Model Reduction and State Estimation for Nonlinear Complex Systems

by

Sarupa Debnath

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Process Control

Department of Chemical and Materials Engineering

University of Alberta

©Sarupa Debnath, 2024

Abstract

Modern industries are increasingly embracing complex, large-scale processes with interconnected units for their economic benefits. The increasing scale of industrial processes and the complexity of unit interactions substantially complicate the development of advanced process control systems. Model reduction has been recognized as a promising framework for managing large-scale complex systems. It involves deriving low-dimensional models of computationally efficient systems, yet accurately represent the behavior of the overall system. Hence, this is essential for simulating high-dimensional systems in real-time, such as for control decisions or process monitoring, where accurate yet rapidly-solvable models are necessary. However, industrial processes bring added complexities along with high dimensionality such as varying time scales, challenges in measuring real-time outputs, nonlinearities, unknown parameters, and dynamic system behaviors. Therefore, this thesis focuses on addressing the high dimensionality of large-scale systems in process monitoring and control to accommodate various aforementioned industrial complexities.

A class of nonlinear systems, expressed by ordinary differential equations (ODE) with implicit two-time-scale behavior, is taken into account. The system is decomposed into fast and slow subsystems based on the singular perturbation theory and a composite solution for the system is proposed. Local estimators are designed for each subsystem and a one-directional communication scheme is used. A benchmark chemical process example is used to illustrate the proposed method. Attention is then given to nonlinear systems with essential process variables that are not measured but need to be monitored accurately from an operational perspective. It is assumed that a mechanistic model is available but is too computationally complex for estimator design and that only a subset of the states needs to be estimated. The aim is to form a reduced state estimation that can estimate the desired variables. A dynamic sensitivity-based approach is obtained to determine the appropriate inputs and outputs for data collection and data-driven model training. The proposed method is applied to a chemical process, and its applicability is demonstrated.

We consider a type of nonlinear system defined by partial differential equations (PDE) for an agrohydrological system. The inherent large-scale nature of the models stems from the discretization of the underlying PDE. The numerical simulation of such large-scale dynamical systems imposes overwhelming demands on computational resources. A reduced framework for large-scale systems is devised, leveraging unsupervised machine learning. However, in dynamic environments, the properties and behavior of the system might undergo changes. We therefore rely on dynamic data-driven reduced models where different reduced models are computed based on the performance of the model. The proposed approach is implemented to estimate the soil moisture of a real agricultural field located in Lethbridge, Canada. We also explore the proposed dynamic model reduction in estimating the soil water content and soil hydraulic parameters in real-case analysis applied to the farm, using experimental data collected in the summer of 2022.

Preface

The results presented in this thesis are part of the research that is under the supervision of Dr. Jinfeng Liu and is funded by Natural Sciences and Engineering Research Council (NSERC) of Canada.

Chapter 2 of this thesis is a revised version of S. Debnath, S. R. Sahoo, B. D. Nelson, J. Liu, Subsystem decomposition and distributed state estimation of nonlinear processes with implicit time-scale multiplicity. AIChE, 68 (5):e17661, 2022.

Chapter 3 of this thesis is a revised version of S. Debnath, S. R. Sahoo, B. Agyeman, J. Liu, Input-Output Selection for LSTM-Based Reduced-Order State Estimator Design, *Mathematics*, 11 (2):400, 2023. A short version has been presented in the 2023 *World Congress of the International Federation of Automatic Control* in Yokohama, Japan.

Chapter 4 of this thesis is a revised version of S. Debnath, B. Agyeman, S. R. Sahoo, X. Yin, J. Liu, Performance triggered adaptive model reduction for soil moisture estimation in precision irrigation, which has been submitted to *International Journal of Adaptive Control and Signal Processing.* A short version of Chapter 4 has been published in *Proceedings of the IEEE Conference on Decision and Control*, Marina Bay Sands, Singapore, 2023.

A short version of Chapter 5 has been submitted to 2024 *Advanced Control of Chemical Processes* in Toronto, Canada.

Acknowledgements

I would like to express my sincere gratitude and appreciation to my advisor, Dr. Jinfeng Liu, for his invaluable guidance, constant availability, and unwavering support throughout the thesis. He allowed the research to be my own while steering me in the right direction whenever I needed it.

I gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) as well as scholarships and awards, including the Alberta Graduate Excellence Award, Doctoral Recruitment Scholarship, Captain Thomas Farrell Greenhalgh Memorial Graduate Scholarship, and Mary Louise Imrie Graduate Student Award.

I am grateful for the camaraderie and assistance of every member of our research group. I want to express my sincere appreciation to Dr. Soumya Ranjan Sahoo, Dr. Xunyuan Yin, Dr. Benjamin Decardi Nelson, Bernard Twum Agyeman, Song Bo, and Sandra Obiri for engaging in discussions and providing valuable suggestions aiding my research.

I would like to extend my sincere gratitude to the teachers and professors at my alma mater, whose guidance and teachings have been the foundation of my academic endeavor. Lastly, my deepest gratitude to my family for their unconditional love, constant encouragement, and selfless sacrifices throughout this journey.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Due to the increasing global demand, the prevalence of large-scale complex processes has become commonplace in the modern process industry. The industrial processes are inherently nonlinear, dynamically complex, and associated with high dimensional variables typically arise from a wide range of complex physical and chemical phenomena. Such systems are becoming increasingly prevalent in various domains; chemical reactions, heat transfer, and agro-hydrological systems are only a few examples. Performing operational tasks such as predictive modeling, process monitoring and optimal design of control often leads to challenges due to the overwhelming demands on computational resources. This encourages us to develop methods of complexity reduction and acquire relevant information about large-scale systems in a more computationally manageable fashion. Dimensionality reduction aims to decrease the computational burden by generating simplified models that are faster and more cost-effective to simulate, while still accurately capturing the behavior of the original large-scale system. During the past few decades, model reduction for dynamical systems has reached a considerable level of maturity, as reflected in [1, 2, 3, 4].

Traditionally, there has been a substantial demand for enhancing operational performance, with significant capital investments dedicated to improving the productivity and profitability of industrial processes. Model predictive control stands out as an advanced strategy in process control, enabling real-time optimization of process operation while adeptly managing constraints and nonlinearities [5, 6]. Nevertheless, implementing advanced control systems often entails the challenge of obtaining comprehensive measurements of the entire process states. State estimation emerges as a technique to reconstruct complete process state estimates, utilizing a process model and output measurements. However, state estimation algorithms face several challenges in terms of computational burden, the high complexity of their dynamic models, nonlinearities, unknown parameters, and sensitivity to modeling errors and measurement noise [7, 8]. In this thesis, we propose computationally efficient state estimators for process monitoring and control to accommodate various industrial complexities. Our objectives are to systematically analyze and reduce large-scale systems, develop reduced-order estimators, and investigate their performance and applications across various processes.

1.2 Literature review

The complexity of modern engineering systems results from unknown nonlinear relationships, system parameters, and numerous variables. The study of complex systems involves exploring how the relationships between their parts give rise to their collective behavior. Understanding and predicting the behavior of these systems necessitate the use of various modeling techniques. The resulting modeling of the relationship between the input and output of a system typically employs either first principles (mechanistic models) or system identification using industrial data. In the case of first-principle models, process behavior is mathematically represented using ordinary differential equations (ODEs) and partial differential equations (PDEs). In the literature review, we examine state estimation techniques, focusing on a specific class of nonlinear systems in each section. We review the background by investigating the system based on its characteristics and also discuss the existing work, along with the challenges associated with it.

1.2.1 Implicit time-scale multiplicity and state estimation

Chemical processes are inherently nonlinear and generally associated with a common feature, time-scale multiplicity. It usually arises due to the strong coupling of the physical and chemical phenomena occurring at disparate time-scales [9]. Typical examples of multiple-time-scale processes include processes with a large recycle [9], reactors with multiple simultaneous reactions [10], and reactor-separator systems [11] which are modeled by nonlinear ODEs. The time-scale multiplicity occurs typically due to the presence of distinctly different time constants or multiple fast and slow reactions. A direct application of standard control or estimation methods without considering time-scale multiplicity may lead to ill-conditioning, or even the loss of closed-loop stability [12]. To deal with such systems, singular perturbation theory provides a natural framework for modeling, stability analysis, model reduction, and controller design [12]. The design of fully centralized nonlinear controllers on the basis of the entire process system is impractical in terms of computational burden, high complexity of its dynamic models, and sensitivity to modeling errors and measurement noise. These considerations generate vigorous interest in decentralized (composite fast and slow control) [13] and distributed control [14] strategies using proportional (P) control [9, 15] or Model predictive control (MPC) [13, 14]. However, relatively much less attention has been paid to state estimation of systems with time-scale multiplicity.

Recently, in [16], two alternative nonlinear observer design approaches, one full-order and one reduced-order are designed for two-time-scale systems. The full-order observer is designed based on linearization of the original model around its stable steady state, whereas, the reduced-order observer is derived based on a lower-dimensional model to reconstruct the slow states which are used to calculate an invariant manifold for the fast state estimation. In [11], a two-time-scale nonlinear system is decomposed into a fast subsystem and several slow subsystems, and a distributed moving horizon estimation (DMHE) scheme is developed. One directional communication from the slow subsystem MHEs to the fast subsystem MHE is established and sufficient conditions on the convergence of the estimation error of the DMHE are derived. However, both [16] and [11] considered two-time-scale systems that can be described in the standard singular perturbation form where the fast and the slow dynamics are associated with distinct process variables and can be separated explicitly. There is a wide range of applications where slow and fast dynamics cannot be separated explicitly. This implicit time-scale multiplicity cannot be expressed in the standard singularly perturbed form [10].

To handle implicit two-time-scale systems, one approach is to find a coordinate change to transform the implicit two-time-scale system into a standard singular perturbation form. In [17], the coordinate change construction for linear two-time-scale systems is addressed using modal analysis. In [18, 19], an ε-independent (ε is a small variable indicates the separation of time scales) approach was developed to find the coordinate change. In [20], an ε-dependent coordinate change for a class of nonlinear two-time-scale systems was also proposed. A series of results was developed to obtain the standard singularly perturbed representation from the original two-time-scale process using both ε-independent and ε-dependent coordinate change [10]. However, for large-scale systems, it is typically challenging to find such a coordination transformation. More recently, fast and slow subsystem reduction-based methods were developed to address the limitation of the coordinate change-based approaches in control system design [9, 15, 21]. Similar approaches have also been used and developed in aerospace control applications [22]. However, there is little attention given to state estimation of implicit two-time-scale systems, which is a common occurrence in chemical processes. To the best of our knowledge, there is no systematic study that considers slow and fast dynamics separation for implicit two-time-scale systems from a state estimation perspective and provides the corresponding composite estimator design.

In Chapter 2, a composite solution for implicit two-time scale processes is proposed and a distributed state estimation method is designed for chemical processes.

1.2.2 Input-output selection and state estimation

In recent decades, modern process industries are increasingly employing complex, large-scale chemical processes due to their economic efficiency. A rigorous dynamic model for a process can consist of hundreds of differential equations to account for the process dynamics. State estimation of essential process variables is quite demanding from an operation point of view to achieve better product quality and optimal utilization of available resources. In process industries, state estimators or observers are commonly used to estimate unknown variables based on a process model and some measurable variables. Using a detailed mechanistic model to perform state estimation is often challenging due to increased complexity and higher computational cost. In many applications, the number of key variables that should be estimated is indeed much smaller than the number of the internal states of the entire system. A reduced-order estimator that can estimate the key variables is sufficient. brunton2016discovering In the literature, there are some results on state estimation based on the reduced-order models. In [23], a state estimation scheme of wastewater treatment plants was developed based on model approximation, in which a reduced-order model was obtained based on the proper orthogonal decomposition (POD) approach. In [24], a state estimation scheme was developed on a reduced-order approximation of the forecast error using a Kalman filter. The reduced-order system was developed by the balanced truncation of the Hankel operator representation of the estimation error. In [25], a model reduction was performed using a matched asymptotic expansions method for an implicit two-time-scale system and a distributed state estimation was implemented to demonstrate the improved computational time and accuracy. In [26], a structure-preserving model reduction method using trajectory-based unsupervised machine learning techniques was used to develop an adaptive moving horizon estimation algorithm. A reduced state observer was developed for a linearized reduced system using the balancing model reduction technique [27]. Different data-driven methods are used in many applications to determine the structure of the reduced-order model and to reveal important physical properties such as sparse regression [28, 29, 30], sparse identification of non-linear dynamics (SINDy) [31] and Koopman operator [32]. In process modeling, hybrid modeling is another data-based modeling, wherein it combines a kinetic model with a data-based model which improves the model accuracy and robustness [33, 34, 35].

In recent years, machine learning techniques, in particular, neural networks [33, 34, 35] have attracted significant attention in reduced-order model and estimator development due to the data-driven nature and easy-to-implement feature of these techniques. In the literature, there are many studies that have used machine learning to develop data-driven models which are in general reduced-order models. For example, in [36, 37, 38], machine learning was integrated with traditional observer or estimator frameworks for data-based state estimation schemes. However, a careful examination of these studies reveals that there lacks a systematical method to determine how to choose the appropriate inputs and outputs for these machine learning-based model development. The well-selected inputs and outputs can make sure that the resulting data-driven model captures the dynamics needed for the estimation of the key target variables and can significantly reduce the model training effort [39, 40].

Another relevant topic in the literature is inferential soft sensors built on process data and explores the correlation between inputs and target outputs. These inferential soft sensors, in general, do not consider the dynamics of the system. There are many applications of soft sensors based on machine learning and statistical techniques such as neural networks [41, 42], principal component regression [43], and partial least squares regression [44]. A soft sensor is developed using a probabilistic slow feature analysis by extracting slowly varying patterns from noisy process data [45]. In [46], an information fusion system is developed to take the benefits from fast-rate sampling of online data and the high accuracy of lab data, which is slow-rate sampled data, for soft sensing. These soft sensors are useful and can provide predictions of unmeasured variables and are typically easy to implement. However, as they do not take the dynamics of the process into account explicitly, their performance may be limited. If a dynamic model can be developed, state estimation provides much-improved estimation performance.

The selection of input-output and a reduced estimator design are provided in detail in Chapter 3.

1.2.3 Adaptive model reduction and soil moisture estimation for agro-hydrological systems

Discretizing PDEs typically gives rise to high-dimensional problems, which pose significant challenges when seeking numerical solutions. This phenomenon is referred to as the "curse of dimensionality," in which the required computational effort scales exponentially with system dimension, making traditional grid-based methods infeasible. In this context, we consider soil water dynamics in agro-hydrological systems modeled by the Richards equation.

