NONLINEAR ESTIMATION FOR MODEL BASED FAULT DIAGNOSIS OF NONLINEAR CHEMICAL SYSTEMS

A Dissertation

by

CHUNYAN QU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2009

Major Subject: Chemical Engineering

NONLINEAR ESTIMATION FOR MODEL BASED FAULT DIAGNOSIS OF NONLINEAR CHEMICAL SYSTEMS

A Dissertation

by

CHUNYAN QU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Juergen Hahn
Committee Members,	Mahmoud El-Halwagi
	M. Sam Mannan
	Alexander G. Parlos
Head of Department,	Michael Pishko

December 2009

Major Subject: Chemical Engineering

ABSTRACT

Nonlinear Estimation for Model Based Fault Diagnosis

of Nonlinear Chemical Systems. (December 2009)

Chunyan Qu, B.Eng., Zhejiang University;

M.Eng., National University of Singapore

Chair of Advisory Committee: Dr. Juergen Hahn

Nonlinear estimation techniques play an important role for process monitoring since some states and most of the parameters cannot be directly measured. There are many techniques available for nonlinear state and parameter estimation, i.e., extended Kalman filter (EKF), unscented Kalman filter (UKF), particle filtering (PF) and moving horizon estimation (MHE) etc. However, many issues related to the available techniques are to be solved. This dissertation discusses three important techniques in nonlinear estimation, which are the application of unscented Kalman filters, improvement of moving horizon estimation via computation of the arrival cost and different implementations of extended Kalman filters.

First the use of several estimation algorithms such as linearized Kalman filter (LKF), extended Kalman filter (EKF), unscented Kalman filter (UKF) and moving horizon estimation (MHE) are investigated for nonlinear systems with special emphasis on UKF as it is a relatively new technique. Detailed case studies show that UKF has advantages over EKF for highly nonlinear unconstrained estimation problems while MHE performs better for systems with constraints.

Moving horizon estimation alleviates the computational burden of solving a full information estimation problem by considering a finite horizon of the measurement data; however, it is non-trivial to determine the arrival cost. A commonly used approach for computing the arrival cost is to use a first order Taylor series approximation of the nonlinear model and then apply an extended Kalman filter. The second contribution of this dissertation is that an approach to compute the arrival cost for moving horizon estimation based on an unscented Kalman filter is proposed. It is found that such a moving horizon estimator performs better in some cases than if one based on an extended Kalman filter. It is a promising alternative for approximating the arrival cost for MHE.

Many comparative studies, often based upon simulation results, between extended Kalman filters (EKF) and other estimation methodologies such as moving horizon estimation, unscented Kalman filter, or particle filtering have been published over the last few years. However, the results returned by the extended Kalman filter are affected by the algorithm used for its implementation and some implementations of EKF may lead to inaccurate results. In order to address this point, this dissertation investigates several different algorithms for implementing extended Kalman filters. Advantages and drawbacks of different EKF implementations are discussed in detail and illustrated in some comparative simulation studies. Continuously predicting covariance matrix for EKF results in an accurate implementation. Evaluating covariance matrix at discrete times can also be applied. Good performance can be expected if covariance matrix is obtained from integrating the continuous-time equation or if the sensitivity equation is used for computing the Jacobian matrix.

To Simon

ACKNOWLEDGMENTS

Foremost, the greatest gratitude is extended to my advisor, Dr. Juergen Hahn, for his thoughtful advice and guidance. He quickly became for me the role model of a successful researcher in the field. His dedication and passion on research and education influenced me positively. His insights and perception on novel approaches as well as on issues and challenges of active research areas inspired me tremendously. He is open-minded and caring for students, which helps to make my research experience focused and fruitful. It is a great honor and pleasure to work with him.

I was also delighted to interact with Dr. Mahmoud El-Halwagi by serving as a teaching assistant for one of his classes and having him as one of my committee members. His knowledge and sense of humor are a rare combination which make me easy and make it enjoyable to work with him. I would as well like to express my appreciation to Dr. M. Sam Mannan and Dr. Alexander G. Parlos for serving on my degree committee and for providing valuable feedback on my research work. I am very much grateful to the Mary Kay O′Connor Process Safety Center for financial support and to the steering committee for comments and feedback from which I have benefited.

Dr. Yannis Dimitratos, Dr. Ross Wilcox and Mr. William Cox deserve special thanks as my boss and colleagues when I interned at the Process Dynamics and Control group at DuPont Engineering. In particular, I would like to thank Dr. Dimitratos for technical guidance and advice, thank Dr. Ross Wilcox for inspiring discussions and Mr. William Cox for valuable questions on my research work. The internship experience broadened my perspective on the practical aspects in the industry and I learned a great deal.

In addition, I am indebted to my peer colleagues, Yunfei Chu, Zuyi Huang, Yu

Zhu and Mitch Serpas for providing a stimulating and fun environment in which to learn and grow. Those great discussions and fun learning times will be memorable for years to come. I am especially thankful to Yunfei Chu. He sets an example of an outstanding student researcher for his rigor and passion on research as well as his desire for constantly knowing why and how. His solid knowledge in mathematics, controls and systems is truly commendable and his sharing and patience are deeply appreciated.

Lastly, and most importantly, I must thank my parents and Simon for their unflagging love and concern. Without their support and encouragement, this dissertation was simply impossible and I could not have gone this far. To them I dedicate my dissertation.

TABLE OF CONTENTS

CHAPTER Page

Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivation

Process monitoring as well as accurate and early fault detection and diagnosis are essential components of operating modern chemical plants as the level of instrumentation in chemical plants increases. These procedures play an essential role in reducing downtime and costs, increasing safety and product quality and minimizing the impact on the environment.

While alarm management is one form of process monitoring, the information contained in the HAZOP (Hazard and Operability) Studies is often very qualitative in nature and the exact threshold for initiating alarms are determined from past experience with the plant. Additionally, alarm management is usually performed by setting threshold for individual variables, thereby neglecting the effect of variables on one another. As a result of this, it often happens that several alarms are initiated at the same time which complicates the response to the abnormal situation. These points need to be addressed by investigating a fault diagnosis system which will be able to determine the type and location of the fault (sensor fault, process fault, actuator fault) as well as the magnitude in the presence of measurement noise and uncertainty in the model of the plant. Subsequently appropriate verification of HAZOP results and alarm thresholds could be determined.

Traditionally fault diagnosis is based on use of extra sensors, actuators, computers and software to measure, monitor or control a variable of interest. The drawbacks in this "hardware redundancy" method are obvious when cost and time of mainte-

The journal model is *IEEE Transactions on Automatic Control.*

nance and space for accommodating equipments are concerned. Additionally, root cause analysis for faults is not possible when multiple alarms are triggered.

With the rapid progress of modern computer technology and the development of powerful techniques of mathematical modeling, state estimation and parameter identification, quantitative model-based method such as analytical redundancy techniques for fault diagnosis become feasible. In addition, knowledge-based approach such as expert systems or fuzzy logic and process history-based methods such as qualitative trend analysis(QTA) or principle component analysis (PCA) also receive a high level of attention.

In the area of quantitative model-based methods, first principles model-based techniques such as Luenberger observers or Kalman filters have been extensively investigated. However, much of the work on fault diagnosis of nonlinear systems has focused on aerospace, mechanical, or electrical engineering applications. Work has to be done on the study of fault diagnosis schemes using nonlinear estimators in complex chemical plants since many important industrial processes such as high purity distillation columns, exothermic chemical reactors and batch systems can exhibit highly nonlinear behavior. Specifically, many techniques are available for designing nonlinear estimators. The question to what degree the results are affected by the approaches used for computing the values of unmeasured states and parameters has not been addressed.

B. Literature Survey

"Hardware redundancy" as early fault detection methods could be found in digital flight control systems such as the AIRBUS 320 and its derivatives [1]. Some other application areas are in safety-critical systems such as nuclear power plants. Due to the obvious cost and space constraints, however, it is sensible to attempt to use analytical or functional relationships between various process and measured variables to diagnose any abnormal event [2] [3] [4]. Within the last three decades, numerous research work and accomplishments in computer-based fault diagnosis methodologies have been published. In terms of the manner how to tackle the problem of fault diagnosis, the classification of quantitative model-based, qualitative model based and process history-based provides a good perspective to understand the different assumptions, deficiencies as well as advantages of various techniques [5] [6] [7]. Figure 1 shows the classification of fault diagnostic algorithms.

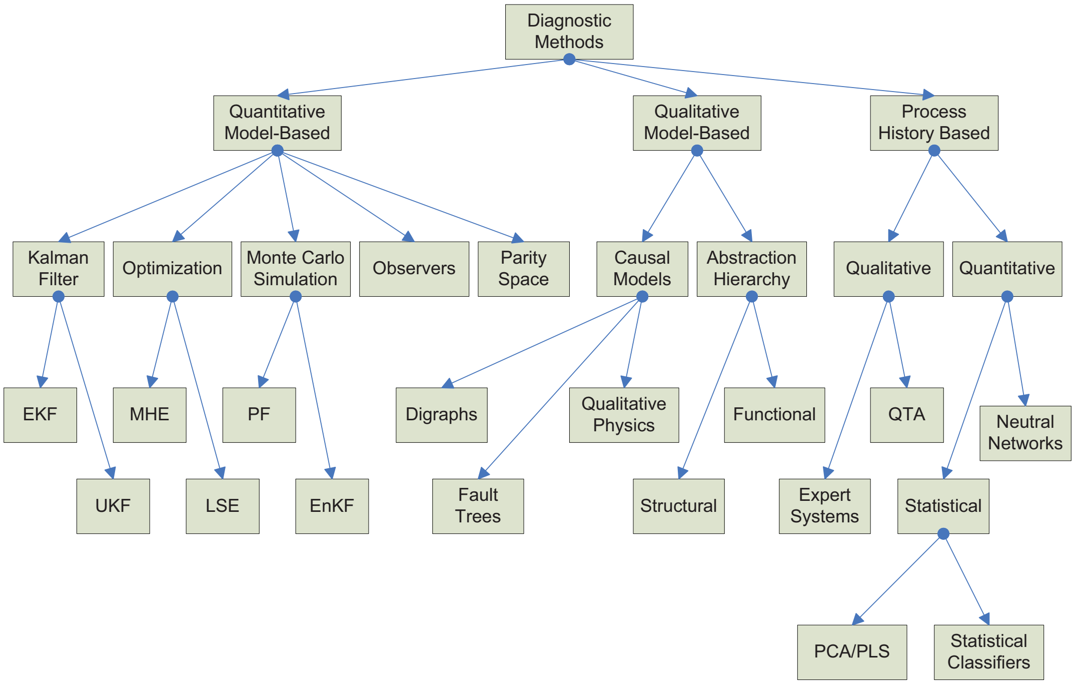


Fig. 1. Classification of Fault Diagnostic Algorithms.

In quantitative model-based (such as first principles, state-space or statistical model) approaches, the most frequently used are diagnostic observers, parity relations, Kalman filters based methods. Some earlier work using diagnostic observers approach can be found in [8] [9] [10]. A comprehensive review of first principles model based fault diagnosis using closed-loop observers or Kalman filters is provided in [11][12][13]. Frank [4] provided a solution to the fundamental problem of robust fault detection by decoupling the effects of faults from each other and from the effects of modeling errors. Diagnostic observers for nonlinear systems have also been generated in the literature. Dingli *et al.* [14] designed observers for bilinear systems. Yang and Saif [15] developed observers based on differential geometric methods for fault-affine model forms. The main concern of observer-based fault detection and identification(FDI) is to generate a set of residuals which detect and uniquely identify different faults. A major advantage of this technique is that residual's sensitivity to faults of a specific frequency range can be tailored. These residuals should be robust in the sense that the decisions are not corrupted by such unknown inputs as unstructured uncertainties like process and measurement noise and modeling uncertainties. The method develops a set of observers, each one of which is sensitive to a subset of faults while insensitive to the remaining faults and the unknown inputs.

Using mechanistic first principle models, Raja *et. al* [16] have proposed an observer-based methodology for diagnosing unknown sensor faults in systems with parametric uncertainties. However, the contribution of first principles model-based fault diagnosis approaches to industrial practice has not been pervasive due to the cost and time required to develop a sufficiently accurate process model for a complex chemical plant [17]. Therefore, Raja *et. al* [18] extended their work to sensor fault diagnosis based on subspace model, which was constructed entirely from historical process data. By performing fault reconstruction and subspace identification at different scales, model identification accuracy and faults detection, isolation and reconstruction for dynamic systems whose normal operational input-output data is known were achieved. However, both of the two methods estimate linear systems via Luenberger observer, which is not adequate in nonlinear applications due to the complexities of nonlinear chemical process.

Dynamic parity relations approach was first introduced by [19] and further explored by [20] [21] [22]. The use of short-term averages of steady state balance equation residuals was suggested by Vaclavek [23] while Almasy and Sztano [24] utilized residuals to identify gross bias faults. The idea is to rearrange model structure and to check the consistency of the plant models with sensor outputs and known process inputs [25]. In 1991, several residual generation methods including diagnostic observers, parity relations, Kalman filters in a consistent framework by [26] , which shows that parity equation and observer based design lead to identical and equivalent residual generators once the desired residual properties have been selected.

Kalman filter (KF) is the optimal estimator for linear systems subject to Gaussian noise and has been widely applied in chemical plants based on the properties that plant disturbances are random and most of the time only their statistical parameters are known. Basseville [27] has demonstrated that Kalman filters can be used for fault isolation when designed on the available process models. In practice, many physical systems exhibit nonlinear dynamics and have states subject to hard constraints, such as nonnegative concentrations and temperatures. Therefore, Kalman filtering which is designed for linear unconstrained systems is no longer directly applicable. As a result, many different types of nonlinear state estimators have been proposed. Daum [28] provides a highly readable and tutorial summary of many of these methods, and Soroush [29] reviews nonlinear estimation with a focus on applications on process control. For state estimation in a probabilistic setting, i.e., both the model and the measurement are potentially subject to random disturbances, estimate techniques such as the extended Kalman filter, unscented Kalman filter, moving horizon estimation and Bayesian estimation etc. receive much attention.

The most common application of the KF to nonlinear systems is in the form of extended Kalman filter [30] [31]. Numerous successful EKF applications have been reported in the literature [32] [33] [34] [35]. Huang *et. al* [36] reported an application of EKF-based FDI system. Mosallaei *et al.* [37] presented an integrated framework to utilize EKF data fusion algorithm for detecting and diagnosing sensor and process faults. The most famous applications of EKF are probably in Boeing 777 and Apollo moon landing [38].

Due to linearization at each time step for EKF application, large errors and divergence of the filter may occur [39][40]. Over 30 years of industrial experience also shows that EKF is difficult to implement and tune for real applications [41]. Although higher order Kalman filters exist, they are more prone to instability. Grewal and Andrews [42] proposed measures to improve numerical stability of EKF as well as Mostov [43] introduced a method to stabilize high order EKF. Chang and Hwang [44] [45] justified suboptimal filtering in fault diagnosis so that the original EKF algorithm can be more robust. Schei [46] proposed a method to improve EKF where a central difference was used to avoid explicit calculation of the Jacobian while Quine developed an implicit way to compute Jacobians [47] and a derivative-free implementation of EKF [48]. Another derivative-free state estimators based on polynomial approximations are derived by Norgarrd *et al* and this estimator performs better than estimators based Taylor approximations under certain assumptions [49]. For a class of state constraints, Ungarala and his coworkers proposed a constrained EKF for nonlinear state estimation [50].

Unscented Kalman filter (UKF) was developed to address the deficiencies of linearization by providing a more direct and explicit mechanism for transforming mean and covariance information. Julier and Uhlmann [41] describes the general unscented transformation along with a variety of special formulations that can be tailored to the specific requirements of different nonlinear filtering and control applications. A new recursive linear estimator that is not restricted to Gaussian distributions was also proposed and demonstrated by Julier and coworkers [51] [52] [53] [54] [55]. The performance of the new estimator lies between those of the modified, truncated second-order filter [56] and the Gaussian second-order filter [57]. The performance of UKF-based nonlinear filtering was evaluated by Xiong and coworkers [58]. Aguirre *et al.* used the UKF to estimate observed variables of nonlinear systems [59] and LaViola applied UKF for estimating quaternion motion [60]. Qu and Hahn [61] investigated the performance of UKF in a large number of case studies including batch reactors, mildly nonlinear CSTRs and highly nonlinear Van de Vusse reactors etc. The application of Unscented transformation was extended to nonlinear dynamic data reconciliation along with optimization strategy by Vachhani and coworkers [62]. Wan and Van de Merwe extended the use of the UKF to a broader class of nonlinear estimation problem, including nonlinear system identification, training of neural networks, and dual estimation problem [63] [64] [65]. In addition, they also explored the use of the UKF as method to improve Particle Filters [66], as well as an extension of the UKF by using a direct Bayesian update [67]. The square-root UKF for state and parameter estimation was also proposed by Van de Merwe and Wan to add benefits of numerical stability and guaranteed positive semi-definiteness of the state covariances [68]. Van de Merwe summarized UKF as one type of Sigma-Point Kalman filters in his Ph.D thesis [69]. Beyer and his coworkers applied a Sigma-point Kalman filter to batch polymerization reactors for adaptive exact linearization control [70]. Constrained state estimation using the Unscented Kalman filter was developed by Kandepu *et al* [71].

The optimal solution to the nonlinear filtering problem requires that a complete description of the conditional probability density is maintained and is infinite dimensional [72]. This exact description requires a potentially unbounded number of parameters and therefore a large number of suboptimal approaches have been developed [31] [73]. These methods usually employ analytical approximations [74] [75] [76] to probability distributions, derivatives of the state transition and observation equations, or numerical Monte Carlo methods [77] which require the use of many thousands of points to approximate the conditional density.

Particle filtering(PF), also called Monte Carlo estimation methods, does not assume a fixed shape of any probability density but approximates the densities of interest via samples or particles. PF can capture the time-varying nature of distributions commonly encountered in nonlinear dynamic problems and any moment can be computed from the sampled particles. In addition, this sampling based approach can solve the estimation problem in a recursive manner without resorting to model approximation. Marseguerra [78] showed the power of particle filtering for fault diagnosis by applying sampling importance resampling to a case study of multi-dimensional states while Li and Kadirkamanathan [79] investigated the PF based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems. T. Chen and his coworkers used particle filters for dynamic data rectification and process change detection [80] [81] and also applied PF for state and parameter estimation in batch processes [82]. Oppenheim *et al.* [83] extended the applications of PF to polymerization reactor, tracking moving bio-cell and depollution of waste water. With the additional use of heuristic optimization methods, Schwaab *et al.* showed that the *so called* particle swarm optimization method is efficient for both minimization and construction of the confidence region of parameter estimates. W. Chen and Lang and their coworkers described and illustrated Bayesian estimation via sequential Monte Carlo sampling for both unconstrained and constrained dynamic systems [84] [85].

Ensemble Kalman filter (EnKF) [86] [87], is related to the particle filter but the EnKF makes the assumption that all probability distributions involved are Gaussian; when it is applicable, it is much more efficient than the particle filter [88] [89]. The cell filter is a piecewise constant approximation of the conditional probability density of the states, whose temporal evolution is modeled by an aggregate Markov chain [90] [91]. Both EnKF and cell filter belongs to Bayesian estimation, as well as particle filters.

Although constrained EKF, UKF and PF were proposed and investigated [50] [71] [85], clipping technique were commonly used, which may result in the failure of filters on providing the accurate estimation. Moving horizon estimation (MHE) [92] [93] [94] [95] has been suggested as a practical strategy to incorporate inequality constraints in estimation building on the success of receding horizon control. The basic strategy of MHE is to reformulate the estimation problem as a quadratic program using a moving, fixed-size estimation window, which is necessary to bound the size of the quadratic program. Stability questions arise as only a subset of the data is considered for estimation. Rao *et al.* proved the stability of MHE for both linear and nonlinear constrained estimation [96][97]. Rao and Rawlings [98] also discussed the application of MHE to constrained process monitoring. Russo and Young [99] applied MHE to an industrial polymerization process while Bemporad *et al.* [100] proposed a new approach via MHE for fault detection and state estimation of hybrid systems. A MHE that evaluates the data obtained by temperature oscillation calorimetry was introduced by Mauntz and his coworkers [101]. While the power of MHE is demonstrated, the computational requirements of real time constrained optimization may make it impractical for large dimensional systems. Darby and Nikolaou [102] addressed the computational efficiency by proposing a parametric programming that bypassed real time optimization. Efficient MHE and data reconciliation via nonlinear programming were also proposed by Tenny and Rawlings [103] and Liebman and coworkers [104].

Data reconciliation is also a model-based filtering technique that attempts to reduce the inconsistency between measured process variables and a process model. Robertson *et al.* [92] showed a typical formulation of the dynamic data reconciliation problem could be presented as a special case of a more general moving horizon state estimation formulation. A technique for dynamic data reconciliation using linear balancing equations to reconcile measured states was described by Almasy [105], and using nonlinear programming techniques for reconciling nonlinear balance equations was demonstrated by Liebman [106]. Ramamurthi *et al.* [107] showed a similar moving horizon data reconciliation strategy to improve closed-loop nonlinear model predictive control performance. Dynamic data reconciliation techniques are also used to detect gross errors, identify bias in measurements and detect outliers [108] [109] [110] [111]. Bias detection and nonlinear dynamic data reconciliation was applied to a single vessel in a process by McBrayer *et al.* [112] and Soderstrom and his coworkers [113] implemented a dynamic data reconciliation application at an ExxonMobil Chemical Company plant. Later a technique that combines data reconciliation and the detection and identification of gross errors within a mixed integer optimization framework was developed by Soderstrom *et al.* [114]. Yelamos *et al.* [115] recently developed an efficient Genetic algorithm for determining time delays to enhance dynamic data reconciliation performance and demonstrated its robustness in a highly nonlinear Tenessee Eastman benchmark process. For simultaneous dynamic optimization strategies, also known as direct transcription methods for solving nonlinear estimation problem or data reconciliation, Kameswaran and Biegler [116] discussed recent advances and outlined a number of challenges.

More reviews on KF-based or optimization-based estimators are found in Chapters III, IV and V.

Under Qualitative model-based method, signed directed graph(SDG), fault trees and qualitative physics are some explored method. In contrast to quantitative method, qualitative models are usually developed based on qualitative functions centered around different units in a process [117] [118]. Then a search strategy, either topographic search or symptomatic search, is used to perform malfunction analysis. After Iri et al. [119] first used SDG for fault diagnosis, considerable research work has been done in the area. Oyeleye and Kramer [120] presented a brilliant work in the field of steady state qualitative simulation using SDG to eliminate spurious solutions without losing completeness. Chang and Yu [121] gave special attention to control loops and reported various techniques that are useful in simplifying SDGs for fault diagnosis. Another important work is the use of fuzzy set theory to improve fault resolution in SDG models by [122]. Later Shih and Lee [123] [124] discussed the use of fuzzy logic principles with SDGs for the removal of spurious solutions. Fault trees have also been developed from digraphs [125]. Ulerich and Powers [126] constructed an AND gate at each primal event to set up a fault detection tree and used the available real-time data to verify events in the fault tree. Considerable work has also been done in the area of derivation of qualitative behavior and representation of causal knowledge [127] [128] [129]. Qualitative simulation (QSIM) and qualitative process theory (QPT) [130] [131] are the popular approaches with respect to applications of qualitative models in fault diagnosis. The major disadvantage of qualitative model is the generation of spurious solutions. Moreover, these methods cannot estimate the shape and size of the fault accurately.

Process history based method extracts either qualitative or quantitative information from the available large amount of historical process data. The extracted information is then presented as a priori knowledge to a diagnostic system. It is based on the fact that more or less model errors exist no matter how accurately the available models are developed. Therefore, process history based method could avoid the inaccuracy by model mismatch. Expert systems [117] [132] and trend modeling methods [133] [134] are the two of major methods which extract qualitative history information while neural networks [135] [136] and multivariate statistics based fault diagnosis using principal component analysis (PCA) [137] [138] and partial least squares (PLS) [139] [140] [141] techniques are those methods that extract quantitative information. There are a number of papers discussing expert systems applications since Henley [142] [143] [144] initially attempted to do so. However, representation power of a specifically developed expert system is quite limited and they are difficult to update. Trend analysis can detect the fault earlier and lead to quick control but false alarms are often triggered due to change in the input level or in operating conditions. Neural networks perform well in terms of robustness to noise and isolability requirements but have weak generalization capability outside of the training data. Among the exhaustive papers on statistical methods, Qin [145] comprehensively reviews the field of statistical process monitoring methods for FDI. Multivariate statistics based methods relies on static models, which assumes that the process operates at a predefined steady-state condition. This is often not the case as the process may undergo throughput changes or exhibit highly nonlinear behavior, which result in dynamic transitions of the process variables [12].

The list of literature in fault diagnosis is far from complete. Among the numerous available methods, not a single method has all the desirable features such as quick detection, robustness, adaptability or less computational requirements. There is always a trade-off between completeness (the proposed fault set includes all the actual faults) and resolution (the fault set to be as minimal as possible) for designing a system of fault diagnosis. Developing diagnostic systems by using some of the available methods which can complement each other is intuitively appealing and practical.

C. Dissertation Overview

Chapter II reviews concepts of nonlinear estimation and several widely-known approaches. Important issues for implementation such as linearization, discretization, scaling and optimization are also described as they are the core bases for this dissertation.

Nonlinear estimation techniques play an important role for process monitoring since some states and most of the parameters cannot be directly measured. Chapter III investigates the use of several estimation algorithms such as linearized Kalman filter (LKF), extended Kalman filter (EKF), unscented Kalman filter (UKF) and moving horizon estimation (MHE) for nonlinear systems with special emphasis on UKF as it is a relatively new technique. Detailed case studies show that UKF has advantages over EKF for highly nonlinear unconstrained estimation problems while MHE performs better for systems with constraints.

Moving horizon estimation alleviates the computational burden of solving a full information estimation problem by considering a finite horizon of the measurement data, however, it is non-trivial to determine the arrival cost. A commonly used approach for computing the arrival cost is to use a first order Taylor series approximation of the nonlinear model and then apply an extended Kalman filter. In Chapter IV, an approach to compute the arrival cost for moving horizon estimation based on an unscented Kalman filter is proposed. The performance of such a moving horizon estimator is compared with the one based on an extended Kalman filter and illustrated in a case study.

Many comparative studies, often based upon simulation results, between extended Kalman filters (EKF) and other estimation methodologies such as moving horizon estimation, unscented Kalman filter, or particle filtering have been published over the last few years. However, the results returned by the extended Kalman filter are affected by the algorithm used for its implementation and some implementations of EKF may lead to inaccurate results. In order to address this point, Chapter V investigates several different algorithms for implementing extended Kalman filters. Advantages and drawbacks of different EKF implementations are discussed in detail and illustrated in a comparative simulation study.

Chapter VI summarizes findings on nonlinear estimation and thoughts for potential future work and extended research areas. The major conclusions are drawn with respect to strength and weakness of each investigated filter. The applicable or inapplicable cases or areas found in the dissertation work are also discussed. Suggestions on future work include a further study on other advanced filters such as Particle filtering or Ensemble filtering, detailed thoughts on selection of a proper filter and extensions of estimation techniques to fault diagnosis.

CHAPTER II

REVIEW OF ESTIMATION TECHNIQUES AND THEIR IMPLEMENTATIONS

This chapter reviews linear estimation and the famous Kalman filter in the first section. Section 2 provides background information for nonlinear state and parameter estimation and briefly reviews existing algorithms for nonlinear estimation, i.e., linearized Kalman filter, extended Kalman filter, moving horizon estimation. Implementation related issues are discussed in the last section as they are the core bases for enhancing the overall performance of numerical algorithms.

A. Linear Estimation

State estimation is to estimate the system states x from measurements y. Estimate x is required to model the system but is often corrupted with process noise w and y with sensor noise v. Therefore it is of interest to estimate the system states x from a set of economically or conveniently measurable variables y which are usually a subset of x. The challenge of state estimation lies in determining a good state estimate in the face of noisy and incomplete output measurements. The desired properties of estimators are

- Unbiased, i.e., expected value of the estimate is the same as that of the quantity being estimated

- Consistent i.e., estimate converges to the true value of x, as *no.* of measurement increases

- Efficient, i.e., error variance is less than or equal to that of any other unbiased estimate

Consider the linear, time invariant model with Gaussian noise

$$\begin{aligned} x_k &= Ax_{k-1} + w_{k-1} \\ y_k &= Cx_k + v_k \\ w &\sim N(0, Q), \qquad v \sim N(0, R), \qquad x_0 \sim N(x(0), p(0)) \end{aligned} \tag{2.1}$$

The parameters of the initial state distribution, $x(0)$ and $P(0)$, are usually not known and often assumed. Probability theory is used to model fluctuations in the data and to develop an optimal state estimator.

A Kalman filter is the optimal estimator for linear unconstrained systems subject to Gaussian noise. It was invented in 1960 by Rudolf. E. Kalman [146] and addresses the age-old question: How to get accurate information out of inaccurate data, i.e., How to estimate the system states x or unknown parameter p from measurements y. The first version was derived as a discrete filter. Figure 2 shows the recursive form of discrete-time Kalman filter and the mechanism of the filter approach is illustrated in Figure 3. The state estimate model is projected forward to obtain the state prediction $\hat{x}^-$. Then a state update occurs at time t_1, which updates the value of the projected state $\hat{x}^-$ to the new state $\hat{x}^+$. Finally this state is used as the initial condition to project the state estimate model to time t_2. The scheme continues forward in time, projecting the state through system models and updating the state with an available measurement. Appendix B describes the derivation of the discrete-time Kalman filter.

Kalman and Bucy extended the filter to the continuous version in 1961 [147]. The famous application of Kalman filter is in Boeing 777 and Apollo moon landing.

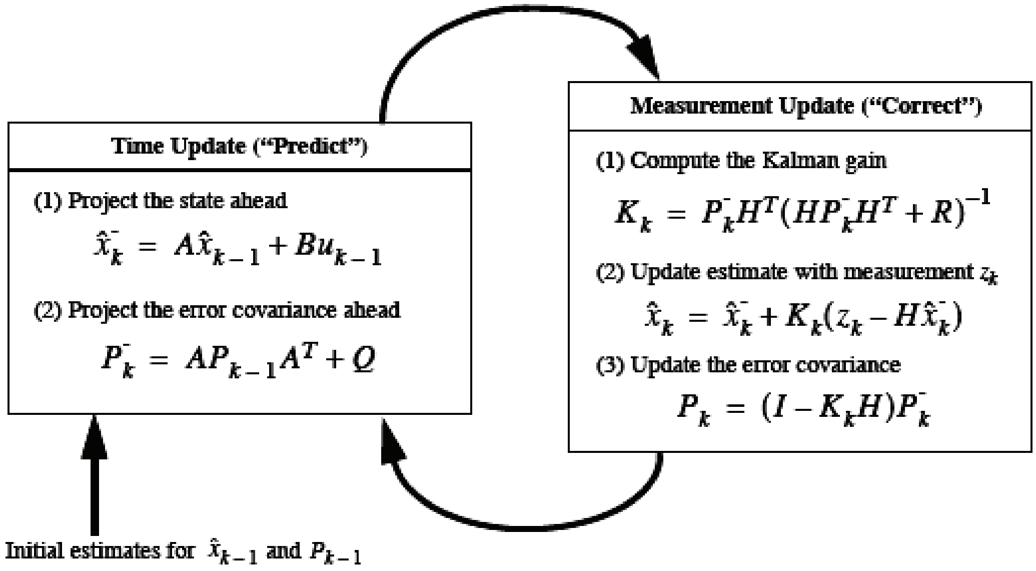

Time Update ("Predict")
(1) Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$
(2) Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update ("Correct")
(1) Compute the Kalman gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
(2) Update estimate with measurement z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Fig. 2. Kalman Filter Recursion

B. Nonlinear Estimation

1. State Estimation

A class of nonlinear systems of interest in state estimation is given by:

$$
\begin{aligned}
x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}) \\
y_k &= h(x_k, u_k, v_k)
\end{aligned}
\tag{2.2}
$$

where $x_k \in \mathbb{R}^n$ is a vector of the state variables, the functions f and h are differentiable functions of the state vector x, $w_k \in \mathbb{R}^n$ is a vector of plant noise, with $E[w_k] = 0$ and $E[w_k w_k^T] = Q_k$; $y_k \in \mathbb{R}^m$ is a vector of the measured variables and $v_k \in \mathbb{R}^m$ is a vector of measurement noise, with $E[v_k] = 0$ and $E[v_k v_k^T] = R_k$; n is the number of states, m refers to the number of measurement variables. The distributions of w and v are not necessarily Gaussian. The initial value x_0 may be assumed to be a Gaussian random variable with known mean and known $n \times n$ covariance matrix P_0.

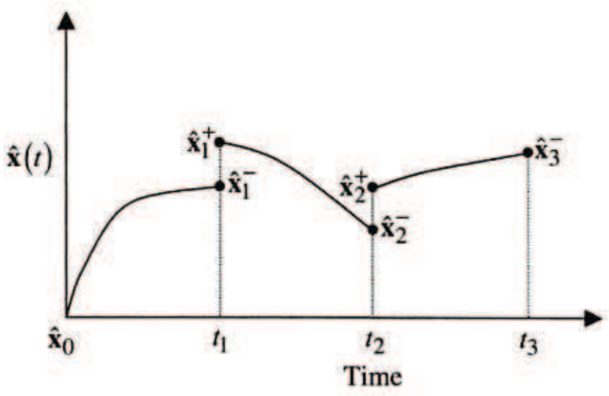


Fig. 3. Mechanism for Kalman Filter

The objective is to find an estimate $\hat{x}_k$ of x_k to minimize the weighted mean-squared error $E(x_k - \hat{x}_k)M(x_k - \hat{x}_k)^T$, where M is any symmetric nonnegative definite weighting matrix. If all estimates weight equally, the objective becomes to minimize the error covariance matrix for an unbiased estimator, given by $P = E(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T$. More specifically, the trace of P is chosen to be minimized resulting in the performance index $J = \frac{1}{2}\mathbf{Tr}[E(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$.

2. Parameter Estimation

Parameter estimation involves a nonlinear mapping of the form:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, w_{k-1}, \theta_k) \\ y_k &= h(x_k, u_k, v_k, \theta_k) \end{aligned} \tag{2.3}$$

where θ_k is a vector parameterizing the nonlinear function f. The description of θ_k corresponds to a stationary process with identity state transition matrix, driven by process noise w_{k-1}.

