CARNEGIE MELLON UNIVERSITY

# Nonlinear Programming Sensitivity Based Methods for Constrained State Estimation

**A DISSERTATION**

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

*for the degree of*

DOCTOR OF PHILOSOPHY

*in*

CHEMICAL ENGINEERING

*by*

RODRIGO LÓPEZ NEGRETE DE LA FUENTE

B.S. CHEMICAL ENGINEERING, UNIVERSIDAD IBEROAMERICANA
M.S. CHEMICAL ENGINEERING, UNIVERSIDAD IBEROAMERICANA

Pittsburgh, Pennsylvania

September, 2011

*To my parents, Juan Carlos and Priscilla, and my sister Pia,*
*for their unconditional love and support.*

"My brother has his sword, King Robert has his warhammer, and I have my mind…and a mind needs books as a sword needs a whetstone, if it is to keep its edge."

–Tyrion Lannister
(GoT, George R. R. Martin)

"From too much study, and from extreme passion, cometh madnesse."

–Sir Isaac Newton

# Acknowledgments

I would like to express my most sincere gratitude to my advisor Larry Biegler, for always being willing to patiently listen to my 'silly' questions, and actually giving me straight answers. Without his insights and help this work would not have been completed in time. Without a doubt he is a great person to work with and, better yet, to learn from.

I would also like to thank the members of my committee – Professors Ignacio Grossmann, William Messner, and Erik Ydstie – for graciously agreeing to read my dissertation, and for all their comments and help. Furthermore, I would like to thank Professor Sachin Patwardhan for all the comments and collaborations these past years, which have helped greatly in the development of my work. I also wish to thank Salvador García-Muñoz for the great conversations and collaborations during my internship at Pfizer. In addition, I would like to express my gratitude to all the members of Prof. Biegler's research group for all the interesting discussions that helped shape my work. Last, but certainly not least, I would also like to thank Prof. Antonio Flores Tlacuahuac for encouraging me apply to a PhD program and for suggesting Carnegie Mellon University.

I would also like to gratefully acknowledge NSF for providing me with financial support for the duration of my PhD.

I would also like to thank my friends and family, without their love and support I would not be where I am now. In particular, I'd like to thank Sebastian for making Pittsburgh a more fun and livable place, filled with great conversations. Also, I'd like to express my thanks to Marcus, Wayo, Rosanna, Jorge, Sylvain, and Victor, for their friendship and support at CMU. From la Ibero I would also like to thank my colleagues and friends Andrea, Javier, Martin, and René for being patient teachers and their friendly advice. In addition, I'd like to thank my friends in México – Ricardo, Martha, José Antonio, Juan Pablo, Dalia, Juan José, Rosario, Juan, and Santiago – for their love, encouragement, and for always believing in me. Finally, I would like to give special thanks to Jeanette for her support and all the good memories.

Rodrigo López Negrete de la Fuente
Pittsburgh, PA
September 2011

# Abstract

Model based control schemes, such as nonlinear model predictive control, assume that the full state vector of the plant is known for feedback control. However, in reality this is not always true. Most times only a set of noisy measurements are available, and thus, the unmeasured states need to be inferred from these measurements. This is done in combination with a detailed model of the system. The most common nonlinear state estimation methods do not have a means to deal with bounds or constraints on the states in an efficient or systematical way. These bounds and constraints are important in chemical engineering processes since states usually have physical meaning, for example, concentrations, molecular weights, and conversions are always positive. Therefore, state estimates must be physically feasible. Since Moving Horizon Estimation (MHE) is optimization based it has become a superior strategy for constrained state estimation because bounds are handled optimally by the Nonlinear Programming (NLP) solver. In the present work we develop strategies for MHE based on NLP sensitivity to reduce the on-line computational expense of solving these problems. These formulations are intended to make the on-line application of MHE feasible, by reducing the potential of delays due to the computational expense of solving the associated NLP.

Here we discuss two approaches to update certain tuning parameters in MHE: one of them allows us to reduce the size of the NLP that is being solved, while the other provides a fast approximation of the covariance of the initial condition. The former method is only suitable for small and medium sized problems, while the latter one is better suited for large-scale systems. Additionally, we also discuss NLP sensitivity theory and extensions that apply to the Interior Point solver we use (i.e., IPOPT). With these extensions we are able to develop fast on-line strategies for NMPC and MHE. However, in this work we focus only in the application of these developments to the latter.

To reduce the horizon window we propose methods to approximate the initial condition parameters based on particle filters and sample based statistics to approximate the conditional probability density function (or its parameters) of the initial condition of the states in the MHE horizon window (i.e., the so-called arrival cost). As mentioned above, this approach is suitable mostly for only certain classes of systems that have few states or almost Gaussian behaviors. Therefore, we also develop other methods for on-line MHE suitable for large-scale systems that uses more measurements to reduce any effects of the simplifications done to approximate the initial condition terms. Thus, using NLP sensitivity we develop strategies that leverage the parametric properties of the MHE problem to formulate fast on-line methods applicable to large-scale systems. Moreover, using NLP sensitivity also allows us to relate the optimality conditions of the associated NLP problem to the stochastic origin of MHE. For example, we show the relationship of the covariance of the state estimates with the reduced Hessian matrix of the NLP. This information can also be used to update the parameters if the initial condition penalty term. Moreover, we also discuss the use of Robust M-Estimators to reduce the effects of outliers or gross errors in the measurements. Finally, we illustrate the use and benefits of these strategies through several small and large-scale examples taken from the literature.

# Contents

**Acknowledgments** **i**

**Abstract** **ii**

**Contents** **iii**

**List of Tables** **vi**

**List of Figures** **vii**

**1 Introduction** **1**
1.1 Background . . . 1
1.2 Motivation for Constrained State Estimation . . . 5
1.3 Research Statement . . . 6
1.4 Literature Review . . . 7
1.5 Dissertation Overview . . . 11

**2 Computational Framework** **13**
2.1 IPOPT - Interior Point Solver . . . 13
2.2 Optimal Sensitivity . . . 16
2.2.1 Sensitivity Properties . . . 18
2.2.2 Barrier Sensitivity Calculation . . . 21
2.2.2.1 Active Set Changes . . . 23
2.2.2.2 Fix-Relax Strategy . . . 25
2.2.2.3 General Upper and Lower Bounds . . . 26
2.2.2.4 Multiple Sequential Parameter Perturbations . . . 27
2.2.2.5 Sensitivity Example . . . 28
2.2.3 Extraction of Reduced Hessian Information . . . 32
2.2.3.1 Reduced Hessian Example . . . 34

**3 Bayesian State Estimation** **37**
3.1 Bayesian Framework . . . 38
3.2 Extended Kalman Filter . . . 43
3.3 Unscented Kalman Filter . . . 45
3.4 Particle Filters . . . 48
3.5 Ensemble Kalman Filters . . . 51
3.6 Conclusions . . . 53

**4 Moving Horizon Estimation** **55**
4.1 Derivation of MHE . . . 55
4.2 Filtered Update of the Arrival Cost . . . 61
4.3 Smoothed Update of the Arrival Cost . . . 62
4.4 Conclusions . . . 65

**5 Filtering Approach for Arrival Cost** **66**
5.1 Posterior PDF for Arrival Cost . . . 67
5.2 Approximation of Arrival Cost Parameters . . . 69
5.3 Approximation of Arrival Cost Conditional Density . . . 71
5.4 Examples and Discussion . . . 72
5.4.1 Example 1: CSTR . . . 74
5.4.2 Example 2: Constrained Batch Reactor . . . 76
5.5 Conclusions . . . 82

**6 Smoother Approach for Arrival Cost** **84**
6.1 Derivation of MHE with Smoothed Arrival Cost . . . 85
6.1.1 Optimality Conditions of MHE . . . 89
6.1.2 Covariance-Reduced Hessian Relation . . . 93
6.1.3 Sensitivity Based MHE . . . 97
6.2 Simulation Examples . . . 98
6.2.1 CSTR Network . . . 98
6.2.2 Large-Scale Distillation Example . . . 101
6.3 Conclusions . . . 102

**7 Multi-Rated State Estimation** **106**
7.1 Multi-rated MHE with Smoothing Update of Arrival Cost . . . 107
7.2 Simulation Examples . . . 112
7.2.1 Example 1: Styrene Polymerization Reactor . . . 112
7.2.2 Example 2: Large Scale Binary Distillation Column . . . 116
7.3 Conclusions . . . 119

**8 Advanced Step Moving Horizon Estimation** **121**
8.1 asMHE and asMMHE Strategies . . . 122
8.2 Simulation Examples . . . 124
8.2.1 Example 1: asMHE with a Binary Distillation Column . . . 124
8.2.2 Example 2: asMMHE with a Binary Distillation Column . . . 125
8.3 Conclusions . . . 128

**9 Robust M-Estimators for MHE** **130**
9.1 Robust Statistics . . . 131
9.2 Robust M-Estimators . . . 132
9.3 M-Estimators in MHE . . . 133

9.4 Simulation Example . . . 135
9.4.1 Example 1: CSTR Case Study . . . 136
9.4.2 Example 2: Distillation Column Case Study . . . 137
9.5 Conclusions . . . 139

**10 Conclusions** **142**
10.1 Summary of the Dissertation and Contributions . . . 142
10.2 Future work . . . 145

**Appendices** **160**

**A Derivation of the Discrete-Time Kalman Filter** **161**
A.1 Prediction/Propagation . . . 162
A.2 Moving Horizon Estimation . . . 164
A.2.1 Horizon Length $N = 1$ . . . 165
A.3 Kalman Filter Derivation . . . 167
A.4 Horizon length $N$ . . . 168

**B Mathematical Models** **172**
B.1 Isothermal Gas Phase Reactor . . . 172
B.2 General 3 State CSTR . . . 173
B.3 Styrene Polymerization CSTR . . . 174
B.4 Distillation Column Model . . . 175
B.5 CSTR Network . . . 180

**C Auxiliary Theorems** **181**

**D Expressions for $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$** **182**

# List of Tables

5.1 Sampling types and bounds for the different filters used to update the arrival cost. *PF with importance distribution. In these cases sampling and update bounds are enforced through *importance density/particle filter*. . . . . 73

6.1 Parameters of noise variables used in Example 6.2.1. . . . . 100
6.2 Summary of parameters for the noise variables for the distillation column example in Chapter 6. . . . . 102

7.1 State and measurement covariance values for the polymerization CSTR example. . . . . 115
7.2 State and measurement covariance values for the distillation example. . . . 117

8.1 Summary of parameters for the noise variables for the distillation column example in Chapter 8. . . . . 125
8.2 State and measurement covariance values for asMMHE with the distillation column in Chapter 8. . . . . 127

9.1 Parameters of noise variables for the CSTR example using M-Estimators with MHE. . . . . 136
9.2 Summary of parameters for the noise variables for the distillation column example in Chapter 9. . . . . 139

# List of Figures


1.1 Hierarchy of decisions in plant operations. . . . . 2
1.2 On-line optimization structure. The highlighted block is the main topic of this dissertation. . . . . 3

3.1 Illustration of the relationship between smoothing, filtering, and predicting. 39
3.2 Nonlinear transformation of Gaussian samples. . . . . 45
3.3 Illustration of propagation of sigma points using a nonlinear model. . . . . 46
3.4 Comparison between the curves of a typical and a truncated Gaussian distribution. . . . . 52

4.1 Illustration of how the information is split between the arrival cost and the horizon window in MHE. . . . . 60
4.2 Illustration of the moving horizon in MHE. . . . . 61
4.3 Illustration the filtered update of the arrival cost. . . . . 63
4.4 Illustration the filtered update of the arrival cost. . . . . 64

5.1 Illustration of sample based filter approximation of arrival cost parameters. 70
5.2 Manipulated variable changes for the CSTR example. . . . . 76
5.3 MSE as a function of horizon length when using (a) unconstrained filters and (b) constrained filters for the arrival cost approximation for the CSTR example. . . . . 77
5.4 EKF with different horizon lengths for the isothermal batch reactor. . . . . 79
5.5 Comparison of MHE with horizon length 5 with all methods using unconstrained filters (a) and constrained filters for arrival cost approximation (b). 80
5.6 SSE as a function of horizon length for the isothermal batch reactor with all methods using unconstrained filters (a) and constrained filters for arrival cost approximation (b). . . . . 81

6.1 Illustration of the shared information between smoothed state estimate and the horizon window. . . . . 87
6.2 Illustration of the propagation of covariances in the smoothing arrival cost approach. . . . . 89
6.3 Concentration (a) and Temperature (b) profiles with 6 reactors in the CSTR network example. . . . . 99
6.4 Approximate number of operations required to estimate the arrival cost covariance matrix for the CSTR network example. . . . . 100
6.5 Liquid composition for light component (a) and liquid molar holdup (b) on tray 14. . . . . 103

6.6 Liquid composition for light component (a) and liquid molar holdup (b) on tray 28. . . . . 104

7.1 Illustration of multi-rated signals: (a) without delay (the sampling and measurement are simultaneous) and (b) with delay (sampling and measurement are obtained at different sampling times). . . . . 107
7.2 Polymerization CSTR simulation example. Estimated states using only fast measurements and multi-rated signals are compared for: (a) initiator concentration, (b) monomer concentration, (c) reactor temperature, (d) cooling jacket temperature, (e) weight average, and (f) number average molecular weights. . . . . 114
7.3 Large scale binary distillation example: vapor and liquid compositions for tray 14 (a) and tray 28 (b), $log_{10}$ of estimation error of vapor composition for trays 14 (c) and 28 (d), $log_{10}$ of estimation error of tray efficiency for trays 14 (e) 28 (f). . . . . 118

8.1 Illustration of the asMHE strategy, where the purple dashed line represents the solution of the approximate problem, and the solid red line represents the updated solution using NLP sensitivity. . . . . 122
8.2 Comparison of the estimated states using MHE and asMHE for the distillation column. . . . . 126
8.3 Comparison of the estimated states and parameters using MHE and asMMHE for the distillation column. . . . . 129

9.1 Comparison of the values of the Least Square Function, the Fair Function, and the re-descending estimator. . . . . 134
9.2 Comparison of the Fair Function, re-descending estimator, and least squares estimator MHE using the CSTR example. . . . . 138
9.3 Comparison of the Fair Function, re-descending estimator, and least squares estimator MHE using the distillation column example. . . . . 140

B.1 Illustration of the flows between any two given trays in the distillation column. 175
B.2 Illustration of a network of CSTRs. . . . . 180

# Chapter 1

# Introduction

In this chapter we discuss the hierarchal organization of decisions taken for process operations. This sets the state to introduce the problem of state estimation and its importance in decision making. During plant operations it is very important to know the current state of the system because this information is used to monitor the behavior of the plant. Additionally, this information is used in conjunction with control strategies such as Nonlinear Model Predictive Control (NMPC), to compute the inputs that should drive the plant into satisfying the specified set points. Therefore, it is important that the state estimates are accurate, and that they can be generated fast enough so that very little delays are introduced to the system.

The rest of this chapter is organized as follows. In Section 1.1 we introduce the hierarchy of decisions in plant operations, and we situate the state estimation problem in this framework. In Section 1.2 we motivate the benefits of using constrained state estimation methods in chemical engineering. In Section 1.3 we discuss the main objectives and results of the present work, and Section 1.4 contains a summary of previous work related to the topics discussed in this thesis. Finally, an overview of the rest of the chapters is presented in Section 1.5.

## 1.1 Background

Operation of chemical processes requires a large amount of decisions that need to be taken at different rates and that consider very different time frames. These decisions can be organized depending on the frequency in which they are taken and time frames considered.

In Figure, 1.1 we illustrate this organization, where as we move towards the upper layers decisions are taken less frequently, but they consider longer time horizons. On the other hand, as we move down to the lower layers decisions are taken more frequently and usually consider shorter time periods [1]. For example, scheduling and planning problems consider time frames of few weeks or months, and these decisions are not taken very frequently (e.g., every few weeks or months). Then again, Real-Time Optimization (RTO) and Model Predictive Control (MPC) problems are solved every few seconds or minutes, and these problems consider time horizons of a few seconds or minutes. Moreover, even the classes of optimization problems solved are different. Planning and scheduling problems usually consist of Mixted-Integer Nonlinear programming problems (MINLP) or Mixed-Integer Linear Problems (MIP). On the other hand, RTO and MPC problems are usually Nonlinear Programming Problems (NLP). Therefore, the solution strategies used for the different layers are very different.

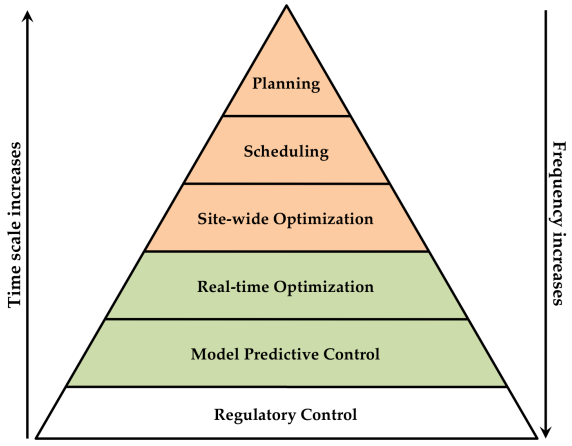



Figure 1.1: Hierarchy of decisions in plant operations.

Figure 1.2 illustrates how the top layers communicate with the bottom ones. Here, the planning and scheduling layers set targets that are sent to the RTO and MPC layers. Note that the only requirement for interaction between the layers is that decision remain feasible

at lower levels. The bottom layers compute values for the inputs that are injected into the plant and managed through the regulatory level. Combining measurement data from the plant and the state and parameter estimation block we generate information about the current state of the plant which is used for model maintenance, feedback and monitoring purposes.

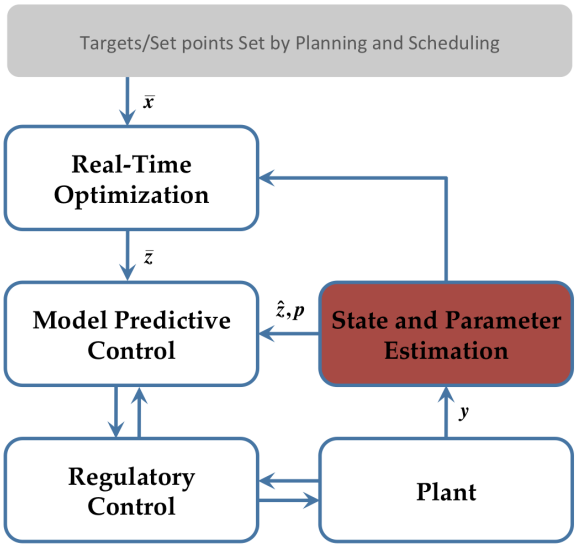



Figure 1.2: On-line optimization structure. The highlighted block is the main topic of this dissertation.

The work presented in this dissertation focuses on problems in the RTO and MPC layers. These blocks are characterized by the types of problems that are used in the decision making process. For example, for NMPC, Dynamic RTO (DRTO) and state estimation the optimization problems are dynamic in nature and constrained with the a first principle predictive model of the plant. However, the objectives used in each case can be very different than in the others. For example, for NMPC the objective could be to minimize the error between the state of the plant and a set point [2], or it could consider some types of economic objectives [3, 4]. On the other hand, state and parameter estimation problems minimize the error between predictions and measurements taken from the plant.

In particular, for this dissertation we focus on the State and Parameter Estimation Block (highlighted in gray in Fig. 1.2). Here, we are especially interested in moving horizon methods that use measurement data from the past and current behavior of the plant to compute estimates of the actual state (and parameters) of the system. Moving Horizon Estimation (MHE) is of particular interest for this work because at the core of these strategies a constrained NLP is solved [5]. Thus, constraints on the states are handled optimally by the NLP solver. Constraints are important since for chemical processes the states that are being estimated have a physical meaning. Thus, for example, concentrations and flows are always positive, mole and mass fractions are always between 0 and 1, or we may even have that some other states are always within a certain range (e.g., temperatures having upper and lower bounds). Not taking into account these bounds and constraints can cause the estimator to diverge, and thus, the estimation error increases [6]. Since monitoring and control strategies depend on these estimates, it is important to have small estimation errors, or their performance would be severely reduced.

In the present work we deal with two of the main problems associated with MHE: Computational expense of solving the problem, and consistency with the stochastic properties of the *constrained* state estimation problem. For the former we develop methods that are based on interior point NLP solvers to extract NLP sensitivity information that can be used to generate fast approximations of neighboring solutions of the optimization problem. In addition, here we also show how NLP sensitivity can be used to extract fast approximations of covariance matrix information of the state estimates that can be used to update some important tuning parameters of MHE. On the other hand, we also provide some analysis on the stochastic properties of the so-called arrival cost. We show that if this term is correctly approximated we no longer need very long horizon windows, and thus we can reduce the computational expense of solving the MHE problems.

## 1.2 Motivation for Constrained State Estimation

As mentioned above constrained state estimation is important for computing predictions of states in chemical plants. Bounds and constraints are of particular interest here since states represent physical quantities. However, the most common nonlinear state estimation methods cannot handle them in a systematic way. For example, the Extended Kalman Filter (EKF), which is the most common state estimation method used in industry [7, 8], has no way of systematically handling them. State estimates computed with EKF have been shown to diverge for certain classes of systems [6]. On the other hand, some EKF formulations handle bounds through clipping. In other words, when a state estimate violates the bounds it is projected back onto it. Nevertheless, this strategy is suboptimal at best, and the performance of the estimator is greatly reduced.

Another consequence of states being bounded is that they will no longer be Gaussian, which is the most common assumption used [8]. Assuming that states remain Gaussian, even when the system is nonlinear or when bounds are present, mostly affects stochastic quantities associated with the estimates, and not so much the performance of the estimator. For example, confidence intervals associated with the estimates may not represent the true ones. However, neglecting to consider the non-Gaussian nature of the states will not impact the estimation error as much as when bounds are not included.

Several methods have been developed that deal with constrained state estimation, non-Gaussian systems, or both. For example, for constrained state estimation clipping can be applied to any method, but Nonlinear Recursive Dynamic Data Reconciliation (NRDDR) has been proposed as a more consistent approach [9]. On the other hand, for non-Gaussian systems sample based filters are used such as the Unscented Kalman Filter UKF [10, 11, 12], Ensemble Kalman Filters (EnKF) [13, 14], and Particle Filters (PF) [15, 16]. Finally, constrained sample base filters also exist to deal with non-Gaussian systems and constrained states. Among these we have the Unscented Recursive Nonlinear Dynamic Data Reconciliation (URNDDR) [9, 17], Constrained Ensemble Kalman Filters (CEnKF) [18], and Con-

strained Particle Filters (CPF) [19, 20].

In contrast to these methods, MHE processes a batch of past measurements to generate the state estimate. By doing this, we avoid problems that could arise if a few measurements are biased. Methods that only use one measurement at a time give full weight to that observation, and this could bias the estimate if there is a large error in it. MHE, on the other hand, distributes the weight of the observation data onto several measurements. That way, if a few of the observed variables are skewed, the rest of the measurements mitigate the effects of the bad ones. Furthermore, since MHE solves a constrained nonlinear program at each sample time, then bounds and constraints are handled systematically by the optimizer. Thus, providing optimally constrained state estimates. It is because of this that MHE also provides a good framework for *constrained* multi-rated state estimation. Since several measurements in the past are being used, it is possible to introduce slower measurements at the correct sample times to include them.

## 1.3 Research Statement

In this section we highlight the main objectives of this dissertation. The main objectives of the present work deal with two main topics: develop fast strategies for NLP sensitivity, and propose efficient MHE methods for constrained state estimation.

We were particularly interested in developing efficient NLP sensitivity strategies based on IPOPT since this solver is particularly well suited for large-scale dynamic optimization problems. This sensitivity information we then use to generate fast approximations of neighboring solutions of the NLP that can be used to make fast MHE or NMPC strategies. Furthermore, we were also interested in extracting reduced Hessian information from the solver, which is not normally formed by IPOPT. Reduced Hessian information provides insights into the quality of the solutions, and we were also interested in associating this information to the stochastic properties of MHE.

On the other hand, we also analyzed common MHE formulations to build on top of

them to provide faster, more efficient strategies. We reduced the computational expense of solving the MHE problem by reducing the required horizon window, without sacrificing the quality of the state estimates, by generating better approximations of the arrival cost term. Additionally, we shifted the computational expense of solving a large-scale NLP problem to the background through the use of NLP sensitivity. Also, through sensitivity, we show a relationship between the optimality conditions of the NLP and the stochastic properties of the MHE problem, which can be exploited to extract covariance information of the state estimates. Furthermore, we also analyzed formulations that can deal with multi-rated measurements and that can deal with outliers or gross errors.

This dissertation is based on the work of others, and in the following section we provide a summary of previous work done in relation to NLP sensitivity and MHE. However, if the reader is interested in delving deeper into some of these topics, he or she is encouraged to also follow the references cited in the following chapters.

## 1.4 Literature Review

In this section we summarize previous work that has been done in relation to Moving Horizon Estimation and related topics. The main focus of the work has been on efficient solution methods for dynamic optimization problems, efficient formulations for on-line implementation, and better understanding of the properties of the MHE problems.

There are vast amounts of work related to nonlinear programming, dynamic optimization, and efficient solution methods of these problems. These topics are widely researched areas, and it would be impossible to give a complete summary of the state of the art here. Thus, the reader is referred to the work of Nocedal and Wright [21], Biegler [22], Edgar and Himmelblau [23] for NLP theory and its application to chemical processes. For details on efficient formulations and numerical solution of dynamic optimization, we refer the reader to the work of Biegler [22], Biegler *et al.* [24], Biegler [25, 26], and the references therein. For NLP Sensitivity the work of Fiacco [27] and the references cited in Chapter 2, are good

sources of information. On the other hand, applications of these techniques to Nonlinear Model Predictive Control can be found in Zavala [28] and Zavala *et al.* [29]. Moreover, process synthesis and control applications of Parametric Programming, which is closely related to sensitivity, are discussed in Pistikopoulos [30], Pistikopoulos *et al.* [31, 32], and the citations therein.

In recent literature, the problem of approximating the initial condition penalty term (or arrival cost) in the MHE objective function still remains an open issue. The arrival cost term appears in the MHE formulation because it was developed to approximate the full information problem by only considering a fixed size moving window of past measurements, also called a horizon window. This is discussed in more details in Chapter 4. This term in the objective function is used to incorporate information from previous measurements that are not included in the horizon window. In other words, it represents the distribution of the state at the beginning of the window given the prior measurement information. The most common way of approximating it is through a weighted 2-norm. For example, for linear systems Rao *et al.* [33] assume that the distribution is Gaussian, and propose the use of a KF and also a Kalman Smoother update of the covariance matrix in the arrival cost term. Moreover, they use both methods to prove that for a linear unconstrained system, the KF solution is obtained from the MHE formulation. They also show that the stability of the MHE can be proved directly from the stability of KF and the stability of the full information problem, thus validating their approach for unconstrained estimates. On the other hand, for nonlinear systems, Rao and Rawlings [34] use an EKF approximation of the arrival cost term, and show that when bounds are not considered the EKF solution can be obtained. Moreover, Tenny and Rawlings [35] compare both the EKF and corresponding smoothing strategies for constrained nonlinear systems, and conclude that the smoothing scheme is superior since the EKF based method induces oscillations due to errors in the arrival cost approximation that are propagated through the horizon. Furthermore, both of these methods assume that the conditional probability densities of the states are well rep-

resented with Gaussian distributions. However, if there are bounds on the states the prior and posterior distributions will not remain Gaussian even if the state and measurement noises are. Moreover, both methods are unconstrained estimators, and using them to approximate the parameters of the arrival cost in constrained MHE results in an inconsistent approximation.

On the other hand, if the system is nonlinear the propagation of a normally distributed random variable through the model will not remain Gaussian. To address this Qu and Hahn [36] propose the use of the Unscented Kalman Filter, that uses sigma points to better approximate the distribution, to update only the covariance matrix in the arrival cost. This resulted in a slight improvement in performance. However, they do not show the effects of using such an update on the horizon window size. Moreover, they also use an unconstrained estimator which, as mentioned earlier, is an inconsistent approach for approximating the arrival cost when states have bounds. Ungarala [37] addresses this difficulty by employing a constrained version of the UKF where clipping is used to satisfy bounds. However, this approach is ad hoc, and leads to suboptimal solutions at best. He also developed approaches for estimating the arrival cost using a constrained type of Particle Filter (PF )[38] and Cell Filter (CF). For the former, the selection of a suitable form of importance function to represent the true posterior density is a crucial step in the particle filter [16, 37, 39]. In the conventional approach, the state transition density (or prior) is used as the importance distribution/importance function. Because the state transition density (being used as importance density function) does not take into account the most recent observation, the particles drawn from the transition prior may have very low likelihood, and their contributions to the posterior estimation become negligible. While this limitation is acknowledged by the author, there is no attempt to overcome this difficulty [37]. It may be noted that the use of appropriate importance functions can significantly reduce the number of particles required for generating accurate estimates, when compared with the conventional particle filter [16], and thereby reduce the computational cost, while

providing a better approximation of the density. Specifically, for systems whose states are bounded we need to take special care in choosing the appropriate importance distribution, and it should be updated in a way that is consistent with the bounds. Among the constrained UKF, PF and CF used by Ungarala [37], the latter turns out to be the best choice for arrival cost estimation in the simulation examples considered. However, as also acknowledged by the author, the computational burden associated with CF renders it suitable only for very small dimensional systems (for example, a two state system required 500 cells with 500 particles per cell). Thus, most of the approaches available in the literature for approximation of the arrival cost propose to employ unconstrained filters, and implicitly assume that the conditional densities of the states can be approximated as Gaussian, which is inconsistent when the states are constrained. While PF and CF proposed recently by Ungarala [37] addresses this problem through the use of constrained PF, the version of PF used does not use the measurement information while constructing importance density.

In some cases the plant measurements are not available at the same rate. For example, temperatures or pressures can be measured very quickly, while, molecular weights or biomass measurements take longer and may even be delayed. The most common state estimation methods usually use only the fast measurements under the assumption that the all states of interest are observable through them. Unfortunately, this assumption does not always hold, and in some cases there are states that are not observable through the fast measurements alone. However, it is sometimes possible to combine different measurements that are available at different rates to change the observability properties of the system, and make some of the unobservable states observable [40, 41]. Therefore, methods that can handle multi-rated measurements with or without delays should be used to take advantage of the slower measurements to improve the quality of the state estimates or their observability.

Several methods have been developed to handle multi-rated measurements, and they can be classified in two possible ways depending on how they deal with the multi-rated

measurements. For example, in [42] two possible classes of methods for EKF based state estimation are defined: *(i)* state augmentation methods, and *(ii)* methods that fuse the slow (possibly delayed) measurement at the moment it arrives. In their manuscript, the authors describe in detail the differences between these methods, and show the differences using simulation examples. In [40, 43] continuous and discrete time reduced order nonlinear observers are used for state estimation in polymerization reactors. The authors propose that, using polynomial extrapolation, the slow measurements be predicted for those sample times where only fast measurements are available. Moreover, the authors also describe a method for tuning the observer gains that makes sure that the estimation error decays to zero asymptotically. Finally, in [44] the authors show a real-time implementation of the observer to estimate molecular weights in a pilot-plant scale reactor. In [45] a state augmentation method is used to estimate states and parameters in a bioreactor using multi-rated measurements. State augmentation is used to take into account delayed slow measurements by generating smoothed state estimates in between major and minor sampling times, where the major sampling time corresponds to the moment when both fast and slow measurements are available; at the minor sampling times only fast measurements are obtained. Also, since the output equation in their case study is nonlinear, they use an iterated EKF to improve the quality of the estimates. Furthermore, the authors analyze the observability of the system using only fast measurements and also, both fast and slow measurements. They conclude that the states of interest are only observable when using multi-rated measurements.

## 1.5 Dissertation Overview

The rest of this dissertation is organized as follows. In Chapter 2 we give an introduction to the computational framework this work is based on. In particular we discuss properties of the NLP solver IPOPT, and the sensitivity results that have been extended to deal with the particulars of the solver. In Chapter 3, we discuss Bayesian state estimation to set the stage

for constrained state estimation through MHE. Chapters 4, provides a detailed discussion of the stochastic origin of MHE, while Chapters 5-8 discuss the main results of the present work in relation to making MHE efficient for on-line application. In Chapter 9 we provide insights into Robust M-Estimators that can be used with MHE to reduce the effects that outliers and gross errors in the measurements have on the state estimates. Finally, we conclude in Chapter 10, where we provide a summary of the main results described in the thesis, and a few suggestions for future work.

# Chapter 2

# Computational Framework

In this chapter we describe the Interior Point algorithm implemented in IPOPT [46] which is used to to solve the nonlinear programs that arise from the state estimation methods described in this work. In addition, we also show some properties related to nonlinear programming sensitivity, which we can use to obtain fast approximations to the solutions of NLPs when parameter values change. Furthermore, we show a simple method to extract reduced Hessian, which is not generated as part of the algorithm used in IPOPT, and that we will use in future chapters to approximate important information of the state estimators.

The rest of the chapter is organized as follows. In Section 2.1 we describe some of the main characteristics of the algorithm implemented in IPOPT (the Interior Point solver), and in Section 2.2 we discuss in detail some important properties of NLPs. Moreover, in this section we also discuss how to extract sensitivity information from the linearized optimality conditions used in IPOPT, and we discuss the implementation of these methods in combination with the solver which leads to *sIPOPT*.

## 2.1 IPOPT - Interior Point Solver

In this section we discuss some of the details of the IPOPT algorithm [46] used to solve the large-scale NLPs that arise form the DAE-constrained optimization problems used in this work. For this we consider the optimization problem of the form

$$\min_x f(x;p) \tag{2.1a}$$

$$\text{s.t. } c(x;p) = 0 \tag{2.1b}$$

$$x \geq 0 \tag{2.1c}$$

with the vectors $x \in \mathbb{R}^{n_x}$, $p \in \mathbb{R}^{n_p}$, and $c(x;p) : \mathbb{R}^{n_x+n_p} \to \mathbb{R}^m$. Note that $p$ represents the parameters of the NLP, and they are *fixed* during the solution of the problem. Also, to simplify notation, here we only consider upper bounds on the variables. NLPs with general upper and lower bounds can be transformed into this form with a simple change of variables, and also, the following derivations can be rewritten to handle both types of bounds as shown in Section 2.2.2.3. Furthermore, IPOPT handles the aforementioned bounds implicitly by substituting a barrier function for the inequality constraints (Eq. (2.1c)). In this way, the barrier problem is defined as follows,

$$\min_x \quad B(x;p,\mu_\ell) = f(x;p) - \mu_\ell \sum_{i=1}^{n_x} ln(x_i) \tag{2.2a}$$

$$\text{s.t.} \quad c(x;p) = 0, \tag{2.2b}$$

$\mu_\ell$ represents the barrier parameter. Solving NLP (2.2) for a decaying sequence of values of the barrier parameter, where $\mu_\ell \to 0$, results in the solution of NLP (2.1). Since the barrier approach handles the bounds implicitly, we avoid the combinatorial complexity of finding the active set of the NLP. At each major iteration $\ell$ of IPOPT we need the solution of the barrier problem for a given value of the barrier parameter $\mu_\ell$. For this we solve NLP (2.2) using this value of the barrier parameter, and for which the optimality conditions are given by

$$\nabla_x \mathcal{L}(x, \lambda, \nu; p) = \nabla_x f(x; p) + \nabla_x c(x; p)\lambda - \nu = 0 \tag{2.3a}$$

$$c(x; p) = 0 \tag{2.3b}$$

$$XVe - \mu_\ell e = 0, \tag{2.3c}$$

where $X = diag\{x\}$, $V = diag\{\nu\}$, $e \in \Re^{n_x}$ is a vector of ones, and $\mathcal{L}(x, \lambda, \nu; p)$ is the associated Lagrange function of the form

$$\mathcal{L} = f(x; p) - \mu_\ell \sum_{i=1}^{n_x} ln(x_i) + \lambda^T c(x; p). \tag{2.4}$$

Moreover, $\lambda \in \Re^{n_\lambda}$ and $\nu \in \Re^{n_x}$ represent the vectors of Lagrange multipliers related to the equality constraints and bounds, respectively. To find the solution of the primal-dual system described with Equation (2.3), IPOPT implements an exact Newton's method with an initial solution iterate given by $s_0^T = \left[x_0^T, \lambda_0^T, \nu_0^T\right]$. Therefore, for the *i*-th Newton iteration, the search direction $\Delta s_i = s_{i+1} - s_i$ is computed with the solution of the linearized KKT conditions (2.3), which are given by

$$\begin{bmatrix} W_i & A_i^T & -\mathbb{I} \\ A_i & 0 & 0 \\ V_i & 0 & X_i \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta \lambda_i \\ \Delta \nu_i \end{bmatrix} = - \begin{bmatrix} \nabla_x f_i + A_i^T \lambda_i - \nu_i \\ c(x_i; p) \\ X_i V_i e - \mu_\ell e \end{bmatrix}, \tag{2.5}$$

where $A_i = \nabla_x c(x_i; p)^T \in \Re^{n_\lambda \times n_x}$ is the Jacobian matrix of the constraints, and $W_i = \nabla_{xx} \mathcal{L}(x_i; p) \in \Re^{n_x \times n_x}$ is the Hessian matrix of the Lagrangian. Also, $\mathbb{I}$ represents the identity matrix of appropriate dimensions. In addition, the left hand side matrix in Equation (2.5) is also known as the KKT or primal-dual matrix. Note that Eq. (2.5) is solved several times at each iteration of IPOPT. Therefore, this is a crucial step in the algorithm, and to make it much more efficient, IPOPT uses state-of-the-art sparse linear solvers. These types of solvers can be very fast since they make efficient use of sparse algebra, and they take advantage of the sparse nature of the system to reduce the amount of memory space required to store the matrices.

Provided that we can can supply IPOPT with exact first and second order derivatives, it is possible to guarantee fast local convergence of Newton's method. Furthermore, it is possible to handle problems with large amounts of degrees of freedom without altering the local convergence properties (as opposed to Quasi-Newton methods). Supplying exact derivatives is easily achievable when using mathematical programming platforms such as AMPL [47], that generate derivative information using automatic differentiation. Moreover, IPOPT implements a filter line search method as a globalizing strategy to promote convergence from bad starting points [48, 49].

## 2.2 Optimal Sensitivity

When analyzing the solution of nonlinear programming problems, optimal sensitivity is a key tool which can provide information about regularity and curvature of the optimal points. Moreover, these methods are useful to assess which variables play a dominant role in the optimization, and they can be used to provide first order estimates for parametric NLPs. Furthermore, when using NLP algorithms that take advantage of exact derivatives, sensitivity can be calculated very efficiently and with very little added computational cost. In this section we discuss how these calculations can be done in the context of IPOPT [50].

The basic strategy for optimal sensitivity for NLP solvers is derived through the use of the Implicit Function Theorem (IFT) on the KKT conditions of a parametric NLP. Originally, Fiacco [51] showed that sensitivities could be obtained by solving a linearization of the optimality conditions, when some suitable regularity conditions are satisfied. Nevertheless, relaxation of some of these regularity conditions induces singularity and inconsistency in this linearized system, and may make the sensitivity calculation much more difficult. Reviews of sensitivity analysis can be found in Fiacco [51], Fiacco and Ishizuka [52], and Büskens and Maurer [53]. More advanced cases have been analyzed by Kyparsis [54], Kojima [55], Kojima and Hirabayashi [56], and Jongen *et al.* [57, 58]. An early implementation of sensitivity analysis applied to barrier methods can be found in Fiacco and

Ghaemi [59].

For over 25 years, NLP sensitivity analysis has been applied to numerous applications in process engineering, as these can be viewed as parametric programs, with parameters that represent uncertain data or unknown inputs. These applications include flow-sheet optimization [60, 61], steady state real-time optimization [62], robust optimization and parameter estimation, nonlinear model predictive control [63], and dynamic real-time optimization [64].

With improvements in fast Newton-based barrier methods, such as IPOPT, sensitivity information can be obtained very easily. As a result, fast on-line algorithms can be constructed where more expensive optimization calculations can be solved in advance, and fast updates to the optimum can be made on-line. This has motivated algorithms for control, estimation and dynamic optimization that execute large optimization models with very little on-line computation [65]. This has been especially successful for so-called *advanced step* strategies that work with large, sparse NLP solvers. To obtain the sensitivity information, we exploit the full space information available through the linearization of the KKT conditions that appears naturally in the solution process.

In the following sections some well known concepts associated to optimal NLP sensitivity are described. These will be helpful to derive the state estimation strategies described in the next few chapters. For these strategies, we are interested in a parametric nonlinear program with the same form as in (2.1), and the associated barrier problem (2.2). At a solution with parameter $p = p_0$ (the nominal value) we compute the sensitivities $\frac{dx^*(p_0)}{dp}$ and

$$\frac{df(x^*;p_0)}{dp}^T = \frac{\partial f(x^*;p_0)}{\partial p}^T + \frac{dx(p_0)}{dp}^T \frac{\partial f(x^*;p_0)}{\partial x}^T \tag{2.6}$$

Thus, to calculate these sensitivities, we first need to consider the properties of the solutions of (2.1) obtained by IPOPT when $p = p_0$ [51, 66].

### 2.2.1 Sensitivity Properties

For the NLP (2.1) with parameter $p = p_0$, the Karush-Kuhn-Tucker (KKT) conditions are defined as:

$$\phi(s^*; p_0) = \begin{bmatrix} \nabla_x f(x^*; p_0) + \nabla_x c(x^*; p_0)\lambda^* - \nu^* \\ c(x^*; p_0) \\ X^*V^*e - \mu_\ell^* e \end{bmatrix} = 0. \tag{2.7}$$

For the KKT conditions to serve as necessary conditions for a local minimum of (2.1), constraint qualifications are needed, such as Linear Independence Constraint Qualification (LICQ) or Mangasarian-Fromowitz Constraint Qualification (MFCQ). Note that in the following definitions $Q_j$ means the *j*-th column of a given matrix $Q$, and $\mathbb{I}$ represents the identity matrix of appropriate dimensions.

**Definition 2.1** (Linear Independence Constraint Qualification (LICQ))**:** *Given a local solution of* (2.1)*, $x^*$ and an active constraint set, consisting of equalities and $n_b$ bound constraints with $E^T x^* = 0$, where $E_i = \mathbb{I}_j$ if $x_j^* = 0$ is the* i*-th bound constraint with $j = 1, \ldots, n$, $i = 1, \ldots, n_b$. LICQ is defined by linear independence of the active constraint gradients, i.e.*

$$[\nabla c(x^*; p_0) \mid E]$$

*is full column rank. Satisfaction of LICQ leads to unique multipliers $\lambda^*, \nu^*$ at a KKT point.*

**Definition 2.2** (Mangasarian-Fromowitz Constraint Qualification (MFCQ))**:** *Given a local solution of* (2.1)*, $x^*$ and an active set, MFCQ is defined by linear independence of the equality constraint gradients and the existence of a search direction $d$, such that:*

$$E^T d > 0, \nabla c(x^*; p_0)^T d = 0.$$

*Satisfaction of MFCQ leads to bounded (but not necessarily unique) multiplers $\lambda^*, \nu^*$ at a KKT point. As noted in [54], these multipliers lie in a bounded, polyhedral set denoted by $P(x^*; p_0)$.*

For a KKT point $x^*$ that satisfies either of the above constraint qualifications the following conditions are sufficient to guarantee a local solution to (2.1).

**Definition 2.3** (Strong Second Order Sufficient Conditions (SSOSC))**:** *SSOC is defined at $x^*$ and some multipliers $\lambda^*$, $\nu^*$ that satisfy the KKT conditions (2.3), and where*

$$d^T \nabla_{xx} \mathcal{L}\left(x^*, \lambda^*, \nu^*; p_0\right) d > 0 \quad \textit{for all} \;\; d \neq 0 \;\; \textit{with} \;\; \nabla c\left(x^*\right)^T d = 0, d_i = 0 \;\; \textit{for} \;\; \nu_i^* > 0. \quad (2.8)$$

**Property 2.1** (Properties of the central path/barrier trajectory)**:** *Consider problem* (2.1) *with $f(x;p)$ and $c(x;p)$ at least twice differentiable in $x$ and once in $p$. Let $x^*$ be a local constrained minimizer of* (2.1) *with the following sufficient optimality conditions at $x^*$:*

1. *$x^*$ is a KKT point that satisfies (2.3)*
2. *LICQ holds at $x^*$*
3. *Strict complementarity (SC) holds at $x^*$ for the bound multipliers $\nu^*$, i.e., $x_i^* + \nu_i^* > 0$*
4. *SSOSC holds for $x^*$, $\lambda^*$, and $\nu^*$ that satisfy (2.3)*

*If we now solve a sequence of barrier problems (2.2) with $\mu_\ell \to 0$, then:*

- *There is at least one subsequence of unconstrained minimizers $(x(\mu_\ell))$ of the barrier function converging to $x^*$*
- *For every convergent subsequence, the corresponding sequence of barrier multiplier approximations is bounded and converges to multipliers satisfying the KKT conditions for $x^*$.*
- *A unique, continuously differentiable vector function $x(\mu)$ of the minimizers of (2.2) exists for $\mu > 0$ in a neighborhood of $\mu = 0$*
- $\lim_{\mu \to 0^+} x(\mu) = x^*$
- $\|x(\mu) - x^*\| = O(\mu)$

**Proof:** The proof follows by noting that LICQ implies MFCQ and invoking Theorem 3.12 and Lemma 3.13 in [66].

This theorem indicates that nearby solutions of (2.2) provide useful information for bounding properties for (2.1) for small positive values of $\mu$. For such cases we now consider the sensitivity of these solutions with respect to changes in values of $p$.

**Property 2.2** (Sensitivity Properties)**:** *For problem* (2.1) *assume that $f(x;p)$ and $c(x;p)$ are $k$ times differentiable in $p$ and $k+1$ times differentiable in $x$. Also, let the assumptions of Property 2.1 hold for problem* (2.1) *with $p = p_0$, then at the solution:*

- *$x^* = x(p_0)$ is an isolated minimizer and the associated multipliers $\lambda^*$ and $\nu^*$ are unique.*
- *For some $p$ in a neighborhood of $p_0$ there exists a $k$ times differentiable function*

$$s(p)^T = \left[x(p)^T \ \lambda(p)^T \ \nu(p)^T\right]$$

  *that corresponds to a locally unique minimum for* (2.1) *and $s(p_0) = s^*$.*
- *For $p$ near $p_0$ the set of binding inequalities is unchanged and complementary slackness holds.*

**Proof:** The result follows directly from Theorem 3.2.2 and Corollary 3.2.5 in [51].

We now consider the barrier formulation and relate sensitivity results between (2.1) and (2.2) with the following result.