With population growth, climate change, and environmental pollution, freshwater scarcity has become a global risk [47]. Agriculture, the primary consumer of freshwater [48], coupled with the persistent problem of low water efficiency [49], intensifies the issue of water scarcity further by utilizing about 38% of the land. Improving water usage efficiency is a critical step in managing the water and land crisis. The open-loop control method, which lacks real-time feedback, is a widely used irrigation practice in agriculture. However, it heavily relies on farmers' experience and observations rather than on actual field conditions like soil moisture content. This can lead to over-irrigation or under-irrigation, both of which can be detrimental to plant growth. In recent years, precision agriculture (PA) has gained popularity in the agricultural community. The technique provides the appropriate amount of water and nutrients for optimal plant growth. A closed-loop irrigation system that uses field soil moisture data to make decisions can significantly increase water use efficiency. However, soil moisture sensors are typically placed at selected locations of the soil profile. To achieve a comprehensive understanding of the dynamic behaviors of the soil moisture across the entire field, the implementation of an appropriate state estimation method is essential. State estimation can enable real-time estimation of the soil moisture across the field by utilizing the measurements collected from the available sensors. Significant progress in developing various computing methods for smart and sustainable closed-loop agriculture irrigation has been made. These methods involve sensor placement [50], soil moisture estimation [51], simultaneous estimation of soil moisture and soil parameters [52, 53], model predictive controller (MPC) design [54], and scheduler design [55, 56]. In particular, various strategies for estimating soil moisture based on real-time field measurements have been explored. Several well-known algorithms such as the extended Kalman filter (EKF) [57], ensemble Kalman filter (EnKF) [58], and moving horizon estimation (MHE) [51] have been applied for data assimilation of soil moisture.

The Richards equation is a nonlinear, and three-dimensional PDE, which assumes local equilibrium between soil water content and soil water potential [59]. To ensure numerical stability and satisfy the local equilibrium assumption, fine discretization (a few centimeters to a few meters) is often required which leads to a high number of nodes, typically ranging from 10^4 to 10^8. Therefore, estimating soil water content is a major challenge due to the high dimensionality of agro-hydrological systems. For instance, the online optimization associated with nonlinear MHE can become intractable due to the high-order of the system. The EKF is a common generalization of the classic Kalman filter to nonlinear systems. However, linearizing the nonlinear model consecutively and propagating the covariance matrix explicitly to handle the nonlinear observation may become infeasible for such a large dimensional system.

Model order reduction is a widely used technique to handle high-dimensional systems and obtain computationally efficient models. Some of the commonly used methods include proper orthogonal decomposition (POD) [60], optimal Hankel norm reduction, [61], slow feature analysis [62], and balanced truncation methods [63]. Yet, these methods typically fall short of preserving the physical meanings of the system's states. In particular, when any of these proposed methods is implemented in an agricultural system, the state of a reduced-order model will no longer accurately reflect the field's soil moisture. Researchers have introduced various techniques for maintaining the topology of the system state in cluster-based model reduction, as discussed in [64, 50]. Recently, in [65, 55], a reduced model based on system trajectories was proposed for an agrohydrological system. This method can effectively represent the system's behavior throughout the entire growing season. Nonetheless, these approaches are typically implemented for linear systems, and a single reduced-order model is built at the beginning of the season to operate for the entire duration. In [51], another approach that employs dynamic model reduction and an MHE-based estimator was proposed. It is important to mention that these trajectories are not updated in real-time. This limitation could lead to a substantial discrepancy between the system model and the actual dynamics of soil water, presenting challenges in adapting to real-time conditions that vary over time. Additionally, MHE, an online optimization method, generally leads to longer computation times than explicit recursive methods such as EKF.

Chapter 4 proposes an adaptive model reduction method with triggering criteria and estimator design for large-scale agricultural fields.

1.2.4 An application of adaptive model reduction to soil moisture and its hydraulic parameter estimation via remote sensing

While both tensiometers and point sensors are reliable for localized measurements, these methods require significant labor, incur substantial costs, and involve intricate procedures [66]. Unlike point sensors, microwave radiometers offer a broader perspective, essential for large-scale moisture analysis. The integration of microwave radiometer sensors into center-pivot irrigation systems presents a unique opportunity for real-time, comprehensive soil moisture monitoring [67]. However, the use of microwave radiometers also introduces new challenges. One significant issue is the constantly changing locations of measurement as the sensors rotate with the irrigation system. Hence, the ability of microwave radiometer sensors to provide a continuous and complete mapping of soil moisture over time and space is limited, resulting in spatial and temporal inconsistencies in soil moisture observations for irrigation systems. A promising approach is the integration of sparse sensor data with advanced modeling techniques. This strategy, known as sequential data assimilation, merges real-time sensor observations with a dynamic mathematical model, creating more accurate and comprehensive soil moisture estimates [57, 68, 69]. They enhance the precision of soil moisture readings by filling in the gaps in sensor data, ensuring that irrigation systems operate more effectively and efficiently.

In improving soil moisture predictions in agricultural models, the accuracy depends on precise soil hydraulic parameters, which have a significant impact on the model's ability to simulate soil moisture content. Recognizing the significant impact these parameters have on a model's performance in predicting soil moisture content, a simultaneous soil moisture and parameters estimation is more suitable as it considers the interdependence between these estimates, leading to enhanced accuracy [52, 70, 71]. In [72], sensitivity analysis is carried out to evaluate parameter estimability for a remote sensor mounted on a center-pivot. The study proceeded to estimate the soil moisture and selected hydraulic parameters by assimilating the soil moisture data from microwave remote sensing into the Richards equation utilizing EKF. Despite the advancements made through this methodology, the study acknowledges that scalability issues in larger agricultural fields present a significant challenge. The data assimilation technique used in the agro-hydrological model is inefficient and demands high computational resources due to the substantial numerical complexities involved in solving the model. This limits many studies to only applying a 1D system or low-dimensional 3D systems [58, 73, 56].

Chapter 4 is expanded to include considerations for rotating measurements and parameter estimation. This is applied and validated using actual field data from microwave remote sensing in Chapter 5 by implementing a data assimilation approach based on an adaptive reduced model.

1.3 Contributions and thesis outline

The rest of the thesis is organized as follows:

In Chapter 2, a subsystem decomposition approach and a distributed estimation scheme are discussed for a class of implicit two-time-scale nonlinear systems. Taking advantage of the time scale separation, A system is first decomposed into fast subsystems and slow subsystems according to the dynamics. Further, an approach that combines the approximate solutions obtained from both the fast and slow subsystems to form a composite solution of the system is presented. Also, based on the fast and slow subsystems, a distributed state estimation scheme is explained to handle the implicit time-scale multiplicity. It is found that the slow estimator is entirely decoupled from the fast estimator which is a significant difference from the control of two-time-scale systems. The decoupling ensures that only one-directional information transmission from the slow estimator to the fast estimator is needed and the fast estimator does not send out any information. To demonstrate the usefulness of the proposed method, a typical chemical process is considered in the simulations and the simulation results demonstrate the effectiveness of the proposed approach.

In Chapter 3, a sensitivity-based approach is considered to construct reduced-order state estimators based on recurrent neural networks (RNN). It is assumed that a mechanistic model is available but is too computationally complex for estimator design and that only some target outputs are of interest and should be estimated. A reduced-order estimator that can estimate the target outputs is sufficient to address such a problem. A systematic method is introduced based on sensitivity analysis to determine how to select the appropriate inputs and outputs for data collection and data-driven model development to estimate the desired outputs accurately. Specifically, a long short-term memory (LSTM) neural network, a type of RNN, is considered as the tool to train the data-driven model. Based on it, an extended Kalman filter (EKF), a state estimator, is designed to estimate the target outputs. Simulations are carried out to illustrate the effectiveness and applicability of the proposed approach. The main contributions of this work include (a) a systematic approach for input and output selection for reduced-order model development based on sensitivity analysis; (b) a modified EKF design that can take advantage of the reduced-order model; (c) detailed simulations illustrating the applicability and effectiveness of the proposed approach.

In Chapter 4, a state estimation method for large-scale agricultural fields is proposed using an error-triggered adaptive model reduction that utilizes a trajectory-based clustering technique. A large-scale system is obtained by discretizing a 3D polar Richards equation that characterizes complex water movement dynamics. An adaptive extended Kalman filter (EKF) is designed accordingly based on the adaptive reduced model. The performance of the proposed method is compared to an EKF designed based on the Richards equation, full state estimator, and an EKF based on a non-adaptive reduced model. A small demo agricultural field is considered and extensive simulations are carried out to compare with the full estimator based on Richards equation. Further, it is applied to a large field, and its performance is compared to a reduced EKF based on a centralized reduced model. Also, a periodic triggering algorithm with the same structural similarities to a centralized method is explored to mitigate weather forecast and irrigation decision uncertainties associated with the centralized method. The proposed approach is applied to different scenarios to show the effectiveness and superiority of the proposed framework.

In Chapter 5, a real case study is presented based on the findings in Chapter 4. We aim to enhance the computational efficiency and accuracy of soil moisture estimates by implementing a reduced-order approach that simultaneously estimates soil moisture and its soil hydraulic parameters. The intermittent nature of data acquisition and the rotating mechanism of microwave remote sensors necessitate an information fusion system that incorporates the 3D cylindrical Richards equation, a reduced EKF, and actual measurements. To address high dimensionality in the estimator, we adopt a sequential triggering approach in developing an adaptive reduced model, minimizing the need for frequent model adjustments. The results demonstrate the effectiveness of the adaptive model reduction approach in improving both the computational efficiency and accuracy of soil moisture estimation within a large-scale agro-hydrological system.

Chapter 6 concludes these works and discusses future research directions.

Chapter 2

Subsystem decomposition and distributed state estimation of nonlinear processes with implicit time-scale multiplicity

In this chapter, we consider a class of implicit two-time-scale nonlinear systems and propose a distributed state estimation method based on fast and slow subsystem decomposition. In a distributed state estimation scheme, a system is decomposed into several subsystems, and a subsystem estimator is designed for each subsystem. These subsystem estimators work together collaboratively by exchanging information to estimate the entire system state. Because of information exchange between subsystems, a distributed framework can give performance that is close to or equal to a centralized framework (when it is feasible). But compared with a centralized framework, a distributed framework provides more structural flexibility, improved fault tolerance, and reduced computational complexity [74, 75, 76].

Specifically, we borrow the idea of the method of matched asymptotic expansions to decompose the system into a fast subsystem and a slow subsystem considering different limiting conditions on ε. The solutions of the fast and slow subsystems are combined to calculate the composite solution of the actual system state. The composite solution approximates the actual system state. Further, a fast EKF is designed for the fast system and a slow MHE is designed for the slow subsystem. The fast EKF and slow MHE form a distributed scheme. It is found that the slow MHE is entirely decoupled from the fast EKF which is a significant difference from the control of two-time-scale systems. The decoupling ensures that only unidirectional information transmission from the slow MHE to the fast EKF is needed and the fast EKF does not send out any information. Also, we make comparisons between the proposed approach and a decentralized scheme and a centralized MHE scheme.

The system description and subsystem decomposition are performed in Section 2.1. Section 2.2 addresses the proposed distributed state estimation design. The application of the proposed method to a chemical process example demonstrates its applicability and effectiveness via simulations in Section 2.3. Finally, we conclude our results in Section 2.4.

2.1 Preliminaries

2.1.1 Notation

The operator $L_f h$ represents the Lie derivative of function h with respect to function f, calculated following $L_f h(x) = \frac{\partial h}{\partial x} f(x)$. $L_f^r h$ represents the r^{th} order Lie derivative of function f, denoted by $L_f^r h(x) = L_f L_f^{r-1} h(x)$. Subscript f and s denote fast and slow subsystems respectively unless mentioned. The subscript ss denotes a variable associated with the steady state. A matrix is full row rank when each of the rows of the matrix are linearly independent and is full column rank when each of the columns of the matrix are linearly independent.

System description

In this work, we consider a class of nonlinear system that can be described in the following singularly perturbed form [77]:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + \frac{1}{\varepsilon}b(x(t))k(x(t)) + w(t) \tag{2.1a}$$

$$y(t) = h(x(t)) + v(t) \tag{2.1b}$$

where $x \in X \subset R^{n_x}$ is the vector of state variables of independent time variable t, $u \in R^{n_u}$ is the vector of the manipulated inputs, $y \in R^{n_y}$ is the vector containing all the measured outputs, $w \in R^{n_x}$ denotes system disturbances and $v \in R^{n_y}$ is measurement noise, the initial condition is $x(0) = x_0$ and ε is a small parameter, such that $0 < \varepsilon << 1$. $f(x)$ and $k(x)$ are analytic vector fields of dimensions n_x and p_x ($p_x < n_x$), $g(x)$ and $b(x)$ are analytic matrices of dimensions $(n_x \times n_u)$, $(n_x \times p_x)$, respectively. In Eq. (2.1), the term $\frac{1}{\varepsilon}b(x(t))k(x(t))$ corresponds to the fast dynamics of the system [77]. We consider that the matrix $b(x)$ and the Jacobian $\frac{\partial k(x)}{\partial x}$ have full column and row rank, respectively. Though the condition on the rank of $b(x)$ is not restrictive. Further, it is assumed that output $y(t)$ is continuously measured [10].

Equation (2.1) describes a class of systems where the separation of fast and slow dynamics is not explicit. There is a wide variety of processes that exhibit time-scale multiplicity. This phenomenon is generally induced by different thermal properties, mass transfer rates, and chemical kinetics of the reaction [77]. These systems are mainly characterized by the presence of a small parameter ε in the explicit mathematical models. The reciprocal of such a small parameter gives a very large term that is responsible for the existence of disparate time-scale features. Typical examples of such processes include those with multiple fast and slow reactions or fast heat/mass transfer rates. These processes typically exhibit two distinctly different magnitudes of gains in different input directions and different time constants: a large and dominant time constant and a small-time constant associated with slow and fast dynamics, respectively. The small parameter ε differentiates the nature of the fast and slow dynamics, and both the dynamics control the speed of the system, as their names suggest. The overall system response is affected by either fast or slow dynamics or both [15].

In the remainder of this section, we will introduce how to derive the fast and slow dynamics of system (2.1) using matched asymptotic expansions method. Further, a composite solution is introduced to retrieve the system's complete dynamics through a correction term.

2.1.2 Two-time-scales decomposition

System (2.1) describes a class of two-time-scale systems with implicit time-scale separation where each state can have both fast and slow dynamics. Solving such a system is quite different from solving a system with explicit time-scale separation [11]. Matched asymptotic expansion is used to decompose and solve the system. It involves finding different approximate solutions or asymptotic expansions, valid for a particular time-scale, and then combining these different solutions to give a single approximate solution valid for the original system. Due to the brisk nature of the fast dynamics, we need a stretched time-scale to capture the dynamics. Conversely, the slow dynamics are sluggish in nature, so a squeezed time-scale is preferably apt for it.

Specially, in asymptotic expansions, the limiting solutions for fast or slow dynamics are obtained. An outer (reduced layer) approximation is obtained for the slow dynamics and inner (boundary-layer) approximation for the fast dynamics is obtained [12]. While combining the inner and outer approximations for the actual dynamics, a correction term needs to be subtracted to avoid considering their overlap value twice. In the following, we consider the nominal deterministic system to illustrate how to decompose the fast and slow dynamics.

