Monitoring of Industrial Processes using Large Scale First Principles Models

Monitoring of Industrial Processes using Large Scale First Principles Models

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr ir J. Fokkema
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 21 december 2006 om 15:00 uur

door

Robert BOS

natuurkundig ingenieur
geboren te Papendrecht

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr ir P. M. J. Van den Hof

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr ir P. M. J. Van den Hof	Technische Universiteit Delft, promotor
dr ir X. J. A. Bombois	Technische Universiteit Delft
Prof. ir O. H. Bosgra	Technische Universiteit Delft
Prof. dr ir A. W. Heemink	Technische Universiteit Delft
Prof. Dr.-Ing. W. Marquardt	Rheinisch-Westfälische Technische Hochschule Aachen
Prof. dr ir G. van Straten	Wageningen Universiteit
dr ir S. Weiland	Technische Universiteit Eindhoven

dr ir X. J. A. Bombois heeft als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

The research has been supported by TNO Science and Industry.

ISBN-10: 90-8759-022-9
ISBN-13: 978-90-8759-022-2

Printed in the Netherlands by JouwBoek.nl.

Contents

Voorwoord

Toen ik in 2001 bijna was afgestudeerd wist ik het zeker: ik wilde graag promoveren. Toen Paul Van den Hof mij een functie als promovendus aanbood, heb ik dan ook niet lang getwijfeld. Nu, inmiddels aan het einde van het promotie traject, heb ik zeker geen spijt van mijn keuze. Paul, bedankt dat je mij deze kans hebt gegeven en dat je mijn promotor wilde zijn.

Behalve een promotor is een goede begeleider erg belangrijk. Ik heb het geluk gehad dat Xavier Bombois mijn begeleiding op zich heeft genomen. Xavier, bedankt voor de hulp die je me gedurende de afgelopen jaren hebt geboden. Onze inhoudelijke discussies hebben er zeker toe geleid dat de resultaten in dit proefschrift veel "strakker" zijn geworden.

Naast Paul en Xavier, ben ik ook de groepen Proces Fysica en Control Engineering van TNO Science and Industry danken voor de input tijdens de voortgangsvergaderingen.

Gedurende mijn vier jaar bij Model gebaseerd Meten en Regelen (MMR) en later het Delft Center for Systems and Control (DCSC) heb ik het uitstekend naar mijn zin gehad. Bij deze wilde ik alle collega's en studenten danken voor prettige werksfeer. In dit verband wil heb ik het bijzonder getroffen met mijn kamergenoten Stijn de Waele en Alex Kalbasenka.

Behalve het werk over het monitoren van behulp van grootschalige fysische modellen, heb ik het geluk gehad om ook aan een aantal andere interessante projecten te mogen werken. Met Piet Broersen heb ik verder mogen werken aan het schatten van tijdreeksmodellen uit onregelmatig bemonsterde data. Piet, bedankt dat je me betrokken hield in dit werk. Met Arjan den Dekker en Jan Sijbers heb ik mogen werken aan het zoeken naar hersenactiviteit in fMRI data. Arjan, Jan, bedankt dat jullie mij bij het fMRI project hebben betrokken.

Ten slotte wil ik ook familie en vrienden danken voor de getoonde interesse, ondersteuning en de nodige afleiding. Ik heb bijzonder wil ik Roeland de Bruijn, Michel Karsdorp, Jurjen Oskam en Ruth Prelicz bedanken voor het nalezen en controleren van het proefschrift.

Robert Bos
December, 2006.

Chapter 1

Introduction

1.1 Motivation and background

The optimal operation of industrial processes has become increasingly important in recent years. In the current global social and economic environment, in order to remain competitive, the process industry continuously needs to lower production costs, increase product quality and consistency, while adhering to ever stricter environmental norms. One possibility to reach these objectives is to optimize the process operation by exploiting all available knowledge about the process.

Efficient process operation requires good process design. Before being operated, the process has to be designed such that it has enough potential to meet the target specifications and requirements. Apart from the process design itself, the instrumentation should be sufficient to obtain the maximum performance from the process design. Enough sensors should be placed to measure process variables with enough accuracy such that the process can be monitored adequately. Enough actuators should be available placed at the right locations in order to be able to efficiently steer the process to its optimal working point and to reject any disturbances.

Once the process has been designed, it must be operated in an optimal manner in order to reach the required performance. A prerequisite for the optimal operation of a process is the availability of a monitoring tool.

In this thesis we shall define monitoring as online estimation of the state of the process using measured inputs and measured outputs of the process. System and control theory defines the state of the system as a time varying vector of fixed dimension that together with the process model and current inputs contains sufficient information to provide the best possible prediction for future process states and outputs [23]. The state vector has to be estimated since it is often impossible to measure it directly.

Apart from the fact that knowledge of the status of the process is valuable to process operators, monitoring enables techniques such as fault detection and isolation and advanced automatic control.

Fault detection attempts to detect deviations from normal process behavior. These deviations from normal process behavior are called faults. Fault detection only detects whether a fault has occurred. Faults can occur for any number of reasons. For instance

a fault could be caused by faulty sensors or faulty actuators. Fault isolation algorithms determine among a number of candidate faults, which fault has occurred. Fault detection and isolation algorithms are important for any automatic process control implementation. Obviously, if process operation is based on erroneous measurements or a bad process model, results will be far from optimal. For fault detection, knowledge of the state of a process is important, because the process state together with input of the process can be used to predict future process outputs. Many fault detection algorithms are based on the statistics of the difference between the predicted measurements and the actually obtained measurements.

Currently available advanced optimal control algorithms can be used to compute the required input signals to move the process to an optimal operation region as efficiently as possible, while rejecting process disturbances. Knowledge of the process state is also important for control applications. If the current state of the system is known, predictive control algorithms can be used to compute inputs for the process such that the predicted process behavior minimizes a predefined cost function, while adhering to the constraints of the process.

Summarizing, monitoring is an important tool that enables efficient process operation, because it provides operators with online information about the current state of the system and enables the use of advanced fault detection and isolation and advanced control algorithms. As already mentioned, monitoring consists of estimating the state vector. To perform this estimation online a state filter is used. A state filter indeed produces estimates of the current state of the process, using known inputs, sensor measurements and a process model.

We have established that to enable near optimal operation of an industrial process, an algorithm for state estimation is required. Apart from this requirement, the extent in which the optimal operation of the process can be obtained is also dependent on the plant model used for process operation.

With the recent increases in computer speed it has become possible to construct detailed first principles models, even for complex industrial processes. These models can provide an accurate description of the process under consideration. The states and parameters in these models have a direct physical interpretation. Simpler processes can sometimes be modelled with relatively simple models. To accurately model more complex industrial processes, very complex nonlinear models are generally required. The dimension of the state vector of these complex models is generally very large. The time required for simulation is often in the same order as the simulated interval. These last two properties have prevented the use of these complex models for state estimation and other online applications.

To circumvent some of the problems of overly complex first principles models of industrial processes, a different modelling approach has attracted attention. Instead of modelling the process using first principles relations specific to the process to be modelled, broadly applicable black-box model structures are considered to describe the process behavior. Parameters in the model structures are determined from experimental process data using identification techniques [56]. An advantage of these type of models is that the structure of the used black box models is often much simpler than

model structures that are obtained by first principles modelling. The computational burden of the models is low enough, so that they can be used for online state estimation, and other process applications. Unfortunately, because the model structures used are so general, it is difficult to assign a physical meaning to states and parameters of black box models. As a result, a black box model may provide reasonable description of a process' input to output behavior, it provides little to no insight in actual process conditions, limiting its use for process monitoring.

Currently the situation found in the process industry is that the simpler black box models are used for the daily operation of the plant, while complex first principles models are only used for offline simulation experiments and process design.

Improved results in process operation could likely be achieved if it would also be possible to use large scale first principles models in online process applications. The main reason for the expected increase in performance is that the state in a detailed first principles model represents a detailed physically interpretable description of the process. For fault detection and isolation applications, this means that more detailed analysis of unexpected process behavior is possible. For monitoring, using first principles models will give operators valuable detailed insights of current physical operating conditions. In control applications, using first principles models opens the possibility of not only controlling the process on an input-output level, but to use far more sophisticated control objectives based on the state of the system.

To enable the use of the large scale first principles models, it should at least be possible to construct a state filter and fault detection algorithms based upon these models.

1.2 State estimation using large scale first principles models

1.2.1 Introduction

As discussed earlier, the availability of a monitoring tool using a detailed first principles models can assist to optimize a process' behavior. As already mentioned, in order to estimate the state of a system online a state filter is required. Designing such a filter for complex detailed first principles models is far from trivial. To highlight some of the difficulties involved, we will first discuss some common properties of detailed first principles models in the process industry. Then we will present methods to design a state filter for relatively simple models. We will argue that popular state filters for simpler models are not feasible for more complex and detailed first principles models. Finally it will be shown that even specialized state filters for large scale models from related fields such as climatological modelling are not directly applicable.

1.2.2 General properties of first principles models of large scale industrial processes

Processes that are currently in use throughout the process industry are very diverse. However, most processes do have at least three properties in common. First of all, most processes in the process industry are MIMO (multiple input multiple output) processes. The number of input actuators and output measurements is often larger than five. Secondly the dynamic input to output behavior is generally nonlinear. This means that the process will react differently in different operating points. Finally, the sampling interval of industrial processes is in general in the order of a minute. Of course, processes with very fast behavior may be sampled faster and processes with very slow behavior may be sampled slower, but in general, the sample times encountered in the process industry are in the order of a minute.

First principles models of large scale industrial processes are often derived using partial differential equations (PDEs) describing the physical and chemical processes within the plant. These equations, together with appropriate boundary and initial conditions form an implicit model of the plant. For process monitoring and control, an explicit model is required. This model can be derived from the implicit PDE model using techniques such as finite elements or finite differences [68]. This technique imposes a fine grid in the spatial dimensions. In each grid element, the process conditions are assumed constant. Using this approximation, the original implicit PDE model can be approximated by an explicit state-space model consisting of a set of ordinary differential equations (ODEs). This ODE model contains an ODE for all process variables in each grid element. The state of the first principles model is a vector containing all the process variables for each grid element. To get an accurate approximation to the implicit PDE model, the grid needs to be very fine. Thus the explicit ODE model that should be used for monitoring and control applications will consist of a very high number of ODEs and have a very large state dimension. State dimensions $>> 10^5$ are not uncommon. When using this model for simulations, the time required to solve the ODEs is often in the same order as the time interval over which the process is simulated.

It should be noted that process models derived in this manner can often very accurately predict the undisturbed process behavior. However, models derived using the described methodology do not contain a description of the properties of possible process disturbances and how these disturbances might alter the behavior of the plant. Also, this type of modelling does not include a description of the errors that occur during the measurements.

In the next sections the difficulties in designing a state filter for these type of models will be discussed. It will be shown that in theory methods to estimate the state of relatively simple models could still be used to estimate the state of the described large-scale first principles models. In practice however, none of the known methods can be implemented due to computational issues associated with the use of detailed first principles process models.

1.2.3 General state estimation techniques for nonlinear models used in the process industry

The most common state filter currently used in the process industry is the EKF (Extended Kalman Filter) [3][42]. The EKF is a generalization of the celebrated Kalman filter [47]. The Kalman Filter is the optimal state estimator in the least squares sense for linear models if both the process disturbances and measurement errors are stochastic variables, with a known Gaussian distribution. The Kalman Filter consists of two steps. In the first step the next future state and output measurements are predicted, and for each prediction an error covariance matrix is computed. This step is often referred to as the time update. Once the next output measurement becomes available, the filter refines the previous state predictions using the difference between the predicted and actually observed output. This step is often referred to as the measurement update. For the computations in both the time update and the measurement update, the Kalman filter uses the assumption that the model is linear. For process models in general, this will not be the case. To overcome this problem the EKF has been developed. The EKF is a modified version of the Kalman filter in which the available model is linearized with respect to the state of the model during both the time update and the measurement update. After the linearization of the model, the usual Kalman equations can be used for both the time update and the measurement update. Since an analytical expression for the required linearization is usually unavailable, the required derivatives are computed numerically. Numerical approximation of the required derivative for first principles models requires at least a model evaluation for every state element. This procedure takes far longer than the sampling interval, thus making online implementation of the EKF impossible. Even if the linearization of the first principles models could be computed well within a sampling interval, there is a second problem preventing the online implementation of an EKF using a complex first principles model. The EKF requires manipulations with square covariance matrices which have the same dimension as the state of the model. For a lot of detailed models even storing these matrices in computer memory is a big problem. Performing manipulations on these matrices, such as inverting them, is practically impossible.

Even though the EKF is an obvious extension of the Kalman filter (which is optimal in the least squares sense for linear models), the EKF can produce state estimates that are far from optimal as the nonlinearities in the process behavior become more pronounced [78][82][98]. To better handle nonlinear models, several other filters have been developed, such as the UKF (Unscented Kalman Filter) and the MHE (Moving Horizon Estimator).

The UKF (Unscented Kalman Filter) [44] is a state estimator which is closely related to the EKF. Instead of computing the required state and output predictions and covariances using linearizations of the process model, the UKF computes the required predictions and associated covariance matrices by experimentally mapping specifically selected points through the process model using simulations. This procedure requires approximately two model evaluations per state element, process disturbance and measurement error. Compared to the EKF, the estimates produced by the UKF can be far more accurate, but computing the estimates requires approximately twice

the computational effort. As a result using the UKF as a state estimator using complex first principles models is computationally not feasible.

Another state filter that is used in the process industry is the MHE (Moving Horizon Estimator) [79]. The MHE computes the current state estimate as a solution of a least squares problem. For linear systems, it can be shown that the solution to this least squares problems is equivalent to the state estimate of the Kalman Filter. The advantage of the MHE approach is that the solution is computed using numerical optimization tools. These optimization tools can incorporate the physical constraints of the system to ensure that the resulting state estimate always has a physically meaningful result. Obvious limitation of MHE is that it requires solving a least squares problem involving the large scale first principles process model, which requires far more computational effort than computing state estimates using the EKF or UKF. To alleviate the computational burden of the required optimization problem, several methods using either sensitivity equations or adjoint models can be used [80][59]. For these techniques additional model equations need to be derived. Even with adjoint and sensitivity based techniques, the optimization problem in the MHE is far too complex to be solved within the sampling interval, thus preventing the practical application of the MHE using a large scale first principles process model.

The problems of implementing an online state filter can be partially addressed using projection based model reduction techniques. Examples of currently popular projection based techniques are POD (Proper Orthogonal Decomposition) and empirical nonlinear balancing [4][51]. These projection based model reduction techniques utilize the property that under normal process conditions the state of the process generally resides within a relatively low dimensional subspace of the high dimensional state-space. This property of most industrial processes can be exploited to build an approximate model that has a much lower state dimension. For linear models, decrease in model order directly also results in a reduction of the CPU time required per model evaluation. For nonlinear models however, it can be shown that simulations using the lower order approximation of the large scale model still requires the same amount of computing time [92]. So for nonlinear models reducing the model order will also reduce some of the computational complexities (e.g. number of model evaluations in the EKF or UKF) to compute a state estimate, but each model evaluation still requires approximately the same computation time as the original model; computing time which is approximately equal to the sampling interval. This prevents the use of traditional nonlinear state filters such as the EKF, UKF and MHE, even using reduced order nonlinear models. An overview of model reduction methods and their use in the field of process control is given in [63][92].

Apart from the computational infeasibilities preventing the design of an efficient state filter, first principles models often do not provide all the necessary knowledge to design an optimal state filter. While first principles models often describe the deterministic behavior of a plant in great detail, they often provide little to no information about the distribution of process disturbances and measurement noise. Unfortunately these distributions are required to design an efficient state filter. The lack of a model for process disturbances and measurement noise thus forms another obstacle for the

implementation of state estimation techniques.

Another problem that is likely to cause problems when implementing any of the state estimators described above, is that the process behavior tends to change over time, causing the behavior predicted by the model to become biased. The change in process behavior can for instance be caused by wear of parts. If the process models become biased, state estimators using these models will generally return biased estimates.

Since advanced state estimation techniques such as the EKF, UKF and MHE cannot be used for large scale first principle models, sub-optimal techniques are often used. In these techniques the state estimation problem is often solved by replacing the complex first principles model with a (reduced order) linearized model for the purpose of designing a state filter. The state estimator is often a simple fixed gain observer (see [59]), which is often tuned manually to improve performance [61][64].

Summarizing we saw that existing techniques for state estimation, such as EKF, UKF and MHE cannot be implemented directly for large first principles process models mainly due to computational issues caused by the large state dimension and the relatively long computational time required for a model evaluation. Of these problems, only the problem of a high state dimension can be solved using empirical model reduction techniques. The CPU time per model evaluation for a reduced order models is not significantly affected by the model reduction. Besides the computational issues, another issue preventing the using of first principles model for state estimation, is the lack of a model for the expected process disturbances. Finally, commonly used state estimation methods do not have methods to perform online calibration of the process model.

1.2.4 State estimation techniques in other fields

Environmental modelling

The need for good state estimation algorithms is not unique for the process industry. In other fields, different aspects of the state estimation problem for complex first principles models are also encountered. For instance, state estimation using large scale first principles models is a crucial technology in the field of environmental modelling.

First principles models of the environment are for instance used to model the weather. Other environmental models are used to estimate the distribution of smog particles [35] or to estimate currents in the oceans [38]. First principles environmental models have a lot in common with first principles process models. Both types of models are generally nonlinear, and the state dimension is very high. As such, state estimation techniques developed in this field of research should also be considered for use in combination with first principles process models. An important difference between environmental models and process models is the sample interval. For environmental models the sampling interval is usually in the order of hours. This is much longer than the sampling interval encountered in process applications.

A currently very popular state estimator for environmental models is the EnKF (Ensemble Kalman filter) [28]. The EnKF is a modified version of the EKF, such

that in order to compute both the time update and the measurement update it is no longer required to linearize the available model. Instead the EnKF computes the required predictions and associated error covariance matrices using Monte Carlo simulations. Good results usually require that approximately 50 to 100 simulations are computed per sampling interval. This is significantly less than the (at least) thousands of simulations required to numerically compute the linearization of a first principles environmental model.

Another state estimation technique that has been proposed for state estimation for atmospherical models is the RRSQRTKF (Reduced Rank Square Root Kalman Filter) [97]. Like the EnKF, the RRSQRTKF is a modified version of the original EKF. The RRSQRTKF solves the problem associated with the computation of the large error covariance matrices by using lower rank approximations of the required error covariance matrices. The rank reduction is performed by using a singular value decomposition. Because the RRSQRTKF uses a reduced rank approximation, it can be shown that it no longer required to compute a full linearization of the model. This limits the number of model simulations that are required per sampling interval. Experience with the RRSQRTKF shows that in practice good results require at least a rank 30-50 approximation of the error covariance matrices. Using finite difference to compute the required linearization of the first principles model thus requires at least 30-50 model simulations per sampling interval. This is often impossible in process applications.

The final state estimation technique commonly encountered in environmental modelling applications is referred to as the 4DVar method [19][53]. This method is very similar to the MHE technique described earlier. In both methods, the state estimate is obtained by solving a least squares optimization problem. Because of the larger sampling interval commonly encountered in environmental applications, the optimization problem can be solved on time using adjoint techniques.

While in all of these methods a dramatic decrease in required computational complexity is observed, the number of required simulations per sampling interval is still much too large for application in the process industry.

Particle filters

Besides the different filters already discussed in this section, particle filters (also referred to as sequential Monte Carlo estimators), are often used to estimate the state of the system [24]. Particle filters are for instance used in the field of automated navigation. In this field, an autonomous vehicle uses state estimation to track its own position [34]. Like the state estimation problem for industrial processes, the models used for state estimation are generally nonlinear. Instead of approximating the best linear estimator for a nonlinear problem like all Kalman based filters, a particle filter attempts to reconstruct the complete probability density function of the state given all available measurements, by conducting Monte Carlo simulations with the process model. The estimated probability density function converges to the true probability density function as the number of Monte Carlo simulations tends to infinity. The reconstructed probability density function can be used to compute the conditional expectation of the system state. Using the estimated probability density function of the state of the

system, it is possible to compute a state estimate. Application of particle filters in the literature show that required number of Monte Carlo runs to obtain good results is often quite large ($>> 50$). Once more, this prevents the use of particle filters for estimating the state of first principles process models.

1.3 Problem formulation

The overview given in the previous sections indicates that although a lot of state estimation techniques can already be found in the literature, no technique is currently able to produce reliable online state estimates using detailed first principles process models. The lack of such a state estimator limits the possibilities of using detailed first principles process models for monitoring and all further applications that require the presence of accurate state knowledge.

These observations have inspired the following central problem formulation for this thesis:

> *Develop a computationally feasible method for the efficient use of large scale physical models in model based monitoring, fault detection and control of industrial processes.*

The to be developed method is required to be computationally feasible. By this we mean that the method can be implemented online using general purpose computers.

The sought methodology is also required to be efficient. This means that the methodology should estimate the state with an accuracy approaching the theoretical optimal accuracy in the mean least squares sense.

Since the first principles models used in the process industry are very diverse, our methodology should not focus on the use of selected models. Instead, the methodology should be broadly applicable.

1.4 Solution strategy

From the discussing in section 1.2 we can identify four main obstacles for developing the methodology as described in section 1.3:

- The state dimension is very large.
- The simulation time is of the same order as the simulated interval.
- The models lack a description of disturbances and measurement noise.
- The models may need to be recalibrated during operation.

In order to develop the methodology as described in section 1.3, we will address each of these obstacles separately.

The first problem concerns the very large state dimension. This problem can be solved by applying available model reduction techniques that are already present in literature. Using these techniques an approximate model is constructed that reduces the state dimension to reasonable magnitudes, while retaining the physical interpretation of the state.

The second obstacle concerns the simulation time of large scale first principles model. While computational issues related to the state dimension of a model can be solved by existing model reduction techniques, no generally applicable techniques exist that can significantly reduce the computational burden to perform simulations with the first principles model, while retaining the physical interpretation of the model states and parameters. If such a technique would be available, it would become possible to use existing techniques to estimate the state of the system. Apart from enabling state estimation techniques, faster models could also be used in fault detection and control tasks.

Once the computational issues preventing the use of state estimation techniques have been solved, there is still the problem that first principles models tend to lack an accurate description of the noises and disturbances. Since it is apparently difficult to model disturbances and noises a priori from first principles, noise models should thus be constructed from actual measurement data, using identification techniques.

Finally, since process behavior changes over time, it is also necessary to update the model online. The process model can be calibrated, by estimating certain process parameters using available data. Unfortunately continuous estimation of process parameters will lead to increased variance of state estimates. An algorithm is thus required that only calibrates the process model when available data suggest that recalibration is necessary.

1.5 Overview of thesis contents

Chapter 2 contains a short summary of elements from state estimation and model reduction theory. The first part of the chapter starts by presenting the state estimation problem and its formal solution using Bayes conditional probability theory. Apart from the theoretical solution, several practical algorithms are presented for both linear and nonlinear models. The second part of chapter 2 contains elements from model reduction theory. In particular, this part describes projection based techniques such as Proper Orthogonal Decomposition and Balancing.

In chapter 3 addresses the problem of the simulation time of large scale first principles models. The main contribution of this chapter is that two methodologies are presented to construct a model that approximates the original first principles process model, but has a significantly reduced simulation time per model evaluation. The approximate models use the same state as the original first principles models, thus retaining the physical interpretation of the state variables.

Chapter 4 considers the problem of how a near optimal state estimator can be constructed when no information is available on the distribution of the process dis-

turbances and measurement errors. Specifically we consider methods to identify an optimal filter using measured input and output data from the process. Especially for linear models several methods to identify a near optimal filter have been published in the literature. This chapter considers the covariance method introduced by Mehra [66]. The contributions of this chapter are twofold. First, in an analysis of Mehra's method, it will be shown that although the method has several favorable properties, it can easily produce poor state estimates for the class of poorly observable systems. Secondly, a improved version of Mehra's covariance method is presented that is more robust, especially if the process is poorly observable.

Chapter 5 addresses the problem of how the process model should be adjusted on-line. It is shown that the problem of determining when a model has to be recalibrated can be written as a model selection problem. Main contribution of this chapter is that the model selection problem is solved by adapting techniques from model selection for system identification to model selection for filtering.

In chapter 6 the theory of the preceding chapters is applied in a simulation study. In this simulation study the goal is to estimate the state of the dryer section of a paper production plant. In this case study a detailed first principles model was provided by TNO Science and Industry.

Chapter 7 contains conclusions and recommendations for further research.

Most results in this thesis have already been published in the form of conference papers. Results of chapter 3 are also contained in [13] and [11]. The main results of chapter 4 have been presented in [12]. Finally, the results of chapter 5 have been published in [14].

Chapter 2

State estimation and model reduction

2.1 Introduction

This chapter introduces basic concepts from the literature that will be used in the following chapters of this thesis. This chapter can be divided into three main sections. The first section discusses the assumptions on the available first principles process models that are available for monitoring. In the second section important elements from state estimation theory are discussed. The final section of this chapter provides a brief overview of projection based model reduction techniques relevant for this research.

2.2 Process model

The process models that will be considered throughout this thesis are so-called first principles models. Such models contain all available chemical and physical insights into the considered process.

Since these models are to be used for monitoring and control purposes, it is generally convenient to reformulate the model such that it has the discrete time state-space form:

$$x(k+1) = f(x(k),u(k),w(k)) \tag{2.1}$$
$$y(k) = h(x(k),u(k),v(k)), \tag{2.2}$$

in which $x(k) \in \mathbb{R}^{n_x \times 1}$ is the state vector of the system at time index $k \in \mathbb{Z}$, $u(k) \in \mathbb{R}^{n_u \times 1}$ is a vector with known inputs, $w(k) \in \mathbb{R}^{n_w \times 1}$ is a vector containing unmeasurable process disturbances, $y(k) \in \mathbb{R}^{n_y \times 1}$ is a vector containing the measurements at time k and finally $v(k) \in \mathbb{R}^{n_v \times 1}$ is a vector containing the measurement errors at time index k. Finally the functions $f(\cdot)$ and $h(\cdot)$ relate the current state and inputs to a subsequent state vector and output vector, respectively.

Unfortunately, most first principles models are not directly available in the discrete time state-space form. Instead these models are generally available only as a set of partial differential equations. As an example of such a PDE consider the following generic PDE describing the flow of some scalar physical quantity $x(r,t)$ as a function of location r (one dimension) and time t:

$$\mathcal{L}\left(x(r,t), \frac{\partial x(r,t)}{\partial r}, \frac{\partial x(r,t)}{\partial t}\right) = 0, \tag{2.3}$$

in which $\mathcal{L}(\cdot)$ is an arbitrary PDE. Note that in contrast to the discrete time state-space model both the spatial coordinate r and the time index t are continuous variables. A PDE such as (2.3) often cannot be solved analytically. Instead the equations are thus solved using numerical methods such as finite differences or finite elements [26][68]. Both methods first impose a fine spatial grid over the physical quantity $x(r,t)$. In each cell of the imposed spatial grid, the physical quantity $x(r,t)$ is assumed uniformly distributed. As a result we can rewrite the scalar spatial distribution of $x(r,t)$ as a vector:

$$x(r,t) \rightarrow x(t) = \begin{bmatrix} x(r_1,t) \\ x(r_2,t) \\ \vdots \\ x(r_N,t) \end{bmatrix} \tag{2.4}$$

in which $r_1, r_2, \ldots, r_N$ are the locations of all the grid cells. The derivatives with respect to the spatial coordinate r can now be removed from the PDE (2.3) using a finite difference approximation. The finite difference approximation assumes that :

$$\left.\frac{\partial x(r,t)}{\partial r}\right|_{r=r_k} \approx \frac{x(r_{k+1},t) - x(r_k,t)}{r_{k+1} - r_k}. \tag{2.5}$$

Substituting this approximation into (2.3) results in a continuous time ODE. This ODE can in turned be solved using various ODE solvers to result in a discrete time model of the form (2.1)-(2.2). In the resulting model the vector $x(t)$ is in fact the state vector.

Note that the dimension of the state vector is proportional to the number of grid elements to approximate the PDE with an ODE. For reasons of accuracy, the number of grid cells is chosen high, which in turn causes the high state dimension of many first principles process models.

Apart from the model equations specified by $f(\cdot)$ and $h(\cdot)$ a description of the expected type of disturbances $w(k)$ and measurement errors $v(k)$ are also required for monitoring. In this thesis it will be assumed that both $w(k)$ and $v(k)$ can be modelled as realizations of stochastic processes with known distributions. Mostly we will assume that both $w(k)$ and $v(k)$ are Gaussian white noise processes with known covariances:

$$\mathbb{E}\left\{\begin{bmatrix} w(k) \\ v(k) \end{bmatrix}\right\} = 0 \tag{2.6}$$

$$\mathbb{E}\left\{\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \begin{bmatrix} w(l)^T & v(l)^T \end{bmatrix}\right\} = \begin{bmatrix} Q(k) & S(k) \\ S(k)^T & R(k) \end{bmatrix} \delta(k-l), \tag{2.7}$$

in which $\mathbb{E}\{\cdot\}$ is the expectation operator and $\delta(k-l)$ is defined as:

$$\delta(k-l) = \begin{cases} 0 & \text{if } k \neq l \\ 1 & \text{if } k = l. \end{cases} \tag{2.8}$$

Note that for the remainder of this thesis it is not required that an explicit model of the form (2.1)-(2.2) is available. It is sufficient that model evaluations of (2.1)-(2.2) can be computed.

2.3 State estimation

2.3.1 State estimation problem

State estimation problem

In order to obtain accurate knowledge of the state vector and its evolution, one option would be to perform online measurements of the state of the system. Unfortunately, the state vector can only rarely be measured directly. As a result, the only method to (approximately) obtain the current state of the system is to estimate the state of the system using all known inputs $u(k)$ and available measurements $y(k)$. Indeed, the state $x(k)$ is related to both inputs $u(k)$ via the state equation (2.1) and outputs $y(k)$ via the measurement relation $h(\cdot)$ (see (2.2)). In this section the current state of the art in state estimation techniques will be discussed. Before discussing the various state estimation techniques we will first provide a more mathematical formulation of the state estimation problem. Denote all known input and output data available at time k as Z^k:

$$Z^k = [u(1),\, y(1),\, \ldots,\, u(k),\, y(k)]. \tag{2.9}$$

Using all available data and the available process model, the state estimation problem consists of finding an estimator $\hat{x}(k,Z^k)$ of $x(k)$ that minimizes a chosen criterion function $\mathcal{C}(x(k)-\hat{x}(k,Z^k))$. By far the most common criterion used in the literature is the mean square error (MSE):

$$\mathcal{C}(x(k)-\hat{x}(k,Z^k)) = \mathbb{E}\{\sum_k \|x(k)-\hat{x}(k,Z^k)\|^2\} \tag{2.10}$$

in which the norm $\|\cdot\|_.$ is defined as:

$$\|x\|_P^2 = x^T P x. \tag{2.11}$$

General solution

For the MSE criterion (2.10) it can be shown that the optimal estimator $\hat{x}(k,Z^k)$ is [3]:

$$\hat{x}(k,Z^k) = \mathbb{E}\{x(k)|Z^k\}. \tag{2.12}$$

The best estimator for $x(k)$ is thus the conditional expectation of $x(k)$ given all available data Z^k.

In order to exactly compute the conditional expectation of $x(k)$ given all data Z^k, we need to know the conditional probability density function of $x(k)$ given Z^k. Denoting this conditional probability density function as $p(x(k)|Z^k)$, the conditional expectation (2.12) can be computed as:

$$\mathbb{E}\{x(k)|Z^k\} = \int x(k)p(x(k)|Z^k)d\{x(k)\}. \tag{2.13}$$

where the integral over $x(k)$ is performed over the entire space $\mathbb{R}^{n_x}$. Suppose $p(x(k)|Z^k)$ is available at some initial time index k_0, then the conditional probability density function at all sampling instants $k > k_0$ can then in principle also be computed exactly, via a two stage recursive algorithm.

The first stage of the algorithm is called the prediction stage. In this step the system model equations (2.1), (2.7) are used to compute the a priori probability density function $p(x(k+1)|Z^k)$ of the future state $x(k+1)$. The a priori probability density function is given by:

$$p(x(k+1)|Z^k) = \int p(x(k+1)|x(k))p(x(k)|Z^k)d\{x(k)\} \tag{2.14}$$

in which $p(x(k+1)|x(k))$ is the probability density function that describes the probability of a state transition from a given state $x(k)$ to a possible state $x(k+1)$. This probability density function can be computed using (2.1) and (2.7) via:

$$p(x(k+1)|x(k)) \quad = \quad \int_{w(k)\in\mathcal{D}_w(x(k+1)|x(k))} p(w(k))d\{w(k)\} \tag{2.15}$$

in which $p(w(k))$ is the probability density function of $w(k)$ and $\mathcal{D}_w(x(k+1)|x(k))$ is a set of realizations $w(k)$ defined as:

$$\mathcal{D}_w(x(k+1)|x(k)) \quad \triangleq \quad \{w(k):\ x(k+1) - f(x(k),u(k),w(k)) = 0\}. \tag{2.16}$$

If the probability density function $p(w(k))$ is known the integral (2.15) can be solved.

The second step of the algorithm is the correction step. The correction step computes the conditional probability $p(x(k+1)|Z^{k+1})$ once a new measurement $y(k+1)$ becomes available. The new conditional probability density function is computed using the well-known Bayes rule [69]:

$$p(a|b,c) = \frac{p(b|c)p(b|a,c)}{p(a|c)}. \tag{2.17}$$

Substituting $x(k+1)$ for a, $y(k+1)$ for b and Z^k for c we thus have:

$$p(x(k+1)|Z^{k+1}) \quad = \quad p(x(k+1)|Z^k, y(k+1)) \tag{2.18}$$

$$\quad = \quad \frac{p(y(k+1)|x(k+1))p(x(k+1)|Z^k)}{p(y(k+1)|Z^k)}. \tag{2.19}$$

The probability density function $p(y(k+1)|x(k+1))$ can be computed using the model equations (2.2) and (2.7):

$$p(y(k+1)|x(k+1)) = \int_{\mathcal{D}_v(y(k+1)|x(k+1))} p(v(k+1))d\{v(k+1)\}, \tag{2.20}$$

in which $\mathcal{D}_v(y(k+1)|x(k+1))$ is the set of realizations of $v(k+1)$ defined as:

$$\mathcal{D}_v(y(k+1)|x(k+1)) \triangleq \{v(k) : \ y(k+1) - h(x(k+1), u(k+1), v(k+1)) = 0\} \tag{2.21}$$

The integral (2.20) thus summarizes the probability mass of all realizations of $v(k+1)$ such that $y(k+1) = h(x(k+1), u(k+1), v(k+1))$.

The final term $p(y(k+1)|Z^{k+1})$ in (2.19) can be computed via:

$$p(y(k+1)|Z^k) = \int p(y(k+1)|x(k+1))p(x(k+1)|Z^k)dx(k+1). \tag{2.22}$$

Of course, analytically computing these equations is often infeasible, hence in practice methods are often used that can approximately compute these equations.

Only in the special case that the functions $f(\cdot)$ and $h(\cdot)$ in the available model are linear functions in $x(k)$, $u(k)$, $w(k)$ and $v(k)$, and $v(k)$ and $w(k)$ Gaussian white noise processes, the conditional expectation (2.12) can be efficiently computed. This can be accomplished using the well known Kalman filter.

2.3.2 Kalman filter

Introduction

As described above, for linear systems with Gaussian distributed process disturbances and measurements errors, the conditional expectation (2.12) that corresponds to the optimal estimator $\hat{x}(k, Z^k)$ can be computed exactly and efficiently using Kalman filter theory [47]. A good summary of the historical developments that have led to the development of the Kalman filter equations is presented in [86]. In the following, only the main results are presented.

Kalman filter recursions

In order to apply Kalman filter theory to produce optimal estimates of the state $x(k)$, it is required that the true process under study can be modelled using a linear state-space model:

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) + w(k) & (2.23)\\ y(k) &= C(k)x(k) + v(k) & (2.24) \end{aligned}$$

with $A(k), B(k), C(k)$ known matrices of appropriate dimension. It is also required that the process noise $w(k)$ and the measurement error $v(k)$ are realizations of stochastic Gaussian white noise processes as described by (2.7). Finally, in order to initialize

the Kalman filter an initial estimate of $x(k_0)$ is required. The error between the initial estimate and the actual value of $x(k_0)$ should be Gaussian distributed with a known error covariance.

Using these assumptions, the Kalman filter can be used to compute state estimates that have an optimal MSE as defined (see (2.10)). The Kalman filter is a two stage recursive algorithm. It consists of a prediction step (also called the time update) and a correction step (also called the measurement update).

Before presenting the actual Kalman filter equations, the following notation is introduced for the conditional expectation of $x(k)$ given data Z^l:

$$\hat{x}(k|l) \triangleq \mathbb{E}\{x(k)|Z^l\}. \tag{2.25}$$

For $k = l$ the conditional expectation is called the *filter estimate*, for $k > l$ it is referred to as the optimal *prediction* and for $k < l$ it is called the *smoothed estimate*. The error covariance matrix of the conditional expectation is denoted as:

$$P_{\hat{x}(k|l)} \triangleq \mathbb{E}\{(x(k) - \hat{x}(k|l))(x(k) - \hat{x}(k|l))^T\} \tag{2.26}$$

This notation is used in the original paper of Kalman [47], and has since been adopted in many texts about Kalman filters.

The first step of the Kalman filter procedure (the prediction step), uses the current filter estimate $\hat{x}(k|k)$ and its associated error covariance matrix $P_{\hat{x}(k|k)}$ to compute the optimal prediction $\hat{x}(k+1|k)$ of the future state $x(k+1)$:

$$\begin{aligned} \hat{x}(k+1|k) &= \mathbb{E}\{x(k+1)|Z^k\} && (2.27)\\ &= A(k)\hat{x}(k|k) + B(k)u(k). && (2.28) \end{aligned}$$

The error covariance of the state prediction can be computed via:

$$\begin{aligned} P_{\hat{x}(k+1|k)} &= \mathbb{E}\{[x(k+1) - \hat{x}(k+1|k)][x(k+1) - \hat{x}(k+1|k)]^T\} && (2.29)\\ &= A(k)P_{\hat{x}(k|k)}A(k)^T + Q(k). && (2.30) \end{aligned}$$

Using the optimal prediction for the state, $\hat{x}(k+1|k)$, the optimal prediction for the next output $y(k+1)$ can be easily computed via:

$$\begin{aligned} \hat{y}(k+1|k) &\triangleq \mathbb{E}\{y(k+1)|Z^k\} && (2.31)\\ &= C\hat{x}(k+1|k). && (2.32) \end{aligned}$$

Once the measurement $y(k+1)$ becomes available, the prediction $\hat{x}(k+1|k)$ can be used to compute the filter estimate $\hat{x}(k+1|k+1)$ and its covariance $P_{\hat{x}(k+1|k+1)}$. This step is often called the measurement update, but is also referred to as the correction step. It can be shown that the measurement update can be computed using (see for instance [3][46]):

$$\begin{aligned} \hat{x}(k+1|k+1) &= \mathbb{E}\{x(k+1)|Z^{k+1}\} && (2.33)\\ &= \hat{x}(k+1|k) + K(k+1)(y(k+1) - \hat{y}(k+1|k)) && (2.34) \end{aligned}$$

in which $K(k+1)$ is the Kalman gain matrix:

$$K(k) = P_{\hat{x}(k|k-1)}C(k)^T(C(k)P_{\hat{x}(k|k-1)}C(k)^T + R(k))^{-1}. \tag{2.35}$$

The error covariance of this filter estimate $\hat{x}(k+1|k+1)$ can be computed using the following equation:

$$\begin{aligned} P_{\hat{x}(k+1|k+1)} &= \mathbb{E}\{[x(k+1) - \hat{x}(k+1|k+1)][x(k+1) - \hat{x}(k+1|k+1)]^T\} \\ &= P_{\hat{x}(k+1|k)} - K(k)C(k)P_{\hat{x}(k+1|k)}. \end{aligned} \tag{2.36}$$

To sum up: we have seen that based on some initial estimate $\hat{x}(k_0|k_0)$ and its error covariance, the Kalman filter allows one to recursively compute optimal state estimates $\hat{x}(k|k)$ for all $k > k_0$ by successively using the presented prediction and correction steps. Compared to the operations required to solve the general filtering problem described earlier, the Kalman filter is computationally a much simpler method to produce state estimates, because it avoids the use of complex integrals over probability density functions.

Kalman filter related properties and definitions

In the years after the invention of Kalman filtering theory, the Kalman filter has become a valuable tool in the fields of signal processing, system identification and control. Given its important role in these fields, the properties of the Kalman filter have been studied in great detail. This section provides a summary of some properties of the Kalman filter that will be used in the remainder of this thesis. For a derivation and discussion of the presented properties the reader is referred to monographs as [3][42].

The innovation signal $e(k)$ of a Kalman filter is defined as:

$$e(k) = y(k) - \hat{y}(k|k-1). \tag{2.37}$$

It can be shown that the innovation sequence is again a zero mean Gaussian distributed white noise process [3]:

$$\mathbb{E}\{e(k)\} = 0 \tag{2.38}$$

$$P_{e(k)e(l)} \triangleq \mathbb{E}\{e(k)e(l)^T\} \tag{2.39}$$

$$= \left[C(k)P_{\hat{x}(k|k-1)}C(k)^T + R(k)\right]\delta(k-l). \tag{2.40}$$

In the Kalman filter equations presented earlier in this section, the linear system matrices were allowed to be time-varying. In most applications however, the matrices $A(k), B(k), C(k), Q(k), R(k)$ are constant. If these matrices are indeed constant, it can be shown that the Kalman gain $K(k)$ converges to a constant matrix as k tends to infinity.

The Kalman state estimate $\hat{x}(k|k)$ can be shown to be equivalent to the solution of a regularized least squares problem (see for instance [85][3]). Suppose that $\hat{\xi}(k-M+$

$1),\ldots,\hat{\xi}(k)$ are the solution to the following least squares problem:

$$\hat{\xi}(k-M+1),\ldots,\hat{\xi}(k) = \arg \min_{\xi(k-M+1),\ldots,\xi(k)} \sum_{i=k-M+1}^{k} \|\hat{v}(i)\|^2_{R(i)^{-1}} + \sum_{i=k-M}^{k-1} \|\hat{w}(i)\|^2_{Q(i)^{-1}} + \|\xi(k-M) - \hat{x}(k-M|k-M)\|^2_{P^{-1}_{\hat{x}(k-M|k-M)}} \quad (2.41)$$

subject to

$$\hat{v}(i) = y(i) - C(i)\xi(i) \quad (2.42)$$

$$\hat{w}(i) = \xi(i+1) - A(i)\xi(i) - B(i)u(i), \quad (2.43)$$

with M any positive integer. Then, it holds that

$$\hat{\xi}(k) = \hat{x}(k|k) = \mathbb{E}\{x(k|Z^k)\} \quad (2.44)$$

$$\vdots$$

$$\hat{\xi}(k-M+1) = \hat{x}(k-M+1|k) = \mathbb{E}\{x(k-M+1|Z^k)\}. \quad (2.45)$$

Several authors (see for instance [42]) have observed that minimizing the least squares criterion (2.41) is equivalent to maximizing the a posteriori conditional probability density function $p(x(k),\ldots,x(k-M+1)|Z^k)$.

Since the error in the prior $x(k-m)-\hat{x}(k-M)$, the disturbances $w(k-1),\ldots,w(k-M)$ and the measurement errors $v(k),\ldots,v(k-M+1)$ are all Gaussian, the maximum of the posteriori conditional probability density function $p(x(k),\ldots,x(k-M+1)|Z^k)$ corresponds to the conditional expectations in (2.44).

Using the least squares problem (2.41) to estimate state is computationally more involved than using the Kalman recursions (2.28)-(2.34). As a result the least squares form is virtually never used to estimate the state of a linear system. However, as will be discussed in the next section, the least squares form (2.41) is easier to generalize to nonlinear systems.

Finally, in the derivations of the Kalman filter and all its properties we have assumed that all disturbances and measurement errors have a Gaussian distribution. Only under this assumption do the Kalman estimates $\hat{x}(k|k)$ correspond to the best possible estimates in the MSE sense. In the more general case in which disturbances and measurement errors are not Gaussian distributed, it still holds that the Kalman filter is the best possible *linear* unbiased estimator of $x(k)$ in the MSE sense.

2.3.3 Nonlinear state estimation

Introduction

For linear systems the Kalman filter recursions can be used to efficiently compute the optimal state estimate (2.12). For the cases in which $f(\cdot)$ or $h(\cdot)$ are nonlinear, the optimal state estimate can only be computed using the complex expressions (2.14)-(2.22). Since it is often infeasible to compute the optimal state estimate using these

relations, approximate filter relations are generally used. The resulting approximate filters only approximate the optimal state estimate $\mathbb{E}\{x(k)|Z^k\}$. The approximate filters can be divided into four main groups: Extended Kalman filters, approximately best linear unbiased filters, least squares based filters and Monte Carlo or particle filters.

Extended Kalman filters

If the functions $f(\cdot)$ and $h(\cdot)$ can be accurately approximated by a linear system of the form (2.23)-(2.24) for all $x(k)$ within the confidence intervals of $\hat{x}(k|k-1)$ and $\hat{x}(k|k)$, then it is reasonable to assume that a linear Kalman estimator based on linearization of the nonlinear model (2.1)-(2.2) will still produce good results.

By far the most popular filter that is based upon this reasoning, is the Extended Kalman filter (EKF). The EKF recursions are given by:

Prediction step:

$$\begin{aligned} \hat{x}(k+1|k) &= f(\hat{x}(k|k),u(k),0) && (2.46)\\ P_{\hat{x}(k+1|k)} &= F(k)P_{\hat{x}(k|k)}F(k)^T + G(k)Q(k)G(k)^T && (2.47) \end{aligned}$$

with

$$\begin{aligned} F(k) &= \left.\frac{\partial f(x,u,w)}{\partial x}\right|_{x=\hat{x}(k|k),u=u(k),w=0} && (2.48)\\ G(k) &= \left.\frac{\partial f(x,u,w)}{\partial w}\right|_{x=\hat{x}(k|k),u=u(k),w=0}. && (2.49) \end{aligned}$$

Correction step:

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K(k)[y(k) - h(\hat{x}(k+1|k)] && (2.50)\\ P_{\hat{x}(k+1|k+1)} &= P_{\hat{x}(k+1|k)} - K(k)H(k)P_{\hat{x}(k+1|k)}, && (2.51) \end{aligned}$$

with

$$K(k) = P_{\hat{x}(k+1|k)}H(k)^T(H(k)P_{\hat{x}(k+1|k)}H(k)^T + R(k))^{-1} \quad (2.52)$$

and

$$H(k) = \left.\frac{\partial h(x,u,v)}{\partial x}\right|_{x=\hat{x}(k+1|k),u=u(k),v=0}. \quad (2.53)$$

As can be seen the EKF requires the computation of the Jacobian of both $f(\cdot)$ and $h(\cdot)$. If the nonlinear model (2.1)-(2.2) is only available in its explicit form, these Jacobians are mostly approximated numerically. The most basic method to approximately compute the Jacobians is using the method of finite differences. Using this

method the $i-th$ column of the Jacobian $F(k)$ is approximated using the following formula:

$$\frac{f(x(k)+\varepsilon e_i,u(k),0)-f(x(k),u(k),0)}{\varepsilon}, \tag{2.54}$$

with ε a small scalar constant and e_i a vector containing only zeros except for its $i-th$ element, which is one:

$$e_i^T = [0\ \cdots\ 0\ 1\ 0\cdots\ 0]. \tag{2.55}$$

Computing the Jacobians numerically in this manner thus requires at least n_x+1 function evaluations; one function evaluation of $f(x(k),u(k),0)$ and n_x further evaluations of $f(x(k)+\varepsilon e_i,u(k),0)$ for $i=1,\ldots,n_x$. Similarly, computing the Jacobian $H(k)$ using the same technique requires n_x+1 function evaluations of $h(\cdot)$.

Approximate best linear unbiased estimators

For linear models, it can be shown that the correction step of the Kalman filter (see (2.34)-(2.36)) exactly corresponds to the Best Linear Unbiased Estimate (BLUE) of $x(k)$ given $e(k)$. To make this apparent first the general form of the BLUE will be presented. Suppose that two random variables x and y have a priori means and covariances given by:

$$\mathbb{E}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} \quad \text{and} \tag{2.56}$$

$$\mathbb{E}\begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}\begin{bmatrix} x-\mu_x \\ y-\mu_y \end{bmatrix}^T = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{xy}^T & R_{yy} \end{bmatrix}. \tag{2.57}$$

Then the BLUE of x given a realization of y is:

$$\hat{x} = \mu_x + R_{xy}R_{yy}^{-1}(y-\mu_y), \tag{2.58}$$

and the error covariance of $\hat{x}$ can be computed via:

$$\mathbb{E}(x-\hat{x})(x-\hat{x})^T = R_{xx} - R_{xy}R_{yy}^{-1}R_{xy}^T. \tag{2.59}$$

From these expressions it is relatively easy to see that the correction step (2.34) in the Kalman filter indeed corresponds to the BLUE estimate of $x(k)$ given $e(k)$. After substituting

$$\hat{x} = \hat{x}(k+1|k+1) \tag{2.60}$$
$$y = e(k+1) \tag{2.61}$$
$$\mu_x = \hat{x}(k+1|k) \tag{2.62}$$
$$\mu_y = 0 \tag{2.63}$$
$$R_{xy} = P_{\hat{x}(k+1|k)}C(k+1)^T \tag{2.64}$$
$$R_{yy} = P_{e(k+1)e(k+1)} \tag{2.65}$$

in (2.58) and (2.59) these equations exactly correspond to (2.34) and (2.36). For convenience, we use the following form of the BLUE:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + P_{(x(k+1)-\hat{x}(k+1|k))e(k+1)} P^{-1}_{e(k+1)e(k+1)} e(k+1), \quad (2.66)$$

with $P_{(x(k+1)-\hat{x}(k+1|k))e(k+1)}$ the cross-covariance between the state prediction error $x(k+1)-\hat{x}(k+1|k)$ and the innovation signal $e(k+1)$.

Note that for the expression for the BLUE it does not matter if the available model is linear or nonlinear. Even though the expression for the BLUE remains unchanged for nonlinear systems, computing the BLUE is generally more difficult. In the linear case all required predictions and associated covariance matrices can be computed exactly using the Kalman time update and measurement update equations. In the nonlinear case generally no computationally easy relations exist to compute either the predictions or their covariances. Thus approximate methods have been developed to compute the required expressions in (2.58)-(2.65).

The simplest method to compute all the required predictions and covariances is by linearizing the model as was done in the Extended Kalman filter. While this approach is often good for nearly linear models, better methods exist for models with more pronounced nonlinear behavior (see [98]).

Instead of linearizing $f(\cdot)$ and $h(\cdot)$ the required predictions and covariances can also be obtained empirically using a procedure that utilizes simulations of $f(\cdot)$ and $h(\cdot)$. Basically these simulation based procedures generally follow the same procedure. As an example we will show how such a simulation based procedure is used to compute the predicted state $\hat{x}(k+1|k)$ and its error covariance $P_{\hat{x}(k+1|k)}$. The first step in the simulation based procedure is to generate a set of points x_i,w_i with $i = 1,\ldots,N$, with a predetermined distribution such that:

$$\begin{aligned}
\mathbb{E}\{x_i\} &= \hat{x}(k|k) && (2.67)\\
\mathbb{E}\{w_i\} &= 0 && (2.68)\\
\mathbb{E}\{(x_i-\hat{x}(k|k))(x_i-\hat{x}(k|k))^T\} &= P_{\hat{x}(k|k)} && (2.69)\\
\mathbb{E}\{w_i {w_i}^T\} &= Q(k). && (2.70)
\end{aligned}$$

The exact methods by which points x_i and w_i are generated differs among the BLUE based estimators. The points x_i and w_i are used to generate points x_i^* via simulations:

$$x_i^* = f(x_i, u(k), w_i). \quad (2.71)$$

The resulting set of points x_i^* is now used to determine $\hat{x}(k+1|k)$ and $P_{\hat{x}(k|k-1)}$. The predicted state is typically computed via:

$$\hat{x}(k+1|k) = \frac{1}{N}\sum_{i=1}^{N} x_i. \quad (2.72)$$

The error covariance of this predicted state is typically computed using the following formula:

$$P_{\hat{x}(k+1|k)} = \frac{1}{N}\sum_{i=1}^{N} (x_i-\hat{x}(k+1|k))^T (x_i-\hat{x}(k+1|k)). \quad (2.73)$$

Similar simulations with $h(\cdot)$ are used to determine $e(k)$ and $P_{e(k)e(k)}$.

As already mentioned, the differences between BLUE based filters are mainly the methods which are used to generate the points x_i and w_i. While some filters use a deterministic method of choosing x_i and w_i, other filters use samples drawn from a predetermined stochastic process to generate the points x_i and w_i. An example of a state estimator that deterministically chooses the simulation points is the Unscented Kalman Filter (UKF) [44]. The selection of the simulation points is called the Unscented Transform. Many authors have reported that the state estimates produced by the UKF are more accurate than the EKF, see for instance [98][44]. The UKF requires $4(n_x+n_w+n_v)$ simulations per state estimate to compute the required predictions and covariances in (2.66).

Like the UKF, the DD2 filter introduced in [73] also chooses its simulation points deterministically. Where the UKF is based on the Unscented Transform to approximate the required means and covariance matrices, the DD2 filter is based on Stirling's approximation of a nonlinear function. The resulting choice for the DD2 filter's simulation points is similar to those of the UKF filter. As a result, the accuracy of both filters is generally very similar [25].

A filter that determines its simulation points using stochastic measures is the Ensemble Kalman Filter (EnKF) [28]. The EnKF draws its set of simulation points from a normal distribution, with a predetermined mean and covariance. For linear systems the EnKF's estimate are only equivalent to the Kalman filter estimates as the number of simulation points tends to infinity.

Least squares approach

For linear systems, it was already mentioned that the optimal state estimate is equivalent to the solution of least squares problem (2.41). Furthermore, the solutions to the least squares problem are equivalent to the maximum a posteriori (MAP) estimate of the state.

It can be shown that for a special class of nonlinear models, where both process noises and measurement errors occur additive in the model equations, i.e. models of the form:

$$x(k+1) = f(x(k),u(k))+w(k) \tag{2.74}$$
$$y(k) = h(x(k),u(k))+v(k), \tag{2.75}$$

that the MAP estimate of the system states are still equivalent to the solution of a least squares problem [79]. This least squares problem is given by (2.41), where the constraints (2.42)-(2.43), are replaced by:

$$\hat{v}(i) = y(i)-h(\xi(i),u(i)) \quad \forall\, i = k-M+1,\ldots,k, \tag{2.76}$$
$$\hat{w}(i) = \xi(i+1)-f(\xi(i),u(i)) \quad \forall\, i = k-M,\ldots,k-1. \tag{2.77}$$

The resulting least squares problem is nonlinear in $\xi(k-M),\ldots,\xi(k)$ and can often only be solved numerically.

The least squares problem contains an initial estimate $\hat{x}(k-M|k-M)$ and its covariance $P_{\hat{x}(k-M|k-M)}$. These are in general computed using a different nonlinear state filter, such as the extended Kalman filter described earlier. It is intuitively clear that the effect of the importance of the initial estimate $\hat{x}(k-M|k-M)$ will in general decrease as the window length M increases.

Probably the best known least squares based filter is the Moving Horizon Estimator (MHE) [78][79]. To constrain the MHE estimates within certain bounds, additional constraints are often used while solving the least squares problem. These bounds are used to ensure that state estimates corresponding to physical quantities that physically bounded are also bounded. Bounds could be used for instance to ensure that states estimates corresponding to chemical concentrations are always greater than or equal to zero. Estimates using an EKF or BLUE based filter are not guaranteed to respect these limits, because both estimators are based on the classic Kalman filter, which assumes all states have a Gaussian distribution.

