NTNU – Trondheim
Norwegian University of
Science and Technology

Techniques for efficient covariance propagation in the Extended/Unscented Kalman Filter based on model reduction

Frode Rønneberg

Master of Science in Engineering Cybernetics (2)
Submission date: May 2013
Supervisor: Lars Imsland, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of Science and Technology

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

MSc THESIS DESCRIPTION SHEET

Name: Frode Rønneberg
Department: Engineering Cybernetics
Thesis title: Techniques for efficient covariance propagation in the Extended/Unscented Kalman Filter based on model reduction

Background
Even though the (recursive) EKF/UKF are considered efficient algorithms for state estimation, it can still be computationally expensive for medium to large systems. The bulk of the complexity stems from the propagation of the state covariance. In this task, the student will consider using techniques from model reduction to make the covariance propagation more efficient, while the full model still is used for state propagation.

Work description

1. Literature study on EKF and UKF, and model reduction for nonlinear systems.
2. What is the complexity of covariance propagation? Perform a literature study on existing efficient filtering methods.
3. Suggest methods for efficient covariance propagation in the EKF and UKF based on a reduced model, while the full model is used for state propagation.
4. Implement for selected test systems.
5. (Compare with other methods for efficient estimation.)
6. Evaluate and discuss.

Start date: January 14, 2013 **Due date:** June 10, 2013

Supervisor: Lars Imsland
Co-advisor(s):

Sammendrag

Fysiske prosesser gir ingeniører og forskere utfordrende oppgaver som modellering, simulering og regulering. Ett av de viktigste stegene i disse prosessene er muligheten til å beregne fremtidig oppførsel, som ofte kan være en beregningsmessig kompleks og krevende oppgave. For høy kompleksitet i prosessene kan føre til at nødvendig informasjon beregnes for sent, gi avbrudd i produksjonssystemer, tap av kostnader, eller det som verre er, tap av liv. Dermed er det å finne mer effektive måter å estimere oppførselen til fysiske prosesser av spesiell interesse.

Denne rapporten presenterer en ny fremgangsmåte for tilstandsestimering ved å redusere beregningene i Extended Kalman Filter (EKF) og Unscented Kalman Filter (UKF) ved hjelp av teknikker fra modellreduksjon. Ved å redusere de mest krevende operasjonene, gitt av kovarianspropageringen i EKF og kalkuleringen av sigmapunktene i UKF, samtidig som den fulle modellen beholdes til tilstandsestimering og transformasjon av sigmapunkt, skal kompleksiteten reduseres.

En en- og to-dimensjonal varmeoverføringsmodell med henholdsvis 400 og 1452 tilstander blir presentert. Den reduserte modellen finnes ved hjelp av Galerkin-projeksjon basert på en redusert basis fra Proper Orthogonal Decomposition (POD), eller balansert trunkering og empiriske Gramians. Algoritmene er så simulert for flere mulig tilfeller for feil eller forstyrrelser som kan forekomme, for å se hvordan dette innvirker på estimeringsresultatet.

Tidsbruken i den nye fremgangsmåten viser seg å være lik de eksisterende fremgangsmåtene for redusert tilstandsestimering, selv om den reduserte modellen kun brukes i de mest krevende operasjonene. Den viser også en betydelig forbedring over de originale algoritmene, gitt at underrommet i den reduserte modellen ikke overskrider en viss prosent av det fulle rommet. Den avledede EKF viser et stort potensiale for et vidt spenn av underrom, forutenom i tilfellet med indusert modellfeil. Den avledede UKF har generelt vist et stort potensiale for underrom større enn en sekstendedel av det totale rommet, og for alle underrom, uavhengig av størrelsen på feilen, i tilfellet med indusert modellfeil.

Den nye fremgangsmåten virker lovende med hensyn til effektivisering av EKF og UKF ved hjelp av modellreduksjon, uten at man mister for mye informasjon i prosessen.

Abstract

Physical processes gives engineers and researchers challenging task, such as modeling, simulation and control. One of the most important steps in these procedures are the ability to predict the systems behavior, which due to the size and complexity of the models often are computationally expensive. A too high computational cost may lead to delays in necessary information, interruptions in production systems, loss of finances, or even worse, loss of lives. Thus, finding more efficient way of predicting the behavior of physical processes are of particular interest.

This thesis presents a new approach of improving the efficiency in the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) through the use of techniques from model reduction. By reducing the main bulk of complexity, given by the covariance propagation in the EKF and the calculation of the sigma points in the UKF, while the full model is used for state propagation and the unscented transformation, respectively, the computational effort is reduced.

A one and two dimensional heat conduction model with 400 and 1452 states are introduced. The reduced order model is obtained by applying the Galerkin projection based on a Proper Orthogonal Decomposition (POD) reduced-order basis or balanced truncation by empirical Gramians. The physical processes are simulated for several possible scenarios of errors one may encounter, to see how the estimation result is affected.

The time complexity in the new approach is shown to be similar to the existing reduced approaches of state estimation, while only using the reduced model in the bulk of complexity. It also show a significantly improvement over the original algorithms, given that the subspace configuration in the reduced model not exceeds a certain percentage of the full space. The derived EKF show great promise for a wide range of subspace configurations, except for in the case of induced model errors. The derived UKF has in general shown great promise for subspace configurations larger than one sixteenth of the full space, and for all subspace configurations in the case of model errors, independent of the magnitude of the error.

The new approach show great promise for improving the efficiency in the EKF and UKF through model reduction, without losing too much information in the process.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in fulfillment of the requirements for the degree Master of Science (MSc). The research presented in this thesis has been carried out in the period of January 2013 through May 2013, at the Department of Engineering Cybernetics, NTNU, under the guidance of Professor Lars Imsland.

I would like to thank Professor Imsland for allowing me to pursue his idea leading up to this thesis, for his reflections and constructive feedback on my work, and guidance throughout the research.

Frode Rønneberg
Trondheim, 2013

Contents

List of Figures

List of Tables

Acronyms

CFD Computational Fluid Dynamics.

DFEKF Derivative-Free Extended Kalman Filter.

DOF Degrees of Freedom.

EKF Extended Kalman Filter.

EnKF Ensemble Kalman Filter.

FLOP Floating point operations.

KF Kalman Filter.

MPD Managed Pressure Drilling.

NTNU Norwegian University of Science and Technology.

PDE Partial Differential Equation.

POD Proper Orthogonal Decomposition.

SDK Software Development Kit.

SSUKF Spherical-Simplex Unscented Kalman Filter.

SVD Singular Value Decomposition.

UKF Unscented Kalman Filter.

1 Introduction

1.1 Applications of state estimation

Weather predictions and control theory are two of many typical applications where state estimation is found necessary. Whether it is predicting the weather for the next day or week, or estimating the values of non-measurable states, the need for state estimation is current.

Estimation is based on previous and current measurements, allowing the implemented desired algorithm to predict the next value, rate or decision to be made. This is a continuous process, where the next estimates are dependent on the current (new) measurements. However, the actual desired information may not always be available for measurements, making the current available measurements the only source of information. It is therefore critical that this information is as accurate and extensive as possible.

The availability of estimating future states makes it possible to increase the performance of some systems. For example, in the oil industry during drilling operations, the downhole pressure is critical to control. In conventional drilling the drill fluid flows through the drill string and drill bit, allowing the cuttings to be transported out of the wellbore through the annulus. However, to avoid fracturing or the collapse of the borehole during drilling operations, the annulus downhole pressure needs to be kept within a given drilling window. Thus, conventional drilling is replaced by Managed Pressure Drilling (MPD), which controls this pressure[1]. The main reason for pressure control is to keep the pressure within the boundaries of the pressure window, such that large financial

[1]For more on MPD, see [Kaasa et al., 2012].

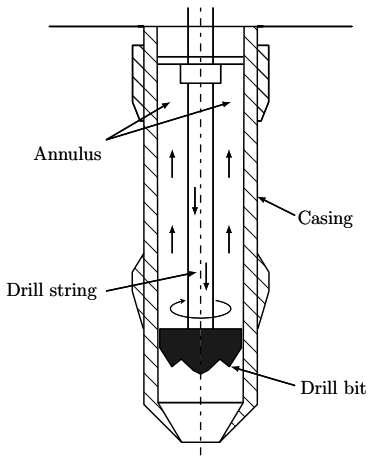

Figure 1.1: Simplified schematic of a typical borehole. Drill mud flows through the drill string and drill bit, transporting the cuttings out of the wellbore through the annulus.

losses, environmental damages or personnel injuries are avoided. Due to the length of the wells, and the need for real-time data during drilling operations, the downhole pressure is not available for direct measurement, and would in addition be affected by disturbances. The need for a satisfying fit-for-purpose model for estimating the states is therefore present. Figure 1.1 illustrates a typical borehole placed down in the seabed. This is one of many applications where state estimation is a crucial part of the overall procedure and performance, where direct measurements are not available.

1.2 Derivation of more efficiently state estimation algorithms

Considering that an estimation scheme approximates the states in a system or process, it is obvious that the computational complexity increase with the size and complexity of the system. Thus, problems may arise when the number of states makes the computational complexity too extensive to calculate within a given time limit, i.e., in real-time operations as mentioned earlier. Due to limited resources in terms of hardware, simpler and more efficient models are

required in order to meet the requirements of some processes. Some approaches for the derivation of simpler models are

- **Finding similar, less complex, models**
  Using models that are less complex, but still is similar enough to use in estimation of the states, so-called fit-for-purpose models, can reduce the computationally complexity. The disadvantage of this approach is that finding a less complex, yet similar to the true model, can be difficult. Finding an appropriate full model by using physical laws can be hard enough.

- **Using model reduction**
  By using existing techniques for model reduction, the existing full model can easily be reduced to a smaller model, and further used in the calculation for estimating the states. The advantage of this approach is that there exists a lot of techniques for model reduction, but it may, however, lead to some loss in information when projecting and reconstructing the full model.

The main difference between the two approaches is that the fit-for-purpose model requires prior knowledge about the process, while using model reduction can generally be used on any existing model (to some degree).

1.3 Problem formulation

The main advantage of reducing the complexity of the state estimation techniques is that the overall computational cost is reduced. However, the estimation result, when reducing the complexity, may be affected such that the error between the true and estimated states should be kept at a minimum. Achieving both of these criteria are highly beneficial, as nature consists of large nonlinear and complex systems, making the process of state estimation a very computationally expensive task.

In state estimation theory, the most well-known state estimation algorithm is the Kalman Filter (KF), developed during a research supported by the U.S. Air Force Office of Scientific Research by [Kalman, 1960]. This algorithm exists both in a linear and nonlinear edition, making it suitable for most of the processes in nature. Similar approaches such as the Unscented Kalman Filter (UKF), [Julier et al., 2004] and the Ensemble Kalman Filter (EnKF), [Evensen, 1994] has also gained popularity since they were developed. Several less computational

expensive variants of these also exists [Julier, 2003, Julier and Uhlmann, 2002, Quine, 2006].

In terms of model reduction, reduced order models can be derived by various methods. Some are for example truncating states, either from a unbalanced or balanced representation [Skogestad and Postlethwaite, 2005], or Proper Orthogonal Decomposition (POD), also known as Karhunen-Loève decomposition [Lall et al., 2002], where each of the techniques have desirable properties depending on the system which is to be reduced. Using Galerkin projection [Lall et al., 2002], allows the projection of the full model to a reduced model, or reconstructing the full model from the reduced model, through similarity transforms. Thus, the technique makes it applicable to only use the reduced model in some parts of the state estimation. Specifically, the reduced model can be used in the covariance propagation in the Extended Kalman Filter (EKF) or calculation of the sigma points in the UKF, while using the full model for propagating the states or transforming the sigma points. This should lead to a improvement in the computational complexity for estimating the states of a given system, while hopefully giving a satisfying estimation result. This is the main motivation of this Masters study.

Given a higher-order nonlinear process, find more efficient ways for state estimation through the use of EKF or UKF, which meets the following characteristics:

- the reduced model should have a substantially lower dimension than the full (original) model,
- the reduced model should keep all relevant dynamics,
- the new approach should retain a sufficient accuracy level,
- the new approach should be more efficient than the original model, and
- the new approach should have benefits over existing reduced approaches.

1.4 Previous work

In the literature regarding model reduction and nonlinear state estimation, it seems that most of the studies are based on using the reduced model for the entire estimation process. On the subject of reducing the computational effort, which in amongst other fields is important for image processing, [Burl, 1993] have shown how combining the EKF in estimation of a image reconstruction and

velocity of objects, gives a large reduction in number of multiplication operations and vastly time savings on the inversion of matrices. The results of [Burl, 1993] also showed that given a known velocity, the image reconstruction was found to be close to the optimal result. A similar study on reducing computational cost by [Farrell and Ioannou, 2001], conducted on a forecast error system of a mid-latitude storm model, showed great approximation and performance using a reduced model. It was shown that the 400 Degrees of Freedom (DOF) model, where the dynamically relevant dimension is much smaller then the full state dimension, could be reduced to a 60 DOF model through balanced truncation, while giving a satisfying accurate approximation.

Another interesting study for reducing the KF is a study on reducing the full system model without model reduction, which have been conducted with good results. According to [Simon, 2007], given that the transition matrix has some given form and is stable, it is shown that the computational effort was substantially decreased in contrast to using normal model reduction techniques. However, this specific method led to an increase in the estimation error and could not guarantee convergence and stability in the computation of the Kalman gain.

In several studies regarding reduction of the computational cost in the EKF, leading up to the paper by [Evensen, 1994], Evensen showed that solving the Kolmogorov's equation using Monte Carlo methods in forecast error statistics, is an alternative to solving the approximate error covariance equation. This technique, known as the EnKF, showed great performance compared to the EKF as there are no closure problem, with only a fraction of the computational load for reasonable accuracy.

1.5 Thesis outline

The thesis is organized as follows.

Chapter 2 presents the relevant theory with respect to state estimation and model reduction. The mathematical outline and algorithms for the most common state estimation techniques are presented, alongside some more efficient derived variants. The most common techniques for model reduction will be presented, both for unbalanced and balanced model reduction, in addition to an overview of the implementations related to this thesis.

Chapter 3 discuss the connection and use of state estimation algorithms in combination with the techniques from model reduction to reduce the bulk of

complexity in the algorithms. The new, derived, approach is presented in full for both the EKF and UKF.

Chapter 4 introduce the Partial Differential Equation (PDE) models used for illustration purposes in this thesis. The models are one and two dimensional heat conduction models, respectively, discretized by the Finite Volume Method, which is one of the most popular discretization techniques in Computational Fluid Dynamics (CFD) software. This chapter also discuss the strategy and approach of finding a reduced order basis corresponding to the full models through the use of POD.

In Chapter 5, the hardware and software setup are presented. This involves the computer specifications, the experimental setup for the simulation, and the conditions imposed on the testing environment to ensure accurate and valid data.

Chapter 6 represent and discuss the results from the simulation of the models. The algorithms are mainly obtained with POD as the model reduction technique, but balanced truncation by empirical Gramians is also considered. The simulations are conducted for different scenarios, inducing errors such as Gaussian white noise, and systematic and model errors. The data of the time complexity, estimation results and problems encountered with the simulations are discussed.

Finally, in Chapter 7, conclusions and recommendations for future research are given.

2 Background material

This chapter introduce the tools and techniques that will be used in the subsequent chapters to develop reduced-order models and perform state estimation. The tools and techniques are either described in full, or summarized by the outline of the algorithm/technique.

2.1 State estimation

State estimation is the technique of approximating or predicting the states in a process based on measurements from the process. In engineering, the information gathered from state estimation may be of interest of their own, or even necessary for example in control purposes. The problem statement, alongside different techniques of state estimation are presented.

2.1.1 Problem statement

The problem statement of state estimation is given as follows. Assume that a real system can be represented by the differential equation

$$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t)\right) + \mathbf{w}(t), \tag{2.1a}$$
$$\mathbf{y}(t) = h\left(\mathbf{x}(t), \mathbf{u}(t)\right) + \mathbf{v}(t), \tag{2.1b}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^p$ is the p inputs to the system, $\mathbf{w} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ are the process and measurement zero-mean Gaussian white noise vectors added to the system, and $\mathbf{y} \in \mathbb{R}^m$ the m measurements of the system. The objective is to estimate the real states,

$$\dot{\hat{\mathbf{x}}}(t) = f\left(\hat{\mathbf{x}}(t), \mathbf{u}(t)\right),$$
$$\hat{\mathbf{y}}(t) = h\left(\hat{\mathbf{x}}(t), \mathbf{u}(t)\right),$$

based on measurements from the real system, such that the estimated states are similar or equal to the true states, i.e., $\hat{\mathbf{x}} \simeq \mathbf{x}$.

Since this thesis is concerned with discrete time systems, the discrete-time versions of the algorithms are presented.

2.1.2 Linear systems

Discrete linear time-invariant systems, or nonlinear systems that are linearized within a small region, are generally described as

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \tag{2.3a}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{v}_k, \tag{2.3b}$$

where the set of equations represents the internal description of the system. Linear systems have the advantage that they are much simpler than their nonlinear counterpart, making one of the most known state estimation techniques apply to linear systems.

The Kalman Filter

[Kalman, 1960] presents a method for solving the Wiener problem and show how estimates of the discretized state vector $\mathbf{x}_k$, based on the knowledge of the system and the measurement vector $\mathbf{y}_k$, at time t_k, can be obtained. This section will present the Kalman Filters recursive equations, which makes up the procedure of estimating the states.

Suppose that the objective is to estimate the linear state-space system (2.3). Given that the data up to and including time t_k can be obtained, an *a posteriori* estimate of the state can be formed. The a posteriori estimate is the estimate after the measurement $\mathbf{y}_k$ is taken into account and is given as

$$\hat{\mathbf{x}}_k^+ = \mathrm{E}\left[\mathbf{x}_k | \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\right],$$

where the operator $\mathrm{E}[\cdot]$ denotes the expected value of $(\cdot)$. Based on the measurements and previous states, the goal is to approximate the states such that the error between the estimated and true states are minimized, i.e.,

$$\mathbf{e}_k^- \triangleq \mathbf{x}_k - \hat{\mathbf{x}}_k^-,$$

with the associated covariance matrix given as

$$\begin{aligned}\mathbf{P}_k^- &= \mathrm{E}\left[\mathbf{e}_k^-(\mathbf{e}_k^-)^{\mathrm{T}}\right] \\ &= \mathrm{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^{\mathrm{T}}\right].\end{aligned} \tag{2.4}$$

The covariance matrices related to the process and measurement noise vectors in (2.3) are given as

$$\mathrm{E}\left[\mathbf{w}_k\mathbf{w}_i^{\mathrm{T}}\right] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases}, \tag{2.5a}$$

$$\mathrm{E}\left[\mathbf{v}_k\mathbf{v}_i^{\mathrm{T}}\right] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases}, \tag{2.5b}$$

$$\mathrm{E}\left[\mathbf{w}_k\mathbf{v}_i^{\mathrm{T}}\right] = 0, \qquad \forall\, i, k, \tag{2.5c}$$

where it is given that the process and measurement noise vectors are uncorrelated. Given the terms above, the approach of the algorithm is outlined next.

Approach The initialization of the estimation process starts with the a posteriori estimate of the initial state vector $\mathbf{x}_0$. Since no measurements are available initially, it is reasonable to set this as the expected value of the initial states, given as

$$\hat{\mathbf{x}}_0^+ = \mathrm{E}(\mathbf{x}_0). \tag{2.6}$$

Further, as the mean of the state propagates with time,

$$\bar{\mathbf{x}}_k = \mathbf{A}_{k-1}\bar{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1},$$

the a priori estimate, known as the time update equation for the estimate $\hat{\mathbf{x}}$, is given in general form as

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{B}_{k-1}\mathbf{u}_{k-1}.$$

The covariance of the state estimation error, denoted by $\mathbf{P}$, is initialized by computing the a posteriori estimate. If one happen to know the initial states (perfectly), one can set $\mathbf{P}_0^+ = 0$, or $\mathbf{P}_0^+ = \infty\mathbf{I}$ if no prior knowledge of the initial states are known. In general, the covariance of the state estimation error is computed by

$$\mathbf{P}_k^+ = \mathrm{E}\left[(\mathbf{x} - \hat{\mathbf{x}}_0^+)(\mathbf{x} - \hat{\mathbf{x}}_0^+)^{\mathrm{T}}\right]. \tag{2.7}$$

Finally, the measurement-update equations for the gain, state estimate and the covariance of the state estimation error are necessary to introduce. These are

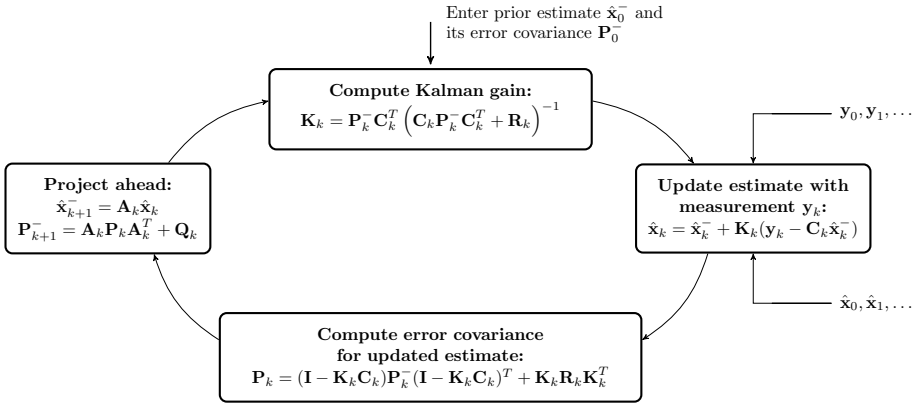

Figure 2.1: The illustration of the Kalman Filter iteration loop (adapted from [Brown and Hwang, 1996]) show the iterative process in the algorithm. Given an initial a priori state estimate $\hat{\mathbf{x}}_0^-$ and belonging error covariance matrix $\mathbf{P}_0^-$, the algorithm computes the Kalman gain and updates the estimate based on the current measurements. The error covariance is then calculated before the a posteriori estimate for the states and the covariance matrix are projected ahead.

given by

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^{\mathrm{T}} \left(\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^{\mathrm{T}} + \mathbf{R}_k\right)^{-1} \tag{2.8a}$$

$$= \mathbf{P}_k^+ \mathbf{C}_k^{\mathrm{T}} \mathbf{R}_k^{-1}, \tag{2.8b}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-\right) \tag{2.8c}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{C}_k\right) \mathbf{P}_k^- \left(\mathbf{I} - \mathbf{K}_k \mathbf{C}_k\right)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}} \tag{2.8d}$$

$$= \left[\left(\mathbf{P}_k^-\right)^{-1} + \mathbf{C}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{C}_k\right]^{-1} \tag{2.8e}$$

$$= \left(\mathbf{I} - \mathbf{K}_k \mathbf{C}_k\right) \mathbf{P}_k^-, \tag{2.8f}$$

where the matrix $\mathbf{K}_k$ is called the Kalman gain, which minimizes the mean-square estimation error. Note that there are three expressions for the covariance matrix of the state estimation error. This is due to that the first equation (2.8d) is valid for any gain, while the others are only valid for the optimal gain condition. To summarize the steps of the algorithm, a graphical illustration of the iteration process is given in Figure 2.1.

2.1.3 Nonlinear systems

Processes are rarely (as good as never) linear, and the linearization of nonlinear systems are not always possible, making state estimation techniques for non-linear systems necessary. There exists several approaches for nonlinear state estimation with different properties, where some of the most known will be presented in the subsequent sections.

The Extended Kalman Filter

The EKF is an nonlinear extension to the original Kalman Filter where the main difference is the linearization of the nonlinear system around the state estimate, making the outline of the algorithms very similar. Thus, when using the EKF to estimate a linear system, the approach of the EKF will be identical to the linear Kalman Filter. The approach of the algorithm is as follows.

Approach As shown in [Simon, 2006b], the discretized nonlinear process of (2.1) is given by

$$\mathbf{x}_k = f_{k-1}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) + \mathbf{w}_{k-1}, \tag{2.9a}$$

$$\mathbf{y}_k = h_k\left(\mathbf{x}_k, \mathbf{u}_k\right) + \mathbf{v}_k, \tag{2.9b}$$

where the covariance matrices for the vectors $\mathbf{v}_k, \mathbf{w}_k$ are the same as in the linear case, given by Equation (2.5). In addition, the initialization of the a posteriori estimate and process covariance is equal to the linear case as given in Equation (2.6) and (2.7), respectively. The necessary linearizations for calculating the time update of the a priori state estimates and the estimation error covariance

matrix, are obtained by computing the Jacobians

$$\mathbf{F}_{k-1} = \left.\frac{\partial f_{k-1}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}^{+}}, \tag{2.10a}$$

$$\mathbf{L}_{k-1} = \left.\frac{\partial f_{k-1}}{\partial \mathbf{w}}\right|_{\hat{\mathbf{x}}_{k-1}^{+}}, \tag{2.10b}$$

which makes it possible to calculate the covariance update and states

$$\mathbf{P}_k^{-} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^{+}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\mathrm{T}}, \tag{2.11a}$$

$$\hat{\mathbf{x}}_k^{-} = f_{k-1}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right). \tag{2.11b}$$

The necessary Jacobians for calculating the Kalman gain, state estimates and a posteriori estimation-error covariance matrix are given by

$$\mathbf{H}_k = \left.\frac{\partial h_k}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^{-}}, \tag{2.12a}$$

$$\mathbf{M}_k = \left.\frac{\partial h_k}{\partial \mathbf{v}}\right|_{\hat{\mathbf{x}}_k^{-}}, \tag{2.12b}$$

$$\tag{2.12c}$$

leading up to the final calculation. The prediction step, given as

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^{\mathrm{T}}\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{\mathrm{T}} + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^{\mathrm{T}}\right)^{-1}, \tag{2.13a}$$

$$\hat{\mathbf{x}}_k^{+} = \hat{\mathbf{x}}_k^{-} + \mathbf{K}_k\left[\mathbf{y}_k - h_k\left(\hat{\mathbf{x}}_k^{-}, \mathbf{u}_k\right)\right], \tag{2.13b}$$

$$\mathbf{P}_k^{+} = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^{-}. \tag{2.13c}$$

which completes the presentation of the EKF with the necessary calculations for estimating a nonlinear system. It should be noted that [Simon, 2006b] use a square-root formulation of the EKF, which gives better numerical stability and precision of the covariance matrix.