**Property 2.3** (Barrier Sensitivity Properties)**:** *For the barrier problem (2.2) assume that $f(x;p)$ and $c(x;p)$ are $k$ times differentiable in $p$ and $k+1$ times differentiable in $x$. Also, let the assumptions of Property 2.1 hold for problem* (2.1)*, then at the solution of (2.2) with a small positive $\mu$:*

- *$x(\mu;p_0)$ is an isolated minimizer and the associated barrier multipliers $\lambda(\mu;p_0)$ and $\nu(\mu;p_0)$ are unique.*
- *For some $p$ in a neighborhood of $p_0$ there exists a $k$ times differentiable function*

$$s(\mu;p)^T = \left[x(\mu;p)^T \ \lambda(\mu;p)^T \ \nu(\mu;p)^T\right]$$

  *that corresponds to a locally unique minimum for (2.2).*

- $\lim_{\mu \to 0, p \to p_0} s(\mu; p) = s(0, p_0) = s^*$

**Proof:** The result follows from Theorem 6.2.1 and Corollary 6.2.2 in [51]. These were originally proved for a mixed penalty function but the proofs are easily modified to deal with barrier functions.

### 2.2.2 Barrier Sensitivity Calculation

Calculation of the sensitivity of the primal and dual variables with respect to $p$ now proceeds from the implicit function theorem (IFT) applied to the optimality conditions of (2.2) at $p_0$. In other words, we apply IFT to (2.7) to yield

$$\frac{d\phi\left(p; s\right)}{dp}^T = \frac{\partial \phi\left(p; s\right)}{\partial p}^T + \frac{d\phi\left(p; s\right)}{ds}^T \frac{\partial s\left(p\right)}{\partial p}^T = 0. \tag{2.9}$$

Thus, defining the quantities:

$$M(s(\mu; p_0)) = \frac{\partial \phi\left(p; s\right)}{\partial s}^T = \begin{bmatrix} W\left(s\left(\mu; p_0\right)\right) & A\left(x\left(\mu; p_0\right)\right)^T & -I \\ A\left(x\left(\mu; p_0\right)\right) & 0 & 0 \\ V\left(\mu; p_0\right) & 0 & X(\mu; p_0) \end{bmatrix} \tag{2.10}$$

and

$$N_p(s(\mu; p_0)) = \frac{\partial \phi\left(p; s\right)}{\partial p}^T = \begin{bmatrix} \nabla_{xp}\mathcal{L}\left(s\left(\mu; p_0\right)\right)^T \\ \nabla_p c\left(x\left(\mu; p_0\right)\right)^T \\ 0 \end{bmatrix}, \text{ and } N_\mu = \begin{bmatrix} 0 \\ 0 \\ -\mu e \end{bmatrix} \tag{2.11}$$

where $W(s(\mu; p_0))$ denotes the Hessian $\nabla_{xx}\mathcal{L}\left(x, \lambda, \nu\right)$ of the Lagrangian function evaluated at $s(\mu; p_0)$, $A(x(\mu; p_0)) = \nabla_x c(x)^T$ evaluated at $x(\mu; p_0)$, $X = diag\left\{x\right\}$ and $V = diag\left\{\nu\right\}$, and making substitutions into (2.9) leads to:

$$M(s(\mu; p_0))\frac{\partial s(\mu; p_0)}{\partial p}^T + N_p(s(\mu; p_0)) = 0. \tag{2.12}$$

When the assumptions of Property 2.1 hold, $M(s(\mu; p_0))$ is nonsingular and the sensitivities can be calculated from:

$$\frac{\partial s(\mu; p_0)}{\partial p}^T = -M\left(s\left(\mu; p_0\right)\right)^{-1} N_p\left(s\left(\mu; p_0\right)\right). \tag{2.13}$$

We note that at the solution of (2.2) these assumptions can be checked by the inertia of $M$ as well as other information in IPOPT (see [50]). Moreover, in IPOPT $M(s(\mu; p_0))$ is directly available in factored form from the solution of (2.2), so the sensitivity can be calculated through a simple backsolve. For small values of $\mu$ and $\|p - p_0\|$ it can be shown from the above properties [51] that

$$s\left(\mu; p\right) = s\left(\mu; p_0\right) - M(s(\mu; p_0))^{-1} N_p(s(\mu; p_0))(p - p_0) + o\|p - p_0\| \tag{2.14}$$

or

$$s(0; p) = s(\mu; p_0) - M(s(\mu; p_0))^{-1} \left[N_p(s(\mu; p_0))(p - p_0) + N_\mu\right] + o\|p - p_0\| + o\|\mu\|. \tag{2.15}$$

Finally, the implementation of NLP sensitivity is simplified if the parameters can be localized in the NLP formulation, so that we write:

$$\begin{aligned} \min_{x,w} \quad & f(x, w) && \text{(2.16a)} \\ \text{s.t.} \quad & c(x, w) = 0, x \geq 0 && \text{(2.16b)} \\ & w - p_0 = 0 && \text{(2.16c)} \end{aligned}$$

Note that the NLP solution is equivalent to (2.1), and it is easy to see that the NLP sensitivity is equivalent as well. Writing the KKT conditions for (2.16) leads to:

$$\nabla_x f(x,w) + \nabla_x c(x,w)\lambda - \nu = 0 \tag{2.17a}$$

$$\nabla_w f(x,w) + \nabla_w c(x,w)\lambda + \bar{\lambda} = 0 \tag{2.17b}$$

$$c(x) = 0 \tag{2.17c}$$

$$XVe = 0 \tag{2.17d}$$

$$w - p_0 = 0 \tag{2.17e}$$

In this definition $\bar{\lambda}$ represents the Lagrange multiplier corresponding to the equation $w - p_0 = 0$. For the Newton step we write:

$$\begin{bmatrix} W & \nabla_{xw}\mathcal{L}(x,w,\lambda,\nu) & A^T & -I & 0 \\ \nabla_{wx}\mathcal{L}(x,w,\lambda,\nu) & \nabla_{ww}\mathcal{L}(x,w,\lambda,\nu) & \nabla_w c(x,w)^T & 0 & I \\ A & \nabla_w c(x,w) & 0 & 0 & 0 \\ V & 0 & 0 & X & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta\lambda \\ \Delta\nu \\ \Delta\bar{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Delta p \end{bmatrix}. \tag{2.18}$$

Since $\Delta w = \Delta p$, the sensitivity computed by this matrix (without the second row) is the same as the optimal sensitivity stated in (2.12). These sensitivity features are embedded in the software *sIPOPT* [67, 68].

#### 2.2.2.1 Active Set Changes

From Property 1, existence of $ds^*/dp$ requires SSOSC, SC and LICQ. Nevertheless, directional derivatives can still be obtained even if these assumptions are relaxed. This issue becomes important if we want to approximate NLP solutions that result from perturbations ($\Delta p = p_f - p_0$) that lead to active set changes for the perturbed solution of (2.1). When $\Delta p$ provokes an active set change, a positive variable may become active at zero or a zero variable may need to become positive. Moreover, even if LICQ and SC hold at

the solution at $p_0$, $\Delta p$ may be too large to maintain the same active set for the perturbed estimate. This case requires special treatment in the sensitivity calculation.

NLP sensitivity with active set changes has been considered through the stepwise application of (2.12); this was developed and described in [53, 69]. On the other hand, a natural extension of (2.12) to deal with active set changes, is through the solution of the quadratic programming problem [60, 61, 64, 70]:

$$\min_{\Delta x} \quad \Phi = \Delta x^T \nabla_{xp}\mathcal{L}\left(s^*(p_0); p_0\right)\Delta p + \frac{1}{2}\Delta x^T \nabla_{xx}\mathcal{L}\left(s^*(p_0); p_0\right)\Delta x \tag{2.19a}$$

$$\text{s.t.} \quad \nabla_p c(x^*; p_0)^T \Delta p + \nabla_x c(x^*; p)^T \Delta x = 0, x^* + \Delta x \geq 0 \tag{2.19b}$$

To justify this approach, we consider the following property due to Kyparsis [54], which allows the calculation of directional derivatives.

**Property 2.4** (Directional Derivatives)**:** *Suppose that MFCQ holds at the solution of* (2.1) *and that SSOSC holds for* all *values of multipliers that satisfy the KKT conditions for* (2.1) *(i.e., in* $P(x^*; p_0)$*). Also, assume that for any subset of the active constraints at the solution of* (2.1)*, the rank of these constraint gradients remains constant near* $(x^*; p_0)$*. and* $K_x$ *is the set of extreme points of the multiplier values in* $P(x^*; p_0)$*. Under these conditions, the following quadratic program:*

$$\min_{\Delta x} \quad \Phi = \Delta x^T \nabla_{xp}\mathcal{L}\left(s(p_0)\right)\Delta p + \frac{1}{2}\Delta x^T \nabla_{xx}\mathcal{L}\left(s(p_0)\right)\Delta x \tag{2.20a}$$

$$\textit{s.t.} \quad \nabla_p c(x^*; p_0)^T \Delta p + \nabla_x c(x^*; p_0)^T \Delta x = 0, \Delta x_j \geq 0 \textit{ for } x_j^* = 0 \tag{2.20b}$$

$$\Delta x_j = 0, \textit{ for some } \nu_j^* > 0 \in K_x \tag{2.20c}$$

*uniquely determines the directional derivative* $\Delta x$ *that corresponds to* $\Delta p$*.*

This property provides the weakest conditions under which a unique directional derivative can be shown [54, 55]. Therefore we note that if LICQ is not satisfied at $s(p_0)$ but the assumptions of Property 2.4 hold, then two observations hold for (2.19) and (2.20).

- The solution, $\Delta x_K$, of (2.20) (with $\|\Delta p\|$ sufficiently small) is also feasible for (2.19).
- For the solution, $\Delta x_{QP}$ of (2.19), we have $\Phi(\Delta x_K) \geq \Phi(\Delta x_{QP})$, and therefore, $\Delta x_{QP}$ provides the optimal first order perturbation for $\Delta p$.

Moreover, if the NLP (2.1) satisfies LICQ at $p_0$, for $p = p_0 + t\,(p_f - p_0)\,, t \in (0, 1]$, we note that there exists a $t \in (0, 1]$ for which LICQ no longer applies at the solution of the NLP with $p = p_0 + t\Delta p$, due to an active set change. Nevertheless, we can still apply (2.19) directly, by considering the following stepwise application of the sensitivity analysis. We first consider a QP for $\Delta p_1 = p_1 - p_0$ with $p_1$ corresponding to a solution where the active set of the QP changes and LICQ no longer holds. Next, we solve the QP (2.19) at $p_1$ with $\Delta p_2 = p_f - p_1$, and all other quantities unchanged. Adding the solutions of these two QPs, i.e., $\Delta x = \Delta x_1 + \Delta x_2$, can be shown to be feasible for (2.19). Therefore, for this case, the solution of (2.19) provides an optimal first order perturbation for $\Delta p$.

#### 2.2.2.2 Fix-Relax Strategy

The current version of *sIPOPT* does not include an implementation of (2.19). Nevertheless, from the solution of the barrier problems, *sIPOPT* provides the elements of a QP-based sensitivity approach through the use of a "fix-relax strategy" which accounts for active set changes [71, 72]. This is illustrated for two cases.

- When the perturbed variable violates its bound, i.e., $x_i = x_i^* + \Delta x_i < 0$, an additional condition is introduced that sets the perturbed variable $x_i$ to its bound (i.e., we write $E_x^T\,(\Delta x + x^*) = 0$). At the same time, the corresponding complementarity condition in (2.12) has to be relaxed with the addition of a new variable $\Delta\bar{\nu}$.

$$\left[\begin{array}{cccc} W & A^T & -I_n & 0 \\ A & 0 & 0 & 0 \\ V & 0 & X & E_x \\ \hline E_x^T & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta\lambda \\ \Delta\nu \\ \hline \Delta\bar{\nu} \end{array}\right] = - \left[\begin{array}{c} \nabla_{px}\mathcal{L}^T\Delta p \\ \nabla_p c^T \Delta p \\ 0 \\ \hline E_x^T x^* \end{array}\right] \tag{2.21}$$

- Similarly in case a perturbed bound multiplier becomes negative because of the new step, the bound multiplier is set to zero (i.e., $E_\nu \Delta (\nu + \nu^*) = 0$), the complementarity condition has to be relaxed, and again a new variable $\Delta\bar{\nu}$ is added.

$$\left[\begin{array}{cccc} W & A^T & -I_n & 0 \\ A & 0 & 0 & 0 \\ V & 0 & X & E_\nu \\ \hline 0 & 0 & E_\nu^T & 0 \end{array}\right] \left[\begin{array}{c} \Delta x \\ \Delta\lambda \\ \Delta\nu \\ \hline \Delta\bar{\nu} \end{array}\right] = - \left[\begin{array}{c} \nabla_{px}\mathcal{L}^T \Delta p \\ \nabla_p c^T \Delta p \\ 0 \\ \hline E_\nu^T \nu^* \end{array}\right] \tag{2.22}$$

With suitable definition of $E$, $r_s$ and $r_1$, the systems (2.21) and (2.22) can be written in the following form:

$$\left[\begin{array}{c|c} K^* & E \\ \hline E^T & 0 \end{array}\right] \left[\begin{array}{c} \Delta s \\ \hline \Delta\bar{\nu} \end{array}\right] = - \left[\begin{array}{c} r_s \\ \hline r_1 \end{array}\right] \tag{2.23}$$

and can be solved using a Schur decomposition for $\Delta s$. This system can be solved in two steps by defining a Schur Matrix $C$:

$$C = -E^T K^{*-1} E \tag{2.24}$$

and solving two linear systems

$$C\Delta\bar{\nu} = E^T K^{*-1} r_s - r_1 \tag{2.25}$$

$$K^* \Delta s = -\left(r_s + E\Delta\bar{\nu}\right). \tag{2.26}$$

An example for relaxing a bound is given by [71], and the process of activating a bound is illustrated in Section 2.2.2.5.

#### 2.2.2.3 General Upper and Lower Bounds

For simplicity the derivations above were described using problems with only lower bounds. However, as shown in [50], the extension for the case with general upper and lower bounds

is straightforward. For this case, the barrier function will be

$$\min \; B(x;p,\mu) = f(x;p) - \mu \sum_{i=1}^{n} ln\left(x_i - x_i^L\right) - \mu \sum_{i=1}^{n} ln\left(x_i^U - x_i\right) \tag{2.27a}$$

$$\text{s.t.} \; c(x;p) = 0. \tag{2.27b}$$

For this case we define the following quantities:

$$M\left(s\left(\mu;p\right)\right) = \begin{bmatrix} W\left(s\left(\mu;p_0\right)\right) & A\left(x\left(\mu;p_0\right)\right)^T & -I & I \\ A\left(x\left(\mu;p_0\right)\right) & 0 & 0 & 0 \\ V_L\left(\mu;p_0\right) & 0 & X - X_L & 0 \\ V_U\left(\mu;p_0\right) & 0 & 0 & X_U - X \end{bmatrix} \tag{2.28}$$

and

$$N_p\left(s\left(\mu;p\right)\right) = \begin{bmatrix} \nabla_{xp}\mathcal{L}\left(s\left(\mu;p_0\right)\right)^T \\ \nabla_p c\left(x\left(\mu;p_0\right)\right)^T \\ 0 \\ 0 \end{bmatrix}, \quad N_\mu = \begin{bmatrix} 0 \\ 0 \\ -\mu e \\ -\mu e \end{bmatrix}, \tag{2.29}$$

where $\nu^L$ and $\nu^U$ are bound multipliers associated with the lower and upper bounds respectively. Also, here we define $X_L = diag\left\{x^L\right\}$, $X_U = diag\left\{x^U\right\}$, $V_L = diag\left\{\nu^L\right\}$ and $V_U = diag\left\{\nu^U\right\}$, and we use the same definitions for $W\left(s\left(\mu;p_0\right)\right)$, $A\left(x\left(\mu;p_0\right)\right)$ and $X$ as in (2.10) and (2.11). Thus, with Equations (2.28), (2.29), along with (2.14) or (2.15) we can obtain the perturbed update when there are both upper and lower bounds. Moreover, since IPOPT handles both upper and lower bounds, this is the way *sIPOPT* calculates the updates.

#### 2.2.2.4 Multiple Sequential Parameter Perturbations

In the derivations in the previous sections we considered changes to the parameter vector. However, in some cases we may be interested in making multiple parameter perturba-

tions in a sequential manner. For example we may want to perturb the current solution $s(\mu; p_0)$ using the parameter vectors $p_1, \ldots, p_{n_{\text{pert}}}$. This amounts to solving system (2.12) with different right hand sides $N_p(s(\mu; p_0))$ (Eq. (2.11)). Note that, because we already have (2.10) factorized at the solution, it is cheap to obtain the $n_{\text{pert}}$ sensitivities. From these and Equation (2.14) (or (2.15)) we can determine the approximated solutions $s(\mu; p_1)$, ..., $s\left(\mu; p_{n_{\text{pert}}}\right)$.

#### 2.2.2.5 Sensitivity Example

To conclude this section we consider a small parametric optimization problem from [60] and also considered in [71]. Here we discuss in detail the updating procedure, and also cover a change in the active set. This problem illustrates the capabilities of the fix-relax strategy described above.

Consider the parametric programming problem:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 + x_3^2 \\ \text{s.t.} \quad & 6x_1 + 3x_2 + 2x_3 - p_1 = 0 \\ & p_2 x_1 + x_2 - x_3 - 1 = 0 \\ & x_1, x_2, x_3 \geq 0, \end{aligned} \tag{2.30}$$

with variables $x_1, x_2$, and $x_3$ and parameters $p_1$, and $p_2$. As programmed, the IPOPT code does not distinguish variables from parameters, but the problem can be reformulated as (2.16) by introducing equations that fix the parameters $p_1, p_2$ to their nominal values $p_{1,a}, p_{2,a}$.

$$
\begin{aligned}
\min \quad & x_1^2 + x_2^2 + x_3^2 && \text{(2.31a)}\\
\text{s.t.} \quad & 6x_1 + 3x_2 + 2x_3 - p_1 = 0 && \text{(2.31b)}\\
& p_2 x_1 + x_2 - x_3 - 1 = 0 && \text{(2.31c)}\\
& p_1 = p_{1,a} && \text{(2.31d)}\\
& p_2 = p_{2,a} && \text{(2.31e)}\\
& x_1, x_2, x_3 \geq 0. && \text{(2.31f)}
\end{aligned}
$$

The KKT conditions for this problem are

$$
\begin{aligned}
2x_1 + 6\lambda_1 + p_2\lambda_2 - \nu_1 &= 0 && \text{(2.32a)}\\
2x_2 + 3\lambda_1 + \lambda_2 - \nu_2 &= 0 && \text{(2.32b)}\\
2x_3 + 2\lambda_1 - \lambda_2 - \nu_3 &= 0 && \text{(2.32c)}\\
-\lambda_1 + \lambda_3 &= 0 && \text{(2.32d)}\\
\lambda_2 x_1 + \lambda_4 &= 0 && \text{(2.32e)}\\
6x_1 + 3x_2 + 2x_3 - p_1 &= 0 && \text{(2.32f)}\\
p_2 x_1 + x_2 - x_3 - 1 &= 0 && \text{(2.32g)}\\
p_1 - p_{1,a} &= 0 && \text{(2.32h)}\\
p_2 - p_{2,a} &= 0 && \text{(2.32i)}\\
\nu_1 x_1 - \mu &= 0 && \text{(2.32j)}\\
\nu_2 x_2 - \mu &= 0 && \text{(2.32k)}\\
\nu_3 x_3 - \mu &= 0 && \text{(2.32l)}\\
x_1, x_2, x_3, \nu_1, \nu_2, \nu_3 &\geq 0. && \text{(2.32m)}
\end{aligned}
$$

The corresponding Newton step is

$$
\begin{bmatrix}
2 & & & & \lambda_2 & 6 & p_2 & & -1 & & & \\
 & 2 & & & & 3 & 1 & & & -1 & & \\
 & & 2 & & & 2 & -1 & & & & -1 & \\
 & & & & & -1 & & 1 & & & & \\
\lambda_2 & & & & & & x_1 & & 1 & & & \\
6 & 3 & 2 & -1 & & & & & & & & \\
p_2 & 1 & -1 & & x_1 & & & & & & & \\
\nu_1 & & & & & & & & x_1 & & & \\
 & \nu_2 & & & & & & & & x_2 & & \\
 & & \nu_3 & & & & & & & & x_3 & \\
 & & & 1 & & & & & & & & \\
 & & & & 1 & & & & & & &
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta p_1 \\ \Delta p_2 \\ \Delta \lambda_1 \\ \Delta \lambda_2 \\ \Delta \nu_1 \\ \Delta \nu_2 \\ \Delta \nu_3 \\ \Delta \lambda_3 \\ \Delta \lambda_4
\end{bmatrix}
= -
\begin{bmatrix}
2x_1^* + 6\lambda_1^* + p_2\lambda_2^* - \nu_1^* \\
2x_2^* + 3\lambda_1^* + \lambda_2^* - \nu_2^* \\
2x_3^* + 2\lambda_1^* - \lambda_2^* - \nu_3^* \\
-\lambda_1^* + \lambda_3^* \\
\lambda_2^* x_1^* + \lambda_4^* \\
6x_1^* + 3x_2^* + 2x_3^* - p_1^* \\
p_2^* x_1^* + x_2^* - x_3^* - 1 \\
\nu_1^* x_1^* - \mu \\
\nu_2^* x_2^* - \mu \\
\nu_3^* x_3^* - \mu \\
p_1^* - p_{1,a} \\
p_2^* - p_{2,a}
\end{bmatrix}
\tag{2.33}
$$

where the right hand side is zero at the solution (note that the ordering of the variables is now as in Eq. (2.18)). Also, note that this Newton step can easily be transformed into (2.18) by rearranging rows and columns. Now consider the exact solutions of two neighboring parameter sets $p_a = [p_{1,a}, p_{2,a}] = [5, 1]$ and $p_b = [4.5, 1]$. The corresponding NLP solutions are

$$s(p_a) = [0.6327, 0.3878, 0.0204, 5, 1, | -0.1633, -0.2857, |0, 0, 0, | -0.1633, 0.1808] \,,$$

and

$$s(p_b) = [0.5, 0.5, 0, 4.5, 1, |0, -1, |0, 0, 1, |0, 0.5] \,.$$

Clearly, there is a change in the active set, when changing the parameters from $p_a$ to $p_b$. This is easily verified from the decrease of $x_3$ to zero. When using $p_b$ the bound is active, while it is inactive when the parameters are set to $p_a$. This change in the active

set is not captured by the linearization of the KKT conditions. For example, using (2.14) would require that we first solve the linear system (2.13), and use the solution to update the values of the variables. In this example we would solve the linear system

$$\begin{bmatrix}
2 & & & & \lambda_2 & 6 & p_2 & -1 & & & & \\
 & 2 & & & & 3 & 1 & & -1 & & & \\
 & & 2 & & & 2 & -1 & & & -1 & & \\
 & & & & & -1 & & & & & 1 & \\
\lambda_2 & & & & & & x_1 & & & & & 1 \\
6 & 3 & 2 & -1 & & & & & & & & \\
p_2 & 1 & -1 & & x_1 & & & & & & & \\
\nu_1 & & & & & & & x_1 & & & & \\
 & \nu_2 & & & & & & & x_2 & & & \\
 & & \nu_3 & & & & & & & x_3 & & \\
 & & & 1 & & & & & & & & \\
 & & & & 1 & & & & & & &
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta p_1 \\ \Delta p_2 \\ \Delta \lambda_1 \\ \Delta \lambda_2 \\ \Delta \nu_1 \\ \Delta \nu_2 \\ \Delta \nu_3 \\ \Delta \lambda_3 \\ \Delta \lambda_4
\end{bmatrix}
= -
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \Delta p_1 \\ 0
\end{bmatrix}, \tag{2.34}$$

where $\Delta p_1 = p_{1,a} - p_{1,b} = 5 - 4.5$. This Newton step yields an updated iterate of

$$s(p_b) = [0.5765, 0.3775, -0.0459, 4.5, 1, | -0.1327, -0.3571, |0, 0, 0, | -0.1327, 0.2099] + o(\|\Delta p\|) \tag{2.35}$$

which violates the bounds on $x_3$. On the other hand, taking into consideration this active set change, we use (2.21) and augment the KKT system fixing the variable to the bound, and relaxing the complementarity condition. The Newton step is now

$$\left[\begin{array}{cccccccccccc|c}
2 & & & & \lambda_2 & 6 & p_2 & -1 & & & & & \\
 & 2 & & & & 3 & 1 & & -1 & & & & \\
 & & 2 & & & 2 & -1 & & & -1 & & & \\
 & & & & & -1 & & & & & 1 & & \\
\lambda_2 & & & & & & x_1 & & & & & 1 & \\
6 & 3 & 2 & -1 & & & & & & & & & \\
p_2 & 1 & -1 & & x_1 & & & & & & & & \\
\nu_1 & & & & & & & x_1 & & & & & \\
 & \nu_2 & & & & & & & x_2 & & & & \\
 & & \nu_3 & & & & & & & x_3 & & 1 & \\
 & & & 1 & & & & & & & & & \\
 & & & & 1 & & & & & & & & \\
\hline
 & & 1 & & & & & & & & & & 
\end{array}\right]
\left[\begin{array}{c}
\Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta p_1 \\ \Delta p_2 \\ \Delta \lambda_1 \\ \Delta \lambda_2 \\ \Delta \nu_1 \\ \Delta \nu_2 \\ \Delta \nu_3 \\ \Delta \lambda_3 \\ \Delta \lambda_4 \\ \hline \Delta \bar{\nu}_3
\end{array}\right]
= -
\left[\begin{array}{c}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \delta p_1 \\ 0 \\ \hline x_3^*
\end{array}\right], \tag{2.36}$$

which yields an updated iterate of

$$s(p_b) = [0.5, 0.5, 0, 4.5, 1, |0, -1, |0, 0, 0, |0, 0.5948] + o(\|\Delta p\|), \tag{2.37}$$

and this is a very good approximation to the optimal solution to the problem with problem data $p_b$. Some differences are expected for the linear system, as $\lambda_4, x_1$ and $\lambda_2$ appear in the nonlinear constraint (2.32e) in the KKT conditions.

### 2.2.3 Extraction of Reduced Hessian Information

An important byproduct of the sensitivity calculation is information related to the Hessian of the Lagrange function pertinent to the second order conditions. At the solution of (2.1) we consider a sensitivity system, $MS = N_{rh}$, with $M$ defined in (2.10), and partition the variables into free and bounded variables, i.e., $x^* = [x_f^T \; x_b^T]$ where $x_f^* > 0, x_b^* = 0$. Assum-

ing strict complementarity (SC), the IFT sensitivity system using (2.10) can be partitioned with:

$$M = \begin{bmatrix} W_{ff}(x^*,\lambda^*) & W_{fb}(x^*,\lambda^*) & A_f(x^*)^T & -\mathbb{I}_f & 0 \\ W_{bf}(x^*,\lambda^*) & W_{bb}(x^*,\lambda^*) & A_b(x^*)^T & 0 & -\mathbb{I}_b \\ A_f(x^*) & A_b(x^*) & 0 & 0 & 0 \\ 0 & 0 & 0 & X_f^* & 0 \\ 0 & V_b^* & 0 & 0 & 0 \end{bmatrix}, S = \begin{bmatrix} S_{x_f} \\ S_{x_b} \\ S_\lambda \\ S_{\nu_f} \\ S_{\nu_b} \end{bmatrix}, \text{ and } N_{rh} = \begin{bmatrix} \bar{E} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.38}$$

where $\bar{E}$ is defined below. From (2.38) it is easy to see that $S_{x_b} = 0, S_{\nu_f} = 0$. These variables and the last two rows can therefore be removed, leading to:

$$\begin{bmatrix} W_{ff}(x^*,\lambda^*) & A_f(x^*)^T & 0 \\ A_f(x^*) & 0 & 0 \\ W_{bf}(x^*,\lambda^*) & A_b(x^*)^T & -\mathbb{I}_b \end{bmatrix} \begin{bmatrix} S_{x_f} \\ S_\lambda \\ S_{\nu_b} \end{bmatrix} = \begin{bmatrix} \bar{E} \\ 0 \\ 0 \end{bmatrix}$$

We now define $S_{xf} = \mathbb{Z}S_\mathbb{Z} + YS_Y$, with $A_f(x^*)\mathbb{Z} = 0$, and $A_f(x^*)Y$ and $R = [Y \mid \mathbb{Z}]$ nonsingular. Using the similarity transform, $H^T M H \tilde{S} = H^T N_{rh}$ with $H = \begin{bmatrix} R & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & 0 & \mathbb{I} \end{bmatrix}$ leads to:

$$\begin{bmatrix} Y^T W_{ff}(x^*,\lambda^*)Y & Y^T W_{ff}(x^*,\lambda^*)\mathbb{Z} & Y^T A_f(x^*)^T & 0 \\ \mathbb{Z}^T W_{ff}(x^*,\lambda^*)Y & \mathbb{Z}^T W_{ff}(x^*,\lambda^*)\mathbb{Z} & 0 & 0 \\ A_f(x^*)Y & 0 & 0 & 0 \\ W_{bf}(x^*,\lambda^*)Y & W_{bf}(x^*,\lambda^*)\mathbb{Z} & A_b(x^*)^T & -\mathbb{I}_b \end{bmatrix} \begin{bmatrix} S_Y \\ S_\mathbb{Z} \\ S_\lambda \\ S_{\nu_b} \end{bmatrix} = \begin{bmatrix} Y^T\bar{E} \\ \mathbb{Z}^T\bar{E} \\ 0 \\ 0 \end{bmatrix}. \tag{2.39}$$

From (2.39) we have $S_Y = 0$ and $S_\mathbb{Z} = (\mathbb{Z}^T W_{ff}\mathbb{Z})^{-1}\mathbb{Z}^T E$. Choosing $\mathbb{Z}^T\bar{E} = \mathbb{I}$ reveals $S_\mathbb{Z}$ as the inverse of the reduced Hessian matrix $\mathcal{H}_\mathcal{R} = \mathbb{Z}^T W_{ff}\mathbb{Z}$. Convenient choices for this transform arise from partitioning $x_f^T = \begin{bmatrix} x_D^T & x_I^T \end{bmatrix}$ with dependent and independent

variables $x_D \in \mathbb{R}^m$, $x_I \in \mathbb{R}^{n_I}$, so that $A_f = [A_D \mid A_I]$ with $A_D$ square and nonsingular. This leads to $Y^T = [\mathbb{I}_m \mid 0]$ and $\mathbb{Z}^T = \left[-A_I^T A_D^{-T} \mid \mathbb{I}_{n_I}\right]$.

Note that the transformation of the sensitivity system (2.39) need not be implemented. Instead, for a chosen set of $n_I \leq n_x - m$ independent variables, $A_D$ nonsingular, $\bar{E}^T = [0 \mid \mathbb{I}_{n_I}]$ and the matrices defined in (2.38), the reduced Hessian can be found directly by solving the following system

$$MS = N_{rh}. \tag{2.40}$$

The computation of $S$ from the above system can be achieved by solving a linear system for each column of $N_{rh}$. In other words, for the *i*-th column in $S$ we solve the linear system $MS = N_i$, where $N_i$ is the *i*-th column of $N_{rh}$. From the choice of $\mathbb{Z}$, $S_{x_f} = \mathbb{Z}S_{\mathbb{Z}}$ in (2.38) and $S_{\mathbb{Z}} = \mathcal{H}_{\mathcal{R}}$, the reduced Hessian can be extracted easily from the rows of $S$.

#### 2.2.3.1 Reduced Hessian Example

Here we illustrate the use of the methods described above to extract the reduced Hessian information from the KKT system at an optimal point. In this case we use a small example given by

$$\begin{aligned} \min\ & (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 \\ & \text{s.t. } x_1 + 2x_2 + 3x_3 = 0. \end{aligned} \tag{2.41}$$

The linearized KKT conditions of the above problem are given by

$$\begin{bmatrix} 2 & & & 1 \\ & 2 & & 2 \\ & & 2 & 3 \\ 1 & 2 & 3 & \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} r_{x_1} \\ r_{x_2} \\ r_{x_3} \\ r_\lambda \end{bmatrix}. \tag{2.42}$$

Note that for this example there are no bounds, and therefore the Hessian with respect to the free variables $W_{ff}$ is the same as the Hessian of the Lagrange function. If we choose

$x_D = x_3$ and $x_I^T = [x_1, x_2]$, then the rows in the solution of (2.42) that correspond to the rows of the reduced Hessian are those corresponding to $[\Delta x_1\ \Delta x_2]$.

To show this we first compute the reduced Hessian at the optimal point, which is given by

$$\mathcal{H}_{\mathcal{R}} = \mathbb{Z}^T W_{ff} \mathbb{Z} = \begin{bmatrix} -\frac{1}{3} & 1 & 0 \\ -\frac{2}{3} & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & & \\ & 2 & \\ & & 2 \end{bmatrix} \begin{bmatrix} -\frac{1}{3} & -\frac{2}{3} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2.2222 & 0.4444 \\ 0.4444 & 2.8889 \end{bmatrix},$$

and the inverted reduced Hessian matrix is

$$\mathcal{H}_{\mathcal{R}}^{-1} = \begin{bmatrix} 0.4643 & -0.0714 \\ -0.0714 & 0.3571 \end{bmatrix}. \tag{2.43}$$

From the partition of dependent and independent variables done above we can define $\bar{E}$ as

$$\bar{E} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

and using the columns of this matrix as the right hand side values in (2.42), we can compute the inverted reduced Hessian. Thus for the first column of $\mathcal{H}_{\mathcal{R}}^{-1}$ we have

$$\begin{bmatrix} 2 & & & 1 \\ & 2 & & 2 \\ & & 2 & 3 \\ 1 & 2 & 3 & \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.44}$$

which yields a solution vector of $S_1^T = \Delta x^T = [0.4643 - 0.0714 - 0.10710.0714]$, and similarly, if we use the second column of $\bar{E}$, we compute the other solution vector $S_2^T =$

$[-0.07140.3571 - 0.21430.1429]$. In other words we have that

$$S = \left[\begin{array}{cc} 0.4643 & -0.0714 \\ -0.0714 & 0.3571 \\ \hline -0.1071 & -0.2143 \\ 0.0714 & 0.1429 \end{array}\right]$$

where we note that the values of the inverted reduced Hessian are those of $S$ if we look at the rows associated with $x_I$. This is easily verified by comparing the values of $S$ above with those of (2.43).

# Chapter 3

# Bayesian State Estimation

In this chapter we discuss the problem of state estimation from the Bayesian perspective, in which we take into consideration prior (initial) data about the states and combine it with measurement information to generate posterior data of the states. In its most general form, the data used for prior, measurement, and posterior information comes in the form of probability density functions, and the state estimates are given by (some of) the moments of these distributions. For example, we may be interested in the mean (minimum variance estimate) or the mode (maximum likelihood estimate), and some means to characterize the goodness of the estimate. The latter of which can be obtained through the covariance matrix of the estimates, which can be used to determine confidence intervals that provide information on the error associated to the estimates.

In the following section we describe a general Bayesian framework, and in the rest of the sections we provide some details on some of the most common state estimation methods used. These descriptions are provided here for convenience since in later chapters they will be used to approximate important quantities required in Moving Horizon Estimation. This chapter is organized as follows: Section 3.1 describes the Bayesian Framework and the well known Kalman Filter (KF). In Section 3.2 a nonlinear extension of the KF is described, Sections 3.3, 3.4, and 3.5 describe the Unscented Kalman Filter, Particle Filters, and the Ensemble Kalman Filter, respectively. Finally, some concluding remarks are given in Section 3.6.

## 3.1 Bayesian Framework

The process in this case is modeled with the following set of equations:

$$z_{l+1} = f(z_l, u_l) + w_l \tag{3.1a}$$

$$y_l = h(z_l) + v_l, \tag{3.1b}$$

where $z_l \in \Re^{n_z}$ is the vector of state variables, $u_l \in \Re^{n_u}$ is the vector of known inputs, $y_l \in \Re^{n_y}$ is the vector of measurements or observations, $w_l \in \Re^{n_w}$ is the vector of unknown disturbances, $v_l \in \Re^{n_y}$ is the vector of measurement noise, $f\colon \Re^{n_z} \mapsto \Re^{n_z}$ is the system function, and $h\colon \Re^{n_z} \mapsto \Re^{n_y}$ is the measurement model. Note that $w_l$ and $v_l$ can be considered deterministic errors of unknown character, i.e., without any stochastic interpretation. However, here $w_l$ and $v_l$ are considered zero mean, independent, and normally distributed random variables. In other words, we have that $w_l \sim \mathcal{N}(0, Q_l)$, and $v_l \sim \mathcal{N}(0, R_l)$. Also, since the inputs $u_l$ are known we can use short hand notation for the state model, i.e., $z_{l+1} = f(z_l) + w_l$.

Since the state and measurements are affected by random variables the state estimate is best described with the conditional probability density function (PDF) $p(z_k \,|y_0, \ldots, y_l)$. Depending on the relationship between the time indexes $l$ and $k$ we can generate 3 types of estimates. This is illustrated in Figure 3.1, where we note that if the measurements are all in the past with respect to time $k$ then we are predicting the states; if $k = l$ we are filtering the states, and if the measurements are in the future with respect to $k$ then we are smoothing the states. For control and monitoring purposes we are usually interested in filtering and prediction, while for analyzing the past behavior of the system we use the smoothed state estimates. Finally, these estimates are summarized below:

1. If $l > k$ we obtain the smoothed state estimate,
2. If $l = k$ we obtain the filtered state estimate,
3. If $l < k$ we obtain the predicted state estimate.

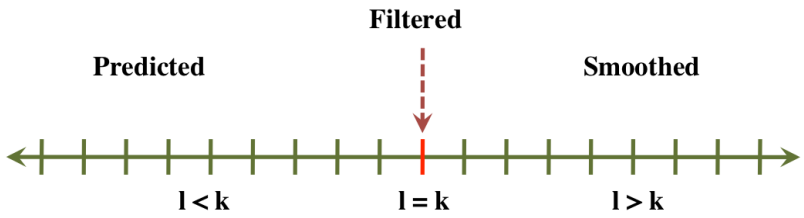



Figure 3.1: Illustration of the relationship between smoothing, filtering, and predicting.

Determining the full structure of this PDF for a general nonlinear system is usually not possible. However, it is possible to make some assumptions in order to simplify the problem. These assumptions are:

**Assumption 3.1:** *Assumptions on states and noise variables [15, 73].*

*(i) The states follow a first order Markov Process (i.e.,* $p\left(z_l\,|z_{l-1},\ldots,z_0\right)=p\left(z_l\,|z_{l-1}\right)$*),*

*(ii) The noises in the observations are independent of those of the states.*

*(iii) The measurements are mutually independent.*

*(iv) The noise variables have zero mean.*

In order to write the posterior PDF as a function of the prior information we apply Bayes' rule. We thus get

$$p\left(z_k\,|y_0,\ldots,y_k\right)=\frac{p\left(y_0,\ldots,y_k|z_k\right)p\left(z_k\right)}{p\left(y_0,\ldots,y_k\right)}= \tag{3.2a}$$

$$=\frac{p\left(y_k|y_0,\ldots,y_{k-1},z_k\right)p\left(y_0,\ldots,y_{k-1}|z_k\right)p\left(z_k\right)}{p\left(y_k|y_0,\ldots,y_{k-1}\right)p\left(y_0,\ldots,y_{k-1}\right)} \tag{3.2b}$$

$$p\left(z_k|y_0,\ldots,y_k\right)=\frac{p\left(y_k|z_k\right)p\left(z_k|y_0,\ldots,y_{k-1}\right)}{p\left(y_k|y_0,\ldots,y_{k-1}\right)}. \tag{3.2c}$$

Note that the information from the measurements is introduced through the likelihood $p\left(y_k|z_k\right)$, and that $p\left(y_k|y_0,\ldots,y_{k-1}\right)$ does not depend on the states. If we assume that the posterior distribution $p\left(z_{k-1}|y_0,\ldots,y_{k-1}\right)$ is known at sampling time $k$ then using the Chapman-Kolmogorov equation we can propagate the probability density function forward between sampling times to obtain the prior distribution $p\left(z_k|y_0,\ldots,y_{k-1}\right)$ [8].

$$p\left(z_k|y_0,\ldots,y_{k-1}\right) = \int p\left(z_k|z_{k-1}\right) p\left(z_{k-1}|y_0,\ldots,y_{k-1}\right) dz_{k-1} \tag{3.3}$$

Thus, using Equation (3.3) we can predict the prior probability density function, which we can then update applying Bayes' rule and the newest measurement (Eq. (3.2c)). These two steps, prediction and correction, are common between the filters that share a Bayesian origin. Furthermore, using Equations (3.2c) and (3.3) we can obtain a difference equation that describes the time evolution of the posterior density function [8].

$$p\left(z_k|y_0,\ldots,y_k\right) = \frac{p\left(y_k|z_k\right)\int p\left(z_k|z_{k-1}\right) p\left(z_{k-1}|y_0,\ldots,y_{k-1}\right) dz_{k-1}}{\int\int p\left(y_k|z_k\right) p\left(z_k|z_{k-1}\right) p\left(z_{k-1}|y_0,\ldots,y_{k-1}\right) dz_{k-1} dz_k} \tag{3.4}$$

Once the posterior distribution is found, the optimal state estimate can be obtained from minimizing a chosen loss function (Eq. (3.5)) [8, 73]. There are several types of loss functions available, but some of the most common are used to obtain the mean, median, or mode of the distribution. For example, the *minimum variance* state estimate is given by the conditional mean, and it can be evaluated with

$$\min \mathbb{E}\left[L\left(z_k\right)\right] = \int L\left(z_k\right) p\left(z_k|y_0,\ldots,y_k\right) dz_k, \tag{3.5}$$

where, for example, the loss function is $L\left(z_k\right) = \gamma_k^T S \gamma_k$. $S$ is a positive definite weight matrix, and $\gamma_k$ is the vector of errors between the predicted and true state. Minimizing this loss function will result in the minimum variance estimate (i.e., the conditional mean) [8].

Unfortunately, closed form solutions to (3.5) or explicit expressions for the Chapman-Kolmogorov Equation and the probability density functions used in Equation (3.3), cannot be determined easily for general nonlinear systems. Nevertheless, for some restrictive cases, such as linear Gaussian systems, it is possible to find expressions that allow us to propagate the full posterior distribution forward in time. For example, in the linear Gaussian case, the solution to (3.5) and also for the Maximum *a Posteriori* Probability (MAP) problem, yields the well known Kalman Filter [74]. This filter is summarized in the following theorem.

**Theorem 3.1** (Discrete time Kalman Filter (KF) [8])**:** *The optimal (minimum variance) filter for the discrete time linear system*

$$\begin{aligned} z_{k+1} &= A_k z_k + \Gamma_k w_k \\ y_k &= H_k z_k + v_k \end{aligned} \tag{3.6}$$

*consists of difference equations for the conditional mean and covariance matrix. Between observations predictions are generated with,*

$$\begin{aligned} z_{k+1|k} &= A_k z_{k|k} \\ M_{k+1|k} &= A_k P_{k|k} A_k^T + \Gamma_k Q_k \Gamma_k^T, \end{aligned} \tag{3.7}$$

*where $M_{k+1|k}$ is the prior covariance matrix. At observations, the predictions are corrected with*

$$\begin{aligned} z_{k|k} &= z_{k|k-1} + K_k \left[ y_k - H_k z_{k|k-1} \right] \\ P_{k|k} &= M_{k|k-1} - K_k H_k M_{k|k-1}, \end{aligned} \tag{3.8}$$

*where $P_{k|k}$ is the posterior covariance matrices, and where*

$$K_k = M_{k|k-1} H_k^T \left[ H_k M_{k|k-1}^{-1} H_k^T + R_k \right] \tag{3.9}$$

*is the Kalman gain.*

**Proof:** There are several well known proofs and derivations of this theorem. Some of them can be found in [7, 8, 74, 75]. □

For the previous theorem, the noise variables in system (3.6) are also white noise sequences where $Q_k$ and $R_k$ are the covariance matrices for the states and measurements, respectively. There are several ways to derive the equations in Theorem 3.1: through orthogonal projections, as a recursive solution of a least squares problem, etc [8]. However here we derive them using a more intuitive path. Assuming that the plant and measurements are given by Equations (3.6), then conditional mean is found with

$$z_{k+1|k} = \mathbb{E}\left[z_{k+1}\left|Y_0^k\right.\right] = \mathbb{E}\left[A_k z_k + B_k u_k + G_k w_k\left|Y_0^k\right.\right] = \tag{3.10a}$$

$$= A_k \mathbb{E}\left[z_k\left|Y_0^k\right.\right] + B_k u_k = \tag{3.10b}$$

$$= A_k z_{k|k} + B_k u_k \tag{3.10c}$$

where $z_{k|k} = \mathbb{E}\left[z_k\left|Y_0^k\right.\right]$ and $z_{k|k-1} = \mathbb{E}\left[z_k\left|Y_0^{k-1}\right.\right]$, $\mathbb{E}\left[w_k\left|Y_0^k\right.\right] = 0$, $Y_0^k = \{y_0, \ldots, y_k\}$, and for future reference, we define $Z_0^k = \{z_0, \ldots, z_k\}$. Furthermore, the prior covariance is determined by

$$M_{k+1|k} = \mathbb{E}\left[\left(z_{k+1} - z_{k+1|k}\right)\left(z_{k+1} - z_{k+1|k}\right)^T\left|Y_0^k\right.\right] = \tag{3.11a}$$

$$= \mathbb{E}\left\{\left[A_k(z_k - z_{k|k}) + G_k w_k\right]\left[A_k(z_k - z_{k|k}) + G_k w_k\right]^T\left|Y_0^k\right.\right\} \tag{3.11b}$$

$$= A_k P_{k|k} A_k^T + G_k Q_k G_k^T \tag{3.11c}$$

where $P_{k|k}$ is the posterior covariance matrix obtained with the following conditional expectation: $\mathbb{E}\left\{\left(z_k - z_{k|k}\right)\left(z_k - z_{k|k}\right)^T\left|Y_0^k\right.\right\}$. Also, we have that $Q_k = \mathbb{E}\left[w_k w_k^T\left|Y_0^k\right.\right]$, $R_k = \mathbb{E}\left[v_k v_k^T\right]$, and $\mathbb{E}\left[w_k v_k^T\left|Y_0^k\right.\right] = \mathbb{E}\left[v_k w_k^T\left|Y_0^k\right.\right] = 0$. This propagates the prior distribution forward in time avoiding the Chapman-Kolmogorov equation. When the new measurement becomes available we can correct the prior using the new information by applying the corrector equations:

$$z_{k+1|k+1} = z_{k+1|k} + K_{k+1}\left[y_{k+1} - H_k z_{k+1|k}\right] \tag{3.12a}$$

$$P_{k+1|k+1} = M_{k+1|k} - K_{k+1} H_{k+1} M_{k+1|k} \tag{3.12b}$$

$$K_{k+1} = M_{k+1|k} H_{k+1}^T \left(H_{k+1} M_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1} \tag{3.12c}$$

where $K_k$ is the Kalman gain. These equations can be derived by solving the maximum likelihood problem, for example, as shown in [7, 8, 76]. A derivation of these equations is also shown in Appendix A.

