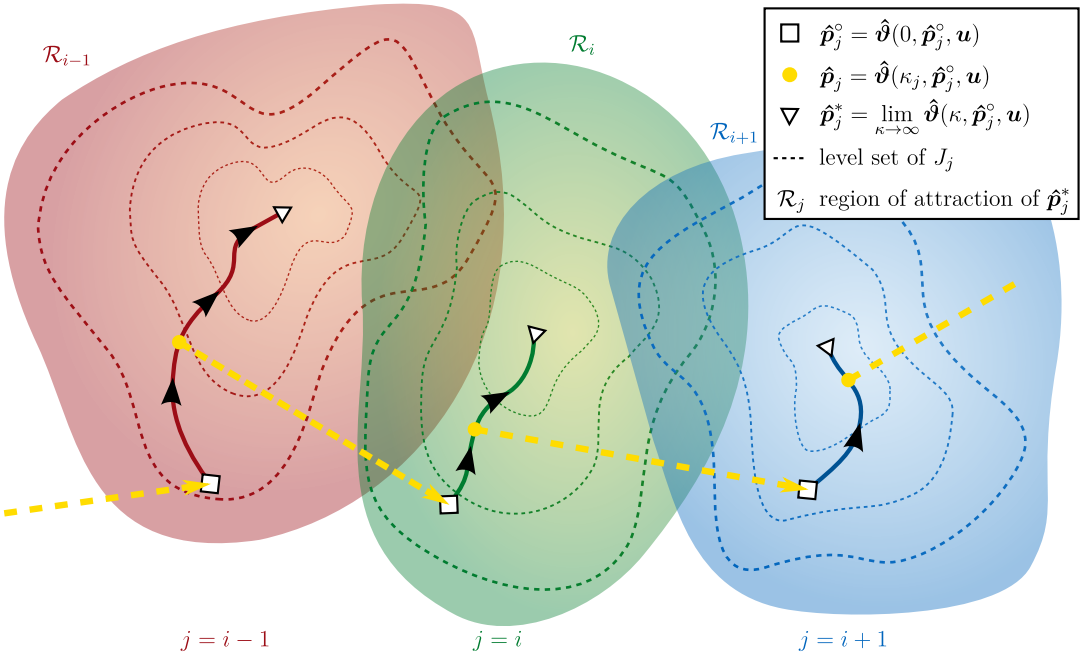


Figure 6.2: Schematic illustration of the stability analysis for the CMHS.

approach to the optimal solutions $\hat{\boldsymbol{p}}_k^*$ as k increases. In the third step, we conclude from this fact stability of the CMHS by invoking the continuous dependence of initial value problems on the initial conditions, times, parameters and vector fields.

6.4 Stability Analysis of the CMHO

In this section, we investigate the stability properties of the nominal CMHO. To this end, we consider the unconstrained case where the update step of the CMHO defined in (3.18) reads as

$$\min_{\hat{\boldsymbol{p}}_k} J_k(\hat{\boldsymbol{p}}_k) \tag{6.8a}$$

with the cost function

$$J_k(\hat{\boldsymbol{p}}_k) = \frac{1}{2} \sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) - \boldsymbol{y}_{i|k}\|^2. \tag{6.8b}$$

Then the associated gradient based optimization system (6.2) becomes with Lemma 4.2.1 and (5.24)

$$\begin{aligned} \dot{\hat{\boldsymbol{p}}}_k &= -\boldsymbol{M}_k \sum_{i=k-N}^{k} \left(\frac{\partial \boldsymbol{h}_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k} \right)^T \left(\boldsymbol{h}_{i|k} - \boldsymbol{y}_{i|k} \right) \\ &= -\boldsymbol{M}_k \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k}^T \left(\underline{\boldsymbol{h}}_k - \underline{\boldsymbol{y}}_k \right), \end{aligned} \tag{6.9}$$

where $\boldsymbol{M}_k$ is a positive definite matrix function, $\underline{\boldsymbol{h}}_k$ is the observation map defined in (5.1) and $\underline{\boldsymbol{y}}_k \triangleq \operatorname{col}(\boldsymbol{y}_{i|k}, i \in \mathcal{I}_k)$ is the stacked measurement vector of the buffer $\mathcal{B}_k$. Moreover, we denote the optimal solution of (6.8) as $\hat{\boldsymbol{p}}_k^* \triangleq \arg\min_{\hat{\boldsymbol{p}}_k} J_k(\hat{\boldsymbol{p}}_k)$. Note that $\hat{\boldsymbol{p}}_k^*$ exists due to the Weierstrass Maximum Theorem (Bronshtein et al., 2007). However, two important questions remain open. First, is $\hat{\boldsymbol{p}}_k^*$ unique and second, how is $\hat{\boldsymbol{p}}_k^*$ related to (6.9)? In order to address both questions, we make the following assumption.

Assumption 10. The undisturbed NCS described in Section 3.1 satisfies the extended observability rank condition in $N+1$ steps and the input $\boldsymbol{u}(t) \in \mathcal{U}$ is extended $N+1$-exciting for all sampling instances $k > N$.

This assumption is quite reasonable since it ensures the possibility of uniquely reconstructing $\hat{\boldsymbol{p}}_k$, see Section 5.2. Moreover, if the sensing model (3.3) is affine, this assumption guarantees for the nominal case positive definiteness of the Hessian calculated by the approximation scheme given in Proposition 4.4.2. With this assumption, we can derive the following result.

Theorem 6.4.1. *Suppose that the Assumption 10 holds. Then the optimal solution $\hat{\boldsymbol{p}}_k^*$ of (6.8) is an isolated asymptotically stable equilibrium of (6.9).*

Proof. Since we consider only a fixed buffer $\mathcal{B}_k$, we ease the proof by omitting the subscripts indicating the buffer affiliation. To prove the theorem, we use the cost function $J(\hat{\boldsymbol{p}})$ of (6.8b) as a Lyapunov function candidate $V(\hat{\boldsymbol{p}})$ for the system (6.9). The derivative of $V(\hat{\boldsymbol{p}})$ along the trajectories of (6.9) is given by

$$\dot{V}(\hat{\boldsymbol{p}}) = -\frac{\partial J}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}})\boldsymbol{M}(\hat{\boldsymbol{p}})\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}). \tag{6.10}$$

By construction, $V(\hat{\boldsymbol{p}})$ and $\dot{V}(\hat{\boldsymbol{p}})$ are positive semi-definite and negative semi-definite around $\hat{\boldsymbol{p}}^*$, respectively. To show that $V(\hat{\boldsymbol{p}})$ and $\dot{V}(\hat{\boldsymbol{p}})$ are positive definite and negative definite around $\hat{\boldsymbol{p}}^*$, respectively, we need to examine the gradient of $J(\hat{\boldsymbol{p}})$, which is given by

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \left(\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}) - \underline{\boldsymbol{y}}\right).$$

Application of the Mean Value Theorem A.2 to $\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})$ results in

$$\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}) = \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}^*).$$

Since $\underline{\boldsymbol{y}} = \underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}^*)$, it follows

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*),$$

which can be rearranged to yield

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = \boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \boldsymbol{a}(\hat{\boldsymbol{p}}), \tag{6.11a}$$

with

$$\boldsymbol{A}(\hat{\boldsymbol{p}}) \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}), \tag{6.11b}$$

$$\boldsymbol{a}(\hat{\boldsymbol{p}}) \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}}) - \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*). \tag{6.11c}$$

The function $\boldsymbol{a}(\hat{\boldsymbol{p}})$ satisfies

$$\|\boldsymbol{a}(\hat{\boldsymbol{p}})\| \leq \left(\sup_{0\leq s\leq 1} \left\| \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \left(\frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}}) - \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}})\right) \right\|\right) \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|.$$

By continuity of $\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})$ and $\partial \underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})/\partial \hat{\boldsymbol{p}}$, we see that

$$\frac{\|\boldsymbol{a}(\hat{\boldsymbol{p}})\|}{\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|} \to 0 \quad \text{as} \quad \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| \to 0.$$

Therefore, for any $\gamma > 0$, there exists $r > 0$ such that

$$\|\boldsymbol{a}(\hat{\boldsymbol{p}})\| < \gamma\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|, \quad \forall\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r. \tag{6.12}$$

By inserting (6.11) into (6.10), the derivative of $V(\hat{\boldsymbol{p}})$ along the trajectories of (6.9) is

$$\begin{aligned}
\dot{V}(\hat{\boldsymbol{p}}) &= -\frac{\partial J}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}})\boldsymbol{M}(\hat{\boldsymbol{p}})\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) \\
&= -\left(\boldsymbol{a}(\hat{\boldsymbol{p}})^T + (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T\boldsymbol{A}(\hat{\boldsymbol{p}})^T\right)\boldsymbol{M}(\hat{\boldsymbol{p}})\Big(\boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \boldsymbol{a}(\hat{\boldsymbol{p}})\Big) \\
&= -(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T\boldsymbol{A}(\hat{\boldsymbol{p}})^T\boldsymbol{M}(\hat{\boldsymbol{p}})\boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) - 2\boldsymbol{a}(\hat{\boldsymbol{p}})^T\boldsymbol{M}(\hat{\boldsymbol{p}})\boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) \\
&\quad - \boldsymbol{a}(\hat{\boldsymbol{p}})^T\boldsymbol{M}(\hat{\boldsymbol{p}})\boldsymbol{a}(\hat{\boldsymbol{p}}).
\end{aligned}$$

Using (6.12) and suppressing henceforth the argument $\hat{\boldsymbol{p}}$ from the notation, we get

$$\dot{V} < -(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + 2\gamma\delta\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|^2 - \boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a}, \quad \forall\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r$$

where

$$\delta \triangleq \sup_{\hat{\boldsymbol{p}}} \|\boldsymbol{M}(\hat{\boldsymbol{p}})\boldsymbol{A}(\hat{\boldsymbol{p}})\|.$$

But

$$(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) \geq \varrho\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|^2,$$

where

$$\varrho \triangleq \min_{\hat{\boldsymbol{p}}} \lambda_{\min}\left(\boldsymbol{A}(\hat{\boldsymbol{p}})^T\boldsymbol{M}(\hat{\boldsymbol{p}})\boldsymbol{A}(\hat{\boldsymbol{p}})\right)$$

with $\lambda_{\min}(\cdot)$ denoting the minimum eigenvalue of a matrix. Thus,

$$\dot{V} < -(\varrho - 2\gamma\delta)\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|^2 - \boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a}, \quad \forall\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r.$$

Note that ϱ is real and nonnegative since $\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}$ is symmetric and nonnegative. Moreover, since $\boldsymbol{M} > \boldsymbol{0}$, ϱ is positive if and only if $\boldsymbol{A}$ has full rank. Assumption 10 guarantees full-rankness of $\partial\underline{\boldsymbol{h}}/\partial\hat{\boldsymbol{p}}$ and therefore full-rankness of $\boldsymbol{A}$ and thus $\varrho > 0$. Then choosing

$$\gamma < \frac{\varrho}{2\delta}$$

ensures that $\dot{V}(\hat{\boldsymbol{p}})$ is negative definite around $\hat{\boldsymbol{p}}^*$ since $-\boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a}$ is negative semi-definite around $\hat{\boldsymbol{p}}^*$. This implies that $\partial J(\hat{\boldsymbol{p}})/\partial\hat{\boldsymbol{p}} \neq \boldsymbol{0}$ for all $\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r, \hat{\boldsymbol{p}} \neq \hat{\boldsymbol{p}}^*$. Therefore, $V(\hat{\boldsymbol{p}})$ is positive definite around $\hat{\boldsymbol{p}}^*$. From application of Theorem A.1 follows that $\hat{\boldsymbol{p}}^*$ is an isolated asymptotically stable equilibrium of (6.9). □

This theorem guarantees that we can find the optimal solution $\hat{\boldsymbol{p}}_k^*$ by a forward simulation of (6.9), provided that the initial value $\hat{\boldsymbol{p}}_k^\circ$ is inside the region of attraction of $\hat{\boldsymbol{p}}_k^*$. Thus, we make the following assumption.

Assumption 11. The initial value $\hat{\boldsymbol{p}}_k^\circ$ is bounded and inside the region of attraction of the isolated asymptotically stable equilibrium $\hat{\boldsymbol{p}}_k^*$ of system (6.9), i. e.

$$\hat{\boldsymbol{p}}_k^\circ \in \mathcal{R}_k \triangleq \{\hat{\boldsymbol{p}}_k^\circ \in \mathbb{R}^{n_p} | \lim_{\kappa\to\infty} \hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u}) = \hat{\boldsymbol{p}}_k^*\}, \quad \forall k \in \mathbb{N}_{N+1}.$$

Note that assumptions of this kind are quite typical for gradient-based algorithms used for solving non-convex optimization problems. It is well known that these algorithms only converge to the global optimal value, if the initial value is close enough to the optimal one, cf. Section 2.4. Thereby, the meaning of close enough is relative and depends on the specific problem. For instance, close enough means for convex optimization problems arbitrary far away while no generally valid statements are possible for unconstrained non-convex problems.

The next question results from the fact that we are interested in a suboptimal solution to (6.8) rather than in the optimal one. This means that the forward simulation of (6.9) is stopped as soon as the decreasing condition (3.15) is fulfilled. Consequently, the arising question is if we can make any statements about the observation error $\boldsymbol{e}(t) \triangleq \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ when we stop prematurely the forward simulation of (6.9) at time κ_k. By means of the following bounded quantities

$$L \triangleq \max_{\boldsymbol{x}\in\mathbb{X},\boldsymbol{u}\in\mathbb{U}} \left\| \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}} \right\|, \quad \delta_f \triangleq \max_{\boldsymbol{x}\in\mathbb{X},\boldsymbol{u}\in\mathbb{U}} \|\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})\|, \tag{6.13}$$

we can give the answer to this question in the following Lemma.

Lemma 6.4.2. *Suppose that Assumptions 10 and 11 hold and that* $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{\vartheta}}(\kappa_k, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$. *Then the norm of the observation error is bounded in the prediction interval* $t \in \left[\max_{j\in\mathcal{I}_k} t_{j|k}, \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}\right]$ *for any* $k \in \mathbb{N}_{N+1}$, $\kappa_k \in \mathbb{R}_0$, $\hat{\boldsymbol{p}}_k^\circ \in \mathcal{R}_k$ *and any admissible* $\boldsymbol{u} \in \mathbb{U}$ *by*

$$\|\boldsymbol{e}(t)\| \leq \left(\|\hat{\boldsymbol{x}}_{k-N|k} - \boldsymbol{x}_{k-N|k}\| + |(\hat{\alpha}_k - \alpha_k)\bar{t}_{k-N|k} + (\hat{\beta}_k - \beta_k)|\delta_f\right) e^{L(t-\hat{\alpha}_k\,\bar{t}_{k-N|k}-\hat{\beta}_k)}. \tag{6.14}$$

Proof. The following holds for all $k \in \mathbb{N}_{N+1}$: Application of Theorem A.4 with $\boldsymbol{y} = \boldsymbol{x}$, $\boldsymbol{z} = \hat{\boldsymbol{x}}$, $\boldsymbol{g}(t) = \boldsymbol{h}(t) = \boldsymbol{0}$, $\boldsymbol{y}_0 = \boldsymbol{x}_{k-N|k}$, $\boldsymbol{z}_0 = \hat{\boldsymbol{x}}_{k-N|k}$, $t_0 = \hat{\alpha}_k\,\bar{t}_{k-N|k} + \hat{\beta}_k$, $t_1 = \alpha_k\,\bar{t}_{k-N|k} + \beta_k$, $t_2 = \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}$, $\delta = \delta_f$ and the Lipschitz constant L reveals the desired result. □

Recall that the cumbersome formulation of the prediction interval is necessary since not every arrived packet is incorporated in the buffer and thus the arrival time stamps t_j cannot be used directly to define the prediction interval, cf. Section 3.7.

Based on the findings so far, we can state the following stability result for the CMHO.

Theorem 6.4.3. *Let the Assumptions 10 and 11 hold. Then the decreasing condition*

$$J_k(\hat{\boldsymbol{p}}_k) \leq \xi_k J_{k-1}(\hat{\boldsymbol{p}}_{k-1}) \tag{6.15}$$

with the decreasing factor $\xi_k \in [0, 1[$ *and the initial cost function value* $J_N(\hat{\boldsymbol{p}}_N) \triangleq J_{N+1}(\hat{\boldsymbol{p}}^\circ_{N+1})$ *is feasible by the optimization algorithm (6.9) for any* $k \in \mathbb{N}_{N+1}$ *and guarantees the CMHO convergence of the observed parameters, i. e.*

$$\lim_{k\to\infty} \hat{\boldsymbol{p}}_k = \boldsymbol{p}_k, \tag{6.16}$$

and convergence of the observation error, i. e.

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0}. \tag{6.17}$$

Proof. The following holds for all $k \in \mathbb{N}_{N+1}$: Assumption 10 implies that Theorem 6.4.1 holds, i. e. $\hat{\boldsymbol{p}}^*_k$ is an isolated asymptotically stable equilibrium of (6.9). Assumption 11 guarantees that the initial value $\hat{\boldsymbol{p}}^\circ_k$ is bounded and inside the region of attraction $\mathcal{R}_k$ of the equilibrium $\hat{\boldsymbol{p}}^*_k$. Since $\hat{\boldsymbol{p}}^*_k$ is the only equilibrium of (6.9) in $\mathcal{R}_k$ and $\boldsymbol{M}_k > \boldsymbol{0}$, it follows from (6.9) that

$$\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k} \neq \boldsymbol{0}, \quad \forall \hat{\boldsymbol{p}}_k \in \mathcal{R}_k \setminus \{\hat{\boldsymbol{p}}^*_k\}.$$

Thus, $J_k(\hat{\boldsymbol{p}}_k)$ is a Lyapunov function for system (6.9) which is positive definite around $\hat{\boldsymbol{p}}^*_k$ for all $\hat{\boldsymbol{p}}_k \in \mathcal{R}_k$. Note that $J_k(\hat{\boldsymbol{p}}_k)$ is zero in $\mathcal{R}_k$ if and only if $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}^*_k$. The derivative of $J_k(\hat{\boldsymbol{p}}_k)$ along the trajectories of (6.9) is

$$\dot{J}_k(\hat{\boldsymbol{p}}_k) = -\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}^T(\hat{\boldsymbol{p}}_k)\boldsymbol{M}_k(\hat{\boldsymbol{p}}_k)\frac{\partial J}{\partial \hat{\boldsymbol{p}}_k}(\hat{\boldsymbol{p}}_k)$$

and is negative definite around $\hat{\boldsymbol{p}}^*_k$ for all $\hat{\boldsymbol{p}}_k \in \mathcal{R}_k$, since $\boldsymbol{M}_k(\hat{\boldsymbol{p}}_k) > 0$. Therefore, the cost function $J_k(\hat{\boldsymbol{p}}_k)$ with $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}^\circ_k, \boldsymbol{u})$ converges to zero as the optimization time κ goes to infinity, i. e.

$$\lim_{\kappa\to\infty} J_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}^\circ_k, \boldsymbol{u})) = 0.$$

Hence, for any $\xi_k \in [0, 1[$, there always exists an optimization time κ_k and an associated optimization variable $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{\vartheta}}(\kappa_k, \hat{\boldsymbol{p}}^\circ_k, \boldsymbol{u})$ such that the decreasing condition (6.15) is satisfied, i. e. $J_k(\hat{\boldsymbol{p}}_k) \leq \xi_k J_{k-1}(\hat{\boldsymbol{p}}_{k-1})$. Furthermore, the only limit point of the resulting sequence of Lyapunov function values $\{J_k(\hat{\boldsymbol{p}}_k)\}$ satisfying the decreasing condition (6.15) is 0. Thus,

$$\lim_{k\to\infty} J_k(\hat{\boldsymbol{p}}_k) = 0,$$

which implies

$$\lim_{k\to\infty} \hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}^*_k.$$

Since the Assumptions 10 and 11 hold, it follows from Lemma 6.4.2 that the observation error $\boldsymbol{e}(t)$ converges to zero if and only if $\hat{\boldsymbol{p}}_k$ converges to $\boldsymbol{p}_k$. Hence,

$$\lim_{t\to\infty} \boldsymbol{e}(t) = \boldsymbol{0},$$

because $\hat{\boldsymbol{p}}_k^* = \boldsymbol{p}_k$. □

The importance of this theorem is substantiated among other things by the following three facts. First of all, this theorem holds for any network which satisfies the quite general Assumptions 5-7. This means that the stability of the nominal CMHO is independent of the packet delay and packet drop statistics and requires only their boundedness. Note that these statistics may be time-varying and do not need to be known at all.

Second, due to the utilized unifying formulation (6.9), all optimization algorithms are permissible which utilize the exact gradient and a positive-definite matrix $\boldsymbol{M}_k$. This includes not only all the algorithms presented in Section 2.4.2 but also line search and trust region methods which utilize the Hessian approximation derived in Proposition 4.4.2. The difference between the algorithms stems from differing vectors fields of (6.9) which results in varying trajectories and different regions of attractions.

Third, instead of the optimal approach, the proposed suboptimal one is sufficient to guarantee stability of the nominal CMHO. This suboptimal concept is straightforward to implement and requires only the choice of the decreasing factors ξ_k. These allow to set up a compromise between the achievable convergence speed and the required computation time. From a performance point of view, these factors should be chosen as small as possible, while from a real-time point of view, these factors should be selected as close as possible to 1.

One may notice that by choosing $\xi_j = 0$, we can instantly get $J_j = 0$. Hence, the following corollary stems directly from Theorem 6.4.3.

Corollary 6.4.4. *Let the Assumptions 10 and 11 hold. If the j-th decreasing factor is chosen as zero, i. e. $\xi_j = 0$, $j \in \mathbb{N}_{N+1}$, then the observed parameters converge in finite time, i. e.*

$$\hat{\boldsymbol{p}}_k = \boldsymbol{p}_k, \quad \forall k \geq j, \quad j \in \mathbb{N}_{N+1} \tag{6.18}$$

and the observation error converges in finite time, i. e.

$$\boldsymbol{e}(t) = \boldsymbol{0}, \quad \forall t \geq \max_{i\in\mathcal{I}_j} t_{i|j}, \quad j \in \mathbb{N}_{N+1}. \tag{6.19}$$

Now we utilize the presented stability analysis to discuss the to be expected CMHO performance by means of the schematic illustration given in Figure 6.3. Without loss of generality, we consider the case where the arrival time stamps can be directly used for defining the prediction intervals. The initial cost function values $J_i(\hat{\boldsymbol{p}}_i^\circ)$ and the suboptimal cost function values $J_i(\hat{\boldsymbol{p}}_i)$ are marked with blue circles and blue squares at the corresponding arrival times t_i, respectively. The suboptimal cost function values $J_i(\hat{\boldsymbol{p}}_i)$ are identical to the Lyapunov function

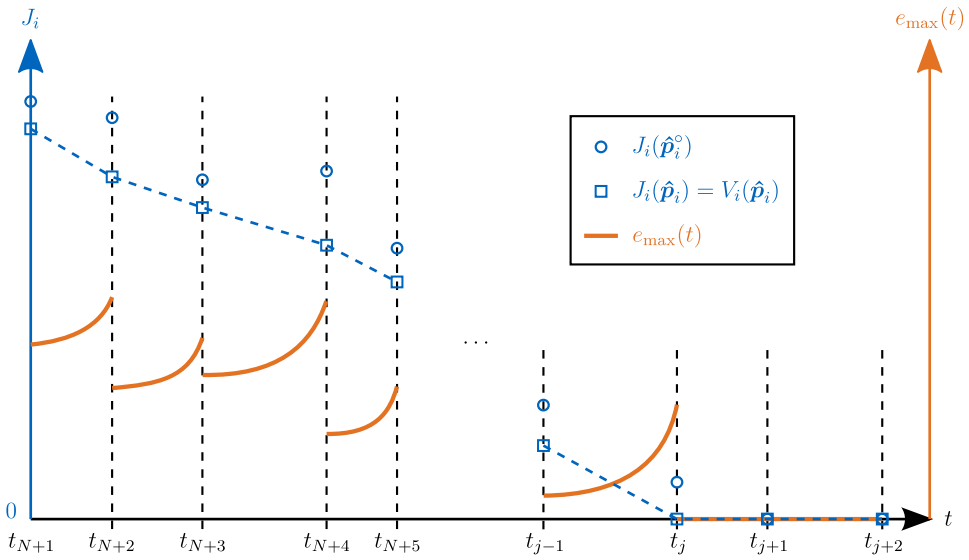


Figure 6.3: Schematic illustration of the CMHO performance.

values $V_i(\hat{\boldsymbol{p}}_i)$ and are connected by a blue dashed line to highlight the effect of the decreasing condition (6.15) which guarantees a steady decline of $J_i(\hat{\boldsymbol{p}}_i)$. The resulting observation error bound $e_{\max}(t)$ derived in (6.14) is depicted as an orange solid line. Note that this bound is conservative, i. e. it represents only the worst case scenario. Moreover, it jumps at the beginning of each prediction interval due to the new information available synthesized in the suboptimal solution to the optimization problem of each update step. From the j-th arrival time stamp onwards, $J_i(\hat{\boldsymbol{p}}_i)$ as well as $e_{\max}(t)$ are vanishing due to the zero choice of the j-th decreasing factor, i. e. $\xi_j = 0$. Note that this implies that not only the state but also the clock parameters are reconstructed exactly. It should be stressed that the presented results hold for all networks which feature bounded packet delay and packet drop statistics.

Remark 6.4.1. Since the network-free MHO is included as a special case of the CMHO, the derived stability results hold for the network-free scenario, i. e. for the MHO.

6.5 Stability Analysis of the CMHE

In this section, we investigate the stability properties of the nominal CMHE. To this end, we consider the unconstrained case where the update step of the CMHE defined in (3.24) is

$$\min_{\hat{\boldsymbol{p}}_k} J_k(\hat{\boldsymbol{p}}_k) \tag{6.20a}$$

with the cost function

$$J_k(\hat{\boldsymbol{p}}_k) = \frac{1}{2}\left\| \begin{bmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k} \end{bmatrix} - \begin{bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \\ \bar{\boldsymbol{x}}_{k-N|k} \end{bmatrix} \right\|^2_{\boldsymbol{P}_k^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) - \boldsymbol{y}_{i|k}\|^2_{\boldsymbol{R}^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_{i|k}\|^2_{\boldsymbol{Q}^{-1}}. \tag{6.20b}$$

Then the associated gradient based optimization system (6.2) becomes with Lemma 4.3.1 and (5.24)

$$\begin{aligned}\dot{\hat{\boldsymbol{p}}}_k = -\boldsymbol{M}_k \Bigg(& \boldsymbol{P}_k^{-1} \left(\begin{bmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k} \end{bmatrix} - \begin{bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \\ \bar{\boldsymbol{x}}_{k-N|k} \end{bmatrix} \right) + \sum_{i=k-N}^{k} \left(\frac{\partial \boldsymbol{h}_{i|k}}{\partial \hat{\boldsymbol{\phi}}_{i|k}} \frac{\partial \hat{\boldsymbol{\phi}}_{i|k}}{\partial \hat{\boldsymbol{p}}_k} \right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{h}_{i|k} - \boldsymbol{y}_{i|k} \right) \\ & + \sum_{i=k-N}^{k-1} \boldsymbol{Q}^{-1} \hat{\boldsymbol{w}}_{i|k} \Bigg) \\ = -\boldsymbol{M}_k \Bigg(& \begin{bmatrix} \boldsymbol{P}_k^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} \left(\hat{\boldsymbol{p}}_k - \begin{bmatrix} \bar{\boldsymbol{p}}_k \\ \boldsymbol{0} \end{bmatrix} \right) + \frac{\partial \underline{\boldsymbol{h}}_k}{\partial \hat{\boldsymbol{p}}_k}^T \underline{\boldsymbol{R}}^{-1} \left(\underline{\boldsymbol{h}}_k - \underline{\boldsymbol{y}}_k \right) \Bigg), \end{aligned} \tag{6.21}$$

where $\boldsymbol{M}_k$ is a positive definite matrix function, $\bar{\boldsymbol{p}}_k \triangleq \operatorname{col}(\bar{\alpha}_k, \bar{\beta}_k, \bar{\boldsymbol{x}}_{k-N})$ is the stacked prediction vector, $\underline{\boldsymbol{R}}^{-1} \triangleq \operatorname{diag}(\boldsymbol{R}^{-1}, \ldots, \boldsymbol{R}^{-1})$ and $\underline{\boldsymbol{Q}}^{-1} \triangleq \operatorname{diag}(\boldsymbol{Q}^{-1}, \ldots, \boldsymbol{Q}^{-1})$ are stacked weighting matrices, $\underline{\boldsymbol{h}}_k$ is the observation map defined in (5.10) and $\underline{\boldsymbol{y}}_k \triangleq \operatorname{col}(\boldsymbol{y}_{i|k}, i \in \mathcal{I}_k)$ is the stacked measurement vector of the buffer $\mathcal{B}_k$. Moreover, we denote the optimal solution of (6.20) as $\hat{\boldsymbol{p}}_k^* \triangleq \arg\min_{\hat{\boldsymbol{p}}_k} J_k(\hat{\boldsymbol{p}}_k)$. Note that $\hat{\boldsymbol{p}}_k^*$ exists due to the Weierstrass Maximum Theorem (Bronshtein et al., 2007). However, just like in the CMHO case, two important questions remain open. First, is $\hat{\boldsymbol{p}}_k^*$ unique and second, how is $\hat{\boldsymbol{p}}_k^*$ related to (6.9)? In order to address both questions, we make the following assumptions.

Assumption 12. The matrices $\boldsymbol{P}_k^{-1}$ are $\forall k \in \mathbb{N}_{N+1}$ positive semi-definite, i. e. $\boldsymbol{P}_k^{-1} \geq \boldsymbol{0}$, and the matrices $\underline{\boldsymbol{R}}^{-1}$ and $\underline{\boldsymbol{Q}}^{-1}$ are positive definite, i. e. $\underline{\boldsymbol{R}}^{-1} > \boldsymbol{0}$ and $\underline{\boldsymbol{Q}}^{-1} > \boldsymbol{0}$.

Assumption 13. The disturbed NCS described in Section 3.1 satisfies the extended observability rank condition in $N+1$ steps and the input $\boldsymbol{u}(t) \in \mathcal{U}$ is extended $N+1$-exciting for all sampling instances $k \in \mathbb{N}_{N+1}$.

Assumption 12 restricts the weighting matrices to reasonable values. Assumption 13 is the counterpart of Assumption 10 for the disturbed case and is quite reasonable since it ensures the possibility of uniquely reconstructing $\hat{\boldsymbol{p}}_k$, see Section 5.3. If the sensing model (3.3) is affine, both assumptions together guarantee for the nominal case positive definiteness of the Hessian calculated by the approximation scheme given in Proposition 4.5.2. With these assumptions, we can derive the following result.

Theorem 6.5.1. *Suppose that the Assumptions 12 and 13 hold. Then the optimal solution $\hat{\boldsymbol{p}}_k^*$ of (6.20) is an isolated asymptotically stable equilibrium of (6.21).*

Proof. Since we consider only a fixed buffer $\mathcal{B}_k$, we ease the proof by omitting the subscripts indicating the buffer affiliation. To prove the theorem, we use $V(\hat{\boldsymbol{p}}) = J(\hat{\boldsymbol{p}}) - J(\hat{\boldsymbol{p}}^*)$ as a Lyapunov function candidate for (6.21). The derivative of $V(\hat{\boldsymbol{p}})$ along the trajectories of (6.21) is given by

$$\dot{V}(\hat{\boldsymbol{p}}) = -\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}). \tag{6.22}$$

By construction, $V(\hat{\boldsymbol{p}})$ and $\dot{V}(\hat{\boldsymbol{p}})$ are positive semi-definite and negative semi-definite around $\hat{\boldsymbol{p}}^*$, respectively. To show that $V(\hat{\boldsymbol{p}})$ and $\dot{V}(\hat{\boldsymbol{p}})$ are positive definite and negative definite around $\hat{\boldsymbol{p}}^*$, respectively, we need to examine the gradient of $J(\hat{\boldsymbol{p}})$, which is given by

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = \begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} \left(\hat{\boldsymbol{p}} - \begin{bmatrix} \bar{\boldsymbol{p}} \\ \boldsymbol{0} \end{bmatrix} \right) + \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \left(\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}) - \underline{\boldsymbol{y}} \right).$$

Application of the Mean Value Theorem A.2 to $\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})$ results in

$$\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}) = \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}^*).$$

Hence,

$$\begin{aligned} \frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = &\left(\begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} + \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds \right) (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) \\ &+ \begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} \left(\hat{\boldsymbol{p}}^* - \begin{bmatrix} \bar{\boldsymbol{p}} \\ \boldsymbol{0} \end{bmatrix} \right) + \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \left(\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}^*) - \underline{\boldsymbol{y}} \right), \end{aligned}$$

which can be rearranged to yield

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) = \boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \boldsymbol{a}_1(\hat{\boldsymbol{p}}) + \boldsymbol{a}_2(\hat{\boldsymbol{p}}), \tag{6.23a}$$

with

$$\boldsymbol{A}(\hat{\boldsymbol{p}}) \triangleq \begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} + \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}), \tag{6.23b}$$

$$\boldsymbol{a}_1(\hat{\boldsymbol{p}}) \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \int_0^1 \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}}) - \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*), \tag{6.23c}$$

$$\boldsymbol{a}_2(\hat{\boldsymbol{p}}) \triangleq \begin{bmatrix} \boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1} \end{bmatrix} \left(\hat{\boldsymbol{p}}^* - \begin{bmatrix} \bar{\boldsymbol{p}} \\ \boldsymbol{0} \end{bmatrix} \right) + \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \left(\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}}^*) - \underline{\boldsymbol{y}} \right). \tag{6.23d}$$

Since $\hat{\boldsymbol{p}}^*$ is an optimal solution of (6.20), we know that

$$\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}^*) = \boldsymbol{0}.$$

Thus, by inspection of (6.23), both functions $\boldsymbol{a}_1(\hat{\boldsymbol{p}})$ and $\boldsymbol{a}_2(\hat{\boldsymbol{p}})$ satisfy

$$\|\boldsymbol{a}_i(\hat{\boldsymbol{p}})\| \to 0 \quad \text{as} \quad \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| \to 0, \quad i = 1, 2.$$

Moreover, application of the Mean Value Theorem A.2 to $\boldsymbol{a}_2(\hat{\boldsymbol{p}})$ results in

$$\boldsymbol{a}_2(\hat{\boldsymbol{p}}) = \int_0^1 \frac{\partial \boldsymbol{a}_2}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \boldsymbol{a}_2(\hat{\boldsymbol{p}}^*)$$

$$= \int_0^1 \frac{\partial \boldsymbol{a}_2}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}})\, ds\, (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*),$$

and the function $\boldsymbol{a}(\hat{\boldsymbol{p}}) \triangleq \boldsymbol{a}_1(\hat{\boldsymbol{p}}) + \boldsymbol{a}_2(\hat{\boldsymbol{p}})$ satisfies

$$\|\boldsymbol{a}(\hat{\boldsymbol{p}})\| \leq \left(\sup_{0\leq s\leq 1} \left\| \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}^T(\hat{\boldsymbol{p}}) \underline{\boldsymbol{R}}^{-1} \left(\frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}}) - \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) \right) + \frac{\partial \boldsymbol{a}_2}{\partial \hat{\boldsymbol{p}}}((1-s)\hat{\boldsymbol{p}}^* + s\hat{\boldsymbol{p}}) \right\| \right) \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|.$$

By continuity of $\underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})$ and $\partial \underline{\boldsymbol{h}}(\hat{\boldsymbol{p}})/\partial \hat{\boldsymbol{p}}$, we see that

$$\frac{\|\boldsymbol{a}(\hat{\boldsymbol{p}})\|}{\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|} \to 0 \quad \text{as} \quad \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| \to 0.$$

Therefore, for any $\gamma > 0$, there exists $r > 0$ such that

$$\|\boldsymbol{a}(\hat{\boldsymbol{p}})\| < \gamma \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|, \quad \forall \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r. \tag{6.24}$$

By inserting (6.23) into (6.22), the derivative of $V(\hat{\boldsymbol{p}})$ along the trajectories of (6.21) is

$$\begin{aligned}
\dot{V}(\hat{\boldsymbol{p}}) &= -\frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \frac{\partial J}{\partial \hat{\boldsymbol{p}}}(\hat{\boldsymbol{p}}) \\
&= -\left(\boldsymbol{a}(\hat{\boldsymbol{p}})^T + (\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T \boldsymbol{A}(\hat{\boldsymbol{p}})^T \right) \boldsymbol{M}(\hat{\boldsymbol{p}}) \left(\boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + \boldsymbol{a}(\hat{\boldsymbol{p}}) \right) \\
&= -(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T \boldsymbol{A}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) - 2\boldsymbol{a}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \boldsymbol{A}(\hat{\boldsymbol{p}})(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) \\
&\quad - \boldsymbol{a}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \boldsymbol{a}(\hat{\boldsymbol{p}}).
\end{aligned}$$

Using (6.24) and suppressing henceforth the argument $\hat{\boldsymbol{p}}$ from the notation, we get

$$\dot{V}(\hat{\boldsymbol{p}}) < -(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T \boldsymbol{A}^T \boldsymbol{M} \boldsymbol{A}(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) + 2\gamma\delta \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|^2 - \boldsymbol{a}^T \boldsymbol{M} \boldsymbol{a}, \quad \forall \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r,$$

where

$$\delta \triangleq \sup_{\hat{\boldsymbol{p}}} \|\boldsymbol{M}(\hat{\boldsymbol{p}}) \boldsymbol{A}(\hat{\boldsymbol{p}})\|.$$

But

$$(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*)^T \boldsymbol{A}^T \boldsymbol{M} \boldsymbol{A}(\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*) \geq \varrho \|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|^2,$$

where

$$\varrho \triangleq \min_{\hat{\boldsymbol{p}}} \lambda_{\min} \left(\boldsymbol{A}(\hat{\boldsymbol{p}})^T \boldsymbol{M}(\hat{\boldsymbol{p}}) \boldsymbol{A}(\hat{\boldsymbol{p}}) \right)$$

with $\lambda_{\min}(\cdot)$ denoting the minimum eigenvalue of a matrix. Note that ϱ is real and nonnegative

since $\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}$ is symmetric and nonnegative. Thus,

$$\dot{V} < -(\varrho - 2\gamma\delta)\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\|_2^2 - \boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a}, \quad \forall\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r.$$

Suppose for the moment that $\varrho > 0$ holds. Then choosing

$$\gamma < \frac{\varrho}{2\delta}$$

ensures that $\dot{V}(\hat{\boldsymbol{p}})$ is negative definite around $\hat{\boldsymbol{p}}^*$ since $-\boldsymbol{a}^T\boldsymbol{M}\boldsymbol{a}$ is negative semi-definite around $\hat{\boldsymbol{p}}^*$. This implies that $\partial J(\hat{\boldsymbol{p}})/\partial\hat{\boldsymbol{p}} \neq \boldsymbol{0}$ for all $\|\hat{\boldsymbol{p}} - \hat{\boldsymbol{p}}^*\| < r, \hat{\boldsymbol{p}} \neq \hat{\boldsymbol{p}}^*$. Therefore, $V(\hat{\boldsymbol{p}})$ is positive definite around $\hat{\boldsymbol{p}}^*$. From application of Theorem A.1 follows that $\hat{\boldsymbol{p}}^*$ is an isolated asymptotically stable equilibrium of (6.21). However, this requires that $\varrho > 0$, which holds if and only if $\boldsymbol{A}$ has full rank for all admissible $\boldsymbol{P}^{-1}$, $\underline{\boldsymbol{Q}}^{-1}$ and $\underline{\boldsymbol{R}}^{-1}$. By introducing the abbreviation

$$\boldsymbol{O} = \begin{bmatrix}\boldsymbol{O}_1 \\ \boldsymbol{O}_2\end{bmatrix} \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \hat{\boldsymbol{p}}}, \text{ where } \boldsymbol{O}_1 \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \operatorname{col}(\hat{\alpha}, \hat{\beta}, \hat{\boldsymbol{x}}_{k-N})}, \boldsymbol{O}_2 \triangleq \frac{\partial \underline{\boldsymbol{h}}}{\partial \operatorname{col}(\hat{\boldsymbol{w}}_i, i \in \overline{\mathcal{I}})},$$

the matrix $\boldsymbol{A}$ defined in (6.23) can be decomposed into

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{P}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{Q}}^{-1}\end{bmatrix} + \begin{bmatrix}\boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1 & \boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_2 \\ \boldsymbol{O}_2^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1 & \boldsymbol{O}_2^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_2\end{bmatrix}. \tag{6.25}$$

By application of Theorem A.6, we see that $\boldsymbol{A}$ has full rank if and only if

$$\det(\boldsymbol{P} + \boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1) \neq 0 \tag{6.26a}$$

$$\det(\boldsymbol{C}_2) \neq 0, \tag{6.26b}$$

with $\boldsymbol{C}_2 \triangleq \underline{\boldsymbol{Q}}^{-1} + \boldsymbol{O}_2^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_2 - \boldsymbol{O}_2^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1(\boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1)^{-1}\boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_2$. The worst case regarding the full-rankness of $\boldsymbol{A}$ results from $\boldsymbol{P}^{-1} = \boldsymbol{0}$. From Assumption 13 follows that $\boldsymbol{O}_1$ has full rank. Thus, (6.26a) is always fulfilled because $\operatorname{rank}(\boldsymbol{O}_1^T\underline{\boldsymbol{R}}^{-1}\boldsymbol{O}_1) = \operatorname{rank}(\boldsymbol{O}_1)$ since $\underline{\boldsymbol{R}}^{-1} > \boldsymbol{0}$ according to Assumption 12. By decomposing $\underline{\boldsymbol{R}}^{-1} = \underline{\boldsymbol{R}}^{-1/2}\underline{\boldsymbol{R}}^{-1/2}$, we get for $\boldsymbol{C}_2$

$$\begin{aligned}\boldsymbol{C}_2 &= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T\breve{\boldsymbol{O}}_2 - \breve{\boldsymbol{O}}_2^T\breve{\boldsymbol{O}}_1(\breve{\boldsymbol{O}}_1^T\breve{\boldsymbol{O}}_1)^{-1}\breve{\boldsymbol{O}}_1^T\breve{\boldsymbol{O}}_2 \\ &= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T(\boldsymbol{I} - \breve{\boldsymbol{O}}_1(\breve{\boldsymbol{O}}_1^T\breve{\boldsymbol{O}}_1)^{-1}\breve{\boldsymbol{O}}_1^T)\breve{\boldsymbol{O}}_2,\end{aligned}$$

where $\breve{\boldsymbol{O}}_1 \triangleq \boldsymbol{O}_1\underline{\boldsymbol{R}}^{-1/2}$ and $\breve{\boldsymbol{O}}_2 \triangleq \boldsymbol{O}_2\underline{\boldsymbol{R}}^{-1/2}$. Using a singular value decomposition of $\breve{\boldsymbol{O}}_1$, i. e. $\breve{\boldsymbol{O}}_1 = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ with $\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Lambda} \\ \boldsymbol{0}\end{bmatrix}$, we get

$$\begin{aligned}\boldsymbol{C}_2 &= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T(\boldsymbol{I} - \breve{\boldsymbol{O}}_1(\breve{\boldsymbol{O}}_1^T\breve{\boldsymbol{O}}_1)^{-1}\breve{\boldsymbol{O}}_1^T)\breve{\boldsymbol{O}}_2 \\ &= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^T)^{-1}\boldsymbol{V}^T\boldsymbol{\Sigma}\boldsymbol{U}^T)\breve{\boldsymbol{O}}_2\end{aligned}$$

$$
\begin{aligned}
&= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T (\boldsymbol{I} - \boldsymbol{U} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{U}^T) \breve{\boldsymbol{O}}_2 \\
&= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T \boldsymbol{U} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{U}^T \breve{\boldsymbol{O}}_2 \\
&= \underline{\boldsymbol{Q}}^{-1} + \breve{\boldsymbol{O}}_2^T \boldsymbol{U} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{U}^T \breve{\boldsymbol{O}}_2.
\end{aligned}
$$

Due to Assumption 12, which guarantees $\underline{\boldsymbol{Q}}^{-1} > \boldsymbol{0}$, $\boldsymbol{C}_2$ is positive definite since it is a sum of a positive definite and a positive semi-definite matrix. Hence, (6.26b) is fulfilled and $\boldsymbol{A}$ has full rank. □

It is interesting to note that if the first term (3.20a) and the third term (3.20c) are not present in the cost function, i. e. $\boldsymbol{P}_k^{-1} = \boldsymbol{0}$ and $\underline{\boldsymbol{Q}}_k^{-1} = \boldsymbol{0}$, then $\hat{\boldsymbol{p}}_k^*$ would not be an isolated asymptotically stable equilibrium of (6.21). In other words, the optimization problem would not be well-posed. Thanks to Assumption 12, Theorem 6.5.1 ensures that we can find the optimal solution $\hat{\boldsymbol{p}}_k^*$ by a forward simulation of (6.21), provided that the initial value $\hat{\boldsymbol{p}}_k^\circ$ is inside the region of attraction of $\hat{\boldsymbol{p}}_k^*$. Thus, we make the following assumption.