Ensemble Kalman Filter

The EnKF by [Evensen, 1994] is a state estimation algorithm based on forecasting the error statistics using Monte Carlo methods. By creating an ensemble of state estimates, statistical samples used in forecasting and analysis, this can be

used to estimate a classical Kalman gain and predict error statistics. The EnKF is considered a highly efficient nonlinear state estimator, and will in this thesis be used for comparison reasons. The EnKF has the advantages that it is suited for systems with a large number of states, and that the covariance matrix is replaced by a sample covariance matrix computed from the ensemble, i.e., using statistics for obtaining the covariance matrix.

EnKF Algorithm The outline of the EnKF presented in this thesis is based on the article by [Gillijns et al., 2006], as given below in Algorithm 1. The reader is referred to [Evensen, 1994, Burgers and et al., 1998] for a more detailed mathematical description of the theory and algorithm.

Algorithm 1: Ensemble Kalman Filter

Input : A priori ensemble analysis estimate $\mathbf{x}_0^{\mathrm{a}_i},\ i = 1, \ldots, q$.

Output: Analysis ensemble estimate $\mathbf{x}_k^{\mathrm{a}_i},\ i = 1, \ldots, q$, analysis mean $\bar{\mathbf{x}}_k^{\mathrm{a}_i},\ i = 1, \ldots, q$

`/* Forecast step */`

for $k \leftarrow 0$ **to** n **do**

$\mathbf{x}_{k+1}^{\mathrm{f}_i} = f\left(\mathbf{x}_k^{\mathrm{a}_i}, \mathbf{u}_k\right) + \boldsymbol{w}_k^i,\ i = 1, \ldots, q;$

$\bar{\mathbf{x}}_{k+1}^{\mathrm{f}} = \frac{1}{q}\sum_{i=1}^{q} \mathbf{x}_{k+1}^{\mathrm{f}_i};$

$\mathrm{E}_k^{\mathrm{f}} = \left[\mathbf{x}_{k+1}^{\mathrm{f}_1} - \bar{\mathbf{x}}_{k+1}^{\mathrm{f}},\ \ldots,\ \mathbf{x}_{k+1}^{\mathrm{f}_q} - \bar{\mathbf{x}}_{k+1}^{\mathrm{f}}\right];$

$\mathrm{E}_{\mathbf{y}_k}^{\mathrm{f}} = \left[\mathbf{y}_k^{\mathrm{f}_1} - \bar{\mathbf{y}}_k^{\mathrm{f}},\ \ldots,\ \mathbf{y}_k{}^{\mathrm{f}_q} - \bar{\mathbf{y}}_k^{\mathrm{f}}\right];$

$\hat{\mathbf{P}}_{\mathbf{x}\mathbf{y}_k}^{\mathrm{f}} = \frac{1}{q-1}\mathrm{E}_k^{\mathrm{f}}\left(\mathrm{E}_{\mathbf{y}_k}^{\mathrm{f}}\right)^{\mathrm{T}},\ \hat{\mathbf{P}}_{\mathbf{y}\mathbf{y}_k}^{\mathrm{f}} = \frac{1}{q-1}\mathrm{E}_{\mathbf{y}_k}^{\mathrm{f}}\left(\mathrm{E}_{\mathbf{y}_k}^{\mathrm{f}}\right)^{\mathrm{T}};$

`/* Analysis step */`

$\hat{\mathbf{K}}_k = \hat{\mathbf{P}}_{\mathbf{x}\mathbf{y}_k}^{\mathrm{f}}\left(\hat{\mathbf{P}}_{\mathbf{y}\mathbf{y}_k}^{\mathrm{f}}\right)^{-1};$

$\mathbf{x}_k^{\mathrm{a}_i} = \mathbf{x}_k^{\mathrm{f}_i} + \hat{\mathbf{K}}_k\left(\mathbf{y}_k + \boldsymbol{v}_k^i - h(\mathbf{x}_k^{\mathrm{f}_i}, \mathbf{u}_k^i)\right),\ i = 1, \ldots, q;$

$\bar{\mathbf{x}}_k^{\mathrm{a}} = \frac{1}{q}\sum_{i=1}^{q} \mathbf{x}_k^{\mathrm{a}_i};$

It should be noted that the index k refers to the time index, n the total number of samples, q the total number of ensembles, and $\boldsymbol{w}_k^i, \boldsymbol{v}_k^i,\ i = 1, \ldots, q$, are zero-mean random variables with a normal distribution and covariance $\mathbf{R}_k, \mathbf{Q}_k$, respectively.

Unscented Kalman Filter

To avoid the limitations of the EKF with the linearization of the state equations, an unscented transformation and state estimation algorithm, were developed by [Julier et al., 2004] to propagate the mean and covariance information through nonlinear transformations. This approach, called the UKF, approximates a probability distribution, rather than approximating an arbitrary nonlinear function, reducing the errors which occurs in the EKF.

Approach Following the approach presented in [Simon, 2006b], given a discrete-time nonlinear system (2.9) with its belonging covariance matrices (2.5), the UKF is initialized similar to the approach of the KF. Due to no initial available measurements, it is reasonable to set the a posteriori state and covariance as given in Equation (2.6) and (2.7), respectively, with respect to the initial state $\mathbf{x}_0$. The procedure starts by obtaining the time update equations through an unscented transformation to estimate the true mean and covariance of a known nonlinear function by a set of deterministic vectors, as illustrated in Figure 2.2. The vectors, called sigma points, denoted by $\hat{\mathbf{x}}_{k-1}^{(i)}$ are, since the best guess of the mean and covariance of $\mathbf{x}_k$ are $\hat{\mathbf{x}}_k^+$ and $\mathbf{P}_{k-1}^+$, chosen as

$$\hat{\mathbf{x}}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1}^+ + \tilde{\mathbf{x}}^{(i)}, \qquad i = 1, \dots, 2n, \tag{2.14a}$$

$$\tilde{\mathbf{x}}^{(i)} = \left(\sqrt{n\mathbf{P}_{k-1}^+}\right)_i^{\mathrm{T}}, \qquad i = 1, \dots, n, \tag{2.14b}$$

$$\tilde{\mathbf{x}}^{(n+i)} = -\left(\sqrt{n\mathbf{P}_{k-1}^+}\right)_i^{\mathrm{T}}, \quad i = 1, \dots, n, \tag{2.14c}$$

where $\sqrt{n\mathbf{P}_{k-1}^+}$ is the matrix square root such that $\left(\sqrt{n\mathbf{P}_{k-1}^+}\right)^{\mathrm{T}} \sqrt{n\mathbf{P}_{k-1}^+} = n\mathbf{P}_{k-1}^+$, and $\left(\sqrt{n\mathbf{P}_{k-1}^+}\right)_i$ is the ith row of $\sqrt{n\mathbf{P}_{k-1}^+}$ (see Appendix B.2 for details of the matrix square root). The sigma points are then transformed into the vectors $\hat{\mathbf{x}}_k^{(i)}$ by using the state equation

$$\hat{\mathbf{x}}_k^{(i)} = f\left(\hat{\mathbf{x}}_{k-1}^{(i)}, \mathbf{u}_k\right) + \mathbf{w}_k, \tag{2.15}$$

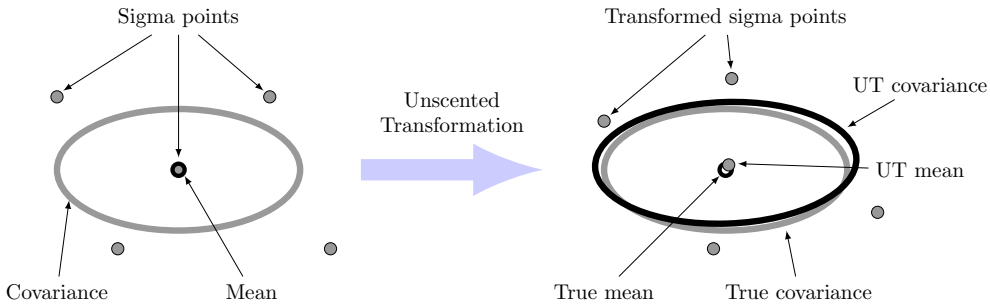

Figure 2.2: Illustration of the unscented transformation in the UKF, which approximates the covariance and mean of the nonlinear system. The left show the true mean and covariance of the system with a deterministically chosen set of points, while the right show the applied transformation with its corresponding estimated covariance and mean.

which are further used to obtain the a priori state estimate and error covariance

$$\hat{\mathbf{x}}_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\hat{\mathbf{x}}_k^{(i)}, \tag{2.16a}$$

$$\mathbf{P}_k^- = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^-\right)\left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^-\right)^{\mathrm{T}} + \mathbf{Q}_{k-1}. \tag{2.16b}$$

Similar to the calculation of the time update equations, as the current best guess for the mean and covariance of $\mathbf{x}_k$ are $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$, the sigma points are chosen as

$$\hat{\mathbf{x}}_k^{(i)} = \hat{\mathbf{x}}_{k-1}^- + \tilde{\mathbf{x}}^{(i)}, \qquad i = 1, \ldots, 2n, \tag{2.17a}$$

$$\tilde{\mathbf{x}}^{(i)} = \left(\sqrt{n\mathbf{P}_k^-}\right)_i^{\mathrm{T}}, \qquad i = 1, \ldots, n, \tag{2.17b}$$

$$\tilde{\mathbf{x}}^{(n+i)} = -\left(\sqrt{n\mathbf{P}_k^-}\right)_i^{\mathrm{T}}, \quad i = 1, \ldots, n. \tag{2.17c}$$

The sigma points are then transformed into predicted measurements $\hat{\mathbf{y}}_k^{(i)}$ by using the state measurement equation, i.e.,

$$\hat{\mathbf{y}}_k^{(i)} = h\left(\hat{\mathbf{x}}_k^{(i)}, \mathbf{u}_k\right), \tag{2.18}$$

making it possible to obtain the predicted measurement and its belong covariance matrix by

$$\hat{\mathbf{y}}_k = \frac{1}{2n}\sum_{i=1}^{2n} \hat{\mathbf{y}}_k^{(i)}, \tag{2.19a}$$

$$\mathbf{P}_y = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right)\left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right)^{\mathrm{T}} + \mathbf{R}_k. \tag{2.19b}$$

It is then necessary to calculate the estimate of the cross covariance between $\hat{\mathbf{x}}_k^-$ and $\hat{\mathbf{y}}_k$,

$$\mathbf{P}_{xy} = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^-\right)\left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k\right)^{\mathrm{T}}, \tag{2.20}$$

which makes the measurement update of the state estimate, using the normal Kalman Filter equations, to be calculated by

$$\mathbf{K}_k = \mathbf{P}_{xy}\mathbf{P}_y^{-1}, \tag{2.21a}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right), \tag{2.21b}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{P}_y\mathbf{K}_k^{\mathrm{T}}. \tag{2.21c}$$

To make the UKF algorithm more rigorous, as it in this approach treats the noise as additives in Equation (2.16) and (2.19), respectively, an augmented version of the algorithm exist. This involves augmenting the state vector as $\mathbf{x}_k^{\mathrm{a}} = \begin{bmatrix}\hat{\mathbf{x}}_k^{\mathrm{T}} & \mathbf{v}_k^{\mathrm{T}} & \mathbf{w}_k^{\mathrm{T}}\end{bmatrix} \in \mathbb{R}^{n_{\mathrm{a}}}$, $n_{\mathrm{a}} = n_{\mathrm{x}} + n_{\mathrm{v}} + n_{\mathrm{w}}$, and using the augmented vector throughout the algorithm. This approach is described in full in [Julier et al., 2004, Wan and van der Merwe, 2000].

Reduced sigma point filters

Due to the computational complexity in the calculation of the sigma points in the UKF, [Julier and Uhlmann, 2002] proved that for an n-dimensional system, only $n+1$ sigma points (minimal skew sigma points) are required to capture the mean

and covariance. This could potentially be a unstable and unreliable algorithm as the set of sigma points can lead to numerical problems. However, based on the minimal skew sigma points, [Julier, 2003] describes a better-behaved sigma point selection strategy, known as the Spherical-Simplex Unscented Kalman Filter (SSUKF), which requires only $n+2$ sigma points. This approach defines a simplex of points that lie on a hypersphere. These points, known as spherical simplex points, lie on the origin or on a hypersphere centered at the origin, making the points lie in a radius proportional to $\sqrt{n}$, and the weights applied proportional to $1/n$. The spherical simplex points are then calculated using the following criterion

$$\boldsymbol{\mathcal{X}}_i = \bar{\mathbf{x}} + \sqrt{\mathbf{P}_x}\boldsymbol{\mathcal{Z}}_i, \qquad i = 0, \ldots, n+1$$

where $\sqrt{\mathbf{P}_x}$ is a matrix square root of $\mathbf{P}_x$ and $\boldsymbol{\mathcal{Z}}_i$ is the ith column of the spherical sigma point matrix calculated by

1. Choose $0 \leq W_0 \leq 1$.
2. Choose weight sequence:

$$W_i = \frac{1 - W_0}{n+1}, \quad i = 1, \ldots, n+1$$

3. Initialize vector sequence as:

$$\boldsymbol{\mathcal{Z}}_0^1 = [0], \quad \boldsymbol{\mathcal{Z}}_1^1 = \begin{bmatrix} -\frac{1}{\sqrt{2W_1}} \end{bmatrix}, \quad \boldsymbol{\mathcal{Z}}_2^1 = \begin{bmatrix} \frac{1}{\sqrt{2W_1}} \end{bmatrix}$$

4. Expand vector sequence for $j = 2, \ldots, n$ according to

$$\boldsymbol{\mathcal{Z}}_i^j = \begin{cases} \begin{bmatrix} \boldsymbol{\mathcal{Z}}_0^{j-1} \\ 0 \end{bmatrix} & \text{for } i = 0 \\ \begin{bmatrix} \boldsymbol{\mathcal{Z}}_i^{j-1} \\ -\frac{1}{\sqrt{j(j+1)W_1}} \end{bmatrix} & \text{for } i = 1, \ldots, j \\ \begin{bmatrix} \mathbf{0}_{j-1} \\ \frac{j}{\sqrt{j(j+1)W_1}} \end{bmatrix} & \text{for } i = j+1 \end{cases}$$

The spherical sigma points are then incorporated into the UKF, with some minor differences from the original algorithm. The reader is referred to the original article for further details.

Derivative-free Extended Kalman Filter

The Derivative-Free Extended Kalman Filter (DFEKF) by [Quine, 2006] is based on the propagation of a minimal ensemble set of $n+1$ state vectors, making it computationally advantageous. It is similar to the EKF as it propagates only the first two moments of any state distribution, and can be shown to be equivalent to the EKF in a limiting case (see [Quine, 2006] for details). The outline of the state estimation algorithm is summarized in pseudo-code below.

1. *Prediction stage*

   (a) Form a set of vectors by using previous or initial state estimates $\hat{\mathbf{x}}$ and covariance $\mathbf{P}_x$

   $$\boldsymbol{\mathcal{X}}_i = \hat{\mathbf{x}} + \frac{\Delta \mathbf{x}_i}{\alpha}, i = 1, \ldots, n, \quad \text{where } \{\Delta \mathbf{x}_1 \ \ldots \ \Delta \mathbf{x}_n\} \equiv \sqrt{n\mathbf{P}_x}$$

   (b) Predict the value of $\boldsymbol{\mathcal{X}}_i, \hat{\mathbf{x}}$ by

   $$\boldsymbol{\mathcal{X}}_i = f\left(\boldsymbol{\mathcal{X}}_i\right), i = 1, \ldots, n, \qquad \hat{\mathbf{x}} = f\left(\hat{\mathbf{x}}\right)$$

   and project the state covariance

   $$\mathbf{P}_x = \frac{\alpha^2}{n} \sum_{i=1}^{n} \left(\boldsymbol{\mathcal{X}}_i - \hat{\mathbf{x}}\right)\left(\boldsymbol{\mathcal{X}}_i - \hat{\mathbf{x}}\right)^{\mathrm{T}} + \mathbf{Q}.$$

   (c) Form a set of vectors for projection

   $$\boldsymbol{\mathcal{Y}}_i = \hat{\mathbf{x}} + \frac{\Delta \mathbf{y}_i}{\alpha}, i = 1, \ldots, n, \quad \text{where } \{\Delta \mathbf{y}_1 \ \ldots \ \Delta \mathbf{y}_n\} \equiv \sqrt{n\mathbf{P}_x}$$

   (d) Predict the observation value of $\boldsymbol{\mathcal{Y}}_i, \hat{\mathbf{x}}$ by

   $$\boldsymbol{\mathcal{Y}}_i = h\left(\boldsymbol{\mathcal{Y}}_i\right), i = 1, \ldots, n, \qquad \hat{\mathbf{y}} = h\left(\hat{\mathbf{x}}\right).$$

2. *Update stage*

   (e) Calculate the covariances $\mathbf{P}_{xy}, \mathbf{P}_{yy}$ by

   $$\mathbf{P}_{xy} = \frac{\alpha^2}{n} \sum_{i=1}^{n} \left(\boldsymbol{\mathcal{X}}_i - \hat{\mathbf{x}}\right)\left(\boldsymbol{\mathcal{Y}}_i - \hat{\mathbf{y}}\right)^{\mathrm{T}},$$

   $$\mathbf{P}_{yy} = \frac{\alpha^2}{n} \sum_{i=1}^{n} \left(\boldsymbol{\mathcal{Y}}_i - \hat{\mathbf{y}}\right)\left(\boldsymbol{\mathcal{Y}}_i - \hat{\mathbf{y}}\right)^{\mathrm{T}} + \mathbf{R}.$$

(f) Form weight matrix and update state and covariance estimates by

$$\begin{aligned}\mathbf{K} &= \mathbf{P}_{xy}\mathbf{P}_{yy}^{-1},\\ \hat{\mathbf{x}} &= \hat{\mathbf{x}} + \mathbf{K}\left(\mathbf{y} - \hat{\mathbf{y}}\right),\\ \mathbf{P}_x &= \mathbf{P}_x - \mathbf{K}\mathbf{P}_{xy}^{\mathrm{T}}.\end{aligned}$$

(g) Repeat recursively to generate time-evolving estimates of state and covariance.

The parameter α is a scaling parameter, which is discussed in full in [Quine, 2006].

2.2 Model reduction

This section presents the theory associated with model reduction, a technique of reducing the number of states in large-scale models to simpler models with similar dynamics.

2.2.1 Problem statement

The problem statement of model reduction is given as follows by [Hovland, 2008, Lall et al., 2002]. Given a system represented by the nonlinear differential equations (2.1), the objective is to find a reduced system represented by the differential equations

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= f_r\left(\mathbf{x}_r(t), \mathbf{u}(t)\right),\\ \mathbf{y}_r(t) &= h_r\left(\mathbf{x}_r(t), \mathbf{u}(t)\right),\end{aligned}$$

where $\mathbf{x}_r \in \mathbb{R}^r$ and $\mathbf{y}_r \in \mathbb{R}^m$ such that $r < n$, where n is the total number of states in the original system (2.1), while having a "small" approximation error, numerical stability and computationally efficiency. A typical measure of the error can according to [Skogestad and Postlethwaite, 2005] be given as

$$\|G(s) - G_r(s)\|_\infty,$$

where $G(s), G_r(s)$ are the transfer functions of the full and reduced system, respectively, and the infinity norm is either a Hankel ($\mathcal{H}_\infty$) or $\mathcal{L}_\infty$ norm.

2.2.2 Projection

Projection is a mathematical approach of mapping a set into a subset, such as transforming points or vectors from one plane (or dimension) to another plane [Antoulas, 2005]. Consider a three-dimensional Euclidian vector $\mathbf{x}(t)$ as given in Figure 2.3, where the vector is projected down onto the subspace $\mathcal{S} \subseteq \mathbb{R}^2$ by a projection operator $\mathbf{\Phi}_r$. The projection operator represent the relationship between the two-dimensional vector $\mathbf{x}_r(t)$ and the original three-dimensional vector $\mathbf{x}(t)$.

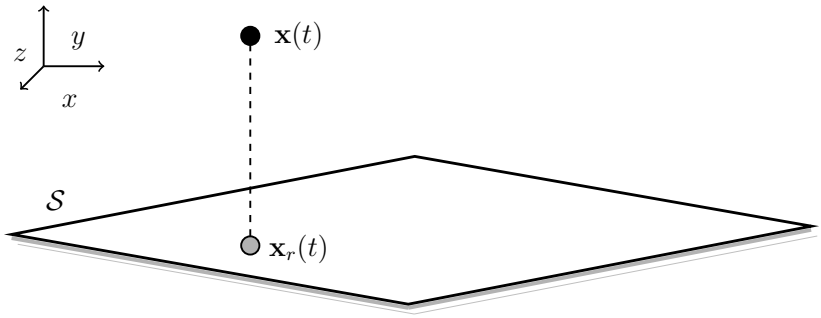

Figure 2.3: Illustration (adapted from [Farhat, 2013]) of orthogonal projection, where the three-dimensional vector $\mathbf{x}(t)$ is projected onto a two-dimensional subspace $\mathcal{S}$ at time t.

In terms of model reduction, projection is more commonly known as Galerkin projection [Lall et al., 2002, Hovland, 2008], where the general idea of projection-based reduction is to search for a subspace spanned by a projection matrix $\mathbf{\Phi} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n\}$ containing the eigenvectors corresponding to the largest eigenvalues of the system. If the eigenvalues are ordered in increasing or decreasing order, the irrelevant eigenvalues can be omitted such that the only relevant information from the system are kept. For a general system (2.1), it is assumed that the state vector $\mathbf{x}$ can be approximated by a linear combination of r basis vectors

$$\mathbf{x} \simeq \mathbf{\Phi}_r \mathbf{x}_r$$

where $\mathbf{x}_r \in \mathbb{R}^r$ is the reduced state and the reduced projection matrix $\mathbf{\Phi}_r \in \mathbb{R}^{n\times r}$ contains the r basis vectors $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_r$ so that $\mathbf{\Phi}_r^{\mathrm{T}}\mathbf{\Phi}_r = \mathbf{I}_r$. This leads to that the general nonlinear reduced system of (2.1) is given by

$$\begin{aligned}\dot{\mathbf{x}}_r(t) &= \mathbf{\Phi}_r^{\mathrm{T}} f_r\left(\mathbf{\Phi}_r\mathbf{x}_r(t), \mathbf{u}(t)\right),\\ \mathbf{y}_r(t) &= h_r\left(\mathbf{\Phi}_r\mathbf{x}_r(t), \mathbf{u}(t)\right).\end{aligned}$$

For the linear discrete case, the projected reduced model of (2.3) becomes

$$\begin{aligned}\dot{\mathbf{x}}_{r,k} &= \mathbf{A}_r\mathbf{x}_{r,k-1} + \mathbf{B}_r\mathbf{u}_{k-1},\\ \mathbf{y}_{r,k} &= \mathbf{C}_r\mathbf{x}_{r,k} + \mathbf{D}_r\mathbf{u}_k,\end{aligned}$$

where the reduced system matrices are projected as

$$\mathbf{A}_r = \mathbf{\Phi}_r^{\mathrm{T}}\mathbf{A}\mathbf{\Phi}_r, \tag{2.25a}$$
$$\mathbf{B}_r = \mathbf{\Phi}_r^{\mathrm{T}}\mathbf{B}, \tag{2.25b}$$
$$\mathbf{C}_r = \mathbf{C}\mathbf{\Phi}_r, \tag{2.25c}$$
$$\mathbf{D}_r = \mathbf{D}. \tag{2.25d}$$

There exist several methods for obtaining the projection matrix $\mathbf{\Phi}$, where some of the methods will be presented in the subsequent sections.

2.2.3 Proper Orthogonal Decomposition

POD, which according to [Holmes et al., 1996, Hovland, 2008] is also known as principal component analysis, Karhunen-Loéve expansion and Singular Value Decomposition (SVD), is a method of least-squares approximation. It was initially developed by [Lumley, 1967] for use in the field of fluid mechanics to study turbulent flows, but has over the years become one of the most popular methods of nonlinear model reduction.

Theory

[Astrid et al., 2002, Lall et al., 2002, Hovland, 2008] states that POD provides a basis for the compact representation of an ensemble of data, such as a finite number of N samples $\mathbf{x}(t_k), k = 1, \ldots, N$ from the system (2.1) in a matrix of snapshots $\mathbf{T}_{\text{snap}}$. By characterizing a subspace $\mathcal{S} \subset \mathbb{R}^n$ by the projection operator $\mathbf{\Phi}_r$, the objective is to find the POD basis that minimize the error

between the original snapshots and their representation in the reduced space, i.e., the total squared distance of the points from the r-plane

$$\min_{\mathbf{\Phi}_r} \ \mathrm{H}(\mathbf{\Phi}_r) = \sum_{k=1}^{N} \|\mathbf{x}(t_k) - \tilde{\mathbf{x}}(t_k)\|_2^2, \tag{2.26}$$

where $\tilde{\mathbf{x}}(t_k) = \mathbf{\Phi}_r \mathbf{\Phi}_r^{\mathrm{T}} \mathbf{x}(t_k)$ and r is the order of approximation. The minimizing solution $\mathbf{\Phi}_r$ can be found by the set of left singular vectors of $\mathbf{T}_{\mathrm{span}}$,

$$\mathbf{T}_{\mathrm{span}} = \mathbf{\Phi} \mathbf{\Sigma} \mathbf{\Psi}^{\mathrm{T}}, \tag{2.27}$$

where $\mathbf{\Phi}$ contains the orthogonal basis vectors and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_s)$ contains the associated singular values. The r most significant basis functions are associated with the r largest singular values $\sigma_i, i = 1, \ldots, r$. If the singular values σ_i has a significant drop in magnitude between $i = r, r+1$, a reduced-order model may be constructed by using the projection operator $\mathbf{\Phi}_r$, corresponding to the r first columns of $\mathbf{\Phi}$.