Approximate Bayesian state estimators

All previously described nonlinear state estimators are in some manner based on the linear Kalman filter. As a result, the state estimators are not guaranteed to converge to the optimal estimate based on the conditional expectation of $x(k)$ given all available data (see (2.12)). In contrast, the approximate Bayesian based estimators that will be described in this section attempt to construct the optimal state estimate (2.12) by using Monte Carlo simulations. The main advantage of the methods discussed in this section is that no assumptions on $f(\cdot)$ or $h(\cdot)$ are used and $w(k)$ and $v(k)$ are allowed to have any distribution.

Approximate Bayesian filters approximate the exact optimal filtering equations (2.14)-(2.22) described in section 2.3.1 using sequential Monte Carlo simulations. In the Monte Carlo simulations the goal is to approximate the various probability density functions in (2.14)-(2.22) using large numbers of samples. Indeed, as the number of samples tends to infinity, any probability density function can be accurately approximated in this manner.

Since the exact solution consists of two stages (prediction stage and a correction stage), Bayesian filters also consist of two stages. In the first stage of the exact equations, the conditional probability density function $p(x(k+1)|Z^k)$ is computed using (2.14)-(2.15). Approximating $p(x(k+1)|Z^k)$ using Monte Carlo simulations is relatively simple. For the Monte Carlo approximation we first draw N_p samples $x_i(k)$, for $i = 1,\ldots,N_p$ are drawn from the distribution $p(x(k)|Z^k)$. For each of these samples $x_i(k)$ new samples $x_i^*(k+1)$ are computed via:

$$x_i^*(k+1) = f(x_i(k), u(k), w_i(k)), \tag{2.78}$$

with $w_i(k)$ a random sample drawn from $p(w(k))$. It is intuitively clear that the samples x_i^* are distributed according to $p(x(k+1)|Z^k)$.

In the correction step of the exact filter, a measurement $y(k+1)$ is used to compute the probability density function $p(x(k+1)|Z^{k+1})$ using equations (2.19)-(2.22).

Approximating this probability density function using Monte Carlo simulations can be done in different manners. Here we shall present a relatively simple method as introduced in [32].

Once a new measurement $y(k+1)$ becomes available, each of the previously computed samples x_i^* is given a normalized weight q_i. This normalized weight is computed using the following equation:

$$q_i(k+1) = \frac{p(y(k+1)|x_i^*(k+1))}{\sum_{i=1}^{N_p} p(y(k+1)|x_i^*(k+1))}. \tag{2.79}$$

To construct samples that are approximately distributed as $p(x(k+1)|Z^{k+1})$ we need to define a discrete probability density function $p(x^*|y(k))$ such that $Pr(x^* = x_i^*(k+1)) = q_i(k+1)$ and again draw N_p samples $x_i(k+1)$ from this distribution such that $Pr(x_i(k+1) = x_i^*(k+1)) = q_i$. It can then be shown that the samples $x_i(k+1)$ will have approximately the same distribution as $p(x(k+1)|Z^{k+1})$ [32]. Using this approximate distribution the state estimate $\hat{x}(k+1|k+1)$ is given by:

$$\hat{x}(k+1|k+1) = \sum_{i=1}^{N_p} x_i(k+1). \tag{2.80}$$

Obviously, the accuracy of an approximate Bayesian filter will increase as the number of samples N_p increases. Convergence results are only available for $N_p \to \infty$. There are no simple results that can be used to chose finite number of samples N_p such that procedures as the one described above will results in an estimation accuracy that is satisfactory. Thus N_p is often determined using trial and error. In general the required number of samples depends on the state dimension n_x and size of the areas for which both the prior $p(x(k)|Z^{k-1})$ and the likelihood $p(y(k)|x(k))$ have a significant value.

An overview of current Monte Carlo based approximate Bayesian filters can be found in [24].

2.3.4 Filtering in the presence of modelling errors

The filtering theory discussed so far assumes that the available model (2.1) - (2.7) is exactly known. If this is indeed the case, the (nonlinear) filtering algorithms presented in the previous sections can be used to generate accurate state estimates.

The assumption that the available model is perfect is often not very realistic. There are many reasons which could cause that the available model to contain modelling errors. For example, during the modelling certain (minor) effects may have been neglected.

For simplicity we rewrite the model equations such that the modelling errors in (2.1)-(2.2) are collected in extra error terms $w_{err}(k)$ and $v_{err}(k)$:

$$\begin{aligned} x(k+1) &= f(x(k),u(k),w(k)) + w_{err}(k) && (2.81)\\ y(k) &= h(x(k),u(k),v(k)) + v_{err}(k). && (2.82) \end{aligned}$$

The problem of how we can generate state estimates with realistic error covariance matrices using an imperfect model has led to the field of robust filtering.

The most basic technique to make state filters more robust is by using a technique similar to stochastic embedding in the fields of parameter estimation and system identification [72]. Even though the error sequences $w_{err}(k)$ and $v_{err}(k)$ are deterministic, the stochastic embedding approach models the error sequences as stochastic processes with a known mean and variance. Using this assumption, the easiest method to obtain a state estimate is to use one of the normal state estimation techniques presented in the previous sections with extra process disturbances and measurement errors corresponding to the assumed behavior of $w_{err}(k)$ and $v_{err}(k)$. This stochastic approach to generate a more "robust" filter is often used when a linear Kalman filter is used to produce state estimates for nonlinear systems [85][42]. In these applications a linearized model is used to estimate the state of the system instead of the original nonlinear system model. It has been shown that under certain circumstances the resulting linear filters become unstable when the modelling error resulting from the linearization is disregarded. It has been shown that including an extra stochastic error term can prevent the divergence of the linear Kalman filter [84].

A second class of robust filters assumes that the modelling errors $w_{err}(k)$ and $v_{err}(k)$ are norm bounded. This assumption has led to the development of various robust filters, see for instance [100], [75] and [90]. The advantage of this category of filters is that these types of robust filters can provide guaranteed uncertainty areas for the estimated state variables. Drawbacks of these robust filters are that the uncertainty region of the state estimates is often very conservative. Also the algorithms that are used to compute these estimates often require that nominal models are linear. Even using linear models the various algorithms are generally far more computationally demanding than the standard Kalman filter. As a result this class of filters is hard to apply for large-scale nonlinear filtering problems.

2.4 Projection based model reduction

2.4.1 Introduction

If the state dimension of the models used for state estimation is very large ($n_x >> 10^3$) then applying the nonlinear state estimation techniques discussed above can become computationally infeasible. This infeasibility has two main causes: the dimension of the error covariance matrix of the state may become too large to store in computer memory and secondly the number of model evaluations required in most of the discussed state estimation algorithms increases at least linearly with n_x; therefore dramatically increasing the time required to compute filter estimates.

A partial solution to the problem consists of reducing the model by defining a new state variable $x_{red}(k)$ that has a dimension $n_{red} << n_x$. To compute these reduced order models several techniques can be found in the literature. An overview of model reduction techniques used for first principles models in the process industry is provided in [63][92]. Model reduction techniques can approximately be divided into two

classes. The first class of methods uses physical insight to derive a lower order model of the process. While good results can be obtained in this manner, the techniques used are generally very problem specific. The second class of methods is the class of projection methods. For many of the model reduction methods in this class no specific physical insight is required. This makes the second class of techniques easier to use in practice. In this section we will limit ourselves to only describing methods from this second class.

In any controllable system with state dimension n_x, the actual state of the system theoretically can vary in a n_x dimensional space. In practice, however, the state of models representing physical systems often largely remains in a lower n_{red} dimensional subspace. This implies that it is possible to construct an accurate lower order approximative model. This model will be of the form:

$$x_{red}(k+1) = f_{red}(x_{red}(k), u(k), w(k)) \tag{2.83}$$
$$y(k) = h_{red}(x_{red}(k), u(k), v(k)). \tag{2.84}$$

As already mentioned above, in this section we will only discuss projection based methods. Projection based model reduction techniques search for a full column rank matrix $T \in \mathbb{R}^{n_x \times n_{red}}$ with $n_{red} << n_x$ that induces a reduced order state vector by:

$$x_{red}(k) = T^{\dagger} x(k), \tag{2.85}$$

in which $T^{\dagger}$ is the pseudo-inverse of T:

$$T^{\dagger} = (T^T T)^{-1} T^T. \tag{2.86}$$

Once such a matrix T has been constructed, the reduced order model is generated using Galerkin projection:

$$f_{red}(x_{red}(k), u(k), w(k)) = T^{\dagger} f(T x_{red}(k), u(k), w(k)) \tag{2.87}$$
$$h_{red}(x_{red}(k), u(k), v(k)) = h(T x_{red}(k), u(k), v(k)). \tag{2.88}$$

Once a suitable projection matrix T has been chosen, state estimation can be done using the reduced order model (2.83)-(2.84). Using the reduced order model for state estimation will result in estimates for the reduced order state $\hat{x}_{red}(k|k)$. From these reduced order state estimates, estimates for the original full-order states can be reconstructed using:

$$\hat{x}_{rec}(k|k) = T \hat{x}_{red}(k|k). \tag{2.89}$$

The problem of reducing the model (2.1)-(2.2) for state estimation problems is thus to find a matrix T of rank n_{red} such that the reconstructed state estimate $\hat{x}_{rec}(k|k)$ is as accurate as possible. Unfortunately no technique exists in the literature to find the desired matrix T for the objective of state estimation.

Since no specialized technique for projection based model reduction for state estimation exists, general purpose projection based model reduction tools are commonly used. In the next sections, the two most important reduction techniques will be discussed.

2.4.2 Proper Orthogonal Decomposition

Reduction criterion and general solution

The Proper Orthogonal Decomposition (POD)[1] model reduction technique is a well known method to derive a reduced order basis. The POD technique has been successfully applied in various fields, such as fluid dynamics [37], pattern analysis [49], and process control applications [5][21][60].

The POD method for model reduction attempts to find a set of n_{red} basis vectors (with $n_{red} < n_x$) for the state of the system $x(k)$ such that the MSE between a state and the closest possible representation using a linear combination of the n_{red} POD basis vectors is minimal. Mathematically this means that the POD method has the objective to find a projection matrix $T_{POD} \in \mathbb{R}^{n_x \times n_{red}}$ for model reduction satisfying:

$$T_{POD} = \arg\min_{T} \overline{\mathbb{E}}\{\|x(k) - TT^T x(k)\|^2\} \quad \text{subject to} \quad T^T T = I, \tag{2.90}$$

in which

$$\overline{\mathbb{E}}\{x(k)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\{x(k)\}. \tag{2.91}$$

In order to solve the reduction criterion (2.90), it is necessary to first determine the pseudo-covariance matrix of the states of the system. The pseudo-covariance matrix of the state, denoted by $P_x \in \mathbb{R}^{n_x \times n_x}$, is defined as:

$$P_x = \overline{\mathbb{E}}\{x(k)x(k)^T\}. \tag{2.92}$$

Note that a pseudo-covariance differs from the normal covariance matrix because it is defined using the $\overline{\mathbb{E}}(\cdot)$ operator instead of the expectation operator $\mathbb{E}(\cdot)$ for a normal covariance matrix.

Assume for the moment that this matrix P_x is known. Since it holds that P_x is a nonnegative pseudo-covariance matrix, it has a complete set of orthogonal eigenvectors $p_1, \ldots, p_{n_x}$ and corresponding set of eigenvalues $\lambda_1 \geq \ldots \lambda_{n_x} \geq 0$. The matrix T_{POD} that minimizes (2.90) can then be found by choosing T_{POD} as a matrix made up of the n_{red} eigenvectors of P_x corresponding to the largest eigenvalues. As a result, the POD method thus retains those modes in the state-space in which $x(k)$ exhibits the largest variability.

It should be noted that as a result of this choice for the reduction criterion, the POD model reduction procedure is sensitive to the scaling of the states of the system. In practical situations, this means that changing the units for some elements of the original state can have consequences for the resulting reduced order model after applying POD model reduction.

Also note that the pseudo covariance matrix P_x is dependent of the input that generated the state sequence $x(k)$. The best reduction results are obtained if the inputs

[1] The POD technique is also known under the names Principle Component Analysis (PCA), Karhunen-Loéve Decomposition and Empirical Eigenfunction Analysis

$u(k)$ used to compute the pseudo-covariance P_x correspond to the inputs used during the application of the reduced order model. In many applications however the input sequence is not yet known when computing the reduced order model and thus a best guess is used for input sequence.

At this point it is important to notice that the method to obtain the POD projection matrix T_{POD} presented above can rarely be used in practice. Apart from the problem that the input sequence to compute P_x is often unknown, the dimensions of P_x are $n_x \times n_x$, thus performing computations with this matrix is generally difficult for models with $n_x >> 10^3$.

As a result, the projection matrix T_{POD} is rarely determined as described above. Instead approximative methods are commonly used. Two of these methods will be described below.

Method of snapshots

The most common method of approximately computing the projection matrix T_{POD} is the method of snapshots. For the method of snapshots, we use the property that (2.92) can be rewritten as:

$$P_x = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\{X(N)^T X(N)\}, \tag{2.93}$$

with $X(N)$ defined as:

$$X(N) = [x(1) \ \dots \ x(N)]. \tag{2.94}$$

The matrix $X(N)$ is often referred to as the snapshot matrix.

There are three main difficulties in computing the matrix P_x using (2.93):

1. In practice we cannot construct a matrix $X(N)$ that consists of an infinite number of columns.
2. The expected value operator $\mathbb{E}\{\cdot\}$ in (2.93) is generally difficult to evaluate.
3. The dimension of P_x is $n_x \times n_x$. For $n_x >> 1000$ computing the eigenvalues and eigenvectors is computationally difficult.

To overcome these difficulties, the method of snapshots computes an approximate POD basis using a pseudo covariance matrix $P_{x,snap}$ that is computed using finite length snapshot matrices $X(N)$:

$$P_{x,snap} = \frac{1}{N}X(N)X(N)^T. \tag{2.95}$$

In practice the number of snapshots N used to compute the approximate pseudo covariance matrix $P_{x,snap}$ is usually in the order of 100. Also note that in the computation of the approximate pseudo covariance matrix, the expectation operator that was present in (2.93) has been removed.

The dimension of $P_{x,snap}$ is $n_x \times n_x$. Thus computing the eigenvalues and eigenvectors of $P_{x,snap}$ is still difficult if $n_x >> 1000$. Often it is more efficient to compute the eigenvalues and eigenvectors of $P_{x,snap}$ using the singular value decomposition of $X(N)$. Define the SVD of $X(N)$ as:

$$USV^T = X(N) \tag{2.96}$$

with U,V unitary matrices and S the matrix containing the decreasing singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$. Using the SVD of $X(N)$ it can be shown that the singular values σ_i are equal to the eigenvalues of $P_{x,snap}$. Also, the $i-th$ column of U can be shown to correspond to the $i-th$ eigenvector of $P_{x,snap}$.

For $n_x >> 1000$ the size of the pseudo covariance matrix $P_{x,snap}$ is typically much larger than the snapshot matrix $X(N)$, because $N < n_x$. Since $X(N)$ is a much smaller matrix, computing the SVD of $X(N)$ is therefore computationally easier than computing an eigenvalue decomposition of $P_{x,snap}$.

Step response based method

A method very similar to POD model reduction has been proposed in [58]. Instead of using the eigenvectors of P_x as the projection basis, the eigenvectors are computed of a similar matrix P_t:

$$P_t = \int_{\mathcal{G}\in\Omega} \sum_{k=0}^{N} x_{\mathcal{G}}(k) x_{\mathcal{G}}(k)^T d\mathcal{G}, \tag{2.97}$$

with $x_{\mathcal{G}}(k)$ the state of the original model as the result of a possible input sequence denoted by $\mathcal{G}$. The integral is computed of the space Ω consisting of all possible trajectories in the working area.

The integral expression for the matrix P_t is difficult to compute exactly for nonlinear systems and thus the paper provides methods how P_t can be approximately computed. In the approximation the original nonlinear model has to be linearized in various working points of the total working area. The interpretation of P_t is quite similar to P_x in that is represents energy introduced in the states as a result of inputs.

Apart from the use of P_t instead of P_x the method is very similar to the more common POD method of snapshots.

Goal-oriented model reduction

The generic POD model reduction techniques described above do not consider the goal for which the reduced order model is used. In our case this means the POD model we obtain with the described methods is generally not the optimal reduced order for monitoring purposes. In fact, currently no model technique exists that can derive a reduced order model that is optimal for monitoring.

Recently a paper was published that modifies the generic POD method to a goal-oriented form of model reduction [99]. The suggested approach to obtain a goal-oriented form of model reduction is to add extra constraints to (2.90). These extra

constraints can be used to influence the properties of the reduced order model. The cited paper does not specifically consider the monitoring problem and as such no specific constraints are provided such that an optimal reduced order model for monitoring can be obtained.

2.4.3 Balancing

Linear balancing

An important model reduction technique often used in control applications is balancing. Balancing attempts to retain those state directions that require little energy to excite and yet are responsible for relatively large fluctuations of the output. This section summarizes the main concepts of balancing. To simplify notation, we will only discuss balanced reduction for deterministic models, so $w(k) = 0$ and $v(k) = 0$ for all k.

In order to move the state of the system from $x(0) = 0$ to $x(\infty) = x^*$ requires that we apply an appropriate input signal $u(k)$. The minimum amount of energy that is required for the input signal $u(k)$ to steer the state from 0 to x^* can be computed using the controllability function $L_c(x)$:

$$L_c(x^*) = \min_{u(k)} \sum_{k=0}^{\infty} u(k)^T u(k) \quad \text{s.t. } x(0) = 0 \text{ and } x(\infty) = x^*. \tag{2.98}$$

For linear systems (2.23)-(2.24) it can be shown that:

$$L_c(x^*) = x^{*T} M_c^{-1} x^*. \tag{2.99}$$

In this expression M_c is called the controllability Gramian of the system. From equation (2.99) it can be concluded that the eigenvectors of M_c corresponding to large eigenvalues require little input energy to reach while state directions that correspond to the eigenvectors of M_c that corresponds to a smaller eigenvalue, require relatively much energy to reach.

Balancing not only considers the amount of input energy required to reach certain states, it also considers the amount of output energy that is observed, while the system state returns from a certain state x^* to its equilibrium. The amount of output energy that is observed as the state returns to its equilibrium is defined as:

$$L_o(x^*) = \sum_{k=0}^{\infty} y(k)^T y(k), \quad \text{when } x(0) = x^* \text{ and } u(k) = 0 \text{ for } k \geq 0. \tag{2.100}$$

Again, for linear systems it can be shown that:

$$L_o(x^*) = x^{*T} M_o x^*, \tag{2.101}$$

in which M_o is called the observability Gramian of the system. From this last equation it can be concluded that the eigenvectors of M_o that correspond to large eigenvalues

are the state direction which cause relatively much output energy. Similarly, the eigenvectors of M_o that correspond to small eigenvalues are the state direction which cause relatively little output energy.

The controllability and observability Gramians have two important properties. First, it can be shown that the eigenvalues and eigenvectors of the product $M_c M_o$ are invariant under similarity transform. The similarity transform of a linear system (2.23)-(2.24) using the full rank matrix P is a new linear system with a new state variable $z(k) = P^{-1}x(k)$, but it still describes exactly the same input-output behavior as the original model. The similarity transform using a full rank matrix P is given by:

$$\begin{aligned} z(k) &= P^{-1}APz(k) + P^{-1}Bu(k) && (2.102)\\ y(k) &= CPz(k) + Du(k). && (2.103) \end{aligned}$$

Secondly, it can by shown that there exists a similarity transform Σ that transforms the system to its balanced form. The balanced form of the system is that similarity transform of the original system for which it holds that:

$$M_c^{bal} = M_o^{bal} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{n_x} \end{bmatrix}. \qquad (2.104)$$

In this balanced form, relatively large eigenvalues λ_i correspond to those directions in the state-space that can both be reached using using little input energy and, at the same time, that cause the most output energy.

Thus, to obtain a reduced order model that retains those directions in the state-space that are both easily excited and produce much output energy, the model reduction matrix T in equation (2.85) is thus chosen to correspond to the first n_{red} columns from Σ.

Comparison between balancing and POD

Whereas the POD method was argued to be sensitive to scaling of the states, it can be shown that balancing is not affected by state transformations. Unfortunately, where POD was insensitive to scaling of the inputs, balancing is sensitive to both scaling of the inputs and scaling of the outputs.

When comparing balancing and POD for the specific purpose of state estimation, the technique that produces the best reduced order models cannot be easily determined in advance. Experience tells us that models obtained using balancing tend to be better for state estimation. This is especially the case if the most controllable directions are not observable. In this case the POD method will tend to retain the controllable states even though the available measurements provide no extra information on the current amplitude of these state directions.

In order to compute the projection matrix T using balancing we require both the input and output Gramian. Computing each Gramian involves solving a Lyapunov

equation. This makes balancing computationally more difficult than the POD. Using specialized algorithms however (see for instance [9]), it can be shown that it is still possible to quickly compute the balanced truncation of systems that have a state dimension in the order of $\sim 10^4$.

Nonlinear balancing

The balancing technique outlined in the previous section can also be extended to nonlinear systems [83]. The extension to nonlinear systems requires finding expressions as (2.98)-(2.104) for the controllability and observability functions $L_c(x)$ and $L_o(x)$ that are valid in the nonlinear case. In [83] it was shown that these functions can be obtained as the solution of Lyapunov and Hamilton-Jacobi type of equations. A nonlinear extension of singular values is used to transform the system into its balanced form. This form can again be used to derive the reduction projection matrix T of (2.85).

Unfortunately, the analytical operations that are required to perform balanced reduction using the results from [83], are not computationally feasible for nonlinear systems above a certain order. As a result several approximate nonlinear balancing algorithms have been developed. For instance, in [71] a Monte Carlo approach was introduced to approximate $L_c(x)$ and $L_o(x)$ for a two dimensional model. Another approach is chosen in [51], where specific test signals are applied to the high order model to find data-driven expressions for the controllability and observability matrix.

2.5 Summary and conclusions

This chapter has mainly served to review some of the available results in the literature and introduce notation that will be used in later chapters. Given the subject of this thesis and the problem formulation in the previous chapter we specifically focused on available results for state estimation and model reduction.

On the subject of state estimation we first mathematically formulated the state estimation problem and presented the general solution. For the special case in which process model is linear and both process disturbances and measurement noises are Gaussian, the optimal solution is given by the famous Kalman filter equations. If state estimation is to be performed using either nonlinear models or models for which disturbances are not Gaussian, computing the optimal state estimates often becomes computationally infeasible. As a result many sub-optimal estimators have been developed. Some of the more commonly used filters were presented in this chapter. The sub-optimal filters were classified into four groups. Although each group of filters has its own properties, all filters require many model evaluations with the nonlinear model to compute the state estimate.

The second part of this chapter described projection based techniques for model reduction. Only projection based techniques were presented because of their generic applicability. Within the class of projection based methods we described the popular POD and balancing techniques.

For the purpose of state estimation, model reduction should ideally aim at delivering reduced order models that provide a state estimate as close as possible to the estimate that would have been obtained using the original model. However, we showed that no currently available model reduction technique is able to achieve this objective.

Chapter 3

Reducing the simulation time of large scale models

3.1 Introduction

As mentioned in Chapter 1, the use of large scale first principles models for online monitoring applications is often hampered by the relatively large amount of CPU time required per model evaluation. As described in Chapter 2, large scale first principles models are generally obtained using finite elements or finite differences methods with a very fine spatial grid. Consequently, the obtained state-space models (see (3.1) and (3.2)) are characterized by a very high state dimension and complicated non-linear functions $f(\cdot)$ and $h(\cdot)$:

$$x(k+1) = f(x(k),u(k),w(k)) \tag{3.1}$$

$$y(k) = h(x(k),u(k),v(k)). \tag{3.2}$$

Often, the amount of CPU time required to compute a single evaluation of $f(\cdot)$ and $h(\cdot)$ is at least in the order of the sampling interval. This large computation time limits the use of large-scale first principles models to off-line simulation studies. An online task such as state estimation is indeed made impossible by that excessive simulation time since the computation of a state estimate based on a nonlinear model such as (3.1)-(3.2) requires many model evaluations. For most advanced state estimation techniques, the required number of model evaluations is in the order of the number of states.

Since large-scale models can not be used for online tasks due to computational constraints, we require methods that can reduce the computational complexity of large-scale models. Based on the considerations above, this can be done using two different methods:

1. by reducing the number of states (in order to reduce the number of evaluations of the model required for the state estimation),
2. by reducing the complexity of the non-linear functions $f(\cdot)$ and $h(\cdot)$ (in order to reduce the time required for each of the remaining evaluations).

Both types of reduction are discussed in the sequel.

As the number of model evaluations that are required for state estimation is dependent on the dimension of the state vector of the model, the computation time required to obtain a state estimate can be reduced drastically by using a reduced order model instead of the original first principles model (3.1)-(3.2) for state estimation. This reduced order model has the general form:

$$x_{red}(k+1) = f_{red}(x_{red}(k), u(k), w(k)) \quad (3.3)$$
$$y(k) = h_{red}(x_{red}(k), u(k), v(k)), \quad (3.4)$$

with $dim(x_{red}) << dim(x)$. This reduced order model can for instance be derived using projection methods, such as described in section 2.4.

If the reduced order model is used for state estimation, the required number of model evaluations per sampling interval is in the order of $dim(x_{red})$. For the original full order model, in the order of $dim(x)$ model evaluations would have been required. The number of model evaluations can thus be drastically reduced using a reduced order model.

Even with the reduction in the required number of model evaluations, using a reduced order model alone does not provide enough computational gain to allow online state estimation. As will be shown in this chapter, the computational effort needed to evaluate the model (3.3)-(3.4) is generally equivalent to the computational effort needed to evaluate the original model [1]. The computation time required to estimate the state will thus still be too large for on-line applications.

To enable state estimation for complex industrial processes, we thus require in addition to state reduction techniques, techniques that can find new models that approximate (3.3)-(3.4), but require considerably less CPU time per simulation. These computationally simpler models should be of the form:

$$x_{red}(k+1) = f_{fast}(x_{red}(k), u(k), w(k)) \quad (3.5)$$
$$y(k) = h_{fast}(x_{red}(k), u(k), v(k)). \quad (3.6)$$

Methods available in the literature that attempt to reduce the computational effort for simulations with large-scale first principles models can roughly be divided into four main categories.

1. The first category of methods consists of optimizing the spatial grid of the finite element method generating the large-scale model (3.3)-(3.4), see section 2.2. This could lead to a reduction of grid cells and thus also to a reduction of the number of computations per model evaluation. The drawback of this approach is that by decreasing the number of grid cells, the model accuracy also decreases. To minimize the decrease in accuracy, specialist knowledge is required to identify those areas in which a courser grid would hardly influence the results.

[1] Note nevertheless that, in cases where the model is available in implicit form, some increase in simulation speed can sometimes be obtained. This decrease in CPU time per model evaluation is mostly only minor [4][92]. In fact, only when the original model (3.1)-(3.2) is linear will model reduction result in a model that can be evaluated significantly faster than the original full order model.

2. The second class of methods attempts to simplify the physical equations that were used to generate the model. Physical relations are simplified by ignoring higher order terms, or neglecting certain effects. While good results can be obtained in this manner [30], this approach is highly problem specific, and requires process specialists to perform the model simplification.

3. A very popular third approach to derive a faster simulation model is to linearize (3.1)-(3.2) in a chosen working point before performing projection based model reduction. For linear models, projection based model reduction instantly results in a model that requires much less computational effort for performing model evaluations. Obvious drawback of this method is that linearization only results in reasonably accurate models if the original system was already close to linear in the intended working area.

4. The final approach encountered in the literature to find faster approximative models is based on system identification. In [40], equation (3.3) is approximated using a linear state-space model that has the states of the reduced order model as outputs:

$$\begin{aligned} \zeta(k+1) &= \hat{A}\zeta(k) + \hat{B}_u u(k) + \hat{B}_w w(k) && (3.7) \\ x_{red}(k) &= \hat{C}\zeta(k) + \hat{D}_u u(k) + v(k). && (3.8) \end{aligned}$$

   The state-space matrices $\hat{A}, \hat{B}_u, \hat{B}_w, \hat{C}, \hat{D}_u$ are identified using a subspace estimator on simulation data generated by (3.3)-(3.4). The resulting model can be interpreted as a linear approximation to the nonlinear model. The fact that the resulting model is linear is also the main drawback of the method. If the original model shows significant nonlinearities, the usefulness of the identified model is limited.

In this chapter, two novel methods to accelerate computations with the original model are presented. The first method attempts the decrease the required amount of computations by only computing the original model for a subset of all the states of the nonlinear model. The remaining states are reconstructed using linear relations that are based on spatial and temporal correlations.

The second method is an identification based approach that can be seen as an extension of the method in [40]. Instead of approximating the reduced order nonlinear system with a single linear model, several locally linear models are used in a quasi-Linear Parameter Varying (qLPV) model structure.

We are approximating a known model (3.3)-(3.4) for simulation purposes. In simulations both the deterministic inputs $u(k)$ and stochastic inputs $w(k)$ and $v(k)$ are known. Since $u(k)$, $w(k)$ and $v(k)$ are all known, there is no fundamental difference between the deterministic inputs $u(k)$ and stochastic inputs $w(k)$ and $v(k)$. Without loss of generality we can therefore simplify notation throughout this section by only considering process models of the form:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) && (3.9) \\ y(k) &= h(x(k), u(k)), && (3.10) \end{aligned}$$

where we only have inputs $u(k)$.

Since the amount of CPU time required to perform a model evaluation is generally dominated by the state equation (3.9), this chapter only considers methods for finding approximations for this function.

3.2 Computational complexity of the reduced order model

If the original first principles model (3.9)-(3.10) is only available in an explicit form (no access to the underlying model equations), than any reduced order model obtained using a projection based model reduction technique will have the following form:

$$x_{red}(k+1) = T^{\dagger} f(T x_{red}(k), u(k)) \tag{3.11}$$
$$y(k) = h(T x_{red}(k), u(k)), \tag{3.12}$$

with $T \in \mathbb{R}^{n_x \times n_{red}}$ the projection matrix that has full column rank and $T^{\dagger}$ is the pseudo-inverse of T defined by:

$$T^{\dagger} = (T^T T)^{-1} T^T. \tag{3.13}$$

To compute the state update for a reduced order model (3.11) the following procedure is required:

Procedure 3.1 *If a model (3.9) is only available in explicit form, computing the state update of the corresponding reduced order model (3.11) is done using the following steps:*

1. *Expand the reduced state $x_{red}(k)$ to approximate $x(k)$ via*

$$x(k) = T x_{red}(k). \tag{3.14}$$

2. *Compute new state $x(k+1)$ via (3.9).*

3. *Compute reduced state $x_{red}(k+1)$ from $x(k+1)$ via*

$$x_{red}(k+1) = T^{\dagger} x(k+1). \tag{3.15}$$

So in order to update the reduced order model, we not only need to compute the original high order model, but we also have to do two linear projections. As a result the number of calculations required to compute (3.11) is actually larger than the number of calculations required to update the original high order model.

3.3 Partitioning the large-scale model

3.3.1 Introduction

Generally, by far the most time consuming step in updating (3.11) using Procedure 3.1 is the update of the original state vector via (3.9). Therefore the computational complexity of a model evaluation could be severely reduced if it would not be necessary to compute (3.9) for all of the original state elements, but only for a subset of the original state elements. For this purpose, we will create an approximate model that only requires a part of the complete full state to be calculated. Later in this section, it will be shown that the information in the new state that is lost by only using a part of the state update function, can be recovered using spatial and temporal correlations.

In order for this acceleration to be possible, we thus require that the original state can be partitioned into at least two parts for which the update can be calculated separately. Mathematically this means that we assume that (3.9) can be rewritten as:

$$\begin{bmatrix} x^{[1]}(k+1) \\ x^{[2]}(k+1) \end{bmatrix} = \begin{bmatrix} f_{[1]}(x(k),u(k)) \\ f_{[2]}(x(k),u(k)) \end{bmatrix}. \tag{3.16}$$

Note that it is allowed to reorder the elements of the original state $x(k)$ before partitioning the state model (3.9) as in (3.16). A requirement for the partition (3.16) is that after partitioning the time required to compute $f_{[1]}(\cdot)$ should be significantly less than the time required to compute the update of the original full order model $f(\cdot)$.

In the remainder of this section, we will use the partitioned model (3.16) to approximately compute a model evaluation of (3.11) using less CPU time. In order to compute this approximate model evaluation, the following procedure will be used:

Procedure 3.2 *Using the partitioned model (3.16), we will use the following procedure to approximately compute the state equation (3.11):*

1. *Expand the reduced order state to approximate $x(k)$ via (3.14).*
2. *Compute the partial state $x^{[1]}(k+1)$ via:*

$$x^{[1]}(k+1) = f_{[1]}(x(k),u(k)). \tag{3.17}$$

3. *Compute the new reduced order state $\tilde{x}_{red}(k+1)$ using only $x^{[1]}(k+1)$. This reduced order state $\tilde{x}(k+1)$ should be computed such that it minimizes*

$$\|x_{red}(k+1) - \tilde{x}_{red}(k+1)\|^2, \tag{3.18}$$

with $x_{red}(k+1)$ computed using (3.11).

In Procedure 3.2 only the elements in $x^{[1]}(k+1)$ are computed using the first principles model. Only computing a partial state update has two important consequences. Firstly, computing a state update using Procedure 3.2 is expected to be faster than computing the state update using Procedure 3.1, because computing the partial state

update using $f_{[1]}(\cdot)$ is simpler than computing a full state update. A second consequence is that if we only compute the state elements $x^{[1]}(k+1)$, we can no longer exactly calculate the new reduced order state using (3.15) in the third step of the procedure. The new reduced state $x_{red}(k+1)$ now has to be approximated using only those elements $x^{[1]}(k+1)$ of the full state vector that are calculated.

The method that is used to construct $\tilde{x}_{red}(k+1)$ using only $x^{[1]}(k+1)$ will greatly influence the accuracy of states computed using Procedure 3.2. In the remainder of this section two methods to perform this estimation will be discussed.

The first method is based on a least squares collocation scheme. Then, a second method is introduced that incorporates more knowledge of the system to reduce the estimation error. Both estimators of $x_{red}(k+1)$ in this section will be linear, since these allow a rapid computation of the estimate for the new reduced state.

Even though we only use linear estimators for the new reduced state, the accelerated model will remain nonlinear, since we update $x^{[1]}(k+1)$ with the first part of the original nonlinear model (see 3.16). Therefore the methods discussed in this section can also be interpreted as a partial linearization of the original system.

3.3.2 Least squares based estimation of $x_{red}(k+1)$

The first method to perform the estimation of $x_{red}(k+1)$ based on $x^{[1]}(k+1)$ is a least squares based method inspired by the missing data problem in [49]. It can be shown that the reduced order state $x_{red}(k+1)$ computed using the Galerkin projection (3.11) is also the solution to the following least squares problem:

$$x_{red}(k+1) = \arg\min_{\hat{x}_{red}(k+1)} \|f(Tx_{red}(k), u(k)) - T\hat{x}_{red}(k+1)\|^2. \tag{3.19}$$

So the new reduced order state is that state which minimizes a least squares criterion in which all states are weighted equally. If we choose to compute only a subset of the original state, a natural method to obtain an approximation of the new reduced state vector would be to replace the least squares criterion over all elements with a criterion that only takes into account the elements of the partial state $x^{[1]}(k+1)$. If we partition the long original state vector as in (3.16) and only compute the elements in $x^{[1]}(k+1)$, the new criterion will thus be:

$$\tilde{x}_{red}(k+1) = \arg\min_{\hat{x}_{red}(k+1)} \|f_{[1]}(Tx_{red}(k), u(k)) - T_1\hat{x}_{red}(k+1)\|^2, \tag{3.20}$$

where T_1 is a matrix consisting of the rows of T that correspond to the elements in $x^{[1]}(k+1)$. The solution of this least squares problem is given by:

$$\tilde{x}_{red}(k+1) = T_1^{\dagger} f_{[1]}(Tx_{red}(k), u(k)), \tag{3.21}$$

where $T_1^{\dagger}$ is the pseudo inverse of T_1. By doing so, our procedure to update $x_{red}(k)$ becomes:

1. Expand the reduced state: $x(k) = Tx_{red}(k)$ via (3.14).

2. Compute new partial state via $x^{[1]}(k+1) = f_{[1]}(x(k), u(k))$.

3. Compute the approximate new reduced state $\tilde{x}_{red}(k+1) = T_1^{\dagger} x^{[1]}(k+1)$.

This new procedure involves considerably less computation time, because the computation time of step 2 (the most time consuming) and 3 are drastically decreased, since the operators $f(\cdot)$ and $T^{\dagger}$ are replaced by operators $f_{[1]}(\cdot)$ and $T_1^{\dagger}$ of lower computational complexity.

The approximate new reduced order state $\tilde{x}_{red}(k+1)$ is the best possible linear estimate using only $x^{[1]}(k+1)$. Experience with this technique shows that quite often it is possible to generate a reasonable approximation even if $x^{[1]}(k+1)$ contains only a limited fraction of the total number of states.

3.3.3 Improved estimation using spatial correlation information

In the second step of Procedure 3.2, we only evaluate the state equation $f_{[1]}(\cdot)$ to compute a partial new state $x^{[1]}(k+1)$. As a result we do not have the information to compute $x_{red}(k+1)$ that was contained in the neglected vector $x^{[2]}(k+1)$. In this subsection we will show how a part of this lost information can be recovered without losing the advantage of a shorter computation time.

To recover some of the information contained in $x^{[2]}(k+1)$ during the computation of $\tilde{x}_{red}(k+1)$, we will use a stochastic framework, even though all variables $x^{[1]}(k+1)$, $x^{[2]}(k+1)$, $u(k)$ are deterministic variables. Using this stochastic framework, better reconstruction of $x_{red}(k+1)$ can be computed by taking into account the correlation between $x^{[1]}(k+1)$ and $x^{[2]}(k+1)$, but also, as will be shown later, the correlation between $x_{red}(k+1)$ and the same vector one time step earlier (ie. $x_{red}(k)$) as well as the correlation between $x_{red}(k+1)$ and $u(k)$. The correlation between $x^{[1]}(k+1)$ and $x^{[2]}(k+1)$ gives us spatial information, because it relates states at the same time. The correlation between $x_{red}(k+1)$ and $x_{red}(k)$ and the correlation between $x_{red}(k+1)$ and $u(k)$ gives us temporal information, since it relates quantities at different times.

In this subsection we will improve the estimate $\hat{x}_{red}(k+1)$ using correlation between $x^{[1]}(k+1)$ and $x^{[2]}(k+1)$ (spatial information). In the next section the estimator will be further improved by adding temporal information.

Our method for improving the accuracy of the reconstructed state $\tilde{x}_{red}(k+1)$ is based on the notion of the BLUE [3]. The BLUE was already discussed in section 2.3.3 but is summarized here for convenience. If two random variables X and Y have means μ_X and μ_Y and covariances:

$$\mathbb{E}\begin{bmatrix} X \\ Y \end{bmatrix}[X\ Y] = \begin{bmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{bmatrix}, \tag{3.22}$$

then the best linear unbiased estimate of X given $Y = y$ is:

$$\hat{x} = \mu_X + R_{XY} R_{YY}^{-1}(y - \mu_Y). \tag{3.23}$$

The error covariance matrix of this estimate is given by:

$$\mathbb{E}(x - \hat{x})(x - \hat{x})^T = R_{XX} - R_{XY} R_{YY}^{-1} R_{YX}. \tag{3.24}$$

In the remainder the BLUE will also be used to exploit the correlations between $x^{[1]}(k)$ and $x^{[2]}(k)$ to improve the accuracy of the reconstructed new reduced order state $\tilde{x}_{red}(k+1)$. Similar to equation (3.22), the required spatial covariance information to enable the use of the BLUE is made up of both the mean μ_x and the covariance matrix R_{xx} of $x(k) = [x^{[1]}(k)^T \; x^{[2]}(k)^T]^T$:

$$\mu_{x(k)} = \begin{bmatrix} \mu_{x^{[1]}(k)} \\ \mu_{x^{[2]}(k)} \end{bmatrix} = \mathbb{E}x(k) \tag{3.25}$$

$$R_{x(k)x(k)} = \begin{bmatrix} R_{x^{[1]}x^{[1]}} & R_{x^{[1]}x^{[2]}} \\ R_{x^{[2]}x^{[1]}} & R_{x^{[2]}x^{[2]}} \end{bmatrix} = \mathbb{E}(x(k) - \mu_{x(k)})(x(k) - \mu_{x(k)})^T. \tag{3.26}$$

Assuming that the required means and covariance information is available, we can use the spatial correlations to improve the accuracy for the approximate new reduced order state $\tilde{x}_{red}(k+1)$ in the third step of Procedure 3.2, using the BLUE equations. The improved reconstruction step for $\tilde{x}_{red}(k+1)$ is presented in the following proposition:

Proposition 3.3 *The linear reconstruction technique for $\tilde{x}_{red}(k+1)$ that minimizes*

$$\mathbb{E}\|\tilde{x}_{red}(k+1) - x_{red}(k+1)\|^2 \tag{3.27}$$

using $x^{[1]}(k+1)$ (see (3.16)), mean $\mu_{x(k)}$ (see (3.25)) and covariance matrix $R_{x(k)x(k)}$ (see (3.26)) in the third step of Procedure 3.2 is:

$$\begin{aligned} &\tilde{x}_{red}(k+1) = \mu_{x_{red}(k+1)} + \\ &\left([T^\dagger]_{[1]} + [T^\dagger]_{[2]} R_{x^{[2]}(k+1)x^{[1]}(k+1)} R^{-1}_{x^{[1]}(k+1)x^{[1]}(k+1)}\right)(x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)}), \end{aligned} \tag{3.28}$$

in which $\mu_{x_{red}(k+1)}$ is given by:

$$\mu_{x_{red}(k+1)} = T\mu_x(k+1) = T\mu_x(k), \tag{3.29}$$

and where $[T^\dagger]_{[1]}$ and $[T^\dagger]_{[2]}$ consist of the columns of $T^\dagger$ corresponding to $x^{[1]}(k+1)$ and $x^{[2]}(k+1)$:

$$T^\dagger = \left[[T^\dagger]_{[1]} \; [T^\dagger]_{[2]}\right]. \tag{3.30}$$

Proof After substituting $x_{red}(k+1)$ for X and $x^{[1]}(k+1)$ for Y in the BLUE equation (3.23) above, we obtain the following expression for $\tilde{x}_{red}(k+1)$:

$$\tilde{x}_{red}(k+1) = \mu_{x_{red}(k+1)} + R_{x_{red}(k+1)x^{[1]}(k+1)} R^{-1}_{x^{[1]}(k+1)x^{[1]}(k+1)} (x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)}) \tag{3.31}$$

in which $\mu_{x_{red}(k+1)}$ is the mean of $x_{red}(k+1)$ and $R_{x_{red}(k+1)x^{[1]}(k+1)}$ is the covariance between $x_{red}(k+1)$ and $x^{[1]}(k+1)$. In the previous expression, covariance matrix $R_{x_{red}(k+1)x^{[1]}(k+1)}$ is equal to:

$$R_{x_{red}(k+1)x^{[1]}(k+1)} = \mathbb{E}(x_{red}(k+1) - \mu_{x_{red}(k+1)})(x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)})^T \tag{3.32}$$

$$
\begin{aligned}
&= \mathbb{E}\left\{ \left[[T^{\dagger}]_{[1]} \; [T^{\dagger}]_{[2]} \right] \begin{bmatrix} x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)} \\ x^{[2]}(k+1) - \mu_{x^{[2]}(k+1)} \end{bmatrix} (x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)})^T \right\} \\
&= [T^{\dagger}]_{[1]} R_{x^{[1]}(k+1)x^{[1]}(k+1)} + [T^{\dagger}]_{[2]} R_{x^{[2]}(k+1)x^{[1]}(k+1)}, \qquad (3.33)
\end{aligned}
$$

Equation (3.28) now follows after substituting (3.33) in (3.31). ∎

By applying (3.24) the error covariance matrix of the BLUE reconstruction $\tilde{x}_{red}(k+1)$ is:

$$
\begin{aligned}
&\mathbb{E}\left\{ (\hat{x}_{red}(k+1) - x_{red}(k+1))(\hat{x}_{red}(k+1) - x_{red}(k+1))^T \right\} = \\
&[T^{\dagger}]_{[2]} \left[R_{x^{[2]}(k+1)x^{[2]}(k+1)} - R_{x^{[2]}(k+1)x^{[1]}(k+1)} R^{-1}_{x^{[1]}(k+1)x^{[1]}(k+1)} R_{x^{[1]}(k+1)x^{[2]}(k+1)} \right] [T^{\dagger}]_{[2]}{}^T . \qquad (3.34)
\end{aligned}
$$

The expected approximation error between $x_{red}(k+1)$ and $\tilde{x}_{red}(k+1)$ (see (3.18)) can be calculated by computing the trace of the error covariance matrix defined in (3.34). As a result, we can use the expression for the error covariance matrix (3.34) to determine a partition (3.16) such that on average the we obtain an optimal approximate new reduced order state $\tilde{x}_{red}(k+1)$.

The required spatial means and covariance information (3.25)-(3.26) can be estimated offline using a procedure that is similar to the manner in which the method of snapshots for POD models reduction computes the matrix P_x (see section 2.4.2). Using inputs similar to those expected in practice, we simulate a state sequence $x(1), x(2), \ldots, x(N)$ using the reduced order model (3.3). The required mean in (3.25) is estimated using:

$$
\hat{\mu}_{x(k)} = \frac{1}{N} \sum_{k=1}^{N} x(k). \qquad (3.35)
$$

The required spatial covariance matix is then estimated using:

$$
\hat{R}_{x(k)x(k)} = \frac{1}{N} \sum_{k=1}^{N} (x(k) - \hat{\mu}_{x(k)})(x(k) - \hat{\mu}_{x(k)})^T. \qquad (3.36)
$$

Note that the estimated mean $\hat{\mu}_{x(k)}$ and covariance matrix $\hat{R}_{x(k)x(k)}$ are dependent on the input signal $u(1), \ldots, u(N)$ used to generate the state sequence $x(1), \ldots, x(N)$. For good results the input sequence $u(k)$ used to estimate $\hat{\mu}_{x(k)}$ and $\hat{R}_{x(k)x(k)}$ in the offline experiment should resemble the inputs used when applying the accelerated Procedure 3.2.

Since that the BLU estimate is optimal given the available spatial information in (3.25)-(3.26), the BLU estimate (3.28) is better than the least squares based estimate (3.21), provided that the spatial information (3.25)-(3.26) is accurate. The fact that the spatial information has been estimated (and is thus only approximately known) can cause that the BLUE is not always the better estimator.

3.3.4 Improved estimates using both spatial and temporal covariances

So far we have used the BLUE to incorporate only spatial information, i.e. we have only used the relation between state elements at the same time instant. We can refine the estimate even further by also including temporal information, i.e. the relation between the state $x_{red}(k+1)$ and the previous state $x_{red}(k)$ and previous input vector $u(k)$. Since the system is in state-space form, all information about the past is contained in these two quantities.

The BLUE constructs an approximate reduced order state $\tilde{x}_{red}(k+1)$ from $x^{[1]}(k+1)$, $x_{red}(k)$ and $u(k)$ using the following relation:

$$\hat{x}_{red}(k+1) = \mu_{x_{red}(k+1)} + VW^{-1} \begin{bmatrix} x^{[1]}(k+1) - \mu_{x^{[1]}(k+1)} \\ x_{red}(k) - \mu_{x_{red}(k)} \\ u(k) - \mu_{u(k)} \end{bmatrix} \tag{3.37}$$

with

$$V = \begin{bmatrix} R_{x_{red}(k+1)x^{[1]}(k+1)} & R_{x_{red}(k+1)x_{red}(k)} & R_{x_{red}(k+1)u(k)} \end{bmatrix} \tag{3.38}$$

and

$$W = \begin{bmatrix} R_{x^{[1]}(k+1)x^{[1]}(k+1)} & R_{x^{[1]}(k+1)x_{red}(k)} & R_{x^{[1]}(k+1)u(k)} \\ R_{x_{red}(k)x^{[1]}(k+1)} & R_{x_{red}(k)x_{red}(k)} & R_{x_{red}(k)u(k)} \\ R_{u(k)x^{[1]}(k+1)} & R_{u(k)x_{red}(k)} & R_{u(k)u(k)} \end{bmatrix}. \tag{3.39}$$

This result follows directly from the BLUE relation (3.23). Like in the previous sections the required means and covariance matrices V and W will have to be estimated in a separate experiment.

Using generic BLUE error covariance equation (3.24) we can show that the expected covariance matrix of $(\tilde{x}_{red}(k+1) - x_{red}(k+1))$ is given by:

$$\mathbb{E}\left\{ (\tilde{x}_{red}(k+1) - x_{red}(k+1))(\tilde{x}_{red}(k+1) - x_{red}(k+1))^T \right\} = R_{x_{red}(k+1)x_{red}(k+1)} - VW^{-1}V^T. \tag{3.40}$$

The new BLU estimator that also includes the temporal information will in general be more accurate than the previous BLU estimate. Accuracy is especially improved if there is significant correlation between the current reduced state and the previous state and input (e.g. slowly varying systems).

The required means $\mu_{u(k)}$, $\mu_{x_{red}(k)}$, $\mu_{x^{[1]}(k+1)}$ and covariance matrices V and W have to be estimated before BLUE can be used to compute $\tilde{x}_{red}(k+1)$. To estimate this information a long input sequence $u(k)$ for $k = 1, \ldots, N$ is used to generate states $x(k)$ using the full order model (3.9). Using this data we construct the vector $\xi(k)$ as:

$$\xi(k)^T = [x^{[1]}(k+1)^T \; x_{red}(k)^T \; u(k)^T]. \tag{3.41}$$

The means $\mu_{u(k)}$, $\mu_{x_{red}(k)}$ and $\mu_{x^{[1]}(k+1)}$ can then be estimated via:

$$\begin{bmatrix} \hat{\mu}_{u(k)} \\ \hat{\mu}_{x_{red}(k)} \\ \hat{\mu}_{x^{[1]}(k+1)} \end{bmatrix} = \mu_\xi = \frac{1}{N}\sum_{k=1}^{N} \xi(k). \tag{3.42}$$

The covariance matrix W can be estimated using:

$$\hat{W} = \frac{1}{N}\sum_{k=1}^{N} (\xi(k) - \hat{\mu}_\xi)(\hat{\xi}(k) - \hat{\mu}_\xi)^T. \tag{3.43}$$

Finally V can be estimated the following equation:

$$\hat{V} = \frac{1}{N}\sum_{k=1}^{N} (x_{red}(k+1) - \mu_{x_{red}(k+1)})(\xi(k) - \hat{\mu}_\xi)^T. \tag{3.44}$$

If all the required means and covariances are accurately known, the new BLUE reconstruction using both spatial and temporal information will be more accurate than both the least squares method and the BLUE using only temporal information.

In practice however, if the means and covariances in (3.37) are not determined accurately enough, then (3.37) even be worse than the previously discussed methods. Similarly, if the required means and covariances have been estimated using an unrepresentative input sequence, using spatial and temporal covariance could cause inaccurate results.

3.3.5 Practical issues for implementation

So far, we have not mentioned how the partitioning (3.16) should be done. The chosen partition can seriously influence the accuracy of the accelerated model. An optimal partition for the BLUE techniques can be found by searching for that partition that minimizes the expected error (3.27). The expected error (3.27) can be computed by taking the trace of (3.34) or (3.40), depending on whether (3.28) or (3.37) is used to compute $\tilde{x}_{red}(k+1)$.

Finding the optimal partition (3.16) can thus be formulated as a optimization problem. Unfortunately, the complexity of the optimization problem increases exponentially with the state dimension n_x, because the number of possible partitions increases exponentially with n_x. Sometimes however, it may be possible to create a good partition using physical insight in the process model.

In both BLU estimators (3.28) and (3.37) the inverse of an auto-covariance matrix is required. Often the auto-covariance matrices will be ill-conditioned. If this is indeed the case, it is advisable to approximately compute the required inverse matrix using a truncated SVD, see [31].

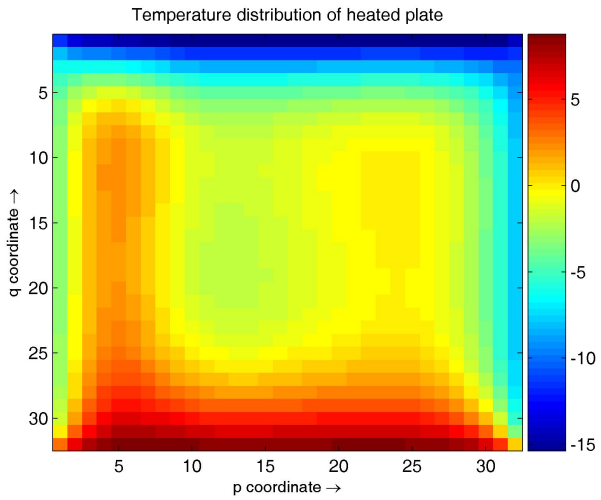

Figure 3.1 Simulated heated plate example. The plate is heated or cooled along the complete edges on all sides. The physical equation are solved on a 32 by 32 grid using explicit Euler integration.

3.4 Simulation example

3.4.1 Simulation model

In this section the presented methods of section 3.3 will be illustrated by a simulation example. The chosen example is that of an iron solid square plate that is heated and cooled at the edges, see Figure 3.1. Before applying the previously described approximation techniques, first the plate model will be discussed.

The sides of the plate are $L = 0.5$ meters in length, the height of the plate is chosen to be $h = 0.01$ meters. All required material properties for iron are obtained from [41]. Each side of the solid plate is connected to a surface of which the temperature can be controlled. The temperatures of each surface will serve as an input. The model thus has four inputs.

The model of the plate is constructed from the energy balance over an infinitesimal small surface area at spatial coordinates (p,q)[2]:

$$\rho\, c\, h\, \mathrm{d}p\, \mathrm{d}q\, \frac{\partial T(p,q,t)}{\partial t} = \nabla \cdot J(p,q,t), \tag{3.45}$$

in which ρ is the mass-density of the plate, c is the specific heat of the plate. The ∇

[2]We use the spatial coordinates (p,q) instead of the more conventional spatial coordinates (x,y) to prevent confusion with the state vector $x(k)$ and measurement vector $y(k)$.

operator is defined as:

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial p} \\ \frac{\partial}{\partial q} \end{pmatrix}. \tag{3.46}$$

The function $J(p,q,t) : \mathbb{R}^3 \to \mathbb{R}^2$ is a vector function that represents the heat transfer at location (p,q) at time t. The heat transfer function $J(p,q,t)$ is given by:

$$J(p,q,t) = \lambda(T(p,q,t))\nabla T(p,q,t), \tag{3.47}$$

with $\lambda(T(p,q,t))$ a temperature-dependent heat conductivity coefficient. For many materials reasonably accurate models can be derived using a constant heat conductivity. Using a constant λ would result in a linear model. In the simulation example however, it was assumed that $\lambda(T(p,q,t))$ is temperature dependent:

$$\lambda(T(p,q,t)) = \frac{1}{2560}T(p,q,t)^3 + \frac{3}{8}T(p,q,t) + 80. \tag{3.48}$$

The heat conductivity as a function of temperature is plotted in Figure 3.2. A temperature dependent heat conductivity is physically more accurate than the more common constant heat conductivity assumption. It should be noted that this particular heat conductivity function is not based on physics. Instead the function was chosen such that the model would have some reasonable nonlinear characteristics such that it provides a more interesting simulation model for testing the previously described model approximation techniques.

The model (3.45)-(3.48) can be rewritten in the nonlinear state-space form using a finite differences method. The finite differences method imposes a grid on the plate. We have chosen to use a grid of 32 by 32 elements. In each element the temperature and all other material properties are assumed to be constant. In the remainder we shall denote the p-coordinate (horizontal position) of a column of grid-cells as $p(k_p)$ with $k_p \in \{1,\ldots,32\}$. Similarly the q-coordinate (vertical position) of a row of cells is denoted as $q(k_q)$ with $k_q \in \{1,\ldots,32\}$. Spatial derivatives with respect to spatial coordinate p and q are approximated using finite differences. For example, the spatial derivative with respect to p is approximated using the following equation:

$$\left.\frac{\partial T(p,q,t)}{\partial p}\right|_{p=p(k_p),q,t} \approx \frac{T(p(k_p+1),q,t) - T(p(k_p),q,t)}{p(k_p+1) - p(k_p)}. \tag{3.49}$$

The partial derivatives with respect to q are approximated in the same manner.