One technique for estimating parameters is to augment the state vector with the parameters to be estimated: $z_k = [x_k^T \ \theta_k^T]^T$. The estimation of both states and parameters can be done recursively by writing the state-space representation as :

$$z_k = \begin{pmatrix} f(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}) \\ \theta_{k-1} + w_{k-1} \end{pmatrix}. \tag{2.4}$$

3. Linearized Kalman Filter

A linearized Kalman filter is the local solution for nonlinear estimation problems based on linearization about a nominal state value. The following equations define the discrete-time form of the LKF:

Prediction equations:

$$\begin{aligned} \hat{x}_{k|k-1} &= A(\hat{x}_{k-1|k-1} - x_0) + x_0 + Bu_{k-1} \\ \hat{y}_k &= C\hat{x}_{k|k-1} + Du_k \end{aligned} \tag{2.5}$$

Update equations:

$$\begin{aligned} P_{k|k-1} &= AP_{k-1|k-1}A^T + GQG^T \\ K_k &= P_{k|k-1}C^T(CP_{k|k-1}C^T + HRH^T)^{-1} \\ P_{k|k} &= (I - K_kC)P_{k|k-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k) \end{aligned} \tag{2.6}$$

where $A \approx \frac{\partial f}{\partial x}|_{x_0}$, $B \approx \frac{\partial f}{\partial u}|_{u_0}$, $C \approx \frac{\partial h}{\partial x}|_{x_0}$, $D \approx \frac{\partial h}{\partial u}|_{u_0}$, $G \approx \frac{\partial f}{\partial w}|_{w_0}$ and $H \approx \frac{\partial h}{\partial v}|_{v_0}$ are the matrices of the linearized system model around the norminal value of the states x_0. The matrices Q and R are the tuning parameters of the Kalman filter. Q is used as a measure of confidence in the process model while R represents a measure of confidence for the sensor readings. If the process noise or uncertainties are relatively large compared to the observation noise, then Q has large values compared to R, and vice versa. The matrix P_0 provides a measure of confidence in the knowledge of the initial states x_0. The notation involving Q, R, and P_0 also applies to other estimation methods such as EKF, UKF or MHE mentioned throughout this thesis.

It is shown from the algorithm that there are no inputs to linear approximation equations from the rest of the estimator. Therefore, the Kalman gain and error covariance can be precomputed off-line. However, the deviation of the actual values from the initial states or steady states tends to be large when input changes or state perturbations occur.

4. Extended Kalman Filter

Linearized Kalman filters assume that a process stays close to the nominal operation point. However, the values of the states can be quite different from the nominal values due to input changes. As a result, offset between state estimates and the actual values may occur for sustained input excitations. Schmidt proposed the idea of an extended Kalman filter to address some of LKF's shortcomings by linearizing the system model along a state trajectory [148].

For different models, there are different forms for EKF. For continuous-time models with discrete-time measurements, as given by

$$\dot{x}(t) = f(x(t), u(t)) + Gw(t) \tag{2.7}$$

$$y_k = h(x(t_k)) + v_k \tag{2.8}$$

$$x(0) \sim N(\bar{x}_0, P_{x_0}),\ w(t) \sim N(0, Q),\ v_k \sim N(0, R_k), \tag{2.9}$$

where $x \in \mathbb{R}^n$ is a vector of the state variables; $w \in \mathbb{R}^n$ is a vector of plant noise; $y_k \in \mathbb{R}^m$ is a vector of the measured variables and $v_k \in \mathbb{R}^m$ is a vector of measurement noise, the following equations define the continuous-discrete form of the EKF:

Prediction equations:

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}, u) \\ \dot{P} &= A(\hat{x})P + PA(\hat{x}) + GQG^{'} \\ \hat{y}_k &= h(\hat{x}(t_k), u)\end{aligned} \tag{2.10}$$

Update equations:

$$\begin{aligned}K_k &= P(t_k)H_k^{'}(H_kP(t_k)H_k^{'} + R)^{-1} \\ P_k &= (I - K_kH_k)P(t_k) \\ \hat{x}_k &= \hat{x}(t_k) + K_k(y_k - \hat{y}_k)\end{aligned} \tag{2.11}$$

where $A(\hat{x}) \approx \frac{\partial f}{\partial x}|_{\hat{x}}$ and $H_k \approx \frac{\partial h}{\partial x}|_{\hat{x}(t_k)}$ are the matrices of the linearized system model, and computed as functions of the estimate for linearization about the estimated trajectory. Lyapunov equations need to be solved at each step for computing the Kalman gain K_k and updating state estimates, $\hat{x}_k$.

Finite difference is the most commonly used method found in the literature for model discretization or computing a Jacobian matrix $A(\hat{x})$ for EKF [93][40]. The estimation errors and computation times are greatly dependent on the step size for computing the finite difference.

5. Moving Horizon Estimation

From a perspective of Bayesian theory, the constrained state estimation problem can be formulated as the solution of the following optimization problem

$$\begin{aligned}\phi_T^* &= \min_{x_0, \{w_k\}_{k=0}^{T-1}} \phi_T(x_0, \{w_k\}) \\ &= \min_{x_0, \{w_k\}_{k=0}^{T-1}} \sum_{k=0}^{T-1} v_k' R^{-1} v_k + w_k' Q^{-1} w_k + (x_0 - \hat{x}_0)' \Pi_0^{-1} (x_0 - \hat{x}_0)\end{aligned} \tag{2.12}$$

subject to

$$\begin{aligned} &x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \\ &y_k = h(x_k, u_k, v_k) \\ &x_k \in \mathbb{X},\ w_k \in \mathbb{W},\ v_k \in \mathbb{V} \end{aligned} \tag{2.13}$$

where the sets $\mathbb{X}$, $\mathbb{W}$ and $\mathbb{V}$ are constraints, $x_k := x(k; x_0, \{w_j\}_{j=0}^{k-1})$ denotes the estimate of the system (2.13) at time k when the initial state is x_0, $\{w_j\}_{j=0}^{k-1}$ is the process noise sequence and $v_k := y_k - Cx(k; x_0, \{w_j\}_{j=0}^{k-1})$. Since all of the available measurements are used to generate an estimate, the problem given by (2.12) and (2.13) is referred to the full information estimator (FIE).

There exist efficient strategies for solving the FIE, which is a nonlinear program. When the process model is stiff or has unstable dynamics, it is beneficial to perform the discretization and optimization simultaneously [149]. When the process model is linear and the constraints are polyhedral convex sets, the nonlinear program simplifies to a quadratic program, which is far less computationally complex. Regardless of the complexity of problems, real time solution to FIE is impossible to be obtained because the size of problems grows unbounded with the number of points in time considered as more data need to be processed. One strategy to make the estimation problem tractable is to bound the problem size by employing a moving horizon approach.

It is fundamental to introduce an arrival cost, resulting in the following optimization problem

$$\min_{x_0, \{w_k\}_{k=0}^{T-1}} \phi_T(x_0, \{w_k\}) = \min_{z, \{w_k\}_{k=T-N}^{T-1}} \sum_{k=T-N}^{T-1} v_k' R^{-1} v_k + w_k' Q^{-1} w_k + \theta_{T-N}(z). \tag{2.14}$$

where $\theta_{T-N}(z)$ is referred to as the arrival cost, which summarizes the effect of the data $\{y_k\}_{k=0}^{T-N-1}$ on the state x_{T-N} and makes it possible to transform the optimization problem into one of lower dimension. However, the best choice of the arrival cost remains an open issue for MHE.

For unconstrained, linear systems, the arrival cost can be expressed explicitly since the MHE optimization simplifies to the Kalman filter and its covariance update formula can be used [98]. Subject to the initial condition Π_0 and assuming the matrix Π_{T-N} is invertible, the arrival cost can then be expressed as

$$\theta_{T-N}(z) = (z - \hat{x}_{T-N})' \Pi_{T-N}^{-1} (z - \hat{x}_{T-N}) + \phi_{T-N}^*. \tag{2.15}$$

where $\hat{x}_{T-N}$ denotes the optimal estimate at time $T-N$ given all of the measurements y_k from time 0 to $T-N-1$, ϕ_{T-N}^* represents the optimal cost at time $T-N$ and Π_{T-N} is computed from the Kalman filter covariance update

$$\Pi_T = A\Pi_{T-1}A^T + GQG^T - A\Pi_{T-1}C^T(C\Pi_{T-1}C^T + HRH^T)^{-1}C\Pi_{T-1}A^T. \tag{2.16}$$

The solution to the problem described by equations (2.14) and (2.15) is the unique optimal pair $(z^*, \{\hat{w}_k^*\}_{k=T-N}^{T-1})$ and it can be integrated to yield the optimal state estimates $\{\hat{x}_k^*\}_{k=T-N+1}^{T}$, where $\hat{x}_k^* := x(k; z^*, \{\hat{w}_j^*\}_{j=T-N}^{k-1})$ denotes the optimal estimate of the system at time k when the initial state is z^* and the estimated process noise sequence is $\{\hat{w}_j^*\}_{j=T-N}^{k-1}$.

For constrained, linear systems, general analytical expressions for the arrival cost are not available. One reasonable strategy is to approximate the arrival cost by the one for the unconstrained problem. The approximation is exact when the inequality constraints are inactive. For nonlinear systems, Tenny and Rawlings estimate the arrival cost by approximating a constrained, nonlinear system as an unconstrained, linear time-varying system [103]. In their work the model functions $f(.)$ and $h(.)$ in Eq.(2.13) are supposed to be sufficiently smooth so that a first-order Taylor series approximation of the model can then be applied, i.e. $A_k := \frac{\partial f}{\partial x}|_{\hat{x}_{k|k-1}}$, $C_k := \frac{\partial h}{\partial x}|_{\hat{x}_{k|k-1}}$, $G_k := \frac{\partial f}{\partial w}|_{w_k}$ and $H_k := \frac{\partial h}{\partial v}|_{v_k}$ can be obtained. The arrival cost $\theta_{T-N}(z)$ in Eq.(2.15) can be computed by solving the matrix Riccati Eq.(2.16) subject to the initial condition Π_0. This is called the MHE problem with an arrival cost computed by EKF.

The horizon length N is a tuning parameter for MHE. As a general rule, the larger the horizon length, the more accurate the estimation results will be, however, this comes at the expense of an increase of the computational burden. A practical rule of thumb is that the length of the horizon should not be less than the number of the system states. Rao and Rawlings recommend to choose the horizon length as twice the order of the system [98].

MHE fixes the computational burden of solving FIE by considering a finite horizon of the previous measurements, however, the computational complexity of itself remains a significant research challenge. Due to advances in numerical optimization, real time solutions to small dimensional nonlinear models could be found in the work of Rawlings [103].

One way to construct an analytic expression for the arrival cost $\theta_{T-N}(z)$ is to use the Kalman filter covariance update formula from equations (2.10) and (2.11), where P is denoted as Π in MHE.

C. Implementation Issues

1. Nonlinear Dynamic Stochastic Systems

A vast majority of nonlinear models are given in continuous-time and measurements are given in discrete-time. Therefore the nonlinear continuous models with discrete measurements are of the major interest in this dissertation. Applications on nonlinear discrete-time models are straightforward and the discretization techniques mentioned in the dissertation could be easily extended and applied to continuous measurements.

The nonlinear continuous model with discrete measurements and gaussian noise can be written as

$$\dot{x}(t) = f(x(t), u(t)) + Gw(t) \tag{2.17}$$

$$y_k = h(x(t_k)) + v_k$$

$$x(0) \sim N(\bar{x}_0, P_{x_0}),\ w(t) \sim N(0, Q),\ v_k \sim N(0, R_k) \tag{2.18}$$

where $x \in \mathbb{R}^n$ is a vector of the state variables; The functions f and h are differentiable functions of the state vector x, $w \in \mathbb{R}^n$ is a vector of plant noise, with $E[w] = 0$ and $E[ww^T] = Q$; $y_k \in \mathbb{R}^m$ is a vector of the measured variables and $v_k \in \mathbb{R}^m$ is a vector of measurement noise, with $E[v_k] = 0$ and $E[v_k v_k^T] = R_k$; n is the number of states, m refers to the number of measurement variables. The distributions of w and v are Gaussian. The initial value x_0 is also a Gaussian random variable with known mean $\bar{x}_0$ and known $n \times n$ covariance matrix P_{x_0}. The sampling time for measurements is T. $x(t)$,$u(t)$ and $w(t)$ are referred to x, u and w, respectively, in the rest of the chapter unless specified.

2. Linearization

Linearization of nonlinear systems is needed in order to apply linear systems theory and solutions to nonlinear systems. The common approach is to expand a nonlinear vector function $f(x)$ in a Taylor series around some nominal point $\bar{x}$, defining $\tilde{x} = x - \bar{x}$:

$$f(x) = f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} \tilde{x} + \frac{1}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_{\bar{x}} \tilde{x}^2 + \frac{1}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_{\bar{x}} \tilde{x}^3 + \dots \tag{2.19}$$

where x is assumed a scalar. When the higher orders of $\tilde{x}$ are small, $f(x)$ can be approximated by $f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} \tilde{x}$.

Supposing that x is a vector, the Taylor expansion becomes

$$f(x) = f(\bar{x}) + D_{\tilde{x}}f + \frac{1}{2!}D_{\tilde{x}}^2 f + \frac{1}{3!}D_{\tilde{x}}^3 f + \dots \tag{2.20}$$

where the operation $D_{\tilde{x}}^k f$ is defined as

$$D_{\tilde{x}}^k f = \left(\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} \right)^k f(x) \bigg|_{\bar{x}} \tag{2.21}$$

If $f(x)$ is expanded around a point that x is closed to $\bar{x}$, then $\tilde{x}$ will be small and the higher powers of $\tilde{x}$ in Equation (5.16) will be negligible. Therefore, $f(x)$ could also be approximated by

$$f(x) \approx f(\bar{x}) + \frac{\partial f}{\partial x}\bigg|_{\bar{x}} \tilde{x} \tag{2.22}$$

3. Discretization

State estimation is almost always implemented through a digital computer while the majority of the nonlinear systems are given in continuous-time. This often requires a discretization of continuous-time systems.

For a continuous-time deterministic linear system,

$$\dot{x} = Ax + Bu \tag{2.23}$$

where x is the state vector, u is the control vector, A is the system matrix, B is the input matrix. If matrices A and B are constant, the solution to Equation (2.23) is given by

$$x(t) = e^{A(t-t_0)} + \int_{t_0}^{t} e^{A\tau} Bu(t-\tau) d\tau \tag{2.24}$$

Let $t = t_k$ at the discrete time instant k and $t_0 = t_{k-1}$ at the previous time instant $k-1$. Assuming that $A(t)$ and $B(t)$ are approximately constant in the integration interval, it is obtained that

$$x(t_k) = e^{A(t_k - t_{k-1})}x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A\tau}Bu(t_k - \tau)d\tau \tag{2.25}$$

Define $T = t_k - t_{k-1}$ to obtain that

$$x_k = e^{AT}x_{k-1} + \int_0^T e^{A\tau}Bu(T - \tau)d\tau. \tag{2.26}$$

This is a linear discrete-time approximation to the continuous-time dynamics given in Equation (2.23). It is called accurate discretization as compared to the method of Euler approximation, where

$$x_k = (I + AT)x_{k-1} + BTu_k. \tag{2.27}$$

4. Simulation of Continuous-time Systems

As important as the discretization and linearization in the sections 2 and 3 is the simulation of continuous-time systems on a digital computer.

For a general ordinary differential equation describing nonlinear systems,

$$\dot{x} = f(x, u, t) \tag{2.28}$$

Solving for $x(t_s)$ at some user-specified value of t_s

$$x(t_s) = x(t_0) + \int_{t_0}^{t_s} f[x(t), u(t), t]dt \tag{2.29}$$

is of the interest.

Rectangular integration, trapezoidal integration or Runge-Kutta integration are often used for computing the Equation (2.29) [150]. Commercial softwares such as Matlab$^{\circledR'}$s *ode*45 can also be used for solving the integration of nonlinear function.

Simulation for stochastic continuous-time systems is not as straightforward as for deterministic systems. It is extremely difficult to simulate a continuous-time system that process noise w enters nonlinearly. For a system that w enters linearly as described by Equation (2.17), process noise w need to be added into the simulated $x(t_s)$ strategically. Such a system is approximately equivalent to the following discrete-time system:

$$\begin{aligned} x_k &= x_{k-1} + \int_{t_{k-1}}^{t_k} f[x(t), u(t), t]dt + w_k \\ y_k &= h(x_k) + v_k \\ w_k &\sim N(0, QT),\ v_k \sim N(0, R_k) \end{aligned} \tag{2.30}$$

5. Optimization

Optimization is to find the best solution to a system or process that is given with constraints. The objective function is an indicator of goodness of solution, for example, cost, yield or profit etc. Constraints are usually physical constraints, or subject to resources or specifications. Decisions variables can be adjusted for optimization.

Optimization is to develop theoretical properties such as convergence or existence from the perspective of mathematicians. To numerical analysts, the implementation of optimization methods is of their interests and therefore the ease of computations, performance and stability for efficient and practical use are their concerns. While from engineers' points of view, it is the most important to apply optimization methods or tools to solve real problem and therefore robustness and efficiency are of the main concerns. The three perspectives are of equal importance to academic researchers.

Given an optimization problem,

$$\begin{aligned} &\max f(x) && (n \text{ variables}) \\ &\text{s.t. } g_i(x) = 0, i \in E && (m \text{ equality constraints}) \\ &\quad g_i(x) \leqslant 0, i \in I && (l \text{ inequality constraints}) \\ &\quad m < n && (n - m \text{ degrees of freedom}) \end{aligned} \tag{2.31}$$

A function is defined to be linear if it could be described by a constant weighted sum of variables and a constant; otherwise it is nonlinear. A variable is continuous if any value can be employed in a specified interval. It is discrete if it is limited to a countable set of values, the choices are generally 0 and 1.

Based on the definitions, the types of optimization problems can be categorized as follows:

- An optimization model is a linear programming (LP) model if the objective function f and all constraints g are linear functions in the decision variables x and all decision variables are continuous.

- An optimization model is a nonlinear programming (NLP) model if the objective model f or any of the constraints g is a nonlinear function in the decision variables x, and all decision variables are continuous.

- An optimization model is a mixed-integer linear programming (MILP or MIP) if the objective function f and all constraints g are linear functions in the decision variables x, and there is at least one discrete variable.

- An optimization model is a mixed-integer nonlinear programming (MINLP if the objective model f or any of the constraints g is a nonlinear function in the decision variables x,, and there is at least one discrete variable.

Table I summarizes the classifications. Nonlinear programming and mixed-integer programming are commonly seen in the applications for chemical engineering.

Table I. Classifications of Optimization Algorithms

	Constraints		
		Linear	Nonlinear
Variables	Continuous	LP	NLP
	Discrete	IP	INLP
	Both	MIP(MILP)	MINLP

6. Scaling and Initial Guess

Scaling is an important factor in obtaining a solution for nonlinear estimation, where numerical methods are often used. Uniform scaling is a linear transformation that enlarges or increases or diminishes state variables. If the original differential equations are not scaled uniformly, nonlinear estimation, especially those solved by optimization, will be solved to different degrees of accuracy.

Take a nonisothermal continuous stirred tank reactor as an example. The CSTR model includes coolant jacket dynamics, where the following exothermic irreversible reaction between sodium thiosulfate and hydrogen peroxide is taking place [16]:

$$2Na_2S_2O_3 + 4H_2O_2 \rightarrow Na_2S_3O_6 + Na_2SO_4 + 4H_2O \tag{2.32}$$

A mole balance for species A (The capital letters A is used to denote the chemical compounds $Na_2S_2O_3$) and energy balances for the reactor and the cooling jacket result in the following nonlinear process model:

$$\begin{aligned}
\frac{dC_A}{dt} &= \frac{F}{V}(C_{Ain} - C_A) - 2k(T)C_A^2 \\
\frac{dT}{dt} &= \frac{F}{V}(T_{in} - T) + 2\frac{(-\Delta H)_R}{\rho c_p}k(T)C_A^2 - \frac{UA}{V\rho c_p}(T - T_j) \\
\frac{dT_j}{dt} &= \frac{F_w}{V}(T_{jin} - T_j) + \frac{UA}{V_w \rho_w c_{pw}}(T - T_j)
\end{aligned} \tag{2.33}$$

The nonlinear model without process noise exhibits multiple steady states, of which the upper steady state (i.e. $C_{Ass} = 0.019$mol/L; $T_{ss} = 384.0\ K$; $T_{js} = 371.3\ K$; $F_w = 30L/min$) is chosen as the point of operation. Clearly concentration is less than temperature by four orders of magnitude. The steady state is used as the nominal point for scaling the process variables to make them dimensionless, which turns the initial states from [0.018 382 371.3] to [0.947 0.995 1] for numerical efficiency.

Initial guess is critical to find a solution for state estimation or optimization. Good initial guess can provide faster convergence and more accurate estimates in solving estimation problems. Normally the measurements provide the best available information and can be used as initial guess for the estimates. At the subsequent time steps, initial guesses can be replaced by the estimates from the previous time step.

7. Sensitivity Analysis

Sensitivity analysis is the study of how uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input [151].

Sensitivity Analysis is common in physics and chemistry, in financial applications, risk analysis, signal processing, neural networks and any area where models are developed. It is used as a tool to ensure the quality of the modeling. There are several important classes of methods to perform sensitivity analysis:

- Local methods is the simple derivative of the output Y with respect to an input factor X_i, where the subscript i indicates that the derivative is taken at some fixed point in the space of the input.

- A sampling-based sensitivity [152] is the method in which the model is executed repeatedly for combinations of values sampled from the distribution of the input factors. Once the sample is generated, several strategies such as simple input-output scatter plots can be used to derive sensitivity measures for the factors.

  - Screening methods is a particular instance of sampling based methods. The objective is to estimate a few active factors in models with many factors [153] [154].

  - Monte Carlo filtering [155] [156] is also sampling-based and the objective is to identify regions in the space of the input factors corresponding particular values e.g. high or low of the output.

- A Bayesian approach [157] is that the value of the output Y of a factor X_i is treated as a stochastic process and estimated from the available computer-generated data points.

- Variance based methods [158] [159] [160] [161] is that the unconditional variance $V(Y)$ of Y is decomposed into terms due to individual factors and terms due to interaction among factors.

- High Dimensional Model Representations [162] [163] [164] is a particular case of the variance based methods where the output Y is expressed as a linear combination of terms of increasing dimensionality.

Local methods for sensitivity analysis is very useful for computing Jacobian matrix of a nonlinear discrete-time model that is not in an explicit form.

Consider a differential equation that describes a nonlinear continuous-time system

$$\dot{x} = f(x, u, t) \tag{2.34}$$

Using any numerical method for discretizing the continuous-time system, the following nonlinear discrete-time system is obtained:

$$x_{k+1} = F(x_k, u, t), \tag{2.35}$$

where F is computed from numerical methods and do not have an explicit mathematical expression, which poses a challenge for computing its jacobian matrix.

The jacobian matrix A of the nonlinear function F is

$$A_k = \frac{\partial F(x_k)}{\partial x_k}, \tag{2.36}$$

which is not expressed in a mathematical formula. However, it is obtained from Eq. (5.30) that

$$A_k = \frac{\partial x_{k+1}}{\partial x_k}, \tag{2.37}$$

which is a form of the sensitivity of x_{k+1} to its initials x_k.

Taking the derivative of the Eq. (2.34) with respect to x_0, it follows that

$$\frac{\partial}{\partial x_0}\dot{x} = \frac{\partial}{\partial x_0} f \tag{2.38}$$

Re-arranging the terms of ∂t and ∂x_0,

$$\frac{\partial}{\partial t}\frac{\partial x}{\partial x_0} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial x_0} \tag{2.39}$$

Denoting $\frac{\partial x}{\partial x_0}$ as A, Eq. (2.39) becomes

$$\dot{A} = \frac{\partial f}{\partial x} A \qquad\qquad A(0) = I \tag{2.40}$$

Solving Eq. (2.40) and (2.34) numerically, the Jacobian matrix A can be computed at any time instant kT, where T is the sampling interval and $A_k = A(kT)$.

CHAPTER III

PROCESS MONITORING AND PARAMETER ESTIMATION VIA UNSCENTED KALMAN FILTERING*

A. Introduction

One important aspect of process safety is detection of abnormal operating conditions. A common approach to this problem is that important states and parameters in a process are monitored and compared against their upper and lower bounds. However, some of these states and most of the parameters cannot be directly measured and instead have to be computed from plant data. This raises the question to what degree the results are affected by the procedure used for computing the values of unmeasured states and parameters.

There are many techniques available for nonlinear state and parameter estimation. Extended Kalman filters (EKF) have found wide-spread use and moving horizon estimation (MHE) is an optimization-based estimator aimed for constrained problems. Unscented Kalman filters (UKF), as recently proposed by Julier and Uhlman [41], could in theory improve upon EKF for state and parameter estimation since linearization is avoided by an unscented transformation and at least second order accuracy is provided. This last point is achieved by carefully choosing a set of sigma points, which capture the true mean and covariance of the given distribution and then passing the mean and covariances of estimated states through a nonlinear transformation. As a result UKF is capable of estimating the posterior mean and covariances accurately

* Part of this chapter is reprinted with permission from "Process Monitoring and Parameter Estimation via Unscented Kalman Filtering" by C. Qu and J. Hahn, 2009. *Journal of Loss Prevention in the Process Industries*, doi:10.1016/j.jlp.2008.07.012, Copyright [2009] by Elsevier.

to a high order. Despite UKF's potential for good performance for state and parameter estimation, only few applications in chemical engineering have been reported [94][39][40].

This chapter investigates advantages and disadvantages of UKF for nonlinear state and parameter estimation and presents a detailed comparison between several state estimation methodologies with a specific emphasis on unscented Kalman filtering as it is a relatively new technique. Each filter has been applied to a CSTR with exothermal irreversible reaction [18] for a variety of scenarios. The case studies show that for a reasonably small sampling time UKF performs as well as EKF. For some applications where a Jacobian matrix is not easy to obtain, UKF may be preferable because the derivation of a Jacobian matrix is not required. Moreover, for applications where the sampling time is rather large it was found that UKF may provide an acceptable performance whereas the same has not been true for EKF. In terms of process monitoring for systems with constraints, UKF as well as EKF have limitations for computing an estimate. However, these limitations were found to be negligible for UKF if physically realistic values of the covariance estimates are used.

This chapter is organized as follows: The UKF algorithm for nonlinear estimation is presented in Section B. Section C compares the performance of each filter for state and parameter estimation and concluding remarks are given in Section D.

B. Process Monitoring via Unscented Kalman Filter

An unscented Kalman filter is the application of the unscented transformation to recursive estimation. The main idea behind UKF is to use an unscented transformation to address the deficiencies of linearization by providing a mechanism for transforming means and covariances information. In the unscented transformation procedure, a set of weighed sigma points are deterministically chosen such that certain properties of these points (e.g., a given mean and covariance) match those of the prior distribution. These sigma points are propagated through a nonlinear mapping and then weighted means and covariances are computed. One approach to determine a set of sigma points that have the same first two moments and all higher odd-ordered central moments as the given distribution is given by the following:

1. Augment the system state vector to an $n^a = n + q + r$ dimensional vector $x^a = [x^T \; w^T \; v^T]^T$ to obtain its augmented mean and covariance,

$$\begin{aligned}\hat{x}^a_{k-1|k-1} &= \begin{pmatrix} \hat{x}_{k-1|k-1} \\ \mathbf{0^{q\times 1}} \\ \mathbf{0^{r\times 1}} \end{pmatrix} \\ P^a_{k-1|k-1} &= \begin{pmatrix} P_{k-1|k-1} & \mathbf{0^{n\times q}} & \mathbf{0^{n\times r}} \\ \mathbf{0^{q\times n}} & Q_{k-1} & P^{wv}_{k-1} \\ \mathbf{0^{r\times n}} & P^{vw}_{k-1} & R_{k-1} \end{pmatrix}\end{aligned} \tag{3.1}$$

where n is the dimension of original state vector, q and r are the dimensions of the original system and measurement noise vectors, respectively and P^{vw}_{k-1} and P^{wv}_{k-1} are the correlations between the system and measurement noise. For ease of computation, P^{vw}_{k-1} and P^{wv}_{k-1} are usually set to zero.

2. Generate a set of $2n^a + 1$ symmetric sigma points

$$\chi^a_{k-1} = \hat{X}^a_{k-1|k-1} + \begin{pmatrix} \mathbf{0} & \sqrt{(n^a+\kappa)P^a_{k-1|k-1}} & -\sqrt{(n^a+\kappa)P^a_{k-1|k-1}} \end{pmatrix} \tag{3.2}$$

where $\hat{X}^a_{k-1|k-1}$ is the expanded $n^a \times (2n^a + 1)$ matrix with $\hat{x}^a_{k-1|k-1}$ as each column. The means are the center points of the set. The reminder of the set is symmetrically located around the means with a distance of the square root of the covariances. $\kappa \in \Re$ is a parameter which determines how far the symmetrical $2n^a$ points are placed from the center point and provides flexibility to fine tune the higher order moments of the approximation. The more higher order moments are taken into account, the less the overall prediction error will be [52]. A useful heuristic is to select $n^a + \kappa = 3$ if $x(k)$ follows a Gaussian distribution.

After a set of sigma points is selected, each of them is propagated through the nonlinear model functions $f(.)$ and $h(.)$. Weighted means and covariances are then computed from the transformed set of points. In the final step, the Kalman filter gain is calculated from the covariances and the predicted states are updated based on the available measurements. This procedure results in the equations defining the unscented Kalman filter as follows:

Prediction equations:

$$\begin{aligned} \chi_k^x &= f(\chi_{k-1}^x, \chi_{k-1}^w, u_{k-1}) \\ \hat{x}_{k|k-1} &= \sum_{i=1}^{2n^a+1} W_i \chi_{i,k}^x \\ \gamma_k &= h(\chi_k^x, \chi_{k-1}^v, u_k) \\ \hat{y}_k &= \sum_{i=1}^{2n^a+1} W_i \gamma_{i,k} \end{aligned} \tag{3.3}$$

Update equations:

$$P_{k|k-1} = \sum_{i=1}^{2n^a+1} W_i[\chi^x_{i,k} - \hat{x}_{k|k-1}][\chi^x_{i,k} - \hat{x}_{k|k-1}]^T \tag{3.4}$$

$$P_{y,k} = \sum_{i=1}^{2n^a+1} W_i[\gamma_{i,k} - \hat{y}_k][\gamma_{i,k} - \hat{y}_k]^T \tag{3.5}$$

$$P_{xy,k} = \sum_{i=1}^{2n^a+1} W_i[\chi^x_{i,k} - \hat{x}_{k|k-1}][\gamma_{i,k} - \hat{y}_k]^T \tag{3.6}$$

$$K_k = P_{xy,k}P^{-1}_{y,k} \tag{3.7}$$

$$P_{k|k} = P_{k|k-1} - K_k P_{y,k} K^T_k \tag{3.8}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_k) \tag{3.9}$$

where $\chi^a = [(\chi^x)^T_{1\times n}\ (\chi^w)^T_{1\times q}\ (\chi^v)^T_{1\times r}]^T$ and W_i are weights as given by Eq. (3.10)

$$W_i = \begin{cases} \frac{\kappa}{2(n_a+\kappa)}, & \text{if i=1;} \\ \frac{1}{2(n_a+\kappa)}, & \text{otherwise.} \end{cases} \tag{3.10}$$

It is worth noting that if process noise or observation noise enter the system and measurements linearly, the system states and the covariance matrix need not be augmented as shown in Eq. (5.16). However, Q and R are then added linearly to Eq. (5.17) and (5.1), respectively.

The most computationally expensive operation of the UKF procedure corresponds to calculating the new set of sigma points at each time update. This task requires computing a matrix square-root of the state covariance matrix, $P = SS^T$. Since computing the square-root of P is an integral part of the UKF, Merwe and Wan [68] developed the square root UKF (SR-UKF), which makes use of three linear algebra techniques: QR decomposition, Cholesky factor updating and least squares computation. In the SR-UKF implementation, S is propagated directly, which avoids refactorization of P at each time step. This improved numerical algorithm can alternatively be used for UKF.

The properties that a set of sigma points can capture are not limited to the first two moments because the unscented transformation offers flexibility to allow more information than mean and covariance to be incorporated. Algorithms that capture the first four moments of a Gaussian distribution and the first three moments of an arbitrary distribution have been presented in [52] [**?**]. Lerner and Tenne also provide results for capturing higher order moments [165][166].

C. Case Studies

To compare the performance of UKF against LKF, EKF and MHE, all algorithms have been applied to a variety of models and a large number of scenarios such as different operating conditions, different tuning parameters Q and R, and different levels of process and measurement noise. This section presents three representative case studies for models with mild nonlinearity, severe nonlinearity and one for a constrained problem. The choice of these three examples highlights observations that have been made about the estimators.

50 Monte Carlo simulations were carried out for each case. Direct numerical integration was used to compute the predicted states for EKF and the transformed points for UKF in order to reduce estimation errors. Additionally, the Jacobian matrices A were directly computed from numerical integration for EKF as this leads to a significant increase in accuracy.

The performance is evaluated using the overall mean-squared error (MSE). The MSE is first averaged over all simulations for each time point and then over time to indicate the long-term behavior of each estimator and the distributions of errors over time.

1. CSTR with Exothermic Irreversible Reaction

The model for this case study is a nonisothermal continuous stirred tank reactor which involves coolant jacket dynamics, where the following exothermic irreversible reaction between sodium thiosulfate and hydrogen peroxide is taking place:

$$2\mathrm{Na_2S_2O_3} + 4\mathrm{H_2O_2} \rightarrow \mathrm{Na_2S_3O_6} + \mathrm{Na_2SO_4} + 4\mathrm{H_2O} \tag{3.11}$$

The capital letters A and B are used to denote the chemical compounds $\mathrm{Na_2S_2O_3}$ and $\mathrm{H_2O_2}$ in the following. The reaction kinetic law is reported in the literature to be [167]:

$$-r_A = k_0 e^{-E/RT} C_A C_B$$

where k_0 is the pre-exponential factor, E is the activation energy, R is the gas constant, T is the temperature, and C_A and C_B are the concentrations of species A and B, respectively. A stoichiometric proportion of species A and B in the feed stream is assumed which results in $C_B(t) = 2C_A(t)$. A mole balance for species A and energy balances for the reactor and the cooling jacket result in the following nonlinear process model:

$$\begin{aligned}
\frac{dC_A}{dt} &= \frac{F}{V}(C_{Ain} - C_A) - 2k(T)C_A^2 \\
\frac{dT}{dt} &= \frac{F}{V}(T_{in} - T) + 2\frac{(-\Delta H)_R}{\rho c_p}k(T)C_A^2 - \frac{UA}{V\rho c_p}(T - T_j) \\
\frac{dT_j}{dt} &= \frac{F_w}{V_w}(T_{jin} - T_j) + \frac{UA}{V_w \rho_w c_{pw}}(T - T_j)
\end{aligned} \tag{3.12}$$

where F is the feed flow rate, V is the volume of the reactor, C_{Ain} is the inlet feed concentration, T_{in} is the inlet feed temperature, F_w is the feed flow rate of the cooling jacket, V_w is the volume of the cooling jacket, T_{jin} is the inlet coolant temperature, c_p is the heat capacity of the reacting mixture, c_{pw} is the heat capacity of the coolant,

ρ is the density of the reaction mixture, ρ_w is the density of the coolant, U is the overall heat-transfer coefficient, and A is the area over which the heat is transferred. The process parameter values are given in the work by Rajaraman et al. [16].