Decomposition of fast dynamics

In the inner approximation, a fast (stretched) time τ is defined as $\tau = \frac{t}{\varepsilon}$. Multiplying both sides of Eq. (2.1a) by ε, converting into τ time-scale, and considering the limiting case $\varepsilon \to 0$, we obtain the fast dynamics of system (2.1):

$$\frac{dx(\tau\varepsilon)}{d\tau} = b(x(\tau\varepsilon))k(x(\tau\varepsilon)) \tag{2.2}$$

Denoting $x(\tau\varepsilon)$ as $x_f(\tau)$, the above equation transforms into,

$$\frac{dx_f}{d\tau} = b(x_f)k(x_f) := f_f(x_f) \tag{2.3}$$

The above system as shown as (2.3) approximates the fast dynamics of the original system (2.1). $f_f(x_f)$ is the analytic vector field of the dimension of n_{x_f}. If the steady state of the fast dynamics is x_{fss}, it satisfies the following condition:

$$k(x_{fss}) = 0, \tag{2.4}$$

We consider systems of the form of Eq. (2.1) for which the matrix $b(x)$ and the Jacobian $\frac{\partial k(x)}{\partial x}$ have full column and row rank, respectively. The condition of full column rank of $b(x)$ ensures that it cannot be zero. However, the condition on the rank of Jacobian $\frac{\partial k(x)}{\partial x}$ assures that in the limit $\varepsilon \to 0$, the differential-algebraic equations (DAE) system that describes the slow dynamics of Eq. (2.1) (in the next section) has a finite index and a well-defined solution, and it is satisfied in typical chemical process applications.

We obtain the set of linearly independent constraints (2.4) that must be satisfied in the slow time-scale t. In the fast time-scale τ, the algebraic constraints Eq. (2.4) are not satisfied unless steady state reaches [77].

The initial condition $x(0) = x_0$ of the system (2.1) applies in the inner approximation i.e. the initial condition of the fast dynamics $x_f(\tau = 0) = x_{f0} = x_0$.

Decomposition of slow dynamics

In the outer approximation, multiplying system (2.1) by ε and considering the limiting case $\varepsilon \to 0$ in the slow time-scale, we obtain the constraint $k(x_s) = 0$ which includes p_x linearly independent scalar equations. Note that x_s denotes the vector of slow states. Note also that the constraint $k(x_s) = 0$ should be satisfied by the slow dynamics for all time. This also implies that the constraint should be satisfied by the slow dynamics at time $t = 0$, which may be used to determine the initial condition for the slow dynamics.

Taking the limit $\varepsilon \to 0$ and defining

$$z = \lim_{\varepsilon \to 0} \frac{k(x_s)}{\varepsilon} \tag{2.5}$$

system (2.1) becomes,

$$\begin{aligned} \frac{dx_s}{dt} &= f(x_s) + g(x_s)u + b(x_s)z \\ k(x_s) &= 0 \end{aligned} \tag{2.6}$$

Note that in (2.6), z is indeterminate. Once the input $u(t)$ is specified (e.g. by a control law), it is possible to differentiate the algebraic constraint in Eq. (2.6) to obtain (after differentiating a sufficient number of times depending on the index number) a solution for the algebraic variable z. Without the loss of generality, it is assumed in this work that by one differentiation in time of the algebraic constraint, a solution of z can be obtained. Based on this assumption, it is obtained that:

$$z = -[L_b(k(x_s))]^{-1}[L_f(k(x_s)) + L_{g(x_s)}(k(x_s))u] \tag{2.7}$$

The matrix $L_b(k(x_s))$ denotes the Lie derivative of function $k(x_s)$ along $b(x_s)$ and is nonsingular. By substituting z in Eq. (2.6), we obtain an approximation of the slow dynamics of system (2.1):

$$\frac{dx_s}{dt} = f(x_s) + g(x_s)u + b(x_s)(-[L_b(k(x_s))]^{-1}[L_f(k(x_s)) + L_g(k(x_s))u]) := f_s(x_s, u) \tag{2.8a}$$

$$k(x_s) = 0 \tag{2.8b}$$

with an initial condition $x_{s0} = x_{fss}$. $f_s(x_s, u)$ is the analytic vector field of the dimension of n_{x_s}.

For more detailed procedures on decomposing fast and slow dynamics, the reader is referred to [77].

Reconstruction of the actual dynamics from fast and slow dynamics

In the preceding discussion, system (2.1) is brought down to a fast subsystem and a slow subsystem. The fast subsystem approximates the fast dynamics in the original system and the slow subsystem approximates the slow dynamics in the original system. We also see these explicit equations for fast and slow subsystems are each valid in their corresponding time-scale τ and t, respectively. An approximation of the actual dynamics of the original system can be constructed based on the fast and slow subsystems.

We use the idea of matching to find out the overlap region of fast and slow subsystems. The overlap region is the intermediate area where both fast and slow approximations should agree for identical values. To elaborate, let us consider the fast subsystem; this approximation dominates in a certain region of its domain. Similarly, the slow subsystem dominates in a specific but distinct area of approximation. However, there is a common region where the approximations overlap. The overlap value is x_{olp} which is the outer limit of the fast subsystem, or the inner limit of the slow subsystem. That is, $x_{olp} = \lim_{\tau\to\infty} x_f(\tau) = \lim_{t\to 0} x_s(t)$ [78, 79].

To obtain the final matched and composite solution, valid on the whole time domain, the uniform method is one of the popular methods. It adds the inner and outer approximations and subtracts their overlap value, x_{olp}, which would otherwise be counted twice. Basically, the overlap value is x_{fss} found from the limits mentioned above ($x_{olp} = x_{fss}$). Therefore, the final composite solution x_{cp} which is applicable

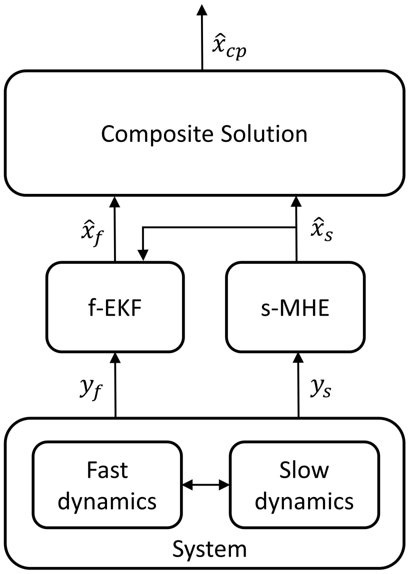


Figure 2.1: A schematic of the proposed distributed state estimation scheme

in the entire time t domain [78, 79].

$$x_{cp}(t) = x_f(\varepsilon\tau) + x_s(t) - x_{fss} \tag{2.9}$$

When the exact solution x for a singular perturbation problem Eq. (2.1) is not available, x_{cp} is an approximate solution of such system that remains uniformly valid in the independent variable t.

2.2 Proposed distributed state estimation scheme

In this section, we propose a distributed state estimation scheme to estimate the state of the two-time-scale system (2.1) based on fast and slow dynamics decomposition. A schematic of the proposed distributed state estimation scheme is presented in Fig. 2.1. A local estimator is designed for each fast subsystem and slow subsystem. Two different estimators are used: extended Kalman filter (EKF) is designed for the fast subsystem, and moving horizon estimation (MHE) is associated with the slow subsystem. The reason to use two different estimators is mainly the existing different time-scales. EKF takes less time to evaluate, which is apt for the fast subsystem, but it cannot take nonlinearity or constraints into account in a systematical way and may give poor performance. The use of EKF for the fast dynamics is a trade-off between computing speed and performance. On the other hand, MHE is more suitable for complex nonlinear and constrained dynamic systems. However, it requires online solutions of dynamic optimization problems, which results in increased computational cost. The sluggish nature of the slow subsystem and the necessary high accuracy in exchange for an increase in computational cost are reasonable enough for the consideration. The EKF and MHE are designed based on the reduced fast subsystem and the reduced slow subsystem derived in the previous section. Note that f-EKF denotes EKF for the fast subsystem, and s-MHE indicates MHE for the slow subsystem. The composite solution is computed using Eq. 2.9 incorporating estimates derived from f-EKF, s-MHE, and the fast system steady state. The composite solution undergoes frequent updates on the fast time-scale.

There is no information exchange from f-EKF to s-MHE. Since each subsystem evolves at different time-scales, it is desirable to use different sampling periods in the local estimator designs for the fast and slow subsystems. Therefore, the sampling period for f-EKF and s-MHE are defined as Δ_f and Δ_s respectively. Without loss of generality, we assume that Δ_s is an integer multiple of Δ_f, i.e., $\Delta_s = n\Delta_f$ where n is a positive integer. In the proposed design, we use $\tau_q := \tau_0 + q\Delta_f$ with $q \geq 0$ and $t_k := t_0 + k\Delta_s$ with $k \geq 0$ to denote the sampling instants of f-EKF and s-MHE, respectively. While $\hat{x}_f(\tau_q)$ denotes the state estimates of f-EKF at τ_q, $\hat{x}_s(t_k)$ is the state estimates of s-MHE at t_k. We denote $y_f(\tau_q)$ and $y_s(t_k)$ as the measurements for f-EKF and s-MHE sampled at τ_q and t_k, respectively. In the end, we find the estimation of the actual state based on composite solution Eq. (2.9). In the following discussion, we illustrate the proposed estimator design procedure that accounts rationally for the nonlinear two-time-scale dynamics.

2.2.1 Proposed implementation algorithms

We decompose two-time-scale systems described in Eq. (2.1) into two separate reduced subsystems evolving in a fast and a slow time-scales as illustrated in the previous section. The fast subsystem is described by Eq. (2.3) and the slow subsystem is described by Eq. (2.8).

It is important to note that the measurements used in the f-EKF and s-MHE are directly obtained from the actual system measurement y of Eq. (2.1b) but sampled every Δ_f and Δ_s respectively.

In the proposed scheme, the f-EKF and s-MHE are designed independently based on the above subsystems. The implementation details of the distributed state estimation are specified in Algorithm 1.

Algorithm 1 Proposed estimation algorithm

```
Initialization Initialize the f-EKF and s-MHE with their initial guesses. Find the steady state of the reduced fast subsystem model
for $q = 0, 1, 2, 3 \ldots$ do
    At $\tau_q$, receive measurement $y_f(q)$
    Evaluate the f-EKF to obtain $\hat{x}_f(\tau_q)$
    if $\frac{q}{n}$is an integer then
        Evaluate the s-MHE to obtain $\hat{x}_s(t_k)$ and send $\hat{x}_s(t_k)$ to the f-EKF
    else
        Obtain open-loop prediction $\hat{x}_s$ from the reduced slow subsystem model and send $\hat{x}_s(\tau_q)$ to the f-EKF
    end if
    Compute $\hat{x}_{cp}(t)$ at time instant $\tau_q$
end for
```

Note that in a typical distributed estimation scheme, bidirectional communication between subsystem estimators is in general used [75]. In the proposed approach, we take into account the specific feature of a two-time-scale system in the design, and only a one-directional communication (from s-MHE to f-EKF) is needed. This is a unique feature of the proposed design and is possible due to the time-scale separation between the fast and slow subsystems.

2.2.2 Design of f-EKF

In this section, we design an EKF estimator based on the reduced fast subsystem model to estimate the state of the fast subsystem. Specifically, in the design of the EKF, we consider the fast subsystem with additive process noise and the system output represented in terms of the fast and slow states based on (2.9) as follows:

$$\frac{dx_f}{d\tau} = f_f(x_f) + w_f \tag{2.10a}$$

$$y(\tau) = h(x_f + x_s - x_{fss}) + v \tag{2.10b}$$

Note that in (2.10a), w_f reflects the modeling error of the subsystem model. The modeling error may come from the system disturbance of the original system (w in (2.1a)) and the assumption of $\varepsilon = 0$ in deriving the fast subsystem. (2.10b) implies that in the design of the f-EKF, information of the slow subsystem state x_s and the steady state information of the fast subsystem x_{fss} is needed.

EKF is a common method used for state estimation of nonlinear systems based on successively linearizing the nonlinear system. It can be divided into two steps, which are prediction and update steps [23].

Prediction step. At a sampling time τ_{q-1}, $q = 1, 2, \ldots$, in an open-loop manner based on the fast subsystem model and the estimate of the fast subsystem state at τ_{q-1}, the f-EKF first predict the state at the next sampling time.

$$\hat{x}_f(\tau|\tau_{q-1}) = \hat{x}_f(\tau_{q-1}|\tau_{q-1}) + \int_{\tau_{q-1}}^{\tau_q} f_f(\hat{x}_f\,(\tau|\tau_{q-1}))\,d\tau \tag{2.11}$$

where $\hat{x}_f(\tau|\tau_{q-1})$ represents the prediction of the state at time instant $\tau \in (\tau, \tau_{q-1}]$. The propagation of the process disturbance is as follows:

$$\dot{P}_f(\tau|\tau_{q-1}) = F_f(\tau, \tau_{q-1})P_f(\tau_{q-1}|\tau_{q-1})F_f^T(\tau, \tau_{q-1}) + \int_{\tau_{q-1}}^{\tau} F_f(\tau, t)Q_f F_f^T(\tau, t)dt \tag{2.12}$$

where P_f and Q_f are the error covariance matrix and the state covariance matrix, respectively, $P(\tau_q|\tau_{q-1})$ is a square matrix containing the a *priori* estimation error covariance information, and $F_f(\tau, \tau_{q-1})$ denotes the state transition matrix of the time-varying linearized system matrix, $A_f(\tau|\tau_{q-1}) := \frac{\partial f_f}{\partial x_f}|_{(\hat{x}_f(\tau|\tau_{q-1}))}$ and can be calculated as follows:

$$\begin{aligned} &\frac{\partial F_f(\tau, \tau_{q-1})}{\partial \tau} = A_f(\tau|\tau_{q-1})F_f(\tau, \tau_{q-1}) \\ &\text{s.t. } F_f(\tau, \tau) = I \end{aligned}$$

for $\tau \in [\tau_{q-1}, \tau_q]$ with I being the identity matrix.