As mentioned above, the KF avoids the formation of the Chapman-Kolmogorov Equation, and provides the optimal solution to the minimum variance, and, for a linear system, also the MAP problem. For the latter case, the problem results in a large-scale least squares problem, for which the KF provides a recursive solution. This will be shown in the following chapters when we discuss the optimality conditions related to the Moving Horizon Estimation (MHE) problem. However, for general nonlinear systems it is not always possible to find the probability density functions associated to the correct noise structure of the system, and therefore some simplifications are required. In the following sections some state estimation methods are described which we will also use with MHE to make it more efficient.

## 3.2 Extended Kalman Filter

The Extended Kalman Filter (EKF) is probably the most common state estimation method used for nonlinear systems in industry [7]. This method assumes that the noise variables $w_k$ and $v_k$ are independent Gaussian random variables, even after propagation through the nonlinear system. Thus, only the mean and covariance are required to completely describe their distributions, as in the KF. Moreover, it assumes that the linearized model provides a good representation of the system which can be used to propagate the covariance of the noise. Also, the mean of the nonlinear function is approximated by evaluating the function using the mean of its arguments. In this way the prediction of the mean is done with

$$z_{k+1|k} = \mathbb{E}\left[f\left(z_k\right) + w_k \middle| Y_0^k\right] \approx f\left(\mathbb{E}\left[z_k \middle| Y_0^k\right]\right) + \mathbb{E}\left[w_k \middle| Y_0^k\right] \tag{3.13a}$$

$$= f\left(z_{k|k}\right), \tag{3.13b}$$

where $\mathbb{E}\left[w_k \middle| Y_0^k\right]$=0. The propagation of the covariance is done with Equation (3.11c), and the correction equations also remain the same. However, the following substitutions are done:

$$A_k = \nabla_{z_k} f\left(z_l, u_l\right)^T \Big|_{z_{k-1|k-1}, u_{k-1}} \tag{3.14a}$$

$$H_k = \nabla_{z_k} h\left(z_l\right)^T \Big|_{z_{k-1|k-1}} . \tag{3.14b}$$

The measurement update equations used are the same as in the KF (Eqs. (3.8)), but with the same substitutions for $A_k$ and $H_k$ used above.

The main drawbacks from using the EKF are the loss of information from using the linearized model, the assumption that Gaussianity will prevail even after nonlinear transformations (e.g., Eqs. (3.13)), and the lack of bounds enforced during the estimation process. These can cause performance problems at best, and at worse the state estimates can diverge from the true values. For example, Haseltine and Rawlings [6], shows a simple two state isothermal gas reactor (see Appendix B.1), where the state estimates become negative. These states represent partial pressures, and therefore, negative values are not physically possible. Moreover, they show that this could be avoided using a constrained state estimator, such as MHE. Nevertheless, it is also possible to use ad-hoc methods for bound handling. For example, when a state estimate violates bounds, it can be projected onto the bound. However, these strategies yield suboptimal estimates, and are usually plagued with bad performance issues [9, 10, 77]. Finally, some of these issues can also be attributed to the inconsistent assumption that the noise structure remains Gaussian, even after nonlinear transformations. For example, this is illustrated in Figure 3.2 where we show a set of points that undergo a nonlinear transformation, and the propagated points are no longer Gaussian. In order to address this issue sample based filters are used. They make use of samples to approximate the distributions and make few assumptions on their structure. Some of these methods will be described below since they will also be used to improve on the moving horizon estimation methods derived in this thesis.

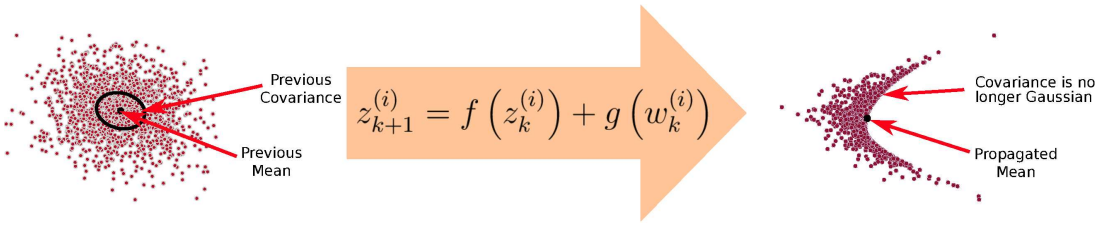



Figure 3.2: Nonlinear transformation of Gaussian samples.

## 3.3 Unscented Kalman Filter

The UKF is based on the idea that *"it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation"* [10, 12, 78]. Thus, the distribution is approximated using a set of deterministically chosen points called sigma points. These are selected so that they have a predetermined mean and covariance (e.g., $z_{k-1|k-1}$ and $P_{k-1|k-1}$), and they will be symmetrically distributed around the previous state estimate $z_{k-1|k-1}$.

The sigma points are generated using an augmented state vector $\zeta^T = [z_{k-1|k-1}^T,\ w_k^T,\ v_k^T]$ with mean $\bar{\zeta} \in \Re^{n_a}$ and covariance $\mathbb{P}_{k-1|k-1} = diag\left\{P_{k-1|k-1}, Q_k, R_k\right\}$ in the following way.

$$\mathcal{Z}_0 = \bar{\zeta} \tag{3.15a}$$

$$\mathcal{Z}_i = \bar{\zeta} + \left(\sqrt{(n_a+\kappa)\mathbb{P}_{k-1|k-1}}\right)_i \ \forall\ i = 1, \ldots, n_a \tag{3.15b}$$

$$\mathcal{Z}_{i+n_a} = \bar{\zeta} - \left(\sqrt{(n_a+\kappa)\mathbb{P}_{k-1|k-1}}\right)_i \ \forall\ i = 1, \ldots, n_a \tag{3.15c}$$

where $\kappa \in \Re$ is a tuning parameter that allows us to determine the spread of the sigma points around the mean $\bar{\zeta}$, $\left(\sqrt{(n_a+\kappa)\mathbb{P}_{k-1|k-1}}\right)_i$ means the $i$-th column of the matrix square root (usually obtained through Cholesky factorization), and $\mathcal{Z} \in \Re^{n_a \times (2n_a+1)}$ is a matrix that contains the $2n_a+1$ sigma points ($\mathcal{Z}_i$ refers to the $i$-th column of $\mathcal{Z}$). Once the sigma points are determined they are propagated forward using the nonlinear model of

the system (Eq. (3.1)) to determine a set of predicted states and measurements $z^{(i)}_{k|k-1}$ and $y^{(i)}_{k|k-1}$ (as illustrated in Figure 3.3), respectively. The weighted mean of these propagated points is used with the new measurement, when it becomes available, to determine the state estimate at the current time point. In other words, there are two steps in this method: first a prediction using the sigma points around the previous estimate and a correction using the means of the propagated points and the new measurement.

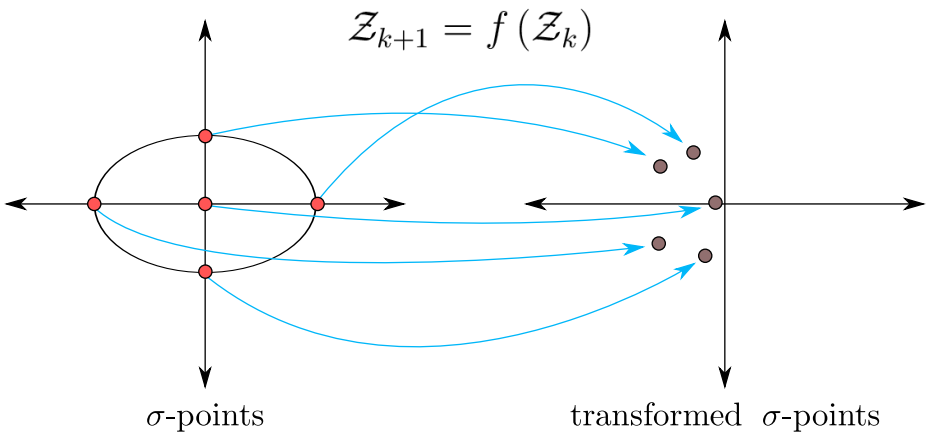



Figure 3.3: Illustration of propagation of sigma points using a nonlinear model.

For the prediction step we generate the sigma points around the previous state estimate $\hat{z}_{k-1|k-1}$ using Equation (3.15), and then they are propagated forward using the mathematical model of the system (3.16) to yield a set of predicted points ($z^{(i)}_{k|k-1}$ and $y^{(i)}_{k|k-1}$).

$$z^{(i)}_{k|k-1} = f\left(\zeta^{(i)}_z\right) + \zeta^{(i)}_w \;\forall\; i = 1, \ldots, 2n_a \tag{3.16a}$$

$$y^{(i)}_{k} = h\left(\zeta^{(i)}_z\right) + \zeta^{(i)}_v \;\forall\; i = 1, \ldots, 2n_a \tag{3.16b}$$

where $\zeta^{(i)}_z$, $\zeta^{(i)}_w$, and $\zeta^{(i)}_v$ correspond to the elements that refer to the states, the model noise, and measurement noise of the augmented vector $\zeta^{(i)}$ for the *i*-th sigma point. The weighted mean of the predicted states and measurements, as well as their covariance matrices, are obtained using equations (3.17) and (3.18) respectively.

$$z_{k|k-1} = \sum_{i=0}^{2n_a} W_i z_{k|k-1}^{(i)} \tag{3.17a}$$

$$\bar{y}_k = \sum_{i=0}^{2n_a} W_i y_k^{(i)} \tag{3.17b}$$

$$P_{k|k-1} = \sum_{i=0}^{2n_a} W_i \left(\bar{z}_{k|k-1} - z_{k|k-1}^{(i)}\right)\left(\bar{z}_{k|k-1} - z_{k|k-1}^{(i)}\right)^T \tag{3.18}$$

where $P_{k|k-1}$ is the prior covariance matrix of the states and the weights are determined with

$$W_0 = \frac{\kappa}{n_a + \kappa} \tag{3.19a}$$

$$W_i = \frac{1}{2(n_a + \kappa)} \; \forall \; i = 1, \ldots, 2n_a \tag{3.19b}$$

Once the new measurement becomes available the predicted mean and covariance can be updated. This is done in a similar way as in the EKF using expression (3.20).

$$K_k = P_{z\bar{y}} P_{\bar{y}\bar{y}}^{-1} \tag{3.20a}$$

$$z_{k|k} = z_{k|k-1} + K_k \left(y_k - \bar{y}_{k|k-1}\right) \tag{3.20b}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{\bar{y}\bar{y}} K_k^T \tag{3.20c}$$

where $P_{\bar{y}\bar{y}}$ and $P_{z\bar{y}}$ are determined as follows.

$$P_{\bar{y}\bar{y}} = \sum_{i=0}^{2n_a} W_i \left(y_{k|k-1}^{(i)} - \bar{y}_{k|k-1}\right)\left(y_{k|k-1}^{(i)} - \bar{y}_{k|k-1}\right)^T \tag{3.21a}$$

$$P_{z\bar{y}} = \sum_{i=0}^{2n_a} W_i \left(z_{k|k-1}^{(i)} - z_{k|k-1}\right)\left(y_{k|k-1}^{(i)} - \bar{y}_{k|k-1}\right)^T \tag{3.21b}$$

Also, note that UKF does not have a direct way to handle bounds. Similarly to EKF clipping can be performed, but this also inherits similar performance issues as described before.

## 3.4 Particle Filters

Particle Filters (PF) take advantage of Sequential Monte Carlo Sampling and integration techniques to approximate the distributions and expectations needed for the estimation problem. Because of this, particle filters are also known as Sequential Monte Carlo Filters. In general, PFs assume that the posterior distribution is represented by the weighted sum of $N_p$ samples [15, 16]:

$$p\left(z_k \left| Y_0^k \right.\right) \approx \frac{1}{N_p} \sum_{n=1}^{N_p} \delta\left(z_k - z_k^{(i)}\right) \equiv \hat{p}\left(z_k \left| Y_0^k \right.\right), \tag{3.22}$$

where $z_k^{(i)}$ are assumed to be independent and identically distributed particles drawn from $p\left(z_k \left| Y_0^k \right.\right)$. When $N_p$ is sufficiently large $\hat{p}\left(z_k \left| Y_0^k \right.\right)$ approximates the true posterior, and with this approximation we can estimate the expected value of a given nonlinear function $F\left(z_k\right)$,

$$\begin{aligned} \mathbb{E}\left[F\left(z_k\right)\right] &\approx \int F\left(z_k\right) \hat{p}\left(z_k \left| Y_0^k \right.\right) dz_k \\ &= \frac{1}{N_p} \sum_{n=1}^{N_p} \int F\left(z_k\right) \delta\left(z_k - z_k^{(i)}\right) dz_k \\ &= \frac{1}{N_p} \sum_{n=1}^{N_p} F\left(z_k^{(i)}\right) \end{aligned} \tag{3.23}$$

This requires sampling from a posterior distribution which can be a difficult task if we do not know the shape or type of the distribution. Instead, it is possible to sample from a trial or proposal distribution (also called importance distribution) from which we assume that we know all that is needed to define it [16]. At each sampling time, particles are drawn from the importance density $\pi\left(z_k \left| Y_0^k \right.\right)$, and the expectation approximation in Equation (3.23) is modified as follows:

$$\begin{aligned}\mathbb{E}\left[F\left(z_k\right)\right] &= \int F\left(z_k\right) p\left(z_k \left| Y_0^k\right.\right) dz_k \\ &= \int F\left(z_k\right) \frac{p\left(z_k \left| Y_0^k\right.\right)}{\pi\left(z_k \left| Y_0^k\right.\right)} \pi\left(z_k \left| Y_0^k\right.\right) dz_k \\ &\approx \frac{1}{N_p} \sum_{n=1}^{N_p} F\left(z_k^{(i)}\right) \omega_k^{(i)}\left(z_k\right)\end{aligned} \tag{3.24}$$

where

$$\omega_k^{(i)}\left(z_k^{(i)}\right) = \frac{p\left(z_k^{(i)} \left| Y_0^k\right.\right)}{\pi_i\left(z_k^{(i)} \left| Y_0^k\right.\right)} \tag{3.25}$$

is the weight function for the samples drawn from $\pi\left(z_k \left| Y_0^k\right.\right)$. This can be applied sequentially to update the weights at each sampling time. For that we assume that the chosen importance distribution has the following recursive form [73].

$$\begin{aligned}\pi\left(z_k \left| Y_0^k\right.\right) &= \pi\left(z_k | z_{k-1}, y_k\right) \pi\left(z_{k-1} | Y_0^{k-1}\right) \\ \pi\left(z_0\right) &\sim p\left(z_0\right)\end{aligned} \tag{3.26}$$

Thus, the weight function can be updated with [73]

$$\begin{aligned}\omega_k^{\star(i)}\left(z_k^{(i)}\right) = \frac{p\left(z_k^{(i)} \left| Y_0^k\right.\right)}{\pi_i\left(z_k^{(i)} \left| Y_0^k\right.\right)} &\propto \frac{p\left(y_k | z_k^{(i)}\right) p\left(z_k^{(i)} | z_{k-1}^{(i)}\right) p\left(Z_0^{k-1} | Y_0^{k-1}\right)}{\pi_i\left(z_k^{(i)} | Z_0^{k-1}, Y_0^{k-1}\right) \pi_i\left(Z_0^{k-1} | Y_0^{k-1}\right)} \\ &\propto \omega_{k-1}^{\star(i)}\left(z_{k-1}^{(i)}\right) \frac{p\left(y_k | z_k^{(i)}\right) p\left(z_k^{(i)} | z_{k-1}^{(i)}\right)}{\pi_i\left(z_k^{(i)} | Z_0^{k-1}, y_k\right)}\end{aligned} \tag{3.27}$$

Finally, the propagated weights are normalized with

$$\omega_k^{(i)}\left(z_k^{(i)}\right) = \frac{\omega_k^{\star(i)}\left(z_k^{(i)}\right)}{\sum_{j=1}^{N_p} \omega_k^{\star(j)}\left(z_k^{(j)}\right)} \tag{3.28}$$

The normalized weights can now be used with (3.24) to approximate the required moments.

We can now outline how the recursive Sequential Monte Carlo Filter works. We begin by assuming that $p\left(y_k|z_k^{(i)}\right)$ and $p\left(z_k^{(i)}|z_{k-1}^{(i)}\right)$ are known. This means that $w_k$ and $v_k$ from (3.1) are random variables with known distributions. The importance distribution can be obtained with the aid of another type of filter such as EKF, UKF, or an ensemble filter. For example if the EKF is chosen, the proposal distribution it is equivalent to

$$\pi_i\left(z_k^{(i)}\left|Y_0^k\right.\right) = \mathcal{N}\left(z_{k|k}^{(i)}, P_{k|k}^{(i)}\right) \tag{3.29}$$

where $z_{k|k}^{(i)}$ is the filtered state estimate for the *i*-th particle, and $P_{k|k}^{(i)}$ is the covariance associated with the posterior state estimate, both obtained from EKF. For this outline EKF was chosen for simplicity, however, this can be generalized to any other method as long as it generates enough information to determine $\pi_i\left(z_k^{(i)}\left|Y_0^k\right.\right)$. The filter is first initialized by drawing samples around $z_{0|0}$, assuming that this is a random variable with known statistics. At sampling time $k$, when a new measurement becomes available, EKF using $y_k$ is applied to each sample $z_{k-1|k-1}^{(i)}$, to generate the set of prior points $z_{k|k-1}^{(i)}$, a set of state estimates $z_{k|k}^{(i)}$, and their associated covariance matrices $P_{k|k}^{(i)}$, which is enough information to update the weights using (3.27) and (3.28). With these we can now determine the new state estimate given by the conditional mean evaluated with (3.24) using $F\left(z_{k|k}\right) = z_{k|k}$. The updated samples $z_{k|k}^{(i)}$ are used again at $k+1$ when the new measurement is obtained, and the process is repeated.

Unfortunately, samples updated this way suffer from a common problem called sample degeneracy (or weight degeneracy), where after some iterations most of the weights become negligible. This implies that most of the computational work is being applied to particles that have no weight in the approximation of the distributions. A simple way to remedy this is through the use of re-sampling that consists of drawing new samples from a weighted sample pool based on the sample weights at each iteration. More samples will

be drawn from the region with high weights, and therefore more information from the samples with more importance will be used [73]. There are several methods described for re-sampling [15, 16]; here we use the Residual Systematic Re-sampling (RSR) algorithm described by Arulampalam *et al.* [16]. The basic idea behind RSR is that at the end of the re-sampling step there will be multiple replicas of the samples with higher weights, therefore increasing their effect in the group. Therefore, after re-sampling the state estimate will be evaluated using (3.30) (where $z_k^{(i)}$ are the re-sampled set).

$$z_{k|k} = \mathbb{E}\left[z_k \left| Y_0^k \right.\right] = \frac{1}{N_p}\sum_{i=1}^{N_p} z_k^{(i)} \tag{3.30}$$

## 3.5 Ensemble Kalman Filters

The Ensemble Kalman Filter (EnKF) also belongs to the family of Monte Carlo Filters, and was first introduced by Evensen [13]. This method was developed to approximate the covariance propagation in EKF using randomly sampled particles instead of the commonly used Riccati Equation. Thus, EnKF is first initialized assuming that $z_{0|0}$ is a random variable with known statistics. It may be noted that the probability densities associated with the initial condition, the state disturbance, and the measurement noise can be arbitrary and need not be Gaussian. EKF is applied to each particle in the ensemble at any given sampling time, and between measurements the particles are propagated forward using the nonlinear model. Therefore the obtained prior (approximated with the samples in the ensemble) will not remain Gaussian as illustrated in Figure 3.2. Also, because each corrected particle comes from a different prediction, the posterior distribution approximated by the EnKF could be non-Gaussian. Notice that this method also consists of the prediction and correction steps of Bayesian Estimation, which it inherits from the EKF and is applied individually to each particle in the ensemble.

EnKF is first initialized assuming that $z_{0|0}$ is a random variable with known statistics (e.g., $z_{0|0} \sim \mathcal{N}(z_{0|0}, Q_0)$). $N_p$ samples are drawn for the initial condition, as well as for the

model and measurement noise variables to generate the sets of particles $z_{k|k}^{(i)}$, $w_k^{(i)}$, and $v_k^{(i)}$. These particles are propagated forward using the mathematical model of the system (Eq. (3.31)) which generates a set of predicted states and measurements.

$$z_{k|k-1}^{(i)} = f\left(z_{k-1|k-1}^{(i)}\right) + w_{k-1}^{(i)} \tag{3.31a}$$

$$y_k^{(i)} = h\left(z_k^{(i)}\right) + v_k^{(i)} \tag{3.31b}$$

The prior distribution is then represented by this ensemble. When the new measurement becomes available, each particle is corrected using Equations (3.12), where $P_{k+1|k}$ is the covariance of the ensemble. In this way, the posterior distribution will be approximated by the set of corrected points. The state estimate will be given by the mean of these corrected points, and the covariance associated to the state estimate will be given by the sample covariance.

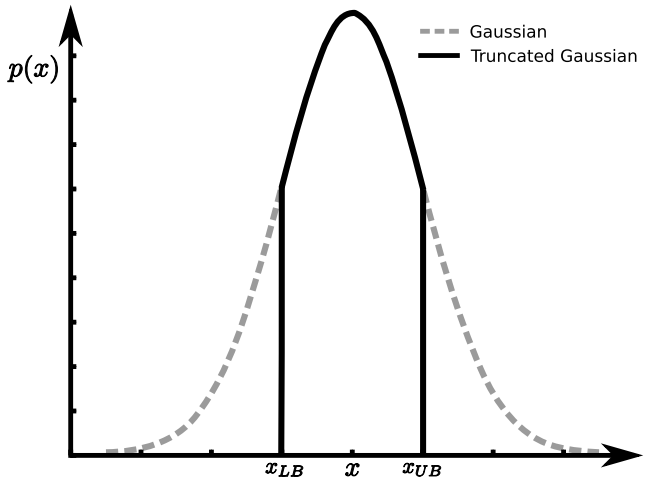



Figure 3.4: Comparison between the curves of a typical and a truncated Gaussian distribution.

A constrained version of the EnKF also exists called Constrained Ensemble Kalman Filter (CEnKF) [19]. It works in a very similar way as the EnKF with the following exceptions.

To satisfy bounds, the initial condition is now assumed to belong to a truncated Gaussian distribution. This distribution can be formed by setting the probability of the random variable outside of the bounds to zero as illustrated in Figure 3.4. The other main difference is in the measurement update of the samples. In this case, instead of using (3.12), a constrained Quadratic Program (QP) is solved for each particle as follows.

$$\hat{z}_{k|k}^{(i)} = \arg \min_{z_k} \left(z_{k|k-1}^{(i)} - z_k\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(z_{k|k-1}^{(i)} - z_k\right)^T + \left\{y_k - \left[h\left(z_k\right) + v_k^{(i)}\right]\right\}^T R_k^{-1} \left\{y_k - \left[h\left(z_k\right) + v_k^{(i)}\right]\right\} \tag{3.32a}$$

$$\text{s.t.} \quad z_{LB} \leq z_k^{(i)} \leq z_{UB} \tag{3.32b}$$

Again, the state estimate will be given by the sample mean of the corrected points, and its covariance can be evaluated from the sample covariance.

## 3.6 Conclusions

State estimation is an important aspect of process monitoring and process control because we do not always have the possibility to measure all of the states in the system. In some cases, some measurements can be obtained only through long laboratory assays, and thus the delays introduced by state estimation can potentially destabilize the system. All the state estimation methods described above deal with the fast available measurements in order to compute current estimates of the unmeasured states. However, these methods do not possess strategies to deal with bounds on the states in a systematic way, and thus may have performance issues such as large biases, or the estimated states may violate physical bounds (e.g., concentrations cannot be negative, temperatures must be in a specific operating region, etc.).

EKF is probably the most commonly used method in industry because of its simplicity. However, implementation of this method tends to be cumbersome due to the need of the Jacobian matrices used in the linearization. On the other hand, the assumptions made

in developing this method are inconsistent with the stochastic properties of the nonlinear system. Sample based methods are developed to deal with the latter. It is possible to represent the distribution of the states through particles, and therefore, approximate the moments of the PDFs through the samples. These methods also use the nonlinear model directly which helps provide a better representation of the probability density function. Unfortunately, neither sample based filters nor EKF can handle bounds in a systematic way, and thus these strategies are sensitive to errors in the measurements or disturbances.

In order to deal with such problems Moving Horizon Estimation was developed [5, 33]. MHE provides a means to do constrained state estimation in a systematic way, without many of the inconsistent assumptions done above. In the next chapter we will describe MHE in detail, and the methods proposed in this dissertation to make it faster and more robust to measurement errors.

# Chapter 4

# Moving Horizon Estimation

In this chapter we derive the Moving Horizon Estimator from the Bayesian perspective described in Chapter 3. Here we will show how MHE can be derived of as an approximation of the so-called full information problem, which uses all the available measurements (present and past) to estimate the trajectory of the states from the first sample time $l = 0$ to the current one $l = k$. In addition, we will show that MHE provides a good framework for constrained state estimation since at each sample time the estimates are computed by solving an NLP problem. This implies that bounds and constraints can be added, and they are handled systematically by the NLP solver. In this way we avoid the use of suboptimal approaches such as clipping. Furthermore, we are able to use the nonlinear system directly to avoid the loss of information due to linearization. Moreover, some discussion is given to highlight some of the strengths and weaknesses of these methods.

The rest of the chapter is organized as follows: in Section 4.1 the most commonly used MHE is derived, Sections 4.2 and 4.3 describe two other possible derivations of the estimator.

## 4.1 Derivation of MHE

Moving Horizon Estimation can be seen as an approximation of the full information problem where all the measurements are used to estimate the full trajectory of the states. In other words, we are interested in finding the conditional probability density function of the trajectory of the states $Z_0^k = \{z_0, \ldots, z_k\}$ given the measurements $Y_0^k = \{y_0, \ldots, y_k\}$. Thus we are interested in finding $p\left(Z_0^k \left| Y_0^k \right.\right)$. The derivation of the full information prob-

lem is done by applying Bayes' rule to this conditional density function. Thus we have

$$p\left(z_0,\ldots,z_k|y_0,\ldots,y_k\right)=\frac{p\left(y_0,\ldots,y_k|z_0,\ldots,z_k\right)p\left(z_0,\ldots,z_k\right)}{p\left(y_0,\ldots,y_k\right)}=\frac{p\left(Y_0^k\left|Z_0^k\right.\right)p\left(Z_0^k\right)}{p\left(Y_0^k\right)}, \quad (4.1)$$

where the PDF in the denominator is a scaling constant since it is not a function of the states (only of the known measurements). On the other hand, the terms in the numerator can be manipulated using the properties of PDFs and Assumptions 3.1, from the previous chapter. Thus taking the likelihood function $p\left(Y_0^k\left|Z_0^k\right.\right)$, and applying the product rule we have

$$p\left(Y_0^k\left|Z_0^k\right.\right)=p\left(y_0\left|Y_1^k,Z_0^k\right.\right)p\left(Y_1^k\left|Z_0^k\right.\right). \quad (4.2)$$

Note that measurements are independent of each other, and also, the measurement at time $l$ is only dependent of the state at the same sample time. Therefore, the above expression can be simplified to

$$\begin{aligned}p\left(Y_0^k\left|Z_0^k\right.\right)&=p\left(y_0\left|z_0\right.\right)p\left(Y_1^k\left|Z_0^k\right.\right)=p\left(y_0\left|z_0\right.\right)p\left(y_1\left|z_1\right.\right)p\left(Y_2^k\left|Z_0^k\right.\right)=\\&=\prod_{l=0}^{k}p\left(y_l\left|z_l\right.\right).\end{aligned} \quad (4.3)$$

On the other hand, the second term in the numerator of Equation (4.1) can also be simplified. In this case, using the product rule of PDFs we can rewrite that expression as follows

$$p\left(Z_0^k\right)=p\left(z_k\left|Z_0^{k-1}\right.\right)p\left(Z_0^{k-1}\right). \quad (4.4)$$

Note that the conditional PDF in the right hand side of Equation (4.4) can be simplified from Assumption 3.1-*(i)*, since the state sequence is a Markov chain, then the state at time $l$ only depends on the state at the previous time $(l-1)$. Therefore Equation (4.4) can be simplified to

$$p\left(Z_0^k\right) = p\left(z_k \left| Z_0^{k-1}\right.\right) p\left(Z_0^{k-1}\right) = p\left(z_k \left| z_{k-1}\right.\right) p\left(z_{k-1} \left| z_{k-2}\right.\right) p\left(Z_0^{k-2}\right) = \\ = p\left(z_0\right) \prod_{l=0}^{k-1} p\left(z_{l+1} \left| z_l\right.\right). \tag{4.5}$$

Substituting Equations (4.3) and (4.5) into (4.1), and defining $\mathcal{K}_1 = p\left(Y_0^k\right)^{-1}$ we get

$$p\left(Z_0^k \left| Y_0^k\right.\right) = \mathcal{K}_1 \, p\left(z_0\right) \left[\prod_{l=0}^{k-1} p\left(z_{l+1} \left| z_l\right.\right)\right] \left[\prod_{l=0}^{k} p\left(y_l \left| z_l\right.\right)\right], \tag{4.6}$$

where the expressions in the products can be related to the process and measurement models (i.e., Eqs. (3.1)) through the probability transfer function theorem (see Theorem C.1 in Appendix C). This implies that we can substitute those terms for the probability density functions associated to the noise variables, thus we obtain

$$p\left(Z_0^k \left| Y_0^k\right.\right) = \mathcal{K}_2 \, p\left(z_0\right) \left[\prod_{l=0}^{k-1} p\left(w_l\right)\right] \left[\prod_{l=0}^{k} p\left(v_l\right)\right], \tag{4.7}$$

where $\mathcal{K}_2$ includes the Jacobian determinant of the nonlinear transformation from Eq. (C.1), which in this case is constant. For a nonlinear system where the noise dynamics are not additive this term might not be constant and a more general treatment of these functions might be needed [79].

The full information problem assumes that the state estimate is given by most probable state, or the mode. Thus we generate the estimates by maximizing the probability density function (4.7) (or in this case its logarithm). This leads to the following expression,

$$\{\hat{z}_{0|k}, \ldots, \hat{z}_{k|k}\} = \arg \max_{\{z_0,\ldots,z_k\}} p\left(Z_0^k | Y_0^k\right) = \\ = \arg \max_{\{z_0,\ldots,z_k\}} \log p\left(z_0\right) + \sum_{l=0}^{l=k} \log p\left(v_l\right) + \sum_{l=0}^{l=k-1} \log p\left(w_l\right), \tag{4.8}$$

where the constant $\mathcal{K}_2$ has been removed since it does not affect the values of the state estimates. Note that in the expression above, the noise variables are functions of the states,

where the relationship comes from Theorem C.1. Finally, we know (from Assumptions 3.1) that the noise variables follow Gaussian distributions of the form $w_k \sim \mathcal{N}(0, Q_k)$ and $v_k \sim \mathcal{N}(0, R_k)$. Moreover, we also assume that the initial condition is a Gaussian random variable with normal distribution given by $z_0 \sim \mathcal{N}(\hat{z}_0, \Pi_0)$. Therefore, substituting the distributions and eliminating constant terms we can rewrite (4.8) as

$$\begin{aligned} \{\hat{z}_{0|k}, \ldots, \hat{z}_{k|k}\} = \arg \min_{\{z_0,\ldots,z_k\}} & \frac{1}{2} (z_0 - \hat{z}_0)^T \Pi_0^{-1} (z_0 - \hat{z}_0) + \\ & \frac{1}{2} \sum_{l=0}^{l=k} v_l^T R_l^{-1} v_l + \frac{1}{2} \sum_{l=0}^{l=k-1} w_l^T Q_l^{-1} w_l \\ \text{s.t. } & z_{l+1} = f(z_l) + w_l \\ & y_l = h(z_l) + v_l. \end{aligned} \tag{4.9}$$

From Equation (4.9) we can obtain optimal estimates of the evolution of the states $\{\hat{z}_{0|k}, \ldots, \hat{z}_{k|k}\}$. Upper and lower bounds on $z_l$, $w_l$, and $v_l$ can also be added to the problem in Eq. (4.9) to include information on the physical properties of the system (e.g., positive concentrations, mole fractions between 0 and 1, etc.). These bounds project the variables onto the feasible space thus yielding a truncated Gaussian distribution (illustrated in Figure 3.4) [5, 34]. Although Rao and Rawlings [34] warn against bounding the measurement noise variables because this may amplify the effects of outliers, adding bounds on the other variables will allow us to introduce pertinent information that we know about the system which could potentially help improve the quality of the state estimates.

The main drawback of Problem (4.9) is that it will grow as more measurements become available; it will grow up to a point when solving it between sample times will be impractical if not impossible. This is especially bad if the state estimates are intended for on-line monitoring or control since the introduced delays could destabilize the closed-loop system. However, it is possible to approximate the solution of the full information problem using ideas from Dynamic Programming [80]. Thus, following Rao [5], we derive the Moving Horizon Estimator. First we split the time into two sets $t_1 = \{l|\ 0 \leq l < k - N\}$ and

$t_2 = \{l|\ k-N \leq l \leq k\}$, where $N$ represents the number of simultaneous measurements we wish to consider (or horizon length), and then we rewrite the objective function of Problem (4.9) using standard dynamic programming methods to yield

$$\Psi(z_0, \ldots, z_{k+1}) = \Phi\left(z_{k-N}\right) + \frac{1}{2}\sum_{l=k-N}^{l=k} \left\|y_l - h\left(z_l\right)\right\|^2_{R_l^{-1}} + \frac{1}{2}\sum_{l=k-N}^{l=k-1} \left\|z_{l+1} - f\left(z_l, u_l\right)\right\|^2_{Q_l^{-1}}, \tag{4.10}$$

where the norms used here are defined as $\|x\|^2_A = x^T A x$ and $\Phi\left(z_{k-N}\right)$ is defined below. This is possible because the states, by assumption, are a Markov process, and therefore the quantity

$$\frac{1}{2}\sum_{l=k-N}^{l=k} \left\|y_l - h\left(z_l\right)\right\|^2_{R_l^{-1}} + \frac{1}{2}\sum_{l=k-N}^{l=k-1} \left\|z_{l+1} - f\left(z_l, u_l\right)\right\|^2_{Q_l^{-1}} \tag{4.11}$$

only depends on the state $z_{k-N}$, the noise sequence $\{w_l|l = k-N, \ldots, k-1\}$, and the measurements $\{y_l|l = k-N, \ldots, k\}$. Furthermore, the term $\Phi(z_{k-N})$ is given by

$$\Phi(z_{k-N}) = \left\{ \begin{array}{l} \min\limits_{z_0,\ldots,z_{k-N}} \dfrac{1}{2}\left(z_0 - \hat{z}_0\right)^T \Pi_0^{-1}\left(z_0 - \hat{z}_0\right) + \dfrac{1}{2}\sum\limits_{l=0}^{l=k-N-1} v_l^T R_l^{-1} v_l + \\ \qquad \dfrac{1}{2}\sum\limits_{l=0}^{l=k-N-1} w_l^T Q_l^{-1} w_l \\ \text{s.t. } z_{l+1} = f\left(z_l\right) + w_l \\ \qquad y_l = h\left(z_l\right) + v_l \end{array} \right\}, \tag{4.12}$$

and is known as the *arrival cost*. This term represents all the information not considered in the horizon, and from the stochastic point of view it represents the conditional probability function of the prior (or initial condition) information. In other words $\Phi(z_{k-N})$ is directly related to $p\left(z_{k-N}\left|Y_0^{k-N-1}\right.\right)$. In Figure 4.1 we illustrate the relationship between the arrival cost and the rest of the terms in Problem (4.10).

On the other hand, note that the objective function of NLP (4.10) only considers the states, noise variables, and measurements in the horizon (i.e., the last $N$ in the sequence).

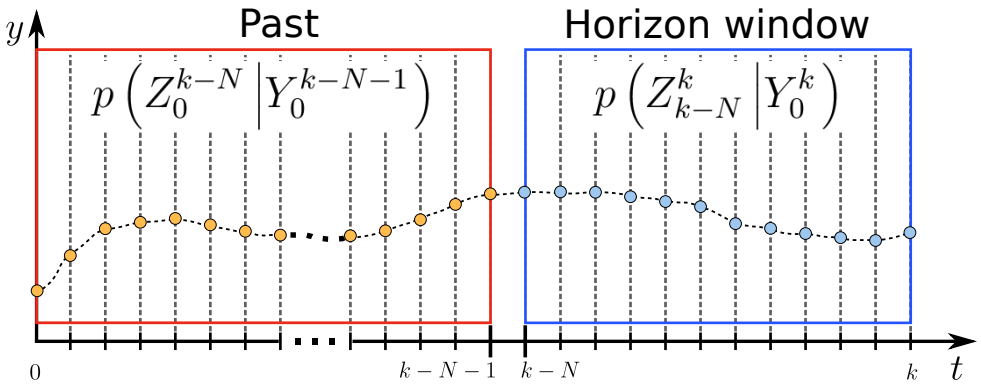



Figure 4.1: Illustration of how the information is split between the arrival cost and the horizon window in MHE.

Therefore, the NLP that we need to solve has been reduced in size, provided that we can supply the arrival cost term or an approximation of it. As time moves forward, when a new measurement becomes available it is appended to the horizon window, and the first measurement is dropped. This is illustrated in Figure 4.2. Furthermore, it is important to note that the arrival cost term will be changing as time moves forward. This is due to the fact that the initial condition will be changing as the horizon window moves forward. Thus, the most common MHE formulation is given by

$$\begin{aligned}
&\min_{z_{k-N},\ldots,z_k} \Phi(z_{k-N}) + \frac{1}{2}\sum_{l=k-N}^{l=k} v_l^T R_l^{-1} v_l + \frac{1}{2}\sum_{l=k-N}^{l=k-1} w_l^T Q_l^{-1} w_l \\
&\text{s.t. } z_{l+1} = f(z_l) + w_l \\
&\quad y_l = h(z_l) + v_l \\
&\quad z_l^{LB} \leq z_l \leq z_l^{UB},
\end{aligned} \tag{4.13}$$

where the arrival cost term $\Phi(z_{k-N})$ has been purposely left as a general expression. However, this term represents $-\log\left[p\left(z_{k-N}\middle|Y_0^{k-N-1}\right)\right]$, and it is also possible to rewrite the above derivation so that the arrival cost term is associated with the posterior distribution $p\left(z_{k-N}\middle|Y_0^{k-N}\right)$ [81] or the smoothed PDF $p\left(z_{k-N}\middle|Y_0^{k-1}\right)$. Nevertheless, it is still

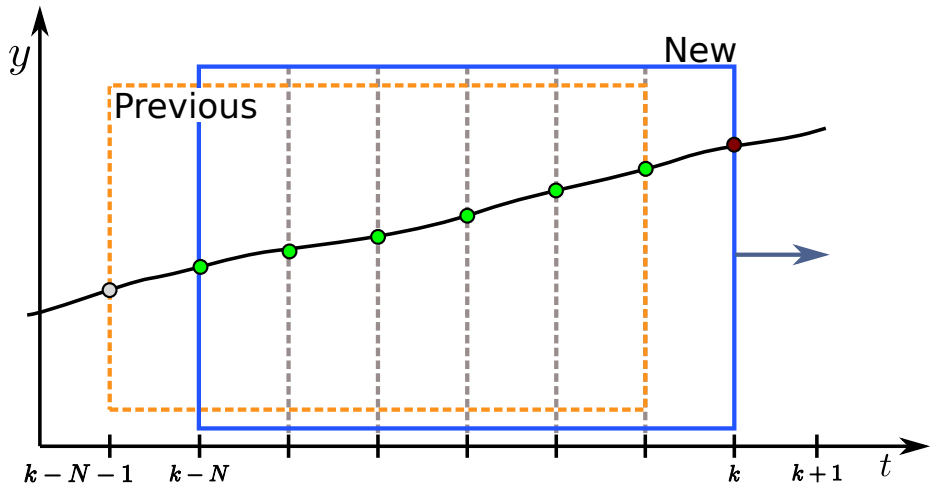



Figure 4.2: Illustration of the moving horizon in MHE.

necessary to find an approximation of the probability density function associated with the arrival cost, regardless of the derivation used. Finally, note that there are two main types of arrival cost formulations: filtered (prior or posterior) or smoothed. Thus in the following subsections we will describe briefly the two main types, and in the following chapters we will describe techniques used to get efficient methods for approximating these probability density functions.

## 4.2 Filtered Update of the Arrival Cost

In the previous section we derived the most common type of Moving Horizon Estimator. In that case the arrival cost term is associated with the prior PDF of the initial state given all the past measurements not included in the horizon window. This formulation is given by the following NLP

$$\min_{z_{k-N},\ldots,z_k} \Phi(z_{k-N}) + \frac{1}{2}\sum_{l=k-N}^{l=k} v_l^T R_l^{-1} v_l + \frac{1}{2}\sum_{l=k-N}^{l=k-1} w_l^T Q_l^{-1} w_l \tag{4.14a}$$

$$\text{s.t. } z_{l+1} = f(z_l) + w_l \tag{4.14b}$$

$$y_l = h(z_l) + v_l \tag{4.14c}$$

$$z_l^{LB} \leq z_l \leq z_l^{UB}, \tag{4.14d}$$

where Equation (4.14d) represents bounds on the states. It is common practice to assume that the arrival cost term is Gaussian, and therefore, it is possible to make the following substitution $\Phi(z_{k-N}) = \frac{1}{2}\left\|z_{k-N} - \hat{z}_{k-N|k-N-1}\right\|^2_{\Pi^{-1}_{k-N|k-N-1}}$, where $\hat{z}_{k-N|k-N-1}$ is a model prediction of the initial condition that is computed using the state estimate $z_{k-N-1|k-N-1}$ and the model of the system. The relationship between the predicted initial condition and the MHE that provides it is given in Figure 4.3. The covariance associated with this prediction ($\Pi_{k-N|k-N-1}$) is generated using the covariance propagation equations from EKF (i.e., Eq. (3.8)) and the linearized system. However, from Chapter 3 we know that this assumption is not consistent when the system is nonlinear or when there are bounds on the states. In reality, the arrival cost term will not remain Gaussian as time moves forward. Moreover, the loss of information due to the linearized model used to propagate the covariance will also introduce errors into the arrival cost term. These errors will propagate through the horizon resulting in errors in the state estimates. The only ways to reduce the effects of these errors are to do a better approximation of the arrival cost or to increase the horizon length. The latter works since the effects of the measurements and model out-weigh the effects of the arrival cost as $N$ increases.

## 4.3 Smoothed Update of the Arrival Cost

A second approach for arrival cost estimation is the so-called smoothed approximation of the arrival cost. For this we re-derive the estimator so that the arrival cost term becomes

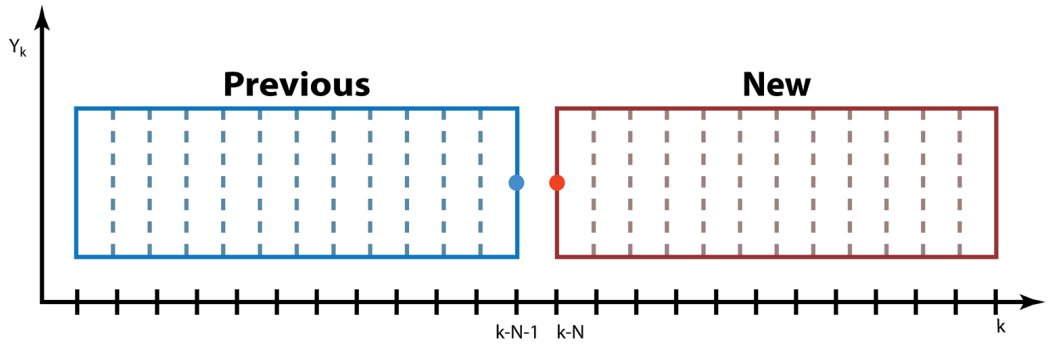



Figure 4.3: Illustration the filtered update of the arrival cost.

$p\left(z_{k-N}\middle|Y_0^{k-1}\right)$. This derivation is done in detail in Chapter 6, and therefore, here we will just show the PDF of the states that gives rise to the MHE with smoothed arrival cost. This PDF is

$$p\left(Z_{k-N}^{k}\middle|Y_0^{k}\right)=\mathcal{K}\,\frac{p\left(z_{k-N}|\,Y_0^{k-1}\right)}{p\left(Y_{k-N}^{k-1}\middle|\,z_{k-N}\right)}\left[\prod_{l=k-N}^{k}p\left(y_l|\,z_l\right)\right]\left[\prod_{l=k-N}^{k-1}p\left(z_{l+1}|\,z_l\right)\right], \tag{4.15}$$

where $\mathcal{K}$ represents a scaling constant and the fraction on the right hand side represents the arrival cost. Here the numerator is the smoothed PDF of the initial condition, while the denominator is a function that allows us to not overweight the measurements that are shared between horizon window and the smoothed initial condition. The relationship between the measurements that are shared between the predicted initial condition ($z_{k-N|k-1}$) and the current horizon window is illustrated in Figure 4.4.

In a similar fashion as in the filtered update MHE we can maximize the PDF in (4.15). This results in a similar NLP as in (4.14). However in this case the arrival cost term becomes $\Phi(z_{k-N}) = -\log\left[p\left(z_{k-N}|\,Y_0^{k-1}\right)/p\left(Y_{k-N}^{k-1}\middle|\,z_{k-N}\right)\right]$. Furthermore, in the original derivation of this method the PDFs in the arrival cost are assumed Gaussian [82]. Thus, after taking this into consideration, the MHE with smoothed arrival cost is given by

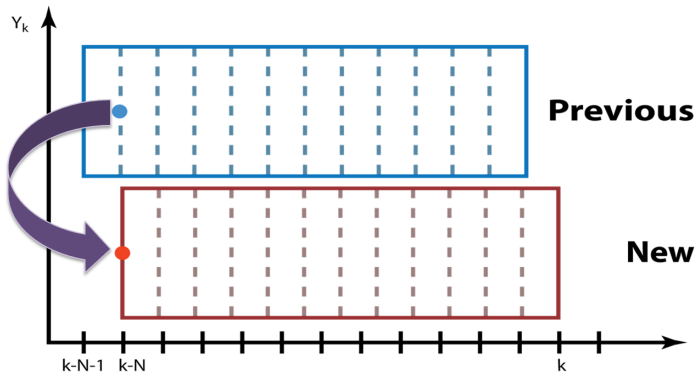



Figure 4.4: Illustration the filtered update of the arrival cost.

$$\min_{z_{k-N}} \quad \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k} v_l^T R_l^{-1} v_l + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l \tag{4.16a}$$

$$\text{s.t.} \quad z_{l+1} - f\left(z_l\right) + w_l = 0 \tag{4.16b}$$

$$y_l - h\left(z_l\right) + v_l = 0 \tag{4.16c}$$

$$z^{LB} \leq z_l \leq z^{UB}, \tag{4.16d}$$

where the terms $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$ are all functions of the shared measurements $Y_{k-N}^{k-1}$. Expressions for these terms are derived in Appendix D. In addition, the smoothed state estimate $\hat{z}_{k-N|k-1}$ is used as the predicted initial condition, and $\Pi_{k-N|k-1}$ is the covariance associated with the prediction. The latter can be approximated using the EKF and Extended Kalman Smoothing (EKS) equations, and the linearized model. This requires the forward propagation of a prior covariance $M_{k-N|k-N-1}$ up to $\Pi_{k-1|k-1}$, and then the backward smoothing of $\Pi_{k-1|k-1}$ to generate $\Pi_{k-N|k-1}$. This can be very computationally expensive since the covariance matrices formed can be very large depending on the number of states, and the EKF and EKS equations require matrix inversions that can be very costly.