After approximating the spatial derivatives using finite differences, the resulting approximate model is a set of 1024 nonlinear ordinary differential equations. These equations are solved using a simple explicit Euler method, using an integration time interval of 20 seconds.

The resulting model is a model of the form (3.9)-(3.10). The state $x(k)$ in the model represents the temperature at each of the 1024 grid elements. The inputs $u(k)$ are the temperatures of each of the sides.

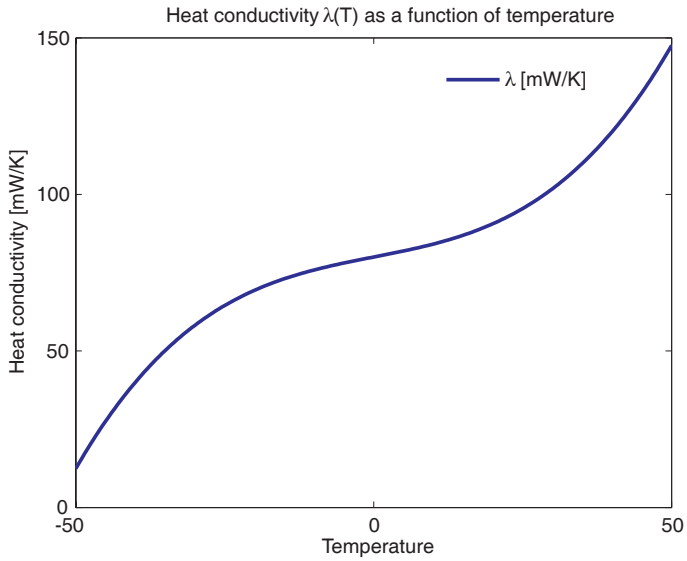

Figure 3.2 Plot of the chosen temperature-dependent heat conductivity function $\lambda(T)$ as given by (3.48).

The input signal for each of the sides is chosen to be a randomly generated series of steps. An example realization of the applied input signals is depicted in Figure 3.3. For $k = 0, 40, 80, \ldots$ a new temperature for each of the sides chosen randomly:

$$u(k) \sim \mathcal{N}(0, Q) \quad \text{with} \quad Q = \text{diag}\{225, 225, 100, 100\} \tag{3.50}$$

with $u(k)$ for $k = 0, 40, 80, \ldots$ a white noise process. For $k \neq 0, 40, 80, \ldots$, (the intermediate sampling instants) the input at time k is determined via:

$$u(k) = u(k-1). \tag{3.51}$$

Since the explicit Euler method is used to approximately solve the nonlinear differential equations, it is easy to partition the state update as in (3.16). It should thus be relatively straightforward to apply the model approximation techniques as described in the previous section.

The computation time required to update only that part of the state vector that is contained in $x^{[1]}$ scales linearly with the dimension of $x^{[1]}$. Denoting the computation time required for a full model update as t_{full} and the computation time required to update only the elements in $x^{[1]}$ as $t_{x^{[1]}}$, it thus holds that:

$$t_{x^{[1]}} \approx \frac{\dim(x^{[1]})}{1024} t_{full}. \tag{3.52}$$

Before the model approximation techniques can be applied, a reduced order model is required. A reduced order model of the plate has been constructed using POD model

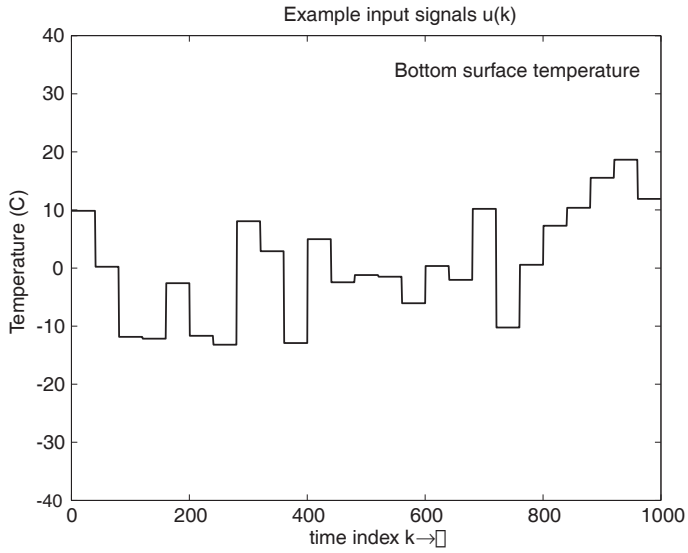

Figure 3.3 Example input signal u(k). The depicted input signal represents the imposed temperatures at the bottom of the heated plate. Similar signals are applied to the other sides of the plate.

reduction as described in 2.4.2. Using the POD technique a very good approximative model could be constructed using a reduced order model of only 25. The input data that was used for estimating the covariance matrix $R_{x(k)x(k)}$ (required determining the projection matrix T) has been estimated using a 2000 point simulation run, with inputs chosen as described above.

3.4.2 Model acceleration results

In the previous section we have derived both a full-order model of the form (3.9) as well as a reduced order model of the form (3.11) for the heated plate. In this section we will apply the techniques described in the previous section to accelerate (approximate) evaluations of the reduced order model. Specifically, we will use the method outlined in Procedure 3.2.

The first step of Procedure 3.2 requires that we have a reduced order model available of the form (3.11). Such a reduced order model has already been derived at the end of the previous section. For the second step it is required that partial state model evaluations can be computed. As argued at the end of the previous section, this too can be easily accomplished for our heated plate model.

Before we can compute partial state updates we first need to partition the full order state $x(k)$ into $x^{[1]}(k)$ and $x^{[2]}(k)$ (see (3.16)). In this simulation example it has been chosen to distribute the states $x^{[1]}(k)$ evenly across the heated plate in a square pattern, an example of such a pattern is shown in Figure 3.4. The number of computed

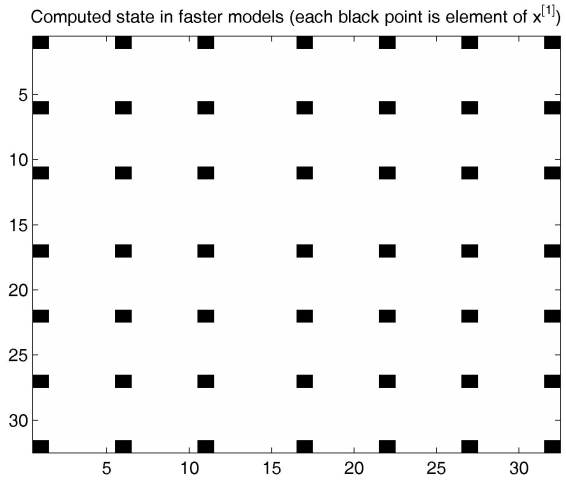

Figure 3.4 Illustration of square pattern chosen for elements $x^{[1]}$. The location of the elements of $x[1]$ are represented by black squares.

elements per side was varied from 2 to 24. Thus the number of states in $x^{[1]}(k)$ was varied from 4 to 576.

The final step in the computationally faster Procedure 3.21 is to compute the approximate new reduced order state $\tilde{x}_{red}(k+1)$ using only the elements in $x^{[1]}(k+1)$. In order to compute $\tilde{x}_{red}(k+1)$ three methods have been presented in the previous sections:

- using the least squares method, see (3.21);
- using spatial correlations, see (3.28);
- using both spatial and temporal equations (3.37).

The required covariance matrices for the methods that use correlations are estimated using a separate experiment. For this experiment 5000 computed states are generated using inputs as described in (3.50)-(3.51). Then the required covariance matrices for the method using only spatial correlations are estimated using (3.35)-(3.36). The required covariance information for the method using both spatial and temporal correlations is estimated using (3.41)-(3.44).

In order to compare the accuracy of the different accelerated models, we need to choose an error criterium. The criterium that will be used is a scaled one step ahead prediction criterium on a new set of $N_{val} = 1000$ point of validation data:

$$Err = \frac{1}{1024} \sum_{k=1}^{N_{val}} \frac{1}{N_{val}} \|x_{red}(k+1) - f_{fast}(x_{red}(k), u(k))\|^2, \tag{3.53}$$

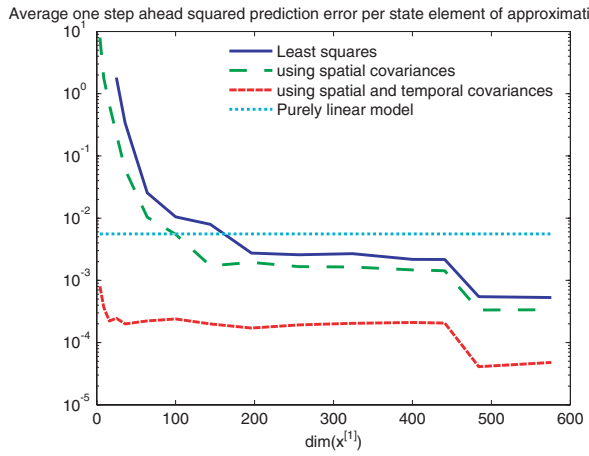

Figure 3.5 The average one step ahead approximation error (3.53) as a function of the number of elements in $x^{[1]}$ for the heated plate example. The three lines correspond to the accuracy of the reconstruction in the third step of Procedure 3.2. For reference the accuracy of a linear model identified using prediction error identification methods is also provided (horizontal line).

with $u(k)$ a new set of inputs generated using (3.50)-(3.51) and $x_{red}(k)$ the corresponding new set of reduced order state vectors. The scaling factor $1/1024$ is used such that the error measure corresponds to the averaged squared prediction error per grid element of the plate. The results of each of the three partitioning variants are plotted in Figure 3.5.

For reference, a linear model of the form:

$$x_{red}(k+1) = \hat{A}x_{red}(k) + \hat{B}u(k) \tag{3.54}$$

has been estimated as well. The matrices $\hat{A}$, $\hat{B}$ have been determined using prediction error identification using the same data that was used to determine the covariance matrices for the partitioning methods. The prediction error of the linear model is also plotted in Figure 3.5.

The least squares method is able to produce good results for $\dim(x^{[1]}) > 200$. With good results it is meant that the method results in smaller prediction errors than the purely linear model (3.54). Since good results can be obtained using $\dim(x^{[1]}) > 200$, (3.52) an approximate model evaluation using Procedure 3.2 can be evaluated approximately five times faster than the reduced order model original model (see (3.52)). Note that the least squares method cannot be implemented for $\dim(x^{[1]}) < 25$. This is because the left inverse $T_1^{\dagger}$ in (3.21), does not exist for these cases.

The accelerated model that exploits knowledge of the spatial covariances is more accurate than the least squares model, irrespective of the dimension of $x^{[1]}$. As a result, good accuracy is already obtained starting $\dim(x^{[1]}) > 100$, so that it is possible to construct a good approximative model that is approximately ten times faster than

Table 3.1 Accuracy of the faster heated plate models obtained using partitioning. The accuracy of each partitioning method is expressed using the one step ahead prediction error measure as defined in (3.53). The simulation time to compute 1000 model evaluations for each of the partitioning methods is provided as well.

$\dim(x^{[1]})$	Err_{ls}	Err_{sp}	$Err_{sp+temp}$	t_{ls}	t_{sp}	$t_{sp+temp}$
4	-	6.51	$6.71 \cdot 10^{-4}$	-	9.2s	10.7s
36	0.19	$6.2 \cdot 10^{-2}$	$1.1 \cdot 10^{-4}$	14.1s	14.6s	15.7s
100	$8.1 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$	29.1	28.3	30.8
196	$2.2 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$	51.1s	50.6s	52.9s
324	$2.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	98.4s	100.4s	101.9s

the original model.

If an approximate model is constructed using both spatial and temporal correlations, the accuracy of the resulting model increases dramatically. Very accurate results are already obtained for $\dim(x^{[1]}) = 4$. As a result, it is thus possible to construct approximate models that require considerably less computations!

3.5 Identifying an approximate quasi-LPV model

3.5.1 Introduction

An alternative for the partitioning method described in the previous sections is the identification based approach. In this approach the reduced order model (3.3)-(3.4) derived from the original first principles model, is approximated using a computationally simpler black box model, while retaining the physical meaning of the state vector $x_{red}(k)$. The objective in the identification based approach is to approximate the already available (but computationally expensive) reduced order model

$$x_{red}(k+1) = f_{red}(x_{red}(k), u(k)) \tag{3.55}$$

with a model with an user defined simpler generic black box model structure $f_{id}(x_{red}(k), u(k), \theta)$:

$$x_{red}(k+1) = f_{id}(x_{red}(k), u(k), \theta), \tag{3.56}$$

with θ a vector of free model parameters.

The main difference between the identification based approach presented in this section and the partitioning approach discussed in the previous sections is that once the resulting faster approximation model $f_{id}(\cdot)$ (3.5) has been identified, the original model will no longer be required to compute (partial) state updates as in Procedure 3.1.

The identification problem posed in this section is to approximate the static nonlinear function $f_{red}(\cdot)$ with $n_x + n_u$ inputs and n_x outputs by a simpler model of the form $f_{id}(\cdot)$.

Especially for nonlinear identification many different model structures for $f_{id}(\cdot)$ can be chosen. Examples of nonlinear model structures found in the literature are Wiener and Hammerstein models [103], NARX [22] models, NARMAX models, Linear Parameter Varying models [94], etcetera. The model structure should be determined such that:

$$\exists \theta \quad \text{s.t.} \quad \overline{\mathbb{E}} \| f_{red}(x_{red}(k), u(k)) - f_{id}(x_{red}, u(k), \theta) \|^2 < \alpha \tag{3.57}$$

with α an user defined limit for the largest acceptable expected prediction error. Finding a model structure which satisfies this criterion is not straightforward. In practice model structure selection is performed using either prior knowledge or by trial and error.

The choice to approximate the reduced order model instead of attempting to approximate the original full order first principles model is mainly motivated by practical issues. Approximating the full order model would quickly become infeasible, because the manipulations required in an identification algorithm are computationally infeasible when the state vector $x(k)$ has a very high dimension. Secondly, even if a successful identification of the full order model would be possible, the model would still need to be reduced in order to be used for state estimation applications.

The identification based approach can be split up into two steps. First a model structure is selected. Once the model structure has been selected, model parameters θ should be identified such that the identified model provides the best possible approximation to the original model.

In this section the quasi-LPV model structure will be used to approximate the model (3.55). A quasi-LPV model uses a state-dependent and input-dependent linear combination of linear models to approximate the original model's behavior:

$$\begin{aligned} f_{id}(x_{red}(k), u(k)) = {} & A_0(\theta) x_{red}(k) + B_0(\theta) u(k) + x_{off,0} + \\ & \sum_{m=1}^{M} \phi_m(x_{red}, u(k), \theta) \left[A_m(\theta) x_{red}(k) + B_m(\theta) u(k) + x_{off,m} \right], \end{aligned} \tag{3.58}$$

with $(A_0, B_0, x_{off,0})$ the state-space matrices representing the best global linear model and $(A_m, B_m, x_{off,m})$ component linear models that are summed after being weighted according to scheduling functions $\phi_m(\cdot)$. Due to the scheduling functions $\phi_m(\cdot)$, the behavior of the qLPV model differs from the basis linear model $(A_0, B_0, x_{off,0})$. The scheduling functions determine how the modelled behavior should change depending on the current state and inputs to the model.

The difference between a standard LPV model and a quasi-LPV model is that in a standard LPV model the scheduling functions $\phi(\cdot)$ are not allowed to be a function of either the inputs $u(k)$ or the current state $x(k)$.

We will refer to model structures specified by (3.58) as quasi Linear Parameter

Varying (qLPV) models, but in the literature these model structures are also referred to as local linear models [94] or fuzzy models [6].

The approach to identify an approximate state model was first suggested in [40]. In this work a linear model structure was used to approximate the $f_{red}(\cdot)$. Compared to the linear model structure used in [40], the model structure (3.58) is obviously more flexible to describe the nonlinear behavior of the original model. Even using a relatively small number of component models and relatively simple scheduling functions $\phi_m(\cdot)$ a wide variety of nonlinear models can be approximated. The main drawback of using the qLPV model structure is that even for a relatively small number of components models $A_m, B_m, x_{off,m}$ the identification problem is already far more difficult than for linear models. Not only is it necessary to determine a single linear model given by $A_0, B_0, x_{off,0}$, but we also need to find optimal component models $A_m, B_m, x_{off,m}$ and scheduling functions $\phi_m(\cdot)$ for $m = 1, 2, \ldots, M$. A second problem with using qLPV type of model structures, is that it is hard to guarantee that the identified models will be stable. It can be shown that even when all models $A_0, B_0, \ldots, A_M, B_M$ are stable, the resulting state dependent sum of stable models can be unstable [81].

The problem of identifying (q)LPV models has been extensively studied in nonlinear identification literature. Current methods to identify model structures of the form (3.58) can be divided into the following categories:

- Simultaneous identifications of scheduling functions $\phi_m(\cdot)$ and models A_m, B_m, $x_{off,m}$ for $m = 0, \ldots, M$, see for instance [95][7][54].
- Functions $\phi_m(\cdot)$ are assumed known, identify only $A_0, B_0, x_{off,0} \ldots A_M$, B_M, $x_{off,M}$.
- Two stage methods: first determine scheduling functions $\phi_m(\cdot)$ then identify $A_m, B_m, x_{off,m}$ for $m = 0, 1, \ldots, M$ [43][6].

All the identification approaches in the literature use experimental data consisting of measured inputs $u(k)$ and measured outputs $y(k)$ to perform the identification. An explicit physical plant model of the form (3.55) is never assumed available. As a result the goal in the identification problem is to find that model which best predicts the output of the model as a function of its inputs. The interpretation of the state in this problem setting is of no consequence. Thus the state vector $x_{red}(k)$ of the identified qLPV model will not have the same physical interpretation as in (3.55).

In this section we will introduce a new method to identify qLPV models. There are two main differences between our approach and the approaches in the literature. The first difference is that our procedure will use simulation data $Z^N = u(1), x_{red}(1), \ldots, u(N), x_{red}(N)$ generated with (3.55) instead of measured input and output signals. The advantages of using simulation data instead of measured data are threefold:

- In simulation data we know the state $x_{red}(k)$ which will allow us to retain the physical interpretation of the state vector x_{red} in the identified model.

- To obtain suitable practical data (expensive) experiments may be required, while simulation data only requires computer time.

- There is no measurement noise in simulation data.

The second difference between our method and qLPV identification methods in literature is that in conventional qLPV identification methods, an available first principles process models is not used, while in our method the availability of the first principles model is a strict requirement.

Our approach is a two stage method:

1. In contrary to most qLPV identification methods we shall first determine matrices $A_0, B_0, x_{off,0} \ldots A_M, B_M, x_{off,M}$ in the model structure (3.58). These matrices are determined using linearizations of the reduced order model (3.55) and by applying techniques similar to those in model reduction.

2. Then in a second step we parameterize functions $\phi_m(\cdot, \theta)$ and estimate the optimal values for θ using simulation data.

The first step of the described method is presented in section 3.5.2, the second step of the method is discussed in section 3.5.3. A discussion of some theoretical aspects of the proposed methods and practical issues for its application are contained in 3.5.4. Section 3.6 illustrates the use of the proposed method in a simulation example.

3.5.2 Determination of $A_0, B_0, x_{off,0}, \ldots, A_M, B_M, x_{off,M}$

In this section a method is presented to determine linear models parameterized by $A_0, B_0, x_{off,0}, \ldots, A_M, B_M, x_{off,M}$ in (3.58). To determine these matrices and vectors, data will be used that is generated by the available simulation model (3.55) :

Assumption 3.4 *The data $Z^N = \{u(1), x_{red}(1), \ldots, u(N), x_{red}(N)\}$ is generated by the reduced order model (3.55). It will be assumed that the available data Z^N is representative for the whole working area of the reduced order model (3.55).*

The assumption above states that the dataset Z^N should be such that it is possible to find a model of the form (3.58) that is accurate in the entire working area. For nonlinear models it is not trivial to verify that a given dataset has this property. In our identification procedure we will ignore this problem for now.

The method presented in this section to determine $A_0, B_0, x_{off,M}, \ldots, A_M$, B_M, $x_{off,M}$ consists of the following steps.

1. Determine $A_0, B_0, x_{off,0}$ such that the model given by $A_0, B_0, x_{off,0}$ is the best linear model for the available simulation data Z^N.

2. Determine the matrices $A_1^l, B_1^l, x_{off,1}^l, \ldots A_N^l, B_N^l, x_{off,N}^l$ and coefficients

$\beta_1^l(k),\ldots,\beta_N^l(k)$ of an intermediate LPV expansion $f_{id}^{long}(\cdot)$ of the form:

$$f_{id}^{long}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{N} \beta_m^l(k)\left[A_m^l x_{red}(k) + B_m^l u(k) + x_{off,m}^l\right] \quad (3.59)$$

such that the resulting expansion matches the reduced order model $f_{red}(\cdot)$ for all $(x_{red}(k),u(k)) \in Z^N$:

$$f_{red}(x_{red}(k),u(k)) = f_{id}^{long}(x_{red}(k),u(k)) \quad \forall \quad (x_{red}(k),u(k)) \in Z^N. \quad (3.60)$$

Note that the length of the expansion (3.59) is N, the number of simulation data. As a result, this expansion is generally very long.

3. In this step we use the previously computed expansion $f_{id}^{long}(\cdot)$ to compute a shorter expansion $f_{id}^{short}(\cdot)$ of the form:

$$f_{id}^{short}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{M} \beta_m(k)\left[A_m x_{red}(k) + B_m u(k) + x_{off,m}\right] \quad (3.61)$$

with $M << N$. For this new expansion $f_{id}^{short}(\cdot)$ new matrices $A_1,B_1,\ldots,A_M,B_M$, vectors $x_{off,1},\ldots,x_{off,M}$ and coefficients $\beta_1(k),\ldots,\beta_M(k)$ will be computed such that for the available simulation data Z^N the prediction error of the new shorter expansion is smaller than an user-defined low constant α:

$$\frac{\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k)) - f_{id}^{short}(x_{red}(k),u(k))\|^2}{\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k))\|^2} < \alpha, \quad (3.62)$$

with α some chosen small value, for instance $\alpha = 0.01$.

The first step of the procedure a model $A_0,B_0,x_{off,0}$ is identified that accounts any possible linear trends in the data Z^N.

In the second step the data $u(m),x(m)$ is mapped to a long expansion in matrices $A_m^l,B_m^l,x_{off,l}$, for $m = 1,\ldots,N$.

After the data has been mapped to the matrices $A_m^l,B_m^l,x_{off,l}$, for $m = 1,\ldots,N$, we can use techniques similar to those used in model reduction to reduce the length of the expansion.

Step 1: Determination of $A_0,B_0,x_{off,0}$

The model structure (3.58) can be interpreted as follows. The original model (3.55) can be approximated to some extent using just a linear model specified by A_0,B_0,

$x_{off,0}$, but due to nonlinearities in the original reduced order model (3.55) it is necessary to adjust the model parameters for the various regions in the operating domain of the original model. The changes in model parameters are parameterized by state and input dependent linear combinations of the linear models $A_1, B_1, x_{off,1} \ldots, A_M, B_M, x_{off,M}$.

In this interpretation the matrices $A_0, B_0, x_{off,0}$ specify the basic linear model that is to be refined later using linear combinations of the parameter change models $A_1, B_1, x_{off,1} \ldots, A_M, B_M, x_{off,M}$. To minimize the required number of required corrections, the model $A_0, B_0, x_{off,0}$ should already approximate the original reduced order model (3.55) as good as possible.

As a result, we will determine the model A_0, B_0, $x_{off,0}$ such that it is the best possible linear approximation to (3.55) for the intended working area. Since the data Z^N is assumed to be representative for the entire working area (see assumption 3.4), this can be accomplished by identifying $A_0, B_0, x_{off,0}$ via:

$$[A_0, B_0, x_{off,0}] = \arg \min_{\tilde{A}_0, \tilde{B}_0, \tilde{x}_{off,0}} \sum_{k=1}^{N} \frac{1}{N} \Big\| x_{red}(k+1) - \tilde{A}_0 x_{red}(k) - \tilde{B}_0 u(k) - \tilde{x}_{off,0} \Big\|^2. \quad (3.63)$$

The presented least squares criterion to determine $A_0, B_0, x_{off,0}$ is linear in $\tilde{A}_0, \tilde{B}_0$ and $\tilde{x}_{off,0}$. As a result, A_0, B_0 and $x_{off,0}$ can be easily determined by solving the linear least squares optimization problem.

Step 2: Determination of $f_{id}^{long}(\cdot)$

The second step is to determine the matrices $A_1^l, B_1^l, x_{off,1}^l, \ldots, A_N^l, B_N^l, x_{off,N}^l$ and coefficients $\beta_1^l(k), \ldots, \beta_N^l(k)$ of an expansion $f_{id}^{long}(\cdot)$ such that for all data $(x_{red}(k), u(k)) \in Z^N$ it holds that:

$$f_{red}(x_{red}(k), u(k)) = f_{id}^{long}(x_{red}(k), u(k)), \quad (3.64)$$

with $f_{red}(\cdot)$ the known reduced order first principles model that is to be approximated, and $f_{id}^{long}(\cdot)$ defined as in (3.59).

For this purpose we introduce the following notation for the linear model $\mathcal{L}(x_{red}(k), u(k))$ obtained by linearizing (3.55) around $(x_{red}(k), u(k))$:

$$\mathcal{L}(x_{red}(k), u(k)) = \mathcal{A}(x_{red}(k), u(k)) x_{red}(k) + \mathcal{B}(x_{red}(k), u(k)) u(k) + \mathcal{X}_{off}(x_{red}(k), u(k)), \quad (3.65)$$

with

$$\mathcal{A}(x_{red}(k), u(k)) = \left. \frac{\partial f_{red}(x_{red}, u)}{\partial x_{red}} \right|_{x_{red}=x_{red}(k), u=u(k)}, \quad (3.66)$$

$$\mathcal{B}(x_{red}(k), u(k)) = \left. \frac{\partial f_{red}(x_{red}, u)}{\partial u} \right|_{x_{red}=x_{red}(k), u=u(k)}, \quad (3.67)$$

and

$$\mathcal{X}_{off}(x_{red}(k),u(k)) = f_{red}(x_{red}(k),u(k)) - \mathcal{A}(x_{red}(k),u(k))x_{red}(k) - \mathcal{B}(x_{red}(k),u(k))u(k). \quad (3.68)$$

Using this notation, the following proposition can be used to construct the expansion $f_{id}^{long}(\cdot)$:

Proposition 3.5 *Given a linear system with offset defined by $A_0, B_0, x_{off,0}$ there exists an expansion $f_{id}^{long}(x_{red}, u(k))$ defined as in (3.59) such that for all pairs $(x_{red}(k), u(k)) \in Z^N$ it holds that:*

$$f_{red}(x_{red}(k),u(k)) = f_{id}^{long}(x_{red}(k),u(k)), \quad (3.69)$$

where $f_{id}^{long}(\cdot)$ is given by:

$$\begin{aligned} A_m^l &= \mathcal{A}(x_{red}(m),u(m)) - A_0 && (3.70)\\ B_m^l &= \mathcal{B}(x_{red}(m),u(m)) - B_0 && (3.71)\\ x_{off,m}^l &= \mathcal{X}_{off}(x_{red}(m),u(m)) - x_{off,0} && (3.72)\\ \beta_m^l(k) &= \begin{cases} 1 & for \quad m = k \\ 0 & for \quad m \neq k \end{cases} && (3.73) \end{aligned}$$

for $m = 1,\ldots,N$.

Proof Using the definitions (3.65)-(3.68) it is easily verified that for all pairs $x_{red}(k), u(k)$ it holds that

$$f_{red}(x_{red}(k),u(k)) = \mathcal{A}(x_{red}(k),u(k))x_{red}(k) + \mathcal{B}(x_{red}(k),u(k))u(k) + \mathcal{X}_{off}(x_{red}(k),u(k)). \quad (3.74)$$

After adding and substraction the identified model $A_0, B_0, x_{off,0}$ we obtain:

$$\begin{aligned} f_{red}(x_{red}(k),u(k)) = {} & A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + [\mathcal{A}(x_{red}(k),u(k)) - A_0]x_{red}(k) \\ & + [\mathcal{B}(x_{red}(k),u(k)) - B_0]u(k) + [\mathcal{X}_{off}(x_{red}(k),u(k)) - x_{off,0}]. \quad (3.75) \end{aligned}$$

Substitution of (3.70)-(3.73) in (3.75) results in:

$$f_{red}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{N} \beta_m(k) \left[A_m^l x_{red}(k) + B_m^l u(k) + x_{off,m}^l \right]. \quad (3.76)$$

And thus for all $(x_{red}(k),u(k)) \in Z^N$ we have:

$$f_{red}(x_{red}(k),u(k)) = f_{id}^{long}(x_{red}(k),u(k)). \quad (3.77)$$

■

Step 3: Determination of $A_1, B_1, \ldots, A_M, B_M$ and M

In the previous section it was shown that an expansion $f_{id}^{long}(\cdot)$ can be constructed that exactly matches $f_{red}(\cdot)$ for the simulation data Z^N. Still, given the length N of the expansion, it is not practical to use the obtained matrices $A_1^l, B_1^l, x_{off,1} \ldots, A_N^l, B_N^l, x_{off,N}$ to construct our final model (3.58).

In this section we shall use the long expansion $f_{id}^{long}(\cdot)$ constructed using proposition 3.5 to construct a much shorter expansion $f_{id}^{short}(\cdot)$ of length $M << N$ with new matrices $A_1, B_1, x_{off,1} \ldots, A_M, B_M, x_{off,M}$ and coefficients $\beta_1(k), \ldots, \beta_M(k)$. The resulting expansion is determined such that $f_{id}^{short}(\cdot)$ satisfies:

$$\frac{\sum_{k=1}^{N} \| f_{red}(x_{red}(k), u(k)) - f_{id}^{short}(x_{red}(k), u(k)) \|^2}{\sum_{k=1}^{N} \| f_{red}(x_{red}(k), u(k)) \|^2} < \alpha, \tag{3.78}$$

in other words, using the computed matrices $A_1, B_1, x_{off,1} \ldots, A_M, B_M, x_{off,M}$ there should exist coefficients $\beta_1(k), \ldots, \beta_M(k)$ such that the relative prediction error (see (3.78)) of the shorter model $f_{id}^{short}(\cdot)$ averaged over the available simulation data Z^N should be smaller than a chosen constant α.

Before presenting the method to construct the expansion $f_{id}^{short}(\cdot)$, we will first introduce some notation. Define parameter vectors $\mathcal{V}_m^s$ for $m = 1, \ldots, M$ as:

$$\mathcal{V}_m^s = [\mathrm{vec}(A_m)^T \ \mathrm{vec}(B_m)^T \ x_{off,m}^T]^T, \tag{3.79}$$

with $\mathrm{vec}(\cdot)$ the operator that transforms an arbitrary matrix into a vector, by stacking its columns. Similarly we define parameter vectors $\mathcal{V}_m^l$ for $m = 1, \ldots, N$ as

$$\mathcal{V}_m^l = [\mathrm{vec}(A_m^l)^T \ \mathrm{vec}(B_m^l)^T \ x_{off,m}^l{}^T]^T. \tag{3.80}$$

Using this notation the problem of finding appropriate matrices $A_m, B_m, x_{off,m}$ for the short expansion $f_{id}^{short}(\cdot)$ is equivalent to the problem of finding parameter vectors V_m^s for $m = 1, \ldots, M$.

The problem of finding vectors $\mathcal{V}_m^s$ for the short expansion such that (3.78) is satisfied is not trivial, because each vector $\mathcal{V}_m^s$ represents a linear model. We propose the following procedure to compute the required component models:

Procedure 3.6 *The following algorithm produces parameter vectors $\mathcal{V}_1^s, \ldots, \mathcal{V}_M^s$ such that (3.78) holds:*

1. *Set $\lambda = 1$.*
2. *Determine parameter vectors $\mathcal{V}_1, \ldots, \mathcal{V}_\lambda$ and coefficients $\beta_1(k), \ldots, \beta_\lambda(k)$ for $k = 1, \ldots, N$ such that:*

$$\mathcal{V}_1^s, \ldots, \mathcal{V}_\lambda^s, \beta_1(1), \ldots, \beta_\lambda(k) = \arg \min_{\tilde{\mathcal{V}}_1^s, \ldots, \tilde{\mathcal{V}}_\lambda^s, \tilde{\beta}_1(1), \ldots, \tilde{\beta}_\lambda(k)} \sum_{k=1}^{N} \left\| \mathcal{V}_k^l - \sum_{m=1}^{\lambda} \tilde{\beta}_m(k) \tilde{\mathcal{V}}_m^s \right\|^2. \tag{3.81}$$

The resulting λ parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$ have the property that on average they can best approximate the parameter vectors $\mathcal{V}_m^l$ for $m=1,\ldots,N$ using a linear combination of the obtained parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$. The parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$ and coefficients $\beta_1(k),\ldots,\beta_\lambda(k)$ for $k=1,\ldots,N$ can be efficiently computed using a singular value decomposition.

3. *Compute the prediction error of the short expansion specified by $\mathcal{V}_1,\ldots,\mathcal{V}_\lambda$ and coefficients $\beta_1(k),\ldots,\beta_\lambda(k)$. If the computed error satisfies (3.78) stop. Otherwise, $\lambda=\lambda+1$ and go to step 2.*

The advantage of using the parameter space criterion is that the parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$ and coefficients $\beta_1(k),\ldots,\beta_\lambda(k)$ can be computed efficiently using a SVD. To use the SVD to compute the parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$, we first need to construct a matrix X whose columns consist of the vectors $\mathcal{V}_1^l,\ldots,\mathcal{V}_N^l$:

$$X=[\mathcal{V}_1^l\ \cdots\ \mathcal{V}_N^l]. \tag{3.82}$$

If the singular value decomposition of X is denoted as:

$$X=USV^T, \tag{3.83}$$

then the parameter vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$ which satisfy (3.81) are given by the first λ columns of U.

The corresponding matrices A_1,B_1,x_{off_1} can be determined by applying the inverse of the vec$(\cdot)$ operator to the obtained vectors $\mathcal{V}_1^s,\ldots,\mathcal{V}_\lambda^s$:

$$[A_m\ B_m\ x_{off,m}]=\text{vec}^{-1}(\mathcal{V}_m^s), \tag{3.84}$$

with vec$^{-1}(\cdot)$ the inverse of the vec$(\cdot)$ operator:

$$\text{vec}^{-1}(\text{vec}(A))=A. \tag{3.85}$$

The coefficients $\beta_1(k),\ldots,\beta_\lambda(k)$ in (3.81) are determined by:

$$\beta_m(k)=\mathcal{V}_m^{s\,T}\mathcal{V}_k^l. \tag{3.86}$$

Note that the described method of determining the expansion $f_{id}^{short}(\cdot)$ is neither guaranteed to result in an expression that minimizes (3.78) using M components models, nor is the length of the resulting expansion M guaranteed to be the shortest expansion that satisfies (3.78). This is to be expected, because the parameter space criterion we used to determine our short expansion $f_{id}^{short}(\cdot)$ is not equivalent to the relative prediction error criterion (3.78). This can be explained by considering the relation between both criteria. Using (3.76) and (3.61) it can be shown that:

$$\begin{aligned} f_{red}(x_{red}(k),u(k))-f_{id}^{short}(x_{red}(k),u(k)) &= [A_k^l-\sum_{m=1}^{M}\beta_m(k)A_m]x_{red}(k) \\ &+[B_m^l-\sum_{m=1}^{M}\beta_m(k)B_m]u(k)+[x_{off}^l-\sum_{m=1}^{M}\beta_m(k)x_{off,m}]. \end{aligned} \tag{3.87}$$

In this expression, define the parameter errors $\mathcal{E}_A(k), \mathcal{E}_B(k)$ and $\mathcal{E}_{x_{off}}(k)$ as

$$\mathcal{E}_A(k) = [A_k^l - \sum_{m=1}^{M} \beta_m(k) A_m] \tag{3.88}$$

$$\mathcal{E}_B(k) = [B_k^l - \sum_{m=1}^{M} \beta_m(k) B_m] \tag{3.89}$$

$$\mathcal{E}_{x_{off}} = [x_{off}^l - \sum_{m=1}^{M} \beta_m(k) x_{off,m}] \tag{3.90}$$

then we see that as the parameter errors tend to zero (i.e. $\mathcal{E}_A(k) \to 0$, $\mathcal{E}_B(k) \to 0$ and $\mathcal{E}_{x_{off}}(k) \to 0$), the relative prediction error (3.78) will also tend to zero. Since Procedure 3.6 results in the matrices A_m, B_m, and vectors $x_{off,m}$ that on average minimize the parameter errors (3.88)-(3.90), the corresponding prediction error is expected to be small as well. Similarly, since the parameter errors quickly decrease with increasing M, using Procedure 3.6 will in general satisfy (3.78) with $M << N$.

Note that the method used to determine the matrices A_m, B_m and vectors $x_{off,m}$ for $m = 1, \ldots, M$ is similar to model reduction using the POD technique using snapshots. Recall from section 2.4.2 that POD model reduction using snapshots of sampled states $x(k)$ to determine a set of basis vectors for the reduced order model. Similarly, Procedure 3.6 uses samples of the parameters of the linearized behavior of the reduced order model to construct a shorter expansion (3.61).

3.5.3 Identification of functions $\phi_i(x_{red}(k), u(k), \theta)$

In the previous steps matrices $A_0, B_0, x_{off,0} \ldots, A_M, B_M, x_{off,M}$ were determined such that for the available identification data there exists a scheduling functions $\beta_m(k)$ such that the one step ahead prediction error (3.78) was smaller than a chosen value α. The criterion was met using scheduling coefficients $\beta_m(k)$ as computed in (3.86). The computed functions $\beta_m(k)$ using Procedure 3.6 have no structure and are highly data dependent. For a new set of data Z^N, new functions $\beta_m(k)$ would thus have to be computed to obtain a small prediction error. Because the functions $\beta_m(k)$ are unstructured, computing new functions $\beta_m(k)$ using (3.86) would involve linearizing $f_{red}(\cdot)$, which is computationally involved.

Since the coefficients $\beta_m(k)$ cannot be used as scheduling functions for computational reasons, we will use structured functions $\phi_m(\cdot)$ in the final model $f_{id}(\cdot)$ (3.58). Using a structured model as scheduling functions means that the model can also be used for new data (e.g. data not the identification set Z^N). In this section scheduling functions $\phi_m(x_{red}(k), u(k), \theta_m)$ will be identified using the available simulation data Z^N. This requires that we first choose a model structure for the functions $\phi_m(x_{red}(k), u(k), \theta_m)$. Since the functions $\phi_m(\cdot, \theta_m)$ are not required to have any physical interpretation, we are free to chose any model structure we would like. The choice for structure of these functions is a tradeoff between complexity and flexibility.

An example of a simple structure is:

$$\phi_m(x_{red}(k),u(k),\theta_m) = [x_{red}(k)^T \; u(k)^T \; 1]\theta_m. \tag{3.91}$$

The simple affine model structure for $\phi_m(\cdot,\theta_m)$ has as main advantages that the structure is linear in its parameter vector θ_m and the number of parameters is relatively small. Drawback of this model structure is that such a structure may not be flexible enough to allow for an accurate model (3.58).

If more complex scheduling functions are required, it is possible to use more complex structures using for instance radial basis functions [94] or fuzzy membership functions [6].

To select an appropriate structure for the scheduling functions, it is often helpful to look at plots of the computed coefficients $\beta_m(k)$ (see (3.86)) as a function of the states and inputs $(x_{red}(k),u(k)) \in Z^N$.

Once a structure has been selected, all that remains is to estimate the parameter vectors θ_m. This can be accomplished by minimizing the prediction error criterion for the available data Z^N:

$$\theta_1,\ldots,\theta_M = \arg\min_{\theta_1,\ldots,\theta_M} \frac{1}{N}\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k) - f_{id}(x_{red}(k),u(k)),\theta_1,\ldots,\theta_M)\|^2, \tag{3.92}$$

with $f_{id}(\cdot)$ defined as:

$$\begin{aligned} f_{id}(x_{red}(k),u(k),\theta_1,\ldots,\theta_M) &= A_0 x_{red}(k) + B_0 u(k) + x_{off,0} \\ &+ \sum_{m=1}^{M} \phi_m(x_{red}(k),u(k),\theta_m)\left[A_m x_{red}(k) + B_m u(k) + x_{off,m}\right], \end{aligned} \tag{3.93}$$

with $A_m, B_m, x_{off,m}$ as determined in section 3.5.2.

Note that if the scheduling functions $\phi_m(\cdot,\theta_m)$ are chosen linear in θ_m such as in (3.91), the resulting minimization problem is a linear least squares problem which can be easily solved. For more complex structures that are not linear in θ_m, the parameters of the scheduling functions have to be determined using nonlinear optimization techniques. For the previously mentioned structures involving radial basis functions or fuzzy membership functions, good initial conditions can be obtained by applying clustering methods on the computed coefficients $\beta_m(k)$ [6].

3.5.4 Discussion of the identification method

Use of impulse response functions

In the presented algorithm, the reduced order model (3.55) is approximated by a linear combination of state-space models, see (3.58). The component linear models specified by $A_m, B_m, x_{off,m}$ are identified using a least squares criterion in the parameter space.

A drawback of such a method is that all parameter changes of equal size are considered equally important, while in fact this is not the case. Certain parameter changes have different consequences than others. As an example consider the following first order scalar model without inputs and offsets:

$$y(k+1) = \frac{u(k)}{1-az^{-1}}, \tag{3.94}$$

with a a parameter that can change within a certain interval. Using the parameter space criterion (3.81), a parameter change in the interval [0.5, 0.7] is indistinguishable from a parameter changes in the interval [0.9, 1.1]. In reality the behavior of the system will not change much for parameter variations in the first interval, but for parameter changes in the second parameter interval, the change in system behavior is dramatic. For $a < 1$, the system is stable, while for $a > 1$ the system is unstable!

This complication can be circumvented by modifying the qLPV model structure (3.58) such that the linear combination of state-space models is replaced by a linear combination of impulse response models:

$$\begin{aligned} f_{id}(x_{red}(k),u(k)) = {} & G_0(k) * u(k) + x_{off,0} \\ & + \sum_{m=1}^{M} \phi_m(x_{red}(k),u(k)) \left[G_m(k) * u(k) + x_{off,m} \right], \end{aligned} \tag{3.95}$$

with $G_m(k)$ finite length impulse response functions of length L, and '$*$' the convolution operator:

$$G_m(k) * u(k) = \sum_{\kappa=1}^{L} G_m(\kappa) u(k-\kappa). \tag{3.96}$$

Using impulse response models, the criterion comparing model parameters directly corresponds to comparing the impulse responses of different models. The described method of identifying qLPV models can be easily adapted to produce models in this form. Unfortunately for systems with slow dynamics, using impulse response models would require the length of the impulse response function L to be very large. For such systems using impulse response functions is impractical.

The use of long impulse response functions can be avoided using orthonormal basis expansions instead of impulse response functions $G_m(k)$ [36]. Using orthonormal basis functions, all impulse response functions $G_m(k)$ are assumed to be of the form:

$$G_m(k) = \sum_{i=1}^{P} \chi_i^m \kappa_i(k), \tag{3.97}$$

with $\kappa_i(k)$ for $i = 1,\ldots,P$ a set of orthogonal basis functions of length L, χ_i^m are scalars that describe the required linear combination to obtain $G_m(k)$ and P the length of expansion. Note that the functions $\kappa_i(k)$ are fixed. Thus to describe a complete impulse response $G_m(k)$ we only need to know the P scalar coefficients χ_i. Using properly chosen basis function $\chi_i(k)$ the length of the expansion (denoted as P) can be relatively short. Methods to generate such basis function are given in [36].

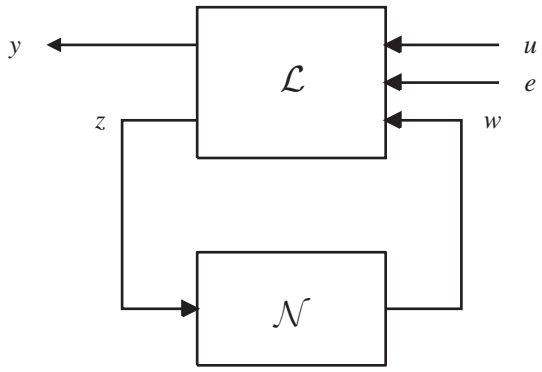

Figure 3.6 Block diagram of the LFT model structure. The block denoted by $\mathcal{L}$ contains all linear dynamic elements (the component linear models of our qLPV model) of the structure, the block $\mathcal{N}$ contains static nonlinearities (the scheduling functions $\phi_i(\cdot)$ of the qLPV model structure.

Practical issues when identifying $\phi_m(\cdot,\theta_m)$

Even for simple linear parameterizations of the scheduling functions $\phi_m(\cdot,\theta_m)$ such as (3.91), determining parameters vectors θ_m can sometimes be difficult. The difficulties occur when the least squares problem (3.92) is ill-conditioned. In such cases it is advisable to approximately solve the least squares criterion (3.92) using a truncated SVD, to effectively reduce the degrees of freedom in the least squares problem. For more details, see [31][70].

Alternative for determining scheduling functions $\phi_i(\cdot)$

Another different method for determining the scheduling functions $\phi_i(\cdot)$ can be found by rewriting the qLPV structure (3.58) as a Linear Fractional Transformation (LFT) model. A generic block diagram of a LFT model structure is depicted in Figure 3.6. Once the model has been rewritten in the LFT form, results from the LFT identification literature can be used to estimate the scheduling functions $\phi_i(\cdot)$. An example of such a technique is given in [39]. In this paper a technique is described that can be used to estimate the nonlinear elements of the LFT structure when all linear elements are known. This is exactly the case we have after we have determined the component models $[A_i, B_i, x_{off,i}]$ using the SVD-based procedure in 3.5.2.

3.6 Simulation example

To illustrate the effectiveness of the presented quasi-LPV identification method, the method will be applied in a simulation example. In the chosen simulation example the quasi-LPV identification method will be used to approximate the heated plate model

described in section 3.4.1. This model is the same model that was used to illustrate partitioning methods described in section 3.3.

Recall that in section 3.4.1 a state-space model has been derived that models the temperature distribution of a heated iron plate on a 32×32 grid. The exact model for the temperature distribution can be written as a nonlinear state-space model of order 1024 (i.e. $\dim(x(k)) = 1024$). Using POD model reduction, a reduced order model has been constructed of the form:

$$x_{red}(k+1) = f_{red}(x_{red}, u(k)). \tag{3.98}$$

This reduced order model is the same reduced order model that was determined in section 3.4.1. The order of the reduced order model is 25. Using the available reduced order model $N = 3000$ points of simulation data $Z^N = \{x_{red}(1), u(1), \ldots, x_{red}(N), u(N)\}$ is generated. The input signal $u(k)$ is chosen to be the same stepping signal (see (3.50)-(3.51)) that was used in the simulation example to illustrate partitioning methods, see section 3.4.1.

In this simulation example the available data Z^N will be used to construct two approximation models: a linear state-space model and a quasi-LPV model. The linear model is estimated so that one can judge what can potentially be gained by using a more complex model structure such as the quasi-LPV model.

The linear state-space approximation model has the following structure:

$$f_{id}(x_{red}(k), u(k)) = A_0 x_{red}(k) + B_0 u(k). \tag{3.99}$$

Note that this chosen linear model structure does not contain an offset term $x_{off,0}$. The offset term was omitted because the heated plate model does not exhibit an offset. The matrices A_0, B_0 can be easily estimated by solving the linear least squares problem (3.63).

Before continuing to construct an qLPV model, first the accuracy of the linear model will be discussed. The accuracy of the identified linear model was evaluated on a separate data set consisting of $N_{val} = 1000$ data points. This data set is the same validation data that was used to test the performance of the various partitioning methods in section 3.4.2. The error measure used to evaluate the accuracy of the linear model is again chosen to be the average one step ahead prediction error:

$$Err = \frac{1}{1024} \sum_{k=1}^{N_{val}} \frac{1}{N_{val}} \left\| f_{red}(x_{red}, u(k)) - f_{id}(x_{red}, u(k)) \right\|^2. \tag{3.100}$$

Using the identified linear model the average one step ahead prediction error over the validation data is 0.0048.

Even though the linear model is already able to closely predict the outcome of the nonlinear reduced order model, we will still attempt to obtain an even more accurate model using the qLPV model structure, using the identification procedure outlined in the previous section.

In order to determine an qLPV model, we first determine the matrices A_m, B_m using the three step procedure outlined in section 3.5.2. The first step of this procedure is to

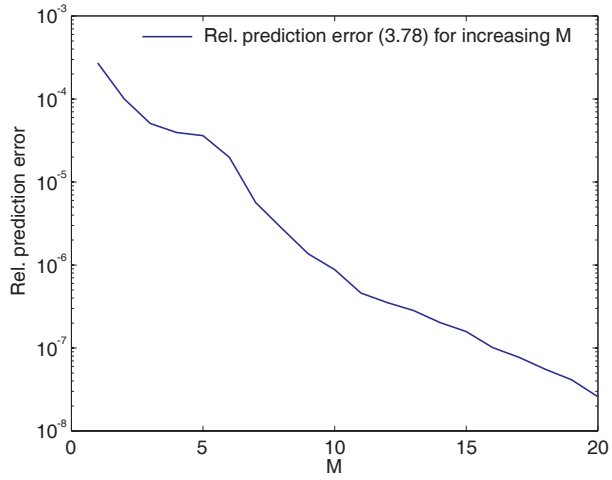

Figure 3.7 Plot of the relative prediction error criterion (3.78) as a function of the number of component models M.

construct a linear model A_0, B_0. This model is the same model as the linear model we have just estimated.

The second step in the procedure is to determine the component models A_m, B_m for $m = 1, \ldots, M$. In order to determine the component models, we first map the data Z^N to a long expansion $f_{id}^{long}(\cdot)$ in matrices A_m^l, B_m^l (see (3.59)). For this purpose we linearize the function $f_{red}(\cdot)$ for all data $(x_{red}(k), u(k)) \in Z^N$ such that the long expansion $f_{id}^{long}(\cdot)$ can be constructed using proposition 3.5. It should be noted that in constructing the expansion $f_{id}^{long}(\cdot)$, no offset terms $x_{off,m}^l$ are used. The offsets were omitted from the expansion because no offset is present.

The long expansion $f_{id}^{long}(\cdot)$ was used to derive a much shorter expansion $f_{id}^{short}(\cdot)$. The new matrices A_m, B_m for $m = 1, \ldots, M$ were computed using Procedure 3.6 as described in section 3.5.2. The number of component models M of the shorter model has been determined by examining the relative prediction error (3.78) of the shorter expansion $f_{id}^{short}(\cdot)$ as a function of M. A plot of the relative prediction error (3.78) as a function of M is given in Figure 3.7. As can be seen in the figure, the relative prediction error decreases almost exponentially for increasing values of M. For the heated plate model we will use a value of $M = 8$. For $M = 8$ the relative prediction error (3.78) for the simulated identification data Z^N using the computed coefficients $\beta_m(k)$ is $2.74 \cdot 10^{-6}$. To put this number is perspective, the linear model specified using only A_0, B_0 identified using the same test data resulted in a relative prediction error of $2.33 \cdot 10^{-4}$, which is approximately a factor 85 times larger than for the qLPV structure.

Once the matrices A_m, B_m for $m = 1, \ldots, M$ have been determined, we still need to

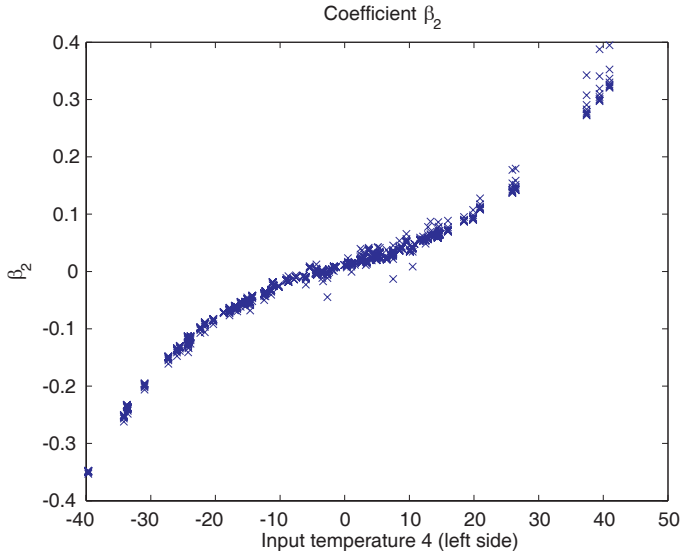

Figure 3.8 Plot of values of $\beta_2(k)$ as computed using (3.86) versus the fourth input component of u(k) (left side temperature).

construct scheduling functions $\phi_m(x_{red}(k),u(k))$ as described in section 3.5.3. To find an appropriate model structure for the scheduling functions $\phi_m(x_{red}(k),u(k))$, we use plots of the computed coefficients β_m as a function of states and inputs. An example of such a plot can be found in Figure 3.8, in which the second scheduling function $\beta_2(k)$ is plotted against the fourth component of the inputs $u_4(k)$. Based on the shape of the Figure 3.8, we can conclude that the coefficients $\beta_2(k)$ seem to behave as a third order polynomial function of $u(k)$. Similar plots of coefficients $\beta_m(k)$ against rows of $x_{red}(k)$ suggest that a similar relation exists between $\beta_m(k)$ and $x_{red}(k)$. This was of course to be expected given that the nonlinearity of the original heated plate model is caused by heat the conductivity function $\lambda(T)$ that is a third order polynomial. As a result, the following structure for the scheduling functions $\phi_m(x_{red}(k),u(k))$ was chosen:

$$\phi_m(x_{red}(k),u(k),\theta_m) = [1\ [x_{red}(k)]_{1:8}^T\ u(k)^T\ [x_{red}(k)]_{1:4}^{3\ T}\ [u(k)^3]^T]\theta_m, \qquad (3.101)$$

with $\theta_m \in \mathbb{R}^{21\times 1}$ the parameter vector of the scheduling function.

Since we have chosen a scheduling function that is linear in the parameter vectors θ_m for $m = 1,\ldots,M$, the parameter vectors θ_m can be determined by solving a least squares problem (see (3.92)).

Now that we have obtained an qLPV model its quality will be assessed using the same 1000 point validation data that was used to test both the linear model and the partitioning methods. The prediction error for the qLPV method on the validation data is $4.8\cdot 10^{-4}$ which is approximately 10 times smaller than for the linear model.

Table 3.2 Averaged prediction errors per grid cell and computation time of constructed approximate faster models of heated plate models. Averaged prediction errors per grid cell and computation time were determined for a 1000 points validation data.

Model	Err	CPU time
Original reduced order model $f_{red}(\cdot)$	0	307 s.
Partitioning using LS (dim $x^{[1]} = 196$)	$2.74 \cdot 10^{-3}$	52 s.
Partitioning using Sp. cov (dim $x^{[1]} = 196$)	$1.94 \cdot 10^{-3}$	55 s.
Partitioning using Sp. + Temp. cov. (dim $x^{[1]} = 196$)	$0.17 \cdot 10^{-3}$	55 s.
Identified Linear model	$4.85 \cdot 10^{-3}$	3 s.
Identified quasi-LPV model	$0.48 \cdot 10^{-3}$	36 s.

In this section we have used the same heated plate model and the same validation model to test the performance of the qLPV identification method, as was used to test the performance of the partitioning methods in section 3.4.2. As a result, we can compare the results of both approaches. The result of both the partitioning methods as well as the results of the qLPV identification method are summarized in Table 3.2. For each approximation method, the table lists the average prediction error and the CPU time required to perform 1000 model evaluations. For reference the CPU time for the reduced order heated plate model is also given.

The listed models obtained using partitioning are all obtained using a state partitioning of the form (3.16) in which $\dim(x^{[1]}) = 196$. The CPU time required to evaluate 1000 model evaluations is approximately 55 seconds. The accuracy of the partitioning methods ranges from $2.74 \cdot 10^{-3}$ for the least squares method (see (3.21)), to $0.17 \cdot 10^{-3}$ for the method that uses both spatial and temporal correlations.