Update step. At each sampling instant τ_q, a state estimate of the actual dynamics of the fast subsystem (denoted as $\hat{x}_f(\tau_q|\tau_q)$) is obtained by performing the measurement-update step. $K(\tau_q)$ is the correction gain updated at τ_q which is used to minimize a *posteriori* error covariance based on the measurement innovation (i.e. $y_f(\tau_q) - Ch(\hat{x}_f(\tau_q|\tau_{q-1}) + \hat{x}_s - x_{fss})$).

$$K(\tau_q) = P_f(\tau_q|\tau_{q-1})H^T(\tau_q)(H(\tau_q)P_{q|q-1}H^T(\tau_q) + R_f)^{-1} \tag{2.13}$$

where $H(\tau_q) = \frac{\partial h}{\partial x_f}|_{(\hat{x}_f(\tau_q|\tau_{q-1}))}$ is the observation matrix, R_f is the covariance matrix of the measurement noise v_f. The updated state estimate is as follows:

$$\hat{x}_f(\tau_q|\tau_q) = \hat{x}_f(\tau_{q-1}|\tau_{q-1}) + K(\tau_q)\left(y_f(\tau_q) - h(\hat{x}_f(\tau_q|\tau_{q-1}) + \hat{x}_s - x_{fss})\right) \tag{2.14}$$

where $\hat{x}_f(\tau_q|\tau_q)$ represents the estimate of x_f at time τ_q given observations up to time τ_q. The updated state covariance is as follows:

$$P_f(\tau_q|\tau_q) = (I - K(\tau_q)H(\tau_q))P_f(\tau_q|\tau_{q-1}) \tag{2.15}$$

where $P(\tau_q|\tau_q)$ is the a *posteriori* error covariance matrix with respect to the estimation error at τ_q, and I is the identity matrix with dimension n_{x_f}. Note that in the above f-EKF design, R_f, Q_f, P_f are three tuning parameters.

Design of s-MHE

In this section, we design the s-MHE based on the reduced slow subsystem model to estimate the slow states. Similarly, we consider a stochastic version of the reduced slow subsystem model in Eq. (2.8) described as in the following form:

$$\frac{dx_s}{dt} = f(x_s) + g(x_s)u + b(x_s)(-[L_b(k(x_s))]^{-1}[L_f(k(x_s)) + L_g(k(x_s))u]) + w_s \tag{2.16a}$$

$$k(x_s) = 0 \tag{2.16b}$$

$$y(t) = h(x_s) + v \tag{2.16c}$$

In (2.16a), w_s accounts the modeling error of this subsystem model which may originate either from the actual system (w in (2.1a)) or the assumption for decomposition of slow subsystem. As stated earlier, when deriving the slow subsystem, it is assumed that the fast dynamics have converged to the corresponding steady state values. Based on the assumption and the expression shown in (2.9), the output equation (2.16) can be obtained for the slow subsystem.

MHE is an online optimization-based estimation method [80]. The proposed s-MHE optimization problem at time t_k is formulated as follows:

$$\min_{x_s(k-N),\hat{w}_s(\cdot)} ||\hat{x}_s(k-N) - \tilde{x}_s(k-N)||^2_{P_s^{-1}} + \sum_{j=k-N}^{k-1} ||\hat{w}_s(j)||^2_{Q_s^{-1}} + \sum_{j=k-N}^{k} ||\hat{v}(j)||^2_{R_s^{-1}} \tag{2.17a}$$

$$\text{s.t. } \hat{x}_s(j+1) = f_s(\hat{x}_s(j), u(j)) + \hat{w}_s(j), \quad j \in [k-N, k-1] \tag{2.17b}$$

$$\hat{v}(j) = y(j) - h(\hat{x}_s(j)), \quad j \in [k-N, k] \tag{2.17c}$$

$$\tilde{x}_s(k-N) = \hat{x}_s(k-N|k-N) \tag{2.17d}$$

$$x_s \in \mathbb{X}_s, \hat{w}_s \in \mathbb{W}, \hat{v} \in \mathbb{V} \tag{2.17e}$$

where $\hat{x}_s$ denotes the estimated value of the slow subsystem state x_s, $\hat{w}_s$ denotes the estimated system disturbance, $\hat{v}$ denotes the estimated measurement noise, and $\mathbb{X}_s$, $\mathbb{W}$, and $\mathbb{V}$ denote the known constraints on the augmented state, the system disturbance, and the measurement noise, respectively. Equation (2.17a) is the cost function the MHE tries to minimize. The objective of the s-MHE is to find the best estimates of the system states such that the model disturbance and measurement noise are minimized. P_s^{-1}, Q_s^{-1} and R_s^{-1} are positive definite weighting matrices which are tuning parameters. The arrival cost, $||\hat{x}_s - \tilde{x}_s||^2_{P_s^{-1}}$ summarizes the information from the initial state of the model up to the beginning of the estimation window of the MHE. N denotes the length of the estimation window. Equations (2.17b) and (2.17c) are the slow subsystem models with system disturbance and measurement noise considered. In Eq. (2.17d), $\hat{x}_s(k-N|k-N)$ represents the estimated state $\hat{x}_s$ at time instant $k-N$, which is estimated at time instant $k-N$. Equation (2.17e) is the known constraints or compact sets that bound the subsystem state, system disturbance, and measurement noise.

Remark 1 *Note that in this work, EKF is used to estimate the fast dynamics. The main consideration is that EKF is of low computational complexity and can be evaluated very fast, which makes it appropriate for the fast dynamics. Other estimator or observer designs of low computational complexity may also be used for the fast subsystem. Regarding EKF, it is widely used for systems with Gaussian noise. When the noise is bounded and is non-Gaussian, Gaussian mixture models may be used to improve the estimation performance as discussed in [81]. Note also that in the proposed distributed estimation scheme, the s-MHE sends the estimated slow state $\hat{x}_s$ to the fast subsystem every slow sampling time. The f-EKF uses $\hat{x}_s$ in the estimation of the fast state x_f. This mechanism may assist the f-EKF in getting an improved* *estimation performance. The reader may refer to [82] for relevant discussion on how the performance of EKF can be improved by re-initializing the EKF frequently using information from MHE.*

Remark 2 *The proposed fast and slow dynamics decomposition is general. While in this work EKF is used in the design of the fast estimator, other estimator or observer designs of low computational complexity may also be used. The idea is also applicable to other applications including process intensification [77]. The reader may refer to [13, 77] for more discussion on how time-scale separation may be used in different applications from a controller design perspective.*

Remark 3 *Note that in this work, the convergence and boundedness of the proposed distributed estimation scheme are not investigated. It is expected that when* ε *is close to 0, the slow dynamics are essentially independent on the fast dynamics. The convergence and boundedness of the s-MHE can be established following a standard centralized MHE design [83]. For the f-EKF, if the estimated slow state* $\hat{x}_s$ *transmitted from the s-MHE to f-EKF can approximate the actual slow state* x_s *accurately, the convergence and boundedness of EKF may also be established locally. However, before the convergence of s-MHE,* $\hat{x}_s$ *could be different from the actual slow state* x_s*. This mismatch between* $\hat{x}_s$ *and* x_s *may lead to divergence of the f-EKF. It is needed to establish sufficient conditions under which the f-EKF can still converge in the presence of the mismatch between* $\hat{x}_s$ *and* x_s*. Similar arguments as used in [75, 11] may be used to find the sufficient conditions. When both s-MHE and f-EKF can converge, the composite solution will also converge to the actual state. We will leave the detailed analysis of the convergence property to our future work.*

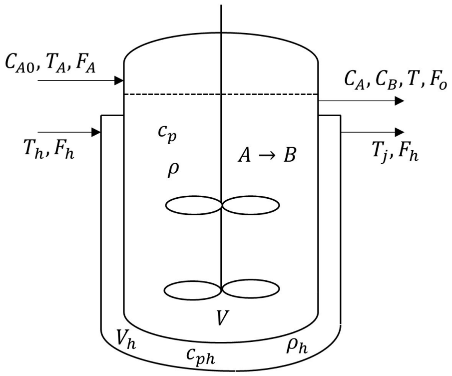


Figure 2.2: A continuous-stirred tank reactor with heating jacket

2.3 Application to a CSTR

2.3.1 Process description

Consider a continuous-stirred tank reactor (CSTR) with a heating jacket as shown in Figure 2.2. Reactant A is fed to the reactor at a flow rate F_A, initial molar concentration C_{A0} and temperature T_A. The reactant A is converted into the product B through the irreversible endothermic reaction $A \to B$, and the product stream is withdrawn at a flow rate $F_0 = F_A$. This implies that the reactor holdup volume V is constant. The reaction rate r_A, is given by the following Arrhenius expression:

$$r_A = k_0 \exp\left(\frac{-E}{RT}\right) C_A V \tag{2.18}$$

where k_0 and E are the reaction rate coefficient and activation energy, respectively, T is the reactor temperature, and C_A is the molar concentration of A in the reactor. Heat is provided to the reactor from the jacket, where a heating fluid is fed at a flow rate F_h and a temperature T_j. The modeling equations for the process include the mole balances for the two components in the reactor and the energy balances in the reactor and the jacket. The resulting dynamic model is as follows:

$$
\begin{aligned}
\dot{C}_A &= \frac{F_A}{V}(C_{A0} - C_A) - r_A \\
\dot{C}_B &= -\frac{F_A}{V}C_B + r_A \\
\dot{T} &= \frac{F_A}{V}(T_A - T) - r_A\frac{\Delta H_r}{\rho c_p} + \frac{UA}{\rho c_p}\left(\frac{T_j - T}{V}\right) \\
\dot{T}_j &= \frac{F_h}{V_h}(T_h - T) - \frac{UA}{\rho_h c_{ph}}\left(\frac{T_j - T}{V_h}\right)
\end{aligned}
\tag{2.19}
$$

where, c_p and c_{ph} are the specific heat capacities of the reaction mixture in the reactor and heating liquid in the jacket, respectively. Similarly, the density of the liquids of the reactor and jacket are ρ and ρ_h, respectively. U is the overall conductance or heat transfer coefficient, A is the heat transfer area of the contact surface between ractor and jacket, and ΔH_r is the heat of reaction which is the enthalpy of the reaction.

It is assumed that the densities and specific heat capacities of the two liquids are the same, i.e. $c_p = c_{ph}$ and $\rho = \rho_h$ and the liquid holdup in the jacket at a temperature T_j has a constant volume V_h. The heat transfer rate by convection thought the contact surface can be expressed as:

$$\dot{Q} = UA(T_A - T) \tag{2.20}$$

Furthermore, we assume that the heat transfer between the heating jacket and the reactor is fast compared to the reaction occurring in the reactor. The large difference in heat transfer and reaction in the model induces two-time-scale behavior in this dynamical system. The ratio between the heat transfer to the reaction is defined as:

$$\frac{1}{\varepsilon} = \frac{UA}{\rho c_p} \tag{2.21}$$

where ε is the small parameter or singular perturbation parameter which indicates the presence of fast and slow transients in time response of the system. Based on the definition of ε, the CSTR model (2.19) can be rewritten as follows:

$$\begin{aligned}
\dot{C}_A &= \frac{F_A}{V}(C_{A0} - C_A) - r_A \\
\dot{C}_B &= -\frac{F_A}{V}C_B + r_A \\
\dot{T} &= \frac{F_A}{V}(T_A - T) - r_A\frac{\Delta H_r}{\rho c_p} + \frac{1}{\varepsilon}\left(\frac{T_j - T}{V}\right) \\
\dot{T}_j &= \frac{F_h}{V_h}(T_h - T) - \frac{1}{\varepsilon}\left(\frac{T_j - T}{V_h}\right)
\end{aligned} \tag{2.22}$$

For this process, it is considered that the state vector is $x = [x_1,\ x_2,\ x_3,\ x_4]^T = [C_A,\ C_B,\ T,\ T_j]^T$, the manipulated input vector is $u = [u_1,\ u_2]^T = [F_A,\ F_h]^T$, and the controlled output vector is $[y_1,\ y_2]^T = [x_2,\ x_4]^T$. The model in (2.22) takes the form of Eq. (2.1a) with the system functions defined as:

$$f(x) = \begin{bmatrix} k_0 \exp\left(\frac{-E}{Rx_3}\right)x_1 \\ -k_0 \exp\left(\frac{-E}{Rx_3}\right)x_1 \\ -k_0 \exp\left(\frac{-E}{Rx_3}\right)x_1\frac{\Delta H_r}{\rho c_p} \\ 0 \end{bmatrix}, \quad g(x) = \begin{bmatrix} \frac{C_{A0} - x_1}{V} & 0 \\ \frac{-x_2}{V} & 0 \\ \frac{T_A - x_3}{V} & 0 \\ 0 & \frac{T_h - x_4}{V_h} \end{bmatrix}, \quad b(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{V} \\ -\frac{1}{V_h} \end{bmatrix},$$

$$k(x) = \begin{bmatrix} x_4 - x_3 \end{bmatrix}$$

2.3.2 Subsystem decomposition

Following the method described in the previous sections, the fast dynamics of the process can be obtained and the fast subsystem is shown below:

$$\begin{aligned}
\frac{dx_{3f}}{d\tau} &= \frac{x_{4f} - x_{3f}}{V} \\
\frac{dx_{4f}}{d\tau} &= -\frac{x_{4f} - x_{3f}}{V_h}
\end{aligned} \tag{2.23}$$

We multiply Eq. (2.22) by ε and consider the limit of an infinitely high heat transfer rate compared to reaction ($\varepsilon \to 0$) in the original time-scale t. In this limiting case, the heat transfer resistance becomes negligible, and the reactor and jacket approach thermal equilibrium. The heat transfer rate $\dot{Q}$ is driven by the thermal equilibrium condition $x_{4s} \to x_{3s}$ instead of the explicit heat transfer correlation. We obtain the constraint $k(x_s)$ which is the linearly independent constraint as follows:

$$x_{4s} - x_{3s} = 0 \tag{2.24}$$

This constraint must be satisfied in the slow time-scale. Also in the limit ($\varepsilon \to 0$), the term $\frac{(x_{4s}-x_{3s})}{\varepsilon}$, which implies that the heat transfer present in the energy balance equations become indeterminate. Therefore, $z = \lim_{\varepsilon \to 0} \frac{(x_{4s}-x_{3s})}{\varepsilon}$ is defined as the algebraic variable – the finite but unknown term. Therefore, the slow dynamics become:

$$\begin{aligned}
\dot{x}_{1s} &= \frac{F_A}{V}(C_{A0} - x_{1s}) - k_0 \exp\left(\frac{-E}{Rx_{3s}}\right) x_{1s} V \\
\dot{x}_{2s} &= -\frac{F_A}{V} x_{2s} + k_0 \exp\left(\frac{-E}{Rx_{3s}}\right) x_{1s} V \\
\dot{x_{3s}} &= \frac{F_A}{V}(T_A - x_{3s}) - k_0 \exp\left(\frac{-E}{Rx_{3s}}\right) x_{1s} V \frac{\Delta H_r}{\rho c_p} + \frac{z}{V} \\
\dot{x_{4s}} &= \frac{F_h}{V_h}(T_h - x_{4s}) - \frac{z}{V_h}
\end{aligned} \tag{2.25}$$

which represents the model of the slow dynamics of the process. The variable z can be obtained after just one differentiation of the algebraic constraint of Eq. (2.24). The values of the process parameters and variables at the nominal steady state are given in Table 2.1. Corresponding to the parameter values shown in Table 2.1, the process has a steady state $[C_A,\ C_B,\ T,\ T_j]^T = [1.205\ \text{mol/l},\ 1.295\,\text{mol/l},\ 302.3\,\text{K},\ 302.6\,\text{K}]^T$ when the input vector of the system is $[F_A,\ F_h]^T = [2.0\,\text{l/s},\ 0.1\,\text{l/s}]^T$. It was verified that these values correspond to a stable steady state of the system (2.22).