## 4.4 Conclusions

In this chapter we derived a general MHE formulation by applying standard dynamic programming techniques to the full information problem. From doing this a term that represents the PDF of the initial condition of the plant at the beginning of the horizon, arises. This so-called arrival cost compresses the information of the measurements that have not been considered in the horizon. Furthermore, since the initial condition moves forward with the horizon, then it is necessary to update the PDF associated with the arrival cost.

In this chapter we also presented two possible ways to represent the arrival cost term. It can be described with a prior distribution (or even a posterior PDF as shown in the next chapter), or a smoothed distribution. However, since the system is nonlinear and there are bounds on the state estimates, the arrival cost PDF is not Gaussian. Thus it important to generate approximations of this term that are consistent with the enforced bounds and constraints, and with the non-Gaussian nature of the states. This will be addressed in Chapter 5, where we will show that it is possible to reduce the size of the horizon window by using the constrained sample based filters described in Chapter 3, to update the parameters of the arrival cost term. Moreover, in Chapter 6 we will show a very efficient strategy to extract arrival cost information directly from the KKT conditions of the NLP. This will be used with the smoothed approach to develop efficient NLP sensitivity based strategies for constrained state estimation.

# Chapter 5

# Filtering Approach for Arrival Cost

In previous chapters we derived the Moving Horizon Estimator, and discussed the origin of the arrival cost term through the Markov property and dynamic programming approach to approximate the full information problem. The arrival cost term compresses the information of the past measurements not included in the horizon, and represents the information given by the probability density function $p\left(z_{k-N}|\, Y_0^{k-N-1}\right)$. However, in this chapter we will provide a more general form of the arrival cost probability density function so that it may be represented by a prior or a posterior PDF. This allows us to provide different ways to approximate it through the use of constrained and unconstrained sample based filters (see Chapter 3). Moreover, we will show that the required horizon length to reduce the error in the estimates is directly related to how well the arrival cost information is approximated.

The rest of this chapter is organized as follows. In Section 5.1 we derive the filtering approach for arrival cost where the associated distribution is the posterior PDF $p\left(z_{k-N}|\, Y_0^{k-N}\right)$, this is useful when using different types of sample based filters to approximate the parameters of this term, in order to make better use of the information provided by them. In Section 5.2 we discuss the use of sample based filters to approximate the first two moments in the arrival cost when using the quadratic form of this term that, because of its simplicity, is the most commonly used approach. In Section 5.3 we use ideas taken from Particle Filters to directly approximate the conditional density of the arrival cost term, and we illustrate the benefits of these methods through two examples in Section 5.4. Finally, Section 5.5 concludes this chapter.

## 5.1 Posterior PDF for Arrival Cost

In Chapter 4 we derived MHE using the filtering approach for arrival cost as a result of applying dynamic programming techniques to the full information problem. In that case, after having split the objective function term for the two time sets, the arrival cost term involves a prior conditional density function of the initial condition. In other words, the term $\Phi\left(z_{k-N}\right) = -\log\left[p\left(z_{k-N}|\, Y_0^{k-N-1}\right)\right]$ represents the penalization of the prediction of the initial condition of the dynamic system in the horizon. However, it is possible to also represent the arrival cost term with a posterior PDF. This can be achieved by splitting the objective function of Problem (4.9) in the following way

$$\begin{aligned}\min\ & \left[\left(z_0-\hat{z}_0\right)^T\Pi_0^{-1}\left(z_0-\hat{z}_0\right)+\sum_{l=0}^{l=k-N}v_l^TR_l^{-1}v_l+\sum_{l=0}^{l=k-N-1}w_l^TQ_l^{-1}w_l\right]+\\ & \left[\sum_{l=k-N+1}^{l=k}v_l^TR_l^{-1}v_l+\sum_{l=k-N}^{l=k-1}w_l^TQ_l^{-1}w_l\right]\\ \text{s.t. } & \text{model constraints and bounds,}\end{aligned} \tag{5.1}$$

where the first term in brackets has one more measurement (i.e., $y_{k-N}$) than Equation (4.10). Also note that the objective function has been scaled to remove the $\frac{1}{2}$, and simplify notation. However, this will not affect the values of the states at solution of the optimization problem, only the value of the objective function which has been doubled. Moreover, to simplify notation this has been done throughout the rest of the dissertation. Furthermore, after applying the forward dynamic properties used previously, we obtain the arrival cost term $\Phi\left(z_{k-N}\right) = -\log\left[p\left(z_{k-N}|\, Y_0^{k-N}\right)\right]$, and the MHE problem with posterior distribution becomes

$$\begin{aligned}\min\quad & \Phi\left(z_{k-N}\right)+\sum_{l=k-N+1}^{l=k}\left\|y_l-h\left(z_l\right)\right\|^2_{R_l^{-1}}+\sum_{l=k-N}^{l=k}\left\|z_{l+1}-f\left(z_l,u_l\right)\right\|^2_{Q_l^{-1}}\\ \text{s.t.}\quad & z_l^{LB}\le z_l\le z_l^{UB}.\end{aligned} \tag{5.2}$$

The arrival cost terms in Problems (4.10) and (5.2) require probability density functions

that are generated in the filtering problem (see Chapter 3), which is why we call these the filtering approach for arrival cost approximation. Unfortunately, as described in Chapter 3 for general nonlinear systems the structure and type of PDFs that describe the states are very hard, if not impossible, to determine. Because of this, in the first descriptions of MHE [34, 83, 84, 85], these terms were assumed Gaussian. This leads to further simplifications, and most importantly, a closed form quadratic expression for the arrival cost. Thus, this well known MHE formulation is given by

$$\begin{aligned} \min \quad & \left\| z_{k-N} - \hat{z}_{k-N|k-N} \right\|^2_{\Pi^{-1}_{k-N|k-N}} + \sum_{l=k-N+1}^{l=k} \left\| y_l - h\left(z_l\right) \right\|^2_{R_l^{-1}} + \\ & \sum_{l=k-N}^{l=k} \left\| z_{l+1} - f\left(z_l, u_l\right) \right\|^2_{Q_l^{-1}} \\ \text{s.t.} \quad & z_l^{LB} \leq z_l \leq z_l^{UB}, \end{aligned} \tag{5.3}$$

where the parameters of the arrival cost, i.e., mean and covariance, are generated from the solution of an MHE problem $k - N$ sample times in the past and the EKF equations, respectively.

Unfortunately, approximating the mean and covariance in this fashion causes the state estimates to oscillate as shown in [35, 82]. This is due to the fact that the predicted initial condition and the current state estimate basically share no measurement information, as in the formulation in Eq. (5.3), or only one measurement as in (4.14) with a Gaussian arrival cost term. A second possibility for generating the initial condition prediction is by generating it with a second filtering algorithm such as EKF or UKF, and the covariance information can be approximated as before. For the rest of this chapter we will focus on using the posterior covariance with the filtering approach.

## 5.2 Approximation of Arrival Cost Parameters

Incorrect approximations of the arrival cost (as with the Gaussian assumption) introduce errors that propagate through the horizon. It is therefore necessary to add more measurement information in order to reduce the weight of the initial condition with respect to the model information and measurement error terms. Furthermore, when bounds are imposed on the states the Gaussian assumption no longer holds. This is sometimes dealt with through the use of truncated distributions, which translates to adjusting the covariance matrix, $P_{k-N|k-N}$ to account for the bounds. The main difficulty with this approach is that multivariate Gaussian or truncated Gaussian distributions may not prove to be a good approximation of the conditional densities for a highly nonlinear system. In fact, in some special cases the conditional densities have been shown to be multi-modal. Nevertheless, in this section we discuss how to approximate the parameters of the arrival cost when using its quadratic form through the use of constrained and unconstrained sample based filters such as UKF and CEnKF (see Sections 3.3 and 3.5).. In the following section we deal with the problem of approximating the conditional densities directly using ideas taken from particle filters (see Section 3.4).

The 2-norm arrival cost is often used because it is simpler to implement, and to generate the information required for it. Even if more measurements are required in the horizon to compensate for the errors in arrival cost estimation. Thus, if for simplicity we choose to use the quadratic form, then we need good approximations of the mean and covariance. For consistency, these should satisfy the bounds and constraints imposed on the states. Since bounds are present the distribution associated with the initial condition $p\left(z_{k-N}\left|Y_0^{k-N}\right.\right)$ will no longer be Gaussian. Thus we use constrained filters to generate parameters that are consistent with these bounds. Moreover, we also use sample based filters that allow us to approximate the moments of the non-Gaussian distribution. These filters are described in Chapter 3 and are also summarized in Table 5.1 below. Here we propose to approximate both mean and covariance of the arrival cost term from the values generated using these

filters, because of this we need to run the filter in parallel to MHE. However, since the initial condition lies in the past, all the measurement information is known. Thus the computational cost of having this second filter is shifted to the background. In fact, it is possible to generate the arrival cost information at every point in the horizon, and store it so it can be used as it is required.

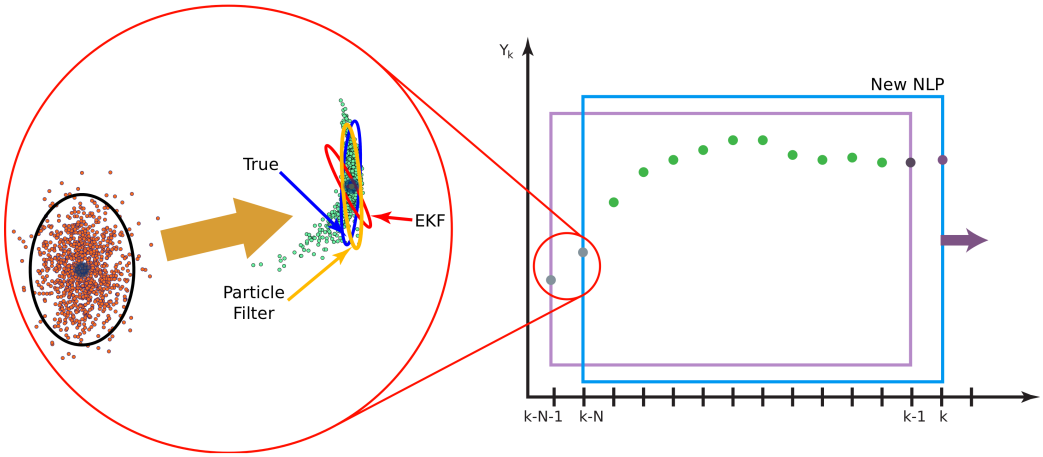



Figure 5.1: Illustration of sample based filter approximation of arrival cost parameters.

Figure 5.1 illustrates how samples are used to approximate the moments of the arrival cost term. Note that the idea is simple, the secondary filter generates approximations of the mean and the covariance, these are then used as $\hat{z}_{k-N|k-N}$ and $\Pi_{k-N|k-N}$ respectively. However, since these parameters are now consistent with bounds and constraints (e.g., when using a constrained filter), and the non-Gaussian nature of the arrival cost term is taken into account through the use of samples, then it is possible to reduce the amount of measurements required in the MHE, and have small estimation errors. Finally, these methods will be illustrated in Section 5.4 with two case studies.

## 5.3 Approximation of Arrival Cost Conditional Density

In the Sections 3.5 and 3.4 different sample based filters (e.g., CEnKF and PF) were described, and they were used in the previous section to approximate the first two moments of the conditional density of the arrival cost term. For those cases, the arrival cost used is the well known 2-norm quadratic expression. However, in this section we describe an alternate approach where we find an approximation of $\Phi_{k-N} = -\log\left[p(z_{k-N}|Y_0^{k-N})\right]$ using constrained PFs. Some PF formulations construct an approximation of the conditional density using random samples $\{z_{k-N}^{(i)} : i = 1, 2, \ldots, N_p\}$ together with the set of weights $\{\omega_{k-N}^{(i)} : i = 1, 2, \ldots, N_p\}$, as follows:

$$p\left(z_{k-N}|Y_0^{k-N}\right) \approx \sum_{i=1}^{N_p} \omega_{k-N}^{(i)} \delta\left(z_{k-N} - z_{k-N}^{(i)}\right), \tag{5.4}$$

where $\delta(\cdot)$ denotes the Dirac delta function. The weights can be viewed as approximations to the relative posterior probabilities of the particles. Since the finite sample approximation of the conditional density involves Dirac delta functions, it is difficult to compute $\Phi_{k-N} = -\log\left[p(z_{k-N}|Y_0^{k-N})\right]$ directly using this density approximation. However, it is possible to develop an estimate of the quantity $\Phi_{k-N}$ through an approximation analogous to (5.4). Here we approximate $-\log\left[p\left(z_{k-N}|Y_0^{k-N}\right)\right]$ from particle samples, with Dirac delta terms associated with the *distribution* of each particle. This leads to:

$$-\log\left[p\left(z_{k-N}|Y_0^{k-N}\right)\right] \approx \sum_{i=1}^{N_p} \omega_{k-N}^{(i)} \left\{-\log\left[p_i\left(z_{k-N}^{(i)}|Y_0^{k-N}\right)\right]\right\}. \tag{5.5}$$

If we approximate $p_i\left(z_{k-N}^{(i)}|Y_0^{k-N}\right)$ for each particle to be Gaussian, then

$$p_i\left(z_{k-N}^{(i)}|Y_0^{k-N}\right) \approx \mathcal{N}\left(\hat{z}_{k-N|k-N}^{(i)}, \hat{P}_{k-N|k-N}^{(i)}\right),$$

where $\hat{z}_{k-N|k-N}^{(i)}$ and $\hat{P}_{k-N|k-N}^{(i)}$ are the mean vector and the covariance matrices associated with *i*-th particle. Since it is proposed to propagate the particles using CEnKF, it is possi-

ble to estimate these quantities from the samples associated with each particle. With this assumption, it follows that

$$-\log\left[p_i\left(z_{k-N}^{(i)}|Y_0^{k-N}\right)\right] \approx \left\|z_{k-N}-\hat{z}_{k-N}^{(i)}\right\|^2_{\left[\hat{P}^{(i)}_{k-N|k-N}\right]^{-1}},$$

and the expression for approximation of $\Phi_{k-N}$ given by (5.5) reduces to

$$\Phi_{k-N} \approx \sum_{i=1}^{N_p} \omega_{k-N}^{(i)} \left\|z_{k-N}-\hat{z}_{k-N}^{(i)}\right\|^2_{\left[\hat{P}^{(i)}_{k-N|k-N}\right]^{-1}}. \tag{5.6}$$

The resulting expression is qualitatively similar to a Gaussian sum approximation, which is often used for approximating non-Gaussian densities. Similar expressions have been derived by Gopaluni [86] in a completely different context (i.e., identification of a nonlinear black box model using the expectation maximization algorithm that employs a particle filter and a particle smoother). When bounds are imposed on the estimated states and $\hat{P}^{(i)}_{k-N|k-N}$ is estimated using some constrained filter (such as CEnKF or some constrained EKF), we can continue to use the proposed approximation. The estimates of $\hat{P}^{(i)}_{k-N|k-N}$ generated using the constrained filter are consistent with the bounds.

## 5.4 Examples and Discussion

In this section two examples are shown where the arrival cost has been estimated according to the descriptions given above using unconstrained and constrained sample based filters. The list of methods used is given in Table 5.1. Also, this table shows how samples are generated and constraints are handled for each of the filters. Some of these filters have not been described in Chapter 3, however they have been discussed extensively in the literature, and information on them can be found in the following references: UKF [10, 11, 12], Unscented Recursive Nonlinear Dynamic Data Reconciliation (URNDDR) [9, 17], Ensemble Kalman Filter (EnKF) [13, 14], Constrained Ensemble Kalman Filter (CEnKF) [18], and PFs [15, 16]. In addition, some of these methods are not constrained, however they are

included here to highlight the effects of not taking bounds into account when updating the arrival cost. Note that in the following figures the labels of each curve are taken from column 1 of Table 5.1. Also, MHE$^{\text{N}}$ represents the case in which only the covariance is approximated using EKF, while the mean is obtained by propagating forward one sampling time the filtered estimate of an MHE solution $k - N - 1$ sampling times in the past as was originally described by Rao *et al.* [33].

| **Filter** | **Sampling** | **Sample bounds** | **Update type** | **Update bounds** |
|---|---|---|---|---|
| EKF | none | none | linear | unconstrained |
| UKF | sigma points | none | linear | unconstrained |
| EnKF | random | none | linear | unconstrained |
| EnKFPF* | random/random | none | linear | unconstrained |
| UKFPF* | sigma points/random | none | linear | unconstrained |
| $\omega$EnKFPF* | random/random | none | linear | unconstrained |
| URNDDR | sigma points | clipping | nonlinear | QP |
| URNDDRPF* | sigma points/random | clipping | weights | QP/through density |
| CEnKF | random | clipping | weights | QP |
| CEnKFPF* | random/random | clipping | weights | QP/through density |
| $\omega$CEnKFPF* | sigma points/random | clipping | weights | QP/through density |

Table 5.1: Sampling types and bounds for the different filters used to update the arrival cost. *PF with importance distribution. In these cases sampling and update bounds are enforced through *importance density/particle filter*.

Since it is assumed that the state and measurement noises are random variables with known statistics, then it is possible to compare each of the methods using the normalized Sum of Squared Errors (SSE). For the *i*-th method and *n*-th horizon length we evaluate the SSE with

$$SSE_n^{(i)} = \sum_{l \in T} \sum_{j \in S} \left( \hat{z}_{j,l|l} - z_{j,l} \right)^2 , \tag{5.7}$$

where $\hat{z}_{j,l|l}$ is the *j*-th element of the estimated state vector at time $l$, $z_{j,l}$ is the *j*-th element of the true state vector at time $l$, $T$ is the set of sampling times and $S$ is the set of state variables. Note that for this the state vectors were also normalized to values between 0 and 1 using the upper and lower bounds, and thus they are dimensionless. Since, for each pair $(i, n)$, 10 repetitions were done, we take the mean ($\overline{SSE}_n^{(i)}$). Finally, this is normalized; thus for each method and horizon length we do the following.

$$\widehat{SSE}_n^{(i)} = \frac{\overline{SSE}_n^{(i)}}{SSE_{max}} \tag{5.8}$$

where $SSE_{max}$ is the highest average error of all methods and all cases (repetitions) used. In other words, we set the highest average error to one with the normalizing factor obtained from

$$SSE_{max} = \max \left\{ \overline{SSE}_n^{(i)} \right\} \tag{5.9}$$

### 5.4.1 Example 1: CSTR

As a first example we use the non-isothermal Continuously Stirred Tank Reactor (CSTR) described in Appendix B.2, where the exothermic reaction between thiosulfate and hydrogen peroxide takes place. In this case the states are the concentration of thiosulfate ($C_A$), the reactor temperature ($T_R$), and the cooling water temperature ($T_{cw}$). For the CEnKF and EnKF based MHE 20 particles were used, and for the particle filters 20 samples were used. In addition, 15 particles were used for the ensemble filters used for the importance distribution, giving a total of 300 particles for ($\omega$)EnKFPF and ($\omega$)CEnKFPF. Finally, the statistical parameters and initial conditions were set to

$$\begin{aligned}
&w_k \sim \mathcal{N}(0, Q_k), \quad Q_k = diag(10^{-8}, 0.25, 0.25) \\
&v_k \sim \mathcal{N}(0, R_k), \quad R_k = 0.25 \\
&z_{0|0} \sim \mathcal{N}(0, P_0), \quad P_0 = diag(10^{-7}, 2.5, 2.5), \quad z_{0|0} = [0.0192, 384.0072, 371.2735]^T.
\end{aligned}$$

In Figure 5.3, $\widehat{SSE}_n^{(i)}$ as a function of horizon length ($N$) is shown for both unconstrained (5.3(a)) and constrained (5.3(b)) filters. Here it is also shown that using EKF to approximate the arrival cost, using MHE$^{\text{N}}$ or EKF, has the worst performance. As expected, on the other hand, using any of the other filters proposed here works better in the sense that the prediction errors are lower. In this example no bounds become active during the time period chosen for the simulation. Thus the difference between using an unconstrained or a constrained filter is not as clear. Moreover, for most of the methods shown after a horizon length of 5, little is gained by extending the number of measurements used. For this case, this suggests that it would be possible to use MHE with a horizon length smaller than that when EKF is used, and still get much better performance. In other words, the computational expense of solving a larger Nonlinear Program (NLP) on-line is traded for simulations required by the sample based filter, which can be carried out using the past measurements in the time interval between two samples.

Moreover, since this process is described by a system of nonlinear Differential Algebraic Equations (DAE), it needs to be transformed into a discrete time model. For this we used the simultaneous approach for dynamic optimization, where we used orthogonal collocation on finite elements to transform the DAE model into a set of algebraic constraints [24]. Because of this the number of variables in the NLP increases with the size of the horizon as $n_v = 2Nn_z(n_{cp} + 1)$ (where $n_{cp}$ is the number of collocation points used, here we used $n_{cp} = 3$). Therefore, a considerable reduction in problem size is achieved when using a better approximation of the arrival cost. For example, in this case if $N = 2$, the number of variables is 19, while if $N = 10$ we get $n_v = 83$. Clearly a reduction in problem size will result in a reduction in computational time. For example, using an Intel Quad-core i7 computer running Linux at 2.8Gz with 9Gb RAM, for a horizon length of 6, the average time to solve the problem was $8.41 \times 10^{-3}$ CPU s, while for $N = 7$ and $N = 8$ it took an average of $9.120 \times 10^{-3}$ CPU s and $9.63 \times 10^{-3}$ CPU s respectively. Note that this is the time for solving the NLP only, the time for approximating the arrival cost is not considered

since it is done in the background. Therefore, using a better approximation of the arrival cost will allow us to use a smaller horizon window, and thus there will be a reduction in the on-line computational effort.

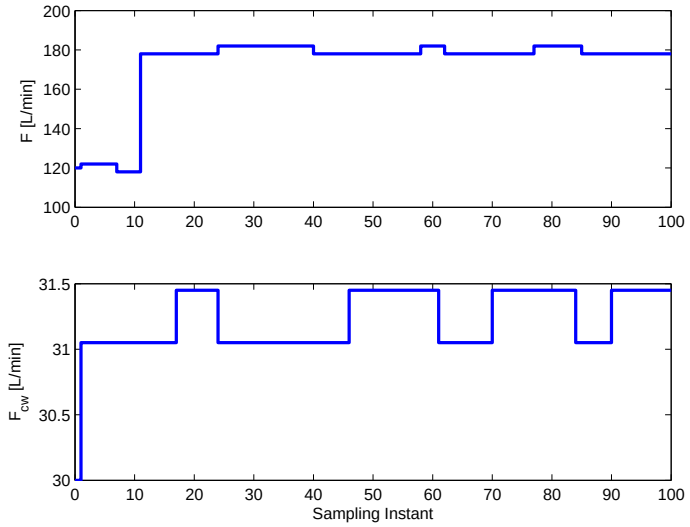
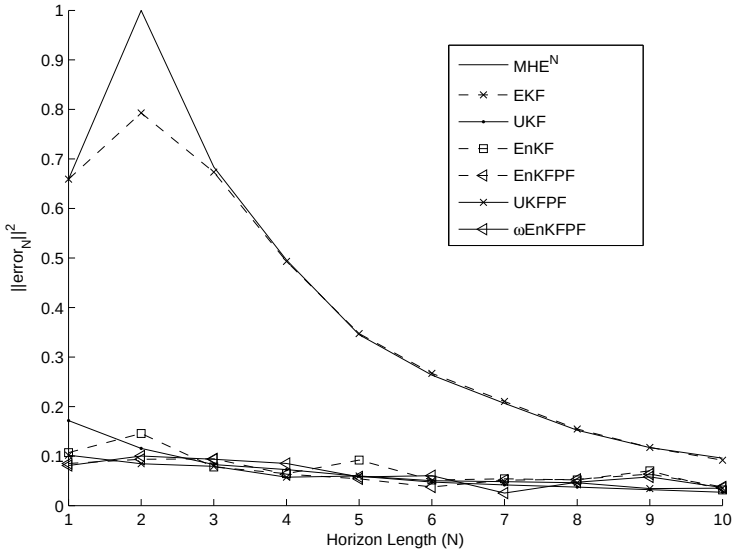


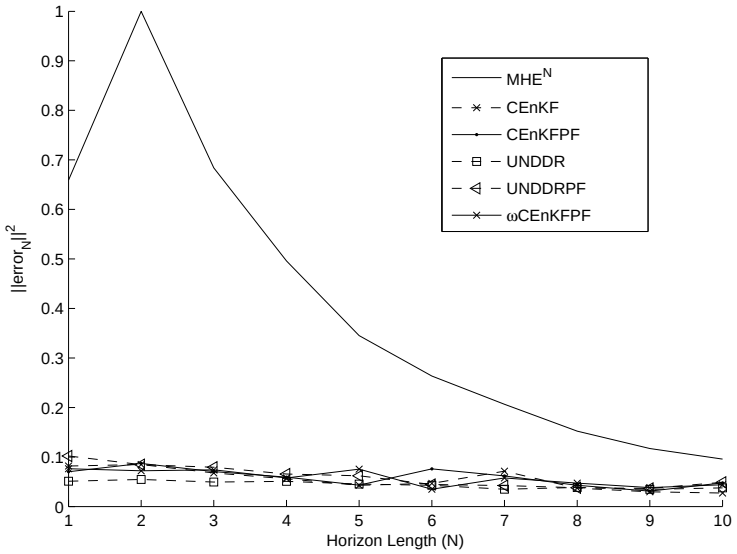

Figure 5.2: Manipulated variable changes for the CSTR example.

### 5.4.2 Example 2: Constrained Batch Reactor

In this example the bounds are active, and they play an important part in the performance of the estimator. Thus in this case the benefit of using constrained filters to approximate the arrival cost will be obvious. This example was chosen because it is known that EKF fails for this system [6]. It is a simple isothermal batch reactor in which an irreversible gas phase reaction takes place at constant volume. The discrete time model of this system is described in Appendix B.1. However, note that $C_{A,k}$ and $C_{B,k}$ are the partial pressures of components A and B, $\Delta t = t_k - t_{k-1} = 0.1$, $\bar{k} = 0.16$ is the kinetic rate constant, and the measurement is given by the total pressure $C_{T,k}$ (Eq. (5.10)).




(a)




(b)

Figure 5.3: MSE as a function of horizon length when using (a) unconstrained filters and (b) constrained filters for the arrival cost approximation for the CSTR example.

$$C_{T,k} = C_{A,k} + C_{B,k} + v_k \tag{5.10}$$

The parameters of the noise variables statistics are $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$, where $Q_k = diag(10^{-6}, 10^{-6})$ and $R_k = 10^{-2}$. Finally, the initial state estimate and the covariance associated with it are $\hat{z}_0 = [0.1,\ 4.5]^T$ and $P_0 = diag(36, 36)$. Note that the state estimate is far from the true initial condition given by $z_0 = [1,\ 3]^T$. Here, 20 particles were used for the CEnKF and EnKF, and for the ($\omega$)EnKFPF and ($\omega$)CEnKFPF 15 particles were used plus 10 particles for the importance distribution (i.e., a total of 150 particles).

Figure 5.4 shows the effect of increasing the horizon length when using an inconsistent approximation of the arrival cost. In this case EKF, which is known to fail for the chosen simulation conditions, was selected for approximating the arrival cost. This figure shows that as the horizon length increases, the effects of the arrival cost on the estimator performance decreases, and eventually correct state estimates are obtained. Figure 5.5 shows the simulation results of applying all of the methods for arrival cost approximation used in the previous example with a horizon length of 5. Here the unconstrained filter based methods have the worst performance. In particular, when using UKF, the concentration oscillates between the upper and lower bound. This happens because the MHE tries to find initial conditions that are close to the predicted ones in the arrival cost, and that also yield trajectories close to the measurements in the horizon. However, in some sampling times it is not possible to go beyond the upper bound or below the lower bound in order to satisfy the measurement model. On the other hand, Figure 5.5 also shows that using an approximation of the arrival cost, that is consistent with bounds, improves the quality of the state estimate. In fact, using any of the constrained methods described here provides much better state estimates and results in better performance of the estimator.

Figure 5.6 shows $\widehat{SSE}_n^{(i)}$ as a function of horizon length when using unconstrained (5.6(a)) and constrained filters (5.6(b)). In this case it is clear that using a constrained filter greatly improves the performance of MHE. Note, that using MHE$^N$ also results in better

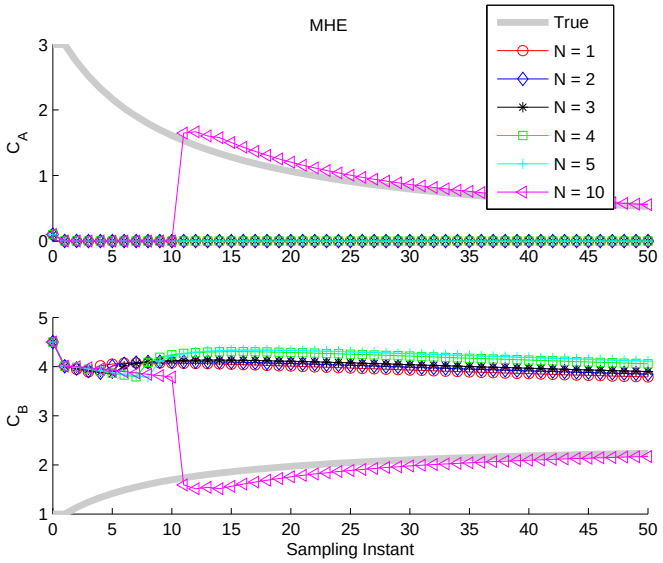
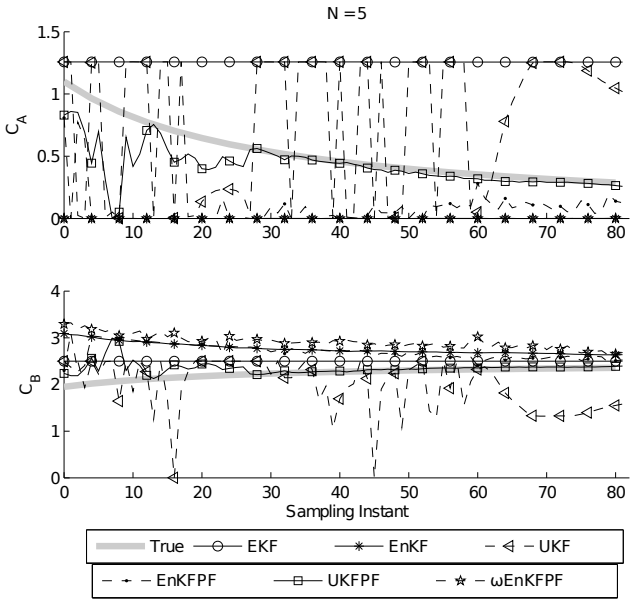


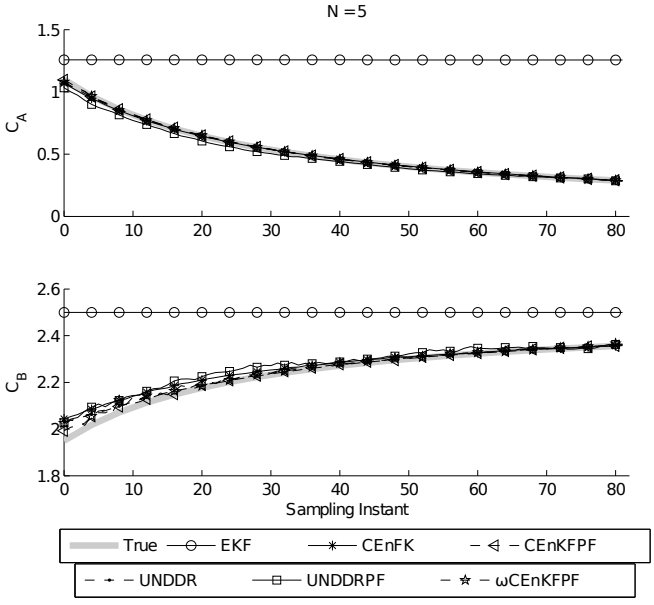

Figure 5.4: EKF with different horizon lengths for the isothermal batch reactor.

performance than taking both mean and covariance from EKF as the parameters of the arrival cost. This is because for MHE$^N$ the predicted initial condition of the arrival cost is taken from a previous MHE (i.e., a constrained prediction), whereas for EKF the initial condition comes from EKF which has been shown to fail for this system. Moreover, since the process is described by a discrete time model, the number of variables in the MHE problem increases as $n_v = 2N(n_z + 1)$. Therefore, when $N = 2$ the problem will have 12 variables, while if $N = 10$ it will have 60 variables. Again, this means that when using constrained sample based filters there will be a reduction in the on-line computational effort, while obtaining better estimates of the states.

It is important to have a smaller horizon length so that the MHE problem can be solved faster on-line, as soon as the new measurement becomes available. On the other hand, the approximation of the arrival cost can be done in the background (i.e., between sampling times) because this only deals with information in the past which we already have. The size of the NLP required for MHE when using the full discretization scheme (used here)

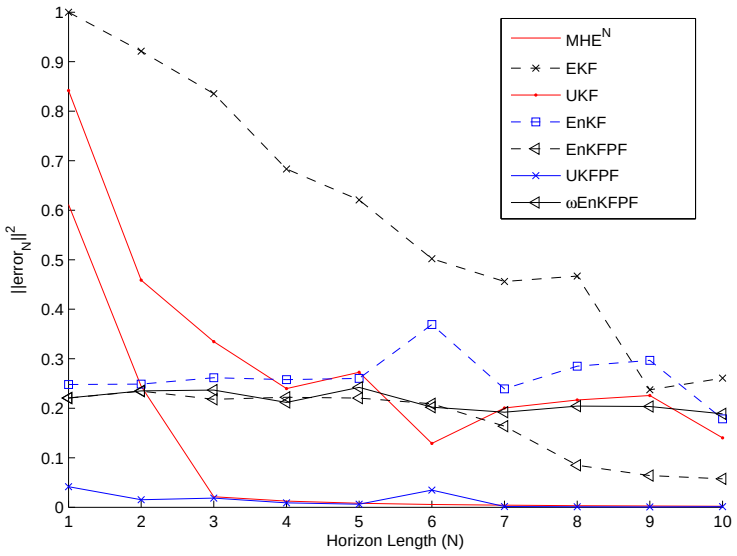



(a)

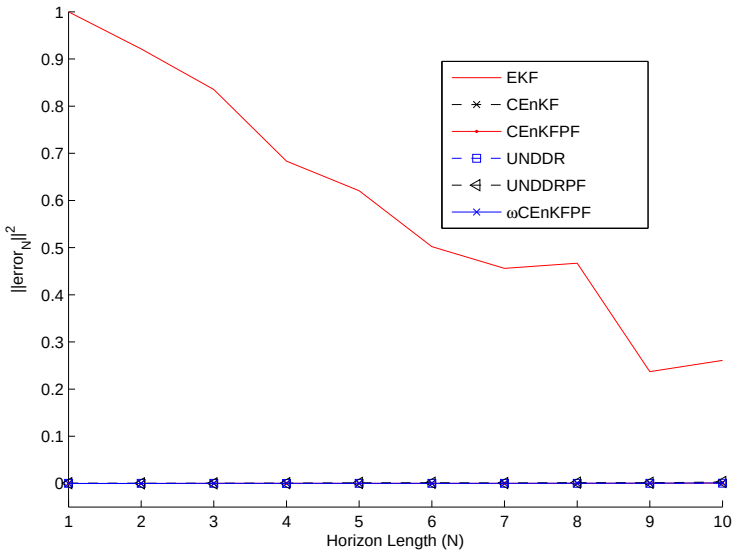



(b)

Figure 5.5: Comparison of MHE with horizon length 5 with all methods using unconstrained filters (a) and constrained filters for arrival cost approximation (b).




(a)




(b)

Figure 5.6: SSE as a function of horizon length for the isothermal batch reactor with all methods using unconstrained filters (a) and constrained filters for arrival cost approximation (b).

grows linearly with the number of states. Moreover, the computational complexity of these schemes scales as $O((n_z + n_w)^\beta)$ (with $\beta = 1$ or at most 2) when using a collocation based approach to discretize the model, and a solver that takes advantage of sparsity and problem structure such as IPOPT [46, 87]. On the other hand, since the sigma points in UKF are deterministically chosen, the problem size grows linearly with the number of states as $O(n_z+n_y+n_w)$. Finally, there is still no property that bounds the computational complexity of PF [88]. The particle filters shown here may suffer from the "curse of dimensionality." In other words, the number of particles required to approximate the distributions can grow exponentially with the number of states [39]. However, this may be avoided by a proper selection of the importance distribution or if the system is "vaguely" Gaussian. For an insightful discussion on this the reader is referred to Daum and Huang [88]. In the present case the importance distribution was chosen because it satisfies constraints (CEnKF) and uses samples to approximate the distributions. Thus it should give close approximations of the real distributions, and therefore aid in avoiding the dimensionality problem associated with PF. Moreover, because the sample based filters are only applied in the background, these types of arrival cost approximations should still be viable for problems of relatively low dimensionality.

## 5.5 Conclusions

Extended Kalman Filter (EKF) approximations of the arrival cost parameters in moving horizon estimation (MHE) introduce unwanted errors that propagate through the horizon window. These errors require the choice of longer horizon lengths that also require a larger optimization problem to be solved on-line. The errors are due to the assumption in EKF that the probability density function of the states, given the previous measurements at the beginning of the horizon, $p\left(z_{k-N}|Y_0^{k-N}\right)$, is Gaussian. In addition, since the states often have bounds, this implies that the EKF propagation of covariance matrices in the arrival cost is inconsistent.

On the other hand, particle-based filters can approximate arrival cost distributions using samples, and thus require few assumptions on the type of distribution. In addition, these methods use the nonlinear model directly. Thus, we also avoid forming the Jacobian matrices needed for EKF. Moreover, CEnKF and constrained PFs handle bounds on the states, and thus provide a more consistent approximation of the arrival cost. Here, we use these features to develop constrained methods for approximating the arrival cost in MHE. Specifically, in the traditional 2-norm approximation, constrained particle based filters are used to generate the first two moments of the arrival cost distribution. In addition, we also develop a method to approximate the arrival cost distribution from the explicit log likelihood approximation of the conditional PDF, using concepts from particle filtering.

The resulting improvements in the arrival cost approximation allow us to use a smaller horizon window for MHE, and a smaller NLP can be solved on-line. Here, different sample-based filters are tested with two benchmark examples taken from the literature, and we show the benefit of using these constrained particle-based arrival cost approximations. As measured by the sum of squares error (SSE), these filters give a better approximation of the arrival cost and demonstrate that a much smaller horizon achieves good performance of the MHE.

Finally, we observe that even when bounds do not become active in the MHE, the constrained filters do not reduce to unconstrained filters. When constrained filters are employed, the particles used in the estimation of arrival cost remain within the specified bounds, which, in turn, significantly improve the estimates of the arrival cost (see Example 5.4.2). Having bounds on the particles used for estimation of the arrival cost also implies that the non-Gaussian nature of the conditional state density is implicitly considered while constructing the estimate. As a consequence, the mean square errors remain smaller for shorter horizons when constrained filters are used to estimate the arrival cost.

# Chapter 6

# Smoother Approach for Arrival Cost

In this chapter we derive the MHE estimator formulation where the arrival cost term is defined by the smoothed density $p\left(z_{k-N}|\, Y_0^{k-1}\right)$. The motivation behind this approach is that, for a large enough system, using a sample based approach to approximate the filtered arrival cost would require a massive amount of particles. The computational expense for computing these parameters might not be easily justified in such a case. On the other hand, for such a system it might be better to use a larger horizon and approximations of the arrival cost parameters based on EKF. For example, using the prediction of the initial condition using the MHE state estimate obtained $k-N-1$ sample times in the past and the covariance from EKF, as originally proposed by [83]. However, this formulation has been shown to produce oscillating state estimates [35], because the filtered arrival cost and the rest of the state estimates in the horizon window share only one measurement. In fact, in the same manuscript, the authors suggest the use of the smoothing arrival cost as a better alternative. However, they propose the approximation of the arrival cost covariance using both the EKF and Extended Kalman Smoother (EKS) equations.

This chapter is organized as follows. In Section 6.1 we derive MHE with the smoothed arrival cost update, and we discuss how we can generate covariance information for the initial condition with the use of the Riccati equations. Additionally, in this section we introduce a property that relates the covariance of the state estimates with the reduced Hessian which we then use to approximate the smoothed covariance. Then, in Section 6.2 we illustrate the use of these strategies in two simulation examples, and finally we conclude in Section 6.3.

## 6.1 Derivation of MHE with Smoothed Arrival Cost

This derivation follows closely the one used for the full information problem. However, in this case we note that the density function of the states in the horizon window is given by $p\left(Z_{k-N}^{k}\middle|\, Y_0^k\right)$. Therefore we can apply the properties of probability density functions to rewrite this density as

$$p\left(Z_{k-N}^{k}\middle|\, Y_0^k\right) = \frac{p\left(Z_{k-N}^{k}, Y_0^k\right)}{p\left(Y_0^k\right)} = \mathcal{K}_1 p\left(Z_{k-N}^{k}, Y_0^k\right) = \tag{6.1a}$$

$$= \mathcal{K}_1 p\left(z_k, y_k\middle|\, Z_{k-N}^{k-1}, Y_0^{k-1}\right) p\left(Z_{k-N}^{k-1}, Y_0^{k-1}\right) = \tag{6.1b}$$

$$= \mathcal{K}_1 p\left(z_k, y_k\middle|\, Z_{k-N}^{k-1}, Y_0^{k-1}\right) p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right) p\left(Y_0^{k-1}\right) = \tag{6.1c}$$

$$= \mathcal{K}_2 p\left(z_k, y_k\middle|\, Z_{k-N}^{k-1}, Y_0^{k-1}\right) p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right), \tag{6.1d}$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are scaling constants that do not depend on the states. We can further simplify this using the Assumptions 3.1. Therefore, we can apply the Markov property to Eq. 6.1d, and since measurements are independent of each other we have

$$p\left(Z_{k-N}^{k}\middle|\, Y_0^k\right) = \mathcal{K}_2 p\left(y_k\middle|\, z_k\right) p\left(z_k\middle|\, z_{k-1}\right) p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right). \tag{6.2}$$

In order to obtain the smoothed covariance we now apply the product rule to $p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right)$, which leads to

$$p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right) = p\left(Z_{k-N+1}^{k-1}\middle|\, z_{k-N}, Y_0^{k-1}\right) p\left(z_{k-N}\middle|\, Y_0^{k-1}\right) = \tag{6.3a}$$

$$= \frac{p\left(Z_{k-N+1}^{k-1}, Y_0^{k-1}\middle|\, z_{k-N}\right)}{p\left(Y_0^{k-1}\middle|\, z_{k-N}\right)} p\left(z_{k-N}\middle|\, Y_0^{k-1}\right), \tag{6.3b}$$

where Eq. (6.3b) comes from applying Bayes' rule. Note that this final expression has the smoothed probability function of the initial condition. We now derive the MHE with smoothed arrival cost by further manipulating the distribution functions. Thus, we apply the product rule to the density function on the numerator of (6.3b) to yield

$$p\left(Z_{k-N+1}^{k-1}, Y_0^{k-1}\middle|\, z_{k-N}\right) = p\left(Y_0^{k-1}\middle|\, Z_{k-N}^{k-1}\right) p\left(Z_{k-N+1}^{k-1}\middle|\, z_{k-N}\right). \tag{6.4}$$

We substitute Equation (6.4) into (6.3b), and reorganize the expressions to form

$$p\left(Z_{k-N}^{k-1}\middle|\, Y_0^{k-1}\right) = \frac{p\left(z_{k-N}|\, Y_0^{k-1}\right)}{p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)} p\left(Y_{k-N}^{k-1}\middle|\, Z_{k-N}^{k-1}\right) p\left(Z_{k-N+1}^{k-1}\middle|\, z_{k-N}\right), \tag{6.5}$$

where we have substituted Equations (6.6) to simplify the density functions. Note that the latter expressions are possible because of the Markov property, and because measurements are independent from each other, and they are given by

$$p\left(Y_0^{k-1}\middle|\, z_{k-N}\right) = p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right) p\left(Y_0^{k-N-1}\right) \tag{6.6a}$$

$$p\left(Y_0^{k-1}\middle|\, Z_{k-N}^{k-1}\right) = p\left(Y_{k-N}^{k-1}\middle|\, Z_{k-N}^{k-1}\right) p\left(Y_0^{k-N-1}\right). \tag{6.6b}$$

The density functions $p\left(Y_{k-N}^{k-1}\middle|\, Z_{k-N}^{k-1}\right)$ and $p\left(Z_{k-N+1}^{k-1}\middle|\, z_{k-N}\right)$ in Equation (6.5) can be treated similarly to Eqs. (4.3) and (4.5), respectively. Thus we can write

$$p\left(Y_{k-N}^{k-1}\middle|\, Z_{k-N}^{k-1}\right) = \prod_{l=k-N}^{k-1} p\left(y_l|\, z_l\right) \tag{6.7a}$$

$$p\left(Z_{k-N+1}^{k-1}\middle|\, z_{k-N}\right) = \prod_{l=k-N}^{k-2} p\left(z_{l+1}|\, z_l\right). \tag{6.7b}$$

Finally, we now substitute Equations (6.7) and (6.5) into (6.2) to obtain the MHE with smoothed arrival cost. This is given by

$$p\left(Z_{k-N}^{k}\middle|\, Y_0^{k}\right) = \mathcal{K}_2 \frac{p\left(z_{k-N}|\, Y_0^{k-1}\right)}{p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)} \left[\prod_{l=k-N}^{k} p\left(y_l|\, z_l\right)\right] \left[\prod_{l=k-N}^{k-1} p\left(z_{l+1}|\, z_l\right)\right], \tag{6.8}$$

where the arrival cost term is represented by

$$\frac{p\left(z_{k-N}|\, Y_0^{k-1}\right)}{p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)}. \tag{6.9}$$

Note that in the numerator of this expression we have the smoothed PDF, while the denominator is a correction term that is used to not overweight the measurements that are shared between the smoothed estimate and the horizon window. This is illustrated in Fig. 6.1, where the shared information is represented by the shadowed area, and the dot represents the smoothed state estimate used to predict the initial condition of the system.