The nonlinear model without process noise exhibits multiple steady states, of which the upper steady state ($C_{Ass} = 0.019$mol/L; $T_{ss} = 384.0\ K$; $T_{js} = 371.3\ K$; $F_w = 30L/min$) is chosen as the point of operation. This is also used as the nominal point for scaling the process variables to make them dimensionless. A plot of steady states C_A, T and T_j as a function of input feed rate F_w, shown in Figure 4 reveals that the system exhibits mild linearity at Point A. Although the system may exhibit severe nonlinearities at Point B where $F_w = 225L/min$, the operating temperature would be below the freezing point, which is not realistic in plant operations. Therefore the performance of each filter is investigated only over the mild nonlinear point A in the case study.

Process and observation noise are assumed to enter the system linearly. A discrete model of the form

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}) + Gw_{k-1} \\ y_k &= Cx_k + v_k \end{aligned} \tag{3.13}$$

is obtained from Eq. (4.13), where $x_k = \begin{bmatrix} C_A & T & T_j \end{bmatrix}^T$, $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ and $v_k \sim \mathcal{N}(0, R_k)$ are zero mean Gaussian noise and $\Delta t = 0.012$ is used for discretizing the systems. The third state T_j is chosen as the only measurable state in this case study.

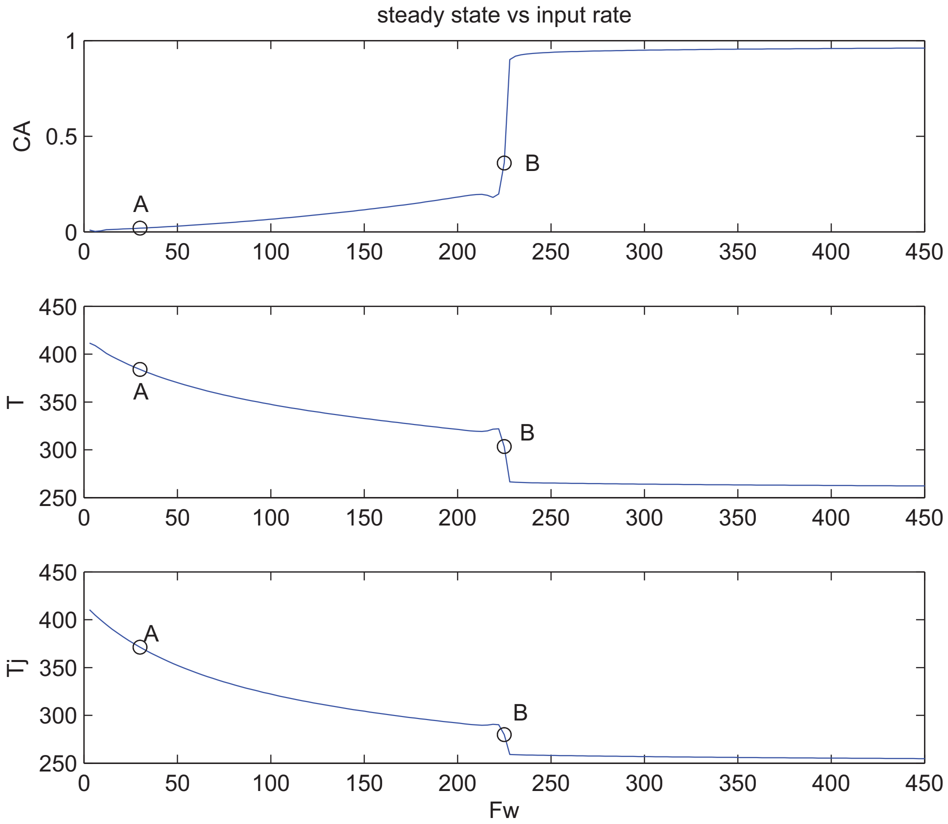


Fig. 4. Steady States as a Function of Reactor Feed Rate for the Nonisothermal Reactor.

The remaining EKF and UKF filter parameters before scaling are

$$\hat{x}_0 = \begin{bmatrix} 0.018 & 382 & 371.3 \end{bmatrix}^T, \qquad R_0 = 2.5 \qquad \text{and} \qquad \kappa = -4,$$

$$\hat{P}_0 = \begin{bmatrix} 10^{-8} & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}, \qquad Q_0 = \begin{bmatrix} 10^{-8} & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}. \qquad (3.14)$$

For fairness of comparison, κ is not further adjusted to fine tune the higher order moments of the approximation for improvement of UKF performance.

Different simulation tests are carried out with different measurement noise parameters. Table II lists the MSE values and computation times for varying measurement noise levels. The nonlinear system behavior and the performance of each filter for $R = 10^{-4}$ are shown in Figure 5.

Table II. MSEs/Computation Time by Varying Measurement Noise Levels for an Exothermic Reaction

		$R = 10^{-6}$	$R = 10^{-4}$	$R = 10^{-2}$
MSE	EKF	4.331×10^{-4}	6.187×10^{-4}	3.940×10^{-3}
	UKF	4.321×10^{-4}	6.194×10^{-4}	3.862×10^{-3}
	% Difference	0.23	0.12	2.02
Computation	EKF	2.03	2.05	2.07
Time	UKF	16.57	16.32	16.21

As shown in Table II and Figure 5, UKF results in a comparable accuracy to EKF. This may be due to system being close to a linear system around the operating point such that EKF is able to sufficiently correct its predictions from the measurements. Additionally, the kurtosis and higher order moments are negligible since the magnitudes of the covariance estimates are significantly smaller than unity. Therefore UKF may not fully utilize its potential for a mildly nonlinear scenario. Computation cost-wise, both algorithms are acceptable, even though UKF requires approximately an order of magnitude more computation time than EKF because a transformation of 15 sigma points is required at each time step compared to one integration performed for EKF.

In the case study it is assumed that the initial conditions of the unmeasured states are not precisely known. Therefore, there are offsets for the first two states by

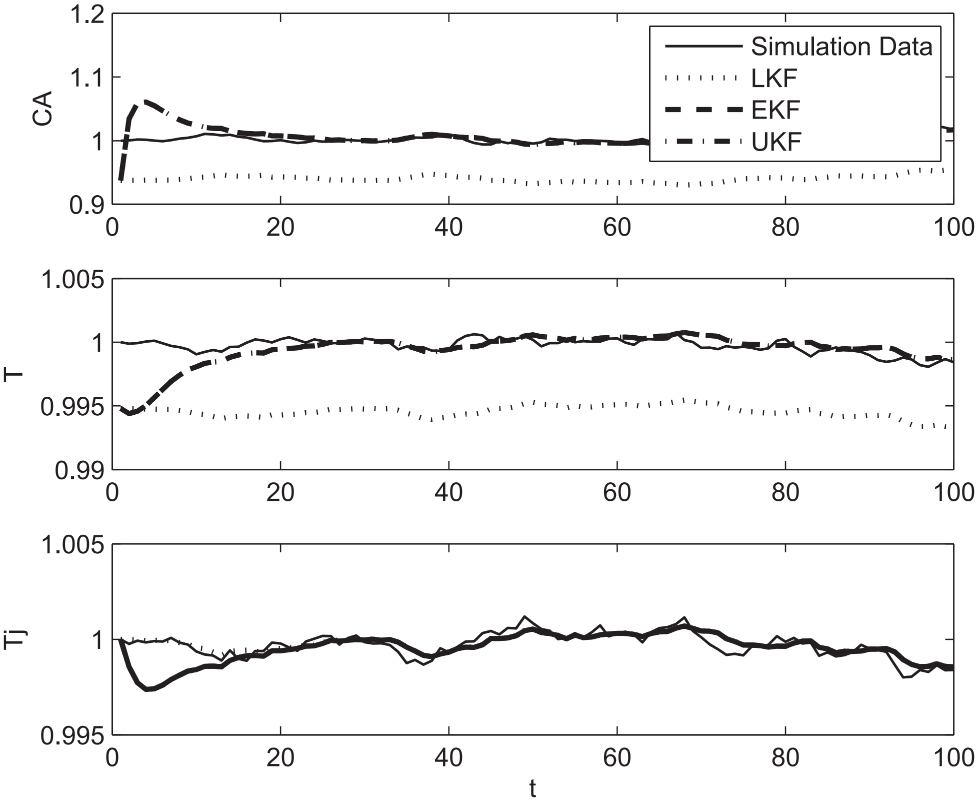


Fig. 5. Performance Comparison for Mildly Nonlinear CSTR.

using LKF. However, UKF exhibits the ability of tracking the system behavior and provide rather good estimates for the system states.

A simulation evaluating UKF's implementation for parameter estimation is also performed. The product of the heat transfer coefficient and the heat transfer area $UA = 4.8 \times 10^{-6} J/s.K$ is considered as the parameter to be estimated. The filter parameters are the same as shown in Eq. (4.18). At time = 150 there is a sudden change in the heat transfer coefficient UA. Figure 6 illustrates the performance of LKF, EKF and UKF for estimating the heat transfer coefficient. Both UKF and EKF perform comparably well. The difference in the results achieved by these two estimators is again minor because the system does not exhibit a strong degree of nonlinearity. However, there is a significant difference between the results computed from EKF, UKF and those returned by LKF.

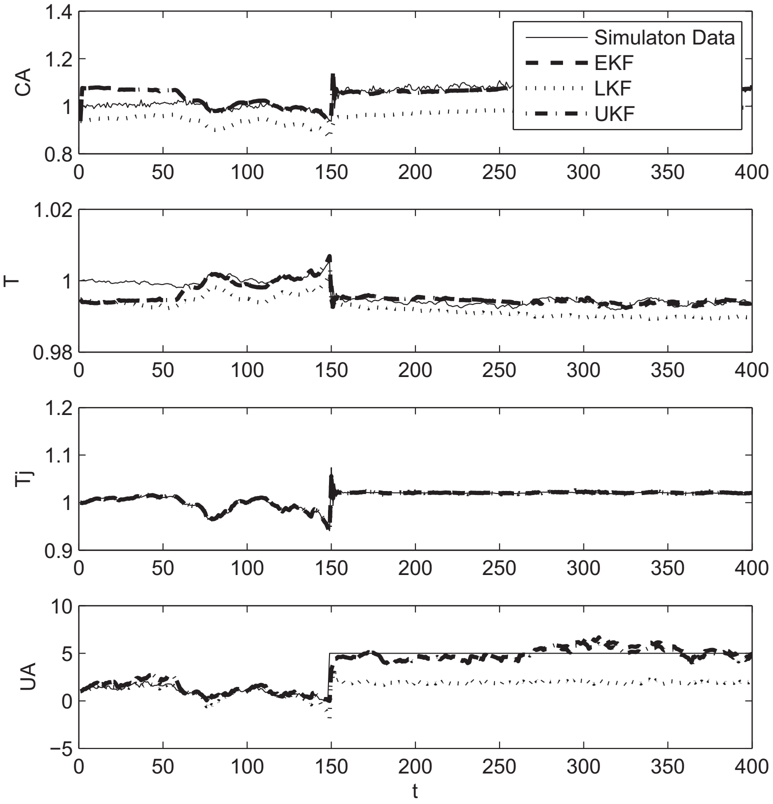


Fig. 6. Performance Comparison for State and Parameter Estimation.

2. Production of Cyclopentanol in a CSTR with van de Vusse Reaction.

In this section performance of UKF and EKF is evaluated for a highly nonlinear system. An isothermal nonlinear CSTR is considered with a competing side reaction governed by van de Vusse reaction kinetics [168]:

$$\begin{aligned} &\mathrm{A} \xrightarrow{\mathrm{k_1}} \mathrm{B} \overset{\mathrm{k2}}{\rightarrow} \mathrm{C} \\ &\mathrm{2A} \overset{\mathrm{k3}}{\rightarrow} \mathrm{D}. \end{aligned} \tag{3.15}$$

Component A is the the reactant cyclopentadiene, B is the product cyclopentanol, C and D are the side products cyclopentandiol and dicylopentadiene. The nonlinear system model is given by the following three differential equations:

$$\frac{dC_A}{dt} = \frac{u}{V}(C_{Ain} - C_A) - k_1 e^{-E_1/RT} C_A - k_3 e^{-E_3/RT} C_A^2 \tag{3.16}$$

$$\frac{dC_B}{dt} = -\frac{u}{V} C_B + k_1 e^{-E_1/RT} C_A - k_2 e^{-E_2/RT} C_B \tag{3.17}$$

$$\begin{aligned} \frac{dT}{dt} =& \frac{1}{\rho c_p}[k_1 e^{-E_1/RT} C_A(-\Delta H_1) + k_2 e^{-E_2/RT} C_B(-\Delta H_2) \\ &+ k_3 e^{-E_3/RT} C_A^2(-\Delta H_3)] + \frac{u}{V}(T_{in} - T) + \frac{Q}{V\rho c_p} \end{aligned} \tag{3.18}$$

where the feed flow rate u is the only controlled variable. The values of the parameters can be found in Tables III and IV [169].

Table III. The Values of the Parameters: Part I

Variable	Value
k_1	$1.287 \times 10^{12}\mathrm{h}^{-1}$
k_2	$1.287 \times 10^{12}\mathrm{h}^{-1}$
k_3	$9.043 \times 10^{9}\mathrm{h}^{-1}/(\mathrm{mol\ h})$
E_1/R	9758.3 K
E_2/R	9758.3 K
E_3/R	8560.0 K

The nonlinear model exhibits multiple steady states, of which the upper steady

Table IV. The Values of the Parameters: Part II

Variable	Value
ΔH_1	4.2 kJ/mol
ΔH_2	-11.0 kJ/mol
ΔH_3	-41.85 kJ/mol
$C_{A,Feed}$	5.1 mol/L
T_{Feed}	403.15 K
V	10.0 L
Q	-4496 kJ/h
ρ	0.9342 kg/L
c_p	3.01 kJ/(kg K)

state ($C_{Ass} = 2.4946$mol/L; $C_{Bss} = 1.1004$mol/L; $T_{ss} = 411.08\ K$; $u = 800L/h$) is chosen as the point of operation. A plot (Figure 7) of steady states C_A, C_B and T as a function of input feed rate u reveals regions where the system exhibits stability (for example, point A) or instability (for example, point B). Parameters for the two operating points are point A ($C_{A,ss} = 2.4946$mol/L; $C_{B,ss} = 1.1004$mol/L; $T_{ss} = 411.08\ K$; $u_{ss} = 800.0\ L/h$) and point B ($C_{A,ss} = 1.0562$mol/L; $C_{B,ss} = 0.8123$mol/L; $T_{ss} = 399.02\ K$; $u_{ss} = 92.5\ L/h$).

In this case study, the system and measurement noise are again assumed to appear linearly. Discretization of continuous differential equations is implemented using finite differences with $\Delta t = 0.002$. The measurable variables are assumed to be the concentration of B and the reactor temperature T. Initial conditions $\hat{x}_0 = \begin{bmatrix} 2.5 & 1.09 & 411.2 \end{bmatrix}^T$. All process variables were scaled to be dimensionless using the upper steady state as the nominal point.

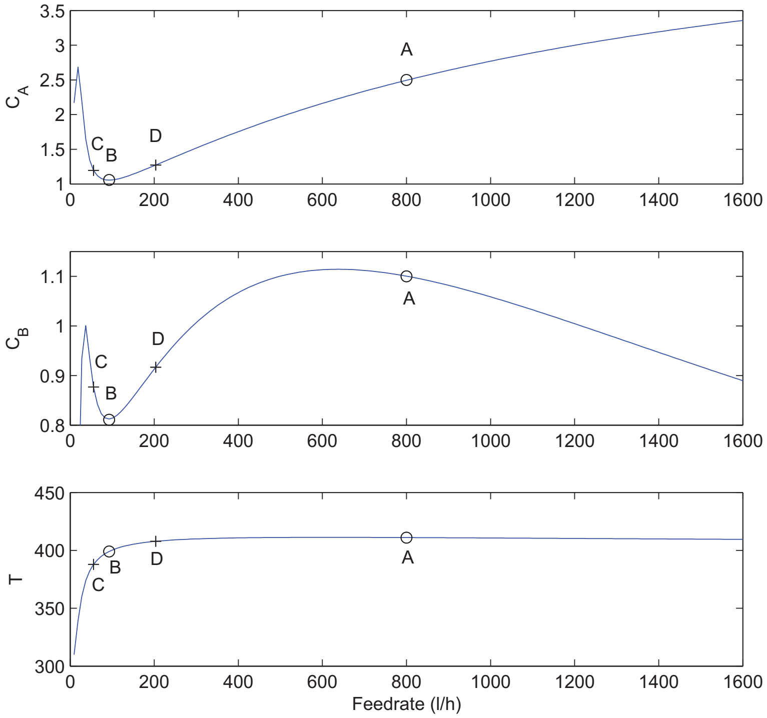


Fig. 7. Steady States as a Function of Reactor Feed Rate for the van de Vusse Reactor

The remaining filter parameters after scaling are given by

$$\hat{P}_0 = diag\{10^{-4}, 10^{-4}, 10^{-4}\} \tag{3.19}$$

$$Q = diag\{10^{-4}, 10^{-4}, 10^{-4}\} \tag{3.20}$$

$$R = diag\{10^{-6}, 10^{-6}\}, \tag{3.21}$$

$$\kappa = -4. \tag{3.22}$$

Several simulation tests were conducted for different scenarios. Table V lists the MSEs generated by EKF and UKF when the system is subjected to different measurement noise levels. UKF outperforms EKF in this case as is illustrated in Figure

Table V. MSEs/Computation Time by Varying Measurement Noise Levels for the Van de Vusse Reactor

		$R = 10^{-6}$	$R = 10^{-4}$	$R = 10^{-2}$
MSE	EKF	1.978×10^{-4}	0.0023	0.0173
	UKF	1.984×10^{-4}	0.0014	0.0134
	% Difference	0.40	39.13	22.54
Computation Time	EKF	1.57	1.58	1.57
	UKF	15.42	15.28	15.64

8, which shows the performance of EKF and UKF when the measurement noise level is set to $R = 10^{-2}$. Similar performance is found when $R = 10^{-4}$. UKF outperforms EKF when the measurement noise level is comparable to or larger than the process noise, because the filters put more emphasis on the model as the measurement noise increases.

3. A Batch Reactor

It should be noted that some of the states of the investigated CSTR reactor models may be subject to constraints, for example, concentrations can not be negative. As neither EKF or UKF can directly deal with constraints, it is warranted to also perform a comparison with MHE. The following reversible gas phase reactions are taking place in a batch reactor [93]:

$$\begin{gathered}\mathrm{A} \underset{\mathrm{k_2}}{\overset{\mathrm{k_1}}{\rightleftharpoons}} \mathrm{B} + \mathrm{C} \\ 2\mathrm{B} \underset{\mathrm{k_4}}{\overset{\mathrm{k_3}}{\rightleftharpoons}} \mathrm{C}\end{gathered} \tag{3.23}$$

The first principles model for a well-mixed, constant volume, isothermal batch

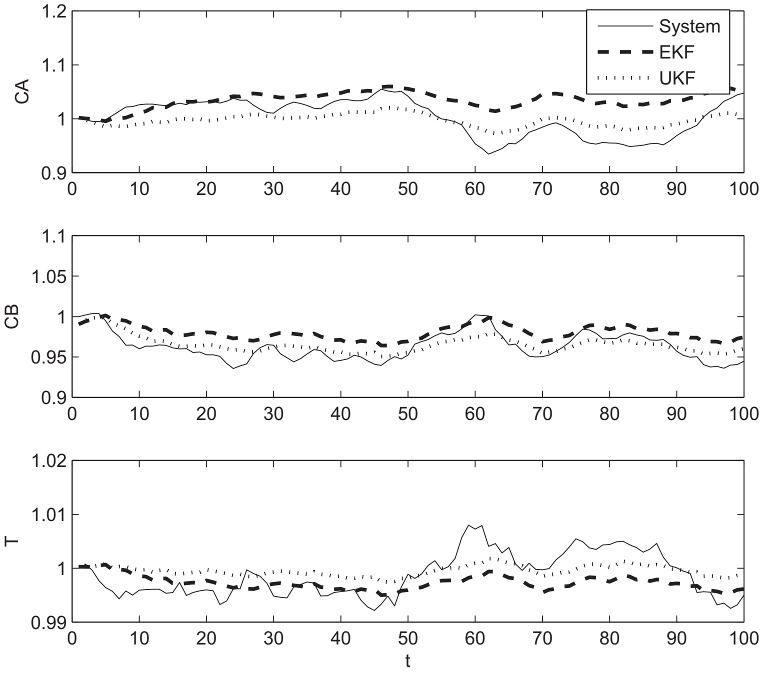


Fig. 8. Performance Comparison for Reactor with van de Vusse Reaction.

reactor is given by the following equations:

$$\frac{dC_A}{dt} = -k_1 C_A + k_2 C_B C_C \tag{3.24}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B C_C - 2k_3 C_B^2 + 2k_4 C_C \tag{3.25}$$

$$\frac{dC_C}{dt} = k_1 C_A - k_2 C_B C_C + k_3 C_B^2 - k_4 C_C \tag{3.26}$$

where C_j denotes the concentration of species j and $[k_1\ k_2\ k_3\ k_4] = [0.5\ 0.05\ 0.2\ 0.01]$. The states are defined as $x = [C_A\ C_B\ C_c]^T$ and the measurement is the pressure which is given by $y = [RT\ RT\ RT]x$. $RT = 32.84$mol atm/L for the case study. A non-negative constraint is enforced on the concentrations at each time step, k. Scaling is not performed given that all the states are concentrations, whose values have the same order of magnitude.

The initial values and filter parameters for state estimation are

$$\Delta t = t_{k+1} - t_k = 0.25, \tag{3.27}$$

$$x_0 = [0.5\ 0.05\ 0]^T, \tag{3.28}$$

$$\hat{x}_0 = [0\ 0\ 4]^T, \tag{3.29}$$

$$\hat{P}_0 = diag(0.5^2, 0.5^2, 0.5^2), \tag{3.30}$$

$$Q_0 = diag(0.001^2, 0.001^2, 0.001^2), \tag{3.31}$$

$$R_0 = 0.25^2, \tag{3.32}$$

$$\kappa = -4, \qquad N = 3$$

Figure 9 illustrates the estimation results of UKF and EKF as compared to MHE. The

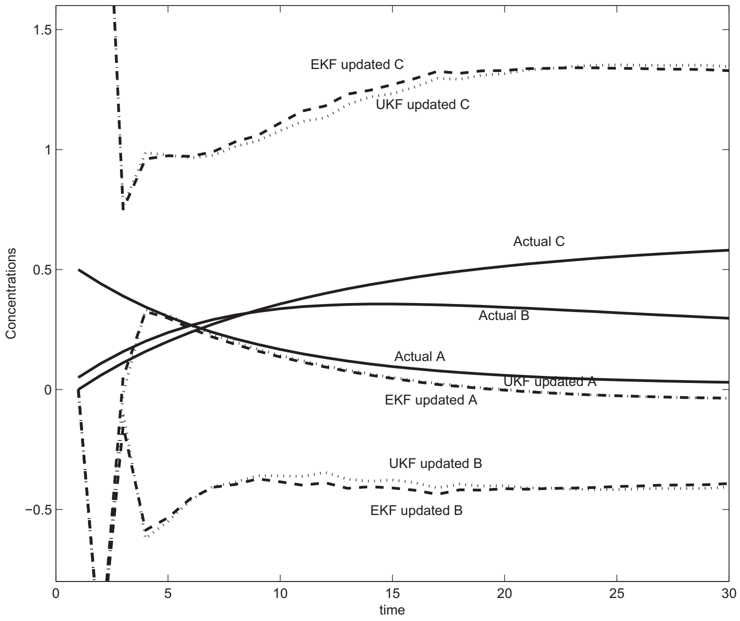


Fig. 9. Performance Comparison of UKF and EKF for Batch Reactor

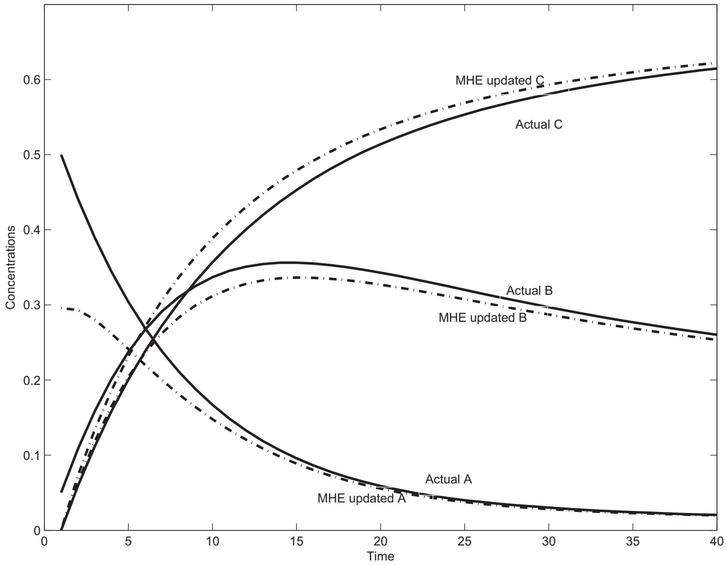


Fig. 10. Performance of MHE for Batch Reactor

solid line represents the system behavior. The dotted line shows the UKF estimates while the EKF results are illustrated by the dashed line. Neither UKE nor EKF converge to the correct state estimates due to their inability to handle constraints. The results using MHE are shown in Figure 10. MHE converges to the correct state values since the state constraints prevent estimation of negative concentrations. It can be concluded that both UKF and EKF may have limitations for computing reasonably good estimates when constraints exist.

D. Conclusions

This chapter presents a comprehensive comparison of UKF with traditional estimation techniques such as LKF, EKF and MHE. It can be seen that the unscented Kalman filter provides a good estimate of the states and parameters and is comparable in performance to EKF when system nonlinearities are not significant. UKF outperforms EKF when severe nonlinearities exist and the measurement noise levels are high.

Additionally, when the structures of the process and measurement functions are not differentiable or are discontinuous, UKF also offers a benefit because it does not require to calculate Jacobian matrices. It should be noted that both EKF and UKF may fail to converge to the true values of the states for constrained problems; MHE may be a better option for constrained problems, however, MHE results in a higher computational burden.

CHAPTER IV

COMPUTATION OF ARRIVAL COSTS FOR MOVING HORIZON ESTIMATION VIA UNSCENTED KALMAN FILTERING*

A. Introduction

For the last two decades, full information estimation (FIE) or optimization-based moving horizon estimation (MHE) have been used in state estimation, data reconciliation and fault detection for nonlinear constrained problems. Many applications of this approach have been investigated [104], [170], [93]. MHE reduces the computational burden as compared to FIE by considering a finite horizon of the available measurements, however, it is non-trivial to summarize the effect of the discarded data on the current states, which is the so called *arrival cost*. For linear unconstrained systems, the Kalman filter covariance can be used to express arrival cost explicitly. However, for nonlinear or constrained systems, a general analytical expression for the arrival cost is rarely available. Tenny and Rawlings [103] estimate the arrival cost by approximating the constrained, nonlinear system as an unconstrained linear time-varying system and applying linearization and standard Kalman Filter. However, this approximation may not be ideal for computing the arrival cost. The best choice of the arrival cost still remains an open issue.

Unscented Kalman filters (UKF), as proposed by Julier and Uhlman [41], avoid the linearization in the Kalman filter update formula by an unscented nonlinear transformation. By carefully choosing a set of sigma points, which capture the true mean

* Part of this chapter is reprinted with permission from "Computation of Arrival Costs for Moving Horizon Estimation via Unscented Kalman Filtering" by C. Qu and J. Hahn, 2009. *Journal of Process Control*, Vol. 19, No. 2, pp. 358-363, doi:10.1016/j.jprocont.2008.04.005, Copyright [2009] by Elsevier.

and covariance of the given distribution and then passing the mean and covariances of estimated states through a nonlinear transformation, UKF is capable of estimating the posterior mean and covariances accurately to an order higher than two. Therefore, UKF can improve the Kalman filter performance in nonlinear estimation. Because of its potential for good performance in nonlinear estimation, a MHE filter based on UKF, which is employed to approximate the arrival cost, is proposed in this paper and its performance is illustrated.

The chapter is organized as follows: The MHE with arrival cost determined by UKF for nonlinear constrained estimation is proposed in Section B. Section C compares the performance of the MHE via EKF to that via UKF for nonlinear state estimation and lastly concluding remarks are given in Section D.

B. Moving Horizon Estimation via Unscented Transformation

As mentioned in Chapter II, algebraic expressions for the arrival cost, which account for data not included in the estimation window, are not available for the majority of systems. Therefore an approximation of the arrival cost is required to implement an estimator.

The main idea behind the presented MHE algorithm is to make use of the advantages that UKF offers over EKF for approximating the arrival cost. A set of weighted sigma points χ_k are selected for computing the arrival cost based on two considerations: Firstly the means and covariances of the set of sigma points need to match those of the prior distribution at time $k = T - N$ in the absence of active bounds. Secondly the distribution of the selected sigma points should be within the feasible region if bounds are active. If $s_{k,i} = \pm(\sqrt{P^a_{k|k}})_i, i = 1, ..., n$ are considered as the directions along which the sigma points are selected and $r_{k,i}$ as the step sizes along these directions, then the step sizes for the selected sigma points in Eq.(**??**) for a general UKF are all equal to $\sqrt{n^a + \kappa}$. In other words all the sigma points chosen are located symmetrically around the current estimate. This selection of sigma points can be applied to the MHE via UKF when the constraints are inactive. However, such selection is not adequate when constraints are active. To better approximate the covariance and then the arrival cost in the presence of active constraints, the set of sigma points is chosen in each direction with a step size of

$$r_{k,i} = \min(\sqrt{n^a + \kappa}, (x_{U,i} - \hat{x}^a_{k|k,i})/s_{k,i}, (x_{L,i} - \hat{x}^a_{k|k,i})/s_{k,i}). \tag{4.1}$$

where $x_{U,i}$ and $x_{L,i}$ are the upper and lower bounds in the direction $s_{k,i}$. The central point $\hat{x}^a_{k|k,0}$ is the same as in the general UKF formulation. The rest of selected sigma points may be asymmetrically around the central point due to the presence of active constraints. These sigma points are identical to those for conventional UKF for an unconstrained problem or if the constraints are not active. However, under the condition that the current estimate is close to the bounds, the selection process takes the constraints into account so that none of the selected sigma points violate the constraints on the state variables.

Considering that weights of all sigma points sum up to unity and that these are the same as those for UKF in Eq.(3.10) in the absence of active bounds, i.e.,

$$\begin{aligned} a\sum_{i=1}^{2n} \theta_i + (2n+1)b &= 1 \\ a\sqrt{n+\kappa} + b &= \frac{1}{2(n+\kappa)}, \end{aligned} \tag{4.2}$$

weights W_i for each sigma point can be calculated as follows:

$$W_i = \begin{cases} \frac{\kappa}{2(n^a+\kappa)}, & \text{if i=1;} \\ ar_i + b, & \text{otherwise.} \end{cases} \tag{4.3}$$

where

$$\begin{aligned} a &= \frac{2\kappa - 1}{2(n^a+\kappa)(S_r - (2n^a+1)(\sqrt{n^a+\kappa}))} \\ b &= \frac{1}{2(n^a+\kappa)} - \frac{2\kappa - 1}{2\sqrt{n^a+\kappa}(S_r - (2n^a+1)(\sqrt{n_a+\kappa}))} \\ S_r &= \sum_{i=1}^{2n^a} r_i. \end{aligned} \tag{4.4}$$

The procedure to compute the weights with constraints is similar to the work presented by Vachhani et al. [62], where a complete mathematical derivation of the weight equations albeit for different purposes can be found.

After a set of sigma points is obtained, each of these sigma points is instantiated through the nonlinear model functions $f(.)$ and $h(.)$ to obtain the transformed sets χ_k^x and γ_k.

$$\begin{aligned} \chi_k^x &= f(\chi_{k-1}^x, \chi_{k-1}^w, u_{k-1}) \\ \gamma_k &= h(\chi_k^x, \chi_{k-1}^v, u_k) \end{aligned} \tag{4.5}$$

The weighted means of system states and measurements and weighted covariances of process and observation noise are then computed from the transformed sets. It should be noted that the weighted predicted estimates $\hat{x}_{k|k-1}$ may not satisfy the constraints.

$$
\begin{aligned}
\hat{x}_{k|k-1} &= \sum_{i=1}^{2n^a+1} W_i \chi_{i,k}^x \\
\hat{y}_k &= \sum_{i=1}^{2n^a+1} W_i \gamma_{i,k} \\
P_{k|k-1} &= \sum_{i=1}^{2n^a+1} W_i [\chi_{i,k}^x - \hat{x}_{k|k-1}][\chi_{i,k}^x - \hat{x}_{k|k-1}]^T \\
P_{y,k} &= \sum_{i=1}^{2n^a+1} W_i [\gamma_{i,k} - \hat{y}_k][\gamma_{i,k} - \hat{y}_k]^T \\
P_{xy,k} &= \sum_{i=1}^{2n^a+1} W_i [\chi_{i,k}^x - \hat{x}_{k|k-1}][\gamma_{i,k} - \hat{y}_k]^T
\end{aligned}
\tag{4.6}
$$

Finally the matrix P_k is calculated from the filter gain and is used to compute the arrival cost in Eq.(4.7). The MHE problem described in the Eq.(2.14) and (2.15) is solved with the approximated arrival cost to obtain the updated estimates $\hat{x}_{k|k}$ as the solutions:

$$
\begin{aligned}
K_k &= P_{xy,k} P_{y,k}^{-1} \\
P_{k|k} &= P_{k|k-1} - K_k P_{y,k} K_k^T \\
\theta_k(z) &= (z - \hat{x}_k)' P_k^{-1} (z - \hat{x}_k) + \phi_k^*
\end{aligned}
\tag{4.7}
$$

The proposed approach to approximate the arrival cost for MHE does not require linearization of the system and measurement functions. As in any MHE filter, the matrices Q and R can be chosen to take uncertainty in the model and measurement noise into account and the size of the estimation horizon serves as an additional tuning parameter.

C. Case Studies

To illustrate the performance of MHE based on UKF (uMHE) compared against the one based on EKF (eMHE), both algorithms have been applied to a variety of models and a large number of scenarios such as different operating conditions, different tuning parameters Q and R, and different process and measurement noise. This section revisits the case studies in Chapter III and shows the performance of uMHE as compared to eMHE. Monte Carlo simulations with 50 sample points have been conduct for each procedure so as not to bias results to one set of data. The performance is evaluated by the overall mean-squared error (MSE). MSE is first averaged over all simulations for each time point and then over time to to take the behavior over the entire time horizon into account. Numerical algorithms for minimizing or maximizing a function provided by commercial software $NAG^{©}$ are used for solving MHEs.