Assumption 14. The initial value $\hat{\boldsymbol{p}}_k^\circ$ is bounded and inside the region of attraction of the isolated asymptotically stable equilibrium $\hat{\boldsymbol{p}}_k^*$ of system (6.9), i. e.

$$
\hat{\boldsymbol{p}}_k^\circ \in \mathcal{R}_k \triangleq \{\hat{\boldsymbol{p}}_k^\circ \in \mathbb{R}^{n_p} | \lim_{\kappa \to \infty} \hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u}) = \hat{\boldsymbol{p}}_k^*\}, \quad \forall k \in \mathbb{N}_{N+1}.
$$

Recall that assumptions of this kind are quite typical for gradient-based algorithms used for solving non-convex optimization problems since they only converge to the global optimal value, if the initial value is close enough to the optimal one, cf. Section 2.4.

Since we are interested in a suboptimal solution to (6.20) rather than in the optimal one, we stop the forward simulation of (6.21) as soon as the decreasing condition (3.21) is fulfilled. This raises the question, if we can make any statements about the estimation error $\boldsymbol{e}(t) \triangleq \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ when we stop the forward simulation of (6.21) prematurely at time κ_k. By means of the Lipschitz constant

$$
L \triangleq \max_{\boldsymbol{x} \in \mathbb{X}, \boldsymbol{u} \in \mathbb{U}} \left\| \frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}} \right\|,
$$

and the bounds

$$
\delta_f \triangleq \max_{\boldsymbol{x} \in \mathbb{X}, \boldsymbol{u} \in \mathbb{U}} \|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})\|, \delta_w \triangleq \max_{\boldsymbol{w} \in \mathbb{W}} \|\boldsymbol{w}\|, \delta_{\hat{w}} \triangleq \max_{\hat{\boldsymbol{w}} \in \hat{\mathbb{W}}} \|\hat{\boldsymbol{w}}\|, \delta_\Delta \triangleq \max_{t \in \mathbb{R}_0} \|\boldsymbol{w}(t) - \hat{\boldsymbol{w}}(t)\| \leq \delta_w + \delta_{\hat{w}},
$$

we can answer this question in the following Lemma.

Lemma 6.5.2. *Suppose that Assumptions 12, 13 and 14 hold and that* $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$. *Then the norm of the estimation error is bounded in the prediction interval* $t \in \left[\max_{j\in\mathcal{I}_k} t_{j|k},\ \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}\right]$ *for any* $k \in \mathbb{N}_{N+1}$, $\kappa \in \mathbb{R}_0$, $\hat{\boldsymbol{p}}_k^\circ \in \mathcal{R}_k$ *and any admissible* $\boldsymbol{u} \in \mathbb{U}$ *by*

$$\begin{aligned}\|\boldsymbol{e}(t)\| \leq \Big(\Big(\|\hat{\boldsymbol{x}}_{k-N|k} - \boldsymbol{x}_{k-N|k}\| + |(\hat{\alpha}_k - \alpha_k)\bar{t}_{k-N|k} + (\hat{\beta}_k - \beta_k)|(\delta_f + \delta_{\hat{w}})\Big) e^{L\hat{\alpha}_k(\bar{t}_{k|k}-\bar{t}_{k-N|k})} \\ + \frac{\delta_\Delta}{L}\left(e^{L\hat{\alpha}_k(\bar{t}_{k|k}-\bar{t}_{k-N|k})} - 1\right)\Big) e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} + \frac{\delta_w}{L}\left(e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} - 1\right).\end{aligned} \tag{6.27}$$

Proof. The following holds for all $k \in \mathbb{N}_{N+1}$. The interval $[\hat{\alpha}_k\,\bar{t}_{k-N|k} + \hat{\beta}_k, \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}]$ is divided into two parts according to the estimated state disturbance. In the first part $[\hat{\alpha}_k\,\bar{t}_{k-N|k} + \hat{\beta}_k, \hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k]$ the estimated state disturbance $\hat{\boldsymbol{w}}(t)$ is present, while in second part $[\hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k, \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}]$ no estimated state disturbance occurs. For each interval, Theorem A.4 is applied and afterwards the results are combined to yield the desired bound.

Application of Theorem A.4 results with $\boldsymbol{y} = \boldsymbol{x}$, $\boldsymbol{z} = \hat{\boldsymbol{x}}$, $\boldsymbol{g}(t) = \boldsymbol{w}(t)$, $\boldsymbol{h}(t) = \hat{\boldsymbol{w}}(t)$, $\boldsymbol{y}_0 = \boldsymbol{x}_{k-N|k}$, $\boldsymbol{z}_0 = \hat{\boldsymbol{x}}_{k-N|k}$ and with the times $t_0 = \hat{\alpha}_k\,\bar{t}_{k-N|k} + \hat{\beta}_k$, $t_1 = \alpha_k\,\bar{t}_{k-N|k} + \beta_k$, $t_2 = \hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k$, and with the bounds $\delta = \delta_f$, $\mu = \delta_{\hat{w}}$, $\rho = \delta_\Delta$ as well as with the Lipschitz constant L for $t \in [\hat{\alpha}_k\,\bar{t}_{k-N|k} + \hat{\beta}_k, \hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k]$ in

$$\begin{aligned}\|\boldsymbol{e}(t)\| \leq \Big(\|\hat{\boldsymbol{x}}_{k-N|k} - \boldsymbol{x}_{k-N|k}\| + |(\hat{\alpha}_k - \alpha_k)\bar{t}_{k-N|k} + (\hat{\beta}_k - \beta_k)|(\delta_f + \delta_{\hat{w}})\Big) e^{L(t-\hat{\alpha}_k\,\bar{t}_{k-N|k}-\hat{\beta}_k)} \\ + \frac{\delta_\Delta}{L}\left(e^{L(t-\hat{\alpha}_k\,\bar{t}_{k-N|k}-\hat{\beta}_k)} - 1\right).\end{aligned} \tag{6.28}$$

Application of Theorem A.4 results with $\boldsymbol{y} = \boldsymbol{x}$, $\boldsymbol{z} = \hat{\boldsymbol{x}}$, $\boldsymbol{g}(t) = \boldsymbol{w}(t)$, $\boldsymbol{h}(t) = \boldsymbol{0}$, $\boldsymbol{y}_0 = \boldsymbol{x}(\hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k)$, $\boldsymbol{z}_0 = \hat{\boldsymbol{x}}_{k|k}$ and with the times $t_0 = t_1 = \hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k$, $t_2 = \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}$, and with the bound $\rho = \delta_w$ as well as with the Lipschitz constant L for $t \in [\hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k, \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}]$ in

$$\|\boldsymbol{e}(t)\| \leq \|\hat{\boldsymbol{x}}_{k|k} - \boldsymbol{x}(\hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k)\| e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} + \frac{\delta_w}{L}\left(e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} - 1\right). \tag{6.29}$$

Bounding $\|\hat{\boldsymbol{x}}_{k|k} - \boldsymbol{x}(\hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k)\|$ in (6.29) with (6.28) for $t = \hat{\alpha}_k\,\bar{t}_{k|k} + \hat{\beta}_k$ yields for $t \in [\max_{j\in\mathcal{I}_k} t_{j|k}, \max_{j\in\mathcal{I}_{k+1}} t_{j|k+1}]$

$$\begin{aligned}\|\boldsymbol{e}(t)\| \leq \Big(\Big(\|\hat{\boldsymbol{x}}_{k-N|k} - \boldsymbol{x}_{k-N|k}\| + |(\hat{\alpha}_k - \alpha_k)\bar{t}_{k-N|k} + (\hat{\beta}_k - \beta_k)|(\delta_f + \delta_{\hat{w}})\Big) e^{L\hat{\alpha}_k(\bar{t}_{k|k}-\bar{t}_{k-N|k})} \\ + \frac{\delta_\Delta}{L}\left(e^{L\hat{\alpha}_k(\bar{t}_{k|k}-\bar{t}_{k-N|k})} - 1\right)\Big) e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} + \frac{\delta_w}{L}\left(e^{L(t-\hat{\alpha}_k\,\bar{t}_{k|k}-\hat{\beta}_k)} - 1\right)\end{aligned}$$

which completes the proof. □

Note that the error bound (6.27) is conservative but constructive. This means that it is only tight for the worst case scenario but it reflects the influence of the different parameters on the performance of the CMHE. For instance, the better the estimation quality of $\hat{\boldsymbol{p}}_k$ is, the smaller is the estimation error bound. Note that the cumbersome formulation of the

prediction interval stems from the fact that not every arrived packet is incorporated in the buffer and thus the arrival time stamps t_j cannot be used directly to define the prediction interval, cf. Section 3.7. The maximal length of the prediction interval is

$$T_{\max,\mathrm{pre}} = (N_{\max,\mathrm{drop}} + 1) \left(\max_{k \in \mathbb{N}_{N+1}} \alpha_k \right) \delta_{\bar{t}} + \tau_{\max}. \tag{6.30}$$

Consequently, a reduction of the prediction interval and thus a diminution of the error bound is accompanied with either a decrease of the time between two consecutive measurements $\delta_{\bar{t}}$ or of the maximum number of consecutive packet drops $N_{\max,\mathrm{drop}}$ or of the maximum packet delay $\tau_{\max}$. The main difference compared to the CMHO lies in the fact that the error bound does not vanish when the estimation derived in the update step is precisely the true state, i. e. $\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{x}_{k|k}$. In this case, the first term in (6.27) vanishes but the second one remains. This is due to the unknown state disturbance acting on the system during the prediction interval.

Nevertheless, we can state the following stability result for the CMHE based on the findings so far.

Theorem 6.5.3. *Let the Assumptions 12, 13 and 14 hold. Then the decreasing condition*

$$J_k(\hat{\boldsymbol{p}}_k) \leq \max\{\xi_k J_{k-1}(\hat{\boldsymbol{p}}_{k-1}), \delta_J\} \tag{6.31}$$

with the decreasing factors $\xi_k \in [0, 1[$, the upper bound for the optimal cost function values $\delta_J \geq \max_{k \in \mathbb{N}_{N+1}} J_k(\hat{\boldsymbol{p}}_k^*)$ and the initial cost function value $J_N(\hat{\boldsymbol{p}}_N) \triangleq J_{N+1}(\hat{\boldsymbol{p}}_{N+1}^\circ)$ is feasible by the optimization algorithm (6.21) for any $k \in \mathbb{N}_{N+1}$. Moreover, the norm of the CMHE estimation error $\|\boldsymbol{e}(t)\|$ is bounded for all $t \in \mathbb{R}_0$.

Proof. The following holds for all $k \in \mathbb{N}_{N+1}$: The Assumptions 12 and 13 imply that Theorem 6.5.1 holds, i. e. $\hat{\boldsymbol{p}}_k^*$ is an isolated asymptotically stable equilibrium of (6.21). Assumption 14 guarantees that the initial value $\hat{\boldsymbol{p}}_k^\circ$ is bounded and inside the region of attraction $\mathcal{R}_k$ of the equilibrium $\hat{\boldsymbol{p}}_k^*$. Since $\hat{\boldsymbol{p}}_k^*$ is the only equilibrium of (6.21) in $\mathcal{R}_k$ and $\boldsymbol{M}_k > \boldsymbol{0}$, it follows from (6.21) that

$$\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k} \neq \boldsymbol{0}, \quad \forall \hat{\boldsymbol{p}}_k \in \mathcal{R}_k \setminus \{\hat{\boldsymbol{p}}_k^*\}.$$

Thus, $V_k(\hat{\boldsymbol{p}}_k) = J_k(\hat{\boldsymbol{p}}_k) - J_k(\hat{\boldsymbol{p}}_k^*)$ is a Lyapunov function for system (6.21) which is positive definite around $\hat{\boldsymbol{p}}_k^*$ for all $\hat{\boldsymbol{p}}_k \in \mathcal{R}_k$. Note that $V_k(\hat{\boldsymbol{p}}_k)$ is zero in $\mathcal{R}_k$ if and only if $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}_k^*$. The derivative of $V_k(\hat{\boldsymbol{p}}_k)$ along the trajectories of (6.21) is

$$\dot{V}_k(\hat{\boldsymbol{p}}_k) = -\frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}^T(\hat{\boldsymbol{p}}_k) \boldsymbol{M}_k(\hat{\boldsymbol{p}}_k) \frac{\partial J_k}{\partial \hat{\boldsymbol{p}}_k}(\hat{\boldsymbol{p}}_k)$$

and is negative definite around $\hat{\boldsymbol{p}}_k^*$ for all $\hat{\boldsymbol{p}}_k \in \mathcal{R}_k$, since $\boldsymbol{M}_k(\hat{\boldsymbol{p}}_k) > \boldsymbol{0}$. Therefore, the Lyapunov function $V_k(\hat{\boldsymbol{p}}_k) = V_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u}))$ and the cost function $J_k(\hat{\boldsymbol{p}}_k) = J_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$

converge to zero and to the optimal value as the optimization time goes to infinity, respectively, i. e.

$$\lim_{\kappa\to\infty} V_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})) = 0 \qquad \text{and} \qquad \lim_{\kappa\to\infty} J_k(\hat{\boldsymbol{\vartheta}}(\kappa, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})) = J_k(\hat{\boldsymbol{p}}_k^*) \leq \delta_J.$$

Hence, for any $\xi_k \in [0, 1[$, there always exists an optimization time κ_k and associated optimization variable $\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{\vartheta}}(\kappa_k, \hat{\boldsymbol{p}}_k^\circ, \boldsymbol{u})$ such that the decreasing condition (6.31) is satisfied, i. e. $J_k(\hat{\boldsymbol{p}}_k) \leq \max\{\xi_k J_{k-1}(\hat{\boldsymbol{p}}_{k-1}), \delta_J\}$. Furthermore, the only limit point of the resulting sequence of cost function values $\{J_k(\hat{\boldsymbol{p}}_k)\}$ satisfying the decreasing condition (6.31) is δ_J, i. e.

$$\lim_{k\to\infty} J_k(\hat{\boldsymbol{p}}_k) = \delta_J.$$

Since the Assumptions 12, 13 and 14 hold, it follows from Lemma 6.5.2 that the norm of the estimation error $\|\boldsymbol{e}(t)\|$ is bounded for all $t \in \mathbb{R}_0$. □

This theorem constitutes the counterpart of Theorem 6.4.3 for the disturbed scenario and has been developed based on the same techniques. Therefore, it is hardly surprising that the major three CMHO facts still hold in the core. Due to their importance, we state them subsequently in adapted form.

First of all, note that Theorem 6.5.3 holds for any network which satisfies the quite general Assumptions 5-7 and any weighting matrices which satisfy Assumption 12. As long as these Assumptions are fulfilled, the specific choice of the weighting matrices, the packet delay and the packet drop statistics do not influence the boundedness of the nominal CMHE. But they do, of course, determine the quality of the estimation error bound and therefore the achievable performance of the CMHE. This means that the weighting matrices are an important tool for setting up the estimation performance, cf. Section 3.5.

Second, due to the utilized unifying formulation (6.21), all optimization algorithms are permissible which utilize the exact gradient and a positive-definite matrix $\boldsymbol{M}_k$. This includes not only all the algorithms presented in Section 2.4.2 but also line search and trust region methods which utilize the Hessian approximation derived in Proposition 4.5.2.

Third, instead of the optimal approach, the proposed suboptimal one is sufficient to guarantee boundedness of the nominal CMHE. This suboptimal concept is straightforward to implement and requires only the choice of the decreasing factors ξ_k. These have the same meaning as for the CMHO and opens up the possibility to set up a compromise between the achievable convergence speed and the required computation time.

One may notice that by choosing

$$\prod_{i=N+1}^{N+1+j} \xi_i \leq \frac{\delta_J}{J_N(\hat{\boldsymbol{p}}_k)}, \quad j \in \mathbb{N}_0, \tag{6.32}$$

we can get $J_{N+1+j} \leq \delta_J$. Hence the following corollary stems directly from Theorem 6.5.3.

Corollary 6.5.4. *Let the Assumptions 12, 13 and 14 hold. Then there exists for every* $j \in \mathbb{N}_0$ *a sequence of decreasing factors* $\{\xi_i\}_{i=N+1}^{N+1+j}$ *such that the cost function values converge in finite time, i. e.*

$$J_k(\hat{\boldsymbol{p}}_k) \leq \delta_J, \quad \forall k \geq N+1+j, \quad j \in \mathbb{N}_0. \tag{6.33}$$

Now we utilize the presented stability analysis to discuss the to be expected CMHE performance and to compare it to the CMHO one by means of the schematic illustration given in Figure 6.4. Without loss of generality, we consider the case where the arrival time stamps can be directly used for defining the prediction intervals. The initial cost function values $J_i(\hat{\boldsymbol{p}}_i^\circ)$, the suboptimal cost function values $J_i(\hat{\boldsymbol{p}}_i)$ and the Lyapunov function values $V_i(\hat{\boldsymbol{p}}_i)$ are marked with blue circles, blue squares and a green rhombuses at the corresponding arrival times t_i, respectively. Note that in contrast to the CMHO, the Lyapunov function values $V_i(\hat{\boldsymbol{p}}_i)$ are generally not identical to suboptimal cost function values $J_i(\hat{\boldsymbol{p}}_i)$ due to the fact that the optimal cost function values $J_i(\hat{\boldsymbol{p}}_i^*)$ are non-vanishing. Each value of $J_i(\hat{\boldsymbol{p}}_i)$ and $V_i(\hat{\boldsymbol{p}}_i)$ is connected by a blue dashed line and a dashed green line, respectively, to highlight the effect of the decreasing condition (6.31). This condition guarantees a steady decline of $J_i(\hat{\boldsymbol{p}}_i)$ but not necessarily of $V_i(\hat{\boldsymbol{p}}_i)$. The resulting maximal estimation error bound $e_{\max}(t)$ derived in (6.27) is depicted as an orange solid line. Note that this bound is conservative, i. e. it is not tight. Moreover, it jumps at the beginning of each prediction interval due to the new information available synthesized in the suboptimal solution to the optimization problem of each update step. From the $(N+1+j)$-th arrival time stamp onwards, $J_i(\hat{\boldsymbol{p}}_i^*)$ is bounded by δ_J due to the decreasing factor choice (6.32). The fact that the suboptimal solutions $\hat{\boldsymbol{p}}_k$ remain in the bounded sublevel sets $\{\hat{\boldsymbol{p}}_k \in \mathbb{P} | J_k(\hat{\boldsymbol{p}}_k) \leq \delta_J, k > N+1+j, j \in \mathbb{N}_0\}$ guarantees the existence of the maximal estimation error bound δ_e for $t \geq t_{N+j+1}$. Note that in contrast to the CMHO, the state and the clock parameters cannot be reconstructed exactly. Nevertheless, it should

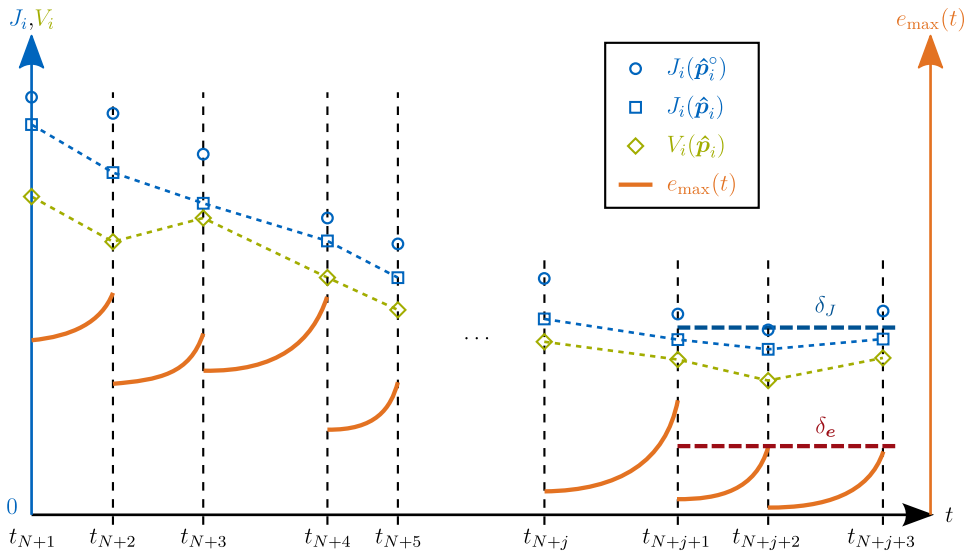


Figure 6.4: Schematic illustration of the CMHE performance.

be stressed that the presented results hold for all networks and disturbances which feature bounded statistics.

6.6 Summary

In this chapter, we have analyzed the stability of the nominal CMHO and the nominal CMHE. Both analyses are founded on the idea of expressing gradient-based optimization algorithms as a single unifying continuous-time dynamical system. The essential characteristic of this system is that its vector field is the product of an arbitrary positive definite matrix function and the negative gradient of the cost function derived in Chapter 4. The freedom in choosing this matrix has opened up the possibility of selecting any line search or trust region method presented in Section 2.4.2. This perception has facilitated the accomplishment of the stability analysis in three steps. In the first step, we have shown that if the NCS is observable in the sense derived in Chapter 5, then the aforementioned unifying continuous-time system possesses an isolated asymptotically stable equilibrium which is the optimal solution to the optimization problem stemming from the corresponding update step. We have proven in the second step that if the suboptimal approach satisfies the decreasing condition, then the suboptimal solutions approach to the optimal ones. In the third step, we have concluded from this fact and by invoking the continuous dependence of initial value problems the following results for the disturbed and undisturbed scenario. For the former case, we have shown boundedness of the CMHE estimation error. For the latter case, i. e. in the absence of disturbances, we have proven asymptotic and even finite-time convergence of the CMHO observation error depending on the choice of the decreasing factors. These factors allow to set up a compromise between the achievable convergence speed and the required computation time. It should be pointed out that the main feature of these results is its generality. This means that the derived analysis hold for the general assumptions made in the framework presented in Chapter 3 which includes, inter alia, the quite general packet delay and packet dropout assumptions.

Chapter 7

Simulation and Experimental Results

In this chapter, we present simulation as well as experimental results for both developed centralized moving horizon strategies (CMHS) applied to a networked version of a nonlinear benchmark system and to a networked pendulum test-rig. A conventional continuous-discrete extended Kalman filter (CDEKF) (see Appendix C) serves in each case as a comparison to the CMHS.

In Section 7.1, we present the simulation results for the centralized moving horizon observer (CMHO), the centralized moving horizon estimator (CMHE) and the CDEKF applied to a networked version of a nonlinear benchmark system. These three methods are compared to each other for various settings of the disturbances, the network conditions and the decreasing factors describing the quality of the suboptimal approach used by the CMHS. A significant characteristic of the benchmark system are the strong nonlinearities not only in the state equations but also in the sensing model. In Section 7.2, we present the experimental results for the CMHE and the CDEKF applied to a networked pendulum test-rig. The transition between two stationary setpoints and the swing-up and stabilization problem serve as an open-loop and closed-loop benchmark, respectively. The latter represents an especially challenging benchmark due to the unstable and non-minimum phase system dynamics along with the network-induced non-negligible packet delays and packet drops. In order to achieve a satisfactory real-time performance, we develop and implement additional steps to exploit the available computational power by reducing the idle time and guaranteeing the strict observance of an upper bound for the execution time. Finally, we conclude the chapter with a summary given in Section 7.3

7.1 Networked Nonlinear Benchmark System

In this section, we present the simulation results for the centralized moving horizon observer (CMHO) and the centralized moving horizon estimator (CMHE) applied to a networked version of a nonlinear benchmark system. Thereby, we compare both methods and contrast them to a conventional continuous-discrete extended Kalman filter (CDEKF) (see Appendix C). All simulations are performed within the centralized NCS architecture depicted in Figure 3.1.

The simulation results are presented in Section 7.1.2 for the simulation setup given in Section 7.1.1.

7.1.1 Simulation Setup

The plant Σ is a commonly used nonlinear scalar benchmark system (Lo, 1994; Alessandri et al., 2008) extended by the control input $u(t)$ and reads as

$$\dot{x}(t) = 2e^{-2x^2(t)} - 1 + u(t) + w(t), \tag{7.1}$$

where the initial state is $x(0) = x^\circ$ and the state disturbance $w(t)$ is uniformly distributed in the interval $[-b_w, b_w]$. The control input $u(t)$ is chosen to be $u(t) = \sin(2t)\sin(0.5t + 2)$ and is sufficiently exciting to enable observability of the NCS, cf. Theorem 5.2.4 and 5.3.4.

The sensor Σ_S posses the sensing model

$$y(\bar{t}) = x^3(\bar{t}) + v(\bar{t}), \tag{7.2}$$

where the measurement disturbance $v(\bar{t})$ is uniformly distributed in the interval $[-b_v, b_v]$. The sensor time $\bar{t}$ is related to the global time via the clock parameters $\alpha_k = 0.9$ and $\beta_k = -1\,\mathrm{s}$. A packet $\mathcal{P}_i$ of the form $\{y_i, \bar{t}_i\}$ is transmitted over the network, if either y or $\bar{t}$ changes by 0.15 or 0.84 s (0.75 s in global time) based on the information in the latest transmitted packet. Note that (7.1) and (7.2) are strongly nonlinear.

The network Σ_N is simulated by a Matlab-based network simulator which enables the emulation of various protocols and network conditions. These lead to different packet delay distributions in the intervals $[b_{\tau,\min}, b_{\tau,\max}]$.

In order to evaluate the effectiveness of the proposed CMHS, the CMHO and the CMHE are compared with the CDEKF. The settings of these three estimators $\hat{\Sigma}$ are described in the following.

Centralized Moving Horizon Observer

The CMHO utilizes the nominal choice of the update step which minimizes the cost function

$$J_k(\hat{\boldsymbol{p}}_k) = \frac{1}{2} \sum_{i=k-N}^{k} \|\hat{x}^3_{i|k} - y_{i|k}\|^2,$$

where the buffer size is $N + 1 = 5$. The initial value $\hat{x}^\circ$ of the CMHO is chosen manually depending on the investigated scenario while the clock parameters $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ are automatically initialized by Method 2, see Section 3.6.2. The optimization problems of the update steps are solved by a forward simulation of the system (6.2) which unifies various gradient-based optimization algorithms. The required gradient of the cost function J_k is calculated according to the sensitivity based method derived in Section 4.2 and the matrix $\boldsymbol{M}_k$ is the inverse of the Hessian approximation $\boldsymbol{H}_1$ derived in Proposition 4.4.2.

Centralized Moving Horizon Estimator

The CMHE utilizes the nominal choice of the update step which minimizes the cost function

$$J_k(\hat{\boldsymbol{p}}_k) = \frac{1}{2}\left\| \begin{bmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \\ \hat{x}_{k-N|k} \end{bmatrix} - \begin{bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \\ \bar{x}_{k-N|k} \end{bmatrix} \right\|^2_{\boldsymbol{P}_k^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k} \|\hat{x}^3_{i|k} - y_{i|k}\|^2_{R^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k-1} \|\hat{w}_{i|k}\|^2_{Q^{-1}}.$$

The buffer size is identical to the one of the CMHO, i. e. $N+1 = 5$. The initial value $\hat{x}^\circ$ of the CMHE is chosen manually depending on the investigated scenario while the clock parameters $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ are automatically initialized by Method 2, see Section 3.6.2. The prediction values are initialized by $\bar{\alpha}_{N+1} = \hat{\alpha}^\circ_{N+1}$, $\bar{\beta}_{N+1} = \hat{\beta}^\circ_{N+1}$ and $\bar{x}_{k-N|k} = \hat{\phi}_{\text{pre}}(t_{N+1|N+1}, 0, \hat{x}^\circ, u)$. The weighting matrices are set to $\boldsymbol{P}_k^{-1} = 0.5\,\boldsymbol{I}$, $R^{-1} = 1$ and $Q^{-1} = 1$. Similar to the CMHO case, the optimization problems of the update steps are solved by a forward simulation of the system (6.2). The required gradient of the cost function J_k is calculated according to the sensitivity based method derived in Section 4.3 and the matrix $\boldsymbol{M}_k$ is the inverse of the Hessian approximation $\boldsymbol{H}_1$ derived in Proposition 4.5.2.

Continuous-Discrete Extended Kalman Filter

The conventional CDEKF is described in Appendix C and is implemented for the networked scenario in the following way. The CDEKF checks whenever a new packet $\mathcal{P}_{i,j}$ arrives if the corresponding sensor time stamp $\bar{t}_i$ is newer than the one of the packet used for the last update step. If this is true, an update step is performed where the arrival time t_j is used as the time where the corresponding sampling occurred, i. e. the time delays are neglected. Otherwise the packet will be dropped and the prediction proceeds. The weighting matrices are chosen as $P_0 = 1$, $Q = 1$ and $R = 1/5$ while the initial value $\hat{x}^\circ$ is identical to the CMHS.

7.1.2 Simulation Results

Figure 7.1 presents the results of the CMHO for the undisturbed scenario and different values of the decreasing factors ξ_i. The initial value of the CMHO is chosen to $\hat{x}^\circ = 1.75$ ($x^\circ = 1.25$) and the clock parameters are initialized by Method 2, cf. Section 3.6.2, which leads to $\hat{\alpha}^\circ_{N+1} = 1$ ($\alpha_{N+1} = 0.9$) and $\hat{\beta}^\circ_{N+1} = -1.037\,\text{s}$ ($\beta_{N+1} = -1\,\text{s}$). The upper plot of Figure 7.1 shows the sensing model $y(\bar{t})$ depicted in global time t as a gray solid line where the event-based generated measurements $y(\bar{t}_i)$ are marked with a blue cross. The blue arrows indicate the run of the corresponding packet $\mathcal{P}_i$ through the network. This means that the time of the beginning of the arrow corresponds to the sensor time stamp depicted in global time while the tail of the arrow correlates to the arrival time stamp. The middle plot of Figure 7.1 presents the estimated state $\hat{x}(t)$ of the CMHO for different values of the decreasing factors ξ_i. The resulting absolute estimation error $|e(t)|$ is illustrated in the lower plot of Figure 7.1. The impact of the update steps can be seen in both plots at the times where the buffer $\mathcal{B}_k$ changes which are indicated by dashed gray lines. At these times, the estimates $\hat{x}(t)$ change abruptly

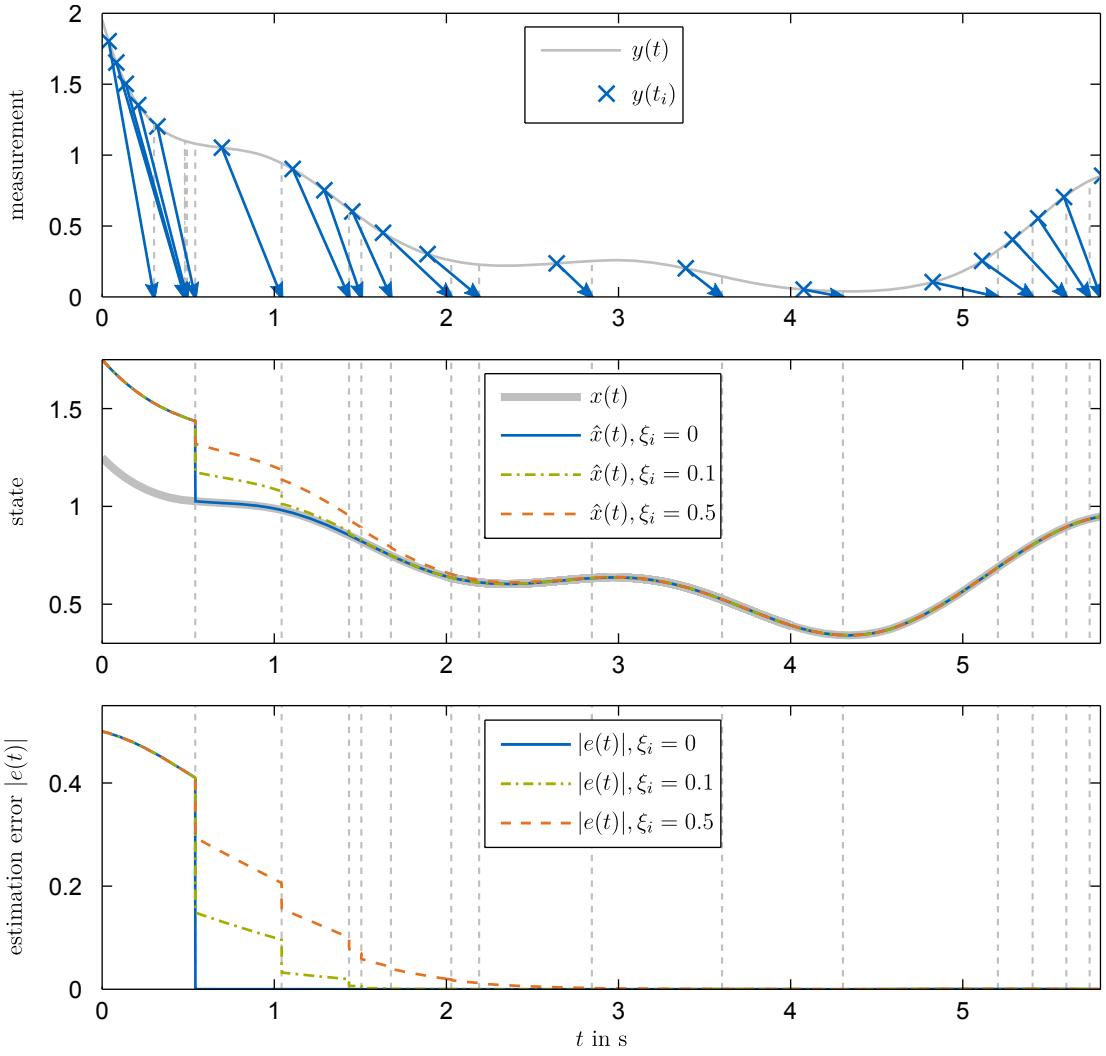


Figure 7.1: Performance of the CMHO for $b_w = 0$, $b_v = 0$, $b_{\tau,\min} = 0.2\,\mathrm{s}$, $b_{\tau,\max} = 0.4\,\mathrm{s}$ and different values of the decreasing factors ξ_i.

towards the true state $x(t)$. Thereby, the rate of convergence depends on the value of the decreasing factors. For the special case of $\xi_i = 0$, we obtain a dead-beat observer, i. e. the estimation error vanishes after the first update step. The condition number $\varrho(\partial \underline{\boldsymbol{h}}_k / \partial \hat{x}_{k-N|k})$ is a measure of observability, cf. Section 5.5.2, and is for the depicted scenario in the interval $[14.1, 295.8]$.

Figure 7.2 is the counterpart of Figure 7.1 and presents the results of the CMHE for the disturbed scenario and different values of the decreasing factors ξ_i. Apart from different values of b_w and b_v, both settings are identical. Similar to the CMHO, we can observe an abrupt change of the estimated states $\hat{x}(t)$ towards the true states $x(t)$ whenever an update step is performed. In contrast to the CMHO, the estimation error $|e(t)|$ of the CMHE does not vanish but remains bounded due to the unknown disturbances. The middle and the lower plot of Figure 7.2 show that $\hat{x}(t)$ converges for $\xi = 0$, $\xi_i = 0.1$ and $\xi_i = 0.5$ after 1, 5 and 8 update steps which corresponds to the times 0.65 s, 1.65 s and 2.81 s, respectively.

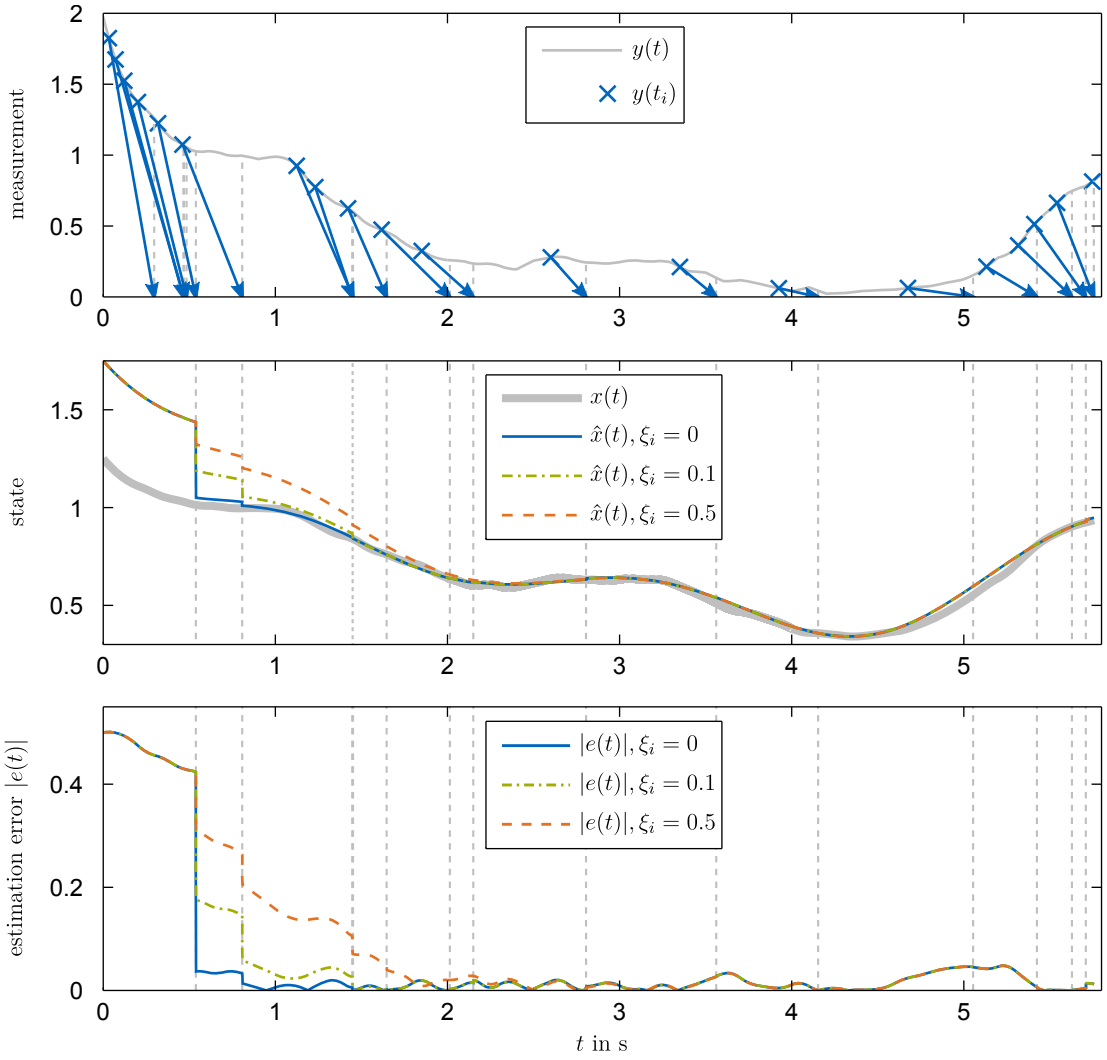


Figure 7.2: Performance of the CMHE for $b_w = 0.25$, $b_v = 0.025$, $b_{\tau,\min} = 0.2\,\mathrm{s}$, $b_{\tau,\max} = 0.4\,\mathrm{s}$ and different values of ξ_i.