Table 2.1: Nominal values of parameters of the CSTR

$C_{A0} = 2.5$ mol/l	$c_p = c_{ph} = 8.0$ J/g K	$\rho = \rho_h = 800$ g/l
$k_0 = 5 \times 10^{10}\ s^{-1}$	$E = 60,000$ J/mol K	$\varepsilon = \frac{\rho c_p}{UA} = 0.1$ s/l
$T_A = 305$ K	$T_h = 330$ K	$\Delta H_r = 20,000$ J/mol
$V = 1.0$ l	$V_h = 0.0494$ l	

2.3.3 f-EKF and s-MHE designs

In this section, we take advantage of the configured subsystem equations and implement the proposed distributed state estimation scheme for the CSTR. Within the proposed distributed framework, two local estimators are designed for the two subsystems. An f-EKF is designed for the fast subsystem, while s-MHE is developed for the slow subsystem. The estimates from the two estimators are used to reconstruct the actual state estimate.

It is assumed that C_B and T_j are the measured outputs of the system. The objective is to estimate the entire state vector of the system based on the two outputs. In the simulations, random process noise is generated following a normal distribution with zero mean and standard deviation 0.1. Similarly, the random measurement noise is considered to be Gaussian white noise with mean zero and standard deviation 0.001 for both measurements. The weighting matrices are diagonal matrices such as $Q_f = 10^{-2}\text{diag}([1,\ 1])$, $R_f = 10^{-6}\text{diag}([1,\ 1])$ and $P_f = 10^{-8}\text{diag}([1,\ 1])$ for the f-EKF. The f-EKF is evaluated at a fast sampling time $\Delta_f = 0.01$ s. For the design of s-MHE, the weighting matrices for s-MHE are $Q_s = 10^{-2}\text{diag}([1,\ 1,\ 1,\ 1])$, $R_s = 10^{-6}\text{diag}([1,\ 1])$ and $P_s = 10^{-8}\text{diag}([1,\ 1,\ 1,\ 1])$. The s-MHE is evaluated at a slower sampling time $\Delta_s = 0.1$ s. In the s-MHE, the arrival cost is approximated using EKF following the approach described in [83]. After testing different values for the estimation window size of the s-MHE, it was found that an estimation window size of 3 gives sufficient estimation performance.

In the following simulations, a couple of indexes are used to evaluate the performance of the estimators. The average relative standard deviation σ_{x_i} is defined as

$$\sigma_{x_i} = \sqrt{\frac{1}{N_{sim}} \sum_{j=0}^{N_{sim}} \left(\frac{\hat{x}_i(t_j) - x_i(t_j)}{x_i(t_j)} \right)^2} \tag{2.26}$$

where N_{sim} indicates the total simulation steps, $\hat{x}_i$ denotes the estimated value, and x_i denotes the actual value of the i^{th} state for $i = 1, 2, ...4$. Another performance index is the average root-mean-square error (RMSE) over the time period

$$\text{RMSE} = \frac{1}{N_{sim}} \sum_{j=0}^{N_{sim}} \sqrt{\frac{1}{4} \sum_{i=1}^{4} \left(\frac{\hat{x}_i(t_j) - x_i(t_j)}{x_i(t_j)} \right)^2} \tag{2.27}$$

These performance indexes are expressed in percentages to evaluate the performance of each scheme. All the simulations were conducted in a desktop computer with an Intel i7 CPU at 3.2 GHz and 16 GB RAM.

2.3.4 Simulation results

First, the effectiveness of the decomposition is investigated by comparing the composite state trajectories with the actual system state trajectories. Figure 2.3 shows one set of the trajectories. The actual system trajectory is solved using an initial condition [2.5, 0.0, 305, 330] for the states C_A, C_B, T, and T_j respectively. For the composite solution, the fast subsystem is integrated with an initial condition [305, 330] for fast states T, and T_j respectively, and the corresponding steady state solution is found to be [309.167, 309.167]. Then, the slow subsystem is solved where the initial condition is [2.5, 0.0, 309.167, 309.167] for C_A, C_B, T, and T_j, respectively. Then, according to Eq.(2.9), the composite solution is evaluated.

It can be seen from Figure 2.3 that the proposed decomposition method is able to track the actual state trajectories very well. The state approximations using the composite solution are very close to the true value obtained from actual model

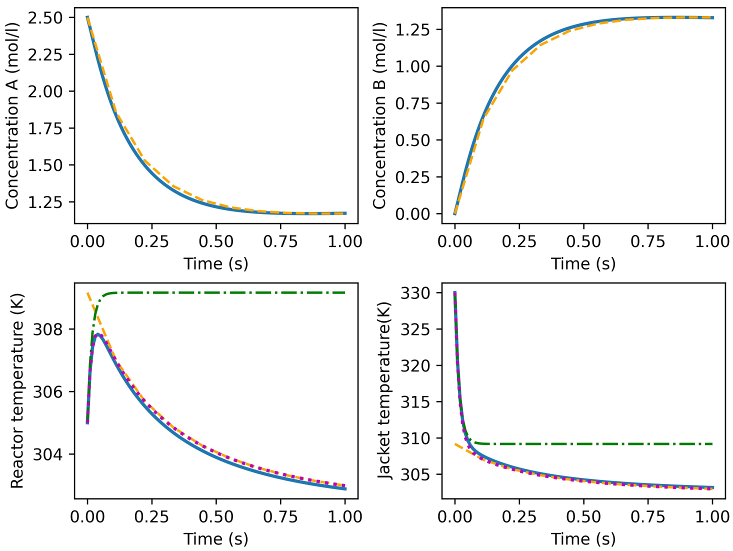


Figure 2.3: Trajectories of the actual process states (blue solid lines), fast subsystem (green dot-dashed lines) slow subsystem (orange dashed lines) and composite solution (red dotted lines)(For interpretation of the references to color in this figure legend, the reader may refer to the web version of this article.)

equations. It can be observed clearly that the concentration trajectories are smooth and have a flat slope but the temperature trends have a very steep slope initially, and then flatten gradually. This indicates that the temperature dynamics exhibit two-time-scale behaviors. The results also show that there is no explicit separation of the fast and the slow dynamics in the system. The composite states based on the fast and slow subsystems and the actual system trajectories do not match 100% because $\varepsilon = 0.1$ instead of 0. The composite solution approaches the actual system states when $\varepsilon = 0$. It should be pointed out that in the case considered ($\varepsilon = 0.1$), the mismatch between the composite solution and the actual model in terms of RMSE is 0.035% which is negligible and shows the decomposed subsystems provide an accurate approximation of the actual system. It was also observed in our simulations that a stable numerical solution of the original nonlinear model cannot be obtained using

Table 2.2: Initial states of the process and the initial guesses used in different estimation schemes.

states	Initial condition	Centralised MHE	f-EKF	s-MHE
C_A (mol/l)	2.5	1.5	N/A	1.5
C_B (mol/l)	0.0	0.0001	N/A	0.0001
T (K)	306	308	308	308
T_j (K)	311	313	313	313

the explicit fourth-order Runge Kutta method (RK4) if the integration step size is greater than 0.04 s. This is because the actual system is a stiff system existing in both fast and slow dynamics and it lacks numerical stability beyond integration step size 0.04 s when a fourth-order integration method like RK4 is used.

Next, we apply the proposed distributed state estimation scheme to the CSTR and compare its performance against two other common schemes. Specifically, we consider three different schemes: (I) the proposed distributed scheme; (II) a decentralized estimation scheme; and (III) a centralized MHE. In the decentralized scheme, the s-MHE and f-EKF in the distributed scheme are still used but they do not communicate. That is, in the decentralized scheme, the s-MHE does not send any information to the f-EKF. The tuning parameters of the decentralized scheme are kept the same as the proposed distributed scheme. The centralized MHE is designed based on the original full nonlinear system of Eq. (2.19). The weighting matrices for the centralized MHE are $Q = 10^{-2}\text{diag}([1,\ 1,\ 1,\ 1])$ and $R = 10^{-6}\text{diag}([1,\ 1])$ and $P = 10^{-8}\text{diag}([1,\ 1,\ 1,\ 1])$. For the centralized scheme, the fast sampling time $\Delta_f = 0.01$ s is used to ensure that both the fast and slow dynamics can be captured accurately. For the centralized MHE, we consider the estimation window is the same as the s-MHE; that is $N = 3$. In such a case, the centralized MHE keeps the same number of decision variables as the s-MHE. In all these designs, the arrival cost is included and is approximated using the EKF-based approach. For all these schemes, the system disturbance and measurement noise are the same for proper comparison.

Table 2.3: Estimation performance of the three estimation schemes

σ_{x_i}	Scheme I	Scheme II	Scheme III
$\sigma_{C_A}(\%)$	12.7	12.7	5.7
$\sigma_{C_B}(\%)$	7.15	7.15	8.42
$\sigma_T(\%)$	0.19	0.55	6.43×10^{-2}
$\sigma_{T_j}(\%)$	2.058×10^{-3}	0.667	7.192×10^{-2}
Average RMSE (%)	2.88	3.26	2.2

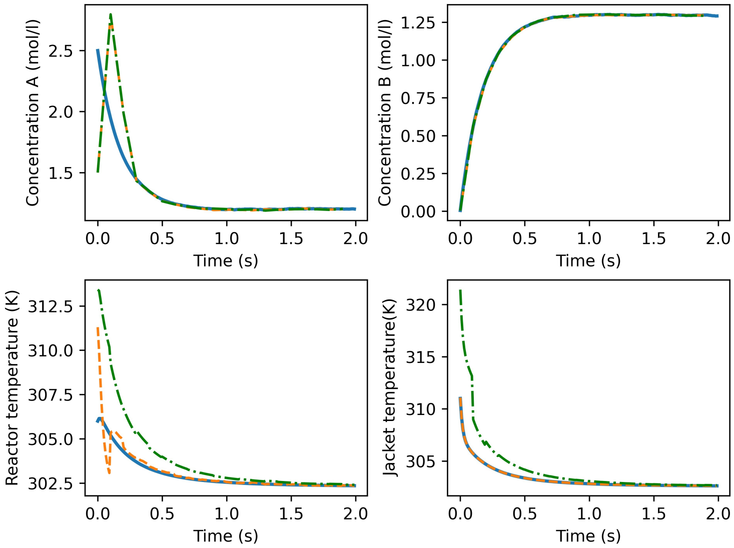


Figure 2.4: Trajectories of the actual states (blue solid line), state estimates based on decomposition under proposed distributed configuration (orange dashed lines), state estimates based on decomposition under decentralized configuration (green dash-dotted lines) (For interpretation of the references to color in this figure legend, the reader may refer to the web version of this article.)

Table 2.3 summarizes the simulation results in terms of the performance measures introduced earlier. Figure 2.4 shows the estimated and the actual state trajectories of scheme I and scheme II. From Figure 2.4, we observe that the estimated states are able to converge to the actual state trajectories in both schemes. From Table 2.3, it

Table 2.4: Average RMSE of the three estimation schemes under different noise levels

	Scheme I	Scheme II	Scheme III
$\sigma_w = 0.001$ $\sigma_v = 0.001$	2.58	2.62	1.75
$\sigma_w = 0.01$ $\sigma_v = 0.01$	3.19	3.22	5.4
$\sigma_w = 0.5$ $\sigma_v = 0.01$	3.58	4.06	5.88
$\sigma_w = 0.9$ $\sigma_v = 0.05$	6.7	7.61	22.93
$\sigma_w = 0.9$ $\sigma_v = 0.1$	10.7	11.6	44.61

can be seen that the proposed scheme I give better estimation performance. Since the s-MHE are the same in both schemes and it is decoupled from the f-EKF, the estimation indexes for the two concentrations σ_{C_A} and σ_{C_B} are the same. Scheme I outperforms scheme II mainly in the estimation of T and T_j. The performance improvement in scheme I is essentially from the information exchange in the proposed distributed scheme. The information exchange in the proposed scheme I leads to faster convergence of the estimated states to the actual states compared with scheme II. Therefore, it can be more favorable to take scheme I for state estimation for the decomposition considered.

Figure 2.5 shows the results of scheme I and scheme III. The state estimators are able to track the actual state trajectories in these two schemes, too. It can be seen from Table 2.3 that the centralized MHE (scheme III) gives improved estimation performance compared with the proposed distributed scheme I in this set of simulations. However, the centralized scheme III is much more computationally intensive. For the distributed or decentralized schemes, the s-MHE takes about 0.032 sec for each evaluation and it evaluates every 0.1 sec. For the centralized MHE, it takes about 0.024 sec for each evaluation, which is a bit smaller than the one for the s-MHE due to the small sampling time used in the centralized MHE but it needs to be evaluated 10 times every 0.01 sec. This indeed makes the centralized MHE much more computationally demanding. Further, the performance of the centralized MHE

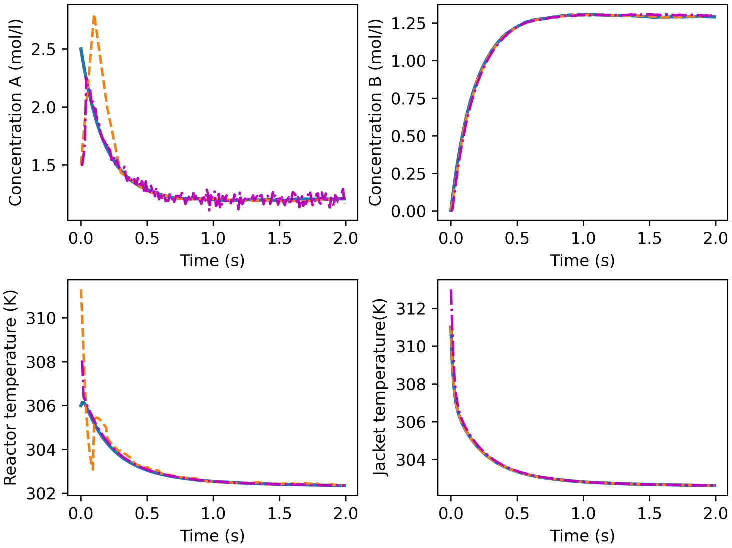


Figure 2.5: Trajectories of the actual states (blue solid line), state estimates based on decomposition under distributed configuration (orange dashed lines), state estimates of actual system without decomposition (maroon dash-dotted line) (For interpretation of the references to color in this figure legend, the reader may refer to the web version of this article.)