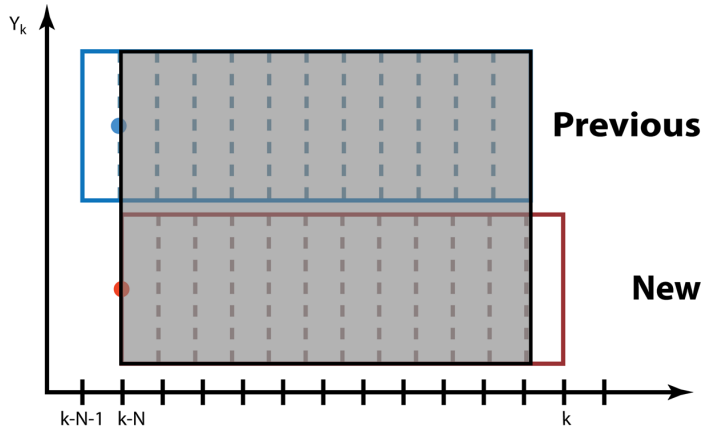



Figure 6.1: Illustration of the shared information between smoothed state estimate and the horizon window.

Finally, to calculate the state estimates we maximize the probability of the states in the horizon given the measurements, in other words we maximize the PDF in (6.1). However, since the noise variables have been assumed Gaussian, maximizing Equation (6.8) can be computed with

$$\min_{z_{k-N}} \quad \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k} v_l^T R_l^{-1} v_l + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l \tag{6.10a}$$

$$\text{s.t.} \quad z_{l+1} - f\left(z_l\right) - w_l = 0 \tag{6.10b}$$

$$y_l - h\left(z_l\right) - v_l = 0 \tag{6.10c}$$

$$z^{LB} \leq z_l \leq z^{UB}, \tag{6.10d}$$

where a further simplification is done by assuming that the densities in the arrival cost term are Gaussian. Also, expressions $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$ in the objective function (Eq. (6.10a)) are generated using the data that the two windows share (see Fig. 6.1), that is $Y_{k-N}^{k-1}$. These values can then be computed directly using the data and the model, and updating them is therefore not a problem. In Appendix D we show expressions that can be used to generate these terms. On the other hand, the arrival cost term parameters need to be generated In this case, the initial condition is predicted with the smoothed MHE solution at sample time $k-1$. Following Tenny and Rawlings [35], the covariance associated with the initial condition is computed using the EKF and EKS equations in combination with a linearized version of the model. To do so, the initial prior covariance $M_{k-N-1|k-N-2}$ is propagated forward up to sample time $k-1$ using

$$\begin{aligned} M_{k+1|k} &= A_k P_{k|k} A_k^T + Q_k \\ P_{k+1|k+1} &= M_{k+1|k} - M_{k+1|k} H_{k+1}^T \left(H_{k+1} M_{k+1|k} H_{k+1}^T + R_{k+1}\right)^{-1} H_{k+1} M_{k+1|k}, \end{aligned} \tag{6.11}$$

where $A_l = \nabla_z f\left(z_l\right)^T$ and $H_l = \nabla_z h\left(z_l\right)^T$. The final posterior covariance $\Pi_{k-1|k-1}$ is smoothed backwards to generate $\Pi_{k-N|k-1}$ with (i.e., EKS equation)

$$\Pi_{l|k} = \Pi_{l|l} + \Pi_{l|l} A_l^T M_{l+1|l}^{-1} \left(\Pi_{l+1|k} - M_{l+1|l}\right) M_{l+1|l}^{-1} A_l \Pi_{l|l}, \tag{6.12}$$

where the same definitions apply for $A_l$. The propagation of the covariances is illustrated in Figure 6.2. However, it is important to note that this propagation requires multiple matrix inversions, and these are repeated at each sample time in the horizon. This could be very computationally expensive for a large scale system, and more importantly it is possible to show that the smoothed covariance information can be extracted directly from the KKT conditions of NLP (6.10), which is the subject of the following sections.

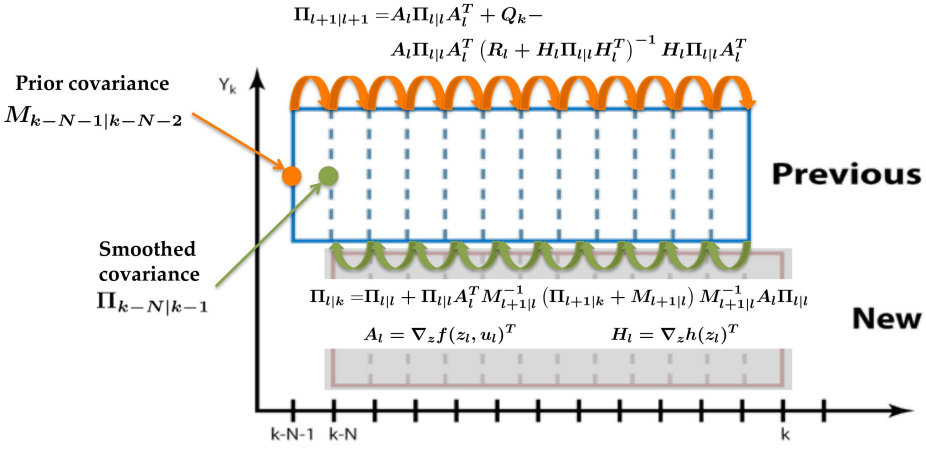



Figure 6.2: Illustration of the propagation of covariances in the smoothing arrival cost approach.

### 6.1.1 Optimality Conditions of MHE

In this section we show how the linearized optimality conditions of NLP (6.10) relate to the Riccati equations used in EKF and EKS (Eqs. (6.11) and (6.12)). This analysis will help us show that it is possible to extract the covariance information from the linearized KKT conditions of the NLP by using ideas from optimal sensitivity of the solutions. Furthermore, we show that this forward and backward propagation of the covariance matrices is done internally in the KKT conditions of MHE at the solution. Thus, we can avoid the expensive propagations using the aforementioned Riccati equations, since this is taken care of by the optimizer.

We establish these relationships using the sensitivity ideas implemented on the solver IPOPT [67], and using the KKT conditions of NLP (6.10). The KKT conditions are found by finding stationary points of the Largrange function

$$\begin{aligned}\mathcal{L} =& \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \\ & \frac{1}{2}\sum_{l=k-N}^{k}\left\|y_l - h\left(z_l\right)\right\|_{R_l^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l + \sum_{l=k-N}^{k-1}\lambda_{l+1}^T\left[z_{l+1}-f\left(z_l\right)-w_l\right]\end{aligned} \tag{6.13}$$

where $\lambda_l$ represent the vectors of Lagrange multipliers associated with the model constraints (Eq. (6.10b)). Thus, the KKT conditions are given by Equations (6.14a)-(6.14e). Also, to simplify the discussion, we have dropped the bound inequalities, which we assume are handled implicitly through the logarithmic barrier function in IPOPT [46].

$$\begin{aligned}\nabla_{z_{k-N}}\mathcal{L} =& \Pi^{-1}_{k-N|k-1}\left(z_{k-N}-\hat{z}_{k-N}\right) + \mathcal{O}^T\mathcal{W}^{-1}\left(\mathcal{Y}-\mathcal{O}z_{k-N}\right) - \\ & A^T_{k-N}\lambda_{k-N+1} - H^T_{k-N}R^{-1}_{k-N}\left[y_{k-N}-h\left(z_{k-N}\right)\right] = 0 \qquad (6.14a)\\ \nabla_{z_l}\mathcal{L} =& -H_l^T R_l^{-1}\left[y_l - h\left(z_l\right)\right] + \lambda_l - A_l^T\lambda_{l+1} = 0, \qquad (6.14b)\\ & \forall \quad l = k-N+1,\ldots,k-1 \\ \nabla_{z_k}\mathcal{L} =& -H_k^T R_k^{-1}\left[y_k - h\left(z_k\right)\right] + \lambda_k = 0 \qquad (6.14c)\\ \nabla_{w_l}\mathcal{L} =& Q_l^{-1}w_l - A_l^T\lambda_{l+1} = 0, \qquad \forall \quad l = k-N,\ldots,k-1 \qquad (6.14d)\\ \nabla_{\lambda_l}\mathcal{L} =& z_{l+1} - f\left(z_l\right) - w_l = 0, \qquad \forall \quad l = k-N,\ldots,k-1 \qquad (6.14e)\end{aligned}$$

For the above equations, the following definitions apply: $A_l^T = \nabla_{z_l} f\left(z_l, w_l\right)$, and $H_l^T = \nabla_{z_l} h\left(z_l, v_l\right)$. NLP solvers apply Newton's method and compute search directions by linearizing the KKT conditions (6.14) around the current point. This leads to the following linear system:

$$P_{k-N}\Delta z_{k-N} - F_{k-N}\Delta w_{k-N} - A^T_{k-N}\Delta\lambda_{k-N+1} = -r_{z_{k-N}} \tag{6.15a}$$

$$P_l\Delta z_l - F_l\Delta w_l + \Delta\lambda_l - A^T_l\Delta\lambda_{l+1} = -r_{z_l}, \tag{6.15b}$$

$$\forall \quad l = k-N+1,\dots,k-1$$

$$P_k\Delta z_k + \Delta\lambda_k = -r_{z_k} \tag{6.15c}$$

$$-F^T_l\Delta z_l + W_l\Delta w_l - G^T_l\Delta\lambda_{l+1} = -r_{w_l}, \tag{6.15d}$$

$$\forall \quad l = k-N,\dots,k-1$$

$$\Delta z_{l+1} - A_l\Delta z_l - G_l\Delta w_l = -r_{\lambda_{l+1}}, \tag{6.15e}$$

$$\forall \quad l = k-N,\dots,k-1,$$

where we define $P_l = \nabla^2_{z_l}\mathcal{L}$, $W_l = \nabla^2_{w_l}\mathcal{L}$, $F_l = \nabla_{w_l z_l}\mathcal{L}$, and $G^T_l = \nabla_{w_l} f(z_l, w_l)$. For the purpose of analysis we obtain a recursive solution of the above KKT system. This is achieved through a forward Riccati decomposition of the system, which leads to the following explicit solution of (6.15):

$$\Delta z_k = -\Pi_k\left(M^{-1}_k r_{M_k} + r_{z_k}\right) \tag{6.16a}$$

$$\Delta\lambda_{l-1} = M^{-1}_{l-1}\left(\Delta z_{l-1} + r_{M_{l-1}}\right) \tag{6.16b}$$

$$\Delta z_{l-1} = \Pi_{l-1}\left(F_{l-1}W^{-1}_{l-1}G^T_{l-1} + A^T_{l-1}\right)\Delta\lambda_l -$$
$$\Pi_{l-1}\left(F_{l-1}W^{-1}_{l-1}r_{w_{l-1}} + M^{-1}_{l-1}r_{M_{l-1}} + r_{z_{l-1}}\right) \tag{6.16c}$$

$$\Delta w_{l-1} = W^{-1}_{l-1}F^T_{l-1}\Delta z_{l-1} + W^{-1}_{l-1}G^T_{l-1}\Delta\lambda_l - W^{-1}_{l-1}r_{w_{l-1}} \tag{6.16d}$$

$$\forall \quad l = k,\dots,k-N,$$

where the following definitions are used

$$P_k = \left(P_k + M_k^{-1}\right)^{-1} \tag{6.17a}$$

$$P_{k-N} = \left(P_{k-N} - F_{k-N} W_{k-N}^{-1} F_{k-N}^T\right)^{-1} \tag{6.17b}$$

$$P_l = \left(P_l - F_l W_l^{-1} F_l^T + M_l^{-1}\right)^{-1} \tag{6.17c}$$

$$M_{l+1} = \left(A_l + G_l W_l^{-1} F_l^T\right) \Pi_l \left(A_l^T + F_l W_l^{-1} G_l^T\right) + G_l W_l^{-1} G_l^T \tag{6.17d}$$

$$\forall \quad l = \; k - N + 1, \ldots, k - 1$$

and finally

$$r_{M_l} = r_{\lambda_l} + G_{l-1} W_{l-1}^{-1} r_{w_{l-1}} + \left(A_{l-1} + G_{l-1} W_{l-1}^{-1} F_{l-1}^T\right) \Pi_{l-1} \left(F_{l-1} W_{l-1}^{-1} r_{w_{l-1}}^T + M_{l-1}^{-1} r_{M_{l-1}} + r_{z_{l-1}}\right) \tag{6.18a}$$

$$r_{M_{k-N+1}} = r_{\lambda_{k-N+1}} + G_{k-N} W_{k-N}^{-1} r_{w_{k-N}} + \left(A_{k-N} + G_{k-N} W_{k-N}^{-1} F_{k-N}^T\right) \Pi_{k-N} \left(F_{k-N} W_{k-N}^{-1} r_{w_{k-N}}^T + r_{z_{k-N}}\right) \tag{6.18b}$$

$$\forall \quad l = \; k - N + 2, \ldots, k.$$

Note that Equations (6.16) propagate the states in the horizon, and for this we use the matrices formed with Equations (6.17). Furthermore, the latter are the forward Riccati equations that propagate the covariance forward in time (in particular see Eqs. (6.17a-6.17d)). In fact, if the dynamic system is linear and the noises are Gaussian, it is possible to show that Equations (6.17)(a-d) represent the forward propagation of the covariances used in the Kalman Filter. In addition, for a nonlinear system with additive noise, Equations (6.17) correspond to the EKF equations used for the propagation of the covariance matrices (if we set $F_l = 0$). However, note that second order information is used for both covariance and state propagation, which is not commonly done in the design of Extended Kalman Filters. Thus, MHE, with an NLP solver that uses exact first and second order information, has the added benefit of including said derivative information in the calculation of the states and covariances. This shows that the forward propagation of the covariances is done

in the optimality conditions of the problem. It is even possible to extract these matrices directly from the linearized KKT system as shown in Zavala *et al.* [87]. However, we still need the smoothed covariance, and in the following subsections we show that this can also be extracted from the linearized optimality conditions of NLP (6.15).

### 6.1.2 Covariance-Reduced Hessian Relation

The relationship between the reduced Hessian and the covariance of the smoothed (and filtered) state estimates is summarized in the following Property.

**Property 6.1** (**Covariance of state estimates.**): *Assume that at the solution of* (6.10) *there are no active bound constraints, and the linear independence constraint qualification (LICQ) and sufficient second order condition hold. Then the inverse of the reduced Hessian of NLP* (6.10) *is an approximation of the smoothed and filtered covariance of the state estimates. In other words, we have that*

$$Cov\left(Z_{k-N}^{k}\right) \approx \left(\mathbb{Z}^{T}\nabla_{x}^{2}\mathcal{L}\mathbb{Z}\right)^{-1} = \mathcal{H}_{\mathcal{R}}^{-1} \tag{6.19}$$

*where $\mathbb{Z}$ is the null space of the Jacobian of constraints* (6.10b)-(6.10c)*.*

**Proof of Property 6.1:** The proof follows by introducing stochastic perturbations of the measurements and the state estimates in the linearized optimality conditions at the optimal point. Then solving for the state estimates and computing the expected value of the outer product of the change in the state estimates. To simplify the notation we make a change of variables with the following definitions

$$y^{T} = \left[y_{k-N}^{T}, \ldots, y_{k}^{T}\right] \qquad\qquad \theta^{T} = \left[z_{k-N}^{T}, \ldots, z_{k}^{T}\right]$$

and

$$c(y,\theta) = \begin{bmatrix} y_{k-N} - h(z_{k-N}) - v_{k-N} \\ \vdots \\ y_k - h(z_k) - v_k \\ z_{k-N+1} - f(z_{k-N}) - w_{k-N} \\ \vdots \\ z_k - f(z_{k-1}) - w_{k-1} \end{bmatrix} \qquad \begin{aligned} V_y &= \text{diag}\left(R_{k-N}, \ldots, R_k, Q_{k-N}, \ldots, Q_{k-1}\right) \\ &= \mathbb{E}\left[\delta_y \delta_y^T\right] \\ V_\theta &= \text{diag}\left(\Pi_{k-N|k-1}, 0, \ldots, 0\right) \\ &= \mathbb{E}\left[\delta_\theta \delta_\theta^T\right] \end{aligned} .$$

Also note that $\mathbb{E}\left[\delta_y \delta_\theta^T\right] = 0$, and thus with the previous definitions we rewrite NLP (6.10) as

$$\min_{\theta,y} \ \frac{1}{2}\left[(y-\hat{y})^T V_y^{-1}(y-\hat{y}) + (\theta-\hat{\theta})^T \hat{V}_\theta^{-1}(\theta-\hat{\theta})\right] \tag{6.20a}$$

$$\text{s.t.} \ \ c(\theta, y) = 0, \tag{6.20b}$$

and the optimality conditions of this NLP are given by the following system of nonlinear equations evaluated at the optimal point

$$\nabla_y \mathcal{L} = V_y^{-1}(y^* - \hat{y}) + \nabla_y c(\theta^*, y^*)\lambda^* = 0 \tag{6.21a}$$

$$\nabla_\theta \mathcal{L} = V_\theta^{-1}\left(\theta^* - \hat{\theta}\right) + \nabla_\theta c(\theta^*, y^*)\lambda^* = 0 \tag{6.21b}$$

$$c(\theta^*, y^*) = 0. \tag{6.21c}$$

From the above equations and LICQ [21] we can define $\lambda^*$ as

$$\lambda^* = -\left(A^T A\right)^{-1} A^T \begin{bmatrix} V_y^{-1}\left(y^* - \hat{y}\right) \\ V_\theta^{-1}\left(\theta^* - \hat{\theta}\right) \end{bmatrix}, \tag{6.22}$$

where $A^T = \nabla c(\theta^*, y^*)$. In addition, a suitable maximum likelihood assumption on (6.20) is that $\mathbb{E}[y^* - \hat{y}] = 0$ and $\mathbb{E}\left[\theta^* - \hat{\theta}\right] = 0$. Therefore, this leads to $\mathbb{E}[\lambda^*] = 0$.

Using the expected value $\lambda^* = 0$, and introducing stochastic perturbations ($\delta_y$, $\delta_\theta$) of the measurements ($\hat{y}$) and data ($\hat{\theta}$), the linearized KKT conditions given by (6.21) are

$$\begin{bmatrix} V_y^{-1} & 0 & A_y^T \\ 0 & V_\theta^{-1} & A_\theta^T \\ A_y & A_\theta & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\theta \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} V_y^{-1}\delta_y \\ V_\theta^{-1}\delta_\theta \\ 0 \end{bmatrix}. \tag{6.23}$$

where the null space is defined with

$$\mathbb{Z} = \begin{bmatrix} -A_y^{-1}A_\theta \\ \mathbb{I} \end{bmatrix}, \tag{6.24}$$

and we have that

$$\mathbb{Z}^T \begin{bmatrix} A_y^T \\ A_\theta^T \end{bmatrix} = 0.$$

Then, assuming $A_y$ is non-singular, we can use the last row of (6.23) to solve for $\Delta y = A_y^{-1}A_\theta\Delta\theta$. We then substitute the latter into (6.23), we drop the last row, and pre-multiply by $\mathbb{Z}^T$ to get

$$\mathbb{Z}^T \begin{bmatrix} V_y^{-1} & 0 \\ 0 & V_\theta^{-1} \end{bmatrix} \begin{bmatrix} -A_y^{-1}A_\theta \\ \mathbb{I} \end{bmatrix} \Delta\theta = \mathbb{Z}^T \begin{bmatrix} V_y^{-1}\delta_y \\ V_\theta^{-1}\delta_\theta \end{bmatrix}. \tag{6.25}$$

Finally, we solve for $\Delta\theta$, which yields

$$\Delta\theta = \left\{ \mathbb{Z}^T \begin{bmatrix} V_y^{-1} & 0 \\ 0 & V_\theta^{-1} \end{bmatrix} \mathbb{Z} \right\}^{-1} \mathbb{Z}^T \begin{bmatrix} V_y^{-1}\delta_y \\ V_\theta^{-1}\delta_\theta \end{bmatrix} = \mathcal{H}_{\mathcal{R}}^{-1}\mathbb{Z}^T \begin{bmatrix} V_y^{-1}\delta_y \\ V_\theta^{-1}\delta_\theta \end{bmatrix}. \tag{6.26}$$

We can now compute the covariance of the state estimates with the expected value $\mathbb{E}\left[\Delta\theta\Delta\theta^T\right]$,

$$\mathbb{E}\left[\Delta\theta\Delta\theta^T\right] = \mathcal{H}_{\mathcal{R}}^{-1}\mathbb{Z}^T \begin{bmatrix} V_y^{-1} & 0 \\ 0 & V_\theta^{-1} \end{bmatrix} \mathbb{E}\left\{ \begin{bmatrix} \delta_y\delta_y^T & \delta_y\delta_\theta^T \\ \delta_\theta\delta_y^T & \delta_\theta\delta_\theta^T \end{bmatrix} \right\} \begin{bmatrix} V_y^{-1} & 0 \\ 0 & V_\theta^{-1} \end{bmatrix} \mathbb{Z}\mathcal{H}_{\mathcal{R}}^{-1}. \tag{6.27}$$

Finally, after canceling out terms, we establish that

$$\mathbb{E}\left[\Delta\theta\Delta\theta^T\right] = \text{Cov}\left(Z_{k-N}^k\right) \approx \mathcal{H}_{\mathcal{R}}^{-1}, \tag{6.28}$$

and this concludes the proof of Property 6.1.

□

The approximation shown in Property 6.1 is similar to using the EKS (e.g., Equation (6.12)). This is easily verified by extracting the reduced Hessian from the linearized optimality conditions in Eq. (6.15), or the recursive solution given by (6.16)-(6.18).

For example we can extract the covariance of the filtered state estimate by setting $r_{z_k} = -\mathbb{I}$, $r_{z_l} = r_{w_l} = r_{\lambda_{l+1}} = 0$ for $l = k-N, \ldots, k-1$, and solving for $\Delta z_k$ using Equations (6.16)-(6.18). Thus, we have that $r_{M_k} = 0$, and from (6.16a)

$$\Delta z_k = \Pi_k. \tag{6.29}$$

As another example, the covariance of the smoothed state estimate at $k-2$ can be computed in a similar fashion. If we set $r_{z_{k-2}} = -\mathbb{I}$, $r_{w_l} = r_{\lambda_{l+1}} = 0$ for $l = k-N, \ldots, k-1$, and $r_{z_l} = 0$ for all $l \neq k-2$. Therefore, we have that $r_{M_l} = 0$ for $l \leq k-2$, and, to simplify notation, we define $\Gamma_l = \Pi_l \left(A_l^T + F_l W_l^{-1} G_l^T\right) M_{l+1}^{-1}$. Now we can solve for $\Delta z_k$, $\Delta\lambda_k$, $\Delta z_{k-1}$, $\Delta\lambda_{k-1}$, then after making all the substitutions into $\Delta z_{k-2}$ and grouping terms, we get

$$\Delta z_{k-2} = \Pi_{k-2} + \Gamma_{k-2}\left[\Pi_{k-1} - M_{k-1} + \Gamma_{k-1}\left(\Pi_k - M_k\right)\Gamma_{k-1}^T\right]\Gamma_{k-2}^T. \tag{6.30}$$

This is further simplified if we define $\Pi_{k-1|k} = \Pi_{k-1} + \Gamma_{k-1}\left(\Pi_k - M_k\right)\Gamma_{k-1}^T$, thus, we finally get

$$\Delta z_{k-2} = \Pi_{k-2} + \Gamma_{k-2} \left( \Pi_{k-1|k} - M_{k-1} \right) \Gamma_{k-2}^T. \tag{6.31}$$

Note that Equation (6.31) is analogous to (6.12), and the same could be done for all $\Delta z_l$, where similar expressions can be found. Furthermore, when using the reduced Hessian to approximate the covariance matrices, as described above, we use second order information, which is not usually done in typical EKS (or EKF) formulations. Finally, depending on the characteristics of the process and measurements models, we have the two following corollaries to Property 6.1.

**Corollary 6.1** (Covariance of nonlinear system with additive noise.)**:** *For a nonlinear system with additive noise models (i.e., $z_l = f(z_{l-1}) + w_{l-1}$ and $y_l = h(z_l) + v_l$), the covariance approximation in Property 6.1 is the Extended Kalman Smoother (with second order information).*

**Corollary 6.2** (Covariance of unconstrained linear Gaussian systems.)**:** *For a linear unconstrained Gaussian system the approximation given in Property 6.1 is exact. In other words, the inverse of the reduced Hessian is the Kalman Smoothing covariance.*

Therefore, using Property 6.1 we can approximate the smoothed covariance of the state estimates in the horizon through the inverted reduced Hessian. Note, however, that IPOPT does not naturally form this matrix as part of the solution algorithm, thus we must extract it directly from the KKT matrix using the strategy described in Section 2.2.3. This methodology requires a single back-solve per column in the covariance matrix. Nevertheless, at the solution we can freeze the already factorized KKT matrix from IPOPT, and simply solve the linear system (2.5) using different right hand sides (one for each column). This is much cheaper than propagating matrices forwards and backwards using EKF and EKS.

### 6.1.3 Sensitivity Based MHE

In the previous sections we described how to extract covariance information from the optimality conditions of the NLP associated with the MHE problem (Eq. (6.10)). Using these methods, we can approximate the arrival cost parameters of the MHE formulation de-

scribed in Eq. (6.10), using the smoothed information taken directly from the optimal solution of the NLP. This formulation takes the prediction of the initial condition for the MHE at sample time $k$ from the solution of the previous NLP at sample time $k-1$, as illustrated in Figure 4.4. In other words, we use $\hat{z}_{k-N|k-1} = z^*_{k-N+1|k-1}$ to approximate the mean of the arrival cost PDF. On the other hand, we avoid the forward and backward propagation of covariance matrices by extracting this information directly from the reduced Hessian at the optimal point, as described in Section 6.1.2.

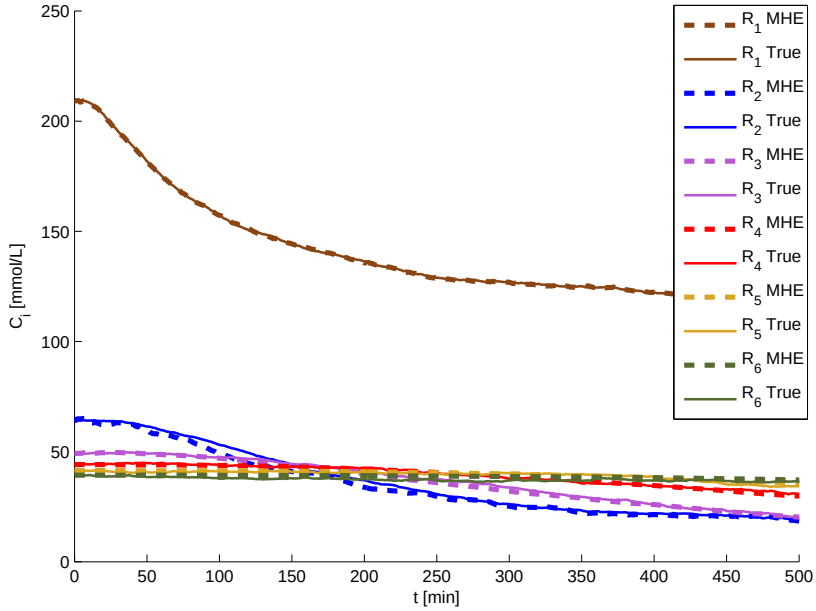
## 6.2 Simulation Examples

In this Section we use two examples to illustrate the use and performance of the methods described in this chapter. The first case is used as a proof of concept example to illustrate the benefits of avoiding the propagation of matrices with EKF and EKS. The second example deals with a large scale binary distillation column for which MHE is used to estimate liquid molar holdups and liquid compositions for the light component, for all trays.

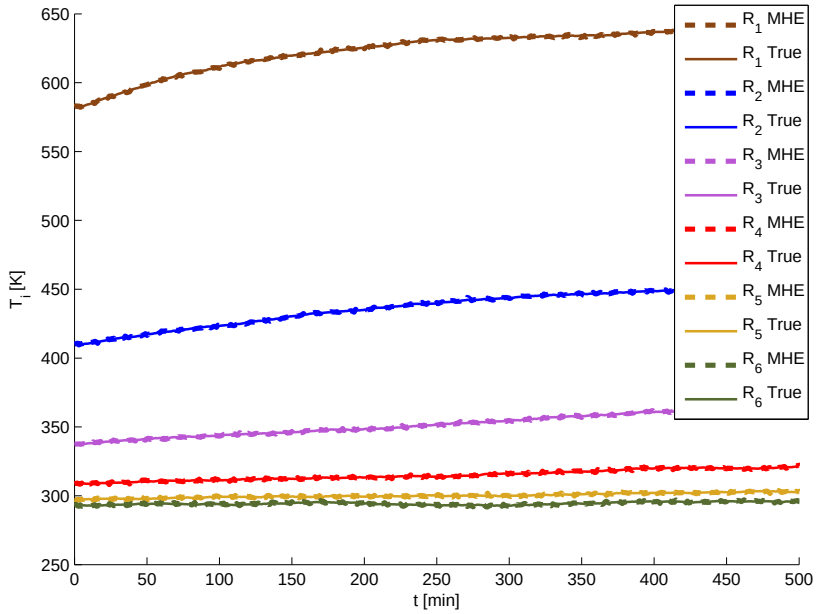
### 6.2.1 CSTR Network

To illustrate the benefits of approximating the arrival cost through the reduced Hessian we use a CSTR network which is modeled using the system of ODEs described in Section B.2. Each reactor in the network has two states: the concentration of component $A$ and the temperature of the reactor. The latter is also used as the measured state. Therefore by increasing the number of reactors in the network, the number of states increases by 2 and the number of measurements increases by 1. The properties of the state and measurement noise used here are summarized in Table 6.1.

Figure 6.3 shows the true and estimated profiles of the concentrations and temperatures when there are 6 reactors in the network, 20 measurements in the horizon. In this example the estimation error is very small due to the small amounts of added noise, and therefore most curves are overlapping each other. However, it is still possible to note that




(a)




(b)

Figure 6.3: Concentration (a) and Temperature (b) profiles with 6 reactors in the CSTR network example.

| State | Model variance | Measurement Variance |
|---|---|---|
| $C_{A,i}$ | $10^{-1}$ | – |
| $T_i$ | $10^{-1}$ | 0.5 |

Table 6.1: Parameters of noise variables used in Example 6.2.1.

the estimate is able to estimate the states properly. Moreover, Figure 6.4 shows the ap-

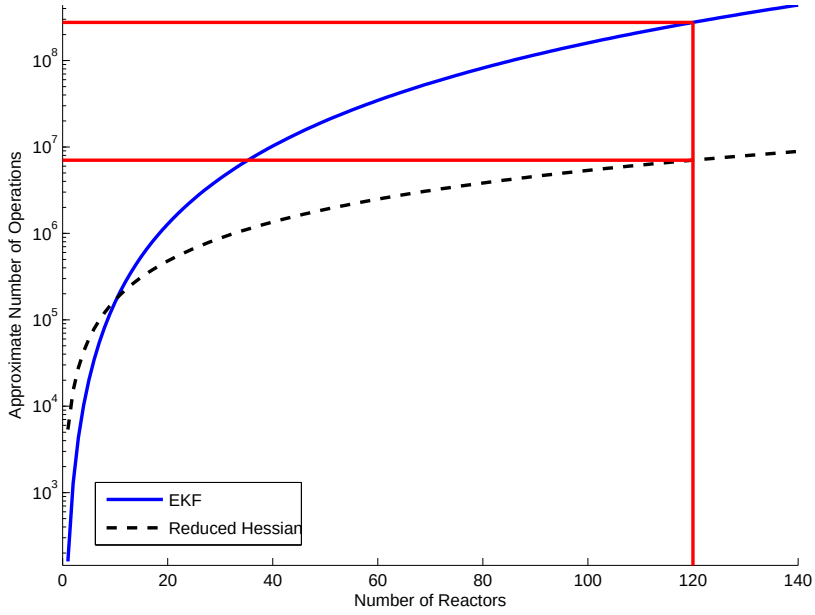



Figure 6.4: Approximate number of operations required to estimate the arrival cost covariance matrix for the CSTR network example.

proximate number of operations required to generate the approximation of the arrival cost covariance with respect to the number of reactors in the network, for this example with 6 reactors and 20 measurements. The number of operations are computed assuming cubic complexity for matrix inversions in EKF/EKS (i.e., $o\left(n_z^3 + n_y^3\right)$). For the reduced Hessian calculations, since the KKT matrix is already factorized, we assumed the complexity scales as $o\left(n_x^{1.5}\right)$ where $n_x$ is the number of variables in the NLP [87]. Note that under 10 re-

actors (i.e., less than 20 states) using EKF/EKS performs slightly better than the reduced Hessian approach. However, this could change if the ratio between states and measurements changes. On the other hand, for larger systems the reduced Hessian approach is much more efficient. For example, if we had 120 reactors in the network propagating matrices forward and backward requires over 4 orders of magnitude more operations than extracting the reduced Hessian from the KKT conditions.

### 6.2.2 Large-Scale Distillation Example

In this section we show the simulation results of the implementation of MHE using the smoothed arrival cost update with the covariance obtained from the reduced Hessian. The model of the distillation column is described in Appendix B.4. For this example we have assumed that the measurement vector consists of the temperatures and liquid volume holdups of all the trays, and the states of interest are given by the compositions of the light component in the liquid phase and the liquid molar holdup on each tray. For this case study, we have considered that the sample time is 60 seconds, and also, we use a horizon length of 10 measurements. In addition, the parameters of the noise variables are summarized in Table 6.2.

Since we have considered a horizon length of 10 measurements and we are using Radau collocation to transform the DAE system into a discrete time model, then the NLP that is solved at each sample time has 21642 variables and 21642 equality constraints. However, this large scale problem is solved in an average of 42.38 CPU seconds which is below the sample time. Furthermore, the calculation of the reduced Hessian takes an average of 1.84 CPU seconds. However, note that it takes 84 back-solves of the KKT system, one for each state, to calculate the covariance matrix.

In Figures 6.5 and 6.6 the state estimates are compared with the true trajectory of the states. Also, the absolute value of the the estimation error is shown, which is calculated with $\epsilon_{j,k} = \left|\zeta_{j,k}^{\text{real}} - \zeta_{j,k}^{\text{estimated}}\right| / \zeta_{j,k}^{\text{real}}$ for the $j$-th state at the $k$-th sample time. Note that

| | Model noise variance | Measurement variance |
|---|---|---|
| $T_i$ | – | $6.25 \times 10^{-2}$ |
| $V^m_{N_T+1}$ | – | $10^{-8}$ |
| $x_i$ | $10^{-5}$ | – |
| $M_0$ | 10 | – |
| $M_i$ | 1 | – |
| $M_{N_T+1}$ | 5 | – |

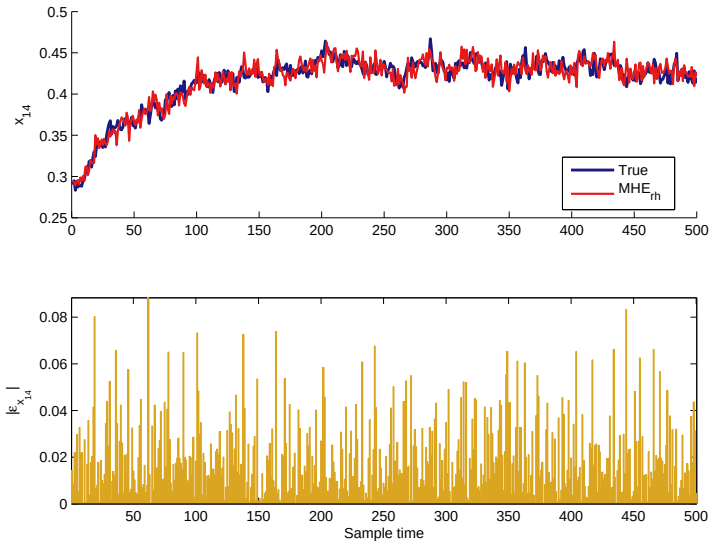
Table 6.2: Summary of parameters for the noise variables for the distillation column example in Chapter 6.

MHE is able to estimate the states with small error, which is important if these estimates are to be used within an NMPC application.

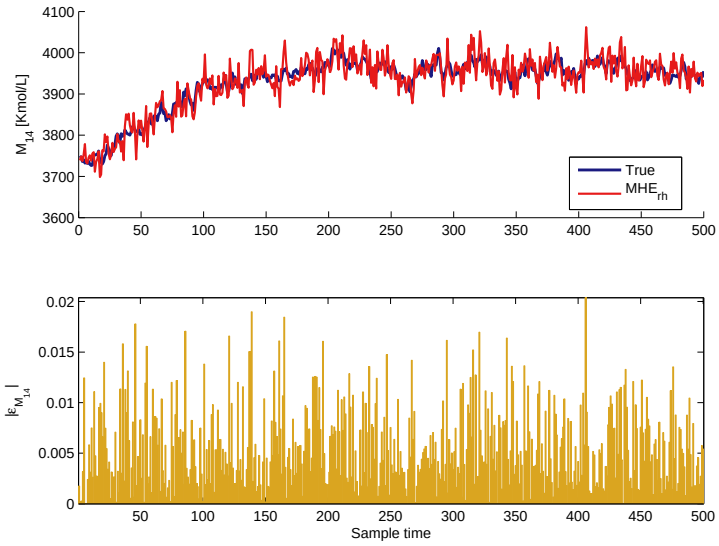
## 6.3 Conclusions

It is possible to extract the mean and covariance information for the arrival cost term directly from the optimal point of the NLP using sensitivity theory. This information can be extracted very cheaply and efficiently from the KKT system using *sIPOPT*, which for large-scale systems is much faster than propagating matrices forwards and backwards. Moreover, this approach avoids oscillations in the estimates, since the prediction of the initial condition is taken from a previously solved MHE problem. Thus, the current horizon and the initial condition share a lot more information than when updating it using the filtering approach (c.f., Fig. 6.1).

It is possible to use the information of the previously solved NLP so that the covariance of the state estimates can be approximated using the inverse of the reduced Hessian of the NLP. If the system and measurement models are linear and Gaussian this approximation reduces to the Kalman Smoother, and MHE generates the Kalman filter solution. On

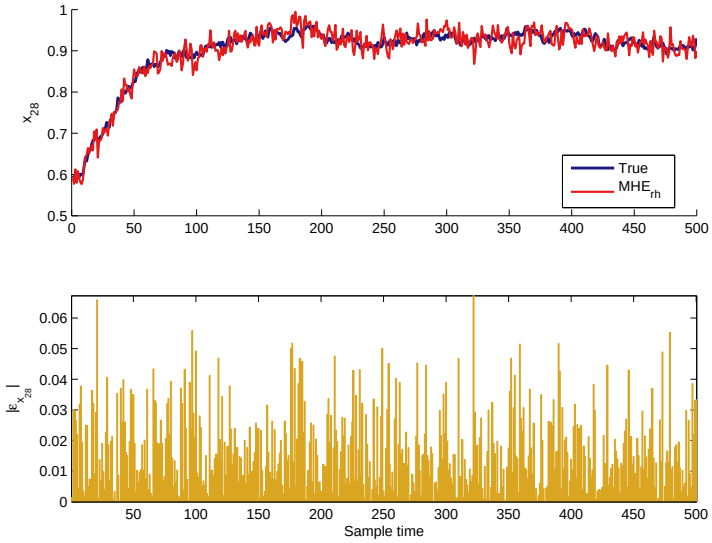



(a)

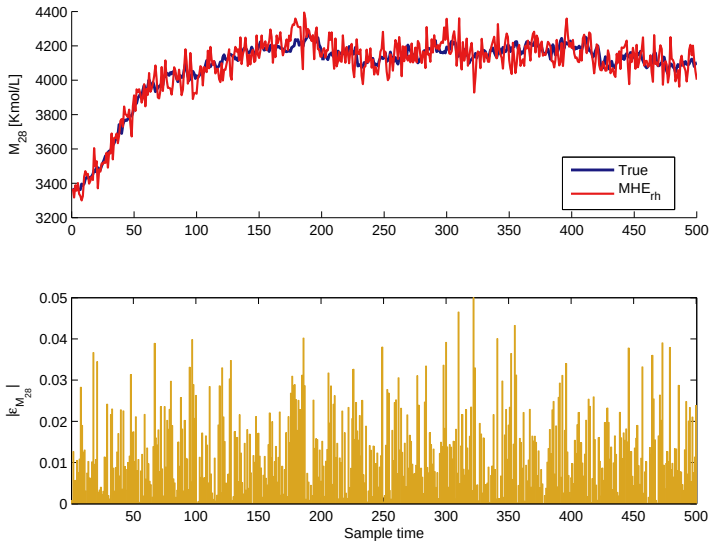



(b)

Figure 6.5: Liquid composition for light component (a) and liquid molar holdup (b) on tray 14.




(a)




(b)

Figure 6.6: Liquid composition for light component (a) and liquid molar holdup (b) on tray 28.

the other hand, for nonlinear systems with additive noise, this approximation reduces to the EKS and EKF. Moreover, extracting the reduced Hessian information from the optimality conditions used in Interior Point solvers can be done cheaply through a series of back-solves. This approach is much cheaper than using EKF and EKS to propagate covariances forward and backward to generate the arrival cost information. In addition, since the solver has access to first and second derivatives computing the state estimates and covariance matrices makes use of second order information. This is an added advantage over traditional EKF formulations that only make use of first order derivatives.

# Chapter 7

# Multi-Rated State Estimation

In previous chapters we have considered the use of measurements that come at the same sample rate. However, in most chemical processes, some state variables (e.g., concentrations, molecular weights, etc.) can be measured infrequently and, probably, with some time delay. The most common state estimation methods usually use only the fast measurements under the assumption that all states of interest are observable through them. Unfortunately, this assumption does not always hold, and in some cases there are states that are not observable through the fast measurements alone. However, it is sometimes possible to combine different measurements that are available at different rates to change the observability properties of the system, and make some of the unobservable states observable [40, 41]. Therefore, methods that can handle multi-rated measurements with or without delays should be used to take advantage of the slower measurements to improve the quality of the state estimates or their observability. Moving Horizon Estimation provides a framework that allows to easily incorporate these infrequent observations because it uses a window of past measurements, where the slower ones can be introduced as they become available. Furthermore, since MHE handles bounds and constraints in a natural way through the solution of an NLP, a multi-rated constrained state estimator will be introduced here. The rest of this chapter is organized as follows: Section 7.1 summarizes the derivation of MHE with the smoothed update for the arrival cost, and introduces the multi-rated state estimator in that framework, and in Section 7.2 a simulation example is used to illustrate the benefits of combining different rated measurements for state estimation. Finally, in Section 7.3 we provide conclusions for this chapter.

## 7.1 Multi-rated MHE with Smoothing Update of Arrival Cost

In the previous section we describe a way to implement a fast MHE that uses the smoothed update of the arrival cost, and the covariance of the arrival cost is approximated through the inverse of the reduced Hessian. For that implementation, only measurements with the same sampling rate were considered. Nevertheless, it is still possible to use measurements that come at a slower rate and that may possibly be delayed. For example, in polymerization reactors, temperatures can be measured very quickly. In contrast, molecular weight information could be measured through Gel Permeation Chromatography (GPC). However, compared to temperatures, these measurements are much slower. Moreover, since the GPC takes some time to analyze the sample taken from the reactor, this measurement will be delayed. Therefore, we can have two classes of multi-rated signals: either the signals have different rates but are obtained instantaneously, or the slower measurements may be delayed. The two types of multi-rated signals are illustrated in Figure 7.1.

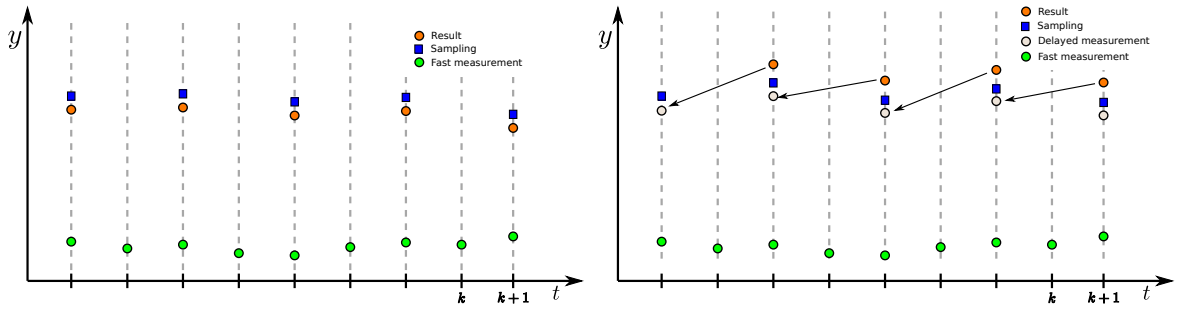



Figure 7.1: Illustration of multi-rated signals: (a) without delay (the sampling and measurement are simultaneous) and (b) with delay (sampling and measurement are obtained at different sampling times).

To derive the multi-rated MHE (MMHE) we use the derivation of the MHE estimator using the so-called smoothing update of the arrival cost described in Chapter 6, and also

in [35, 83]. In this formulation, the prediction of the initial condition of the plant at the beginning of the horizon (at time $k-N$) is given by the smoothed state estimate taken from the solution of the previous MHE at time $k-1$, that is $z_{k-N|k-1}$ (see Fig. 4.4). In addition, we are interested in the trajectory of the states in the horizon window ($Z_{k-N}^{k} = \{z_{k-N}, \dots, z_k\}$) given the measurements in the set $Y_0^k = \{y_0, \dots, y_k\}$. Thus, after applying the assumptions in Assumption 3.1 to the conditional probability density function of the states $p\left(Z_{k-N}^{k} \middle| Y_0^k\right)$ we get

$$p\left(Z_{k-N}^{k} \middle| Y_0^k\right) = \mathcal{K}_2 \frac{p\left(z_{k-N} \middle| Y_0^{k-1}\right)}{p\left(Y_{k-N}^{k-1} \middle| z_{k-N}\right)} \left[\prod_{l=k-N}^{k} p\left(y_l \middle| z_l\right)\right] \left[\prod_{l=k-N}^{k-1} p\left(z_{l+1} \middle| z_l\right)\right], \tag{7.1}$$

where $\mathcal{K}_2$ is a proportionality constant. The expressions on the right hand side of the equation represent the arrival cost term, and the process and measurement noise distributions, which are $p\left(z_{l+1} \middle| z_l\right)$ and $p\left(y_l \middle| z_l\right)$, respectively. These we know are Gaussian, from the assumptions described before. To avoid the complexity of using sample based filters to approximate the non-Gaussian arrival cost, as described in Chapter 5, here we use the reduced Hessian approach to generate the covariance information. Thus we assume a Gaussian PDF for the arrival cost term to simplify computations. This approach is described in detail in Chapter 6, therefore here we will summarize the main parts of the derivation to aid in the introduction of the multi-rated estimator. Applying logarithms to Equation (7.1), and after maximizing the probability density function of the states in the horizon we get

$$\min_{z_{k-N}} \quad \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k} v_l^T R_l^{-1} v_l + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l \tag{7.2a}$$

$$\text{s.t.} \quad z_{l+1} - f\left(z_l\right) - w_l = 0 \tag{7.2b}$$

$$y_l - h\left(z_l\right) - v_l = 0 \tag{7.2c}$$

$$z^{LB} \leq z_l \leq z^{UB}, \tag{7.2d}$$

where (7.2d) are bounds that allow us to constrain the state estimates to values that represent the physical properties of the system (e.g., concentrations must be positive or molar fractions are between 0 and 1). Note, also, that $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$ in the objective function (Eq. (7.2a)) are generated using the data that the two windows share (see Fig. 6.1), that is $Y_{k-N}^{k-1}$. These values can then be computed directly using the data and the model. In Appendix D we show expressions that can be used to generate these terms.