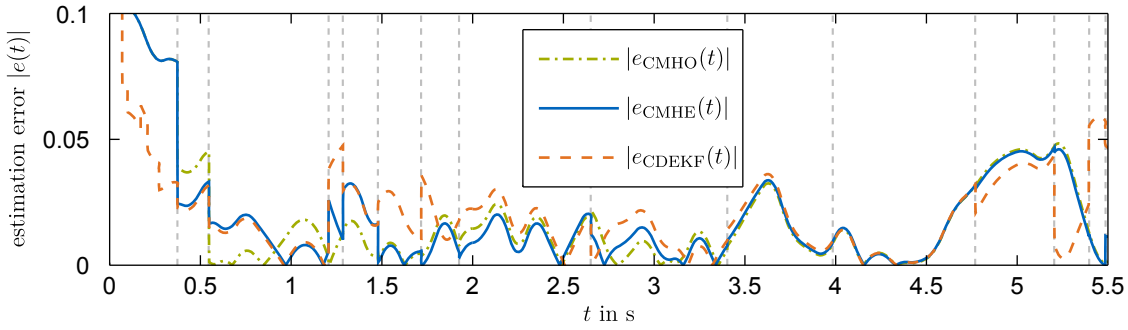
For the sake of comparison, we consider the performance index given by the root mean square error (RMSE)

$$\mathrm{RMSE} \triangleq \left(\int_{t_1}^{t_2} \frac{|e(t)|^2}{t_2 - t_1} dt \right)^{\frac{1}{2}}, \tag{7.3}$$

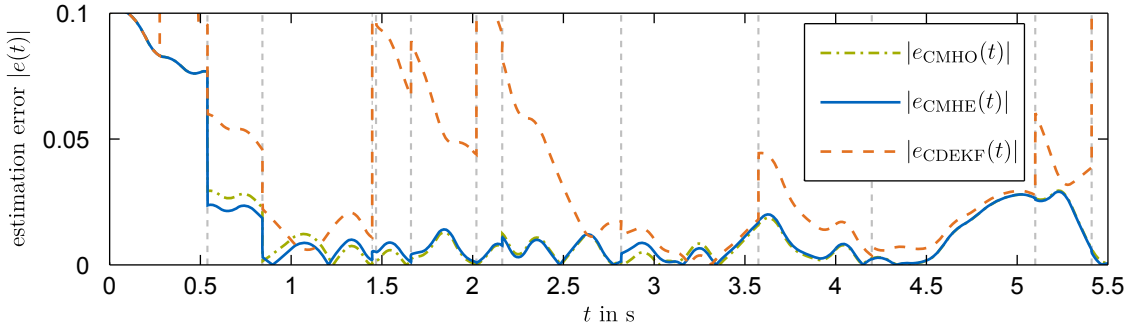
where $|e(t)|$ is the absolute value of the estimation error at time t, t_1 is the time where the first update step of the corresponding estimator is performed and t_2 is 5.6 s. The RMSE values of the CMHO, CMHE and CDEKF for different settings are depicted in Table 7.1. The absolute estimation errors $|e(t)|$ of the CMHO, CMHE and CDEKF for the first setting of the time delay variation and for the second setting of the disturbance variation are depicted in Figure 7.3(a) and 7.3(b), respectively. The dashed gray lines in Figure 7.3 indicate the times where an update step of the CMHS respective CDEKF is performed. In the presence of high disturbances and low delays, the performance of the CDEKF is almost comparable to those of the CMHO and CMHE. For increasing delays, however, the CMHO and the CMHE

Table 7.1: Performance of the CMHO, CMHE and CDEKF for $\hat{x}^\circ = 1.35$, $\hat{\alpha}^\circ_{N+1}$ and $\hat{\beta}^\circ_{N+1}$ according to Method 2, $\xi_i = 0$ and different values of b_w, b_v, $b_{\tau,\min}$ and $b_{\tau,\max}$.

		time delay variation			disturbance variation		
	b_w		0.25		0.05	0.15	0.25
	b_v		0.025		0.005	0.015	0.025
	$b_{\tau,\min}$	0.05 s	0.2 s	0.4 s		0.2 s	
	$b_{\tau,\max}$	0.1 s	0.4 s	0.7 s		0.4 s	
RMSE	CMHO	0.019687	0.021246	0.024809	0.004191	0.012852	0.021246
	CMHE	0.019360	0.019950	0.019955	0.005059	0.012364	0.019950
	CDEKF	0.024437	0.053526	0.080830	0.053997	0.053373	0.053526



(a) Parameter setting: $b_w = 0.25$, $b_v = 0.025$, $b_{\tau,\min} = 0.05\,\mathrm{s}$ and $b_{\tau,\max} = 0.1\,\mathrm{s}$.



(b) Parameter setting: $b_w = 0.15$, $b_v = 0.015$, $b_{\tau,\min} = 0.2\,\mathrm{s}$ and $b_{\tau,\max} = 0.4\,\mathrm{s}$.

Figure 7.3: Absolute estimation error $|e(t)|$ of the CMHO, CMHE and CDEKF for different parameter settings.

significantly outperform the CDEKF. This is hardly surprising, since the CDEKF is not constructed for dealing with the networked scenario. The CDEKF performs only reasonable in regions where the delays have low influence, i. e. $y(t_i) \approx y(t_i - \tau_i)$. For instance, this situation can be seen in the interval $[2.5\,\mathrm{s}, 4\,\mathrm{s}]$ in both plots of Figure 7.3. As to the comparison between the CMHO and the CMHE, the CMHO is slightly superior to the CMHE for the almost undisturbed scenario. However, this situation changes in favor to the CMHE for increasing disturbances. This becomes even more noticeable for high disturbances combined with high time delays.

7.2 Networked Pendulum

In this section, we present the experimental validation of the centralized moving horizon estimator (CMHE) and contrast it to those of a conventional continuous-discrete extended Kalman filter (CDEKF). All experiments are conducted on a networked pendulum test-rig within the centralized NCS architecture depicted in Figure 7.4. Thereby, we utilize a two-degree of freedom controller Σ_{C}. This means that the feedforward controller Σ_{FF} is supported by the state feedback controller Σ_{FB} with one of the above two estimators $\hat{\Sigma}$ in order to stabilize the pendulum system Σ along the nominal trajectories $\boldsymbol{x}^{\star}(t)$ provided by the signal generator $\Sigma^{\star}$. The transition between two stationary and stable setpoints and the more challenging swing-up and stabilization problem serve as an open-loop and closed-loop benchmark, respectively. A feature of the proposed CMHE is the compensation of network induced imperfections. This means that the feedforward controller Σ_{FF} as well as the feedback controller Σ_{FB} are designed as if no network Σ_{N} is present. In the following subsections, the above mentioned points are addressed.

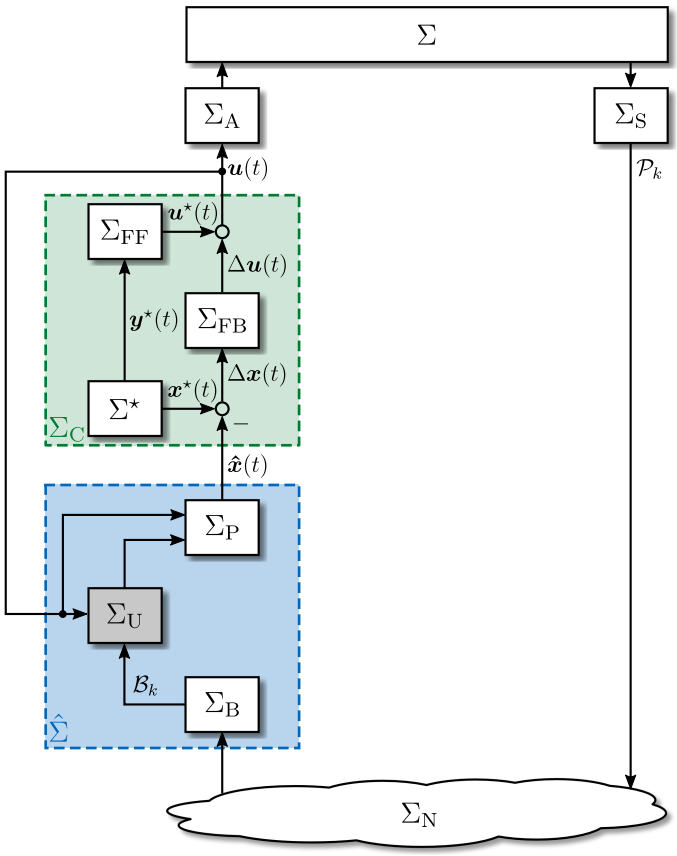
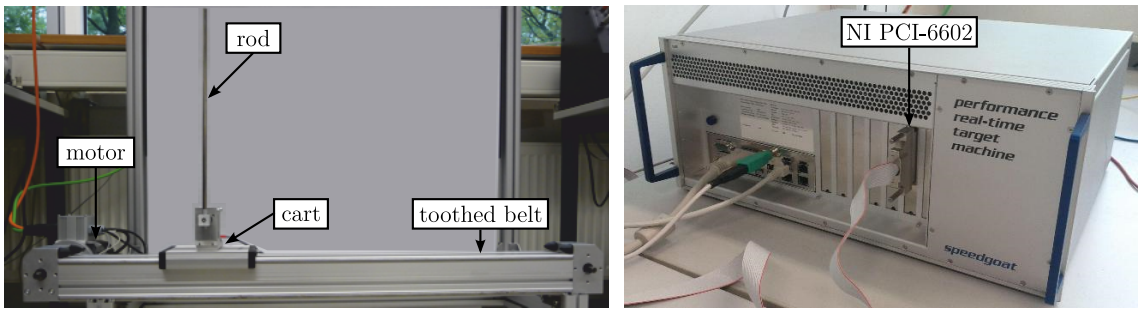


Figure 7.4: Centralized NCS architecture with pendulum system Σ, sensor Σ_{S}, network Σ_{N}, estimator $\hat{\Sigma}$ (consisting of buffer Σ_{B}, updater Σ_{U} and predictor Σ_{P}), two-degree-of-freedom controller Σ_{C} (consisting of signal generator $\Sigma^{\star}$, feedforward controller Σ_{FF} and feedback controller Σ_{FB}) and actuator Σ_{A}.

7.2.1 Pendulum Test Rig

The pendulum test rig used for the experiments is shown in Figure 7.5(a). The cart is driven by a toothed belt connected to a synchronous motor with a maximum input of $|u_{\max}| = 45\,\mathrm{V}$. The position of the cart x and the angle of the rod φ are measured by incremental angle encoders which have a resolution of $2\pi/4096\,\mathrm{rad}$. Both measurements are combined to emulate the sensor Σ_{S} depicted in Figure 7.4 with one local sensor time $\bar{t}$ and the clock parameters $\alpha_k = 1.2$ and $\beta_k = -3\,\mathrm{s}$. A packet $\mathcal{P}_i$ of the form $\{x_i, \varphi_i, \bar{t}_i\}$ is transmitted over the network, if either x, φ or $\bar{t}$ changes by $0.01\,\mathrm{m}$, $0.0175\,\mathrm{rad}$ or $10\,\mathrm{ms}$ based on the information in the latest transmitted packet.



(a) Pendulum test-rig. (b) Real-time target machine.

Figure 7.5: The networked pendulum test-rig.

The real-time control of the pendulum test-rig is realized via xPC Target[1] from MathWorks. This enables the execution of Simulink models in real-time on x86-based real-time target machines. The one used for the experiments is depicted in Figure 7.5(b). It possesses an Intel Core i5-680 CPU with 3.6 GHz, runs a xPC Target kernel with dual core support and is equipped with a NI PCI-6602 data acquisition board for the signal processing. Apart from one exception (see Section 7.2.5), all tasks are performed on the first CPU core with a sampling time of $T_A = 1\,\mathrm{ms}$. In order to test different network protocols and to guarantee reproducibility of the network conditions, the network Σ_{N} is emulated by a Matlab-based hardware in the loop (HiL) network simulator. It enables the emulation of various network protocols and network conditions which leads to different packet delay and packet dropout distributions.

7.2.2 Equations of Motion

The model of the pendulum is derived within the Lagrangian mechanics framework. To this end, we consider the pendulum schematics shown in Figure 7.6 and the associated description of the pendulum parameters in Table 7.2. Based on the absolute position $\boldsymbol{r}(t) = [x(t) - l\sin\varphi(t), l\cos\varphi(t)]^T$ of the center of mass of the rod, the kinetic and potential energies can

[1] http://www.mathworks.de/products/xpctarget/

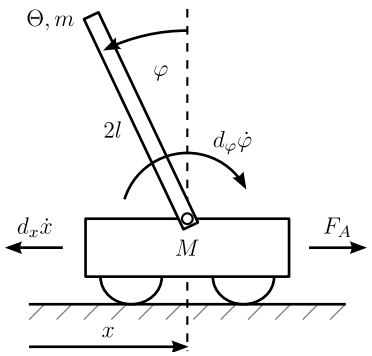


Figure 7.6: Pendulum schematics.

description	symbol	value	unit
half pendulum length	l	0.1925	m
rod mass	m	0.146	kg
cart mass	M	5.9	kg
moment of inertia	Θ	$1.8 \cdot 10^{-3}$	kgm^2
rotational friction constant	d_φ	$5.4 \cdot 10^{-4}$	Nms
horizontal friction constant	d_x	843	Ns/m
motor constant	b	24.95	N/V
gravitation constant	g	9.81	N/kg

Table 7.2: Pendulum Parameters.

be expressed as

$$T(t) = \frac{1}{2}M\dot{x}(t)^2 + \frac{1}{2}\Theta\dot{\varphi}(t)^2 + \frac{1}{2}m\left(\dot{x}(t)^2 + l^2\dot{\varphi}(t)^2 - 2\dot{x}(t)\dot{\varphi}(t)l\cos\varphi(t)\right) \tag{7.4a}$$

$$V(t) = mgl\cos\varphi(t). \tag{7.4b}$$

The non-conservative forces projected in the space of the generalized coordinates $\boldsymbol{r}(t) = [x(t), \varphi(t)]^T$ are

$$F_1(t) = \begin{bmatrix} -d_x\dot{x}(t) \\ 0 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0 \\ -d_\varphi\dot{\varphi}(t) \end{bmatrix}, \quad F_3 = \begin{bmatrix} F_A(t) \\ 0 \end{bmatrix}, \tag{7.5}$$

where $F_1(t)$ and $F_2(t)$ are the viscous friction between the cart and the rail and in the link, respectively. The dynamics of the electrical part of the motor are much faster than the dynamics of the mechanical system and are thus neglected. This leads to the linear relation $F_A(t) = b\,u(t)$ between the voltage $u(t)$ applied to the motor and the force $F_A(t)$ on the cart. The equations of motion are derived by means of the Lagrange equation of second kind

$$\frac{\partial}{\partial t}\frac{\partial L(t)}{\partial \dot{\boldsymbol{r}}} - \frac{\partial L(t)}{\partial \boldsymbol{r}} = \sum_{i=1}^{3} F_i(t), \tag{7.6}$$

with the Lagrangian function $L(t) = T(t) - V(t)$. The resulting dynamics of the pendulum can be written with $\boldsymbol{x}(t) = [{}_1x(t), {}_2x(t), {}_3x(t), {}_4x(t)]^T = [x(t), \dot{x}(t), \varphi(t), \dot{\varphi}(t)]^T$ in state space form $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), u(t))$ as

$${}_1\dot{x}(t) = {}_2x(t) \tag{7.7a}$$

$$\begin{aligned} {}_2\dot{x}(t) = {} & \frac{(\Theta + ml^2)bu(t) - (\Theta + ml^2)d_x\,{}_2x(t) - mld_\varphi\,{}_4x(t)\cos{}_3x(t)}{(\Theta + ml^2)(M+m) - m^2l^2\cos^2{}_3x(t)} \\ & + \frac{-(\Theta + ml^2)ml\,{}_4x(t)^2\sin{}_3x(t) + m^2l^2g\cos{}_3x(t)\sin{}_3x(t)}{(\Theta + ml^2)(M+m) - m^2l^2\cos^2{}_3x(t)} \end{aligned} \tag{7.7b}$$

$$ {}_3\dot{x}(t) = {}_4x(t) \tag{7.7c} $$

$$ {}_4\dot{x}(t) = \frac{bml\cos{}_3x(t)u(t) - mld_x{}_2x(t)\cos{}_3x(t) - (M+m)d_\varphi{}_4x(t) - mld_\varphi{}_4x(t)\cos{}_3x(t)}{(\Theta+ml^2)(M+m) - m^2l^2\cos^2{}_3x(t)} + \frac{-(M+m)mlg\sin{}_3x(t) + m^2l^2g{}_4x(t)^2\cos{}_3x(t)\sin{}_3x(t)}{(\Theta+ml^2)(M+m) - m^2l^2\cos^2{}_3x(t)}. \tag{7.7d} $$

The parameters of the pendulum have been measured and identified at the experimental device and are summarized in Table 7.2.

7.2.3 Feedforward Controller

The overall feedforward control problem can be treated as a sequence of transition problems between two stationary setpoints

$$ [\boldsymbol{x}^\star(0), u^\star(0)]: \qquad \boldsymbol{f}(\boldsymbol{x}^\star(0), u^\star(0)) = \boldsymbol{0} \tag{7.8a} $$

$$ [\boldsymbol{x}^\star(T), u^\star(T)]: \qquad \boldsymbol{f}(\boldsymbol{x}^\star(T), u^\star(T)) = \boldsymbol{0} \tag{7.8b} $$

of system (7.7) within a finite time interval $t \in [0, T]$ subject to the limited voltage $u_{\max}$ of the DC-motor. Thereby, the feedforward variables are indicated by the superscript $\star$. The actual constraint $u^\star_{\max}$ for the feedforward control is set to 40 V and is tighter compared to the physical limit of 45 V in order to leave sufficient reserve for the underlying feedback controller. This imposes the following boundary conditions (BCs) on system (7.7)

$$ \boldsymbol{x}^\star(0) = [{}_1x^\star(0), 0, {}_3x^\star(0), 0]^T \tag{7.9a} $$

$$ \boldsymbol{x}^\star(T) = [{}_1x^\star(T), 0, {}_3x^\star(T), 0]^T \tag{7.9b} $$

$$ |u^\star(t)| \leq u^\star_{\max}, \quad t \in [0, T]. \tag{7.9c} $$

The determination of the feedforward control trajectory $u^\star(t)$ is strongly related to the design of the desired output trajectory $\boldsymbol{y}^\star(t)$ and the corresponding state trajectory $\boldsymbol{x}^\star(t)$. These are provided by the signal generator $\Sigma^\star$. The trajectories must appropriately connect the initial and terminal values given in (7.9a) and (7.9b) while satisfying the constraints (7.9c). In the following, we apply the inversion-based design method presented in Graichen et al. (2005) in a suitable manner.

In order to determine all trajectories $u^\star(t)$, $\boldsymbol{y}^\star(t)$ and $\boldsymbol{x}^\star(t)$, we need an appropriate system representation in input-output coordinates. This is achieved by applying the transformation

$$ [y(t), \dot{y}(t), \eta(t), \dot{\eta}(t)]^T = \Phi(\boldsymbol{x}(t)) \triangleq [{}_1x(t), {}_2x(t), {}_3x(t), {}_4x(t)]^T \tag{7.10} $$

to the system (7.7). For ease of presentation, we suppress the argument t of all functions when the meaning is clear. In the new coordinates, the inverted pendulum has the relative

degree $r = 2$ and is given as

$$\ddot{y} = \chi(\dot{y}, \eta, \dot{\eta}, u) \tag{7.11a}$$
$$\ddot{\eta} = \psi(\eta, \dot{\eta}, \dot{y}, u), \tag{7.11b}$$

where (7.11a) is the input-output dynamic and (7.11b) is the internal dynamic. The zero dynamic of the system following from (7.11b) with $u = 0$ is unstable. This means that the pendulum is non-minimum phase. The inversion-based feedforward control $u^\star$ results from inverting the input-output dynamics

$$u^\star = \chi^{-1}(\dot{y^\star}, \ddot{y^\star}, \eta^\star, \dot{\eta^\star}) \tag{7.12}$$

and depends on the output trajectory $y^\star$ and the state $\eta^\star$ of the internal dynamics (7.11b). To calculate $\eta^\star$, the feedforward control (7.12) is inserted into (7.11b) to yield

$$\ddot{\eta^\star} = \bar{\psi}(\eta^\star, \dot{\eta}^\star, \ddot{y}^\star) = \frac{ml\ddot{y}^\star \cos\eta^\star + mlg \sin\eta^\star - d_\varphi \dot{\eta^\star}}{\Theta + ml^2}. \tag{7.13}$$

Note that $\eta^\star$ depends on the second time derivative of the output trajectory $y^\star$.

The basic idea for tackling the BCs (7.9a) and (7.9b) is to formulate the feedforward control problem as the two-point boundary value problem (BVP)

$$\ddot{y}^\star = \chi(\dot{y}^\star, \eta^\star, \dot{\eta}^\star, u^\star) \tag{7.14a}$$
$$\ddot{\eta}^\star = \bar{\psi}(\eta^\star, \dot{\eta}^\star, \ddot{y}^\star) \tag{7.14b}$$

$$y^\star(0) = {}_1x^\star(0), \quad y^\star(T) = {}_1x^\star(T), \quad \dot{y}^\star(0) = 0, \quad \dot{y}^\star(T) = 0, \tag{7.14c}$$
$$\eta^\star(0) = {}_3x^\star(0), \quad \eta^\star(T) = {}_3x^\star(T), \quad \dot{\eta}^\star(0) = 0, \quad \dot{\eta}^\star(T) = 0, \tag{7.14d}$$

where the BCs (7.14c) and (7.14d) result from the transformed BCs (7.9a) and (7.9b), respectively. This BVP is overdetermined with the two second-order ODEs and the eight BCs. As a necessary condition for the solvability of the BVP (7.14), we provide four free parameters $\boldsymbol{p} = [{}_1p, {}_2p, {}_3p, {}_4p,]^T$ in a set-up function

$$\Omega(t, \boldsymbol{p}) = \sum_{i=1}^{4} {}_ip \sin \frac{k\pi t}{T} \tag{7.15}$$

for the second-order derivative of the output trajectory, i. e. $\ddot{y} = \Omega(t, \boldsymbol{p})$. The solution of the resulting BVP with free parameters compromises the parameter set $\boldsymbol{p}$ as well as the states $y^\star, \dot{y}^\star, \eta^\star, \dot{\eta}^\star$. The numerical solution is a standard task in numerics and is performed by using the MATLAB function `bvp4c`[2] which is designed for the solution of two-point BVPs

[2] http://www.mathworks.com/bvp_tutorial

with unknown parameters. Consequently, the feedforward control $u^\star$ can be calculated by evaluating (7.12).

However, since this approach does not explicitly considers the control constraints (7.9c), it can result in a feedforward control $u^\star$ which violates (7.9c). The observance of these control BCs can be guaranteed by means of the feedforward control $u_\Omega^\star = \alpha^{-1}(\Omega(t, \boldsymbol{p}), \dot{y}^\star, \eta^\star, \dot{\eta}^\star)$ resulting from the set-up function $\Omega(t, \boldsymbol{p})$. Whenever the control constraints are exceeded, the second-order derivative of the output trajectory will be set to the second-order derivative resulting from the system dynamics with $u^\star = \pm u_{\max}^\star$, i. e. $\ddot{y}^\star = \alpha(\dot{y}^\star, \eta^\star, \dot{\eta}^\star, \pm u_{\max}^\star)$. Consequently, the overall BVP can be written as

$$\ddot{y}^\star = \begin{cases} \alpha(\dot{y}^\star, \eta^\star, \dot{\eta}^\star, -u_{\max}^\star) & \text{if } u_\Omega^\star < -u_{\max}^\star \\ \Omega(t, \boldsymbol{p}) & \text{if } |u_\Omega^\star| \leq u_{\max}^\star \\ \alpha(\dot{y}^\star, \eta^\star, \dot{\eta}^\star, u_{\max}^\star) & \text{if } u_\Omega^\star > u_{\max}^\star \end{cases} \tag{7.16a}$$

$$\ddot{\eta}^\star = \bar{\psi}(\eta^\star, \dot{\eta}^\star, \ddot{y}^\star) \tag{7.16b}$$

with the BCs (7.14c) and (7.14d). Note that this choice for $\ddot{y}^\star$ always satisfies $\ddot{y}^\star(0) = 0$ and $\ddot{y}^\star(T) = 0$ and hence implies $\mathcal{C}^0$-continuity of the feedforward trajectory $u^\star$ at the bounds $t = 0$ and $t = T$. This property is used to assemble the overall open and closed loop trajectories from a sequence of suitable single trajectories. In Table 7.3, snapshots of the trajectories ${}_1x^\star$ and ${}_3x^\star$ are given at certain times t. These values uniquely define the boundary conditions (7.9) such that each single trajectory in between can be calculated by means of the presented approach. The overall trajectories $\boldsymbol{x}^\star$ for the open-loop and closed-loop case are depicted in Figure 7.9 and 7.10, respectively. In Figure 7.7, we can see that the feedforward control $u^\star$ goes into saturation during the swing-up phase, i. e. during the first 1.7 s.

Table 7.3: Boundary conditions of the feedforward controls.

	open-loop			closed-loop					
	$t=0\,\mathrm{s}$	$t=1\,\mathrm{s}$	$t=2\,\mathrm{s}$	$t=0\,\mathrm{s}$	$t=1.7\,\mathrm{s}$	$t=3.7\,\mathrm{s}$	$t=6.4\,\mathrm{s}$	$t=8.4\,\mathrm{s}$	$t=11.1\,\mathrm{s}$
${}_1x^\star$	0	0.5	0	0	0	0	0.5	0.5	0
${}_3x^\star$	$-\pi$	$-\pi$	$-\pi$	$-\pi$	0	0	0	0	0

7.2.4 Feedback Controller

The pendulum is stabilized along the nominal trajectories by the feedback controller. In order to compensate for a possible steady-state error in the cart position ${}_1x(t)$, we augment the pendulum model (7.7) by the additional state ${}_5x(t) \triangleq \int_0^t {}_1x(\tau)\, d\tau$ resulting in the overall state vector $\boldsymbol{x}(t) = [{}_1x(t), {}_2x(t), {}_3x(t), {}_4x(t), {}_5x(t)]^T \in \mathbb{R}^5$. Due to the compensation of the network induced imperfections by the CMHE and the accuracy of the nonlinear feedforward controller,

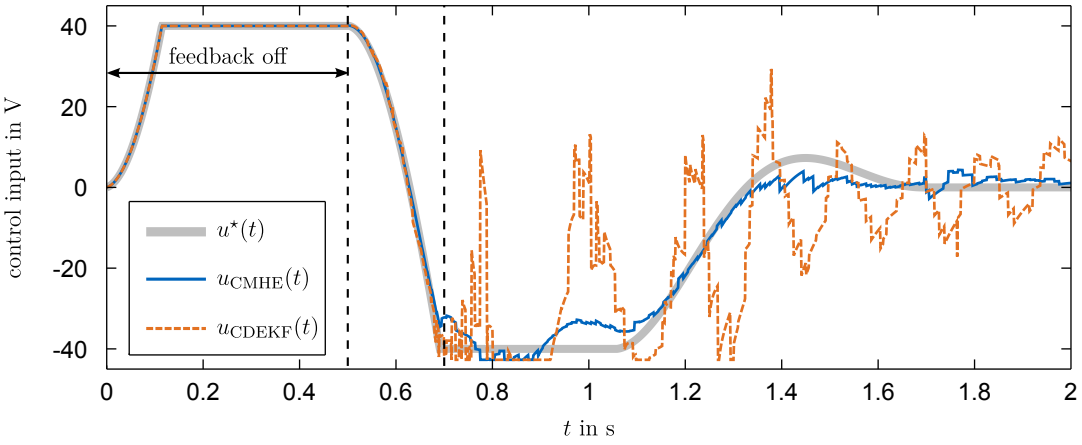


Figure 7.7: Comparison of the control inputs $u^\star(t)$, $u_{\mathrm{CDEKF}}(t)$ and $u_{\mathrm{CMHE}}(t)$ for the closed-loop case with the network scenario A depicted in Figure 7.11(a,b).

the feedback controller is designed with linear methods by linearizing the extended state space model around the nominal trajectories $\boldsymbol{x}^\star(t) = [{}_1x^\star(t), {}_2x^\star(t), {}_3x^\star(t), {}_4x^\star(t), {}_5x^\star(t)]^T$ (note that $x_5(t)^\star = \int_0^t x_1^\star(\tau)\, d\tau$) and $u^\star(t)$. This leads to the linear time-varying system

$$\Delta\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(t)\Delta\boldsymbol{x}(t) + \boldsymbol{b}(t)\Delta u(t), \tag{7.17a}$$

with

$$\Delta\boldsymbol{x}(t) \triangleq \boldsymbol{x}(t) - \boldsymbol{x}^\star(t), \qquad \boldsymbol{A}(t) \triangleq \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}^\star(t),u(t)} \tag{7.17b}$$

$$\Delta u(t) \triangleq u(t) - u^\star(t), \qquad \boldsymbol{b}(t) \triangleq \left.\frac{\partial \boldsymbol{f}}{\partial u}\right|_{\boldsymbol{x}^\star(t),u(t)}. \tag{7.17c}$$

For this system, we design an optimal linear quadratic time varying feedback control $\Delta u(t) = -\boldsymbol{k}(t)\Delta\boldsymbol{x}(t)$ which minimizes the cost functional

$$J = \frac{1}{2}\Delta\boldsymbol{x}(T)^T\boldsymbol{S}\Delta\boldsymbol{x}(T) + \frac{1}{2}\int_0^T \Delta\boldsymbol{x}(t)^T\boldsymbol{Q}\Delta\boldsymbol{x}(t) + \Delta u(t)R\Delta u(t)\, dt, \tag{7.18}$$

where $\boldsymbol{S}, \boldsymbol{Q} \in \mathbb{R}^{5\times5}$ are symmetric positive semidefinite matrices and $R > 0$ is a positive scalar. The solution $\boldsymbol{P}(t)$, $t \in [0, T]$ of the Riccati ODE

$$\dot{\boldsymbol{P}}(t) = -\boldsymbol{P}(t)\boldsymbol{A}(t) - \boldsymbol{A}^T(t)\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{b}(t)R^{-1}\boldsymbol{b}^T(t)\boldsymbol{P}(t) - \boldsymbol{Q} \tag{7.19a}$$

$$\boldsymbol{P}(T) = \boldsymbol{S} \tag{7.19b}$$

determines the feedback gains

$$\boldsymbol{k}(t) = R^{-1}\boldsymbol{b}(t)^T\boldsymbol{P}(t). \tag{7.20}$$

The terminal condition $\boldsymbol{P}(T) = \boldsymbol{S}$ for the reverse time integration of the Riccati equation (7.19a) is determined by solving the algebraic Riccati equation following from (7.19a) with $\dot{\boldsymbol{P}} = \boldsymbol{0}$. First, the MATLAB function `lqr` is used to calculate $\boldsymbol{S}$ and afterwards, a standard ODE solver of MATLAB is used to perform the reverse-time integration.

The weighting matrices in (7.18) are chosen to $\boldsymbol{Q} = \mathrm{diag}(10^7, 10^3, 10^5, 10^3, 10^3)$ and $R = 10$. The end time T is 11.1 s according to Table 7.3. In Figure 7.8, the time-varying feedback gains $\boldsymbol{k}(t) = [{}_1k(t), {}_2k(t), {}_3k(t), {}_4k(t), {}_5k(t)]$ are shown in the time interval $t = [0, 2\,]s$. Outside of this interval, the feedback gains are only subject to minor variations and are thus omitted from presentation. As the CDEKF and the CMHE are initialized with wrong initial conditions, the feedback control is turned off during the first 0.5 s. On the one hand, this provides enough time for the estimators to converge and on the other hand, to prevent the feedback controller from acting based on wrong estimated states of the system. In the bordering interval $t \in [0.5, 0.7\,]s$, the feedback gains $\boldsymbol{k}(t)$ are linearly interpolated between zero and the respective gain values in order to smoothly switch on the feedback controller.

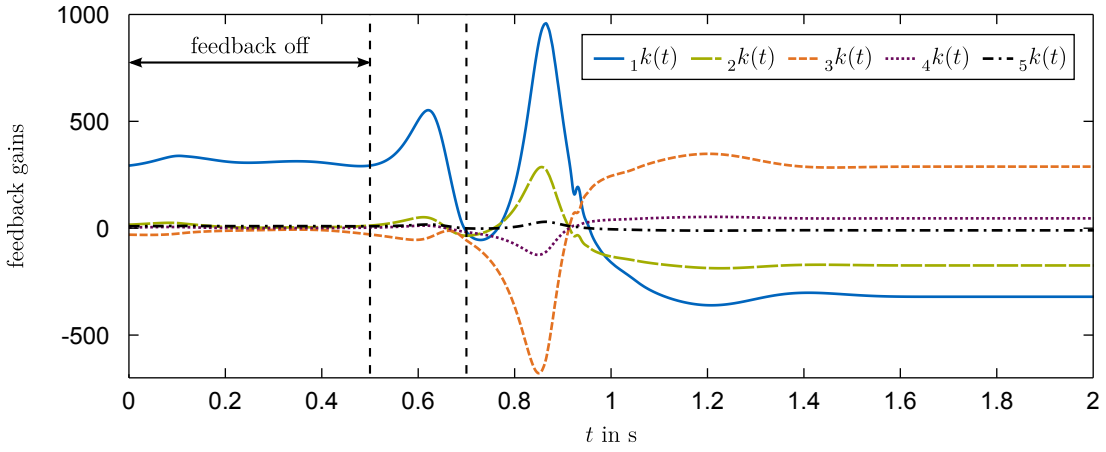


Figure 7.8: Illustration of the time-varying feedback gains $\boldsymbol{k}(t) = [{}_1k(t), {}_2k(t), {}_3k(t), {}_4k(t), {}_5k(t)]$.

7.2.5 Centralized Moving Horizon Estimator

The CMHE utilizes the nominal choice of the update step which minimizes the cost function

$$J_k(\hat{\boldsymbol{p}}_k) = \frac{1}{2}\left\| \begin{bmatrix} \hat{\alpha}_k \\ \hat{\beta}_k \\ \hat{\boldsymbol{x}}_{k-N|k} \end{bmatrix} - \begin{bmatrix} \bar{\alpha}_k \\ \bar{\beta}_k \\ \bar{\boldsymbol{x}}_{k-N|k} \end{bmatrix} \right\|^2_{\boldsymbol{P}_k^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k} \|\boldsymbol{h}(\hat{\boldsymbol{\phi}}_{i|k}(\hat{\boldsymbol{p}}_k, \boldsymbol{u})) - \boldsymbol{y}_{i|k}\|^2_{\boldsymbol{R}^{-1}} + \frac{1}{2}\sum_{i=k-N}^{k-1} \|\hat{\boldsymbol{w}}_{i|k}\|^2_{\boldsymbol{Q}^{-1}}. \tag{7.21}$$

The initial value of the CMHE is chosen to $\hat{\boldsymbol{x}}^\circ = [0.2, -0.2, -\pi{+}0.2, -0.2]^T$ ($\boldsymbol{x}^\circ = [0, 0, -\pi, 0]^T$) and the clock parameters are initialized by Method 2, see Section 3.6.2. This leads, e. g. for the open-loop scenario depicted in Figure 7.9, to $\hat{\alpha}^\circ_{N+1} = 1$ ($\alpha_{N+1} = 1.2$) and $\hat{\beta}^\circ_{N+1} = -2.485\,\mathrm{s}$ ($\beta_{N+1} = -3\,\mathrm{s}$). The prediction values are initialized by $\bar{\alpha}_{N+1} = \hat{\alpha}^\circ_{N+1}$, $\bar{\beta}_{N+1} = \hat{\beta}^\circ_{N+1}$ and $\bar{\boldsymbol{x}}_{k-N|k} = \hat{\boldsymbol{\phi}}_{\mathrm{pre}}(t_{N+1|N+1}, 0, \hat{\boldsymbol{x}}^\circ, \boldsymbol{u})$. The weighting matrices are set to $\boldsymbol{P}_k^{-1} = \min\{10^{0.1^k}, 10^6\}\,\boldsymbol{I}$,

$\boldsymbol{R}^{-1} = \mathrm{diag}(200, 4)\cdot 10^6$ and $\boldsymbol{Q}^{-1} = \mathrm{diag}(200, 20, 4, 1)\cdot 10^2$. Thereby, the increase of the weighting $\boldsymbol{P}_k^{-1}$ in the first 60 steps reflects the gradually gained confidence in the predicted values from initially low $10^{0.1}$ to finally high 10^6. The vector field of the pendulum dynamic (7.7) is used for the definition of the function $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\phi}}_{\mathrm{pre}}$ in Definition 3.5.2 and 3.7.1, respectively. Although the minimal buffer size is $N + 1 = 3$, we have set the actual buffer size to $N + 1 = 25$ to enable the incorporation of almost all successfully received packets in the estimation process which contain outdated information.

The predefinition of the decreasing factors ξ_i is possible but not the most efficient way to fully exploit the computing power of the target machine. This can be seen by the following consideration. Each optimization problem requires a different number of iteration steps of the optimization algorithm until the decreasing condition is satisfied. This leads to different execution times for the estimator. However, a guaranteed fixed upper bound on the execution time is required for the implementation on the real-time target machine. The obvious remedy is to set this bound to the largest possible execution time, if procurable. Of course this approach is conservative since it contains a lot of idle time.

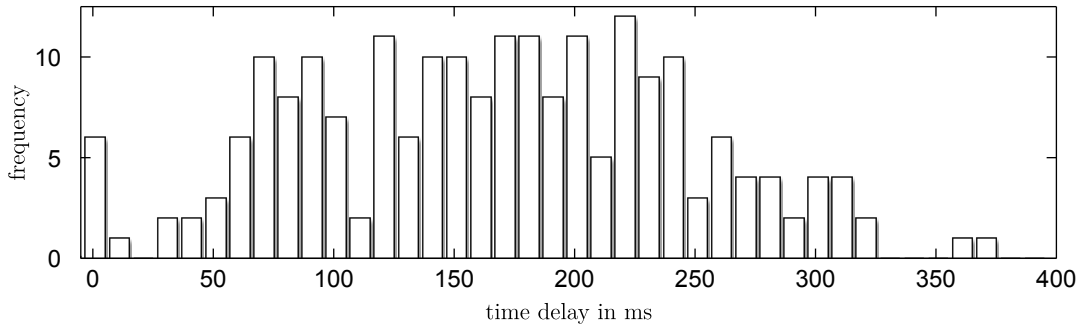
Thus, we introduce the following modifications which exploit the dual core support of the xPC Target Kernel. All elements except the updater Σ_{U} are performed on the first CPU core with the above mentioned sampling time $T_A = 1\,\mathrm{ms}$. Only the updater Σ_{U} is executed on the second CPU core with the sampling time T_B. To guarantee the observance of a fixed T_B and to exploit the computing power of the target machine, we modify the updater Σ_{U} as follows. Whenever the updater Σ_{U} is executed, it first checks if the buffer Σ_{B} has changed. If this is the case, the first iteration step of the new optimization problem is performed. Otherwise, one additional iteration step of the previous optimization problem is executed. In any case, the predictor Σ_{P} is updated afterwards. This approach allows a sampling time T_B as low as 6 ms which is broken down as follows. The calculation of the cost function (7.21), its gradient and Hessian according to Corollary 4.3.8 and Proposition 4.5.2 ($\boldsymbol{H}_2$ is neglected), respectively, takes 1.4 ms. Note that a conventional finite-difference approximation of the derivatives takes 244 ms and is thus 174 times slower and therefore not suitable for the real-time implementation. The execution of the IPOPT solver (Wächter and Biegler, 2006), which is used for calculating one iteration step, requires 4.3 ms. Since this solver is written in C, we have developed a C MEX S-FUNCTION[3] such that IPOPT can be used in both MATLAB/SIMULINK and REAL-TIME WORKSHOP and thus for the real-time target machine. In summary, we efficiently utilize the computing power of the target machine at the cost of not predefining the decreasing factors ξ_i directly but indirectly via the number of iteration steps.