1. A Batch Reactor

In this section, the gas phase reversible reactions in the following are revisited [93].

$$\begin{aligned} &\mathrm{A} \underset{\mathrm{k_2}}{\overset{\mathrm{k_1}}{\rightleftharpoons}} \mathrm{B} + \mathrm{C} \\ &2\mathrm{B} \underset{\mathrm{k_4}}{\overset{\mathrm{k_3}}{\rightleftharpoons}} \mathrm{C} \end{aligned} \tag{4.8}$$

The first principle model for a well-mixed, constant volume, isothermal batch reactor is described by the following equations:

$$\frac{dC_A}{dt} = -k_1 C_A + k_2 C_B C_C \tag{4.9}$$

$$\frac{dC_B}{dt} = k_1 C_A - k_2 C_B C_C - 2k_3 C_B^2 + 2k_4 C_C \tag{4.10}$$

$$\frac{dC_C}{dt} = k_1 C_A - k_2 C_B C_C + k_3 C_B^2 - k_4 C_C \tag{4.11}$$

where C_j denotes the concentration of species j, $[k_1\ k_2\ k_3\ k_4] = [0.5\ 0.05\ 0.2\ 0.01]$. The state is defined to be $x = [C_A\ C_B\ C_c]^T$ and the measurement $y = [RT\ RT\ RT]x$. It is assumed that the ideal gas law holds (high temperature and low pressure).

The initial values and filter parameters for state estimation are

$$\Delta t = t_{k+1} - t_k = 0.25$$

$$x_0 = [0.5\ 0.05\ 0]^T$$

$$\hat{x}_0 = [0\ 0\ 4]^T$$

$$\hat{P}_0 = diag(0.5^2, 0.5^2, 0.5^2)$$

$$Q_0 = diag(0.001^2, 0.001^2, 0.001^2)$$

$$R_0 = 0.25^2, \qquad \kappa = -4, \qquad N = 3$$

The performance of MHE via either EKF or UKF is presented in Figure 11. As a comparison to the performance of EKF and UKF in Chapter III, Both MHE approaches converge to the true state estimates after the state constraints are applied to prevent estimation of negative concentrations. The MSE error of uMHE is 0.0067 while eMHE generates a MSE error of 0.0133. uMHE provides slightly better performance than eMHE does.

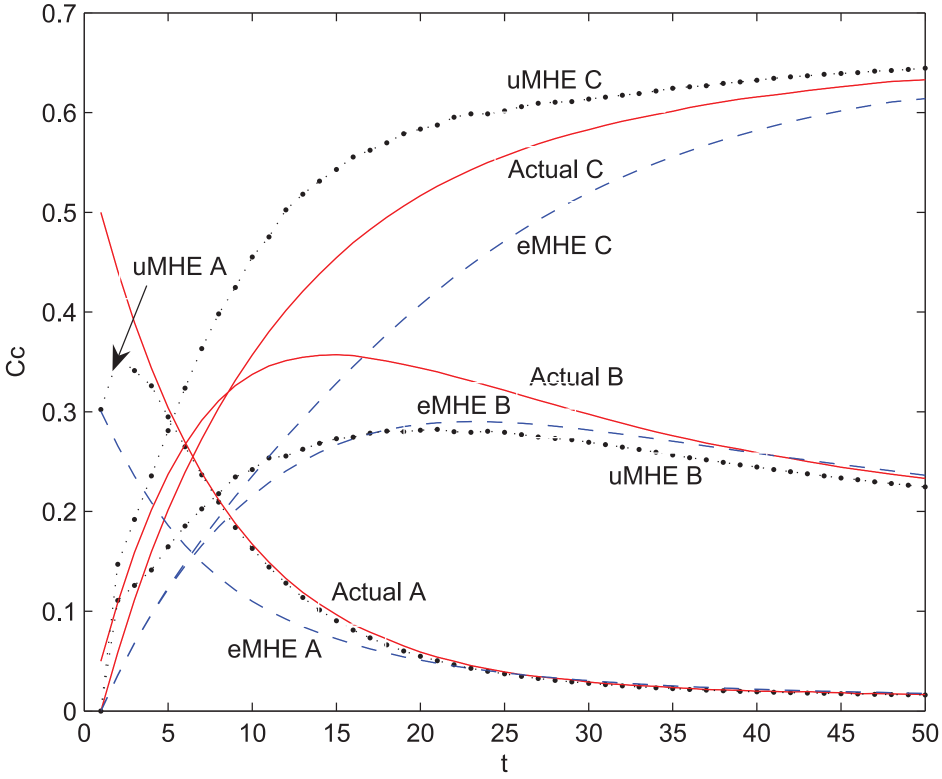


Fig. 11. Performance Comparison of uMHE and eMHE

2. CSTR with Exothermal Irreversible Reaction

The model of this section is the nonisothermal continuous stirred tank reactor with coolant jacket dynamics, where the following exothermic irreversible reaction between sodium thiosulfate and hydrogen peroxide is taking place:

$$2Na_2S_2O_3 + 4H_2O_2 \rightarrow Na_2S_3O_6 + Na_2SO_4 + 4H_2O \tag{4.12}$$

The capital letters A and B are used to denote the chemical compounds $Na_2S_2O_3$ and H_2O_2 in the following. The reaction kinetic law is reported in the literature to be

[167]:

$$-r_A = k_0 e^{-E/RT} C_A C_B$$

where k_0 is the pre-exponential factor, E is the activation energy, R is the gas constant, T is the temperature, and C_A and C_B are the concentrations of species A and B, respectively. A stoichiometric proportion of species A and B in the feed stream is assumed which results in $C_B(t) = 2C_A(t)$. A mole balance for species A and energy balances for the reactor and the cooling jacket result in the following nonlinear process model:

$$\begin{aligned}
\frac{dC_A}{dt} &= \frac{F}{V}(C_{Ain} - C_A) - 2k(T)C_A^2 \\
\frac{dT}{dt} &= \frac{F}{V}(T_{in} - T) + 2\frac{(-\Delta H)_R}{\rho c_p}k(T)C_A^2 - \frac{UA}{V\rho c_p}(T - T_j) \\
\frac{dT_j}{dt} &= \frac{F_w}{V}(T_{jin} - T_j) + \frac{UA}{V_w \rho_w c_{pw}}(T - T_j)
\end{aligned} \tag{4.13}$$

where F is the feed flow rate, V is the volume of the reactor, C_{Ain} is the inlet feed concentration, T_{in} is the inlet feed temperature, F_w is the feed flow rate of the cooling jacket, V_w is the volume of the cooling jacket, T_{jin} is the inlet coolant temperature, c_p is the heat capacity of the reacting mixture, c_{pw} is the heat capacity of the coolant, ρ is the density of the reaction mixture, ρ_w is the density of the coolant, U is the overall heat-transfer coefficient, and A is the area over which the heat is transferred. The process parameter values are given in the work by Rajaraman et al. [16].

The nonlinear model without process noise exhibits multiple steady states, of which the upper steady state (i.e. $C_{Ass} = 0.019$mol/L; $T_{ss} = 384.0\ K$; $T_{js} = 371.3$ K; $F_w = 30L/min$) is chosen as the point of operation.

Process and observation noise are assumed to enter the system linearly. By discretizing the system model in (4.13), a discrete model of the form

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, k-1) + G w_{k-1} \\ y_k &= C x_k + v_k \end{aligned} \tag{4.14}$$

is obtained, where $x_k = \begin{bmatrix} C_A & T & T_j \end{bmatrix}^T$, $w_{k-1} \sim \mathcal{N}(0, Q_{k-1})$ and $v_k \sim \mathcal{N}(0, R_k)$ are zero mean Gaussian noise and a sampling rate of 1.4Hz is used. The third state T_j is chosen as the only measurable state in this case.

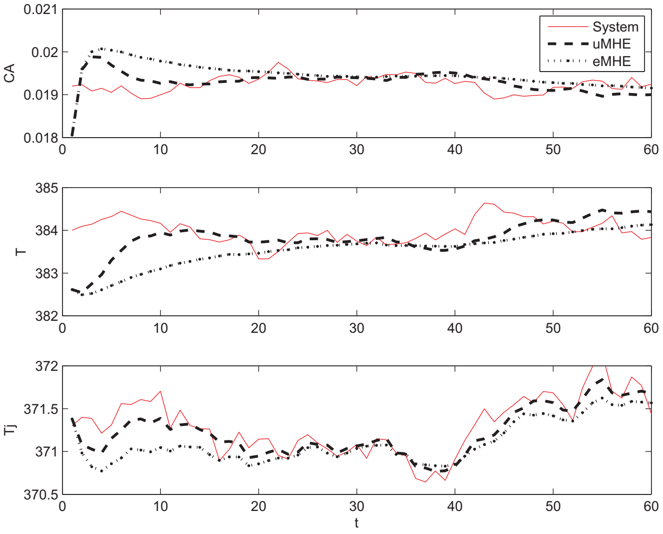


Fig. 12. Performance Comparison of uMHE and eMHE (N=3, R=0.25)

The initial conditions and filter parameters are as follows:

$$\hat{x}_0 = \begin{bmatrix} 0.018 & 382 & 371.3 \end{bmatrix}^T, \tag{4.15}$$

$$\hat{P}_0 = diag\{10^{-7}, 2.5, 2.5\}, \tag{4.16}$$

$$Q_0 = diag\{10^{-8}, 0.25, 0.25\}, \tag{4.17}$$

$$R_0 = 0.25. \tag{4.18}$$

The dimension of the augmented state vector is 7 and the set of sigma points is

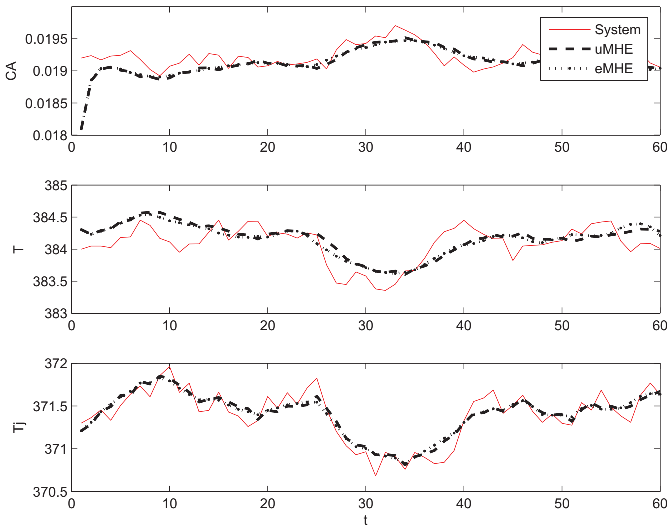


Fig. 13. Performance Comparison of uMHE and eMHE (N=6, R=0.25)

composed of 15 elements. The additional tuning parameter of the UKF, κ, is set to -4 for the case study according to the heuristic mentioned in Section B. For fairness of comparison, κ is not further adjusted to fine tune the higher order moments of the approximation. A non-negative constraint is enforced on the concentration C_A for both MHE formulations.

Figure 12 illustrates the system behavior and the performance of the uMHE and the eMHE with a horizon length set to 3 for one simulation. Based on the overall mean-squared error, the performance of each MHE is evaluated for horizon lengths N=3, 4, 6, and 10. Figure 13 shows the performance for N=6 and R=0.25, where both MHEs perform similarly well.

Further simulations have also been carried out by varying measurement noise parameters for a fixed horizon length. Figure 14 presents the performance of eMHE

Table VI. MSEs Comparison for uMHE & eMHE by Varying Measurement Noise Levels and Horizon Lengths for an Exothermic Reaction

MSE	N=3		N=4		N=6		N=10	
	eMHE	uMHE	eMHE	uMHE	eMHE	uMHE	eMHE	uMHE
R = 25	8.45	5.64	5.56	4.94	5.14	4.76	2.66	2.52
R = 0.25	2.69	1.45	1.43	1.21	0.89	0.84	0.87	0.70
R = 0.01	1.01	0.86	0.80	0.78	0.59	0.56	0.48	0.45

and uMHE with $R_0 = 25, N = 3$, while the results when process noise dominates ($R_0 = 0.01, N = 3$) are shown in Figure 15. Both of the figures demonstrate that uMHE performs slightly better than eMHE.

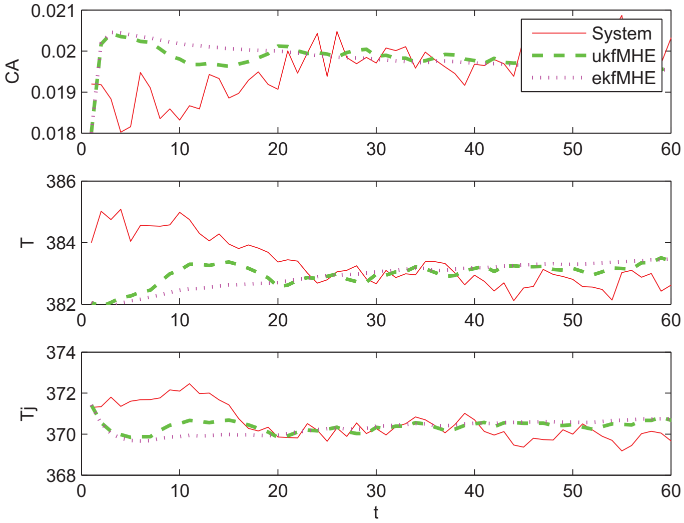


Fig. 14. Performance with Large Measurement Noise(N=3, R=25)

Table VI provides a summary of the results by varying measurement noise parameters and horizon lengths. It can be seen that uMHE performs better than eMHE

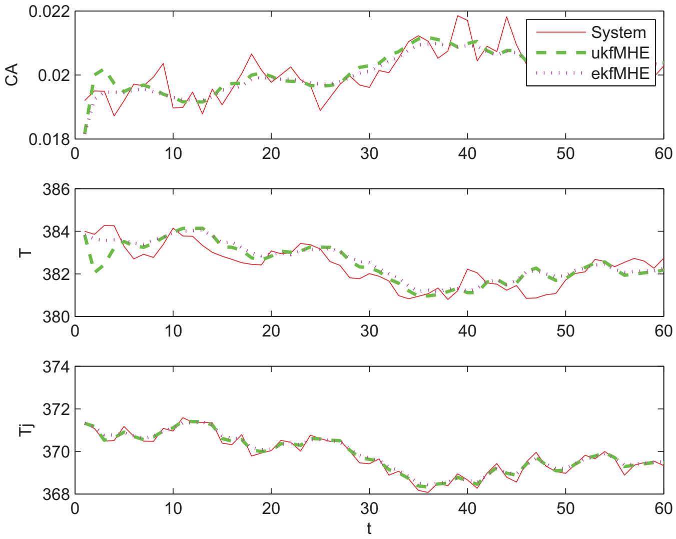


Fig. 15. Performance with Small Measurement Noise(N=3, R=0.01)

for all the investigated horizon lengths and measurement noise levels. The MSEs for both uMHE and eMHE are decreasing with increasing lengths of the horizons, i.e, the performance of both uMHE and eMHE improves as more data are included in a horizon. If N is chosen to be large, the arrival cost could be accurately computed with either approach. Therefore the advantages of uMHE over eMHE decrease for large N.

Table VII presents the computation time for eMHE and uMHE for varying measurement noise levels and horizon lengths. Since the computation time depends upon the specific algorithm used for implementing a MHE, all shown results are scaled by the time required for eMHE, N=3, R=0.25. As the horizon length increases, the computational burden increases. When comparing computation times for uMHE and eMHE, no clear trend can be observed.

The MSEs for MHE without arrival cost are shown in Table VIII. As com-

Table VII. Computation Cost for uMHE & eMHE by Varying Measurement Noise Levels and Horizon Lengths for an Exothermic Reaction

Computation Ratio	N=3		N=4		N=6		N=10	
	eMHE	uMHE	eMHE	uMHE	eMHE	uMHE	eMHE	uMHE
R = 25	1	1.05	1.86	1.97	4.37	3.93	9.97	9.87
R = 0.25	1.97	2.60	2.85	2.87	5.91	5.57	10.12	11.06
R = 0.01	1.03	0.92	2.28	2.21	8.93	10.24	13.99	18.47

pared to uMHE and eMHE, MHE without arrival cost generates significantly large MSEs. Therefore approximating arrival cost is important for obtaining a good MHE performance.

Table VIII. MSEs Comparison for uMHE & eMHE by Varying Measurement Noise Levels for an Exothermic Reaction

	R = 25	R = 0.25	R = 0.01
MHE w/o arrival cost	16.991	8.5659	2.2010
eMHE	3.8349	2.2187	0.8516
uMHE	2.9231	1.1557	0.6553

3. Production of Cyclopentanol in a CSTR with van de Vusse Reaction.

In this section, the performance of the eMHE and the uMHE is evaluated on an isothermal, perfectly mixed, nonlinear CSTR with a myriad of competing side reactions governed by the van de Vusse reaction kinetics shown in equation (5.30)

[169]

$$\begin{aligned}&\mathrm{A} \xrightarrow{\mathrm{k_1}} \mathrm{B} \overset{\mathrm{k2}}{\rightarrow} \mathrm{C}\\ &2\mathrm{A} \overset{\mathrm{k3}}{\rightarrow} \mathrm{D}\end{aligned} \tag{4.19}$$

Component A is the the reactant cyclopentadiene, B is the product cyclopentanol, C and D are the side products cyclopentandiol and dicylopentadiene. Three differential equations given by eq.(5.31)–(5.33) describe the nonlinear system model.

$$\frac{dC_A}{dt} = \frac{u}{V}(C_{Ain} - C_A) - k_1 e^{-E_1/RT} C_A - k_3 e^{-E_3/RT} C_A^2 \tag{4.20}$$

$$\frac{dC_B}{dt} = -\frac{u}{V} C_B + k_1 e^{-E_1/RT} C_A - k_2 e^{-E_2/RT} C_B \tag{4.21}$$

$$\begin{aligned}\frac{dT}{dt} =& \frac{1}{\rho c_p}[k_1 e^{-E_1/RT} C_A(-\Delta H_1) + k_2 e^{-E_2/RT} C_B(-\Delta H_2)\\ &+ k_3 e^{-E_3/RT} C_A^2(-\Delta H_3)] + \frac{u}{V}(T_{in} - T) + \frac{Q}{V\rho c_p}\end{aligned} \tag{4.22}$$

where the feed flow rate u is the only controlled variable. The measurable variables are assumed to be the concentration of B and the reactor temperature T. The values of the parameters can be found by Hahn and Edgar [169].

The nonlinear model exhibits multiple steady states. A plot of steady states C_A, C_B and T as a function of input feed rate u (Figure 16) reveals regions where the system exhibits stability (for example, point A) or instability (for example, point B). Steady states of each operating point are listed in Table IX.

In this case study, the system and measurement noise are again assumed to appear linearly. Discretization of continuous differential equations is implemented using finite differences with $\Delta t = 0.002$. The measurable variables are assumed to be the concentration of B and the reactor temperature T.

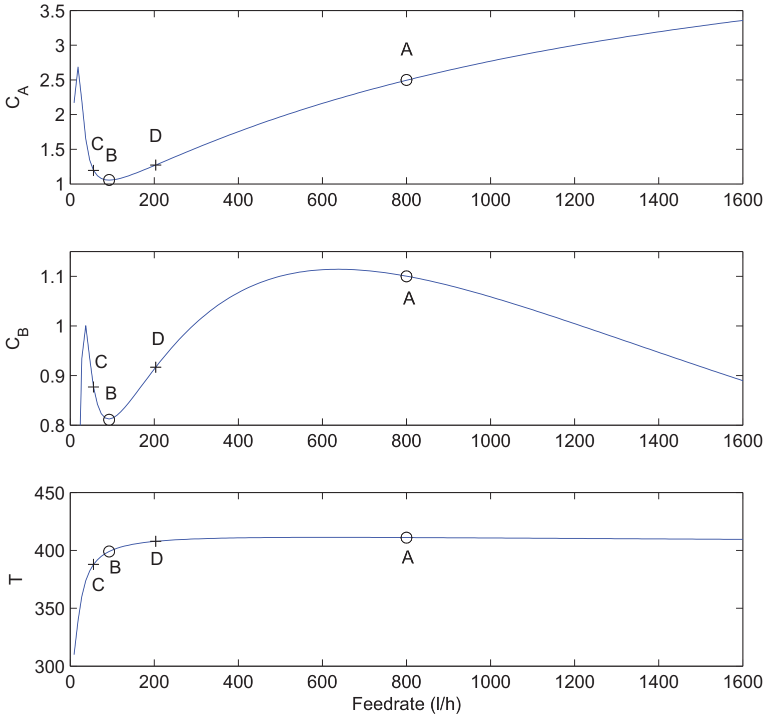


Fig. 16. Steady States as a Function of Reactor Feed Rate for the van de Vusse Reactor

The initial conditions for operating point A, B and D are as follows

$$\hat{x}_{A0} = \begin{bmatrix} 2.5 & 1.09 & 411.2 \end{bmatrix}^T, \tag{4.23}$$

$$\hat{x}_{B0} = \begin{bmatrix} 1.04 & 0.8 & 399 \end{bmatrix}^T, \tag{4.24}$$

$$\hat{x}_{D0} = \begin{bmatrix} 1.12 & 0.91 & 407.8 \end{bmatrix}^T. \tag{4.25}$$

Table IX. Parameters at Steady States for the van de Vusse Reactor

	Point A	Point B	Point C	Point D
C_A (mol/L)	2.4946	1.0562	1.195	1.273
C_B (mol/L)	1.1004	0.8123	0.877	0.9168
T (K)	411.08	399.02	387.9	407.9
u (L/h)	800	92.5	55.5	203.5

The remaining filter parameters are given by

$$\hat{P}_0 = diag\{10^{-2}, 10^{-2}, 1\}, \tag{4.26}$$

$$Q = diag\{10^{-4}, 10^{-4}, 1\}, \tag{4.27}$$

$$R = diag\{10^{-4}, 1\}, \tag{4.28}$$

$$\kappa = -4, \qquad N = 3. \tag{4.29}$$

Table X. MSEs Comparison for uMHE & eMHE by Varying Input Rates for the van de Vusse Reactor

	eMHE	uMHE
u = 800	1.2958	0.9480
u = 92.5	1.2595	1.1881
u = 203.5	0.7950	0.7682
u varies	0.9552	0.9434

Several simulation tests were conducted for different input rates. Table X lists the MSEs generated by uMHE and eMHE when the system is subjected to different input flowrate u. uMHE performs again insignificantly better than eMHE. This is

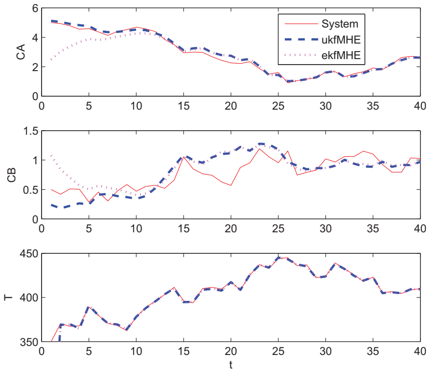


Fig. 17. Performance Comparison of uMHE and eMHE (u=800)

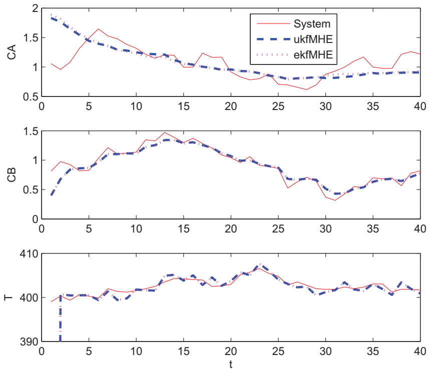


Fig. 18. Performance Comparison of uMHE and eMHE (u=92.5)

further illustrated by Figures 17 $\sim$ 19, which shows the performance of each filter at the operating point A, B and D respectively. The performance of each filter is further investigated by enlarging the operating region by operating between point D and point C, where the sign of the first derivative of nonlinearities changed. At time $t = 50$ when the input feed rate is decreased from 203.5 L/h to 55.5 L/h, the system is driven from steady state D to steady state C. uMHE slightly outperforms eMHE again with a MSE error of 0.9434 versus 0.9552.

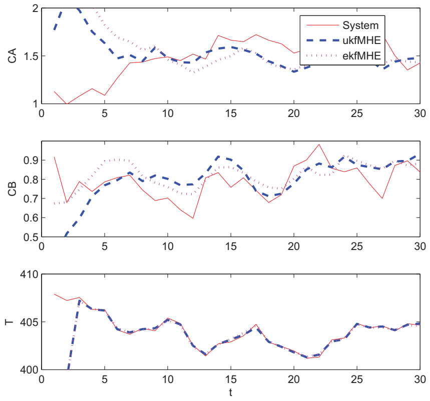


Fig. 19. Performance Comparison of uMHE and eMHE (u=203.5)

D. Conclusions

This chapter presented a MHE formulation where the arrival cost is computed by UKF. The unscented transformation and a set of selected sigma points are used to compute the covariances and then the arrival cost. The selection procedure for the sigma points is the same as the one used for unscented Kalman filtering if the constraints are inactive, however, a modification is used that satisfies the state variable constraints when the constraints are active. Linearization of the model is not required for the presented approach.

The presented method performed slightly better than the commonly used eMHE for all investigated cases. Therefore, the method can be a promising alternative for approximating the arrival cost for MHE.

CHAPTER V

INVESTIGATION OF DIFFERENT EXTENDED KALMAN FILTER IMPLEMENTATIONS

A. Introduction

Extended Kalman filters have found wide-spread use in nonlinear state and parameter estimation. In order to apply Kalman filters to nonlinear systems, EKF uses a first-order Taylor series expansion to linearize the nonlinear model along its trajectory and assumes a Gaussian noise distribution. This inevitably leads to EKF's limitations when applied to nonlinear systems or non-Gaussian noise processes. Numerous estimation methodologies have been proposed in the literature to address the problems that EKF encounters and comparisons between the presented new approaches and EKF have often been made. To name a few, Rawlings and coworkers compared moving horizon estimation (MHE) to EKF for estimation of constrained problem and pointed out that EKF may fail to converge to the true values [93]. Chen *et al*. investigated particle filtering (PF) performance in industrial batch processes and compared the results to EKF [82]. Similar comparisons on a continuous stirred-tank reactor (CSTR) were made by Chen and his coworkers [84]. In the area of unscented Kalman filter (UKF) applications, Romanenko *et al.* applied EKF and UKF to a nonlinear exothermic chemical CSTR [39] and a pH system [40]. The authors showed improvements in the performance of UKF over EKF in both cases. Kandepu *et al*. conducted comparisons between UKF and EKF for four cases including a Van der Pol oscillator and a reversible reaction [171]. They also showed that the UKF performs better than the EKF in terms of robustness and speed of convergence. Contrary to some of these findings, the work by Qu and Hahn [61] found the difference between

UKF and EKF to be minor for mildly nonlinear systems and the advantages of UKF over EKF can be mainly seen for highly-nonlinear systems. It is evident from several of these findings that only a few generally applicable conclusions can be drawn when comparing different estimators. This is especially so as the implementation of an estimator can affect its performance.

This work performs a detailed study of EKF implementations with a focus on several key procedures such as discretization, first order linearization and computation of the Jacobian matrix for nonlinear continuous-time model functions. Comparisons among the implementations are made based upon a chemical reactor model with Van de Vusse reaction kinetics.

This chapter is organized as follows. Section B presents different approaches for EKF implementations. Remarkable notes on each of those methods are given in Section C. Section D compares the performance of each EKF implementation based upon application to a continuous stirred tank reactor exhibiting nonlinear dynamic behavior. Concluding remarks are given in Section E.

B. Implementations of EKF

In order to implement an extended Kalman filter, attention has to be paid to discretization and linearization of nonlinear continuous-time model along its trajectory. The order in which discretization and linearization are performed may result in differences in performance. The approach used for discretizing systems, such as Euler's method or Runge-Kutta method, may also lead to different results. In addition, computation of Jacobian matrices via sensitivity equations or via finite differences affect the accuracy of EKF.

Taking these points into account, several different EKF implementations are discussed in this section. The reason for doing so is that when results for EKF and other estimation methods are reported in the literature, there is often very little discussion of the discretization scheme used, yet the choice of a discretization scheme has a major effect on the outcome. In this work, the system model is assumed to be a nonlinear continuous dynamic system with discrete measurements, such as the one shown in Eqs. (B.6) - (2.9). The functions f and h are differentiable functions of the state vector x, $w \in \mathbb{R}^n$ is a vector of plant noise, with $E[w] = 0$ and $E[ww^T] = Q$; $y_k \in \mathbb{R}^m$ is a vector of the measured variables and $v_k \in \mathbb{R}^m$ is a vector of measurement noise, with $E[v_k] = 0$ and $E[v_k v_k^T] = R_k$; n is the number of states, m refers to the number of measurement variables. The distributions of w and v are Gaussian. The initial value x_0 is also a Gaussian random variable with known mean $\bar{x}_0$ and known $n \times n$ covariance matrix P_{x_0}. The sampling time for measurements is T. $x(t)$,$u(t)$ and $w(t)$ are referred to x, u and w, respectively, in the rest of the chapter unless specified.

1. Implementations via Linearization and Continuous KF for Covariance Prediction

This algorithm (**Algorithm 1**) linearizes the model along its trajectory and then predicts the covariance matrix P via continuous KF.

The state estimate $\hat{x}$ is computed from the nonlinear differential equation, i.e.,

$$\dot{\hat{x}} = f(\hat{x}, u)^1 \tag{5.1}$$

[1] Consider the first-order Taylor series expansion of f(x,u) about the current estimate (i.e., conditional mean) $\hat{x}$:

$$f(x,u) \cong f(\hat{x},u) + \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}} [x - \hat{x}],$$

The Jacobian matrix $A(\hat{x})$ of $f(\hat{x})$ is found to be

$$A(\hat{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial \hat{x}_1} & \cdots & \frac{\partial f_1}{\partial \hat{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \hat{x}_1} & \cdots & \frac{\partial f_m}{\partial \hat{x}_n} \end{bmatrix}. \tag{5.2}$$

The covariance matrix P is then propagated through the Lyapunov equation

$$\dot{P} = A(\hat{x})P + PA(\hat{x})^{'} + GQG^{'}. \tag{5.3}$$

Since the initial values $\hat{x}_0$ and P_0 are known, ODEs (5.1) $\sim$ (5.3) form an initial value problem that can be solved using commercial ODE solvers such as Matlab$^{\circledR'}$s *ode*45.

The predictions at any sampling point kT are given by

$$\hat{x}(kT) = \hat{x}_k^-, \; P(kT) = P_k^-. \tag{5.4}$$

Since measurements are only available at the sampling time, the Kalman gain is calculated based on the predicted discrete covariance:

$$K_k = P_k^- H_k^{'}(H_k P_k^- H_k^{'} + R)^{-1}, \text{where } H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^-}. \tag{5.5}$$

In a last step, corrections are made based upon the predictions and the new available measurement.

$$P_k = (I - K_k H_k)P_k^- \tag{5.6}$$

$$\hat{x}_k = \hat{x}_k^- + K_k[y_k - h(\hat{x}_k^-)]. \tag{5.7}$$

Table XI provides a summary of this algorithm. In this algorithm, both mean

where $\hat{x}$ is close to x. Taking the expectation of both sides of the above equation gives

$$E\{f(x,u)\} = f(\hat{x},u),$$

Therefore the state estimate $\hat{x}$ is predicted via $\dot{\hat{x}} = f(\hat{x},u)$.

$\hat{x}$ and covariance matrix P are solved in a continuous manner. The numerical solver determines the step size for integration of $\hat{x}$ and P during each sampling interval. This increases the accuracy of integration compared to methods with a fixed step size. This method produces an error resulting from linearization at each integration step only.

Table XI. Summary of Procedure for Algorithm 1

Initialization	$\hat{x}_0 = \bar{x}_0,\ P_0 = P_{x_0}$
Prediction	$\dot{\hat{x}} = f(\hat{x}, u)$ $\dot{P} = A(\hat{x})P + PA(\hat{x})' + GQG'$, where $A(\hat{x}) = \frac{\partial f}{\partial x}\Big\vert_{x=\hat{x}}$ $\hat{x}(kT) = \hat{x}_k^-,\ P(kT) = P_k^-$
Kalman gain	$K_k = P_k^- H_k'(H_k P_k^- H_k' + R)^{-1}$, where $H_k = \frac{\partial h}{\partial x}\Big\vert_{x=\hat{x}_k^-}$
Correction	$P_k = (I - K_k H_k)P_k^-$ $\hat{x}_k = \hat{x}_k^- + K_k[y_k - h(\hat{x}_k^-)]$

Remark: There are possible alternatives to this implementation. In Algorithm 1, the Jacobian matrix A is considered to be time-varying for solving the ODEs. If A is assumed to be time-invariant, the matrix could be calculated at each sampling interval, i.e., $A(\hat{x}_k) = \frac{\partial f}{\partial x}\Big\vert_{x=\hat{x}_k}$. The ODEs (5.1) and (5.3) thus can be solved separately (**Algorithm 1.1**), which may reduce computation costs. However,the computation accuracy may be decreased concurrently since the error is affected by linearization at each sampling interval, which is usually significantly larger than the integration step size used in Algorithm 1.

2. Implementations via Linearization and Discrete KF for Covariance Prediction

In the second algorithm (**Algorithm 2**), linearization of the nonlinear continuous-time model along its trajectory is performed and the covariance predictions are computed from discrete information.

As in Algorithm 1, the nonlinear differential equation

$$\dot{\hat{x}} = f(\hat{x}, u) \tag{5.8}$$

is used for predicting the state vector $\hat{x}_k$. The Jacobian matrix A is then computed at each sampling time,

$$A(\hat{x}_k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k}. \tag{5.9}$$

The linearized model is given by

$$\dot{\tilde{x}} = A(\hat{x}_k)\tilde{x} + B\tilde{u} + f(\hat{x}_k, u_k), \tag{5.10}$$

where $\tilde{x} = \hat{x} - \hat{x}_k$ and $\tilde{u} = u - u_k$.

Solving the ODEs (5.10) and (5.3) for computing the discrete covariance matrix P results in

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k \tag{5.11}$$

$$P^{-}_{k+1} = A_k P A_k^{'} + GQ_kG^{'} \tag{5.12}$$

where $A_k = e^{A(\hat{x}_k)T}$ is the state transition matrix, $GQ_kG^{'} = \int_0^T e^{A(\hat{x}_k)\tau} GQG^{'} e^{A(\hat{x}_k)^{'}\tau} d\tau$ is the process noise matrix, and $B_k = \int_0^T e^{A(\hat{x}_k)\tau}[Bu(T-\tau) + f(\hat{x}_k, u_k)]d\tau$ is the input matrix with $B = \left.\frac{\partial f}{\partial u}\right|_{u=u_k}$.

Corrections for mean and covariance matrices using a Kalman filter are computed by Eqs. (5.5) $\sim$ (5.7) as discrete measurements are the source for estimation updates.

Table XII summarizes the procedure of this algorithm. During each measurement sampling interval, the nonlinear system is considered as a linear first-order system with constant coefficients, which results in Eqn. (5.11) and (5.12). This incurs a larger error for computing the covariance matrix P than Algorithm 1, where P is integrated using a time-varying state transition matrix A.