When projecting onto a subspace $\mathcal{S}$, the energy conservation in the model is of concern. Particularly, how close the approximation of data provided by an r-dimensional subspace is important and can be measured by comparing the fraction of the total 'energy'

$$P = \frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{N} \sigma_i}, \tag{2.28}$$

where the goal is to choose r such that the energy in the subspace is approximately the same as the full space, i.e., $P \simeq 1$, while keeping r sufficiently small. Keep in mind that the validity of approximations is defined by the correlation matrix $\mathbf{T}_{\mathrm{span}}$, and is limited to how well this represent the dynamics of the system. In addition, if the system is unstable, the corresponding responses may make an approximation by orthonormal basis impossible.

POD Algorithm

From the previous section, it comes clear for the perceptive reader that the calculation of Equation (2.27) is simply the SVD of the snapshot matrix $\mathbf{T}_{\mathrm{span}}$. The outline of the POD algorithm is restated from [Hovland, 2008] below in Algorithm 2.

Algorithm 2: Proper Orthogonal Decomposition

Input : Matrix of N snapshots, $\mathbf{T}_{\text{snap}} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}\}$.
Output: Matrix of the r most significant basis vectors, $\boldsymbol{\Phi}_r$

1 *Compute the SVD of the snapshot matrix* $\mathbf{T}_{\text{span}}$

$$\mathbf{T}_{\text{span}} = \boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Psi}^{\mathrm{T}}$$

2 *Extract the r most significant basis vectors* $\boldsymbol{\Phi}_r$ *based on the singular values* σ_i *of* $\boldsymbol{T}_{snap}$

3 *Project the state equations onto the reduced basis as described in Section 2.2.2.*

2.2.4 Model reduction by balanced truncation

Balanced truncation is merely the method of truncating a balanced system. A balanced system, or balanced realization, is given by [Green and Limebeer, 1994] as the approach of balancing the past input and output "energy" in the system. More generally, the controllability and observability Gramians are equal and diagonal.

Truncating a system corresponds to ordering the states based on the information they contain, and removing the states which contain little or irrelevant information. Thus, the truncated (reduced) system share approximately the same dynamics as the full system.

Linear systems

A general linear state-space system is given on the form presented in Equation (2.3), where $\mathbf{x}, \mathbf{u}$ and $\mathbf{y}$ have the same dimensions as the vectors given in Equation (2.1). A balanced realization of a linear system (2.3), is an asymptotically stable minimal realization in which the controllability and observability Gramians are equal and diagonal. According to [Green and Limebeer, 1994], a balanced realization is most commonly defined as

Definition 2.2.1. A realization ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) is balanced if $\mathbf{A}$ is asymptotically stable and

$$\mathbf{A}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\mathbf{A}^{\mathrm{T}} + \mathbf{B}\mathbf{B}^{\mathrm{T}} = 0,$$
$$\mathbf{A}^{\mathrm{T}}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\mathbf{A} + \mathbf{C}^{\mathrm{T}}\mathbf{C} = 0,$$

in which

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1\mathbf{I}_{r_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_m\mathbf{I}_{r_m} \end{bmatrix}, \quad \sigma_i \neq \sigma_j,\ i \neq j \ and\ \sigma_i > 0\ \forall i.$$

Note that $n = r_1 + \ldots + r_m$ is the size (called the McMillan degree) of $\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ and that r_i is the multiplicity of σ_i. We say that the realization is an ordered balanced realization if, in addition, $\sigma_1 > \sigma_2 > \ldots > \sigma_m > 0$. ■

To be able to validate the controllability and observability of a system, the associated Gramians needs to be calculated. The definitions of the controllability and observability Gramian are defined in [Green and Limebeer, 1994] as given below.

Definition 2.2.2. The dynamical system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$

or the pair $(\mathbf{A}, \mathbf{B})$, is state controllable if and only if the Gramian matrix

$$\mathbf{W}_c(t) \triangleq \int_0^{\mathrm{T}} e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^{\mathrm{T}}e^{\mathbf{A}^{\mathrm{T}}\tau}\mathrm{d}\tau$$

has full rank for any $t > 0$. Equivalently, for a stable system, the pair $(\mathbf{A}, \mathbf{B})$ is state controllable if and only if the controllability Gramian

$$\mathbf{O}_c \triangleq \int_0^{\infty} e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^{\mathrm{T}}e^{\mathbf{A}^{\mathrm{T}}\tau}\mathrm{d}\tau$$

is positive definite. $\mathbf{O}_c$ may also be obtained as the solution to the Lyapunov equation

$$\mathbf{A}\mathbf{O}_c + \mathbf{O}_c\mathbf{A}^{\mathrm{T}} = -\mathbf{B}\mathbf{B}^{\mathrm{T}}.$$

■

Definition 2.2.3. The dynamical system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u},$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

or the pair $(\mathbf{A}, \mathbf{C})$, is state observable if and only if the observability Gramian

$$\mathbf{O}_o \triangleq \int_0^\infty e^{\mathbf{A}^{\mathrm{T}}\tau}\mathbf{C}^{\mathrm{T}}\mathbf{C}e^{\mathbf{A}\tau}\mathrm{d}\tau$$

has full rank. $\mathbf{O}_o$ may also be obtained as the solution to the Lyapunov equation

$$\mathbf{A}\mathbf{O}_o + \mathbf{O}_o\mathbf{A}^{\mathrm{T}} = -\mathbf{C}^{\mathrm{T}}\mathbf{C}.$$

■

Using the theory above for obtaining a balanced realization, and assuming that the system (2.3) has been balanced, the system can be partitioned as

$$\mathbf{x}_{1,k} = \mathbf{A}_{11}\mathbf{x}_{1,k-1} + \mathbf{A}_{12}\mathbf{x}_{2,k-1} + \mathbf{B}_1\mathbf{u}_{k-1},$$
$$\mathbf{x}_{2,k} = \mathbf{A}_{21}\mathbf{x}_{1,k-1} + \mathbf{A}_{22}\mathbf{x}_{2,k-1} + \mathbf{B}_2\mathbf{u}_{k-1},$$
$$\mathbf{y}_k = \mathbf{C}_1\mathbf{x}_{1,k} + \mathbf{C}_2\mathbf{x}_{2,k} + \mathbf{D}\mathbf{u}_k,$$

where $\mathbf{x}_1 \in \mathbb{R}^{n-r}$, $\mathbf{x}_2 \in \mathbb{R}^r$. In addition, assuming that the r states associated with $\mathbf{x}_2$ contains no relevant dynamics, the states can be truncated making the reduced system share approximately the same amount of information as the full system.

Similar approaches to model reduction using projection through balanced truncation for the linear case exists, for example by using the method of balanced truncation by [Laub et al., 1987].

Nonlinear systems

The theory of the Gramians from Definition 2.2.2-2.2.3 is not applicable for nonlinear systems, and since nonlinear energy functions are difficult to obtain, [Lall et al., 2002, Hahn and Edgar, 2002b] presents the method of using empirical Gramians in model reduction for control purposes. This is considered an extension to balancing linear systems, as described in the previous section. The approach involves computing controllability and observability covariance matrices, where a transformation can be used with a Galerkin projection to achieve a nonlinear balanced form. The balanced equations can then be reduced by truncating the states corresponding to small Hankel singular values.

Theory The empirical Gramians have to be determined from data, either generated experimentally or by simulation, where the data should be collected within the region where the process is to be controlled. To define the empirical Gramians for a nonlinear system (2.1), the following sets are required:

$$\mathcal{T}^n = \left\{ \mathbf{T}_1, \ldots, \mathbf{T}_j;\ \mathbf{T}_i \in \mathbb{R}^{n\times n},\ \mathbf{T}_i^{\mathrm{T}}\mathbf{T}_i = \mathbf{I},\ i = 1, \ldots, j \right\} \tag{2.30a}$$

$$\mathcal{M} = \{c_1, \ldots, c_s;\ c_i \in \mathbb{R},\ c_i > 0,\ i = 1, \ldots, s\} \tag{2.30b}$$

$$\mathcal{E}^n = \{\boldsymbol{\upsilon}_1, \ldots, \boldsymbol{\upsilon}_n;\ \text{standard unit vectors in } \mathbb{R}^n\} \tag{2.30c}$$

where j is the number of matrices for perturbation directions, s the number of different perturbation sizes for each direction and n the number of states of the full order system. The sets $\mathcal{T}$ and $\mathcal{E}$ are used to determine the input directions, and $\mathcal{M}$ specify the size of the inputs and states we are interested in, where the dynamics should evolve in a region close to the operating area. An initial reasonable choice of the sets (2.30) are $\mathcal{T} = [\mathbf{I},\ -\mathbf{I}]$, $\mathcal{E} = \mathbf{I}$, while the set $\mathcal{M}$ can be chosen arbitrarily according to the desired perturbation magnitude. The choice of the set $\mathcal{T}$ corresponds to using both positive and negative inputs (or initial states) on each input separately.

The main advantage of the following approach is that it requires only the solution of the standard linear matrix eigenvalue problem, where the empirical Gramians are restated from [Hahn and Edgar, 2002a] in Definition 2.2.4 and 2.2.5 below.

Definition 2.2.4. Let $\mathcal{T}^p, \mathcal{E}^p$ and $\mathcal{M}$ be given sets as described above, where p is the number of inputs. The discrete empirical controllability Gramian is defined by

$$\mathbf{W}_{ec} \triangleq \sum_{l=1}^{j}\sum_{m=1}^{s}\sum_{i=1}^{p}\frac{1}{jsc_m^2}\sum_{k=0}^{q}\boldsymbol{\Theta}_k^{ilm}\Delta t_k$$

where $\boldsymbol{\Theta}_k^{ilm} \in \mathbb{R}^{n\times n}$ is given by $\boldsymbol{\Theta}_k^{ilm} = \left(\mathbf{x}_k^{ilm} - \mathbf{x}_{ss}^{ilm}\right)\left(\mathbf{x}_k^{ilm} - \mathbf{x}_{ss}^{ilm}\right)^{\mathrm{T}}$, $\mathbf{x}_k^{ilm}$ is the state of the nonlinear system (2.1) at time step k corresponding to the input $\mathbf{u}_k = c_m\mathbf{T}_l\boldsymbol{\upsilon}_i\delta_i + \mathbf{u}_{ss,0}$[1], and $\mathbf{x}_{ss}^{ilm}$ is the desired system trajectory. ■

Definition 2.2.5. Let $\mathcal{T}^n, \mathcal{E}^n$ and $\mathcal{M}$ be given sets as described above, where n is the number of states. The discrete empirical observability Gramian is defined by

$$\mathbf{W}_{eo} \triangleq \sum_{l=1}^{j}\sum_{m=1}^{s}\frac{1}{jsc_m^2}\sum_{k=0}^{q}\mathbf{T}_l\boldsymbol{\Psi}_k^{lm}\mathbf{T}_l^{\mathrm{T}}\Delta t_k$$

[1] δ denotes Dirac's delta function.

where $\mathbf{\Psi}_k^{lm} \in \mathbb{R}^{n\times n}$ is given by $\mathbf{\Psi}_k^{lm}{}_{ij} = \left(\mathbf{y}_k^{ilm} - \mathbf{y}_{ss}^{ilm}\right)\left(\mathbf{y}_k^{ilm} - \mathbf{y}_{ss}^{ilm}\right)^{\mathrm{T}}$, $\mathbf{y}_k^{ilm}$ is the output of the nonlinear system (2.1) corresponding to the initial condition $\mathbf{x}_0 = c_m\mathbf{T}_l\boldsymbol{\upsilon}_i + \mathbf{x}_{ss}$, and $\mathbf{y}_{ss}^{ilm}$ is the steady state the system will reach after this perturbation. ■

From the empirical Gramians it is possible to obtain a balanced realization through the subspace approach by [Lall et al., 2002], or by the algorithm given in [Hahn and Edgar, 2002a]. The approach by [Lall et al., 2002] is developed for control purposes, where the computation for balancing the empirical Gramians is as follows. Apply the Cholesky decomposition (see Appendix B.2) to $\mathbf{W}_{ec}$ so that $\mathbf{W}_{ec} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, where $\mathbf{L}$ is a lower triangular matrix with non-negative diagonal entries. Then let $\mathbf{U}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{T}}$ be the SVD of $\mathbf{L}^{\mathrm{T}}\mathbf{W}_{eo}\mathbf{L}$, and

$$\mathbf{\Phi} = \mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^{\mathrm{T}}\mathbf{L}^{-1}$$

be the change of coordinates such that the system is balanced. The states with small Hankel singular values σ_i are then truncated. If the states are ordered according to decreasing singular values, this is equivalent to applying a Galerkin projection where the r most significant states are kept. The projection can easily be performed by introducing the "identity" vector $\mathbf{I}_{\mathrm{g}} = [\mathbf{I}_r\ \mathbf{0}]$, which results in the following balanced reduced system

$$\begin{aligned}\tilde{\mathbf{x}}_k &= \mathbf{I}_{\mathrm{g}}\mathbf{\Phi}_r f_{k-1}\left(\mathbf{\Phi}_r^{-1}\mathbf{I}_{\mathrm{g}}^{\mathrm{T}}\tilde{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}\right),\\ \mathbf{y}_k &= h_k\left(\mathbf{\Phi}_r^{-1}\mathbf{I}_{\mathrm{g}}^{\mathrm{T}}\tilde{\mathbf{x}}_k, \mathbf{u}_k\right).\end{aligned}$$

It should be mentioned that [Hahn and Edgar, 2002b] shows that this approach is limited to control-affine systems, requiring modifications when the systems steady-state is different from zero. However, the approach by [Hahn and Edgar, 2002a] is limited to stable nonlinear systems, and due to quite intricate computations, the reader is asked to refer to the original paper for details.

2.2.5 Balanced Proper Orthogonal Decomposition

The balanced POD by [Rowley, 2005] obtains an approximation to balanced truncation that is computationally effective for large systems.

Algorithm 3: Balanced Proper Orthogonal Decomposition

1 *Integrate solutions* $\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_n(t)$ *of the system* $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$*, with initial conditions* $\boldsymbol{x}_k(0) = \boldsymbol{b}_k$*, where* $\boldsymbol{b}_k$ *denotes the k-th column of the* $\boldsymbol{B}$ *matrix in the system (2.3).*

2 *Compute POD modes* $\boldsymbol{\phi}_k$ *of the dataset* $\{\boldsymbol{C}\boldsymbol{x}_1(t), \ldots, \boldsymbol{C}\boldsymbol{x}_n(t)\}$*, and choose a projection rank* r *such that the error*

$$\|\mathbf{G} - \mathbf{P}_r\mathbf{G}\|_2^2 = \sum_{j=r+1}^{p} \lambda_j$$

where $\boldsymbol{G}$ *is the impulse response matrix,* $\boldsymbol{P}_r$ *is an orthogonal projection with rank* r *and* p *is the number of outputs, is acceptable.*

3 *Integrate solutions* $\boldsymbol{z}_1(t), \ldots, \boldsymbol{z}_r(t)$ *of the adjoint system* $\dot{\boldsymbol{z}} = \boldsymbol{A}^T\boldsymbol{z}$*, with initial conditions* $\boldsymbol{z}_k(0) = \boldsymbol{C}^T\boldsymbol{\phi}_k$*.*

4 *Form the data matrices* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *for the primal and dual solutions as*

$$\mathbf{X} = \left[\mathbf{x}_1(t_1)\sqrt{\delta_1} \ \ldots \ \mathbf{x}_1(t_m)\sqrt{\delta_m} \ \ldots \ \mathbf{x}_n(t_1)\sqrt{\delta_1} \ \ldots \ \mathbf{x}_n(t_m)\sqrt{\delta_m}\right],$$

where δ_j *are quadrature coefficients.*

5 *Compute the SVD of* $\boldsymbol{Y}^T\boldsymbol{X}$

$$\mathbf{Y}^{\mathrm{T}}\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^{\mathrm{T}} \\ \mathbf{V}_2^{\mathrm{T}}\end{bmatrix},$$

and the balanced POD modes are given by

$$\mathbf{T}_1 = \mathbf{X}\mathbf{V}_1\boldsymbol{\Sigma}_1^{-\frac{1}{2}},$$
$$\mathbf{S}_1 = \boldsymbol{\Sigma}_1^{-\frac{1}{2}}\mathbf{U}_1^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}.$$

The features of this technique are a combination of the features for balanced

realization and POD, making further elaboration unnecessary. The drawback of balanced POD is that it requires the dual system, which does not exist in nonlinear settings. However, this can be solved by linearizing the system and forming the dual, or by the method of empirical Gramians as described earlier. The procedure is summarized in Algorithm 3.

2.2.6 Goal-oriented model-constrained reduction

Goal-oriented model-constrained reduction is, in contrast to the previously presented techniques, a optimization problem proposed by [Bui-Thanh et al., 2007], which has the objective to find an optimal rth order basis $\mathbf{\Phi}_r$ and corresponding reduced states $\mathbf{x}_r$ that minimize a criteria similar to (2.26) in the POD.

Consider a finite set with $\mathcal{I}$ different instances of the system (2.1), where the corresponding reduced-order system is of the form (2.2), the optimization problem on the interval $(0, t_f)$ is given as

$$\min_{\mathbf{\Phi}_r, \mathbf{x}_r} \ \mathcal{G} = \frac{1}{2} \sum_{k=1}^{\mathcal{I}} \int_0^{t_f} \left(\mathbf{y}^k - \mathbf{y}_r^k\right)^{\mathrm{T}} \left(\mathbf{y}^k - \mathbf{y}_r{}^k\right) \ \mathrm{dt} + \frac{\beta}{2} \sum_{j=1}^{n} \left(1 - \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{\phi}_j\right)^2 + \frac{\beta}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \left(\boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\phi}_j\right)^2 \tag{2.32a}$$

$$\text{subject to} \quad \dot{\mathbf{x}}_r^k = \mathbf{\Phi}_r^{\mathrm{T}} f_r\left(\mathbf{\Phi}_r \mathbf{x}_r^k, \mathbf{u}^k\right), \quad k = 1, \ldots, \mathcal{I}, \tag{2.32b}$$

$$\mathbf{x}_{r_0}^k = \mathbf{\Phi}_r^{\mathrm{T}} \mathbf{x}_0^k, \qquad k = 1, \ldots, \mathcal{I}, \tag{2.32c}$$

$$\mathbf{y}_r^k = h_r\left(\mathbf{\Phi}_r \mathbf{x}_r^k, \mathbf{u}^k\right), \qquad k = 1, \ldots, \mathcal{I}, \tag{2.32d}$$

where β is a regularization parameter penalizing the deviation of the basis vectors from an orthonormal set, and $\mathbf{x}_0$ is the initial states. If the relationship between the outputs and states are linear, the objective function is given as

$$\mathcal{G} = \frac{1}{2} \sum_{k=1}^{\mathcal{I}} \int_0^{t_f} \left(\mathbf{x}^k - \hat{\mathbf{x}}^k\right)^{\mathrm{T}} \mathbf{H}^k \left(\mathbf{x}^k - \hat{\mathbf{x}}^k\right) \ \mathrm{dt} + \frac{\beta}{2} \sum_{j=1}^{n} \left(1 - \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{\phi}_j\right)^2 + \frac{\beta}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \left(\boldsymbol{\phi}_i^{\mathrm{T}} \boldsymbol{\phi}_j\right)^2,$$

where $\mathbf{H}^k = \left(\mathbf{C}^k\right)^{\mathrm{T}} \mathbf{C}^k$ is considered a weighting matrix defining the relevant states to the specified output. The optimization problem (2.32) minimizes the output error, whereas the POD cost is related to minimizing the error over the

entire state domain, making it more suitable for output-feedback implementation. However, by altering the minimizing criteria, for example to the largest degree of observability or minimizing the error in a specific state, the approach could potentially provide a better basis than the POD.

Solving the optimization problem will in general be more computationally expensive than using the POD. The cost of solving the offline optimization problem should therefore be taken into consideration when choosing the procedure for obtaining the basis $\mathbf{\Phi}_r$.

2.3 Differentiation

In order to be able to perform the linearization, the calculation of the Jacobians, in the EKF, a differentiation technique is required. The choice for this thesis, due to its simplicity is a simple difference quotient. There exist many other techniques such as automatic and symbolic differentiation that are more numerical stable, but many of the other techniques are more computationally expensive.

2.3.1 Difference quotient

The most basic and simple numerical differentiation scheme is the difference quotient computation, which has the objective to add a small step length to approximate the next function value. Derived from Taylors theorem, the difference quotient is given by [Hass et al., 2011] as

$$\nabla f(\bar{x}) = \lim_{\alpha \to 0} \frac{f(\bar{x} + \alpha) - f(\bar{x})}{\alpha} + O(\alpha), \quad \alpha \neq 0$$

where α is a small positive scalar, $\bar{x}$ a column vector and $O(\alpha)$ is the error, provided that this limit exists. The notation $O(\alpha)$ should be read 'big oh of α' and indicates the upper limit of the error, which behaves linearly.

The advantage of the difference quotient is that it only needs a valid function, but has the disadvantage that the accuracy is highly dependent on the value of α. The choice of α is somewhat of a paradox. A small value of α can reduce the number of significant figures, while if α is not small enough, truncation errors may become significant.

2.4 Implementation

This section will summarize which algorithms and techniques that are actually implemented during the work related to the thesis. First, all of the state estimation algorithms are implemented according to the theory presented. The implementation of the algorithms are only assumed to be valid, as it is difficult to make satisfying tests which validate the state estimation algorithms sufficiently.

Most of the model reduction techniques are implemented, but only the POD and the balanced truncation by [Lall et al., 2002] are actually used in this thesis. The two used implementations of the model reduction techniques are tested through "textbook examples" from their respective references or similar books/journals. Table 2.1 gives an overview of which methods from this chapter that are implemented and if they are validated.

Table 2.1: Overview of the different techniques and methods from the background theory that are implemented and/or validated.

Description	Implemented?	Validated?
EKF	Yes	No
UKF	Yes	No
EnKF	Yes	No
SSUKF	Yes	No
DFEKF	Yes	No
POD	Yes	Yes
Balanced truncation, Lall	Yes	Yes
Balanced truncation, Hahn	Yes♠	No
Balanced POD	Yes♠	No
Goal-oriented POD	No	No
Difference quotient	Yes	Yes

♠ Implemented, but not used in this thesis.

3 Efficiency improvement in state estimation

3.1 Introduction

The complexity of state estimation, as discussed initially in the introduction, increase with the size and complexity of a system. Nevertheless, and regardless of the objective, a high computational complexity is never preferable, which in the context of this thesis, leads to seek for more efficient state estimation algorithms. From the previous work presented in Chapter 1, on the subject of efficiency improvement, the common approach is to use the reduced model throughout the entire state estimation algorithms. This includes using reduced models in both the covariance and state propagation in the EKF, or reduced sigma points and transformations in the UKF. These algorithms will be referred as the traditionally reduced algorithms for the rest of the thesis. This chapter is concerned with the subject of increasing the efficiency in state estimation algorithms according to the problem formulation presented in Section 1.3.

As this thesis is interested in several state estimation algorithms, the bulk of complexity is not a common denominator and stems from different parts in the algorithms. However, the one thing they have in common which make the complexity more expensive, is the necessary matrix operations used in the calculation of the estimates. By deriving a new approach of estimating the states, which emphasize on reducing the bulk of complexity in the algorithms based on projections from the full system, the computational complexity should be reduced without any significant loss in information.

The following sections will introduce the necessary reconstructions to the full space and their importance in the new approach, the main complexity in

the most common known nonlinear state estimators, and the new, derived, state estimators.

3.2 Reconstructions

The mapping from Section 2.2.2 plays a significantly part of the efficiency improvement presented in this thesis when projecting to a reduced system, as the reconstruction of the full order system is necessary operation. Based on the

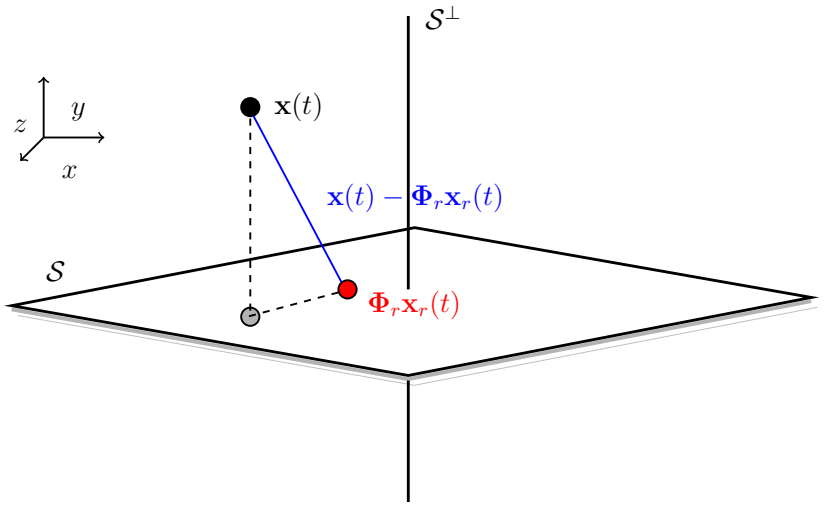

Figure 3.1: Illustration (adapted from [Farhat, 2013]) of the reconstruction of the Euclidian three-dimensional vector $\mathbf{x}(t)$ (black) from the projected vector $\mathbf{x}_r(t)$ (gray) by the use of the projection operator $\mathbf{\Phi}_r$. The blue line indicates the error between the actual vector and the reconstructed vector (red). This illustrates the main problem of getting the error as small as possible, when reconstructing the states.

projection operator, the reconstructed full-order state solution

$$\mathbf{x}^{\star} = \mathbf{\Phi}_r\mathbf{x}_r$$

must be obtained. In general, the desired objective is to obtain the mapping that yields the smallest error between the true and reconstructed states,

$$\mathbf{e} = \mathbf{x} - \mathbf{x}^{\star}, \tag{3.1}$$

such that the reconstructed states are as close to the true states as possible. Based on the theory from [Antoulas, 2005], and to illustrate the problems that may occur during reconstruction of the states, the example of the three-dimensional Euclidian vector from Section 2.2.2 (Figure 2.3) is considered. When reconstructing the vector there is a high probability that the reconstruction will differ from the initial vector. As shown in Figure 3.1, the reconstructed vector is not in the vicinity of the initial vector, but located close to the subspace $\mathcal{S}$. Hence, the numerical value of the error (3.1) is not equal to zero, as indicated by the blue line in the figure. Thus, there is a loss in information when reconstructing the vector. By obtaining a other projection basis could potentially result in a better reconstruction and a smaller error. The method used for obtaining the basis, and the size of the basis is therefore of importance. Note that all reconstructed matrices and vectors will from here on be indicated by superscript $\star$.