[3] http://www.mathworks.de/help/toolbox/simulink/slref/sfunction.html

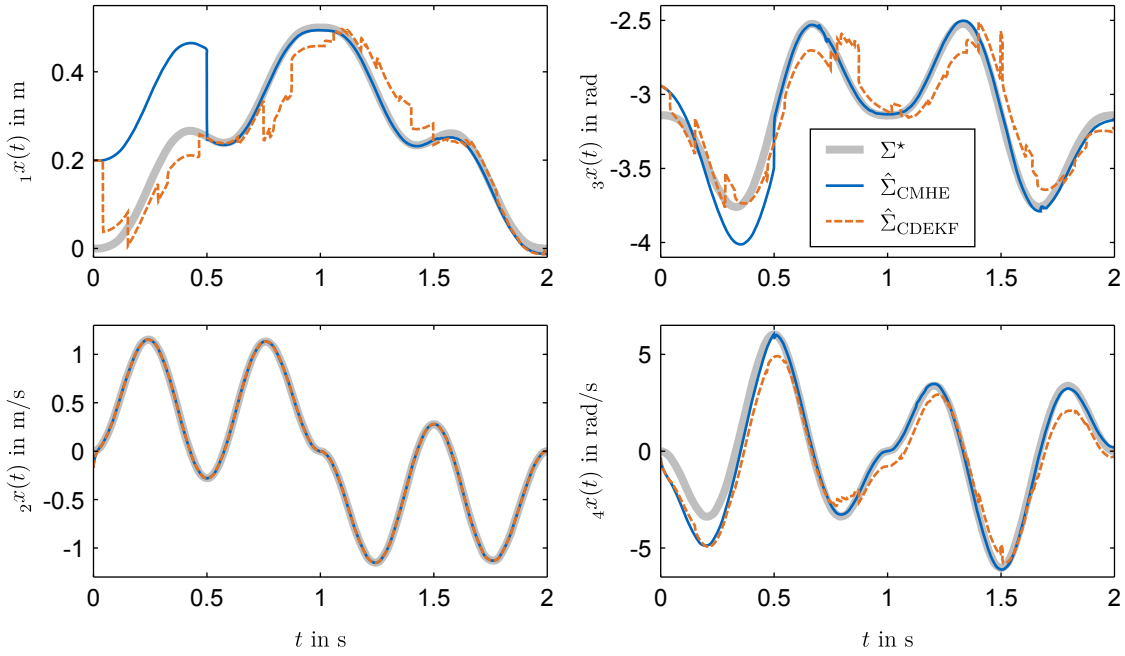
7.2.6 Experimental Results

7.2.6.1 Open-Loop Results

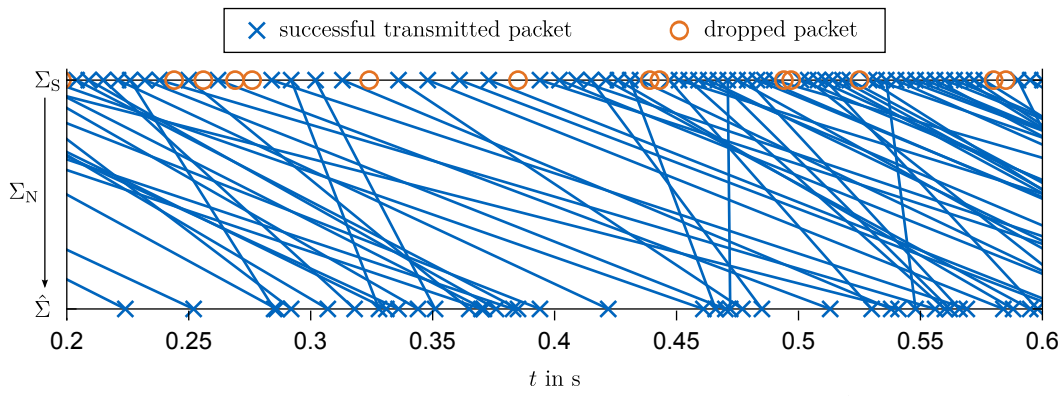
Figure 7.9 presents the results for the open-loop case which involves the transition between two stable downward equilibria, see Table 7.3. The chosen network protocols and settings



(a) Histogram of the time delays τ_i of the successful transmitted packets $\mathcal{P}_i$.



(b) Comparison between the nominal trajectory $\boldsymbol{x}^*(t)$ and the estimated states $\hat{\boldsymbol{x}}(t)$.



(c) Run of the packets $\mathcal{P}_i$ from the sensor Σ_{S} through the network Σ_{N} to the estimator $\hat{\Sigma}$.

Figure 7.9: Illustration of the open-loop results.

result in a packet loss rate of 19.8 % and in a time-delay distribution of the successfully transmitted packets $\mathcal{P}_i$ within the interval $[0\,\mathrm{ms}, 370\,\mathrm{ms}]$. The histogram of this distribution is shown in Figure 7.9(a).

The run of the packets $\mathcal{P}_i$ from the sensor Σ_S through the network Σ_N to the estimator $\hat{\Sigma}$ is schematically illustrated in Figure 7.9(c). The packets $\mathcal{P}_i$ generated by the sensor Σ_S are marked with either a blue cross or a red circle, depending on whether or not the corresponding packet is successfully transmitted. These packets are located on the vertical axis at the level of Σ_S at the sensor time stamps $\bar{t}_i$ transformed into the global time, i. e. $\alpha_i\,\bar{t}_i + \beta_i$. Note that the non-equidistant spacing of the packets is caused by the event-based sampling strategy of the sensor. The successfully received packets $\mathcal{P}_{i,j}$ are marked with a blue cross and are located on the vertical axis at the level of $\hat{\Sigma}$ at the arrival time stamps t_j. Each pair $(\mathcal{P}_i, \mathcal{P}_{i,j})$ is connected by a solid line. Note that packet reordering has taken place whenever two lines intersect.

Figure 7.9(b) shows the comparison between the nominal trajectory $\boldsymbol{x}^\star(t)$ and the estimated states $\hat{\boldsymbol{x}}(t)$ generated by the CDEKF and CMHE. In contrast to the CMHE, the CDEKF conducts the first update step right after the first packet has arrived. The CMHE, however, has to wait until 25 packets are stored in the buffer Σ_B which is approximately after 0.5 s. Afterwards, the CMHE significantly outperforms the CDEKF. The estimation error of the CMHE for the clock parameters is within 2 %. If we look closely, then we can observe that the estimates generated by the CDEKF are shifted by the unknown time delays which cannot be compensated. Consequently, the CDEKF performs only reasonable when the trajectory dynamics are slow compared to the time delays, i. e. $\boldsymbol{y}(t_i) \approx \boldsymbol{y}(t_i - \tau_i)$.

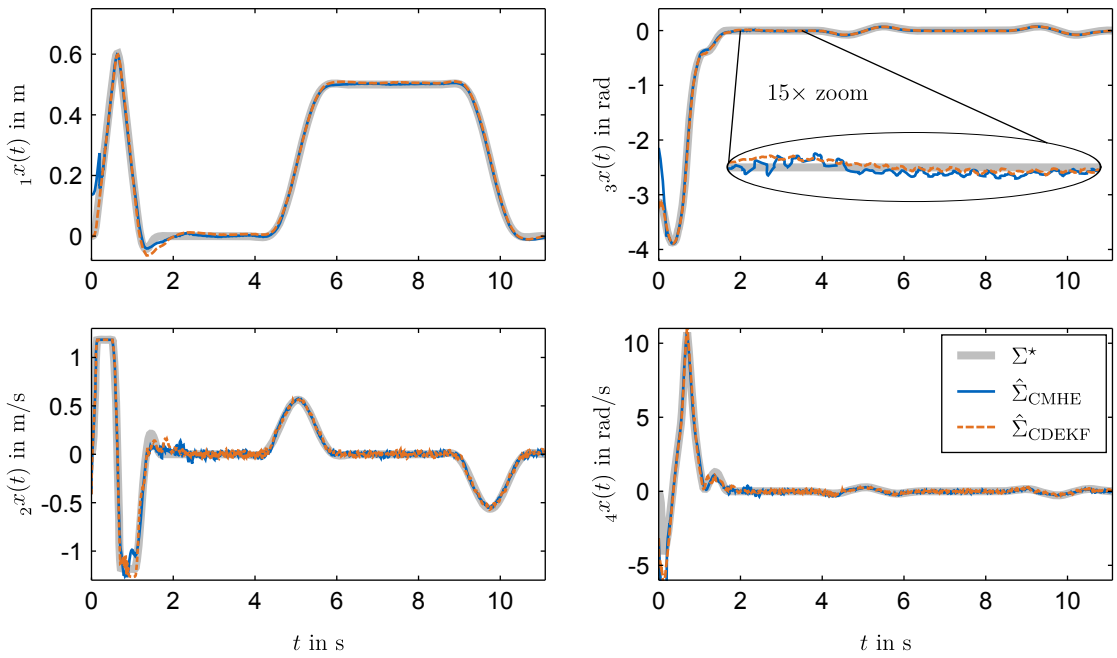
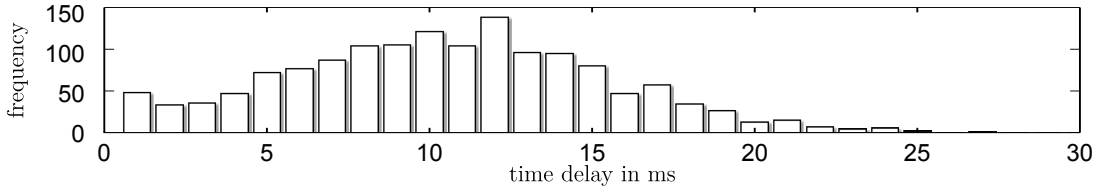
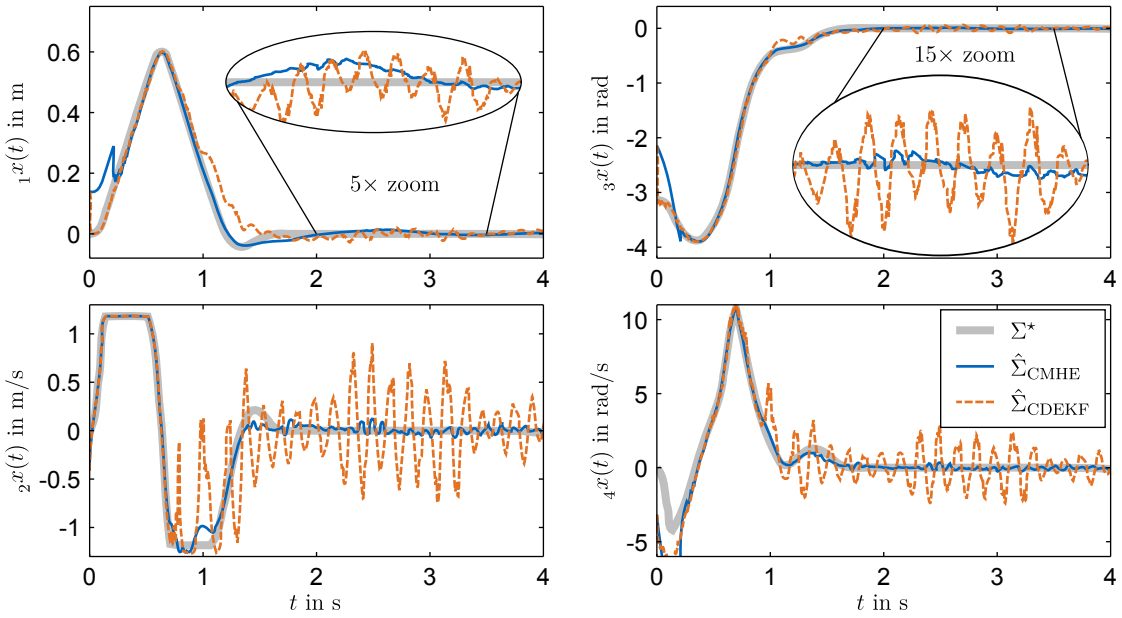


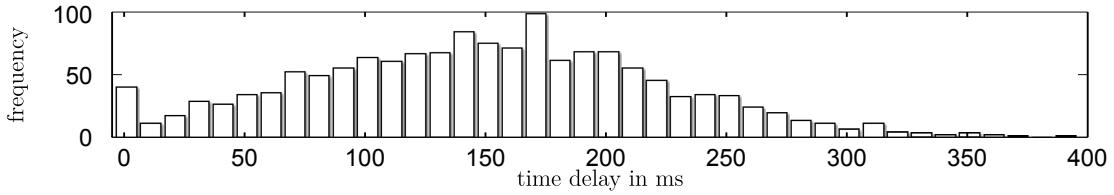
Figure 7.10: Illustration of the closed-loop results without a network.



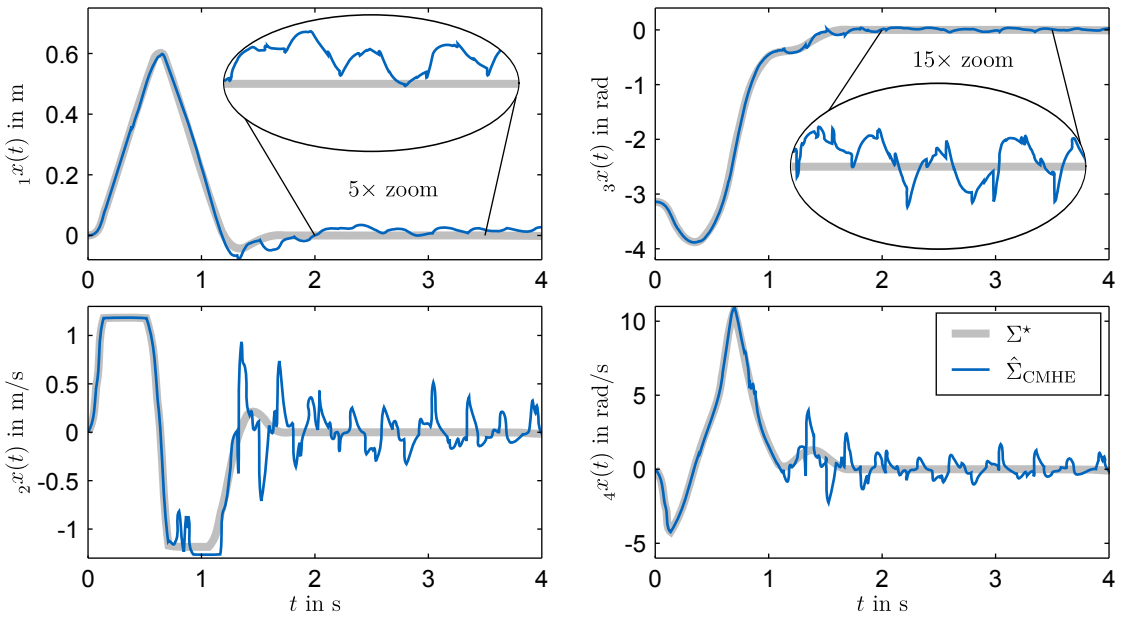
(a) Scenario A: Histogram of the time delays τ_i of the successful transmitted packets $\mathcal{P}_i$.



(b) Scenario A: Comparison between the nominal trajectory $\boldsymbol{x}^*(t)$ and the estimated states $\hat{\boldsymbol{x}}(t)$.



(c) Scenario B: Histogram of the time delays τ_i of the successful transmitted packets $\mathcal{P}_i$.



(d) Scenario B: Comparison between the nominal trajectory $\boldsymbol{x}^*(t)$ and the estimated states $\hat{\boldsymbol{x}}(t)$.

Figure 7.11: Illustration of the closed-loop results for the network scenario A and B.

7.2.6.2 Closed-Loop Results

The objective in the closed-loop scenario is to swing-up the pendulum followed by a transition between two unstable upward equilibria, see Table 7.3, for three different network scenarios: the nominal case, scenario A and B depicted in Figure 7.10, 7.11(a,b) and 7.11(c,d), respectively. The chosen network protocols and settings for scenario A and B result in a packet loss rate of 4.8 % and 15.2 % and in a time-delay distribution of the successfully transmitted packets $\mathcal{P}_i$ in the intervals $[0\,\text{ms}, 27\,\text{ms}]$ and $[0\,\text{ms}, 390\,\text{ms}]$, respectively. The histogram of these time delay distributions for the scenario A and B are shown in Figure 7.11(a) and 7.11(c), respectively. The last scenario results in an especially challenging control task, since the pendulum has a high dynamic due to the short pendulum length and the low moment of inertia (cf. Table 7.2). Starting from the initial condition $\boldsymbol{x}^\circ = \pi/180\,[0, 0, 2, 20]^T$, the pendulum rotates in 200 ms about 8.8° and in 400 ms about 30.4°. Thus, the influence of the time delays is significant.

The results for the nominal case, i. e. with an ideal network, are depicted in Figure 7.10 while the ones for scenario A and B are shown in Figure 7.11. In order to improve the comparability of the estimated states resulting from the three settings, the same scale is used for every plot and particular sections of the cart position and the rod angle are magnified by a factor of 5 and 15, respectively. The observability of the scenario B has been investigated in Section 5.5.3. While the CDEKF performs quite nicely in the nominal case, the closed-loop system is near the border of stability for scenario A. The corresponding control input is depicted in Figure 7.7. Note that it was not possible to either swing-up or stabilize the pendulum with the CDEKF for the harsh setting of scenario B. In contrast, the CMHE can cope with all three settings without major impact. Of course, the performance decreases with increasing time delays. As in the open-loop case, the estimation error of the CMHE for the clock parameters is within 2 %.

7.3 Summary

In this chapter, we have presented simulation and experimental results for the developed centralized moving horizon strategies and the continuous-discrete extended Kalman filter. We have analyzed the performance for each of these three methods by evaluating the simulation results of a networked version of a common nonlinear benchmark system. Without the influence of a network but with small disturbances, the performance of the CDEKF was almost comparable to those of both CMHS. For increasing delays, however, the CMHO and the CMHE significantly outperform the CDEKF. While in this scenario the CMHO is for low disturbances slightly superior to the CMHE, the situation changes in favor to the CMHE for increasing disturbances. By choosing the decreasing factors to $\xi_i = 0$, we could verify the convergence of the CMHS in finite-time.

The experimental results have been conducted on a networked pendulum test rig within a two-degree-of-freedom control scheme for different network scenarios. These confirm the conclusions derived in simulation and underline the performance of the CMHE in general and especially in comparison to the CDEKF. In order to achieve a satisfactory real-time performance of the CMHE, we have developed and implemented two additional steps to increase the efficient utilization of the available computing power of the target machine. First, we have indirectly predefined the decreasing factors via the number of iteration steps. Second, we have divided the overall execution task into two tasks which run on separate CPU cores.

Chapter 8

Conclusions and Future Work

In Part II of this thesis, we have presented the *centralized moving horizon observer* (CMHO) and the *centralized moving horizon estimator* (CMHE) for state estimation within the undisturbed and disturbed centralized NCS architecture, respectively. Furthermore, we have developed methods which facilitate the real-time implementation of both strategies which includes structure-exploiting techniques to efficiently calculate the involved derivatives. We have also introduced a notion of observability for undisturbed and disturbed NCS and have derived necessary conditions for the control input to be sufficiently exciting which enables the estimation of all unknown parameters. Finally, we have analyzed the stability of the CMHO as well as the CMHE and have validated their performance not only in simulations of a networked benchmark system but also in experiments on a networked pendulum test-rig.

In this chapter, we summarize each chapter of Part II of this thesis. In addition, we also outline possible and natural extensions, as well as broader ideas for future work.

8.1 Summary

Chapter 3: Centralized Moving Horizon Strategies

In this chapter, we have presented the CMHO and the CMHE within a common framework for the undisturbed and the disturbed centralized NCS architecture, respectively. Both strategies have been explicitly designed to deal simultaneously with the network-induced imperfections of unknown and variable packet delays which include the possibility of packet reordering, unknown and variable packet drops, unsynchronized sensor clocks, and limited energy supplies of the sensors. To overcome these challenges, we have introduced event-based sampling along with time stamping of the resulting measurements. Moreover, we have proposed an affine clock model for the sensor clocks and have extended the moving horizon to a buffer logic. This buffer serves as the information basis for both centralized moving horizon strategies and has enabled us along with the other introduced steps to formulate the state estimation problems as suitable optimization problems where we additionally estimate the unknown clock parameters of the sensors. To achieve a practical feasible implementation of these optimization problems in real-time and thus of the CMHO and CMHE, we have proposed the following two steps which

significantly contribute to this goal. First, we have introduced a suboptimal approach which requires only suboptimal instead of optimal solutions. Second, we have provided efficient methods for generating proper initial conditions for finding (sub)optimal solutions.

Chapter 4: Efficient Derivative Calculation

In this chapter, we have presented an efficient, parallelizable and sensitivity-based method to calculate the gradient and Hessian of the Lagrangian to the optimization problem of the CMHO and CMHE. The gradient computation method has been derived by applying the chain rule to the Lagrangian and depends on the respective first-order state sensitivities. These sensitivities have been calculated based on the solution of first-order sensitivity matrix differential equations as an answer to the difficulties stemming from the special nature of the optimization problem formulation where the clock parameters arise in the integral bounds. This perception facilitates the accomplishment of exploiting the structure of the state sensitivities such that the number of ODEs required for calculating the gradient in the CMHE case is independent of the number of optimization variables and equal to the adjoint case, namely $2n_x^2 + n_x$. The Hessian computation method is founded on the idea of partitioning the exact Hessian into two parts. The first part can be readily calculated due to the already available first-order state sensitivities while the second part can either be approximated by a modified BFGS method or completely neglected. In contrast to the adjoint method, the proposed sensitivity-based method provides an excellent approximation of the Hessian for arbitrary cost functions almost for free. The theoretical benefits have been substantiated by a numerical case study of a continuously-stirred tank reactor where the proposed method has been compared to the finite-difference approach.

Chapter 5: Observability of Networked Control Systems

In this chapter, we have introduced an observability notion for undisturbed and disturbed NCSs. This notion ensures (at least locally) the well-posedness of the respective observation map by guaranteeing the full-rankness of its derivative with respect to $\mathrm{col}(\hat{\alpha}_k, \hat{\beta}_k, \hat{\boldsymbol{x}}_{k-N|k})$. The observability of NCSs does not only depend on the structure of the system but also on the information content of the control input. To this end, we have derived for the undisturbed as well as for the disturbed NCSs a necessary condition for the control input to be sufficiently exciting. The key idea for achieving this condition was to establishing a relation between the control input and the first-order state sensitivities appearing in the elements of the derivative of the observation map. The resulting conditions are almost identical for the undisturbed and disturbed NCSs and the only difference stems from the dependence of the estimated state disturbance $\hat{\boldsymbol{w}}(t)$ on the clock parameter $\hat{\alpha}$. Moreover, we have revealed the relation between the introduced notion of observability to the conventional Kalman observability matrix for linear systems and to the update step as well as to the gradient and Hessian of the Lagrangian of the CMHO and CMHE. Finally, we have clarified the introduced terminology and the derived conditions by a linear example system and pointed out their practical relevance by investigating a real networked pendulum test-rig.

Chapter 6: Stability Analysis

In this chapter, we have analyzed the stability of the nominal CMHO and the nominal CMHE. Both analyses are founded on the idea of expressing gradient-based optimization algorithms as a single unifying continuous-time dynamical system. The essential characteristic of this system is that its vector field is the product of an arbitrary positive definite matrix function and the negative gradient of the cost function derived in Chapter 4. The freedom in choosing this matrix has opened up the possibility of selecting any line search or trust region method presented in Section 2.4.2. This perception has facilitated the accomplishment of the stability analysis in three steps. In the first step, we have shown that if the NCS is observable in the sense derived in Chapter 5, then the aforementioned unifying continuous-time system possesses an isolated asymptotically stable equilibrium which is the optimal solution to the optimization problem stemming from the corresponding update step. We have proven in the second step that if the suboptimal approach satisfies the decreasing condition, then the suboptimal solutions approach to the optimal ones. In the third step, we have concluded from this fact and by invoking the continuous dependence of initial value problems the following results for the disturbed and undisturbed scenario. For the former case, we have shown boundedness of the CMHE estimation error. For the latter case, i. e. in the absence of disturbances, we have proven asymptotic and even finite-time convergence of the CMHO observation error depending on the choice of the decreasing factors. These factors allow to set up a compromise between the achievable convergence speed and the required computation time. It should be pointed out that the main feature of these results is its generality. This means that the derived analysis hold for the general assumptions made in the framework presented in Chapter 3 which includes, inter alia, the quite general packet delay and packet dropout assumptions.

Chapter 7: Simulation and Experimental Results

In this chapter, we have presented simulation and experimental results for the developed centralized moving horizon strategies and the continuous-discrete extended Kalman filter. We have analyzed the performance for each of these three methods by evaluating the simulation results of a networked version of a common nonlinear benchmark system. Without the influence of a network but with small disturbances, the performance of the CDEKF was almost comparable to those of both CMHS. For increasing packet delays, however, the CMHO and the CMHE significantly outperform the CDEKF. While in this scenario the CMHO is for low disturbances slightly superior to the CMHE, the situation changes in favor to the CMHE for increasing disturbances. By choosing the decreasing factors to $\xi_i = 0$, we could verify the convergence of the CMHS in finite-time.

The experimental results have been conducted on a networked pendulum test rig within a two-degree-of-freedom control scheme for different network scenarios. These confirm the derived simulative conclusions and underline the performance of the CMHE in general and especially in comparison to the CDEKF. In order to achieve a satisfactory real-time perfor-

mance of the CMHE, we have developed and implemented two additional steps to increase the efficient utilization of the available computing power of the target machine. First, we have indirectly predefined the decreasing factors via the number of iteration steps. Second, we have divided the overall execution task into two tasks which run on separate CPU cores.

8.2 Future Work

Moving horizon strategies are a powerful tool for state estimation with a vast potential of multifaceted industrial applications. With the explicit consideration of network induced imperfections, the research presented in Part II of this thesis provides a significant contribution to extend the range of application to networked control systems. However, there are still unresolved issues. Some recommendations for future work are the following:

- *Optimal choice of the degrees of freedom* - The presented CMHS provide several degrees of freedom like the choice of the weighting matrices, the buffer size and the decreasing factors. On the one hand, these parameters allow the customization of the CMHS to the specific circumstances of the particular application. On the other hand, the choice of these parameters is crucial for the estimation performance and can lead in the worst case to unstable observers/estimators. The conditions derived in Chapter 6 guarantee convergence of the observation error to zero for the CMHO and boundedness of the estimation error for the CMHE. However, these conditions do not answer the following two questions which are important from a practical point of view: What are admissible ranges of the free parameters such that the results derived in Chapter 6 hold? What is the optimal parameter combination such that the smallest estimation error and thus the optimal performance of the CMHE can be obtained? Therefore, it is desirable to develop methods which replace the current trial and error approach for choosing the parameters of the CMHS by a systematic procedure.
- *Stability analyses for constrained state estimation* - A favorable property of the moving horizon strategies for state estimation is the possibility of directly handling constraints on the states as well as the disturbances. This is useful from an engineering point of view since in practice often additional information in form of constraints is available. For instance, the range of some values in the system is known. Incorporating this prior knowledge into the estimation process might improve the performance of the CMHS. While the practical implementation of this approach has been shown in Part II of this thesis, the stability analysis for constrained state estimation remains unanswered.
- *Simple, robust and fast algorithms* - One of the biggest obstacles which prevents moving horizon strategies for state estimation from broader application in industrial environments is the complexity involved with setting up such methods. This is mainly due to the lack of optimization algorithms which are specially tailored to satisfy the needs of moving horizon strategies. The desired algorithms should be simple to set up, sufficiently

robust to operate in practical environments and fast enough to satisfy the real-time requirements of the particular application. In Part II of the thesis, we have proposed a first step in this direction by combining powerful off-the-shelf optimization algorithms with the structure exploiting derivative methods presented in Chapter 4. The next step could be the development of an algorithm which eases the overall set up by automatically providing the efficient calculated derivatives to off-the-shelf optimization algorithms. In a broader scope, another desirable direction is to develop algorithms which facilitates automatic code generation and implementation on embedded hardware, e.g. field programmable gate arrays (FPGAs) or digital signal processors (DSPs). This would further reduce computation times and widen the application scope.

Part III

Distributed Moving Horizon Strategies

Chapter 9

Distributed Moving Horizon Strategies

In this chapter, we develop the distributed moving horizon observer (DMHO) and the distributed moving horizon estimator (DMHE) within a common framework for the undisturbed and disturbed distributed NCS architecture, respectively. As discussed in Chapter 1, the distributed NCS architecture combines the benefits of the centralized (Kailath, 1980; Lunze, 1992; Goodwin et al., 2001; Khalil, 2002) and decentralized (Sandell et al., 1978; Siljak, 2007; Bakule, 2008) framework of conventional control systems. The idea is to exploit the capabilities of modern actuators and sensors which are equipped with small microcontrollers. These offer computational capabilities as well as wired or wireless intercommunication and thus render a central processing unit redundant. To overcome the performance limitations of decentralized control approaches, a centralized controller is designed but implemented in a distributed manner on the actuators. This requires distributed knowledge of all states of the system which is provided by both distributed moving horizon strategies (DMHS).

Distributed problems arise in several areas like in decentralized receding horizon control (Keviczky et al., 2006), quasi-decentralized state estimation and control (Sun and El-Farra, 2008), distributed estimation (Farina et al., 2010), consensus problems (Olfati-Saber and Murray, 2004) and in dynamic networks (Dai and Mesbahi, 2011). The common concept is to accomplish an overall objective while multiple subsystems interact with one another.

If the objective can be posed as an optimization problem, the dual decomposition technique (see Section 2.4.4.2) constitutes a powerful tool to distribute an optimization problem into smaller subproblems. Successful applications have been recently reported in several areas. In particular, the dual decomposition is used in Xiao et al. (2004) for simultaneous routing and resource allocation in data networks, in Raffard et al. (2004) for trajectory optimization of formation flights and in Rantzer (2009) for distributed control. Another field of application is distributed model predictive control (DMPC), where in Wakasa et al. (2008) a DMPC method is developed for systems consisting of multiple SISO subsystems with coupling output constraints and a fixed communication chain structure. In Giselsson and Rantzer (2010), a stopping criterion is designed for DMPC such that the closed-loop performance above a certain pre-specified degree is achieved and asymptotic stability of the closed-loop system is guaranteed. Dual decomposition has also been applied in distributed estimation for static problems (Samar et al., 2007) and has been used for distributed receding horizon Kalman filtering for

systems consisting of multiple subsystems with coupling output constraints (Maestre et al., 2010).

Both DMHS are essentially derived by the dual decomposition technique applied to the respective centralized moving horizon strategies (CMHS) presented in Part II of this thesis. In particular, the main steps to derive the distributed versions of the CMHS are as follows: First, we investigate the issue of how to allocate the measurements from the various sensors to the DMHO/DMHE. In a next step, we model the communication topology among the distributed estimators as a directed graph and extend the optimization problems of the centralized moving horizon strategies by additional consensus constraints which reflect the interconnection structure. Finally, we apply the dual decomposition technique to reveal a separable dual problem which we solve by a suitable subgradient method to achieve a distributed algorithm.

The remainder of this chapter is organized as follows. In Section 9.1, we introduce the distributed NCS architecture and the resulting two problem definitions for the disturbed and undisturbed scenario. At the heart of this chapter in Section 9.2, we develop the DMHS based on the CMHS for the given scenarios. The resulting algorithm of the DMHO and DMHE is presented in Section 9.3. The stability of both DMHS is analyzed in Section 9.4. In Section 9.5, we derive a distributed Kalman filter as a special case of the DMHE. In Section 9.6, we sketch several extensions for the DMHS. Finally, we conclude this chapter with a summary given in Section 9.7.

9.1 Problem Formulation

Consider the distributed NCS architecture depicted in Figure 9.1. The plant Σ is described

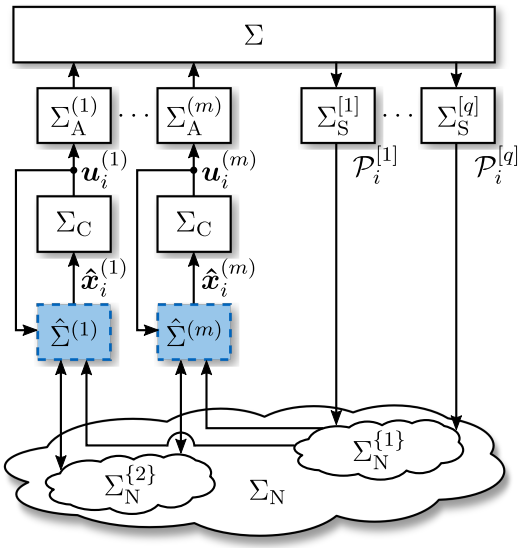


Figure 9.1: Distributed NCS architecture with system Σ, sensors $\Sigma_{\mathrm{S}}^{[j]}$, network Σ_{N} consisting of networks $\Sigma_{\mathrm{N}}^{\{1\}}$ and $\Sigma_{\mathrm{N}}^{\{2\}}$, observers/estimators $\hat{\Sigma}^{(l)}$, controller Σ_{C} and actuators $\Sigma_{\mathrm{A}}^{(l)}$.

by the discrete-time linear time-invariant system

$$\boldsymbol{x}_{i+1} = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\begin{bmatrix}\boldsymbol{u}_i^{(1)}\\ \vdots \\ \boldsymbol{u}_i^{(m)}\end{bmatrix} + \boldsymbol{w}_i, \tag{9.1}$$

where $\boldsymbol{x}_i \in \mathbb{R}^{n_x}$ is the state with the initial value $\boldsymbol{x}_0 \in \mathbb{R}^{n_x}$, $\boldsymbol{u}_i^{(l)} \in \mathbb{R}^{n_{u,l}}$, $l \in \mathcal{A} \triangleq \{1, 2, \ldots, m\}$ is one of the m control inputs, $\boldsymbol{w}_i \in \mathbb{R}^{n_x}$ is the state disturbance and $i \in \mathbb{N}_0$ is the discrete global time. By introducing the control input $\boldsymbol{u}_i \triangleq \operatorname{col}(\boldsymbol{u}_i^{(l)}, l \in \mathcal{A}) \in \mathbb{R}^{n_u}$, we can write (9.1) compactly as

$$\boldsymbol{x}_{i+1} = \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i + \boldsymbol{w}_i. \tag{9.2}$$

Each of the q sensors $\Sigma_{\mathrm{S}}^{[j]}$ with $j \in \mathcal{S} \triangleq \{1, 2, \ldots, q\}$ is equipped with an individual clock which possesses the discrete global time $i \in \mathbb{N}_0$. This means that all clocks in the architecture depicted in Figure 9.1 are synchronized. Moreover, each sensor generates a sequence of packets $\mathcal{P}_i^{[j]} = \{\boldsymbol{y}_i^{[j]}, i\}$ consisting of measurements $\boldsymbol{y}_i^{[j]} \in \mathbb{R}^{n_{y,j}}$ and corresponding time stamps $i \in \mathbb{N}_0$. These measurements are derived according to the measurement model

$$\boldsymbol{y}_i^{[j]} = \boldsymbol{C}^{[j]}\boldsymbol{x}_i + \boldsymbol{v}_i^{[j]}, \quad j \in \mathcal{S}, \quad i \in \mathbb{N}_0, \tag{9.3}$$

where $\boldsymbol{v}_i^{[j]} \in \mathbb{R}^{n_{y,j}}$ are the measurement disturbances. Each packet $\mathcal{P}_i^{[j]}$ is transmitted to one of the observers/estimators $\hat{\Sigma}^{(l)}$ over the network $\Sigma_{\mathrm{N}}^{\{1\}}$.

The key feature of the distributed NCS architecture consists of implementing decentralized any centralized designed controller Σ_{C}. In contrast to decentralized controllers, centralized controllers consider the overall system dynamics and thus have in general a superior performance compared to pure decentralized solutions. Just like in the centralized NCS architecture depicted in Figure 3.1, the controller Σ_{C} is designed for the nominal centralized case, i. e. without the network Σ_{N}, and is described as

$$\tilde{\boldsymbol{x}}_{i+1} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{r}_i), \tag{9.4a}$$

$$\boldsymbol{u}_i = \begin{bmatrix}\boldsymbol{u}_i^{(1)}\\ \vdots \\ \boldsymbol{u}_i^{(m)}\end{bmatrix} = \tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{r}_i) = \begin{bmatrix}\tilde{\boldsymbol{h}}^{(1)}(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{r}_i)\\ \vdots \\ \tilde{\boldsymbol{h}}^{(m)}(\tilde{\boldsymbol{x}}_i, \boldsymbol{x}_i, \boldsymbol{r}_i)\end{bmatrix}, \tag{9.4b}$$

where $\tilde{\boldsymbol{x}}_i \in \mathbb{R}^{n_{\tilde{x}}}$ is the control state with the initial value $\tilde{\boldsymbol{x}}_0 \in \mathbb{R}^{n_{\tilde{x}}}$ and $\boldsymbol{r}_i \in \mathbb{R}^{n_r}$ is the reference input. The control input $\boldsymbol{u}_i^{(l)}$ of the l-th actuator results from the corresponding rows of (9.4b). However, the successful decentralized implementation of the controller Σ_{C} requires distributed and identical knowledge about the full state of the system. Since the measurement information is only distributed available, the observers/estimators have the possibility to exchange information over the network $\Sigma_{\mathrm{N}}^{\{2\}}$ in order to agree upon the estimates.

The separation of the network Σ_{N} into the two parts $\Sigma_{\mathrm{N}}^{\{1\}}$ and $\Sigma_{\mathrm{N}}^{\{2\}}$ is motivated by their different tasks which result in different properties. The network $\Sigma_{\mathrm{N}}^{\{2\}}$ is responsible for the intercommunication between the observers/estimators and describes a reliable wired high-speed network. In contrast, the network $\Sigma_{\mathrm{N}}^{\{1\}}$ models several different possibilities of measurement transmission from the sensors to the observers/estimators: conventional wired transmission, wired network transmission and wireless network transmission. Thus, the properties of $\Sigma_{\mathrm{N}}^{\{1\}}$ depend on the considered transmission types and might vary significantly from those of $\Sigma_{\mathrm{N}}^{\{2\}}$.

In order to set up reasonable problem formulations, we make the following assumptions:

Assumption 15. The initial value $\boldsymbol{x}_0$, the disturbances $\boldsymbol{w}_i$ and $\boldsymbol{v}_i^{[j]}$, $j \in \mathcal{S}$, are uncorrelated Gaussian sequences with mean $\bar{\boldsymbol{x}}_0$, $\mathbf{0}$ and $\mathbf{0}$ and covariance $\boldsymbol{P}_0 > \mathbf{0}$, $\boldsymbol{Q} > \mathbf{0}$ and $\boldsymbol{R}^{[j]} > \mathbf{0}$, $j \in \mathcal{S}$, i. e. $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{P}_0)$, $\boldsymbol{w}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$ and $\boldsymbol{v}_i^{[j]} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}^{[j]})$, $j \in \mathcal{S}$, respectively.

Assumption 16. The pair $(\boldsymbol{A}, \boldsymbol{B})$ is stabilizable.

Assumption 17. The pair $(\mathrm{col}(\boldsymbol{C}^{[j]}, j \in \mathcal{S}), \boldsymbol{A})$ is observable.

Assumption 18. Each packet $\mathcal{P}_i^{[j]}$ is $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}_0$ transmitted over the ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ to only one observer/estimator $\hat{\Sigma}^{(l)}$ which may vary over time.

Assumption 19. The packet transmission between two observers/estimators $\hat{\Sigma}^{(i)}$ and $\hat{\Sigma}^{(j)}$ over the network $\Sigma_{\mathrm{N}}^{\{2\}}$ is $\forall i, j \in \mathcal{A}$, $i \neq j$, either ideal or impossible.

Note that the Assumption 16 is not necessary for the design of the observers/ estimators $\hat{\Sigma}^{(l)}$. But it is reasonable from a practical point of view since it allows a controller design such that a satisfactory overall performance within the distributed NCS architecture can be achieved. In contrast, Assumption 17 is necessary since it facilitates the possibility of reconstructing the state vector by implying global observability. Assumption 18 ensures that the available measurements for each observer/estimator $\hat{\Sigma}^{(i)}$ are always different. This excludes the trivial solution which consists of implementing m identical centralized observers/estimators where each one would require all measurements. Compared to this trivial approach, Assumption 18 favors solutions which possess lower local computational load and can handle reduced local information. Note that Assumption 18 facilitates a time-varying measurement allocation which, e. g., is helpful for incorporating mobile sensors. Since the measurement information is only distributed available, the observers/estimators $\hat{\Sigma}^{(l)}$ have to exchange information in order to agree upon the estimates. Note that Assumption 19 imposes no restrictions on the topology of the network $\Sigma_{\mathrm{N}}^{\{2\}}$. Moreover, both Assumptions 18 and 19 state that the respective network, i. e. $\Sigma_{\mathrm{N}}^{\{1\}}$ and $\Sigma_{\mathrm{N}}^{\{2\}}$, are ideal, i. e. without packet delay nor packet drop. While this assumption is justified for $\Sigma_{\mathrm{N}}^{\{2\}}$, it might be inadequate for $\Sigma_{\mathrm{N}}^{\{1\}}$ to describe all of the above mentioned measurement transmission possibilities. Therefore, we will drop later on the assumption of an ideal network and present in Section 9.6.3 the extension to the case of a packet-delaying and packet-dropping network $\Sigma_{\mathrm{N}}^{\{1\}}$.