The identified linear model (3.99) requires only 3 seconds to perform 1000 model evaluations, but its prediction error is the largest. The identified qLPV model requires 36 seconds to perform 1000 model evaluations, but its accuracy is far better than the linear model.

The heated plate model uses an explicit Euler solver to compute state updates (see section 3.4.1). Even when such a simple solver is used to compute state updates, the approximation models obtained using partitioning methods and qLPV modelling use significantly less time per model evaluation. The reduction in computational complexity of the reduced order methods leads to a reduction in CPU time by a factor of 5.6 for the partitioning methods and a factor of 8.5 for the qLPV method. Had a more complex solver been used in the original model, the expected speedup factor using the approximate models probably would have been higher.

3.7 Summary and conclusions

Model reduction alone does not enable the use of large-scale first principles models for online state estimation. Indeed, we showed that even though model reduction decreased the number of model evaluations required for state estimation, each model evaluation of the reduced order model $f_{red}(\cdot)$ still requires the evaluation of the original first principles model $f(\cdot)$.

Consequently, reduction of the state dimension alone is thus not sufficient to be able to use first principles for online state estimation. The CPU time required for a model evaluation has also to be drastically decreased. In this section, we have proposed two methodologies in order to obtain a model $f_{fast}(\cdot)$ that closely approximates $f_{red}(\cdot)$ while requiring much less CPU time per model evaluation. The first method is a partitioning approach. Here, the faster model is generated by only computing a state update for a part of the state. The remaining states are approximated using linear operations. The accuracy of the approximated states increases when spatial and temporal covariance information is used.

The second approach obtains a function $f_{fast}(\cdot)$ using identification techniques. In section 3.5 a new identification algorithm has been presented that can be used to identify a qLPV model $f_{fast}(\cdot)$ from simulated data generated using the reduced order model $f_{red}(\cdot)$.

The use of both methods has been illustrated using a simulation example, see sections 3.4.2 and 3.6. In the simulation examples faster models were computed for a first principles model of a heated plate. Both partitioning and qLPV identification resulted in models that accurately predicted the new states of the original reduced order model $f_{red}(\cdot)$ of the heated plate using only a fraction of the CPU time.

Chapter 4

Kalman filtering for poorly observable systems without noise information

4.1 Introduction

In the previous chapters it has already been discussed that both state dimension and computational complexity can be obstacles that prevent the use of state estimation techniques. In this chapter another obstacle preventing optimal filter design will be discussed. Consider the case in which the process model is a state-space model of reasonable dimension, such that both state dimension and computational complexity do not pose problems for state estimation. For such models the states of the process can be estimated using measurements related to the state. The estimated states are produced using a state filter such as the Kalman filter. The standard design methodology for such a filter requires not only a full description of the relation between the states and measurements, but also a full description of the noise affecting the states and the measurements. Like in the previous chapters we assume that such a model relating the measurements to the states has been obtained using first principles modelling. At this point it is important to note that first principles models of physical plants are generally not fully controllable and observable, due to the limited amount of measurements (i.e. the limited dimension of the measurement vector with respect to the dimension of the state vector). Moreover, although the deterministic behavior can often be derived using first principles modelling, this form of modelling only very rarely admits an accurate description of the stochastic behavior. Based on these considerations, we will address the problem of designing a reliable Kalman filter when no noise properties are available and when some singular values of the observability matrix are small or even zero. For simplicity, we will consider linear systems of the form

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+w(k) && (4.1)\\ y(k) &= Cx(k)+v(k), && (4.2) \end{aligned}$$

with $x(k) \in \mathbb{R}^{n_x \times 1}$ the state of the system at time index k, $u(k) \in \mathbb{R}^{n_u \times 1}$ the known input of the system, $w(k) \in \mathbb{R}^{n_x \times 1}$ the unknown process noise, $y(k) \in \mathbb{R}^{n_y \times 1}$ the measurement vector and $v(k) \in \mathbb{R}^{n_y \times 1}$ the measurement noise. The noises $w(k)$ and $v(k)$ are assumed to be Gaussian with:

$$\mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} = 0, \tag{4.3}$$

$$\mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix}^T \begin{bmatrix} w(l) & v(l) \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{k,l}. \tag{4.4}$$

Limiting ourselves to the class of linear models, the results of this chapter can only be applied to processes whose behavior can be modelled linearly in the intended working area. The advantage of limiting ourselves to linear processes is that we are able to provide more detailed analysis, without having to resort to approximations.

For the considered linear systems the optimal estimator for the state vector $x(k)$ is the Kalman filter [3]. The Kalman filter is optimal is the sense that it minimizes the MSE:

$$\sum_k \mathbb{E}\|x(k) - \hat{x}(k|k)\|^2. \tag{4.5}$$

The optimal predictor (in the MSE sense) can be written in the Kalman innovations form:

$$\hat{x}(k+1|k) = A\hat{x}(k|k-1) + Bu(k) + Ke(k) \tag{4.6}$$

$$e(k) = y(k) - C\hat{x}(k|k-1), \tag{4.7}$$

with $K \in \mathbb{R}^{n_x \times n_y}$ the Kalman gain and $e(k) \in \mathbb{R}^{n_y \times 1}$ the so-called innovation process. The filter estimates $\hat{x}(k|k)$ can be computed via:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + A^{-1}Ke(k). \tag{4.8}$$

The optimal Kalman gain is generally computed as a function of (A,B,C) and (Q,R,S). As already mentioned, we consider the case where the matrices A, B, C are available, but matrices Q, R, S are not available.

If the covariance matrices Q, R, S are not known, an optimal Kalman filter of the form (4.6)-(4.7) can no longer be computed. Instead, a sub-optimal filter could be constructed that has the same form as (4.6)-(4.7). The Kalman gain K in the suboptimal filter is replaced by a manually tuned gain matrix L. The filter gain can be determined using pole placement techniques, such as described in [59]. The resulting filters are generally sub-optimal in the sense that there is no guarantee that the criterion (4.5) is minimized.

If Q, R, S are not available, but data $u(1), y(1), \ldots, u(N), y(N)$ are available, many methods can be found in the literature to design a filter that is equivalent to the Kalman filter as the number of data N tends to infinity. Indeed, a filter designed using one of these data-based methods will minimize the MSE error of the estimated states (see

(4.5)). In practice the available number of measurements N is finite. As a result the filters designed using these data-based methods result in sub-optimal results.

The data based methods to compute a Kalman filter can be divided into two groups of methodologies:

- Parametric methods. This group of methods parameterizes either $Q(\theta), R(\theta), S(\theta)$ or $K(\theta)$. The parameter vector θ is estimated using a series of data $u(1), \ldots, u(N), y(1), \ldots, y(N)$. Issues regarding the parametrization of either $Q(\theta), R(\theta), S(\theta)$ or $K(\theta)$ are discussed below. Examples of different estimators for θ are also given below. Using the estimated parameter vector $\hat{\theta}$, the Kalman filter is constructed using the usual relations.

- Non-parametric methods. This group of methods directly derives a Kalman filter using test data $u(1), \ldots, u(N), y(1), \ldots, y(N)$, without the need to parameterize Q, R, S or K.

The group of parametric methods can again be divided into to groups. The first group parameterizes $Q(\theta), R(\theta)$ and $S(\theta)$, the second group parameterizes $K(\theta)$. If $Q(\theta), R(\theta), S(\theta)$ are parameterized, the dimension of parameter vector θ can often be very high. Full parametrization of Q, R, S requires $\frac{1}{2}n_x(n_x+1) + \frac{1}{2}n_y(n_y+1) + n_x n_y$ parameters (discounting the symmetric ones). The number of parameters can be reduced by for instance only parameterizing the diagonals of matrices, but this may limit the accuracy of the resulting Kalman filter.

If $K(\theta)$ is parameterized the number of parameters required for a full parametrization is $n_x n_y$. Since the number of parameters to be estimated is smaller then when $Q(\theta), R(\theta), S(\theta)$ are parameterized, parameterizing $K(\theta)$ is generally preferable.

Once a suitable parametrization for either $K(\theta)$ or $Q(\theta), R(\theta), S(\theta)$ has been chosen, the vector θ can be estimated. Estimators for θ encountered in the literature can be classified into four categories: Bayesian estimators, Maximum Likelihood (ML) estimators, estimators based on Prediction Error Identification and estimators that use correlation or covariance matching.

Using the collected data $u(1) \ldots u(N), y(1) \ldots y(N)$, the Bayesian estimator for θ is given by :

$$\hat{\theta} \quad = \quad \mathbb{E}(\theta | u(1) \ldots u(N), y(1) \ldots u(N)), \tag{4.9}$$

where the expected value is taken over θ, since in Bayesian estimation θ is modelled as a stochastic variable. Computing the expected value in the equations above involves numerical integration over a possibly large dimensional parameter space. As such, Bayesian estimation is a computationally intensive estimation method. More details about Bayesian estimation of covariance information can be found in [62].

The second class of estimators are the ML estimators. The ML estimator of θ is given by:

$$\hat{\theta} = \arg\max_{\theta} Pr(y_1 \ldots y_n | \theta, u_1 \ldots u_N), \tag{4.10}$$

with $Pr(y|\theta,u)$ the conditional probability that y occurs given θ and u. Like the Bayesian method of estimation, the ML estimator is also computationally intensive, as it usually involves numerical nonlinear minimization. The use of the ML estimator to construct a Kalman filter is described in more detail in [102].

The third group of estimators for θ use the Prediction Error Identification method as described in [56]. The estimator $\hat{\theta}$ for this method is given by:

$$\hat{\theta} = \arg\min_{\theta} \sum_{k=1}^{N} \|\hat{y}(k,\theta) - y(k)\|^2, \tag{4.11}$$

with $\hat{y}(k,\theta)$ the best one step ahead prediction of $y(k)$ given all information $y(1)\dots y(k-1), u(1)\dots u(k-1)$ as a function of θ. Like the ML estimation procedure, Prediction Error Identification often requires nonlinear minimization. Thus the parameter estimation procedure is computationally difficult. It can be shown that the Prediction Error estimate is closely related to the ML estimator, see for instance [77].

The fourth and final class of estimators consists of covariance matching methods. In these methods, θ is solved by considering the autocovariance function of the innovations process $e(k)$ (see (4.7)). For an optimal filter it can be shown that this sequence should be a white noise [3]. This property is used in [66] and [18] to iteratively construct estimates of the optimal choices for either Q,R,S or K.

All the parametric approaches described require that a nonlinear optimization problem has to be solved in order to construct the Kalman filter. Even when the dimension of the parameter vector θ is small, the criterion function in the optimization problem can exhibit local minima, meaning that the problem can only be solved if accurate starting values for θ are available.

In summary, there are two difficulties that prohibit the use of parametric methods to construct a Kalman filter for practical data. The first reason is that finding a suitable parametrization for either $Q(\theta),R(\theta),S(\theta)$ of $K(\theta)$ is nontrivial. Secondly the resulting optimization problem to estimate θ is generally nonlinear.

The second group of methodologies to construct a Kalman filter from practical data are the nonparametric methods. An example of a nonparametric method is the covariance based approach described in [67]. In the described method, the covariance of the stochastic part of the signal $y(k)$ is used to directly identify the Kalman gain matrix K, without the need for K to be parameterized. Moreover, for the construction of the Kalman gain K, only linear operations are necessary.

Given these attractive properties of the nonparametric approach presented in [67], we will only consider this approach to construct a Kalman filter from practical data.

For both parametric and nonparametric approaches, (including the covariance based method we are considering), it is assumed that the system described by (4.1)-(4.2) is completely observable and also that the corresponding observability matrix has no small singular values. This can be a problem, because for many physical systems, this is not necessarily the case.

This chapter contains three main contributions:

- We will show that an ill conditioned observability matrix for the system (4.1)-

(4.2) can lead to poor results of the Kalman Filter derived using the covariance-based approach (even if the number of available test data is reasonably large).

- We will show that this problem can be solved by using the normal covariance based approach for only a part of the state directions and derive a separate estimator for the remaining state directions.

- We will show that even if a system is completely unobservable in certain directions of the state-space, it is still possible to construct a useful Kalman filter.

This chapter is organized as follows: section 4.2 starts with a summary of the covariance based Kalman filter design method. Then, in the second part of section 4.2, we demonstrate how this covariance based Kalman filter design method can be used to construct an optimal filter using known matrices A,B,C and a series of test measurements. Section 4.3 then analyze the sensitivity of this method to estimation errors and also shows how this sensitivity can be reduced. Section 4.4 presents a simulation example, in which the results of the previous sections are applied. Section 4.5 provides a discussion of the properties of the results. The chapter ends with a summary of the main results and some concluding remarks in section 4.6.

4.2 Covariance based design of a Kalman filter

4.2.1 Introduction

As mentioned in the introduction our methodology for designing a state filter is based on a covariance based design procedure of the Kalman filter. In the first part of this section a short overview of the covariance based design procedure for the Kalman filter is given.

The second part of this section describes how this method can be used to obtain a state filter using only knowledge of the matrices A,B,C, and some test data consisting of a series of inputs $u(k)$ and corresponding outputs $y(k)$.

4.2.2 Summary of the covariance based Kalman filter design

As opposed to the standard method for the design of a Kalman filter which requires knowledge of the matrices (Q,R,S), the covariance based method does not require this information. Instead, the covariance of the measurement signal y_k is used to construct the Kalman filter. The resulting filter is equivalent to the filter designed using the common design method.

The model (4.1)-(4.2) can be split into two parts, a purely deterministic part, and a purely stochastic part. The deterministic part is given by:

$$x_d(k+1) = Ax_d(k)+Bu(k) \tag{4.12}$$
$$y_d(k) = Cx_d(k). \tag{4.13}$$

The stochastic part of the system is given by:

$$\begin{aligned} x_s(k+1) &= Ax_s(k)+w_k && (4.14)\\ y_s(k) &= Cx_s(k)+v_k. && (4.15)\end{aligned}$$

The deterministic and stochastic systems are related to the complete system by:

$$\begin{aligned} x(k) &= x_d(k)+x_s(k) && (4.16)\\ y(k) &= y_d(k)+y_s(k). && (4.17)\end{aligned}$$

If we assume that the matrix A is stable and the process is in steady-state, the covariance function of the stochastic part of the measurements $y_s(k)$ can be written as:

$$R_{y_s}(i,k) = \mathbb{E}[y_s(k+i)y_s(k)^T] = \begin{cases} CA^{i-1}M(k) & i>0 \\ C^T\Phi(k)C+R_e(k) & i=0 \\ M(k)^T(A^T)^{(-i-1)}C & i<0 \end{cases} \quad (4.18)$$

In the equation above, $M(k)$ is the cross-covariance between the stochastic part of the state $x_s(k+1)$ and stochastic part of the output $y_s(k)$:

$$M(k) = \mathbb{E}\left[x_s(k+1)y_s(k)^T\right], \quad (4.19)$$

$R_e(k)$ is the covariance of the innovation sequence $e(k)$ (see (4.7)):

$$R_e(k) = \mathbb{E}[e(k)e(k)^T] \quad (4.20)$$

and $\Phi(k)$ is the covariance of the stochastic part of the state:

$$\Phi(k) = \mathbb{E}[x_s(k)x_s(k)^T]. \quad (4.21)$$

Using the definitions in this section, the Kalman filter (for prediction) is given by (4.6)-(4.7), with

$$\begin{aligned} K(k) &= [M(k)-A\Sigma(k)C^T]R_e(k)^{-1} \text{ with} && (4.22)\\ R_e(k) &= R_{y_s}(0,k)-C\Sigma(k)C^T, && (4.23)\\ \Sigma(k+1) &= A\Sigma(k)A^T+K(k)R_e(k)K(k)^T, \\ \Sigma(0) &= 0. \end{aligned}$$

In these last equations, $\Sigma(k)$ can be interpreted as the covariance of the estimated state $\hat{x}_s(k)$:

$$\Sigma(k) = \mathbb{E}[\hat{x}_s(k)\hat{x}_s(k)^T]. \quad (4.24)$$

As can be seen from (4.22) B plays no role when computing the Kalman gain K. To simplify the notation in the remainder of this report without loss of generality it is assumed that $B=0$.

Since we assumed that the stochastic process is in steady state, the matrices $M(k)$, $\Sigma(k)$, $\Phi(k)$ and $R_e(k)$ will quickly converge to constant values.

Summarizing, if the matrix $M(k)$ (see (4.19)) and the covariance function $R_{y_s}(k)$ are known, it is possible to compute the Kalman gain K (see (4.6)) even when the covariance matrices Q,R,S are not known. In practice, situations in which Q,R,S are unknown, but where M and $R_{y_s}(k)$ are known will never occur. Fortunately, if data $u(0),y(0),u(1),y(1),\ldots$ generated with the original system is available, M and R_{y_s} can be estimated. Using the estimates for M and R_{y_s} we can then use (4.22) to construct a state filter.

4.2.3 Estimation of M, $R_{y_s}(0)$

As already mentioned in the previous section, we will need to estimate the unknown quantities M and $R_{y_s}(k)$, in order to be able to construct a state filter using (4.22). In this section it is demonstrated how these can be estimated, using N points of test-data generated by (4.1)-(4.2). Note that (once the process is in steady state) these quantities M and $R_{y_s}(k)$ are constants and both are independent on the input signal $u(k)$ that was used to generate $y(k)$.

We start by showing how M can be estimated. An estimator for M is easily derived by realizing that the following relation holds:

$$\begin{bmatrix} R_{y_s}(1) \\ \vdots \\ R_{y_s}(N-1) \end{bmatrix} = \mathcal{O}M \quad \text{with} \tag{4.25}$$

$$\mathcal{O} = \begin{bmatrix} C^T & (CA)^T & \ldots & (CA^{L-1})^T \end{bmatrix}^T, \tag{4.26}$$

with L a constant integer for which holds that $L \geq n_x/n_y$. Denote the left inverse of $\mathcal{O} \in \mathbb{R}^{Ln_y \times n_x}$ as $\mathcal{O}^\dagger$, i.e.

$$\mathcal{O}^\dagger = (\mathcal{O}^T\mathcal{O})^{-1}\mathcal{O}^T. \tag{4.27}$$

Using this left inverse, M can be obtained via:

$$M = \mathcal{O}^\dagger \begin{bmatrix} R_{y_s}(1) \\ \vdots \\ R_{y_s}(L) \end{bmatrix}. \tag{4.28}$$

The covariance function $R_{y_s}(k)$ is unknown, but as will be shown in the sequel, it can be estimated using the available test data.

In order to estimate the covariance function $R_{y_s}(k)$ we require $y_s(k)$, the stochastic part of the measurements $y(k)$. This signal can be recovered from measurements $y(k)$ using:

$$\begin{aligned} y_s(k) &= y(k) - y_d(k) \\ &= y(k) - CA^k x_d(0) - \sum_{i=0}^{k-1} CA^{k-1-i}Bu(i). \end{aligned} \tag{4.29}$$

In the equation above the initial state of the deterministic part, $x_d(0)$ is unknown. However, if A is stable, $\lim_{k\to\infty} A^k x_d(0) = 0$, so for large k we can write:

$$y_s(k) \approx \tilde{y}_s(k) \triangleq y(k) - \sum_{i=0}^{k-1} CA^{k-1-i}Bu(i) \quad \forall\, k >> 1. \tag{4.30}$$

Using the signal $\tilde{y}_s(k)$, the covariance function $R_{y_s}(i)$ can be estimated using different methods:

- The simplest method to estimate the covariance function $R_{y_s}(k)$ is using a lagged product estimator:

$$\hat{R}_{y_s}(i) = \frac{1}{N}\sum_{k=0}^{N-i} y_s(k+i)y_s(k)^T. \tag{4.31}$$

- The covariance function $R_{y_s}(k)$ can also be estimated using a model based approach. In the model based approach the stochastic signal $y_s(k)$ is assumed to be generated using:

$$\begin{aligned} \zeta(k+1) &= A_s\zeta(k) + K_s\varepsilon(k) && (4.32)\\ y_s(k) &= C_s\zeta(k) + \varepsilon(k), && (4.33) \end{aligned}$$

  with $\varepsilon(k)$ a zero mean white noise process with covariance R_s. The matrices A_s, K_s, C_s, R_s can be estimated using subspace algorithms such as CCA [52], N4SID [93] or MOESP [96]. Denoting the estimated matrices as $\hat{A}_s, \hat{K}_s, \hat{C}_s$, and $\hat{R}_s$, the estimate for $R_{y_s}(k)$ is:

$$\hat{R}_{y_s}(k) = \begin{cases} \hat{C}_s P_\zeta \hat{C}_s^T + \hat{R}_s & \text{for} \quad k = 0 \\ \hat{C}_s \hat{A}_s^k P_\zeta \hat{C}_s^T & \text{for} \quad k > 0, \end{cases} \tag{4.34}$$

  with P_ζ the solution of the Lyapunov equation:

$$P_\zeta = \hat{A}_s P_\zeta \hat{A}_s^T + \hat{K}_s \hat{R}_s \hat{K}_s^T. \tag{4.35}$$

- If the signal $y_s(k)$ is scalar, we can using ARMA time-series modelling to estimate $R_{y_s}(k)$. In this approach the signal $y_s(k)$ is modelled as:

$$\begin{aligned} y_s(k) + a_1 y_s(k-1) + \ldots + a_p y_s(k-p) = \\ \varepsilon(k) + b_1\varepsilon(k-1) + \ldots + b_q\varepsilon(k-q), \end{aligned} \tag{4.36}$$

  with $\varepsilon(k)$ a zero mean white noise process. Using the data $y_s(k)$ the parameters $a_1, \ldots, a_p, b_1, \ldots, b_q$ can be estimated, using time series software (see for instance [17]). The estimated parameters can then be used to construct the estimate for the covariance function $y_s(k)$ using the Levinson-Durbin relations [87].

Using the estimated covariance function $\hat{R}_{y_s}(i)$, a natural estimator for M is:

$$\hat{M} = \mathcal{O}^{\dagger} \begin{bmatrix} \hat{R}_{y_s}(1) \\ \vdots \\ \hat{R}_{y_s}(L) \end{bmatrix}. \tag{4.37}$$

Using the estimate for M and the estimate $\hat{R}_{y_s}(0)$ of $R_{y_s}(0)$, an approximate Kalman filter can be constructed using (4.22), after substituting $\hat{M}$ for M and $\hat{R}_{y_s}(0)$ for $R_{y_s}(0)$.

4.3 Improved estimation

4.3.1 Introduction

If the technique described in the previous section is directly used to obtain an approximate Kalman filter, the accuracy of the estimates of the obtained Kalman filter can vary from being very accurate to being very poor. It can be shown that results are poor when the matrix $\mathcal{O}$ satisfies the following assumption:

Assumption 4.1 *Denote the decreasing singular values of $\mathcal{O}$ as $\sigma_1 \geq \ldots \geq \sigma_l \geq \ldots \geq \sigma_{n_x}$. It will be assumed that the matrix C and corresponding measurements $y(k)$ are scaled such that*

$$\sigma_1 = 1 \tag{4.38}$$

and

$$\sigma_i << 1 \quad \forall \quad i > l. \tag{4.39}$$

We will start by explaining why the estimated states can be very poor under this assumption. The poor quality of estimated states can be easily explained if we consider the estimation error for $\hat{M}$. Denote the SVD of $\mathcal{O}$ as:

$$\mathcal{O} = U \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{n_x} \\ & 0 & \end{bmatrix} V^T \tag{4.40}$$

in which $U \in \mathbb{R}^{(N-1)n_y \times (N-1)n_y}$ and $V \in \mathbb{R}^{n_x \times n_x}$ are unitary matrices. Using this notation, the estimation error for $\hat{M}$ can be written as:

$$\hat{M} - M = [V_{1:l}\ V_{l+1:n_x}] \begin{bmatrix} \sigma_1^{-1} & & \\ & \ddots & 0 \\ & & \sigma_{n_x}^{-1} \end{bmatrix} \begin{bmatrix} U_{1:l}^T \\ U_{l+1:n_x}^T \\ U_{n_x+1:(N-1)n_y}^T \end{bmatrix} \Delta_R, \tag{4.41}$$

with

$$\Delta_R = \begin{bmatrix} \hat{R}_{y_s}(1) - R_{y_s}(1) \\ \vdots \\ \hat{R}_{y_s}(N-1) - R_{y_s}(N-1) \end{bmatrix} \tag{4.42}$$

and $V_{i:j}$ and $U_{i:j}$ matrices consisting of the columns i through j of V and U, respectively. From (4.41) we see that an error Δ_R in the direction of U_i corresponds to an error $\hat{M} - M$ in the direction V_i after multiplication with σ_i^{-1}. Due to Assumption 4.1 small errors ΔR in the directions of $U_{l+1:n_x}$ induce large errors $\hat{M} - M$ in the directions of $V_{l+1:n_x}$.

For some estimators of the covariance function $R_{y_s}(k)$ an expression for both the mean and covariance of Δ_R is available. If the bias and variance of Δ_R are given as:

$$\mathbb{E}\{\Delta_R\} = \overline{\Delta}_R \quad \text{and} \quad \mathbb{E}\{(\Delta_R - \overline{\Delta}_R)(\Delta_R - \overline{\Delta}_R)^T\} = \mathcal{R}_{\Delta_R}, \tag{4.43}$$

then it holds that

$$\mathbb{E}(\hat{M} - M)(\hat{M} - M)^T = Y\overline{\Delta}_R\overline{\Delta}_R^T Y^T + Y\mathcal{R}_{\Delta_R}Y^T, \tag{4.44}$$

in which

$$Y = [V_{1:l}\ V_{l+1:n_x}] \begin{bmatrix} \sigma_1^{-1} & & \\ & \ddots & 0 \\ & & \sigma_{n_x}^{-1} \end{bmatrix} \begin{bmatrix} U_{1:l}^T \\ U_{l+1:n_x}^T \\ U_{n_x+1:(N-1)n_y}^T \end{bmatrix}. \tag{4.45}$$

If a filter is designed using $\hat{M}$ instead of M, the estimation error $\hat{M} - M$ will result in an increase in the MSE of the estimated states $\hat{x}(k|k)$. Unfortunately an explicit equation describing the increase of the MSE for $\hat{x}(k|k)$ as a result of the estimation error $\hat{M} - M$ is not known, but experience shows that when $\hat{M} - M$ is large, the MSE for a filter designed using $\hat{M}$ instead of M is much higher than the MSE of the Kalman filter estimates. Under assumption 4.1 small errors ΔR in the directions of $U_{l+1:n_x}$ induce large errors $\hat{M} - M$ in the directions of $V_{l+1:n_x}$, thus large estimations errors $\hat{M} - M$ are likely to occur and the state estimates using an unmodified covariance methods are often poor.

In the remainder of this section we will show how a good estimate of the state can be obtained under Assumption 4.1. When deriving this estimator, it is convenient to first apply the following similarity transform:

$$\tilde{x}(k) = V^T x(k) = \begin{bmatrix} V_{1:l}^T \\ V_{l+1:n_x}^T \end{bmatrix} x(k) = \begin{bmatrix} \tilde{x}_1(k) \\ \tilde{x}_2(k) \end{bmatrix}. \tag{4.46}$$

The state $\tilde{x}_1(k)$ is thus the component of the original state in the directions of $V_{1:l}$, and $\tilde{x}_2(k)$ corresponds to the component of $x(k)$ in the directions $V_{l+1:n}$. After having derived an estimate $\hat{\tilde{x}}(k)$ the corresponding estimate $\hat{x}(k)$ can be simply computed via

$\hat{x}(k) = V\hat{\tilde{x}}(k)$. Due to the similarity transform, the various state related matrices also need to be transformed:

$$\tilde{A} = V^T A V = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix} \tag{4.47}$$

$$\tilde{C} = CV = \begin{bmatrix} \tilde{C}_1 & \tilde{C}_2 \end{bmatrix} \tag{4.48}$$

$$\tilde{M} = V^T M = \begin{bmatrix} \tilde{M}_1 \\ \tilde{M}_2 \end{bmatrix} \tag{4.49}$$

$$\tilde{\Sigma}(k) = V^T \Sigma(k) V = \begin{bmatrix} \tilde{\Sigma}_{11}(k) & \tilde{\Sigma}_{12}(k) \\ \tilde{\Sigma}_{21}(k) & \tilde{\Sigma}_{22}(k) \end{bmatrix} \tag{4.50}$$

$$\tilde{K}(k) = V^T K(k). \tag{4.51}$$

Note that after the similarity transform, $\tilde{M}_1$ corresponds to the directions in which M can be accurately estimated using (4.37) and $\tilde{M}_2$ corresponds to the directions in which M is difficult to estimate.

The estimator for $\tilde{x}(k)$ will be derived in two steps, first we will construct an estimator for $\tilde{x}_1(k)$ then an estimator for $\tilde{x}_2(k)$ will be discussed.

4.3.2 Estimator for $\tilde{x}_1(k)$

Our estimator for the state component $\tilde{x}_1(k)$ is based on the following proposition:

Proposition 4.2 *Define $\tilde{M}_l$ as:*

$$\tilde{M}_l = \begin{bmatrix} \tilde{M}_1 \\ 0 \end{bmatrix}. \tag{4.52}$$

Suppose that $\sigma_i = 0$ for $i > l$ and assume we construct a Kalman filter using (4.22) using $\tilde{M}_l$ instead of $\tilde{M}$. Denote the vectors and matrices $\tilde{\Sigma}(k)$, $\tilde{R}_e(k)$, $\hat{\tilde{x}}(k)$ that are computed using $\tilde{M}_l$ as $\tilde{\Sigma}_l(k)$, $\tilde{R}_{e,l}(k)$, $\hat{\tilde{x}}_l(k)$. Then it holds that:

- $R_{e,l}(k) = R_e(k)$
- $\tilde{\Sigma}_{11,l}(k) = \tilde{\Sigma}_{11}(k)$
- $\hat{\tilde{x}}_{1,l}(k) = \hat{\tilde{x}}_1(k)$.

Proof. The proof can be found in section 4.A ∎

This proposition implies that the estimation result $\hat{\tilde{x}}_1(k)$ does not change if, instead of using the exact Kalman filter, we use a Kalman filter obtained by replacing $\tilde{M}$ by $\tilde{M}_l$ in (4.22).

Combining the result with the fact that we can accurately estimate $\hat{\tilde{M}}_1$ with $\hat{\tilde{M}}_1 = V_{1:l}^T \hat{M}$, a natural estimator for $\tilde{x}_1(k)$ is thus the Kalman Filter computed with (4.22) where $\tilde{M}$ is replaced by $[\hat{\tilde{M}}_1\ 0]^T$.

Note that even though Proposition 4.2 holds for $\sigma_i = 0\ \forall\ i > l$, we assume that for a Kalman filter constructed using $[\hat{\tilde{M}}_1\ 0]^T$ under Assumption 4.1 it holds that:

- $R_{e,l}(k) \approx R_e(k)$
- $\tilde{\Sigma}_{11,l}(k) \approx \tilde{\Sigma}_{11}(k)$
- $\hat{\tilde{x}}_{1,l}(k) \approx \hat{\tilde{x}}_1(k)$.

4.3.3 Estimator for $\tilde{x}_2(k)$

Using Proposition 4.2 we can construct an estimator for $\tilde{x}_1(k)$. In this section we will construct an estimator for the remaining states $\tilde{x}_2(k)$. The optimal estimate for $\tilde{x}_2(k)$ corresponding to the optimal Kalman filter is:

$$\begin{aligned}\hat{\tilde{x}}_2(k+1|k) = \tilde{A}_{21}\left(\hat{\tilde{x}}_1(k|k-1) + \mathbb{E}[(\tilde{x}_1(k) - \hat{\tilde{x}}_1(k|k-1))|e(k)]\right) \\ + \tilde{A}_{22}\left(\hat{\tilde{x}}_2(k) + \mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k|k-1))|e(k)]\right).\end{aligned} \tag{4.53}$$

Both conditional expectations can be expressed as functions of $\tilde{M}$, $\tilde{\Sigma}(k)$, $\tilde{C}$, $\tilde{R}_e(k)$ and $e(k)$ using the following proposition:

Proposition 4.3 *Under the assumption that $\sigma_i = 0$ for $i > l$, the expected value terms $\mathbb{E}[(\tilde{x}_1(k) - \hat{\tilde{x}}_1(k|k-1))|e(k)]$ and $\mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k|k-1))|e(k)]$ in (4.53) can be written as:*

$$\tilde{A}_{21}\mathbb{E}[(\tilde{x}_1(k) - \hat{\tilde{x}}_1(k|k-1))|e(k)] = \tilde{A}_{21}(\tilde{A}_{11}^{-1}\tilde{M}_1 - \tilde{\Sigma}_{11}(k))\tilde{C}_1^T R_e^{-1}(k)e(k) \tag{4.54}$$

and

$$\tilde{A}_{22}\mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k|k-1))|e(k)] = (\tilde{M}_2 - \tilde{A}_{21}\tilde{A}_{11}^{-1}\tilde{M}_1 - \tilde{A}_{22}\tilde{\Sigma}_{21}(k)\tilde{C}_1^T)R_e^{-1}(k)e(k). \tag{4.55}$$

Proof. The proof of the proposition is contained in section 4.B ∎

In this proposition we see that the conditional expectation for the error in $\hat{\tilde{x}}_1(k)$ can be computed, since the expression contains only elements that are either known, or can be accurately estimated.

Computing the expectation $\mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k))|e(k)]$ requires an accurate estimate of $\hat{\tilde{M}}_2$, which is unavailable. This means we cannot reliably compute the conditional expectation for the error in $\tilde{x}_2(k)$. Therefore we choose not to use this conditional expectation.

Even though we cannot reliably compute $\mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k))|e(k)]$, we can show that

$$\mathbb{E}_{e(k)}\left(\mathbb{E}[(\tilde{x}_2(k) - \hat{\tilde{x}}_2(k))|e(k)]\right) = 0. \tag{4.56}$$

This means that the a-priori (i.e. before $e(k)$ was known) expectation of the last term in (4.53) is zero. This result is easily obtained using (4.55). As a result an a-priori

unbiased prediction $\hat{\tilde{x}}_2(k+1|k)$ can be obtained using:

$$\hat{\tilde{x}}_2(k+1|k) = \tilde{A}_{21}\left(\hat{\tilde{x}}_1(k|k-1) + \mathbb{E}[(\tilde{x}_1(k) - \hat{\tilde{x}}_1(k|k-1))|e(k)]\right) + \tilde{A}_{22}\hat{\tilde{x}}_2(k|k-1). \tag{4.57}$$

Note that even though the new estimator for $\tilde{x}_2(k+1)$ is unbiased, it is not optimal, because we cannot reliably use the correlation that exists between $e(k)$ and $\tilde{x}_2(k) - \hat{\tilde{x}}_2(k|k-1)$. Still because (4.57) can be reliably computed, whereas (4.53) cannot, the estimate $\hat{\tilde{x}}_2(k+1|k)$ obtained using (4.57) is often better (in terms of its MSE) than the estimate obtained using (4.53).

Combining (4.54) and (4.57), the complete estimator for $\tilde{x}_2(k+1)$ is obtained by replacing $\tilde{M}_1$ with the estimate $\hat{\tilde{M}}_1$:

$$\begin{aligned}\hat{\tilde{x}}_2(k+1|k) = \tilde{A}_{21}\hat{\tilde{x}}_1(k|k-1) + \tilde{A}_{22}\hat{\tilde{x}}_2(k|k-1) \\ + \tilde{A}_{21}(\tilde{A}_{11}^{-1}\hat{\tilde{M}}_1 - \tilde{\Sigma}_{11,l}(k))\tilde{C}_1^T R_{e,l}^{-1}(k)e(k).\end{aligned} \tag{4.58}$$

4.3.4 Summary

A data based Kalman filter can be estimated using the procedure described in section 4.2.3. In practice, the estimation results of Kalman filter constructed in this manner are often unsatisfactory. In section 4.3.1 it was argued that the poor performance of a data based Kalman filter may be caused by large estimation errors for M, especially when the system is poorly observable (see assumption 4.1).

In sections 4.3.2 and 4.3.3 an alternative method was derived to construct a data based Kalman filter. The methodology was designed to result in accurate data based Kalman filters, even if the system model is poorly observable.

The new filter is constructed by first applying a similarity transform $\tilde{x}(k) = V^T x(k)$. Then the elements corresponding to the first l columns of V $(= \tilde{x}_1(k))$ are estimated using an approximate Kalman filter estimate that is constructed by inserting $[\tilde{M}_1^T \ 0]^T$ for M in (4.22). Using the results of this filter, $\tilde{x}_2(k)$ can be estimated using (4.58).

4.4 Simulation example

In this section the method for constructing a Kalman filter as outlined in the previous sections will be demonstrated using a simulation example. For the example, we will again use the heated plate model introduced in section 3.4.1. To modify the heated plate model such that the model structure complies with the structure of (4.1)-(4.2) the following steps are taken:

1. Instead of using the temperature dependent heat transfer specified in (3.48), we will use a constant heat conductivity coefficient:

$$\lambda = 80. \tag{4.59}$$

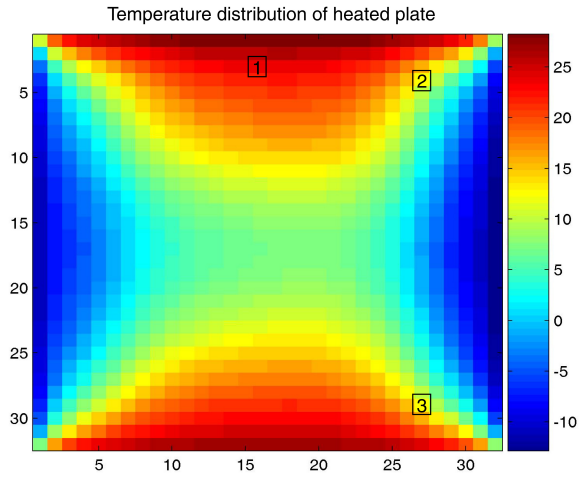

Figure 4.1 Example of the temperature distribution of the heated plate as computed using heated plate model. The numbers in the figure correspond to the locations where the temperature of the plate is measured.

It can be shown that using a constant heat conductivity, the model for the heated plate becomes a linear state equation:

$$x(k+1) = Ax(k) + Bu(k). \tag{4.60}$$

2. A zero mean Gaussian process noise term $w(k)$ is added to the model as a process noise. The covariance matrix Q of the process noise $w(k)$ is chosen as a diagonal matrix. Each element on the diagonal of Q is chosen randomly between 0 and 5.

3. A measurement equation of the form (4.2) is added. The measurements $y(k)$ are chosen to correspond to the measurement locations indicated in Figure (4.1). The measurement error $v(k)$ is chosen to be a zero mean Gaussian white noise, with covariance matrix $R = \text{diag}(0.1\ 0.1\ 0.1)$. The cross covariance S between $w(k)$ and $v(k)$ is chosen S=0.

4. To simplify the computations involved in constructing a data based Kalman filter, the model order was reduced to from 1024 to 12, using the POD technique described in section 2.4.2.

After these modifications the heated plate model is a linear model of the form (4.1)-(4.2), with all matrices A, B, C, Q, R, S known. The singular values of the observability matrix $\mathcal{O}$ for this system are presented in Table 4.1. As can be seen the singular values are quickly decreasing. Thus the system can be said to be poorly observable.

Table 4.1 Singular values of the observability matrix of the linear heated plate model

i	σ_i	i	σ_i
1	0.35	7	0.0032
2	0.26	8	0.0016
3	0.10	9	$4.6 \cdot 10^{-4}$
4	0.045	10	$4.6 \cdot 10^{-5}$
5	0.028	11	$2.1 \cdot 10^{-5}$
6	0.0056	12	$2.6 \cdot 10^{-7}$

The goal of the simulation example is to test, assuming that the matrices A, B, C, are known, and the set of test measurements y is available, if it is possible to construct a Kalman filter using the procedure outlined in the previous sections. The accuracy of the filter designed using this procedure will be compared with the accuracy of the optimal filter, that can be constructed since in the simulations the true values of Q, R and S are also available.

The first step in constructing a data based Kalman filter is to estimate the auto-covariance function of the stochastic part of the measurements. For this purpose, 1500 measurements using the simulation model were generated. After subtracting the deterministic part of the system using (4.30), the computed sequence y_s was used to estimate the autocovariance function $\hat{R}_{y_s}(k)$ of the stochastic part of measurements. The covariance function was determined by estimating a state-space model of the measurements using the N4SID subspace algorithm [93], using the subspace based technique described earlier in section 4.2.3.

After estimating the auto-covariance function $\hat{R}_{y_s}(k)$, the estimated matrix $\hat{M}$ is constructed using (4.37). Finally the data based Kalman filter is designed by substituting the estimated values $\hat{R}_{y_s}(0)$ and $\hat{M}$ for $R_{y_s}(0)$ and M in (4.22) as was described in section 4.2.

In order to evaluate the performance of the data based Kalman filter design method, the accuracy of the resulting filters is compared with the optimal Kalman filter, the filter can be derived if Q, R, S are known. The comparison is conducted by using the true system to generate an additional series of 1000 data points. The error measure that is used to compare the performance is the averaged squared error of the predicted state, which will be denoted by $\mathcal{E}$:

$$\mathcal{E} = \frac{1}{N} \sum_{k=0}^{N-1} \|x(k) - \hat{x}(k|k-1)\|^2 . \tag{4.61}$$

The averaged errors $\mathcal{E}$ for the optimal and approximate Kalman filter can be found in Table 4.2.

The resulting averaged squared prediction error using the unmodified covariance technique as described in section 4.2.3 is presented in Table 4.2. The filter designed using the unmodified covariance technique is unstable, and as a result the average

Table 4.2 Mean squared one-step-ahead prediction errors $\mathcal{E}$ of both the optimal Kalman filter and approximations obtained using methods outlined in sections 2 and 3.

Filter	$\mathcal{E}$
Optimal filter	368
Direct approx. filter (sect. 2)	$2.5 \cdot 10^{26}$
Improved filter $l = 1$ (sect. 3)	500
Improved filter $l = 2$ (sect. 3)	447
Improved filter $l = 3$ (sect. 3)	399
Improved filter $l = 4$ (sect. 3)	402
Improved filter $l = 5$ (sect. 3)	425
Improved filter $l = 6$ (sect. 3)	611
Improved filter $l = 7$ (sect. 3)	$2.7 \cdot 10^4$

squared prediction error of the filter is very large. This may be explained by the theory of section 4.3 if $\mathcal{O}$ has small singular values. The poor results of the approximate Kalman filter are likely caused by the smallest values, see (4.41).

Apart from the direct method of section 4.2.3, the improved filter of section 4.3 are also tested. In order to apply the results of section 4.3, we first need to apply the similarity transform $\tilde{x}(k) = V^T x(k)$ and choose the number of elements in $\tilde{x}_1(k)$ and $\tilde{x}_2(k)$. The number of elements in $\tilde{x}_1(k)$ is denoted by l. Normally a single partition would be chosen using the computed singular values but in this example we have tried all options for $l = 1, \ldots, 7$. The results using the improved estimator are also presented in Table 4.2.

Results in Table 4.2 show that using $l = 1$ the resulting Kalman filter produces much better estimates than the unmodified covariance method. Results improve as l is increased up to 3. For these values of l the matrix $\hat{M}$ can be estimated accurately enough, such that including these modes in the approximate Kalman filter improves the accuracy of the resulting filter. For $l = 3$ the quality of the data based Kalman filter is at its best. For values of $l > 3$, the quality of the resulting approximate filter deteriorates. For $l > 3$ we can conclude that it is no longer beneficial to include extra modes in $\tilde{M}_l$, because $\hat{\tilde{M}}_l$ is no longer accurate enough.

4.5 Discussion

In this chapter we have considered covariance method for designing a Kalman filter.

As was shown in the simulation example in section 4.4, using the unmodified covariance method can result in a filter that produces poor state estimates. In section 4.3.1 we have shown that the poor accuracy of filter derived using the unmodified method is likely to occur if the observability matrix of the system is poorly conditioned. The poor condition number of the observability matrix results in inaccurate estimate $\hat{M}$. As a result estimates of the data based filter constructed using the esti-

mated $\hat{M}$ are also poor.

It should be noted however, that when using the covariance method, the error in the estimate for M is not the only reason that a resulting state filter may be poor in practice. As can be seen, from equation (4.22), the estimation error in $\hat{R}_{y_s}(0)$ can also cause that the resulting state filter is not optimal.

From equation (4.23) we can conclude that if the $\hat{R}_{y_s}(0)$ is estimated too low, the covariance matrix $R_e(k)$ in (4.23) can become negative, even though this auto covariance matrix should always be semi-positive definite. In practice we see that data based Kalman filters are generally unstable if $R_e(k) > 0 \forall k$ does not hold. Underestimating $R_{y_s}(0)$ can thus have severe consequences.

On the other hand, if the estimate $\hat{R}_{y_s}(0) > R_{y_s}(0)$, than we have that $\hat{R}_e(k) > R_e(k)$. If a $\hat{R}_e(k) > R_e(k)$ is used to construct a data based Kalman filter, the Kalman gain of the resulting data based filter is too low (see 4.22). While a too low filter gain reduces the quality of the state estimates, this will not cause an unstable filter. It thus appears that the consequences of underestimating $R_{y_s}(0)$ are far more severe than the consequences of overestimating $R_{y_s}(0)$.

Therefore, since the consequences of underestimating $R_{y_s}(0)$ are less severe than the consequences of underestimating $R_{y_s}(0)$, it is preferable to use a biased estimator that overestimates $R_{y_s}(0)$ to construct the approximate state filter.

If M and $R_{y_s}(0)$ are perfectly estimated, and $\sigma_i = 0$ for all $i > l$ (as is assumed in Propositions 4.2 and 4.3) then the reduced order filter is stable. In practice, the singular values $\sigma_l + 1, \ldots, \sigma_{n_x}$ are not exactly zero. While the consequences of $\sigma_i << 1$ instead of $\sigma_i = 0$ are difficult to analyze, experiments show that this can sometimes cause the approximate filter to become unstable. This is especially true if the order of the system is relatively large. The same experiments show that using $(1-\varepsilon)A$ with $\varepsilon > 0$ instead of A itself to construct the approximate Kalman filter again stabilizes the approximate Kalman filter. While this method to stabilize the approximate Kalman filter has worked in simulation trails, a better analysis for the case $\sigma_i << 1$ is required to derive structured methodology to stabilize experimentally obtained state filters.

4.6 Summary and conclusions

Classical state estimation has the objective to minimize the mean square error (MSE) between the estimated state and the actual state vector. In order to reach this objective, knowledge of the covariances of both the process noise and measurement disturbances is required. In most first principles models this knowledge is not available.

For linear systems, the problem of unknown covariances for the process noise and measurement disturbances can be tackled via multiple methods. A common point in all these methods is that they use input and output data collected from the physical system to determine the missing information. In this chapter we considered the use of the so-called covariance method. This covariance method has the advantage that the state filter can be derived using only linear operations.

The covariance method has been analyzed in section 4.3. In this section we showed

that the method can lead to poor state estimates when the observability matrix of the system is poorly conditioned. Even very small estimation errors for the disturbance properties can indeed result in major estimation errors if the observability matrix is poorly conditioned.

To solve this problem, we proposed a modified covariance procedur. In this modified procedure the state is estimated into two stages. In the first stage (see section 4.3.2) a state filter that only estimates the components of the vector that correspond to the easily observable directions is designed. Then a separate state estimator for the states that are poorly observable is derived.

In section 4.4 the effectiveness of the modified method was demonstrated in a simulation example. For this example, using the unmodified covariance method resulted in a unstable state filter. In contrast, using the modified covariance method a filter can be obtained that has nearly the same MSE as the optimal Kalman filter that would be derived if the covaraiances for the process noise and measurement disturbances had been available.

Unfortunately, there appears to be no easy generalization of the modified covariance method such that it can also be used for nonlinear process models.

4.A Proof of proposition 4.2

4.A.1 Introduction

Although the proof of proposition 4.2 is quite straightforward, it is tedious and lengthy. For readability purposes it was chosen to split up this section into two subsections. In the first subsection some supporting lemmas are presented. These lemmas will be used in the main proof which is presented in the second subsection.

4.A.2 Supporting lemmas

Lemma 4.4 *Consider the observability matrix* $\mathcal{O} \in \mathbb{R}^{Nn_y \times 1}$*:*

$$\mathcal{O} = \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \\ \vdots \\ \tilde{C}\tilde{A}^{N-1} \end{bmatrix} \tag{4.A.1}$$

with $N \geq n_x$ *and* $\tilde{A} \in \mathbb{R}^{n_x \times n_x}$ *and* $\tilde{C} \in \mathbb{R}^{n_y \times n_x}$*. Furthermore define sets of vectors* W_1 *and* W_2 *as:*

$$W_1 \triangleq \left\{ w_1 : \; w_1 = \begin{bmatrix} u_1 \\ 0 \end{bmatrix}, u_1 \in \mathbb{R}^l \backslash \{0\} \right\} \tag{4.A.2}$$

$$W_2 \triangleq \left\{ w_2 : \; w_2 = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, u_2 \in \mathbb{R}^{n_x - l} \right\}. \tag{4.A.3}$$

If for all possible $w_1 \in W_1$ *and* $w_2 \in W_2$ *it holds that*

$$\mathcal{O}w_1 \neq 0 \tag{4.A.4}$$

$$\mathcal{O}w_2 = 0, \tag{4.A.5}$$

then it holds that:

$$\tilde{A}w_2 \in W_2. \tag{4.A.6}$$

Proof Suppose that there would exist $w_2 \in W_2$ such that:

$$\tilde{A}w_2 \notin W_2. \tag{4.A.7}$$

Then it must hold that

$$\tilde{A}w_2 = r_1 + r_2, \tag{4.A.8}$$

with $r_1 \in W_1$ and $r_2 \in W_2$. This implies that

$$\mathcal{O}\tilde{A}w_2 = \mathcal{O}r_1 + \mathcal{O}r_2 \neq 0, \tag{4.A.9}$$

because $\mathcal{O}r_1 \neq 0$ (see (4.A.4)) and $\mathcal{O}r_2 = 0$ (see (4.A.5)). Since $\mathcal{O}\tilde{A}w_2 \neq 0$ there must exist a value of $p \in \{1, 2, \ldots\}$ such that $\tilde{C}\tilde{A}^p w_2 \neq= 0$. Using the definition of (4.A.1) this means that:

$$\mathcal{O}w_2 \neq 0, \tag{4.A.10}$$

which contradicts (4.A.5). ■

Lemma 4.5 *Consider the transformed system matrix $\tilde{C}$ as defined in (4.48). If $\sigma_i = 0$ for $i > l$, then $\tilde{C}$ has the form $\tilde{C} = [CV_{1:l}^T\ 0]$.*

Proof The results is easy to see by realizing that the columns of $V_{l+1:n}$ all correspond to non-observable directions, which by definition implies that:

$$\mathcal{O}V_{l+1:n} = 0. \tag{4.A.11}$$

Given the first block row of $\mathcal{O}$ in (4.26), we thus also have that:

$$CV_{l+1:n} = 0. \tag{4.A.12}$$

The transformed matrix $\tilde{C}$ is thus:

$$\begin{aligned} \tilde{C} &= CV && (4.A.13)\\ &= [CV_{1:l}\ CV_{l+1:n}] && (4.A.14)\\ &= [\tilde{C}_1\ 0]. && (4.A.15) \end{aligned}$$

■

Lemma 4.6 *Let the SVD of $\mathcal{O}$ as defined in (4.26) is given by (4.40) and let Π be defined as:*

$$\Pi = \begin{bmatrix} I_l & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.A.16}$$

with I_l the $l \times l$ identity matrix. If $\sigma_i = 0$ for $i > l$ then

a.

$$\Pi\tilde{A}\Pi = \Pi\tilde{A}. \tag{4.A.17}$$

b.

$$\Pi\tilde{C}^T = \tilde{C}^T \tag{4.A.18}$$

Proof

a. Because $\tilde{A}_{12} = 0$, it is easily shown that:

$$\begin{aligned} \Pi\tilde{A} &= \begin{bmatrix} \tilde{A}_{11} & 0 \\ 0 & 0 \end{bmatrix} && (4.A.19)\\ \Pi\tilde{A}\Pi &= \begin{bmatrix} \tilde{A}_{11} & 0 \\ 0 & 0 \end{bmatrix}, && (4.A.20) \end{aligned}$$

   and thus $\Pi\tilde{A} = \Pi\tilde{A}\Pi$.

b. Result follows immediately from Lemma 4.5 and definition of Π.

■

Corollary 4.7 *As a result of Lemma 4.6 it can be shown that for any matrix W of appropriate dimension it holds that:*

a.

$$\Pi\tilde{A}W\tilde{C}^T = \Pi\tilde{A}\Pi W\Pi\tilde{C}^T \tag{4.A.21}$$

b.

$$\Pi\tilde{A}W\tilde{A}^T\Pi = \Pi\tilde{A}\Pi W\Pi\tilde{A}^T\Pi. \tag{4.A.22}$$

Proof Both results are easily proven by applying Lemma 4.6 on both sides of W. ∎

4.A.3 Main proof

The main proof of proposition 4.2 is split up into 2 parts. In the first part it will be proved that $\tilde{\Sigma}_{11,l}(k) = \tilde{\Sigma}_{11}(k)$ and $R_{e,l}(k) = R_e(k)$. Afterwards it will be shown that $\hat{\tilde{x}}_{1,l}(k) = \hat{\tilde{x}}_1(k)$.

The proof that $\tilde{\Sigma}_{11,l}(k) = \tilde{\Sigma}_{11}(k)$ and $R_{e,l}(k) = R_e(k)$ can be constructed via induction. First we verify that for $k = 0$ it holds that $\tilde{\Sigma}_{11,l}(0) = \tilde{\Sigma}_{11}(0)$ and $R_{e,l}(0) = R_e(0)$. This is easy to verify, since at $k = 0$ the Kalman filter is initialized using $\tilde{\Sigma}(0) = 0$ and $R_e(0) = R_{y_s}(0)$.

To prove that $\tilde{\Sigma}_{11,l}(k) = \tilde{\Sigma}_{11}(k)$ and $R_{e,l}(k) = R_e(k)$ for $k > 0$ the following three implications need to be proved:

$$\Pi\tilde{\Sigma}_l(k)\Pi = \Pi\tilde{\Sigma}(k)\Pi \quad \Rightarrow \quad R_{e,l}(k) = R_e(k) \tag{4.A.23}$$

$$\left.\begin{array}{rcl} \Pi\tilde{\Sigma}_l(k)\Pi &=& \Pi\tilde{\Sigma}(k)\Pi \\ R_{e,l}(k) &=& R_e(k) \end{array}\right\} \quad \Rightarrow \quad \Pi\tilde{K}_l(k) = \Pi\tilde{K}(k) \tag{4.A.24}$$

$$\left.\begin{array}{rcl} \Pi\tilde{\Sigma}_l(k)\Pi &=& \Pi\tilde{\Sigma}(k)\Pi \\ R_{e,l}(k) &=& R_e(k) \\ \Pi\tilde{K}_l(k) &=& \Pi\tilde{K}(k) \end{array}\right\} \quad \Rightarrow \quad \Pi\tilde{\Sigma}_l(k+1)\Pi = \Pi\tilde{\Sigma}(k+1)\Pi, \tag{4.A.25}$$

with Π defined as in (4.A.16).