Table XII. Summary of Procedure for Algorithm 2

Initialization	$\hat{x}_0 = \bar{x}_0,\ P_0 = P_{x_0}$
Prediction	$\dot{\hat{x}} = f(\hat{x}, u)$ $P^-_{k+1} = A_k P A'_k + G Q_k G'$ where $A(\hat{x}_k) = \frac{\partial f}{\partial x}\Big\vert_{x=\hat{x}_k},\ A_k = e^{A(\hat{x}_k)T},$ $GQ_kG' = \int_0^T e^{A(\hat{x}_k)\tau} GQG' e^{A(\hat{x}_k)'\tau} d\tau$
Kalman gain	$K_k = P^-_k H'_k (H_k P^-_k H'_k + R)^{-1}$, where $H_k = \frac{\partial h}{\partial x}\Big\vert_{x=\hat{x}^-_k}$
Correction	$P_k = (I - K_k H_k) P^-_k$ $\hat{x}_k = \hat{x}^-_k + K_k[y_k - h(\hat{x}^-_k)]$

An alternative (**Algorithm 2.1**) to this algorithm is to use an Euler approximation for discretization of continuous-time models. The matrices for covariance prediction are then replaced by the following:

$$A_k = I + A(\hat{x}_k)T,\ Q_k = QT,\ \text{and } B_k = BTu(kT). \tag{5.13}$$

Due to the lower accuracy of Euler's method compared to, e.g., a Runge-Kutta method used by conventional ODE solvers, Algorithm 2.1 will in theory result in poorer performance than Algorithm 2. The errors are due to both linearization and discretization of nonlinear continuous-time models. Additionally, when large sampling times are used, this method may produce unstable results.

3. Implementations via Discretization Followed by Linearization

In this algorithm (**Algorithm 3**), the nonlinear continuous-time model is discretized first and then linearized along its trajectory. It is well known that ODE solvers or Euler approximations are two common methods used for discretization of continuous-time models. Euler's method discretizes models with a fixed step size while the step size for discretization is adjusted for different dynamic behaviors when ODE solvers are used. Therefore ODE solvers can result in more accurate discrete data than if Euler approximations are used.

In order to use ODE solvers such as Matlab$^{®'}$s *ode*45 for discretization, however, a continuous noise signal $w(t)$ in Eq. (2.9) is needed. This is unlikely to be simulated and implemented in a digital computer. As a compromise, a discrete signal $w_k = w(kT)$ can be generated. After solving an initial value problem for finding a solution of the ODE

$$\dot{x} = f(x, u), \tag{5.14}$$

w_k is linearly added to $x_k = x(kT)$ at each sampling interval kT.

Once a discrete-time model is obtained, the next step is to compute the Jacobian matrix A_k of the nonlinear function f at each time step kT. Since numerical discretization of the continuous-time model is executed at the first step, no analytical form for the model is available and A_k also needs to be computed numerically. One approach uses the sensitivity matrix

$$\dot{A} = \frac{\partial f}{\partial x'} A. \tag{5.15}$$

The state vector predictions $\hat{x}_k^-$ and the Jacobian matrix A_k of f can be solved simultaneously

$$\dot{\hat{x}} = f(\hat{x}, u) \tag{5.16}$$

$$\dot{A} = \frac{\partial f}{\partial \hat{x}'} A \tag{5.17}$$

with $\hat{x}_0 = \bar{x}_0$ and $A_0 = I$.

Remarks: One alternative (**Algorithm 3.01**) is to compute A_k by using finite difference such as central differences, i.e.,

$$A_k = \frac{f(\hat{x}_k + \Delta x) - f(\hat{x}_k - \Delta x)}{2\Delta x}. \tag{5.18}$$

Numerically it is non-trivial to find an appropriate difference Δx. A finite difference method may be less accurate than solving the sensitivity equation.

Once the state vector estimate $\hat{x}_k^-$ and the matrix A_k are computed using Eqs. (5.16) and (5.17), the covariance matrix P is computed

$$P_{k+1}^- = A_k P_k A_k' + G Q_k G', \tag{5.19}$$

where Q_k is approximated by QT.

The Kalman gain is calculated in the same discrete-time form as in Eq. (5.5),

$$K_k = P_k^- H_k' (H_k P_k^- H_k' + R)^{-1}, \text{where } H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^-}. \tag{5.20}$$

Updates for the state estimate $\hat{x}_k$ and the covariance matrix P_k are made in the same manner as in Eqs. (5.6) and (5.7),

$$P_k = (I - K_k H_k) P_k^- \tag{5.21}$$

$$\hat{x}_k = \hat{x}_k^- + K_k [y_k - h(\hat{x}_k^-)]. \tag{5.22}$$

Table XIII summarizes the procedure, where the sensitivity matrix is used for

Table XIII. Summary of Procedure for Algorithm 3

Initialization	$\hat{x}_0 = \bar{x}_0,\ P_0 = P_{x_0},\ A_0 = I$
Prediction	$\dot{\hat{x}} = f(\hat{x}, u), \dot{A} = \frac{\partial f}{\partial \hat{x}'} A$ $\hat{x}(kT) = \hat{x}_k^-,\ A(kT) = A_k$ $P_{k+1}^- = A_k P_k A_k^{'} + GQ_kG^{'}$
Kalman gain	$K_k = P_k^- H_k^{'}(H_k P_k^- H_k^{'} + R)^{-1}$, where $H_k = \frac{\partial h}{\partial x}\Big\vert_{x=\hat{x}_k^-}$
Correction	$P_k = (I - K_k H_k) P_k^-$ $\hat{x}_k = \hat{x}_k^- + K_k[y_k - h(\hat{x}_k^-)]$

computing the Jacobian matrix of the nonlinear function f at each time step.

As mentioned in Algorithm 2.1, a discretization of continuous-time models can be replaced by an Euler approximation. Algorithm 3 then results in:

$$\hat{x}_{k+1}^- = \hat{x}_k + Tf(\hat{x}_k, u_k) \tag{5.23}$$

$$A_k = I + T \frac{\partial f(\hat{x}, u_k)}{\partial \hat{x}}\Bigg\vert_{\hat{x}=\hat{x}_k} \tag{5.24}$$

$$P_{k+1}^- = A_k P_k A_k^{'} + GQTG^{'}, \tag{5.25}$$

which is referred to as **Algorithm 3.1** here. The Kalman gain and correction equations remain the same. Unstable filters may be generated when the step size is large for the Euler approximation. This method results in errors from linearization of the nonlinear model along its trajectory at each sampling interval and discretization of continuous-time models for computing the covariance matrix, similar to Algorithm 2.1. Additionally, an error is resulting from prediction of the states using an Euler approximation. It is estimated that this method performs worse than Algorithm 2.1.

C. Discussions

The algorithms discussed in the last section involve linearization and discretization of nonlinear continuous-time models at different steps. The sequence and the specific technique for executing them define each algorithm. To be more specific, Algorithms 1 and 1.1 execute linearization of the model along its trajectory first and then use a continuous KF to predict the covariance P. Algorithms 2 and 2.1 also linearize the model along its trajectory first but use a discrete KF to propagate P. Algorithms 3, 3.01 and 3.1 first discretize the model and then linearize the model along its trajectory for computing P. With respect to linearization, Algorithm 1, 3 and 3.01 evaluate the Jacobian matrix A continuously while the others compute it only at the sampling time.

Table XIV. Summary of the Algorithms

<table>
<tr><td><b>Initialization</b></td><td colspan="3">$\hat{x}_0 = \bar{x}_0,\ P_0 = P_{x_0},\ A_0 = I$</td></tr>
<tr><td><b>Mean Prediction</b></td><td colspan="3">$\dot{\hat{x}} = f(\hat{x}, u)$
$(Exception : \hat{x}_{k+1}^- = \hat{x}_k + Tf(\hat{x}_k, u_k)\ for\ Algorithm\ 3.1)$</td></tr>
<tr><td rowspan="3"><b>Covariance Prediction</b></td><td></td><td><b>Evaluate A at</b> $x(t)$</td><td><b>Evaluate A at</b> x_k</td></tr>
<tr><td>$\dot{P} = A(\hat{x})P$
$+PA(\hat{x})' + GQG'$</td><td>Algorithm 1</td><td>Algorithm 1.1</td></tr>
<tr><td>$P_{k+1}^- = A_k P_k A_k'$
$+GQ_kG'$</td><td>Algorithm 3, 3.01</td><td>Algorithm 2, 2.1, 3.1</td></tr>
<tr><td><b>Kalman gain</b></td><td colspan="3">$K_k = P_k^- H_k'(H_k P_k^- H_k' + R)^{-1}$, where $H_k = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_k^-}$</td></tr>
<tr><td><b>Correction</b></td><td colspan="3">$P_k = (I - K_k H_k)P_k^-$
$\hat{x}_k = \hat{x}_k^- + K_k[y_k - h(\hat{x}_k^-)]$</td></tr>
</table>

In spite of the classifications, all algorithms share the same formulation for state predictions and mean and covariance corrections with the exception of Algorithm 3.1 where the state is predicted via an Euler approximation. The main differences between the methods are given by the computation of the covariance matrix P and the approaches used to evaluate the Jacobian matrix A. Table XIV summarizes the results. Algorithm 1 and 1.1 propagate P via a continuous KF while a discrete KF is used to compute P for the other algorithms. Further, Algorithms 1, 3 and 3.01 evaluate A continuously while Algorithms 1.1, 2 and 2.1 compute A at each sampling point x_k.

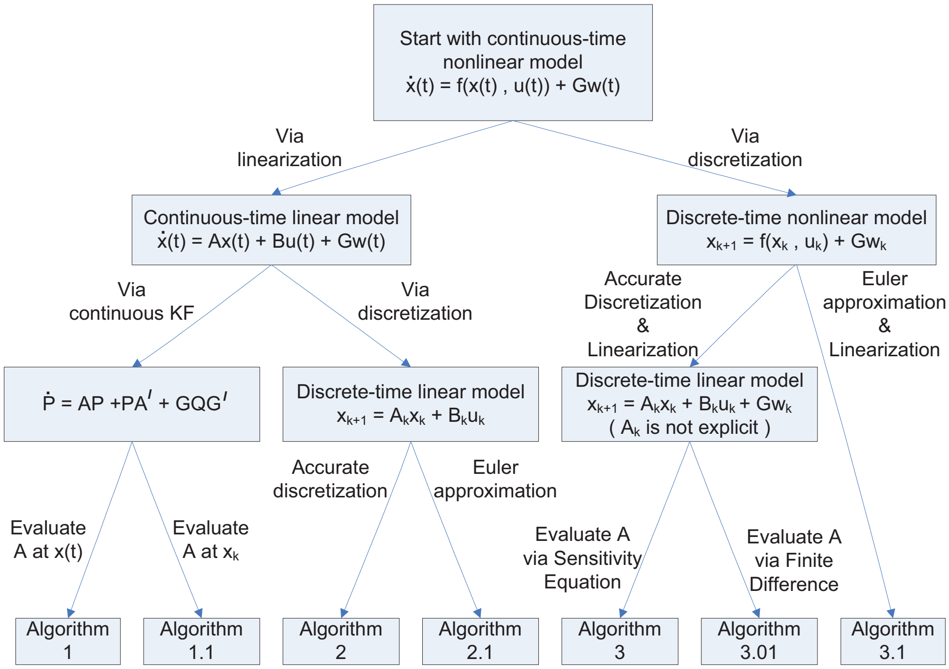

Start with continuous-time nonlinear model
$\dot{x}(t) = f(x(t), u(t)) + Gw(t)$
Via linearization
Via discretization
Continuous-time linear model
$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t)$
Discrete-time nonlinear model
$x_{k+1} = f(x_k, u_k) + Gw_k$
Via continuous KF
Via discretization
Accurate Discretization & Linearization
Euler approximation & Linearization
$\dot{P} = AP + PA' + GQG'$
Discrete-time linear model
$x_{k+1} = A_k x_k + B_k u_k$
Discrete-time linear model
$x_{k+1} = A_k x_k + B_k u_k + Gw_k$
(A_k is not explicit)
Accurate discretization
Euler approximation
Evaluate A at x(t)
Evaluate A at x_k
Evaluate A via Sensitivity Equation
Evaluate A via Finite Difference
Algorithm 1
Algorithm 1.1
Algorithm 2
Algorithm 2.1
Algorithm 3
Algorithm 3.01
Algorithm 3.1

Fig. 20. Comparison of Different Algorithms for Implementing EKF.

Figure 20 provides a graphic overview of the discussed EKF implementations for a continuous-time model with discrete-time measurements.

Remarks:

1. Algorithms 1 and 1.1 predict P via solution of the Lyapunov Eq. $\dot{P} = A(\hat{x})P + PA(\hat{x})' + GQG'$ which distinguishes them from the other algorithms. The difference between 1 and 1.1 lies in the Jacobian matrix calculation of model functions. Algorithm 1 computes the Jacobian A at each integration step $x(t)$ while Algorithm 1.1 computes A only at time kT which implies that A_k is time-invariant during each integration period. Algorithm 1.1 should be less accurate than Algorithm 1 though it can save computation time.

2. Algorithms 2, 2.1, 3, 3.01 and 3.1 propagate P in a discrete manner $P_{k+1}^{-} = A_k P_k A_k' + GQ_k G'$.

3. Algorithms 2 and 2.1 linearize the model along its trajectory first and then carry out discretization of the state equations. Algorithm 2 uses $A_k = e^{A(\hat{x}_k)T}$ for discretization while Algorithm 2.1 makes use of an Euler approximation which results in $A_k = I + A(\hat{x}_k)T$. If the sampling time T is small, then Algorithm 2.1 may produce results comparable to Algorithm 2. Additional note: For computing the integral $\int_0^T e^{A(\hat{x}_k)\tau} GQG' e^{A(\hat{x}_k)'\tau} d\tau$ for Algorithm 2, a simple alternative involving the matrix exponential computation is presented by Van Loan [172].

4. Algorithms 3, 3.01 and 3.1 implement discretization of the model first and then use linearization for the same model along its trajectory. Similar to Algorithms 2 and 2.1, the discretization scheme distinguish Algorithms 3 and 3.01 from Algorithm 3.1 where an Euler approximation is used. Therefore both Algorithm 3 and 3.01 should be superior to Algorithm 3.1 in terms of accuracy. Algorithm 3 and 3.01 differ in the approach for computing the Jacobian matrix A. The

former uses solution of the sensitivity equation while finite differences are chosen for the latter.

5. Algorithms 2.1 and 3.1 are identical except for how they predict the state $\hat{x}$. The former uses direct integration while the latter makes use of an Euler approximation. Therefore Algorithm 2.1 potentially performs better than 3.1. However, both of them may produce unstable filters when the sampling interval is large due to the Euler approximation for discretizing the model.
6. Algorithm 2 is identical to Algorithm 1.1 although the implementations are not the same. Please refer to the Appendix for a detailed proof.

D. Case Studies

To evaluate the performance of EKF using each implementation, the algorithms given in Section C have been applied to models including ones with mild nonlinearity as well as some with strong degrees of nonlinearity and a large number of scenarios such as different operating conditions, different tuning parameters Q and R, and different process and measurement noise levels. Monte Carlo simulations of 50 runs were carried out for each scenario. The performance is evaluated by the overall mean-squared error (MSE). The MSE is first averaged over all simulations for each time point and then over time to indicate the long-term behavior of each estimator and the distribution of errors over time.

This section shows two representative case studies: Van der Pol oscillator and a CSTR with Van de Vusse reactions.

1. Van der Pol Oscillator

The model for this case study is the Van der Pol oscillator which is widely used in the literature. The Van der Pol oscillator is a type of nonconservative oscillator with nonlinear damping. It evolves in time according to the second order differential equation:

$$\frac{d^2x}{dt^2} + \mu(x^2 - 1)\frac{dx}{dt} + x = 0 \tag{5.26}$$

where x is the position coordinate which is a function of the time t, and μ is a scalar parameter indicating the strength of the nonlinear damping. When $\mu < 0$ the system will be damped and exhibit an unstable limit cycle. When $\mu = 0$, there is no damping function and the system is a form of the simple harmonic oscillator. When $\mu \geq 0$, the system will enter a stable limit cycle where energy continues to be conserved.

The state and measurement equations chosen in the study of a stable limit cycle are

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 0.2(1 - x_1^2)x_2 - x_1 \\ y &= [x_1\ x_2]^T \end{aligned} \tag{5.27}$$

The system is discretized with a sampling interval of 0.1. The remaining EKF filter parameters are

$$x_0 = \begin{bmatrix} 0.5 & 0 \end{bmatrix}^T, \qquad \hat{x}_0 = \begin{bmatrix} 5 & -1 \end{bmatrix}^T, \tag{5.28}$$

$$\hat{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad Q_0 = R_0 = 10^{-3}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.29}$$

Figure 21 $\sim$ Figure 24 shows the results of each algorithm for EKF implemen-

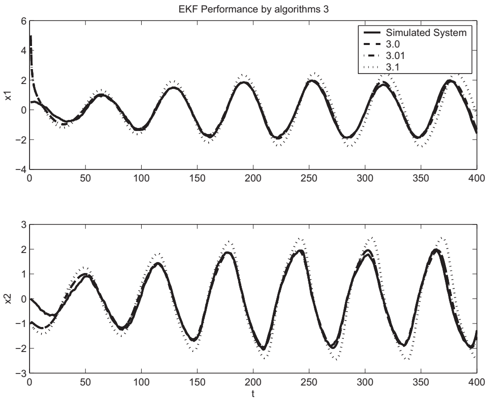


Fig. 21. EKF Performance by Algorithm 3 and Its Derivatives for van der Pol Oscillator.

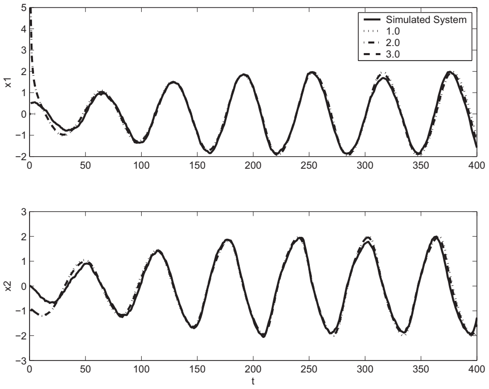


Fig. 22. EKF Performance Comparison for van der Pol Oscillator.

tations. All algorithms provide good estimates for both system states except that Algorithm 3.1 where Euler approximation is used for discretization generates a significant large error than the other algorithms.

2. Production of Cyclopentanol in a CSTR with van de Vusse Reaction.

The first case study in last section provides a overview of the performance of each EKF implementation. In this section, a detailed examination on the performance of each algorithm is conducted through an isothermal nonlinear CSTR with a competing

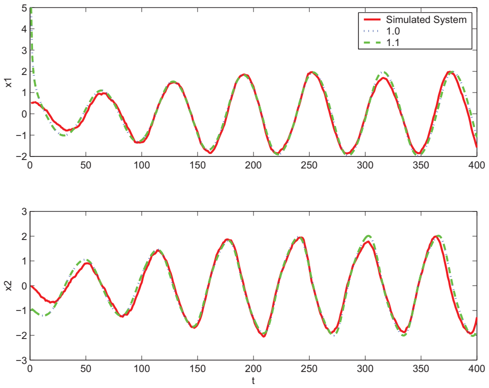


Fig. 23. EKF Performance by Algorithm 1 and Its Derivatives for van der Pol Oscillator.

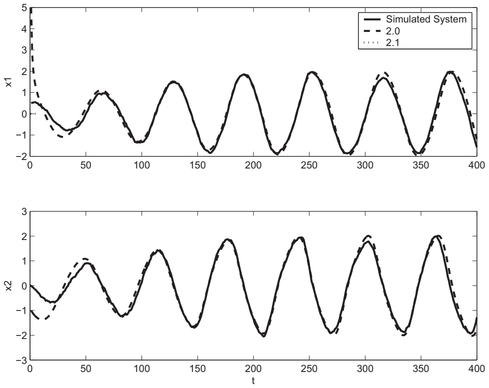


Fig. 24. EKF Performance by Algorithm 2 and Its Derivatives for van der Pol Oscillator.

side reaction governed by van de Vusse reaction kinetics used for the production of Cyclopentanol [168]:

$$\begin{aligned} &\mathrm{A} \xrightarrow{k_1} \mathrm{B} \overset{k2}{\to} \mathrm{C} \\ &2\mathrm{A} \overset{k3}{\to} \mathrm{D}. \end{aligned} \tag{5.30}$$

Component A is the the reactant cyclopentadiene, B is the product cyclopentanol, C and D are the side products cyclopentandiol and dicylopentadiene. The nonlinear system model is given by the following three differential equations:

$$\begin{aligned}\frac{dC_A}{dt} =& \frac{u}{V}(C_{Ain} - C_A) - k_1 e^{-E_1/RT} C_A - k_3 e^{-E_3/RT} C_A^2 && (5.31)\\ \frac{dC_B}{dt} =& -\frac{u}{V}C_B + k_1 e^{-E_1/RT} C_A - k_2 e^{-E_2/RT} C_B && (5.32)\\ \frac{dT}{dt} =& \frac{1}{\rho c_p}[k_1 e^{-E_1/RT} C_A(-\Delta H_1) + k_2 e^{-E_2/RT} C_B(-\Delta H_2) && (5.33)\\ & + k_3 e^{-E_3/RT} C_A^2(-\Delta H_3)] + \frac{u}{V}(T_{in} - T) + \frac{Q}{V\rho c_p}\end{aligned}$$

where the feed flow rate u is the only controlled variable. The values of the parameters can be found in the work by Hahn and Edgar [169].

The nonlinear model exhibits multiple steady states, of which the upper steady state ($C_{Ass} = 2.49$mol/L; $C_{Bss} = 1.1$mol/L; $T_{ss} = 411K$; $u = 800L/h$) is chosen as the point of operation. The measurable variable is assumed to be the reactor temperature T. Initial conditions are chosen to be $\hat{x}_0 = \begin{bmatrix} 2.5 & 1.09 & 411 \end{bmatrix}^T$ and all process variables were scaled to be dimensionless using the upper steady state as the nominal point. The sampling time for the measurements is 0.02 min.

The remaining filter parameters after scaling are given by

$$\begin{aligned}\hat{P}_0 &= diag\{100, 100, 100\},\\ Q &= diag\{10^{-2}, 10^{-2}, 10^{-2}\},\\ R &= diag\{10^{-2}, 10^{-2}\}.\end{aligned}$$

Table XV. MSEs for Algorithms ($\Delta t = 0.02, R = 0.01I$) for EKF Implementations

Algorithms	1	1.1	2	2.1	3	3.01	3.1
MSEs	0.626	0.627	0.627	1.039	0.653	0.675	1.537

The overall MSEs for $\Delta t = 0.02, R = 0.01I$ are shown in Table XV and it can

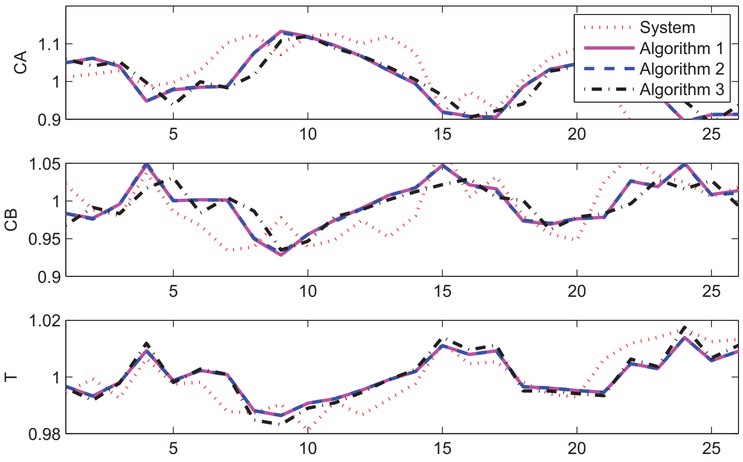


Fig. 25. EKF Performance Comparison for the van de Vusse Reactor.

be seen that Algorithms 1, 2 and 3 produce relatively small MSEs. The predictions of the states made by these algorithms are also shown in Figure 25.

Compared to Algorithm 1, Algorithm 1.1 also performs reasonably well although its performance is slightly worse than Algorithm 1 for large variations in the operating conditions.

As discussed in Remark 3, Algorithm 2 produces more accurate results than Algorithm 2.1 due to the Euler approximations used in the latter. This can be seen by comparing the MSEs, which are 0.627 for Algorithm 2 and 1.039 for Algorithm 2.1. As shown in the Appendix, Algorithm 1.1 and 2 are identical and generate the same MSEs.

The MSE Results using Algorithms 3, 3.01 and 3.1 are 0.653, 0.675, and 1.537, respectively. Algorithm 3.01 has poorer performance than Algorithm 3 because of the finite differences used for computing the Jacobian instead of solving the sensitivity equation. The difference between Algorithm 3 and 3.1 is significant as Algorithm 3.1 failed to produce reasonably good estimates when the sampling interval is large due to the Euler approximation.

Methods that involve Euler approximations, such as Algorithm 2.1 and 3.1, produce relatively large values of the MSEs as given by 1.039 and 1.537, respectively. The two methods share the formulation for the covariance matrix propagation but differ in their mean predictions.

Further investigations are also carried out for different measurement noise levels and different sampling interval lengths, and are summarized in Table XVI. When measurement noise is relatively small ($\Delta t = 0.02, R = 0.0001I$), methods involving Euler approximations such as Algorithm 2.1 and 3.1 generate acceptable MSEs. However, for large measurement noise($\Delta t = 0.02, R = 1I$), both Algorithm 2.1 and 3.1 exhibit unstable properties due to the Euler Approximation. Similarly, all algorithms perform comparably well when the sampling time is relatively small ($\Delta t = 0.002$). However unstable filters can result from Algorithm 2.1 or 3.1 when large sampling intervals are used. Therefore approaches such as Algorithm 2.1 or 3.1 should only be considered when measurement noise levels and the sampling time are relatively small.

Table XVI. Summary of MSEs for All Algorithms for EKF Implementations

Algorithm	1	1.1	2	2.1	3	3.01	3.1
$\Delta t = 0.02, R = 0.0001I$	0.250	0.247	0.247	0.339	0.361	0.368	0.420
$\Delta t = 0.02, R = 1I$	0.686	0.686	0.686	6.158	0.689	0.691	11.370
$\Delta t = 0.002, R = 0.01I$	0.151	0.152	0.152	0.151	0.151	0.151	0.151
$\Delta t = 0.2, R = 0.01I$	3.233	3.249	3.249	23.842	3.059	3.064	423.310

Scenarios with input changes are investigated for different measurement noise levels and different sampling times. Table XVII provides a summary of MSE results for the case where the input flowrate increases from $800L/h$ to $1200L/h$ at the

1500th sampling interval. This set of results are consistent with the case without input changes or any disturbance. Algorithm 1, 2 and 3 provide reasonably good performance for EKF while Algorithm 2.1 and 3.1, which involve Euler approximations, generate unstable filters.

Table XVII. MSEs for Algorithms with a 50% Input Change for EKF Implementations

Algorithm	1	1.1	2	2.1	3	3.01	3.1
$\Delta t = 0.02, R = 0.01I$	1.297	1.297	1.297	17.090	1.296	1.296	380.76
$\Delta t = 0.02, R = 0.0001I$	0.978	0.986	0.986	7.658	1.095	1.140	24.654
$\Delta t = 0.2, R = 0.01I$	5.678	5.692	5.692	NaN	5.763	5.768	NaN

E. Conclusions

This chapter compares different implementations of EKF for a class of continuous-time nonlinear models with discrete-time measurements. The algorithm can be classified by the sequence and methods used for linearization and discretization of nonlinear continuous-time models. The main difference between the methods lies in the methodology for computing the covariance matrix P. The conclusions are that continuously predicting P for EKF results in an accurate implementation. Evaluating P at discrete times can also be applied. In this case, good performance can be expected if P is obtained from integrating the continuous-time equation or if the sensitivity equation is used for computing the Jacobian matrix A. Instead, if a finite difference approach is chosen for computing A, the sampling time of the finite difference scheme needs to be small for acceptable performance of EKF. Approaches involving Euler approximations show good behavior only when the sampling interval is reasonably small and therefore they are not recommended for processes with long sampling time intervals.

CHAPTER VI

CONCLUSIONS

A. Findings and Contributions

This dissertation addresses some key issues in nonlinear estimation for process monitoring and fault diagnosis. Specifically, the application of unscented Kalman filters, improvement of moving horizon estimation via computation of the arrival cost and different implementations of extended Kalman filters are investigated.

Chapter III compares UKF with traditional estimation techniques such as LKF, EKF and MHE. It is found that the unscented Kalman filter provides a good estimate of the states and parameters and is comparable in performance to EKF when system nonlinearities are not significant. UKF outperforms EKF when strong nonlinearities exist and the measurement noise levels are high.

Additionally, when the structures of the process and measurement functions are not differentiable or are discontinuous, UKF also offers a benefit because it does not require to calculate Jacobian matrices. Moreover, UKF is applicable to any black box models given the properties of input and output information. This distinguishes UKF from EKF and makes UKF preferable in industrial applications as first principle models are often not available.

However, both EKF and UKF may fail to converge to the true values of the states for constrained problems; MHE may be a better option for constrained problems, however, MHE results in a higher computational burden.

Another important property of these advanced estimation algorithms is that UKF reduces to the Kalman filter for linear systems; simulations were conducted to investigate this finding. MHE also simplifies to the Kalman filter for unconstrained linear systems.

In chapter IV a MHE formulation where the arrival cost is computed by UKF is proposed and MHE performance is further investigated. The unscented transformation and a set of selected sigma points are used to compute the covariances and then the arrival cost. The selection procedure for the sigma points is the same as the one used for unscented Kalman filtering if the constraints are inactive, however, a modification is used that satisfies the state variable constraints when the constraints are active. Linearization of the model is not required for the presented approach.

The presented method performed slightly better than the commonly used eMHE for all investigated cases. Therefore, the method can be a promising alternative for approximating the arrival cost for MHE.

Case studies without the approximating arrival cost were also investigated. It was shown that the estimates were much less accurate than those with arrival cost approximation, which implied that it is important to compute arrival cost for implementing MHE accurately.

Scaling is a key factor for obtaining an optimal solution for MHE in the application of numerical algorithms. It is common in chemical engineering to be concerned with a system which is composed of concentrations and temperatures. The units of concentrations and temperatures usually differ by several orders of magnitude. Poor scaling of such a system results in ill-posed problems.

Chapter V compares different implementations of EKF for a class of continuous-time nonlinear models with discrete-time measurements. The algorithm can be classified by the sequence and methods used for linearization and discretization of nonlinear continuous-time models. The main difference between the methods lies in the methodology for computing the covariance matrix P. The conclusions are that continuously predicting P for EKF results in an accurate implementation. Evaluating P at discrete times can also be applied. In this case, good performance can be expected if P is obtained from integrating the continuous-time equation or if the sensitivity equation is used for computing the Jacobian matrix A. Instead, if a finite difference approach is chosen for computing A, the sampling time of the finite difference scheme needs to be small for acceptable performance of EKF. Approaches involving Euler approximations show good behavior only when the sampling interval is reasonably small and therefore they are not recommended for processes with long sampling time intervals.

This portion of the work provides a good reference for practicing engineers as it is found from the literature that Euler approximations are commonly chosen for EKF implementations where inaccuracy of EKF performance and biased conclusions are usually given.

B. Suggestions for Further Work

Besides the EKF, UKF and MHE discussed in the dissertation there are several important filtering methods, e.g. particle filtering, ensemble Kalman filtering. Particle filter (PF), also known as the sequential Monte Carlo method (SMC), is an advanced estimation technique based on simulations. It can be an alternative to EKF or UKF with the advantage that the Bayesian optimal estimate is approximated directly by the method and a more accurate estimation can be made than either the EKF or UKF. However, particle filtering requires much more computational resources since a sufficiently large number of samples are required in the simulation. The ensemble Kalman filter (EnKF) has been proposed as a version of the Kalman filter where the covariance matrix is replaced by the sample covariance. EnKF is related to the PF but EnKF makes the assumption that all probability distributions involved are Gaussian. Due to this assumption, it is much more efficient than the particle filter. These approaches would benefit from detailed studies to gain a more complete understanding of nonlinear estimation. Each filtering technique can also be combined with optimization strategies for nonlinear systems with constraints, where efficient numerical methods are commonly required. This is an area that industry shows great interest in.

It may also be interesting to investigate different implementations of UKF or MHE since the implementation of a filtering method is not unique. This dissertation addresses the work of different implementations for EKF and it is evident that different implementations lead to significantly different mean squared errors and the difference in the results is even larger than that returned by different filtering methods. Investigation of the implementation is of great importance to attain a good performance for a filter. For UKF there are also different implementations, e.g. calculation of the squared root of the covariance matrix and these different implementations in UKF can be investigated in the future work. For MHE the implementation plays an even more important role since a lot of numerical algorithms for nonlinear optimization exist and the estimate of MHE is highly dependent on the quality of the optimal solution found. Generally it is difficult in practice to guarantee that the global optimal solution is found. Therefore it may be worthwhile to investigate the decrease in the performance of MHE if a local solution is found.

The horizon length of MHE is an important tuning parameter. The performance of MHE improves as the horizon length increases. Theoretically, the horizon length needs to be greater than the order of the system in order to make MHE stable. Though sufficiently large moving horizon leads to diminishing returns, the computational cost also increases with the length of horizon. The choice of horizon length is a compromise between computation efforts and estimation accuracy. At the same time, time constant of a process may affect the selection of horizon length. Similar to Model predictive control strategies, the larger the time constant, i.e., the slower the system dynamics, the greater the horizon length is required. On the other hand, it is difficult to design a MHE method if the time constant of a system is significantly small. Investigating how the time constant affects the choice of horizon length and the computational cost would be beneficial to improve MHE performance. Adaptive horizon length may also enhance MHE performance as a fixed horizon length is sub-optimal. Work on choosing horizon length adaptively would be valuable as well.

The comparison of different filters in the dissertation focuses on the performance where the model used in filtering is assumed to be accurate. However, in practice mismatch of model is inevitable. Investigation of robustness of filters is important for applications to real systems. A large number of methods are developed to search for optimal H^2 filtering to minimize the variance of the estimate. The robust H^2 filtering aims to design a filter such that the worst case mean square estimation error is minimized for all admissible uncertainties under the assumption of the process noise input and the structure of the model uncertainty. Robust filtering is able to guarantee the performance of the filter under the model uncertainty. However robust filtering is much more difficult than the conventional filters even for linear time invariant systems and reports of the nonlinear robust filtering are rare.

EKF is a linearization method and it is recognized that EKF is able to achieve an adequately good performance if the nonlinearity of a model is mild while it has a poor performance if the nonlinearity of a system is severe. However, quantitatively there is no criteria to measure to what degree of the nonlinearity EKF is likely to fail. Many methods have been developed to test the nonlinearity of a dynamic system [173] [174] [175] [176] [169] and to check if a linearized model accurately approximates the nonlinear model. If some quantitative relationship can be built between a nonlinearity measure and the performance of EKF it can be used to provide a guide for choosing a proper filter given the fact that EKF is often the first choice among various filters and an alternative advanced filter is desired only when EKF fails.