Problem 9.1 (Distributed Observer Design)**.** *Let the presented undisturbed distributed NCS architecture be given consisting of the undisturbed system* Σ*, i. e.* $\boldsymbol{Q} = \boldsymbol{0}$*, without information about the initial state, i. e.* $\boldsymbol{P}_0^{-1} = \boldsymbol{0}$*, the disturbed sensors* $\Sigma_S^{[j]}$*, i. e.* $\boldsymbol{R}^{[j]} > \boldsymbol{0}$*,* $j \in \mathcal{S}$*, the controller* Σ_C*, the actuators* $\Sigma_A^{(l)}$*,* $l \in \mathcal{A}$*, and the network* Σ_N*.*

The problem is to design under the given circumstances

- *m distributed observers* $\hat{\Sigma}^{(l)}$*,* $l \in \mathcal{A}$*,*
- $\forall j \in \mathcal{S}$ *and* $\forall i \in \mathbb{N}_0$ *the allocation of the measurements* $\boldsymbol{y}_i^{[j]}$ *over the network* $\Sigma_N^{\{1\}}$ *to the m distributed observers* $\hat{\Sigma}^{(l)}$*,* $l \in \mathcal{A}$*, and*
- *the topology of the network* $\Sigma_N^{\{2\}}$*,*

such that the current state of the system Σ *is reconstructed in a distributed manner where all observed states are identical to those of a centralized moving horizon observer, i. e.* $\hat{\boldsymbol{x}}_i^{(l)} = \hat{\boldsymbol{x}}_i^{CMHO}$*,* $\forall l \in \mathcal{A}$*,* $\forall i \in \mathbb{N}_0$*.*

Problem 9.2 (Distributed Estimator Design)**.** *Let the presented disturbed distributed NCS architecture be given consisting of the disturbed system* Σ*, i. e.* $\boldsymbol{Q} > \boldsymbol{0}$*, with information about the initial state, i. e.* $\boldsymbol{P}_0^{-1} > \boldsymbol{0}$*, the disturbed sensors* $\Sigma_S^{[j]}$*, i. e.* $\boldsymbol{R}^{[j]} > \boldsymbol{0}$*,* $j \in \mathcal{S}$*, the controller* Σ_C*, the actuators* $\Sigma_A^{(l)}$*,* $l \in \mathcal{A}$*, and the network* Σ_N*.*

The problem is to design under the given circumstances

- *m distributed estimators* $\hat{\Sigma}^{(l)}$*,* $l \in \mathcal{A}$*,*
- $\forall j \in \mathcal{S}$ *and* $\forall i \in \mathbb{N}_0$ *the allocation of the measurements* $\boldsymbol{y}_i^{[j]}$ *over the network* $\Sigma_N^{\{1\}}$ *to the m distributed estimators* $\hat{\Sigma}^{(l)}$*,* $l \in \mathcal{A}$*, and*
- *the topology of the network* $\Sigma_N^{\{2\}}$*,*

such that the current state of the system Σ *is reconstructed in a distributed manner where all estimated states are identical to those of a centralized moving horizon estimator, i. e.* $\hat{\boldsymbol{x}}_i^{(l)} = \hat{\boldsymbol{x}}_i^{CMHE}$*,* $\forall l \in \mathcal{A}$*,* $\forall i \in \mathbb{N}_0$*.*

It is important to note that throughout Part III of this thesis including the current chapter, the *Assumptions 15-19 are supposed to hold without explicitly stating them.*

9.2 From Centralized to Distributed Moving Horizon Strategies

Problem 9.1 and 9.2 require among other things that all distributed estimates are identical to those of a centralized moving horizon observer (CMHO) and a centralized moving horizon estimator (CMHE), respectively. The strategy we pursue to fulfill this requirement is to develop distributed versions of the CMHS. In this section, we present the steps to achieve this goal.

The starting point of this endeavor are the CMHS for which we assume that all packets of the sensors are transmitted over the ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ to a single CMHO/CMHE. Since the network is ideal and all clocks of the distributed NCS architecture are synchronized, we do not need either the clock model or the notation introduced in Section 3.3. By using the simpler notation of the moving horizon framework presented in Section 2.1, we can state the update step and the prediction step of the CMHS as follows.

Centralized Moving Horizon Observer

The update step of the CMHO for the given problem consists of the following optimization problem

$$\min_{\hat{\boldsymbol{x}}_{k-N},\ldots,\hat{\boldsymbol{x}}_k} J_k(\hat{\boldsymbol{x}}_{k-N},\ldots,\hat{\boldsymbol{x}}_k) \tag{9.5a}$$

subject to

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}\hat{\boldsymbol{x}}_i + \boldsymbol{B}\boldsymbol{u}_i, \quad i \in \overline{\mathcal{I}}_k, \tag{9.5b}$$

where the cost function is given by

$$J_k(\hat{\boldsymbol{x}}_{k-N},\ldots,\hat{\boldsymbol{x}}_k) = \frac{1}{2}\sum_{i=k-N}^{k}\sum_{j=1}^{q} \|\boldsymbol{C}^{[j]}\hat{\boldsymbol{x}}_i - \boldsymbol{y}_i^{[j]}\|^2_{\boldsymbol{R}^{[j]-1}}. \tag{9.5c}$$

Based on the solution of (9.5), the actual observed state is calculated by

$$\hat{\boldsymbol{x}}_{k-N+i} = \boldsymbol{A}^i\hat{\boldsymbol{x}}_{k-N} + \sum_{j=1}^{i}\boldsymbol{A}^{i-j}\boldsymbol{B}\boldsymbol{u}_{k-N-1+j}. \tag{9.6}$$

Note that $\hat{\boldsymbol{x}}_{k-N}$ is the best linear unbiased minimum-variance estimator for problem (9.5) (Humpherys and West, 2010). Moreover, the CMHO becomes a dead-beat observer in the case of noise-free measurements, i. e. $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k$, $\forall k \in \mathbb{N}_N$.

Centralized Moving Horizon Estimator

The update step of the CMHE for the given problem consists of the following optimization problem

$$\min_{\substack{\hat{\boldsymbol{x}}_{k-N},\ldots,\hat{\boldsymbol{x}}_k \\ \hat{\boldsymbol{w}}_{k-N},\ldots,\hat{\boldsymbol{w}}_{k-1}}} J_k(\hat{\boldsymbol{x}}_{k-N},\ldots,\hat{\boldsymbol{x}}_k,\hat{\boldsymbol{w}}_{k-N},\ldots,\hat{\boldsymbol{w}}_{k-1}) \tag{9.7a}$$

subject to

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}\hat{\boldsymbol{x}}_i + \boldsymbol{B}\boldsymbol{u}_i + \hat{\boldsymbol{w}}_i, \quad i \in \overline{\mathcal{I}}_k, \tag{9.7b}$$

where the cost function is given by

$$J_k(\hat{\boldsymbol{x}}_{k-N},\dots,\hat{\boldsymbol{x}}_k,\hat{\boldsymbol{w}}_{k-N},\dots,\hat{\boldsymbol{w}}_{k-1}) = \Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) + \frac{1}{2}\sum_{i=k-N}^{k}\sum_{j=1}^{q}\|\boldsymbol{C}^{[j]}\hat{\boldsymbol{x}}_i - \boldsymbol{y}_i^{[j]}\|^2_{\boldsymbol{R}^{[j]-1}} + \frac{1}{2}\sum_{i=k-N}^{k-1}\|\hat{\boldsymbol{w}}_i\|^2_{\boldsymbol{Q}^{-1}}. \tag{9.7c}$$

The function $\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N})$ in (9.7c) is the *arrival cost* and summarizes the information not explicitly accounted for in the current moving horizon. Based on the solution of (9.7), the actual state estimate is calculated by

$$\hat{\boldsymbol{x}}_{k-N+i} = \boldsymbol{A}^i\hat{\boldsymbol{x}}_{k-N} + \sum_{j=1}^{i}\boldsymbol{A}^{i-j}\boldsymbol{B}\boldsymbol{u}_{k-N-1+j} + \sum_{j=1}^{\min(i,N)}\boldsymbol{A}^{i-j}\hat{\boldsymbol{w}}_{k-N-1+j}. \tag{9.8}$$

This expression computes a smoothed state estimate when $i < N$, a filtered state estimate when $i = N$, and a predicted state estimate when $i > N$.

Recall that according to Theorem 2.1.1, we can express the arrival cost in this case algebraically and establish a relation between the CMHE and the Kalman filter (KF). More precisely, the state estimate $\hat{\boldsymbol{x}}_k$ and $\hat{\boldsymbol{x}}_{k+1}$ are equivalent to the one derived by a Kalman filter (Rao, 2000) and an one-step prediction Kalman filter (Muske et al., 1993), respectively, if the arrival cost is chosen as

$$\Gamma_{k-N}(\hat{\boldsymbol{x}}_{k-N}) = \frac{1}{2}\|\hat{\boldsymbol{x}}_{k-N} - \bar{\boldsymbol{x}}_{k-N}\|^2_{\boldsymbol{P}^{-1}_{k-N}}, \tag{9.9}$$

where $\bar{\boldsymbol{x}}_{k-N}$ denotes the optimal CMHE estimate at time $k-N$ given the measurements up to time $k-N-1$ and where the covariance matrix $\boldsymbol{P}_{k-N}$ is updated by the Riccati equation

$$\boldsymbol{P}_{k-N} = \boldsymbol{A}\boldsymbol{P}_{k-N-1}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{P}_{k-N-1}\boldsymbol{C}^T\big(\boldsymbol{C}\boldsymbol{P}_{k-N-1}\boldsymbol{C}^T + \boldsymbol{R}\big)^{-1}\boldsymbol{C}\boldsymbol{P}_{k-N-1}\boldsymbol{A}^T + \boldsymbol{Q}, \tag{9.10}$$

where $\boldsymbol{R} \triangleq \operatorname{diag}(\boldsymbol{R}^{[1]},\dots,\boldsymbol{R}^{[q]})$. Positive definiteness (and thus invertibility) of $\boldsymbol{P}_{k-N}$ is assured by the following Lemma:

Lemma 9.2.1. *If $(\boldsymbol{C},\boldsymbol{A})$ is detectable and $(\boldsymbol{A},\boldsymbol{Q}^{1/2})$ is controllable, then $\lim_{k\to\infty}\boldsymbol{P}_{k-N} = \boldsymbol{P}_\infty$ where $\boldsymbol{P}_\infty > \boldsymbol{0}$ is the unique steady sate solution to the Riccati equation (9.10). Furthermore, if $\boldsymbol{P}_0 \geq \boldsymbol{P}_\infty$ (meaning that $\boldsymbol{P}_0 - \boldsymbol{P}_\infty$ is nonnegative definite), then $\boldsymbol{P}_{k-N}$ is positive definite for all $k \in \mathbb{N}_N$.*

Proof. The proof of the former and latter part can be found in de Souza et al. (1986) and Bitmead et al. (1985), respectively. □

Consequently, if we choose $\boldsymbol{P}_{k-N}$ as the steady state covariance matrix $\boldsymbol{P}_\infty$ in (9.9), the filtered or predicted state estimate derived by (9.8) coincides with the respective steady state Kalman filter.

The CMHO: A Special Case of the CMHE

It is important to mention that we obtain the optimization problem of the CMHO stated in (9.5) from the one of the CMHE presented in (9.7) by removing the estimated state disturbances $\hat{\boldsymbol{w}}_i$ and the arrival cost Γ_{k-N}. For this reason, we do not derive the distributed versions of the CMHO and CMHE separately. Instead, we detail only the derivation of the DMHE and derive the DMHO afterwards as a special case of the DMHE.

Reformulation by Vector and Matrix Representation

The prediction step (9.8) of the CMHE is already available in a form suitable for distributed implementation. Therefore, we focus in the following only on the update step given by the optimization problem (9.7). By denoting

$$\begin{aligned} \boldsymbol{R} &\triangleq \operatorname{diag}(\boldsymbol{R}^{[1]}, \dots, \boldsymbol{R}^{[q]}), & \underline{\boldsymbol{R}} &\triangleq \operatorname{diag}(\boldsymbol{R}, \dots, \boldsymbol{R}), \\ & & \underline{\boldsymbol{Q}} &\triangleq \operatorname{diag}(\boldsymbol{Q}, \dots, \boldsymbol{Q}), \\ \underline{\boldsymbol{y}}_k &\triangleq \operatorname{col}(\boldsymbol{y}_i, i \in \mathcal{I}_k), & \underline{\boldsymbol{u}}_k &\triangleq \operatorname{col}(\boldsymbol{u}_i, i \in \overline{\mathcal{I}}_k), \\ \underline{\boldsymbol{v}}_k &\triangleq \operatorname{col}(\boldsymbol{v}_i, i \in \mathcal{I}_k), & \underline{\boldsymbol{w}}_k &\triangleq \operatorname{col}(\boldsymbol{w}_i, i \in \overline{\mathcal{I}}_k), \end{aligned}$$

$$\boldsymbol{F} \triangleq \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{N} \end{bmatrix}, \boldsymbol{G} \triangleq \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{B} & \boldsymbol{0} & \ddots & \vdots \\ \boldsymbol{C}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{B} & \ddots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{C}\boldsymbol{A}^{N-2}\boldsymbol{B} & \dots & \boldsymbol{C}\boldsymbol{B} \end{bmatrix}, \boldsymbol{H} \triangleq \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{0} & \ddots & \vdots \\ \boldsymbol{C}\boldsymbol{A} & \boldsymbol{C} & \ddots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}^{N-1} & \boldsymbol{C}\boldsymbol{A}^{N-2} & \dots & \boldsymbol{C} \end{bmatrix},$$

we can write the dynamics of $\underline{\boldsymbol{y}}_k$ as

$$\underline{\boldsymbol{y}}_k = \boldsymbol{F}\boldsymbol{x}_{k-N} + \boldsymbol{G}\underline{\boldsymbol{u}}_k + \boldsymbol{H}\underline{\boldsymbol{w}}_k + \underline{\boldsymbol{v}}_k. \tag{9.11}$$

Consequently, we can express the constrained problem (9.7) with the arrival cost (9.9) in terms of vectors and matrices as the unconstrained problem

$$\min_{\hat{\boldsymbol{x}}_{k-N}, \underline{\hat{\boldsymbol{w}}}_k} J_k(\hat{\boldsymbol{x}}_{k-N}, \underline{\hat{\boldsymbol{w}}}_k), \tag{9.12a}$$

with the cost function

$$\begin{aligned} J_k(\hat{\boldsymbol{x}}_{k-N}, \underline{\hat{\boldsymbol{w}}}_k) = \frac{1}{2}\|\underline{\boldsymbol{y}}_k - \boldsymbol{F}\hat{\boldsymbol{x}}_{k-N} - \boldsymbol{G}\underline{\boldsymbol{u}}_k - \boldsymbol{H}\underline{\hat{\boldsymbol{w}}}_k\|^2_{\underline{\boldsymbol{R}}^{-1}} \\ + \frac{1}{2}\|\underline{\hat{\boldsymbol{w}}}_k\|^2_{\underline{\boldsymbol{Q}}^{-1}} + \frac{1}{2}\|\hat{\boldsymbol{x}}_{k-N} - \bar{\boldsymbol{x}}_{k-N}\|^2_{\boldsymbol{P}^{-1}_{k-N}}, \end{aligned} \tag{9.12b}$$

where the covariance matrix $\boldsymbol{P}_{k-N}$ is either updated according to the Riccati equation (9.10) or chosen as the steady state solution $\boldsymbol{P}_\infty$.

Measurement Allocation

The first step in deriving a distributed algorithm is to allocate the measurements $\underline{\boldsymbol{y}}_k$ of the k-th moving horizon to the m estimators. To this end, we recall the sets $\mathcal{I}_k = \{k-N, \dots, k\}$ and $\mathcal{S} = \{1, 2, \dots, q\}$ whose elements represent the time indices of the k-th moving horizon and the sensors, respectively. The resulting index set $\mathcal{Y}_k = \mathcal{I}_k \times \mathcal{S}$ associated to $\underline{\boldsymbol{y}}_k$ is separated in m index sets $\mathcal{Y}_k^{(i)}$ such that $\bigcup_{i=1}^m \mathcal{Y}_k^{(i)} = \mathcal{Y}_k$, $\mathcal{Y}_k^{(i)} \cap \mathcal{Y}_k^{(j)} = \{\emptyset\}$, $\forall i, j \in \mathcal{A}$, $i \neq j$, and $\mathcal{Y}_k^{(i)} \neq \{\emptyset\}$, $\forall i \in \mathcal{A}$. This ensures that each measurement $\boldsymbol{y}_i^{[j]}$ with $(i,j) \in \mathcal{Y}_k$ is assigned precisely to one estimator. The measurements available to the i-th estimator at the time k are denoted by $\underline{\boldsymbol{y}}_k^{(i)} \triangleq \mathrm{col}(\boldsymbol{y}_i^{[j]}, (i,j) \in \mathcal{Y}_k^{(i)})$. Consequently, we can rearrange (9.11) as follows

$$\begin{bmatrix} \underline{\boldsymbol{y}}_k^{(1)} \\ \vdots \\ \underline{\boldsymbol{y}}_k^{(m)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_k^{(1)} \\ \vdots \\ \boldsymbol{F}_k^{(m)} \end{bmatrix} \boldsymbol{x}_{k-N} + \begin{bmatrix} \boldsymbol{G}_k^{(1)} \\ \vdots \\ \boldsymbol{G}_k^{(m)} \end{bmatrix} \underline{\boldsymbol{u}}_k + \begin{bmatrix} \boldsymbol{H}_k^{(1)} \\ \vdots \\ \boldsymbol{H}_k^{(m)} \end{bmatrix} \underline{\boldsymbol{w}}_k + \begin{bmatrix} \boldsymbol{v}_k^{(1)} \\ \vdots \\ \boldsymbol{v}_k^{(m)} \end{bmatrix} \tag{9.13}$$

and thus rewrite (9.12b) as

$$\begin{aligned} J_k(\hat{\boldsymbol{x}}_{k-N}, \underline{\hat{\boldsymbol{w}}}_{k-N}) = \frac{1}{2} \sum_{i=1}^{m} \| \underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{F}_k^{(i)} \hat{\boldsymbol{x}}_{k-N} - \boldsymbol{G}_k^{(i)} \underline{\boldsymbol{u}}_k - \boldsymbol{H}_k^{(i)} \underline{\hat{\boldsymbol{w}}}_k \|^2_{\underline{\boldsymbol{R}}_k^{(i)^{-1}}} \\ + \frac{1}{2} \| \underline{\hat{\boldsymbol{w}}}_k \|^2_{\underline{\boldsymbol{Q}}^{-1}} + \frac{1}{2} \| \hat{\boldsymbol{x}}_{k-N} - \bar{\boldsymbol{x}}_{k-N} \|^2_{\boldsymbol{P}_{k-N}^{-1}}. \end{aligned} \tag{9.14}$$

Decoupling of the Cost Function

Although the measurements have been assigned to the m estimators, the optimization problem stemming from (9.14) cannot be solved in a distributed manner because of the coupling variables $\hat{\boldsymbol{x}}_{k-N}$ and $\underline{\hat{\boldsymbol{w}}}_k$. Thus, we introduce local estimates $\hat{\boldsymbol{x}}_{k-N}^{(i)}$ and $\underline{\hat{\boldsymbol{w}}}_k^{(i)}$ for each estimator and divide the cost function (9.14) into m decoupled parts

$$J_k(\hat{\boldsymbol{x}}_{k-N}^{(1)}, \underline{\hat{\boldsymbol{w}}}_k^{(1)}, \dots, \hat{\boldsymbol{x}}_{k-N}^{(m)}, \underline{\hat{\boldsymbol{w}}}_k^{(m)}) = \sum_{i=1}^{m} J_k^{(i)}(\hat{\boldsymbol{x}}_{k-N}^{(i)}, \underline{\hat{\boldsymbol{w}}}_k^{(i)}), \tag{9.15a}$$

where the addend $J_k^{(i)}$ is

$$\begin{aligned} J_k^{(i)}(\hat{\boldsymbol{x}}_{k-N}^{(i)}, \underline{\hat{\boldsymbol{w}}}_k^{(i)}) = \frac{1}{2} \| \underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{F}_k^{(i)} \hat{\boldsymbol{x}}_{k-N}^{(i)} - \boldsymbol{G}_k^{(i)} \underline{\boldsymbol{u}}_k - \boldsymbol{H}_k^{(i)} \underline{\hat{\boldsymbol{w}}}_k^{(i)} \|^2_{\underline{\boldsymbol{R}}_k^{(i)^{-1}}} \\ + \frac{1}{2m} \| \underline{\hat{\boldsymbol{w}}}_k^{(i)} \|^2_{\underline{\boldsymbol{Q}}^{-1}} + \frac{1}{2m} \| \hat{\boldsymbol{x}}_{k-N}^{(i)} - \bar{\boldsymbol{x}}_{k-N} \|^2_{\boldsymbol{P}_{k-N}^{-1}}. \end{aligned} \tag{9.15b}$$

The reason for this choice of $J_k^{(i)}$ will become clear later. The solution to the problem stemming from (9.15a) is identical to the solution of (9.7) if and only if each local solution $\hat{\boldsymbol{x}}_{k-N}^{(i)}, \underline{\hat{\boldsymbol{w}}}_k^{(i)}$ coincides with $\hat{\boldsymbol{x}}_{k-N}, \underline{\hat{\boldsymbol{w}}}_k$. In order to reach this consensus, the estimators have to exchange information. The information flow of the primal variables $\hat{\boldsymbol{p}}_k^{(i)} \triangleq \left[\hat{\boldsymbol{x}}_{k-N}^{(i)^T}, \underline{\hat{\boldsymbol{w}}}_k^{(i)^T} \right]^T \in \mathbb{R}^{(N+1)n_x}$ among the estimators is described by the directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the nodes in $\mathcal{V} = \{1, 2, \dots, m\}$ represent the estimators and the edge (i,j) in the set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ models

that estimator i can transmit information about $\hat{\boldsymbol{p}}_k^{(i)}$ to estimator j. By means of the adjacency matrix $\boldsymbol{\Omega}(\mathcal{G}) = [\omega_{ij}] \in \mathbb{R}^{m\times m}$, with entries

$$\omega_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{E} \\ 0 & (i,j) \notin \mathcal{E}, \end{cases} \tag{9.16}$$

we can express (9.7) as

$$p_k^* = \min_{\hat{\boldsymbol{p}}_k^{(1)}} J_k^{(1)}(\hat{\boldsymbol{p}}_k^{(1)}) + \ldots + \min_{\hat{\boldsymbol{p}}_k^{(m)}} J_k^{(m)}(\hat{\boldsymbol{p}}_k^{(m)}) \tag{9.17a}$$

$$\text{subject to} \qquad \hat{\boldsymbol{p}}_k^{(i)} = \hat{\boldsymbol{p}}_k^{(j)}, \quad \forall (i,j) \in \mathcal{E}, \tag{9.17b}$$

if the set $\mathcal{E}$ is such that the consensus constraints (9.17b) ensure equality of all local solutions $\hat{\boldsymbol{p}}_k^{(i)}$, i. e. $\hat{\boldsymbol{p}}_k^{(i)} = \hat{\boldsymbol{p}}_k^{(j)}, \forall i,j \in \mathcal{V}$, and thus identity to the central solution $\hat{\boldsymbol{p}}_k$. This is guaranteed by the following assumption.

Assumption 20. The directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is weakly connected and has no self-loops.

Dual Problem

Although the cost function in (9.17a) is decoupled, the primal problem (9.17) still cannot be solved in a distributed manner because of the coupling through the consensus constraints (9.17b). Decoupling can be achieved by dual decomposition (cf. Section 2.4.4.2) which utilizes the dual problem of (9.17). To this end, we introduce the dual variables $\boldsymbol{\lambda}_k^{(i,j)} \in \mathbb{R}^{(N+1)n_x}$ which are transmitted from estimator i to estimator j. The information flow of $\boldsymbol{\lambda}_k^{(i,j)}$ among the estimators is described by the reverse graph $\bar{\mathcal{G}} = \{\mathcal{V}, \bar{\mathcal{E}}\}$, which is obtained by reversing the order of the nodes of all the pairs in $\mathcal{E}$. Recall that the adjacency matrix $\bar{\boldsymbol{\Omega}} = [\bar{\omega}_{ij}] \in \mathbb{R}^{m\times m}$ associated to the graph $\bar{\mathcal{G}}$ is $\bar{\boldsymbol{\Omega}} = \boldsymbol{\Omega}^T$.

By denoting

$$\underline{\hat{\boldsymbol{p}}}_k \triangleq \text{col}(\hat{\boldsymbol{p}}_k^{(i)}, i \in \mathcal{V}), \qquad \underline{\boldsymbol{\lambda}}_k \triangleq \text{col}(\boldsymbol{\lambda}_k^{(i,j)}, (i,j) \in \bar{\mathcal{E}}), \tag{9.18}$$

we can decouple the Lagrangian of (9.17) regarding the primal variables as follows

$$L_k(\underline{\hat{\boldsymbol{p}}}_k, \underline{\boldsymbol{\lambda}}_k) = \sum_{i=1}^{m} L_k^{(i)}(\hat{\boldsymbol{p}}_k^{(i)}, \underline{\boldsymbol{\lambda}}_k), \tag{9.19}$$

where $L_k^{(i)}(\hat{\boldsymbol{p}}_i, \underline{\boldsymbol{\lambda}}_k)$ is

$$L_k^{(i)}(\hat{\boldsymbol{p}}_i, \underline{\boldsymbol{\lambda}}_k) = J_k^{(i)}(\hat{\boldsymbol{p}}_k^{(i)}) - \sum_{j=1}^{m} \left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)} - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}\right)^T \hat{\boldsymbol{p}}_k^{(i)}. \tag{9.20}$$

Consequently, we can also decouple the dual function $q_k(\underline{\boldsymbol{\lambda}}_k)$

$$\begin{aligned} q_k(\underline{\boldsymbol{\lambda}}_k) &= \min_{\underline{\hat{\boldsymbol{p}}}_k} L_k(\underline{\hat{\boldsymbol{p}}}_k, \underline{\boldsymbol{\lambda}}_k) \\ &= \min_{\underline{\hat{\boldsymbol{p}}}_k} \sum_{i=1}^{m} L_k^{(i)}(\hat{\boldsymbol{p}}_k^{(i)}, \underline{\boldsymbol{\lambda}}_k) \\ &= \sum_{i=1}^{m} q_k^{(i)}(\underline{\boldsymbol{\lambda}}_k), \end{aligned} \tag{9.21}$$

where

$$q_k^{(i)}(\underline{\boldsymbol{\lambda}}_k) = \min_{\hat{\boldsymbol{p}}_k^{(i)}} L_k^{(i)}(\hat{\boldsymbol{p}}_k^{(i)}, \underline{\boldsymbol{\lambda}}_k). \tag{9.22}$$

Recall that the dual function (9.21) is concave and yields according to Lemma 2.4.10 for any $\underline{\boldsymbol{\lambda}}_k$ a lower bound on the optimal value p_k^* of the primal problem (9.17). The question about the best lower bound d_k^* that can be obtained leads to the dual problem

$$\begin{aligned} d_k^* &= \max_{\underline{\boldsymbol{\lambda}}_k} q_k(\underline{\boldsymbol{\lambda}}_k) \\ &= \max_{\underline{\boldsymbol{\lambda}}_k} \sum_{i=1}^{m} q_k^{(i)}(\underline{\boldsymbol{\lambda}}_k). \end{aligned} \tag{9.23}$$

Due to the Assumptions 17 and 20, the primal problem (9.17) is strictly convex and strictly feasible. This implies according to Theorem 2.4.11 that strong duality holds, i. e. $d_k^* = p_k^*$. In other words, we can use the dual problem (9.23) to solve the primal problem (9.17) in a distributed manner.

Subgradient Method

The method of choice to solve the dual problem is the subgradient method presented in Section 2.4.2.5. To this end, we transform this maximization problem into a minimization problem in the following way

$$d_k^* = \max_{\underline{\boldsymbol{\lambda}}_k} q_k(\underline{\boldsymbol{\lambda}}_k) = \min_{\underline{\boldsymbol{\lambda}}_k} -q_k(\underline{\boldsymbol{\lambda}}_k). \tag{9.24}$$

Since $-q_k(\underline{\boldsymbol{\lambda}}_k)$ is a convex function, we can apply the subgradient method which consists for the given problem in the following iterative procedure

$$\boldsymbol{\lambda}_k^{(i,j)}[l+1] = \boldsymbol{\lambda}_k^{(i,j)}[l] - \gamma_k[l]\, \boldsymbol{s}_k^{(i,j)}[l], \quad \forall (i,j) \in \bar{\mathcal{E}}, \quad l \in \mathbb{N}_0, \tag{9.25}$$

where l is the iteration counter, $\boldsymbol{s}_k^{(i,j)}[l]$ are any subgradients of the negative dual function $-q_k$ at the current iterate $\underline{\boldsymbol{\lambda}}_k^{(i,j)}[l]$ and $\gamma_k[l]$ is the step size of the l-th iteration. The required subgradients are given as follows.

Lemma 9.2.2. *Let $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ denote $\forall i \in \mathcal{V}$ the optimal solutions to the dual functions $q_k^{(i)}(\underline{\boldsymbol{\lambda}}_k[l])$ given in (9.22). Then*

$$\boldsymbol{s}_k^{(i,j)}[l] = \hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \tag{9.26}$$

is $\forall (i,j) \in \bar{\mathcal{E}}$ a subgradient of $-q_k(\boldsymbol{\lambda}_k^{(i,j)}[l])$ at the point $\boldsymbol{\lambda}_k^{(i,j)}[l]$.

Proof. By choosing an arbitrary $\boldsymbol{\lambda}_k^{(i,j)}[l]$, $(i,j) \in \bar{\mathcal{E}}$ as the dependent variable and all other $\boldsymbol{\lambda}_k^{(a,b)}[l]$, $(a,b) \in \bar{\mathcal{E}} \setminus \{(i,j)\}$ as the independent variables, (9.21) becomes

$$q_k(\boldsymbol{\lambda}_k^{(i,j)}[l]) = \min_{\underline{\hat{\boldsymbol{p}}}_k[l]} \Big(\bar{L}_k - \boldsymbol{\lambda}_k^{(i,j)T}[l] \left(\hat{\boldsymbol{p}}_k^{(i)}[l] - \hat{\boldsymbol{p}}_k^{(j)}[l] \right) \Big),$$

where

$$\bar{L}_k = \sum_{a=1}^{m} J_k^{(a)}\left(\hat{\boldsymbol{p}}_k^{(a)}[l]\right) - \sum_{\substack{a=1 \\ a \neq i}}^{m} \sum_{\substack{b=1 \\ b \neq j}}^{m} \bar{\omega}_{ab} \boldsymbol{\lambda}_k^{(a,b)T}[l] \left(\hat{\boldsymbol{p}}_k^{(a)}[l] - \hat{\boldsymbol{p}}_k^{(b)}[l] \right).$$

Let $\bar{L}_k^*$ denote the optimal value of $\bar{L}_k$ corresponding to the optimal solution $\underline{\hat{\boldsymbol{p}}}_k^*[l]$ of $q_k(\boldsymbol{\lambda}_k^{(i,j)}[l])$. Then the following relationship holds for all $\boldsymbol{\lambda}_k^{(i,j)}[l]$, $(i,j) \in \bar{\mathcal{E}}$

$$\begin{aligned}
q_k(\boldsymbol{\xi}) &= \min_{\hat{\boldsymbol{p}}_k[l]} \Big(\bar{L}_k - \boldsymbol{\xi}^T \left(\hat{\boldsymbol{p}}_k^{(i)}[l] - \hat{\boldsymbol{p}}_k^{(j)}[l] \right) \Big) \\
&\leq \bar{L}_k^* - \boldsymbol{\xi}^T \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right) \\
&= \bar{L}_k^* - \boldsymbol{\lambda}_k^{(i,j)T}[l] \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right) + \boldsymbol{\lambda}_k^{(i,j)T}[l] \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right) - \boldsymbol{\xi}^T \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right) \\
&= q_k(\boldsymbol{\lambda}_k^{(i,j)}[l]) - \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right)^T \left(\boldsymbol{\xi} - \boldsymbol{\lambda}_k^{(i,j)}[l] \right)
\end{aligned}$$

and thus

$$-q_k(\boldsymbol{\xi}) \geq -q_k(\boldsymbol{\lambda}_k^{(i,j)}[l]) + \left(\hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l] \right)^T \left(\boldsymbol{\xi} - \boldsymbol{\lambda}_k^{(i,j)}[l] \right).$$

Consequently, according to Definition 2.4.5, $\boldsymbol{s}_k^{(i,j)}[l] = \hat{\boldsymbol{p}}_k^{(i)*}[l] - \hat{\boldsymbol{p}}_k^{(j)*}[l]$ is $\forall (i,j) \in \bar{\mathcal{E}}$ a subgradient of the convex function $-q_k(\boldsymbol{\lambda}_k^{(i,j)}[l])$ at the point $\boldsymbol{\lambda}_k^{(i,j)}[l]$. □

The calculation of the subgradients require the optimal solutions $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ which are given as follows.

Lemma 9.2.3. *The optimal solutions $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ to the dual functions $q_k^{(i)}(\boldsymbol{\lambda}_k[l])$ given in (9.22) are $\forall i \in \mathcal{V}$*

$$\hat{\boldsymbol{p}}_k^{(i)*}[l] = \boldsymbol{N}_k^{(i)-1} \Big(\boldsymbol{o}_k^{(i)} + \sum_{j=1}^{m} \left(\bar{\omega}_{ij} \boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji} \boldsymbol{\lambda}_k^{(j,i)}[l] \right) \Big), \tag{9.27a}$$

where

$$\boldsymbol{N}_k^{(i)} \triangleq \begin{bmatrix} \boldsymbol{F}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \boldsymbol{F}_k^{(i)} + \frac{1}{m} \boldsymbol{P}_{k-N}^{-1} & \boldsymbol{F}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \boldsymbol{H}_k^{(i)} \\ \boldsymbol{H}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \boldsymbol{F}_k^{(i)} & \boldsymbol{H}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \boldsymbol{H}_k^{(i)} + \frac{1}{m} \underline{\boldsymbol{Q}}^{-1}, \end{bmatrix} \tag{9.27b}$$

$$\boldsymbol{o}_k^{(i)} \triangleq \begin{bmatrix} \boldsymbol{F}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \\ \boldsymbol{H}_k^{(i)^T} \underline{\boldsymbol{R}}_k^{(i)^{-1}} \end{bmatrix} \left(\underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{G}_k^{(i)} \underline{\boldsymbol{u}}_k \right) + \begin{bmatrix} \frac{1}{m} \boldsymbol{P}_{k-N}^{-1} \\ \boldsymbol{0} \end{bmatrix} \bar{\boldsymbol{x}}_{k-N}. \tag{9.27c}$$

Proof. The proof is sketched as follows: The dual function (9.22) is $\forall i \in \mathcal{V}$ an unconstrained strictly concave quadratic problem. Thus, the optimal solution can be obtained analytically by solving $\partial L_k^{(i)} / \partial \hat{\boldsymbol{p}}_k^{(i)} = \boldsymbol{0}$. The matrix $\boldsymbol{N}_k^{(i)}$ is always positive definite because it can be decomposed into the sum of the positive definite matrix $\frac{1}{m} \operatorname{diag}(\boldsymbol{P}_{k-N}, \underline{\boldsymbol{Q}})^{-1}$ and the positive semi-definite matrix $\boldsymbol{M}_k^{(i)^T} \boldsymbol{M}_k^{(i)}$, where

$$\boldsymbol{M}_k^{(i)} = \frac{1}{\sqrt{2}} \begin{bmatrix} \underline{\boldsymbol{R}}_k^{(i)^{-1/2}} & \boldsymbol{0} \\ \boldsymbol{0} & \underline{\boldsymbol{R}}_k^{(i)^{-1/2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_k^{(i)} & \boldsymbol{H}_k^{(i)} \\ \boldsymbol{F}_k^{(i)} & \boldsymbol{H}_k^{(i)} \end{bmatrix}.$$

□

Note that the distribution of the arrival cost among the local cost functions $J_k^{(i)}$ in (9.15) guarantees positive definiteness of $\boldsymbol{N}_k^{(i)}$ regardless of the matrices $\boldsymbol{F}_k^{(i)}$ and $\boldsymbol{H}_k^{(i)}$. As a consequence, the solvability of the dual functions (9.22) is independent of the measurement allocation. This means that there is no need for the measurement allocation to imply local observability of the states. This would have been necessary if the local cost functions $J_k^{(i)}$ are independent of the arrival cost. In this case, it is easy to see that $\boldsymbol{N}_k^{(i)}$ is positive definite if and only if $\boldsymbol{F}_k^{(i)}$ has full rank for which local observability is required. In any case, if solvability of the local problems is given, the consensus of the local variables $\hat{\boldsymbol{p}}_k^{(i)}$ to the optimal solution $\hat{\boldsymbol{p}}_k^*$ is guaranteed by the constraints (9.17b).

The DMHO: A Special Case of the DMHE

As stated earlier, we derive the DMHO as a special case of the DMHE by removing the estimated state disturbances $\hat{\boldsymbol{w}}_i$, the arrival cost Γ_{k-N} and the associated terms. Consequently, the primal variables $\hat{\boldsymbol{p}}_k^{(i)}$ of the DMHO reduce to the local initial values $\hat{\boldsymbol{x}}_{k-N}^{(i)}$, i. e. $\hat{\boldsymbol{p}}_k^{(i)} \triangleq \hat{\boldsymbol{x}}_{k-N}^{(i)} \in \mathbb{R}^{n_x}$. The remaining procedure of the DMHO is identical to the one of the DMHE in adapted form. However, there is one step where special care has to be taken. As mentioned above, if there is no arrival cost, then the optimal solutions $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ required for the calculation of the subgradients can only be determined if $\boldsymbol{F}_k^{(i)}$ has full rank. This is stated in the following Lemma which is the adapted version of Lemma 9.2.3.

Lemma 9.2.4. *Suppose that* $\forall i \in \mathcal{V}$, $\operatorname{rank}(\boldsymbol{F}_k^{(i)}) = n_x$. *Then the optimal solutions* $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ *to the dual functions* $q_k^{(i)}(\boldsymbol{\lambda}_k[l])$ *given in* (9.22) *are* $\forall i \in \mathcal{V}$

$$\hat{\boldsymbol{p}}_k^{(i)*}[l] = \boldsymbol{N}_k^{(i)^{-1}}\Big(\boldsymbol{o}_k^{(i)} + \sum_{j=1}^{m}\left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}[l]\right)\Big), \tag{9.28a}$$

where

$$\boldsymbol{N}_k^{(i)} \triangleq \boldsymbol{F}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\boldsymbol{F}_k^{(i)}, \tag{9.28b}$$

$$\boldsymbol{o}_k^{(i)} \triangleq \boldsymbol{F}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\left(\underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{G}_k^{(i)}\underline{\boldsymbol{u}}_k\right). \tag{9.28c}$$

The only way to achieve $\forall i \in \mathcal{V}$ full rankness of $\boldsymbol{F}_k^{(i)}$ is to develop a suitable measurement allocation algorithm. Its objective can be stated as follows.

Problem 9.3. *The problem is to design a measurement allocation algorithm such that the following conditions are satisfied*

$$\begin{aligned}
&(C0) \quad \textstyle\bigcup_{i=1}^{m}\mathcal{Y}_k^{(i)} = \mathcal{Y}_k,\\
&(C1) \quad \mathcal{Y}_k^{(i)} \cap \mathcal{Y}_k^{(j)} = \{\emptyset\}, \quad \forall i,j \in \mathcal{V}, i \neq j,\\
&(C2) \quad \mathcal{Y}_k^{(i)} \neq \{\emptyset\}, \quad \forall i \in \mathcal{V},\\
&(C3) \quad \operatorname{rank}(\boldsymbol{F}_k^{(i)}) = n_x, \quad \forall i \in \mathcal{V}.
\end{aligned}$$

Our strategy to tackle this problem is as follows. First, we note that a necessary condition for $\operatorname{rank}(\boldsymbol{F}_k^{(i)}) = n_x$ is that the number of rows of $\boldsymbol{F}_k^{(i)}$, which is equivalent to the dimension of $\underline{\boldsymbol{y}}_k^{(i)}$, is at least n_x. This means that we have to chose the size of the moving horizon sufficiently large. Once this is ensured, the basic idea of the proposed measurement allocation algorithm is to assign the measurements periodically to the observers, i. e. the $(i - 1 + j\,m)$-th measurement is assigned to the i-th observer with $i \in \mathcal{V}$, $j \in \mathbb{N}_0$ and $m = |\mathcal{V}|$. We illustrate this idea by an exemplary situation depicted in Table 9.1. In order to derive a general description for this approach, we introduce the greatest integer operator (Graham et al., 1994).

Definition 9.2.1. *Let* $x \in \mathbb{R}$ *and* $z \in \mathbb{Z}$ *be a real number and an integer, respectively. The greatest integer operator* $\lfloor\cdot\rfloor$ *is defined as*

$$\lfloor x \rfloor = \max_z z, \ \textit{subject to } z \leq x,$$

and maps x *to the greatest integer less than or equal to* x.

Then we can state the periodic measurement allocation algorithm in the following Theorem.