Using Lemma 4.5 it can be shown that:

$$\begin{aligned} \tilde{C}\Pi &= \tilde{C} && (4.A.26)\\ \tilde{C}(I-\Pi) &= 0. && (4.A.27) \end{aligned}$$

These last two equations together with (4.23) can be used to prove (4.A.23):

$$\begin{aligned} R_e(k) &= R_{y_s}(0) - \tilde{C}\tilde{\Sigma}(k)\tilde{C}^T && (4.A.28)\\ &= R_{y_s}(0) - \tilde{C}[\Pi + (I-\Pi)]\tilde{\Sigma}(k)[\Pi + (I-\Pi)]\tilde{C}^T \\ &= R_{y_s}(0) - \tilde{C}\Pi\tilde{\Sigma}(k)\Pi\tilde{C}^T && (4.A.29)\\ &= R_{y_s}(0) - \tilde{C}\Pi\tilde{\Sigma}_l(k)\Pi\tilde{C}^T && (4.A.30)\\ &= R_{y_s}(0) - \tilde{C}\tilde{\Sigma}_l(k)\tilde{C}^T && (4.A.31)\\ &= R_{e,l}(k). && (4.A.32) \end{aligned}$$

To prove the second induction equation (4.A.24), we start by expanding the expression of $\Pi\tilde{K}(k)$ using (4.22)):

$$\Pi\tilde{K}(k) = [\Pi\tilde{M} - \Pi\tilde{A}\tilde{\Sigma}(k)\tilde{C}^T]R_e^{-1}(k) \tag{4.A.33}$$

Using Corollary 4.7 twice we have that:

$$\begin{aligned} \Pi\tilde{A}\tilde{\Sigma}(k)\tilde{C}^T &= \Pi\tilde{A}\Pi\tilde{\Sigma}(k)\Pi\tilde{C}^T && (4.A.34)\\ &= \Pi\tilde{A}\Pi\tilde{\Sigma}_l(k)\Pi\tilde{C}^T && (4.A.35)\\ &= \Pi\tilde{A}\tilde{\Sigma}_l(k)\tilde{C}^T, && (4.A.36) \end{aligned}$$

since we assume that $\Pi\tilde{\Sigma}_l(k)\Pi = \Pi\tilde{\Sigma}\Pi$. It was also assumed that $R_e(k) = R_{e,l}(k)$ and from the definition of M_l in (4.52) we have that $\Pi\tilde{M} = \Pi\tilde{M}_l = \tilde{M}_l$. Using these relations, (4.A.33) can be rewritten as:

$$\begin{aligned} \Pi\tilde{K}(k) &= [\Pi\tilde{M} - \Pi\tilde{A}\tilde{\Sigma}(k)\tilde{C}^T]R_e^{-1}(k) && (4.A.37)\\ &= \Pi[\tilde{M}_l - \tilde{A}\tilde{\Sigma}_l(k)\tilde{C}^T]R_{e,l}^{-1}(k) && (4.A.38)\\ &= \Pi\tilde{K}_l(k). && (4.A.39) \end{aligned}$$

The third and final induction relation can be proved by using Corollary 4.7. We will start with the normal recursive equation (4.23) for $\Pi\tilde{\Sigma}(k)\Pi$:

$$\Pi\tilde{\Sigma}(k+1)\Pi = \Pi\tilde{A}\tilde{\Sigma}(k)\tilde{A}^T\Pi + \Pi\tilde{K}R_e(k)\tilde{K}(k)^T\Pi. \tag{4.A.40}$$

Using Lemma 4.7 twice, the first term on the right hand side satisfies:

$$\begin{aligned} \Pi\tilde{A}\tilde{\Sigma}(k)\tilde{A}^T\Pi &= \Pi\tilde{A}\Pi\tilde{\Sigma}(k)\Pi\tilde{A}^T\Pi && (4.A.41)\\ &= \Pi\tilde{A}\Pi\tilde{\Sigma}_l(k)\Pi\tilde{A}^T\Pi && (4.A.42)\\ &= \Pi\tilde{A}\tilde{\Sigma}_l(k)\tilde{A}^T\Pi. && (4.A.43) \end{aligned}$$

Using the assumptions that $R_e(k) = R_{e,l}(k)$ and $\Pi\tilde{K}(k) = \Pi\tilde{K}_l(k)$ the second term on the right-hand side of (4.A.40) satisfies:

$$\Pi\tilde{K}(k)R_e(k)\tilde{K}(k)^T\Pi = \Pi\tilde{K}_l(k)R_{e,l}\tilde{K}_l^T(k)\Pi. \tag{4.A.44}$$

Substituting (4.A.43) and (4.A.44) in (4.A.40) results in:

$$\begin{aligned} \Pi\tilde{\Sigma}(k+1)\Pi &= \Pi\tilde{A}\tilde{\Sigma}_l(k)\tilde{A}^T\Pi + \Pi\tilde{K}_l(k)R_{e,l}(k)\tilde{K}_l(k)^T\Pi\\ &= \Pi\tilde{\Sigma}_l(k+1)\Pi. && (4.A.45) \end{aligned}$$

In the second part of this proof it still needs to be shown that $\hat{\tilde{x}}_{1,l}(k) = \hat{\tilde{x}}_1(k)$. Again, the easiest method to prove this is via induction. The initial estimate is independent of the choice of M so for $k = 0$ we have that

$$\hat{\tilde{x}}_{1,l}(0) = \hat{\tilde{x}}_1(0). \tag{4.A.46}$$

Using Lemma 4.4 and 4.5 new estimates for of $\hat{\tilde{x}}(k+1)$ are constructed using:

$$\hat{\tilde{x}}_1(k+1) = \tilde{A}_{11}\hat{\tilde{x}}_1(k) + \tilde{K}_1(k)[y(k) - \tilde{C}_1\hat{\tilde{x}}_1(k)] \tag{4.A.47}$$

with $\tilde{K}_1(k)$ the first l rows of $\tilde{K}(k)$ and $\tilde{C}_1$ the first l columns of $\tilde{C}$. The new estimates for $\hat{\tilde{x}}_{1,l}(k)$ are computed via:

$$\hat{\tilde{x}}_{1,l}(k+1) = \tilde{A}_{11}\hat{\tilde{x}}_{1,l}(k) + \tilde{K}_{1,l}(k)[y(k) - \tilde{C}_1\hat{\tilde{x}}_{1,l}(k)]. \tag{4.A.48}$$

In (4.A.39) it was already proven that $\tilde{K}_{1,l}(k) = \tilde{K}_1(k)$ for all k. The right hand sides of (4.A.47) and (4.A.48) are equal, and thus we can conclude that $\hat{\tilde{x}}_{1,l}(k) = \hat{\tilde{x}}_1(k)$ for all k. ∎

4.B Proof of proposition 4.3

Using the assumption that $e(k)$ is zero mean and Gaussian, we use the BLUE to show that (see for instance [3])

$$\mathbb{E}\left[(\tilde{x}(k)-\hat{\tilde{x}}(k)|e(k))\right] = R_{(\tilde{x}(k)-\hat{\tilde{x}}(k))e(k)}R_e^{-1}(k)e(k) \tag{4.B.49}$$

with

$$\begin{aligned} R_{(\tilde{x}(k)-\hat{\tilde{x}}(k))e(k)} &= \mathbb{E}(\tilde{x}(k)-\hat{\tilde{x}}(k))e(k)^T. && (4.B.50)\\ &= \mathbb{E}(\tilde{x}(k)-\hat{\tilde{x}}(k))(\tilde{x}(k)-\hat{\tilde{x}}(k))^T\tilde{C}^T. && (4.B.51)\\ &= \tilde{P}(k)\tilde{C}^T. && (4.B.52)\end{aligned}$$

In the step from (4.B.50) to (4.B.51) we use (4.7) and in the step from (4.B.51) to (4.B.52) $\tilde{P}(k)$ is defined as:

$$\tilde{P}(k) = \mathbb{E}(\tilde{x}(k)-\hat{\tilde{x}}(k))(\tilde{x}(k)-\hat{\tilde{x}}(k))^T. \tag{4.B.53}$$

Since the estimation error $\tilde{x}(k)-\hat{\tilde{x}}(k)$ is uncorrelated to $\hat{\tilde{x}}(k)$ we can show that

$$\begin{aligned} \tilde{\Phi}(k) &= \mathbb{E}\{\tilde{x}(k)\tilde{x}(k)^T\} && (4.B.54)\\ &= \mathbb{E}\{[\hat{\tilde{x}}(k)-(\hat{\tilde{x}}(k)-\tilde{x}(k))][\hat{\tilde{x}}(k)-(\hat{\tilde{x}}(k)-\tilde{x}(k))]^T\} && (4.B.55)\\ &= \mathbb{E}\{\hat{\tilde{x}}(k)\hat{\tilde{x}}(k)^T+(\hat{\tilde{x}}(k)-\tilde{x}(k))(\hat{\tilde{x}}(k)-\tilde{x}(k))^T\} && (4.B.56)\\ &= \tilde{\Sigma}(k)+\tilde{P}(k), && (4.B.57)\end{aligned}$$

see also [46]. This means that $\tilde{P}(k)$ satisfies:

$$\tilde{P}(k) = \tilde{\Phi}(k)-\tilde{\Sigma}(k). \tag{4.B.58}$$

Inserting (4.B.52) - (4.B.58) into (4.B.49) results in:

$$\mathbb{E}\left[(\tilde{x}(k)-\hat{\tilde{x}}(k)|e(k))\right] = [\tilde{\Phi}(k)-\tilde{\Sigma}(k)]\tilde{C}^TR_e^{-1}(k)e(k). \tag{4.B.59}$$

Using (4.14), (4.19) and (4.21), the matrix $\tilde{\Phi}(k)$ can be expressed as a function of $\tilde{M}(k)$ via:

$$\tilde{M}(k) = \tilde{A}\tilde{\Phi}(k)\tilde{C}^T, \tag{4.B.60}$$

and so, (4.B.59) can also be written as:

$$\mathbb{E}\left[(\tilde{x}(k)-\hat{\tilde{x}}(k)|e(k))\right] = [\tilde{A}^{-1}\tilde{M}(k)-\tilde{\Sigma}(k)\tilde{C}^T]R_e^{-1}(k)e(k). \tag{4.B.61}$$

The term $\tilde{A}^{-1}\tilde{M}(k)$ can also be written as:

$$\begin{aligned} \tilde{A}^{-1}\tilde{M} &= \begin{bmatrix} \tilde{A}_{11}^{-1} & 0 \\ -\tilde{A}_{22}^{-1}\tilde{A}_{21}\tilde{A}_{11}^{-1} & \tilde{A}_{22}^{-1}\end{bmatrix}\begin{bmatrix}\tilde{M}_1(k)\\ \tilde{M}_2(k)\end{bmatrix}\\ &= \begin{bmatrix} \tilde{A}_{11}^{-1}\tilde{M}_1(k) \\ \tilde{A}_{22}^{-1}\tilde{M}_2(k)-\tilde{A}_{22}^{-1}\tilde{A}_{21}\tilde{A}_{11}^{-1}\tilde{M}_1(k)\end{bmatrix}. && (4.B.62)\end{aligned}$$

Similarly, $\tilde{\Sigma}(k)\tilde{C}^T$ can be written as:

$$\begin{bmatrix}\tilde{\Sigma}_{11}(k) & \tilde{\Sigma}_{12}(k)\\ \tilde{\Sigma}_{21}(k) & \tilde{\Sigma}_{22}(k)\end{bmatrix}\begin{bmatrix}\tilde{C}_1\\ 0\end{bmatrix} = \begin{bmatrix}\tilde{\Sigma}_{11}(k)\tilde{C}_1^T\\ \tilde{\Sigma}_{21}(k)\tilde{C}_1^T\end{bmatrix}. \tag{4.B.63}$$

Inserting (4.B.62) and (4.B.63) into (4.B.61) results in

$$\mathbb{E}\left[\begin{array}{c}(\tilde{x}_1(k)-\hat{\tilde{x}}_1(k))|e(k)\\(\tilde{x}_2(k)-\hat{\tilde{x}}_2(k))|e(k)\end{array}\right]= \left[\begin{array}{c}\tilde{A}_{11}^{-1}\tilde{M}_1(k)-\tilde{\Sigma}_{11}(k)\tilde{C}_1^T\\ \tilde{A}_{22}^{-1}\tilde{M}_2(k)-\tilde{A}_{22}^{-1}\tilde{A}_{21}\tilde{A}_{11}^{-1}\tilde{M}_1(k)-\tilde{\Sigma}_{21}(k)\tilde{C}_1^T\end{array}\right] R_e^{-1}(k)e(k). \quad (4.B.64)$$

Multiplying the first line by $\tilde{A}_{21}$ and the second line by $\tilde{A}_{22}$ concludes the proof of Proposition 4.3. ■

Chapter 5

On online model selection for state estimation

5.1 Introduction

A common assumption in state estimation problems is that the available large scale first principles model perfectly describes the true process that is to be monitored. Specifically it is assumed that the available large scale model is of the form:

$$x(k+1) = f(x(k),u(k),w(k),\theta_0) \quad (5.1)$$
$$y(k) = h(x(k),u(k),v(k),\theta_0) \quad (5.2)$$

with $\theta_0 \in \mathbb{R}^{n_\theta \times 1}$ a fixed known vector of process variables. The assumption that (5.1)-(5.2) perfectly describes the process behavior is an important assumption because it can be shown that the accuracy of the model can greatly affect the accuracy of the estimated states, see for instance [55][90][101].

In practice the assumption that (5.1)-(5.2) perfectly describes the process behavior is not realistic. Even if the available model of the form (5.1)-(5.2) is a perfect description of the process at a certain time instant $k = k_0$, it is not assured that this model description will remain perfect over time. Deviations might arise slowly (for instance due to wear of process components) or might occur abruptly (for instance due to sudden sensor or actuator faults). In this chapter we will introduce a framework that can deal with both types of errors.

In this chapter we will relax the assumption that (5.1)-(5.2) perfectly describes process behavior. Instead of assuming that (5.1)-(5.2) is a perfect model, we will assume that there exists a time varying parameter vector $\theta(k)$ such that the model of the form:

$$x(k+1) = f(x(k),u(k),w(k),\theta(k)) \quad (5.3)$$
$$y(k) = h(x(k),u(k),v(k),\theta(k)) \quad (5.4)$$

is able to model the behavior of the process. The exact sequence of parameters $\theta(k)$ that satisfy this assumption are assumed unknown. Note that although this assumption

is a relaxation of the perfect plant assumption, it does not encompass all possible modelling errors.

Although the parameters $\theta(k)$ are not known, it is often possible to estimate these parameters simultaneously with the process states. A commonly used method to estimate both states $x(k)$ and parameters $\theta(k)$ is to create an extended version of the state equation (5.1) in which time evolution of the parameter vector $\theta(k)$ is modelled as a random walk:

$$\begin{bmatrix} x(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} f(x(k),u(k),w(k),\theta(k)) \\ \theta(k)+w_\theta \end{bmatrix} \tag{5.5}$$

with $w_\theta(k)$ zero mean Gaussian white noise with an user defined covariance matrix Q_θ.

After introducing an extended state vector $\tilde{x}(k)$ and an extended process noise $\tilde{w}(k)$ defined as:

$$\tilde{x}(k) = \begin{bmatrix} x(k) \\ \theta(k) \end{bmatrix} \quad \tilde{w} = \begin{bmatrix} w(k) \\ w_\theta(k) \end{bmatrix} \tag{5.6}$$

it is possible to write the extended model in the general state-space form:

$$\tilde{x}(k+1) = \tilde{f}(\tilde{x}(k),u(k),\tilde{w}(k)) \tag{5.7}$$

$$y(k) = \tilde{h}(\tilde{x}(k),u(k),v(k)). \tag{5.8}$$

Since the model is now in the usual state-space form, we can use any nonlinear state estimation technique to simultaneously estimate the states $x(k)$ and the parameters $\theta(k)$.

After tuning the noise covariance matrix Q_θ of the random walk, we thus have a model which can cope with variations in process variables θ. The price we have to pay for the extra robustness in the simulation model is that if both states and parameters are estimated, the variance error of the estimated states will increase. The additional variance error increases with the number of parameters $\theta(k)$ that have to be estimated.

Instead of using a single model fixed to predict and estimate the state of the system better results could be obtained using a multiple model approach. In this multiple model approach we estimate the state $x(k)$ and parameters $\theta(k)$ of the system using multiple models. The first model is the nominal model (5.1)-(5.2) in which the parameters are assumed constant ($\theta(k) = \theta(k-1)$). Besides the nominal model we also estimate the state of the system with one or more extended models of the form (5.3)-(5.4). Based upon the fit of each model to the available measurements $y(k)$ we then (ideally) determine which filter has produced the best estimates for $x(k)$ and $\theta(k)$ in terms of the mean squared error. At time $k+1$ the selected estimation results for $x(k)$ and $\theta(k)$ form the starting point for each of the filters to predict and estimate the state at time $k+1$, and so on.

The advantage of the multiple model approach is that it enables the user to get a filter estimate with a relatively small variance. The relatively small variance is obtained by using the nominal model when the data indicates the $\theta(k)$ has not changed. The extended model (5.5) only has to be used when the data indicates that $\theta(k)$ has changed

significantly. In all cases we avoid using a model structure that contains unnecessary parameters that have to be estimated.

Using multiple models to estimate the state of a system is a technique that is often encountered in Fault Detection and Isolation (FDI) literature [29]. In FDI literature the model parameters $\theta(k)$ usually have the interpretation of possible faults. The parameters should nominally remain at $\theta = 0$, but if a fault in the process occurs, it is assumed that $\theta(k) \neq 0$. To isolate which fault has occurred, the parameter vector $\theta(k)$ is estimated, often using a random walk model for $\theta(k)$ such as in (5.5). Those elements that deviate from zero correspond to the fault location. If no fault has occurred, on average the best estimate of the state is obtained using the nominal model in which $\theta = 0$. When a fault has indeed occurred, better results are likely obtained using the fault model (in which θ is a free parameter). The model selection problem in FDI literature is thus to decide when it is necessary to estimate the parameters $\theta(k)$.

The FDI literature about model selection problems can be roughly divided in two separate categories. The first category uses the nominal model to generate a residual signal $r(k)$. Once statistical tests such as simple linear χ^2-tests or nonlinear tests as CUSUM (=Cumulative Sum) or SPRT (Sequential Probability Ratio Test) [74] on the residual show that the nominal model is no longer valid, the extended model is used to isolate the fault. For linear systems with Gaussian noise disturbances a popular method to generate the required residual signal is to use the innovation sequence $e(k)$ (see (2.37) of the linear Kalman filter, so $r(k) = e(k)$ [65]. This is commonly extended to nonlinear systems by using nonlinear filters such as the extended Kalman filter or particle filters instead of the linear Kalman filter to generate the innovation sequence $e(k)$ [20][45].

Instead of only using the fault model once it has been determined that the nominal model is not valid, the second class of model selection algorithms uses a different procedure. At time k the states and parameters are estimated using all available models (both nominal and fault models). After the estimation a selection algorithm is used to determine which of the filter models will most likely have resulted in the best estimates. The results of this model are then used again as a starting point for all available models at time $k+1$ and so on. The algorithm we will introduce in this chapter also belongs in this class of methods.

In order to determine which model has resulted in the best possible state and parameter estimates, many different algorithms have already been suggested. The model selection procedures in the second class of selection algorithms are often based on statistical criteria, see for instance [33]. Assume that n models of the form (5.1)-(5.2) are available, denote these models as $\mathcal{M}_i$, with $i = 1, 2, \ldots, n$. Then the filtering procedure is carried out for each model, on data y. Afterwards, using Bayes conditional probability theory, the conditional probability $p(\mathcal{M}_i|y)$ is computed. The model with the highest conditional probability is then selected, and the state estimates based on this model are used. The conditional probability $p(\mathcal{M}_i|y)$ can be computed via:

$$p(\mathcal{M}_i|y) = p(y|\mathcal{M}_i)p(\mathcal{M}_i)/p(y). \tag{5.9}$$

Using this equation for complex process models is generally difficult in practice, because the term $p(y|\mathcal{M}_i)$ is not trivial to compute for non-linear systems and knowledge

of the a-priori probability of each model $p(\mathcal{M}_i)$ is rarely available.

An alternative approach to the model selection problem is presented in [91]. If a moving horizon state estimator (MHE) is used for state estimation, the state estimation problem is written as a weighted and regularized least squares problem. The problem of model selection is therefore similar to model selection in system identification theory. Indeed in system identification, model selection is used to determine the best candidate model based upon the least squares fit of each candidate model to estimation data. Given this similarity, the model selection is done using a selection criterion from identification literature. In the case of [91] the Akaike Information Criterion (AIC) is chosen. Advantages of this approach are that exact probability distributions are no longer required, and the technique can also be easily adapted for non-linear models. Drawbacks of this approach are that the technique can only be used in conjunction with moving horizon estimators. Another drawback is that the AIC criterion may not be the best criterion, since it was derived only for least squares problems without weighting and regularization, while states are estimated with weighting and regularization.

In this chapter a model selection procedure will be considered that is closely related to [91]. Instead of using the generic AIC criterion, a specialized criterion for weighted and regularized least squares problems is derived. It will be shown that such a selection criterion can also by used in conjunction with other filters than just the MHE filter. Apart from deriving the new model selection algorithm, we will also present the statistical properties of the new criterion.

The remainder of this chapter is organized as follows; in section 5.2 we will introduce our model selection procedure. In section 5.3 we will derive the statistical properties of the selection algorithm proposed in the previous section. Section 5.4 modifies the proposed selection scheme such that the results can be tuned. In the final section of this chapter, the effectiveness of the model selection scheme will be demonstrated in simulation examples.

5.2 Model selection for state estimation

As mentioned in the introduction, our objective is to estimate the state of a process with the smallest possible mean squared error. To this end we assume we have available known inputs $u(k)$, measured inputs $y(k)$ and a set of candidate models (possibly nonlinear). For each candidate model it is possible to design a state filter to and estimate the state of the system. From all estimates, we wish to select that estimate which has the smallest mean squared error.

In order to derive our model selection procedure we will (for simplicity) first assume that the system is described by the following linear model:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+w(k) && (5.10)\\ y(k) &= Cx(k)+v(k), && (5.11) \end{aligned}$$

where $w(k)$ and $v(k)$ are Gaussian noise processes with

$$\mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} = 0 \tag{5.12}$$

$$\mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} \begin{bmatrix} w_l^T & v_l^T \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \delta_{k,l}. \tag{5.13}$$

If the model (5.10)-(5.11) perfectly describes the true system, an optimal estimate of the state vector $x(k)$ at each time k can be obtained using a Kalman filtering procedure. As discussed in section 2.3.2 such a filtering procedure consists of two steps. In the prediction step, a prediction $\hat{x}(k+1|k)$ of the state vector is given, along with its covariance matrix $\hat{P}_{x(k+1|k)}$. In the measurement update, the measurement $y(k+1)$ is used together with the predicted state $\hat{x}(k+1|k)$ to compute the estimate $\hat{x}(k+1|k+1)$ and to produce its covariance matrix $\hat{P}_{x(k+1|k+1)}$. An important result for the remainder is that the measurement update in a Kalman filtering procedure can be seen as the solution of a Weighted and Regularized Least Squares (WRLS) problem [46]:

$$\hat{x}(k|k) = \arg\min_x \|y(k) - Cx\|^2_{R^{-1}} + \|x - \hat{x}(k|k-1)\|^2_{\hat{P}^{-1}_{x(k|k-1)}}, \tag{5.14}$$

with $\|z\|^2_W = z^T W z$ for $z \in \mathbb{R}^n$ (see also (2.41)).

If the system is non-linear, such as e.g. in (5.1)-(5.2), an estimate of the state vector $x(k)$ can again be obtained using very similar procedures (e.g. using the EKF or UKF instead of the normal Kalman Filter) which also consist of two steps: a prediction step and a measurement update step. Define $\hat{y}(k,x)$ as the predictor of the output vector $y(k)$ using the state vector x and the available model (5.1)-(5.2) i.e.

$$\hat{y}(k,x) = \mathbb{E}_{v(k)} h(x, u(k), v(k)). \tag{5.15}$$

If $(x - \hat{x}(k|k-1))$ is Gaussian distributed, and

$$\hat{y}(x) \approx Cx \quad \forall \quad \|(x - \hat{x}(k|k-1))\|^2_{P^{-1}_{\hat{x}(k|k-1)}} < \chi^2_\alpha, \tag{5.16}$$

then the the measurement update in the filtering procedure is still approximately a WRLS problem:

$$\hat{x}(k|k) = \arg\min_x \|y(k) - \hat{y}(k,x)\|^2_{R^{-1}} + \|x - \hat{x}(k|k-1)\|^2_{P^{-1}_{\hat{x}(k|k-1)}}, \tag{5.17}$$

In practice the measurement update in a filtering procedure will still be accurately described by (5.17), even if $(x - \hat{x}(k|k-1))$ is not exactly Gaussian distributed.

The state estimation procedure delivers an estimate $\hat{x}(k) = \hat{x}(k|k)$ for $x(k)$ under the assumption that the available model perfectly describes the true system. As stated in the introduction, often the available model is only an approximation of the true system and the quality of the estimate of $x(k)$ will depend on the quality of the chosen model. Consequently, the model will have to be chosen in such a way that the state estimation procedure based on this model delivers a good estimate of $x(k)$. As already

mentioned, we will consider the particular situation where we have several candidate-models (e.g. a model with fixed parameters and a similar model with extra time-varying parameters) and we have to select, among these models, the model which will deliver the best estimate of the state vector. To make this selection, only known inputs and measured outputs are available.

Our selection procedure will be based on a measure of the quality of the model for the estimation of the state vector. Different measures can be considered for this purpose. In this chapter, we will define a measure of quality which is very similar to the quality measures used in system identification. Given a model of the type (5.1)-(5.2) and given the estimate $\hat{x}(k)$ of the state vector at time k obtained using the WRLS problem (5.17), the measure of quality $\mathcal{V}(k)$ at time k is defined as follows:

$$\mathcal{V}(k) = \mathbb{E}_{\hat{x}(k)} \overline{V}(\hat{x}(k),k) \tag{5.18}$$

$$\text{with } \overline{V}(x,k) = \mathbb{E}_{y(k)} \|y(k) - \hat{y}(x)\|^2_{R^{-1}}. \tag{5.19}$$

Using (5.19), we see that $\overline{V}(\hat{x}(k),k)$ represents the ability of the model and the available estimate $\hat{x}(k)$ to predict not only the particular realization of the output vector $y(k)$ that we used to estimate $\hat{x}(k)$, but also all other possible realizations of $y(k)$. In (5.19) the prediction error $y(k) - \hat{y}(x)$ is weighted with R^{-1}, to take into account the variability of the measurements $y(k)$. The quantity $\overline{V}(\hat{x}(k),k)$ is still a random variable since $\hat{x}(k)$ is determined using noisy data. Therefore, it is better to consider its mean as measure of quality for the model such as we have done in (5.18). From the definition of $\mathcal{V}(k)$, we see that the smaller the time function $\mathcal{V}(k)$ is, the better is the model.

We have thus defined a measure $\mathcal{V}(k)$ of the quality of a model. This quantity can only be used in a quality assessment procedure if it is possible to compute (or to approximate) $\mathcal{V}(k)$ using the available data. In order to find a method for computing (or estimating) $\mathcal{V}(k)$, we first notice the strong analogy between $\mathcal{V}(k)$ and the Akaike's Final Prediction Error (FPE) $\overline{J}_p(\mathcal{M})$ that is used to assess the quality of a model $\mathcal{M}$ in system identification theory. The FPE is defined as (see [56]):

$$\overline{J}_p(\mathcal{M}) = \mathbb{E}_{\hat{\theta}} \overline{V}_{sysid}(\hat{\theta}) \tag{5.20}$$

in which

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{k=1}^{N} \|y(k) - \hat{y}(k,\theta)\|^2 \quad \text{and} \tag{5.21}$$

$$\overline{V}_{sysid}(\theta) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}_{y(k)} \|y(k) - \hat{y}(k,\theta)\|^2, \tag{5.22}$$

with $y(k)$ for $k = 1,\ldots,N$ a set of given output measurements and $\hat{y}(k,\theta)$ the prediction of the output using model parameters θ.

We can indeed interpret the FPE as a special case of $\mathcal{V}(k)$. After substituting $\hat{x}(k)$ for θ both criteria are equivalent if $\hat{x}(k)$ is estimated using (5.14) with $R = I$ and $\hat{P}^{-1}_{x(k+1|k)} = 0$.

Note that the expected value operator $\mathbb{E}_{\hat{\theta}}$ in (5.20) corresponds to the expected value operator $\mathbb{E}_{\hat{x}}$ in (5.18).

From system identification literature we know that the exact FPE cannot be computed as this would require an infinite amount of data. Therefore in practice the following approximation is commonly used [1]:

$$\overline{J}_p(\mathcal{M}) \approx \frac{1}{N}\sum_{k=1}^{N}\left[\|y(k)-\hat{y}(k,\hat{\theta})\|^2\right] + \lambda_0\frac{2d_{\mathcal{M}}}{N}, \tag{5.23}$$

with $d_{\mathcal{M}}$ the dimension of θ and $\lambda_0 = \overline{V}_{sysid}(\theta_0)$, with θ_0 the true model parameters. Since λ_0 is generally unknown, the following estimate of λ_0 is often used:

$$\hat{\lambda}_0 = \frac{1}{N-d_{\mathcal{M}}}\sum_{k=1}^{N}\left[\|y(k)-\hat{y}(k,\hat{\theta})\|^2\right]. \tag{5.24}$$

Substitution $\hat{\lambda}_0$ for λ_0 in (5.23) results in:

$$\overline{J}_p(\mathcal{M}) \approx \frac{N+d_{\mathcal{M}}}{N-d_{\mathcal{M}}}\frac{1}{N}\sum_{k=1}^{N}\left[\|y(k)-\hat{y}(k,\hat{\theta})\|^2\right]. \tag{5.25}$$

Like the FPE $\overline{J}_p(\mathcal{M})$, it is also impossible to compute our model quality criterion $\mathcal{V}(k)$ and thus we require a method to estimate our model quality measure. In [56] an approximation result for $\mathcal{V}(k)$ is given for the case where $\hat{x}(k)$ has been estimated by minimizing the WRLS problem (5.17) with $R = I$ and with $\hat{P}^{-1}_{\hat{x}(k+1|k)} = \delta I$ with δ a positive real constant. However, in practice $\hat{P}^{-1}_{\hat{x}(k+1|k)}$ cannot be written as δI, which implies that the approximations in [56] can not be used directly. Therefore the results of [56] need to be extended in order to be able to find a computable expression to accurately approximate $\mathcal{V}(k)$.

Proposition 5.1 *Let us consider the time instant k and the output vector $y(k)$ collected from the true system at that instant. Let us also consider the measure of quality $\mathcal{V}(k)$ defined in (5.18)-(5.19). Furthermore assume that the estimate $\hat{x}(k)$ of the state vector in (5.18)-(5.19) is obtained via the following weighted and regularized least squares problem which is equivalent to (5.17):*

$$\hat{x}(k) = \arg\min_{x}\left(V(x,k) \ +(x-x(k)^{\#})^T P^{-1}(x-x(k)^{\#})\right) \tag{5.26}$$

in which $x(k)^{\#}$ is a pre-specified state vector, P^{-1} is a positive semi-definite regularization matrix, and $V(x,k)$ is a weighted least squares criterium:

$$V(x,k) = \|y(k)-\hat{y}(x)\|^2_{R^{-1}}. \tag{5.27}$$

Then if $x(k)^{\#} \approx x^*(k) = \arg\min_x \overline{V}(x,k)$ the following expression is a generalized version of (5.23) and can be used to approximately compute $\mathcal{V}(k)$:

$$\mathcal{V}(k) \approx V(\hat{x}(k),k) + 4tr\left\{\left[(\psi^T R^{-1} L R^{-T}\psi)\right]\left[\overline{V}''(x^*(k),k) + 2P^{-1}\right]^{-1}\right\}, \tag{5.28}$$

where $L = \mathbb{E}(ee^T)$ *with* $e = y(k) - \hat{y}(x^*(k))$ *and*

$$\psi = \left.\frac{\partial \hat{y}(x)}{\partial x}\right|_{x=x^*(k)}, \tag{5.29}$$

$$\overline{V}''(x^*(k),k) = \left.\frac{\partial^2 \overline{V}(x,k)}{\partial x^2}\right|_{x=x^*(k)}, \tag{5.30}$$

under the condition that $y(k) - \hat{y}(x^*(k))$ *is approximately a white noise and that the dimension of the vector* $y(k)$ *is sufficiently large.*

Proof See Appendix 5.A. ∎

As mentioned in the statement of Proposition 5.1, the derivation of the approximation (5.28) requires that: $x(k)^{\#} \approx x^*(k) = \arg\min_x \overline{V}(x,k)$. Using this particular value for $x^{\#}(k)$ ensures that the estimate $\hat{x}(k)$ is unbiased. This additional requirement is also present in the less general version of [56] [1]. In practice the value $x^*(k)$ is generally unknown and one can only assume that $x(k)^{\#} \approx x^*(k)$.

The estimated asymptotic fit $\mathcal{V}(k)$ in (5.28) is dependent on L, which represents the covariance of the minimal asymptotic prediction error. For perfectly modelled systems with additive white noise, $L = R$, the covariance of the additive noise. The estimated asymptotic fit $\mathcal{V}(k)$ is the sum of the achieved fit $V(\hat{x}(k),k)$ on the measurement data and a term containing $\overline{V}''(x^*(k),k)$ and ψ. $\overline{V}''(x^*(k),k)$ can be approximated by the second derivative of $V(\hat{x}(k),k)$. Similarly, ψ can be approximated using the derivative of $\hat{y}(x)$ evaluated in $\hat{x}(k)$ instead of $x^*(k)$. If the measurement equation of the model is linear, for instance as is the case in (5.11), such approximations are not necessary as illustrated in the following corollary:

Corollary 5.2 *Consider the situation as assumed in Propostition 5.1. If additionally the measurement equation is of the form:*

$$y(k) = Cx(k) + v(k) \quad \textit{with} \tag{5.31}$$

$$\mathbb{E}\{v(k)\} = 0 \quad \textit{and} \quad \mathbb{E}\{v(k)v(k)\} = R, \tag{5.32}$$

then

$$\mathcal{V}(k) \approx V(\hat{x}(k),k) + 2tr\left\{ \left[(C^T R^{-1} L R^{-T} C)\right] \left[C^T R^{-1} C + P^{-1}\right]^{-1} \right\}, \tag{5.33}$$

where $L = \mathbb{E}(ee^T)$ *with* $e = y(k) - Cx^*(k)$.

Proof Using the linear output equation (5.31) it is easy to show that:

$$\psi = \left.\frac{\partial \hat{y}(x)}{\partial x}\right|_{x=x^*(k)} = C \tag{5.34}$$

[1] Note that expression (5.28) does not correspond to expression (16.36) in [56] as should be the case for $R = I$ and $P^{-1}_{\hat{x}(k+1|k)} = \delta I$. As a result of Proposition 5.1 an oversight in the derivation of (16.36) in [56] has been found, as acknowledged in [57].

and

$$\overline{V}''(x^*(k),k) = \left.\frac{\partial^2 \overline{V}(x,k)}{\partial x^2}\right|_{x=x^*(k)} = 2C^T R^{-1} C. \tag{5.35}$$

Substituting these results in (5.28) directly leads to (5.33). ■

Let us now summarize and define our model selection procedure. We wanted to select, among a set of candidate-models, the model which delivers the best estimate of the state vector. We have defined for this purpose a measure $\mathcal{V}(k)$ of the quality of a model. This measure is a time function and can be approximated using (5.28). In order to make the selection, we use this quality measure in the following procedure:

Procedure 5.3 *We want to determine the best filter model among n available models denoted $\mathcal{M}_1,\ldots,\mathcal{M}_n$. Denote the state estimates and associated covariances obtained using each of these models at time k as $\hat{x}^{[i]}(k|k)$ and $P^{[i]}_{\hat{x}(k|k)}$ with $i = 1,\ldots,n$. Then the selection procedure is as follows:*

1. *At some initial time k initialize all filters such that both $\hat{x}^{[i]}(k|k)$ and $P^{[i]}_{\hat{x}(k|k)}$ are the same for all $i = 1,\ldots,n$.*
2. *Compute state estimates and estimation covariances $\hat{x}^{[i]}(k+1|k+1)$ and $P^{[i]}_{\hat{x}(k+1|k+1)}$ for all models.*
3. *Estimate the model quality $\mathcal{V}(k+1)$ for all filter models using (5.28). Denote the estimated model quality at time $k+1$ for the i-th model as $\hat{\mathcal{V}}^{[i]}(k+1)$.*
4. *Determine the model which has the best model quality via:*

$$i_{sel} = \arg\min_i \hat{\mathcal{V}}^{[i]}(k+1). \tag{5.36}$$

5. *Set*

$$k = k+1 \tag{5.37}$$
$$\hat{x}^{[i]}(k|k) = \hat{x}^{[i_{sel}]}(k|k) \quad \forall \quad i = 1,\ldots,n \tag{5.38}$$
$$P^{[i]}_{\hat{x}(k|k)} = P^{[i_{sel}]}_{\hat{x}(k|k)} \quad \forall \quad i = 1,\ldots,n \tag{5.39}$$

and go to step 2.

5.3 Analysis of the selection procedure

5.3.1 Introduction

Most of the model selection algorithms that are currently in use such as CUSUM, SPRT and Bayesian or likelihood based model have their origin in the field of statistics.

As a result, the performance of these selection algorithms is often expressed in statistic terms such as the probability of choosing the optimal model.

In contrast, the chosen selection criterion $\mathcal{V}(k)$ (5.18) is a deterministic quantity. As a result, it has no statistical properties. In practice, however, the criterion (5.18) is never used, because it cannot be evaluated using finite amounts of data. In the practical model selection Procedure 5.3 we thus used the approximate criterion $\hat{\mathcal{V}}(k)$ defined in (5.28). This approximation is dependent on measurements $y(k)$ that are influenced by stochastic terms $w(k)$ and $v(k)$. As a result the outcome of the selection Procedure 5.3 itself is also a stochastic variable. Since the outcome of the selection procedure is a stochastic variable we can describe its properties in terms of the probabilities of choosing either the right or wrong model.

In this section we will analyze the selection Procedure 5.3. We will express the performance of the selection procedure in terms of the probability of selecting either the right or wrong model. Also we will compute the expected quality of the state estimate obtained using model selection. For reasons of complexity we will limit our analysis to the specific case in which the selection algorithm has to decide between two linear models denoted by $\mathcal{M}_1$ and $\mathcal{M}_2$. These models have system matrices $A^{[m]}, B^{[m]}, C^{[m]}, Q^{[m]}$ and $R^{[m]}$ for $m \in \{1,2\}$.

We limit the analysis to linear models in order to ensure that the mathematics involved in the analysis remains tractable. Even for linear models the resulting expressions to compute the statistical properties of the selection procedure are long and not trivial to evaluate. Using nonlinear models would quickly result in expressions which can no longer be evaluated analytically.

By limiting the analysis to linear model structures we can only use the results of the analysis to compute the statistical properties of selecting between linear model structures. For linear models it thus allows us to compare the proposed model selection procedure with existing selection techniques. The results of the analysis cannot be used to compute the properties of the selection algorithm when nonlinear models are used.

The probability of selecting either $\mathcal{M}_1$ and $\mathcal{M}_2$ will in general depend on previous selection results. To eliminate this complication in the analysis we will use the following assumption:

Assumption 5.4 *For time samples* $0 \ldots (k-1)$ *the true system has been equivalent to model* $\mathcal{M}_1$*, and the state vector has been estimated using model* $\mathcal{M}_1$.

The previous assumption implies that the estimated state at time $k-1$, $\hat{x}(k-1|k-1)$ is unbiased and the estimation error covariance is indeed $P_{x(k-1|k-1)}$.

5.3.2 False alarm and detection probabilities

If we could exactly compute $\mathcal{V}_1(k)$ and $\mathcal{V}_2(k)$, the optimal state estimate would always be selected. However since the selection is performed using estimates of these quantities, a wrong model could be selected. Given the assumption 5.4, there are only two possibilities to select the wrong model:

- Situation S_1: Model $\mathcal{M}_2$ is selected while $\mathcal{M}_1$ still describes the true system.
- Situation S_2: Model $\mathcal{M}_1$ is selected while the true system has switched to model $\mathcal{M}_2$.

In the field of FDI, $Pr(S_1)$, the probability that situation 1 occurs, i.e. $\mathcal{M}_2$ is selected while $\mathcal{M}_1$ still describes the true system, is called the false alarm probability. The probability $1-Pr(S_2)$, with $Pr(S_2)$ the probability that situation 2 occurs, i.e. $\mathcal{M}_2$ is selected when the true system has indeed switched to $\mathcal{M}_2$, is called the detection probability.

In this section we will show how $Pr(S_1)$ can be computed. The detection probability $1-Pr(S_2)$ can be computed using a similar calculation.

The probability $Pr(S_1)$ is equivalent with the probability that

$$\hat{\mathcal{V}}_2(k) < \hat{\mathcal{V}}_1(k) \tag{5.40}$$

while $\mathcal{M}_1$ still describes the true system. Using (5.33) this can be rewritten as:

$$\begin{aligned} &V_1(\hat{x}^{[1]}(k|k),k) + 2\text{tr}\left(C^{[1]^T}R^{[1]^{-T}}C^{[1]}\left[C^{[1]}R^{[1]^{-1}}C^{[1]^T} + P^{[1]}_{\hat{x}(k|k-1)}{}^{-1}\right]^{-1}\right) > \\ &V_2(\hat{x}^{[2]}(k|k),k) + 2\text{tr}\left(C^{[2]^T}R^{[2]^{-1}}R^{[1]}R^{[2]^{-T}}C^{[2]}\left[C^{[2]}R^{[2]^{-1}}C^{[2]^T} + P^{[2]}_{\hat{x}(k|k-1)}{}^{-1}\right]^{-1}\right) \end{aligned} \tag{5.41}$$

with $V_1(\hat{x}^{[1]}(k|k),k)$ and $V_2(\hat{x}^{[2]}(k|k),k)$ the achieved fits of the estimated states to the measurement data using model $\mathcal{M}_1$ and model $\mathcal{M}_2$ respectively, $P^{[1]}_{k|k-1}$ and $P^{[2]}_{k|k-1}$ are error covariance matrices of the predicted state using Kalman filters based on both models. To obtain (5.41) from (5.33) we used $L = R^{[1]}$ since $\mathcal{M}_1$ is assumed to perfectly describe the true system.

As can be seen in this last equation the $V_i(\hat{x}^{[i]}(k|k),k)$ terms are the only data dependent terms. This means that the condition for a false alarm can be abbreviated to

$$V_1(\hat{x}^{[1]}(k|k),k) - V_2(\hat{x}^{[2]}(k|k),k) > \mathcal{P} \tag{5.42}$$

with $\mathcal{P}$ a constant that is only dependent on the estimation models, i.e. $\mathcal{P}$ is defined as:

$$\begin{aligned} \mathcal{P} = 2\text{tr}&\left(C^{[2]^T}R^{[2]^{-1}}R^{[1]}R^{[2]^{-T}}C^{[2]}\left[C^{[2]}R^{[2]^{-1}}C^{[2]^T} + P^{[2]}_{\hat{x}(k|k-1)}{}^{-1}\right]^{-1}\right) - \\ &2\text{tr}\left(C^{[1]^T}R^{[1]^{-T}}C^{[1]}\left[C^{[1]}R^{[1]^{-1}}C^{[1]^T} + P^{[1]}_{\hat{x}(k|k-1)}{}^{-1}\right]^{-1}\right). \end{aligned} \tag{5.43}$$

Behavior of the achieved fits $V_i(\hat{x}^{[i]}(k|k),k)$

To understand when condition (5.42) is met, it is necessary to understand how phenomena such as model inaccuracies, errors in previous estimates and stochastic effects

in the measurement data influence the left hand side of (5.42). Specifically, we want to compute the probability density function of the left hand side of (5.42). In order to compute this distribution function, we first need to know how model errors and stochastic effects influence each of the achieved fits $V_i(\cdot)$, $i = \{1,2\}$. This behavior of the achieved fit is computed in the following proposition:

Proposition 5.5 *Suppose that at time $k-1$ we have an unbiased estimate $\hat{x}(k-1|k-1)$ of the true state $x(k-1)$, with known error covariance matrix $P_{\hat{x}(k-1|k-1)}$. Assume that at at next time step k the true system is described by the model (A,B,C,Q,R), but the estimate $\hat{x}(k|k)$ of the true state vector $x(k)$ is achieved using a model $\mathcal{M} = (A^{[m]},B^{[m]},C^{[m]},Q^{[m]},R^{[m]})$ which can be different from the "true" model. Then, the fit $V(\hat{x}(k|k),k)$ at time k (see (5.27)) can be expressed as follows:*

$$\begin{aligned} V(\hat{x}(k|k),k) &= (\gamma+\Psi e)^T {R^{[m]}}^{-1}(\gamma+\Psi e) && (5.44)\\ &= \|\gamma+\Psi e\|^2_{R^{[m]-1}}, && (5.45)\end{aligned}$$

with

- *e a zero mean Gaussian noise vector which has a covariance I and which is independent of the choice of the model $\mathcal{M}$,*
- $\Psi \in \mathbb{R}^{n_y\times(2n_x+n_y)}$ *given by*

$$\Psi = (I-C^{[m]}K^{[m]})\begin{bmatrix} C^{[m]}A^{[m]}P^{\frac{1}{2}}_{\hat{x}(k-1|k-1)} & CQ^{\frac{1}{2}} & R^{\frac{1}{2}} \end{bmatrix}, \quad (5.46)$$

- $\gamma \in \mathbb{R}^{n_y\times 1}$ *defined as*

$$\begin{aligned}\gamma = (I-C^{[m]}K^{[m]})\Big[(C-C^{[m]})(Ax(k-1)+Bu(k-1))+C^{[m]}\Big((A-A^{[m]})x(k-1)\\ +(B-B^{[m]})u_{k-1}\Big)\Big] \quad (5.47)\end{aligned}$$

with $K^{[m]}$ the Kalman gain computed using the estimation model $\mathcal{M}$.

Proof The proof can be found in the Appendix. ∎

Note that if the "true" model is used to estimate state of the system, i.e. $A = A^{[m]}$, $B = B^{[m]}$, $C = C^{[m]}$, the estimate $\hat{x}(k|k)$ is unbiased and thus $\gamma = 0$. This result is a consequence of (5.47).

False alarm condition

Let us now return to the determination of the probability of selecting a wrong model when using the model selection procedure of Section 5.2. In the context of Assumption 5.4, this probability is equivalent to the probability that (5.42) occurs when $\mathcal{M}_1$

still represents the true system at time k. Using Proposition 5.5 in this context, we have that:

$$V_1(\hat{x}^{[1]}(k|k),k) = \|\Psi_1 e\|^2_{R^{[1]-1}} \text{ and } V_2(\hat{x}^{[2]}(k|k),k) = \|\gamma_2 + \Psi_2 e\|^2_{R^{[2]-1}} \tag{5.48}$$

with γ_i and Ψ_i ($i = 1,2$) equal to the γ and Ψ of Proposition 5.5 with $(A^{[m]},B^{[m]},C^{[m]},Q^{[m]},R^{[m]}) = (A^{[i]},B^{[i]},C^{[i]},Q^{[i]},R^{[i]})$ and $(A,B,C,Q,R) = (A^{[1]},B^{[1]},C^{[1]},Q^{[1]},R^{[1]})$. Note that $\gamma_1 = 0$ since, at time k, we assume $\mathcal{M}_1$ to be equivalent to the true system. Note also that the noise e is the same in the expressions of $V_1(\hat{x}^{[1]}(k|k),k)$ and $V_2(\hat{x}^{[2]}(k|k),k)$ since e is independent of the choice of the model. Finally, note that the unknown x_{k-1} in the expression of γ_2 will be approximated by its estimate.

Under the condition that $\mathcal{M}_1$ still describes the true system at time k, (5.42) can thus be rewritten as:

$$\|\Psi_1 e\|^2_{R^{[1]-1}} - \|\gamma_2 - \Psi_2 e\|^2_{R^{[2]-1}} > \mathcal{P}. \tag{5.49}$$

Consequently, the probability that (5.42) holds when $\mathcal{M}_1$ still describes the true system at time k is equal to the probability that (5.49) holds. To compute this probability, the exact distribution function of the left hand side of (5.49) is required. In the general case, there is no standard distribution which describes the left hand side of (5.49). Only in the specific case that estimates of model $\mathcal{M}_2$ are unbiased, which means $\gamma_2 = 0$, the exact distribution can be found. An example of a situation in which estimates of model $\mathcal{M}_2$ are unbiased is when model $\mathcal{M}_2$ is an overparametrized version of model $\mathcal{M}_1$ meaning for instance that it models system parameters that are actually constants as variable states using a random walk model. This example was also considered in the introduction. We shall first compute the exact false alarm probability in the specific case that $\gamma_2 = 0$ and then give approximative results for the more general case where $\gamma_2 \neq 0$.

False alarm probability if $\gamma_2 = 0$ ($\mathcal{M}_2$ unbiased)

So first assume that $\gamma_2 = 0$. In this specific case, (5.49) becomes:

$$e^T \mathcal{Q} e > \mathcal{P} \tag{5.50}$$

with

$$\mathcal{Q} = \Psi_1^T {R^{[1]}}^{-1} \Psi_1 - \Psi_2^T {R^{[2]}}^{-1} \Psi_2. \tag{5.51}$$

The left hand side of (5.50) is the weighted 2-norm of a Gaussian stochastic variable. Define $p_{e^T\mathcal{Q}e}(\xi)d\xi$ as the probability that $e^T\mathcal{Q}e \in (\xi, \xi + d\xi)$. In the special case in which $\mathcal{Q} = I$ the probability density function $p_{e^T\mathcal{Q}e}(\xi)$ is of course equivalent with the χ^2 distribution. For arbitrary $\mathcal{Q}$ the distribution function $p_{e^T\mathcal{Q}e}(\xi)$ can be computed using its moment generating function. The moment generating function $\phi(t)$ that describes the distribution $p_{e^T\mathcal{Q}e}(\xi)$ is [88][sect. 15.15]:

$$\phi(t) = \prod_p (1 - 2jt\lambda_p)^{-\frac{1}{2}}, \tag{5.52}$$

with λ_p the p^{th} eigenvalue of $\mathcal{Q}$ and $j^2 = -1$. The probability density function of $p_{e^T \mathcal{Q} e}(\xi)$ can now be found by taking the inverse Fourier transform of $\phi(t)$:

$$p_{e^T \mathcal{Q} e}(\xi) = \int_{-\infty}^{\infty} \phi(t) e^{j\xi t} dt. \tag{5.53}$$

With this probability density function the probability that model $\mathcal{M}_2$ is selected, which shall be denoted by P_2, can be computed via:

$$Pr(S_1) = \int_{\mathcal{P}}^{\infty} p_{e^T \mathcal{Q} e}(\xi) d\xi. \tag{5.54}$$

False alarm probability if $\gamma_2 \neq 0$ ($\mathcal{M}_2$ biased)

Returning to the more general case, in which $\gamma_2 \neq 0$, then the exact probability density function of $V_1(\cdot) - V_2(\cdot)$ cannot be computed exactly. As a result we cannot directly compute the expected false alarm rate using (5.54). Since we cannot compute the exact distribution function, we will construct an approximate distribution. Even though it may not be possible to construct the full true distribution function of $V_1(\cdot) - V_2(\cdot)$, we can compute its first two moments, i.e. its mean and variance. The first two moments can be computed using the following proposition:

Proposition 5.6 *Let $V_1(\hat{x}^{[1]}(k|k),k)$ and $V_2(\hat{x}^{[2]}(k|k)$ be defined as in (5.48), with Ψ_1, Ψ_2, $R^{[1]}$, $R^{[2]}$ fixed known matrices, γ_2 a known fixed vector and e a Gaussian white noise vector with $e \sim \mathcal{N}(0,I)$. Then the mean and variance of $V_1(\cdot) - V_2(\cdot)$ are given by:*

$$\mathbb{E}\left[V_1(\hat{x}_k^{[1]},k) - V_2(\hat{x}_k^{[2]},k)\right] = -\gamma_2^T {R^{[2]}}^{-1} \gamma_2 + tr(\mathcal{Q}) \tag{5.55}$$

$$var\left[V_1(\hat{x}_k^{[1]},k) - V_2(\hat{x}_k^{[2]},k)\right] = 4\gamma_2^T {R^{[2]}}^{-T} \Psi_2^T \Psi_2 {R^{[2]}}^{-1} \gamma_2 + tr\left(\mathcal{Q}(\mathcal{Q}+\mathcal{Q}^T)\right) \tag{5.56}$$

with $\mathcal{Q}$ defined as in (5.51).

Proof For the proof of this proposition, the reader is referred to section 5.C. ∎

To find an approximate distribution of $V_1(\cdot) - V_2(\cdot)$ first consider the special case in which $\gamma_2 = 0$ and $\mathcal{Q} = I$. As already discussed above, we know that for this particular case the distribution for $V_1(\cdot) - V_2(\cdot)$ is known to be a χ^2 distribution. The exact distribution of $V_1(\cdot) - V_2(\cdot)$ can be easily constructed using a χ^2-distribution with a number of degrees of freedom ν such that its first two moments match (5.55)-(5.56). For other values of γ_2 and $\mathcal{Q}$ it is no longer possible to find a value for ν such that the first two moments of the resulting χ^2-distribution match (5.55)-(5.56). Accordingly simulations show that for $\gamma \neq 0$ and $\mathcal{Q} \neq I$ the χ^2-distribution is a poor approximate for the true distribution of $V_1(\cdot) - V_2(\cdot)$. Although the χ^2-distribution is able to exactly approximate the distribution of $V_1(\cdot) - V_2(\cdot)$ for special choices of γ_2 and $\mathcal{Q}$, we thus

see that it is not generic enough to approximate the distribution for $V_1(\cdot) - V_2(\cdot)$ for arbitrary γ_2 and $\mathcal{Q}$.

The Γ-distribution is a generalized version of the χ^2-distribution. Instead of a single shape parameter ν, the Γ distribution has two shape parameters a and b, which can be used to independently set the mean and variance of the distribution. The extra degree of freedom allows us to the better approximate the distribution the true distribution of $V_1(\cdot) - V_2(\cdot)$. The pdf of the Γ-distribution, denoted by $f_\Gamma(x|a,b)$ is given by:

$$f_\Gamma(x|a,b) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{\frac{x}{b}}, \tag{5.57}$$

with a,b the scalar parameters that can be used to determine the shape of the Γ-distribution. It can be shown that for every χ^2 distribution with ν degrees of freedom, there exists a specific choice for a and b such that the resulting Γ-distribution is equivalent to the χ^2-distribution. The mean and variance of the Γ-distribution, denoted as μ_Γ and σ^2_{Gamma} respectively, can be determined via:

$$\mu_\Gamma = \int_x x f_\Gamma(x|a,b)dx = ab \tag{5.58}$$

$$\sigma^2_\Gamma = \int_x (x-\mu_\Gamma)^2 f_\Gamma(x|a,b)dx = ab^2. \tag{5.59}$$

Simulations show that if we choose a and b such that the first two moments of the Γ-distribution (5.58)-(5.59) match the first two moments of $V_1(\cdot) - V_2(\cdot)$ (see (5.55)-(5.56)), then the resulting Γ-distribution is a good approximation for the true distribution of $V_1(\cdot) - V_2(\cdot)$. In order to match the first two moments of $V_1(\cdot) - V_2(\cdot)$, the shape parameters a and b should be chosen as:

$$a = \frac{-\gamma_2^T R^{[2]^{-1}} \gamma_2 + tr(\mathcal{Q})}{\left\{4\gamma_2^T R^{[2]^{-T}} \Psi_2^T \Psi_2 R^{[2]^{-1}} \gamma_2 + \mathrm{tr}\left(\mathcal{Q}(\mathcal{Q}+\mathcal{Q}^T)\right)\right\}^{\frac{1}{2}}}, \tag{5.60}$$

$$b = \frac{4\gamma_2^T R^{[2]^{-T}} \Psi_2^T \Psi_2 R^{[2]^{-1}} \gamma_2 + \mathrm{tr}\left(\mathcal{Q}(\mathcal{Q}+\mathcal{Q}^T)\right)}{-\gamma_2^T R^{[2]^{-1}} \gamma_2 + tr(\mathcal{Q})}. \tag{5.61}$$

Simulations also show that the approximation is especially good if $\mathbb{E}[V_1(\cdot) - V_2(\cdot)] > 0$. An example of such a simulation result is provided in Figure 5.1. Indeed in this Figure we see that the approximate Γ-distribution obtained by matching the first two moments is an accurate approximation of the experimentally obtained histogram.