3.3 Extended Kalman Filter

The bulk of complexity in the EKF from Section 2.1.3, stems from the propagation of the state covariance matrix (2.11a), which requires the calculation of the system Jacobians (2.10) and (2.12). The bulk of complexity can be illustrated by the systems

$$\text{(a)} \quad \dot{\mathbf{x}}_k^l = \mathbf{A}\mathbf{x}_{k-1}^l + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}, \qquad \text{(b)} \quad \dot{\mathbf{x}}_k^{nl} = f\left(\mathbf{x}_{k-1}^{nl}, \mathbf{u}_{k-1}\right) + \mathbf{w}_{k-1},$$

where (a) is a linear system, and (b) is nonlinear system. The associated Jacobians, i.e., differentiating the system equation with respect to the state vector, are given as

$$\text{(a}_1\text{)} \quad \left.\frac{\partial \dot{\mathbf{x}}_k^l}{\partial \mathbf{x}_k^l}\right|_{\mathbf{x}_k^l = \mathbf{x}_{k-1}^l} = \mathbf{A}, \qquad \text{(b}_1\text{)} \quad \left.\frac{\partial \dot{\mathbf{x}}_k^{nl}}{\partial \mathbf{x}_k^{nl}}\right|_{\mathbf{x}_k^{nl} = \mathbf{x}_{k-1}^{nl}} = \left.\frac{\partial f}{\partial \mathbf{x}_k^{nl}}\right|_{\substack{\mathbf{x}_k^{nl} = \mathbf{x}_{k-1}^{nl} \\ \mathbf{u}_k = \mathbf{u}_{k-1} \\ \mathbf{w}_k = \mathbf{w}_{k-1}}}. \tag{3.2}$$

The result of the linear case, given that the differentiation is numerically stable, can be considered time invariant with a constant numerical result. This corresponds to that the EKF is equivalent to the KF if the system is linear, as

mentioned earlier. The nonlinear case ($3.2b_1$) is relatively more computationally complex, as the result of the Jacobian is time variant and dependent on the value of $\mathbf{x}^{nl}, \mathbf{u}$ and $\mathbf{w}$ at time t_{k-1}. If the system matrices in addition are time variant, the complexity would increase in both cases, making the nonlinear case even more complex. From this it is given that the computational complexity in the EKF is dependent on the number of states in the system, both due to the differentiation and the number of elements in the system matrices.

3.3.1 Reduced Extended Kalman Filter

The following sections will introduce new derived methods of the EKF, which incorporate techniques from model reduction and use the reduced models to achieve a more efficient approach. The methods involve using the reduced system equations for calculating the Jacobians, while the full model is used for state propagation. The difference between the two subsequent methods are the placement of the similarity transformations.

Full covariance matrix

This approach involves using the full covariance matrix, $\mathbf{P} \in \mathbb{R}^{n \times n}$, in the covariance update (2.11a). Thus, the Jacobians of the reduced system equations, $\mathbf{F}_r$ and $\mathbf{L}_r$, respectively, will be calculated in the subspace and must be reconstructed to the full space. Specifically, incorporating the similarity transform (2.25) in the covariance update (2.11a), leads to the modified covariance update

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}^\star \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^{\star^{\mathrm{T}}} + \mathbf{L}_{k-1}^\star \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^{\star^{\mathrm{T}}},$$

where the reconstructed Jacobians are given as

$$\mathbf{F}_{k-1}^\star = \left(\mathbf{\Phi}_r^{\mathrm{T}}\right)^\dagger \mathbf{F}_{r,k-1} \mathbf{\Phi}_r^\dagger,$$
$$\mathbf{L}_{k-1}^\star = \left(\mathbf{\Phi}_r^{\mathrm{T}}\right)^\dagger \mathbf{L}_{r,k-1} \mathbf{\Phi}_r^\dagger,$$

and $\mathbf{\Phi}^\dagger$ denotes the pseudoinverse of $\mathbf{\Phi}$[1]. In addition to the above, the Jacobians of the measurement equation (2.12), $\mathbf{H}_r$ and $\mathbf{M}_r$, will be calculated in the projected subspace. These are essential components in the calculation of the

[1]Note that the pseudoinverse is not always necessary to calculate, as $\mathbf{\Phi}^\dagger = \mathbf{\Phi}^{\mathrm{T}}$ may apply.

Kalman gain (2.13a), and are reconstructed to the full space by

$$\mathbf{H}_k^\star = \mathbf{H}_{r,k}\boldsymbol{\Phi}_r^\dagger, \tag{3.4a}$$

$$\mathbf{M}_k^\star = \mathbf{M}_{r,k}\boldsymbol{\Phi}_r^\dagger. \tag{3.4b}$$

This ensures that the Kalman gain corresponds to the total number of states in the full system and leads to the modified Kalman gain update

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{\star^{\mathrm{T}}} \left(\mathbf{H}_k^\star \mathbf{P}_k^- \mathbf{H}_k^{\star^{\mathrm{T}}} + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^{\mathrm{T}} \right)^{-1}.$$

Thus, the incorporated changes above allows the use of the reduced Jacobians while keeping the full model for state propagation, reducing the bulk of complexity. The algorithm is given in full in Appendix A.1.

Reduced covariance matrix

In the previous method, the Jacobians are reconstructed to ensure that they correlate to the size of the covariance matrix. The following approach does the exact opposite, i.e., projecting the covariance matrix to correlate to the dimensions of the reduced Jacobians, leading to a reduced version of the covariance update (2.11a). The reduced covariance update, $\mathbf{P}_{r,k}^-$, makes the involved matrix operations more efficient due to less matrix operations, but leads to problems with the dimensions in the calculation of the Kalman gain update. Thus, the reduced covariance update must be reconstructed to the full space to ensure correct dimensions in the calculation of the final prediction step

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^{\star-} \left(\mathbf{H}_k^\star\right)^{\mathrm{T}} \left(\mathbf{H}_k^\star \mathbf{P}_k^{\star-} \left(\mathbf{H}_k^\star\right)^{\mathrm{T}} + \mathbf{M}_k^\star \mathbf{R}_k \left(\mathbf{M}_k^\star\right)^{\mathrm{T}} \right)^{-1}, \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{y}_k - h_k\left(\hat{\mathbf{x}}_k^-, \mathbf{u}_k\right) \right], \\
\mathbf{P}_k^+ &= \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^{\star-},
\end{aligned}$$

where the reconstructed covariance update is given by

$$\mathbf{P}_k^{\star-} = \left(\boldsymbol{\Phi}_r^{\mathrm{T}}\right)^\dagger \mathbf{P}_k^- \boldsymbol{\Phi}_r^\dagger, \tag{3.6}$$

and $\mathbf{H}^\star, \mathbf{M}^\star$ are as given in Equation (3.4), further reducing the computational complexity. The algorithm is given in full in Appendix A.1.

To understand how this approach is less computationally expensive, a short introduction of matrix operations and measurements of such operations are

given. The complexity of matrix operations can be measured in Floating point operations (FLOP)s, a measure of how many necessary operations the matrix multiplications require. For example, from [Golub and Van Loan, 1996] it is given that the multiplication of an $M \times N$ and $N \times L$ matrix requires $2MNL - ML$ operations. By projecting the a priori covariance in the covariance update (where the main complexity, besides the Jacobians, of the EKF is located) instead of reconstructing the Jacobians, the number of FLOPs is reduced. Thus, the reduced estimation error and process noise covariance matrices can be used in the calculation, making the overall state estimation more efficient.

Even though this approach requires less FLOPs, it should approximately lead to the same estimation result. The main difference is that the necessary projections and reconstructions to ensure correct dimensions are placed more strategically to achieve a larger efficiency. However, as a preliminary precaution to the reader, there may be numerical differences between the methods as the projections or reconstructions may cause a greater loss in information.

3.4 Unscented Kalman Filter

The UKF, as presented in Section 2.1.3, relies on the calculation of $2n+1$ sigma points, which makes up the bulk of complexity in this type of filters (sigma point filters). Thus, the method presented in the following section focus on reducing the complexity associated with the calculation of the sigma points.

3.4.1 Reduced Unscented Kalman Filter

The following section will cover the new, derived, method of the UKF, where similar to the reduced EKF, the techniques for model reduction are incorporated. The following method minimize the calculation of the sigma points by using the reduced system in the calculation. The state vector and the covariance matrix needs to be projected onto the subspace in order to calculate the

reduced sigma points

$$\begin{aligned}
\hat{\mathbf{x}}_{r,k-1}^{(i)} &= \mathbf{\Phi}_r^{\mathrm{T}} \hat{\mathbf{x}}_{k-1}^{+} + \tilde{\mathbf{x}}_r^{(i)}, && i = 1, \ldots, 2n_r, \\
\tilde{\mathbf{x}}_r^{(i)} &= \left(\sqrt{n_r \mathbf{\Phi}_r^{\mathrm{T}} \mathbf{P}_{k-1}^{+} \mathbf{\Phi}_r} \right)_i^{\mathrm{T}}, && i = 1, \ldots, n_r, \\
\tilde{\mathbf{x}}_r^{(n_r+i)} &= -\left(\sqrt{n_r \mathbf{\Phi}_r^{\mathrm{T}} \mathbf{P}_{k-1}^{+} \mathbf{\Phi}_r} \right)_i^{\mathrm{T}}, && i = 1, \ldots, n_r,
\end{aligned}$$

which are derived from the sigma points (2.14) in the original UKF. The number of sigma points is then reduced to $2n_r + 1$, where n_r is the size of the reduced system, greatly reducing the complexity if n_r is sufficiently small.

This approach, in addition to calculating the reduced sigma points, use the full system equation in the transformation of the sigma points (2.15), as using the reduced system equation in the transformations, would make the algorithm equal to similar (traditional) approaches conducted by others. Thus, the reduced sigma points must be reconstructed to the full space,

$$\hat{\mathbf{x}}_{k-1}^{\star,(i)} = \mathbf{\Phi}_r \hat{\mathbf{x}}_{r,k-1}^{(i)}.$$

Further, to calculate the next round of sigma points (2.17), the a priori state estimate must be projected onto the subspace,

$$\begin{aligned}
\hat{\mathbf{x}}_{r,k-1}^{(i)} &= \mathbf{\Phi}_r^{\mathrm{T}} \hat{\mathbf{x}}_{k-1}^{-} + \tilde{\mathbf{x}}_r^{(i)}, && i = 1, \ldots, 2n_r, \\
\tilde{\mathbf{x}}_r^{(i)} &= \left(\sqrt{n_r \mathbf{P}_{r,k}^{-}} \right)_i^{\mathrm{T}}, && i = 1, \ldots, n_r, \\
\tilde{\mathbf{x}}_r^{(n_r+i)} &= -\left(\sqrt{n_r \mathbf{P}_{r,k}^{-}} \right)_i^{\mathrm{T}}, && i = 1, \ldots, n_r.
\end{aligned}$$

Hence, once gain, the reduced sigma points must be reconstructed to the full space, as the second transformation of the sigma points (2.18) use the full measurement equation,

$$\hat{\mathbf{y}}_k^{(i)} = h\left(\mathbf{\Phi}_r \hat{\mathbf{x}}_{r,k}^{(i)}, \mathbf{u}_k, \mathbf{v}_k \right),$$

which is also a required operation for the calculation of the cross covariance (2.20). The final step is to reconstruct the error covariance back to its full

dimensions, as shown for the reduced EKF in Equation (3.6), in order to obtain the full error covariance in the calculation of the measurement update (2.21). These changes makes up the derived method, where the algorithm is presented in full in Appendix A.3.

4 Estimation of heat conduction models

4.1 Introduction

In this chapter, two PDE models, discretized by the Finite Volume approach, which is a popular discretization technique in CFD software, are introduced. The models presented are heat conduction models in one and two dimensional plates, as these models are easy to understand and well suited for illustrative purposes.

The method of POD was in Chapter 2 presented as a model reduction technique. The method is based on using simulation data to obtain POD basis functions, which in combination with projection is used to obtain a reduced-order model. This chapter will present the implementation of the POD on the given discretized model, as the POD has the advantage that it minimize the error over the entire domain, which is of particular interest as this thesis is concerned with state estimation and, thus, the entire domain.

4.2 One dimensional heat conduction model

The model presented is given by [Astrid et al., 2002, Astrid, 2004] and is a one dimensional heat conduction process with the following specifications. Consider a slab of length $L = 0.1$m, with two heat sources located at the leftmost end and at the middle of the slab, marked 1 and 2, respectively, as given by the

sketch of the slab in Figure 4.1. The slab is divided into 400 equally spaced grid points, with a grid size of $\Delta x = 0.00025$m. The heat transfer by conduction

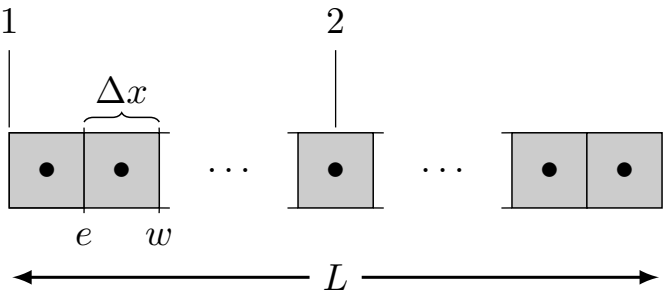

Figure 4.1: Sketch of the slab (adapted from [Astrid et al., 2002]). The figure show the grid points representing the full length L, while the heat sources are marked with 1 and 2. The markings e, w represent the east and west side of the grid point.

along the slab is given by the PDE

$$\rho c_p \frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2},$$

where κ is the thermal conductivity, ρ is the density of the slab and ρ_c is the heat capacity. The physical parameters are listed in Table 4.1.

Table 4.1: The physical parameters of the slab, with notations on the left, value in the middle and a description to the right.

Physical parameters, slab			
Notation	Value	Unit	Description
κ	100	[W/(m K)]	Thermal conductivity (taken as a constant)
L	0.1	[m]	Length of slab
ρc_p	10^7	[J/(m^3 K)]	Multiplication of density and heat capacity

The discretized CFD model, derived in full in [Astrid et al., 2002, Astrid, 2004],

with the time space divided into $\Delta t = 2s$ is summarized by

$$a_p T(x_P, t+\Delta t) = a_P^0 T(x_P, t) + a_E T(x_{P+1}, t+\Delta t) + a_W T(x_{P-1}, t+\Delta t) - S_P u(t),$$

where $T(x_P, \Delta t)$ denotes the temperatures at a specific coordinate x_P. The linear system equation is given as

$$\mathbf{A}\mathbf{T}_k = \mathbf{A}_0\mathbf{T}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}, \tag{4.1}$$

where $\mathbf{A}_0 \in \mathbb{R}^{400\times 400}$ is a diagonal matrix with the constant a_P^0 on the diagonal, and $\mathbf{A} \in \mathbb{R}^{400\times 400}$ is a tridiagonal matrix generally given as

$$\mathbf{A} = \begin{bmatrix} a_P & -a_E & 0 & \dots & 0 \\ -a_W & a_P & -a_E & \dots & 0 \\ 0 & -a_W & a_P & -a_E & 0 \\ \vdots & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{4.2}$$

where the coefficients are the contributions from the previous time step and the eastern and western neighboring grid points. The matrix $\mathbf{B} \in \mathbb{R}^{400\times p}$ is a collection of the row vectors $\mathbf{S}_U \in \mathbb{R}^{1\times p}$, where p is the number of actuators. The coefficients are generally given as

$$\begin{aligned} a_E &= \frac{\kappa}{\Delta x} \\ a_W &= \frac{\kappa}{\Delta x} \\ a_P^0 &= \rho c_p \frac{\Delta x}{\Delta t} \\ S_P &= -\frac{2\kappa}{\Delta x} \\ S_U &= \frac{2\kappa}{\Delta x} \end{aligned}$$

where S_P denotes the contributions from the neighboring cells *if* these are the boundary points or the actuators. Thus, this concludes the mathematical presentation of the model.

4.3 Reduced modeling of one dimensional heat conduction model

In this thesis, POD is used as the main model reduction method. This method is described in Section 2.2.3, and is an appropriate choice even though the

model is unstable. The instability of the model is sufficiently slow, such that the responses of the simulation makes them still relevant.

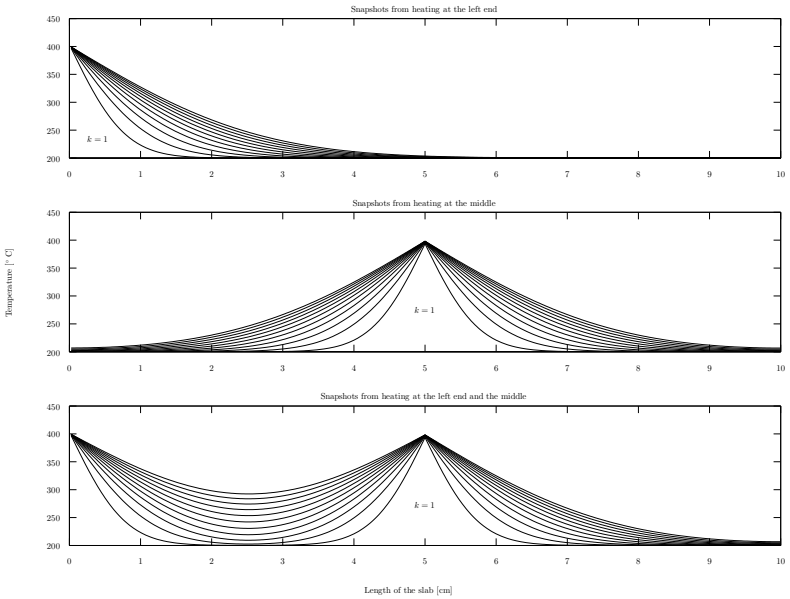

Figure 4.2: Three simulations of different actuator settings of the one dimensional heat conduction model; heating at the left end, the middle and left end and middle, making up the snapshot matrix $\mathbf{T}_{\text{snap}}$. The lines in the plot indicate the heat development at time t_k, $k = 1, 10, 20, \ldots$.

To obtain the most accurate dynamics of the model, a set of 120 snapshots are obtained from each of the three following actuator settings;

- left actuator,
- middle actuator, and
- left and middle actuator.

The data from the actuator settings are then acquired through simulation, as illustrated in Figure 4.2, where the order of the subplots correspond to the list

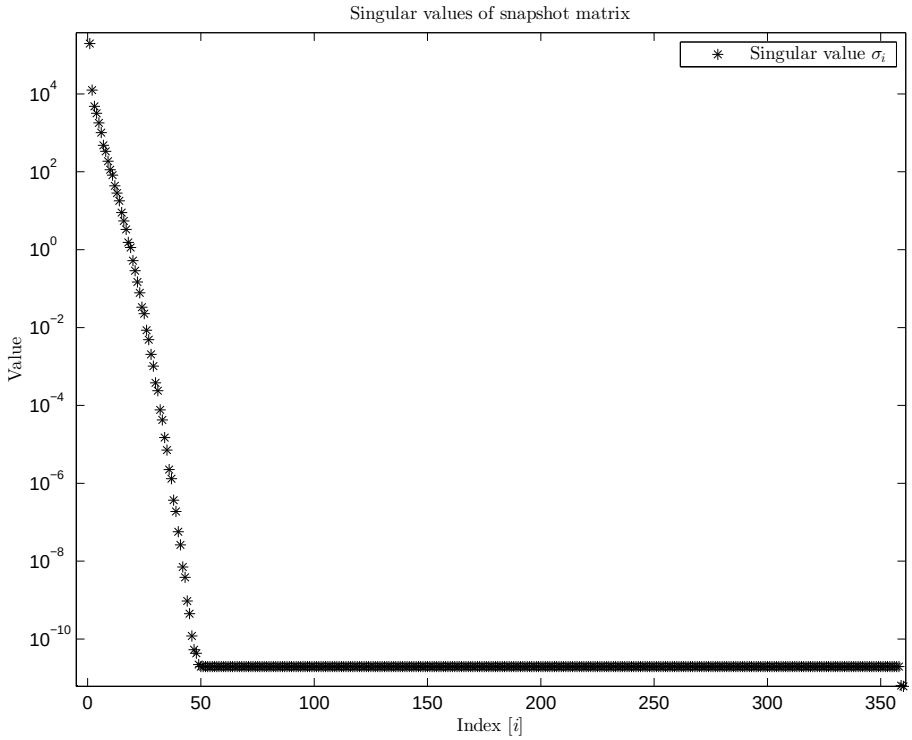

Figure 4.3: The singular values of the snapshot matrix $\mathbf{T}_{\text{span}}$ (4.3). The numerical values of the singular values drops rapidly, making only the first "few" enough to preserve the energy from the original system. Note that the y-axis is scaled logarithmically.

presented above. The lines in the figure indicates the temperature development of the heat in the slab over time. The data is then gathered in the snapshot matrix

$$\mathbf{T}_{\text{snap}} = \begin{bmatrix} \mathbf{T}_l(1) & \dots & \mathbf{T}_l(120) & \mathbf{T}_m(1) & \dots & \mathbf{T}_m(120) & \mathbf{T}_{lm}(1) & \dots & \mathbf{T}_{lm}(120) \end{bmatrix},$$

where the subscripts l, m and lm corresponds to the left, middle, and left and middle, respectively.

The corresponding singular values of the snapshot matrix (2.27) is represented graphically in Figure 4.3, where the number of singular values to retain

should be chosen such that the energy condition (2.28) is satisfied. The figure show that the values of the singular values falls quite rapidly, with many of the values close to or equal zero. Selecting the size of the basis to be minimal, i.e., $2, 3$ and 4, the conserved energy is given as $0.9990, 0.9996$ and 0.9999, respectively. Thus, the full system can be projected to a substantially smaller subspace, while keeping most of the total energy preserved.

4.4 Two dimensional heat conduction model

The two dimensional heat conduction model is given by [Astrid, 2004, Astrid et al., 2003], where the model is a rectangular plate, illustrated by the sketch in Figure 4.4. The dimension of the plate is $0.3m \times 0.4m \times 0.01m$, corresponding to length (L), height (H) and thickness, respectively. The thermal conductivity of the plate material is given as $\kappa = 1000\text{W/mK}$, and the plate is divided into 33×44 grid cells, making the grid size in horizontal and vertical direction to be $\Delta x = \Delta y = 0.0091m$.

The heat conduction model is given by

$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa \frac{\partial T}{\partial y}\right),$$

which when solved and discretized, with the time space divided into $\Delta t = 1\text{s}$, is summarized by

$$\begin{aligned} a_p T(x_P, k+1) = a_P^0 T(x_P, k) + a_N T(x_N, k+1) + a_S T(x_S, k+1) \\ + a_E T(x_E, k+1) + a_W T(x_W, k+1) - S_P u(t), \end{aligned}$$

where x_P is a specific coordinate and x_E, x_W, x_N, x_S denotes the east, west, north and south coordinate, respectively, according to the grid point P. This is similar to the solution of the one dimensional model (4.1), only now with $\mathbf{A}_0 \in \mathbb{R}^{1452\times 1452}$ as a diagonal matrix with a_P^0 on the diagonal and $\mathbf{A} \in \mathbb{R}^{1452\times 1452}$

as a pentadiagonal-like matrix with the structure

$$\mathbf{A} = \begin{bmatrix} \overbrace{\begin{matrix} a_P & a_E & 0 & \dots & 0 \end{matrix}}^{GC_x} & a_S & 0 & \dots & & & 0 & \dots \\ a_W & a_P & a_E & 0 & \dots & 0 & a_S & 0 & \dots & & 0 & \ddots \\ 0 & a_W & a_P & a_E & 0 & \dots & 0 & a_S & 0 & \dots & 0 & \ddots \\ \vdots & & \ddots & \ddots & \ddots & & & & \ddots & & 0 & \ddots \\ 0 & 0 & \dots & a_W & a_P & 0 & \dots & & 0 & a_S & 0 & \ddots \\ a_N & 0 & \dots & & 0 & a_P & a_E & 0 & \dots & & a_S & \ddots \\ 0 & a_N & 0 & \dots & 0 & a_W & a_P & a_E & 0 & \dots & \ddots & \ddots \\ \vdots & & \ddots & & & \ddots & & & \ddots & & & \ddots \\ \\ \vdots & & & & \ddots & & & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & & & 0 & a_N & 0 & \dots & 0 & a_W & a_P & a_E \\ 0 & \dots & & & & 0 & a_N & 0 & \dots & 0 & a_W & a_P \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \end{matrix}\right\} GC_x \quad ,$$

where GC_x is the number of grid cells in the x-direction. The structure makes the respective grid cell able to obtain the contributions from the neighboring cells, where the matrix $\mathbf{A}$ is structured such that the first row corresponds to the upper-left grid cell, and the last row to the lower-right grid cell. The values of the coefficients are given as

$$\begin{aligned} A &= LH, \\ a_E = a_W &= \frac{\kappa A}{\Delta x}, \\ a_N = a_S &= \frac{\kappa A}{\Delta y}, \\ a_P^0 &= \rho c_p \frac{\Delta x \Delta y}{\Delta t}, \end{aligned}$$

$$S_P = -\frac{2\kappa}{\Delta x},$$
$$S_U = \frac{2\kappa}{\Delta x},$$

with $\rho c_p = 10^7 \mathrm{J}/(\mathrm{m}^3\ \mathrm{K})$ and $\kappa = 1000\mathrm{W}/(\mathrm{m\ K})$.