It can be also helpful for selection of a filter to investigate how large the difference in mean squared error (MSE) of state estimates is practically significant as the MSE is often used to evaluate the performance of filters. Subsequently, it may also be possible to discover if the difference in MSE may generate different results for a fault diagnosis system that is based on a certain estimator, i.e., an approach of fault diagnosis is sensitive to the state estimates provided by a selected filter. If a fault diagnosis is not sensitive to the state estimates a simple filter of EKF may be a proper choice even though it returns an apparently larger MSE than advanced one. While the difference in MSE significantly affects the performance of a fault diagnosis an advanced filter may be needed even though the difference in MSE is small in numerical value.

Applying available estimation techniques to fault diagnosis could be advantageous such that the location and nature of a fault can be diagnosed. For example, an actuator fault will influence the behavior of the process which will then result in sensor readings of several process variables deviating from normal behavior. While this may set off several alarms at the same time, this does not mean that there are several faults occurring instantaneously which have to be counteracted. Instead, the abnormal readings are the result of one fault affecting several process variables to varying degrees. Desired is a fault diagnosis system via nonlinear estimation that is able to determine that an actuator fault has occurred and what needs to be done in order to counteract this effect. At the same time it could also indicate that it is sufficient to sound one alarm for the actuator instead of several for each of the variables affected by the fault. Particularly, it may be interesting to find out what the statistical properties of the estimates are after they undergo a nonlinear transformation since they could be used to be a guide for setting up thresholds for alarms. In most cases it is believed that the statistical properties such as the second moment, i.e., covariance, are changed and therefore it may not be reasonable to use the previously known covariance σ based on industrial experience for setting the alarm thresholds in alarm management systems.

In addition to applying such a method for fault diagnosis, it may also be used for design, verification and validation of HAZOP results. While application of HAZOP studies to new facilities is always required, the constant demand of HAZOP efforts as a part of Management of Change issues has indicated that alarms are a common and growing addition to the plant safeguard system, but without any real study as to the effectiveness of the alarm or its impact on the identification of that particular fault to the DCS operator. Hence growing alarm systems can sometimes cloud the actual event from the operator, with resulting alarm floods. Since it will be possible to study the effect of different faults on a system, it can be directly used in order to determine thresholds for the individual alarms. It is especially important to determine with high confidence when a fault can be detected in the presence of measurement noise and also in the presence of other parameters that may signal alarm condition as a result of the original fault. Countermeasures can be initiated quickly if a fault has been detected early and the occurrence of plant shutdowns can be minimized.

REFERENCES

[1] C. Favre, "Fly-by-wire for commercial aircraft - the airbus experience," *International Journal of Control*, vol. 59, no. 1, pp. 139–157, 1994.

[2] M. Basseville, "Detecting changes in signals and systems - a survey," *Automatica*, vol. 24, no. 3, pp. 309–326, 1988.

[3] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 603–614, 1984.

[4] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results," *Automatica*, vol. 26, no. 2, pp. 459–474, 1990.

[5] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri, "A review of process fault detection and diagnosis - part I: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, pp. 293–311, 2003.

[6] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri, "A review of process fault detection and diagnosis - part II: Qualitative models and search strategies," *Computers and Chemical Engineering*, vol. 27, pp. 313–326, 2003.

[7] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri, "A review of process fault detection and diagnosis - part III: Process history based methods," *Computers and Chemical Engineering*, vol. 27, pp. 327–346, 2003.

[8] R. N. Clark, "The dedicated observer approach to instrument fault detection," in *Proceedings of the 15th IEEE-CDC*, Fort Lauderdale, FL, Dec. 1979, pp. 237–241.

[9] M. A. Massoumnia, "A geometric approach to the synthesis of failure detection filters," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 839–846, 1986.

[10] P.M. Frank and J. Wunnenberg, *Robust Fault Diagnosis using Unknown Input Observer Schemes*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[11] J. Chen and R. J. Patton, *Robust Model Based Fault Diagnosis for Dynamic Systems*, Boston, MA: Kluwer Academic Publishers, 1999.

[12] M.1 Yoshimura, P.M. Frank, and X. Ding, "Survey of robust residual generation and evaluation methods in observer-based fault detection systems," *Journal of Process Control*, vol. 10, pp. 356–379, 1974.

[13] E. A. Garcia and P. M. Frank, "Deterministic nonlinear observer-based approaches to fault diagnosis: A survey," *Control Engineering Practice*, vol. 5, no. 5, pp. 663–670, 1997.

[14] Y. Dingli, J. B. Gomm, D. N. Shields, D. Williams, and K. Disdell, "Fault diagnosis for a gas-fired furnace using bilinear observer method," in *Proceedings of the American Control Conference*, Seattle, WA, Jun. 1995, pp. 1127–1131.

[15] H. Yang and M. Saif, "Nonlinear adaptive observer design for fault detection," in *Proceedings of the American Control Conference*, Seattle, WA, Jun. 1995, pp. 1136–1139.

[16] S. Rajaraman, J. Hahn, and M.S. Mannan, "A methodology for fault detection, isolation and identification for nonlinear processes with parametric uncertainties," *Industrial & Engineering Chemistry Research*, vol. 43, no. 21, pp. 6774–6786, 2004.

[17] E. Russell, L. H. Chiang, and R. D. Braatz, *Data Driven Methods for Fault Detection and Diagnosis in Chemical Processes*, New York: Springer, 2000.

[18] S. Rajaraman, J. Hahn, and M. S. Mannan, "A methodology for fault detection, isolation, and identification for nonlinear processes with parametric uncertainties," *Industrial & Engineering Chemistry Research*, vol. 43, no. 21, pp. 6774 –6786, 2004.

[19] A.S. Willsky and H.L. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.

[20] J. Gertler, X. Fang, and Q. Luo, "Detection and diagnosis of plant failure: The orthogonal parity equation approach," *Control and Dynamic Systems*, vol. 37, pp. 159–216, 1990.

[21] J. Gertler, M. Costin, X. Fang, Z. Kowalczuk, M. Kunwer, and R. Monajemy, "Model based diagnosis for automative engines - algorithm development and testing on a productin vehicle," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 61–69, 1995.

[22] J. Gertler and R. Monajemy, "Generating directional residuals with dynamic parity relations," *Automatica*, vol. 31, no. 4, pp. 627–635, 1995.

[23] V. Vaclavek, "Gross systematic errors or biases in the balance calculations," in *Papers of the Prague Institute of Technology*, Prague, Czech, 1984.

[24] G. A. Almasy and T. Sztano, "Checking and correction of measurements on the basis of linear system models," *Problems of Control and Information Theory*, vol. 4, pp. 57–69, 1975.

[25] M. Desai and A. Ray, "A fault detection and isolation methodology theory and application," in *Proceedings of the American Control Conference*, San Diego, CA, 1984, pp. 262–270.

[26] J. Gertler, "Analytical redundancy methods in fault detection and isolation," in *Proceedings of the IFAC Symposium SAFEPROCESS*, Baden-Baden, Germany, 1991, pp. 9–21.

[27] M. Basseville and A. Benveniste, "Detection of abrupt changes in signals and dynamic systems: Some statistical aspects," *Lecture Notes in Control and Information Sciences*, vol. 62, pp. 143–155, 1984.

[28] F. Daum, "Nonlinear filters: Beyond the Kalman filter," *IEEE A&E Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[29] M. Soroush, "State and parameter estimations and their applications in process control," *Computers and Chemical Engineering*, vol. 23, no. 2, pp. 229–245, 1998.

[30] H. W. Sorenson Ed., *Kalman Filtering: Theory and Application*, New York: IEEE Press, 1985.

[31] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.

[32] R. M. Oisiovici and S. L. Cruz, "State estimation of batch distillation columns using an extended Kalman filter," *Chemical Engineering Science*, vol. 55, no. 20, pp. 4667–4680, 2000.

[33] E. Boje and M. Petrick, "Application of the extended Kalman filter to a lysine hydrochlorination process," *Control Engineering Practice*, vol. 8, no. 3, pp.

291–297, 2000.

[34] V. M. Becerra, P. D. Roberts, and G. W. Griffiths, "Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations," *Control Engineering Practice*, vol. 9, no. 3, pp. 267–281, 2001.

[35] R. Li, A. B. Corripioa, M. A. Henson, and M. J. Kurtzc, "On-line state and parameter estimation of epdm polymerization reactors using a hierarchical extended Kalman filter," *Journal of Process Control*, vol. 14, no. 8, pp. 837–852, 2004.

[36] Y. Huang, S. Dash, G.V. Reklaitis, and V. Venkat, "EKF based estimator for FDI in model IV FCCU," in *Proceedings of the IFAC Symposium SAFEPROCESS*, Budapest, Hungary, Jun. 2000.

[37] M. Mosallaei, K. Salahshoor, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended Kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, 2008.

[38] B. Cipra, "Engineers look to Kalman filtering for guidance," *SIAM News*, vol. 26, no. 5, pp. 8, 1993.

[39] A. Romanenko, L. O. Santos, and P. A. F. N. A. Afonso, "Unscented Kalman filtering of a simulated pH system," *Industrial & Engineering Chemistry Research*, vol. 43, no. 23, pp. 7531–7538, 2004.

[40] A. Romanenko and J. A.A.M. Castro, "The unscented filter as an alternative to the EKF for nonlinear state estimation: A simulation case study," *Computers and Chemical Engineering*, vol. 28, no. 3, pp. 347–355, 2004.

[41] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[42] M. S. Grewal and A. P. Andrews., *Kalman Filtering : Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[43] K. Mostov, “Fuzzy adaptive stabilization of higher order Kalman filters,” M.S. thesis, University of California, Berkley, CA, 1996.

[44] C. T. Chang and J. I. Hwang, “Simplification techniques for ekf computations in fault diagnosis - suboptimal gains,” *Chemical Engineering Science*, vol. 53, no. 22, pp. 3853–3862, 1998.

[45] C. T. Chang and J. I. Hwang, “Simplification techniques for ekf computations in fault diagnosis - model decomposition,” *AIChE Journal*, vol. 44, no. 6, pp. 1392–1403, 1998.

[46] T. S. Schei, “A finite difference method for linearization in nonlinear estimation algorithms,” *Automatica*, vol. 33, no. 11, pp. 2051–2058, 1997.

[47] B. M. Quine, J. K. Uhlmann, and H. F. Durrant-Whyte, “Implicit jacobians for linearized state estimation in nonlinear systems,” in *Proceedings of the American Control Conference*, Seattle, WA, Jun. 1995, vol. 3, pp. 1645–1646.

[48] B. M. Quine, “A derivative-free implementation of the extended Kalman filter,” *Automatica*, vol. 42, pp. 1927–1934, 2006.

[49] M. Norgaard, N. K. Poulsen, and O. Ravn, “New developements in state estimation for nonlinear systems,” *Automatica*, vol. 36, no. 11, pp. 1627–1638, 2000.

[50] S. Ungarala, E. Dolence, and K. Li, "Constrained extended Kalman filter for nonlinear state estimation," in *8th International IFAC Symposium on Dynamics and Control of Process Systems*, Cancun, Mexico, 2007, pp. 63–68.

[51] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference*, Seattle, WA, 1995, pp. 1628–1632.

[52] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium Aerospace/Defense Sensing, Simulation and Controls*, Orlando, FL, 1997, pp. 182–193.

[53] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[54] S. J. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probablity distributions," *Online at http://www.robts.ox.ac.uk/∼siju*, 1996.

[55] S. J. Julier, "The scaled unscented transformation," in *Proceedings of the American Control Conference*, Anchorage, AK, 2002, pp. 4555–4559.

[56] P. S. Maybeck Ed., *Stochastic Models, Estimation, and Control* I, New York: Academic Press, 1979.

[57] M. Athans, R. P. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 504–518, 1968.

[58] K. Xiong, H. Y. Zhang, and C. W. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, 2006.

[59] L. A. Aguirre, B. O. S. Teixeira, and L. A. B. Torres, "Using data-driven discrete-time models and the unscented Kalman filter to estimate unobseved variables of nonlinear systems," *Physical Review*, vol. 72, pp. 026226, Aug.2005.

[60] J. J. LaViola Jr., "A comparison of unscented and extended Kalman filtering for estimating quaterion motion," in *Proceedings of the American Control Conference*, Denver, CO, Jun. 2003, pp. 2435–2440.

[61] C. Qu and J. Hahn, "Process monitoring and parameter estimation via unscented Kalman filtering," In Press, 2009.

[62] P. Vachhani, S. Narasimhan, and R Rengaswamy, "Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation," *Journal of Process Control*, vol. 16, pp. 1075–1086, 2006.

[63] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *The IEEE of Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[64] E. A. Wan, R. Van Der Merwe, and A. T. Nelson, "Dual estimation and the unscented transformation," *MIT Press Advances in Neural Information Processing Systems*, vol. 12, pp. 666–672, 2000.

[65] E. A. Wan and A. T. Nelson, "Neural dual extended Kalman filtering: Applications in speech enhancement and monaural blind signal separation," in *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, Amelia Island, FL, Sep. 1997, pp. 466–475.

[66] R. Van Der Merwe, J. F. G. de Freitas, A. Doucet, and E. A. Wan, "The unscented particle filter," Technical report, Department of Engineering, University of Cambridge, UK, 2000.

[67] E. A. Wan and R. Van Der Merwe, "The unscented bayes filter," Technical report, The Center for Spoken Language Understanding, Oregon Health & Science University, Portland, OR, 2000.

[68] R. van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter estimation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, 2001, pp. 3461–3464.

[69] R. van der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D dissertation, Oregon Health and Science University, Portland, OR, 2004.

[70] M-A Beyer, W. Grote, and G. Reinig, "Adaptive exact linearization control of batch polymerization reactors using a sigma-point Kalman filter," *Journal of Process Control*, vol. 18, no. 7, pp. 663–675, 2007.

[71] R. Kandepu, L. Imsland, and B. A. Foss, "Constrained state estimation using the unscented Kalman filter," in *Proceedings of 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, Jun. 2008, pp. 1453–1458.

[72] H. J. Kushner, "Dynamical equations for optimal nonlinear filtering," *Journal of Differential Equations*, vol. 3, pp. 179–190, 1967.

[73] P. S. Maybeck Ed., *Stochastic Models, Estimation, and Control* II, New York: Academic Press, 1982.

[74] H. J. Kushner, “Approximations to optimal nonlinear filters,” *IEEE Transactions on Automatic Control*, vol. 12, pp. 546–556, 1967.

[75] H. W. Sorenson and A. R. Stubberud, “Nonlinear filtering by approximation of the *a posteriori* density,” *International Journal of Control*, vol. 8, no. 1, pp. 33–51, 1968.

[76] F. E. Daum, “New exact nonlinear filters,” in *Bayesian Analysis of Time Series and Dynamic Models*, J. C. Spall Ed., New York: Marcel Dekker, 1988, pp. 199–226.

[77] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *IEEE Proceedings of Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[78] M. Marseguerra and E. Zio, “Monte carlo simulation for model-based fault diagnosis in dynamic systems,” *Reliablity Engineering & System Safety*, vol. 94, no. 2, pp. 180–186, 2009.

[79] P. Li and V. Kadirkamanathan, “Particle filtering based likelihood ratio approach to fault diagnosis in nonlinear stochastic systems,” *IEEE Transactions on Systems Man and Cybernetics Part C - Applications and Reviews*, vol. 31, no. 3, pp. 337–343, 2001.

[80] T. Chen, J. Morris, and E. Martin, “Particle filters for dynamic data rectification and process change detection,” in *IFAC Fault Detection, Supervision and Safety of Technical Processes*, Beijing, China, 2006, pp. 204–209.

[81] T. Chen, J. Morris, and E. Martin, “Dynamic data rectification using particle filters,” *Computers & Chemical Engineering*, vol. 32, pp. 451–462, 2008.

[82] T. Chen, J. Morris, and E. Martin, "Particle filters for state and parameter estimation in batch processes," *Journal of Process Control*, vol. 15, pp. 665–673, 2005.

[83] G. Oppenheim, A. Philippe, and J. de Rigal, "The particle filters and their applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 91, pp. 87–93, 2008.

[84] W. Chen, B. R. Bakshi, P. K. Goel, and S. Ungarala, "Bayesian estimation via sequential Monte Carlo sampling: Unconstrained nonlinear dynamic systems," *Industrial & Engineering Chemistry Research*, vol. 43, no. 14, pp. 4012–4025, 2004.

[85] L. Lang, W Chen, B. R. Bakshi, P. K. Goel, and S. Ungarala, "Bayesian estimation via sequential Monte Carlo sampling: Constrained dynamic systems," *Automatica*, vol. 43, no. 9, pp. 1615–1622, 2007.

[86] G. Evenson, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[87] S. Gillijns, O. B. Mendoza, J. Chandrasekar, B. L. R. de Moor, D. S. Bernstein, and D. S. Ridley, "What is the ensemble Kalman filter and how well does it work," in *Proceedings of the American Control Conference*, Minneapolis, MN, 2006, pp. 4448–4453.

[88] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, "Dual state parameter estimation of hydrological models using ensember Kalman filter," *Advances in Water Resources*, vol. 28, no. 2, pp. 135–147, 2005.

[89] X. Luo and IM. Moroz, "Ensemble Kalman filter with the unscented transform," *Physica D - Nonlinear Phenomena*, vol. 238, no. 5, pp. 549–562, 2009.

[90] S. Ungarala, K. Li, and Z.Z. Chen, "Constrained bayesian state estimation using a cell filter," *Industrial & Engineering Chemistry Research*, vol. 47, no. 19, pp. 7312–7322, 2008.

[91] S. Ungarala and K. Li, "The use of a cell filter for state estimation in closed-loop NMPC of low dimensinal systems," *Journal of Process Control*, vol. 19, no. 3, pp. 550–556, 2009.

[92] D. G. Robertson, J. H. Lee, and J. B. Rawlings, "A moving horizon-based approach for least-squares estimation," *AIChE Journal*, vol. 42, no. 8, pp. 2209–2224, 1996.

[93] E. L. Haseltine and J. B. Rawlings, "Critical evaluation of extended Kalman filtering and moving horizon estimation," *Industrial & Engineering Chemistry Research*, vol. 44, no. 8, pp. 2451–2460, 2005.

[94] J. B. Rawlings and B R. Bakshi., "Particle filtering and moving horizon estimation," *Computers and Chemical Engineering*, vol. 30, no. 10-12, pp. 1529–1541, 2006.

[95] E. L. Haseltine and J. B. Rawlings, "A critical evaluation of extended Kalman filtering and moving horizon estimation," Technical report, TWMCC, Department of Chemical Engineering, University of Wisconsin-Madison, Madison, WI, 2002.

[96] C.V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimation: A moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, 2001.

[97] C.V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–258, 2003.

[98] C. V. Rao and J. B. Rawlings, "Constrained process monitoring: Moving horizon approach," *AIChE Journal*, vol. 48, no. 1, pp. 97–109, 2002.

[99] L. P. Russo and R. E. Young, "Moving horizon state estimation applied to an industrial polymerization process," in *Proceedings of the American Control Conference*, San Diego, CA, 1999, pp. 1129–1133.

[100] A. Bemporad, D. Mignone, and M. Morari, "Moving horizon estimation for hybrid systems and fault detection," in *Proceedings of the American Control Conference*, San Diego, CA, 1999, pp. 2471–2475.

[101] W. Mautz, M. Diehl, and S. Engell, "Moving horizon estimation and optimal excitation in temperature oscillation calorimetry," in *Proceedings of 8th International IFAC Symposium on Dynamics and Control of Process Systems*, Cancun, Mexico, 2007, pp. 57–62.

[102] M. L. Darby and M. Nikolaou, "A parametric programming approach to moving horizon state estimation," *Automatica*, vol. 43, no. 5, pp. 885–891, 2007.

[103] M.J. Tenny and J. B. Rawlings, "Efficient moving horizon estimation and nonlinear model predictive control," in *Proceedings of the American Control Conference*, Anchorage, AK, 2002, pp. 4475–4480.

[104] M. Liebman, T. Edgar, and L. Lasdon, "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," *Computers and Chemical Engineering*, vol. 16, no. 10-11, pp. 963–986, 1992.

[105] G. A. Almasy, “Principles of dynamic balancing,” *AIChE Journal*, vol. 36, no. 9, pp. 1321–1330, 1990.

[106] M. J. Liebman, “Reconciliation of process measurements using statistical and nonlinear programming techniques,” Ph.D dissertation, The University of Texas at Austin, 1991.

[107] Y. Ramamurthi, P. B. Sistu, and B. W. Bequette, “Control relevant dynamic data reconcliation and parameter estimation,” *Computers and Chemical Engineering*, vol. 17, no. 1, pp. 41–59, 1993.

[108] J. Chen and J. A. Romagnoli, “A strategy for simultaneous dynamic data reconciliation and outlier detection,” *Computers and Chemical Engineering*, vol. 22, no. 4-5, pp. 559–562, 1998.

[109] J. S. Albuquerque and L. T. Biegler, “Data reconciliatin and gross error detection for dynamic systems,” *AIChE Journal*, vol. 42, no. 10, pp. 2841–2856, 1996.

[110] K. F. McBrayer and T. F. Edgar, “Bias detection and estimation in dynamic data reconciliation,” *Journal of Process Control*, vol. 5, no. 4, pp. 285–289, 1995.

[111] K. F. McBrayer, “Detection and identification of bias in nonlinear dynamic processes,” Ph.D dissertation, The University of Texas at Austin, Austin, TX, 1996.

[112] K. F. McBrayer, T. A. Soderstrom, T. F. Edgar, and R. E. Young, “The application of nonlinear dynamic data reconciliation to plant data,” *Computers and Chemical Engineering*, vol. 22, no. 12, pp. 1907–1911, 1998.

[113] T. A. Soderstrom, T. F. Edgar, L. P. Russo, and R. E. Young, "Industrial application of a large-scale dynamic data reconciliation strategy," *Industrial & Engineering Chemistry Research*, vol. 39, no. 6, pp. 1683–1693, 2000.

[114] T. A. Soderstrom, D. M. Himmelblau, and T. F. Edgar, "A mixed integer optimization approach for simultaneous data reconciliation and identification of measurement bias," *Control Engineering Practice*, vol. 9, no. 8, pp. 869–876, 2001.

[115] I. Yelamos, C. Mendez, and L. Puigjaner, "Enhancing dynamic data reconciliation performance through time delays identification," *Chemical Engineering and Processing*, vol. 46, no. 12, pp. 1251–1263, 2007.

[116] S. Kameswaran and L. T. Biegler, "Simultaneous dynamic optimization strategies: Recent advances and challenges," *Computer and Chemical Engineering*, vol. 30, no. 10-12, pp. 1560–1575, 2006.

[117] S. H. Rich and V. Venkatasubramanian, "Model based reasoning in diagnostic expert systems for chemical process plants," *Computers and Chemical Engineering*, vol. 11, no. 2, pp. 111–122, 1987.

[118] V. Venkatasubramanian and S. H. Rich, "An object-oriented two-tier architecture for integrating compiled and deep-level knowledge for process diagnosis," *Computers and Chemical Engineering*, vol. 12, no. 9-10, pp. 903–921, 1988.

[119] M. Iri, K. Aoki, E. O'Shima, and H. Matsuyamu, "An algorithm for diagnosis of system failures in the chemical process," *Computers and Chemical Engineering*, vol. 3, no. 1-4, pp. 489–493, 1979.

[120] O. O. Oyeleye and M. A. Kramer, "Qualitative simulation of chemical process

systems: Steady state analysis," *AIChE Journal*, vol. 34, no. 9, pp. 1441–1454, 1988.

[121] C. C. Chang and C. C. Yu, "On-line fault diagnosis using the signed directed graph," *Industrial & Engineering Chemistry Research*, vol. 29, no. 7, pp. 1290–1299, 1990.

[122] C. Han, R. Shih, and L. Lee, "Quantifying signed directed graphs with the fuzzy set for fault diagnosis resolution improvement," *Industrial & Engineering Chemistry Research*, vol. 33, no. 8, pp. 1943–1954, 1994.

[123] R. Shih and L. Lee, "Use of fuzzy cause-effect digraph for resolution fault diagnosis for process plant I. fuzzy cause-effect digraph," *Industrial & Engineering Chemistry Research*, vol. 34, no. 5, pp. 1688–1702, 1995.

[124] R. Shih and L. Lee, "Use of fuzzy cause-effect digraph for resolution fault diagnosis for process plant II: Diagnostic algorithm and applications," *Industrial & Engineering Chemistry Research*, vol. 34, no. 5, pp. 1703–1717, 1995.

[125] S. A. Lapp and G. A. Powers, "Computer-aided synthesis of fault trees," *IEEE Transactions on Reliability*, vol. 26, no. 1, pp. 2–13, 1977.

[126] N. H. Ulerich and G. A. Powers, "Online hazard aversion and fault diagnosis in chemical processes: The diagraph + fault tree method," *IEEE Transactions on Reliability*, vol. 37, no. 2, pp. 171–177, 1988.

[127] H. A. Simon, *Models of Discovery*, Boston, MA: Reidel Publishing Company, 1990.

[128] Y. Iwasaki and H. A. Simon, "Causality in device behavior," *Artificial Intelligence*, vol. 29, no. 1, pp. 3–32, 1986.

[129] J. de Kleer and S. Brown, "A qualitative physics based on confluences," *Artificial Intelligence*, vol. 24, no. 1-3, pp. 7–83, 1984.

[130] S. D. Grantham and L. H. Ungar, "A first principles approach to automated troubleshooting of chemical plants," *Computers and Chemical Engineering*, vol. 14, no. 7, pp. 783–798, 1990.

[131] S. D. Grantham and L. H. Ungar, "Comparative analysis of qualitative models when the model changes," *AIChE Journal*, vol. 37, no. 6, pp. 931–943, 1991.

[132] S. H. Rich, V. Venkatasubramanian, M. Masrallah, and C. Matteo, "Development of a diagnostic expert system for a whipped toppings process," *Journal of Loss Prevention in the Process Industries*, vol. 2, no. 3, pp. 145–154, 1989.

[133] J. T. Cheung and G. Stephanopoulos, "Representaton of process trends part I. A formal representation framework," *Computers and Chemical Engineering*, vol. 14, pp. 495–510, 1990.

[134] R. Rengaswamy and V. Venkatasubramanian, "A syntactic pattern-recognition approach for process monitoring and fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 1, pp. 35–51, 1995.

[135] A. B. Bulsari Ed., *Neural Networks for Chemical Engineers*, Amsterdam: Elsevier Science, 1995.

[136] V. Venkatasubramanian and T. J. McAvoy, "Special issue on neural network applicatons in chemical engineering," *Computers and Chemical Engineering*, vol. 16, no. 4, 1992.

[137] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine Series B*, vol. 2, no. 6, pp. 559–572, 1901.

[138] H. Hotelling, “Multivariate quality control illustrated by the testing of sample bombsights,” in *Selected Techniques of Statistical Analysis*, O. Eisenhart Ed., New York: McGraw-Hill, 1947, pp. 111–184.

[139] S. Wold, C. Albano, W. J. Dunn, U. Edland, K. Esbensen, P. Geladi, S. Hellberg, E. Johansson, W. Lindberg, and M. Sjostrom, “Multivariate data analysis in chemistry,” in *Chemometrics, Mathematics and Statistics in Chemistry*, B. Kowalski Ed., New York: Springer, 1984, pp. 4–96.

[140] S. Wold, A. Ruhe, H. Wold, and W. Dunn, “The collinearity problem in linear regression. The partial least squares approach to generalized inverses,” *SIAM Journal of Science Statistical Computer*, vol. 5, no. 3, pp. 735–743, 1984.

[141] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Sysems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[142] E.J. Henley, “Application of expert systems to fault diagnosis,” in *Proceedings of AIChE Annual Meeting*, San Francisco, CA, 1984.

[143] D. Chester, D. Lamb, and P. Dhurjati, “Rule-based computer alarm analysis in chemical process plants,” in *Proceedings of 7th Micro-Delcon*, Los Alamitos, CA, 1984, pp. 22–29.

[144] K. Niida, “Expert system experiments in processing engineering,” in *Institution of Chemical Engineering Symposium Series*, 1985, pp. 529–583.

[145] J. S. Qin, “Statistical process monitoring: Basics and beyond,” *Journal of Chemometrics*, vol. 17, no. 8-9, pp. 480–502, 2003.

[146] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *ASME Journal of Basic Engineering*, vol. 82, no. 3, pp. 35–45, 1960.

[147] R. E. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *ASME Journal of Basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.

[148] G. T. Schmidt, *Practical Aspects of Kalman Filtering Implementation*, NATO Advisory Group for Aerospace Research and Development, AGARD-LS-82, London, 1974.

[149] H. G. Bock, M. Diehl, D. B. Leineweber, and J. P. Schloser, "A direct multiple shooting method for real-time optimization of nonlinear DAE processes," in *Proceedings of International Symposium on Nonlinear Model Predictive Control*, Ascona, Switzerland, 1998.

[150] D. Simon, *Optimal State Estimation - Kalman, H_∞, and Nonlinear Approaches*, Hoboken, NJ: Wiley-Interscience, 2006.

[151] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global Sensitivity Analysis: The Primer*, New York: Wiley-Interscience, 2008.

[152] J.C. Helton, J.D. Johnson, C.J. Salaberry, and C.B. Storlie, "Survey of sampling based methods for uncertainty and sensitivity analysis," *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1175–1209, 2006.

[153] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.

[154] F. Campolongo, J. Cariboni, and A. Saltelli, "An effective screening design for sensitivity analysis of large models," *Environmental Modelling and Software*, vol. 22, no. 10, pp. 1509–1518, 2007.

[155] G. Hornberger and R. Spear, "An approach to the preliminary analysis of environmental systems," *Journal of Environmental Management*, vol. 7, no. 12, pp. 7–18, 1981.

[156] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, New York: John Wiley and Sons, 2004.

[157] J. Oakley and A. O. Hagan, "Probabilistic sensitivity analysis of complex models: A bayesian approach," *Journal of the Royal Statistical Society Series B*, vol. 66, pp. 751–769, 2004.

[158] I. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Mathematical Modeling & Computational Experiment*, vol. 2, no. 1, pp. 112–118, 1990.

[159] T. Homma and A. Saltelli, "Importance measures in global sensitivity analysis of nonlinear models," *Reliability Engineering and System Safety*, vol. 52, no. 1, pp. 1–17, 1996.

[160] A. Saltelli, K. Chan, and M. Scott Eds., *Sensitivity Analysis*, New York: John Wiley and Sons, 2000.

[161] A. Saltelli and S. Tarantola, "On the relative importance of input factors in mathematical models: Safety assessment for nuclear waste disposal," *Journal of American Statistical Association*, vol. 97, no. 9, pp. 702–709, 2002.

[162] H. Rabitz, "System analysis at molecular scale," *Science*, vol. 246, no. 4927, pp. 221–226, 1989.

[163] G. Li, W. S. Wang, and H. Rabitz, "Practical approaches to construct RS-HDMR component functions," *Journal of Physical Chemistry*, vol. 106, no. 37, pp. 8721–8733, 2002.

[164] G. Li, J. Hu, S. W. Wang, P. Georgopoulos, J. Schoendorf, and H. Rabitz, "Random sampling-high dimensional model representation and orthogonality of its different order component functions," *Journal of Physical Chemistry*, vol. 110, no. 7, pp. 2474–2485, 2006.

[165] U. N. Lerner, "Hybrid bayesian networks for reasoning about complex systems," Ph.D dissertation, Stanford University, Stanford, CA, 2002.

[166] D. Tenne and T. Singh, "The higher order unscented filter," in *Proceedings of the American Control Conference*, Denver, CO, 2003, pp. 2441–2446.

[167] S. A. Vejtasa and R. A. Schmitz, "An experimental study of steady-state multiplicity and stability in an adiabatic stirred reactor," *AIChE Journal*, vol. 16, no. 3, pp. 410–419, 1970.

[168] A. J. Stack and F. J. Doyle, "Application of a control-law nonlinearity measure to the chemical reactor analysis," *AIChE Journal*, vol. 43, no. 2, pp. 425–439, 1997.

[169] J. Hahn and T. F. Edgar, "A gramian based approach to nonlinearity quantification and model classification," *Industrial & Engineering Chemistry Research*, vol. 40, no. 24, pp. 5724–5731, 2001.

[170] E. Gatzke and F. J. Doyle, "Use of multiple models and qualitative knowledge for on-line moving horizon disturbance estimation and fault diagnosis," *Journal of Process Control*, vol. 12, no. 2, pp. 339–352, 1994.

[171] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented Kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7-8, pp. 753–768, 2008.

[172] C. F. Van Loan, "Computing integrals involving the matrix exponential," *IEEE Transactions on Automatic Control*, vol. 23, no. 3, pp. 395–404, 1978.

[173] D. M. Bates and D. G. Watts, "Relative curvature measures of nonlinearity," *Journal of the Royal Statistical Society Series B*, vol. 42, no. 1, pp. 1–25, 1980.

[174] M. Guay, P. G. McLellan, and D. W. Bacon, "Measurement of nonlinearity in chemical process control systems: The steady state map," *Canadian Journal of Chemical Engineering*, vol. 73, no. 6, pp. 868–882, 1995.

[175] A. Helbig, W. Marquardt, and F. Allgower, "Nonlinearity measures: Definition, computation and applications," *Journal of Process Control*, vol. 10, no. 2-3, pp. 113–123, 2000.

[176] D. Sun and K. A. Hoo, "Nonlinearity measures for a class of SISO nonlinear systems," *International Journal of Control*, vol. 73, no. 1, pp. 29–37, 2000.

APPENDIX A

PROOF THAT ALGORITHM 2 IS EQUIVALENT TO ALGORITHM 1.1.

It is adequate to show that P_{k+1}^{-} in both algorithms is identical for proving that Algorithm 2 is equivalent to Algorithm 1.1.

From Algorithm 2 in Table XII, it can be seen that,

$$\begin{aligned} P_{k+1}^{-} &= A_k P_k^{+} A_k^{'} + G Q_k G^{'} \\ &= e^{A(\hat{x}_k)T} P_k^{+} e^{A(\hat{x}_k)^{'}T} + \int_0^T e^{A(\hat{x}_k)\tau} G Q G^{'} e^{A(\hat{x}_k)^{'}\tau} d\tau, \text{where } A(\hat{x}_k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k}. \end{aligned} \tag{A.1}$$

Also Algorithm 1.1 states that

$$\dot{P} = A(\hat{x}_k)P + PA(\hat{x}_k)^{'} + GQG^{'}, \text{where } A(\hat{x}_k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}_k} \tag{A.2}$$

with $P(0) = P_k^{+}$ and $P(T) = P_{k+1}^{-}$.

Noting that $A(\hat{x}_k)$ in Eqn. (A.1) and Eqn. (A.2) is identical, it can be replaced by A as a change of notation.