In MHE, since we have a history of past measurements, it is simple to include the signals that come at different sample rates. For example, in Figure 7.1b the slow measurements, can be placed in their proper locations in the measurement history. Note that even if some of these measurements are delayed, we can still place them in their corresponding sample times. In this case, however, the appropriate measurement covariance matrices must be used in each sample time.

For the Multi-Rate MHE (MMHE) we can define a vector of only fast measurements $y_l^F$ and a vector of fast and slow measurements $y_l^{SF}$. Therefore, assuming that the first available measurement is $y_0^F$ and that every second sampling time we have a slow and delayed measurement available, then we have the set of measurements $Y_0^k = \left\{y_0^F, y_1^{SF}, \ldots, y_{k-1}^F, y_k^F\right\}$. Note that here the last two measurement vectors only have information of the fast measurements. This is because the sample taken from the plant at time $k$ still needs to be processed, and the measurement information will not be available until time $k+2$. A

similar situation would happen if the first measurement had been of type $y_0^{SF}$ or if the slow measurements are available at a different rate. However, since the analysis shown below can be adapted to different measurement structures in a straightforward way, we will focus on the measurement set shown above. Also, we use the following conditional probability density functions in Equation (7.1):

$$p_l\left(y_l|\, z_l\right) = \begin{cases} p\left(y_l^F \middle|\, z_l\right) & \text{if} \quad l \in \mathbb{I}_F \\ p\left(y_l^{SF} \middle|\, z_l\right) & \text{if} \quad l \in \mathbb{I}_{SF} \end{cases}, \tag{7.3}$$

where $\mathbb{I}_F$ and $\mathbb{I}_{SF}$ represent the set of indices for the sample times with only fast measurements and fast and slow measurements, respectively. Note that the way MHE handles the delayed measurements is through these sets. In the case described here we have $\mathbb{I}_F = \{0, 2, \ldots, k-1, k\}$ and $\mathbb{I}_{SF} = \{1, 3, \ldots, k-2\}$, assuming $k$ is odd. However, these sets can be tailored to account for different types of measurement rate structures. Moreover, again here we emphasize that, even thought we may have sampled the plant at time $k$, the slow measurement information will not be available at that instant. Instead, because it is delayed, it will be available at time $k+2$, and this is why the last elements of set $Y_0^k$ are only fast measurements. Furthermore, after applying logarithms and maximizing the probability density function of the states (Eq. (7.1)), taking into account the availability of fast and slow measurements as described above, we would have

$$\min_{z_{k-N}} \quad \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l +$$

$$\frac{1}{2}\left(\sum_{\substack{l=k-N\\ l\in\mathbb{I}_F}}^{k}\left\|v_l^F\right\|_{\left(R_l^F\right)^{-1}} + \sum_{\substack{l=k-N\\ l\in\mathbb{I}_{SF}}}^{k}\left\|v_l^{SF}\right\|_{\left(R_l^{SF}\right)^{-1}}\right) \tag{7.4a}$$

$$\text{s.t.} \quad z_{l+1} - f\left(z_l\right) - w_l = 0 \tag{7.4b}$$

$$y_l^F - h^F\left(z_l\right) - v_l^F = 0 \qquad \forall\, l \in \mathbb{I}_F \tag{7.4c}$$

$$y_l^{SF} - h^{SF}\left(z_l\right) - v_l^{SF} = 0 \qquad \forall\, l \in \mathbb{I}_{SF} \tag{7.4d}$$

$$z^{LB} \leq z_l \leq z^{UB}, \tag{7.4e}$$

where $R_l^F$ and $R_l^{SF}$ represent the covariance matrices (of appropriate dimensions) for the measurement vector with only fast measurements and the one with both, respectively. Also, we would have measurement models for both types of measurement vectors, and these are given by Eqs. (7.4c) and (7.4d), respectively.

The arrival cost is approximated in a similar fashion as described in Chapter 6, where the initial condition is approximated with $\hat{z}_{k-N|k-1} = z^*_{k-N+1|k-1}$, i.e., taken from the solution of the NLP at time $k-1$. The covariance of the arrival cost is approximated using the inverse of the reduced Hessian using Property 6.1. Fortunately, this is a general approach that can handle multi-rated measurements, and it takes advantage of both slow and fast measurements to improve the approximation of the covariance. Moreover, because of the variable structure nature of this formulation, we need to be careful when constructing the expressions for $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$, and we need to make sure that the correct measurement covariance matrices are used in the objective function. Having the slow measurements can provide enough information to make some of the unobservable states (if any), observable [40, 41], which is important to be able to monitor the behavior of these states. Finally, since more data are being used when fast and slow measurements are combined, the multi-rated MHE should perform better in the presence of errors or large disturbances. In other

words, the multi-rated estimator will converge faster to the true states. The behavior of the proposed method will be shown in the next section through some simulation examples.

## 7.2 Simulation Examples

In this section we test the proposed estimator on two simulation examples. The first example deals with the polymerization of styrene, where we are interested in estimating the molecular weight distributions. The second example deals with a large scale binary distillation column where parameter and state estimation is done simultaneously. Here we compare the application of the MHE using only fast measurements (MHE$_f$), and using multi-rated measurements (MMHE).

The models used here are index 1 Differential Algebraic Equation (DAE) systems. However, the model used in (3.1) is discrete time system. Thus, to transform the continuous time DAE systems we apply orthogonal collocation on finite elements. More information on this techniques can be found in [22]. This increases the size of the optimization problems needed to find the state estimates, and therefore we use state-of-the-art NLP solvers. In particular we use IPOPT [46] to solve the NLPs, and also, we use sIPOPT [67] (an optimal sensitivity package specifically designed to work with IPOPT) to compute reduced Hessians.

### 7.2.1 Example 1: Styrene Polymerization Reactor

For this example the polystyrene Continuously Stirred Tank Reactor (CSTR) is considered. The model is described in Appendix B.3, the example was originally described in [40]. Here the states are $C_i$, $C_s$, and $C_m$ that represent the concentrations of the initiator, solvent, and monomer, respectively. Also, $T_R$ and $T_J$ that are the temperatures of the reactor and cooling water, and finally $\lambda_0$, $\lambda_1$, and $\lambda_2$ that are the first moments of the molecular weight distributions. For this case study the molecular weight averages are of particular importance since these are used to determine specific physical properties of the produced

materials [89]. It is therefore common to use these states as controlled variables. In addition, the molecular weight averages by weight ($M_W$) and number ($M_N$) can be calculated with

$$M_N = \frac{\lambda_1}{\lambda_0} \tag{7.5a}$$

$$M_W = \frac{\lambda_2}{\lambda_1} \tag{7.5b}$$

.

Here we assume that the fast measurements are given by the temperatures of the reactor and the cooling jacket. The molecular weight moments are considered to be the slow measurements, and these are obtained through GPC. We have also assumed that the fast sampling time is 6 min, while the slow sampling time is 12 min. In addition, there is a delay between the time when the sample is taken from the reactor and when the molecular weight moments are obtained. This delay is assumed equal or less than 12 min. The predicted state in the arrival cost for the first sample time is assumed to be very corrupted for the fast, single-rated MHE (MHE$_f$) and the multi-rated MHE (MMHE). This is done to illustrate the benefits of having the added information from the slow and delayed measurements. This scenario is not completely unrealistic, since it is possible that noise and unmeasured disturbances may shift the plant in such a way that previous estimates are away from the true values of the plant states. Therefore, the estimators used must be able to recover from such an error as fast as possible.

Table 7.1 shows the values of the covariance values for the states, as well as the initial value of the arrival cost covariances, used during the simulations. The NLP, using 20 measurements in the horizon and the discretized model, consists of 1,832 variables and 1,729 constraints. On average this problem is solved in 1.810 CPU s, and the computation of the arrival cost covariance takes 0.003 CPU s. In Figure 7.2 we compare the estimated states using only the fast measurements with the estimates obtained using the combined

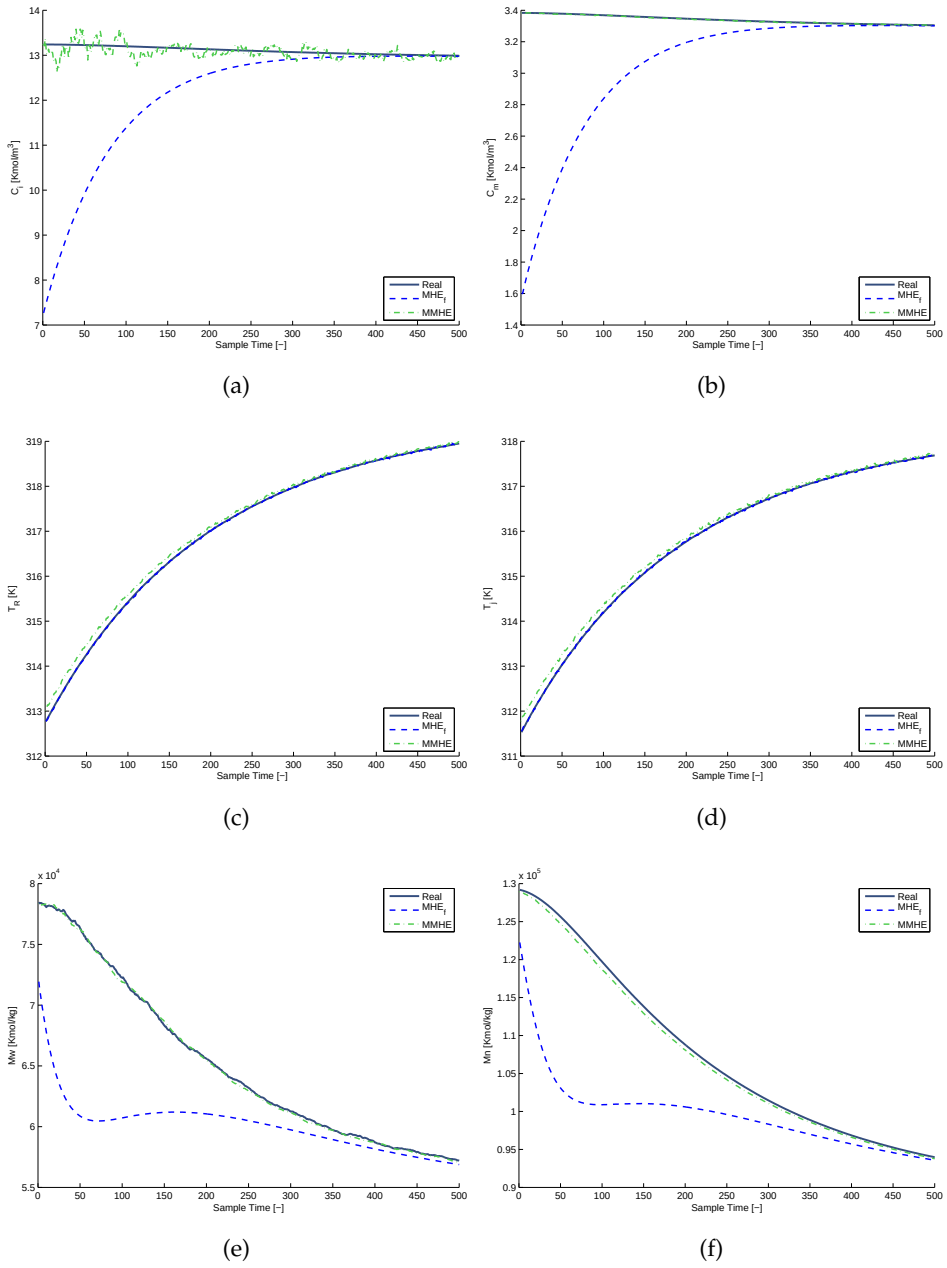



Figure 7.2: Polymerization CSTR simulation example. Estimated states using only fast measurements and multi-rated signals are compared for: (a) initiator concentration, (b) monomer concentration, (c) reactor temperature, (d) cooling jacket temperature, (e) weight average, and (f) number average molecular weights.

| State/Measurement | $R_l^F$ | $R_l^{SF}$ | $Q_l$ | $\bar{\Pi}_0$ |
|---|---|---|---|---|
| $C_i$ | - | - | $10^{-4}$ | $10^{-1}$ |
| $C_s$ | - | - | $10^{-4}$ | $10^{-1}$ |
| $C_m$ | - | - | $10^{-4}$ | $10^{-1}$ |
| $T_R$ | $2.5\times10^{-2}$ | $2.5\times10^{-2}$ | $10^{-3}$ | $10^{-1}$ |
| $T_J$ | $2.5\times10^{-2}$ | $2.5\times10^{-2}$ | $10^{-3}$ | $10^{-1}$ |
| $\lambda_0$ | - | $10^{-3}$ | $10^{-3}$ | $10^{-1}$ |
| $\lambda_1$ | - | $10^{-2}$ | $10^{-2}$ | $10^{-1}$ |
| $\lambda_2$ | - | $10^{-1}$ | $10^{-1}$ | $10^{-1}$ |

Table 7.1: State and measurement covariance values for the polymerization CSTR example.

multi-rated measurements and with the real trajectory of the states. Note that when only fast measurements are being used there are large errors at the first sampling times.

The benefit of using multi-rated measurements can be seen especially in the covariance matrices approximated with the reduced Hessian. Using only fast measurements, the arrival cost covariance at the first sample time has a condition number of $3.1464 \times 10^8$, while in the multi-rated case the condition number is $1.1010 \times 10^4$, a decrease of 4 orders of magnitude. This implies that the ill-conditioning of covariance matrix increases when we do not consider the slow measurements, and thus the system is close to being unobservable. On the other hand, adding the information of the measured molecular weight moments allows the estimator to quickly recover from a bad initial value of the arrival cost. Thus, if during normal operations a large disturbance occurs, MMHE would most likely recover faster than MHE$_f$. This agrees with the result in Tatiraju *et al.* [40], which considers the observability of the states with respect to *continuous* measurements. They show that, for example, the molecular weight moments are unobservable when only the fast measurements are used. While our examples are given by *discrete* time models and are still observable as assessed by IPOPT (see [87]), the ill-conditioning of the MHE$_f$ problem seen here

corresponds directly to their analysis.

### 7.2.2 Example 2: Large Scale Binary Distillation Column

In this section we present the application of the MHE to estimate states and parameters for a large scale binary distillation problem proposed by [70] using multi-rated signals. The column is used to separate Methanol (1) and n-Propanol (2), and it consists of 40 trays, a total condenser, and a reboiler. Also, for this example we show how using measurements that are obtained at a different rate without delay, improves the performance of MHE to estimate parameters (tray efficiencies) and the states. In particular we are interested in estimating the tray temperatures, the compositions of the liquid and vapor phases, as well as the molar holdup of the liquid phase for each of the trays. To achieve this we consider that the fast measurements are given by the temperatures and liquid volume holdup for each tray, while the slow measurements are given by adding the compositions of the light component in the liquid and vapor phases.

The column is modeled with an index 1 DAE system consisting of the so-called MESH equations (Mass balances, Equilibrium, Summation of compositions, and energy balances). In this case the vapor molar holdup is assumed negligible. This assumption is well known to yield high index DAEs, and thus index reduction must be performed. More details on this can be found in López-Negrete and Flores-Tlacuahuac [90] and Cervantes and Biegler [91]. Moreover, the equilibrium is modeled using Raoult's law, and non-ideal behavior is added through tray efficiencies. Finally, the liquid flow rate from the trays is modeled using Francis' weir equation. In total, the model consists of 84 differential equations and 168 algebraic equations, and it is described in Appendix B.4. Moreover, to simultaneously estimate states and parameters, the latter are formulated as random walks. Thus the state vector is augmented as $\zeta_k^T = \left[z_k^T,\, p_k^T\right]$ where $p_k^T = [\alpha_1, \ldots, \alpha_{N_T}]^T$ is the vector of tray efficiencies used in Equations (B.17) and (B.18). Thus the state model equation is changed as follows

$$\zeta_{k+1} = \begin{bmatrix} z_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} f(\zeta_k, w_k) \\ w_k^p \end{bmatrix} \tag{7.6a}$$

$$y_k = h(\zeta_k, v_k), \tag{7.6b}$$

where $w_k^p$ is a white noise variable associated with the parameter. For simulation purposes it is assumed that the fast measurement sample rate is 60 s and the slow sample rate is 120 s. In Table 7.2 we show the covariance values used for this example. Here the variables shown are molar liquid holdup, tray efficiency, light component liquid composition, tray temperature, liquid volume, and light component vapor composition.

| State/Measurement | $R_l^F$ | $R_l^{SF}$ | $Q_l$ | $\bar{\Pi}_0$ |
|---|---|---|---|---|
| $M_R$ | - | - | 10 | 10 |
| $M_i$ | - | - | 1 | 10 |
| $M_C$ | - | - | 5 | 10 |
| $\alpha_i$ | - | - | $10^{-3}$ | 10 |
| $x_i$ | - | $10^{-6}$ | $10^{-3}$ | 10 |
| $T_i$ | $2.5\times10^{-3}$ | $2.5\times10^{-3}$ | - | - |
| $V_i^m$ | $10^{-4}$ | $10^{-4}$ | - | - |
| $y_i$ | - | $10^{-6}$ | - | - |

Table 7.2: State and measurement covariance values for the distillation example.

The NLP in this case, using the discretized model, consists of 23,482 variables and 22,078 equality constraints. This optimization problem is solved on average in 41.57 CPU s, while the covariance approximation takes around 2.62 CPU s. Figure 7.3 shows the comparison between some of the estimated states and parameters using both MHE$_f$ and MMHE, with respect to the true values. In particular, Figure 7.3(a,b) display the compositions of the light component in the liquid and vapor phases of trays 14 and 28. Note that the most visible

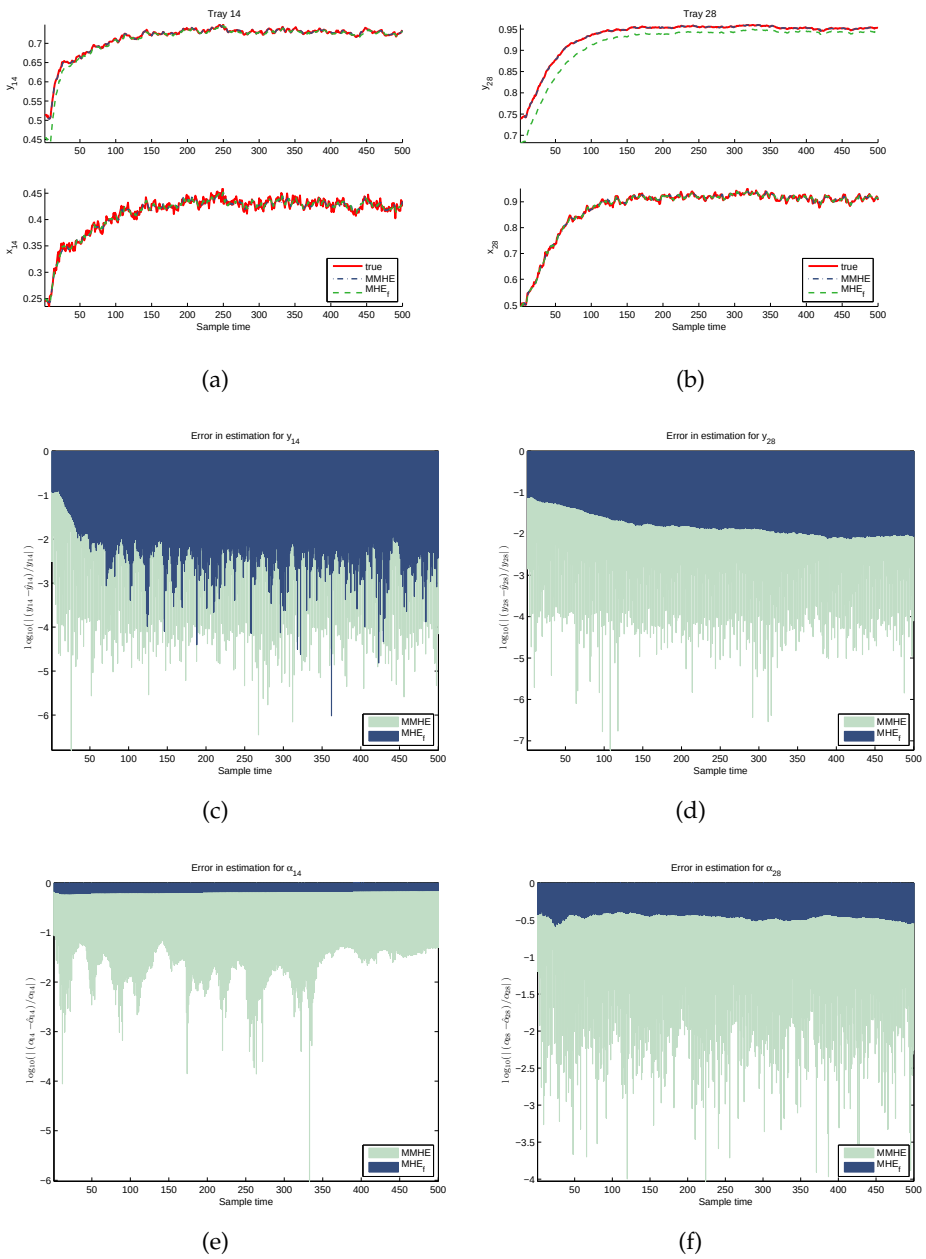



Figure 7.3: Large scale binary distillation example: vapor and liquid compositions for tray 14 (a) and tray 28 (b), $log_{10}$ of estimation error of vapor composition for trays 14 (c) and 28 (d), $log_{10}$ of estimation error of tray efficiency for trays 14 (e) 28 (f).

difference can be seen in the vapor composition of tray 28. Therefore, to better show the effects of using a combination of fast and slow measurements, we evaluate the $log$ of the normalized error. This is given by

$$\log(\epsilon_k) = \log\left(\left|\frac{\zeta_k - \hat{\zeta}_k}{\zeta_k}\right|\right), \tag{7.7}$$

where $\hat{\zeta}_k$ represents the estimated quantity and $\zeta_k$ its true value.

In Figures 7.3(c,d) we use bar plots to show the effects of using the two types of estimators on the estimation error in the composition of the light component in the vapor phase. It is clear that there is a benefit of using the MMHE, since the estimation error for both compositions is orders of magnitude smaller than when using MHE$_f$. On the other hand, in Figures 7.3(e,f) we see the estimation error for the calculated tray efficiencies. Again, in this case, the estimation error of the parameter is greatly reduced when using multi-rated measurements.

## 7.3 Conclusions

Moving Horizon Estimation provides a framework for constrained estimation that systematically handles bounds and constraints. Since this strategy processes a batch of measurements in the past it is straightforward to combine fast, slow and even delayed measurements. Using multi-rated measurements reduces the estimation error since more information about the system is used, and, since the structure of measurement model changes, it is possible to make some unobservable states observable.

The reduced Hessian approach for updating the covariance matrix takes into account the changes in the measurement model and vector sizes automatically. Thus, it is a general approach of the variable structure multi-rated MHE that generates the covariance matrix using smoothed information from the horizon. Moreover, the addition of the slower measurements makes it possible to reduce estimation errors, and this is illustrated in the

simulation examples.

# Chapter 8

# Advanced Step Moving Horizon Estimation

In previous chapters we have assumed that the MHE takes a negligible amount of time to solve, even when this is not true for some of the large scale systems used here. However, the performance of the controllers that use these estimates could be affected when we take into account these delays. Closed-loop stability could even be lost if the delays are long enough. This is similar to the delay in solving the NMPC problem [29]. Moreover, the delayed estimate will no longer correspond to the true state of the system, especially if a non-negligible amount of time has passed from the moment the plant is sampled to the moment we obtain the estimate from MHE. Thus, it is important to develop fast strategies that can overcome these limitations. To reduce the computational delay in model predictive controllers an advanced step NMPC has been developed [29]. This approach takes advantage of NLP sensitivity to approximate the solution of the NMPC problem on-line. This method shifts the expense of solving the NLP to the background, and the only on-line computational expense is solving the KKT system given by (2.12) (or (2.13)). A similar approach exists for MHE, and it will be discussed here. Moreover, this approach is also extended to handle multi-rated measurements in the advanced step Multi-Rated MHE (asMMHE).

The rest of this chapter is organized as follows: Section 8.1 introduces the asMHE strategy, and here we describe the extension to handle multi-rated measurements, in Section 8.2 we illustrate the benefits of the advanced step strategies, and finally we conclude the chapter in Section 8.3.

## 8.1 asMHE and asMMHE Strategies

The advanced step strategy takes advantage of the parametric properties of the moving horizon NLP to reduce the on-line computational expense. If we know the the previous state estimate we can make a model prediction of the current measurement. Using this prediction we can solve an approximate NLP, which we update as the new measurement becomes available using NLP sensitivity. This is illustrated in Figure 8.1. Note that this

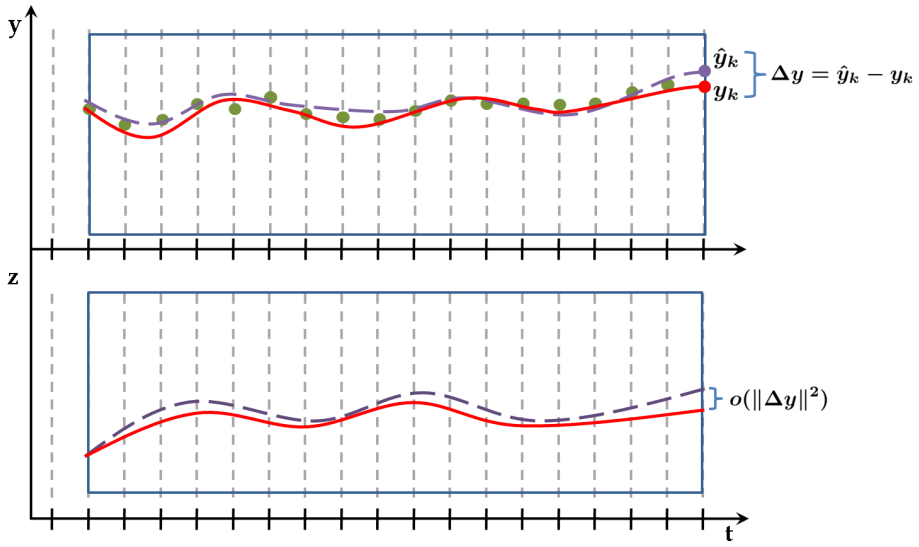



Figure 8.1: Illustration of the asMHE strategy, where the purple dashed line represents the solution of the approximate problem, and the solid red line represents the updated solution using NLP sensitivity.

method has two components: the off-line solution of the NLP and the on-line update of the solution using Equation (2.14). The details of on-line and off-line components are summarized as follows.

**Off-line component (at time $k-1$):**

1. Generate a model prediction of the measurement $\hat{y}_k$ using the state estimate $\hat{z}_{k-1|k-1}$,

and Equations (3.1).

2. Update the measurement sequence to yield $\hat{Y}_{k-N}^{k} = \{y_{k-N}, \dots, \hat{y}_k\}$.
3. Solve the approximate NLP in between sample times $k-1$ and $k$, and hold the factorized KKT matrix at the solution.
4. Generate the smoothed covariance information through the reduced Hessian.

**On-line component (at time $k$):**

1. Update the NLP parameters (measurements) with the true measurement $y_k$.
2. Compute the updated solution using Equations (2.13) and (2.14) and the KKT matrix from the solution of the approximate NLP.

In this strategy, the parameter perturbation is given by $\left|Y_{k-N}^{k} - \hat{Y}_{k-N}^{k}\right| = |y_k - \hat{y}_k|$. Thus, the difference between the state estimate computed with ideal MHE, where the computational delay is assumed negligible, and the advanced step is given by

$$\left|\hat{z}_k^{MHE_i} - \hat{z}_k^{asMHE}\right| = O\left(|y_k - \hat{y}_k|^2\right). \tag{8.1}$$

In other words, as long as the error between the true and the predicted measurements is small, then the error in the state estimate obtained with the asMHE strategy will be very close to the ideal case.

The advanced step strategy can also be implemented with multi-rated measurements. The extension is simple, as long as care is taken when generating the simulated measurement $\hat{y}_k$. For example if at sample time $k$ the true measurement will consist of both fast and slow observations, then $\hat{y}_k = \hat{y}_k^{SF}$. On the other hand, if the true measurement will only consist of the fast observation, then we have that $\hat{y}_k = \hat{y}_k^{F}$. In the following section we show two examples of the implementation of the asMHE strategy: one using only fast measurements and the second case considers multi-rated measurements.

## 8.2 Simulation Examples

In this section we show two simulation examples to illustrate the behavior of the asMHE; both examples are implemented using the model for the binary distillation column described in Appendix B.1. However, in the first case study we consider that only fast measurement information is available, while for the second example we implement the advanced step multi-rated MHE (asMMHE).

The column is used to separate Methanol (1) and n-Propanol (2), and it consists of 40 trays, a total condenser, and a reboiler. The model of the column is given by an index 1 DAE system that consists of the MESH equations. In total there are 84 differential equations and 168 algebraic equations. Moreover, note that the measurement equations (for both case studies) are nonlinear functions of the states, represented by some of the algebraic equations of the model.

### 8.2.1 Example 1: asMHE with a Binary Distillation Column

For this case study we assume that the measurements are given by the temperatures and volumetric holdups of each tray. These are used to estimate the composition of the light component in the liquid phase and the liquid molar holdup at each tray. Additionally, we consider that the sample time is 60 seconds, a horizon length of 10 measurements, and the parameters of the noise variables are summarized in Table 8.1.

Since we have considered a horizon length of 10 measurements and we are using Radau collocation to transform the DAE system into a discrete time model, then the NLP that is solved at each sample time has 21642 variables and 21642 equality constraints. The approximate NLP is solved on average in of 42.38 CPU sec, and the covariance for the arrival cost is computed using the reduced Hessian approach in an average of 1.84 CPU seconds. Note that these computations are done in the background, and in between sample times. Thus, the only on-line computational expense is given by the NLP sensitivity update which takes on average 0.529 CPU seconds. This is orders of magnitude faster than solving

| | Model noise variance | Measurement variance |
|---|---|---|
| $T_i$ | – | $6.25 \times 10^{-2}$ |
| $V^m_{N_T+1}$ | – | $10^{-8}$ |
| $x_i$ | $10^{-5}$ | – |
| $M_0$ | 10 | – |
| $M_i$ | 1 | – |
| $M_{N_T+1}$ | 5 | – |

Table 8.1: Summary of parameters for the noise variables for the distillation column example in Chapter 8.

the NLP on-line; thus this approach drastically reduces the delay in generating the state estimate once the plant measurement is available.

In Figure 8.2 we compare the behavior of the estimated states using the ideal MHE (MHE$_i$) with those obtained with the asMHE. In the same figure we also compare the errors in estimation using both methods. Both methods track the true trajectory of the states properly. In fact, looking at the estimation errors, computed with $\epsilon_l = |z_l - \hat{z}_l| / z_l$, we can see that the largest error is 8%, for the composition in tray 14. However, when using the ideal MHE, the mean error for the compositions in tray 14 is 2% and in tray 28 is 1.6%, while for the liquid holdups the mean errors are 0.5% and 1.4%. Moreover, the errors when using the advanced step MHE are very close to those of the ideal case. Therefore, the performance of the asMHE is almost identical to the performance of the ideal scenario, but the computational delay that could be introduced by solving the full NLP on-line is avoided.

### 8.2.2 Example 2: asMMHE with a Binary Distillation Column

For this example we implemented the asMMHE for state and parameter estimation using the distillation column used above. Here we combine fast and slow measurements to im-

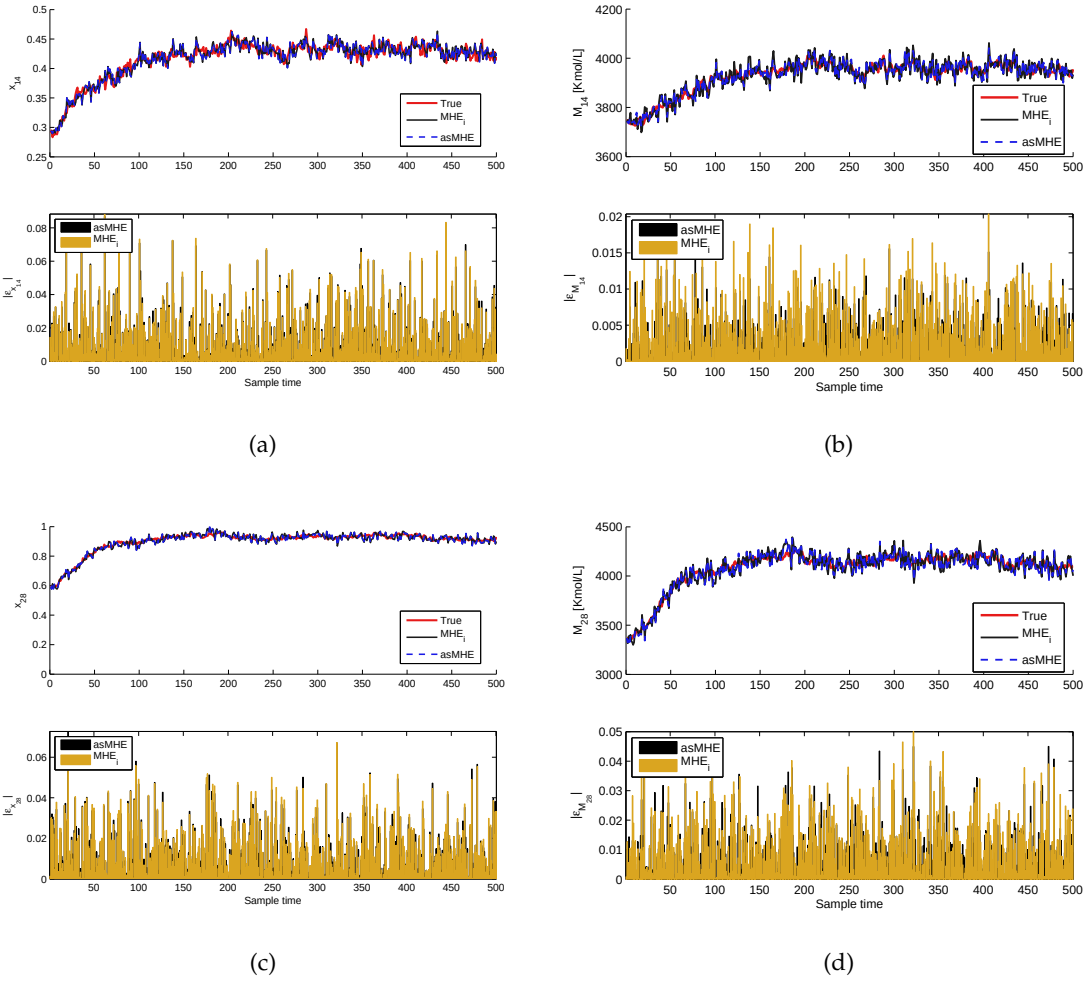



Figure 8.2: Comparison of the estimated states using MHE and asMHE for the distillation column.

prove the performance of MHE to estimate parameters (tray efficiencies) and the states. In particular we are interested in estimating the tray temperatures, the compositions of the liquid and vapor phases, as well as the molar holdup of the liquid phase for each of the trays. To achieve this we consider that the fast measurements are given by the temperatures and liquid volume holdup for each tray, while the slow measurements are given by adding the compositions of the light component in the liquid and vapor phases. Similarly as before, we consider sample times of 60 seconds, a horizon length of 10 measurements, and the parameters of the noise variables are summarized in Table 8.2.

| State/Measurement | $R_l^F$ | $R_l^{SF}$ | $Q_l$ | $\bar{\Pi}_0$ |
|---|---|---|---|---|
| $M_R$ | - | - | 10 | 10 |
| $M_i$ | - | - | 1 | 10 |
| $M_C$ | - | - | 5 | 10 |
| $\alpha_i$ | - | - | $10^{-3}$ | 10 |
| $x_i$ | - | $10^{-6}$ | $10^{-3}$ | 10 |
| $T_i$ | $2.5\times10^{-3}$ | $2.5\times10^{-3}$ | - | - |
| $V_i^m$ | $10^{-4}$ | $10^{-4}$ | - | - |
| $y_i$ | - | $10^{-6}$ | - | - |

Table 8.2: State and measurement covariance values for asMMHE with the distillation column in Chapter 8.

In this case, the solution of the ideal MMHE (i.e., MMHE$_i$) takes an average of 38.26 CPU seconds, while the on-line component of the asMMHE takes 0.78 CPU seconds. Again we note a drastic reduction of the on-line computational expense, which implies that we are reducing the computational delay of estimating the states. This is important, as mentioned above, when we use these estimates in combination with a controller because delays can potentially destabilize the system.

In Figure 8.3 we compare the behavior and estimation errors of the ideal case (MMHE$_i$)

with that of the asMMHE. As in the previous example we see that the estimator is able to track the true states of the system as well as the parameters. Furthermore, the estimation errors are very small which means that the estimates are not far from the true values. Moreover, the estimation error of the asMMHE is very close to the ideal case. Therefore, the behavior of both estimators is very similar, but the asMMHE introduces a negligible delay.

## 8.3 Conclusions

The advanced step strategy described in this chapter is capable of drastically reducing the computational delay associated with the solution of the MHE problem. It is also possible to extend the asMHE to use multi-rated measurements to take advantage of fast and slow measurement information to estimate states and parameters. For the cases shown in this chapter, the behavior of the ideal MHE and the asMHE are basically the same. This means that the computational delay associated with asMHE is in fact negligible, and that the quality of the estimates is not affected by the advances step calculations.

The implementation of these strategies if further simplified with the use of *sIPOPT* and AMPL. The former extends IPOPT to compute fast sensitivity updates of solutions for the on-line component, and computes the reduced Hessian used to approximate the covariance of the arrival cost. Moreover, AMPL provides exact first and second derivatives, which improves the computations obtained with *sIPOPT*.

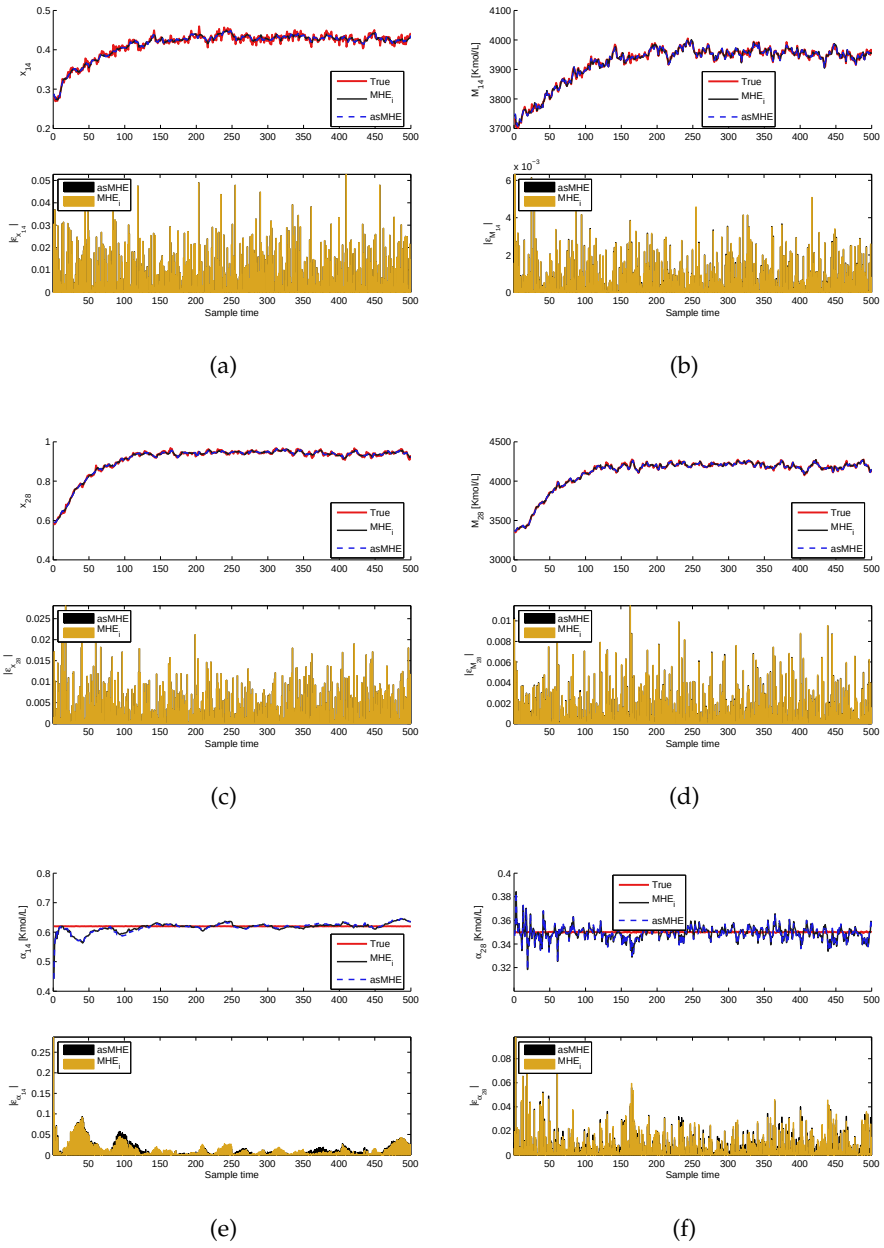



Figure 8.3: Comparison of the estimated states and parameters using MHE and asMMHE for the distillation column.

# Chapter 9

# Robust M-Estimators for MHE

In previous chapters we have assumed that measurement information has Gaussian noise added to it. However, outliers and gross errors occur naturally during process operations. For example, measurement errors occur when operators make errors reading sensor information, or during laboratory assays. Also, sensors may get stuck or fail, which introduce gross errors in observations. Thus, it is important to take this into account to develop strategies that can deal with outliers and gross errors systematically to obtain state estimates. One possible way to do this is through the use of robust statistics, that provide estimators less sensitive to large residual values. For example, here we introduce the use of two types for M-Estimators: the Fair Function and Hampel's re-descending estimator. These functions behave as the least squares estimator for small values of the residuals, but when residual values increase the function values increase much slower than that of the least squares estimator. In fact, Hampel's re-descending estimator has three regions that behave differently; for small enough residuals it behaves as the least squares, then it grows linearly, and finally it inflects and becomes constant for large enough residuals. Thus, combining these types of estimators with MHE we propose a formulation that is able to compute state estimates that are not biased by errors in measurement information. Finally, in the next section we describe robust statistic which will allow us to introduce the M-Estimators.

The rest of this chapter is organized as follows. In Section 9.1 we describe the concept of a robust statistics. This concept is then used in Section 9.2 to describe the M-Estimators. Here we describe two different types of functions: the Fair Function and Hampel's re-descending estimator. In Section 9.3 we propose MHE strategies based on

the M-Estimators that are able to deal with gross errors and outliers to provide estimates with low estimation errors. In Section 9.4 we illustrate the performance of these strategies with two simulation examples, and finally this chapter is concluded in Section 9.5.

## 9.1 Robust Statistics

The performance of MHE is reduced in the presence of outliers and gross errors, since the information on the true state of the system is incorrect. In order to mitigate the effects outliers and gross errors have on the state estimates we can use robust statistics.

The errors in the measurements are assumed to follow Gaussian distributions, in which case the associated likelihood function is the least squares estimator. However, when measurement data is corrupted with outliers or gross errors it is difficult to determine its distribution, and using a fixed distribution is not consistent. In such cases, robust estimators may be used. These types of estimators are mostly distribution independent and produce unbiased results in the presence of data derived from some ideal distribution, and are less sensitive to deviations from ideality [92, 93]. The main advantage of these estimators is that they give less weight to measurements that have been corrupted. Suppose that $\{\xi_1, \ldots, \xi_n\}$ are drawn from a distribution $f(\xi)$, and let $T$ be an unbiased estimator $\hat{\theta} = T[f(\xi)]$ of parameter $\theta$. If $g(\xi)$ represents an approximate distribution model, then the estimate will be given by $\tilde{\theta} = T[g(\xi)]$. The distributions of the estimators based on $f(\xi)$ and $g(\xi)$ will be $\Upsilon\left(\hat{\theta}, f\right)$ and $\Upsilon\left(\tilde{\theta}, g\right)$, respectively. Thus, the estimator $T(\cdot)$ is robust iff

$$d(f,g) < \eta \implies d\left[\Upsilon\left(\hat{\theta}, f\right), \Upsilon\left(\tilde{\theta}, g\right)\right] < \epsilon, \tag{9.1}$$

where $d(\cdot)$ is a distance function. Therefore, a bounded shift from the ideal distribution will lead to a bounded shift in the estimates. The influence function can be used to assess the robustness of the estimator. This function measures the importance of an observation on the estimator and it is defined by

$$\mathcal{IF} = \psi(\xi_0) = \lim_{t \to 0} \frac{\Upsilon\left[(1-t)\, f + t\delta\left(\epsilon - \epsilon_0\right)\right] - \Upsilon\left[f\right]}{t}, \tag{9.2}$$

where $\delta(\epsilon - \epsilon_0)$ is the Dirac delta function centered on the particular observation $\epsilon_0$. Moreover, for the estimator to be robust, its influence function has to be bounded as the observations go to infinity. In the following section we describe two robust M-Estimators that can be used in combination with MHE to derive a state estimator that is robust to outliers or gross errors.