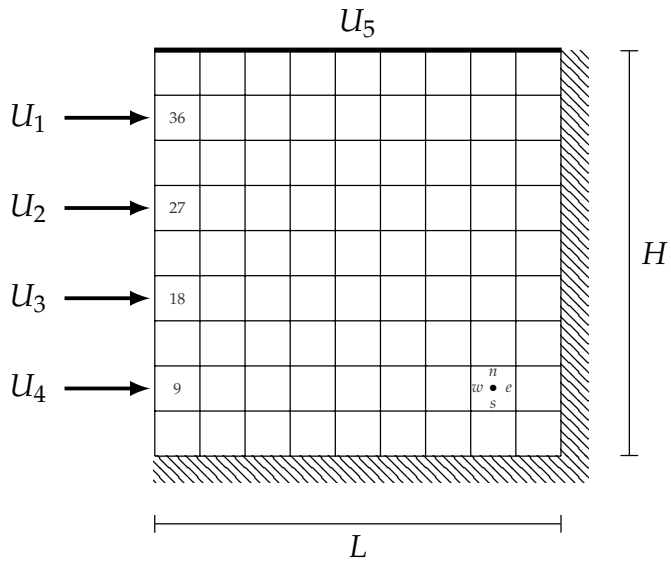

Figure 4.4: Sketch of the heated plate (adapted from [Astrid, 2004]). A grid point is illustrated at the lower right corner, marked with a bullet point, while the heat sources are marked with $U_i, i = 1, \ldots, 5$. The markings n, s, e, w near the grid point represent the north, south, east and west side of the grid point, respectively. The pinstripe diagonal lines indicates the insulated sides of the plate.

4.5 Reduced modeling of two dimensional heat conduction model

To obtain the most accurate dynamics of the model, a set of 1000 snapshots from each of the following three actuator settings are collected for the POD;

- constant temperature of 100°C at the top,
- constant temperature of 100°C at the top and left, and
- constant temperature of 100°C at the top with varying temperatures at the left actuators.

The simulations are given in Figure 4.5, where the figure show the three settings in individual subplots. For illustrative purposes, the figure only show the first and last time time step of the simulation. The first time step is given by the lower surface plot, while the upper surface plot indicates the final time step of the simulation, which should illustrate the entire heat development in the plate. The data from the simulations are then gathered in the snapshot matrix,

$$\mathbf{T}_{\text{snap}} = \begin{bmatrix} \mathbf{T}_t(1) & \dots & \mathbf{T}_t(1000) & \mathbf{T}_{tl}(1) & \dots & \mathbf{T}_{tl}(1000) & \mathbf{T}_{tlv}(1) & \dots & \mathbf{T}_{tlv}(1000) \end{bmatrix},$$

where the subscript t, tl and tlv indicates the top, top and left, and top and varying left actuator setting, respectively. The reason for adding all of the actuator settings are to represent the dynamics of the model as good as possible. To obtain a POD basis, the procedure is identical to the one presented for the one dimensional model. That is, the POD basis is obtained based on the singular values of the snapshot matrix $\mathbf{T}_{\text{snap}}$, such that the energy (2.28) in the projection is preserved.

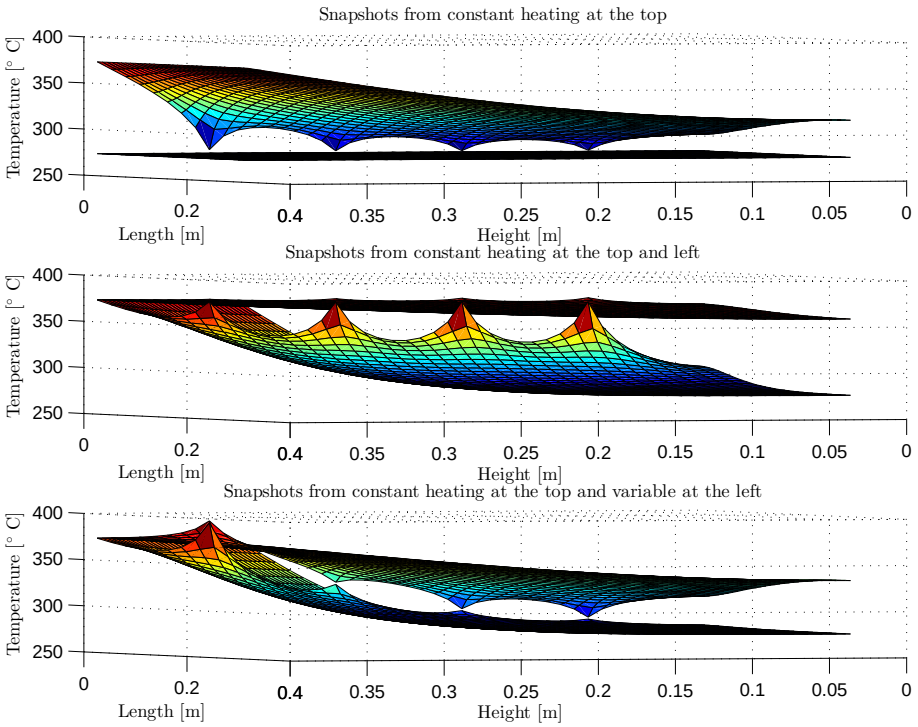

Figure 4.5: Three simulations of different actuator settings of the two dimensional heat conduction model; constant heating at the top, constant heating at the top and the left actuators, and constant heating on the top and varying heating at the left actuators, making up the snapshot matrix $\mathbf{T}_{\text{snap}}$. The lower surface in the plots indicate the plate at the first time step in the simulation, while the upper surface represent the last time step.

5 Hardware and software setup

This chapter presents the hardware specifications used for simulation and profiling, in addition to the experimental setup and the requirements imposed on the testing environment. This is meant as an introduction to how the performance and numerical results of the simulations will be conducted, such that the results can be used in other similar or further studies. This is especially significant for the profiling results, which are to some degree dependent on the hardware specifications, i.e., processor time, memory, FLOPs, et cetera.

5.1 Hardware specification

To be able to justify the performance of the estimation algorithms from Chapter 2, all of the simulations and performance evaluations are conducted on the same machine, i.e., with the same architecture, hardware specifications and operating system. The machine is a portable computer of the brand Lenovo, model T61p, which had its release in 2008.
With todays acceleration in electronic components[1], the internal components are less powerful than newer machines and their components. It is therefore considered essential to present the components used for the simulations, in order to allow the reader to evaluate the current, or possible, upcoming results in similar studies. The hardware specifications of the machine is summarized in Table 5.1, where all necessary information such as chipset, processor and memory should be provided.

[1] Moore's law - `http://www.intel.com/content/dam/www/public/us/en/documents/backgrounders/standards-22-nanometers-technology-backgrounder.pdf`, 25/05/2013

Table 5.1: Hardware specifications used in the simulation and profiling. The specifications of the internal components are for a Lenovo T61p, a portable computer released in 2008.

System specifications	
Operating System	Microsoft Windows 7 Professional 64-bit SP1
Motherboard	Intel® PM965, 800MHz FSB
Processor	Intel® Core™ 2 Duo T9500 @ 2.60GHz, 6MB cache
Memory	4032 MB DDR2 SDRAM, PC2-5300 (667MHz)
Hard Drive	SATA-I 1.5Gb/s, 7200RPM, 16384KB buffer size

5.2 Experimental setup

The software used for the simulations and calculations is MathWorks `MATLAB` R2012a under Microsoft Windows 7 (with service pack 1 installed). Both the software and the operating system run the x64 (64-bits) architecture, where `MATLAB` is installed with Microsoft Windows Software Development Kit (SDK) 7.1 as its default compiler. The performance evaluations with respect to time and memory are performed by using the built-in profiler in `MATLAB`, and the numerical estimation results are to be evaluated by defining a criteria suited to the respective model.

In the simulations, the standard deviations for the process and measurement are kept constant, given as $r = q = 0.1$, with the belonging covariance matrices $\mathbf{R} = r^2$ and $\mathbf{Q} = q^2\mathbf{I}$, respectively. Note that $\mathbf{I}$ denotes the identity matrix. The covariance matrices corresponds to the matrices given in Equation (2.5). The initial states are chosen arbitrarily, while the initial error covariance matrix (2.4) is given by $\mathbf{P} = 0.1\mathbf{I}$. The standard ensemble size in the EnKF is set to 10 and 50 in the one and two dimensional heat conduction model, respectively, unless otherwise specified.

In addition to the above, there are some software-specific conditions that are found necessary. To ensure that a direct comparison of the algorithms using the projected subspace is valid, random varying data, such as process and measurement noise, will be predetermined. Thus, the random variables will be equal for all simulations, making the comparison applicable. Some model-specific details regarding the noise follows;

- **One dimensional heat conduction model**
  - The applied process and measurement noise will be uniformly distributed, of the same magnitude, for all states according to the above.

- **Two dimensional heat conduction model**
  - The process noise is to be applied with different magnitude at certain areas of the plate. This should make the disturbances more realistic, as it simulates contributions from draft, external heat sources et cetera, in addition to making the estimation more challenging.
  - The measurement noise is the same as for the one dimensional model.

5.3 Testing environment

Some precautions are made in order to ensure that the data represented in this thesis is as accurate as possible. All of the simulations and tests that require a stable environment when gathering data, will be conducted under the following conditions;

- The processors affinity of `MATLAB` will be set to one core
  - ensures that all scheduling and processor tasks are performed under the same conditions
- Running processes will be kept at a minimum
  - makes the processor and memory less affected by other processes
  - only necessary system services are allowed to be running in the background
- The mean of data is to be used where it is possible
  - ensures good statistical data

6 Simulation and estimation results

6.1 Introduction

This chapter presents the results gathered from the simulations of the one and two dimensional heat conduction model from Chapter 4, with the setup and conditions as presented in Chapter 5. The focus of this chapter will mainly be on the estimation results of non-measured states, as estimating measured states are considered trivial. The reduced models are mainly obtained through the use of POD basis functions, as described in Chapter 4. In addition, a short assessment by the use of balanced truncation, obtained through the use of empirical Gramians, will be presented for comparison reasons.

The simulations will be conducted for different subspace configurations in order to cover and evaluate the complexity and numerical results corresponding to a wide span of the full space. The chapter will start by evaluating the time complexity for the original and derived algorithms, as this data will be unaffected by induced errors and disturbances. The chapter will further cover several common errors and disturbances that can occur during state estimation and modeling. More specifically, Gaussian white noise and a systematic error are induced as disturbances in the measurement equation, while parameter errors are added to the model, all of which are evaluated separately. The results will show how the derived algorithms compare to the original algorithms by a criteria based on the error between the true and estimated states, specifically designed to suit each of the models.

6.2 Time complexity

The data related to the time complexity of the state estimation algorithms is obtained by measuring the time used in the calculation of one time step instead of the overall time use. This allows a better reflection of the actual time consumption in the algorithm, as the simulation time may generally vary. As described in Chapter 5, the data is obtained through profiling and represents the mean of the entire simulation, to ensure good statistical properties. The results from both the one and two dimensional heat conduction model will be presented in the following sections.

6.2.1 One dimensional heat conduction model

The one dimensional heat conduction model from Section 4.2 is simulated for a duration of 800 time steps. The original algorithms, which are independent of the projected subspace, should in a perfect testing environment yield a identical time complexity in each simulation. However, due to some system services that are running in the background during the simulations, which affects the profiling differently at each simulation, the time complexity varies slightly along with the size of the subspace. The variations are shown in Figure 6.1, where the algorithms with subscript r and r_2 indicates the derived algorithms with the full and reduced covariance matrix (EKF), or the use of the reduced model in the calculation of the sigma points (UKF), respectively. The algorithms with subscript tr are the algorithms which have the "traditional" approach, i.e., by using the reduced model throughout the entire algorithm. These notations will be used for the rest of the thesis. The results of the original algorithms are merely represented to work as indicators of any abnormalities during the calculations, allowing corresponding corrections with respect to the other algorithms to be made.

The efficiency improvement and the gain of using the reduced models in comparison to the full algorithms are clearly beneficial. Figure 6.1 show that the EKF_{tr} and EKF_{r_2} have a near identical time complexity, giving the new approach the advantage of only using the reduced system equations for covariance propagation. This could potentially give a better estimation result, while sharing the same low computational time. The EKF_r, which is assumed to produce the same estimation result as EKF_{r_2}, has a much larger time complexity than the EKF_{r_2}, and is the reduced algorithm with the least improvement. The UKF_{tr} is more efficient than the EKF_{r_2} and EKF_{tr} at the smaller subspaces, but is less efficient when the subspace exceeds approximately 250 states ($\tilde{6}2.5\%$

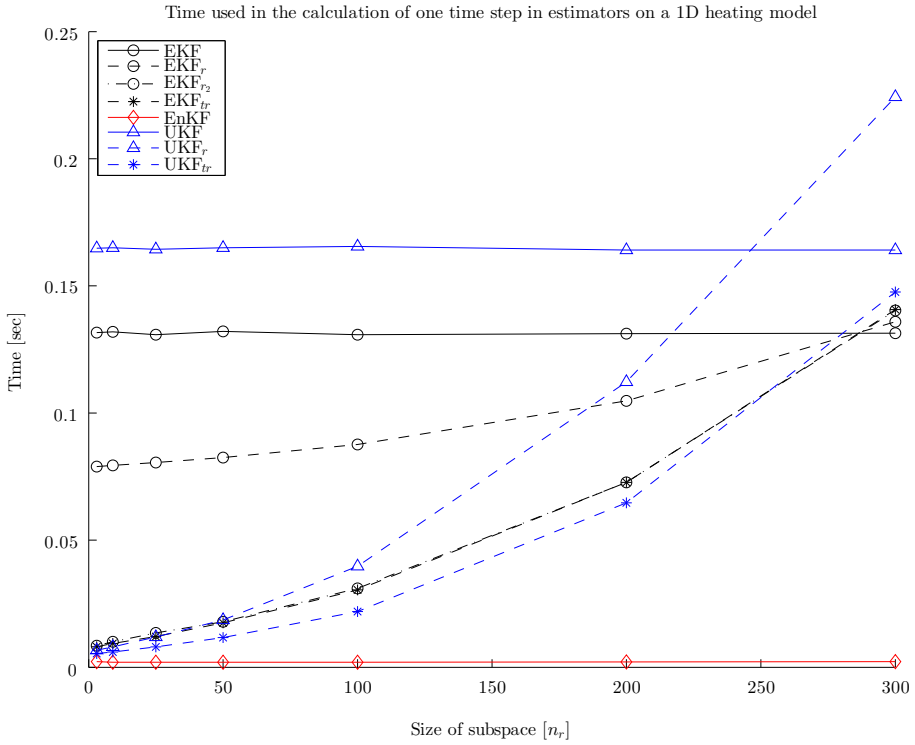

Figure 6.1: The time complexity of calculating one time step in of the state estimators of the slab. The lines indicate how the different size of subspaces affects the computational time complexity in the algorithms. The time complexity of the full original algorithms (solid lines) are close to constant regardless of the size of the subspace (small variations are related to background processes during the simulation), while the reduced algorithms (dashed and dashed-dot lines) increase up towards, and even above, the computational time of the original algorithms. This is related to the computational cost of multiplying the similarity transformation matrices, which increase with the dimension of the subspace. The algorithms marked by an asterisk is the "traditional" (reduced model in the entire algorithm) approach of the respective algorithms.

of the full space). In general, the EnKF is the most efficient algorithm and is

superior when compared to the EKF and the UKF, with approximately 95-98% faster computational time in the calculation of one time step.

The algorithms that use a subspace share that the time complexity varies along with the size of the subspace, the numerical differentiation, the calculation of the sigma points and the associated matrix operations. Thus, when increasing the subspace, the time complexity in the algorithms increase, as shown in Figure 6.1. From the figure it is obvious that the size of the subspace (percentage-wise) correlates with the time complexity, where the algorithms should intuitively, as long as the subspace does not span the entire full space, be less computational expensive than the original algorithm. However, the necessary similarity transformations, as described in Section 2.2.2 and 3.2, leads to significantly more matrix operations, which are particularly expensive when the size of the subspace is close to the full space. The data in the figure indicates a critical point, or upper limit, to the dimension of the projected subspace, as the computational time of the reduced algorithms exceed the computational time of the original algorithms. The UKF_r show a upper limit on the subspace of approximately 68% of the full space, while the other derived algorithms show a upper limit of approximately 73% of the full space. Clearly, the subspace must be kept below these limits to ensure more computationally efficient algorithms. The computational cost of the matrix operations, are briefly explained in Section 3.3.1.

It should be noted that even though the EKF and UKF share the same asymptotic complexity [Thrun et al., 2005], they do not yield the exact same computational time. This is due to that the numerical differentiation used in the EKF is a simple difference quotient, which is perturbed in only one direction and optimized to perform two separate differentiations in one function call, i.e., with respect to both the state and noise vector. Thus, this makes the EKF slightly more efficient than the UKF.

The reduced sigma point filters (SSUKF, DFEKF) are not a part of the main focus of the thesis and, hence, not mentioned in this chapter. However, they are briefly compared and evaluated up against the data presented in this section in Appendix C.

6.2.2 Two dimensional heat conduction model

The data of the two dimensional heat conduction model is, similar to the one dimensional heat conduction model to ensure good statistical properties, gathered over a total of 300 time steps. This model has more than triple the number of states than the slab, making it significantly more computationally expensive.

The data obtained from the simulations are represented in Table 6.1.

Table 6.1: Time consumption in the two dimensional heat conduction model of calculating one time step in the full and reduced estimators of Section 2.1. The data is obtained from calculating the mean of 300 measurements, where the subspace in the reduced algorithms varies. The EKF, UKF and EnKF are the algorithms that use the full scale model, which makes the data approximately consistent for any size of the subspace. The EKF and UKF share the same asymptotically complexity, which are well reflected from the data. The computational time of the reduced algorithms increase as the subspace increase, due to the associated similarity transformations and matrix operations, and exceeds the complexity of the original algorithms at large subspaces.

	Size of subspace				
	2	**50**	**100**	**700**	**1400**
EKF	4.9376s	4.9621s	4.9568s	4.9364s	5.1355s
EKF$_r$	3.2565s	3.2834s	3.3364s	4.4025s	7.5605s
EKF$_{r_2}$	0.1472s	0.2327s	0.3557s	2.9016s	9.8252s
EKF$_{tr}$	0.1254s	0.2209s	0.3385s	2.9040s	9.3236s
UKF	5.6162s	5.5962s	5.5745s	5.5775s	5.7540s
UKF$_r$	0.1114s	0.2209s	0.3787s	4.2467s	15.1453s
UKF$_{tr}$	0.0899s	0.1434s	0.2211s	2.4438s	10.1449s
EnKF	0.0603s	0.0584s	0.0617s	0.0627s	0.0654s

Studying the table for overall performance show that the most expensive reduced algorithm is the EKF$_r$, that the EKF$_{r_2}$, EKF$_{tr}$ and UKF$_r$ share approximately the same computational time, and that the most time-efficient algorithm, besides the EnKF, is the traditional UKF$_{tr}$. At the smallest subspace configuration, $n_r = 3$, the UKF$_r$ is actually close to the computational time of the EnKF. The general behavior is similar to the result presented for the one dimensional model. Although the analysis of the data in this section is a bit short, the data show similar results and ratio between the full and reduced algorithms as in the one dimensional model. Hence, approximately the same in-depth analysis applies for this section.

Generally, the derived algorithms show great promise when it comes to reducing the computational time complexity. However, the data from the table further enhances that there exist an upper (critical) limit to the dimension of the subspace, occurring at the same percentage between the derived and full

algorithms as for the slab. Thus, it is crucial and necessary to choose a subspace configuration below the critical limit in order to achieve any benefits and computational gains.

6.3 Gaussian white noise

The first induced error, to validate the accuracy and result of the different algorithms, is Gaussian white noise added to both the process and measurement equation. As both the EKF and UKF assumes a Gaussian distribution, both the original and derived algorithms should hopefully produce good estimates. It should be noted that the disturbances share the same magnitudes as the process and measurement covariance matrix, making them closely related and correlated.

6.3.1 One dimensional heat conduction model

The simulation of the one dimensional heat conduction model is performed with the first actuator engaged, heating the slab at the left end, while measuring the temperature at the opposite end, with an additional "hidden" measurement at the middle. The purpose of the hidden measurement is to make a comparison of a estimate other than the actual measurement. This makes the evaluation of the algorithms more accurate, as it gives a broader perspective of the actual estimation result. The simulation results are given in Figure 6.2, where the true and estimated states are presented. The upper subplot show that the estimate of the measured state is similar to the true state, where the three original algorithms, EKF, UKF and EnKF, yield approximately the same result. If the measurement was represented instead of the true state, the plot would show that the algorithms filters some of the noise added to the measurement equation, making the estimates more accurate than the actual measurement. The lower subplot represents the non-measured state, where in this case, the estimates filters most of the noise, making the numerical estimation result very close to the true state.

To make the comparison more detailed, the absolute mean error between the true and estimated states are calculated, i.e.,

$$\bar{\mathbf{e}} = \frac{1}{n_s} \sum_{k=1}^{n_s} \left| \mathbf{y}_{\mathrm{T},k} - \hat{\mathbf{y}}_k \right|,$$

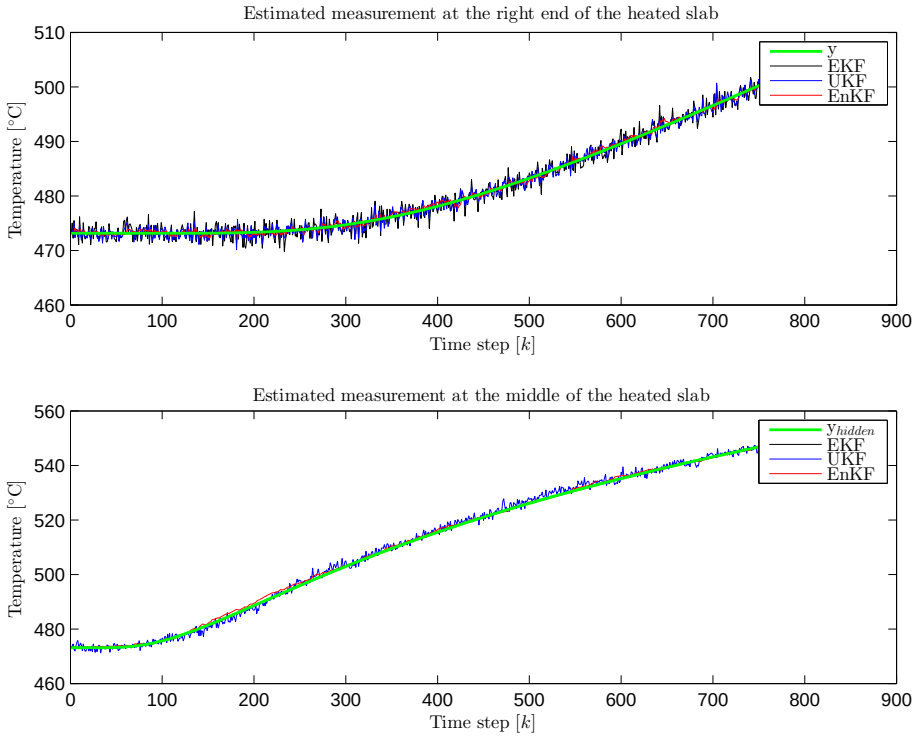

Figure 6.2: Comparison of the measured and estimated states in the one dimensional heat conduction model. The upper subplot show the true measured state, i.e., without noise, and its estimates, while the lower subplot show the hidden measured state and its estimates. The estimation results of the EKF, UKF and EnKF are clearly affected by the measurement and process noise, but filters the actual noise. All estimators are close to the true states in the process, although the three algorithms have different approaches for estimating the states.

where n_s is the total number of time samples, $\mathbf{y}_\mathrm{T}$ is the true states and $\hat{\mathbf{y}}$ is the estimated states, making a small value yield a good approximation. Using this criteria on the estimation results founds the basis of the data presented in Table 6.2, in addition to a graphical presentation given in Figure 6.3. The data show that the numerical values are exactly the same for the EKF and

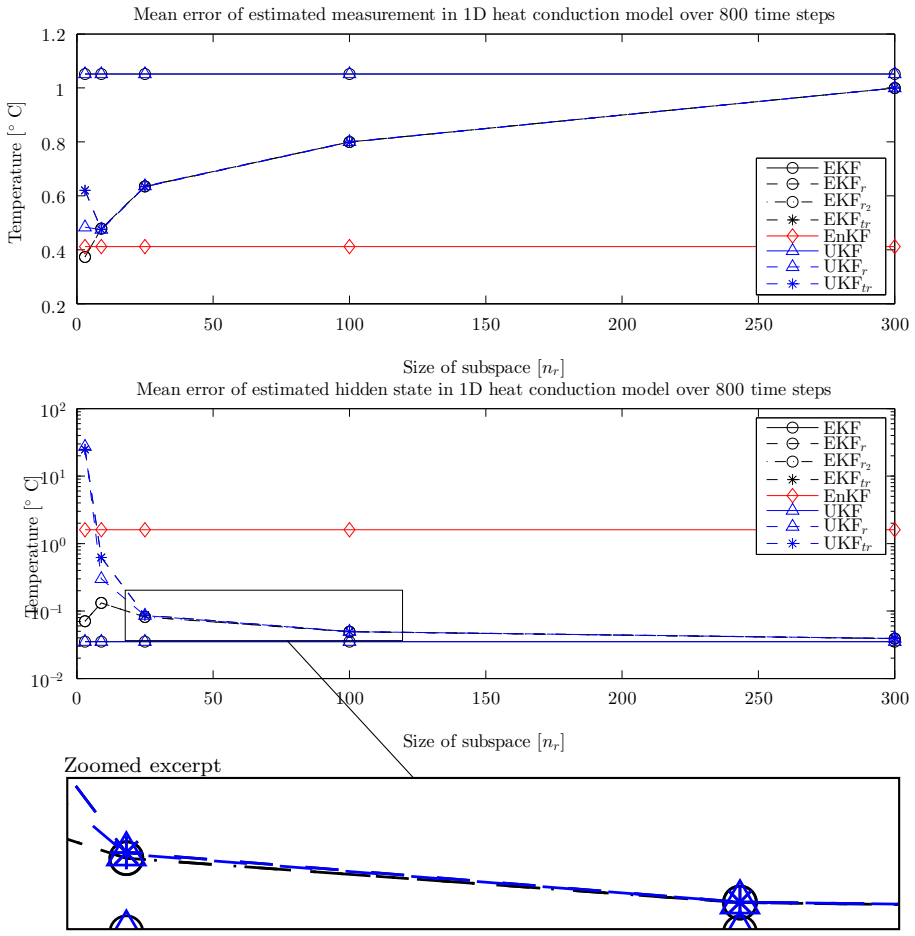

Figure 6.3: Absolute mean error [°C] between the true and estimated states for the slab with random Gaussian white noise added on both the process and the measurement equation. The data is gathered over 800 time steps. The upper and lower subplot represent the data for the mean error between the estimates and the true measured and hidden state, respectively. The interesting data is the non-measured state in the lower subplot, where the mean error of the reduced algorithms decrease as the subspace increase. The reduced algorithms share approximately the same results for subspace configurations larger than $n_r = 25$, as given in the zoomed excerpt. The traditionally reduced algorithms (denoted by subscript *tr* and marked by asterisk) share identical values.