Substituting the approximating Γ-distribution for the true distribution in (5.54), we can approximately compute the expected false alarm rate of our selection procedure using:

$$Pr(S_1|\mathcal{M}_1) \approx \int_{\mathcal{P}}^{\infty} f_\Gamma(\xi,a,b)d\xi. \tag{5.62}$$

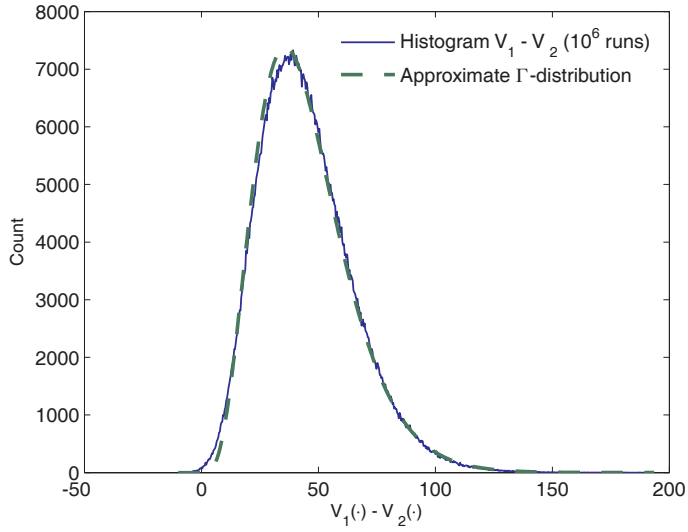

Figure 5.1 Simulation example in which an experimentally obtained histogram of the distribution of $V_1(\cdot) - V_2(\cdot)$ (for $\gamma \neq 0$) is compared with the approximate Γ-distribution whose first two moments are chosen equal to (5.55)-(5.56).

Computing the detection probability

In the previous subsections, we demonstrated how the false alarm probability can be computed. In this section we will show how we can compute the detection probability of the selection procedure.

In order to compute the detection probability in this subsection we will assume that up to the time instant $k-1$ the true model of the process is given by $\mathcal{M}_1$. We also assume that model $\mathcal{M}_1$ has been used to estimate the state of the system. At time k the true process model changes to $\mathcal{M}_2$. The detection probability in this situation corresponds with the probability that the selection procedure chooses to estimate the state of the system using model $\mathcal{M}_2$.

Using Procedure 5.3 it is easily shown that the detection probability corresponds with the probability that

$$\hat{\mathcal{V}}_2(k) < \hat{\mathcal{V}}_1(k) \tag{5.63}$$

while $\mathcal{M}_2$ describes the true system. Using (5.33) the previous expression can be

rewritten as:

$$V_1(\hat{x}^{[1]}(k|k),k) + 2\text{tr}\left(C^{[1]^T}R^{[1]^{-T}}C^{[1]}\left[C^{[1]}R^{[1]^{-1}}C^{[1]^T} + P^{[1]}_{\hat{x}(k|k-1)}\right]^{-1}\right) > V_2(\hat{x}^{[2]}(k|k),k) + 2\text{tr}\left(C^{[2]^T}R^{[2]^{-1}}R^{[1]}R^{[2]^{-T}}C^{[2]}\left[C^{[2]}R^{[2]^{-1}}C^{[2]^T} + P^{[2]}_{\hat{x}(k|k-1)}\right]^{-1}\right) \tag{5.64}$$

or

$$V_1(\hat{x}^{[1]}(k|k),k) - V_2(\hat{x}^{[2]}(k|k),k) > \mathcal{P} \tag{5.65}$$

with $\mathcal{P}$ defined as in (5.43).

Note that the condition for detection (5.65) is exactly the same as the condition for a false alarm (5.42). The only difference between the two conditions is that when computing the false alarm probability, we assume that the true model of the system corresponded to $\mathcal{M}_1$, while in order to compute the detection probability we assume that $\mathcal{M}_2$ is the true model for the system.

Since the condition for a false alarm is similar to the condition for a successful detection, the same expressions that were previously use to compute the false alarm probability can again be used to compute the detection probability, with only two modifications resulting from the fact that $\mathcal{M}_2$ describes true system instead of $\mathcal{M}_1$ in the previous subsections:

1. When computing the false alarm probability we used $\gamma_1 = 0$, because the state estimate of model $\mathcal{M}_1$ was unbiased. Since $\mathcal{M}_1$ no longer corresponds the the true model, we cannot guarantee that it still holds that $\gamma_1 = 0$.

2. While computing the detection probability, we know that the the state estimate using model $\mathcal{M}_2$ is unbiased. As a result we have that $\gamma_2 = 0$.

Apart from these alteration the computation of the false alarm probability is completely similar to the computation of the false alarm probability.

Expected state estimation error after selection

Besides statistical properties such as the false alarm and detection probabilities, we are also interested in the accuracy of the estimated state after model selection. In this section we will assume that the following error measure is used to assess the accuracy of the estimated state:

$$\mathcal{E}(k) = \mathbb{E}\|x(k) - \hat{x}(k)\|^2_W. \tag{5.66}$$

The measure above is the expected quadratic estimation error at time k, weighted using an user defined semi-positive definite weighting matrix W.

If we do not consider model selection for the moment (i.e. we use a fixed model to estimate the state of the system), we can distinguish two situations when estimating the state of a system:

1. the state of the system is estimated using the true model of the system;

2. the state of the system is estimated using a model other than the true system.

In both cases the expected state estimation error can be computed using the following proposition:

Proposition 5.7 *Suppose at time $k-1$ we have an unbiased estimate $\hat{x}(k-1|k-1)$ of the true state $x(k-1)$, with known error covariance matrix $P_{\hat{x}(k-1|k-1)}$. Consider now time k. At time k, the true system is still described by the model (A,B,C,Q,R), but the estimate $\hat{x}^{[m]}(k|k)$ of the true state vector $x(k)$ is determined using a model $\mathcal{M} = (A^{[m]},B^{[m]},C^{[m]},Q^{[m]},R^{[m]})$ which can be different from the "true" model. Then, the estimation error $\mathcal{E}(k)$ can be written as:*

$$\begin{aligned} \mathcal{E}(k) &= E\left[(\kappa+\Phi e)^T W(\kappa+\Phi e)\right] && (5.67)\\ &= \kappa^T W\kappa + tr\left(\Phi^T W\Phi\right), && (5.68)\end{aligned}$$

with $\kappa \in \mathbb{R}^{n_x\times 1}$ defined as:

$$\begin{aligned}\kappa = (I-K^{[m]}C^{[m]})\left[(A-A^{[m]})x(k)+(B-B^{[m]})u(k)\right]\\ -K^{[m]}(C-C^{[m]})(Ax(k)+Bu(k)) \quad (5.69)\end{aligned}$$

and $\Phi \in \mathbb{R}^{n_x\times(2n_x+n_y)}$ defined as

$$\Phi = \left[\begin{array}{ccc} (I-K^{[m]}C^{[m]})A^{[m]}P_{\hat{x}(k-1|k-1)}^{\frac{1}{2}} & (I-K^{[m]}C)Q^{\frac{1}{2}} & K^{[m]}R^{\frac{1}{2}}\end{array}\right]. \quad (5.70)$$

Proof

The proof can be obtained similarly to the proof of proposition 5.5. The proposition follows after substituting (5.B.15) and (5.B.16) into (5.66) and regrouping the deterministic and stochastic terms. ∎

Note that the vector e in (5.67) is still the same Gaussian noise vector as in (5.44).

Note that in the special case in which $A = A^{[m]}$, $B = B^{[m]}$, $C = C^{[m]}$ it can be shown that $\Phi\Phi^T = P_{\hat{x}(k|k)}$, the error covariance matrix of the Kalman estimate at time k (see (2.36)).

Proposition 5.7 gives an expression for $\mathcal{E}(k)$ when a fixed model used to estimate the state of the system. Now consider the situation where two models are available and the Procedure 5.3 detailed in section 5.2 is again used to determine which model will generate the estimated state. Like in the previous section we will again assume that model $\mathcal{M}_1$ perfectly describes the true model, while model $\mathcal{M}_2$ contains some modeling errors. In this case, the expected error $\mathcal{E}(k)$ needs to be computed differently. First define $\mathcal{D}_1$ as all realizations of the noise vector e which causes condition (5.49) to be met (meaning the the correct model is selected), and $\mathcal{D}_2$ as all the realizations of

e for which (5.49) is violated (meaning the wrong model is selected):

$$\mathcal{D}_1 = \{e : e^T \Psi_1^T {R^{[1]}}^{-1} \Psi_1 e - (\gamma_2 - \Psi_2 e) {R^{[2]}}^{-1} (\gamma_2 - \Psi_2 e) < \mathcal{P}\} \quad (5.71)$$

$$\mathcal{D}_2 = \{e : e^T \Psi_1^T {R^{[1]}}^{-1} \Psi_1 e - (\gamma_2 - \Psi_2 e) {R^{[2]}}^{-1} (\gamma_2 - \Psi_2 e) > \mathcal{P}\}. \quad (5.72)$$

Since e is just a Gaussian distributed noise vector, with known mean and variance, its probability density function (pdf) is completely known. Denote the pdf of e as $p_e(e)$. Using these definitions the expected estimation error of the state can be expressed as:

$$\mathcal{E}(k) = \int_{e \in \mathcal{D}_1} (\kappa_1 + \Phi_1 e)^T W (\kappa_1 + \Phi_1 e) p_e(e) de + \int_{e \in \mathcal{D}_2} (\kappa_2 + \Phi_2 e)^T W (\kappa_2 + \Phi_2 e) p_e(e) de. \quad (5.73)$$

Unfortunately, to the knowledge of the author there is no analytical solution to this last expression, meaning that one has to resort to numerical methods for computing the expected estimation error. If the dimension of the state vector x or the measurement vector y is large then the usual gridding methods cannot be used. In those cases randomized algorithms such as [89] may be a viable alternative.

If the user is only interested in a rough approximation of the expected estimation error, the following approximation can also be used:

$$\mathcal{E}(k) \approx (1 - Pr(S_1|\mathcal{M}_1))\mathcal{E}_1(k) + Pr(S_1|\mathcal{M}_1)\mathcal{E}_2(k), \quad (5.74)$$

with $Pr(S_1)$ the probability that model $\mathcal{M}_2$ is selected as calculated with either (5.54) or (5.62) and $\mathcal{E}_i$ the expected estimation error if only model $\mathcal{M}_i$ had been used ($i = \{1,2\}$), as computed in (5.67).

5.4 Tuning the selection algorithm

In the previous section we analyzed the properties of the model selection using Procedure 5.3. The result of the previous section allows us, for specific cases to compute the properties of the selection algorithm. Unfortunately, Procedure 5.3 provides no method of influencing the results as there are no tuning parameters in the selection procedure. In this section we will present two modifications to the filter model selection procedure that allow the user to influence the properties of the selection process.

Our first modification regards the second term in (5.28), which we will call the penalty term. This penalty term is a generalized version of the penalty term $\lambda_0 \frac{2d_{\mathcal{M}}}{N}$ in (5.23). In time series analysis it has been shown that using this penalty the criterion to select models using a finite amount of measurement data y generally causes a model to be selected for which the number of parameters is too high. This is especially true if the number of data N is low [15][16]. Many solutions for this problem have been suggested. In [15][16] the suggested solution to this problem is to use a modified

selection algorithm with altered penalty term:

$$\overline{J}_p(\mathcal{M},\alpha) \approx \frac{1}{N}\sum_{k=1}^{N}\left[\|y(k)-\hat{y}(k,\hat{\theta})\|^2\right] + \alpha\lambda_0\frac{2d_{\mathcal{M}}}{N}. \tag{5.75}$$

The extra parameter α can be used to influence the selection results using the modified FPE criterion. In the field of time series analysis good results have been reported using $\alpha = 1.5$.

Because our filter model quality criterion $\hat{\mathcal{V}}(k)$ is very similar to the FPE criterion it is likely that it also tends to select models with too many degrees of freedom. In the example discussed in the introduction, this means that the model with the extra random walk parameter would be chosen more often than necessary. A solution can be easily constructed by also including the factor α. The corrected criterion thus becomes:

$$\hat{\mathcal{V}}(k,\alpha) = V(\hat{x}(k),k) + 4\alpha\mathrm{tr}\left\{\left[(\psi^T R^{-1} L R^{-T}\psi)\right]\left[\overline{V}''(x^*(k),k) + 2P^{-1}\right]^{-1}\right\}. \tag{5.76}$$

In terms of our analysis in the previous section, using larger values for α will increase $\mathcal{P}$ in (5.43). In practice this means that the model selection procedure is less likely to select models that have a high number of free variables.

Although our chosen filter model quality measure is a deterministic quantity, both the approximation (5.28) and the corrected version (5.76) are stochastic variables, because they are both dependent of the data $y(k)$ via $V(\hat{x}(k),k)$. Consequently the results of the selection procedure are dependent on the realization of measurements $y(k)$. With our second modification to the selection procedure we aim to reduce variability of the estimated model quality $\hat{\mathcal{V}}(\cdot)$. Instead of using the estimated model quality at one time instant only, it is possible to use the average model quality of the last M time instances for model selection. By using an averaged model quality the probability of false alarms and missed detection should decrease.

After both modifications, Procedure 5.3 becomes:

Procedure 5.8 *Assume that we want to determine the best filter model among n available models denoted $\mathcal{M}_1,\ldots,\mathcal{M}_n$. Denote the state estimates and associated covariances obtained using each of these models at time k as $\hat{x}^{[i]}(k|k)$ and $P^{[i]}_{\hat{x}(k|k)}$ with $i = 1,\ldots,n$. Then the selection procedure is*

1. *At some initial time k_0 initialize all filters such that both $\hat{x}^{[i]}(k_0|k_0)$ and $P^{[i]}_{\hat{x}(k_0|k_0)}$ are the same for all $i = 1,\ldots,n$.*
2. *Compute state estimates and estimation covariances $\hat{x}^{[i]}(k|k)$ and $P_{\hat{x}(k|k)^{[i]}}$ for all models for $k = k_0+1,\ldots,k_0+M$.*
3. *Compute the estimated model quality $\mathcal{V}(k)$ for $k = k_0+1,\ldots,k_0+M$ for all filter models using (5.76). Denote the approximate model quality at time k for the i-th model as $\hat{\mathcal{V}}^{[i]}(k)$.*

4. *Determine the model which on average had the best model quality via:*

$$i_{sel} = \arg\min_i \frac{1}{M} \sum_{k=k_0+1}^{k=k_0+M} \hat{V}^{[i]}(k). \tag{5.77}$$

5. *Set*

$$\begin{aligned} k_0 &= k_0 + M && (5.78)\\ \hat{x}^{[i]}(k_0|k_0) &= \hat{x}^{[i_{sel}]}(k_0|k_0) \quad \forall \quad i = 1,\ldots,n && (5.79)\\ P^{[i]}_{\hat{x}(k_0|k_0)} &= P^{[i_{sel}]}_{\hat{x}(k_0|k_0)} \quad \forall \quad i = 1,\ldots,n && (5.80) \end{aligned}$$

and go to step 2.

In the new procedure the available tuning variables are α and M.

5.5 Simulation example

5.5.1 Example 1: Low order toy-model

To illustrate how the selection procedure can be used in practice, two simulation examples are provided. In the first example we will estimate the state of a simple time varying model of low order. Later, in the second example model selection will be used to estimate the temperature distribution of the heated plate example, that was introduced in chapter 3.

In the first simulation example we will use model selection to estimate states of a time varying system. Consider the following simple system:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} \theta_k & 0.7 \\ 0 & 0.9 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} u(k) + w(k) && (5.81)\\ y(k) &= [1\ 1]x(k) + v(k). && (5.82) \end{aligned}$$

The input signal $u(k)$ is chosen as a random binary signal with a switching probability of 0.05, the noises $w(k)$ and $v(k)$ were chosen as Gaussian with:

$$\begin{aligned} \mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} &= 0 && (5.83)\\ \mathbb{E}\begin{bmatrix} w(k) \\ v(k) \end{bmatrix} [w(k)^T v(k)^T] &= \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} && (5.84)\\ Q &= I && (5.85)\\ R &= 10. && (5.86) \end{aligned}$$

If the parameter $\theta(k)$ is constant, these system equations describe a linear system for which the standard Kalman filter is the optimal state estimator. In this example

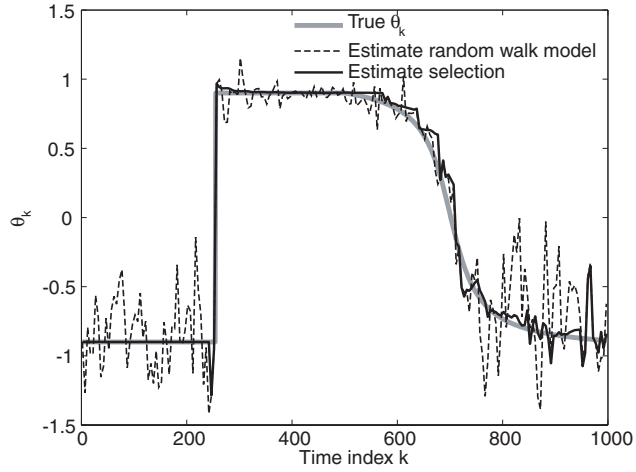

Figure 5.2 True and estimated parameter $\theta(k)$ in the simulation example. The estimates for $\theta(k)$ have been produced using a random walk model and using the model selection procedure described in Procedure 5.8.

however the parameter $\theta(k)$ is chosen to be time varying. The parameter $\theta(k)$ jumps from -0.90 to 0.90 at $k = 255$ and starting from $k = 500$, the parameter gradually returns to its original value, see Figure 5.2. This way, both a sudden jump (fault) and slowly changing behavior are modelled.

Using this system $N = 1000$ measurements $y(k)$ are generated. The true states $x(k)$ and parameters $\theta(k)$ are also stored. Afterwards, using only the recorded measurements and inputs, the state of the system is estimated using the procedure based upon model selection as presented in the previous section.

For the model selection procedure two candidate models of the system are available. The choice of the candidate models has been chosen to correspond with the selection scenario described in the introduction. This means that the first available model is equal to the true model (5.81)-(5.82), but uses constant parameters, i.e.:

$$\theta(k+1) = \theta(k). \tag{5.87}$$

The initial value for the constant parameter is -0.90, which is equal to the true starting value of $\theta(k)$. In the second model the value of the parameter is allowed to vary according to a random walk, i.e. the model (5.81)-(5.82) augmented by:

$$\theta(k+1) \quad = \quad \theta(k) + w^{\theta}(k) \tag{5.88}$$

with $w^{\theta}(k)$ chosen as a Gaussian white noise with variance 0.1.

In this example, the optimal model for state estimation will be selected using the model selection procedure of section 5.4. Two UKF filters are designed using the

Table 5.1 Mean squared errors of estimated states for a time varying system. States were estimated using a fixed parameter model, a random walk parameter model, and using model selection. Results are averaged over 200 simulation runs.

Fixed parameter	Flexible par. filter	Selection filter
$1.1 \cdot 10^4$	15.1	11.8

given models. At time instant $k = k_0$, both filters are initialized with the same estimate $\tilde{x}_{k_0|k_0}$ and $\tilde{P}_{k_0|k_0}$. From that point on the model selection Procedure 5.8 will be used to estimate the state of the system, using $M = 15$ and $\alpha = 1$.

For illustration purposes, we have also estimated the state vector using only the fixed parameter model and using only the flexible parameter model.

In order to compare the accuracy of each filter we compute the average estimation error $\mathcal{E}_{est}$ defined as:

$$\mathcal{E}_{est} = \frac{1}{N} \sum_{k=1}^{N} \|x(k) - \hat{x}(k|k)\|^2. \tag{5.89}$$

The results of all the simulations are given in Table 5.1. As expected, the fixed parameter model alone produces very poor results, due to the bias caused by using an inaccurate model. The results obtained using the flexible parameter model alone are much better. The extra degree of freedom in this model allows the filter to correct the parameter in the measurement update of the filtering procedure. The results using both models and our new model selection procedure are, as expected, much better than the results using the fixed model alone and better than the results of using the flexible parameter model alone. By looking at Figure 5.2, we indeed see that the estimate using our new procedure is made using the fixed parameter model in the time periods where the true parameter remains constant, so that there is no extra variance in the estimated state.

5.5.2 Example 2: Heated plate example

For this second simulation example we consider a more complex process. In this simulation example we use a slightly modified version of the heated plate model of section 3.4.1, altering the model in the following ways:

- In section 3.4.1 we assumed that the heat conductivity coefficient λ (see (3.48)) is temperature dependent. As a result the heat conductivity at time k could be different for different regions of the plate, depending on the temperature of each region. In this example we assume that at each time instant k the heat conductivity $\lambda(k)$ is constant for the entire plate, but this global heat conductivity can change over time.

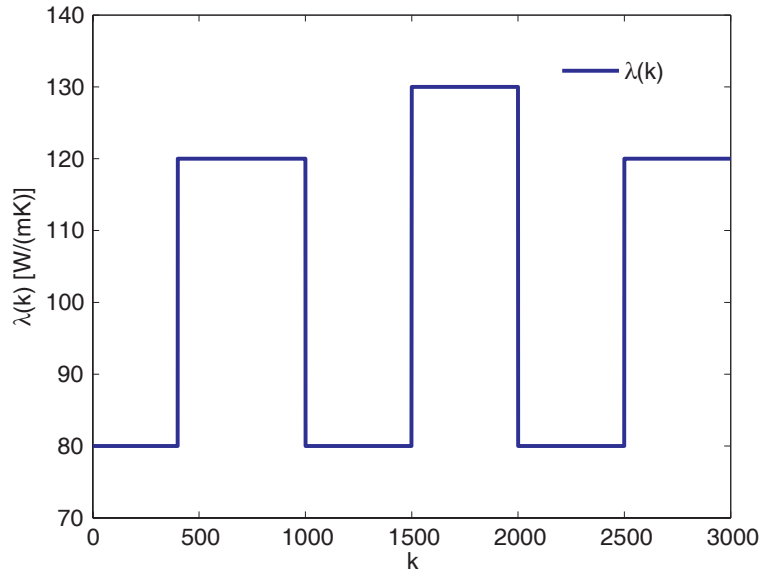

Figure 5.3 Plot of the chosen time-dependent heat conductivity coefficient $\lambda(k)$ that was used in the simulation example to generate data.

For this particular example we have chosen a heat conductivity $\lambda(k)$ that is piecewise constant. Its exact behavior is depicted in Figure 5.3. Note that a heat conductivity will never change in such a manner in reality.

- We have added an additive Gaussian process noise $w(k)$ to the system. The variance matrix Q of the process noise matrix is assumed known: $Q = I$.
- It is assumed that the temperature of the plate is measured at 5 positions. The measurement locations are indicated in Figure 5.4. The measurements subject to a zero mean Gaussian measurement noise $v(k)$. The covariance matrix R of the measurement noise is assumed known, $R = 0.01 \cdot I$.

After the modifications the model can be written in the following form:

$$x(k+1) = f(x(k),u(k),\lambda(k)) + w(k) \tag{5.90}$$
$$y(k) = h(x(k)) + v(k). \tag{5.91}$$

If the time-dependent heat conductivity coefficient is assumed known, the heated plate model (5.90)-(5.91) reduces to a time varying state-space model of the form (2.23)-(2.24). As was discussed in chapter 2, for such models the optimal state estimator is the Kalman filter.

In this example however, we will assume that the time dependent heat-transfer coefficient $\lambda(k)$ is not known. Instead, the heat-transfer coefficient will be estimated

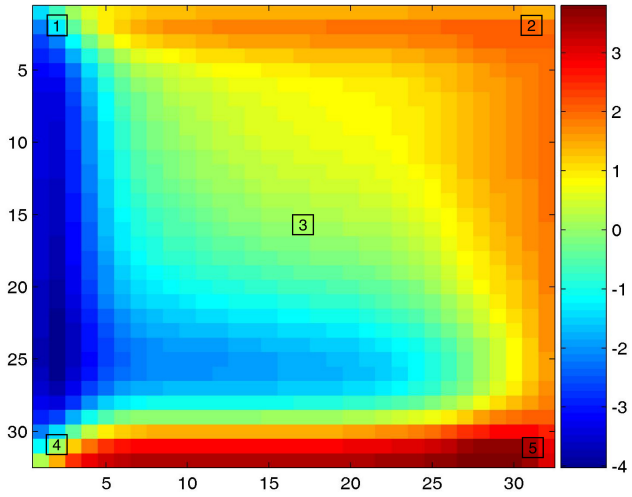

Figure 5.4 Realization of the heated plate. At every time instant the temperature is measure at five locations. These locations are indicated by the numbered squares.

online. To this end we will create two extended models of the form (5.5). Again these models are chosen as first described in section 5.1: in the first extended model the heat conductivity coefficient $\lambda(k)$ is modeled as a constant:

$$\lambda(k+1) = \lambda(k). \tag{5.92}$$

In the second model we model the heat conductivity coefficient with a random walk:

$$\lambda(k+1) = \lambda(k) + w_\lambda(k), \tag{5.93}$$

with $w_\lambda(k)$ a zero mean Gaussian white noise with covariance $\mathbb{E}(w_\lambda(k)w_\lambda(k)^T) = 25$).

Using these two models for $\lambda(k)$ we shall construct three state estimation filters:

1. A filter in which the heat-transfer coefficient is assumed constant at $\lambda(k) = 80$. Using this assumption the plate model (5.90)-(5.91) reduces to a linear model. As a result, the linear Kalman filter is used to estimate the state of the system.

2. A filter in which the heat-transfer coefficient is modelled as a random walk as given in (5.93). Using this assumption the filter model is nonlinear. The Unscented Kalman filter is used to estimate the state of the system.

3. The state of the system is estimated using online filter model selection. Procedure 5.8 is used to determine online which model to determine the best model

Table 5.2 Averaged mean squared errors (see (5.94)) of estimated states for the time varying heated plate model. The states in this example represent temperature distribution of the heated plate. The different filter models used are described in section 5.5.2.

$\lambda(k)$ constant	$\lambda(k)$ random walk	Selection	$\lambda(k)$ assumed known
312.2	130.0	128.5	125.9

for state estimation. The available models are the extended models in which the heat conductivity coefficient is either assumed fixed (see (5.92) or a random walk (5.93). The Unscented Kalman filter is again used to estimate the state of both candidate models. For the selection procedure we used the tuning parameters $M = 20$ and $\alpha = 1.5$.

For reference we shall also construct a state filter in which $\lambda(k)$ is known. Since $\lambda(k)$ is now a known quantity, the model (5.90)-(5.91) reduces to a time varying linear state-space model of the form (2.23)-(2.24). Since the model equation are linear, the state is estimated using the Kalman filter.

To quantify and compare the accuracy of each of the filters, we shall use the following quality measure:

$$Err = \frac{1}{N}\sum_{k=1}^{N} \|x(k) - x(k|k)\|^2. \tag{5.94}$$

The accuracy of each of the filters is presented in Table 5.2. In the table we see once again that the results of the first filter, in which $\lambda(k)$ was modelled as a constant, are very poor. This error is caused by the bias due to a erroneous value of $\lambda(k)$.

The filter using the random walk model (5.93) already performs much better. The estimation error using this filter is already quite close to the estimation error of the optimal filter in which $\lambda(k)$ was known. The additional error of the random walk model is caused due to the assumed uncertainty in $\lambda(k)$.

The filter results using the online filter model selection algorithm outline in Procedure 5.8 are a slight improvement over the random walk based model. The reason for the increase in accuracy is most easily explained by considering the estimated values for $\lambda(k)$ as plotted in Figure 5.5. As can be seen in this figure, the model selection procedure selects results from the fixed parameter model (5.92) when $\lambda(k)$ is indeed a constant (thus avoiding the extra variability of the random walk model), but uses the results of the random walk model when the heat conductivity coefficient jumps to a new value. As in the previous example we thus see that the selection procedure is able to combine the advantages of both the available models.

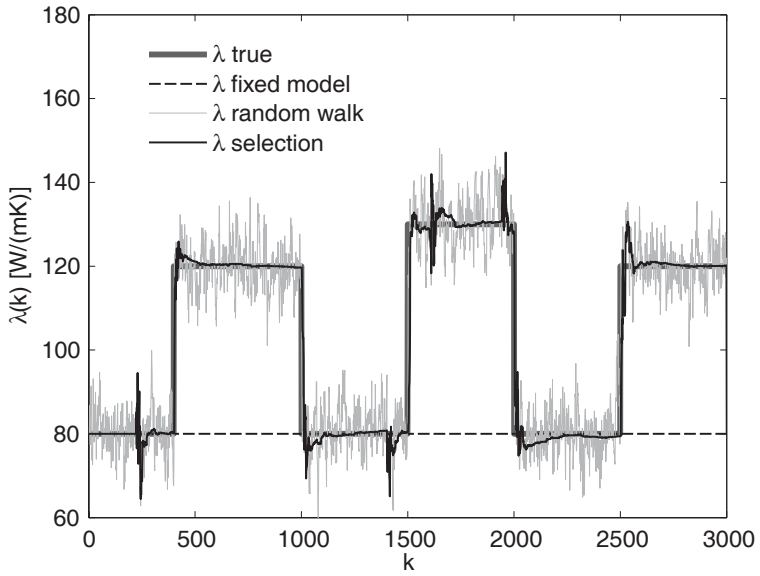

Figure 5.5 True and estimated values for $\lambda(k)$. The estimates were obtained using models in which $\lambda(k)$ was assumed constant (5.92), $\lambda(k)$ modelled as a random walk (5.93) and finally using model selection Procedure 5.8 that had to determine the best alternative from the both previous models.

5.6 Summary and conclusions

In many state estimation applications it is assumed that the available process and measurement model perfectly describes the true process. In this chapter we have considered the possibility that in fact the behavior of the true process might change over time, in particular we assume that the process parameters might change over time. If model parameters could change over time, they would have to be estimated together with the state of the system. Estimating states and parameters simultaneously leads to a increase of the variance of the estimated states. As we have shown in this chapter, the additional variance can be avoided by using accurate model selection criteria which automatically determine using the available input-output data when the model parameters need to be re-estimated.

For this purpose a new model selection criterium has been introduced in section 5.2. The model selection criterium has been constructed by extending the existing model selection criteria used in System Identification literature.

For the case where the available models are linear, an analysis of the statistical properties of the new model selection algorithm has been presented in section 5.3.

Section 5.4 has introduced two tuning parameters in the selection algorithm that can be used to improve the behavior of the selection algorithm.

Finally, the effectiveness of the new selection procedure has been demonstrated in section 5.5. In this section the model selection procedure is used to obtain improved state estimates in both a toy example and for the heated plate model introduced earlier.

5.A Proof of Proposition 5.1

Define $W(x,k)$ as:

$$W(x,k) = V(x,k) + \|x - x(k)^{\#}\|^2_{P^{-1}}, \tag{5.A.1}$$

then

$$\hat{x}(k) = \arg\min_x W(x,k). \tag{5.A.2}$$

Define also $\overline{W}(x,k)$ as:

$$\overline{W}(x,k) = \overline{V}(x,k) + \|x - x(k)^{\#}\|^2_{P^{-1}}. \tag{5.A.3}$$

In order to prove the proposition, let us first expand $\overline{V}(\hat{x}(k),k)$ around $x^*(k) = \arg\min_x \overline{V}(x,k)$:

$$\overline{V}(\hat{x}(k),k) = \overline{V}(x^*(k),k) + \frac{1}{2}(\hat{x}(k) - x^*(k))^T \overline{V}''(\zeta_k,k)(\hat{x}(k) - x^*(k)). \tag{5.A.4}$$

In the equation above, the symbol $'$ means taking the partial derivative with respect to x. Similarly since $W'(\hat{x}(k),k) = 0$:

$$W(x^*(k),k) = W(\hat{x}(k),k) + \frac{1}{2}(\hat{x}(k) - x^*(k))W''(\overline{\zeta}_k,k)(\hat{x}(k) - x^*(k)), \tag{5.A.5}$$

which is easily rewritten into:

$$W(\hat{x}(k),k) = W(x^*(k),k) - \frac{1}{2}(\hat{x}(k) - x^*(k))W''(\overline{\zeta}_k,k)(\hat{x}(k) - x^*(k)). \tag{5.A.6}$$

The second derivative of $W(x,k)$ is:

$$W''(x,k) = V''(x,k) + 2P^{-1}. \tag{5.A.7}$$

Inserting (5.A.3) and (5.A.7) into (5.A.6) and using the assumption that $x_k^{\#} = x^*(k)$, we obtain:

$$V(\hat{x}(k),k) \quad = \quad V(x^*(k),k) - \frac{1}{2}(\hat{x}(k) - x^*(k))(V''(\overline{\zeta}_k,k) + 4P^{-1})(\hat{x}(k) - x^*(k))$$

Take the expectations of (5.A.4) and (5.A.8) and use the following asymptotical relations:

$$\begin{aligned}
&\mathbb{E}_{\hat{x}(k)}(\hat{x}(k) - x^*(k))(\overline{V}''(\zeta_k,k))(\hat{x}(k) - x^*(k)) \\
&\qquad = \mathbb{E}_{\hat{x}(k)} \operatorname{tr}\left\{\overline{V}''(\zeta_k,k)(\hat{x}(k) - x^*(k))(\hat{x}(k) - x^*(k))^T\right\} \\
&\qquad\qquad \approx \operatorname{tr} \overline{V}''(x^*(k),k)P_x
\end{aligned} \tag{5.A.8}$$

in which P_x is the asymptotic covariance matrix of $\hat{x}(k)$. Also,

$$\mathbb{E}_{\hat{x}(k)}(\hat{x}(k) - x^*(k))(V''(\overline{\zeta}_k,k))(\hat{x}(k) - x^*(k)) \quad \approx \quad \operatorname{tr}\overline{V}''(x^*(k),k)P_x \tag{5.A.9}$$

$$\text{and} \quad V(x^*,k) \quad \approx \quad \overline{V}(x^*,k). \tag{5.A.10}$$

Using the last three relations together with (5.A.4) and (5.A.8) gives:

$$\begin{aligned}
\mathbb{E}_{\hat{x}(k)}\overline{V}(\hat{x}(k),k) \quad &\approx \quad \overline{V}(x^*(k),k) + \frac{1}{2}\operatorname{tr}\overline{V}''(x^*(k),k)P_x \\
\mathbb{E}_{\hat{x}(k)}V(\hat{x}(k),k) \quad &\approx \quad \overline{V}(x^*(k),k) - \frac{1}{2}\operatorname{tr}(\overline{V}''(x^*(k),k) + 4P^{-1})P_x.
\end{aligned}$$

Combining the two last expressions yields:

$$\mathbb{E}_{\hat{x}(k)}\overline{V}(\hat{x}(k),k) \approx \mathbb{E}_{\hat{x}(k)}V(\hat{x}(k),k) + \mathrm{tr}(\overline{V}''(x^*(k),k) + 2P^{-1})P_x. \tag{5.A.11}$$

Using the theory of chapter 9, pages 281-282 in [56], it can be shown that the covariance matrix P_x equals:

$$\begin{aligned} P_x &= 4\left[\overline{W}''(x^*(k),k)\right]^{-1}\left[\psi^T RLR^T\psi\right]\left[\overline{W}(x^*(k),k)\right]^{-1} \\ &= 4\left[\overline{V}''(x^*(k),k)+2P^{-1}\right]^{-1}\left[\psi^T RLR^T\psi\right]\left[\overline{V}''(x^*(k),k)+2P^{-1}\right]^{-1} \end{aligned} \tag{5.A.12}$$

with ψ and L as defined in the statement of the theorem. To derive this last expression, we have used the assumption that $y(k)-\hat{y}(x^*(k))$ is approximately a white noise and that $\overline{W}''(x^*(k),k)$ exists and is regular.

If we take result (5.A.11) (after replacing $\mathbb{E}_{\hat{x}(k)}V(\hat{x}(k))$ with $V(\hat{x}(k))$, the only observation we have of it) and combine it with (5.A.12), we obtain the expression (5.28). ∎

5.B Proof of Proposition 5.5

Proof By (5.27), we have:

$$V(\hat{x}(k|k),k) = \|y(k) - C^{[m]}\hat{x}(k|k)\|^2_{R^{[m]^{-1}}}. \tag{5.B.13}$$

Using (5.11) this becomes:

$$V(\hat{x}(k|k),k) = \|Cx(k) + v(k) - C^{[m]}\hat{x}(k|k)\|^2_{R^{[m]^{-1}}}. \tag{5.B.14}$$

Since the Kalman filter was used to determine $\hat{x}(k|k)$, we can write according to (2.34):

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K^{[m]}(y(k) - C^{[m]}\hat{x}(k|k-1)). \tag{5.B.15}$$

Now, using (2.28) and denoting by $\varepsilon = x(k-1) - \hat{x}(k-1|k-1)$ the estimation error at time $k-1$, we have that:

$$\begin{aligned} x(k) - \hat{x}(k|k-1) = (A - A^{[m]})x(k-1) + (B - B^{[m]})u(k-1) \\ + w(k-1) - A^{[m]}\varepsilon. \end{aligned} \tag{5.B.16}$$

Using these last two equations and (2.28), we can rewrite (5.B.14) as follows:

$$\begin{aligned} V(\hat{x}(k|k),k) = \Big\| &(I - C^{[m]}K^{[m]})(C - C^{[m]})(Ax(k-1) + Bu(k-1)) \\ &+ C^{[m]}(I - K^{[m]}C^{[m]})\left((A - A^{[m]})x(k-1) + (B - B^{[m]})u(k-1)\right) \\ &+ (I - C^{[m]}K^{[m]})Cw(k-1) - (I - K^{[m]}C^{[m]})A^{[m]}\varepsilon \\ &+ (I - C^{[m]}K^{[m]})v(k)\Big\|^2_{R^{[m]^{-1}}}. \end{aligned} \tag{5.B.17}$$

In this last equation, the first three terms are all deterministic quantities, while the last three terms are all stochastic variables. Using the definition (5.47) of γ, the deterministic terms can be lumped together:

$$V(\hat{x}(k|k),k) = \|\gamma + (I - C^{[m]}K^{[m]})Cw(k-1) \\ - C^{[m]}(I - K^{[m]}C^{[m]})A^{[m]}\varepsilon + (I - C^{[m]}K^{[m]})v(k)\|^2_{R^{[m]-1}}. \quad (5.B.18)$$

The last three terms describe how the three noise terms $v(k), w(k-1)$ and ε influence the achieved fit of the model. Note that, since $\hat{x}(k-1|k-1)$ has been determined using an exact model, ε is a white noise with covariance $P_{k-1|k-1}$ and is not correlated with $w(k-1)$ and $v(k)$. Therefore, the stochastic terms in the last equation can be rewritten as:

$$(I - C^{[m]}K^{[m]})Cw(k-1) - C^{[m]}(I - K^{[m]}C^{[m]})A^{[m]}\varepsilon + (I - C^{[m]}K^{[m]})v(k) = \Psi e, \quad (5.B.19)$$

with Ψ defined as in (5.46) and e a zero mean Gaussian noise vector which has a covariance I and which is independent of the choice of the model $\mathcal{M} = (A^{[m]}, B^{[m]}, C^{[m]}, Q^{[m]}, R^{[m]})$. The dimension of the vector e is the sum of the lengths of the vectors $v(k), w(k)$ and ε. Substituting this last result in (5.B.18), gives equations (5.44)-(5.45). ∎

5.C Proof of Proposition 5.6

5.C.1 Outline of proof

The proof for proposition 5.6 presented here consists of two steps: first we will present two lemmas in which the expectation and variance of expressions of the form: $e^T We$, with $e \in \mathbb{R}^{n_e \times n_e}$ a normally distributed vector with $e \sim \mathcal{N}(0, \sigma^2 I)$ and $W \in \mathbb{R}^{n_e \times n_e}$ an arbitrary fixed matrix. Then, using the derived properties we shall move to the proof of proposition 5.6.

Lemma 5.9 *Suppose $e \in \mathbb{R}^{n_e \times n_e}$ is a Gaussian distributed vector with $e \sim \mathcal{N}(0, \sigma^2 I)$ and $W \in \mathbb{R}^{n_e \times n_e}$ is an arbitrary fixed matrix. Then it holds that:*

$$\mathbb{E}e^T We = \sigma^2 trace(W) \quad (5.C.20)$$

$$var(e^T We) = \sigma^4 trace(W(W + W^T)). \quad (5.C.21)$$

Proof The expected value of $e^T We$ in (5.C.20) can be easily computed. Since $e^T We$ is a scalar, it holds that:

$$\mathbb{E}e^T We = \text{trace}(\mathbb{E}e^T We). \quad (5.C.22)$$

Using the property of the trace operator that $\text{trace}(AB) = \text{trace}(BA)$ we easily obtain:

$$\text{trace}(\mathbb{E}e^T We) = \text{trace}(\mathbb{E}ee^T W) = \sigma^2 \text{trace}(W), \quad (5.C.23)$$

which proves (5.C.20). The variance expression is somewhat harder to prove. We start the proof of the variance expression by writing:

$$\text{var}(e^T We) = \mathbb{E}\left(e^T We - \mathbb{E}\{e^T We\}\right)^2 \quad (5.C.24)$$

$$= \mathbb{E}\left(e^T Wee^T We - 2e^T We\mathbb{E}\{e^T We\} + [\mathbb{E}\{e^T We\}]^2\right) \quad (5.C.25)$$

$$= \mathbb{E}\left(e^T Wee^T We\right) - \sigma^4 \left(\text{trace}(W)\right)^2. \quad (5.C.26)$$

Consider the first term on the right hand side of (5.C.26). If we denote the element of W on row i and column j as $[W]_{ij}$ and the i-th row of e as e_i we can write:

$$\mathbb{E}\left(e^T W e e^T W e\right) = \sum_{i,j,k,l} \mathbb{E}[W]_{ij}[W]_{kl} e_i e_j e_k e_l. \tag{5.C.27}$$

In which all the summation variables i, j, k, l range from 1 to n_e. Because $e \sim \mathcal{N}(0, \sigma^2 I)$ the terms of this summation are only non-zero in the following four cases:

1. If $i = j$, $k = l$, but $i \neq k$. For this particular case we have:

$$\begin{aligned}
\sum_{i=j,k=l,i\neq k} \mathbb{E}[W]_{ij}[W]_{kl} e_i e_j e_k e_l &= \sum_{i\neq k} \mathbb{E}[W]_{ii}[W]_{kk} e_i e_i e_k e_k \\
&= \sigma^4 \sum_{i\neq k} [W]_{ii}[W]_{kk} \\
&= \sigma^4 \sum_{i,k} [W]_{ii}[W]_{kk} - \sigma^4 \sum_{i=k} [W]_{ii}[W]_{kk} \\
&= \sigma^4 \sum_{i,k} [W]_{ii}[W]_{kk} - \sigma^4 \sum_{i} [W]_{ii}[W]_{ii}.
\end{aligned} \tag{5.C.28}$$

2. If $i = k$, $j = l$, but $i \neq j$. For this particular case we have:

$$\begin{aligned}
\sum_{i=k,j=l,i\neq j} \mathbb{E}[W]_{ij}[W]_{kl} e_i e_j e_k e_l &= \sum_{i\neq j} \mathbb{E}[W]_{ij}[W]_{ij} e_i e_i e_j e_j \\
&= \sigma^4 \sum_{i\neq j} [W]_{ij}[W]_{ij} \\
&= \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ij} - \sigma^4 \sum_{i=j} [W]_{ij}[W]_{ij} \\
&= \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ij} - \sigma^4 \sum_{i} [W]_{ij}[W]_{ij}.
\end{aligned} \tag{5.C.29}$$

3. If $i = l$, $j = k$, but $i \neq j$. For this particular case we have:

$$\begin{aligned}
\sum_{i=l,j=k,i\neq j} \mathbb{E}[W]_{ij}[W]_{kl} e_i e_j e_k e_l &= \sum_{i\neq j} \mathbb{E}[W]_{ij}[W]_{ji} e_i e_i e_j e_j \\
&= \sigma^4 \sum_{i\neq j} [W]_{ij}[W]_{ji} \\
&= \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ji} - \sigma^4 \sum_{i=j} [W]_{ij}[W]_{ji} \\
&= \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ji} - \sigma^4 \sum_{i} [W]_{ij}[W]_{ji}.
\end{aligned} \tag{5.C.30}$$

4. If $i = j = k = l$. For this last situation it holds that:

$$\begin{aligned}
\sum_{i=j=k=l} \mathbb{E}[W]_{ij}[W]_{kl} e_i e_j e_k e_l &= \sum_{i} \mathbb{E}[W]_{ii}[W]_{ii} e_i e_i e_i e_i \\
&= 3\sigma^4 \sum_{i} [W]_{ii}[W]_{ii}.
\end{aligned} \tag{5.C.31}$$

Returning to (5.C.27) we thus have that (5.C.27) = (5.C.28) + (5.C.29) + (5.C.30) + (5.C.31):

$$\begin{aligned}\mathbb{E}(e^T W e e^T W e) &= \sigma^4 \sum_{i,j} [W]_{ii}[W]_{jj} + \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ij} + \sigma^4 \sum_{i,j} [W]_{ij}[W]_{ji}, \\ &= \sigma^4 \left((\text{trace}(W))^2 + \text{trace}(WW^T) + \text{trace}(WW) \right), \end{aligned} \tag{5.C.32}$$

where in the last step we used that

$$\text{trace}(WW) \triangleq \sum_i [WW]_{ii} = \sum_{ij} [W]_{ij}[W]_{ji}, \tag{5.C.33}$$

which can be easily deduced from

$$[WW]_{ii} = \sum_j [W]_{ij}[W]_{ji}, \tag{5.C.34}$$

Finally substituting (5.C.32) in (5.C.26) results in

$$\text{var}(e^T W e) = \sigma^4 \text{trace}(W(W + W^T)). \tag{5.C.35}$$

■

5.C.2 Main proof

Using the results derived in Lemma 5.9, we now prove Proposition 5.6

Proof

$$\begin{aligned}\mathbb{E}\left[V_1(\hat{(}x)(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] &= \mathbb{E}\left(\|\Psi_1 e\|^2_{R^{[1]^{-1}}} - \|\gamma_2 + \Psi_2 e\|^2_{R^{[2]^{-1}}} \right) \\ &= \mathbb{E}\left(-\gamma_2^T R^{[2]^{-1}} \gamma_2 - 2\gamma_2^T R^{[2]^{-1}} \Psi_2 e + e^T \mathcal{Q} e \right) \end{aligned}$$

using $\mathcal{Q}$ as defined in (5.51). Of the three terms of the right hand side of the last expression, the first is a constant, the second has expectation zero and the expectation of the last term can be easily computed using Lemma 5.9. As a result we thus indeed have that:

$$\mathbb{E}\left[V_1(\hat{x}(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] = -\gamma_2^T R^{[2]^{-1}} \gamma_2 + \text{trace}(\mathcal{Q}). \tag{5.C.36}$$

To prove the variance result we first write:

$$\begin{aligned}\text{var}\left[V_1(\hat{x}(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] = \mathbb{E}\Big(&\left[V_1(\hat{x}(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] - \\ &\mathbb{E}\left[V_1(\hat{x}(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] \Big)^2. \end{aligned} \tag{5.C.37}$$

Using (5.48) and (5.67) this becomes:

$$\begin{aligned}&\text{var}\left[V_1(\hat{x}(k|k)^{[2]}, k) - V_2(\hat{x}(k|k)^{[2]}, k) \right] = \\ &\quad \mathbb{E}\left(\gamma_2^T R^{[2]^{-1}} \gamma_2 + 2\gamma_2^T R^{[2]^{-1}} \Psi_2 e + e^T \mathcal{Q} e - \left(-\gamma_2^T R^{[2]^{-1}} \gamma_2 + \text{trace}(\mathcal{Q}) \right) \right)^2 \end{aligned} \tag{5.C.38}$$

$$= \mathbb{E}\left(4e^T\Psi_2^T R^{[2]^{-T}}\gamma_2\gamma_2^T R^{[2]^{-1}}\Psi_2 e\right) + \mathbb{E}\left(e^T\mathcal{Q}e - \text{trace}(Q)\right)^2. \tag{5.C.39}$$

In this last equation we see that the first term is of the form e^TWe. This means that this expectation is easily computed using (5.C.20). The second term in the last equation is equivalent to the variance of $e^T\mathcal{Q}e$. This term is easily computed using (5.C.21). After substituting the results from Lemma 5.9, we indeed have:

$$\text{var}\left[V_1(\hat{x}_k^{[1]},k) - V_2(\hat{x}_k^{[2]},k)\right] = 4\gamma_2^T R^{[2]^{-T}}\Psi_2^T\Psi_2 R^{[2]^{-1}}\gamma_2 + \text{tr}\left(\mathcal{Q}(\mathcal{Q}+\mathcal{Q}^T)\right) \tag{5.C.40}$$

■

Chapter 6

Case study: Dryer section of paper production machine

6.1 Introduction

In this chapter the theory of the preceding chapters will be illustrated using a simulation case study. In this case study we will attempt to monitor the states of the dryer section of a paper mill.

Before continuing with a detailed description of the dryer section of a paper machine, we will first discuss the paper making process in general. Then we will focus on the role of the dryer section in this process.

Most current industrial paper mills are based on the Fourdrinier machine invented in 1798 by Nicolas Louis Robert. A Fourdrinier machine performs all tasks required to transform wood pulp (the source product) into paper. Like the original Fourdrinier machine, current paper machines consist of four main sections: the wet section, the press section, the dryer section and the calender section.

In the wet section of a paper machine water and wood pulp are combined with other source materials such as sizing, fillers, colors, and possibly waste paper called broke. The source materials are converted to fibers and passed on to the refiner where the fibres are subjected to brushing and rubbing. After brushing and rubbing the fibre mixture then enters the headbox that loads the pulp onto a moving wire conveyor. Suction boxes below the wire gently remove water from the pulp with a slight vacuum. The pulp mixture on the conveyor belt that leaves the wet-section on average contains over 90% water.

The next section is the press section. The main purpose of the press section is to mechanically remove water from the paper using sets of rolling presses. A secondary effect of pressing the paper is that it smoothes the surface of the paper. After the press section approximately 50% of the mass of the paper mixture on the conveyor consists of water.

After the press section the paper enters the dryer section. The goal of the dryer section is to lower the water content of the paper mixture on the conveyor to approxi-

mately 5% - 10%. In the dryer section the water fraction is lowered by vaporizing the water from the paper. For this purpose the paper web on the conveyor belt is heated by steam heated cylinders. During the drying process additional agents are often sprayed on the paper to influence the paper properties.

The final section of a paper machine is the calender section. In the calender section the dried paper is wound up on big reels. Wax, starch or other products may be added to the paper in the calender section to obtain the desired finish for the final product.

The dryer section of a paper machine is by far the most energy consuming section of a paper machine. As a result it is a natural target for process optimization. As discussed earlier, this requires that we develop an effective monitoring tool for the dryer section.

To develop a monitoring tool, we will use a first principles model of the dryer section. Different models for a dryer section can be found in the literature, see for instance [10][27][48]. Here, we will use a detailed first principles model developed by TNO Science and Industry [50].

This chapter is organized as follows. First, in Section 6.2 we will discuss the available nonlinear process model for a dryer section and present its main equations. Section 6.3 demonstrates the use of model reduction techniques to reduce the order of the available dryer model. Since model reduction does not decrease the CPU time required per simulation, in section 6.4 an approximate reduced order dryer model will be derived that allows us to speed up simulations, while retaining the physical interpretation of the states. Section 6.5 illustrates how we can use measured data to construct an approximate Kalman filter for a linearized dryer model even when no covariance information is available. In section 6.6 we use the nonlinear model to estimate the state of the dryer section. Once a good state filter has been derived, section 6.7 applies the model selection techniques of chapter 5 to determine online if it is necessary to adjust model parameters. The chapter ends with some conclusions and discussions of the obtained results.

6.2 Modelling of a dryer section

6.2.1 Dryer section description

Before presenting the available dryer section model that will be used in the case study, first the modelled dryer section setup will be discussed. This particular setup was supplied by TNO Science and Industry. The supplied setup is similar to actual dryer configurations used in industrial practice.

In the considered dryer section approximately 55% of the mass that comprises the paper web is water. To remove the water content, the considered dryer section has 54 drying cylinders available. After the 32nd cylinder the modelled setup contains a coater unit which sprays water mixed with coating materials to improve the properties of the produced paper. After passing the coater unit, the paper web is again dried such that the paper web leaving the dryer contains less than 10% water.

In this chapter we will refer to the first 32 cylinders in this particular dryer setup as

the pre-dryer section (or PDR section), the 22 cylinder after the coater will be referred to as the after-dryer section (or ADR section).

All cylinders in the dryer section are heated by pressurized and saturated steam passing through the interior of the (hollow) cylinders. The pressure (and thus the temperature) of the saturated steam can be controlled by an operator. The pressure of the steam entering the cylinders cannot be controlled per individual cylinder. Instead cylinders are divided into six groups. The pressure of the steam heating the cylinders is the same for all cylinders within a cylinder group. The steam pressure for each of the six cylinder groups, however, can be controlled by an operator.

Apart from the pressure for each of the 6 cylinder groups, the operator can also control the speed of the paper web and the amount of coating materials sprayed onto the web at the coater unit.

The main monitoring objective in a paper machine is to obtain an accurate estimate of the dryness of the paper at the entrance of the dryer section. The dryness of the paper entering the dryer section is of critical importance because it determines the maximum stress the paper can endure before breaking. If the paper web is too wet when entering the dryer section, the web will break. If the web breaks, production needs to be halted, to allow operators to clean up the broken web. When the paper web at the entrance of the dryer section is relatively dry, it may be possible to operate the dryer section at a higher velocity. Unfortunately the dryness of the paper at the entrance of the dryer section cannot be easily measured directly. In fact, in the studied plant the dryness of the paper is only measured after the 32nd and 54th cylinder pairs.

Another important quantity to be monitored is the efficiency of the heat transfer between the dryer cylinders and the paper web for both the PDR and ADR sections of the dryer. The heat transfer between the paper web on the conveyor and a dryer cylinder can decrease as a result of dirt accumulating on the dryer cylinder. Once the heat transfer between a cylinder and the paper web becomes unacceptably low, a cylinder needs to be cleaned. The efficiency of the heat transfer between the paper web and the dryer cylinders can be expressed quantitatively using correction factors on the theoretical heat transfer between the dryer cylinders and the paper web.

In order to monitor the dryer section, two measurements are available at each sampling instant. These measurements consist of paper dryness measurements between the 32nd cylinder and the coater unit (just after the PDR section of the dryer) and after the final dryer cylinder of the dryer. The dryness measurements are available every 60 seconds.

6.2.2 Model layout

In this section we will discuss the available model of the dryer section. This model consists of three parts. Before discussing each part of the model in detail we will first present a brief overview of the available sub-models:

1. TNO Dryer model as described in [50]. This dryer model - by far the most complex and detailed part of the total model - has been provided by TNO Science and Industry. This model describes virtually all aspects of the dryer section:

(a) Heat transfer between steam and cylinders,

(b) Heat and mass (water in liquid and vapour form) transfer within the paper web,

(c) Heat and mass transfer from the paper web to its surroundings (air, cylinders, felts),

(d) Shrinking of the paper as it becomes dryer,

(e) Coater unit.

Unfortunately the TNO Dryer model was developed under the assumption that the dryer is in steady state. As a result the gradual heating of the dryer cylinders is ignored.

2. Cylinder model. To obtain a model under dynamical process conditions, the gradual heating and cooling of the individual dryer cylinders has also been modelled.

3. The model for the behavior of the input dryness and heat transfer correction factors for the PDR and ADR sections.

In the remainder of this section we will discuss the various submodels in more detail.

6.2.3 TNO Dryer model

In the first part of the dryer model the properties of the paper web are modelled as it travels through the dryer section. This part of the model can be used to compute both the dry-mass fraction of the paper as well as the paper temperature at each point within the dryer section. The level of detail in this part of the model is such that it is even possible to compute temperature and dryness profiles for the interior of the paper web. An example of a temperature and dryness profile as computed by this part of the model is presented in Figure 6.1. In this figure we can easily identify the increase in paper web temperature at the top and at the bottom of the paper web when it comes into contact with a dryer cylinder. In the dryness profile we can see how the dry-mass fraction gradually increases until the paper reaches the coater unit. The coater unit increases the water fraction by spraying water and chemicals on the paper web to enhance the final paper properties. After the coater unit, the dryness of the paper web again gradually increases until it reaches a dryness of approximately 90% at the end of the dryer section.