## 9.2 Robust M-Estimators

M-Estimators can be used as the likelihood functions for the measurements in the objective function of (4.13). If we define the likelihood function $\mathbb{L}(x_j|p)$ of the observation $x_j$ dependent on parameters $p$, then the overall likelihood function of the errors in $N$ observations is given by

$$\mathbb{L} = \prod_{j=1}^{N} \mathbb{L}(x_j|p)\,. \tag{9.3}$$

The M-Estimator associated with the previous likelihood function is given by

$$\rho^M = \sum_{j=1}^{N} \rho_j = -\log(\mathbb{L}) = -\sum_{j=1}^{N} \log\left[\mathbb{L}(x_j|p)\right], \tag{9.4}$$

where $\rho^M$ is the overall M-Estimator, and $\rho_j$ is the estimator associated with the *j*-th observation. There are two particular estimators that are of interest for the present work. These are Huber's fair function [94] and Hampel's re-descending estimator [95], which are defined with the following expressions:

**Huber's Fair Function:**

$$\rho_j^F = C^2 \left[\frac{|\epsilon_j|}{C} - \log\left(1 + \frac{|\epsilon_j|}{C}\right)\right], \tag{9.5}$$

where $C$ represents a tuning parameter, and $\epsilon_j = (y_j - \hat{y}_j)/\sigma_y$ is the studentized prediction error. Note that here $\sigma_y$ represents the standard deviation of the residual.

**Hampel's Re-descending Estimator:**

$$\rho_j^R = \begin{cases} \dfrac{1}{2}\epsilon_j^2, & 0 \leq |\epsilon_j| \leq a \\ a\,|\epsilon_j| - \dfrac{a^2}{2}, & a < |\epsilon_j| \leq b \\ ab - \dfrac{a^2}{2} + \dfrac{a\,(c-b)}{2}\left[1 - \left(\dfrac{c - |\epsilon_j|}{c-b}\right)^2\right], & b < |\epsilon_j| \leq c \\ ab - \dfrac{a^2}{2} + \dfrac{a\,(c-b)}{2}, & |\epsilon_j| > c, \end{cases} \tag{9.6}$$

$a$, $b$, and $c$ represent tuning parameters that satisfy $c \geq b + 2a$ and allow us to define 4 regions in the estimator. These regions represent different types of behavior depending on the magnitude of the residuals. In the first region, when residuals are small, the estimator has the idealized behavior; in this case it behaves as the least squares estimator. The value of the estimator becomes linear with respect to the residuals in the second region, e.g., if $a < |\epsilon_j| \leq b$. As the residuals increase in value, the function value enters the re-descending region. Finally, when residuals are greater than $c$, the function becomes constant. This is illustrated in Figure 9.1, where we compare the least squares function with the Fair Function and Hampel's re-descending estimator. Note that the value of the Fair Function increases more slowly than that of the least squares function. Thus, if outliers or gross errors are present in the observations the values of the M-Estimators are lower than that of the least squares function. In other words, outliers and gross errors have less influence on the estimators for these functions.

## 9.3 M-Estimators in MHE

In Chapters 4-6 we developed MHE considering that the measurement noise is Gaussian. However, as mentioned above, in the presence of outliers or gross errors it is hard to de-

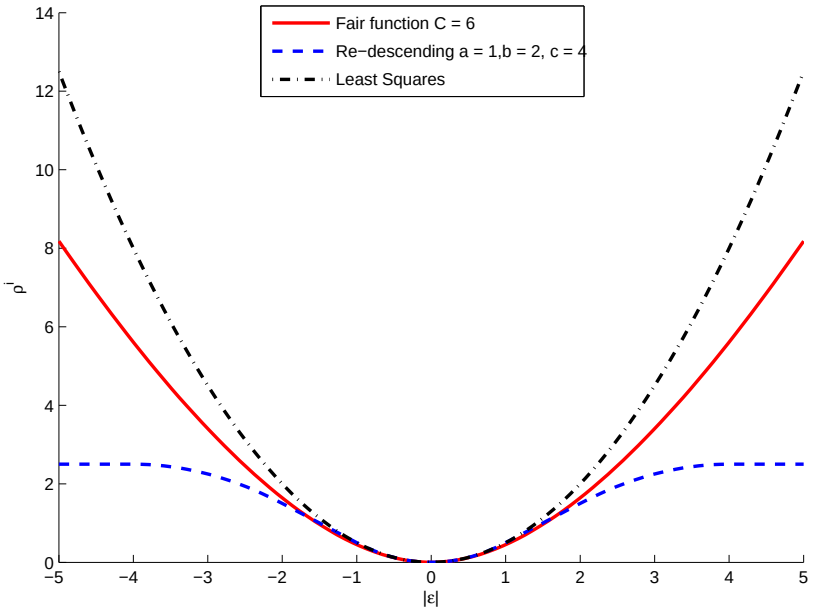



Figure 9.1: Comparison of the values of the Least Square Function, the Fair Function, and the re-descending estimator.

termine the correct distribution of the observations. Thus, it is possible to assume that the ideal behavior of the residuals will be Gaussian, and in order to reduce the effects of errors in the observations we can use M-Estimators. In other words, we can derive a robust MHE using a smoothed update of the arrival cost in the following way. For this case, the PDF of the states in the horizon is given by

$$p\left(Z_{k-N}^{k}\middle|\, Y_{0}^{k}\right) = \mathcal{K}\,\frac{p\left(z_{k-N}|\, Y_{0}^{k-1}\right)}{p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)}\left[\prod_{l=k-N}^{k} p\left(y_l|\, z_l\right)\right]\left[\prod_{l=k-N}^{k-1} p\left(z_{l+1}|\, z_l\right)\right], \tag{9.7}$$

where $\mathbb{L} = \prod_{l=k-N}^{k} p\left(y_l|\, z_l\right)$ represents the likelihood function. Previously we have considered Gaussian noise, and thus the estimator associated with this likelihood function is the least squares function. However, here we consider that the estimator can be substituted with an M-Estimator such as the Fair Function or the re-descending Estimator. Thus, after taking logarithms and maximizing the probability we obtain

$$\min_{z_{k-N}} \quad \frac{1}{2}\left\|z_{k-N}-\hat{z}_{k-N|k-1}\right\|^2_{\Pi^{-1}_{k-N|k-1}} - \frac{1}{2}\left\|\mathcal{Y}-\mathcal{O}z_{k-N}\right\|^2_{\mathcal{W}^{-1}} + \frac{1}{2}\sum_{l=k-N}^{k}\rho_l^{ME}\left(v_l\right) + \frac{1}{2}\sum_{l=k-N}^{k-1} w_l^T Q_l^{-1} w_l \tag{9.8a}$$

$$\text{s.t.} \quad z_{l+1} - f\left(z_l\right) - w_l = 0 \tag{9.8b}$$

$$y_l - h\left(z_l\right) - v_l = 0 \tag{9.8c}$$

$$z^{LB} \leq z_l \leq z^{UB}, \tag{9.8d}$$

where $\rho_l^{ME}\left(v_l\right)$ represents the robust M-Estimator of our choosing; either Eq. (9.5) or (9.6) depending on whether we wish to use the Fair Function or the re-descending estimator. Note that if we choose the latter, the objective function in Eq. (9.8) will be non-smooth. Thus, it is important to use a smoothed approximation of the M-Estimator as shown in Arora and Biegler [93]. A consequence of using Hampel's re-descending estimator is that, if the observations have large errors, then the estimator function will become constant. This implies that some terms in Equation (9.8a) will be constant, and thus those measurements are not considered. On the other hand, if residuals are not too large and we are using either of the M-Estimators discussed above, then the terms associated with these observations will have less influence on the objective function. Therefore, the state estimates will not be affected too much by outliers or gross errors. In the following section we illustrate the behavior of these estimators on two simulation examples.

## 9.4 Simulation Example

In this section we illustrate the use of robust M-Estimators in combination with MHE to show how the state estimator handles gross errors. These errors are simulated by assuming that one of the thermocouples drifts away from the true value of the measurement. In the first example we compare the use of the Fair Function (FF-MHE) and the re-descending Estimator ($\text{MHE}_{\text{RED}}$), with the traditional least squares MHE. For this first case study we use

| | Measurement Noise Variance | State Noise Variance |
|---|---|---|
| $C_A$ | – | $10^{-8}$ |
| $T_R$ | 0.0625 | $10^{-4}$ |
| $T_{cw}$ | 0.0625 | $10^{-4}$ |

Table 9.1: Parameters of noise variables for the CSTR example using M-Estimators with MHE.

the 3 state CSTR model. Finally, the second example we use the binary distillation model, and compare the use of the re-descending estimator with the least squares estimator when the temperature measured at tray 14 drifts away from the true measurement value.

### 9.4.1 Example 1: CSTR Case Study

The model used in this example is described in Appendix B.2, and it describes a CSTR where component A is transformed into B. The model consists of 3 states, the concentration of A ($C_A$), the temperature of the reactor $T_R$, and the temperature of the cooling jacket $T_{cw}$. The measurements here are the two temperatures. To show the effects of having gross errors, here we assume that the observed temperature of the reactor drifts away from the true measurement. The maximum error is 5 degrees above the true measurement, and we increase the gross error by 0.0625 degrees per sample time. Moreover, the parameters of the noise variables are summarized in Table 9.1.

The simulation results are shown in Figure 9.2. Note that at sample time 80 the error in the measured temperature of the reactor is slowly added. Figures 9.2(a,c,e) show the estimated states compared with the different methods, while Figures 9.2(b,d,f) show the estimation error for each state. In these figures it is clear that the MHE with the least squares estimator has the worst performance. On the other hand, when the Fair Function is used the effects of the measurement error has less effect on the objective function of Problem 9.8. Therefore, the estimation errors are smaller than in the least square case. Finally, the

re-descending estimator MHE ($MHE_{RED}$) has the best performance. As the residuals start to increase we can see that errors begin to grow. However, when the errors are large enough, then Hampel's estimator becomes constant and the measurements are ignored. Thus, the effects of the errors are eliminated, and only the measured cooling water temperature is used to estimate the states. Finally, the effects of the different regions in Hampel's re-descending estimator can be seen, for example, in Figure 9.2b. As the temperature drifts away from the true measurement in sample times 80-150 the estimation errors increase. Once the residuals are past the threshold, and the estimator becomes constant, the estimation error drops drastically. Finally, when the gross error is eliminated after sample time 250 we can see that the estimation error starts to increase and then decrease as the residuals move through the different regions.

### 9.4.2 Example 2: Distillation Column Case Study

For this case study we assume that the measurements are given by the temperatures and volumetric holdups of each tray. These are used to estimate the composition of the light component in the liquid phase and the liquid molar holdup at each tray. Additionally, we consider that the sample time is 60 seconds, a horizon length of 10 measurements, and the parameters of the noise variables are summarized in Table 9.2. Moreover, here we simulate a drift in the measured temperature of tray 14 which starts at sample time 30. The error is increased to a maximum of 40 degrees at a rate of 0.888 degrees per sample time. After sample time 90, this error is reduced at the same rate until it reaches a value of zero. Finally, here we compare the performance of the least squares MHE with the re-descending MHE ($MHE_{RED}$).

Figure 9.3 shows the results of the simulation example. For tray 14, when the temperature observation drifts away from the true measurement we can see that both the composition and holdup estimation errors increase when using the traditional least squares MHE. On the other hand, when we use $MHE_{RED}$ the effects of the gross error are reduced. In fact,

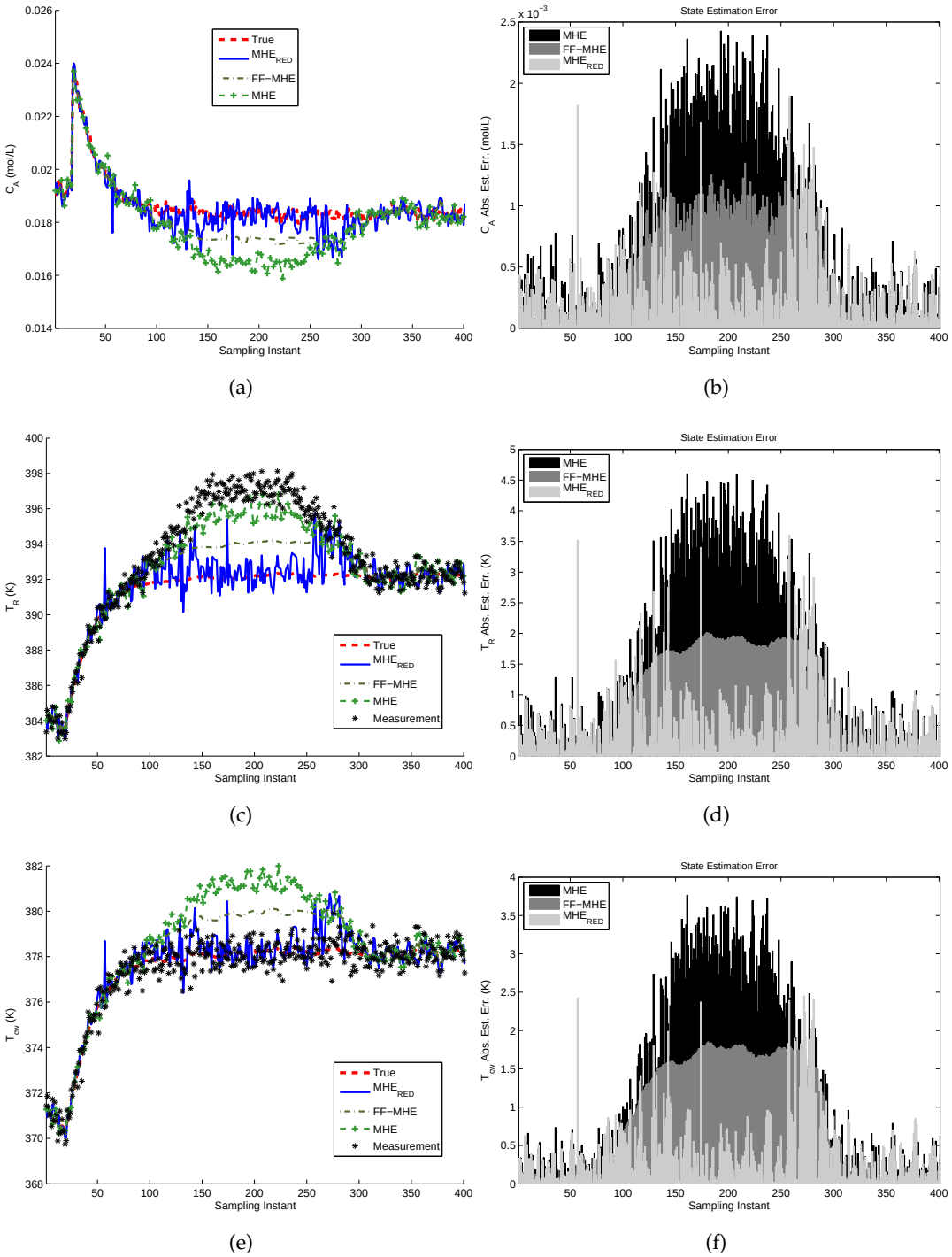



Figure 9.2: Comparison of the Fair Function, re-descending estimator, and least squares estimator MHE using the CSTR example.

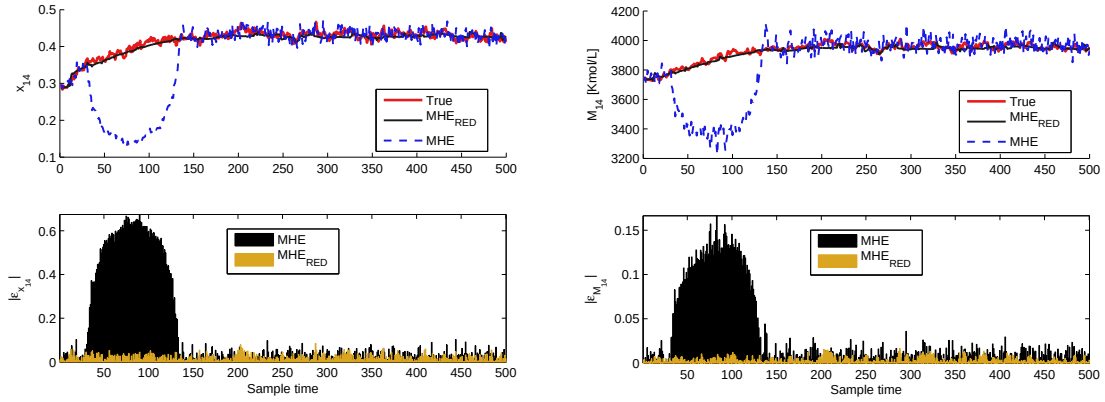
| | Model noise variance | Measurement variance |
|---|---|---|
| $T_i$ | – | $6.25 \times 10^{-2}$ |
| $V^m_{N_T+1}$ | – | $10^{-8}$ |
| $x_i$ | $10^{-5}$ | – |
| $M_0$ | 10 | – |
| $M_i$ | 1 | – |
| $M_{N_T+1}$ | 5 | – |

Table 9.2: Summary of parameters for the noise variables for the distillation column example in Chapter 9.

in Figures 9.3a and 9.3b we see that the estimation errors practically do not change, even in the presence of the temperature drift. Furthermore, for the trays above tray 14, we see that the temperature drift has a negligible effect on the liquid composition. However, for the liquid holdup estimates (Fig. 9.3d) we see a spike in the estimation errors for MHE starting at sample time 90, which coincides with the addition of the gross error. For this same figure, we can see that, when using the $\text{MHE}_{\text{RED}}$, this does not happen. This is because the M-Estimator reduces the effects of the large residuals, and the estimates are obtained using the other measured variables.

## 9.5 Conclusions

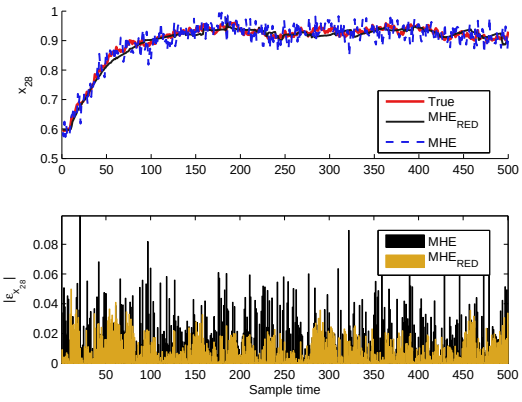
In this chapter we introduced robust estimators that change their behavior depending on the magnitude of the residuals. These estimators result in smaller values than the least squares estimator when evaluated with large residuals, and thus it is possible to combine them with MHE to reduce the effects of outliers and gross errors. Furthermore, here we compared the use of the Fair Function and Hampel's re-descending estimator with a small CSTR example, and from the simulation results we can conclude that the latter



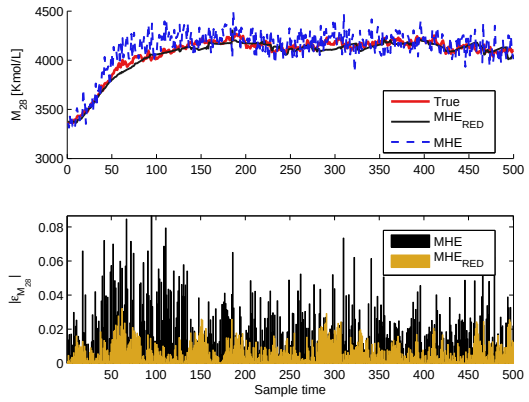

(a)




(b)




(c)




(d)

Figure 9.3: Comparison of the Fair Function, re-descending estimator, and least squares estimator MHE using the distillation column example.

has better performance since very large measurement errors imply that those observations are dropped. Moreover, for the second example we see similar behavior, where the M-Estimator greatly reduces the effects of the measurement errors, and performance increases when estimation errors are compared with those of the least squares MHE. Finally, note that it is possible to use M-Estimators in combination with any of the MHE formulations shown in previous chapters. Thus it is possible to construct a robust multi-rated MHE for state and parameter estimation by simply changing the likelihood function with any of Equations (9.5), (9.6), or a different robust M-Estimator (e.g., contaminated Gaussian distributions [92]).

# Chapter 10

# Conclusions

In the present work we have described techniques for constrained state estimation based on NLP sensitivity and robust M-Estimators. These approaches make use of NLP solvers to handle constraints on states and parameters to compute the estimates. Moreover, here we discuss implementations that allow us to use these methods in real time by reducing the computational expense associated with the solution of these problems. Additionally, we show that there exists a relationship between the covariance of the state estimates and the reduced Hessian, which can be used for further analysis on the estimates and to update the arrival cost term in the objective function. Finally, using M-Estimators we propose formulations that are able to reduce the effects of outliers and gross errors in the measurements. In this way, it is possible to efficiently compute state estimates in real time that are not biased by the errors. Finally, in this chapter we conclude the dissertation. In the following section we summarize the main contributions described in each chapter, and we briefly describe the main topics discussed in each one. Finally, in Section 10.2 we suggest possible research topics for future work related to state estimation and data reconciliation.

## 10.1 Summary of the Dissertation and Contributions

In Chapter 2 we discuss interior point solvers, and some properties associated with these methods that allow us to extract sensitivity information directly from the KKT conditions. In addition, we show how to extract reduced Hessian information, which is not normally generated by interior point solvers, and we introduce the software *sIPOPT* that integrates these methods to extend IPOPT's post optimal analysis capabilities.

Chapter 3 introduces the problem of Bayesian state estimation, and the most common methods used to solve this problem. Here we also discuss some of the typical drawbacks associated with these methods that justify the development of MHE.

In Chapter 4 Moving Horizon Estimation is introduced from the Bayesian perspective. We also discuss how the arrival cost term is used to summarize all the measurement information that is not included in the horizon. Moreover, we describe two possible ways to formulate MHE, where the arrival cost is either updated with filtered or with smoothed information.

Chapter 5 introduces an MHE formulation that uses a posterior distribution to describe the arrival cost term. In this chapter we discuss two possible ways to approximate the initial condition penalty term. The first one assumes that this term can be approximated with the least squares estimator resulting in the original formulation proposed in [5]. For this approach we also show how to generate approximations of the covariance that are consistent with bounds and the non-Gaussian properties of the initial condition, using constrained sample based filters. Furthermore, in this chapter we introduce a novel approach to directly approximate the arrival cost density function using ideas from particle filters. In this way we substitute the traditional 2-norm form for the approximated PDF. Finally, using simulation examples, we illustrate how the horizon window size can be reduced with these approaches, and still compute estimates with low estimation error. This is possible since a better approximation of the arrival cost will not introduce errors in the initial condition. Thus, we no longer need to use a lot of measurement information to reduce the weight of the initial condition decreasing the size of the horizon window.

Chapter 6 discusses a second MHE formulation that uses smoothed information to approximate the arrival cost term. In this chapter we introduce a novel approach that extracts covariance information directly from the KKT conditions of the NLP problem through sensitivity. Traditionally this strategy uses EKF and EKS to propagate the covariance matrix forwards and backwards in the horizon to approximate the smoothed covariance of the ini-

tial condition. However, here we show a property of the MHE optimization problem that relates the covariance of the state estimates with the reduced Hessian. Using this property we avoid the propagation of covariances, which results in a cheaper approximation of the covariance information. Moreover, this approach is superior to the one shown in the previous chapter, since the latter requires very large amounts of samples that can increase exponentially with the number of states.

In Chapter 7 we discuss the possibility of combining slow, possibly delayed observations with fast measurements to improve the quality of the state estimates. Multi-rated state estimation can also help make some unobservable states observable. Furthermore, MHE provides a framework where this slow measurements can be incorporated systematically for constrained state estimation. This is easily achieved since in MHE we have a horizon of past measurements, and thus the slower observations only need to be placed in their proper locations in the past history. Moreover, using the smoothed approach for arrival cost approximation, we can use the reduced Hessian to generate the arrival cost covariance matrix. Thus, MMHE results in a superior approach for multi-rated constrained state estimation.

Chapter 8 we describe a method that reduces on-line computational expense of solving the MHE problem, thus making MHE suitable for real-time applications. Here we use NLP sensitivity to generate fast approximations of the solution of the NLP, thus shifting the solution of the problem to the background. Moreover, we discuss the application of the advanced step strategies to multi-rated state estimation, and finally we illustrate these methods with two simulation examples.

Finally, in Chapter 9 we describe robust statistics and robust M-Estimators. These functions reduce the effects that large residuals have on the estimator, and thus decrease the influence that outliers or gross errors could have on the estimates. Here we also show a novel MHE formulation leverages robust estimators overcome the effects that measurement errors might have. For this strategy we also use reduced Hessian information to

update the arrival cost term. This strategies are illustrated with two simulation examples where we show that Hampel's re-descending estimator has superior performance when compared to the fair function or least squares.

## 10.2 Future work

In this section we make some recommendations for future work in the topics discussed in this dissertation. In particular we propose research ideas in NLP sensitivity, stability and observability of MHE, and fault detection and identification. With these ideas we believe that we can extend the understanding of state estimation, and propose improved MHE strategies that can leverage the measurement information to detect faults. Moreover, we also make suggestions that relate to NLP sensitivity that should make the detection of active set changes more efficient.

### NLP Sensitivity

In Chapter 2 we show some strategies that allow us to correct the computed sensitivities when the parameter perturbation causes an active set change. In particular, this strategy sets the offending variables to their bounds using the fix-relax method. To detect active set changes we simply check if any variable or bound multiplier violates bounds. However, this method is not scale independent, and therefore it is prone to errors due to improper tuning parameter settings (e.g., tolerances). Thus, indicator functions have been proposed, that exhibit specific properties that allow us to identify points that violate bounds [96, 97]. These properties basically imply that the indicator function will provide enough information to determine when it is evaluated at feasible or an infeasible point. Moreover, these functions are scale independent, and presumably no parameters need to be tuned. Thus, by evaluating them before and after the sensitivity update they provide enough information to detect active set changes. Regardless, of scaling problems or numerical instabilities. The use of these functions has been described in the development of interior point algo-

rithms [98]. However, to our knowledge, there is no current work that relates indicator functions with NLP sensitivity.

On the other hand, the fix-relax strategy described in Section 2.2.2.1 can be improved. The main drawback of the proposed method is, that once an active set change is detected and the fix-relax step is performed, there is no guarantee that the new updated solution will satisfy bounds. Thus, following the work of [60, 61, 64] a QP can be solved to find an optimal search direction that results in a feasible step. Thus, we substitute the fix-relax strategy for a QP based approach to generate sensitivity information. This is briefly described in Section 2.2.2.1. Therefore, implementation of a QP sensitivity step in IPOPT could prove to be superior to the current implementation in *sIPOPT* to deal with especially large parameter perturbations that may cause active set changes.

### Stability and Observability with Robust M-Estimators

In Chapter 9 we discuss the possibility to reduce the negative effects that outliers and gross errors may have in the estimated states. Here we noted that Hampel's re-descending estimator deals better with these errors than the other estimators. However, it is important to note that if there are not enough redundant measurements observability of the states may be lost when enough measurements are dropped. Thus, the analysis of the detectability and observability of the Robust MHE should be studied. Specifically, through the concepts described in Soroush [99, 100] conditions for detectability and observability could be derived. These could be used to better design and tune the parameters of the M-Estimators.

### Fault Detection and Identification

Detecting gross errors is an important aspect of state estimation since the quality of the estimated is related to the quality of the measurements. These errors can severely bias the estimates, which can cause operational problems. Using robust estimators, as shown in Chapter 9, can greatly reduce the effects outliers and gross errors have on the estimated

states. However, it is important to be able to detect when faults occur and where, so that they can be fixed. Some methods have been developed for Data Reconciliation Problems (DRP), and since both state estimation and DR are similar in the sense that both use measurement information to compute data of the plant, then it is natural to think that these methods could be used to extend the MHE formulations described here. The fault detection problem has been the topic of many research projects, and several strategies have been developed. A very complete survey of the subject is given by Venkatasubramanian *et al.* [101, 102, 103].

Tjoa and Biegler [104] propose the use of tailored distribution functions that combine the effects of gross errors and random errors. In this way, the weight of each type of error in the objective function is chosen automatically based on the magnitude of the residuals. Here, they also discuss specialized SQP methods that can deal with the proposed distributions in an efficient way. Through simulation examples they show that the method is able to compute unbiased estimates, and it is effective to detect gross errors. However, this approach was developed for steady state systems. On the other hand, Albuquerque and Biegler [92] discuss strategies for Dynamic Data Reconciliation and gross error detection based on robust estimators such as the Fair Function and the contaminated normal function. These functions provide a certain degree of robustness towards gross errors. Moreover, box plots are constructed using the residuals to detect faults, and they show the efficacy of their approach with simulation examples.

Soderstrom *et al.* [105] propose a different approach for fault detection and identification. They use a Mixed-Integer Nonlinear Programming (MINLP) method in combination with data reconciliation. In their approach they assume steady states, thus only static linear models are considered. This approach substitutes the quadratic objective function used in DRP with an $L_1$ penalty function of the measurement error with a penalty term to minimize the number of chosen faults. They successfully show how this approach works with simulation examples, and they compare the accuracy of their method with other typical

methods used for fault detection and identification. They conclude that their approach is better since it is able to make accurate predictions using less data. Thus smaller MINLP problems are required. Finally, Arora and Biegler [93] use M-Estimators to reduce the effects of outliers and gross errors in simultaneous dynamic data reconciliation and parameter estimation problems. Moreover, they derive a method for fault detection based on Akaike's Information Criterion (AIC) [106, 107]. This approach leads to MINLP formulations that resemble the problems obtained by [105], however here the formulations are supported by previous theory.

Extending MHE to provide fault detection and identification should be straightforward if we use the work cited above. For example, using Hampel's re-descending estimator we can easily detect faults by looking at the residuals. All those measurements whose residuals lie in the constant region of the function (i.e., $\{\epsilon_j : \epsilon_j > c \, \forall j\}$) can be considered faults. Moreover, it should be possible to extend the advanced step formulation so that when the new measurement is available we can formulate an Mixed Integer QP (MIQP) that will compute the optimal sensitivity and provide fault detection. This would be similar to the MINLP formulations described above. In addition, it is important to understand the effects that M-Estimators have on the approximation of the covariance matrix through the reduced Hessian. For example, comparing the effects on the covariance matrix when using M-Estimators and without them for linear Gaussian systems, should provide insights into the effects that these functions have on the noise structure. In other words, these insights might prove useful to show whether the computed confidence intervals can be used to determine if the estimates are accurate or not.

# Bibliography

[1] Biegler, L.T. and Zavala, V.M., Large-Scale Nonlinear Programming using IPOPT: An Integrating Framework for Enterprise-Wide Dynamic Optimization. *Comput. Chem. Eng.* 33, **2009**, pp. 575–582.

[2] Rawlings, J.B. and Mayne, D.Q., *Model Predictive Control: Theory and Design*. Nob Hill Publishing, LLC., **2009**.

[3] Huang, R.; Harinath, E.; and Biegler, L.T., Economicaly-Oriented Nonlinear Model Predictive Control for Energy Applications. *J. Process Control* 21(4), **2011**, pp. 501–509.

[4] Huang, R.; Harinath, E.; and Biegler, L.T., Robust Stability of an Economically-Oriented Infinite Horizon Nmpc for Cyclic Processes., **2011**, in Press.

[5] Rao, C.V., *Moving-Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. Ph.D. thesis, University of Wisconsin-Madison, **2000**.

[6] Haseltine, E.L. and Rawlings, J.B., Critical Evaluation of Extended Kalman Filtering and Moving-Horizon Estimation. *Ind. Eng. Chem. Res.* 44, **2005**, pp. 2451–2460.

[7] Bryson, A.E. and Ho, Y.C., *Applied Optimal Control: Optimization, Estimation, and Control*. Taylor and Francis, New York, USA, **1975**.

[8] Jazwinski, A.H., *Stochastic Processess and Filtering Theory*. Dover Publications, Inc., Mineola, New York, **2007**.

[9] Vachhani, P.; Rengaswamy, R.; Gangwal, V.; and Narasimhan, S., Recursive Estimation in Constrained Nonlinear Dynamical Systems. *AIChE Journal* 51(3), **2004**, pp. 946–959.

[10] Julier, S.J. and Uhlmann, J.K., Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE* 92(3), **2004**, pp. 401–422.

[11] Julier, S.; Uhlmann, J.; and Durrant-Whyte, H., A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, **1995**, pp. 1628–1632.

[12] Julier, S. and Uhlmann, J., A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, **1997**, pp. 182–193.

[13] Evensen, G., Sequential Data Assimilation with a Nonlinear Quasi-Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics. *Journal of Geophisical Research* 99(C5), **1994**, pp. 143–162.

[14] Burgers, G.; van Leeuwen, P.J.; and Evensen, G., Analysis Scheme in the Ensemble Kalman Filter. *Monthly Weather Review* 126(6), **1998**, pp. 1719–1724.

[15] Chen, Z., Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond. Technical report, Adaptive Syst. Lab., McMaster University, Hamilton, ON, Canada, **2003**.

[16] Arulampalam, S.; Maskell, S.; Gordon, N.; and Clapp, T., A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing* 50, **2002**, pp. 174–188.

[17] Vachhani, P.; Narasimhan, S.; and Rengaswamy, R., Robust and Reliable Estimation via Unscented Recursive Nonlinear Dynamic Data Reconciliation. *J. Process Control* 16, **2006**, pp. 1075–1086.

[18] Prakash, J.; Patwardhan, S.; and Shah, S., Constrained Nonlinear State Estimation Using Ensemble Kalman Filter. *Ind. Eng. Chem. Res.* 49(5), **2010**, pp. 2242–2253.

[19] Prakash, J.; Shah, S.; and Patwardhan, S., Constrained State Estimation Using Particle Filters. In *Proceedings of the 17th IFAC World Congress*, eds. M.J. Chung and P. Misra, volume 17, **2008**, pp. 6472–6477.

[20] Prakash, J.; Shah, S.; and Patwardhan, S., On the Choice of Importance Distributions for Unconstrained and Constrained State Estimation Using Particle Filter. *J. Process Control* 21, **2011**, pp. 3–16.

[21] Nocedal, J. and Wright, S.J., *Numerical Optimization*. 2nd edition, Springer, New York, USA, **2006**.

[22] Biegler, L.T., *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM, **2010**.

[23] Edgar, T.F. and Himmelblau, D.M., *Optimization of Chemical Processes*. 2nd edition, McGraw-Hill Science/Engineering/Math, USA, **2001**.

[24] Biegler, L.T.; Cervantes, A.M.; and Wächter, A., Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.* 57(4), **2002**, pp. 575–593.

[25] Biegler, L.T., Efficient Solution of Dynamic Optimization and NMPC Problems. In *Nonlinear Model Predictive Control*, eds. F. Allgöwer and A. Zheng, volume 26 of *Progress in Systems and Control Theory*, Birkhäuser Basel, **2000**.

[26] Biegler, L.T., An Overview of Simultaneous Strategies for Dynamic Optimization. *Comput. Chem. Eng.* 46, **2007**, pp. 1043–1053.

[27] Fiacco, A.V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, **1983**.

[28] Zavala, V.M., *Computational Strategies for the Optimal Operation of Large-Scale Chemical Processes*. Ph.D. thesis, Carnegie Mellon University, **2008**.

[29] Zavala, V.M.; Laird, C.D.; and Biegler, L.T., Fast Implementations and Rigorous Models: Can Both be Accommodated in NMPC? *Journal of Robust and Nonlinear Control* 18(8), **2008**, pp. 800–815.

[30] Pistikopoulos, E.N., Perspectives in Multiparametric Programming and Explicit Model Predictive Control. *AIChE Journal* 55(8), **2009**, pp. 1918–1925.

[31] Pistikopoulos, E.N.; Georgiadis, M.C.; and Dua, V., *Multi-Parametric Programming: Theory, Algorithms, and Applications*. Wiley-VCH, **2007**.

[32] Pistikopoulos, E.N.; Georgiadis, M.C.; and Dua, V., *Multi-Parametric Model-Based Control: Theory and Applications*. Wiley-VCH, **2007**.

[33] Rao, C.V.; Rawlings, J.B.; and Lee, J.H., Constrained Linear State Estimation–A Moving Horizon Approach. *Automatica* 37, **2001**, pp. 1619–1628.

[34] Rao, C.V. and Rawlings, J.B., Constrained Process Monitoring: Moving-Horizon Approach. *AIChE Journal* 48(1), **2002**, pp. 97–109.

[35] Tenny, M.J. and Rawlings, J.B., Efficient Moving Horizon Estimation and Nonlinear Model Predictive Control. In *Proceedings of the American Control Conference*, Anchorage, Alaska, **2002**, pp. 4475–4480.

[36] Qu, C.C. and Hahn, J., Computation of Arrival Cost for Moving Horizon Estimation via Unscented Kalman Filtering. *J. Process Control* 19, **2009**, pp. 358–363.

[37] Ungarala, S., Computing Arrival Cost Parameters in Moving Horizon Estimation Using Sampling Based Filters. *J. Process Control* 19, **2009**, pp. 1576–1588.

[38] Lang, L.; Chen, W.; Bakshi, B.R.; Goel, P.K.; and Ungarala, S., Bayesian Estimation via Sequential Monte Carlo Sampling-Constrained Dynamic Systems. *Automatica* 43, **2007**, pp. 1615–1622.

[39] Rawlings, J.B. and Bakshi, B.R., Particle Filtering and Moving Horizon Estimation. *Comput. Chem. Eng.* 30, **2006**, pp. 1529–1541.

[40] Tatiraju, S.; Soroush, M.; and Ogunnaike, B.A., Multirate Nonlinear State Estimation with Application to a Polymerization Reactor. *AIChE Journal* 45(4), **1999**, pp. 769–780.

[41] Zambare, N.; Soroush, M.; and Grady, M.C., Real-Time Multirate State Estimation in a Pilot-Scale Polymerization Reactor. *AIChE Journal* 48(5), **2002**, pp. 1022–1033.

[42] Gopalakrishnan, A.; Kaisare, N.S.; and Narasimhan, S., Incorporating Delayed and Infrequent Measurements in Extended Kalman Filter Based Nonlinear State Estimation. *J. Process Control* 21, **2011**, pp. 119–129.

[43] Zambare, N.; Soroush, M.; and Ogunnaike, B.A., A Method for Robust Multi-Rate State Estimation. *J. Process Control* 13, **2003**, pp. 337–355.

[44] Zambare, N.; Soroush, M.; and Grady, M., Multi-Rate Nonlinear State Estimation in a Polymerization Reactor: A Real-time Study. In *American Control Conference, 2002. Proceedings of the 2002*, volume 4, **2002**, pp. 2701–2706 vol.4.

[45] Gudi, R.D.; Shah, S.L.; and Gray, M.R., Adaptive Multirate State and Parameter Estimation Strategies with Application to a Bioreactor. *AIChE Journal* 41(11), **1995**, pp. 2451–2464.

[46] Wächter, A. and Biegler, L.T., On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming* 106(1), **2006**, pp. 25–57.

[47] Fourer, R.; Gay, D.M.; and Kernighan, B.W., *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Pacific Grove, **2002**.

[48] Wächter, A. and Biegler, L.T., Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM J. Comput.* 16(1), **2005**, pp. 1–31.

[49] Wächter, A. and Biegler, L.T., On the Implementation of an Interior-Point FIlter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming* 106, **2006**, pp. 25–57.

[50] Wächter, A. and Biegler, L.T., On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming* 106(1), **2006**, pp. 25–57.

[51] Fiacco, A.V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, **1983**.

[52] Fiacco, A.V. and Ishizuka, Y., Sensitivity and Stability Analysis for Nonlinear Programming. *Annals of Operations Research* 27, **1990**, pp. 215–236.

[53] Büskens, C. and Maurer, H., Sensitivity analysis and real-time control of parametric control problems using nonlinear programming methods. In *Online Optimization of Large-scale Systems*, eds. M. Grötschel; S. Krumke; and J. Rambau, Springer-Verlag, **2001**, pp. 57–68.

[54] Kyparsis, J., Sensitivity Analysis for Nonlinear Programs and Variational Inequalities with Nonunique Multipliers. *Mathematics of Operations Research* 15(2), **1990**, pp. 286–298.

[55] Kojima, M., Strongly State Stationary Solutions in Nonlinear Programs. In *Analysis and Computation of Fixed Points*, ed. S.M. Robinson, Academic Press, New York, **1980**.

[56] Kojima, M. and Hirabayashi, R., Continuous Deformation of Nonlinear Programs. *Mathematical Programming Study* 21, **1984**, pp. 150–198.

[57] Jongen, H.T.; Jonker, P.; and Twilt, F., *Nonlinear Optimization in Finite Dimensions*. Kluwer Academic Publishers, **2000**.

[58] Jongen, H.T.; Meer, K.; and Triesch, E., *Optimization Theory*. Kluwer Academic Publishers, **2004**.

[59] Fiacco, A.V. and Ghaemi, A., A user's manual for SENSUMT. A penalty function computer program for solution, sensitivity analysis and optimal bound value calculation in parametric nonlinear programs. Technical Report T-434, Management Science and Engineering, George Washington University, **1980**.

[60] Ganesh, N. and Biegler, L.T., A reduced hessian strategy for sensitivity analysis of optimal flowsheets. *AIChE* 33, **1987**, pp. 282–296.

[61] Wolbert, D.; Joulia, X.; Koehret, B.; and Biegler, L.T., Flowsheet Optimization and Optimal Sensitivity Analysis Using Exact Derivatives. *Comput. Chem. Eng.* 18, **1994**, p. 1083.

[62] Forbes, J. and Marlin, T.E., Design cost: a systematic approach to technology selection for model-based real-time optimization systems. *Comput. Chem. Eng.* 20, **1996**, pp. 717–734.

[63] Diehl, M.; Findeisen, R.; and Allgöwer, F., *Real-Time PDE-Constrained Optimization*, chapter A Stabilizing Real-Time Implementation of Nonlinear Model Predictive Control. SIAM, **2007**, pp. 25–52.

[64] Kadam, J. and Marquardt, W., Sensitivity-based Solution Updates in Closed-loop Dynamic Optimization. In *Proceedings of the DYCOPS 7 Conference*, Elsevier, **2004**.

[65] Zavala, V.M. and Biegler, L.T., The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica* 45(1), **2009**, pp. 86 – 93.

[66] Forsgren, A.; Gill, P.E.; and Wright, M.H., Interior Point Methods for Nonlinear Optimization. *SIAM Review* 44(4), **2002**, pp. 525–597.

[67] Pirnay, H.; López-Negrete, R.; and Biegler, L.T., Optimal Sensitivity Based on IPOPT, submitted to *Math Prog Comp*, **2011**.

[68] Pirnay, H.; López-Negrete, R.; and Biegler, L.T., *sIPOPT Reference Manual*, **2011**.

[69] Beltracchi, T.J. and Gabriele, G.A., An Investigation of New Methods for Estimating Parameter Sensitivities. Technical Report 4245, NASA Contractor Report, **1989**.

[70] Diehl, M., *Real-Time Optimization for Large Scale Nonlinear Processes*. Ph.D. thesis, Universität Heidelberg, **2001**.

[71] Zavala, V.M., *Computational Strategies for the Operation of Large-Scale Chemical Processes*. Ph.D. thesis, Carnegie Mellon University, **2008**.

[72] Bartlett, R.A. and Biegler, L.T., QPSchur: A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming. *Optimization and Engineering* 7, **2006**, pp. 5–32.

[73] Chen, W.S.; Bakshi, B.R.; Goel, P.K.; and Ungarala, S., Bayesian Estimation via Sequential Monte Carlo Sampling: Unconstrained Nonlinear Dynamic Systems. *Ind. Eng. Chem. Res.* 43(14), **2004**, pp. 4012–4025.

[74] Kalman, R.E., Contributions to the Theory of Optimal Control. *Bull. Soc. Math. Mex.* 5, **1960**, pp. 102–119.

[75] Söderström, T., *Discrete-Time Stochastic Systems: Estimation and Control*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, **2002**.

[76] Stengel, R.F., *Optimal Control and Estimation*. Dover Publications, Inc., New York, USA, **1994**.

[77] Kandepu, R.; Foss, B.; and Imsland, L., Applying the Unscented Kalman Filter for Nonlinear State Estimation. *J. Process Control* 18(7-8), **2008**, pp. 753–768.

[78] Wan, E.A. and Van der Merwe, R., Chapter 7: The Unscented Kalman Filters. In *Kalman Filtering and Neural Networks*, ed. S. Haykin, Wiley, New York, **2001**, pp. 221–280.

[79] Friedland, B. and Bernstein, I., Estimation fo the State of a Nonlinear Process in the Presence of Nongaussian Noise and Disturbances. *Journal of the Franklin Institute* 281(6), **1966**, pp. 455–480.

[80] Betrsekas, D.P., *Dynamic Programming and Optimal Control*, volume I. 3rd edition, Athena Scientific, Belmont, Massachussetts, **2005**.

[81] López-Negrete, R.; Patwardhan, S.C.; and Biegler, L.T., Constrained particle filter approach to approximate the arrival cost in Moving Horizon Estimation. *J. Process Control* 21(6), **2011**, pp. 909–919.

[82] Tenny, M.J., *Computational Strategies for Nonlinear Model Predictive Control*. Ph.D. thesis, University of Wisconsin-Madison, **2002**.

[83] Rao, C.V. and Rawlings, J.B., Nonlinear Moving Horizon State Estimation. In *Nonlinear Model Predictive Control*, eds. F. Allgöwer and A. Zheng, volume 26 of *Progress in Systems and Control Theory*, Birkhäuser Verlag, Basel, Switzerland, **2000**, pp. 45–69.

[84] Robertson, D.G.; Lee, J.H.; and Rawlings, J.B., A Moving Horizon Based Approach for Least-Squares Estimation. *AIChE Journal* 42(8), **1996**, pp. 2209–2224.

[85] Rao, C.V.; Rawlings, J.B.; and Mayne, D.Q., Constrained State Estimation for Nonlinear Discrete-Time Systems: Stability and Moving Horizon Approximations. *IEEE Trans. Auto. Cont.* 48(2), **2003**, pp. 246–258.

[86] Gopaluni, R.B., Nonlinear System Identification Under Missing Observations: The Case of Unknown Model Structure. *J. Process Control* 20, **2010**, pp. 314–324.

[87] Zavala, V.M.; Laird, C.D.; and Biegler, L.T., A Fast Moving Horizon Estimation Algorithm Based on Nonlinear Programming Sensitivity. *J. Process Control* 18(9), **2008**, pp. 876–884.

[88] Daum, F. and Huang, J., Curse of dimensionality and particle filters. In *Proceedings of the IEEE Aerospace Conference*, volume 4, **2003**, pp. 1979–1993.

[89] Flores-Tlacuahuac, A.; Bielger, L.T.; and Saldívar-Guerra, E., Optimal Grade Transitions in the High-Impact Polystyrene Polymerization Process. *Ind. Eng. Chem. Res.* 45, **2006**, pp. 6175–6189.

[90] López-Negrete, R. and Flores-Tlacuahuac, A., Optimal Start-Up and Product Transition Policies of a Reactive Distillation Column. *Ind. Eng. Chem. Res.* 46, **2007**, pp. 2092–2111.