Table 6.2: Absolute mean error [°C] between the true and estimated states for the slab with random Gaussian white noise added on both the process and measurement equation. I) represent the data for the mean error between the estimated and true measured state, while II) represent the data for the mean error between the estimated and hidden true state. As the data from I) is considered trivial, the data in II) show that the mean error is equal for both the original EKF and UKF, whereas the error of the reduced algorithms decrease as the subspace increase, as one would expect. The EnKF gives the largest mean error in the estimation of the hidden state, and the other original algorithms gives the smallest.

		Size of subspace $[n_r]$				
		3	**9**	**25**	**100**	**300**
I)	**EKF**♠	*1.0517*	*1.0517*	*1.0517*	*1.0517*	*1.0517*
	EKF$_r$	0.3731	0.4783	0.6351	0.7995	0.9995
	EKF$_{r_2}$	0.3731	0.4783	0.6351	0.7995	0.9995
	EKF$_{tr}$	0.6203	0.4748	0.6341	0.7992	0.9994
	UKF♠	*1.0517*	*1.0517*	*1.0517*	*1.0517*	*1.0517*
	UKF$_r$	0.4834	0.4750	0.6360	0.8002	0.9997
	UKF$_{tr}$	0.6203	0.4748	0.6341	0.7992	0.9994
	EnKF♠	*0.4115*	*0.4115*	*0.4115*	*0.4115*	*0.4115*
II)	**EKF**♠	*0.0351*	*0.0351*	*0.0351*	*0.0351*	*0.0351*
	EKF$_r$	0.0705	0.1315	0.0817	0.0494	0.0390
	EKF$_{r_2}$	0.0705	0.1315	0.0817	0.0494	0.0390
	EKF$_{tr}$	24.8045	0.6203	0.0868	0.0495	0.0392
	UKF♠	*0.0351*	*0.0351*	*0.0351*	*0.0351*	*0.0351*
	UKF$_r$	27.3374	0.2991	0.0857	0.0495	0.0391
	UKF$_{tr}$	24.8045	0.6203	0.0868	0.0495	0.0392
	EnKF♠	*1.5990*	*1.5990*	*1.5990*	*1.5990*	*1.5990*

♠ Constant data, independent of the subspace n_r

UKF, both for the measured and non-measured state, which is also the case for the two traditionally reduced algorithms. Since the algorithms share the same asymptotic complexity and characteristics, this is as expected. The EnKF has the smallest mean error of the original algorithms for the measured state, but the largest error in the non-measured state. This can be coherent to the fact

that the EnKF use Monte Carlo methods, which makes the estimation result perturb around the true value of the state due to its statistics and, thus, most noticeable in the non-measured state, as the noise is not as well compensated for as in the measured state.

The derived algorithms show an opposite behavior with respect to the measured and non-measured state. The absolute mean error of the measured state is smallest when the subspace is at its smallest, increasing towards the mean error of the original algorithms as the size of the subspace increase. The non-measured state show the exact opposite, meaning that the mean error decrease as the size of the subspace increase. The illogical behavior of a increasing error with the increase of the subspace is considered to be related to that the white noise added to the measurement equation has a greater impact on the estimate of the measured state, in addition to that the structure of the obtained POD basis may emphasize on non-measured states. Focusing on the mean error of the non-measured state, Table 6.2 show that the two derived versions of the EKF, i.e., with the use of the full and reduced error covariance matrix, yields exactly the same values. Thus, given that they produce the same numerical result, the obvious choice is to use the version with the reduced error covariance matrix, denoted with the subscript r_2, due to its lower time complexity. This is based on the results represented in Figure 6.1 or Table 6.3, as there is a clear difference in the time complexity of the two algorithms when the subspace is small. However, they also show sign of an erratic behavior as the mean error near doubles when increasing the subspace from 3 to 9, while the mean error reduces when increasing the subspace further. At a subspace configuration of $n_r = 25$, which is one sixteenth of the total number of states, the mean error of the EKF_{r_2} is 130% larger than its original algorithm. Increasing the subspace configuration further to $n_r = 100$ (1/4 of the full space), produce a mean error that is 40% larger than its original algorithm. The best estimation result is obtained at the largest subspace $n_r = 300$, with approximately 11% larger error than the original algorithm. However, this configuration would cause a larger computational complexity than the original algorithm, making it a poor choice either way.

The mean error of the non-measured state in UKF_r show the expected logical behavior of constantly decreasing as the subspace increase. However, Figure 6.3 show that there is a clear difference between the derived EKF and UKF when projecting to the smallest subspace, where the derived EKFs have a similar magnitude for the different size of the subspace configurations, the derived UKF show a relative large error when the subspace is small (780% larger than its original algorithm at a subspace size of $n_r = 3$), before converging to the values

given by the derived EKFs for larger subspaces.

The traditionally reduced algorithms, EKF_{tr} and UKF_{tr}, have identical values at each subspace configuration, similar to the behavior of the original EKF/UKF. They also share a similar behavior to the UKF_r, with respect to the mean error, i.e., by yielding a large mean error for small subspace configurations. Thus, the new derived approach of the EKF show great promise, as the estimation result produced in this algorithm is closer to the actual state than the traditional approach, while sharing the same computational complexity (as seen in the previous section) for subspace configurations below the critical limit.

6.3.2 Two dimensional heat conduction model

The heated plate from Section 4.4 is simulated with the actuators at the left end disabled, only heated by the actuator at the top surface with a constant temperature of 100°C. The measurement is located at the insulated grid point in the right bottom corner. In a similar manner to the one dimensional model, the need for determining the estimates of the non-measured states are present. However, instead of selecting a random point on the plate to estimate, the estimation of the entire plate is considered when selecting an appropriate evaluation

Table 6.3: Absolute mean error [°C] in the estimation of the heat conduction plate over 300 measurements, i.e., the mean error over the whole plate throughout the entire simulation. The mean error of the three original algorithms share the same order of magnitude, while the mean error in the reduced algorithms decrease as the size of the subspace increase.

	Size of subspace $[n_r]$				
	2	**50**	**100**	**700**	**1400**
EKF♠	*21.6568*	*21.6568*	*21.6568*	*21.6568*	*21.6568*
$\textbf{EKF}_r$	33.0510	31.3779	30.2206	29.4540	28.6365
$\textbf{EKF}_{r_2}$	33.0510	31.3779	30.2206	29.4540	28.6365
$\textbf{EKF}_{tr}$	41.4265	31.3615	31.1050	29.4859	28.6652
UKF♠	*21.6568*	*21.6568*	*21.6568*	*21.6568*	*21.6568*
$\textbf{UKF}_r$	40.5068	31.3620	30.2770	29.4856	28.6653
$\textbf{UKF}_{tr}$	41.4265	31.3615	31.1050	29.4859	28.6652
EnKF♠	*19.5861*	*19.5861*	*19.5861*	*19.5861*	*19.5861*

♠ Constant data, independent of the subspace n_r

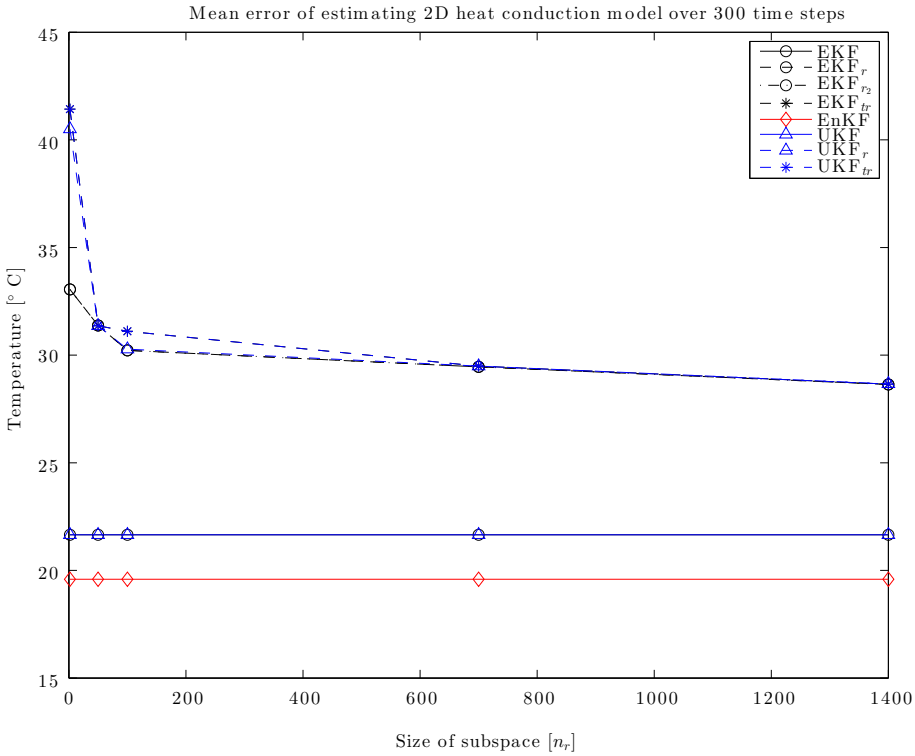

Figure 6.4: Absolute mean error [°C] in the estimation of the heat conduction plate over 300 measurements, i.e., the mean error over the whole plate through the entire simulation. The mean error of the original algorithms are in the same order of magnitude, while the projected algorithms decrease as the size of the subspace increase, as one would expect.

criteria. The criteria is chosen to evaluate the absolute mean error of the entire plate, given by

$$\bar{\mathbf{e}} = \frac{1}{n_s} \sum_{k=1}^{n_s} \sum_{i=1}^{n} \left| \mathbf{y}_{\mathrm{T},k}^{i} - \hat{\mathbf{y}}_{k}^{i} \right|,$$

where n_s is the total number of time samples and $\mathbf{y}_{\mathrm{T}}^{i}$ and $\hat{\mathbf{y}}^{i}$ is the ith true and estimated states, respectively. It should be noted that the magnitude of

the mean error between the models with this criteria will be larger than the previous result, and can not be directly compared.

The EKF and UKF and their traditional reduced alike, produce the same estimation result as shown by the values in Table 6.3 or graphically in Figure 6.4, which was also the case for the one dimensional model in the previous section. In overall, the EnKF has the smallest mean error with a couple degrees below the EKF and UKF (lower is better). The new algorithms have a significantly larger mean error when the subspace size is at its minimum ($n_r = 2$), especially the UKF$_r$ and the traditionally reduced algorithms, but similar to the slab, the mean error of the algorithms decrease as the size of the subspace increase. When using almost the entire space in the projected algorithms, the total mean error is approximately 30% larger than the original algorithms in all of the reduced algorithms. In terms of temperature, this is equal to about 7°C spread over 1452 states, which on average yields a very small mean error on each state. The estimation result is therefore assumed to be similar to the results presented for the one dimensional model, and not analyzed in further details.

6.4 Systematic error

The case of a systematic error, a bias in the measurement, is introduced, while Gaussian white noise is still added to the system equation. The constant error of 15°C to the measurement equation is simulated for two scenarios;

- the covariance matrix is kept at the same values as used for the Gaussian white noise, and
- the covariance matrix is modified to correlate with the constant error added.

There should be a clear difference between the two scenarios, as the first case will have an uncorrelated covariance matrix with the noise and should produce an estimation result with a large deviation from the true state. The second case should produce good estimates, or at least converge, to the true state. Figure 6.5 show the simulation result of the true and measured state, respectively, when added a systematic error of 15°C. Note that the disturbance in the measured state is white noise added to the process equation.

Based on the results from the previous sections, and as the two models are homogeneous and share similar dynamics, it is assumed that the estimation results will be similar for this study as well. Thus, only the one dimensional

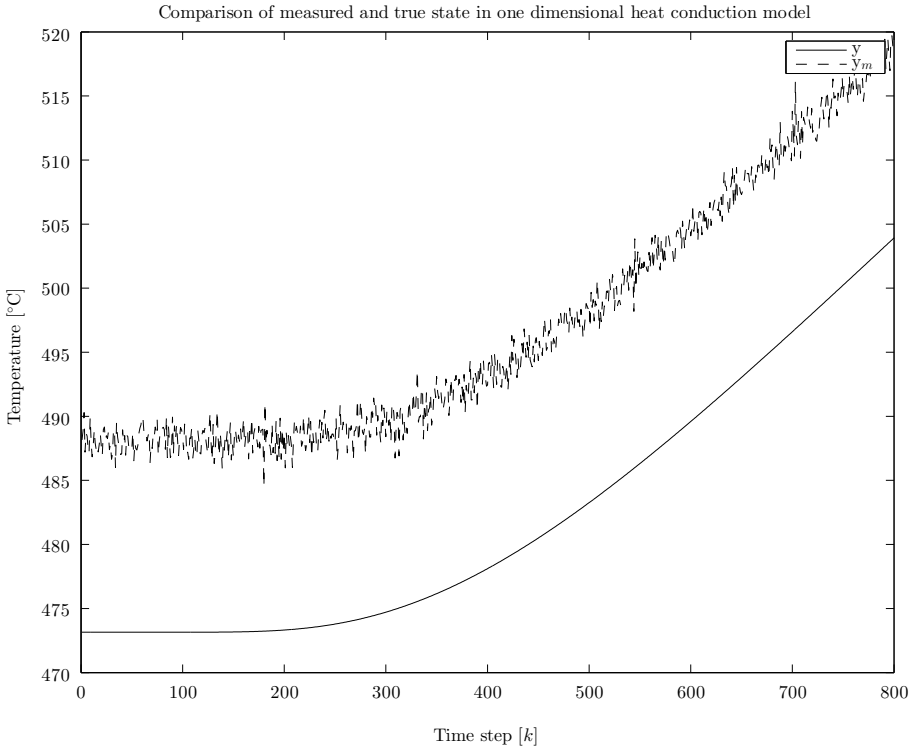

Figure 6.5: Comparison of the true state y (solid line) and the measured state y_m (dashed line), when the measurement equation is added a systematic error (constant bias) of 15°C.

heat conduction model is used in this and following simulations, as it is less computational expensive than the two dimensional model. This also enables to give a more direct comparison with respect to the non-measured state.

6.4.1 Unchanged measurement covariance matrix

Allowing the measurement covariance matrix to be unchanged, i.e., the same as in the case of the Gaussian white noise (see Section 5.2), the covariance matrix is correlated with white noise and not with the constant offset added

to the measurement equation. This should result in a poor estimation result, where the constant error should be clearly dominating. The absolute mean error

Table 6.4: Absolute mean error [°C] between the true and estimated states for the slab with white noise added to the process and a constant bias of 15°C added on the measurement equation. I) represent the data for the mean error between the estimated and the true measured state, while II) represent the data for the mean error between the estimated and the hidden true state. The estimates of the measured state are better correlated with the measurement than the actual state, giving an error similar to the bias. The non-measured state is better estimated.

		Size of subspace $[n_r]$				
		3	**9**	**25**	**100**	**300**
I)	**EKF**♠	*14.8875*	*14.8875*	*14.8875*	*14.8875*	*14.8875*
	EKF$_r$	14.7929	14.8129	14.8261	14.8467	14.8768
	EKF$_{r_2}$	14.7929	14.8129	14.8261	14.8467	14.8768
	EKF$_{tr}$	15.2780	14.8211	14.8259	14.8467	14.8768
	UKF♠	*14.8875*	*14.8875*	*14.8875*	*14.8875*	*14.8875*
	UKF$_r$	15.1171	14.8587	14.8291	14.8486	14.8813
	UKF$_{tr}$	15.2780	14.8211	14.8259	14.8467	14.8768
	EnKF♠	*14.0265*	*14.0265*	*14.0265*	*14.0265*	*14.0265*
II)	**EKF**♠	*4.8078*	*4.8078*	*4.8078*	*4.8078*	*4.8078*
	EKF$_r$	7.5664	5.0686	4.8423	4.8265	4.8154
	EKF$_{r_2}$	7.5664	5.0686	4.8423	4.8265	4.8154
	EKF$_{tr}$	15.7834	5.9828	4.8549	4.8273	4.8170
	UKF♠	*4.8078*	*4.8078*	*4.8078*	*4.8078*	*4.8078*
	UKF$_r$	18.0325	5.4308	4.8449	4.8271	4.8168
	UKF$_{tr}$	15.7834	5.9828	4.8549	4.8273	4.8170
	EnKF♠	*3.6912*	*3.6912*	*3.6912*	*3.6912*	*3.6912*

♠ Constant data, independent of the subspace n_r

between the true and estimated states are given in Table 6.4, where the data for the measured state, indicated by roman numeral I), obviously shows that all of the algorithms correlate with the measurement and not the actual state. All of the algorithms produce a estimated state that has the bias dominating, as the error is almost equivalent to the bias. The absolute mean error of the non-

measured state, indicated by roman numeral II) in the table, has a much lower mean error than the measured state, but a much larger error than in the case of the white noise (see Table 6.2). However, the estimates of the non-measured state are more accurate than the measured, which can be related to that the non-measured state is less affected by noise, due to no direct comparison of measurements.

The data from Table 6.4 are also given in Figure 6.6, providing a better overview of the development of the mean error as the subspace increase. The general behavior and development of the results with respect to the error as the subspace increase are, however, similar to the study with Gaussian white noise applied to the measurement equation.

6.4.2 Modified measurement covariance matrix

The measurement covariance matrix is modified to correlate better with the systematic error in an attempt to filter the bias, and thus, improve the estimation result of the state estimators. Table 6.5 represent the data obtained through simulation with the new measurement covariance matrix. The bias from the previous simulation (Table 6.4) is now eliminated in all of the algorithms except for the EnKF, which has an significantly increase in the error. While most of the algorithms have a surprisingly small mean error within one hundredth of a degree Celsius, the mean error of the EnKF are off by more than thirteen and eight thousand degree Celsius for the measured and non-measured state, respectively. The poor results produced by the EnKF is probably due to the Monte-Carlo statistics, making the algorithm greatly affected by the magnitude of the values in the measurement covariance matrix. The measurement covariance matrix, which in this thesis is a scalar due to only one measurement, is set to a high numerical value in order to compensate for the systematic error, which has a significantly affect on the EnKF, as shown in Table 6.5.

The derived UKF and the traditionally reduced algorithms show once again a significantly larger mean error when the subspace is small. The behavior for the derived UKF for larger subspaces, in addition to the general behavior of the other algorithms, is the same as seen in the previous sections. The mean error of the non-measured state, due to the decimal precision, is equal for all algorithms (except the EnKF) at a subspace of size $n_r = 100$ (1/4 of the full space). The derived EKFs show the greatest promise, as the mean error is close to the value of the original algorithm at a subspace of $n_r = 9$.

In order to do a fair comparison, an attempt to improve the EnKF was conducted. Altering the measurement covariance matrix showed that the esti-

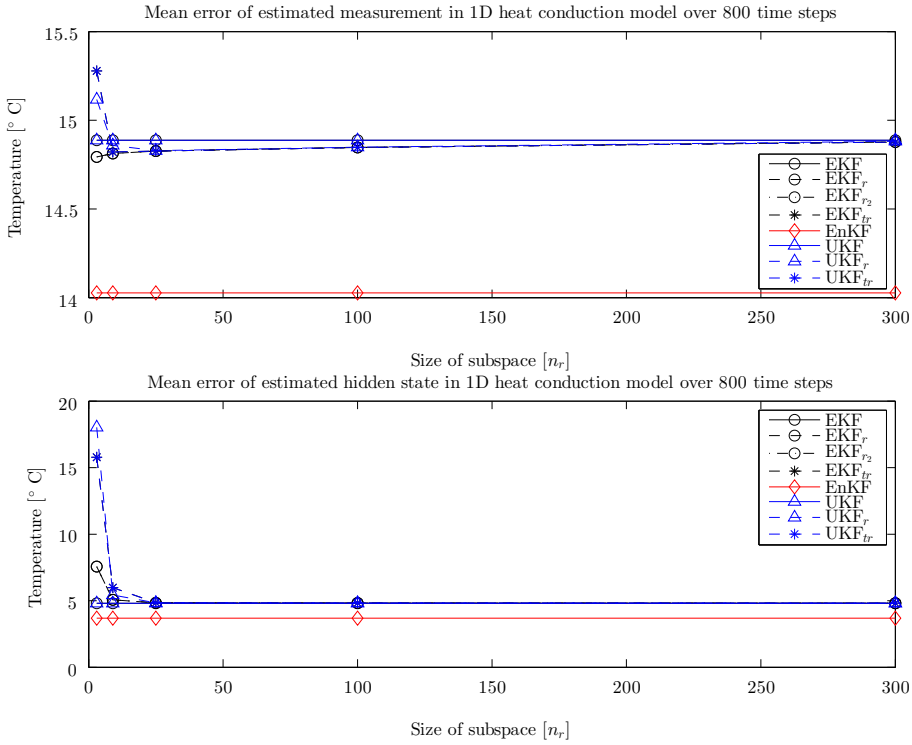

Figure 6.6: Absolute mean error [°C] of the estimated states compared to the true state with white noise added to the process and a constant bias of 15°C added on the measurement equation, gathered through simulation over 800 time steps. The upper subplot represent the data for the mean error between the estimated and the true measured state, while the lower represent the data for the mean error between the estimated and the hidden true state. The estimates of the measured state are better correlated with the measurement than the actual state, giving an error similar to the bias. The non-measured state is less affected by the disturbance and has a smaller error than the measured state.

mation result became more accurate, yielding a smaller mean error than given in Table 6.5. However, the result is not nearly as good as the results of the EKF and UKF presented in this section. Even by increasing the size of the ensemble

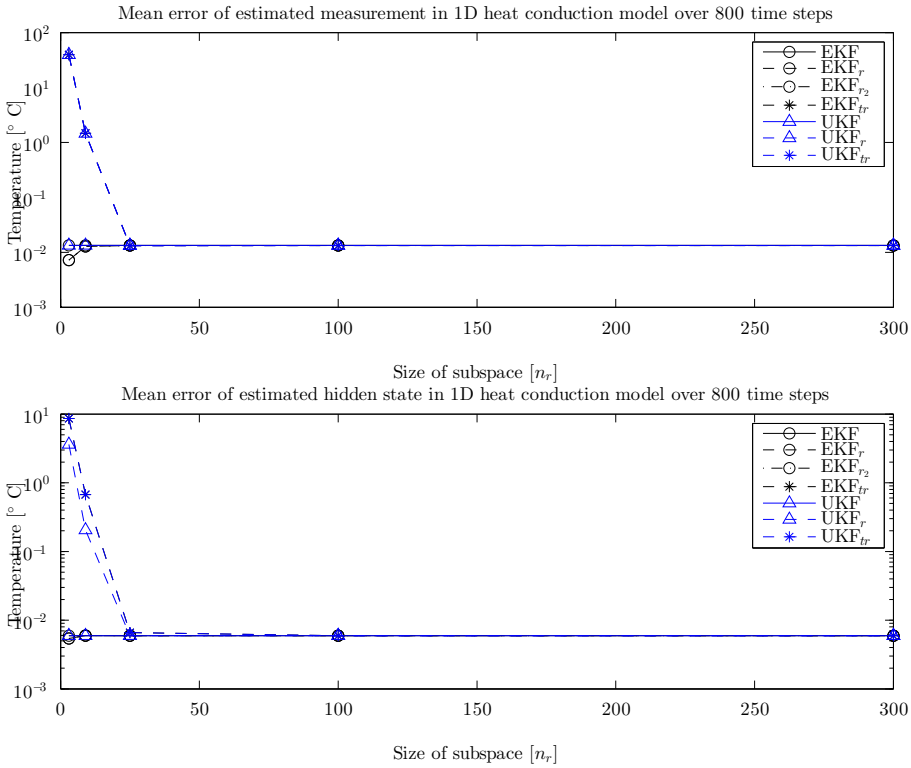

Figure 6.7: Absolute mean error [°C] between the true and estimated states for the slab with white noise added to the process and a constant bias of 15°C added on the measurement equation. In addition, the covariance matrix is modified to better correlate with the constant bias. The data is gathered through simulation over 800 time steps. The upper subplot represent the data for the mean error between the estimated and the true measured state, while the lower represent the data for the mean error between the estimated and the hidden true state. Both of which show great estimation results. Note that the EnKF is excluded due to poor estimation and that the y-axes are scaled logarithmically.