Table 9.1: Elements in the sets $\mathcal{Y}_k^{(i)}$ for $i = 1, 2, 3$, $k = 1, 2, \ldots, 11$, $m = 3$ and $N+1 = 6$. The current index of each step is framed to highlight the periodicity of the measurement allocation.

	discrete time k											
	0	1	2	3	4	5	6	7	8	9	10	11
$\mathcal{Y}_k^{(1)}$	$\boxed{1}$	1	1	1	1	1	4	4	4	7	7	7
	–	–	–	$\boxed{4}$	4	4	$\boxed{7}$	7	7	$\boxed{10}$	10	10
$\mathcal{Y}_k^{(2)}$	–	$\boxed{2}$	2	2	2	2	2	5	5	5	8	8
	–	–	–	–	$\boxed{5}$	5	5	$\boxed{8}$	8	8	$\boxed{11}$	11
$\mathcal{Y}_k^{(3)}$	–	–	$\boxed{3}$	3	3	3	3	3	6	6	6	9
	–	–	–	–	–	$\boxed{6}$	6	6	$\boxed{9}$	9	9	$\boxed{12}$

Theorem 9.2.5. *Suppose that the pair $(\boldsymbol{C}\boldsymbol{A}^j, \boldsymbol{A}^m)$ is observable for all $j \in \{0, 1, \ldots, m-1\}$. If the length of the moving horizon is*

$$N + 1 = (\lfloor \frac{n_x}{n_y} \rfloor + 1 + \beta)m \tag{9.29}$$

where $\beta \in \mathbb{N}_0$ is a free parameter determining the overall moving horizon size and if the index sets $\mathcal{Y}_k^{(i)}$ are chosen $\forall i \in \mathcal{V}$ as

$$\mathcal{Y}_k^{(i)} = \begin{cases} \bigcup_{j=1}^{|\mathcal{Y}_k^{(i)}|} \left\{ {}_1\mathcal{Y}_k^{(i)} + (j-1)m \right\}, & |\mathcal{Y}_k^{(i)}| > 0 \\ \{\emptyset\}, & |\mathcal{Y}_k^{(i)}| = 0 \end{cases}, \quad i \in \mathcal{V} \tag{9.30a}$$

with

$$|\mathcal{Y}_k^{(i)}| = \min \left\{ \lfloor \frac{k+1-i}{m} \rfloor + 1, \frac{N+1}{m} \right\}, \tag{9.30b}$$

$${}_1\mathcal{Y}_k^{(i)} = \max \left\{ \lfloor \frac{k+1-i}{m} \rfloor + 1 - \frac{N+1}{m}, 0 \right\} m + i, \tag{9.30c}$$

where $|\mathcal{Y}_k^{(i)}|$ is the cardinality of $\mathcal{Y}_k^{(i)}$ and ${}_1\mathcal{Y}_k^{(i)}$ is the first element of $\mathcal{Y}_k^{(i)}$, then

$$\operatorname{rank}(\boldsymbol{F}_k^{(i)}) = n_x, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathbb{N}_N. \tag{9.31}$$

Proof. The consequence of the periodic measurement allocation is that

$$\text{number of rows of } \boldsymbol{F}_k^{(i)} = \dim(\underline{\boldsymbol{y}}_k^{(i)}) = \frac{N+1}{m} n_y, \quad \forall i \in \mathcal{V}.$$

From (9.29) we know that

$$\frac{N+1}{m} n_y = \left(\lfloor \frac{n_x}{n_y} \rfloor + 1 + \beta \right) n_y \geq n_x.$$

Combining both results ensures that $\forall i \in \mathcal{V}$ and $\forall k \in \mathbb{N}_N$ the number of rows of $\boldsymbol{F}_k^{(i)}$ is at least n_x which is a necessary condition for full rankness of $\boldsymbol{F}_k^{(i)}$. According to the index sets (9.30), we can write the matrix $\boldsymbol{F}_k^{(i)}$ as

$$\boldsymbol{F}_k^{(i)} = \begin{bmatrix} \boldsymbol{C}\boldsymbol{A}^{\kappa} \\ \boldsymbol{C}\boldsymbol{A}^{\kappa}\boldsymbol{A}^{m} \\ \boldsymbol{C}\boldsymbol{A}^{\kappa}\boldsymbol{A}^{m2} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{\kappa}\boldsymbol{A}^{m\left(\frac{N+1}{m}-1\right)} \end{bmatrix}, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathbb{N}_N, \tag{9.32}$$

where $\kappa = (m-1) - ((k+1-i) \bmod m) \in \{0, 1, \ldots, m-1\}$. Since the pair $(\boldsymbol{C}\boldsymbol{A}^j, \boldsymbol{A}^m)$ is observable for all $j \in \{0, 1, \ldots, m-1\}$ and $(N+1)/m - 1 \geq n_x - 1$, $\boldsymbol{F}_k^{(i)}$ has $\forall i \in \mathcal{V}$ and $\forall k \in \mathbb{N}_N$ full rank. □

Note that if rank $\boldsymbol{A} = n_x$, then the pair $(\boldsymbol{C}\boldsymbol{A}^j, \boldsymbol{A}^m)$ is observable for all $j \in \{0, 1, \ldots, m-1\}$ if and only if $(\boldsymbol{C}, \boldsymbol{A}^m)$ is observable.

9.3 Overall Algorithm

The algorithm that is followed by the i-th observer/estimator of both DHMS is presented in Algorithm 7. The differences between the DMHO and the DMHE in this algorithm are indicated by different text colors. It is important to note that the distinguishing difference between the DMHO and the DMHE is not visible in the presented algorithm since it concerns the different requirements for allocating the measurements. As discussed in the previous section, the measurement allocation has to be designed for the DMHO such that some form of local observability is ensured (cf. Theorem 9.2.5), while there are no restrictions for the DMHE.

When choosing the input to Algorithm 7, we have to consider the following aspects. As a consequence of the measurement allocation, the moving horizon size $N+1$ for the DMHO has to be chosen sufficiently large (cf. Theorem 9.2.5), while there are no restrictions for the DMHE. To ensure equality of all local estimates $\hat{\boldsymbol{x}}_k^{(i)}$, the graph $\mathcal{G}$, which describes the intercommunication of the observers/estimators, has to be weakly connected (cf. Assumption 20). The step sizes $\gamma_k[l]$ have to satisfy in general certain conditions (see Section 9.4) in order to guarantee stability of the DMHS.

Starting from zero initial values for the primal and dual variables (line 3), the optimization procedure alternates between updating and transmitting the primal and dual variables

Algorithm 7 Distributed moving horizon observer/estimator $\hat{\Sigma}^{(i)}$. (Black, green and red text color valid for DMHO & DMHE, DMHO only, DMHE only, respectively.)

Input: initial state $\hat{\boldsymbol{x}}_0$, communication graph $\mathcal{G}$, moving horizon size $N+1$, step sizes $\gamma_k[l]$, $\forall k \in \mathbb{N}_N$, $\forall l \in \mathbb{N}_0$, covariance matrix $\boldsymbol{R}$, covariance matrix $\boldsymbol{Q}$, covariance matrices $\boldsymbol{P}_k$, $\forall k \in \mathbb{N}_0$ (either calculated by Riccati equation (9.10) or set constant to steady state solution $\boldsymbol{P}_\infty$)

1: **for all** $k \in \mathbb{N}_0$ **do**
2: **if** $k \geq N$ **then**
3: Initialization: $l = 0$, $\quad \boldsymbol{\lambda}_k^{(i,j)}[0] = \boldsymbol{0}$, $\forall (i,j) \in \bar{\mathcal{E}}$, $\quad \hat{\boldsymbol{p}}_k^{(i)*}[0] = \boldsymbol{0}$, $\forall i \in \mathcal{V}$
4: **while** termination condition = false **do**
5: Increase iteration counter: $l \leftarrow l + 1$
6: Update dual variables $\boldsymbol{\lambda}_k^{(i,j)}[l]$:

$$\boldsymbol{\lambda}_k^{(i,j)}[l] = \boldsymbol{\lambda}_k^{(i,j)}[l-1] - \gamma_k[l-1]\left(\hat{\boldsymbol{p}}_k^{(i)*}[l-1] - \hat{\boldsymbol{p}}_k^{(j)*}[l-1]\right), \quad \forall (i,j) \in \bar{\mathcal{E}}$$

7: Communicate $\boldsymbol{\lambda}_k^{(i,j)}[l]$ according to the graph $\bar{\mathcal{G}}$.
8: Update primal variable $\hat{\boldsymbol{p}}_k^{(i)*}[l]$:

$$\hat{\boldsymbol{p}}_k^{(i)*}[l] = \boldsymbol{N}_k^{(i)^{-1}}\left(\boldsymbol{o}_k^{(i)} + \sum_{j=1}^{m}\left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}[l]\right)\right)$$

with $\boldsymbol{N}_k^{(i)}$ and $\boldsymbol{o}_k^{(i)}$ defined in (9.28b)/(9.27b) and (9.28c)/(9.27b), respectively.
9: Communicate $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ according to the graph $\mathcal{G}$.
10: **end while**
11: Predict state $\hat{\boldsymbol{x}}_k^{(i)}$ based on $\hat{\boldsymbol{p}}_k^{(i)} = \hat{\boldsymbol{p}}_k^{(i)*}[l]$ according to (9.6)/(9.8).
12: **else**
13: Predict state $\hat{\boldsymbol{x}}_k^{(i)}$ based on $\hat{\boldsymbol{x}}_0$ according to (9.6).
14: **end if**
15: **end for**

(line 5-9) until a terminal condition is satisfied (line 4). While the exact solution can in general only be derived for infinitely many iterations (see Section 9.4), a practical approximate solution can be found by prematurely stopping the optimization procedure. In this case, several suitable termination conditions exist, like the convergence tolerance of the dual variables and many more (Bertsekas et al., 2003; Shor, 1985; Kiwiel, 2004). It is important to note that there is no need for the i-th observer/estimator to exchange information in order to calculate the control inputs. All these values can be calculated fully decentralized based on the decentralized available controller Σ_C. Since each observer/estimator is initialized identically by $\hat{\boldsymbol{x}}_0$ and each observer/estimator converges to the identical optimal primal variable $\hat{\boldsymbol{p}}_k$, the decentralized calculated control inputs are identical for all observers/estimators. Moreover, if the measurement allocation scheme is known in advance, the inverse of the matrix $\boldsymbol{N}_k^{(i)}$ necessary for calculating $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ in line 8 can be calculated offline such that only simple matrix vector multiplications have to be performed online in each iteration step.

9.4 Stability Analysis

In this section, we analyze the stability of the DMHS presented in Algorithm 7. To this end, we recall that the CMHO described in (9.5) and (9.6) as well as the CMHE described in (9.7) and (9.8) are stable. Consequently, both DMHS are stable if all distributed estimates $\hat{\boldsymbol{x}}_k^{(i)}$ are identical to the corresponding centralized counterpart $\hat{\boldsymbol{x}}_k$. Since the prediction steps of the distributed and centralized moving horizon strategies are identical, it is sufficient to show that all local solutions $\hat{\boldsymbol{p}}_k^{(i)}$ are identical to the centralized solution $\hat{\boldsymbol{p}}_k$. As previously mentioned, this property depends on the connectivity of the observers/estimators described by the graph $\mathcal{G}$. Thus, we analyze the stability of the DMHS for the general case of a weakly connected directed graph in Section 9.4.1 and for the special case of a complete directed graph in Section 9.4.2. Note that the latter is tantamount to an all-to-all communication scheme.

9.4.1 General Case: Weakly Connected Directed Communication Graph

The convergence behavior of $\hat{\boldsymbol{p}}_k^{(i)}$ and thus the stability of the DMHS for a weakly connected directed graph $\mathcal{G}$ is analyzed in the following Theorem.

Theorem 9.4.1. *Suppose that Assumption 20 holds. If the step size sequences* $\{\gamma_k[l]\}_{l=0}^{\infty}$ *satisfy* $\forall k \in \mathbb{N}_N$

$$\gamma_k[l] > 0, \quad \lim_{l\to\infty} \gamma_k[l] = 0, \quad \sum_{l=0}^{\infty} \gamma_k[l] = \infty, \quad \gamma_k[l]\, \|\boldsymbol{s}_k^{(i,j)}[l]\| \leq c, \ \forall (i,j) \in \bar{\mathcal{E}}, \tag{9.33}$$

where $\boldsymbol{s}_k^{(i,j)}[l]$ *is the subgradient defined in Lemma 9.2.2 and* $c \in \mathbb{R}_0$ *is a positive constant, then the sequences of local optimal solutions* $\{\hat{\boldsymbol{p}}_k^{(i)*}[l]\}_{l=0}^{\infty}$ *generated in line 8 by the DMHS Algorithm 7 satisfy*

$$\lim_{l\to\infty} \hat{\boldsymbol{p}}_k^{(i)*}[l] = \hat{\boldsymbol{p}}_k, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathbb{N}_N. \tag{9.34}$$

Proof. Since $-q_k(\underline{\boldsymbol{\lambda}}_k)$ is a strictly convex function and the sequences of step lengths $\{\gamma_k[l]\}_{l=0}^{\infty}$ satisfy $\forall k \in \mathbb{N}_N$ the conditions stated in (9.33), we can apply Theorem 2.4.6. Since the condition $\gamma_k[l]\, \|\boldsymbol{s}_k^{(i,j)}[l]\| \leq c$ holds, this theorem ensures that the sequences $\{\boldsymbol{\lambda}_k^{(i,j)}[l]\}_{l=1}^{\infty}$, $(i,j) \in \bar{\mathcal{E}}$, generated by the update rule stated in line 6 of the DMHS Algorithm 7 satisfy

$$\lim_{l\to\infty} \boldsymbol{\lambda}_k^{(i,j)}[l] = \boldsymbol{\lambda}_k^{(i,j)*}, \quad \forall (i,j) \in \bar{\mathcal{E}}, \quad \forall k \in \mathbb{N}_N, \tag{9.35a}$$

$$\lim_{l\to\infty} q_k(\underline{\boldsymbol{\lambda}}_k[l]) = d_k^*, \quad \forall k \in \mathbb{N}_N. \tag{9.35b}$$

Since (i) strong duality holds, (ii) the Lagrangian $L_k(\underline{\hat{\boldsymbol{p}}}_k, \underline{\boldsymbol{\lambda}}_k^*)$ is $\forall k \in \mathbb{N}_N$ a strictly convex function of $\underline{\hat{\boldsymbol{p}}}_k$ and (iii) Assumption 20 holds, which ensures equality of all local solutions $\hat{\boldsymbol{p}}_k^{(i)}$, i. e. $\hat{\boldsymbol{p}}_k^{(i)} = \hat{\boldsymbol{p}}_k^{(j)}, \forall i,j \in \mathcal{V}$, the sequences $\{\hat{\boldsymbol{p}}_k^{(i)*}[l]\}_{l=1}^{\infty}$ generated by the update rule stated in

line 8 of the DMHS Algorithm 7 satisfy

$$\lim_{l\to\infty} \hat{\boldsymbol{p}}_k^{(i)*}[l] = \hat{\boldsymbol{p}}_k, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathbb{N}_N, \tag{9.36a}$$

$$\lim_{l\to\infty} L_k(\underline{\hat{\boldsymbol{p}}}_k[l], \underline{\boldsymbol{\lambda}}_k[l]) = p_k^* = d_k^*, \quad \forall k \in \mathbb{N}_N, \tag{9.36b}$$

and thus (9.34) holds. □

Note that boundedness of the subgradient $\boldsymbol{s}_k^{(i,j)}[l]$ depends on boundedness of $\hat{\boldsymbol{p}}_k^{(i)*}[l]$ and $\boldsymbol{\lambda}_k^{(i,j)}[l]$. This has two severe consequences. First, boundedness of $\boldsymbol{s}_k^{(i,j)}[l]$ cannot be guaranteed in advance. Second, not all step size sequences $\{\gamma_k[l]\}_{l=0}^{\infty}$ which satisfy the first three conditions in (9.33) lead to bounded and convergent sequences $\{\hat{\boldsymbol{p}}_k^{(i)*}[l]\}_{l=1}^{\infty}$ and $\{\boldsymbol{\lambda}_k^{(i,j)}[l]\}_{l=1}^{\infty}$. Therefore, the step size sequences have to additionally satisfy the fourth condition in (9.33). For instance, the step size $\gamma_k[l] = a/\sqrt{l}$ satisfies for all positive a the first three conditions in (9.33), but the satisfaction of the last one depends on the choice of a.

9.4.2 Special Case: Complete Directed Communication Graph

The convergence behavior of $\hat{\boldsymbol{p}}_k^{(i)}$ and thus the stability of the DMHS for a complete directed graph $\mathcal{G}$ is analyzed in the following Theorem.

Theorem 9.4.2. *Suppose that $\mathcal{G}$ is a complete directed graph without any self-loops. If the update rule stated in line 6 of the DMHS Algorithm 7 is replaced by*

$$\boldsymbol{\lambda}_k^{(i,j)*}[l] = -\boldsymbol{N}_k^{(j)} \left(\sum_{a=1}^{m} \boldsymbol{N}_k^{(a)} \right)^{-1} \boldsymbol{o}_k^{(i)}, \quad (i,j) \in \bar{\mathcal{E}}, \tag{9.37}$$

then the optimal solution $\hat{\boldsymbol{p}}_k$ is attained in the first iteration of the update rule in line 8 of the DMHS Algorithm 7, i. e.

$$\hat{\boldsymbol{p}}_k^{(i)*}[1] = \hat{\boldsymbol{p}}_k, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathbb{N}_N. \tag{9.38}$$

Proof. Since $\mathcal{G}$ is a complete directed graph without self-loops, its adjacency matrix is all 1's except for 0's on the diagonal. By denoting $\boldsymbol{N}_k \triangleq \sum_{a=1}^{m} \boldsymbol{N}_k^{(a)}$ and $\boldsymbol{o}_k \triangleq \sum_{i=1}^{m} \boldsymbol{o}_k^{(i)}$, we can write (9.37) as $\boldsymbol{\lambda}_k^{(i,j)*} = -\boldsymbol{N}_k^{(j)} \boldsymbol{N}_k^{-1} \boldsymbol{o}_k^{(i)}$. Inserting both results into the update rule stated in line 8 of the DMHS, we get

$$\begin{aligned}
\hat{\boldsymbol{p}}_k^{(i)*}[1] &= \boldsymbol{N}_k^{(i)^{-1}} \Big(\boldsymbol{o}_k^{(i)} + \sum_{\substack{j=1 \\ j\neq i}}^{m} \Big(-\boldsymbol{N}_k^{(j)} \boldsymbol{N}_k^{-1} \boldsymbol{o}_k^{(i)} + \boldsymbol{N}_k^{(i)} \boldsymbol{N}_k^{-1} \boldsymbol{o}_k^{(j)} \Big) \Big) \\
&= \boldsymbol{N}_k^{(i)^{-1}} \Big(\boldsymbol{o}_k^{(i)} - \big(\boldsymbol{N}_k - \boldsymbol{N}_k^{(i)} \big) \boldsymbol{N}_k^{-1} \boldsymbol{o}_k^{(i)} + \boldsymbol{N}_k^{(i)} \boldsymbol{N}_k^{-1} \big(\boldsymbol{o}_k - \boldsymbol{o}_k^{(i)} \big) \Big) \\
&= \boldsymbol{N}_k^{-1} \boldsymbol{o}_k = \hat{\boldsymbol{p}}_k,
\end{aligned}$$

which holds for all $i \in \mathcal{V}$. □

9.5 Special Case: Distributed Kalman Filter

Consider the case where the DMHE involves only the most recent measurements $\boldsymbol{y}_k^{[1]}, \dots, \boldsymbol{y}_k^{[q]}$, i. e. $N+1=1$, distributed among the m estimators such that each one is assigned with at least one measurement $\boldsymbol{y}_k^{[i]}$, i. e. $q \geq m$. Then Algorithm 7 becomes a distributed Kalman filter (DKF).

Lemma 9.5.1. *Suppose that $N=0$ and $q \geq m$. Then the update rule in line 8 of the DMHS Algorithm 7 reduces to the i-th state estimate update*

$$\hat{\boldsymbol{p}}_k^{(i)*}[l] = \hat{\boldsymbol{x}}_k^{(i)*}[l] = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k^{(i)}\left(\underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{C}_k^{(i)}\bar{\boldsymbol{x}}_k\right) + m\boldsymbol{P}_k^{(i)} \sum_{j=1}^{m} \left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}[l]\right) \tag{9.39a}$$

with the i-th Kalman gain matrix

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_k \boldsymbol{C}_k^{(i)^T}\left(\boldsymbol{C}_k^{(i)}\boldsymbol{P}_k\boldsymbol{C}_k^{(i)^T} + \frac{1}{m}\boldsymbol{R}_k^{(i)}\right)^{-1} \tag{9.39b}$$

and the i-th updated covariance matrix

$$\boldsymbol{P}_k^{(i)} = \left(\boldsymbol{I} - \boldsymbol{K}_k^{(i)}\boldsymbol{C}_k^{(i)}\right)\boldsymbol{P}_k. \tag{9.39c}$$

Proof. By setting $N=0$ and $q \geq m$, we have from Lemma 9.2.3 that $\boldsymbol{N}_k^{(i)} = \boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\boldsymbol{C}_k^{(i)} + \left(m\boldsymbol{P}_k\right)^{-1}$ and $\boldsymbol{o}_k^{(i)} = \boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\underline{\boldsymbol{y}}_k^{(i)} + \left(m\boldsymbol{P}_k\right)^{-1}\bar{\boldsymbol{x}}_k$. Consequently, we can write the update rule in line 8 of the DMHS Algorithm 7 as

$$\begin{aligned}\hat{\boldsymbol{p}}_k^{(i)*}[l] = \hat{\boldsymbol{x}}_k^{(i)*}[l] = \bar{\boldsymbol{x}}_k &+ \left(\boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\boldsymbol{C}_k^{(i)} + \left(m\boldsymbol{P}_k\right)^{-1}\right)^{-1} \\ &\cdot \left(\boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\left(\underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{C}_k^{(i)}\bar{\boldsymbol{x}}_k\right) + \sum_{j=1}^{m}\left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}[l]\right)\right).\end{aligned}$$

Combining this with the definition of the i-th Kalman gain matrix

$$\begin{aligned}\boldsymbol{K}_k^{(i)} &\triangleq \left(\boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\boldsymbol{C}_k^{(i)} + \left(m\boldsymbol{P}_k\right)^{-1}\right)^{-1}\boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}} \\ &= m\boldsymbol{P}_k\boldsymbol{C}_k^{(i)^T}\left(\boldsymbol{C}_k^{(i)}m\boldsymbol{P}_k\boldsymbol{C}_k^{(i)^T} + \underline{\boldsymbol{R}}_k^{(i)}\right)^{-1} \\ &= \boldsymbol{P}_k\boldsymbol{C}_k^{(i)^T}\left(\boldsymbol{C}_k^{(i)}\boldsymbol{P}_k\boldsymbol{C}_k^{(i)^T} + \frac{1}{m}\underline{\boldsymbol{R}}_k^{(i)}\right)^{-1}\end{aligned}$$

and the definition of the i-th updated covariance matrix

$$\begin{aligned}m\boldsymbol{P}_k^{(i)} &\triangleq \left(\boldsymbol{C}_k^{(i)^T}\underline{\boldsymbol{R}}_k^{(i)^{-1}}\boldsymbol{C}_k^{(i)} + \left(m\boldsymbol{P}_k\right)^{-1}\right)^{-1} \\ &= \left(\boldsymbol{I} - \boldsymbol{K}_k^{(i)}\boldsymbol{C}_k^{(i)}\right)m\boldsymbol{P}_k.\end{aligned}$$

yields (9.39) and concludes the proof. □

Table 9.2: Summary of the centralized and distributed discrete Kalman filter equations.

	centralized Kalman filter (CKF)	distributed Kalman filter (DKF)
System Model	$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}$	$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}$
Measurement Model	$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k$	$\underline{\boldsymbol{y}}_k^{(i)} = \boldsymbol{C}^{(i)}\boldsymbol{x}_k, \quad i \in \mathcal{V}$
State Estimate Prediction	$\hat{\boldsymbol{x}}_{k\|k-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1\|k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}$	$\hat{\boldsymbol{x}}_{k\|k-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1\|k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1}$
Covariance Extrapolation	$\boldsymbol{P}_{k\|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1\|k-1}\boldsymbol{A}^T + \boldsymbol{Q}$	$\boldsymbol{P}_{k\|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1\|k-1}\boldsymbol{A}^T + \boldsymbol{Q}$

State Estimate Update	$\hat{\boldsymbol{x}}_{k\|k} = \hat{\boldsymbol{x}}_{k\|k-1} + \boldsymbol{K}_k\big(\boldsymbol{y}_k - \boldsymbol{C}\hat{\boldsymbol{x}}_{k\|k-1}\big)$	$\boldsymbol{\lambda}_k^{(i,j)}[0] = \boldsymbol{0}, \forall (i,j) \in \bar{\mathcal{E}}, \quad \hat{\boldsymbol{x}}_{k\|k}^{(i)}[0] = \boldsymbol{0}, \forall i \in \mathcal{V}$
		for $l = 0$ **to** $l_{\max}$ **do**
		$\boldsymbol{\lambda}_k^{(i,j)}[l] = \boldsymbol{\lambda}_k^{(i,j)}[l-1] - \gamma_k[l-1]\left(\hat{\boldsymbol{x}}_{k\|k}^{(i)}[l-1] - \hat{\boldsymbol{x}}_{k\|k}^{(j)}[l-1]\right)$
		$\hat{\boldsymbol{x}}_{k\|k}^{(i)}[l] = \hat{\boldsymbol{x}}_{k\|k-1} + \boldsymbol{K}_k^{(i)}\left(\underline{\boldsymbol{y}}_k^{(i)} - \boldsymbol{C}_k^{(i)}\hat{\boldsymbol{x}}_{k\|k-1}\right)$
		$+ m\boldsymbol{P}_{k\|k}^{(i)} \sum_{j=1}^{m}\left(\bar{\omega}_{ij}\boldsymbol{\lambda}_k^{(i,j)}[l] - \bar{\omega}_{ji}\boldsymbol{\lambda}_k^{(j,i)}[l]\right)$
		end for

Kalman Gain Matrix	$\boldsymbol{K}_k = \boldsymbol{P}_{k\|k-1}\boldsymbol{C}^T\Big(\boldsymbol{C}\boldsymbol{P}_{k\|k-1}\boldsymbol{C}^T + \boldsymbol{R}\Big)^{-1}$	$\boldsymbol{K}_k = \boldsymbol{P}_{k\|k-1}\boldsymbol{C}^T\Big(\boldsymbol{C}\boldsymbol{P}_{k\|k-1}\boldsymbol{C}^T + \boldsymbol{R}\Big)^{-1}$ $\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_{k\|k-1}\boldsymbol{C}_k^{(i)^T}\Big(\boldsymbol{C}_k^{(i)}\boldsymbol{P}_{k\|k-1}\boldsymbol{C}_k^{(i)^T} + \frac{1}{m}\underline{\boldsymbol{R}}_k^{(i)}\Big)^{-1}$
Covariance Update	$\boldsymbol{P}_{k\|k} = \big(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}\big)\boldsymbol{P}_{k\|k-1}$	$\boldsymbol{P}_{k\|k} = \big(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}\big)\boldsymbol{P}_{k\|k-1}$ $\boldsymbol{P}_{k\|k}^{(i)} = \big(\boldsymbol{I} - \boldsymbol{K}_k^{(i)}\boldsymbol{C}_k^{(i)}\big)\boldsymbol{P}_{k\|k-1}$

The equations for the centralized Kalman filter (CKF) and the distributed Kalman filter (DKF) are summarized in Table 9.2 in order to contrast the similarities and differences between both versions. Therefore, we use the established notation in the estimation framework, where the notation $\hat{\boldsymbol{x}}_{i|j}$ represents the estimate of $\boldsymbol{x}$ at time i given observations up to, and including at time j. Note that in this notation, $\bar{\boldsymbol{x}}_k$ is described as $\hat{\boldsymbol{x}}_{k|k-1}$.

The equations for the state estimate prediction, covariance extrapolation, Kalman gain matrix and the covariance update of the CKF and the DKF are identical. However, since the measurement $\boldsymbol{y}_k$ is only distributed available for the DKF, the state estimate update equation differs and results for the DKF in the adapted version of Algorithm 7. The first part of the update equation for $\hat{\boldsymbol{x}}_{k|k}^{(i)}[l]$ is a conventional Kalman filter which utilizes due to the local measurement $\underline{\boldsymbol{y}}_k^{(i)}$ a local covariance matrix $\boldsymbol{P}_k^{(i)}$ and a local Kalman gain matrix $\boldsymbol{K}_k^{(i)}$. The second part is a consensus term which guarantees convergence of each local updated state estimate $\hat{\boldsymbol{x}}_{k|k}^{(i)}$ to the updated global state estimate $\hat{\boldsymbol{x}}_{k|k}$. Since the DKF is a special version of the DMHE, the stability analysis presented in Section 9.4 also hold for the DKF.

9.6 Extensions

In this section, we sketch the extensions for the presented DMHS to the following three cases:

① linear time-variant system and measurement model (Section 9.6.1),

② state and disturbance constraints (Section 9.6.2), and

③ packet-delaying and packet-dropping network $\Sigma_{\mathrm{N}}^{\{1\}}$ (Section 9.6.3).

9.6.1 Linear Time-Variant System and Measurement Model

Consider the problems given in Section 9.1 where the plant Σ is described instead of the linear time-invariant system (9.2) by the linear time-variant system

$$\boldsymbol{x}_{i+1} = \boldsymbol{A}_i \boldsymbol{x}_i + \boldsymbol{B}_i \boldsymbol{u}_i + \boldsymbol{w}_i \tag{9.40}$$

and the linear time-invariant measurement model (9.3) is replaced by its linear time-variant counterpart

$$\boldsymbol{y}_i^{[j]} = \boldsymbol{C}_i^{[j]} \boldsymbol{x}_i + \boldsymbol{v}_i^{[j]}, \quad j \in \mathcal{S}, \quad i \in \mathbb{N}_0. \tag{9.41}$$

The extension of the DMHS to deal with this situation is straightforward. Roughly speaking, the main challenge consists in replacing the time-invariant matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ by its time-variant counterparts $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i$, respectively. Everything else including strong duality and the stability analyzes presented in Section 9.4 still hold. The only difference in Algorithm 7 is that the matrix $\boldsymbol{N}_k^{(i)}$ and the vector $\boldsymbol{o}_k^{(i)}$ required in line 8 are build with the time-variant matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i$ instead of the time-invariant counterparts.

9.6.2 State and Disturbance Constraints

As discussed in Section 2.1, one favorable property of the moving horizon strategies is the possibility of directly handling constraints. Consider the problems given in Section 9.1 where the states and disturbances satisfy additionally $\forall i \in \mathbb{N}_0$ the following constraints

$$\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad \boldsymbol{w}_i \in \mathcal{W} \subset \mathbb{R}^{n_x}, \quad \boldsymbol{v}_i^{[j]} \in \mathcal{V}^{[j]} \subset \mathbb{R}^{n_{y,j}}, j \in \mathcal{S}, \tag{9.42}$$

where the sets $\mathcal{X}$, $\mathcal{W}$ and $\mathcal{V}^{[j]}$, $j \in \mathcal{S}$ are nonempty, bounded and convex with $\mathbf{0} \in \mathcal{W}$ and $\mathbf{0} \in \mathcal{V}^{[j]}$, $j \in \mathcal{S}$.

To deal with this situation, we extend the DMHS as follows. First, we express the constraints (9.42) in terms of constraints for the primal variable $\hat{\boldsymbol{p}}_k$, i. e. $\hat{\boldsymbol{p}}_k \in \mathcal{P}_k$. Consequently, we extend (9.17) by these constraints as follows

$$p_k^* = \min_{\hat{\boldsymbol{p}}_k^{(1)}} J_k^{(1)}(\hat{\boldsymbol{p}}_k^{(1)}) + \ldots + \min_{\hat{\boldsymbol{p}}_k^{(m)}} J_k^{(m)}(\hat{\boldsymbol{p}}_k^{(m)}) \tag{9.43a}$$

$$\text{subject to} \qquad \hat{\boldsymbol{p}}_k^{(i)} = \hat{\boldsymbol{p}}_k^{(j)}, \qquad (i,j) \in \mathcal{E}, \tag{9.43b}$$

$$\hat{\boldsymbol{p}}_k^{(i)} \in \mathcal{P}_k, \qquad i \in \mathcal{V}. \tag{9.43c}$$

This problem is still convex and strictly feasible which implies strong duality to hold according to Theorem 2.4.11. Thus, we can use once again the dual problem to solve the primal problem. The additional introduced constraints carry over as follows to the dual functions

$$q_k^{(i)}(\underline{\boldsymbol{\lambda}}_k) = \min_{\hat{\boldsymbol{p}}_k^{(i)} \in \mathcal{P}_k} L_k^{(i)}(\hat{\boldsymbol{p}}_k^{(i)}, \underline{\boldsymbol{\lambda}}_k). \tag{9.44}$$

The resulting dual problem is now solved by the *projected* subgradient method

$$\boldsymbol{\lambda}_k^{(i,j)}[l+1] = P_{\mathcal{P}_k}\left(\boldsymbol{\lambda}_k^{(i,j)}[l] - \gamma_k[l]\, \boldsymbol{s}_k^{(i,j)}[l]\right), \quad \forall (i,j) \in \bar{\mathcal{E}}, \quad l \in \mathbb{N}_0, \tag{9.45}$$

where

$$P_{\mathcal{P}_k}(\boldsymbol{a}) \triangleq \arg\min_{\boldsymbol{b} \in \mathcal{P}_k} \|\boldsymbol{a} - \boldsymbol{b}\| \tag{9.46}$$

is the projector on $\mathcal{P}_k$. The remaining quantities are identical as for the subgradient method (9.25) including the subgradients. Similar convergence Theorems exists for the projected subgradient method (Bertsekas et al., 2003; Kiwiel, 2004). As discussed in Section 2.1, the CMHS and therefore the DMHS are stable estimators for linear constrained problems. The only difference in Algorithm 7 is that the update step in line 8 is replaced by its projected version (9.45).

9.6.3 Non-Ideal Network

As discussed in Section 9.1, the assumption of an ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ might be inadequate for describing all possibilities of measurement transmission. Therefore, we consider the problems given in Section 9.1 where the Assumption 18 is replaced by the following one.

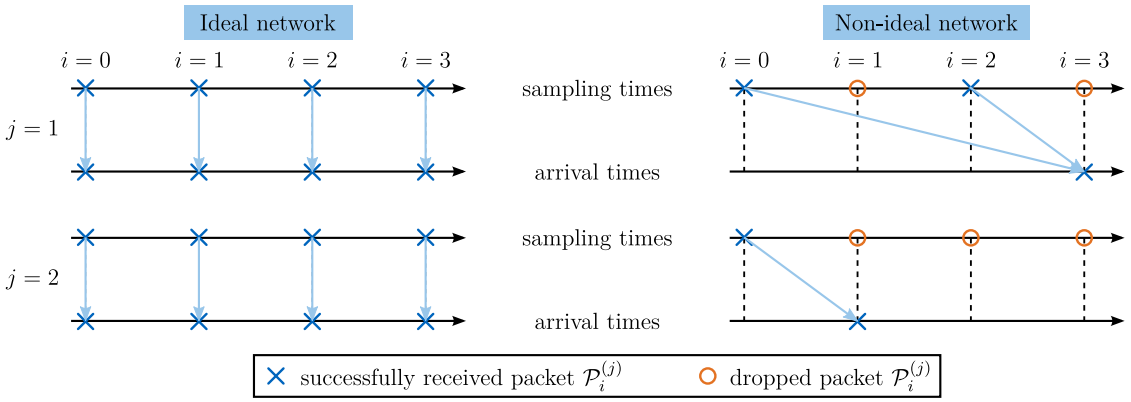
Assumption 21. Each packet $\mathcal{P}_i^{[j]}$ is $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}_0$ transmitted over the non-ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ to only one observer/estimator $\hat{\Sigma}^{(l)}$ which may vary over time. Thereby, the statistics of the time delays $\tau_i^{[j]} \in \mathbb{N}_0$ of the packets $\mathcal{P}_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}_0$ unknown. Consequently, the time delays $\tau_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}_0$ considered as deterministic variables of unknown character which take their values from the bounded set

$$\tau_i^{[j]} \in [0, \tau_{\max}^{[j]}] \subset \mathbb{N}_0, \quad j \in \mathcal{S}, \quad i \in \mathbb{N}_0. \tag{9.47}$$

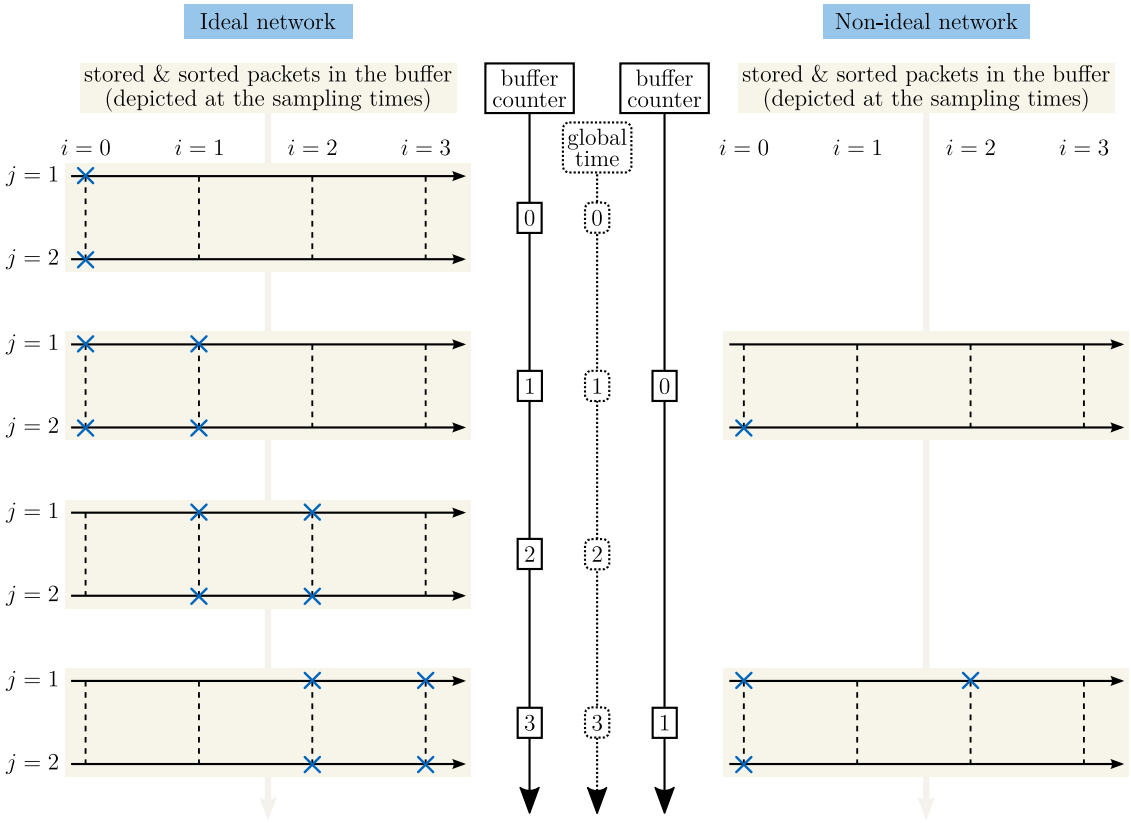
Similar, the statistics of the packet drop probability of the packets $\mathcal{P}_i^{[j]}$ are $\forall j \in \mathcal{S}$ and $\forall i \in \mathbb{N}_0$ unknown. Consequently, the maximum number of consecutive packet drops for the j-th sensor is bounded by $N_{\max,\mathrm{drop}}^{[j]} \in \mathbb{N}_0$, $j \in \mathcal{S}$.

This assumption models a non-ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ where, in general, only bounds but not the precise network statistics are known in advance. Note that this assumption is similar to the Assumptions 5 and 6.

The extension of the DMHS to the present case is more a notational challenge than a theoretical problem. Since the network is non-ideal, we cannot use the previous notation anymore. Thus, we have to extend the buffer based notation introduced in the Sections 3.2 and 3.3 to the present case of m decentralized buffers. This requires the introduction of a cumbersome notation which is necessary to formally describe the DMHS. Fortunately, this step is unnecessary for describing the basic ideas required for extending the DMHS. To this end, we consider the exemplary situation depicted in Figure 9.2(a) where two sensors $\Sigma_{\mathrm{S}}^{[j]}$, $j = 1, 2$, transmit their packets $\mathcal{P}_i^{[j]}$ for $i = 0, 1, 2, 3$ one time over the ideal network and another time over the non-ideal network to a single buffer $\mathcal{B}_k$. For the latter case, we can utilize the notation introduced in the Sections 3.2 and 3.3 and do not need an extended version. Depending on whether or not the packets $\mathcal{P}_i^{[j]}$ are successfully transmitted over the network, they are marked on the sampling time axises with either a blue cross or a red circle. Moreover, the successfully transmitted packets are connected to their corresponding counterparts located on the arrival time axis by an arrow. This representation illustrates the fact that in contrast to the case of an ideal network, the arrival times are not necessarily tantamount to the sampling times for the case of a non-ideal network. The resulting consequences combined with those of packet drops are illustrated for the buffer $\mathcal{B}_k$ in Figure 9.2(b) which depicts the temporal evolvement of the stored packets in the buffer for a buffer size of $N + 1 = 2$. For the ideal network, the buffer index k is identical to the global time i and the structure of the stored packets in the buffer is always identical. However, the situation for the non-ideal network is different. In this case, the buffer index k generally differs from the global time i and the structure of the



(a) Run of the packets $\mathcal{P}_i^{[j]}$ through the network $\Sigma_{\mathrm{N}}^{\{1\}}$ to the buffer $\mathcal{B}_k$ for $i = 0, 1, 2, 3$ and $j = 1, 2$.



(b) Temporal evolvement of the stored packets in the buffer $\mathcal{B}_k$ for a buffer size of $N + 1 = 2$ and of the buffer counter k resulting from the scenario depicted in (a).