The TNO Dryer model is described in detail in [50]. The dryer model consists of four main mass and energy balances. These balances and their main equations are:

1. External mass transport balance. This mass balance describes the transport of water vapor from the surface of the paper to the surrounding air. The driving forces behind the evaporation process are the internal pressures of the water

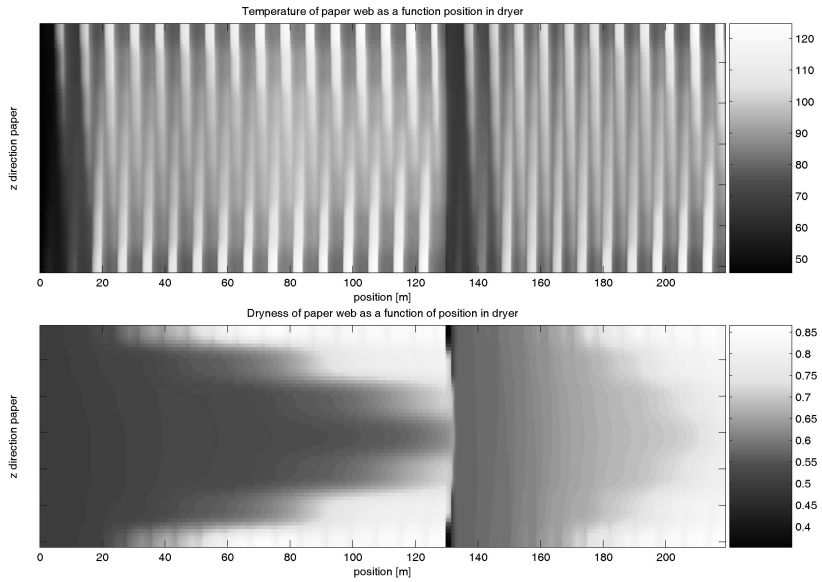

Figure 6.1 Example of a temperature and dryness profile for the cross-section of the paper web as a function of the position inside the dryer section. This profile was computed using the TNO Dryer model.

vapor within the paper. The evaporation process can be modelled using Stefan's law:

$$J_{w,ext} = \frac{kM_v P_{tot}}{RT_{avg}} \ln\left(\frac{P_{tot} - P_{v,a}}{P_{tot} - P_{v,p}}\right). \tag{6.1}$$

In this equation $J_{w,ext}$ is the mass flux of water, k is the mass-transfer coefficient, M_v is the molecular mass of the water vapor, T_{avg} is the average temperature over the paper-air interface, $P_{v,a}$, $P_{v,i}$ are the partial pressures of the water vapor and the air and P_{tot} is the total pressure.

2. Internal mass transport balance. This mass balance describes the transport of water within the paper. This mass balance consists of two parts that model water in its liquid and vapor phases respectively. The transport of liquid water is modelled using Darcy's law, the transport of vapor is described using Fick's equation. According to Darcy's law for the flow of liquids it should hold that the total transport of liquid water $J_{l,int}$ the z direction (perpendicular to the plane of the paper) is

$$J_{l,int} = \frac{k}{\mu} \frac{\partial P_h}{\partial z} \tag{6.2}$$

with k the permeability of the paper, A the capillary surface, μ the viscosity of the water and P_h the hydrostatic pressure of the water.

According to Fick's law the vapor transport $J_{v,int}$ is equal to

$$J_{v,int} = \frac{DM_v}{RT} \frac{P_{tot}}{P_{tot} - P_v} \frac{\partial P_v}{\partial z}, \tag{6.3}$$

with D the effective diffusion coefficient.

3. External heat transfer. This energy balance describes heat exchange between the paper web and either the surrounding air or a cylinder wall depending on the position of the paper web. The main equation that describes the external energy flux q_{ext} is

$$q_{ext} = \alpha(T_{paper} - T_{air}) + J_{v,ext} h_v, \tag{6.4}$$

in which α is a heat-transfer coefficient and h_v is the enthalpy of the vapor leaving the paper.

4. Internal heat transfer. This final equation describes the heat transfer within the paper. The heat flux within the web consists of three parts. The first two parts are caused by the movement of the water in either liquid of vapor form. The final part consists of heat conduction. Combined, this leads to the equation

$$q_{int} = J_{l,int} h_l + J_{v,int} h_v + \lambda_{paper} \frac{\partial T_{paper}}{\partial z}, \tag{6.5}$$

with $J_{l,int}$, $J_{v,int}$ as modelled in (6.2) and (6.3), h_l, h_v the enthalpy of the water inside the paper, λ_{paper} the heat conductivity of the paper and T_{paper} the temperature of the paper.

Combining these equations leads to a coupled set of PDEs that provides a complete description of the paper web. The boundary conditions for these PDEs are dependent on the machine configuration. The combined set of PDEs and boundary equations are solved numerically by applying a finite difference technique.

Even though the solution to the PDEs describes the complete paper web at all points of the machine, we will only use a part of the solution of the TNO Dryer model. Specifically we will only use the TNO Dryer model to compute all the paper dryness just before the coater and at the end of the dryer section, the average temperature of the paper web that is in contact with the various cylinders and all heat transfer coefficients between the cylinder surface and the paper for each dryer cylinder:

$$[d_{out}(t), d_R(t), T_{pap,1\ldots,54}, \alpha_{c\to p,1,\ldots 54}, \alpha_{c\to p,1,\ldots 54}, \alpha_{c\to p,1,\ldots 54}] = f_{TNO}(T_{cil,1,\ldots 54}, d_{in}(t), c_{PDR}(t), c_{ADR}, c_{ADR}(t)). \quad (6.6)$$

For the meaning of each variable in this function the reader is referred to Table 6.1. The TNO Dryer model has been implemented in 'C' by TNO Science and Industry.

As already mentioned, in order to use the TNO Dryer model the user has to provide the several inputs to the model, i.e. the average temperature of each dryer cylinder, the dryness of the paper at the start of the dryer section as well as both correction factors for the heat transfer in the PDR and ADR sections of the dryer.

6.2.4 Cylinder model

In this part of the dryer model the temperature evolution of each of the dryer cylinders is described.

Each dryer cylinder gains thermal energy by steam that passes through the hollow interior of the cylinder. At the same time a cylinder looses thermal energy to the paper web and to the surrounding air. The temperature evolution of a dryer cylinder can be approximately modelled using a simplified energy balance [2]. For a cylinder in the PDR section of the dryer section the resulting differential equation is:

$$\frac{m_{cil}c_p}{c_{PDR}}\frac{\partial T_{cil,i}(t)}{\partial t} = A_{c\to p}\alpha^i_{c\to p}(t)[T_{cil,i}(t) - T_{pap,i}(t)] + A_{c\to s}\alpha_{c\to s}(t)[T_{cil,i}(t) - T_{steam,i}(t)] + A_{c\to a}\alpha_{c\to a}(t)[T_{cil,i}(t) - T_{air}(t)]. \quad (6.7)$$

For a cylinder in the ADR section c_{PDR} would be replaced by c_{ADR}. The meaning of all variables in this expression can be found in Table 6.1.

To derive the previous energy balance two approximating assumptions were made. Firstly, it has been assumed that the temperature of each cylinder is uniform. This assumption can be justified since the cylinders consist of iron, which has a very good heat conductivity. Secondly the temperatures of both the air and the paper web touching the cylinders are assumed constant. This is not a crucial assumption, since we are only interested in the total heat transfer from a dryer cylinder to its surroundings.

For our model for the cylinder temperature, we require the average heat transfer coefficients from the cylinder to its surroundings. These heat transfer coefficients are

Table 6.1 Symbols of the dryer model, their meaning and units.

Symbol	Unit	Explanation
$\alpha_{c\to a}$	$\frac{W}{m^2 °C}$	heat transfer coefficient between i-th dryer and air
$\alpha_{c\to p,i}$	$\frac{W}{m^2 °C}$	heat transfer coefficient between i-th dryer and paper
$\alpha_{c\to s,i}$	$\frac{W}{m^2 °C}$	heat transfer coefficient between i-th dryer and steam
c_{ADR}	[1]	correction factor for heat transfer in after dryer section (final 22 cylinders)
c_{PDR}	[1]	correction factor for heat transfer in the pre dryer section (first 32 cylinders)
γ	$\frac{J}{kg°C}$	specific heat of the cylinder
d_{in}	[1]	mass fraction dry paper in the paper mixture entering the dryer section ($d_{in} = m_{pap}/(m_{pap} + m_{water})$)
d_{out}	[1]	measured mass fraction of dry paper measured after the pre- and after-dryer sections
g	gr/m^2	average weight of mass entering dryer per m^2
m_{cil}	kg	mass of a dryer cylinder
v_{pap}	m/s	speed at which the paper web moves through the dryer section
$A_{c\to a}$	m^2	contact surface between a dryer cylinder and surrounding air
$A_{c\to p}$	m^2	contact surface between a dryer cylinder and the paper web
$A_{c\to s}$	m^2	contact surface between a dryer cylinder and steam heating the cylinder
$P_{steam,i}$	bar	pressure of saturated steam that heats the cylinders in the i-th cylinder group
T_{air}	$°C$	temperature of air surrounding the dryer section
$T_{cil,i}$	$°C$	temperature of the i-th dryer cylinder
$T_{pap,i}$	$°C$	temperature of paper surface at the i-th cylinder
$T_{steam,i}$	$°C$	temperature of steam inside the i-th cylinder

not trivial to compute as they may depend on the dryness of the paper at the cylinder and the average temperature of the paper in contact which the cylinder. Fortunately the TNO Dryer model can be used to compute the various heat conductivities and the average paper temperatures of paper at the contact surface at every cylinder, see (6.6).

6.2.5 Model for input dryness and correction factors

From equations (6.6) and (6.7) we see that both the TNO dryer model and the cylinder model are dependent on the input dryness $d_{in}(k)$ and on the correction factors for the PDR and ADR sections of the dryer, respectively.

Both the input dryness and the correction factors are empirically modelled (in discrete time) as first order autoregressive model processes that fluctuate around a given mean value. For example, the behavior of the dryness of the paper at the entrance of the dryer section is assumed to correspond to the following first order autoregressive process:

$$d_{in}(k+1) - \overline{d_{in}} = a_{d_{in}}(d_{in}(k) - \overline{d_{in}}) + w_{d_{in}}(k), \tag{6.8}$$

with $\overline{d_{in}}$ the average dryness of the paper at the entrance of the dryer section, the $a_{d_{in}}$ the autoregressive parameter of the process and $w_{d_{in}}(k)$ a zero mean Gaussian distributed white noise sequence. For the simulation examples the autoregressive parameter $a_{d_{in}}$, the average input dryness $\overline{d_{in}}$ and the variance of $w_{d_{in}}(k)$ are assumed known. The values for these parameters can be found in Table 6.2. In the table, the variance of $w_{d_{in}}(k)$ is denoted as $Q_{w_{d_{in}}}$.

As mentioned before, the behavior of the correction factors $c_{PDR}(k)$ and c_{ADR} are also assumed to correspond to first order autoregressive processes:

$$c_{PDR}(k+1) - \overline{c_{PDR}} = a_{c_{PDR}}(c_{PDR}(k) - \overline{c_{PDR}}) + w_{c_{PDR}}(k) \tag{6.9}$$

$$c_{ADR}(k+1) - \overline{c_{ADR}} = a_{c_{ADR}}(c_{ADR}(k) - \overline{c_{ADR}}) + w_{c_{ADR}}(k), \tag{6.10}$$

with $\overline{c_{PDR}}$, $\overline{c_{ADR}}$ the average values for the correction factors, $a_{c_{PDR}}$, $a_{c_{ADR}}$ the autoregressive parameters of the processes and $w_{c_{PDR}}(k)$, $w_{c_{ADR}}(k)$ zero mean Gaussian white noise processes with variances $Q_{w_{c_{PDR}}}$ and $Q_{w_{c_{ADR}}}$. As was the case for the input dryness behavior, we assume that all parameters in each of the autoregressive processes are known. All values used in the simulation experiments are given in Table 6.2.

6.2.6 The combined discrete time process model

The various sub-models of the dryer section can be combined to form a complete dynamic model for the entire dryer section. The simulation of this complete model requires that we numerically resolve the set of equations (6.6)-(6.10). For this purpose we have used the fourth order Runge-Kutta method [76] using integration steps of 15 seconds.

Since the actual sampling time of the paper dryness measurements is 60 seconds, the discrete time model for the dryer section can be obtained by using every fourth Runge-Kutta numerical integration result.

Table 6.2 Parameter values for the autoregressive models of input dryness d_{in} and both correction factors c_{PDR} and c_{ADR} (see (6.8)-(6.10)).

Parameter	Value
$a_{d_{in}}$	0.96
$a_{c_{PDR}}$	0.999
$a_{c_{ADR}}$	0.9995
$\overline{d_{in}}$	0.48
$\overline{c_{PDR}}$	5.88
$\overline{c_{ADR}}$	0.73
$Q_{w_{d_{in}}}$	$1,764 \cdot 10^{-5}$
$Q_{w_{c_{PDR}}}$	$2.5594 \cdot 10^{-5}$
$Q_{w_{c_{ADR}}}$	$5.1174 \cdot 10^{-3}$

The obtained process model for the dryer section can be written in the form of a discrete time nonlinear state equation:

$$x(k+1) = f(x(k), u(k)) + w(k) \tag{6.11}$$

with

$$x(k) = [T_{cyl,1}(k), \ldots, T_{cyl,54}(k), d_{in}(k), c_{PDR}(k), c_{ADR}(k)]^T \tag{6.12}$$

$$u(k) = [T_{steam,1}(k), \ldots, T_{steam,6}(k), v_{pap}(k), g(k), coater(k)]^T \tag{6.13}$$

$$w(k) = [w_{T_{cyl,1}}(k), \ldots, w_{T_{cyl,54}}(k), w_{d_{in}}(k), w_{c_{PDR}}(k), w_{c_{ADR}}(k)]^T, \tag{6.14}$$

with w(k) an additive Gaussian white noise process with mean zero and covariance Q specified by:

$$Q = \mathbb{E}(w(k)w(k)^T) = diag(0.1^2, \ldots, 0.1^2, Q_{d_{in}}, Q_{c_{PDR}}, Q_{c_{ADR}}). \tag{6.15}$$

In equation (6.11) an additive white noise term $w(k)$ has been added. This additive white noise has been introduced to allow for irreproducible effects that lead to slight variations in cylinder temperature. Moreover, this additive noise will later allow a state filter designed using this model to compensate for minor modelling inaccuracies (see also section 2.3.4).

An important remark for the sequel is that the CPU time required to compute the state update of the combined dryer model is approximately 48 seconds on an 1400MHz AMD Athlon based computer. Note that the required CPU time per model evaluation is of the same order as the sampling interval of 60 seconds.

6.2.7 Measurement model

Besides a model for the state dynamics, we also require a model for the measurements, i.e. the measured dry-mass fraction of the paper web measured just before the coater and just after the final dryer cylinder, where the paper leaves the dryer section.

The measurement model is easily obtained, since the existing TNO Dryer model (see 6.6) can be used to compute the dry-mass fraction at all possible positions in the dryer section. So by rewriting the TNO Dryer model, we can construct a measurement equation of the form:

$$y(k) = h(x(k), u(k)) + v(k), \tag{6.16}$$

with $y(k) = [d_{out}(k)\ d_{coater}(k)]^T$, $x(k)$ and $u(k)$ defined as before, $h(\cdot)$ the nonlinear measurement equation derived form (6.6). The term $v(k)$ is an additive Gaussian white noise which has been added to the measurement model to describe the measurement errors. The dry-mass fraction sensors used in practice have a standard deviation of approximately $25 \cdot 10^{-3}$. The covariance matrix of the measurement errors is chosen accordingly:

$$R = \mathbb{E}\{v(k)v(k)^T\} = \text{diag}(6.25 \cdot 10^{-4}, 6.25 \cdot 10^{-4}). \tag{6.17}$$

The measurement errors $v(k)$ and the process disturbances $w(k)$ are assumed to be independent.

6.2.8 Summary of dryer model properties

In this section we have discussed the model that will be used in order to test the properties of the methodologies discussed in previous chapters. In this section we will summarize the properties of the dryer model and verify to which extent the dryer model corresponds to the type of nonlinear process models that we considered in the problem statement in the beginning of this thesis (see section 1.3).

In the problem statement of this thesis we focussed on nonlinear large scale first principles models of complex processes in the process industry. We assumed that the CPU time per model evaluation of the model is at least of the same order as the sampling time of the data. Finally, we assumed that in these models the properties of noise disturbances acting on states and measurements may be unknown and the models may need to be recalibrated during operation due to changing process conditions.

The dryer model has 57 states. This is relatively low compared to finite difference models such as the heated plate model we used earlier in this thesis. The fact that the dryer model is of relatively low order will prevent an effective demonstration of the capacity of model reduction techniques.

The simulation time of the dryer model is 48 seconds, while the measurements are sampled every 60 seconds. The CPU time per model evaluation is thus of the same order as the sampling time per measurement. As a result the dryer section case can be used to show the usefulness of the methodologies introduced in chapter 3 to speed up model computations.

In this case study we will only use simulated data. As a result the properties of process disturbances and measurement errors are exactly known. However we can still use the dryer model to test the techniques introduced in chapter 4 to construct a Kalman filter without covariance information.

Finally, the dryer model currently does not contain any parameters that may need to be re-estimated online. In order to test the model selection techniques as introduced in chapter 5, we will extend the model with additional parameters, which we can easily change over time.

6.3 Model reduction

The total dryer model has 57 states. The first 54 states represent the temperature of each dryer cylinder, the remaining three states are the input dryness and the two correction factors (see (6.8)-(6.10)). In this section we will use the POD model reduction technique described in section 2.4.2 to reduce the order of this model.

The reduction of the model is not motivated by a very high state dimension of the original model. A model order of 57 is indeed not that high. However if it is possible to obtain a reduced order model that has a significantly lower state dimension, this will reduce the computational load when constructing a state filter or when deriving approximative models.

For the POD model reduction we will only reduce the number of states that represent the cylinder temperatures. The reason for only using POD model reduction on this part of the state vector is that the POD model reduction technique is sensitive to the scaling of the state vector. The POD method only retains those directions of the state-space which exhibit the most variability (see section 2.4.2). In the case of the dryer model, the variation in the input dryness and in both correction factors will generally be much smaller than the variation of the cylinder temperatures. This means that the states in which we were most interested, would likely be removed by model reduction. This problem is easily circumvented by only considering the first 54 states (cylinder temperatures) for model reduction.

To reduce the number of cylinder temperature states we need to construct a snapshot matrix X_{snap} as defined in (2.94). For this purpose we construct a simulation run of 16500 samples with the complete dryer model (6.11).

The input sequence $u(k)$ used to generate this sequence should be representative of input data that would be encountered in practice. Unfortunately, we only have a sequence of 300 inputs available from an actual industrial production run. Consequently, a procedure is needed to extend the input sequence of 300 measured inputs to a sequence of 16500 inputs. For the dryer case, the following procedure has been used:

Procedure 6.1 *In order to extend the available sequence on 300 inputs $u(k)$ to a sequence of arbitrary length, the following procedure has been used:*

1. *For each component of the input vector $u(k)$, examine the available sequence of 300 inputs.*
2. *Identify for each input component a time series model of the input sequence using ARMA models or step-functions. For ARMA models the model order and the*

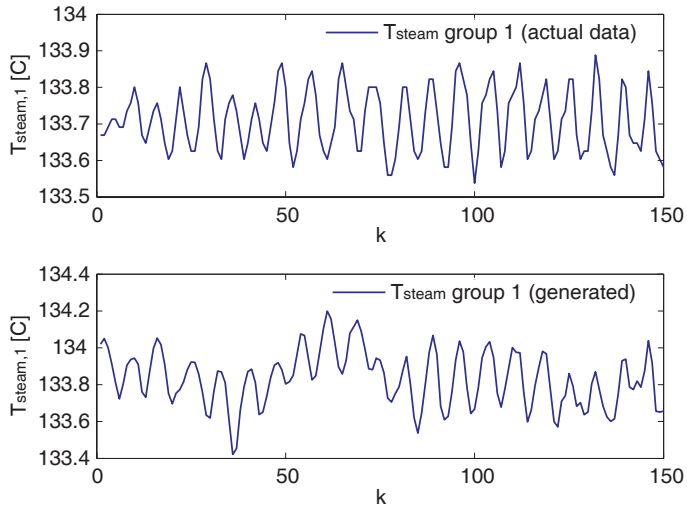

Figure 6.2 Measured and generated data of the temperature of the steam flowing to the first cylinder group, $T_{steam,1}(k)$. The properties of the generated data (mean, variability, autocorrelation) are similar.

parameter values can be determined using the ARMASA estimation procedure [17].

3. *Generate new input data per component using the models obtained in the previous step.*

This procedure has the advantage that the simulated data have the same properties (mean, covariance function), as the original data. Examples of measured data and data generated using the described procedure can be found in Figures 6.2 and 6.3. Figure 6.2 shows the available temperature data for the steam supplied to the first cylinder group, Figure 6.3 shows measured and generated data for the mass of dry material entering the dryer section.

As opposed to the other inputs, which are modelled using ARMA processes, the machine speed $v_{pap}(k)$ has been modelled using a series of step functions. Step functions were chosen because the original data also consisted of a series of steps.

Using the simulated inputs, the full order simulation model (6.11) is used to generate 16500 states $x(1),\ldots,x(16500)$. These states can then be used to determine the POD basis using the method of snapshots previously described in section 2.4.2.

After subtracting the mean values of the states generated using the simulated inputs, the snapshot matrix X_{snap} is constructed:

$$X_{\text{snap}} = \left[\begin{array}{ccc} x(1)-\overline{x} & \cdots & x(16500)-\overline{x} \end{array} \right], \tag{6.18}$$

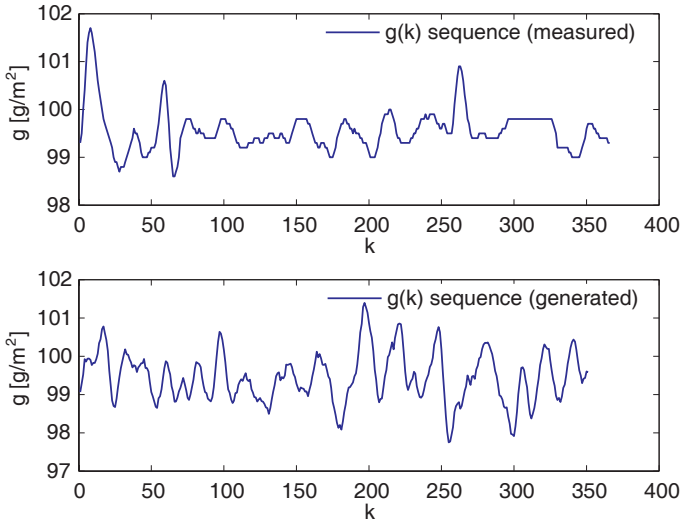

Figure 6.3 Measured and generated data of the average weight per square meter entering the dryer section, $g_k(k)$. The properties of the generated data (mean, variability, autocorrelation) are similar.

in which $\overline{x}$ is the mean of the simulated states:

$$\overline{x} = \frac{1}{N} \sum_{k=1}^{16500} x(k). \tag{6.19}$$

To compute the POD basis for the simulation data we compute the singular value decomposition of the snapshot matrix X_{snap}. The decreasing singular values $\sigma_1, \ldots, \sigma_{54}$ of the snapshot matrix are depicted in Figure 6.4. In this figure we see that the computed singular values quickly decrease until $k = 20$. From approximately $k = 25$ the singular values have decreased at least two orders of magnitude compared to first singular value σ_1. This leads us to reduce the number of temperature states from 54 to 25 by projecting the cylinder temperatures onto the first 25 vectors corresponding to the 25 largest singular values of X_{snap}.

Since we are not reducing the complete state vector, but only the cylinder temperature states, we have to employ a slightly modified version of the Galerkin projection technique to obtain the reduced order model. Denoting T as a $\mathbb{R}^{54\times 25}$ matrix whose columns consist of the singular vectors corresponding to the 25 largest singular values of X_{snap}, the reduced order state model is given by:

$$\begin{aligned} x_{red}(k+1) &= f_{red}(x_{red}(k), u(k), w(k)) && (6.20)\\ &= P^T f(P x_{red}(k), u(k), w(k)), && (6.21) \end{aligned}$$

with $f(\cdot)$ the original full order state model and $P \in \mathbb{R}^{54\times 28}$ a projection matrix defined

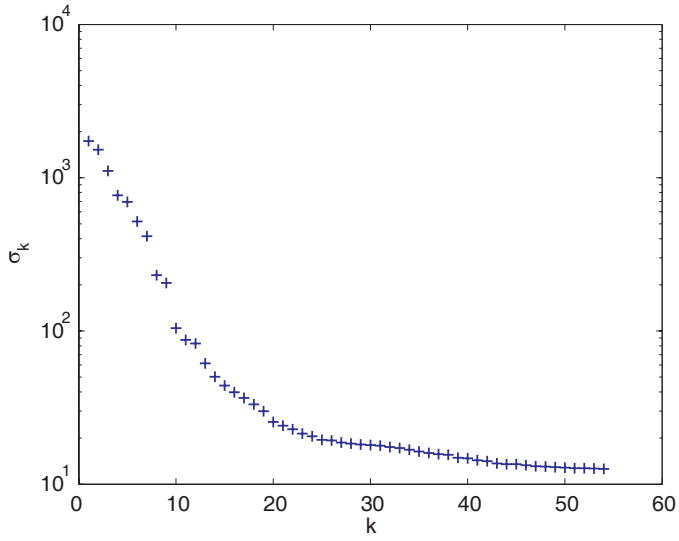

Figure 6.4 Singular values obtained by computing the Singular Value Decomposition of the snapshot matrix for the dryer section. The singular values for $k > 25$ have decreased approximately 2 orders of magnitude compared to the first singular value.

as:

$$P = \begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix}. \tag{6.22}$$

The reduced order measurement equation for the dryer model can be constructed similarly:

$$\begin{aligned} y(k) &= h_{red}(x_{red}(k), u(k), v(k)) && (6.23) \\ &= h(Px_{red}, u(k), v(k)), && (6.24) \end{aligned}$$

with P defined as in (6.22).

To verify that the reduced order model obtained by the POD procedure described above is an accurate approximation of the original full order model, the step responses of computed output dryness of the reduced order model have been compared to the step responses of the original model. As an example, Figure 6.5 depicts the step response of both original and reduced order model to a step in the steam temperature of the second cylinder group, $T_{steam,2}$. As can be seen the predicted output dryness of both models overlap.

The step responses for other changes in inputs and states were also computed, but no significant deviations from the original model were identified. The obtained reduced order model is thus an accurate representation of the original full order dryer model.

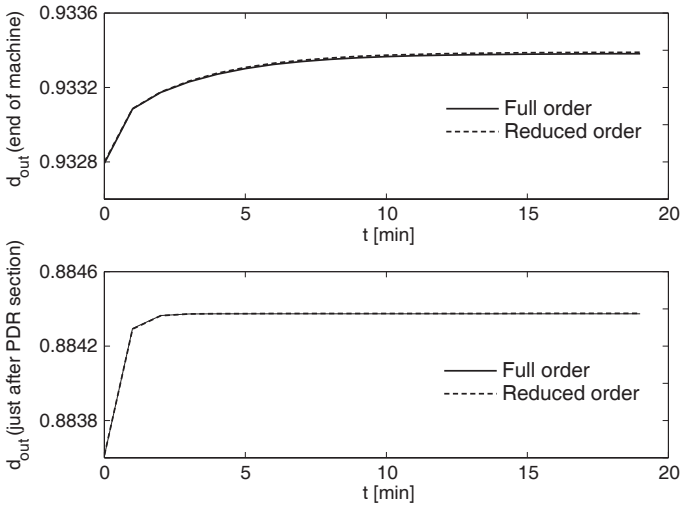

Figure 6.5 Responses of the measured dryness to a step increase of the steam pressure (and thus temperature) of the second cylinder group as computed by both the original and reduced order model. The top figure displays the response for the dryness at the end of the dryer section, the bottom picture shows the response for the dryer just after the PDR section. The increase steam pressure for the 2nd cylinder group was $0.5 \cdot 10^5 Pa$. The reduced order model was obtained by reducing the number of cylinder temperature states from 54 to 25 using the POD technique.

For the dryer case, we have managed to construct an accurate reduced order model. The order of the reduced order model is 28 compared to 57 for the full order model. The model order has thus been reduced by approximately 50%. Although the order of the dryer model has been significantly reduced, the reduction in model order is relatively minor compared to other examples in the literature (see for instance [4][8]). The modest result is due to the fact the original order of the dryer order is relatively low. As a result, lowering the model order by two orders of magnitude (as observed in [4][8]) is not possible for the particular example of the dryer model.

Even though the model order has been reduced, the time to compute a full time step with the reduced order model is 48 seconds, unchanged from the CPU time required for a time step with the original full order model. The required CPU time is not reduced by model reduction, because in order to compute a state update using the reduced order model, a model evaluation of the full order model is still required (see section 3.2).

6.4 Approximation using qLPV identification

6.4.1 Introduction

The computation time required to perform a single model evaluation (corresponding to 1 minute of production) using either the full order model or the reduced order model is approximately 48 seconds on a 1400Mhz AMD Athlon PC. As a result neither model can be used for online state estimation using any of the techniques described in section 2.3, as each of the described state estimation methods requires that many model evaluations can be performed in the time between sampled measurements. To cope with this problem we will construct a new (approximate) dryer model which has a significantly shorter simulation time, but still retains the physical interpretable state vector. In chapter 3 two methods were presented for this specific purpose, either by partitioning the original large scale model (see section 3.3), or by approximating the reduced order model with a qLPV model structure (see section 3.5).

For the dryer model partitioning is not useful, because the dryer model does not satisfy the requirement that computing a partial state update is significantly faster than computing the state update for all states (see equation (3.16)). This is a result of the model structure of the dryer model. Consequently only the qLPV method described in section 3.5 will be used to generate a computationally faster model.

The qLPV model that will be identified is of the form:

$$f_{id}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{M} \phi_m(x_{red},u(k)) \left[A_m x_{red}(k) + B_m u(k) + x_{off,m}\right], \quad (6.25)$$

as proposed in (3.58). To identify this model, we will use the method as described in section 3.5. The identification of the model (6.25) will be split up into two parts.

First matrices $A_0, B_0, \ldots, A_M, B_M$ and vectors $x_{off,0}, \ldots x_{off,M}$ will be identified as has been described in section 3.5.2. Then in the second step the scheduling functions $\phi_1(x(k),u(k)), \ldots, \phi_M(x(k),u(k))$ will be determined using the methodology described in section 3.5.3.

In order to identify the model (6.25), we will use the same 16.500 points of data that were used for model reduction. It is assumed that this data covers the entire operational envelope in which the identified model is to be used in practice.

6.4.2 Identification of $A_0, B_0, x_0, \ldots, A_M, B_M, x_{off,M}$

The methodology outlined in section 3.5.2 to identify matrices $A_0, B_0, \ldots, A_M, B_M$ and vectors $x_{off,0}, \ldots x_{off,M}$ consists of three steps:

1. Determine $A_0, B_0, x_{off,0}$ such that the model given by $A_0, B_0, x_{off,0}$ is the best linear model for the available simulation data.

2. Determine the matrices $A_1^l, B_1^l, x_{off,1}^l, \ldots A_N^l, B_N^l, x_{off,N}^l$ and coefficients $\beta_1^l(k), \ldots, \beta_N^l(k)$ of a long intermediate LPV expansion $f_{id}^{long}(\cdot)$ of the form:

$$f_{id}^{long}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{N} \beta_m^l(k) \left[A_m^l x_{red}(k) + B_m^l u(k) + x_{off,m}^l \right] \quad (6.26)$$

such that the resulting expansion matches the reduced order model $f_{red}(\cdot)$ for all $(x_{red}(k),u(k)) \in Z^N$:

$$f_{red}(x_{red}(k),u(k)) = f_{id}^{long}(x_{red}(k),u(k)) \quad \forall \quad (x_{red}(k),u(k)) \in Z^N. \quad (6.27)$$

3. Finally, use the previously computed expansion $f_{id}^{long}(\cdot)$ to compute a shorter expansion $f_{id}^{short}(\cdot)$ of the form:

$$f_{id}^{short}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{M} \beta_m(k) \left[A_m x_{red}(k) + B_m u(k) + x_{off,m} \right] \quad (6.28)$$

with $M << N$. For this new expansion $f_{id}^{short}(\cdot)$ new matrices $A_1, B_1, \ldots, A_M, B_M$ and coefficients $\beta_1(k), \ldots, \beta_M(k)$ will be computed such that for the available simulation data Z^N the prediction error of the new shorter expansion is lower than an user-defined threshold α:

$$\frac{\sum_{k=1}^{N} \| f_{red}(x_{red}(k),u(k)) - f_{id}^{short}(x_{red}(k),u(k)) \|^2}{\sum_{k=1}^{N} \| f_{red}(x_{red}(k),u(k)) \|^2} < \alpha, \quad (6.29)$$

with α some chosen small value, for instance $\alpha = 0.01$.

In the first step of the identification procedure we use the data to estimate an affine model of the form:

$$x_{red}(k+1) = A_0 x_{red}(k) + B_0 u(k) + x_{\text{off},0}. \tag{6.30}$$

As described in section 3.5.2, system matrices A_0, B_0 and offset vector $x_{off,0}$ can be determined by minimizing the least squares criterion (3.63).

The second step in the identification is to construct a long intermediate qLPV model $f_{id}^{long}(\cdot)$ that perfectly matches all data points in the available simulation data. The matrices $A_1^l, B_1^l, \ldots, A_N^l, B_N^l$ and vectors $x_{off,1}^l, \ldots, x_{off,N}^l$ of this expansion are constructed by linearizing the available dryer model (6.21) at every datapoint, see Proposition 3.5. Due to computer constraints, we chose to construct the intermediate model (6.26) only for every 10th data point. The to the nature of the process, we assume that only using every 10th point will not significantly affect the results.

In the third step, we use the computed long intermediate model to compute a shorter qLPV model $f_{id}^{short}(\cdot)$ (see (6.28)) with the final matrices $A_0, B_0, \ldots, A_M, B_M$ and vectors $x_{off,0}, \ldots x_{off,M}$. This model is constructed using Procedure 3.6.

Here the number of component models is chosen to be $M = 25$. The number of component models has been determined by evaluating the relative error criterion (6.29) for increasing values of M, see Figure 6.6. As can be seen in the Figure, the relative prediction error (6.29) decreases rapidly until approximately $M = 25$. For $M > 25$, the decrease in the relative prediction error is less pronounced. We thus choose $M = 25$, as a compromise between the complexity of the qLPV model and the attainable accuracy of the qLPV model.

6.4.3 Identification of scheduling functions $\phi_i(x(k), u(k), \theta_i)$

In the previous section we have constructed the matrices $A_0, B_0, \ldots, A_M, B_M$ and vectors $x_{off,0}, \ldots, x_{off,M}$, such that criterion (6.29) is satisfied. To satisfy this criterion, we used coefficients $\beta_1(k), \ldots, \beta_M(k)$, that have be computed according to (3.86). These coefficients $\beta_1(k), \ldots, \beta_M(k)$ are data dependent. In this section we will construct scheduling functions $\phi_i(x(k), u(k), \theta_i)$ for $i = 1 \ldots M$ to replace the coefficients $\beta_1(k), \ldots, \beta_M(k)$. In order to construct the scheduling functions $\phi_i(x(k), u(k), \theta_i)$, we will use the methodology described in section 3.5.3.

The proposed methodology in section 3.5.3 to determine the scheduling functions $\phi_i(x(k), u(k), \theta_i)$ consists of two main steps:

1. Determine a suitable model structure for the functions $\phi_i(x(k), u(k), \theta_i)$;
2. Determine parameter vectors θ_i such that the resulting qLPV model minimizes the quadratic criterion (3.92).

To select an appropriate model structure for the scheduling functions $\phi_i(x(k), u(k), \theta_i)$ some knowledge of the process is required. Specifically, we need to know which states and inputs are mainly responsible for the nonlinearities of the system.

For the dryer model, step response experiments show that the main states that cause the nonlinear behavior of the dryer model are:

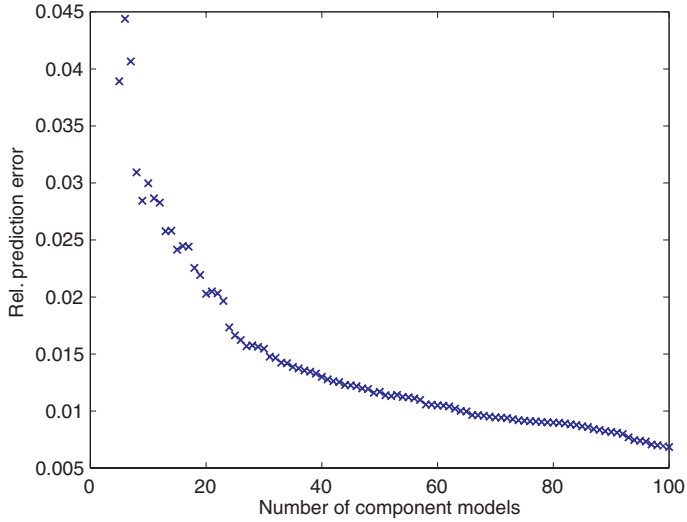

Figure 6.6 Relative error (6.29) of the shorter qLPV model $f_{id}^{short}(\cdot)$ as a function of the number of component models M. The relative errors rapidly decreases until approximately $M = 25$. The rate at which the relative prediction errors decreases is slower for larger values of M.

- input dryness, $d_{in}(k)$,
- correction factors for heat transfer, $c_{PDR}(k)$ and $c_{ADR}(k)$),
- machine speed, $v_{pap}(k)$,
- first five POD cylinder temperature states $(x_{red,1},\ldots,x_{red,5})$.

Of these, the input dryness has the largest impact on the behavior of the dryer process. The effect of the other listed states on the process behavior is relatively minor. In order to select an appropriate structure for the functions $\phi_i(x(k),u(k),\theta_i)$, section 3.5.3 suggests plotting the coefficients $\beta_i(k)$ as a function of the selected states. An example of such a plot is given in Figure 6.7. In Figure 6.7, $\beta_2(k)$ is plotted as a function of $d_{in}(k)$. Unfortunately, the Figure does not clearly indicate how the structure of the functions $\phi_i(x(k),u(k),\theta_i)$ should be chosen.

The following model structure for the functions $\phi_i(x(k),u(k),\theta_i)$ has been selected:

$$\phi_i(x_{red}(k),u(k),\theta_i) = \\ [1\ d_{in}(k)\ d_{in}(k)^2\ c_{PDR}(k)\ c_{ADR}(k)\ x_{red,1}(k)\ \ldots\ x_{red,5},v_{pap}(k)]\theta_i, \quad (6.31)$$

with $\theta_i \in \mathbb{R}^{11\times 1}$ a parameter vector that is yet to be determined. The motivation for choosing this model structure is:

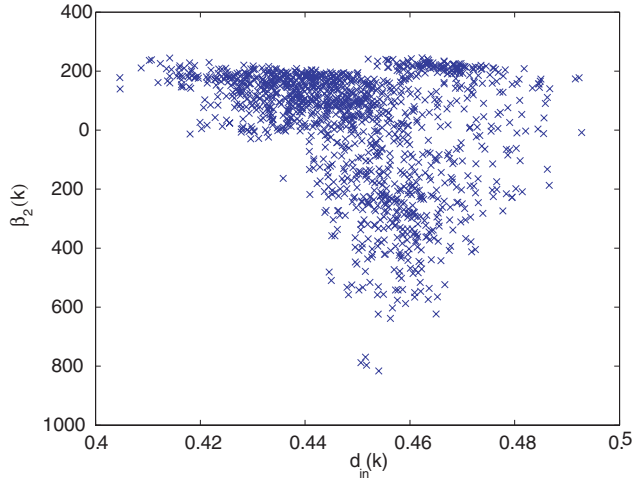

Figure 6.7 Scatter plot of coefficients $\beta_2(k)$ as a function of $d_{in}(k)$. As can be seen there is no obvious relation between the coefficients $\beta_2(k)$ and $d_{in}(k)$.

- Model structure is affine in states $c_{PDR}(k), c_{ADR}(k), v_{pap}(k), x_{red,1}(k), \ldots, x_{red,5}(k)$. Since the nonlinear behavior of the process is not strongly influenced by these states, a simple affine model structure is chosen. Quadratic or higher order terms are not expected to result in significantly better models and are thus omitted. The advantage of using an affine model above a linear model is that offsets can be modelled.

- Model structure is quadratic in the input dryness $d_{in}(k)$. Since the proposes scheduling function (6.31) also contains a quadratic term for $d_{in}(k)$, the scheduling function should be better equipped describe the effect on changes in behavior better than using just a linear term.

- Model is linear in θ_i. This means that the parameters θ_i can be estimated by solving a linear least squares problem.

Now that we have a model structure for the scheduling functions $\phi_i(x(k), u(k), \theta_i)$, all that remains is to determine the parameter vectors θ_i. These parameters θ_i are estimated by minimizing the prediction error of the total model (6.25) as a function of θ_i:

$$\theta_1, \ldots, \theta_M = \arg \min_{\tilde{\theta}_1, \ldots, \tilde{\theta}_M} \frac{1}{N} \sum_{k=1}^{N} \left\| f_{red}(x_{red}(k), u(k)) - f_{id}(x_{red}(k), u(k), \tilde{\theta}_1, \ldots, \tilde{\theta}_M) \right\|^2, \quad (6.32)$$

with $f_{id}(\cdot)$ defined as in (6.25):

$$f_{id}(x_{red}(k),u(k),\theta_1,\ldots,\theta_M) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0} + \sum_{m=1}^{M} \phi_m(x_{red}(k),u(k),\theta_m)\left[A_m x_{red}(k) + B_m u(k) + x_{off,m}\right], \quad (6.33)$$

with $A_m, B_m, x_{off,m}$ as determined in section 6.4.2. Because our chosen model structure for $\phi_i(x(k),u(k),\theta_i)$ is linear in θ_i, problem (6.33) is a linear least squares minimization problem which can be solved efficiently.

Due to computer memory constraints we did not use the complete dataset of 16.500 points. Instead only 2200 datapoints were used to determine the parameter vectors θ_i.

6.4.4 Accuracy of the qLPV model

To determine the accuracy of the constructed qLPV model, we introduce the following averaged quadratic error measure:

$$\mathcal{E} = \frac{1}{N}\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k)) - f_{id}(x_{red}(k),u(k))\|^2. \quad (6.34)$$

Table 6.3 contains the average quadratic error $\mathcal{E}$ of the constructed qLPV model. The average quadratic error is provided using both the 2200 point data set that was used to determine the parametervectors θ_i, as well as using a new data set consisting of 800 points. For reference this table also gives the average quadratic error $\mathcal{E}$ for a simple affine model given by:

$$f_{id}(x_{red}(k),u(k)) = A_0 x_{red}(k) + B_0 u(k) + x_{off,0}, \quad (6.35)$$

with matrices A_0, B_0 and vector $x_{off,0}$ as determined in section 6.4.2.

For the estimation data (the data that has also been used to determine the parameter vectors θ_i), the quadratic error for the qLPV model is about 10% lower than that of the simpler affine model. For the validation data the qLPV model is approximately 6% more accurate.

While the qLPV model is more accurate than the affine model, the improvement in accuracy is relatively modest. This is because in the considered operating regime the behavior of the dryer model is quite close to the behavior of a linear model.

Apart from an absolute error measure such as (6.34) we can also express the quality of the estimated linear and qLPV models using a relative error measure. For this purpose, consider the following relative error measure:

$$\mathcal{E}_{rel} = \frac{\frac{1}{N}\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k)) - f_{id}(x_{red}(k),u(k))\|^2}{\frac{1}{N}\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k))\|^2}. \quad (6.36)$$

Table 6.3 Average quadratic error $\mathcal{E}$ (see (6.34)) of the identified quasi-LPV model. Estimation data contained 2200 elements that were used to determine parameter vectors θ_i, validation data set consisted 800 new points. For reference, the average quadratic error $\mathcal{E}$ of the linear model $f_{id}(x_{red}(k),u(k)) = A_0(\theta)x_{red}(k) + B_0(\theta)u(k) + x_{off,0}$ is also provided.

	Linear	quasi LPV
Estimation data	2.84	2.55
Validation data	3.27	3.09

Table 6.4 Relative errors $\mathcal{E}_{rel}$ (see 6.36) of the identified quasi-LPV model. Estimation data contained 2200 elements that were used to determine parameter vectors θ_i, validation data set consisted 800 new points. For reference, the relative error $\mathcal{E}_{rel}$ of the linear model $f_{id}(x_{red}(k),u(k)) = A_0(\theta)x_{red}(k) + B_0(\theta)u(k) + x_{off,0}$ is also provided.

	Linear	quasi LPV
Estimation data	$6.1 \cdot 10^{-3}$	$5.5 \cdot 10^{-3}$
Validation data	$7.0 \cdot 10^{-3}$	$6.7 \cdot 10^{-3}$

The numerator of the relative error is exactly the same as the previous quadratic error, but in the relative error the quadratic errors is divided by $\frac{1}{N}\sum_{k=1}^{N} \|f_{red}(x_{red}(k),u(k))\|^2$. The resulting relative error thus represents the prediction error of the identified model as a fraction of the fluctuations of the original model.

The relative errors $\mathcal{E}_{rel}$ for both the linear model and the qLPV model are given in Table 6.4. As can be seen in the table, both models are able to approximate the original reduced order model to within 1%.

6.5 Construction of a Kalman filter without noise information

In our dryer model both the covariances of the disturbances in the state and measurement equations are known, because we chose the values for these matrices. As a result, we have all the information required to construct a state filter.

In practice however, the covariances matrices of process disturbances and measurement noises are often not available. When there is no noise covariance information available, it is no longer possible to design a state filter. For those situations in which there is no noise information available, we can use the techniques developed in chapter 4 to construct a Kalman filter using only measurement $u(k), y(k)$. In this section we will thus apply the techniques discussed in chapter 4 to the dryer model.

The techniques in chapter 4 require that the process model is linear. This is not

the case for our dryer model. So instead of using the dryer model, we will use a linear dryer model. The linear model that will be used has the following form:

$$\begin{aligned} x_{red}(k+1) &= A_0 x_{red}(k) + B_0 u(k) + x_{\text{off},0} + w(k) && (6.37)\\ y(k) &= C_0 x_{red}(k) + D_0 u(k) + y_{\text{off},0} + v(k), && (6.38) \end{aligned}$$

in which $w(k)$ and $v(k)$ have the same covariance matrices Q and R as in the nonlinear dryer model. The matrices A_0, B_0, C_0 and D_0 and offset vectors $x_{\text{off},0}, y_{\text{off},0}$ are identified using a least squares criterion:

$$[A_0, B_0, x_{\text{off},0}] = \arg\min_{A,B,x_0} \sum_{k=1}^{N} \| f_{red}(x_{red}(k), u(k)) - A x_{red}(k) - B u(k) - x_0 \|^2 \quad (6.39)$$

$$[C_0, D_0, y_{\text{off},0}] = \arg\min_{C,D,y_0} \sum_{k=1}^{N} \| h_{red}(x_{red}(k), u(k)) - C x_{red} - D u(k) - y_0 \|^2. \quad (6.40)$$

The data for this identification are again the set of 16.500 data elements, that have also been used for model reduction in section 6.3 and for the qLPV identification in section 6.4. As a result, the matrices A_0, B_0 and offset vector $x_{off,0}$ are the same matrices and vector that were identified in the previous section.

Using a linear model instead of a nonlinear model will likely not affect the accuracy of the state estimation by much in this particular case. As we already discussed in section 6.4.4, a linear model can capture over 99% percent of the dryer model dynamics.

The identification method as described in chapter 4 constructs a Kalman filter from N data samples $u(k), y(k)$ generated by the noisy system. Using the process model we thus generate $N = 600$ measurements. Using this data we will first construct an approximate filter using the unmodified technique by Mehra [66], as outlined in section 4.2.3. Then we will investigate if it is possible to construct a more accurate filter using the suggested improved technique in section 4.3.

For both techniques in sections 4.2.3 and 4.3 the first required step to compute the approximate Kalman filter is to estimate the covariance function $R_{y_s}(k)$ of $y_s(k)$, with $y_s(k)$ the stochastic part of the measurements. The stochastic part of the measurements can be (approximately) computed from $y(k)$ using (4.30). To estimate the covariance of $y_s(k)$ we have used the N4SID algorithm [93]. To construct a Kalman filter using the unmodified Mehra technique, the estimated covariance function $\hat{R}_{y_s}(k)$ is in turn used to construct the estimated covariance matrix $\hat{M}$ using (4.37). Using the estimated covariance matrix $\hat{M}$, an approximate Kalman filter is constructed, by substituting $\hat{M}$ for M and $\hat{R}_{y_s}(0)$ for $R_{y_s}(0)$ in (4.22). The filter obtained in this manner unfortunately turned out to be unstable.

Table 6.5 Singular values of the observability matrix of the linearized reduced order dryer model

i	1	2	3	4	5	6	$\cdots$	28
σ_i	4.02	0.63	0.0233	0.0088	0.0054	0.0023		$2.8 \cdot 10^{-13}$

The unstable behavior of the filter might be caused by the poor observability of the system. Recall from section 4.3 that for a poorly observable system even very small errors in the estimated covariance function $\hat{R}_{y_s}(k)$ can lead to large estimation errors in $\hat{M}$ which in turn will most likely result in a poor state filter. To investigate if this is indeed the case, the singular values of the observability matrix have been computed. The computed singular values are listed in Table 6.5. From this table we see that the system is indeed poorly observable. After the third singular value, the remaining singular values of the system are close to zero.

Given the poor observability of the system, it is likely that better results can be obtained using the improved covariance method presented in section 4.3. The singular values in Table 6.5 suggest that better results could likely be obtained using only the first two (or possibly three) observable directions of $x(k)$, corresponding to the first two (or three) largest singular values of the observability matrix.

For this particular example however it turned out that the improved procedure by itself was not enough to obtain a stable approximate Kalman filter. Independent of the number of observable modes used to construct the filter, the resulting filter was unstable, causing the estimation errors to increase exponentially for increasing time k. Possible reasons for the instability of the constructed filters are given in section 4.5. In a nutshell, section 4.5 lists two possible reasons why the filters constructed using the methods of section 4.3 can become unstable:

- $\hat{R}_{y_s}(0)$ is underestimated.

- By using only a subset of all observable directions, the poles of the resulting filter move outside the unit circle.

Section 4.5 suggests that both causes can be resolved by intentionally using an overestimated $\hat{R}_{y_s}(0)$ and using $(1-\varepsilon)A$ in combination with the reduced order method from 4.3. Indeed, using $2\hat{R}_{y_s}(0)$ instead of $\hat{R}_{y_s}(0)$ and $0.97A$ instead of A in (4.22) stable state estimators could be constructed.

To evaluate the accuracy of both Mehra's direct covariance method as well as the improved covariance method (including the stabilizing adjustments just described), a new data set is generated that consists of 2000 inputs $u(k)$, states $x(k)$ and measurements $y(k)$. The average accuracy of both state predictions $\hat{x}(k|k-1)$ and state estimates $\hat{x}(k|k)$ has been determined. The prediction and estimation accuracies are

expressed using the following error measures:

$$\mathcal{E}_{pred} = \frac{1}{2000}\sum_{k=1}^{2000} \|x(k) - \hat{x}(k|k-1)\|^2, \tag{6.41}$$

$$\mathcal{E}_{est} = \frac{1}{2000}\sum_{k=1}^{2000} \|x(k) - \hat{x}(k|k)\|^2. \tag{6.42}$$

For reference the estimation and prediction accuracy of the optimal Kalman filter is also computed. The results of each filter can be found in Table 6.6. In the table l denotes the number of observable directions that are used to construct M_l (see (4.52)) in the improved algorithm described section 4.3.

Recall that Mehra's unmodified covariance method results in an unstable filter. This translates into a huge estimation error. When we computed the singular values for the observability matrix, we predicted that a better filter can be constructed when only the first two observable directions (i.e. $l = 2$) of M are used to construct a Kalman filter. In Table 6.6 we see that the filter constructed using just the first two observable directions in M indeed results in a better filter.

To illustrate the effect of choosing the number of observable modes in M either too high or too low, Kalman filters are also constructed using the improved method using $l = 0, \ldots, 8$ observable directions to construct M_l. The accuracy of each of these filters is given in Table 6.6 as well. The results show that choosing l too low decreases the filters accuracy, because a significant amount of information contained in M is not utilized. The results for $l \geq 4$ illustrate that if l is chosen too high, filter accuracy decreases because M_l cannot be determined accurately enough. No results are given for $l = 3$, because for $l = 3$ the approximate Kalman filter becomes unstable.

6.6 State estimation

In this section we will perform the most important part of the case-study; we will test the accuracy with which it is possible to estimate the state $x(k)$ of the dryer section model using only known input signals $u(k)$ and output measurements $y(k)$.

In order to judge the accuracy of the state estimates we will only test the monitoring algorithms using simulated data generated by the full-order dryer model. The test data consisted of 120 minutes of simulated production, resulting in 120 input vectors $u(k)$, true states $x(k)$ and measurement $y(k)$. An advantage of using simulated data instead of practical data is that using simulated data the exact true state of the system is known, allowing us to compare the estimated state with the actual values. Also by generating data with a known model, complicating issues such as model mismatch can be avoided.

To estimate the state of the dryer section, three filters have been constructed:

1. A nonlinear Unscented Kalman Filter based on the reduced order nonlinear dryer model (6.21)-(6.23);

Table 6.6 Average state prediction ($\mathcal{E}_{pred}$ - see (6.41)) and estimation errors ($\mathcal{E}_{est}$ - see (6.42)) of the approximate Kalman filters constructed using Mehra's direct covariance method and the improved estimator of section 4.3. For the improved estimator l denotes the number of observable directions that are used to construct M_l (see 4.52). For reference the results of the optimal Kalman filter for the dryer model are also given.

Filter	Average prediction error $\mathcal{E}_{pred}$	Average estimation error $\mathcal{E}_{est}$
Kalman (Q, R, known)	15.49	15.41
Unmodified covariance	$3.5 \cdot 10^{26}$	$2.0 \cdot 10^{26}$
Improved ($l = 0$)	78.52	78.52
Improved ($l = 1$)	72.13	50.85
Improved ($l = 2$)	27.94	34.13
Improved ($l = 3$)	-	-
Improved ($l = 4$)	32.10	35.28
Improved ($l = 5$)	90.23	95.66
Improved ($l = 6$)	425.69	433.28
Improved ($l = 7$)	387.88	395.51
Improved ($l = 8$)	1469.2	1505.8

2. A nonlinear Unscented Kalman Filter using the identified qLPV model (see section 6.4) as state equation and using (6.38) as output equation;

3. A linear Kalman filter model based on the identified linear model (6.37)-(6.38) for the dryer section.

In theory the first filter should be able to provide the best result of all filters, since the model used should be the most accurate. The only modelling errors in the model used for estimation are the errors introduced by model reduction, which were already verified to be negligible (see section 6.3). Although the filter should provide the best results, the computation time required for this filter is expected to be much larger than the simulated interval, making online implementation of this filter impossible.

Like the first filter, the second filter is based on a nonlinear state equation. However, instead of using the reduced order state model, this filter uses the qLPV state model identified in section 6.4. Although this alone would already greatly speed up computations in the UKF, experiments showed that the CPU required to perform a prediction and correction step in the UKF, would still require too much CPU time for online implementation. The CPU time per estimate is still too high because the output model still contains the TNO dryer model (see 6.2.6). To obtain a filter that can be implemented online the measurement equation for the UKF filter has been replaced by the identified linear output model (6.38). Because both the state and measurement equations contain minor errors, the resulting filter is expected to be less accurate than the filter based on the reduced order model.

The final filter that is used is a linear filter. The linear model used for this filter is the identified linear state-space model that was also used in section 6.5. The main

reason for including this model in the case study is to investigate if the use of a nonlinear dryer model for the dryer section results in a significant increase in estimation accuracy.

For both the linear filter and the filter using the qLPV model, the filter model does not perfectly describe the data. In order to construct a more robust state filter we used the stochastic embedding technique described in section 2.3.4 to obtain a state filter. For the qLPV model this means that we have modelled the error $w_{err}(k)$ (see (2.81)-(2.82)) of the qLPV model as an additional zero mean Gaussian white noise process. The covariance Q_{qLPV} of the additional process noise was determined experimentally via:

$$Q_{qLPV} = \frac{1}{N}\sum_{k=1}^{N}\zeta(k)\zeta(k)^T, \quad \text{with} \tag{6.43}$$

$$\zeta(k) = f_{red}(x_{red}(k),u(k),w(k)) - f_{qLPV}(x_{red}(k),u(k),w(k)). \tag{6.44}$$

Since the measurement equation was also altered, the stochastic embedding technique was also used to obtain a robust filter with respect to the errors in the measurement equation.