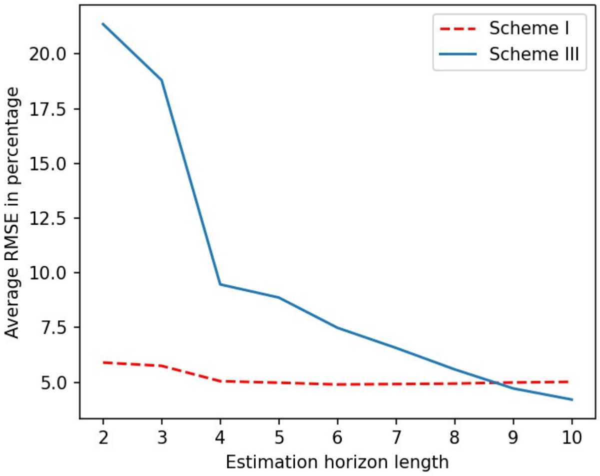


Figure 2.6: Average RMSE of the proposed distributed scheme I and the centralized scheme III when different estimation window sizes are considered.

is sensitive to the level of noise present in the system and deteriorates quickly as the noise level increases as shown in the next set of simulations.

In this set of simulations, we run the three estimation schemes with different initial conditions and different noise levels. Table 2.4 summarizes the results. In Table 2.4, each data point is calculated based on three simulation runs with different initial conditions and different noise sequences. From Table 2.4, we see that the performance of the three schemes overall decreases as the noise level increases. However, the performance of the centralized MHE deteriorates quickly as the noise level increases. When the noise level is low, the centralized MHE gives the best estimation performance but as the noise level increases, the centralized MHE gives the poorest performance. This may be due to the centralized model (which is stiff) used in the centralized MHE. The stiff model makes the centralized MHE sensitive to uncertainty and is less robust. This shows the advantage of decomposing the dynamics into fast and slow subsystems as in the proposed distributed scheme. From Table 2.4, it can be seen that the proposed distributed scheme (scheme I) outperforms the decentralized scheme (scheme II) in all the noise levels. This again demonstrates the benefits of information exchange between the s-MHE and the f-EKF in the proposed distributed scheme.

Note that in the above simulations, the centralized MHE and the s-MHE both used an estimation window size of 3. The robustness of the centralized MHE may be improved by using a longer horizon. To study the dependence of the s-MHE and the centralized MHE on the length of the estimation window, we perform another set of simulations. Figure 2.6 shows the average RMSE of the two schemes under different window lengths. Each RMSE value is the average value calculated based on simulations with different noise levels as shown in Table 2.4. From the figure, it can be seen that the performance of the proposed distributed scheme has less dependence on the size of the estimation window. The centralized MHE (scheme III) may achieve a similar performance as the proposed scheme I with $N = 3$ when the estimation window is about 8. However, when $N = 8$, the evaluation time of the centralized MHE is about 0.057 s, which is much larger than the evaluation time (0.032 s) of the s-MHE. Given the fact that the centralized MHE needs to be evaluated 10 times every 0.1 sec, it leads to a much higher computational load. This further demonstrates the proposed decomposition and distributed estimation scheme in obtaining a balanced performance in terms of estimation performance and computational complexity.

2.4 Summary

In this chapter, we developed a distributed state estimation method based on EKF and MHE for a class of implicit two-time-scale nonlinear systems, where some of the state variables inhibit both the fast and slow dynamics. The nonlinear system was decomposed into fast and slow subsystems based on singular perturbed parameter ε. In the proposed design, a one-directional communication strategy was established and the method was applied to a chemical process. A series of simulations were carried out to compare the proposed architecture with centralized and decentralized techniques from a computational time and accuracy point of view. Owing to different sampling times, the system is a stiff problem where step size plays an important role in the numerical stability of the solution instead of accuracy requirements. However, our design ensures numerical stability, moderate accuracy, and low computational time.

The potential challenges in implementing the proposed approach may include (a) the determination of the small parameter ε for a specific process, which may not be straightforward for some applications, and (b) the solution of the indeterminate variable z since it involves the calculation of Lie derivatives, which could be computationally expensive for large-scale processes. Future work aiming at addressing these challenges should be conducted.

Chapter 3

Input-output selection for LSTM-based reduced-order state estimator design

In this chapter, we consider a sensitivity-based approach to construct reduced-order state estimators based on recurrent neural networks (RNN). It is assumed that a mechanistic model is available but is too computationally complex for estimator design and that only some target outputs are of interest and should be estimated. A reduced-order estimator that can estimate the target outputs is sufficient to address such a problem. We propose an approach to find the most appropriate inputs and outputs for data-driven reduced-order model development for target variable estimation purposes. Specifically, we assume that a mechanistic model of the actual system is available and we are only interested in estimating a small set of the desired outputs instead of the entire state vector. To address such a problem, a reduced-order estimator that can estimate the desired outputs is sufficient to meet the requirements. In the proposed approach, there are three steps. In the first step, the sensitivity matrix of the target outputs to the initial state is evaluated based on process data. Then the singular value decomposition (SVD) is applied to the matrix to find the dominant singular values and the most important state elements that contribute to the dominant singular values. The most important state elements are selected as the reduced state vector. Once the reduced state vector is determined, the sensitivity of the inputs to the reduced state vector is evaluated and similarly, the most important inputs are determined. These inputs are selected as the elements in the reduced input vector. In the second step, process data are collected based on simulating the process model, and a data-driven model in the form of a Long-Short-Term-Memory (LSTM) neural network is designed to approximate the dynamics between the selected reduced input vector and the reduced state vector. In the last step, an extended Kalman filter is designed based on the reduced-order LSTM model to estimate the target output. The proposed approach is applied to a chemical process and extensive simulations will be performed to show its applicability and effectiveness.

3.1 Introduction

3.2 Preliminaries

In this section, we first provide a description of the discrete-time nonlinear system and define the objectives of the work. This section also discusses the formulation of the reduced-order model briefly.

3.2.1 System description

We consider a class of discrete-time nonlinear systems described as follows:

$$x(t+1) = f(x(t), u(t)) \tag{3.1a}$$

$$y(t) = h(x(t)) \tag{3.1b}$$

$$y_t(t) = h_t(x(t)) \tag{3.1c}$$

where $x(t) \in R^{n_x}$ is the vector of state variables at t, $u(t) \in R^{n_u}$ is the vector of the manipulated inputs, and $y(t) \in R^{n_y}$ is the vector containing all the measured outputs. $f(\cdot)$ and $h(\cdot)$ denote the nonlinear state and measured output equations, respectively. $y_t(t) \in R^{n_{y_t}}$ represents the vector of target process variables which is to be estimated, and the function $h_t(\cdot)$ characterizes the relation between the state x and the target output y_t. It is assumed that the state vector x is observable based on the measurements of y. It is also assumed that the dimension of y_t is smaller than the dimension of x (the number of target outputs n_{y_t} is smaller than the number of states n_x). In a process system, the number of states can be many but there is typically a relatively much smaller number of important states that need to be monitored very closely. The above assumption is considered from a viewpoint of the operation. It is not a condition but a scenario of the operation. For convenience, we will refer to the system in (3.1) as the actual system in the remainder of this work.

3.2.2 Problem formulation

The main objective of this work is to develop a reduced-order estimator to estimate the target output y_t based on measurements of y. Aiming at the y_t estimation, one can develop a full-order estimator using the available measurements y and the process model (3.1a). Different estimation algorithms can be readily applicable to estimate all the states x of the system (3.1). However, in many applications, it is not necessary to estimate all the states of the system when we are only interested in y_t. A full-order estimation can be computationally expensive, and the estimation performance may also be compromised due to estimating all the states with limited number of measured output variables. It is expected that a reduced-order estimator that only estimates a smaller subset of the state variables that are closely related to the target outputs is sufficient to achieve the objective. Compared to a full-order state estimator, a reduced-order estimator has the potential to decrease the computational cost and improve the estimation performance.

In this work, we consider the development of a reduced-order model for the system (3.1) and the associated reduced-order estimator design. Specifically, we propose to first use the full-order system model (3.1) to generate data, and then to identify a reduced-order model using RNN for y_t estimation purposes. Subsequently, a reduced-order estimator is designed based on the RNN to estimate y_t based on the measurements of y. In particular, we are interested in identifying a reduced-order model in the following form:

$$\tilde{x}(t+1) = \tilde{f}(\tilde{x}(t), \cdots, \tilde{x}(t-n_l), \tilde{u}(t), \cdots, \tilde{u}(t-n_l)) \tag{3.2}$$

where $\tilde{x} \in R^{\tilde{n}_x}$ is the vector of reduced (selected) state variables, n_l is the length of the sequence of data, $\tilde{u} \in R^{\tilde{n}_u}$ is the vector of the reduced (selected) manipulated inputs, and $\tilde{f}(\cdot)$ describes the dynamics of the reduced-order model. It is also expected that the elements of the reduced variables $\tilde{x}$ and $\tilde{u}$ are the same as the respective elements in the actual system variables x and u. The measured output y and the target output y_t are also expected to be able to be described using the reduced state $\tilde{x}$. Let us denote the relations as follows:

$$y(t) = \tilde{h}(\tilde{x}(t)) \tag{3.3}$$

$$y_t(t) = \tilde{h}_t(\tilde{x}(t)) \tag{3.4}$$

where $\tilde{h}$ and $\tilde{h}_t$ define the measured output equation and target output equation with respect to the reduced state vector $\tilde{x}$.

We will discuss how $\tilde{x}$ and $\tilde{u}$ should be selected so that a reduced model as shown in (3.2)-(3.4) can be identified and the target output y_t can be accurately estimated using the measurements of y based on the reduced-order model (3.2). It is expected that the dimension $\tilde{n}_x$ of the reduced state vector $\tilde{x}$ is much smaller than the dimension n_x of the actual system state vector x ($\tilde{n}_x < n_x$). Similarly, $\tilde{n}_u < n_u$.

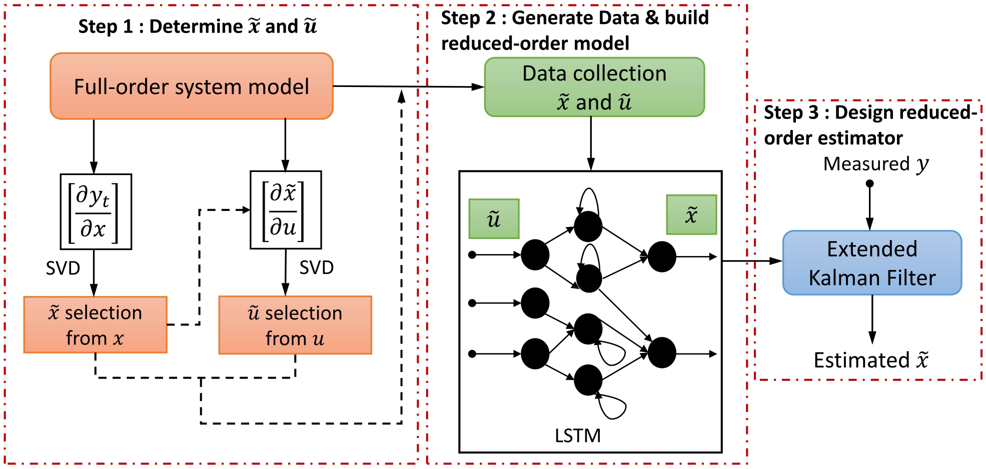


Figure 3.1: The flow chart of the proposed approach.

3.3 Proposed reduced input and state vectors selection approach

Figure 3.1 shows a flow chart of the proposed approach. In the first step, we select the reduced-order state $\tilde{x}$ and the corresponding $\tilde{u}$ for the purpose of estimating y_t. Based on the full-order system model, we first construct the sensitivity matrix $\frac{\partial y_t}{\partial x}$ and normalized. By analyzing $\frac{\partial y_t}{\partial x}$, we can then determine the elements in the state vector x that are most closely related to y_t, and these elements are selected to construct the reduced-order system state $\tilde{x}$. Once $\tilde{x}$ is determined, we further construct the sensitivity matrix $\frac{\partial \tilde{x}}{\partial u}$ and normalized [84, 85]. Based on $\frac{\partial \tilde{x}}{\partial u}$, we then find a subset of u, $\tilde{u}$, that has a significant impact on $\tilde{x}$. In the determination of $\tilde{x}$ and $\tilde{u}$, the singular value decomposition (SVD) method will be used. In the second step, based on the full-order system, we vary $\tilde{u}$ and collect the trajectories of $\tilde{x}$. Then, based on the data of $\tilde{u}$ and $\tilde{x}$, an LSTM model is identified. This LSTM model is a reduced-order model. In the third step, an estimator is designed based on the reduced-order model identified in the second step. In this work, we will show how an extended Kalman filter (EKF) may be designed based on the reduced-order model to estimate y_t based on measurements of y.

3.3.1 Sensitivity matrix for reduced state selection

In order to select the reduced states, the sensitivity matrix of the target output y_t to state x is considered. A larger element in the sensitivity matrix indicates that the target output is more sensitive to the corresponding state. That is, a small perturbation of the states can generate a larger change in the target output. In the literature, sensitivity matrices have been used to identify relevant connections between model outputs and inputs to develop reduced-order model [86, 87, 88] since the sensitivity is closely related to both observability and controllability of a system[71, 89]. To find the sensitivity matrix, a practical method is to linearize the nonlinear system at different points along its trajectories and find the observability matrix at each point.

Consider $N+1$ sampling points from t_0 to t_N along a trajectory of the system (3.1). Defining $A(t) := \frac{\partial f}{\partial x}(t)$, $B(t) := \frac{\partial f}{\partial u}(t)$, $C_1(t) := \frac{\partial h}{\partial x}(t)$, and $C_2(t) := \frac{\partial h_t}{\partial x}(t)$, the linearized system at a sampling time t can be obtained as follows:

$$x(t+1) = A(t)x(t) + B(t)u(t) + F^{'} \tag{3.5}$$

$$y(t) = C_1(t)x(t) + H_1^{'} \tag{3.6}$$

$$y_t(t) = C_2(t)x(t) + H_2^{'} \tag{3.7}$$

where $F^{'}$, $H_1^{'}$, and $H_2^{'}$ are additional constant terms resulting from the linearization at sampling point $(x(t), u(t))$.