Figure 9.2: Comparison of the effects on the buffer $\mathcal{B}_k$ caused by packet transmission over an ideal and a non-ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ based on an exemplary situation.

stored packets in the buffer varies with the buffer index k. This structure has to be reflected in the constraints of the DMHS which describe the dynamics of the measurements stored in the current buffer. While these constraints are for the ideal network time-invariant, i. e.

$$\hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}\hat{\boldsymbol{x}}_i + \boldsymbol{B}\boldsymbol{u}_i + \hat{\boldsymbol{w}}_i \tag{9.48a}$$

$$\boldsymbol{y}_i = \boldsymbol{C}\hat{\boldsymbol{x}}_i + \hat{\boldsymbol{v}}_i, \tag{9.48b}$$

they become *time-variant* for the non-ideal network due to the time-varying structure, i. e.

$$\hat{\boldsymbol{x}}_{i+1|k} = \boldsymbol{A}_i \hat{\boldsymbol{x}}_{i|k} + \boldsymbol{g}_i(\boldsymbol{A}, \boldsymbol{B}) \underline{\boldsymbol{u}}_k + \hat{\boldsymbol{w}}_{i|k} \tag{9.49a}$$

$$\boldsymbol{y}_{i|k} = \boldsymbol{C}_i \hat{\boldsymbol{x}}_{i|k} + \hat{\boldsymbol{v}}_{i|k}. \tag{9.49b}$$

The time-variance in (9.49a) and (9.49b) is due to the non-equidistant spacing of the sampling times associated to the buffer $\mathcal{B}_k$ and the non-availability of all measurement at a certain sampling time associated to the buffer $\mathcal{B}_k$, respectively. The former and latter case can be seen in Figure 9.2 in the non-ideal case by vertical and horizontal gaps in the stored packets of the buffer $\mathcal{B}_2$, respectively. Moreover, note that in general the measurements $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i|k}$ are not identical and thus the estimated disturbances $\hat{\boldsymbol{w}}_i$ and $\hat{\boldsymbol{w}}_{i|k}$ as well as $\hat{\boldsymbol{v}}_i$ and $\hat{\boldsymbol{v}}_{i|k}$ differ. As a consequence, we have to replace the time-invariant dynamics of the measurements corresponding to the buffer $\mathcal{B}_k$

$$\underline{\hat{\boldsymbol{y}}}_k = \boldsymbol{F} \hat{\boldsymbol{x}}_{k-N} + \boldsymbol{G} \underline{\boldsymbol{u}}_k + \boldsymbol{H} \underline{\hat{\boldsymbol{w}}}_k + \underline{\hat{\boldsymbol{v}}}_k \tag{9.50}$$

with the time-variant counterpart

$$\underline{\hat{\boldsymbol{y}}}_k^{\text{non-ideal}} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k-N|k} + \boldsymbol{G}_k \underline{\boldsymbol{u}}_k + \boldsymbol{H}_k \underline{\hat{\boldsymbol{w}}}_k^{\text{non-ideal}} + \underline{\hat{\boldsymbol{v}}}_k^{\text{non-ideal}} \tag{9.51}$$

As discussed in Section 9.6.1, it is straightforward to extend the DMHS to deal with the present time-variant but linear case. The only difference in Algorithm 7 is that the matrix $\boldsymbol{N}_k^{(i)}$ and the vector $\boldsymbol{o}_k^{(i)}$ required in line 8 are build based on the time-variant representation (9.49) instead of the time-invariant counterpart (9.48).

Due to the time-varying structure of the stored packets in the buffer $\mathcal{B}_k$, observability and therefore solvability of the primal and dual problem is only guaranteed if the moving horizon size $N+1$ is chosen properly. This means that depending on the network parameters $\tau_{\max}^{[j]}$, $N_{\max,\text{drop}}^{[j]}$, $j \in \mathcal{S}$, the size of the moving horizon $N+1$ has to be chosen sufficiently large or adapted dynamically. Once this is assured, the stability analyzes presented in Section 9.4 hold, i. e. the state estimates derived by the DMHS are identical to those of the CMHS. For the stability analyses of the CMHS, we can invoke the results derived in Chapter 6 after adapting some of the assumptions made in Section 9.1 to the ones made in Section 3.1.

9.7 Summary

In this chapter, we have presented the DMHO and the DMHE within a common framework for the undisturbed and disturbed distributed NCS architecture, respectively. The main feature of this architecture is the possibility of implementing any centralized designed controller fully decentralized. This is enabled by both DMHS which provide distributed knowledge about the full state of the system. The key step to derive the DMHS is the extension of the CMHS

optimization problems by additional suitably designed consensus constraints which reflect the communication topology among the observers/estimators by means of a directed graph. This reformulation allows us to apply the dual decomposition technique to derive a decoupled dual optimization problem which we have solved by a suitable subgradient method. The resulting algorithm possesses a simple structure and alternates between updating and transmitting the primal and dual variables. Convergence of this algorithm and therefore stability of the DMHS is assured for several step sizes if the following three conditions are fulfilled. First, the system has to be observable. Second, the directed graph, which represents the communication topology among the observers/estimators, has to be weakly connected. Third, the allocation of the measurements to the observers of the DMHO has to imply full-rankness of a certain matrix which is assured by a periodic measurement allocation provided that the system satisfies a certain form of observability. As a consequence, the DMHO requires a certain minimum moving horizon length while the DMHE can operate with measurements from one time step only. In this case, the DMHE becomes a distributed Kalman filter. Finally, we have sketched the extension of the DMHS to the cases of time-variant system and measurement models, state and disturbance constraints, and non-ideal networks.

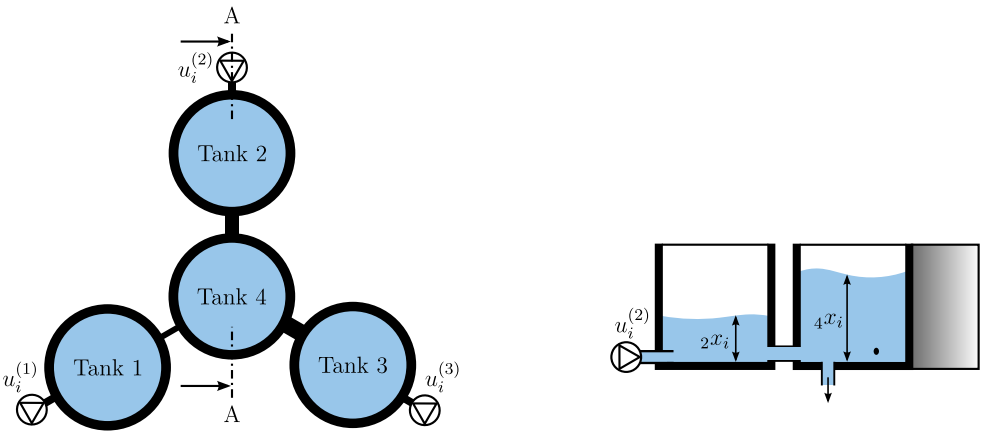
Chapter 10

Simulation Results

In this chapter, we present simulation results for both developed distributed moving horizon strategies (DMHS) applied to a networked four tank system. In Section 10.1, we describe the simulation setup of the considered distributed NCS architecture. Thereby, the plant Σ consists of four interconnected tanks. Three of these tanks are equipped with an actuator $\Sigma_{\mathrm{A}}^{(j)}$, a sensor $\Sigma_{\mathrm{S}}^{[j]}$ and a distributed observer/estimator $\hat{\Sigma}^{(j)}$. This setup provides the possibility of investigating several strategies for the measurement allocation as well as the communication topology among the observers/estimators. The corresponding results are presented in Section 10.2 for varies settings of the distributed moving horizon observer (DMHO) and the distributed moving horizon estimator (DMHE). This also includes a comparison between both methods. Finally, we conclude this chapter with a summary given in Section 10.3.

10.1 Simulation Setup

Consider the distributed NCS architecture depicted in Figure 9.1. The plant Σ is the four tank system shown in Figure 10.1 and can be described by the fourth-order discrete-time



(a) Top view.

(b) Cross section A-A.

Figure 10.1: Schematic diagram of the four tank system.

linear system

$$\boldsymbol{x}_{i+1} = \begin{bmatrix} 0.50 & 0 & 0 & 0.07 \\ 0 & 0.50 & 0 & 0.35 \\ 0 & 0 & 0.50 & 0.56 \\ 0.07 & 0.35 & 0.56 & -0.48 \end{bmatrix} \boldsymbol{x}_i + \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.8 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{u}_i + \boldsymbol{w}_i. \tag{10.1}$$

This system involves four states $\boldsymbol{x}_i = [{}_1x_i, {}_2x_i, {}_3x_i, {}_4x_i]^T \in \mathbb{R}^4$ corresponding to the fill levels of the four tanks with the initial condition $\boldsymbol{x}_0 = [20, 30, 40, 23.1]^T$, the zero-mean state disturbance $\boldsymbol{w}_i \in \mathbb{R}^4$ with covariance $\boldsymbol{Q} \in \mathbb{R}^{4\times4}$ and the control inputs $\boldsymbol{u}_i = [u_i^{(1)}, u_i^{(2)}, u_i^{(3)}]^T \in \mathbb{R}^3$ of the three pump voltages.

The first three states are measured by the three sensors $\Sigma_{\mathrm{S}}^{[j]}$, $j = 1, 2, 3$ according to the measurement models

$$y_i^{[1]} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_i + v_i^{[1]}, \tag{10.2a}$$
$$y_i^{[2]} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}_i + v_i^{[2]}, \tag{10.2b}$$
$$y_i^{[3]} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}_i + v_i^{[3]}, \tag{10.2c}$$

where $v_i^{[j]}$, $j = 1, 2, 3$, are zero-mean measurement disturbances with variance $R^{[j]} \in \mathbb{R}$, $j = 1, 2, 3$. Note that the system (10.1) is not observable by one of the three sensors alone, i. e. the pairs $(\boldsymbol{C}^{[j]}, \boldsymbol{A})$, $j = 1, 2, 3$ are not observable, but by all sensors combined, i. e. the pair $(\boldsymbol{C}, \boldsymbol{A})$ with $\boldsymbol{C} \triangleq \mathrm{col}(\boldsymbol{C}^{[j]}, j = 1, 2, 3)$ is observable. Each measurement $y_i^{[j]}$ is transmitted in a packet of the form $\mathcal{P}_i^{[j]} = \{y_i^{[j]}, i\}$ over the ideal network $\Sigma_{\mathrm{N}}^{\{1\}}$ to one of the $m = 3$ observers/estimators. These are placed next to the different pumps. The measurements are allocated to the observers/estimators according to one of the following two methods:

Local measurement allocation: The measurements of the j-th sensor are allocated to the j-th estimator. The length of the moving horizon is set to $N + 1 = 1$ and the associated sets are $\mathcal{I}_k = \{k\}$ and $\mathcal{S} = \{1, 2, 3\}$. This results for $k \in \mathbb{N}_0$ in the following index sets

$$\begin{aligned} \mathcal{Y}_k^{(1)} &= \{(k, 1)\}, \\ \mathcal{Y}_k^{(2)} &= \{(k, 2)\}, \\ \mathcal{Y}_k^{(3)} &= \{(k, 3)\}. \end{aligned} \tag{10.3}$$

Periodic measurement allocation (Theorem 9.2.5): The $(j - 1 + 3i)$-th measurements of all sensors are allocated to the j-th observer/estimator for $j = 1, 2, 3$ and $i \in \mathbb{N}_0$. By choosing $\beta = 0$ in (9.29), the length of the moving horizon becomes $N + 1 = 6$ and the associated sets are $\mathcal{I}_k = \{k - 5, k - 4, k - 3, k - 2, k - 1, k\}$ and $\mathcal{S} = \{1, 2, 3\}$. This

results for $k \in \mathbb{N}_5$ according to (9.30) in the following index sets

$$\begin{aligned}
\mathcal{Y}_k^{(1)} &= \Big\{(3\lfloor\tfrac{k}{3}\rfloor \quad -2, 1), (3\lfloor\tfrac{k}{3}\rfloor \quad -2, 2), (3\lfloor\tfrac{k}{3}\rfloor \quad -2, 3), \\
&\qquad (3\lfloor\tfrac{k}{3}\rfloor \quad +1, 1), (3\lfloor\tfrac{k}{3}\rfloor \quad +1, 2), (3\lfloor\tfrac{k}{3}\rfloor \quad +1, 3)\Big\}, \\
\mathcal{Y}_k^{(2)} &= \Big\{(3\lfloor\tfrac{k-1}{3}\rfloor -1, 1), (3\lfloor\tfrac{k-1}{3}\rfloor -1, 2), (3\lfloor\tfrac{k-1}{3}\rfloor -1, 3), \\
&\qquad (3\lfloor\tfrac{k-1}{3}\rfloor +2, 1), (3\lfloor\tfrac{k-1}{3}\rfloor +2, 2), (3\lfloor\tfrac{k-1}{3}\rfloor +2, 3)\Big\}, \\
\mathcal{Y}_k^{(3)} &= \Big\{(3\lfloor\tfrac{k-2}{3}\rfloor \quad , 1), (3\lfloor\tfrac{k-2}{3}\rfloor \quad , 2), (3\lfloor\tfrac{k-2}{3}\rfloor \quad , 3), \\
&\qquad (3\lfloor\tfrac{k-2}{3}\rfloor +3, 1), (3\lfloor\tfrac{k-2}{3}\rfloor +3, 2), (3\lfloor\tfrac{k-2}{3}\rfloor +3, 3)\Big\}.
\end{aligned} \tag{10.4}$$

The local measurement allocation can only be applied to the DMHE while the periodic measurement allocation is suitable for both DMHS. Note that the assumptions of Theorem 9.2.5 are fulfilled because $\boldsymbol{A}$ has full rank and the pair $(\boldsymbol{C}, \boldsymbol{A}^3)$ is observable.

Each of the three observers/estimators $\hat{\Sigma}^{(j)}$, $j = 1, 2, 3$, are identically initialized with $\hat{\boldsymbol{x}}_0 = [0, 0, 0, 0]^T$ and use the step size rule $\gamma_k[l] = a/\sqrt{l}$ where the positive parameter a is adapted for the respective situation. The intercommunication between the observers/estimators is according to one of the three graphs depicted in Figure 10.2. These graphs $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_3$ represent a line, a ring and an all-to-all communication topology, respectively. Note that the latter case allows the usage of the optimal Lagrange variables derived in Theorem 9.4.2.

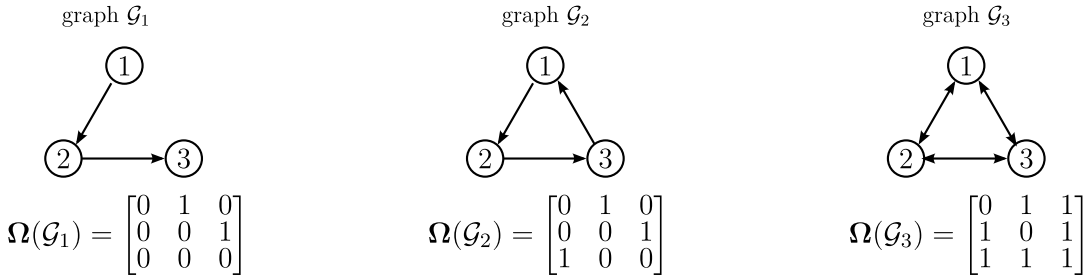


Figure 10.2: Three cases for the communication topology of the observers/estimators represented by the graphs $\mathcal{G}_i$ and the associated adjacency matrices $\boldsymbol{\Omega}(\mathcal{G}_i)$: line ($i = 1$), ring ($i = 2$) and all-to-all ($i = 3$).

The controller Σ_C is a *centralized designed but decentralized implemented* combined linear static feedforward and linear quadratic feedback controller

$$\boldsymbol{u}_i = -\boldsymbol{K}\boldsymbol{x}_i + (\boldsymbol{K}\boldsymbol{M}_x + \boldsymbol{M}_u)\boldsymbol{r}_i \tag{10.5}$$

with the reference input $\boldsymbol{r}_i = [{}_1r_i, {}_2r_i, {}_3r_i]^T \in \mathbb{R}^3$ and the matrices $\boldsymbol{K} \in \mathbb{R}^{3\times4}$, $\boldsymbol{M}_x \in \mathbb{R}^{4\times3}$, $\boldsymbol{M}_u \in \mathbb{R}^{3\times3}$. The feedforward part is calculated as the solution to

$$\begin{bmatrix} \boldsymbol{A} - \boldsymbol{I} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_x \\ \boldsymbol{M}_u \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}, \tag{10.6}$$

which results from the requirement that in the stationary case, i. e. $\boldsymbol{x}_{i+1} - \boldsymbol{x}_i = \boldsymbol{0}$, the conditions $\boldsymbol{y}_i = \boldsymbol{r}_i$ and $\boldsymbol{x}_i = \boldsymbol{x}_{i,\text{desired}} = \boldsymbol{M}_x \boldsymbol{r}_i$ have to be fulfilled. The LQR feedback gain $\boldsymbol{K}$ is determined for the state weighting matrix $\boldsymbol{Q}_{\text{LQR}} = \text{diag}(1, 1, 1, 10)$ and the control input weighting matrix $\boldsymbol{R}_{\text{LQR}} = \text{diag}(1, 1, 1)$, see Kwakernaak and Sivan (1972).

10.2 Simulation Results

In this section, we present the simulation results for the distributed moving horizon observer (DMHO) and the distributed moving horizon estimator (DMHE) as well as the comparison of both methods.

10.2.1 Distributed Moving Horizon Observer

Figure 10.3 depicts the closed-loop performance of the DMHO for the state disturbance covariance $\boldsymbol{Q} = \boldsymbol{0}$ and the measurement disturbance variances $R^{[j]} = 2.5$, $j = 1, 2, 3$. Thereby, the DMHO utilizes the periodic measurement allocation and the all-to-all communication topology $\mathcal{G}_3$ in combination with the optimal dual variables derived in Theorem 9.4.2 which provides convergence of the DMHO in the first iteration step. As long as $k < 5$, the feedback part of the controller is turned off and the state is predicted by a forward simulation based on $\hat{\boldsymbol{x}}_0$. After the moving horizon is filled, i. e. $k \geq 5$, the observers reconstruct the state of the system from the noisy measurements. Note that the depicted performance is identical to the case where a CMHO is used within a centralized NCS architecture.

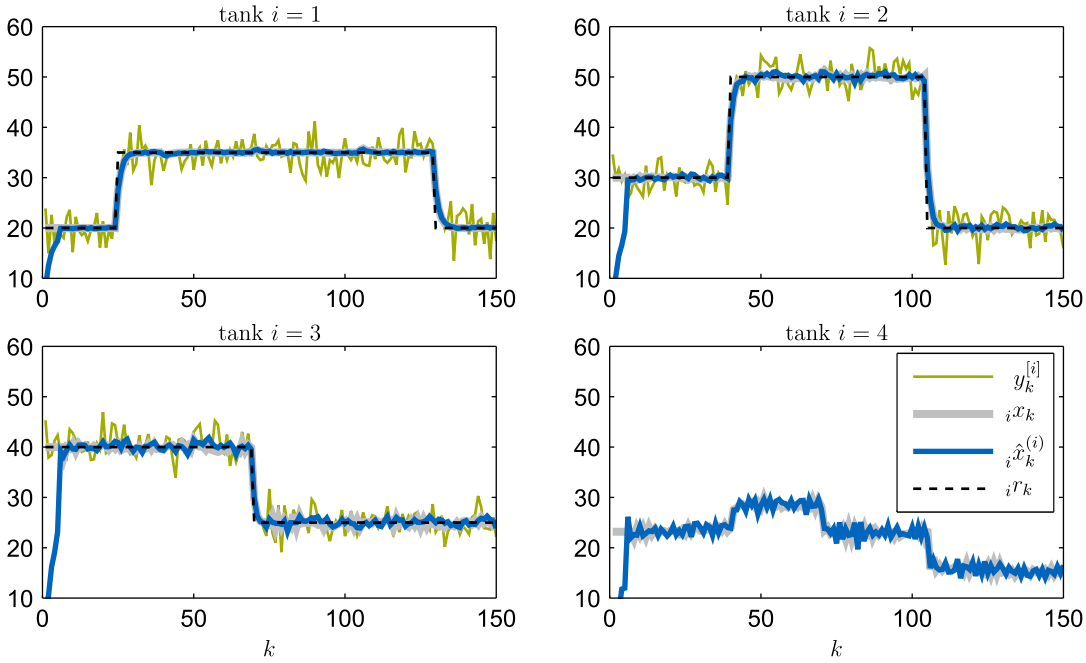


Figure 10.3: Comparison between the noisy measurements $y_k^{[i]}$, nominal trajectories ${}_i r_k$, true states ${}_i x_k$ and estimated states ${}_i \hat{x}_k^{(i)}$ generated by the DMHO (all-to-all communication topology $\mathcal{G}_3$ & optimal dual variables) for the tanks $i = 1, 2, 3, 4$.

10.2.2 Distributed Moving Horizon Estimator

Figure 10.4 depicts the closed-loop performance of the DMHE for the state disturbance covariance $\boldsymbol{Q} = \mathrm{diag}(1,1,1,1)$ and the measurement disturbance variances $R^{[j]} = 2.5$, $j = 1,2,3$. Thereby, the DMHE utilizes the local measurement allocation, the update of the covariance $\boldsymbol{P}_{k-N}$ according to the Riccati equation (9.10) with the initial value $\boldsymbol{P}_0 = \mathrm{diag}(1,1,1,1)$, the step size $\gamma_k[l] = 0.35/\sqrt{l}$ and the line communication topology $\mathcal{G}_1$. Since the length of the moving horizon is $N+1 = 1$, the DMHE reduces to the DKF presented in Section 9.5. The DMHE is able to reconstruct the state of the system right from the start, i. e. for all $k \geq 0$. Note that the depicted performance is identical to the case where a Kalman filter is used within a centralized NCS architecture.

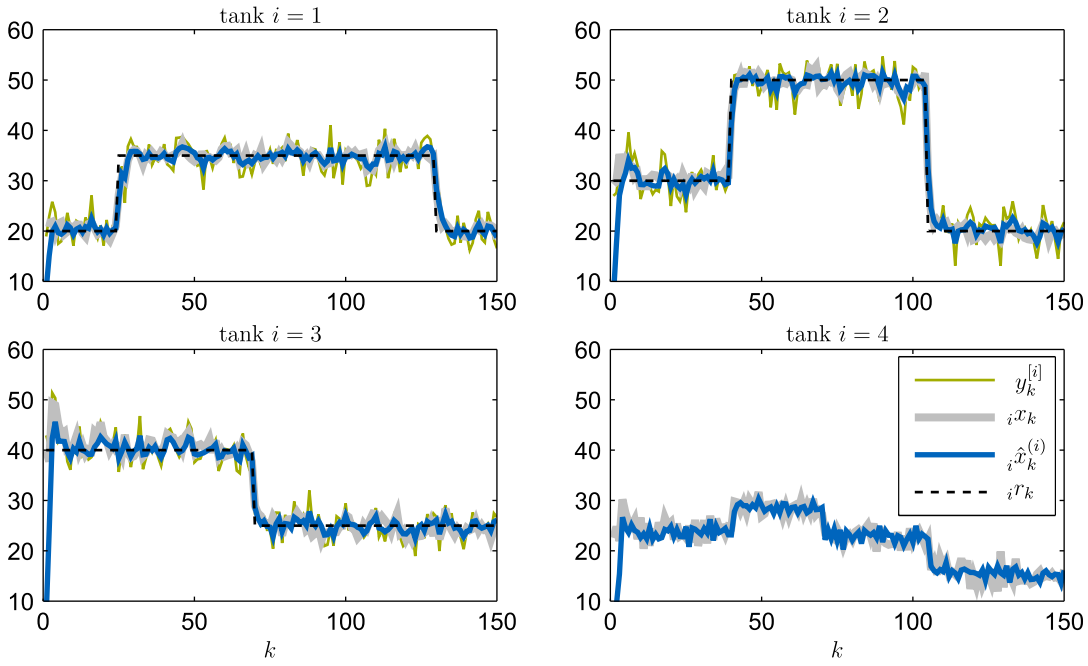


Figure 10.4: Comparison between the noisy measurements $y_k^{[i]}$, nominal trajectories ${}_i r_k$, true states ${}_i x_k$ and estimated states ${}_i \hat{x}_k^{(i)}$ generated by the DMHE (line communication topology $\mathcal{G}_1$) for the tanks $i = 1,2,3,4$.

Since optimality of the DMHE is only guaranteed for infinitely many iterations, we investigate a suboptimal approach where we prematurely stop the optimization after only 25 iterations per time step. Thereby, we use the relative error $d_k^{(i)}$ between the distributed estimates $\hat{\boldsymbol{x}}_k^{(i)}$ and the centralized estimate $\hat{\boldsymbol{x}}_k$ defined as

$$d_k^{(i)} = \frac{\|\hat{\boldsymbol{x}}_k^{(i)} - \hat{\boldsymbol{x}}_k\|}{\|\hat{\boldsymbol{x}}_k\|}, \quad i = 1,2,3, \tag{10.7}$$

to judge the performance of this approach. Figure 10.5 shows the relative errors for the scenario depicted in Figure 10.4. We can observe that the relative errors are below $7 \cdot 10^{-3}$ and the distributed estimators have almost reached a consensus after only 25 iterations per time step for the line communication topology $\mathcal{G}_1$.

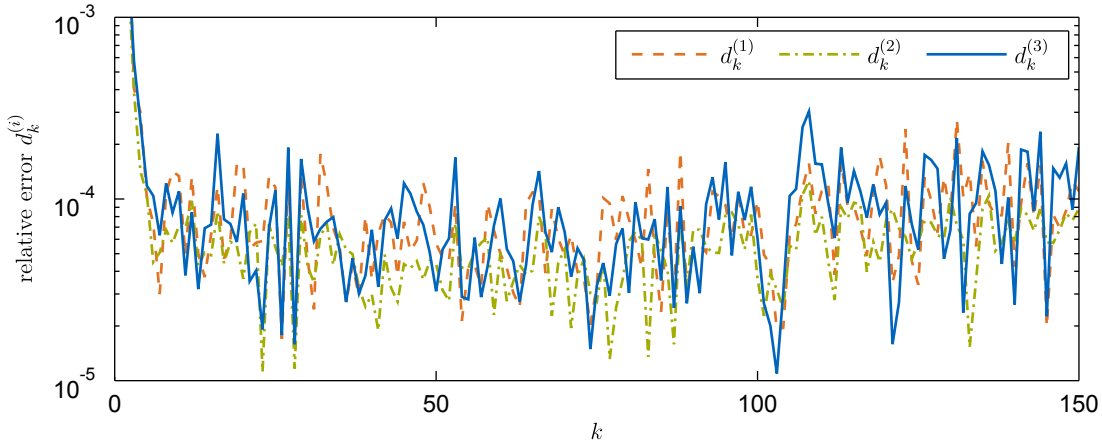


Figure 10.5: Relative errors $d_k^{(i)}$, $i = 1, 2, 3$, of the three observers for a line communication topology $\mathcal{G}_1$ and 25 iterations per time step.

Figure 10.6 reveals the influence of the communication topology on the duality gap $p_0^* - d_0^*[l]$. Thereby, the parameter a of the step size rule $\gamma_k[l] = a/\sqrt{l}$ is determined for each communication topology individually such that a maximum reduction in each iteration step is achieved while the assumptions of Theorem 9.4.1 are satisfied and thus stability is maintained. This results in the maximum values $a = 0.35$, $a = 0.2$ and $a = 0.12$ for the line, ring and all-to-all communication topology, respectively. We can observe that a higher connectivity results in an increased reduction of the duality gap. This confirms the intuitive expectation that a higher connectivity of the estimators results in an increased convergence rate.

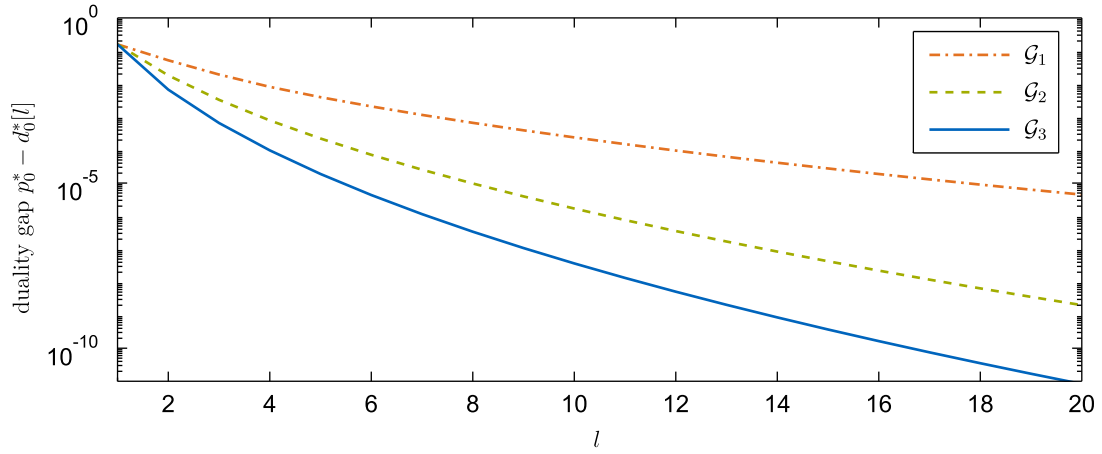


Figure 10.6: Duality gap $p_0^* - d_0^*[l]$ for different communication topologies in combination with the respective maximum admissible step size parameter a of the step size rule $\gamma_k[l] = a/\sqrt{l}$.

10.2.3 Comparison between the Distributed Moving Horizon Strategies

For the comparison between the DMHO and the DMHE, we consider the open-loop scenario with the state disturbance covariance $\boldsymbol{Q} = \mathrm{diag}(1, 1, 1, 1)$, the measurement disturbance variances $R^{[j]} = 2.5$, $j = 1, 2, 3$, and the reference input $\boldsymbol{r}_i$ according to the one depicted in

Figure 10.3 and 10.4. Since the DMHO is incapable of dealing with the local measurement allocation, we utilize the periodic measurement allocation for both DMHS. Moreover, both DMHS are interconnected by the ring communication topology $\mathcal{G}_2$ and utilize the step size $\gamma_k[l] = a/\sqrt{l}$ with the maximum admissible values $a = 0.03$ for the DMHO and $a = 0.25$ for the DMHE. The covariance $\boldsymbol{P}_{k-N}$ of the DMHE is updated according to the Riccati equation (9.10) with the initial value $\boldsymbol{P}_0 = \mathrm{diag}(1, 1, 1, 1)$.

For the sake of comparison, we consider the performance index given by the root mean square error (RMSE)

$$RMSE[l] \triangleq \sqrt{\sum_{i=1}^{3} \sum_{k=t_1}^{t_2} \frac{\|\hat{\boldsymbol{x}}_k^{(i)}[l] - \boldsymbol{x}_k\|^2}{3(t_2 - t_1)}}, \tag{10.8}$$

where $\hat{\boldsymbol{x}}_k^{(i)}[l]$ is the estimate derived by the i-th observer/estimator when prematurely stopping the optimization after l iterations, t_1 is the time where the moving horizon is filled, i. e. $t_1 = N = 5$ and t_2 is the time where the simulation is stopped, i. e. $t_2 = 150$. Note that for infinitely many iterations, the RMSE values of the DMHS coincide with those of the respective CMHS. Moreover, we consider the performance index (PI)

$$PI[l] \triangleq \lim_{j \to \infty} \frac{RMSE[1] - RMSE[l]}{RMSE[1] - RMSE[j]}, \tag{10.9}$$

which indicates the performance of a DMHS compared to the respective centralized counterpart on a scale between 0 and 1. A value of 1 means that the performance of the DMHS is identical to the one of the respective CMHS.

The $RMSE$ and corresponding PI values of the DMHO and DMHE for different number of iterations per step are depicted in Figure 10.7. Thereby, the state and measurement disturbance sequences are identical for all simulations. Based on these values, we can conclude the following. First, the DMHE is superior to the DMHO but requires a higher computational

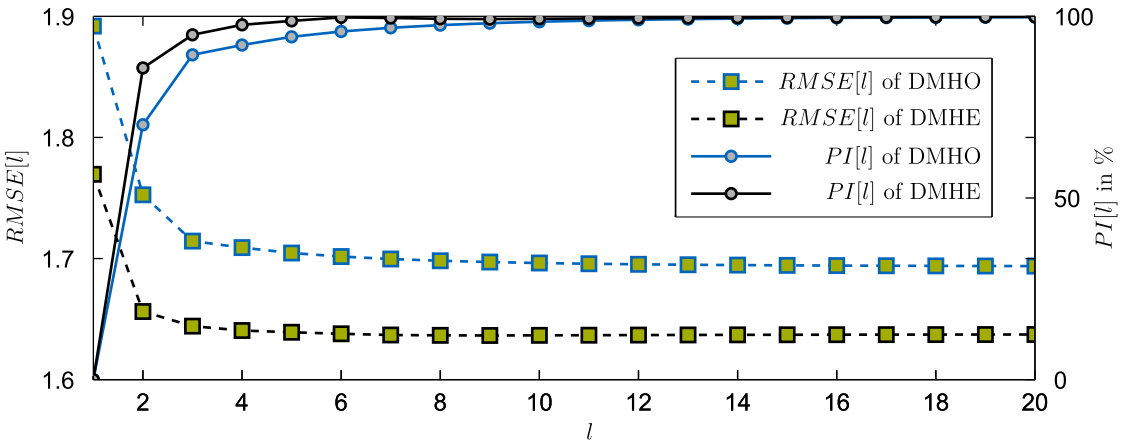


Figure 10.7: $RMSE[l]$ and corresponding $PI[l]$ values for a ring communication topology $\mathcal{G}_2$ among the DMHO and DMHE.

load for the given setup. However, the DMHE can utilize the local measurement allocation which results in a significantly reduced computational load compared to the DMHO. Second, the performance of the DMHO and the DMHE improves with increased number of iterations. Third, it is sufficient to perform only 12 and 6 iterations per time step for the DMHO and DMHE, respectively, because in these cases, the respective $RMSE$ values are almost identical to the centralized case and the PI values are above 99%.

10.3 Summary

In this chapter, we have presented simulation results for the developed distributed moving horizon strategies. We have analyzed the performance for each of these two methods including the special case of a distributed Kalman filter by evaluating the simulation results of a networked four tank system for various settings. These results confirm the high performance of all strategies whereby the CMHE has been slightly superior to the CMHO at the price of requiring a higher computational load for identical moving horizon lengths. Since optimality of all DMHS is only guaranteed for infinitely many iterations if the communication topology is not described by a complete directed graph, we have proposed and investigated a suboptimal approach where we prematurely stop the optimization after a certain number of iterations. Thereby, we have made two interesting observations. First, a higher connectivity among the observers/estimators results in an increased convergence rate. Second, the overall performance of the suboptimal approach reaches for only view iterations almost the performance of the optimal one.

Chapter 11

Conclusions and Future Work

In Part III of this thesis, we have presented the *distributed moving horizon observer* (DMHO) and the *distributed moving horizon estimator* (DMHE) for state estimation within the undisturbed and disturbed distributed NCS architecture, respectively. Both strategies provide distributed knowledge about the full state of the system which facilitates the decentralized implementation of any centralized controller. Furthermore, we have devised a method for the allocation of the measurements to the distributed observers. We have also investigated the impact of the interconnection topology among the distributed observers/estimators within the stability analysis of the DMHO as well as the DMHE. Finally, we have validated the performance of both strategies in simulations of a networked four tank system.

In this chapter, we summarize each chapter of Part III of this thesis. In addition, we also outline possible and natural extensions, as well as broader ideas for future work.

11.1 Summary

Chapter 9: Distributed Moving Horizon Strategies

In this chapter, we have presented the DMHO and the DMHE within a common framework for the undisturbed and disturbed distributed NCS architecture, respectively. The main feature of this architecture is the possibility of implementing any centralized designed controller fully decentralized. This is enabled by both DMHS which provide distributed knowledge about the full state of the system. The key step to derive the DMHS is the extension of the CMHS optimization problems by additional suitably designed consensus constraints which reflect the communication topology among the observers/estimators by means of a directed graph. This reformulation allows us to apply the dual decomposition technique to derive a decoupled dual optimization problem which we have solved by a suitable subgradient method. The resulting algorithm possesses a simple structure and alternates between updating and transmitting the primal and dual variables. Convergence of this algorithm and therefore stability of the DMHS is assured for several step sizes if the following three conditions are fulfilled. First, the system has to be observable. Second, the directed graph, which represents the communication topology among the observers/estimators, has to be weakly connected. Third, the allocation of

the measurements to the observers of the DMHO has to imply full-rankness of a certain matrix which is assured by a periodic measurement allocation provided that the system satisfies a certain form of observability. As a consequence, the DMHO requires a certain minimum moving horizon length while the DMHE can operate with measurements from one time step only. In this case, the DMHE becomes a distributed Kalman filter. Finally, we have sketched the extension of the DMHS to the cases of time-variant system and measurement models, state and disturbance constraints, and non-ideal networks.

Chapter 10: Simulation Results

In this chapter, we have presented simulation results for the developed distributed moving horizon strategies. We have analyzed the performance for each of these two methods including the special case of a distributed Kalman filter by evaluating the simulation results of a networked four tank system for various settings. These results confirm the high performance of all strategies whereby the CMHE has been slightly superior to the CMHO at the price of requiring a higher computational load for identical moving horizon lengths. Since optimality of all DMHS is only guaranteed for infinitely many iterations if the communication topology is not described by a complete directed graph, we have proposed and investigated a suboptimal approach where we prematurely stop the optimization after a certain number of iterations. Thereby, we have made two interesting observations. First, a higher connectivity among the observers/estimators results in an increased convergence rate. Second, the overall performance of the suboptimal approach reaches for only view iterations almost the performance of the optimal one.

11.2 Future Work

The following recommendations for future work are proposed to enhance the DMHS presented in Part III of this thesis:

- *Stability analysis for the suboptimal approach* - In general, optimality for the DMHS is only guaranteed for infinitely many iterations. Therefore, we have investigated in Chapter 10 a suboptimal approach, where we prematurely stop the optimization after a fixed number of iterations per time step. This approach has shown for the considered scenario almost the identical performance as the optimal one. However, the theoretical analysis of this procedure remains open. In this context, one might also contemplate a warm start strategy for the dual variables along with a decreasing condition for the cost functions similar to the one of the CMHS. In this way, it might be possible to design a suboptimal approach which converges to the optimal solution.
- *Speed of convergence analysis* - The stability analysis presented in Chapter 9 answers the question when the DMHS converge to the optimal solution. However, this analysis does not give any answer about how fast this happens. The simulation results of Chapter 10 suggest that the higher the connectivity among the distributed observers/estimators are,

the higher is the convergence speed. This expectation is reasonable because we know from consensus algorithms that their speed of convergence is quantified by the algebraic connectivity of the communication graph which is the second smallest eigenvalue of the graph Laplacian (Olfati-Saber et al., 2007). Since the DMHS algorithm looks similar to a consensus algorithm, it is not unreasonable to hope to derive similar results for the DMHS by applying the analysis techniques of the consensus algorithms. In this context, it is desirable to develop methods which replaces the current trial and error approach for choosing the maximal allowable step size parameter for a given communication topology by a systematic procedure.

- *Robustness enhancement* - Throughout the derivation of the DMHS, we have assumed that the networks $\Sigma_{\mathrm{N}}^{\{1\}}$ and $\Sigma_{\mathrm{N}}^{\{2\}}$ are ideal. However, this assumption might not be adequate to model all real-world circumstances. Therefore, we have presented in Section 9.6.3 the extension to an imperfect network $\Sigma_{\mathrm{N}}^{\{1\}}$. However, a similar extension for the network $\Sigma_{\mathrm{N}}^{\{2\}}$, which describes the communication topology among the distributed observers/estimators, remains open. Reasonable extensions should consider time-varying communication topologies, packet delays, packet loss and asynchronous operation and communication. A first port of call in this context might be the work of Gatsis and Giannakis (2012) which investigates asynchronous subgradient methods with unbounded delays.

- *Nonlinear systems and unsynchronized sensor clocks* - Finally, it would be desirable to derive distributed versions of the full CMHS including nonlinear systems and unsynchronized sensor clocks. While this does not constitute a practical problem within the presented framework, it raises the question whether or not this approach is reasonable. In both cases, the main challenge is that the primal problem becomes non-convex and thus, in general, strong duality does not hold any longer. Therefore, the duality gap should be investigated if it enables us to draw any conclusions regarding the quality of the primal solutions calculated based on the optimal dual ones. Another approach might be to restrict ourselves to a certain class of nonlinear systems for which it is possible to find a reformulation of the optimal estimation problem such that strong duality holds.

Appendix A

Background Material

For the sake of completeness, we state in this appendix several Definitions, Lemmas and Theorems which are of central importance for this thesis.

Definiteness of Functions

Definition A.1 (Slotine and Li, 1991)**.** *A scalar continuous function* $V(\boldsymbol{x}) : \mathcal{D} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ *is said to be*

- *locally positive semi-definite around* $\boldsymbol{x}^* \in \mathcal{D}$ *if* $V(\boldsymbol{x}^*) = 0$ *and* $V(\boldsymbol{x}) \geq 0$, $\boldsymbol{x} \in \{\mathcal{D} | \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq r, \boldsymbol{x} \neq \boldsymbol{x}^*\}$.
- *locally positive definite around* $\boldsymbol{x}^* \in \mathcal{D}$ *if* $V(\boldsymbol{x}^*) = 0$ *and* $V(\boldsymbol{x}) > 0$, $\boldsymbol{x} \in \{\mathcal{D} | \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq r, \boldsymbol{x} \neq \boldsymbol{x}^*\}$.
- *locally negative semi-definite around* $\boldsymbol{x}^* \in \mathcal{D}$ *if* $-V(\boldsymbol{x})$ *is positive semi-definite around* $\boldsymbol{x}^* \in \mathcal{D}$.
- *locally negative definite around* $\boldsymbol{x}^* \in \mathcal{D}$ *if* $-V(\boldsymbol{x})$ *is negative definite around* $\boldsymbol{x}^* \in \mathcal{D}$.