If it can be shown that

$$P(t) = e^{At} P_k^{+} e^{A^{'}t} + \int_0^t e^{A\tau} G Q G^{'} e^{A^{'}\tau} d\tau \tag{A.3}$$

is the solution of

$$\dot{P} = AP + PA^{'} + GQG^{'}, \text{with } P(0) = P_k^{+}, \tag{A.4}$$

then it can be concluded that Algorithm 2 is identical to Algorithm 1.1.

It can be derived from Eqn. (A.3) that for $t = 0$,

$$P(0) = e^0 P_k^+ e^0 + \int_0^0 e^{A\tau} GQG' e^{A'\tau} d\tau = P_k^+. \tag{A.5}$$

Taking the derivative of Eqn. (A.3), it is derived that

$$\dot{P} = Ae^{At} P_k^+ e^{A't} + e^{At} P_k^+ e^{A't} A' + e^{At} GQG' e^{A't}. \tag{A.6}$$

On the other hand, substituting Eqn. (A.3) into the right side of Eqn. (A.4), the following equations

$$\begin{aligned} AP + PA' + GQG' = {} & Ae^{At} P_k^+ e^{A't} + \int_0^t Ae^{A\tau} GQG' e^{A'\tau} d\tau \\ & + e^{At} P_k^+ e^{A't} A' + \int_0^t e^{A\tau} GQG' e^{A'\tau} A' d\tau + GQG' \end{aligned} \tag{A.7}$$

are obtained.

Examining the second and the fourth terms in the above equation, it is found that

$$\begin{aligned} & \int_0^t Ae^{A\tau} GQG' e^{A'\tau} d\tau + \int_0^t e^{A\tau} GQG' e^{A'\tau} A' d\tau \\ & = \int_0^t d(e^{A\tau}) GQG' e^{A'\tau} + \int_0^t e^{A\tau} GQG' d(e^{A'\tau}) \\ & = \int_0^t d(e^{A\tau} GQG' e^{A'\tau}) \\ & = e^{A\tau} GQG' e^{A'\tau} \Big|_0^t \\ & = e^{At} GQG' e^{A't} - GQG' \end{aligned} \tag{A.8}$$

Therefore Eqn. (A.7) becomes

$$AP + PA' + GQG' = Ae^{At} P_k^+ e^{A't} + e^{At} P_k^+ e^{A't} A' + e^{At} GQG' e^{A't}. \tag{A.9}$$

Comparing Eqn. (A.6) to Eqn. (A.9), it can be shown that

$$\dot{P} = AP + PA^{'} + GQG^{'}. \tag{A.10}$$

Taking Eqn.(A.5) into account, it can be concluded that Eqn. (A.3) is the solution of Eqn. (A.4). Therefore Algorithm 2 is equivalent to Algorithm 1.1.

APPENDIX B

DERIVATION OF DISCRETE-TIME KALMAN FILTER

Consider the linear, dynamic, discrete-time, stochastic system:

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1} \tag{B.1}$$

$$z_k = H_k x_k + v_k \tag{B.2}$$

where it is assumed that,

- Initial statistical properties of *a prior* distribution are known, i.e., $x_0 = E[x(0)]$ and $P_0 = E[(x(0) - x_0)(x(0) - x_0)^T)]$ and
- Statistical models of the system and measurement noises are known. $w_k \sim N(0, Q_k)$ and $v_k \sim N(0, R_k)$ are zero mean, Gaussian white noise processes and
- w_k, v_k, and $x(0)$ are mutually uncorrelated $E[w_k v_k^T] = E[x(0) w_k^T] = E[x(0) v_k^T] = 0$

The objective is to find optimal estimates $\hat{x}$ for the given noisy system, i.e., to minimize the mean squared error (variance). In this case, the cost function is chosen as the trace (i.e., the sum of the diagonal elements) of the error covariance matrix

$$P_k = E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T], \tag{B.3}$$

which is equivalent to minimize the mean squared error for an unbiased estimator. The choice of the cost function offers mathematical convenience for derivation of a solution.

Define the estimation error

$$\tilde{x}_k^+ = \hat{x}_k^+ - x_k \tag{B.4}$$

$$\tilde{x}_k = \hat{x}_k^- - x_k \tag{B.5}$$

where x_k represents the true value of the system states, $\hat{x}_k^-$ is a prior state estimate and $\hat{x}_k^+$ is a posterior state estimate at time t_k.

Given *a prior* estimate $\hat{x}_k^-$, an update estimate $\hat{x}_k^+$ is sought, based on the measurement z_k. In order to avoid a growing memory lter, the estimate is sought in the linear recursive form:

$$\hat{x}_k^+ = M_k\hat{x}_k^- + K_k z_k \tag{B.6}$$

where M_k and K_k are time-varying weighting matrices, as yet un- specified. Clearly, there is no need to store past measurements for the purpose of computing present estimates.

From the recursive Eq. (B.6), an equation for the estimation error can be obtained

$$\tilde{x}_k^+ = M_k\hat{x}_k^- + K_k z_k - x_k \tag{B.7}$$

Substitute the measurement z_k in Eq. (B.7),

$$\tilde{x}_k^+ = M_k\hat{x}_k^- + K_k(H_k x_k + v_k) - x_k \tag{B.8}$$

$$= M_k\hat{x}_k^- + (K_k H_k - I)x_k + K_k v_k \tag{B.9}$$

$$= M_k\tilde{x}_k^- + (M_k + K_k H_k - I)x_k + K_k v_k \tag{B.10}$$

By assumption $E[v_k] = 0$ and $E[\tilde{x}_k^-] = 0$. To seek an unbiased estimator, $E[\tilde{x}_k^+] = 0$ only if

$$M_k = I - K_k H_k. \tag{B.11}$$

Therefore, the estimator takes the form

$$\hat{x}_k^+ = (I - K_k H_k)\hat{x}_k^- + K_k z_k \tag{B.12}$$

or alternatively,

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \tag{B.13}$$

which is the update equation for $\hat{x}_k^+$ based on the measurements. K_k is the matrix for weighting measurement residual, as to be specified.

ERROR COVARIANCE UPDATE

From the posterior state update equation, the estimation error equation

$$\tilde{x}_k^+ = (I - K_k H_k)\tilde{x}_k^- + K_k v_k \tag{B.14}$$

Substituting $\tilde{x}_k^+$ into the error covariance matrix

$$P_k^+ = E[(\tilde{x}_k^+)(\tilde{x}_k^+)^T] \tag{B.15}$$

gives

$$P_k^+ = E[\{(I - K_k H_k)\tilde{x}_k^- + K_k v_k\}\{(I - K_k H_k)\tilde{x}_k^- + K_k v_k\}^T] \tag{B.16}$$

$$= E[(I - K_k H_k)\tilde{x}_k^- \tilde{x}_k^{-T}(I - K_k H_k)^T + (I - K_k H_k)\tilde{x}_k^- v_k^T K_k^T \tag{B.17}$$

$$+ K_k v_k \tilde{x}_k^{-T}(I - K_k H_k)^T) + K_k v_k v_k^T K_k^T] \tag{B.18}$$

By definition,

$$E[(\tilde{x}_k^-)(\tilde{x}_k^-)^T] = P_k^- \tag{B.19}$$

$$E[v_k v_k^T] = R_k \tag{B.20}$$

and, as a result of measurement errors being uncorrelated,

$$E[\tilde{x}_k^- v_k^T] = E[v_k \tilde{x}_k^{-T}] = 0 \tag{B.21}$$

Thus,

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \tag{B.22}$$

OPTIMUM CHOICE OF K_k The criterion is to minimize

$$J_k = trace[P_k^+]. \tag{B.23}$$

To find the optimal value of K_k, it is necessary to take the partial derivative of J_k with respect to K_k and equate it to zero, i.e.,

$$\frac{\partial J_k}{\partial K_k} = 0 \tag{B.24}$$

Since

$$\frac{\partial}{\partial A}[trace(ABA^T)] = 2AB, \tag{B.25}$$

it is obtained that

$$\frac{\partial}{\partial K_k}[trace(P_k^+)] = -2(I - K_k H_k) P_k^- H_k^T + 2K_k R_k = 0. \tag{B.26}$$

Solving for K_k,

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \tag{B.27}$$

which is referred to as the *Kalman gain matrix*. Examination of the Hessian of J_k reveals that this value of K_k does indeed minimize J_k.

Substituting K_k into P_k^+ gives, after some manipulation,

$$P_k^+ = (I - K_k H_k) P_k^-. \tag{B.28}$$

Thus far, the discrete state estimate and error covariance matrix behavior *across* a measurement has been described. The equations

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - H_k\hat{x}_k^-) \tag{B.29}$$

$$P_k^+ = (I - K_kH_k)P_k^- \tag{B.30}$$

are so called *update equations* or *measurement update*.

The following section discusses the discrete state estimate and error covariance matrix behavior $\hat{x}$ and P *between* measurements.

PROPAGATION OF STATE VECTOR AND ERROR COVARIANCE

The system equation is x

$$x_k = \Phi_{k-1}x_{k-1} + w_{k-1}. \tag{B.31}$$

The known state transition matrix Φ_k is used to predict the estimate ($E[w_k] = 0$),

$$\hat{x}_k^- = \Phi_{k-1}\hat{x}_{k-1}^+. \tag{B.32}$$

The estimation error

$$\tilde{x}_k^- = \hat{x}_k^- - x_k \tag{B.33}$$

$$= \Phi_{k-1}\hat{x}_{k-1}^+ - \Phi_{k-1}x_{k-1} - w_{k-1} \tag{B.34}$$

$$= \Phi_{k-1}\tilde{x}_{k-1}^+ - w_{k-1}. \tag{B.35}$$

The expected value of the error is

$$E[\tilde{x}_k^-] = \Phi_{k-1}E[\tilde{x}_{k-1}^+] - E[w_{k-1}] = 0 \tag{B.36}$$

under the assumptions that $\tilde{x}_{k-1}^+$ and w_{k-1} are unbiased. This is, extrapolation of the state vector estimate through the state transition matrix Φ_{k-1} does not introduce a bias to $\hat{x}_k^-$.

The error covariance matrix P projected from time t_{k-1} to t_k is

$$\begin{aligned} P_k^- &= E[\tilde{x}_k^- \tilde{x}_k^{-T}] && \text{(B.37)} \\ &= E[\Phi_{k-1}\tilde{x}_{k-1}^+ \tilde{x}_{k-1}^{+T}\Phi_{k-1}^T - w_{k-1}\tilde{x}_{k-1}^{+T}\Phi_{k-1}^T - \Phi_{k-1}\tilde{x}_{k-1}^+ w_{k-1}^T + w_{k-1}w_{k-1}^T] && \text{(B.38)} \\ &= \Phi_{k-1}P_{k-1}^+\Phi_{k-1}^T + Q_{k-1}. && \text{(B.39)} \end{aligned}$$

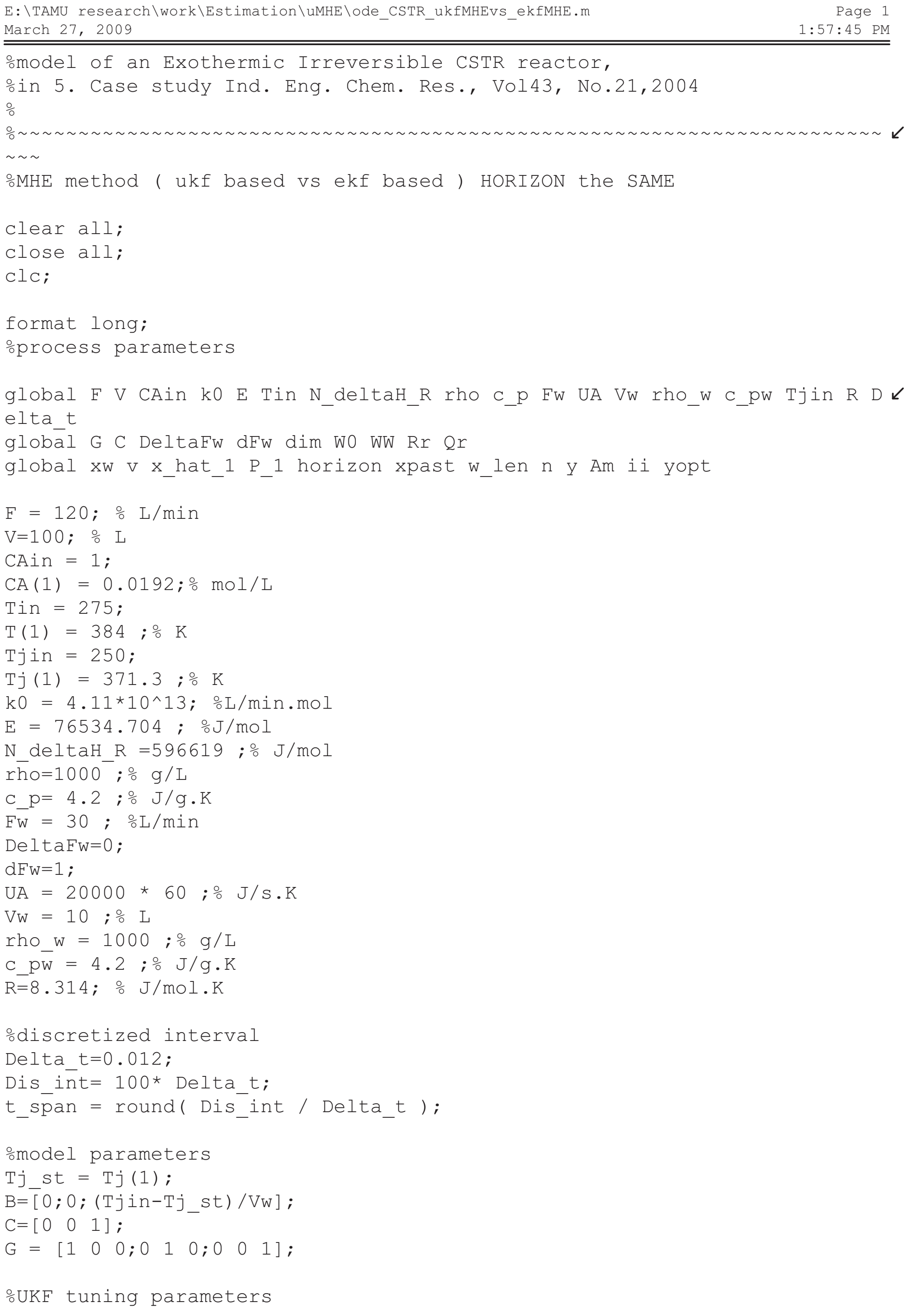
The equations

$$\begin{aligned} \hat{x}_k^- &= \Phi_{k-1}\hat{x}_{k-1}^+ && \text{(B.40)} \\ P_k^- &= \Phi_{k-1}P_{k-1}^+\Phi_{k-1}^T + Q_{k-1}. && \text{(B.41)} \end{aligned}$$

are called prediction equations or time update.

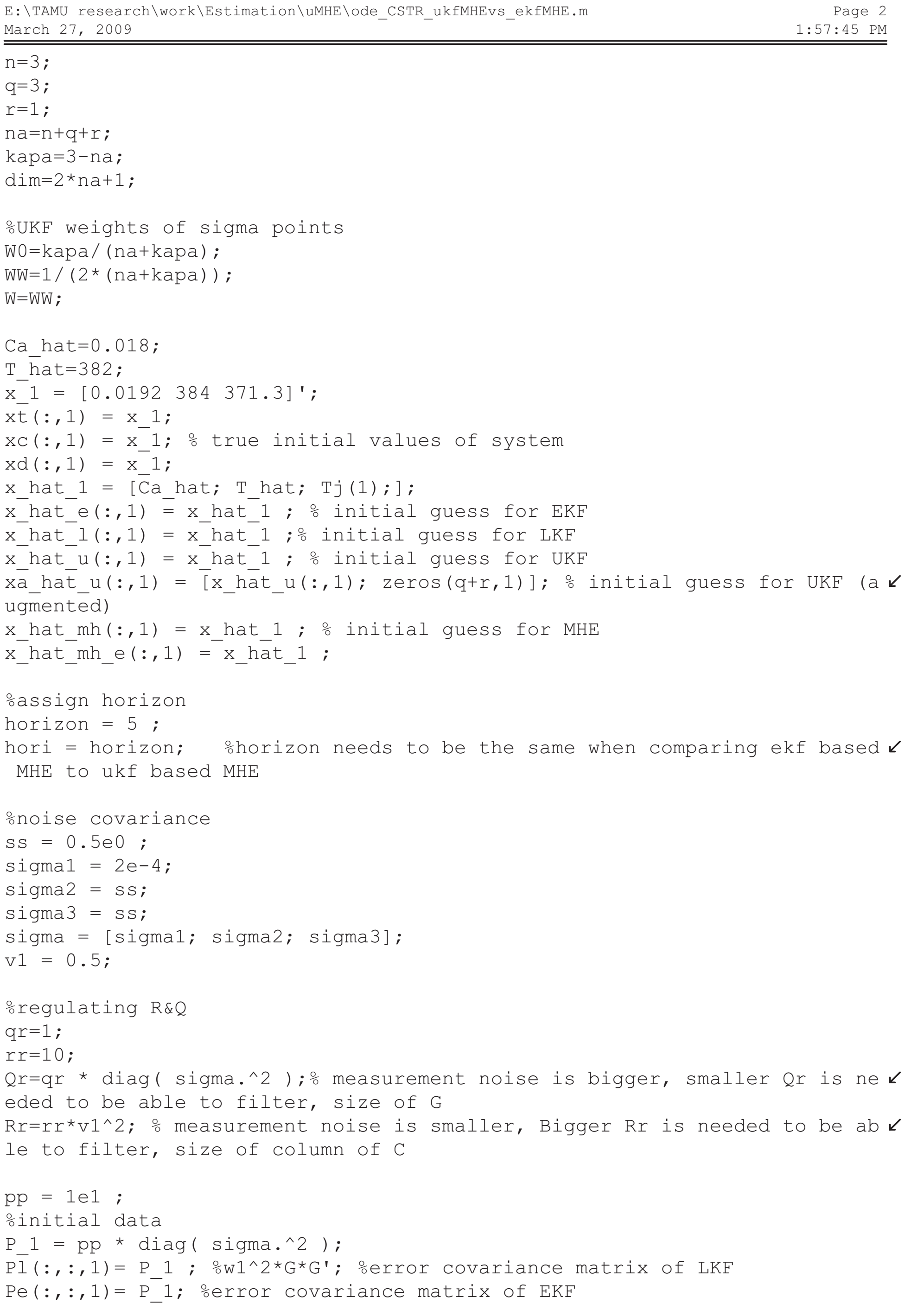
APPENDIX C

MATLAB CODE FOR CASE STUDIES



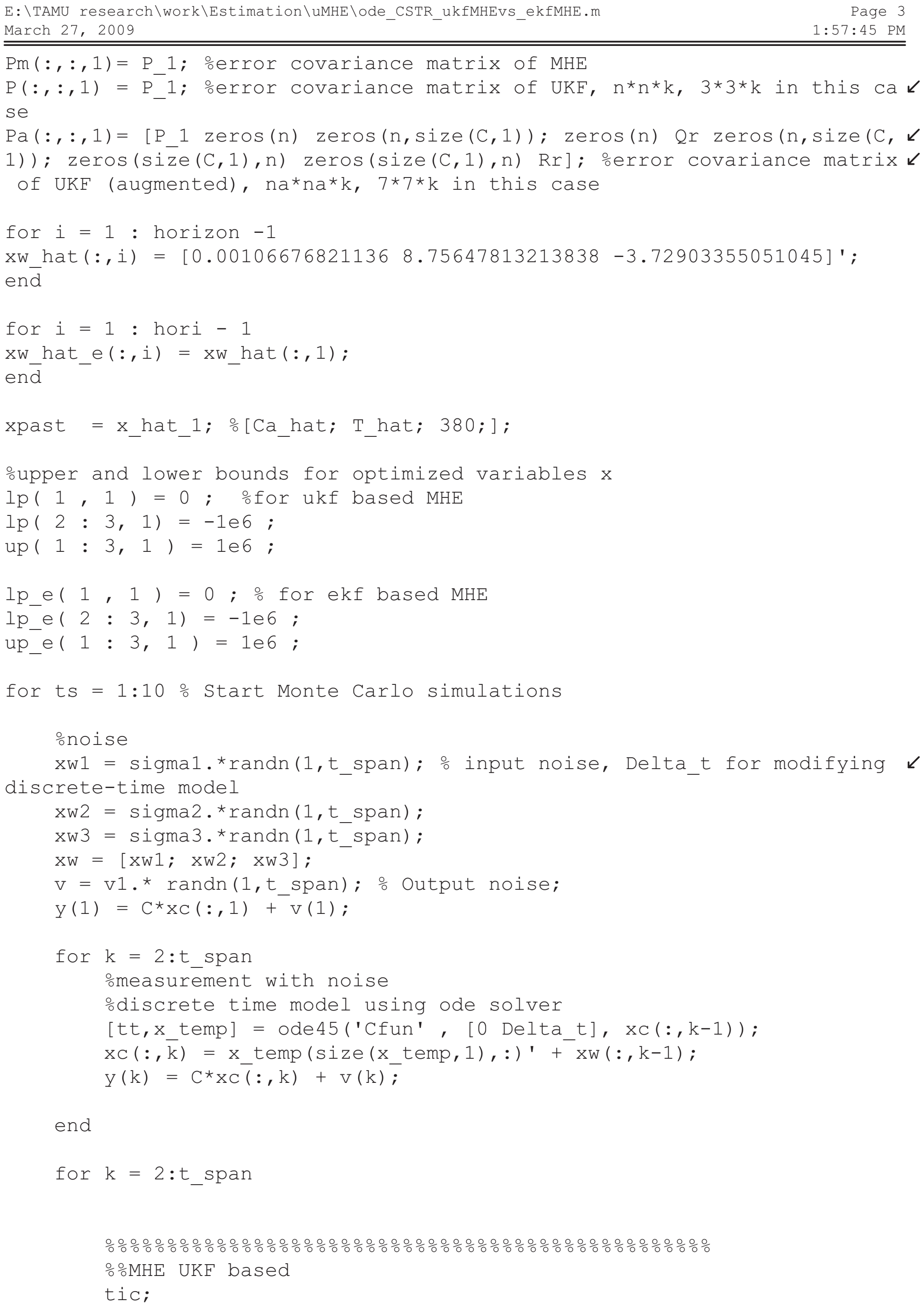
E:\TAMU research\work\Estimation\uMHE\ode_CSTR_ukfMHEvs_ekfMHE.m Page 1
March 27, 2009 1:57:45 PM

```
%model of an Exothermic Irreversible CSTR reactor,
%in 5. Case study Ind. Eng. Chem. Res., Vol43, No.21,2004
%
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ↙
~~~
%MHE method ( ukf based vs ekf based ) HORIZON the SAME

clear all;
close all;
clc;

format long;
%process parameters

global F V CAin k0 E Tin N_deltaH_R rho c_p Fw UA Vw rho_w c_pw Tjin R D↙
elta_t
global G C DeltaFw dFw dim W0 WW Rr Qr
global xw v x_hat_1 P_1 horizon xpast w_len n y Am ii yopt

F = 120; % L/min
V=100; % L
CAin = 1;
CA(1) = 0.0192;% mol/L
Tin = 275;
T(1) = 384 ;% K
Tjin = 250;
Tj(1) = 371.3 ;% K
k0 = 4.11*10^13; %L/min.mol
E = 76534.704 ; %J/mol
N_deltaH_R =596619 ;% J/mol
rho=1000 ;% g/L
c_p= 4.2 ;% J/g.K
Fw = 30 ; %L/min
DeltaFw=0;
dFw=1;
UA = 20000 * 60 ;% J/s.K
Vw = 10 ;% L
rho_w = 1000 ;% g/L
c_pw = 4.2 ;% J/g.K
R=8.314; % J/mol.K

%discretized interval
Delta_t=0.012;
Dis_int= 100* Delta_t;
t_span = round( Dis_int / Delta_t );

%model parameters
Tj_st = Tj(1);
B=[0;0;(Tjin-Tj_st)/Vw];
C=[0 0 1];
G = [1 0 0;0 1 0;0 0 1];

%UKF tuning parameters
```



E:\TAMU research\work\Estimation\uMHE\ode_CSTR_ukfMHEvs_ekfMHE.m Page 2
March 27, 2009 1:57:45 PM

```
n=3;
q=3;
r=1;
na=n+q+r;
kapa=3-na;
dim=2*na+1;

%UKF weights of sigma points
W0=kapa/(na+kapa);
WW=1/(2*(na+kapa));
W=WW;

Ca_hat=0.018;
T_hat=382;
x_1 = [0.0192 384 371.3]';
xt(:,1) = x_1;
xc(:,1) = x_1; % true initial values of system
xd(:,1) = x_1;
x_hat_1 = [Ca_hat; T_hat; Tj(1);];
x_hat_e(:,1) = x_hat_1 ; % initial guess for EKF
x_hat_l(:,1) = x_hat_1 ;% initial guess for LKF
x_hat_u(:,1) = x_hat_1 ; % initial guess for UKF
xa_hat_u(:,1) = [x_hat_u(:,1); zeros(q+r,1)]; % initial guess for UKF (a↙
ugmented)
x_hat_mh(:,1) = x_hat_1 ; % initial guess for MHE
x_hat_mh_e(:,1) = x_hat_1 ;

%assign horizon
horizon = 5 ;
hori = horizon;   %horizon needs to be the same when comparing ekf based↙
 MHE to ukf based MHE

%noise covariance
ss = 0.5e0 ;
sigma1 = 2e-4;
sigma2 = ss;
sigma3 = ss;
sigma = [sigma1; sigma2; sigma3];
v1 = 0.5;

%regulating R&Q
qr=1;
rr=10;
Qr=qr * diag( sigma.^2 );% measurement noise is bigger, smaller Qr is ne↙
eded to be able to filter, size of G
Rr=rr*v1^2; % measurement noise is smaller, Bigger Rr is needed to be ab↙
le to filter, size of column of C

pp = 1e1 ;
%initial data
P_1 = pp * diag( sigma.^2 );
Pl(:,:,1)= P_1 ; %w1^2*G*G'; %error covariance matrix of LKF
Pe(:,:,1)= P_1; %error covariance matrix of EKF
```

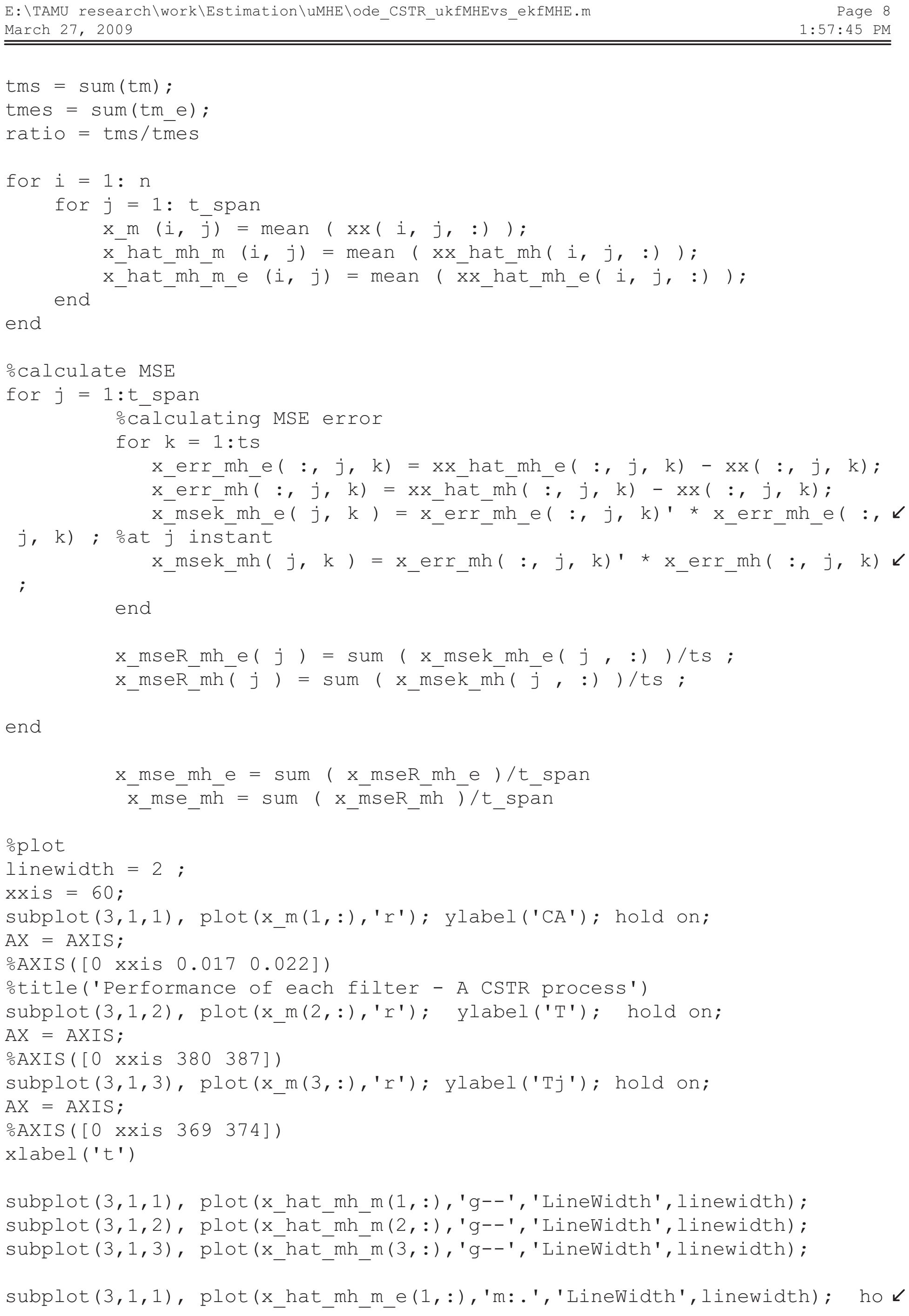


E:\TAMU research\work\Estimation\uMHE\ode_CSTR_ukfMHEvs_ekfMHE.m Page 3
March 27, 2009 1:57:45 PM

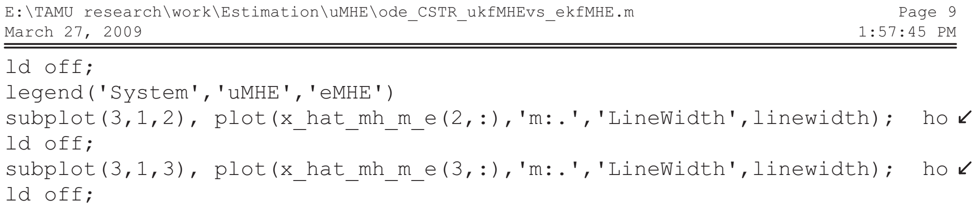
```
Pm(:,:,1)= P_1; %error covariance matrix of MHE
P(:,:,1) = P_1; %error covariance matrix of UKF, n*n*k, 3*3*k in this ca ↙
se
Pa(:,:,1)= [P_1 zeros(n) zeros(n,size(C,1)); zeros(n) Qr zeros(n,size(C, ↙
1)); zeros(size(C,1),n) zeros(size(C,1),n) Rr]; %error covariance matrix ↙
 of UKF (augmented), na*na*k, 7*7*k in this case

for i = 1 : horizon -1
xw_hat(:,i) = [0.00106676821136 8.75647813213838 -3.72903355051045]';
end

for i = 1 : hori - 1
xw_hat_e(:,i) = xw_hat(:,1);
end

xpast  = x_hat_1; %[Ca_hat; T_hat; 380;];

%upper and lower bounds for optimized variables x
lp( 1 , 1 ) = 0 ;  %for ukf based MHE
lp( 2 : 3, 1) = -1e6 ;
up( 1 : 3, 1 ) = 1e6 ;

lp_e( 1 , 1 ) = 0 ; % for ekf based MHE
lp_e( 2 : 3, 1) = -1e6 ;
up_e( 1 : 3, 1 ) = 1e6 ;

for ts = 1:10 % Start Monte Carlo simulations

    %noise
    xw1 = sigma1.*randn(1,t_span); % input noise, Delta_t for modifying  ↙
discrete-time model
    xw2 = sigma2.*randn(1,t_span);
    xw3 = sigma3.*randn(1,t_span);
    xw = [xw1; xw2; xw3];
    v = v1.* randn(1,t_span); % Output noise;
    y(1) = C*xc(:,1) + v(1);

    for k = 2:t_span
        %measurement with noise
        %discrete time model using ode solver
        [tt,x_temp] = ode45('Cfun' , [0 Delta_t], xc(:,k-1));
        xc(:,k) = x_temp(size(x_temp,1),:)' + xw(:,k-1);
        y(k) = C*xc(:,k) + v(k);

    end

    for k = 2:t_span


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%MHE UKF based
        tic;
        %assign optimized variables x and w

        if k >= horizon ;

            ii = k - horizon + 1; %optim = [x(k - horizon + 1), w(k - ho↙
rizon + 1), w1, ..., w(k-1)] and its dimension grows with w

            %set initial guess for optimizer --start fr x0 or x(k-horizo↙
n+1)
            optim( 1 : 3, 1) =  x_hat_mh(:,ii) ;
            xpast = x_hat_mh(:,ii);

            jj = ii ;
            for i = 2 : horizon

                %upper and lower bounds for optimized variables w
                lp( 3*i - 2 : 3*i, 1) = -1e6 ;
                up( 3*i - 2 : 3*i, 1 ) = 1e6 ;
                %set initial guess for optimizer --dimension changes
                optim( 3*i-2 : 3*i, 1) = xw_hat(:,jj) ;

                jj = jj + 1;
            end

            % solve optimization problem

            x_len = length(optim) ;
            w_len = ( length(optim) - n )/ n  ;

            % get optimized initial state x_mh(0) and process noise seri↙
es xw (1...k or N of them)(column vector)
            %~~~~~~dimension of optim grows with the available measureme↙
nts
            optimnew(:, k) = e04jaf(optim, lp, up);

            x_hat_mh(:,ii) = optimnew( 1:n, k);
            %*******************************************
            xw_hat(:, ii:(k-1))  = reshape ( optimnew ( n+1 : x_len, k), ↙
 n, w_len) ;
            % assign w
            xw_hat(:,k) = xw_hat(:,k-1);

            %integrate to obtain the current N estimates based on the
            %current optimized initials
            for j = ii : (k - 1)
                %update x_hat_mh from estimated x0 and w sequence for th↙
e
                %initial values in optim when k > horizon
                [tt,x_mh] = ode45('Cfun' , [0 Delta_t], x_hat_mh(:,j));  ↙
%ode solver
                x_hat_mh(:,j+1) = x_mh(size(x_mh,1),:)' + xw_hat(:,j);

            end
```

```
            %arrival cost by UKF %update P_1 by UKF
            xa_hat_mh(:,ii) = [x_hat_mh(:,ii); zeros(q+r,1)];
            Pa_mh(:,:,ii)= [P_1 zeros(n) zeros(n,size(C,1)); zeros(n) Qr ↙
 zeros(n,size(C,1)); zeros(size(C,1),n) zeros(size(C,1),n) Rr];
            [Chi_mh(:,:,ii), Chi_0_mh(:,:,ii)] = UF(xa_hat_mh(:,ii),Pa_m ↙
h(:,:,ii),na,kapa);