The sensitivity of the target output $y_t(t)$ to the initial state $x(t_0)$ is defined as $\frac{\partial y_t(t)}{\partial x(t_0)}$. Defining the sensitivity of the state to the initial condition as $\frac{\partial x(t)}{\partial x(t_0)}$, from (3.5) and (3.7) the following two equations can be written:

$$\frac{\partial x(t+1)}{\partial x(t_0)} = A(t)\frac{\partial x(t)}{\partial x(t_0)} \tag{3.8}$$

$$\frac{\partial y_t(t)}{\partial x(t_0)} = C_2(t)\frac{\partial x(t)}{\partial x(t_0)} \tag{3.9}$$

with the initial value at $t = t_0$, $\dfrac{\partial x(t_0)}{\partial x(t_0)} = I$. Using (3.8) and (3.9), it can be rewritten at the sampling point t as:

$$\frac{\partial y_t(t)}{\partial x(t_0)} = C_2(t)A(t-1)A(t-2)\cdots A(0) \tag{3.10}$$

From sampling time t_0 to t_N, we can construct the sensitivities, $\dfrac{\partial y_t(t)}{\partial x(t_0)}$, $t = 0, \ldots, N$, and stack them to form a sensitivity matrix S_O:

$$S_O(t_0, \ldots, t_N) = \begin{bmatrix} \frac{\partial y_t(t_0)}{\partial x_1(t_0)} & \frac{\partial y_t(t_0)}{\partial x_2(t_0)} \cdots \frac{\partial y_t(t_0)}{\partial x_{n_x}(t_0)} \\ \\ \frac{\partial y_t(t_1)}{\partial x_1(t_0)} & \frac{\partial y_t(t_1)}{\partial x_2(t_0)} \cdots \frac{\partial y_t(t_1)}{\partial x_{n_x}(t_0)} \\ & \vdots \\ & \vdots \\ \frac{\partial y_t(t_N)}{\partial x_1(t_0)} & \frac{\partial y_t(t_N)}{\partial x_2(t_0)} \cdots \frac{\partial y_t(t_N)}{\partial x_{n_x}(t_0)} \end{bmatrix} \tag{3.11}$$

The matrix is a series of snapshots of the sensitivities stacked vertically in the time span t_0 to t_N. We can test the rank of $S_O(t_0 \ldots t_N)$ along a typical trajectory from t_0 to t_N. The sensitivity matrix S_O will be used to select the reduced state $\tilde{x}$. The SVD analysis will be used to achieve this goal and will be discussed in Section 3.3.2.

3.3.2 Reduced state selection via singular value decomposition

Once the sensitivity matrix S_O is constructed, we propose to use the singular value decomposition (SVD) analysis to analyze S_O to find the subset of state elements in x that are closely related to the target output y_t. The SVD algorithm [90] provides a way to represent the sensitivity matrix S_O as a summation of equally sized matrices that decrease in dominance:

$$S_O = u_1\sigma_1 v_1^T + u_2\sigma_2 v_2^T + \ldots + u_{n_x}\sigma_{n_x} v_{n_x}^T \tag{3.12}$$

where σ_i's ($i = 1, \ldots, n_x$) are the singular values of the matrix S_O sorted such that σ_1 is the largest singular value and σ_{n_x} is the smallest singular value, and u_i, v_i ($i = 1, \ldots, n_x$) are the associated left and right unitary matrices, respectively. Equation (3.12) implies that the sensitivity information contained in S_O can be projected onto n_x directions represented by the n_x singular vectors v_i, $i = 1, \ldots, n_x$. The magnitude of the singular value reflects the amount of information contained in the associated direction.

After SVD, we first examine the singular values and can select the dominant ones that contain most of the information in S_O for further analysis. One way to select the dominant singular values is to identify a significant gap in the singular values [91, 92].

Suppose that m out of the n_x singular values are selected for further analysis. We next analyze the associated singular vectors v_i, $i = 1, \ldots, m$. The absolute value of the j-th, $j = 1, \ldots, n_x$, element in v_i, denoted as v_{ij}, reflects the contribution of the j-th state element of x to the variance/information of the target variable y_t along the direction of v_i [93]. Therefore, in this work, the following measure is used to reflect the overall effect of the j-th state element on the target output:

$$D_j = \frac{\sum\limits_{i=0}^{m} |\sigma_i v_{ij}|}{\sum\limits_{i=0}^{m} |\sigma_i|} \tag{3.13}$$

where $0 \leq D_j \leq 1$. A large value of D_j implies a larger impact on the target output. Based on the above measure, a partitioning of the elements of x into two groups can be performed. The first group has a comparatively large effect on the target variable than the other group. Therefore, the elements belonging to the first group are selected to be elements in $\tilde{x}$. After the above process, $\tilde{x}$ contains elements that are most closely related to y_t. In order to be able to use the measured output y, we should check whether y can be expressed using the elements in the current $\tilde{x}$. This is, whether an expression in (3.3) can be obtained. If the current $\tilde{x}$ is not sufficient to express y, the missing state elements should be added to $\tilde{x}$.

3.3.3 Sensitivity matrix for input selection and reduced input vector selection

After $\tilde{x}$ is determined, we can proceed to select the reduced input vector $\tilde{u}$. The sensitivity of $\tilde{x}$ with respect to the actual input vector u, $\frac{\partial \tilde{x}}{\partial u}$ will be used.

Considering again the sampling points from t_0 to t_N along a trajectory of the system in (3.1a) and (3.1b), from equation (3.5), it can be written the following equation:

$$\frac{\partial x(t+1)}{\partial u(t_0)} = A(t)\frac{\partial x(t)}{\partial u(t_0)} + B(t)\frac{\partial u(t)}{\partial u(t_0)} \tag{3.14}$$

where $\frac{\partial x(t)}{\partial u(t_0)}$ is the sensitivity of the state vector $x(t)$ with respect to the input $u(t_0)$ and $\frac{\partial u(t)}{\partial u(t_0)}$ is the sensitivity of input vector u at t with respect to the input at t_0. As the inputs are not dependent on each other, $\frac{\partial u(t)}{\partial u(t_0)} = 0$ for all the t except t_0. At $t = t_0$, $\frac{\partial u(t)}{\partial u(t_0)} = I$. From (3.14), $\frac{\partial x(t)}{\partial u(t_0)}$ is evaluated as follows:

$$\frac{\partial x(t)}{\partial u(t_0)} = A(t-1)A(t-2)\cdots A(1)B(0) \tag{3.15}$$

Note that the sensitivity of the reduced state vector $\tilde{x}$ to the input vector u, $\frac{\partial \tilde{x}(t)}{\partial u(t_0)}$, can be obtained by taking the corresponding elements (rows) from $\frac{\partial x(t)}{\partial u(t_0)}$ since $\tilde{x}$ is composed of selected elements of x. By calculating $\frac{\partial \tilde{x}(t)}{\partial u(t_0)}$ from t_0 to t_N, and stacking these sensitivities in a matrix, we can form the following sensitivity matrix:

$$S_C(t_0, \ldots, t_N) = \begin{bmatrix} \dfrac{\partial \tilde{x}_1(t_0)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_1(t_0)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_1(t_0)}{\partial u_{n_u}(t_0)} \\ \dfrac{\partial \tilde{x}_2(t_0)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_2(t_0)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_2(t_0)}{\partial u_{n_u}(t_0)} \\ & \vdots \\ \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_0)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_0)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_0)}{\partial u_{n_u}(t_0)} \\ & \vdots \\ & \vdots \\ \dfrac{\partial \tilde{x}_1(t_N)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_1(t_N)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_1(t_N)}{\partial u_{n_u}(t_0)} \\ \dfrac{\partial \tilde{x}_2(t_N)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_2(t_N)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_2(t_N)}{\partial u_{n_u}(t_0)} \\ & \vdots \\ \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_N)}{\partial u_1(t_0)} & \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_N)}{\partial u_2(t_0)} \cdots \dfrac{\partial \tilde{x}_{\tilde{n}_x}(t_N)}{\partial u_{n_u}(t_0)} \end{bmatrix} \tag{3.16}$$

For determining the reduced input vector $\tilde{u}$, a similar approach to the selection of $\tilde{x}$ is considered based on the sensitivity matrix S_C. First, by applying SVD to S_C, an equation similar to equation (3.12) can be derived. Next, by examining the singular values, we can find the dominant ones. Next, we analyze the associated singular vectors to find the important elements in u that have more significant impact on the reduced state $\tilde{x}$. A similar measure to D_j can be used in determining the important elements in u and these elements in u form the reduced input vector $\tilde{u}$.

3.4 Proposed reduced-order estimator design approach

In the previous section, we discussed how to select the reduced state and input vectors $\tilde{x}$ and $\tilde{u}$. In this section, we discuss how to develop a reduced-order model using LSTM to describe the dynamics between $\tilde{u}$ and $\tilde{x}$. We will also discuss how to design a reduced-order estimator in the framework of EKF based on the reduced-order model.

3.4.1 Reduced-order model development

We propose to use LSTM neural networks to develop the reduced-order model. LSTM networks are typically used for modeling sequential time-series data, such as the trajectories of dynamical chemical processes. While the traditional RNNs suffer from losing error information pertaining to long data sequences, LSTM models can deal with this problem by protecting error information from decaying using learnable gates [94, 95].

As illustrated in Step 2 in Figure 3.1, to develop the reduced-order model, we need to collect data. The data for model development can be generated based on the actual system model as shown in (3.1) based on extensive simulations with different initial states and randomly generated multi-step input sequences. The multi-step input sequence ensures to capture of most of the system's dynamics. Note that since only the inputs in the selected $\tilde{u}$ are important for the target output y_t, only these inputs need to be considered in the data generation and the other inputs can be kept constant which is described in detail in the application section. In the extensive simulations, the trajectories of $\tilde{u}$ and $\tilde{x}$ are collected. It is noted that the generation of these time-series data is an important step of the LSTM modeling. When the data is collected, the entire data set is divided into training, validation, and testing data sets for the LSTM model development. A brief step-by-step procedure for identifying an LSTM model is outlined as follows:

1. Normalise the dataset so that all values are within the range of 0 and 1.
2. Determine the number of layers of the LSTM model, the number of nodes in

each layer; the outputs of the LSTM model should be $\tilde{x}$ and the input to the LSTM model should be $\{\tilde{x}, \tilde{u}\}$.

3. Train the LSTM model using the training dataset. This may be done using, for example, the Keras Library in Python for artificial neural networks.
4. Use the validation and test datasets to validate and evaluate the model performance, respectively. If the model performance (both single-step and multi-step ahead predictions) is not acceptable, go back to Step 2 and retrain the LSTM model. If the performance is good, save the model parameters.

After the LSTM model is trained, the LSTM model parameters can be extracted and the model can be described in the form of (3.2)-(3.4).

3.4.2 Extended Kalman filter design

EKF and its variants are standard methods used for state estimation of nonlinear systems based on successively linearizing the nonlinear system [25, 96]. A traditional EKF is modified to accommodate the sequence length of the LSTM model. Note that EKF is based on successive linearization of the original nonlinear system. If a system cannot be linearized well, numerical approaches may be used to find the Jacobian matrix treating the nonlinear system as a black box. The EKF is divided into two steps - prediction and update steps.

Prediction step. At a sampling time t, using the past n_l estimated reduced state $\hat{\tilde{x}}$ from $t-1$ to $t-n_l$, the EKF predicts the reduced state at the sampling time t. The predicted reduced state is as follows:

$$\hat{\tilde{x}}(t|t-1) = \tilde{f}(\hat{\tilde{x}}(t-1), \ldots, \hat{\tilde{x}}(t-n_l), \tilde{u}(t-1), \ldots, \tilde{u}(t-n_l)) \qquad (3.17)$$

where $\hat{\tilde{x}}(t|t-1)$ represents the prediction of the reduced state at time instant t based on past estimated reduced state $\hat{\tilde{x}}$ from $t-n_l$ to $t-1$.

The propagation of the process disturbance is as follows:

$$P(t|t-1) = \sum_{m=t-n_l}^{t-1} A_d(m+1)P(m)A_d(m+1) + Q \tag{3.18}$$

where P and Q are the reduced state covariance matrix and process noise covariance matrix. The state-transition matrix, $A_d(t) = \left.\frac{\partial \tilde{f}}{\partial \tilde{x}}\right|_{\hat{\tilde{x}}(t-1)}$.

To find the expression of P in (3.18), let us consider the reduced-order model with additive process noise w:

$$\tilde{x}(t) = \tilde{f}(\tilde{x}(t-1), \cdots, \tilde{x}(t-n_l), \tilde{u}(t-1), \cdots, \tilde{u}(t-n_l)) + w(t-1) \tag{3.19}$$

The error between the actual reduced state $\tilde{x}(t)$ and the predicted value from (3.17), $\hat{\tilde{x}}(t|t-1)$, is given as:

$$\begin{aligned} \tilde{x}(t) - \hat{\tilde{x}}(t|t-1) =& \tilde{f}(\tilde{x}(t-1), \cdots, \tilde{x}(t-n_l), \tilde{u}(t-1), \cdots, \tilde{u}(t-n_l)) + w(t-1) \\ &- \tilde{f}(\hat{\tilde{x}}(t-1), \ldots, \hat{\tilde{x}}(t-n_l), \tilde{u}(t-1), \ldots, \tilde{u}(t-n_l)) \end{aligned} \tag{3.20}$$

The estimation error can be approximated using by only considering the linear approximations of the nonlinear system equation:

$$\begin{aligned} \tilde{x}(t) - \hat{\tilde{x}}(t|t-1) \approx& \left.\frac{\partial \tilde{f}}{\partial \tilde{x}}\right|_{\hat{x}(t-1)} (\tilde{x}(t-1) - \hat{\tilde{x}}(t-1)) + \cdots \\ &+ \left.\frac{\partial \tilde{f}}{\partial \tilde{x}}\right|_{\hat{x}(t-n_l)} (\tilde{x}(t-n_l) - \hat{\tilde{x}}(t-n_l)) + w(t-1) \\ =& \sum_{m=t-n_l}^{t-1} A_d(m+1)(\tilde{x}(m) - \hat{\tilde{x}}(m)) + w(t-1) \end{aligned} \tag{3.21}$$

The covariance matrix $P(t|t-1)$ can be calculated as follows:

$$P(t|t-1) = E\big[(\tilde{x}(t) - \hat{\tilde{x}}(t|t-1))(\tilde{x}(t) - \hat{\tilde{x}}(t|t-1))^T\big] \tag{3.22}$$

Based on (3.21) and (3.22), the following equation can be written:

$$P(t|t-1) = E\Bigg[\bigg(\sum_{m=t-n_l}^{t-1} A_d(m+1)(\tilde{x}(m) - \hat{\tilde{x}}(m)) + w(t-1)\bigg)$$

$$\left(\sum_{m=t-n_l}^{t-1} A_d(m+1)(\tilde{x}(m) - \hat{\tilde{x}}(m)) + w(t-1) \right)^T \Bigg] \tag{3.23}$$

Given that the noise $w(t-1)$ is not correlated with the estimates at and before $t-1$ and neglecting the correlation between the estimated reduced states at different time instants, $P(t|t-1)$ can be approximated as follows:

$$P(t|t-1) = \sum_{m=t-n_l}^{t-1} A_d(m+1)P(m)A_d(m+1) + Q \tag{3.24}$$

which is the expression using in (3.18).

Update step. At each sampling instant t, an estimate of the current reduced state $\hat{\tilde{x}}(t)$ is obtained by performing the measurement-update step based on the predicted value $\hat{\tilde{x}}(t|t-1)$ as follows:

$$\hat{\tilde{x}}(t) = \hat{\tilde{x}}(t|t-1) + K_t(y(t) - C\hat{\tilde{x}}(t|t-1)) \tag{3.25}$$

where $\hat{\tilde{x}}(t)$ represents the estimated $\tilde{x}$ at time t given the observations of y up to time t and the observation matrix, $C = \left.\frac{\partial h}{\partial \tilde{x}}\right|_{\hat{\tilde{x}}(t|t-1)}$. The correction gain K_t at time t used to minimize a *posteriori* error covariance based on the measurement innovation (i.e. $y(t) - C\hat{\tilde{x}}(t|t-1)$) can be determined as below:

$$K_t = P(t|t-1)C^T(R + CP(t|t-1)C^T)^{-1} \tag{3.26}$$

where R is the covariance matrix of the measurement noise. The covariance matrix is also updated as follows:

$$P(t) = (I - K_tC)P(t|t-1) \tag{3.27}$$

where $P(t)$ is the *posteriori* error covariance matrix of the estimation error at t and I is an identity matrix. Note that $P(0)$, Q, and R are three tuning parameters for the EKF.