Table 6.5: Absolute mean error [°C] between the true and estimated states for the slab with white noise added to the process and a constant bias of 15°C added on the measurement equation. In this simulation the covariance matrix is modified to better correlate with the constant bias. I) represent the data for the mean error between the estimated and true measured state, while II) represent the data for the mean error between the estimated and true "hidden" state. There is a clear difference of the reduced algorithms for small subspace configurations, but the values converge to the mean error of the original algorithms as the subspace increase.

		Size of subspace $[n_r]$				
		3	**9**	**25**	**100**	**300**
I)	**EKF**♠	*1.334E-2*	*1.334E-2*	*1.334E-2*	*1.334E-2*	*1.334E-2*
	EKF$_r$	7.187E-3	1.276E-2	1.330E-2	1.331E-2	1.333E-2
	EKF$_{r_2}$	7.187E-3	1.276E-2	1.330E-2	1.331E-2	1.333E-2
	EKF$_{tr}$	39.6058	1.4672	1.324E-2	1.331E-2	1.333E-2
	UKF♠	*1.334E-2*	*1.334E-2*	*1.334E-2*	*1.334E-2*	*1.334E-2*
	UKF$_r$	39.4886	1.4603	1.324E-2	1.331E-2	1.333E-2
	UKF$_{tr}$	39.6058	1.4672	1.324E-2	1.331E-2	1.333E-2
	EnKF♠	*1.3166E+4*	*1.3166E+4*	*1.3166E+4*	*1.3166E+4*	*1.3166E+4*
II)	**EKF**♠	*5.933E-3*	*5.933E-3*	*5.933E-3*	*5.933E-3*	*5.933E-3*
	EKF$_r$	5.400E-3	5.982E-3	5.934E-3	5.933E-3	5.933E-3
	EKF$_{r_2}$	5.400E-3	5.982E-3	5.934E-3	5.933E-3	5.933E-3
	EKF$_{tr}$	8.6041	0.6740	6.599E-3	5.933E-3	5.933E-3
	UKF♠	*5.933E-3*	*5.933E-3*	*5.933E-3*	*5.933E-3*	*5.933E-3*
	UKF$_r$	3.6226	0.2054	5.934E-3	5.933E-3	5.933E-3
	UKF$_{tr}$	8.6041	0.6740	6.599E-3	5.933E-3	5.933E-3
	EnKF♠	*8.7683E+3*	*8.7683E+3*	*8.7683E+3*	*8.7683E+3*	*8.7683E+3*

♠ Constant data, independent of the subspace n_r

in the EnKF did not show any improvements, leading to the conclusion that a systematic error on the measurements will heavily affect the performance of the EnKF, and requires extensively tuning of the covariance matrix.

6.5 Model error

The model error is added to the system matrix $\mathbf{A}$ in the one dimensional heat conduction model (4.2) by adding a percentage p of the existing diagonal $\mathbf{A}_{\text{diag}}$ to the diagonal, i.e.,

$$\mathbf{A}_{me} = \mathbf{A} + p \cdot \mathbf{A}_{\text{diag}}. \tag{6.1}$$

The simulation is conducted with three different percentages to fully see the development of the estimation results as the error increase. During the simulation the process noise is set to zero, such that the only discrepancy in the model is the change in the system matrix. As before, the study will only be conducted for the one dimensional model, due to similar behavior of the two models. Figure 6.8 illustrates the data of the absolute mean error from the simulations, where the upper, middle and lower row represent the percentages $p = 10\%, 50\%$ and 100%, respectively, and the columns represent the measured (left) and non-measured state (right). The measured state show similar numerical results in all three simulations as one would expect, besides the spike that occurs in the derived EKFs in the lower left subplot. As before, the mean error in the measured state of the reduced algorithms increase up towards the value of the EKF and UKF as the subspace increase. The estimates of the non-measured state do not have a persistent estimation result as the model error increase. Studying the right column of subplots in Figure 6.8, it is obvious that the derived EKFs and the traditionally reduced algorithms have a substantial increase in the mean error as the model error increase. The EnKF, the original EKF and UKF, and the derived UKF show the same estimation result for each of the three different errors, yielding a mean error smaller than 1°C. Thus, the derived UKF is the most promising algorithm, showing good results for subspace configurations smaller than the full space.

As the elements on the diagonal of the original system matrix $\mathbf{A}$ are quite small, a experiment of adding a multiple of the identity to the system matrix, i.e.,

$$\mathbf{A}_{me_2} = \mathbf{A} + 0.3 \cdot a_P \mathbf{I},$$

is conducted. The extra contribution to the system matrix is significantly larger than the previous simulation (6.1), with a contribution in the order of 10^5 (the value of a_P is relatively large, see Section 4.2), which makes the parameter on the diagonal much larger than the true process parameter. The results are given in Figure 6.9, where only the EKF, EnKF, UKF and UKF$_r$ are represented. This is due to that the derived EKFs and traditionally reduced algorithms have

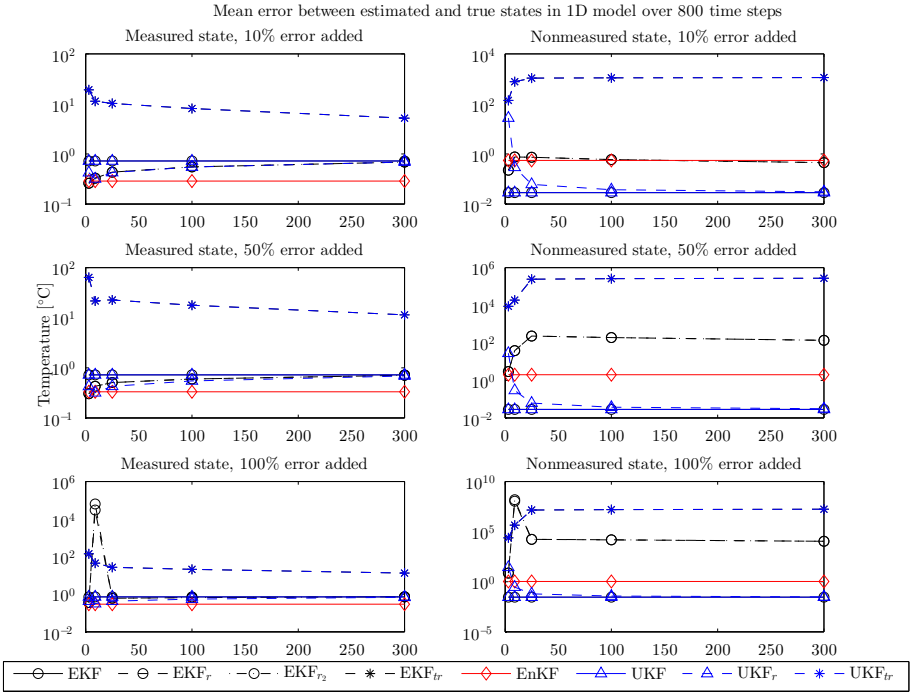

Figure 6.8: Absolute mean error [°C] between the true and estimated states for the slab added model errors, gathered through simulation over 800 time steps. The left column represent the mean error of the measured state and its estimates, while the right represents the mean error of the non-measured state and its estimates. The upper row is the simulation data when added a 10% ($p = 0.1$) error on the diagonal of the system matrix. The middle and lower row represent an error of 50% and 100%, respectively. Note that all y-axis of the subplots are logarithmic scaled for better representation.

estimation results that yield `NaN`[1] or `Inf`[2]. These results are due to the large contribution from the model error, which consequently leads to large values in the estimated covariance matrices. This affects both the calculation of the estimated states, and even makes the modified Cholesky factorization to loop

[1] Not a Number - occurs for example when dividing by zero.
[2] Infinity

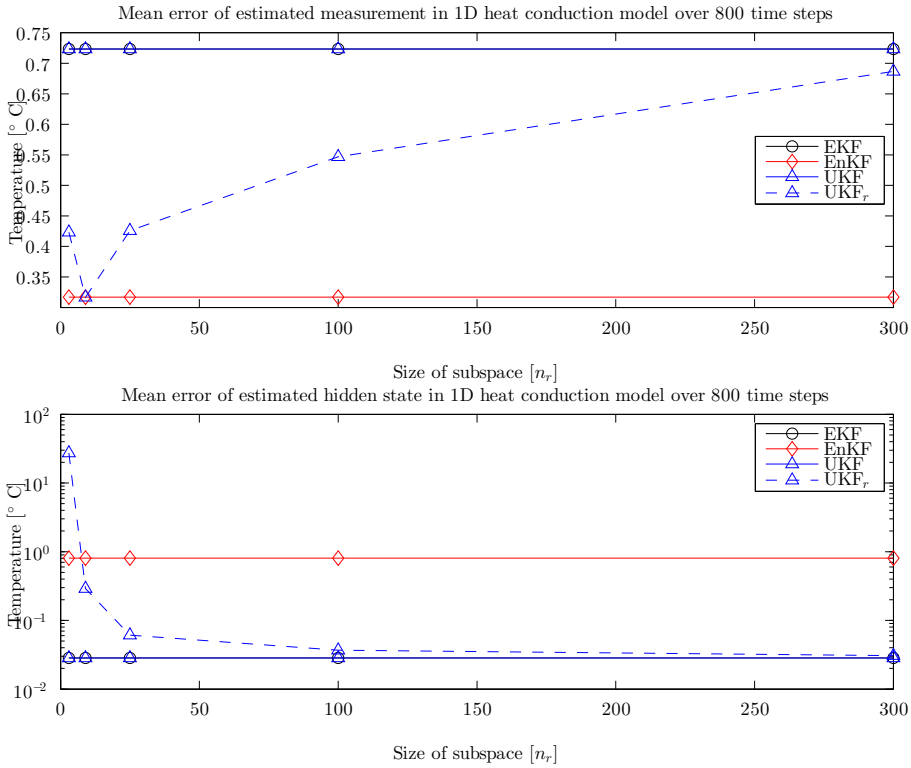

Figure 6.9: Absolute mean error [°C] between the true and estimated states for the slab added a large model error on the diagonal of the system matrix, and white noise as disturbances on the measurement equation. The data is gathered through simulation over 800 time steps. The upper subplot represent the data for the mean error between the estimated and the true measured state, while the lower represent the data for the mean error between the estimated and the hidden true state. Some of the algorithms have been excluded due to `Inf` or `NaN` in the estimation result. The behavior and development of the represented algorithms are exactly as described in the previous sections. Note that the y-axis of the lower subplot is logarithmic scaled.

infinitely. The non-measured state, represented by a logarithmic scale on the y-axis, is by all of the state estimators presented in the figure estimated close to

its true state. The results are identical to the previous simulation with smaller contributions to the diagonal (6.1), further enhancing the promise of the derived algorithm.

6.6 Balanced truncation

To not be limited strictly to a reduced-order basis obtained from POD, the method of balanced truncation from Section 2.2.4 is incorporated. The first simulations are conducted by applying Gaussian white noise to the measurement equation, in order to make a direct comparison to the results from Section 6.3.1. The absolute mean error between the true and estimated states are given in Figure 6.10, where a quick glance at the figure may give the impression that the figure or the simulations are incomplete. The missing data is related to that the reduced algorithms produce numerical values of `NaN` or `Inf`. However, evaluating the presented results show that the derived EKFs produce, unlike the other simulations, an identical behavior for both the measured and non-measured state. That is, the mean error rapidly grows in magnitude as the subspace increase, and while the estimation result is quite good for small subspaces, the last valid subspace configuration yields a worse result than their full original algorithm for the non-measured state. Extending the plotted lines in the figure would show a exponential growth of the mean error, given that the algorithms could produce valid results for higher subspaces. The product of the derived UKF yields zero at every time step, except for the first time step due to the initial conditions. The traditionally reduced algorithms differs in numerical results for the first time, where EKF_{tr} provides valid results for a subspace configuration up to $n_r = 9$, while UKF_{tr} also gives results for the next subspace configuration at $n_r = 25$ before starting to yield invalid numbers. However, they produce states that are close to zero at each time step (similar to the derived UKF), and should therefore not be considered as an option. Thus, when using balanced truncation for model reduction, the subspace configuration in the reduced EKFs should be kept at a minimum.

Introducing a systematic error instead of Gaussian white noise in the measurement equation, with the initial configuration of the measurement covariance matrix as described in Section 5.2, show that the derived UKF and traditionally reduced algorithms exhibit a behavior equal to the balanced Gaussian case, i.e., yielding estimates that are equal to zero at every time step. The original algorithms show the behavior seen in Section 6.4, correlating to the measurement, while the derived EKFs correlate with the true states, as shown in Figure

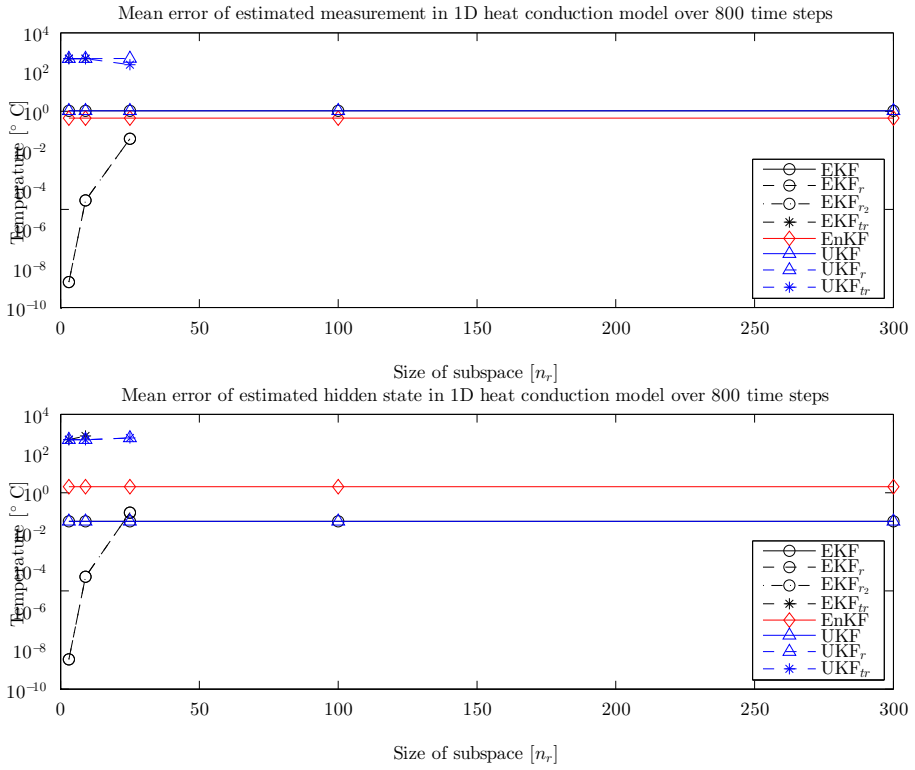

Figure 6.10: Absolute mean error [°C] between the true and estimated states for the slab with random Gaussian white noise added on both the process and the measurement equation. The reduced basis is obtained by balanced truncation, and the data is gathered through simulation over 800 time steps. The upper subplot represent the data for the mean error between the estimated and the true measured state, while the lower represent the data for the mean error between the estimated and the hidden true state. The clipping of the lines in the figure is due to that the estimation algorithms yielded `Inf` or `NaN` at given subspaces. This can be illustrated by considering the derived EKFs and extending the lines, which should give an exponential behavior. Note that both y-axis in the plots are logarithmic scaled.

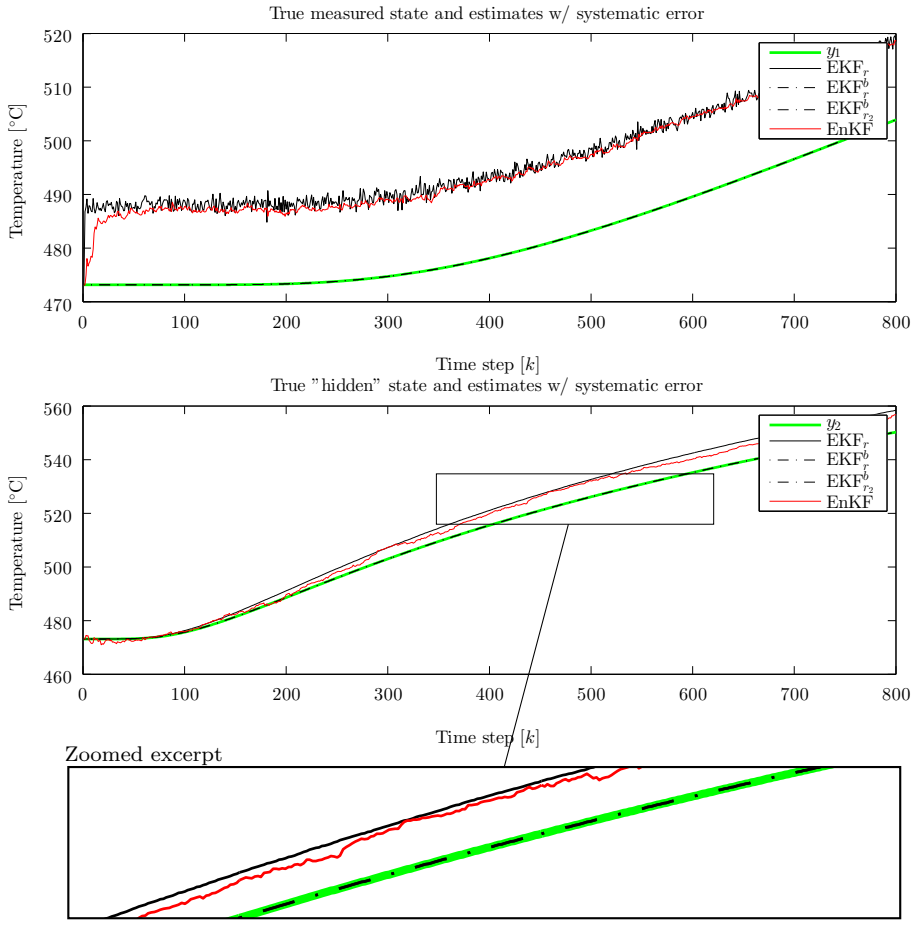

Figure 6.11: Comparison of the measured and estimated state for the slab with a systematic error of 15°C added. The reduced basis ($n_r = 3$) is obtained through balanced truncation. The upper and lower subplot show the true measured and non-measured state, i.e., without noise, and their estimates. The upper subplot show that the original algorithms produce a estimate correlated with the measurement, while the derived, balanced, algorithms correlate with the true state. The derived algorithms reproduce the non-measured state near perfectly, while the original algorithms have an increasing deviation from the true state, as shown in the zoomed excerpt at the bottom of the figure. The derived UKF and traditionally reduced approaches are omitted due to no relevant data.

6.11. The correlation with the true states is an highly unexpected behavior, especially considering that the covariance matrices are uncorrelated with the noise. In fact, changing the covariance matrices to correlate with the constant bias do not change the results in the derived EKFs. This can be a indication of that the balanced approach is less sensitive of noise, and the choice of the covariance matrices may be uncorrelated with the disturbances. However, it should be noted that this is only valid for the same small subspaces as before, as the algorithms produce numerical values of `NaN` or `Inf` at higher subspaces.

It should be noted that when the balanced truncation is applied to the full model, the unbalanced and balanced model does not yield the same characteristics as they should. The method has during implementation been tested to "textbook" examples, where it produced desired results. It is therefore assumed that the problem is related to the model, or closely related to the magnitude of the empirical Gramians, which have elements with magnitudes in the order of 10^9. The need for considering another balancing approach may therefore be necessary.

6.7 Uncertainties

This section presents a few points that should be addressed, as these can be considered as uncertainties in the simulation and estimation results.

1. **The process and measurement noise is correlated with the error covariance matrices**
   This is considered to be not entirely realistic, as the covariance matrices seldom are fully correlated with the noise. Based on this, the state estimators in this thesis may have produced better estimation results than what one would expect in a realistic simulation with an real physical model, where the noise is considered a unknown factor. This is mainly directed at the case of Gaussian white noise and the one dimensional model, as the Gaussian white noise for the two dimensional model had a varying magnitude at different areas throughout the plate, which is a more realistic approach.

2. **The numerical differentiation may yield incorrect numerical values**
   The difference quotient used in this thesis is only perturbed in one direction, possibly making it less exact and stable than similar approaches that are perturbed in more than one dimension. Even though the model is

discretized linear, making the result of the numerical differentiation equal to the system matrices, the differentiation could experience numerical deviations for large systems, which have been the case during testing.

3. **The results from the EnKF could have been better (or worse)** The estimation results of the EnKF are directly dependent on the size of the ensemble and the perturbations added to the ensemble. In this thesis, there are two factors that affects this result:

   - A fixed size of 10 and 50 for the one and two dimensional model, respectively, is applied. Thus, both the complexity and the estimation result could have had a different outcome if other options were considered.
   - The perturbations on the ensemble are added by multiplying the process noise covariance matrix $\mathbf{Q}$ with a sequence of normally distributed pseudorandom numbers. The perturbations, and hence, the product of the algorithm, are therefore not necessarily optimal, as the estimation results are dependent on the sequence of generated numbers.

7 Conclusion and recommendations

In this thesis, a new approach of improving the efficiency in the EKF and UKF has been proposed. The approach focus on reducing the main bulk of complexity, located in the covariance propagation in the EKF and in the calculation of the sigma points in the UKF, through the use of techniques from model reduction, while the full model is used for state propagation and the unscented transformation, respectively.

The POD has been used as the preferred model reduction technique, as this method minimize the error over the entire domain. The POD basis functions are derived from simulation data, as shown in Chapter 4. The validity of the reduced-order model will depend on the quality of the data set. If the data set sufficiently represents the typical variations of the system, the reduced-order model will be valid for simulations within this operating range.

The time complexity of the new approach has been shown to be similar to the traditionally reduced approaches of state estimation, and shown a significantly improvement over the original algorithms for given subspace configurations. The new derived approach has the advantage of using the reduced model solely in the bulk of complexity, while sharing the same complexity of the traditional (existing) approaches. However, when exceeding a subspace configuration of approximately 60-70% of the full space, it is shown that the complexity of the derived approach is computationally *more* expensive than the original algorithms.

The heat conduction models have been simulated under different induced errors, such as Gaussian white noise and a systematic error on the measurement equation, in addition to adding model errors to the system equation. The new approach for the EKF has shown great promise for a wide range of subspace configurations substantially smaller than the full space, except for in the case of induced model error, where the estimation result was shown to increase along with the size of the error. The derived UKF has in general shown great promise for subspace configurations larger than one sixteenth of the full space, and for *all* subspace configurations in the case of model errors, independent of the magnitude of the added error.

When using balanced truncation through empirical Gramians, the derived EKF has shown satisfying estimation results for subspace configurations smaller

than one sixteenth of the full space, as it encounters numerical problems at larger subspaces. The derived UKF should not be used when balanced truncation is considered the preferred method of model reduction, as it yields a output of zero for the entire simulation.

Although the thesis focus on homogeneous models with slow dynamics, the new approach shows great promise for improving the efficiency in the EKF and UKF through model reduction, without losing too much information in the process.

7.1 Recommendations for future research

The following are recommendations for future research:

1. **Introduce and apply state estimators to other models/processes**
   This thesis has focused on two homogeneous heat conduction models, which share similar properties and dynamics. As the models are discretized CFD models, and thus considered linear, make the elements in the system matrices time independent. This makes the models less complex and makes in particular the numerical differentiation and parameter estimation significantly easier to calculate. To make an assessment of the impact of the numerical differentiation in the EKF, a nonlinear and more complex model should be introduced. The model should contain faster dynamics than the models used in this thesis and potentially have different dynamics in the states, allowing the measured and non-measured state(s) to differ. Using a more complex model will also make for a deeper and broader understanding of the estimation result, due to more complexity in the estimation of the covariance matrix and state vector. As this probably makes it necessary to perturb the numerical differentiation in more than one dimension, other differentiation techniques and the increase of computational complexity should be researched.

2. **Application of goal-oriented POD**
   The performance of the estimation results are related to the applied reduced-order basis. This thesis has mainly been concerned with the use of POD, barely touching onto the subject of balanced truncation through empirical Gramians due to restricted time. The goal-oriented POD is a reduction algorithm based on an optimization problem seeking the best orthonormal basis, given a minimizing criteria and constraints. For example, if the estimated model has the objective to estimate one specific state, as this

state is considered more important than the others, the minimizing criteria in the objective function can be altered to fully fit the desired purpose. As this is more specific (goal-oriented) than the POD, which minimizes the error over the entire domain, it could potentially obtain a better basis for the desired purpose. However, the optimization problem can become quite large, making the offline cost of solving it computationally expensive, which must be accounted for.

3. **Analysis of reduced models**
   Throughout the thesis the state estimators are evaluated and analyzed by the estimation results, with no preliminary conditions on the reduced models. In future research, conditions such as robustness and stability will be beneficial to determine the validity range of the reduced models and to be able to guarantee stable reduced order models.
   The change of physical parameters in nonlinear systems may cause the POD basis functions to not approximate the new system correctly. For example, the validity of the model, given that the parameters are within some limitations, is important for the operators to decide whether the reduced order model is adequate or not. Also, given that the original model is internally stable, the POD have no guarantee that the reduced model will be stable. [Prajna, 2003] gives a sufficient condition for preserving stability for POD based reduced order models, and proposes a stability-preserving POD model reduction scheme. Since the models are used for predictions, it is desirable that the stability is preserved in the reduced-order model.