            %update sigma points by nonlinear function
            for i=1:dim
                [tt,x_mh] = ode45('Cfun' , [0 Delta_t], Chi_mh(1:3,i,ii) ↙
); %system model
                Chi_mh(1:3,i,ii+1) = x_mh(size(x_mh,1),:)' +  [Chi_mh(4, ↙
i,ii), Chi_mh(5,i,ii), Chi_mh(6,i,ii)]';

            end


            %prediction
            [x_hat_p_mh(:,ii+1),dx_mh(:,:,ii+1)] = UKF_Pred(Chi_mh(1:n,: ↙
,ii+1),n);

            %measurement
            Chi_y_mh(1,:,ii+1) = C*Chi_mh(1:n,:,ii+1) + Chi_mh(n+4,:,ii+ ↙
1-1); %1*15*k
            y_hat_mh(ii+1) = W0*Chi_y_mh(1,1,ii+1) + W*(sum(Chi_y_mh(1,2 ↙
:dim,ii+1),2)) ;
            Y_hat_mh(1,:,ii+1) =y_hat_mh(ii+1)*ones(1,dim); %1*dim*k
            dy_mh(1,:,ii+1) =Chi_y_mh(1,:,ii+1) - Y_hat_mh(1,:,ii+1); %1 ↙
*dim*k
            dymh(ii+1) = y(ii+1) - y_hat_mh(ii+1);

            %UKF mean and covariance update
            [x_hat_mh(:,ii+1), P_1, xa_hat_mh(:,ii+1), Pa_mh(:,:,ii+1)]  ↙
= UKF_upd( x_hat_p_mh(:,ii+1), dx_mh(:,:,ii+1), dy_mh(r,:,ii+1), dymh(ii ↙
+1), Qr, Rr, n, q, r);

        end


        tm(k) = toc;

    end

   P_1 = pp * diag( sigma.^2 );

     for k = 2:t_span

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %%MHE ekf based
        tic;

        %assign optimized variables x and w
        if k >= hori ;

            ii = k - hori + 1; %optim = [x(k - hori + 1), w(k - hori + 1 ↙
), w1, ..., w(k-1)] and its dimension grows with w

            %set initial guess for optimizer --start fr x0 or x(k-hori+1 ↙
)
            optim_e( 1 : 3, 1) =  x_hat_mh_e(:,ii) ;
            xpast = x_hat_mh_e(:,ii);

            jj = ii ;
            for i = 2 : hori

                %upper and lower bounds for optimized variables w
                lp_e( 3*i - 2 : 3*i, 1) = -1e6 ;
                up_e( 3*i - 2 : 3*i, 1 ) = 1e6 ;
                %set initial guess for optimizer --dimension changes
                optim_e( 3*i-2 : 3*i, 1) = xw_hat_e(:,jj) ;

                jj = jj + 1;
            end


            x_len = length(optim_e) ;
            w_len = ( length(optim_e) - n )/ n  ;

            % get optimized initial state x_mh(0) and process noise seri ↙
es xw (1...k or N of them)(column vector)
            %~~~~~~dimension of optim grows with the available measureme ↙
nts
            optimnew_e(:, k) = e04jaf(optim_e, lp_e, up_e);

            x_hat_mh_e(:,ii) = optimnew_e( 1:n, k);
            %*******************************************
            xw_hat_e(:, ii:(k-1))  = reshape ( optimnew_e( n+1 : x_len,  ↙
k), n, w_len) ;
            % assign w
            xw_hat_e(:,k) = xw_hat_e(:,k-1);

           for j = ii : (k - 1)

                x_int = [x_hat_mh_e(:,j)' 1 0 0 0 1 0 0 0 1] ;
                %update x_hat_mh from estimated x0 and w sequence for th ↙
e
                %initial values in optim when k > hori
                [tt,x_mh_e] = ode45('Efun_cstr' , [0 Delta_t], x_int); % ↙
ode solver
                x_hat_mh_e(:,j+1) = x_mh_e( size(x_mh_e , 1), 1 : 3 )' + ↙
 xw_hat_e(:,j);
                Am( : , 1 , j+1 ) = x_mh_e( size(x_mh_e , 1), 4 : 6 )' ;
                Am( : , 2 , j+1 ) = x_mh_e( size(x_mh_e , 1), 7 : 9 )' ;
                Am( : , 3 , j+1 ) = x_mh_e( size(x_mh_e , 1), 10 : 12 )' ↙
```

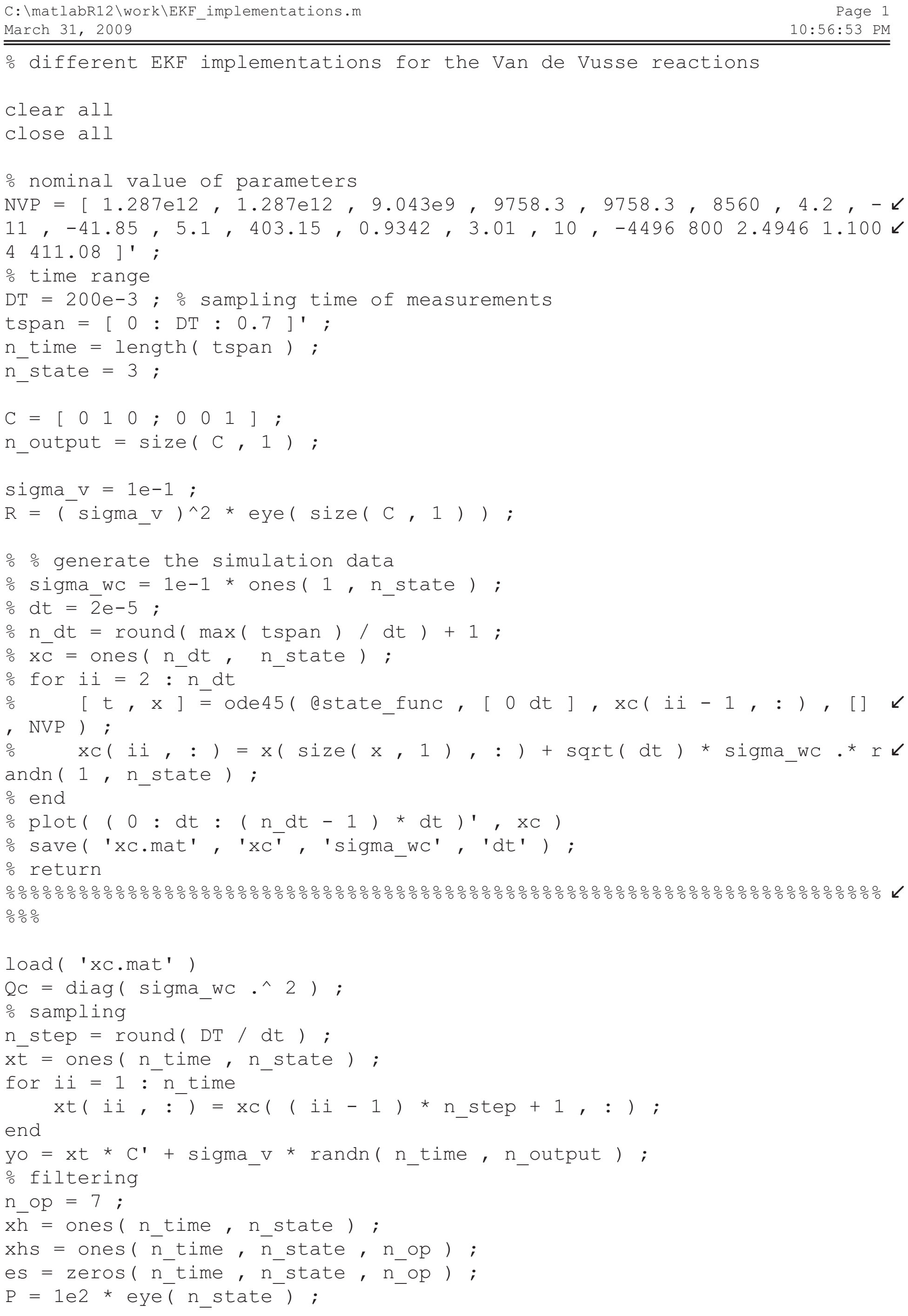
```
 ;
            end


            P_1 = Am( : , : , ii ) * P_1 * Am( : , : , ii )' + G * Qr * ↙
G';
            LL = P_1 * C' * inv(C * P_1 * C' + Rr);
            P_1 = (eye(n,n) - LL*C) * P_1;

        end


        tm_e(k) = toc;

     end  %end for k = 2:t_span

   xx(:, :, ts) = xc;
   xx_hat_mh(:, :, ts) = x_hat_mh;
   xx_hat_mh_e(:, :, ts) = x_hat_mh_e;

   %for next round
   clear   xw_hat;
   clear   xw_hat_e;
   clear   lp;
   clear   lp_e;
   clear   up;
   clear   up_e;
   clear   optim;
   clear   optim_e;
   clear   optimnew;
   clear   optimnew_e;

   for i = 1 : horizon -1
xw_hat(:,i) = [0.00106676821136 8.75647813213838 -3.72903355051045]';
end

for i = 1 : hori - 1
xw_hat_e(:,i) = xw_hat(:,1);
end

   xpast  = x_hat_1;
   P_1 = pp * diag( sigma.^2 );

    %upper and lower bounds for optimized variables x
    lp( 1 , 1 ) = 0 ;
    lp( 2 : 3, 1) = -1e6 ;
    up( 1 : 3, 1 ) = 1e6 ;

    lp_e( 1 , 1 ) = 0 ;
    lp_e( 2 : 3, 1) = -1e6 ;
    up_e( 1 : 3, 1 ) = 1e6 ;

end % End monte carlo simulations
```



```
E:\TAMU research\work\Estimation\uMHE\ode_CSTR_ukfMHEvs_ekfMHE.m                    Page 8
March 27, 2009                                                                 1:57:45 PM

tms = sum(tm);
tmes = sum(tm_e);
ratio = tms/tmes

for i = 1: n
    for j = 1: t_span
        x_m (i, j) = mean ( xx( i, j, :) );
        x_hat_mh_m (i, j) = mean ( xx_hat_mh( i, j, :) );
        x_hat_mh_m_e (i, j) = mean ( xx_hat_mh_e( i, j, :) );
    end
end

%calculate MSE
for j = 1:t_span
         %calculating MSE error
         for k = 1:ts
            x_err_mh_e( :, j, k) = xx_hat_mh_e( :, j, k) - xx( :, j, k);
            x_err_mh( :, j, k) = xx_hat_mh( :, j, k) - xx( :, j, k);
            x_msek_mh_e( j, k ) = x_err_mh_e( :, j, k)' * x_err_mh_e( :, ↙
 j, k) ; %at j instant
            x_msek_mh( j, k ) = x_err_mh( :, j, k)' * x_err_mh( :, j, k) ↙
 ;
         end

         x_mseR_mh_e( j ) = sum ( x_msek_mh_e( j , :) )/ts ;
         x_mseR_mh( j ) = sum ( x_msek_mh( j , :) )/ts ;

end

         x_mse_mh_e = sum ( x_mseR_mh_e )/t_span
          x_mse_mh = sum ( x_mseR_mh )/t_span

%plot
linewidth = 2 ;
xxis = 60;
subplot(3,1,1), plot(x_m(1,:),'r'); ylabel('CA'); hold on;
AX = AXIS;
%AXIS([0 xxis 0.017 0.022])
%title('Performance of each filter - A CSTR process')
subplot(3,1,2), plot(x_m(2,:),'r');  ylabel('T');  hold on;
AX = AXIS;
%AXIS([0 xxis 380 387])
subplot(3,1,3), plot(x_m(3,:),'r'); ylabel('Tj'); hold on;
AX = AXIS;
%AXIS([0 xxis 369 374])
xlabel('t')

subplot(3,1,1), plot(x_hat_mh_m(1,:),'g--','LineWidth',linewidth);
subplot(3,1,2), plot(x_hat_mh_m(2,:),'g--','LineWidth',linewidth);
subplot(3,1,3), plot(x_hat_mh_m(3,:),'g--','LineWidth',linewidth);

subplot(3,1,1), plot(x_hat_mh_m_e(1,:),'m:.','LineWidth',linewidth);  ho ↙
```

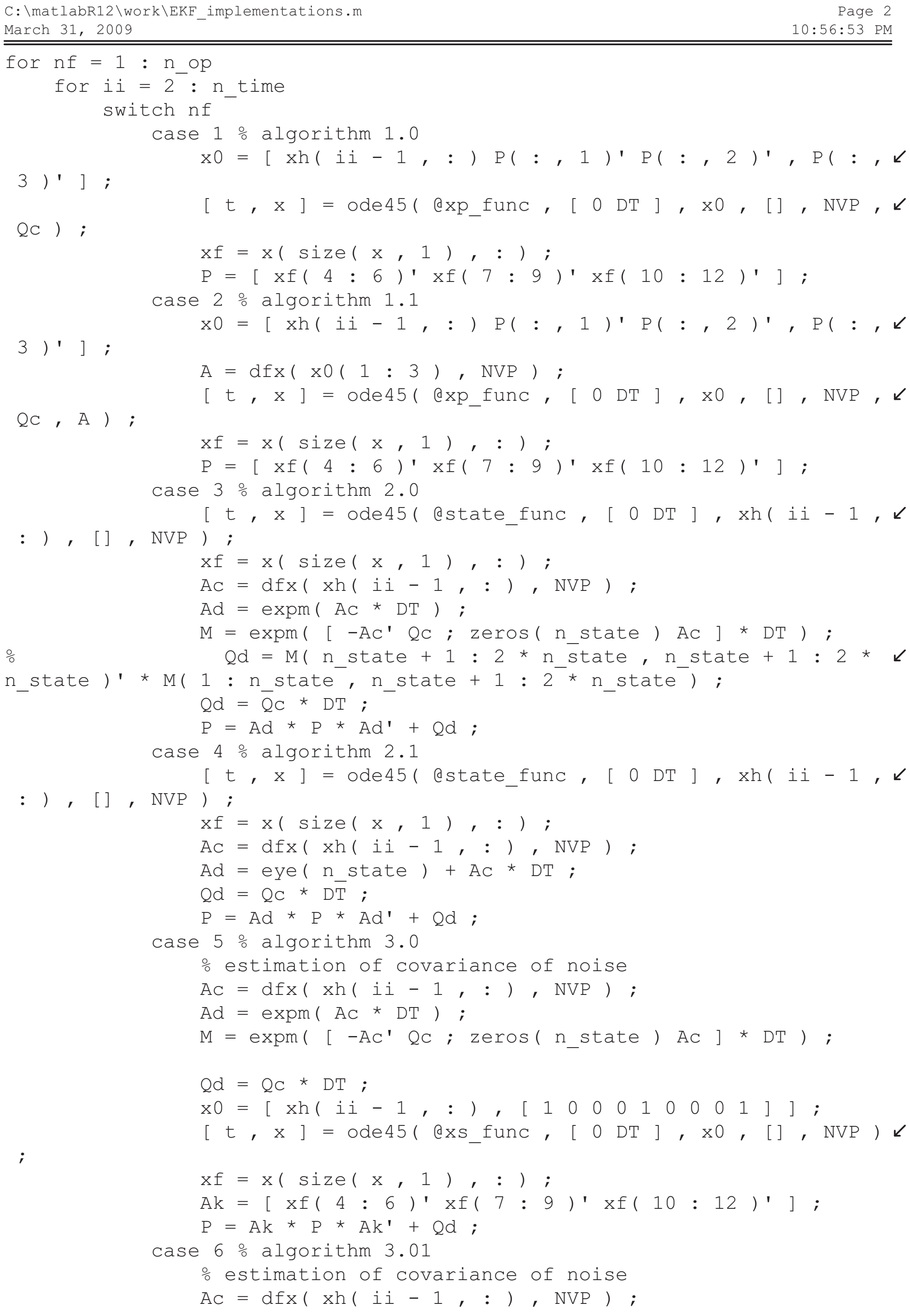


```
ld off;
legend('System','uMHE','eMHE')
subplot(3,1,2), plot(x_hat_mh_m_e(2,:),'m:.','LineWidth',linewidth);  ho ↙
ld off;
subplot(3,1,3), plot(x_hat_mh_m_e(3,:),'m:.','LineWidth',linewidth);  ho ↙
ld off;
```



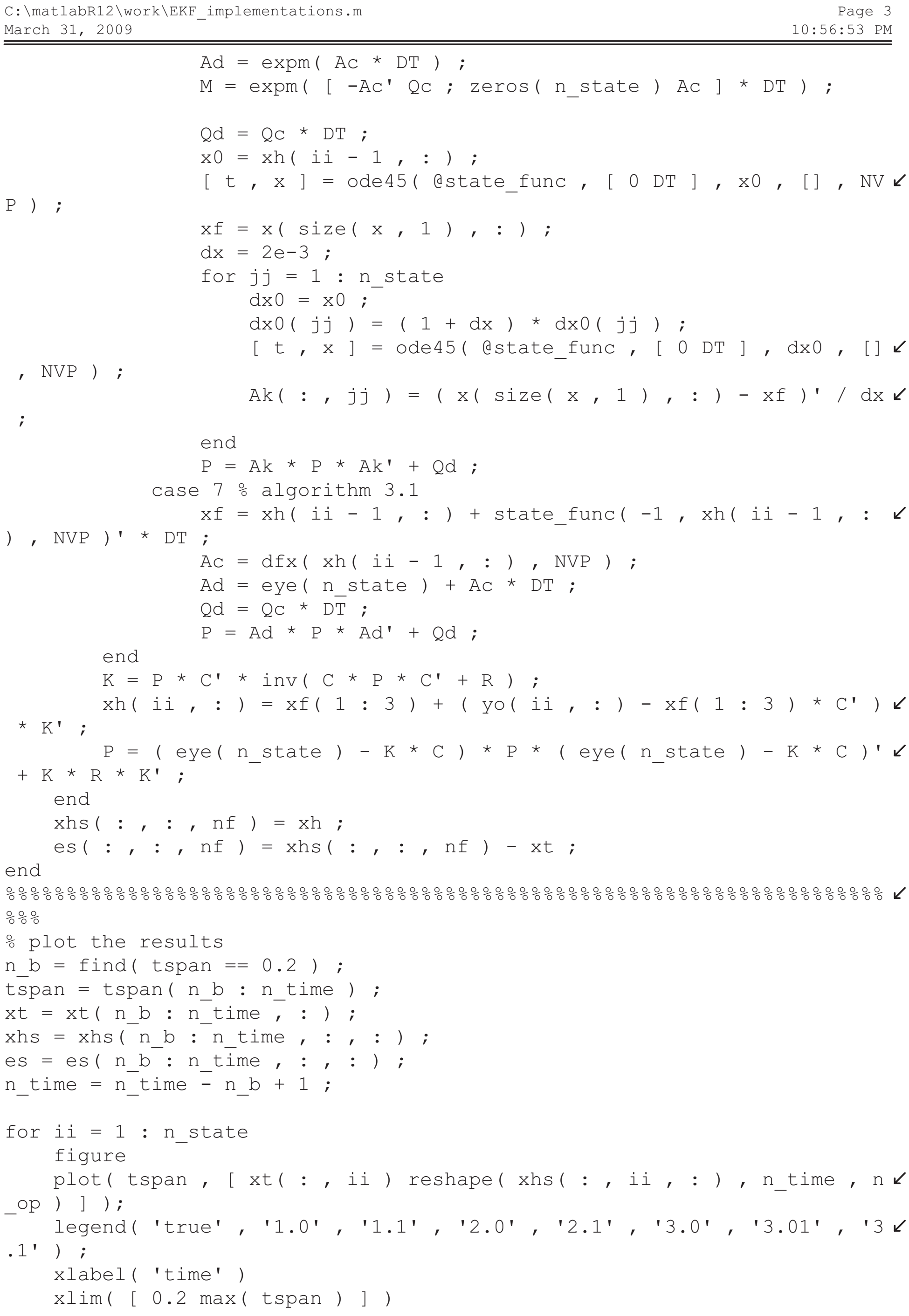
```
C:\matlabR12\work\EKF_implementations.m                                    Page 1
March 31, 2009                                                        10:56:53 PM
```

```
% different EKF implementations for the Van de Vusse reactions

clear all
close all

% nominal value of parameters
NVP = [ 1.287e12 , 1.287e12 , 9.043e9 , 9758.3 , 9758.3 , 8560 , 4.2 , -↙
11 , -41.85 , 5.1 , 403.15 , 0.9342 , 3.01 , 10 , -4496 800 2.4946 1.100↙
4 411.08 ]' ;
% time range
DT = 200e-3 ; % sampling time of measurements
tspan = [ 0 : DT : 0.7 ]' ;
n_time = length( tspan ) ;
n_state = 3 ;

C = [ 0 1 0 ; 0 0 1 ] ;
n_output = size( C , 1 ) ;

sigma_v = 1e-1 ;
R = ( sigma_v )^2 * eye( size( C , 1 ) ) ;

% % generate the simulation data
% sigma_wc = 1e-1 * ones( 1 , n_state ) ;
% dt = 2e-5 ;
% n_dt = round( max( tspan ) / dt ) + 1 ;
% xc = ones( n_dt ,  n_state ) ;
% for ii = 2 : n_dt
%     [ t , x ] = ode45( @state_func , [ 0 dt ] , xc( ii - 1 , : ) , [] ↙
, NVP ) ;
%     xc( ii , : ) = x( size( x , 1 ) , : ) + sqrt( dt ) * sigma_wc .* r↙
andn( 1 , n_state ) ;
% end
% plot( ( 0 : dt : ( n_dt - 1 ) * dt )' , xc )
% save( 'xc.mat' , 'xc' , 'sigma_wc' , 'dt' ) ;
% return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%%%

load( 'xc.mat' )
Qc = diag( sigma_wc .^ 2 ) ;
% sampling
n_step = round( DT / dt ) ;
xt = ones( n_time , n_state ) ;
for ii = 1 : n_time
    xt( ii , : ) = xc( ( ii - 1 ) * n_step + 1 , : ) ;
end
yo = xt * C' + sigma_v * randn( n_time , n_output ) ;
% filtering
n_op = 7 ;
xh = ones( n_time , n_state ) ;
xhs = ones( n_time , n_state , n_op ) ;
es = zeros( n_time , n_state , n_op ) ;
P = 1e2 * eye( n_state ) ;
```



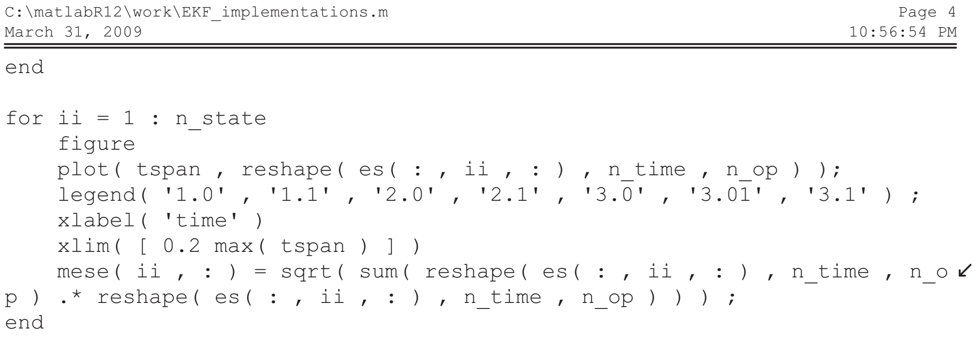
C:\matlabR12\work\EKF_implementations.m Page 2
March 31, 2009 10:56:53 PM

```
for nf = 1 : n_op
    for ii = 2 : n_time
        switch nf
            case 1 % algorithm 1.0
                x0 = [ xh( ii - 1 , : ) P( : , 1 )' P( : , 2 )' , P( : , ↙
 3 )' ] ;
                [ t , x ] = ode45( @xp_func , [ 0 DT ] , x0 , [] , NVP , ↙
 Qc ) ;
                xf = x( size( x , 1 ) , : ) ;
                P = [ xf( 4 : 6 )' xf( 7 : 9 )' xf( 10 : 12 )' ] ;
            case 2 % algorithm 1.1
                x0 = [ xh( ii - 1 , : ) P( : , 1 )' P( : , 2 )' , P( : , ↙
 3 )' ] ;
                A = dfx( x0( 1 : 3 ) , NVP ) ;
                [ t , x ] = ode45( @xp_func , [ 0 DT ] , x0 , [] , NVP , ↙
 Qc , A ) ;
                xf = x( size( x , 1 ) , : ) ;
                P = [ xf( 4 : 6 )' xf( 7 : 9 )' xf( 10 : 12 )' ] ;
            case 3 % algorithm 2.0
                [ t , x ] = ode45( @state_func , [ 0 DT ] , xh( ii - 1 , ↙
 : ) , [] , NVP ) ;
                xf = x( size( x , 1 ) , : ) ;
                Ac = dfx( xh( ii - 1 , : ) , NVP ) ;
                Ad = expm( Ac * DT ) ;
                M = expm( [ -Ac' Qc ; zeros( n_state ) Ac ] * DT ) ;
%                 Qd = M( n_state + 1 : 2 * n_state , n_state + 1 : 2 * ↙
n_state )' * M( 1 : n_state , n_state + 1 : 2 * n_state ) ;
                Qd = Qc * DT ;
                P = Ad * P * Ad' + Qd ;
            case 4 % algorithm 2.1
                [ t , x ] = ode45( @state_func , [ 0 DT ] , xh( ii - 1 , ↙
 : ) , [] , NVP ) ;
                xf = x( size( x , 1 ) , : ) ;
                Ac = dfx( xh( ii - 1 , : ) , NVP ) ;
                Ad = eye( n_state ) + Ac * DT ;
                Qd = Qc * DT ;
                P = Ad * P * Ad' + Qd ;
            case 5 % algorithm 3.0
                % estimation of covariance of noise
                Ac = dfx( xh( ii - 1 , : ) , NVP ) ;
                Ad = expm( Ac * DT ) ;
                M = expm( [ -Ac' Qc ; zeros( n_state ) Ac ] * DT ) ;

                Qd = Qc * DT ;
                x0 = [ xh( ii - 1 , : ) , [ 1 0 0 0 1 0 0 0 1 ] ] ;
                [ t , x ] = ode45( @xs_func , [ 0 DT ] , x0 , [] , NVP ) ↙
 ;
                xf = x( size( x , 1 ) , : ) ;
                Ak = [ xf( 4 : 6 )' xf( 7 : 9 )' xf( 10 : 12 )' ] ;
                P = Ak * P * Ak' + Qd ;
            case 6 % algorithm 3.01
                % estimation of covariance of noise
                Ac = dfx( xh( ii - 1 , : ) , NVP ) ;
```



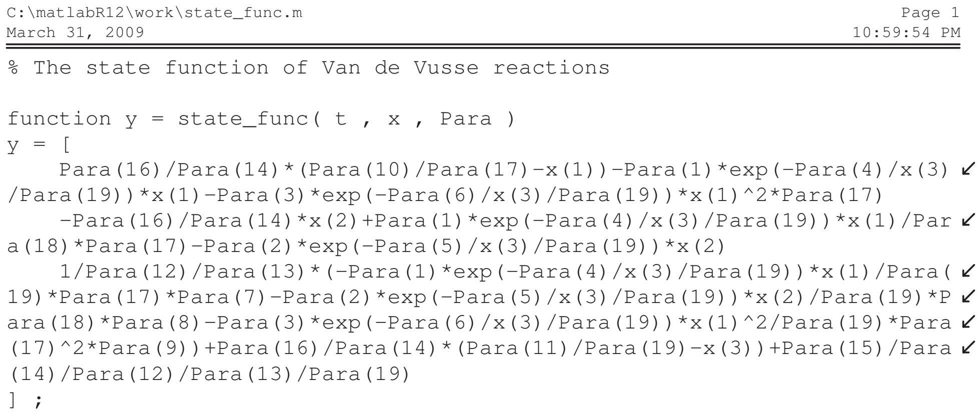
C:\matlabR12\work\EKF_implementations.m Page 3
March 31, 2009 10:56:53 PM

```
                Ad = expm( Ac * DT ) ;
                M = expm( [ -Ac' Qc ; zeros( n_state ) Ac ] * DT ) ;

                Qd = Qc * DT ;
                x0 = xh( ii - 1 , : ) ;
                [ t , x ] = ode45( @state_func , [ 0 DT ] , x0 , [] , NV↙
P ) ;
                xf = x( size( x , 1 ) , : ) ;
                dx = 2e-3 ;
                for jj = 1 : n_state
                    dx0 = x0 ;
                    dx0( jj ) = ( 1 + dx ) * dx0( jj ) ;
                    [ t , x ] = ode45( @state_func , [ 0 DT ] , dx0 , [] ↙
 , NVP ) ;
                    Ak( : , jj ) = ( x( size( x , 1 ) , : ) - xf )' / dx ↙
 ;
                end
                P = Ak * P * Ak' + Qd ;
            case 7 % algorithm 3.1
                xf = xh( ii - 1 , : ) + state_func( -1 , xh( ii - 1 , :  ↙
) , NVP )' * DT ;
                Ac = dfx( xh( ii - 1 , : ) , NVP ) ;
                Ad = eye( n_state ) + Ac * DT ;
                Qd = Qc * DT ;
                P = Ad * P * Ad' + Qd ;
        end
        K = P * C' * inv( C * P * C' + R ) ;
        xh( ii , : ) = xf( 1 : 3 ) + ( yo( ii , : ) - xf( 1 : 3 ) * C' ) ↙
 * K' ;
        P = ( eye( n_state ) - K * C ) * P * ( eye( n_state ) - K * C )' ↙
 + K * R * K' ;
    end
    xhs( : , : , nf ) = xh ;
    es( : , : , nf ) = xhs( : , : , nf ) - xt ;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%↙
%%%
% plot the results
n_b = find( tspan == 0.2 ) ;
tspan = tspan( n_b : n_time ) ;
xt = xt( n_b : n_time , : ) ;
xhs = xhs( n_b : n_time , : , : ) ;
es = es( n_b : n_time , : , : ) ;
n_time = n_time - n_b + 1 ;

for ii = 1 : n_state
    figure
    plot( tspan , [ xt( : , ii ) reshape( xhs( : , ii , : ) , n_time , n↙
_op ) ] );
    legend( 'true' , '1.0' , '1.1' , '2.0' , '2.1' , '3.0' , '3.01' , '3↙
.1' ) ;
    xlabel( 'time' )
    xlim( [ 0.2 max( tspan ) ] )
```



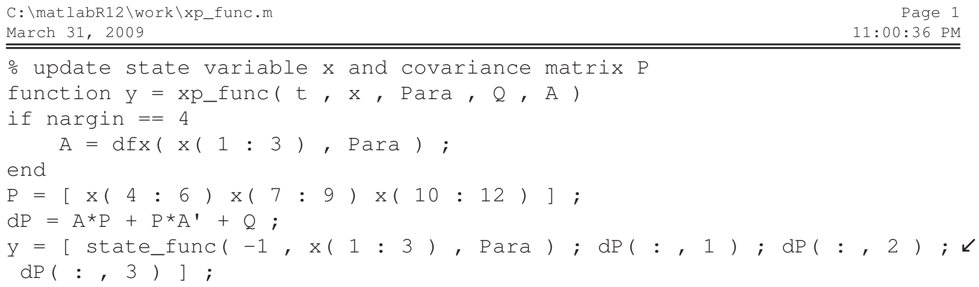
C:\matlabR12\work\EKF_implementations.m Page 4
March 31, 2009 10:56:54 PM

```
end

for ii = 1 : n_state
    figure
    plot( tspan , reshape( es( : , ii , : ) , n_time , n_op ) );
    legend( '1.0' , '1.1' , '2.0' , '2.1' , '3.0' , '3.01' , '3.1' ) ;
    xlabel( 'time' )
    xlim( [ 0.2 max( tspan ) ] )
    mese( ii , : ) = sqrt( sum( reshape( es( : , ii , : ) , n_time , n_o ↙
p ) .* reshape( es( : , ii , : ) , n_time , n_op ) ) ) ;
end
```



```
C:\matlabR12\work\state_func.m                                      Page 1
March 31, 2009                                                 10:59:54 PM
```

```
% The state function of Van de Vusse reactions

function y = state_func( t , x , Para )
y = [
    Para(16)/Para(14)*(Para(10)/Para(17)-x(1))-Para(1)*exp(-Para(4)/x(3) ↙
/Para(19))*x(1)-Para(3)*exp(-Para(6)/x(3)/Para(19))*x(1)^2*Para(17)
    -Para(16)/Para(14)*x(2)+Para(1)*exp(-Para(4)/x(3)/Para(19))*x(1)/Par ↙
a(18)*Para(17)-Para(2)*exp(-Para(5)/x(3)/Para(19))*x(2)
    1/Para(12)/Para(13)*(-Para(1)*exp(-Para(4)/x(3)/Para(19))*x(1)/Para( ↙
19)*Para(17)*Para(7)-Para(2)*exp(-Para(5)/x(3)/Para(19))*x(2)/Para(19)*P ↙
ara(18)*Para(8)-Para(3)*exp(-Para(6)/x(3)/Para(19))*x(1)^2/Para(19)*Para ↙
(17)^2*Para(9))+Para(16)/Para(14)*(Para(11)/Para(19)-x(3))+Para(15)/Para ↙
(14)/Para(12)/Para(13)/Para(19)
] ;
```



C:\matlabR12\work\xp_func.m Page 1
March 31, 2009 11:00:36 PM

```
% update state variable x and covariance matrix P
function y = xp_func( t , x , Para , Q , A )
if nargin == 4
    A = dfx( x( 1 : 3 ) , Para ) ;
end
P = [ x( 4 : 6 ) x( 7 : 9 ) x( 10 : 12 ) ] ;
dP = A*P + P*A' + Q ;
y = [ state_func( -1 , x( 1 : 3 ) , Para ) ; dP( : , 1 ) ; dP( : , 2 ) ; ↙
 dP( : , 3 ) ] ;
```

VITA

Chunyan (Cheryl) Qu received the B.Eng. degree in control science and engineering from Zhejiang University, China, in 2000 and the M.Eng. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2003. Her master thesis was on adaptive control of nonlinear discrete time systems. From 2003 to 2005, she was a monitoring and control system engineer with Ch2mHill Singapore Pte. Ltd. Since Fall 2005, she has been pursuing her Ph.D. degree in the Artie McFerrin Department of Chemical Engineering at Texas A&M University and she graduated in 2009. She was a student member of the American Institute for Chemical Engineers from 2007 to 2009 and was a student member of the Institute of Electrical and Electronics Engineers between 2001 and 2002.

Address: Artie McFerrin Department of Chemical Engineering, c/o Dr. Juergen Hahn, TAMU 3122, College Station, TX 77843.

The typist for this dissertation was Chunyan Qu.