[91] Cervantes, A.M. and Biegler, L.T., Large-Scale DAE Optimization Using a Simultaneous NLP Formulation. *AIChE Journal* 44(5), **1998**, pp. 1038–1050.

[92] Albuquerque, J.S. and Biegler, L.T., Data Reconciliation and Gross-Error Detection for Dynamic Systems. *AIChE Journal* 42(10), **1996**, pp. 2841–2856.

[93] Arora, N. and Biegler, L.T., Redescending Estimatiors for Data Reconciliation and Parameter Estimation. *Comput. Chem. Eng.* 25, **2001**, pp. 1585–1599.

[94] Huber, P.J., *Robust Statistics*. John Wiley and Sons, New York, **1981**.

[95] Hampel, F.R., The Influence Curve and its Role in Robust Estimation. *Journal of the American Statistical Association* 69(346), **1974**, pp. 383–393.

[96] Tapia, R.A., Role of Slack Variables in Quasi-Newton Methods for Constrained Optimization. In *Numerical Optimization of Dynamic Systems*, eds. L.C.W. Dixon and G.P. Szegö, Elsevier Science Ltd, **1979**.

[97] Facchinei, F.; Fischer, A.; and Kanzow, C., On the Identification of Zero Variables in an Interior-Point Framework. *SIAM J. Optim.* 10(4), **1998**, pp. 1058–1078.

[98] El-Bakry, A.S.; Tapia, R.A.; and Zhang, Y., A Study of Indicators for Identifying Zero Variables in Interior-Point Methods. *SIAM Review* 36(1), **1994**, pp. 45–72.

[99] Soroush, M., Nonlinear State-Observer Design with Application to Reactors. *Chem. Eng. Sci.* 52(3), **1997**, pp. 387–404.

[100] Soroush, M., State and Parameter Estimations and their Applications in Process Control. *Comput* 23(2), **1998**, pp. 229–245.

[101] Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; and Kavuri, S.N., A Review of Process Fault Detection and Diagnosis: Part I: Quantitative Model-Based Methods. *Comput. Chem. Eng.* 27(3), **2003**, pp. 293–311.

[102] Venkatasubramanian, V.; Rengaswamy, R.; and Kavuri, S.N., A Review of Process Fault Detection and Diagnosis: Part II: Qualitative Models and Search Strategies. *Comput. Chem. Eng.* 27(3), **2003**, pp. 313–326.

[103] Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S.N.; and Yin, K., A Review of Process Fault Detection and Diagnosis: Part III: Process History Based Methods. *Comput. Chem. Eng.* 27(3), **2003**, pp. 327–346.

[104] Tjoa, I.B. and Biegler, L.T., Simultaneous Strategies for Data Reconciliation and Gross Error Detection of Nonlinear Systems. *Comput. Chem. Eng.* 15(10), **1991**, pp. 679–690.

[105] Soderstrom, T.A.; Himmelblau, D.M.; and Edgar, T.F., A Mixed Integer Optimization Approach for Simultaneous Data Reconciliation and Identification oControl Engineering Practicef Measurement Bias. *Control Engineering Practice* 9(8), **2001**, pp. 869–876.

[106] Akaike, H., A New Look at the Statistical Model Identification. *IEEE Trans. Automat. Contr.* 19, **1974**, pp. 716–723.

[107] Yamamura, K.; Nakajima, M.; and Matsuyama, H., Detection of Gross Errors in Process Data Using Mass and Energy Balances. *International Chemical Engineering* 28(1), **1988**, pp. 91–98.

[108] Rajaraman, S.; Hahn, J.; and Mannan, M.S., A Methodology for Fault Detection, Isolation, and Identification for Nonlinear Processes with Parametric Uncertainties. *Ind. Eng. Chem. Res.* 43(21), **2004**, pp. 6774–6786.

[109] Diehl, M.; Bock, H.G.; Schlöder, J.P.; Findeisen, R.; Nagy, Z.; and Allgöwer, F., Real-Time Optimization and Nonlinear Model Predictive Control of Process Governed by Differential-Algebraic Equations. *J. Process Control* 12, **2002**, pp. 577–585.

[110] Hicks, G.A. and Ray, W.H., Approximation Methods for Optimal Control Synthesis. *Can. J. Chem. Eng*. 49(4), **1971**, pp. 522–528.

# Appendix A

# Derivation of the Discrete-Time Kalman Filter

Here we derive the discrete time Kalman Filter following the maximum likelihood approach. The prediction of the mean is obtained by propagating the previous known estimate using the linear model. The correction step is done as a linear weighted minimum least squares problem that results from applying the maximum likelihood approach.

For a linear system of the following form, where $w_k$ is a random variable that describes the state noise:

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}, \tag{A.1}$$

with measurements ($v_k$ is the measurement noise):

$$y_k = H_k x_k + v_k \tag{A.2}$$

For such a system the Kalman Filter equations are defined as follows.

**Propagation or Prediction**

$$x_{k|k-1} = A_{k-1}x_{k-1|k-1} + B_{k-1}u_{k-1} \tag{A.3}$$

$$\Gamma_{k|k-1} = A_{k-1}\Gamma_{k-1|k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \tag{A.4}$$

**Correction**

$$x_{k|k} = x_{k|k-1} + K_k \left( y_k - H_k x_{k|k-1} \right) \tag{A.5}$$

where

$$K_k = \Gamma_{k|k-1} H_k^T \left( H_k \Gamma_{k|k-1} H_k^T + R_k \right)^{-1} \tag{A.6}$$

$$\Gamma_{k|k} = \left( \Gamma_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k \right)^{-1} \tag{A.7}$$

Here $x_{k|k-1}$ and $x_{k|k}$ are the predicted mean before the measurement and the corrected mean after the measurement respectively. $\Gamma_{k|k-1}$ and $\Gamma_{k|k}$ are the predicted covariance matrix before the measurement and the corrected covariance after the measurement. $K_k$ is the filter gain matrix and $R_k$ is the measurement noise covariance matrix.

To derive this equations in Section A.1 we first define the way that the mean value of the states and the covariance propagate through the linear system. Then in Section A.2 we define a recursive weighted linear least squares estimator, through the use of Moving Horizon Estimation (MHE), that can correct previous estimates of the states using the information of a new measurement. In Section A.3 we derive the Kalman Filter equations shown above using the tools generated in the previous sections. Finally, in the last sections we generalize the solution of the MHE problem to obtain a recursive analytical solution.

## A.1 Prediction/Propagation

The propagation of the mean value of the state is done by calculating $\mathbb{E}[x_k] = \bar{x}_k$, i.e. if the model is

$$x_k = A_{k-1} x_{k-1} + B_{k-1} u_{k-1} + G_{k-1} w_{k-1} \tag{A.8}$$

Here it is assumed that $\mathbb{E}[w_k] = 0$, and $u_k$ is a known non-random variable, then

$$\begin{aligned}\mathbb{E}\left[x_k\right] =& \mathbb{E}\left[f\left(x_{k-1}, u_{k-1}, w_{k-1}\right)\right] = \\ & \mathbb{E}\left[A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + G_{k-1}w_{k-1}\right] = \\ & A_{k-1}\mathbb{E}\left[x_{k-1}\right] + B_{k-1}u_{k-1}\end{aligned} \tag{A.9}$$

therefore,

$$\bar{x}_k = A_{k-1}\bar{x}_{k-1} + B_{k-1}u_{k-1} \tag{A.10}$$

For the covariance a similar approach can be taken if we assume the following

$$\mathbb{E}\left[\left(x_k - \bar{x}_k\right)\left(x_k - \bar{x}_k\right)^T\right] = \bar{\Gamma}_k \tag{A.11}$$

$$\mathbb{E}\left[w_k w_j^T\right] = \delta_{k,j}Q_k \tag{A.12}$$

Where $\delta_{k,j}$ is the Kronecker delta. Taking all of this into account we can evaluate the error between the state and the mean by

$$x_k - \bar{x}_k = A_{k-1}(x_{k-1} - \bar{x}_{k-1}) + B_{k-1}\cancel{(u_{k-1} - u_{k-1})} + G_{k-1}w_{k-1} \tag{A.13}$$

Thus the covariance is calculated by equation (A.11) as

$$\begin{aligned}\bar{\Gamma}_k =& A_{k-1}\mathbb{E}\left[(x_{k-1} - \bar{x}_{k-1})(x_{k-1} - \bar{x}_{k-1})^T\right] A_{k-1}^T + \\ & A_{k-1}\mathbb{E}\left[(x_{k-1} - \bar{x}_{k-1})w_k^T\right] G_{k-1}^T + \\ & G_{k-1}\mathbb{E}\left[w_k(x_{k-1} - \bar{x}_{k-1})^T\right] A_{k-1}^T + G_{k-1}\mathbb{E}\left[w_k w_k^T\right] G_{k-1}^T\end{aligned} \tag{A.14}$$

Which finally turns out to be

$$\begin{aligned}\bar{\Gamma}_k =& A_{k-1}\bar{\Gamma}_{k-1}A_{k-1}^T + A_{k-1}M_{k-1}G_{k-1}^T + \\ & G_{k-1}M_{k-1}^T A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T\end{aligned} \tag{A.15}$$

and

$$M_k = \mathbb{E}\left[(x_k - \bar{x}_k)w_k^T\right] \tag{A.16}$$

$$Q_k = \mathbb{E}\left[w_k w_k^T\right] \tag{A.17}$$

Another possible simplifying assumption that can be done is to assume white noise, and, therefore, $\mathbb{E}\left[(x_k - \bar{x}_k)w_k^T\right] = 0$. This assumption will be made throughout the rest of this text. This way equation (A.15) becomes

$$\bar{\Gamma}_k = A_{k-1}\bar{\Gamma}_{k-1}A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \tag{A.18}$$

## A.2 Moving Horizon Estimation

The Kalman Filter estimation equations can be derived through the use of Moving Horizon Estimation [34]. The problem is to estimate the states at a certain time $T$. This problem, using the MHE formulation, is defined as follows, where the horizon length is $N$ sampling points. In this case time point $k = 0$ is the time at $T - N$.

$$\begin{aligned}
&\min_{x_0,\{w_k\}_{k=0}^{N-1}} \Phi(x_0, \{w_k\}_{k=0}^{N-1}) = \mathcal{P}_{x_0} + \sum_{k=0}^{N-1} \parallel v_k \parallel_{R_k}^2 + \parallel w_k \parallel_{Q_k}^2 \\
&\text{subject to} \\
&\quad x_{k+1} = A_k x_k + B_k u_k + G_k w_k, \;\; k = 0 \ldots N-1 \\
&\quad y_k = H_k x_k + v_k, \;\; k = 0 \ldots N-1,
\end{aligned} \tag{A.19}$$

where $\mathcal{P}_{x_0}$ is the arrival cost and it is defined as follows:

$$\mathcal{P}_{x_0} = \parallel x_0 - \bar{x}_0 \parallel_{\Pi_0}^2 . \tag{A.20}$$

Here $\bar{x}_0$ is the predicted value of $x_0$ (i.e., the propagated mean value). From this formulation we would obtain the estimate of the state at point $N-1$ (i.e., $x_{N-1}$ is the state estimate at $T$), and the predicted value of the state at point $N$ ($x_N$ in the future time $T+1$). That is to say that we are evaluating the corrected (after the measurement) estimate of the state, and the propagated value (before measurement) with this formulation.

### A.2.1 Horizon Length $N = 1$

To generate the estimator of the Kalman Filter we first consider the Karush Kuhn Tucker (KKT) conditions for the above problem using a horizon length $N = 1$. In this case problem (A.19) reduces to the following:

$$\min_{x_0,\{w_0\}} \Phi(x_0, w_0) = (x_0 - \bar{x}_0)^T \Pi_0 (x_0 - \bar{x}_0) + (y_0 - H_0 x_0)^T R_0 (y_0 - H_0 x_0) + w_0^T Q_k w_0 \tag{A.21}$$

subject to

$$x_1 = A_0 x_0 + B_0 u_0 + G_0 w_0$$

To evaluate the necessary conditions for the above problem we generate the following Lagrangian function.

$$\mathcal{L}(x_0, w_0) = (x_0 - \bar{x}_0)^T \Pi_0 (x_0 - \bar{x}_0) + (y_0 - H_0 x_0)^T R_0 (y_0 - H_0 x_0) + w_0^T Q_k w_0 + \lambda_1^T (x_1 - A_0 x_0 - B_0 u_0 - G_0 w_0) \tag{A.22}$$

The KKT conditions for the problem are the following [21].

$$\nabla_{x_0}\mathcal{L} \quad = 0 = \quad \Pi_0\left(x_0 - \bar{x}_0\right) - H_0^T R_0\left(y_0 - H_0 x_0\right) - A_0^T\lambda_1 \tag{A.23}$$

$$\nabla_{x_1}\mathcal{L} \quad = 0 = \quad \lambda_1 \tag{A.24}$$

$$\nabla_{w_0}\mathcal{L} \quad = 0 = \quad Q_0 w_0 - G_0^T\lambda_1 \tag{A.25}$$

$$\nabla_{\lambda_1}\mathcal{L} \quad = 0 = \quad x_1 - A_0 x_0 - B_0 u_0 - G_0 w_0 \tag{A.26}$$

From equation (A.23) and (A.24) we can solve for $x_0$ to obtain.

$$x_0 = \left(\Pi_0 + H_0^T R_0 H_0\right)^{-1}\left(\Pi_0\bar{x}_0 + H_0^T R_0 y_0\right) \tag{A.27}$$

If we substitute equations (A.27) and (A.24) into (A.26) we get the following after some algebraic manipulation.

$$x_1 = A_0(\Pi_0 + H_0^T R_0 H_0)^{-1}(\Pi_0\bar{x}_0 + H_0 R_0 y_0) + B_0 u_0$$

Using the matrix inversion lemma for the above, and after some algebra we get.

$$x_1 = B_0 u_0 + A_0\left[\bar{x}_0 + \Pi_0 H_0^T(H_0\Pi_0 H_0^T + R_0)^{-1}(y_0 - H_0\bar{x}_0)\right] \tag{A.28}$$

From equation (A.28) we can extract the recursive estimator if we consider that $x_{0|-1} = \bar{x}_0$ and $\Pi_{0|-1} = \Pi_0$. Thus we get the following equation.

$$x_{0|0} = x_{0|-1} + \Pi_{0|-1} H_0^T(H_0\Pi_{0|-1}H_0^T + R_0)^{-1}(y_0 - H_0 x_{0|-1}) \tag{A.29}$$

Also, from equation (A.27) and the assumptions made above, we can extract the following covariance update equation.

$$\Pi_{0|0} = (\Pi_{0|-1}^{-1} + H_0^T R_0 H_0)^{-1} \tag{A.30}$$

If we compared these last equations to equations (A.5) to (A.7) we can notice that they are the same. Therefore, these equations will correspond to the recursive estimator that

we can use to obtain the Kalman Filter equations. This simple example also shows that the value of $x_1$ is the propagated mean value at $N = 1$ (i.e., $x_{N|N-1}$), and that $x_0$ in equation (A.27) is in fact $x_{0|0}$ (the corrected value of the state). This will help out in the final section where we obtain the solution of the general MHE problem.

## A.3 Kalman Filter Derivation

To derive the Kalman Filter equations we can use the information of the previous sections. Here we use the following notation: for the expected value of the state at time $k$ before the measurement (i.e., given $k-1$ measurements) we use $x_{k|k-1}$, for the corrected value of the state (after the measurement) we will use $x_{k|k}$, and the same will apply for the values of the covariance matrix ($\Gamma_{k|k-1}$ and $\Gamma_{k|k}$) before and after the measurement. We also define the following:

$$\mathbb{E}\left[w_k\right] = 0, \quad \text{and} \quad \mathbb{E}\left[w_k w_k^T\right] = Q_k \tag{A.31}$$

This means that $w_k$ is a random vector with mean value zero and covariance $Q_k$. Also, since the state $x_k$ is a random vector, and if $w_k$ and $x_k$ are independent we will be able to use equations (A.10) and (A.18).

First let's assume that a linear system makes a discrete transition from a state $k-1$ to a state $k$. This transition will occur as in equation (A.1), and we also know a value for the mean and the covariance at $k$ given the past knowledge of the process ($x_{k|k-1}$ and $\Gamma_{k|k-1}$). If we now take a new measurement we can use the recursive estimator equations (A.5) to (A.6) derived in Section A.2.1 to correct the value of the state prior to the measurement. That is to say that we evaluate the following.

$$x_{k|k} = x_{k|k-1} + K_k\left(y_k - H_k x_{k|k-1}\right) \tag{A.32}$$

where,

$$K_k = \Gamma_{k|k-1} H_k^T \left( H_k \Gamma_{k|k-1} H_k^T + R_k \right)^{-1} \tag{A.33}$$

We can also update (or correct) the covariance matrix using equation (A.7), thus we get the following.

$$\Gamma_{k|k} = \left( \Gamma_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k \right)^{-1} \tag{A.34}$$

This gives the corrected state estimate $x_{k|k}$ at time point $k$. If we now wish to evaluate a new estimate at time point $k+1$ we would first need an estimate of the mean and covariance (at $k+1$) with all the past knowledge. For this we could use equations (A.10) and (A.18) to propagate the corrected estimates we obtained before to the next time point. This means that we need to evaluate the following.

$$\begin{aligned} x_{k|k-1} =& A_{k-1} x_{k-1|k-1} + B_{k-1} u_{k-1} \\ \Gamma_{k|k-1} =& A_{k-1} \Gamma_{k-1|k-1} A_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \end{aligned} \tag{A.35}$$

Finally, equations (A.32) to (A.35) form the discrete Kalman Filter.

## A.4 Horizon length $N$

To derive the Kalman Filter equations (A.3) to (A.7) we can treat the general case in which the horizon length is $N$. The MHE problem to solve is defined by (A.19). For this problem the Lagrangian function is defined as follows.

$$\begin{aligned} \mathcal{L}(x_0, \{w_k\}_{k=0}^{N-1}) =& (x_0 - \bar{x}_0)^T \Pi_0 (x_0 - \bar{x}_0) + \\ & \sum_{k=0}^{N-1} \left\{ (y_k - H_k x_k)^T R_k (y_k - H_k x_k) + w_k^T Q_k w_k \right\} + \\ & \sum_{k=0}^{N-1} \lambda_{k+1}^T (x_{k+1} - A_k x_k - B_k u_k - G_k w_k) \end{aligned} \tag{A.36}$$

The necessary or KKT conditions for this problem are defined by the following equations.

$$\nabla_{x_0}\mathcal{L} = 0 = \Pi_0(x_0 - \bar{x}_0) - H_0^T R_0(y_0 - H_0 x_0) - A_0^T \lambda_1 \tag{A.37}$$

$$\nabla_{x_k}\mathcal{L} = 0 = -H_k^T R_(y_k - H_k x_k) + \lambda_k - A_k^T \lambda_{k+1},\ k = 1, \ldots, N-1 \tag{A.38}$$

$$\nabla_{x_N}\mathcal{L} = 0 = \lambda_N \tag{A.39}$$

$$\nabla_{w_k}\mathcal{L} = 0 = Q_k w_k - G_k^T \lambda_{k+1},\ k = 0, \ldots, N-1 \tag{A.40}$$

$$\nabla_{\lambda_{k+1}}\mathcal{L} = 0 = x_{k+1} - B_k u_k - A_k x_k - G_k w_k,\ k = 0, \ldots, N-1 \tag{A.41}$$

These equations can be solved to obtain a recursive analytical solution. The recursive solution is based on assuming an affine relation $x_k = M_k \lambda_k + r_{M_k}$ which can be substituted to solve for $x_N$. To do this we can start by solving equation (A.37) for $x_0$, and substituting it into equation (A.41) along with the values of $\lambda_1$ and $w_0$ from equations (A.38) and (A.40), respectively using the relation between $x_k$ and $\lambda_k$. If we do this for all $x_k$, and each time we substitute each $x_{k-1}$, $\lambda_k$ and $w_{k-1}$ in equation (A.41) a solution for $x_N$ can be obtained. Once this value is calculated we can recursively evaluate the rest of the states ($x_k$) starting from $k = N-1$ to $k = 0$. The solution is as follows.

$$\lambda_N = 0 \tag{A.42}$$

$$x_N = r_{M_N} \tag{A.43}$$

$$\lambda_k = M_k(x_k - r_{M_k}),\ k = N-1, \ldots, 1 \tag{A.44}$$

$$x_k = P_k A_k^T \lambda_{k+1} + P_k(M_k r_{M_k} + H_k^T R_k y_k),\ k = N-1, \ldots, 0 \tag{A.45}$$

$$w_k = Q_k G_k^T \lambda_{k+1},\ k = N-1, \ldots, 0, \tag{A.46}$$

where the following definitions have been used.

$$
\begin{aligned}
M_0 &= \Pi_0 && \text{(A.47)}\\
M_{k+1} &= A_k P_k A_k^T + G_k Q_k G_k^T && \text{(A.48)}\\
P_k &= (M_k + H_k^T R_k H_k)^{-1},\ k = 0, \ldots, N-1 && \text{(A.49)}\\
r_{M_0} &= \bar{x}_0 && \text{(A.50)}\\
r_{M_{k+1}} &= B_k u_k + A_k P_k (M_k r_{M_k} + H_k^T R_k y_k),\ k = 0, \ldots, N-1 && \text{(A.51)}
\end{aligned}
$$

In this case $M_k$ is the propagated matrix $\Gamma_{k|k-1}$ (before measurement) through the sampling points in the MHE, while $P_k$ is its corrected value $\Gamma_{k|k}$. Again, the estimated state (after measurement) is given by $x_{N-1}$, while $x_1$ is the propagated state used as $\bar{x}_0$ for the new MHE problem that will be solved when the new measurement is available at the next time point. The solution will be given by $x_{N-1}$, that is we will obtain $x_{N-1|N-1}$. This can be shown by applying Equation (A.42) to (A.45) for $k = N-1$ to get the following:

$$
x_{N-1} = P_{N-1} \left( M_{N-1} r_{M_{N-1}} + H_{N-1} R_{N-1} y_{N-1} \right) \tag{A.52}
$$

Also we can apply Equation (A.49) to the above to get:

$$
x_{N-1} = \left( M_{N-1} + H_{N-1}^T R_{N-1} H_{N-1} \right)^{-1} \left( M_{N-1} r_{M_{N-1}} + H_{N-1} R_{N-1} y_{N-1} \right) \tag{A.53}
$$

We can use the matrix inversion lemma for the first term of Equation (A.53), and after some algebraic manipulation we get the following equation.

$$
\begin{aligned}
x_{N-1} =& r_{M_{N-1}} - M_{N-1} H_{N-1}^T \left( H_{N-1} M_{N-1} H_{N-1}^T + R_{N-1} \right)^{-1} H_{N-1} r_{M_{N-1}} + \\
& M_{N-1} H_{N-1}^T \left[ \mathbb{I} - \left( H_{N-1} M_{N-1} H_{N-1}^T + R_{N-1} \right)^{-1} H_{N-1} M_{N-1} H_{N-1}^T \right] R_{N-1} y_{N-1}
\end{aligned} \tag{A.54}
$$

Here $\mathbb{I}$ is the identity matrix of proper dimensions. We can proceed now by defining the following equality.

$$\mathbf{I} - \left(H_{N-1}M_{N-1}H_{N-1}^T + R_{N-1}\right)^{-1} H_{N-1}M_{N-1}H_{N-1}^T = \left(H_{N-1}M_{N-1}H_{N-1}^T + R_{N-1}\right)^{-1} R_{N-1} \tag{A.55}$$

If we apply this to Equation (A.54), and after grouping terms we get the following.

$$x_{N-1} = r_{M_{N-1}} + M_{N-1}H_{N-1}^T \left(H_{N-1}M_{N-1}H_{N-1}^T + R_{N-1}\right)^{-1} \left(y_{N-1} - H_{N-1}r_{M_{N-1}}\right) \tag{A.56}$$

Equation (A.56) is clearly equal to Equation (A.5) with $x_{N-1|N-2} = r_{M_{N-1}}$, $\Gamma_{N-1|N-2} = M_{N-1}$, and a gain matrix defined as follows.

$$K_{N-1} = M_{N-1}H_{N-1}^T \left(H_{N-1}M_{N-1}H_{N-1}^T + R_{N-1}\right)^{-1} \tag{A.57}$$

Finally, we can also see that if $x_{N-1}$ is the corrected value of the state, then by substituting Equations (A.51) and (A.52) into (A.43), we will obtain the predicted mean value at $k = N$ as if using Equation (A.3).

# Appendix B

# Mathematical Models

The mathematical models of the different examples used are described in this chapter. The first system is a discrete time model, and the last three are index 1 DAE systems that describe the dynamic behavior of 4 different processes. The systems are:

1. Isothermal gas phase batch reactor
2. General 3 state CSTR
3. Styrene polymerization CSTR
4. Binary distillation column
5. CSTR Network

## B.1 Isothermal Gas Phase Reactor

This model is of a simple isothermal batch reactor in which an irreversible gas phase reaction takes place at constant volume. The reaction taking place here is

$$2A \xrightarrow{\bar{k}} B.$$

Thus, the discrete time model of this system is given by

$$C_{A,k} = \frac{C_{A,k-1}}{2\bar{k}\Delta t C_{A,k-1} + 1} \tag{B.1a}$$

$$C_{B,k} = C_{B,k-1} + \frac{\bar{k}\Delta t C_{B,k-1}^2}{2\bar{k}\Delta t C_{A,k} + 1} \tag{B.1b}$$

$$C_{A,k} \in [0, 1.5],\ C_{B,k} \in [0, 2.5], \tag{B.1c}$$

where $C_{A,k}$ and $C_{B,k}$ are the partial pressures of components A and B, $\Delta t = t_k - t_{k-1} = 0.1$, $\bar{k} = 0.16$ is the kinetic rate constant.

## B.2 General 3 State CSTR

This model describes the dynamic behavior of a non-isothermal CSTR where the exothermic reaction between thiosulfate and hydrogen peroxide takes place. For this example the states are the concentration of thiosulfate ($C_A$), the reactor temperature ($T_R$), and the cooling water temperature ($T_{cw}$). The model, as reported by Qu and Hahn [36] is given by the following index 1 system of differential-algebraic equations, where the model parameters are given in Rajaraman *et al.* [108].

$$\frac{dC_A}{dt} = \frac{F}{V}\left(C_A^{in} - C_A\right) - 2k\left(T_R\right)C_A^2 \tag{B.2a}$$

$$\frac{dT_R}{dt} = \frac{F}{V}\left(T_R^{in} - T_R\right) + \frac{2(-\Delta H_R)k\left(T_R\right)C_A^2}{\rho C_P} - \frac{UA}{V\rho C_p}\left(T_R - T_{cw}\right) \tag{B.2b}$$

$$\frac{dT_{cw}}{dt} = \frac{F_{cw}}{V_{cw}}\left(T_{cw}^{in} - T_{cw}\right) + \frac{UA}{V_{cw}\rho_{cw}C_{pcw}}\left(T_R - T_{cw}\right) \tag{B.2c}$$

$$k\left(T_R\right) = k_o \exp\left[\frac{-E_a}{RT_R}\right] \tag{B.2d}$$

$$C_A \in [0,1],\ T_R \in [200,420],\ T_{cw} \in [200,420], \tag{B.2e}$$

where the feed and cooling water flow rates, $F$ and $F_{cw}$, are considered known inputs, $C_A^{in}$ and $T_R^{in}$ are the concentration of $A$ and the temperature of the inlet stream, and the inlet temperature of the cooling water is given by $T_{cw}^{in}$. Moreover, the following are parameters: reactor volume and area ($V$ and $A$, respectively), reaction heat ($\Delta H_R$), inlet feed stream density and heat capacity ($\rho$ and $C_P$, respectively), global heat transference coefficient ($U$). Finally, similar parameters are used for the dynamics of the cooling water, and these have the subindex $cw$.

## B.3 Styrene Polymerization CSTR

The polystyrene CSTR considered here is the same as the system described in [40]. The model describes the mass and energy balances of the CSTR, and it consists of the following set of differential-algebraic equations:

$$\frac{dC_i}{dt} = \frac{Q_i C_{ii}}{V} - \left(k_i + \frac{1}{\tau}\right) C_i \tag{B.3a}$$

$$\frac{dC_s}{dt} = \frac{Q_m C_{ss} + Q_i C_{si}}{V} - \frac{C_s}{\tau} \tag{B.3b}$$

$$\frac{dC_m}{dt} = \frac{Q_m C_{mm} - Q_t C_m}{V} - k_p C_m P \tag{B.3c}$$

$$\frac{dT_R}{dt} = \gamma k_p C_m P - \frac{UA\left(T_R - T_j\right)}{\rho C_p V} + \frac{Q_t\left(T_{in} - T_R\right)}{V} \tag{B.3d}$$

$$\frac{dT_j}{dt} = \frac{UA\left(T_R - T_j\right)}{\rho_w C_{pw} V_j} + \frac{Q_w\left(T_{win} - T_j\right)}{V_j} \tag{B.3e}$$

$$\frac{d\lambda_0}{dt} = -\frac{\lambda_0}{\tau} + \left(k_{fm} C_m + k_{td} P + k_{fs} C_s\right) \alpha P + \frac{1}{2} k_{tc} P^2 \tag{B.3f}$$

$$\frac{d\lambda_1}{dt} = -\frac{\lambda_1}{\tau} + \left[\left(k_{fm} C_m + k_{td} P + k_{fs} C_s\right)\left(2\alpha - \alpha^2\right) + k_{tc} P\right] \frac{P M_m}{1-\alpha} \tag{B.3g}$$

$$\frac{d\lambda_2}{dt} = -\frac{\lambda_2}{\tau} + \left[\left(k_{fm} C_m + k_{td} P + k_{fs} C_s\right)\left(\alpha^3 - 3\alpha^2 + 4\alpha\right) + k_{tc} P\left(\alpha + 2\right)\right] \frac{P M_m^2}{(1-\alpha)^2} \tag{B.3h}$$

$$\alpha = \frac{k_p C_m}{\left(k_p + k_{fm}\right) C_m + k_{fs} C_s + k_t P} \tag{B.3i}$$

$$P = \sqrt{\frac{2 f^\star C_i k_i}{k_t}} \tag{B.3j}$$

$$k_j = k_{0,j} e^{-\frac{E_j}{RT}} \;\forall\; j \in \{i, p, t, fm, fs, td, tc\} \tag{B.3k}$$

where $C_i$, $C_s$, and $C_m$ represent the concentrations of the initiator, solvent, and monomer, respectively. $T_R$ and $T_j$ are the temperatures of the reactor and cooling water, and finally $\lambda_0$, $\lambda_1$, and $\lambda_2$ are the first moments of the molecular weight distributions. In addition, $C_{ii}$, $C_{ss}$, and $C_{mm}$ represent the concentrations of the initiator, solvent, and monomer in their respective feed streams. $C_{si}$ is the concentration of the solvent in the initiator feed stream, and $P$ represents the concentration of the live polymer. Finally, the inlet temperatures of

the reactor and cooling water are $T_{in}$ and $T_{win}$, respectively. In this case, the molecular weight averages are of particular importance since these are used to determine specific physical properties of the produced materials [89]. It is therefore common to use these states as controlled variables. Thus, the molecular weight averages by weight ($M_W$) and number ($M_N$) can be calculated with

$$M_N = \frac{\lambda_1}{\lambda_0} \tag{B.4a}$$

$$M_W = \frac{\lambda_2}{\lambda_1} \tag{B.4b}$$

.

## B.4 Distillation Column Model

Here a detailed description of the mathematical model for the binary distillation column used in the examples is given. Note that coefficients for the physical property equations (enthalpy, Antoine equation, etc.) can be found in [70, 109]. The top tray is indexed with $i = N_T$, the condenser with $i = N_T + 1$, and the reboiler with $i = 0$. Figure B.1 illustrates the flows between any two given trays in the column.

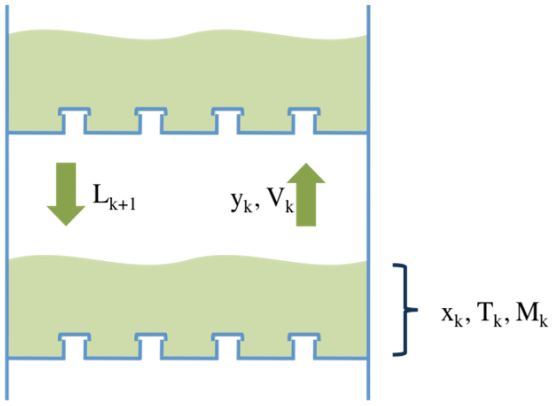



Figure B.1: Illustration of the flows between any two given trays in the distillation column.

**Mass balances:** For each stage, the overall mass balance must hold, and the rate of change of the molar hold-up ($M_i$) is given by the difference in the flows to and from the adjacent stages

$$\dot{M}_i = V_{i-1} - V_i + L_{i+1} - L_i + F_i, \tag{B.5}$$

with $V_i$ and $L_i$ being the vapor and liquid flow rates. Here the molar feed flow $F_i$ is entering the column at stage $i = 21$ (all other feeds are set to zero). For the condenser, the distillate is given by $D$, thus we have

$$\dot{M}_{N_T+1} = V_{N_T} - D - L_{N_T+1}. \tag{B.6}$$

On the other hand, for the reboiler the bottoms flow rate is $B$.

$$\dot{M}_0 = -V_0 - B + L_1 \tag{B.7}$$

Moreover, the reflux stream is modeled with

$$R = \frac{D}{L_{N_T+1}} \tag{B.8}$$

Additionally, the tray component-wise mass balance, assuming that only liquid molar holdup is of importance, is given by

$$\dot{M}_i x_i + M_i \dot{x}_i = V_{i-1} y_{i-1} - V_i y_i + L_{i+1} x_{i+1} - L_i x_i + F_i z_{f,i}, \tag{B.9}$$

where $x_i$ is the molar fraction of the volatile component in the liquid phase. Using (B.5) the above equation yields

$$M_i \dot{x}_i = V_{i-1}\left(y_{i-1} - x_i\right) + L_{i+1}\left(x_{i+1} - x_i\right) - V_i(y_i - x_i) + F_i\left(z_{f,i} - x_i\right). \tag{B.10}$$

For the reboiler and condenser we have,

$$M_0\dot{x}_0 = L_1(x_1 - x_0) - V_0(y_0 - x_0) \tag{B.11}$$

$$M_{N_T+1}\dot{x}_{N_T+1} = V_{N_T}(y_{N_T} - x_{N_T+1}), \tag{B.12}$$

respectively.

**Equilibrium and Summation Equations:** The total pressure $P_i$ on each tray is assumed to be constant, with a pressure drop $\Delta P_i$ from top to bottom.

$$P_{i-1} = P_i + \Delta P_i, \quad i = 1, \ldots, N_T + 1 \tag{B.13}$$

The pressure of the condenser is set to $P_{N_T+1} = 93.9$ KPa, while the pressure drop was set to $250$ Pa per tray for the stripping section and $190$ Pa per tray for the rectifying section. Thermodynamic equilibrium was modeled using Raoult's Law, so that the pressure of each tray is

$$P_i = P^s_{i,1}(T_i)x_i + (1 - x_i)P^s_{i,2}(T_i), \tag{B.14}$$

where the vapor pressures are computed using Antoine's Equation

$$P^s_{i,j} = \exp\left(A_j - \frac{B_j}{T_i + C_j}\right). \tag{B.15}$$

Note that the tray temperatures are implicitly defined by Equation (B.14). However, to reduce the index of the model, we need the time derivative of the temperature, which is determined by applying the implicit function theorem to Equation (B.14) to yield

$$\dot{T}_i = -\frac{\left(P^s_{i,1} - P^s_{i,2}\right)\dot{x}_i}{\frac{\partial P^s_{i,1}}{\partial T_i}x_i + \frac{\partial P^s_{i,1}}{\partial T_i}(1 - x_i)} \tag{B.16}$$

The partial derivatives of the vapor pressures can be obtained from the Antoine Equations above.

Moreover, to account for non-equilibrium behavior in the mixtures tray efficiencies $\alpha_i$ are considered in the summation equations. Thus for each tray we have

$$y_i = \alpha_i x_i \frac{P^s_{i,1}}{P_i} + (1-\alpha_i) y_{i-1}, \tag{B.17}$$

and for $i = 0$ the term simplifies to

$$y_0 = x_0 \frac{P^s_{0,1}}{P_0}. \tag{B.18}$$

**Energy balances:** The liquid enthalpy is determined with

$$h^L_i(x_i, T_i) = x_i \bar{h}^L_{i,1}(T_i) + (1-x_i)\bar{h}^L_{i,2}(T_i), \tag{B.19}$$

and the vapor enthalpy is given by

$$h^V_i(y_i, T_i, P_i) = y_i \bar{h}^V_{i,1}(T_i, P_i) + (1-y_i)\bar{h}^V_{i,2}(T_i, P_i). \tag{B.20}$$

In addition, the pure liquid and vapor enthalpies required in the above equations are defined by the following terms

$$\begin{aligned} \bar{h}^L_{i,j}(T_i) &= C\left[g_{j,1}\left(T_i - T_{\text{ref}}, j\right) + g_{j,2}\left(T_i - T_{\text{ref},j}\right)^2 + g_{j,3}\left(T_i - T_{\text{ref},j}\right)^3\right] \\ \bar{h}^V_{i,j}(T_i, P_i) &= \bar{h}^L_{i,j}(T_i) + \hat{\alpha} R T^c_j \sqrt{1 - \frac{P_i}{P^c_j}\left(\frac{T_i}{T^c_j}\right)^{-3}} \\ \hat{\alpha} &= \left[\hat{a} - \hat{b}\frac{T_i}{T^c_j} + \hat{c}\left(\frac{T_i}{T^c_j}\right)^7 + \Omega_j\left(\hat{d} - \hat{e}\frac{T_i}{T^c_j} + \hat{f}\left(\frac{T_i}{T^c_j}\right)^7\right)\right]. \end{aligned} \tag{B.21}$$

The energy balance for the trays are given by

$$\begin{aligned} \dot{M}_i h^L_i + M_i\left(\dot{x}_i \frac{\partial h^L_i}{\partial x_i} + \dot{T}_i \frac{\partial h^L_i}{\partial T_i}\right) =& V_{i-1} h^V_{i-1} - V_i h^V_i + \\ & L_{i+1} h^L_{i+1} - L_i h^L_i + F_i h^L\left(z_{f,i}, T_{f,i}, P_{f,i}\right). \end{aligned} \tag{B.22}$$

For the reboiler the added heat duty is $Q_R$, and a loss term ($Q_{\text{loss}}$) is also considered.

$$\dot{M}_0 h_0^L + M_0 \left( \frac{\partial h_0^L}{\partial x_0} \dot{x}_0 + \frac{\partial h_0^L}{\partial T_0} \dot{T}_0 \right) = Q_R - Q_{\text{loss}} - V_0 h_0^V + L_1 h_1^L - B h_0^L \tag{B.23}$$

The energy balance for the condenser is

$$\dot{M}_{N_T+1} h_{N_T+1}^L + M_{N_T+1} \left( \frac{\partial h_{N_T+1}^L}{\partial x_{N_T+1}} \dot{x}_{N_T+1} + \frac{\partial h_{N_T+1}^L}{\partial T_{N_T+1}} \dot{T}_{N_T+1} \right) = V_{N_T} \left( h_{N_T}^V - h_{N_T+1}^L \right) - Q_C, \tag{B.24}$$

where the condenser heat duty is $Q_c$. The partial derivatives of the previous equations can be derived from Equations (B.19) and (B.21). Note that substituting equations (B.5)-(B.7), (B.10)-(B.12), and (B.16) into the energy balance equations yields a set of purely algebraic equations for the vapor flow rates and the condenser heat duty.

**Hydrodynamics:** To determine the liquid flow rates from each tray we use Francis weir formula. For that first we need to determine the liquid volume holdup of each tray. These are related the molar volume $V_i^m(x_i, T_i)$ by the following equation

$$n_i^v = M_i V_i^m(x_i, T_i), \tag{B.25}$$

and the molar volumes are computed as the sum of the molar volumes of the pure components weighted by their molar fraction

$$V_i^m(x_i, T_i) = x_i \bar{V}_{i,1}^m(T_i) + (1 - x_i) \bar{V}_{i,2}^m(T_i), \tag{B.26}$$

where $\bar{V}_{m,1}(T_i)$ and $\bar{V}_{m,2}(T_i)$ are the temperature dependent molar volumes of pure Methanol and n-Propanol. These are computed using

$$\bar{V}_{i,j}^m(T_i) = \frac{1}{\bar{a}_j} \bar{b}_j^{1+(1-T_i/\bar{c}_j)^{\bar{d}_j}}. \tag{B.27}$$

Finally, the liquid flow rates are calculated by the so-called Francis weir equation given by

$$L_i V_i^m (x_i, T_i) = W_i \left( n_i^v - n_i^{v,\text{ref}} \right)^{\frac{3}{2}}, \tag{B.28}$$

where $W_i$ is the Francis weir constant.

## B.5 CSTR Network

This model consists of the mass and energy balance of each reactor in a network where the reaction that is occurring is given by $A \longrightarrow B$. The network is depicted in Figure B.2, and the model is given by

$$\frac{dC_i}{dt} = \frac{C_{i-1} - C_i}{\theta_i} - k_o C_i e^{-\frac{E_R}{RT_i}} \tag{B.29a}$$

$$\frac{dT_i}{dt} = \frac{T_{i-1} - T_i}{\theta_i} - \frac{\Delta H_{\text{reac}}}{\rho C_p} k_o C_i e^{-\frac{E_R}{RT_i}} + \frac{2U}{r \rho C_p} (T_{cw} - T_i) \tag{B.29b}$$

where $C_i$ and $T_i$ are the concentration and temperature for the *i*-th reactor, respectively. Also, $C_0$ and $T_0$ are the feed concentration and temperature. Other model parameters are taken from [110].

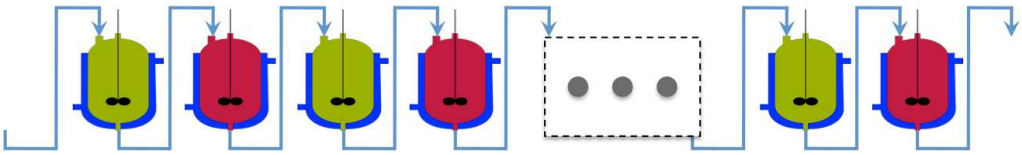


Figure B.2: Illustration of a network of CSTRs.

# Appendix C

# Auxiliary Theorems

In this appendix some general theorems and proofs are collected. These are used throughout the thesis, and for convenience they are shown here. The theorems collected here are related to Probability Theory.

**Theorem C.1** (Probability transfer function theorem [8].)**:** *Let $x$, $y$ be random n-vectors with $y = f(x)$. Suppose $f^{-1}$ exists and that both $f$ and $f^{-1}$ are continuously differentiable. Then*

$$p_y(y) = p_x\left(f^{-1}(y)\right)\left\|\frac{\partial f^{-1}(y)}{\partial y}\right\|, \tag{C.1}$$

*where $\left\|\partial f^{-1}(y)/\partial y\right\| > 0$ is the absolute value of the Jacobian determinant.*

**Proof:** The proof follows directly from well known integral transformations and Theorem 2.7 in Jazwinski [8]. □

# Appendix D

# Expressions for $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$

In the objective function of NLP (6.10) the initial condition penalty term includes expressions that use $\mathcal{Y}$, $\mathcal{O}$, and $\mathcal{W}$. These can be obtained by assuming that $p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)$ is Gaussian, and from the properties of stochastic difference equations. Moreover, we use the linearized versions of the model to generate the expression that follow. Also, we assume that for the linear model the noise variables are additive. Thus, linearized model is given by

$$z_l = A_{l-1} z_{l-1} + \omega_{l-1} \tag{D.1a}$$

$$y_l = H_l z_l + \nu_l, \tag{D.1b}$$

and since $p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right)$ is Gaussian, we also have that

$$p\left(Y_{k-N}^{k-1}\middle|\, z_{k-N}\right) \sim \mathcal{N}\left(\bar{\mathcal{Y}}, \mathcal{W}\right). \tag{D.2}$$

The parameters of this Gaussian distribution are generated as follows. The mean ($\bar{\mathcal{Y}}$) is obtained from

$$\bar{\mathcal{Y}} = \begin{bmatrix} \bar{y}_{k-N} \\ \bar{y}_{k-N+1} \\ \bar{y}_{k-N+2} \\ \vdots \\ \bar{y}_{k-1} \end{bmatrix} = \begin{bmatrix} H_{k-N} \\ H_{k-N+1} A_{k-N} \\ H_{k-N+2} A_{k-N+1} A_{k-N} \\ \vdots \\ H_{k-1} A_{k-2} A_{k-3} \cdots A_{k-N} \end{bmatrix} z_{k-N} = \mathcal{O} \bar{z}_{k-N}, \tag{D.3}$$

where we note that $\mathcal{O}$ is the observability matrix. On the other hand, the covariance matrix ($\mathcal{W}$) is given by $\mathbb{E}\left[\left(\mathcal{Y}-\bar{\mathcal{Y}}\right)\left(\mathcal{Y}-\bar{\mathcal{Y}}\right)^T\right]$. For this, we now define the following equation

$$\mathcal{Y} = \mathcal{O} z_{k-N} + \mathcal{M}\omega + \nu, \tag{D.4}$$

and, also, the following definition holds

$$\mathcal{M} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ H_{k-N+1} & 0 & 0 & 0 \\ H_{k-N+2}A_{k-N+1} & H_{k-N+2} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_{k-1}A_{k-2}A_{k-3}\cdots A_{k-N+1} & H_{k-1}A_{k-2}A_{k-3}\cdots A_{k-N+2} & \dots & H_{k-1} \end{bmatrix}, \tag{D.5}$$

where, the vectors in Eq. (D.4) are defined by

$$\omega = \begin{bmatrix} \omega_{k-N} \\ \omega_{k-N+1} \\ \omega_{k-N+2} \\ \vdots \\ \omega_{k-1} \end{bmatrix} \quad \text{and} \quad \nu = \begin{bmatrix} \nu_{k-N} \\ \nu_{k-N+1} \\ \nu_{k-N+2} \\ \vdots \\ \nu_{k-1} \end{bmatrix}.$$

Finally, taking the expected value of $\mathbb{E}\left[\left(\mathcal{Y}-\bar{\mathcal{Y}}\right)\left(\mathcal{Y}-\bar{\mathcal{Y}}\right)^T\right]$ and using Assumptions 3.1(*ii*–*iii*) we obtain the expression for $\mathcal{W}$:

$$\mathcal{W} = \mathcal{O}\bar{\Pi}_{k-N|k-1}\mathcal{O}^T + \mathcal{M}\mathcal{Q}\mathcal{M}^T + \mathcal{R}, \tag{D.6}$$

where $\mathcal{Q} = \text{diag}\left(Q_{k-N}, \dots, Q_{k-1}\right)$ and $\mathcal{R} = \text{diag}\left(R_{k-N}, \dots, R_{k-1}\right)$.