If the above properties hold $\forall \boldsymbol{x} \in \mathcal{D}$*, then the above statements are valid globally.*

Stability of Autonomous Systems

Theorem A.1 (Slotine and Li, 1991; Khalil, 2002)**.** *Consider the autonomous system*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{A.1}$$

where $\boldsymbol{f} : \mathcal{D} \mapsto \mathbb{R}^n$ *is a locally Lipschitz map from a domain* $\mathcal{D} \subset \mathbb{R}^n$ *into* $\mathbb{R}^n$*. Let* $\boldsymbol{x}^* \in \mathcal{D}$ *be an equilibrium point of* (A.1) *and let* $V : \mathcal{D} \mapsto \mathbb{R}$ *be a continuously differentiable function such that*

i) $V(\boldsymbol{x}^*) = 0$ *and* $V(\boldsymbol{x}) > 0$, $\boldsymbol{x} \in \mathcal{D} \backslash \{\boldsymbol{x}^*\}$

ii) $\dot{V}(\boldsymbol{x}) \leq 0$, $\boldsymbol{x} \in \mathcal{D}$.

Then $\boldsymbol{x}^*$ *is stable. Moreover, if*

iii) $\dot{V}(\boldsymbol{x}) < 0$, $\boldsymbol{x} \in \mathcal{D} \backslash \{\boldsymbol{x}^*\}$,

then $\boldsymbol{x}^*$ *is asymptotically stable.*

Mean Value Theorem

Theorem A.2 (Abraham et al., 2002, pg. 77)**.** *Let $\mathcal{E}$ and $\mathcal{F}$ be real Banach spaces, $\boldsymbol{f}(\boldsymbol{z}) : \mathcal{U} \subset \mathcal{E} \mapsto \mathcal{F}$ a continuously differentiable map, $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$, and $\boldsymbol{c}$ a continuously differentiable arc in $\mathcal{U}$ connecting $\boldsymbol{x}$ to $\boldsymbol{y}$, i. e., $\boldsymbol{c}$ is a continuous differentiable map $\boldsymbol{c} : [0, 1] \mapsto \mathcal{U}$, for which $\boldsymbol{c}(0) = \boldsymbol{x}$ and $\boldsymbol{c}(1) = \boldsymbol{y}$ holds. Then*

$$\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{f}(\boldsymbol{x}) = \int_0^1 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}(\boldsymbol{c}(t))\, \boldsymbol{c}'(t)\, dt.$$

If $\mathcal{U}$ is convex and $\boldsymbol{c}(t) = (1 - t)\boldsymbol{x} + t\boldsymbol{y}$, then

$$\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{f}(\boldsymbol{x}) = \int_0^1 \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{z}}((1 - t)\boldsymbol{x} + t\boldsymbol{y})\, dt\, (\boldsymbol{y} - \boldsymbol{x}).$$

Gronwall-Bellman Inequality

Lemma A.3 (Khalil, 2002, pg. 651)**.** *Let $\lambda : [a, b] \mapsto \mathbb{R}$ be continuous and $\mu : [a, b] \mapsto \mathbb{R}$ be continuous and non-negative. If a continuous function $y : [a, b] \mapsto \mathbb{R}$ satisfies*

$$y(t) \leq \lambda(t) + \int_a^t \mu(s)y(s)ds$$

for $a \leq t \leq b$, then on the same interval

$$y(t) \leq \lambda(t) + \int_a^t \lambda(s)\mu(s) \exp\left(\int_s^t \mu(\tau)d\tau\right) ds.$$

In particular, if $\lambda(t) \triangleq \lambda$ is a constant, then

$$y(t) \leq \lambda \exp\left(\int_a^t \mu(\tau)d\tau\right).$$

If, in addition, $\mu(t) \triangleq \mu \geq 0$ is a constant, then

$$y(t) \leq \lambda \exp\left(\mu(t - a)\right).$$

Continuous Dependence on Initial Conditions, Times and Parameters

The following Theorem is derived by extending Theorem 3.4 given in Khalil (2002, pg. 95) by additionally considering the continuous dependence on initial times. The proof heavily relies on the Gronwall-Bellman Inequality stated in Lemma A.3.

Theorem A.4. *Let $\boldsymbol{f}(t, \boldsymbol{x})$ be piecewise continuous in t and Lipschitz in $\boldsymbol{x}$ on $[t_0, t_2] \times \mathcal{W}$ with a Lipschitz constant L, where $\mathcal{W} \subset \mathbb{R}^n$ is an open connected set. Let*

$$\|\boldsymbol{f}(t, \boldsymbol{x})\| \leq \delta, \quad \forall (t, \boldsymbol{x}) \in [t_0, t_2] \times \mathcal{W}$$

for some $\delta > 0$. Let $\boldsymbol{y}(t)$ and $\boldsymbol{z}(t)$ be solutions of

$$\begin{aligned} \dot{\boldsymbol{y}} &= \boldsymbol{f}(t, \boldsymbol{y}) + \boldsymbol{g}(t), & \boldsymbol{y}(t_1) &= \boldsymbol{y}_0 \\ \dot{\boldsymbol{z}} &= \boldsymbol{f}(t, \boldsymbol{z}) + \boldsymbol{h}(t), & \boldsymbol{z}(t_0) &= \boldsymbol{z}_0 \end{aligned}$$

with $t_1 \in [t_0, t_2]$, such that $\boldsymbol{y}(t), \boldsymbol{z}(t) \in \mathcal{W}$ for all $t \in [t_0, t_2]$. Suppose that

$$\begin{aligned} \|\boldsymbol{h}(t)\| &\leq \mu, \quad \forall t \in [t_0, t_2] \\ \|\boldsymbol{g}(t) - \boldsymbol{h}(t)\| &\leq \rho, \quad \forall t \in [t_0, t_2] \end{aligned}$$

for some $\mu, \rho > 0$. Then

$$\|\boldsymbol{y}(t) - \boldsymbol{z}(t)\| \leq \|\boldsymbol{y}_0 - \boldsymbol{z}_0\| e^{L(t-t_0)} + |t_1 - t_0|(\delta + \mu) e^{L(t-t_0)} + \frac{\rho}{L}\left(e^{L(t-t_0)} - 1\right)$$

$\forall t \in [t_0, t_2]$.

Proof. The solutions $\boldsymbol{y}(t)$ and $\boldsymbol{z}(t)$ are given by

$$\begin{aligned} \boldsymbol{y}(t) &= \boldsymbol{y}_0 + \int_{t_0}^{t} \boldsymbol{f}(s, \boldsymbol{y}(s)) + \boldsymbol{g}(s)\, ds \\ \boldsymbol{z}(t) &= \boldsymbol{z}_0 + \int_{t_0}^{t_1} \boldsymbol{f}(s, \boldsymbol{z}(s)) + \boldsymbol{h}(s)\, ds + \int_{t_1}^{t} \boldsymbol{f}(s, \boldsymbol{z}(s)) + \boldsymbol{h}(s)\, ds. \end{aligned}$$

Subtracting the two equations and taking norms yield

$$\begin{aligned} \|\boldsymbol{y}(t) - \boldsymbol{z}(t)\| &\leq \|\boldsymbol{y}_0 - \boldsymbol{z}_0\| + \int_{t_0}^{t_1} \|\boldsymbol{f}(s, \boldsymbol{z}(s))\| + \|\boldsymbol{h}(s)\|\, ds \\ &\quad + \int_{t_1}^{t} \|\boldsymbol{f}(s, \boldsymbol{y}(s)) - \boldsymbol{f}(s, \boldsymbol{z}(s))\|\, ds + \int_{t_1}^{t} \|\boldsymbol{g}(s) - \boldsymbol{h}(s)\|\, ds \\ &\leq \gamma + (t - t_1)\rho + \int_{t_1}^{t} L\|\boldsymbol{y}(s) - \boldsymbol{z}(s)\| ds \end{aligned}$$

where $\gamma = \|\boldsymbol{y}_0 - \boldsymbol{z}_0\| + |t_1 - t_0|(\delta + \mu)$. Application of the Gronwall-Bellman inequality stated in Lemma A.3 to the function $\|\boldsymbol{y}(t) - \boldsymbol{z}(t)\|$ results in

$$\|\boldsymbol{y}(t) - \boldsymbol{z}(t)\| \leq \gamma + (t - t_1)\rho + \int_{t_1}^{t} (\gamma + (t - t_1)\rho) L e^{L(t-s)}\, ds.$$

Integrating the right-hand side by parts, we obtain

$$\begin{aligned} \|\boldsymbol{y}(t) - \boldsymbol{z}(t)\| &\leq \gamma + (t - t_1)\rho - \gamma - (t - t_1)\rho + \gamma e^{L(t-t_1)} + \int_{t_1}^{t} \rho e^{L(t-s)}\, ds. \\ &= \|\boldsymbol{y}_0 - \boldsymbol{z}_0\| e^{L(t-t_0)} + |t_1 - t_0|(\delta + \mu) e^{L(t-t_0)} + \frac{\rho}{L}\left(e^{L(t-t_0)} - 1\right) \end{aligned}$$

which completes the proof. □

Leibniz-Integral Rule

Lemma A.5. *Let $\alpha(p)$ and $\beta(p)$ be $\mathcal{C}^1$-functions. Suppose that both $\boldsymbol{f}(p,t)$ and $\partial \boldsymbol{f}(p,t)/\partial p$ are continuous in the variables p and t. Then $\int_{\alpha(p)}^{\beta(p)} \boldsymbol{f}(p,t)\,dt$ exists as a continuously differentiable function of p, with derivative*

$$\frac{\partial}{\partial p}\int_{\alpha(p)}^{\beta(p)} \boldsymbol{f}(p,t)\,dt = \int_{\alpha(p)}^{\beta(p)} \frac{\partial \boldsymbol{f}(p,t)}{\partial p}\,dt + \frac{\partial \beta(p)}{\partial p}\boldsymbol{f}(p,\beta(p)) - \frac{\partial \alpha(p)}{\partial p}\boldsymbol{f}(p,\alpha(p)).$$

Block matrices

Lemma A.6 (Kailath, 1980; Petersen and Pedersen, 2012)**.** *Given a block matrix*

$$\begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix}, \quad \text{with } \boldsymbol{A}_{11} \in \mathbb{R}^{n\times n}, \boldsymbol{A}_{12} \in \mathbb{R}^{n\times m}, \boldsymbol{A}_{21} \in \mathbb{R}^{m\times n} \text{and } \boldsymbol{A}_{22} \in \mathbb{R}^{m\times m}.$$

Then the determinant of this block matrix is

$$\begin{aligned} \det \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} &= \det(\boldsymbol{A}_{11})\det(\boldsymbol{C}_2), \quad \text{if } \boldsymbol{A}_{11} \text{ is invertible} \\ &= \det(\boldsymbol{A}_{22})\det(\boldsymbol{C}_1), \quad \text{if } \boldsymbol{A}_{22} \text{ is invertible,} \end{aligned}$$

with

$$\begin{aligned} \boldsymbol{C}_1 &\triangleq \boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}, \quad \text{if } \boldsymbol{A}_{22} \text{ is invertible} \\ \boldsymbol{C}_2 &\triangleq \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}, \quad \text{if } \boldsymbol{A}_{11} \text{ is invertible.} \end{aligned}$$

Appendix B

Derivatives of a Specific Function

At several places in this thesis, we need the first-order and second-order derivative of the scalar function $\boldsymbol{c}^T \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{b}))$ with respect to $\boldsymbol{b}$ which are stated in the following Lemma. Note that this Lemma contains the first-order and second-order derivatives of the scalar function $f(\boldsymbol{x}(\boldsymbol{b}))$ with respect to $\boldsymbol{b}$ as a special case by setting ${}_ic = 1/k$ and ${}_if = f$.

Lemma B.1. *Let the functions* $\boldsymbol{f}(\boldsymbol{x}) = [{}_1f(\boldsymbol{x}), \ldots, {}_kf(\boldsymbol{x})]^T$ *and* $\boldsymbol{x}(\boldsymbol{b}) = [{}_1x(\boldsymbol{b}), \ldots, {}_nx(\boldsymbol{b})]^T$ *be at least twice continuously differentiable mappings* $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^k$ *and* $\boldsymbol{x} : \mathbb{R}^m \mapsto \mathbb{R}^n$, *respectively. Moreover, let* $\boldsymbol{c} = [{}_1c, \ldots, {}_kc]^T$ *be a vector with* $\boldsymbol{c} \in \mathbb{R}^k$. *Then the first-order derivative of the scalar function* $\boldsymbol{c}^T \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{b}))$ *with respect to* $\boldsymbol{b}$ *is*

$$\frac{\partial}{\partial \boldsymbol{b}} \left(\boldsymbol{c}^T \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{b})\right) \right) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^T \boldsymbol{c} \tag{B.1a}$$

and the second-order derivative with respect to $\boldsymbol{b}$ *is*

$$\frac{\partial^2}{\partial \boldsymbol{b}^2} \left(\boldsymbol{c}^T \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{b})\right) \right) = \sum_{i=1}^{k} {}_ic \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial^2 {}_if}{\partial \boldsymbol{x}^2} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}} + \sum_{j=1}^{n} \left(\boldsymbol{c}^T \frac{\partial \boldsymbol{f}}{\partial {}_jx} \right) \frac{\partial^2 {}_jx}{\partial \boldsymbol{b}^2}. \tag{B.1b}$$

Proof. Recalling the notation of the gradient as a column vector (see Notations), the first-order derivative (B.1a) is derived by

$$\frac{\partial}{\partial \boldsymbol{b}} \left(\mathbf{c}^T \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{b})\right) \right) = \sum_{i=1}^{k} {}_ic \frac{\partial {}_if}{\partial \boldsymbol{b}} = \sum_{i=1}^{k} {}_ic \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial {}_if}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \sum_{i=1}^{k} {}_ic \frac{\partial {}_if}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^T \mathbf{c}.$$

Equation (B.1b) is expressed as

$$\frac{\partial^2}{\partial \boldsymbol{b}^2} \left(\mathbf{c}^T \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{b})\right) \right) = \sum_{i=1}^{k} {}_ic \frac{\partial^2 {}_if}{\partial \boldsymbol{b}^2},$$

where the Hessian of ${}_if$ is

$$\frac{\partial^2 {}_if}{\partial \boldsymbol{b}^2} = \begin{bmatrix} \frac{\partial^2 {}_if}{\partial {}_1b \partial {}_1b} & \cdots & \frac{\partial^2 {}_if}{\partial {}_1b \partial {}_mb} \\ \vdots & & \vdots \\ \frac{\partial^2 {}_if}{\partial {}_mb \partial {}_1b} & \cdots & \frac{\partial^2 {}_if}{\partial {}_mb \partial {}_mb} \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} \frac{\partial}{\partial_1 b}\left(\frac{\partial_i f}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{x}}{\partial_1 b}\right) & \cdots & \frac{\partial}{\partial_1 b}\left(\frac{\partial_i f}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{x}}{\partial_m b}\right) \\ \vdots & & \vdots \\ \frac{\partial}{\partial_m b}\left(\frac{\partial_i f}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{x}}{\partial_1 b}\right) & \cdots & \frac{\partial}{\partial_m b}\left(\frac{\partial_i f}{\partial \boldsymbol{x}}^T \frac{\partial \boldsymbol{x}}{\partial_m b}\right) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial_1 b}\sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial_j x}{\partial_1 b} & \cdots & \frac{\partial}{\partial_1 b}\sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial_j x}{\partial_m b} \\ \vdots & & \vdots \\ \frac{\partial}{\partial_m b}\sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial_j x}{\partial_1 b} & \cdots & \frac{\partial}{\partial_m b}\sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial_j x}{\partial_m b} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^n \frac{\partial \boldsymbol{x}}{\partial_1 b}^T \frac{\partial_i f^2}{\partial_j x \partial \boldsymbol{x}}\frac{\partial_j x}{\partial_1 b} & \cdots & \sum_{j=1}^n \frac{\partial \boldsymbol{x}}{\partial_1 b}^T \frac{\partial_i f^2}{\partial_j x \partial \boldsymbol{x}}\frac{\partial_j x}{\partial_m b} \\ \vdots & & \vdots \\ \sum_{j=1}^n \frac{\partial \boldsymbol{x}}{\partial_m b}^T \frac{\partial_i f^2}{\partial_j x \partial \boldsymbol{x}}\frac{\partial_j x}{\partial_1 b} & \cdots & \sum_{j=1}^n \frac{\partial \boldsymbol{x}}{\partial_m b}^T \frac{\partial_i f^2}{\partial_j x \partial \boldsymbol{x}}\frac{\partial_j x}{\partial_m b} \end{bmatrix} \\
&\quad + \begin{bmatrix} \sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial_1 b \partial_1 b} & \cdots & \sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial_1 b \partial_m b} \\ \vdots & & \vdots \\ \sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial_m b \partial_1 b} & \cdots & \sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial_m b \partial_m b} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial \boldsymbol{x}}{\partial_1 b}^T \frac{\partial_i f^2}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial_1 b} & \cdots & \frac{\partial \boldsymbol{x}}{\partial_1 b}^T \frac{\partial_i f^2}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial_m b} \\ \vdots & & \vdots \\ \frac{\partial \boldsymbol{x}}{\partial_m b}^T \frac{\partial_i f^2}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial_1 b} & \cdots & \frac{\partial \boldsymbol{x}}{\partial_m b}^T \frac{\partial_i f^2}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial_m b} \end{bmatrix} + \sum_{j=1}^n \frac{\partial_i f}{\partial_j x} \begin{bmatrix} \frac{\partial^2{}_j x}{\partial_1 b \partial_1 b} & \cdots & \frac{\partial^2{}_j x}{\partial_1 b \partial_m b} \\ \vdots & & \vdots \\ \frac{\partial^2{}_j x}{\partial_m b \partial_1 b} & \cdots & \frac{\partial^2{}_j x}{\partial_m b \partial_m b} \end{bmatrix} \\
&= \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial^2{}_i f}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}} + \sum_{j=1}^n \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial \boldsymbol{b}^2}.
\end{aligned}
$$

Finally, the desired derivative is

$$
\begin{aligned}
\frac{\partial^2}{\partial \boldsymbol{b}^2}\, \mathbf{c}^T \boldsymbol{f}\left(\boldsymbol{x}(\boldsymbol{b})\right) &= \sum_{i=1}^k \left({}_i c \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial^2{}_i f}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}} + \sum_{j=1}^n {}_i c \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial \boldsymbol{b}^2} \right) \\
&= \sum_{i=1}^k {}_i c \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial^2{}_i f}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}} + \sum_{j=1}^n \sum_{i=1}^k {}_i c \frac{\partial_i f}{\partial_j x}\frac{\partial^2{}_j x}{\partial \boldsymbol{b}^2} \\
&= \sum_{i=1}^k {}_i c \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}}^T \frac{\partial^2{}_i f}{\partial \boldsymbol{x}^2}\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{b}} + \sum_{j=1}^n \left(\mathbf{c}^T \frac{\partial \boldsymbol{f}}{\partial_j x} \right) \frac{\partial^2{}_j x}{\partial \boldsymbol{b}^2},
\end{aligned}
$$

which concludes the proof. □

Appendix C

Continuous-Discrete Extended Kalman Filter

Consider the continuous-time nonlinear system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{w}(t), \tag{C.1a}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ is the state, $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ is the control input and $\boldsymbol{w}(t) \in \mathbb{R}^{n_x}$ is the zero mean state disturbance with covariance $\boldsymbol{Q}(t)$, i. e. $\boldsymbol{w}(t) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(t))$. The mean and the covariance of the initial state at the initial time t_0 are assumed to be $\hat{\boldsymbol{x}}_{0|0}$ and $\boldsymbol{P}_{0|0}$, respectively. The state vector is observed through the discrete-time measurement equation

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}(t_k)) + \boldsymbol{v}_k, \qquad k = 1, 2, \ldots, \tag{C.1b}$$

where $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ is the measurement and $\boldsymbol{v}_k \in \mathbb{R}^{n_y}$ is the zero mean measurement disturbance with covariance $\boldsymbol{R}_k$, i. e. $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$.

The continuous-discrete extended Kalman filter (CDEKF) alternates between a prediction step and an update step which may be stated as follows, see Gelb (1974).

Prediction Step

The state and covariance predictions are computed as the solutions to the system of differential equations

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t)), \tag{C.2a}$$

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{F}(t)\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{F}(t)^T + \boldsymbol{Q}(t), \tag{C.2b}$$

in which

$$\boldsymbol{F}(t) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}(t)=\hat{\boldsymbol{x}}(t)}, \tag{C.2c}$$

with the initial conditions $\hat{\boldsymbol{x}}(t_{k-1}) = \hat{\boldsymbol{x}}_{k-1|k-1}$ and $\boldsymbol{P}(t_{k-1}) = \boldsymbol{P}_{k-1|k-1}$. The one-step ahead predictions required for the subsequent update step are for the state $\hat{\boldsymbol{x}}_{k|k-1} = \hat{\boldsymbol{x}}(t_k)$ and for

the covariance $\boldsymbol{P}_{k|k-1} = \boldsymbol{P}(t_k)$.

Update Step

Utilizing the gain matrix

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k\right)^{-1}, \tag{C.3a}$$

the filtered state $\hat{\boldsymbol{x}}_{k|k}$ and its covariance $\boldsymbol{P}_{k|k}$ are calculated by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1})\right), \tag{C.3b}$$

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{P}_{k|k-1}, \tag{C.3c}$$

where

$$\boldsymbol{H}_k = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}(t_k)=\hat{\boldsymbol{x}}_{k|k-1}}. \tag{C.3d}$$

List of Figures

List of Tables

List of Algorithms

Bibliography

R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. Springer, 3rd edition, 2002.

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

M. Alamir. Optimization based non-linear observers revisited. *International Journal of Control*, 72(13):1204–1217, 1999.

M. Alamir and L. A. Calvillo-Corona. Further results on nonlinear receding-horizon observers. *IEEE Transactions on Automatic Control*, 47(7):1184–1188, 2002.

A. Alessandri, M. Baglietto, and G. Battistelli. Receding-horizon estimation for discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48(3):473–478, 2003.

A. Alessandri, M. Baglietto, and G. Battistelli. Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44 (7):1753–1765, 2008.

P. Antsaklis and J. Baillieul. Guest editorial special issue on networked control systems. *IEEE Transactions on Automatic Control*, 49(9):1421–1423, 2004.

P. Antsaklis and J. Baillieul. Special issue on technology of networked control systems. *Proceedings of the IEEE*, 95(1):5–8, 2007.

U. M. Ascher, J. Christiansen, and R. D. Russell. A collocation solver for mixed order systems of boundary value problems. *Mathematics of Computation*, 33(146):659–679, 1979.

U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Classics in Applied Mathematics 13. Society for Industrial and Applied Mathematics, 1995.

L. Bakule. Decentralized control: An overview. *Annual Reviews in Control*, 32(1):87–98, 2008.

R. Bellman. *Dynamic Programming*. Princeton University Press, 1st edition, 1957.

A. Bemporad, M. Heemels, and M. Johansson. *Networked Control Systems*. Springer, 2010.

L. T. Berger and K. Iniewski. *Smart Grid: Applications, Communications, and Security*. Wiley, 2012.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

G. Besançon, editor. *Nonlinear Observers and Application*. Lecture Notes in Control and Information Sciences. Springer, 2007.

S. Biegacki and D. VanGompel. The application of devicenet in process control. *ISA Transactions*, 35(2):169–176, 1996.

T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, and O. Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 295–339. Springer, 2001.

C. H. Bischof, A. Carle, P. Khademi, and A. Mauer. ADIFOR 2.0: Automatic differentiation of Fortran 77 programs. *IEEE Computational Science & Engineering*, 3(3):18–32, 1996.

R. R. Bitmead, M. R. Gevers, I. R. Petersen, and R. J. Kaye. Monotonicity and stabilizability properties of solutions of the Riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples. *Systems & Control Letters*, 5(5):309–315, 1985.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

M. S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Mühlig. *Handbook of Mathematics*. Springer, 5th edition, 2007.

A. E. Bryson and M. Frazier. Smoothing for linear and nonlinear dynamic systems. In *Proceedings of the Optimum System Synthesis Conference*, pages 353–364, 1963.

C. Califano, S. Monaco, and D. Normand-Cyrot. On the observer design in discrete-time. *Systems & Control Letters*, 49(4):255–265, 2003.

Y. Cao, S. Li, and L. Petzold. Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. *Journal of Computational and Applied Mathematics*, 149(1):171–191, 2002.

F. L. Chernousko and A. A. Lyubushin. Method of successive approximations for solution of optimal control problems. *Optimal Control Applications and Methods*, 3(2):101–114, 1982.

M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1): 255–312, 2007.

B. Colson, P. Marcotte, and G. Savard. Bilevel programming: A survey. *4OR-A Quarterly Journal of Operations Research*, pages 87–107, 2005.

J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

H. Cox. On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on Automatic Control,*, 9(1):5–12, 1964.

R. Dai and M. Mesbahi. Optimal topology design for dynamic networks. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1280–1285, 2011.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–11, 1960.

C. de Souza, M. Gevers, and G. Goodwin. Riccati equations in optimal filtering of nonstabilizable systems having singular state transition matrices. *IEEE Transactions on Automatic Control*, 31(9):831–838, 1986.

V. DeMiguel and W. Murray. A local convergence analysis of bilevel decomposition algorithms. *Optimization and Engineering*, 7(2):99–133, 2006.

M. Diehl, H. J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In L. Magni, D. M. Raimondo, and F. Allgöwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 391–417. Springer, 2009.

R. Diestel. *Graph Theory*. Springer, 4th edition, 2012.

J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*, pages 147–163, Boston, 2002.

J. E. Elson. *Time Synchronization in Wireless Sensor Networks*. PhD thesis, University of California, Los Angeles, 2003.

M. Farina, G. Ferrari-Trecate, and R. Scattolini. Distributed moving horizon estimation for linear constrained systems. *IEEE Transactions on Automatic Control*, 55(11):2462–2475, 2010.

M. C. Ferris and O. L. Mangasarian. Parallel variable distribution. *SIAM Journal on Optimization*, 4:815–832, 1994.

H. Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627, 1973.

W. Fleming. *Functions of Several Variables*. Undergraduate Texts in Mathematics. Springer, 1977.

M. Fliess, J. Lévine, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61:1327–1361, 1995.

N. Gatsis and G. B. Giannakis. Asynchronous subgradient methods with unbounded delays for communication networks. In *Proceedings of the 51th IEEE Conference on Decision and Control*, pages 5870–5875, 2012.

A. Gelb. *Applied Optimal Estimation*. The MIT Press, 1974.

R. Giering and T. Kaminski. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software*, 24(4):437–474, 1998.

P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7272–7277, 2010.

C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

G. C. Goodwin, S. F. Graebe, and M. E. Salgado. *Control System Design*. Prentice Hall, 2001.

G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh. A moving horizon approach to networked control system design. *IEEE Transactions on Automatic Control*, 49(9):1427–1445, 2004.

R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Professional, 2nd edition, 1994.

K. Graichen, V. Hagenmeyer, and M. Zeitz. A new approach to inversion-based feedforward control design for nonlinear systems. *Automatica*, 41:2033–2041, 2005.

A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Other Titles in Applied Mathematics. SIAM, Philadelphia, PA, 2nd edition, 2008.

A. Griewank, D. Juedes, and J. Utke. Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, 22(2):131–167, 1996.

S. Hanba. On the "uniform" observability of discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 54(8):1925–1928, 2009.

S. Hanba. Further results on the uniform observability of discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 55(4):1034–1038, 2010.

R. F. Hartl, S. P. Sethi, and R. G. Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, 37(2):181–218, 1995.

E. L. Haseltine and J. B. Rawlings. Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44(8):2451–2460, 2005.

J. P. Hespanha, M. McLaughlin, G. S. Sukhatme, M. Akbarian, R. Garg, and W. Zhu. Haptic collaboration over the Internet. In *Proceedings of the Fifth Phantom Users Group Workshop*, pages 9–13, 2000.

J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, 2007.

G. A. Hicks and W. H. Ray. Approximation methods for optimal control synthesis. *The Canadian Journal of Chemical Engineering*, 49:552–529, 1971.

S. Hirche. *Haptic Telepresence in Packet Switched Communication Networks*. PhD thesis, Technische Universität München, 2005.

Y. C. Ho and D. L. Pepyne. Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570, 2002.

D. Hristu-Varsakelis and W. S. Levine. *Handbook of Networked and Embedded Control Systems*. Birkhäuser, 2005.

J. Humpherys and J. West. Kalman filtering with Newton's method. *IEEE Control Systems Magazine*, 30(6):101–106, 2010.

A. Isidori. *Nonlinear Control Systems*. Springer, 3rd edition, 1995.

ISO 7498. Osi basic reference model, ISO 7498. American National Standards Institute, 1984.

S. S. Jang, B. Joseph, and H. Mukai. Comparison of two approaches to on-line parameter and state estimation of nonlinear systems. *Industrial & Engineering Chemistry Process Design and Development*, 25(3):809–814, 1986.

A. Jazwinski. Limited memory optimal filtering. *IEEE Transactions on Automatic Control*, 13(5):558–563, 1968.

A. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, 1970.

L. Ji'an, C. Li, and J. Peng. Moving horizon state estimation for nonlinear systems using quantized data from wireless sensor networks. In *Proceedings of the 27th Chinese Control Conference*, pages 442–446, 2008.

Z. Jin, C.-K. Ko, and R. M. Murray. Estimation for nonlinear dynamical systems over packet-dropping networks. In *Proceedings of the American Control Conference*, pages 5037–5042, 2007.

T. A. Johansen, D. Sui, and R. Nybø. Regularized nonlinear moving-horizon observer with robustness to delayed and lost data. To appear in: IEEE Transactions on Control Systems Technology, 2013.

T. Kailath. *Linear Systems*. Prentice Hall, 1980.

I. Karafyllis and C. Kravaris. On the observer problem for discrete-time control systems. *IEEE Transactions on Automatic Control*, 52(1):12–25, 2007.

T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.

K. C. Kiwiel. Convergence of approximate and incremental subgradient methods for convex optimization. *SIAM Journal on Optimization*, 14(3):807–840, 2004.

G. Kreisselmeier. Adaptive observers with exponential rate of convergence. *IEEE Transactions on Automatic Control*, 22(1):2–8, 1977.

G. Kreisselmeier and R. Engel. Nonlinear observers for autonomous lipschitz continuous systems. *IEEE Transactions on Automatic Control*, 48(3):451–464, 2003.

A. J. Krener and W. Respondek. Nonlinear observers with linerizable error dynamics. *SIAM Journal on Control and Optimization*, 23(2):197–216, 1985.

M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos. *Nonlinear and Adaptive Control Design*. John Wiley and Sons, 1st edition, 1995.

H. J. Kushner. On the differential equations satisfied by conditional probabiltity densities of markov processes, with applications. *SIAM Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 2(1):106–119, 1964.

H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley Interscience, 1972.

V. J. Law and Y. Sharma. Computation of the gradient and sensitivity coefficients in sum of squares minimization problems with differential equation models. *Computers and Chemical Engineering*, 21(12):1471–1479, 1997.

F. Lian, J. R. Moyne, and D. M. Tilbury. Performance evaluation of control networks. *IEEE Control System Magazine*, pages 66–83, 2001.

M. J. Liebman, T. F. Edgar, and L. S. Lasdon. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers & Chemical Engineering*, 16(10–11):963–986, 1992.

W. Lin and C. I. Byrnes. Remarks on linearization of discrete-time autonomous systems and nonlinear observer design. *Systems & Control Letters*, 25(1):31–40, 1995.

A. Liu, L. Yu, and W.-A. Zhang. Moving horizon estimation for networked systems with packet dropouts. In *Proceedings of the 51th IEEE Conference on Decision and Control*, pages 763–768, 2012.

A. Liu, L. Yu, W.-A. Zhang, and M. Z. Q. Chen. Moving horizon estimation for networked systems with quantized measurements and packet dropouts. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(7):1823–1834, 2013.

L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 2nd edition, 1999.

J. T.-H. Lo. Synthetic approach to optimal filtering. *IEEE Transactions on Neural Networks*, 5(5):803–811, 1994.

G. Ludyk. *Theoretische Regelungstechnik 1*. Springer, 1995.

J. Lunze. *Feedback Control of Large Scale Systems*. Prentice Hall, 1992.

J. Lunze, editor. *Control Theory of Digitally Networked Dynamic Systems*. Springer, 2014.

J. M. Maestre, P. Giselsson, and A. Rantzer. Distributed receding horizon kalman filter. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 5068–5073, 2010.

M. Maroti, G. Simon, B. Kusy, and A. Ledeczi. The flooding time synchronization protocol. In *Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, pages 39–49, Baltimore, 2004.

C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian. Remote surgery case: robot-assisted teleneurosurgery. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 819–823, 2004.

H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006, 1995.

D. L. Mills. Internet time synchronisation: the network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.

P. E. Moraal and J. W. Grizzle. Observer design for nonlinear systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 40(3):395–404, 1995.

J. R. Moyne and D. M. Tilbury. The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proceedings of the IEEE*, 95(1):29–47, 2007.

K. R. Muske, J. B. Rawlings, and J. H. Lee. Receding horizon recursive state estimation. In *Proceedings of the American Control Conference*, pages 900–904, 1993.

H. M. Newman. Integrating building automation and control products using the bacnet protocol. *ASHRAE Journal*, 28(11):36–42, 1996.

H. Nijmeijer and A. J. Van Der Schaft. *Nonlinear Dynamical Control Systems*. Springer, 1990.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2nd edition, 2006.

P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

D. P. Palomar and M. Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, 2006.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, 2012. URL http://www2.imm.dtu.dk/pubdb/p.php?3274.

P. Philipp. Zustands- und Parameterschätzung für nichtlineare Networked Control Systems. In G. Roppenecker and B. Lohmann, editors, *Methoden und Anwendung der Regelungstechnik*, pages 1–13. Shaker, Aachen, 2009.

P. Philipp. Structure exploiting derivative computation for moving horizon estimation. In *Proceedings of the American Control Conference*, pages 4263–4268, San Francisco, USA, 2011a.

P. Philipp. Exact state sensitivity calculation and its role in moving horizon estimation for nonlinear networked control systems. In G. Roppenecker and B. Lohmann, editors, *Methoden und Anwendung der Regelungstechnik*, pages 27–39. Shaker, Aachen, 2011b.

P. Philipp. Observability of nonlinear NCS with unsynchronized sensor clocks. In *Proceedings of the 51th IEEE Conference on Decision and Control*, pages 1349–1355, Maui, Hawaii, USA, 2012.

P. Philipp and S. Altmannshofer. Experimental validation of a new moving horizon estimator approach for networked control systems with unsynchronized clocks. In *Proceedings of the American Control Conference*, pages 4939–4944, Montreal, Canada, 2012.

P. Philipp and B. Lohmann. Gauss-newton moving horizon observer for state and parameter estimation of nonlinear networked control systems. In *Proceedings of the European Control Conference*, pages 1740–1745, Budapest, Hungary, 2009.

P. Philipp and B. Lohmann. Moving horizon estimation for nonlinear networked control systems with unsynchronized timescales. In *Proceedings of the IFAC World Congress*, pages 12457–12464, Milano, Italy, 2011.

P. Philipp and B. Lohmann. Centralized and decentralized moving horizon estimation for networked control system. In J. Lunze, editor, *Control Theory of Digitally Networked Systems*, pages 86–99. Springer, 2014.

P. Philipp and T. Schmid-Zurek. Distributed moving horizon estimation via dual decomposition. In *Proceedings of the 51th IEEE Conference on Decision and Control*, pages 4792–4798, Maui, Hawaii, USA, 2012.

P. Philipp and M. Schneider. A decentralized moving horizon observer for distributed implementation of centralized controllers. In *Proceedings of the American Control Conference*, pages 2490–2496, Washington D.C., USA, 2013.

L. S. Pontryagin, V. G. Boltyanskii, G. R. V., and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.

G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Proceedings of the 43th IEEE Conference on Decision and Control*, volume 3, pages 2453–2459, 2004.

A. Rantzer. Dynamic dual decomposition for distributed control. In *Proceedings of the American Control Conference*, pages 884–888, 2009.

C. V. Rao. *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin-Madison, 2000.

C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation-a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

J.-P. Richard. Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39(10):1667–1694, 2003.

K. Rieger, K. Schlacher, and J. Holl. On the observability of discrete-time dynamic systems - a geometric approach. *Automatica*, 44(8):2057–2062, 2008.

D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8):2209–2224, 1996.

E. Rosenwasser and R. Yusupov. *Sensitivity of Automatic Control Systems*. CRC Press, 1999.

C. A. Sagastizábal and M. V. Solodov. Parallel variable distribution for constrained optimization. *Computational Optimization and Applications*, 22(1):111–131, 2002.

A. P. Sage and J. L. Melsa. *Estimation Theory with Applications to Communications and Control*. McGraw-Hill, 1971.

S. Samar, S. Boyd, and D. Gorinevsky. Distributed estimation via dual decomposition. In *Proceedings of the European Control Conference*, pages 1511–1516, 2007.

N. R. Sandell, P. Varaiya, M. Athans, and M. G. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.

A. Sandu, D. N. Daescu, and G. R. Carmichael. Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part i - theory and software tools. *Atmospheric Environment*, 37(36):5083–5096, 2003.

G. Schickhuber and O. McCarthy. Disturbed fieldbus and control network systems. *Computing and Control Engineering Journal*, 8(1):21–32, 1997.

F. C. Schweppe. Algorithms for estimating a re-entry body's position, velocity and ballistic coefficient in real time or for post flight analysis. Technical report, Massachusetts Institute of Technology, Lincoln Laboratory, 1964.

P. Seiler. *Coordinated control of unmanned aerial vehicles*. PhD thesis, University of California, Berkeley, 2001.

N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer, 1985.

D. D. Siljak. *Large-Scale Dynamic Systems: Stability and Structure*. Dover, 2007.

J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

C. W. Straka. Adf95: Tool for automatic differentiation of a fortran code designed for large numbers of independent variables. *Computer Physics Communications*, 168:123–139, 2005.

O. Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, 1992.

Y. Sun and N. H. El-Farra. Quasi-decentralized state estimation and control of process systems over communication networks. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 5468–5475, 2008.

B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

Y. A. Thomas. Linear quadratic optimal estimation and control with receding horizon. *Electronics Letters*, 11(1):19–21, 1975.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. Wiley, 1977.

I. B. Tjoa and L. T. Biegler. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 30(2):376–385, 1991.

A. Uppal, W. H. Ray, and A. B. Poore. On the dynamic behavior of continuous stirred tank reactors. *Chemical Engineering Science*, 29:967–985, 1974.

J. Utke, U. Naumann, M. Fagan, N. Tallent, M. Strout, P. Heimbach, C. Hill, and C. Wunsch. OpenAD/F: A modular, open-source tool for automatic differentiation of Fortran codes. *ACM Transactions on Mathematical Software*, 34(4):18:1–18:36, 2008.

F. Valencia, J. D. Lopez, A. Marquez, and J. J. Espinosa. Moving horizon estimator for measurement delay compensation in model predictive control schemes. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6678–6683, 2011.

J. R. Vig. Introduction to quartz frequency standards. Technical Report SLCET-TR-92-1, Army Research Laboratory, Electronics and Power Sources Directorate, 1992.

Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi. Decentralized model predictive control via dual decomposition. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 381–386, 2008.

A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106:25–57, 2006.

Y.-C. Wu, Q. Chaudhari, and E. Serpedin. Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, 28(1):124–138, 2011.

F. Xia and Y. Sun. *Control and Scheduling Codesign: Flexible Resource Management in Real-Time Control Systems*. Springer, 2008.

L. Xiao, M. Johansson, and S. P. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, 52(7):1136–1144, 2004.

B. Xue, S. Li, and N. Li. Moving horizon state estimation for constrained networked control systems with missing measurements. In *Proceedings of the International Conference on Modelling, Identification Control*, pages 719–724, 2012a.

B. Xue, S. Li, and Z. Quanmin. Moving horizon state estimation for networked control systems with multiple packet dropouts. *IEEE Transactions on Automatic Control*, 57(9): 2360–2366, 2012b.

B. Xue, N. Li, S. Li, and Q. Zhu. Moving horizon scheduling for networked control systems with communication constraints. *IEEE Transactions on Industrial Electronics*, 60(8):3318–3327, 2013.

K. Yogesan, S. Kumar, L. Goldschmidt, and J. Cuadros. *Teleophthalmology*. Springer, 2006.

S. Zampieri. Trends in networked control systems. In *Proceedings of the IFAC World Congress*, pages 2886–2894, 2008.

V. M. Zavala. *Computational strategies for the optimal operation of large-scale chemical processes*. PhD thesis, Carnegie Mellon University, 2008.

G. Zimmer. State observation by on-line minimization. *International Journal of Control*, 60 (4):595–606, 1994.

H. Zimmermann. OSI reference model - the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.

Y. Zou and S. Li. Receding horizon estimation to networked control systems with multirate scheme. *Science in China Series F: Information Sciences*, 52(7):1103–1112, 2009.