The estimation results for all filters including the approximate CPU time for each filter is presented in Table 6.7. The accuracy of each filter for (a part of) the state estimate is expressed in the average squared error:

$$\mathcal{E}_{est}(x) = \frac{1}{120}\sum_{k=1}^{120}\|x(k)-\hat{x}(k|k)\|^2. \tag{6.45}$$

Comparing the total state estimation error of each filter, we see that as expected the filter using the reduced order dryer model (6.21)-(6.23) results in the best estimates. This was to be expected since this filter uses virtually the same model that was used to generate the estimation data. The error of the qLPV based filter is relatively close to that of the reduced order model filter, the error of the linear filter is twice as large as the filter using the reduced order model. The reason that the qLPV model is more accurate than the linear model is because the qLPV model better describes the nonlinear behavior of the plant.

Table 6.7 also lists the average state estimation errors for parts of the state. The errors for the cylinder temperatures $T_{cil,1\ldots54}$ and the after-dryer correction factor $c_{ADR}(k)$ again show that best results were obtained using the reduced order model, that the qLPV model results are relatively close and that the linear model performs the worst. For the input dryness d_{in} and the pre-dryer correction factor c_{PDR} the results are unexpected. Here we see that the results of the filter using the reduced order model are actually not as good as those of the qLPV and linear models. Since the difference in estimation accuracy is relatively minor, this can likely be resolved by tuning the UKF filter.

Finally, Table 6.7 also lists the amount of CPU time required to produce the filter estimates. The filter using the reduced order model needs more than 3 days to process 120 minutes of data. This obviously means that online implementation of this filter is

Table 6.7 Average state estimation errors for the dryer section. Average estimation errors $\mathcal{E}(\cdot)$ are computed over a single data set consisting of 120 measurements (thus corresponding to 120 minutes of production).

	Red. order dryer model	qLPV state model and lin. output model	Lin. state and output model
$\mathcal{E}_{est}(x)$	10.44	12.99	20.27
$\mathcal{E}_{est}(T_{cil,1\ldots54})$	10.07	12.93	20.24
$\mathcal{E}_{est}(d_{in})$	$2.37 \cdot 10^{-5}$	$1.48 \cdot 10^{-5}$	$2.08 \cdot 10^{-5}$
$\mathcal{E}_{est}(c_{pdr})$	$37.1 \cdot 10^{-2}$	$5.73 \cdot 10^{-2}$	$3.22 \cdot 10^{-2}$
$\mathcal{E}_{est}(c_{adr})$	$7.32 \cdot 10^{-4}$	$10.2 \cdot 10^{-4}$	$23.0 \cdot 10^{-4}$
CPU time:	$\sim$ 3 days	25.5 sec.	3.9 sec.

not possible. In contrast, the filters using the qLPV and linear models only required 25.5 en 3.9 seconds to produce the required estimates. This means that the required CPU time to compute the estimates has decreased by a factor of 10^4 for the qLPV model and a factor $6 \cdot 10^4$ for the linear model. As a result, both filters could easily be implemented online.

6.7 Model selection

In this section we will demonstrate the effectiveness of the model selection procedure that was introduced in chapter 6. To test the model selection procedure, we will not use the nonlinear dryer model, but instead we will again use the linear dryer model that was also used in the previous section. The motivation for using the linear dryer model instead of the nonlinear dryer model is that state estimation results for the linear model are very similar to the results of the nonlinear model, but estimation results can be computed in a fraction of the time (see also Table 6.7).

In order to use the linear dryer model to test the model selection procedure we need to extend the linear dryer model with an extra state (or time dependent parameter) that could affect the model behavior in such a way that it would affect the state estimation of the model. For this purpose we have extended the linear dryer model with an additional state $x_b(k)$. The extra state $x_b(k)$ represents a possible bias in the measurements. Ignoring a bias in measurements will likely lead to erroneous state estimates. We assume that the true behavior of the bias is unknown.

To estimate the bias state we will introduce two models. In the first model the bias state $x_b(k)$ is assumed constant:

$$x_b(k+1) = x_b(k). \tag{6.46}$$

In the second model the bias state is assume to be a random walk:

$$x_b(k+1) = x_b(k) + w_b(k), \tag{6.47}$$

with $w_b(k)$ a zero mean Gaussian white noise, with covariance $Q_b = 1$. It is important to realize that these models do not necessarily correspond to the actual behavior of the bias state $x_b(k)$.

The linear dryer model can be easily extended with the bias states. The state equation for the extended linear model is of the form:

$$\begin{bmatrix} x_{red}(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} A_0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{red}(k) \\ x_b(k) \end{bmatrix} + \begin{bmatrix} B_0 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} x_0 \\ 0 \end{bmatrix} + \begin{bmatrix} w(k) \\ w_b(k) \end{bmatrix}. \quad (6.48)$$

For the constant bias model we use $w_b(k) = 0$ and for the random walk bias we assume that $w_b(k)$ is a zero mean Gaussian white noise, with covariance $Q_b = 1$.

The measurement equation of the extended model is:

$$y(k) = \begin{bmatrix} C_0 & \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} x_{red}(k) \\ x_b(k) \end{bmatrix} + D_0 u(k) + y_0 + v(k). \quad (6.49)$$

Note that the extension of the dryer model using the bias states is just one possible extension. Other extensions could also have been used to test the selection procedure. However this particular extension has the advantage the the resulting model remains linear.

Now that the linear dryer model has been extended with both models for the bias state $x_b(k)$, the state of the system along with the bias state can be estimated using the usual Kalman filter equations.

The accuracy of the state estimation results will be dependent on how suitable each of the bias models is for estimating the bias state $x_b(k)$. In this simulation example we will estimate the state of the system using the following techniques:

1. A Kalman filter using the extended model in which $x_b(k) = 0.833$ is constant, so $w_b(k) = 0$.
2. A Kalman filter using the extended model in which $x_b(k)$ is modelled as a random walk, so $w_b(k)$ is a zero mean Gaussian white noise with covariance $Q_b = 1$.
3. The state of the system is estimated using the two previous Kalman filters (options 1 and 2). Then Procedure 5.8 is used to determine best estimation model.
4. A Kalman filter in which the correct values for $x_b(k)$ are given. For this filter only the original states $x(k)$ have to be estimated.

Given these four filters, the best estimation results are expected using the fourth filter, for which the sequence $x_b(k)$ is given. This filter will only serve to provide an upper limit on achievable estimation accuracy.

The estimation data consists of 1000 points generated with the linear model in which the bias state $x_b(k) = 0$ for $k < 333$, $x_b(k) = 0.25$ for $334 \leq k < 667$ and $x_b(k) = 0$ for $k \geq 667$ (see also Figure 6.8).

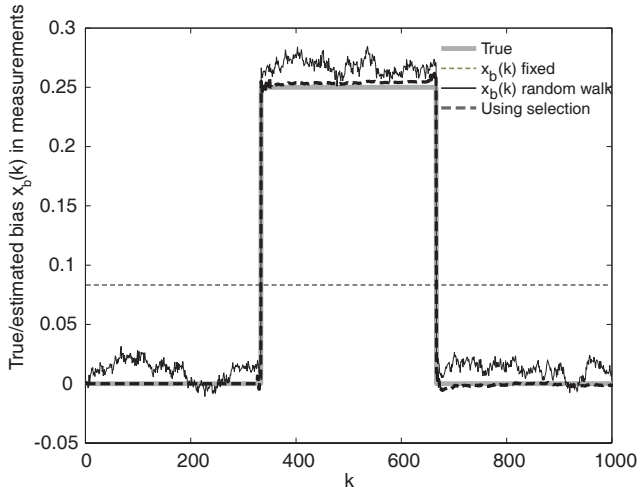

Figure 6.8 Estimated values for the bias state $x_b(k)$ in (6.48) and (6.49). The model for the bias state is either a constant (see 6.46), a random walk (see 6.47) or both models are used to estimate the state of the system and selection Procedure 5.8 is used to determine the optimal model online.

The quality of the estimated estimated states will be assessed using the following error criterion:

$$\mathcal{E} = \sum_{k=1}^{1000} \frac{1}{1000} (\hat{x}(k|k) - x(k))^T (\hat{x}(k|k) - x(k)). \tag{6.50}$$

The results are averaged over 20 simulation runs. The results for each of the estimation scenarios is summarized in Table 6.8.

The estimation results using a fixed model for $x_b(k)$ are very poor. Ignoring the change in the bias state $x_b(k)$ results in a large bias in the estimation results of $x(k)$.

The results using a random walk model for $x_b(k)$ are better than when using a fixed

Table 6.8 Average state estimation errors. The state of the system has been estimated using either a fixed bias model (6.46), a random walk bias model (6.47) or using both fixed and random walk model and using the model selection Procedure 5.8 to determine which model to use for state estimation. For reference the results of an optimal filter for which the values of $x_b(k)$ were given are also included.

$x_b(k)$ fixed	$x_b(k)$ random walk	Selection	Optimal
3539	86.4	30.7	14.2

model for $x_b(k)$. However the estimation error using the random walk model is still quite large compared to the results of the optimal filter for which the exact sequence $x_b(k)$ was given. The additional error using the random walk model can be explained if we consider Figure 6.8, in which the estimation results for the bias state $x_b(k)$ have been plotted. Although the bias in the estimates appear reduced compared to the fixed $x_b(k)$ model, they exhibit a large variance. The variance causes the additional error in the state estimation results.

The results using the selection procedure are better than those of both other options and are in fact relatively close to the results of the filter for which the exact sequence $x_b(k)$ is given. This is because the model selection procedure uses the fixed model for the regions in which the bias state is constant, but uses the random walk model when the bias state jumps from 0 to 0.25 (and back). In this manner it thus combines the low variance of the fixed model with the flexibility of the random walk model. This is confirmed in Figure 6.8. Figure 6.8 depicts the estimation results for $x_b(k)$ of the various filters. The selection model indeed only uses the random walk model when $x_b(k)$ changes, and the fixed parameter model at other times.

6.8 Summary and conclusions

In this chapter we presented a simulation study in which the applicability of the various techniques presented in the previous chapters is tested using a practical example. For this simulation study a nonlinear model of the dryer section of a paper machine has been used.

The dryer model was supplied by TNO Science and Industry. The model structure, its main equations and its properties have been discussed in section 6.2. The main properties of the dryer model are:

- Model order is equal to 57.
- The CPU time per model evaluation is approximately 48 seconds on a 1400 MHz AMD Athlon computer.
- The sample time of measurements is 60 seconds.
- The model is (mildly) nonlinear in operating regime.

In section 6.3 we use the POD model reduction technique to reduce the order of the model. We managed to reduce the order of the dryer model from 57 to 28, but, as expected, the CPU time per model evaluation has not been reduced.

The dryer model requires too much CPU time per model evaluation to be used for online nonlinear state estimation. An approximate model for the dryer model that can be evaluated much faster is constructed in section 6.4. Specifically, a model that has a qLPV model structure is identified. The resulting qLPV model is a good approximation of the dryer model. Compared to the accuracy of a linear approximation to the dryer model, the accuracy of the qLPV model is only 6-10% better. It is argued that

the main reason for the limited increase in accuracy is that the dryer section model is almost linear, such that big improvements were not to be expected.

In section 6.5 we tested the techniques introduced in chapter 4 for constructing a Kalman filter without covariance information. Techniques are tested using a linear approximation of the dryer section model. From the singular values of its observability matrix it is clear that this linear model is poorly observable. As a result a direct application of Mehra's covariance method results in poor state estimates. Better results are obtained if a Kalman filter is constructed using only the state directions corresponding to the largest singular values of the observability matrix, as described in the improved technique introduced in section 4.3.

In section 6.6 the state of the dryer section model is estimated using the reduced order model, the identified qLPV model and a linear model of the dryer section. Using each of the models, it is possible to accurately estimate the state of the system. It is however important to note that the state estimates using the qLPV model or the linear model can be computed approximately 10^4 times faster than when the reduced order dryer model is used.

The linear dryer model is extended with additional bias states in section 6.7. To estimate the bias states two candidate models are given. The model selection technique described in chapter 5 is used to determine online which model is optimal for state estimation. Results using model selection are better than the results of each of the candidate models alone.

As a final conclusion for this simulation study, we would like to note that, even though we obtained satisfying results, the relatively low complexity of the dryer section model has not allowed us to demonstrate the full extent of the methods developed in this thesis. Indeed, the results for the model reduction would have been more illustrative if the order of the dryer model had been higher. If the model nonlinearities had been more pronounced, it would have constituted a better test for the qLPV approximation technique.

Chapter 7

Conclusions and recommendations

7.1 Conclusions

This thesis has aimed to contribute to the development of computationally feasible methods for the efficient use of large scale physical models in model based monitoring, fault detection and control of industrial processes.

In the introduction we have identified four properties of large scale first principles models that prevent the development of such methods. These properties are:

- The state dimension of models is very large.
- The simulation time of models is of the same order as the sampling interval.
- The models often lack description of disturbances and measurement noise.
- The models may need to be recalibrated during operation.

This thesis has addressed each of these obstacles. The main results and conclusions of the thesis are summarized below.

Since the goal of the thesis mainly pertains monitoring, the first section of Chapter 2 contains a survey of currently used state estimation methods for nonlinear systems. The main conclusions of this survey are:

- State estimation techniques for nonlinear systems can be divided into four main categories: techniques based on the Extended Kalman Filter equations, techniques that approximately implement the Best Linear Unbiased Estimate (BLUE), techniques based on the least squares formulation of the state estimation problem and techniques that approximately compute the Bayes estimate.
- All state estimation methods for nonlinear methods will generally require many model evaluations.

To address the problem of models with large state dimensions, the second part of chapter 2 describes two projection based model reduction techniques from the literature: Proper Orthogonal Decomposition (POD) and Balancing. The main conclusion from this section is:

- Projection based model reduction techniques such as Balancing or POD can often reduce the order of a large scale first principles model by orders of magnitude.

Chapter 3 addresses the problem that the simulation time per model evaluation of a first principles process model can be of the same order of magnitude as the sampling interval. This prevents the use of such models for online state estimation. This problem can be solved by introducing methods that can be used to generate a new model, approximating the behavior of the original model, but requiring far less simulation time per model evaluation. Main results and conclusions of this chapter are:

- For nonlinear systems, model reduction alone does not necessarily result in a reduced simulation time per model evaluation. (For linear systems the simulation time per model evaluation is automatically reduced by model reduction.)

- We introduced a method that approximates a nonlinear model by only computing the state update for a part of the state vector. A linear estimator is used to approximately compute the state update for the remaining states. It was shown that the accuracy of the states computed using the linear estimator can be improved by using the averaged spatial covariance (the covariance between different elements of the state vector at the same time instant) and temporal covariances (the covariance between current states and previous states and inputs).

- We introduced a second technique that approximates the known (reduced order) first principles model with a model that has the quasi-LPV model structure. The quasi-LPV model is identified using data generated by the first principles model. It was shown that the component linear models for this qLPV model structure can be efficiently computed using a singular value decomposition.

Chapter 4 considers the problem of designing a Kalman filter when the normally required covariance information is not available, and the available model is poorly observable. A system is called poorly observable when the observability matrix is near singular. We consider Mehra's covariance method which identifies the lacking information using input-output data collected from the real-life system. The main results for this chapter are:

- After an analysis of Mehra's direct covariance method it has been shown that the method can be sensitive to estimation errors induced by the identification of the missing statistical properties. Poor results are especially likely for poorly observable systems (systems for which the observability matrix is nearly singular).

- We introduced an improvement to Mehra's covariance method that is more robust to estimation errors. The improved method achieves this by only using those state directions for filter design, that correspond to the largest singular values of the observability matrix. For the remaining state directions a separate estimator has been developed.

During monitoring it may be necessary to recalibrate the process model. The problem of determining when to recalibrate the process model can be considered a model selection problem. Chapter 5 discussed a new method for online model selection for the purpose of state estimation. The main results of this chapter are:

- We derived a new version of the classical FPE model selection criterion for models whose parameters have been estimated by solving a weighted and regularized least squares problem.
- A procedure was developed to apply the new FPE model selection criterion to determine online the optimal model for state estimation.
- We analyzed the false alarm and detection probabilities of the new online filter model selection procedure for the special case of linear models.

Finally in chapter 6, the methodologies of the preceding chapters have been tested in a case study. In this case study the states of a model of the dryer section of a paper production machine were estimated.

Using the results of this thesis several problems that are associated with the development of generally applicable monitoring algorithms can thus be overcome. As mentioned above, computational issues have been addressed in chapters 2 and 3. The lack of proper noise and disturbance models has been addressed in chapter 4. The problem with model recalibration can be addressed using the model selection technique introduced in chapter 5. All these results were developed without considering specific applications. Thus the results should be applicable to a wide range of process models. This is illustrated by the good results that were obtained when these techniques were applied in the heated plate simulation example and the dryer model case study in chapter 6. However, given the wide range of processes found in the process industry, the results of applying these techniques may vary. Also, the techniques presented in this thesis are not yet at the stage where they can be applied by operators that have no background knowledge of the specific techniques.

7.2 Recommendations for future work

Although the results of the thesis as described in the previous section provide solutions to some obstacles for the development of monitoring algorithms for large scale first principles models in the process industry, there are still problems that need to be solved. Besides these open problems there still is potential for improving the accuracy of some of the developed methods.

Some suggestions for further study are given below:

- The literature survey as provided in Chapter 2 revealed that currently used model reduction techniques such as POD and Balancing, do not take into account that the reduced order model is to be used for monitoring. It would be worth investigating if model reduction techniques could be developed for the specific purpose of state estimation. A first step in this direction is presented in [99].

- An important decision in state estimation for nonlinear systems is which nonlinear state estimation algorithm to use. Methodologies should be developed that aid an user to decide which estimator to use. Important parameters for such a methodology include a measure of the nonlinearity of the system, the time required per model evaluation and the sampling interval.

- In chapter 3 a method has been presented to create an approximate model by partitioning the original system (see section 3.3). Results of this section are easily applicable if an explicit solver is used. In [4], the partitioning methods (without the use of spatial of temporal covariances) are applied to models which use implicit Euler solvers. It should be relatively easy to adapt the methods that include covariance information such that they too can be applied to systems which use implicit solvers.

- The structure of the qLPV model used to approximate the reduced order first principles model in section 3.5 is a linear combination of state-space models. Using linear combinations of state-space models has the drawback that the resulting model can be unstable, even if all the component models themselves are stable. This could possibly be avoided if the qLPV model would consist of a linear combination of orthogonal basis functions instead of state-space models. The component orthogonal basis functions could still be constructed using a SVD.

- The main problem in the qLPV model approximation procedure is to choose an appropriate structure for the scheduling functions $\phi_i(x_{red}(k), u(k), \theta_i)$. It would be worth investigating if the scheduling functions can be estimated more easily. A possible option for determining the scheduling functions $\phi_i(x_{red}(k), u(k), \theta_i)$ is to try and rewrite the qLPV as a LFT model. If possible, we could determine the scheduling functions using techniques such as described in [39], where the static nonlinearities of a LFT model structure are identified when all linear elements are known.

- Instead of determining fixed scheduling functions $\phi_i(x_{red}(k), u(k), \theta_i)$, it might also be possible to model the behavior of the scheduling functions as extra states, which are modelled as a stationary stochastic processes. These extra states could be estimated along with the original states of the model.

- In chapter 4 it is shown that if the improved version of Mehra's covariance method is used to construct a Kalman filter, the number of observable directions have to be distinguished from the nearly unobservable directions. Choosing the

optimal number of observable directions is not an easy task and a wrong choice can have a large effect on the accuracy of the resulting filter. It would be worth investigating if it is possible to determine the optimal number of directions l in an objective manner.

- For the covariance method, there is no explicit relation between the identification errors in the required covariance matrices on the one hand and the performance of the constructed Kalman filter on the other hand. If such an explicit link would be available, better methods to construct a Kalman filter could likely be derived.

- In chapter 6 the methodologies proposed in this thesis have been tested in a case study. In the case study, a model of a dryer section of a paper machine has been used. Due to the properties of this model (model order 57, almost linear behavior in operating region), the full capabilities of methods such as model reduction and approximating a known process model with a qLPV model structure could not be fully tested. For these methods, a more challenging case study is advisable.

Bibliography

[1] H. Akaike. Fitting autoregressive models for prediction. *Ann. Inst. Stat. Math.*, 21:243–247, 1969.

[2] H. E. A. Akker and R. F. Mudde. *Fysische Transportverschijnselen I*. Delftse Universitaire Pers, Delft, The Netherlands, 1996.

[3] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, N.J., 1979.

[4] P. Astrid. *Reduction of Process Simulation models - a Proper Orthogonal Decomposition Approach*. PhD thesis, Eindhoven University of Technology, 2004.

[5] P. Astrid, L. Huisman, S. Weiland, and A.C.P.M Backx. Reduction and predictive control design of a computational fluid dynamics model. In *Proceedings of 41st IEEE Conference in Decision and Control*, Las Vegas, 2002.

[6] R. Babuška. *Fuzzy Modeling for Control*. Kluwer Academic, Boston/Dordrecht/London, 1998.

[7] B. Bamieh and L. Giarré. Identification of linear paramter varying models. *Int. J. Nonlinear Control*, 12:841–853, 2002.

[8] H.T. Banks, C.H. del Rosario, and R. C. Smith. Reduced-order model feedback control design: Numerical implementation in a thin shell model. *IEEE Trans. Autom. Control*, 45(7):1312–1324, 2000.

[9] S. Barrachina, P. Benner, E. Quintana-orti, and G. Quintana-orti. Parallel algorithms for balanced truncation of large-scale unstable systems. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 2248–2253, Seville, Spain, 2005.

[10] M. Berrada, S. Tarasiewicz, M. E. Elkadiri, and H. Radziszewski, P. A state model for the drying paper in the paper product industry. *IEEE Transactions on Industrial Electronics*, 44(4):579 – 586, 1997.

[11] R. Bos, X. Bombois, and P. M. J. Van den Hof. Accelerating simulations of first principle models of complex industrial systems using quasi linear parameter varying models. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 4140–4145, Seville, Spain, 2005.

[12] R. Bos, X. Bombois, and P. M. J. Van den Hof. Designing a Kalman filter when no noise covariance information is available. In *Proceedings of 16th IFAC World Congress*, Prague, 2005.

[13] R. Bos, X. Bombois, and P.M.J. Van den Hof. Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations. In *Proceedings of American Control Conference 2004*, pages 3705–3710, Boston, 2004.

[14] R. Bos, X. J. A. Bombois, and P. M. J. Van den Hof. On model selection for state estimation for nonlinear systems. In *Proceedings of 7th International Symposium on Dynamics and Control of Process Systems*, Boston, 2004.

[15] P .M. T Broersen. Finite sample criteria for autoregressive order selection. *IEEE Trans. Signal Processing*, 48(12):3550–3558, December 2000.

[16] P. M. T. Broersen and H. E. Wensink. On the penalty factor for autoregressive order selection in finite samples. *IEEE Trans. Signal Processing*, 44(3):748–752, March 1996.

[17] P.M.T. Broersen. MATLAB toolbox ARMASA. `http://www.tn.tudelft.nl/mmr`, 2001.

[18] B. Carew and P.R. Bélanger. Identification of optimum filter steady-state gain for systems with unknown noise covariances. *IEEE Trans. on Automatic Control*, 6(18):582–587, 1973.

[19] A. Caya, J. Sun, and C. Snyder. A comparison between the 4DVAR and the ensemble Kalman filter techniques for radar data assimilation. *Monthly Weather Review*, 133(11):3081–3094, 2005.

[20] Y. Chetanouani. Fault detection by using the innovation signal: application to an exothermic reaction. *Chemical Engineering and Processing*, 43, 2004.

[21] P. G. Cizmas, A. Palacios, T. O'Brian, and M. Syamlal. Proper-orthogonal decomposition of spatio-temporal patterns in fluidized beds. *Chemical Engineering Science*, 58:4417–4427, 2003.

[22] G. De Nicolao and G. Ferrari Trecante. Consistent identification of NARX models via regularization networks. *IEEE Transaction on Automatic Control*, 44(11):2045–2049, 1999.

[23] P. M. Derusso, R. J. Roy, and C. M. Close. *State Variables for Engineers*. John Wiley and Sons, New York, 1965.

[24] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[25] J. Dunik, M. Šimandl, O. Straka, and L. Král. Performance analysis of derative-free filters. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 1941–1946, Seville, Spain, 2005.

[26] A. Ern and J. Guermond. *Theory and Practice of Finite Elements*. Springer, New York, 2004.

[27] A. B. Etemoglu, M Can, A. Avci, and E. Pulat. Theoretical study of combined heat and mass transfer process during paper drying. *Heat and Mass Transfer*, 41:419–427, 2005.

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

[29] P. M. Frank. Fault diagnostics in dynamic systems using analytical and knowledge-based redundancy – a survey and some new results. *Automatica*, 26(3):459–474, 1990.

[30] N. Ganesh and L. T. Biegler. A robust technique for process flowsheet optimization using simplified model approximations. *Computers and Chemical Eng.*, 11(6):553–565, 1987.

[31] G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins Univ. Press, Baltimore, USA, second edition, 1989.

[32] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gausian bayesian state estimation. *IEE Proceedings-F*, 40(2):107–113, 1993.

[33] F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley & Sons, Chichester, United Kingdom, 2000.

[34] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, 2002.

[35] A. W. Heemink and A. J. Segers. Modeling and prediction of environmental data in space and time using Kalman filtering. *Stochastic Environmental Research and Risk Assesment*, (16):225–240, 2002.

[36] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg, editors. *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer Verlag, New York, 2005.

[37] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1996.

[38] I. Hoteit, Dinh-Tuan Pham, and J. Blum. A simplified reduced order Kalman filtering and application to altrimetric data assimilation in the tropical Pacific. *J. of Marine Systems*, (36):101–127, 2002.

[39] K. Hsu, T. Vincent, C. Novara, M. Milanese, and K. Poolla. Identification of nonlinear maps in interconnected systems. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 6430–6435, Seville, Spain, 2005.

[40] L. Huisman and S. Weiland. Identification and model predictive control of an industrial glass-feeder. In *Proceedings 13th IFAC Symposium on System Identification*, pages 1685–1689, Rotterdam, 2003.

[41] L. P. B. M. Janssen and M. M. C. G. Warmoeskerken. *Transport phenomena data companion*. Delftse Uitgevers Maatschappij, Delft, The Netherlands, 1987.

[42] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, Inc., New York and London, 1970.

[43] T. A. Johansen and B. A. Foss. Identification of non-linear system structure and parameters using regime decomposition. *Automatica*, 31(2):321–326, 1995.

[44] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the non-linear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control*, 45(3):477–482, 2000.

[45] V. Kadirkamanathan, P. Li, M. H. Jaward, and S. G. Fabri. Particle filtering-based fault detection in non-linear stochastic systems. *International Journal of Systems Science*, 33(4):259–265, 2002.

[46] T. Kailath, A.H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, Upper Saddle River, 1999.

[47] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal Basic Engineering, Transactions ASME*, 82(1):35–45, 1960.

[48] M. Karlsson and S. Stenström. Static and dynamic modeling of cardboard drying part 1: Theoretical model. *Drying Technology*, 23(1-2):143–163, 2005.

[49] M. Kirby. *Geometric Data Analysis*. Wiley and Sons, New York, 2001.

[50] W.M.A. Kruf and W. J. Coumans. *Fysische transportverschijnselen bij het papier- en droogproces, Appendix bij: Een rekenmodel voor de droogpartij*. TNO-TPD, 1994.

[51] S. Lall, J. E. Marsden, and S. Glavaski. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal on Robust and Nonlinear Control*, 12:519–535, 2002.

[52] W. E. Larimore. System identification, reduced order filters and modeling via the canonical variate analysis. In *Proceedings 1983 American Control Conference*, pages 445–451, Piscataway, NJ., 1983.

[53] F-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observartions: Theoretical aspects. *Tellus*, 37A:309–322, 1986.

[54] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *J. of Dynamic Systems, Measurement and Control*, 121:71–78, 1999.

[55] T.H. Lee, W. S. Ra, T.S. Yoon, and J. B. Park. Robust Kalman filtering via Krein space estimation. *IEE Proceeding Control Theory Applications*, 151(1):59–63, 2004.

[56] L. Ljung. *System Identification-Theory for the User*. Prentice Hall, Upper Saddle River, second edition, 1999.

[57] L. Ljung. System identification - errata list.
. `http://www.control.isy.liu.se/~ljung/sysid/errata`, 2005.

[58] H. Löffler and W. Marquardt. Order reduction of non-linear differential process models. *Journal of Process Control*, 1(1):32–40, 1991.

[59] D. G. Luenberger. *Introduction to Dynamic Dystems; Theory, Models and Applications*. Wiley, New York, 1979.

[60] H. V. Ly and H. T. Tran. Modelling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer modelling*, 33(1):223–236, 2001.

[61] J. M. Maciejowski. *Predictive Control: With Constraints*. Pearson Education, Harlow, U.K., 2002.

[62] D.T. Magill. Optimal adaptive estimation of samples stochastic processes. *IEEE Trans. On Automatic Control*, 10(4):434–439, 1965.

[63] W. Marquardt. Nonlinear model reduction for optimization based control of transient chemical processes. *AIChE Symp. Ser. 326*, 98:12–42, 2002.

[64] W. Marquardt and H. Auracher. An observer-based solution of inverse heat conduction problems. *Int. J. Heat Mass Transfer*, 33(7):1545–1562, 1990.

[65] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7, 1971.

[66] R.K. Mehra. On the identification of variances and adaptive kalman filtering. *IEEE Trans. on Automatic Control*, 15(2):175–184, 1970.

[67] R.K. Mehra. Approaches to adaptive filtering. *IEEE Trans. on Automatic Control*, 17(5):693–698, 1973.

[68] P. Moin. *Fundamentals of Engineering Numerical Analysis*. Cambridge Univesity Press, Cambridge, U.K., 2001.

[69] A. A. Mood, F. A. Graybill, and D. C Boes. *Introduction to the Theory of Statistics*. McGraw-Hill, Tokyo, 1974.

[70] R. Murray-Smith and T. A. Johansen, editors. *Multiple Model Approaches to Modelling and Control*. The Taylor and Francis Systems and Control Book Series. Taylor and Francis, 1997.

[71] A. Newman and P. S. Krishnaprasad. Nonlinear model reduction for rtcvd". In *IEEE Proceedings 32nd Conf. Information Sciences and Systems*, Princeton, 1998.

[72] B. Ninness and G. C. Goodwin. Estimation of model quality. *Automatica*, 31(12):1771–1797, 1995.

[73] M. Nørgaard, N. K. Poulsen, and O. Ravn. New developments in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638, 2000.

[74] E. S. Page. Continous inspection schemes. *Biometrika*, 41, 1954.

[75] I. R. Peterson and D. C. McFarlane. Optimal guaranteed cost control and filtering for uncertain linear systems. *IEEE Transactions on Automatic Control*, 39(9):1971–1977, 1994.

[76] W. H. Press, S. A. Teukolsky, T. V. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1992.

[77] M. B. Priestley. *Spectral Analysis and Time Series*. Ac. Press, London, 1981.

[78] C. V. Rao, B. R. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 48(2):246–258, 2003.

[79] D. G. Robertson and J. H. Lee. A least squares formulation for state estimation. *J. Process Control*, 5(4):291–299, 1995.

[80] O. Rosen and R. Luus. Evaluation of gradients for piecewise constant optimal control. *Computers and Chem. Eng.*, 15(4):273–281, 1991.

[81] Wilson J. Rugh. *Linear System Theory*. Prentice-Hall, Upper Saddle River, New Jersey, second edition, 1996.

[82] M. Sanjeev Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[83] J. M. A. Scherpen. Balancing for nonlinear systems. *Systems and Control Letters*, pages 143–153, 1993.

[84] F. H. Schlee, C. J. Standish, and N. F. Toda. Divergence in the Kalman filter. *AIAA Journal*, 5:1114–1120, 1967.

[85] H. W. Sorenson. *Kalman filtering techniques*, volume 3 of *Advances in Control Systems*. Academic Press, New York, 1966.

[86] H. W. Sorenson. Least-squares estimation, from Gauss to Kalman. *IEEE Spectrum*, 7:63–68, 1970.

[87] P. Stoica and R. L. Moses. *Introduction to Spectral Analysis*. Prentice Hall, Upper Saddle River, 1997.

[88] A. Stuart and J. K. Ord. *Kendall's Advanced Theory of Statistics Volume 1 Distribution Theory*. Edward Arnold, London, 1994.

[89] R. Tempo and F. Dabbene. *Randomized algorithms for the analysis and control of uncertain systems: An overview*. Perspectives in robust control. Springer, London, 2001.

[90] Y. Theodor, U. Shaked, and de Souza. C. E. A game theory approach to robust discrete-time H_∞-estimation. *IEEE Transactions on Automatic Control*, 42(6):1486–1494, 1994.

[91] M. L. Tyler, K. Asano, and M. Morari. Application of moving horizon estimation based fault detection to cold tandem steel mill. *Int. J. Control*, 73(5):427–438, 2000.

[92] J. van den Berg. *Model Reduction for Dynamic Real-Time Optimization of Chemical Processes*. PhD thesis, Delft University of Technology, 2005.

[93] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[94] V. Verdult. *Nonlinear System Identification - A State-Space Approach*. PhD thesis, University of Twente, 2002.

[95] V. Verdult, L. Ljung, and M. Verhaegen. Identification of composite local linear state-space models using a projected gradient search. *Int. J. Control*, 75:1385–1398, 2002.

[96] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica*, 30(1):61–74, 1994.

[97] M. Verlaan and A. W. Heemink. Tidal flow forecasting using reduced rank square root filters. *Stochastic Hydrology and Hydraulics*, 5(11):349–368, 1997.

[98] E. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, Lake Louise, Canada, 2000.

[99] K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B. Bader. An optimization framework for goal-oriented model-based reduction of large-scale systems. In *Proceedings of 44th IEEE Conference on Decision and Control and European Control Conference*, pages 2265–2271, Seville, Spain, 2005.

[100] L. Xie and Y. C. Soh. Robust Kalman filtering for uncertain systems. *System and Control Letters*, 22:123–129, 1994.

[101] M-G Yoon, V. A. Ugrinovskii, and I. R. Petersen. Robust finite horizon minimax filtering for discrete-time stochastic uncertain systems. *System and Control Letters*, (52):99–112, 2004.

[102] J. Zhou and R.H. Luecke. Estimation of the covariances of the process noise and measurement noise for a linear discrete dynamic system. *Computers chem. Engng.*, 19(2):187–195, 1994.

[103] Y. Zhu. *Multivariable System Identification for Process Control*. Pergamon, Amsterdam, The Netherlands, 2001.

List of Publications

Authored or coauthored journal papers

- P. M. T. Broersen and R. Bos. Estimating time-series models from irregulary spaced data. *IEEE Trans. on Instrumentation and Measurement*, 55(4):1124-1131, 2006.

- J. Sijbers, A.J. den Dekker and R. Bos. A likelihood ratio test for functional MRI data analysis to account for colored noise. *Lecture Notes in Computer Science*, Vol. 3708, p. 538-546, 2005.

- P. M. T. Broersen, S. de Waele and R. Bos. Autoregressive spectral analysis when observations are missing. *Automatica*, vol. 40, p 1495-1504, 2004.

- P. M. T. Broersen, S. de Waele and R. Bos. Application of autoregressive spectral analysis to missing data problems. *IEEE Trans. on Instrumentation and Measurement*, 53(4):981-986, 2004.

- R. Bos, S. de Waele and P. M. T. Broersen. AR spectral estimation by application of the Burg algorithm to irregularly sampled data. *IEEE Trans. on Instrumentation and Measurement*, 51(6):1289-1294, 2002. 2001.

Authored or coauthored conference papers

- D. Poot, J. Sijbers, A. J. den Dekker and R. Bos. Estimation of the noise variance from the background histogram mode of an magnetic resonance image. In *Proceedings of SPS-DARTS 2006, the second annual IEEE BENELUX/DSP Valley Signal Processing Symposium*, pages 159-162, Antwerp, Belgium, 2006.

- J. Sijbers, A. J. den Dekker, D. Poot, R. Bos, M. Verhoye, N. Van Camp and A. Van der Linden. Robust estimation of the noise variance from background MR data. In *Proceedings of SPIE, Medical Imaging 2006: Image Processing*, Volume: 6144, p. 2018-2028, San Diego, 2006.

- R. Bos, X. Bombois and P. M. J. Van den Hof. Acceleration simulations of first principles models of complex industrial processes using quasi linear parameter varying models. In *Proceedings 44th IEEE Conference on Decision and Control and European Control Conference*, pages 4140-4145, Seville, Spain, 2005.

- R. Bos, X. Bombois and P. M. J. Van den Hof. Designing a Kalman filter when no noise covariance information is available. In *Proceedings of 16th IFAC World Congress,* Prague, 2005.

- P. M. T. Broersen and R. Bos. Estimating time series models from irregularly sampled data. In *Proceedings IEEE/IMTC Conference*, p 1723-1728, Ottawa, Canada, 2005.

- R. Bos, X. Bombois and P. M. J. Van den Hof. Accelerating large-scale non-linear models for monitoring and control using spatial and temporal correlations. In *Proceedings of American Control Conference 2004*, pages 3705-3710, Boston, 2004.

- R. Bos, X. Bombois and P. M. J. Van den Hof. On model selection for state estimation for nonlinear systems. In *Proceedings of 7th International Symposium on Dynamics and Control of Process Systems,* Boston, 2004.

- P. M. T. Broersen and R. Bos. Autoregressive order selection in missing data problems. In *Proceedings XII European Signal Processing Conference Eusipco-2004*, pages 2159-2162, 2004.

- P. M. T. Broersen and R. Bos. Estimation of time series spectra with randomly missing data. In *Proceedings IEEE/IMTC Conf.*, pages 1718-1723, Como, Italy, 2004.

- P. M. T. Broersen and R. Bos. Order selection for autoregressive spectral estimation with randomly missing data. In *Proceedings Fourth IEEE Benelux Signal Processing Symposium SPS2004*, pages 33-36, Hilvarenbeek, The Netherlands, 2004.

- P. M. T. Broersen, S. de Waele and R. Bos. Autoregressive spectral analysis with randomly missing data. In *Proceedings SYSID 2003, 13th IFAC Symp. on System Identification*, pages 1125-1131, Rotterdam, The Netherlands, 2004.

- P. M. T. Broersen, S. de Waele and R. Bos. Estimation of autoregressive spectra with randomly missing data. In *Proceedings IEEE/IMTC Conf.*, pages 1154-1159, Vail, USA, 2003.

- P. M. T. Broersen, R. Bos and S. de Waele, Spectral analysis of irregularly sampled data with autoregressive models. In *Proceedings 15th IFAC World Conference*, Barcelona, Spain, 2002.

- R. Bos, S. de Waele and P. M. T. Broersen. AR spectral estimation by application of the Burg algorithm to irregularly sampled data. In *Proceedings of IMTC 2001 Conference*, pages 1208-1213, Budapest, Hungary, 2001.

- P. M. T. Broersen, S. de Waele, and R. Bos. The accuracy of time series analysis for laser doppler velocimetry. In *Proc. 10th Int. Symp. On Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 2000.

Authored or coauthored abstracts

- D. Poot, J. Sijbers, A.J. den Dekker and R. Bos. Estimation of the noise variance from the background histogram mode of an MR image. In *25th Benelux Meeting on Systems and Control, Book of Abstracts*, Heeze, The Netherlands, 2006.
- R. Bos, X. Bombois, P. M. J. Van den Hof. Designing a Kalman Filter for Poorly Observable Systems Without Noise Covariance Information. In *24th Benelux Meeting on Systems and Control, Book of Abstracts*, Houffalize, Belgium, 2005.
- A. J. den Dekker, J. Sijbers, R. Bos and A. Smolders Brain activation detection from functional magnetic resonance imaging data using likelihood based hypothesis tests. In *24th Benelux Meeting on Systems and Control, Book of Abstracts*, Houffalize, Belgium, 2005.
- R. Bos, X. Bombois, P. M. J. Van den Hof. Model selection for state estimation. In *23rd Benelux Meeting on Systems and Control, Book of Abstracts*, Helvoirt, The Netherlands, 2004.
- R. Bos and P. M. J. Van den Hof. Model reduction for physical models using proper orthogonal decomposition and the best linear unbiased estimator. In *22nd Benelux Meeting on Systems and Control, Book of Abstracts*, Lommel, Belgium, 2003.
- R. Bos, S. de Waele and P.M.T. Broersen. Autoregressive spectral estimation by application of the Burg algorithm to irregularly sampled data. In *20-th Benelux Meeting on Systems and Control, Book of Abstracts*, Houffalize, Belgium, 2001.

Glossary

Operators

$(\cdot)^{\dagger}$	pseudo-inverse
$\mathbb{E}(\cdot)$	expectation operator
$\overline{\mathbb{E}}(\cdot)$	time averaged expectation (see (2.91))
$\mathrm{tr}(\cdot)$	trace operator
$\mathrm{vec}(\cdot)$	transforms a matrix into a vector by stacking its columns
$\mathrm{vec}^{-1}(\cdot)$	inverse $\mathrm{vec}(\cdot)$ operator, transforms vector back into matrix

Greek

$\alpha_{c\to a}$	heat conductivity coefficient between i-th dryer and air
$\alpha_{c\to p,i}$	heat conductivity coefficient between i-th dryer and paper
$\alpha_{c\to s,i}$	heat conductivity coefficient between i-th dryer and steam
$\beta(k)$	unstructured scheduling coefficient for qLPV model computed using (3.86)
γ	specific heat of the cylinder
$\phi_m(\cdot)$	scheduling function for the m-th component model of a LPV model
$\lambda(T)$	heat conductivity of heated plate model
$\mu_{a(k)}$	time average of $a(k)$
σ_i	i-th singular value
θ	parameter vector
$\Phi(k)$	autocovariance matrix of $x_s(k)$
$\Sigma(k)$	autocovariance matrix of $\hat{x}_s(k)$

Latin

c_{ADR}	correction factor for heat transfer in after dryer section (final 22 cylinders)
c_{PDR}	correction factor for heat transfer in the pre dryer section (first 32 cylinders)
d_{in}	mass fraction dry paper in the paper mixture entering the dryer section ($d_{in} = m_{pap}/(m_{pap} + m_{water})$)
d_{out}	measured mass fraction of dry paper measured after the pre- and after-dryer sections
$e(k)$	innovations sequence
$f(\cdot)$	nonlinear process model
$f_{id}(\cdot)$	identified approximate process model
$f_{red}(\cdot)$	reduced order nonlinear process model

$f_{fast}(\cdot)$	faster approximate reduced order process model
g	mass of dry material per m^2
$h(\cdot)$	nonlinear output model
h_{red}	reduced order nonlinear output model
h_{fast}	faster approximate reduced order output model
k	discrete time index
m_{cil}	mass of a dryer cylinder
m_{pap}	mass of paper
m_{water}	mass of water
$p(a\|b)$	conditional probability density function of a given b
$u(k)$	input vector at time k
$v(k)$	measurement noise at time k
v_{pap}	speed at which the paper web moves through the dryer section
$w(k)$	process noise at time k
$x(k)$	state vector at time k
$x_d(k)$	discrete part of the state $x(k)$
$x_s(k)$	stochastic part of the state $x(k)$
$\tilde{x}_{red}(k)$	approximate reduced order state, computed using Procedure 3.2
$\hat{x}(k\|l)$	estimate for state $x(k)$ using data Z^l
$x_{off,m}$	offset vector for m-th component model of a qLPV model
$x_{red}(k)$	reduced order state at time k
$y(k)$	measurement vector at time k
$y_d(k)$	discrete part of measurement vector $y(k)$
$y_s(k)$	stochastic part of measurment vector $y(k)$
A	state transition matrix of state-space model
$A_{c\rightarrow a}$	contact surface between a dryer cylinder and surrounding air
$A_{c\rightarrow p}$	contact surface between a dryer cylinder and the paper web
$A_{c\rightarrow s}$	contact surface between a dryer cylinder and steam heating the cylinder
A_m	state transition matrix of m-th component model of a qLPV model structure
B	input matrix of state-space model
B_m	input matrix of m-th component model of a qLPV model structure
C	output matrix of state-space model
D	direct feedthrough matrix of state-space model
$\mathcal{E}(\cdot)$	averaged quadratic error
$Err(\cdot)$	averaged quadratic error
$\overline{J}_p(\mathcal{M})$	FPE for model $\mathcal{M}$
$K(k)$	Kalman gain at time k
$M(k)$	cross covariance matrix between $x_s(k+1)$ and $y_s(k)$
$\mathcal{M}_i$	candidate model i for model selection
$\mathcal{O}$	observability matrix
$P_{x(k)}$	autocovariance matrix of $x(k)$
$P_{x(k)y(k)}$	cross covariance matrix between $x(k)$ and $y(k)$
$P_{steam,i}$	pressure of saturated steam that heats the cylinders in the i-th cylinder group
$Pr(a)$	probability that a occurs
$Q(k)$	autocovariance matrix of process noise $w(k)$

$R(k)$	autocovariance matrix of measurement noise $v(k)$
$R_e(k)$	autocovariance matrix of the innovations sequence $e(k)$
$R_{y_s}(i)$	autocovariance function for stochastic sequence $y_s(k)$
T	projection matrix for model reduction
T_{air}	temperature of air surrounding the dryer section
$T_{cil,i}$	temperature of the i-th dryer cylinder
$T_{pap,i}$	temperature of paper surface at the i-th cylinder
$T_{steam,i}$	temperature of steam inside the i-th cylinder
$\mathcal{V}(k)$	filter model quality at time k (see 5.18)
$\overline{V}(x,k)$	expected prediction error given x at time k (see 5.19)
Z^N	identification data $[u(1),y(1),\ldots,u(N),y(N)]$

Abbreviations

AIC	Akaike Information Criterion
BLUE	Best Linear Unbiased Estimate
CUSUM	Cumulative Sum
EKF	Extended Kalman Filter
EnKF	Ensemble Kalman Filter
FDI	Fault Detection and Isolation
FPE	Final Prediction Error
LFT	Linear Fractional Transformation
LPV	Linear Parameter Varying
MAP	Maximum a posteriori
MHE	Moving Horizon Estimator
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
qLPV	quasi-Linear Parameter Varying
RRSQRTKF	Reduced Rank Square Root Kalman Filter
SPRT	Sequential Probability Ratio Test
SVD	Singular Value Decomposition
UKF	Unscented Kalman Filter
WRLS	Weighted and Regularized Least Squares

Summary

The process industry is increasingly looking for methods to increase its efficiency. One possibility to increase the efficiency of processes is to monitor critical process variables. Often, critical process variables are often not directly measurable. In order to monitor the unmeasurable process variables, these parameters could be inferred using available detailed first principles models. Unfortunately, no monitoring techniques currently exists that can infer the process variables online, due to the complexity of the available first principles models.

This thesis has aimed to contribute to the development of computationally feasible methods for the efficient use of large scale physical models in model based monitoring, fault detection and control of industrial processes.

Four properties of large scale first principles models prevent the development of such methods. These properties are:

- The state dimension of models is very large;
- The simulation time of models is of the same order as the sampling interval;
- The models often lack description of disturbances and measurement noise;
- The models may need to be recalibrated during operation.

A literature survey of currently available state estimation techniques and projection based model reduction techniques is given in chapter 2.

Chapter 3 shows that for nonlinear models, model reduction alone does not reduce the time required to perform simulations. To reduce the required time for simulations, two new methods are introduced to generate approximate models that require less time per model evaluation. In the first method to generate a faster approximate model, the process equations are only used to compute a state update for a part of the state vector. The remaining states are then computed using linear operations. The second method to generate an approximate process model is an identification based approach. In this approach, data generated using the reduced order process model is used to identify a model that has a quasi-Linear Parameter Varying (qLPV) model structure. To identify the qLPV model, a singular value based approach is introduced that simplifies the identification of the component models.

Chapter 4 considers the problem of designing a Kalman filter when the normally required covariance information is not available, and the available model is poorly observable. A system is called poorly observable when the observability matrix is

near singular. We considered Mehra's covariance method which identifies the lacking information using input-output data collected from the real-life system. After an analysis of Mehra's direct covariance method it has been shown that the method can be sensitive to estimation errors induced by the identification of the missing statistical properties. Poor results are especially likely for poorly observable systems. We introduced an improvement to Mehra's covariance method that is more robust to estimation errors. The improved method achieves this by only using those state directions for filter design, that correspond to the largest singular values of the observability matrix. For the remaining state directions a separate estimator has been developed.

During monitoring it may be necessary to recalibrate the process model. The problem of determining when to recalibrate the process model can be considered a model selection problem. Chapter 5 discussed a new method for online model selection for the purpose of state estimation. We derived a new version of the classical FPE model selection criterion for models whose parameters have been estimated by solving a weighted and regularized least squares problem. Using the new selection criterion chapter 5 introduces a new method to determine the optimal model for state estimation online.

Finally in chapter 6, the methodologies of the preceding chapters have been tested in a case study. In this case study the states of a model of the dryer section of a paper production machine were estimated.

Using the results of this thesis several problems that are associated with the development of generally applicable monitoring algorithms can thus be overcome. All these results were developed without considering specific applications. Thus the results should be applicable to a wide range of process models. This is illustrated by the good results that were obtained when these techniques were applied in the heated plate simulation example and the dryer model case study in chapter 6. However, given the wide range of processes found in the process industry, the results of applying these techniques may vary. Also, the techniques presented in this thesis are not yet at the stage where they can be applied by operators that have no background knowledge of the specific techniques.

Samenvatting

De procesindustrie is voortdurend opzoek naar methodes om de efficiëntie van processen te verhogen. Een mogelijkheid om de efficiëntie van processen te verhogen is door het optimaliseren van de procesbesturing. Voorwaarde is dan wel dat men cruciale procesvariabelen online kan monitoren. Jammer genoeg is het vaak niet mogelijk om alle gewenste procesvariabelen online te meten. Indien het niet mogelijk is om procesvariabelen direct te meten, is het noodzakelijk om deze varabelen te schatten, door gebruik te maken van grootschalige fysische modellen. De op dit moment gangbare schattingstechnieken kunnen echter vaak niet overweg met complexiteit van grootschalige fysische modellen.

In dit proefschrift is gezocht naar rekentechnisch haalbare methodes die efficiënt gebruik maken van grootschalige fysische modellen om cruciale procesvariabelen online te kunnen schatten, ten behoeve van o.a. fout detectie en proces aansturing.

Grootschalige fysische modellen hebben in het algemeen eigenschappen die het ontwikkelen van een dergelijke methodiek moeilijk maken. Vier van deze eigenschappen zijn:

- de dimensie van een toestandvector in grootschalige fysische modellen is erg hoog;
- de tijd benodigd voor een enkele simulatie stap is vaak in de zelfde orde van grootte als de tijd tussen metingen;
- grootschalige fysische modellen bevatten geen statische beschrijving van procesverstoringen en meetruis;
- het procesmodel moet soms opnieuw gekalibreerd worden.

In hoofdstuk 2 wordt een overzicht gegeven van de huidige stand van de techniek op het gebied van online toestandsschatters en modelreductie.

In hoofdstuk 3 wordt uitgelegd dat model reductie alleen niet automatisch leidt tot een kortere rekentijd per simulatie stap. Om de rekentijd van complexe fysische modellen te reduceren worden er twee nieuwe methoden geïntroduceerd. In beide methodes worden benaderende modellen gemaakt, die veel minder rekentijd per modelevaluatie nodig hebben. Eerst wordt er een techniek gepresenteerd, waarbij het originele procesmodel alleen nog gebruikt om slechts een deel van de toestanden te berekenen. De overige toestanden worden vervolgens benaderend berekend, waarbij slechts van lineaire operaties gebruikt wordt gemaakt. Ook bij de tweede techniek wordt een

benaderend model gemaakt. Hier wordt echt gebruik gemaakt van technieken uit de systeemidentificatie. Data gegenereerd met het originele model wordt gebruikt om een model te schatten met een qLPV (quasi Lineair Parameter Varying) modelstructuur. Bij de identificatie van het qLPV model wordt een nieuwe techniek gebruikt om submodellen binnen de qLPV structuur eenvoudig te kunnen bepalen. Deze techniek is gebaseerd op een singuliere waarde decompositie op gelineariseerde modellen.

Voor het construeren van een Kalman filter op de gangbare methodes is het noodzakelijk dat de covariantie-matrices van procesverstoringen en meetfouten bekend zijn. Hoofdstuk 4 beschrijft methoden die in staat zijn om een Kalman filter te construeren zonder dat deze informatie beschikbaar is. In het bijzonder wordt er gekeken naar systemen die "nauwelijks" observeerbaar zijn. Een systeem is "nauwelijks" observeerbaar als de observeerbaarheidsmatrix van het systeem bijna singulier is. In het bijzonder wordt er gekeken naar Mehra's covariantie methode, waarmee de ontbrekende covariantie informatie kan worden geïdentificeerd uit data gegenereerd met het fysieke systeem. Uit een analyse van de Mehra's methode blijkt dat de methode gevoelig is identificatiefouten indien het systeem "nauwelijks" observeerbaar is. Behalve deze analyse wordt ook een verbeterde versie van Mehra's covariantie methode gegeven, die minder gevoelig is voor deze identificatiefouten. De verbeterde methode is minder gevoelig door eerst een filter te ontwerpen dat alleen de goed observeerbare toestanden schat. Daarna wordt een aparte schatter gegeven voor de slecht observeerbare toestanden.

Gedurende het online schatten van toestanden kan het op den duur nodig zijn om het fysische model opnieuw te kalibreren. De bepaling van het moment waarop het model opnieuw gekalibreerd moet worden, kan geschreven worden als een model selectie probleem. In hoofdstuk 5 wordt een nieuwe methode voor modelselectie voor het schatten van toestanden gepresenteerd. Eerst wordt het bestaande FPE selectiecriterium generaliseerd, zodat het ook kan worden gebruikt voor schattingen bepaald uit gewogen en geregulariseerde kleinste kwadraten problemen. Daarna wordt het gegeneraliseerde FPE criterium gebruikt om het optimale model te selecteren voor het schatten van toestanden.

In hoofdstuk 6 worden de resultaten uit de voorafgaande hoofdstukken getest in een casestudy. In deze case worden de eerder afgeleide technieken toegepast op een model van de droogsectie van een papiermachine.

Bij de afleiding van de eerder genoemde nieuwe technieken zijn nauwelijks specifieke aannamen gedaan met betrekking tot de structuur van de gebruikte modellen. Daardoor is het mogelijk om de geïntroduceerde technieken op een diverse fysische modellen. Dit blijkt uit goede resultaten die in dit proefschrift zijn verkregen bij tests met het model van een verwarmde plaat en in de casestudy van de droogsectie van een papierfabriek. In de procesindustrie worden echter vele modellen gebruikt met uiteenlopende eigenschappen. Deze uiteenlopende eigenschappen zouden ervoor kunnen zorgen dat de mate van succes per toepassing varieert. Ten slotte zijn de gepresenteerde technieken nog niet vergenoeg gevorderd dat de technieken gebruikt kunnen worden zonder enige kennis van de achterliggende methoden.

Curriculum Vitae

Robert Bos was born in Papendrecht, The Netherlands in 1977. He obtained his secondary eduction diploma (VWO) from the Willem de Zwijger College in 1995. In that same year he started to study applied physics at the Delft University of Technology. He received his MSc. degree in applied physics in 2001. For his MSc. thesis he was awarded the "study-award". His MSc. thesis was also selected to be one of the finalists in the national control award.

After obtaining his MSc., he joined the Systems, Signals and Control group at the Delft University of Technology as a PhD student. The Systems, Signals and Control group would later merge with similar groups to form the Delft Center for Systems and Control. In these groups he worked on the "Monitoring using large-scale first-principles models from the process industry" project which was sponsored by TNO Science and Industry.

Currently, he is employed at Shell International Exploration and Production as a reservoir engineer.