4. **Adaptive POD basis functions**
   As proposed in [Astrid, 2004], the POD basis functions are only guaranteed to be valid for the segments of snapshot collected. It is therefore proposed using an adaptive scheme for updating the POD basis functions by neural network algorithm. This could potentially increase the validity range of the reduced order model, which is of particular interest.

5. **Reduced sigma point filters**
   This thesis have not been focused on the reduced sigma point filters and, hence, only studied and briefly compared the time complexity of these filters. Further research should include a study on the estimation result of these filters, which should include both the filters as they are, in addition to reducing these further by the techniques presented in this thesis. The study should be conducted to see if the reduced sigma point filters can pro-

duce better, or equal, estimation results than the new derived algorithms, due to their low computational time.

A Reduced filters

This appendix presents the new derived filters discussed in Section 3.4 in full. The necessary changes are emphasized by red font color, which indicates either the need for projecting or reconstructing the full model to obtain the correct dimensions. Two derived versions of the EKF and one version of the UKF is given.

A.1 Projected EKF, full covariance matrix

$$\mathbf{F}_{r,k-1} = \left.\frac{\partial f_{r,k-1}}{\partial \mathbf{x}_r}\right|_{\boldsymbol{\Phi}_r^{\mathrm{T}}\hat{\mathbf{x}}_{k-1}^{+}}, \tag{A.1a}$$

$$\mathbf{L}_{r,k-1} = \left.\frac{\partial f_{r,k-1}}{\partial \mathbf{w}_r}\right|_{\boldsymbol{\Phi}_r^{\mathrm{T}}\hat{\mathbf{x}}_{k-1}^{+}}, \tag{A.1b}$$

$$\mathbf{F}_{k-1}^{\star} = \left(\boldsymbol{\Phi}_r^{\mathrm{T}}\right)^{\dagger}\mathbf{F}_{r,k-1}\boldsymbol{\Phi}_r^{\dagger} \tag{A.1c}$$

$$\mathbf{L}_{k-1}^{\star} = \left(\boldsymbol{\Phi}_r^{\mathrm{T}}\right)^{\dagger}\mathbf{L}_{r,k-1}\boldsymbol{\Phi}_r^{\dagger}, \tag{A.1d}$$

$$\mathbf{P}_k^{-} = \mathbf{F}_{k-1}^{\star}\mathbf{P}_{k-1}^{+}\mathbf{F}_{k-1}^{\star^{\mathrm{T}}} + \mathbf{L}_{k-1}^{\star}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\star^{\mathrm{T}}}, \tag{A.1e}$$

$$\hat{\mathbf{x}}_k^{-} = f\left(\hat{\mathbf{x}}_{k-1}^{+}, \mathbf{u}_{k-1}\right), \tag{A.1f}$$

$$\mathbf{H}_{r,k} = \left.\frac{\partial h_{r,k}}{\partial \mathbf{x}_r}\right|_{\boldsymbol{\Phi}_r^{\mathrm{T}}\hat{\mathbf{x}}_k^{-}}, \tag{A.1g}$$

$$\mathbf{M}_{r,k} = \left.\frac{\partial h_{r,k}}{\partial \mathbf{v}_r}\right|_{\boldsymbol{\Phi}_r^{\mathrm{T}}\hat{\mathbf{x}}_k^{-}}, \tag{A.1h}$$

$$\mathbf{H}_k^{\star} = \mathbf{H}_{r,k}\boldsymbol{\Phi}_r^{\dagger}, \tag{A.1i}$$

$$\mathbf{M}_k^\star = \mathbf{M}_{r,k}\boldsymbol{\Phi}_r^\dagger, \tag{A.1j}$$
$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{\star\mathrm{T}} \left(\mathbf{H}_k^\star \mathbf{P}_k^- \mathbf{H}_k^{\star\mathrm{T}} + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^\mathrm{T}\right)^{-1}, \tag{A.1k}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{y}_k - h_k\left(\hat{\mathbf{x}}_k^-, \mathbf{u}_k\right)\right], \tag{A.1l}$$
$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^-. \tag{A.1m}$$

A.2 Projected EKF, reduced covariance matrix

$$\mathbf{F}_{r,k-1} = \left.\frac{\partial f_{r,k-1}}{\partial \mathbf{x}_r}\right|_{\boldsymbol{\Phi}_r^\mathrm{T}\hat{\mathbf{x}}_{k-1}^+}, \tag{A.2a}$$
$$\mathbf{L}_{r,k-1} = \left.\frac{\partial f_{r,k-1}}{\partial \mathbf{w}_r}\right|_{\boldsymbol{\Phi}_r^\mathrm{T}\hat{\mathbf{x}}_{k-1}^+}, \tag{A.2b}$$
$$\mathbf{P}_{r,k}^- = \mathbf{F}_{r,k-1}\boldsymbol{\Phi}_r^\mathrm{T}\mathbf{P}_{k-1}^+\boldsymbol{\Phi}_r\mathbf{F}_{r,k-1}^\mathrm{T} + \mathbf{L}_{r,k-1}\boldsymbol{\Phi}_r^\mathrm{T}\mathbf{Q}_{k-1}\boldsymbol{\Phi}_r\mathbf{L}_{r,k-1}^\mathrm{T}, \tag{A.2c}$$
$$\hat{\mathbf{x}}_k^- = f\left(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}\right), \tag{A.2d}$$
$$\mathbf{H}_{r,k} = \left.\frac{\partial h_{r,k}}{\partial \mathbf{x}_r}\right|_{\boldsymbol{\Phi}_r^\mathrm{T}\hat{\mathbf{x}}_k^-}, \tag{A.2e}$$
$$\mathbf{M}_{r,k} = \left.\frac{\partial h_{r,k}}{\partial \mathbf{v}_r}\right|_{\boldsymbol{\Phi}_r^\mathrm{T}\hat{\mathbf{x}}_k^-}, \tag{A.2f}$$
$$\mathbf{H}_k^\star = \mathbf{H}_{r,k}\boldsymbol{\Phi}_r^\dagger, \tag{A.2g}$$
$$\mathbf{M}_k^\star = \mathbf{M}_{r,k}\boldsymbol{\Phi}_r^\dagger, \tag{A.2h}$$
$$\mathbf{P}_k^{\star^-} = \left(\boldsymbol{\Phi}_r^\mathrm{T}\right)^\dagger \mathbf{P}_{r,k}^- \boldsymbol{\Phi}_r^\dagger \tag{A.2i}$$
$$\mathbf{K}_k = \mathbf{P}_k^{\star^-} \mathbf{H}_k^{\star\mathrm{T}} \left(\mathbf{H}_k^\star \mathbf{P}_k^{\star^-} \mathbf{H}_k^{\star\mathrm{T}} + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^\mathrm{T}\right)^{-1}, \tag{A.2j}$$
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left[\mathbf{y}_k - h_k\left(\hat{\mathbf{x}}_k^-, \mathbf{u}_k\right)\right], \tag{A.2k}$$
$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^{\star^-}. \tag{A.2l}$$

A.3 Projected Unscented Kalman Filter

Sigma points

$$\hat{\mathbf{x}}_{r,k-1}^{(i)} = \boldsymbol{\Phi}_r^{\mathrm{T}} \hat{\mathbf{x}}_{k-1}^{+} + \tilde{\mathbf{x}}_r^{(i)}, \qquad i = 1, \dots, 2n_r, \tag{A.3a}$$

where (A.3b)

$$\tilde{\mathbf{x}}_r^{(i)} = \left(\sqrt{n_r \boldsymbol{\Phi}_r^{\mathrm{T}} \mathbf{P}_{k-1}^{+} \boldsymbol{\Phi}_r} \right)_i^{\mathrm{T}}, \qquad i = 1, \dots, n_r, \tag{A.3c}$$

$$\tilde{\mathbf{x}}_r^{(n_r+i)} = -\left(\sqrt{n_r \boldsymbol{\Phi}_r^{\mathrm{T}} \mathbf{P}_{k-1}^{+} \boldsymbol{\Phi}_r} \right)_i^{\mathrm{T}}, \quad i = 1, \dots, n_r, \tag{A.3d}$$

$$\hat{\mathbf{x}}_k^{(i)} = f\left(\boldsymbol{\Phi}_r \hat{\mathbf{x}}_{r,k-1}^{(i)}, \mathbf{u}_k, \mathbf{v}_k \right), \tag{A.3e}$$

$$\hat{\mathbf{x}}_k^{-} = \frac{1}{2n_r} \sum_{i=1}^{2n_r} \hat{\mathbf{x}}_k^{(i)}, \tag{A.3f}$$

$$\mathbf{P}_{r,k}^{-} = \frac{1}{2n_r} \boldsymbol{\Phi}_r^{\mathrm{T}} \left(\sum_{i=1}^{2n_r} \left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^{-} \right) \left(\hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^{-} \right)^{\mathrm{T}} + \mathbf{Q}_{k-1} \right) \boldsymbol{\Phi}_r, \tag{A.3g}$$

Sigma points

$$\hat{\mathbf{x}}_{r,k}^{(i)} = \boldsymbol{\Phi}_r^{\mathrm{T}} \hat{\mathbf{x}}_{k-1}^{-} + \tilde{\mathbf{x}}_r^{(i)}, \qquad i = 1, \dots, 2n_r, \tag{A.3h}$$

where (A.3i)

$$\tilde{\mathbf{x}}_r^{(i)} = \left(\sqrt{n_r \mathbf{P}_{r,k}^{-}} \right)_i^{\mathrm{T}}, \qquad i = 1, \dots, n_r, \tag{A.3j}$$

$$\tilde{\mathbf{x}}_r^{(n_r+i)} = -\left(\sqrt{n_r \mathbf{P}_{r,k}^{-}} \right)_i^{\mathrm{T}}, \qquad i = 1, \dots, n_r, \tag{A.3k}$$

$$\hat{\mathbf{y}}_k^{(i)} = h\left(\boldsymbol{\Phi}_r \hat{\mathbf{x}}_{r,k}^{(i)}, \mathbf{u}_k, \mathbf{v}_k \right), \tag{A.3l}$$

$$\hat{\mathbf{y}}_k = \frac{1}{2n_r} \sum_{i=1}^{2n_r} \hat{\mathbf{y}}_k^{(i)}, \tag{A.3m}$$

$$\mathbf{P}_y = \frac{1}{2n_r} \sum_{i=1}^{2n_r} \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k \right) \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k \right)^{\mathrm{T}} + \mathbf{R}_k, \tag{A.3n}$$

$$\mathbf{P}_{xy} = \frac{1}{2n_r} \sum_{i=1}^{2n_r} \left(\boldsymbol{\Phi}_r \hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k^{-} \right) \left(\hat{\mathbf{y}}_k^{(i)} - \hat{\mathbf{y}}_k \right)^{\mathrm{T}}, \tag{A.3o}$$

$$\mathbf{K}_k = \mathbf{P}_{xy}\mathbf{P}_y^{-1}, \tag{A.3p}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right), \tag{A.3q}$$

$$\mathbf{P}_k^+ = \left(\boldsymbol{\Phi}_r^{\mathrm{T}}\right)^{\dagger}\mathbf{P}_{r,k}^-\boldsymbol{\Phi}_r^{\dagger} - \mathbf{K}_k\mathbf{P}_y\mathbf{K}_k^{\mathrm{T}}. \tag{A.3r}$$

B Matrix square root and positive definiteness

This appendix introduce the conditions of positive definite matrices and the matrix square root.

B.1 Symmetric and positive definite matrices

Symmetric and positive definite matrices are in [Nocedal and Wright, 2006a] defined as

Definition B.1.1. A matrix $\mathbf{A}$ is symmetric if

$$\mathbf{A} = \mathbf{A}^{\mathrm{T}},$$

where $\mathbf{A}^{\mathrm{T}}$ denotes the transpose of $\mathbf{A}$. ■

Definition B.1.2. A matrix $\mathbf{A}$ is positive definite if

$$\exists\ \alpha \in \mathbb{R} > 0 \mid \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} \geq \alpha \mathbf{x}^{\mathrm{T}}\mathbf{x},\ \forall \mathbf{x} \in \mathbb{R}^n.$$

■

Positive definite matrices have the properties that all eigenvalues are strictly positive, where there exist an unique lower triangular matrix $\mathbf{L}$ from the Cholesky factorization.

B.2 Cholesky factorization

While the definition of the square of a matrix is not a given standard, a common notation given in [Simon, 2006a] is $\mathbf{P} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, instead of the scalar definition $\mathbf{P} = \mathbf{L}^2$. A positive definite matrix always has a square root, and since there may exist more than one solution, the square root is not necessarily unique.

The sigma point filters requires the square root of the covariance matrix in the calculation of the sigma points. In a comparison study of matrix square root

methods in the UKF by [Rhudy et al., 2011], where methods such as diagonalization, Schur-decomposition and Cholesky amongst others are compared, show that the Cholesky factorization gives the best compromise between performance and time, and is therefore used in the implementations and, thus, presented in this thesis. As given in [Simon, 2006a], the algorithm below computes an lower triangular matrix $\mathbf{L}$ such that $\mathbf{P} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, where $\mathbf{P} \in \mathbb{R}^{n \times n}$.[1]

Algorithm 4: Cholesky Factorization

Input : Symmetric positive definite matrix $\mathbf{P}$.
Output: Lower triangular matrix $\mathbf{L}$.

for $i \leftarrow 1$ **to** n **do**
  $\mathbf{L}_{ii} = \sqrt{\mathbf{P}_{ii} - \sum_{j=1}^{i-1} \mathbf{L}_{ij}^2}$;
  for $j \leftarrow 1$ **to** n **do**
    $\mathbf{L}_{ji} = 0, \quad j < i;$
    $\mathbf{L}_{ji} = \frac{1}{\mathbf{L}_{ii}} \left(\mathbf{P}_{ji} - \sum_{k=1}^{i-1} \mathbf{L}_{jk}\mathbf{L}_{ik} \right), \quad j > i;$

In terms of covariance matrices, this is generally a condition that should apply. However, due to that the matrices are estimated, the estimation result may give rise to the problem of the matrices not being positive definite. Thus, it may be necessary to apply a modification to the matrices which ensures this.

B.2.1 Matrix modifications to obtain positive definiteness

A given matrix that is not positive definite, can according to [Nocedal and Wright, 2006b] be modified by one of the following approaches

- eigenvalue decomposition,
  - Given a matrix $\mathbf{A}$ with eigenvalues $\lambda_i < 0$, replace the negative values by a small value $\delta > 0$, or flip sign of the negative eigenvalues,
- adding a multiple of the identity,
  - Given a matrix $\mathbf{A}$, find a scalar $\tau > 0$, such that $\mathbf{A} + \tau\mathbf{I}$ is sufficiently positive definite,

[1] In `MATLAB` one may use the existing function `chol`.

- modified Cholesky factorization, or
  - Given a matrix $\mathbf{A}$, increase the diagonal elements encountered during the factorization (where necessary) to ensure sufficiently positiveness.
- modified symmetric indefinite factorization,
  - Perform a factorization similar to the Cholesky factorization.

to guarantee that the matrix is positive definite. For full details about the approaches, see the referral.

C Time complexity in reduced sigma point filters

This appendix is an extension to the result presented in Section 6.2, where the main focus will be on the time complexity of the reduced sigma point filters from Section 2.1.3.

The time complexity of the state estimation algorithms is illustrated in Figure C.1, where, in addition to the earlier discussed algorithms, the reduced sigma point filters are included. The time complexity associated with the SSUKF and DFEKF correlate with the theory presented earlier, as they only require the calculation of $n + 2$ and $n + 1$ sigma points, respectively, and are more efficient than the EKF and UKF. The derived version of the SSUKF and DFEKF yield similar behavior to the derived versions of the EKF and UKF (described in Section 6.2), where the time complexity increase along with the size of the subspace and exceeds their derived algorithm at a given subspace configuration due to the similarity transformations and the associated matrix operations. The time complexity of the derived SSUKF and DFEKF share approximately the same time complexity as the derived versions EKF_{r_2} and UKF_r. This is also the case for the traditionally reduced versions of the SSUKF and DFEKF at small subspace configurations, but these are more efficient at larger subspaces.

Comparing the full to reduced ratio between the algorithms indicate that the reduced sigma point filters (SSUKF, DFEKF) do not improve the efficiency as one would initially assume. Hence, the main benefit of using the SSUKF or DFEKF is concluded to lie in the use of the original version of the algorithms, when compared to the original EKF or UKF. There is no obvious reason for choosing the derived SSUKF or DFEKF above the derived EKF or UKF, based on that the SSUKF_r and DFEKF_r are assumed to have a greater loss of information due to the number of sigma points and necessary projections and reconstructions in the algorithms, in comparison to the derived EKF or UKF, while yielding the same computational complexity.

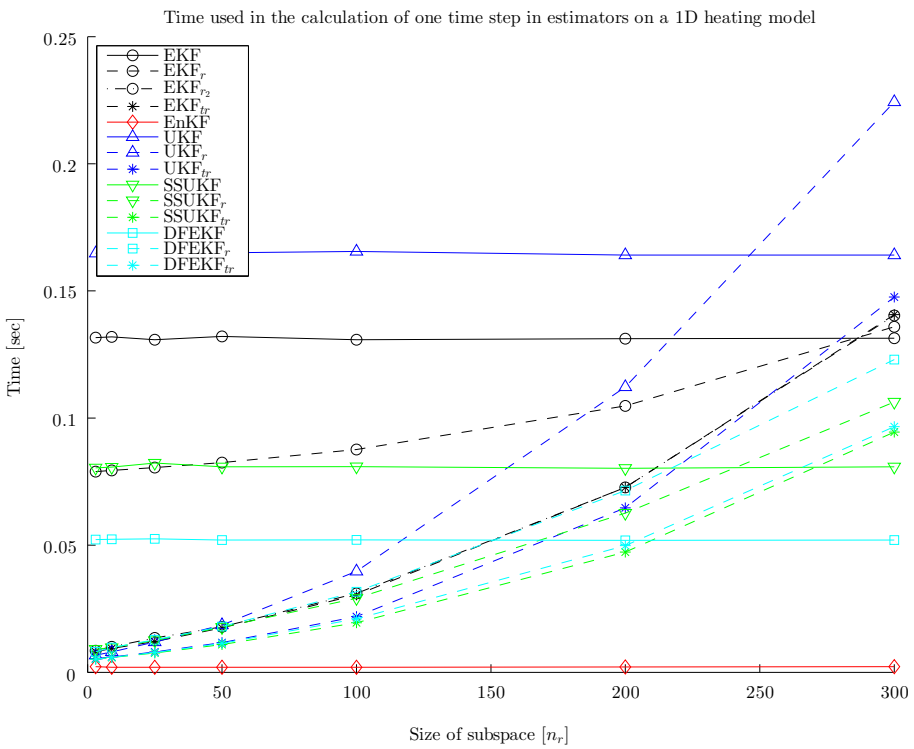

Figure C.1: The lines indicate how the different size of subspaces affects the computational time complexity in the algorithms. The time complexity of the full original algorithms (solid lines) are close to constant regardless of the size of the subspace (small variations are related to background processes during the simulation), while the projected algorithms (dashed and dashed-dot lines) increase up towards, and even above, the computational time of the original algorithms. This is related to the computational cost of multiplying the similarity transformation matrices, which increase with the dimension of the subspace. The algorithms marked by an asterisk is the "traditional" (reduced model in the entire algorithm) approach of the respective algorithms.

References

[Antoulas, 2005] Antoulas, A. C. (2005). *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control) (Advances in Design and Control)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[Astrid, 2004] Astrid, P. (2004). *Model Reduction for Process Simulations: A proper orthogonal decomposition approach*. PhD thesis, Technische Universiteit Eindhoven.

[Astrid et al., 2003] Astrid, P., Huisman, L., and Hazenberg, M. (2003). Low order modeling and optimal control design of a heated plate. In *2003, Proceedings of the European Control Conference*.

[Astrid et al., 2002] Astrid, P., Huisman, L., Weiland, S., and Backx, A. (2002). Reduction and predictive control design for a computational fluid dynamics model. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, pages 3378 – 3383 vol.3.

[Brown and Hwang, 1996] Brown, R. G. and Hwang, P. Y. C. (1996). The discrete kalman filter, state-space modeling, and simulation. In *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises*, chapter 5. John Wiley & Sons.

[Bui-Thanh et al., 2007] Bui-Thanh, T., Willcox, K., Ghattas, O., and Waanders, B. v. B. (2007). Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224(2):880–896.

[Burgers and et al., 1998] Burgers, G. and et al. (1998). On the analysis scheme in the ensemble kalman filter. *Monthly Weather Review*, 126:1719–1724.

[Burl, 1993] Burl, J. (1993). A reduced order extended kalman filter for sequential images containing a moving object. *Image Processing, IEEE Transactions on*, 2(3):285 –295.

[Evensen, 1994] Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162.

[Farhat, 2013] Farhat, C. (2013). Lecture notes in CME345: Model Reduction - Projection Based Reduced-Order Modeling. `http://www.stanford.edu/group/frg/course_work/CME345/CA-CME345-Ch3.pdf`, May 7, 2013.

[Farrell and Ioannou, 2001] Farrell, B. F. and Ioannou, P. J. (2001). State estimation using a reduced-order kalman filter. *Journal of the atmospheric sciences*, 58:3666–3680.

[Gillijns et al., 2006] Gillijns, S., Mendoza, O., Chandrasekar, J., Moor, B. D., Bernstein, D., and Ridley, A. (2006). What is the ensemble kalman filter and how well does it work? pages 4448–4453, Minneapolis, Minnesota, USA. IEEE American Control Conference. E-ISBN: 1-4244-0210-7.

[Golub and Van Loan, 1996] Golub, G. H. and Van Loan, C. F. (1996). *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

[Green and Limebeer, 1994] Green, M. and Limebeer, D. J. (1994). Model reduction by truncation. In *Linear Robust Control*, chapter 9. Prentice Hall.

[Hahn and Edgar, 2002a] Hahn, J. and Edgar, T. F. (2002a). Balancing approach to minimal realization and model reduction of stable nonlinear systems. *Industrial & Engineering Chemistry Research*, 41(9):2204–2212.

[Hahn and Edgar, 2002b] Hahn, J. and Edgar, T. F. (2002b). An improved method for nonlinear model reduction using balancing of empirical gramians. *Computers & Chemical Engineering*, 26(10):1379 – 1397.

[Hass et al., 2011] Hass, J. R. et al. (2011). Differentiation. In *University Calculus*, chapter 3. Pearson.

[Holmes et al., 1996] Holmes, P., Lumley, J. L., and Berkooz, G. (1996). Proper orthogonal decomposition. In *Turbulence, coherent structures, dynamical systems, and symmetry*, chapter 3. Cambridge University Press.

[Hovland, 2008] Hovland, S. (2008). *Model Reduction and Control in Computational Fluid Dynamics*. PhD thesis, NTNU.

[Julier, 2003] Julier, S. (2003). The spherical simplex unscented transformation. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2430 – 2434 vol.3.

[Julier and Uhlmann, 2002] Julier, S. and Uhlmann, J. (2002). Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *American Control Conference, 2002. Proceedings of the 2002*, volume 2, pages 887 – 892 vol.2.

[Julier et al., 2004] Julier, S. J., Jeffrey, and Uhlmann, K. (2004). Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422. Issue: 3.

[Kaasa et al., 2012] Kaasa, G.-O., Stamnes, . N., Imsland, L., and Aamo, O. M. (2012). Simplified hydraulics model used for intelligent estimation of downhole pressure for a managed-pressure-drilling control system. *SPE Drill & Completion*, 27(1):127–138.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45.

[Lall et al., 2002] Lall, S., Marsden, J. E., and Glavaški, S. (2002). A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535.

[Laub et al., 1987] Laub, A., Heath, M., Paige, C., and Ward, R. (1987). Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *Automatic Control, IEEE Transactions on*, 32(2):115 – 122.

[Lumley, 1967] Lumley, J. L. (1967). The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio propagation*, pages 166–178. edited by A.M. Yaglom and V.I. Tatarski, Moscow: Nauka.

[Nocedal and Wright, 2006a] Nocedal, J. and Wright, S. (2006a). Appendix a.1 - elements of linear algebra. In *Numerical optimization*, Springer series in operations research and financial engineering, chapter A.1, page 599. Springer, 2. ed. edition.

[Nocedal and Wright, 2006b] Nocedal, J. and Wright, S. (2006b). Newton's method with hessian modification. In *Numerical optimization*, Springer series in operations research and financial engineering, chapter 3, pages 48–56. Springer, 2. ed. edition.

[Prajna, 2003] Prajna, S. (2003). Pod model reduction with stability guarantee. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 5, pages 5254–5258 Vol.5.

[Quine, 2006] Quine, B. M. (2006). A derivative-free implementation of the extended kalman filter. *Automatica*, 42(11):1927 – 1934.

[Rhudy et al., 2011] Rhudy, M., Gu, Y., Gross, J., and Napolitano, M. R. (2011). Evaluation of matrix square root operations for ukf within a uav gps/ins sensor fusion application. *International Journal of Navigation and Observation*, 2001.

[Rowley, 2005] Rowley, C. (2005). Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013.

[Simon, 2006a] Simon, D. (2006a). The kalman filter. In *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, part ii 6, page 160. Wiley-Blackwell.

[Simon, 2006b] Simon, D. (2006b). Nonlinear kalman filtering. In *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, part iv 13. Wiley-Blackwell.

[Simon, 2007] Simon, D. (2007). Reduced order kalman filtering without model reduction. *Control and Intelligent Systems*, 35(2):169–174.

[Skogestad and Postlethwaite, 2005] Skogestad, S. and Postlethwaite, I. (2005). Model reduction. In *Multivariable Feedback Control*, chapter 11. Wiley-Interscience.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). The ukf algorithm. In *Probabilistic Robotic*, Intelligent Robotics and Autonomous Agents Series, chapter 3, pages 67–71. MIT Press.

[Wan and van der Merwe, 2000] Wan, E. and van der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153 –158.