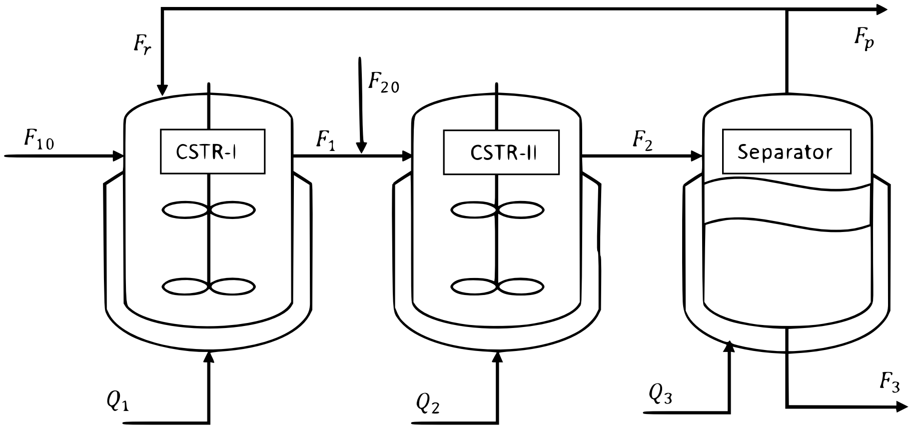


Figure 3.2: Two continuous-stirred tank reactors and a flash separator process

3.5 Application to a chemical process

In this section, we apply the propose reduced-order estimator design approach to a chemical process to illustrate its applicability and effectiveness.

3.5.1 Process description and simulation settings

A chemical process consisting of two continuous stirred tank reactors (CSTR) and a flash separator in series is considered [11]. A process schematic is shown in Figure 3.2. Pure material A is fed at the rate of F_{10} and F_{20} respectively, into the two CSTRs, in which the first-order irreversible exothermic reactions take place, i.e. $A \rightarrow B$ and $B \rightarrow C$. The reactors are assumed to be perfectly mixed, with constant density, liquid volume, and heat capacity. The outlet of the second CSTR is fed into the flash separator at a flow rate F_2. The overhead of the separator is condensed and passed to a downstream unit at flow rate F_p with a recycle to the first reactor F_r and the bottom product stream is removed at flow rate F_3. Each tank is equipped with a jacket to heat or cool the tank and Q_1, Q_2, and Q_3 are heat inputs/removals. A detailed model of the process in the form of ordinary differential equations (ODEs) is described in [11]. The parameter values of the model are shown in Table 3.1. In the model,

Table 3.1: Parameter values

$T_{01} = 300$ K	$\Delta H_1 = -6.0 \times 10^4$ kJ/kmol
$T_{02} = 300$ K	$\Delta H_2 = -7.0 \times 10^4$ kJ/kmol
$E_1 = 5 \times 10^4$ kJ/kmol	$R = 8.314$ kJ/ kmol K
$E_2 = 6 \times 10^4$ kJ/kmol	$k_1 = 9.972 \times 10^6$ hr^{-1}
$c_p = 4.2$ kJ/(kg·K)	$k_2 = 9.36 \times 10^6$ hr^{-1}
$\Delta H_{vapA} = -3.57 \times 10^4$ kJ/kmol	$V_1 = 4$ m^3
$\Delta H_{vapB} = -1.57 \times 10^4$ kJ/kmol	$V_2 = 4$ m^3
$\Delta H_{vapC} = -4.07 \times 10^4$kJ/kmol	$V_3 = 4$ m^3
$\alpha_A = 3.5$	$\alpha_B = 1.0$
$\alpha_C = 0.5$	$X_{A10} = 1.0$
$X_{B10} = 0.0$	$X_{A20} = 1.0$
$X_{B20} = 0.0$	$\rho = 1000$ kg/m^3
$Q_1 = 3 \times 10^6$ kJ/hr	$Q_2 = 1 \times 10^6$ kJ/hr
$Q_3 = 3 \times 10^6$ kJ/hr	$F_p = 0.5$ m^3/hr
$F_{10} = 12.0$ m^3/hr	$F_{20} = 3.0$ m^3/hr
$F_r = 13.4$ m^3/hr	

there are in total nine ODEs corresponding to the dynamics of the concentrations and temperatures of each tank. It is assumed that the temperatures of all the tanks are measurable, so the system outputs $y = [T_1, T_2, T_3]^T$. The concentrations of components A and B in each tank and the temperatures of the tanks are the states of the process; that is, $x = [X_{A1}, X_{B1}, T_1, X_{A2}, X_{B2}, T_2, X_{A3}, X_{B3}, T_3]^T$ and input vector is $u = [F_{10}, F_{20}, Q_1, Q_2, Q_3, F_r, F_p]^T$.

In order to control and monitor the quality of the product, it is assumed that the concentration of component B in the separator X_{B3} is an important process variable and is considered as the target variable; that is, $y_t = [X_{B3}]$. It is desired to estimate X_{B3} at each sampling time based on the process information of the input u and the measured output y. Since we are mainly concerned about the target output X_{B3}, it

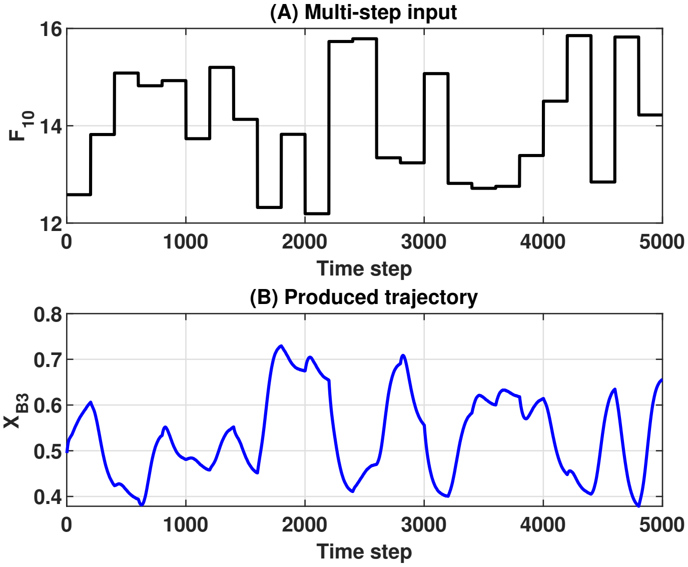


Figure 3.3: Trajectories of input F_{10} and the target output X_{B3}

is not necessary to estimate all the other states of the process.

A first principle model of the process was built based on physio-chemical phenomena to connect different unit operations by mass and heat balances [23]. It is verified that based on the measurements of y and the first principle model, the entire state x can be estimated. In this section, we illustrate how a reduced-order estimator may be designed using the proposed approach to estimate X_{B3} and will compare the performance of the reduced-order estimator with a full-order estimator based on the actual first principle model to show the benefits of using a reduced-order estimator.

3.5.2 Selection of the reduced state and input vectors

Following the steps illustrated in Figure 3.1, firstly, to obtain sensitivity matrices for reduced state and input selection, we perform open-loop simulations based on the first principle model of the process to generate data. In the simulations, the values of the process parameters used are shown in Table 3.1. A random process noise $w(t)$

is added and the noise is generated following a Gaussian white noise with zero mean and standard deviation of 0.01. The process model is solved using the fourth-order Runge Kutta method with a sampling time of 0.01 hr. The entire data is produced using randomly generated inputs within the allowable ranges. The inputs change every 2 hours. Figure 3.3 shows the trajectories of one of the inputs and the target output.

We find the sensitivity matrix of the target output to the state S_O following (3.11). Then, we apply SVD to the sensitivity matrix to find the singular values and the associated singular vectors. Figure 3.4A represents the nonzero singular values of the sensitivity matrix in a semi-log plot, in descending order, along with its index number. There are eight entries in this plot therefore it has a zero singular value. It can be observed from the plot that there is a clear gap between the 3^{rd} and 4^{th} singular values. Therefore, the first three singular values are considered the dominant ones.

After determining the dominant singular values, we calculate measure (3.13) based on the three dominant singular values. Figure 3.4B shows the D_j values of each of the state elements or the contributions of the state elements to the three singular values. From Figure 3.4B, it can be seen that T_1, X_{B3}, T_2, X_{B2} have relatively larger D_j values, which implies that these state elements contribute to the three dominant eigenvalues most. Further, X_{B1}, X_{B2}, X_{A2}, X_{A1}, X_{A3} have a much smaller D_j values and T_3 has a D_j value equal to 0. These smaller D_j values imply that the corresponding state elements contribute much less to the three dominant singular values. Therefore, the initial reduced state vector can include T_1, X_{B3}, T_2, and X_{B2}, which have relatively larger impact on the three dominant singular values. That is, $\tilde{x} = [T_1, X_{B3}, T_2, X_{B2}]^T$. Note that the current $\tilde{x}$ is determined based on y_t. Next, we continue to check whether the measured outputs $y = [y_1, y_2, y_3]^T = [T_1, T_2, T_3]^T$ can be expressed based on $\tilde{x}$. It can be found that $y_3 = T_3$ cannot be expressed based

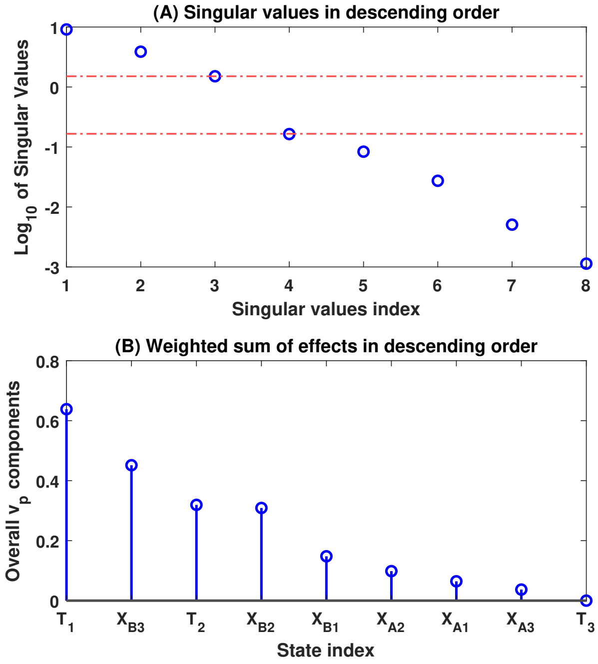


Figure 3.4: Singular values of S_O and the D_j values associated with the dominant singular values.

on $\tilde{x}$. However, since T_3 is not related to the three dominant singular values ($D_j = 0$ for T_3), y_3 is not really useful in estimating the target output. Instead of expanding $\tilde{x}$ to include T_3, we can remove y_3 from the measurements used in estimating y_t.

Once $\tilde{x} = [T_1, X_{B3}, T_2, X_{B2}]^T$ is determined, we continue to determine the reduced input vector $\tilde{u}$. Similarly, based on open-loop simulation data, we calculate the sensitivity matrix S_C following (3.16). Then, we apply SVD to S_C to find the dominant singular values of S_C and then the closely related inputs. Figure 3.5A shows the singular values of S_C. From the figure, it can be seen that there is a significant gap between the 6^{th} and 7^{th} singular values. Therefore, these six singular values are considered to be the dominant ones. For these singular values, we further calculate the D_j values, which are shown in Figure 3.5B. From Figure 3.5B, it can be seen that

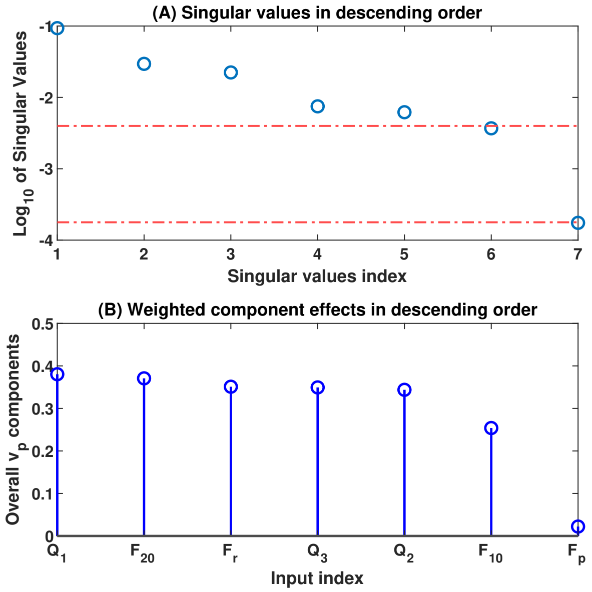


Figure 3.5: Singular values of S_C and the D_j values associated with the dominant singular values.

there are six inputs that contribute significantly to the singular values. These inputs are included in the reduced input vector. That is, $\tilde{u} = [Q_1, F_{20}, F_r, Q_3, Q_2, F_{10}]^T$.

The elements of the reduced state vector $\tilde{x}$, the reduced input vector $\tilde{u}$, and the measured outputs used for estimating y_t are summarized in Table 3.2.

3.5.3 Reduced-order model and estimator

Based on the selected $\tilde{x}$, $\tilde{u}$, further open-loop simulations of the actual process model are performed and data is collected for LSTM model development. Noted the data is only collected for $\tilde{x}$ and $\tilde{u}$.

In the training of the LSTM, different n_l values were considered and it was found that $n_l = 2$ is sufficient to get a good LSTM model. The LSTM has two hidden layers of 50 neurons and one dense layer of 4 neurons as the output layer.

Once the LSTM model is developed, the EKF is designed based on the LSTM

Table 3.2: Elements of the reduced state and input vectors and the used measured outputs

State	X_{B3}	T_1	T_2	X_{B2}		
Output	T_1	T_2				
Input	Q_1	F_{20}	F_r	Q_3	Q_2	F_{10}

model. For the EKF estimator, the weighting matrices are diagonal matrices with $Q = diag\{[0.005^2, 0.005^2, 0.005^2, 0.005^2]\}$, $R = diag\{[20^2, 20^2]\}$, and the matrix $P(0) = diag\{[100^2, 100^2, 100^2, 100^2]\}$.

Although, the LSTM was selected due to its ability to remember both short-term and long-term temporal dependencies, the real benefits of the LSTM are expected to be observed when the system under study considers a longer sequence length. Therefore, it is possible to apply different modeling approaches to find the structure of the model, and some of the methods are mentioned briefly for this application. It is the case that for the sequence length used in the simulation example, a vanilla RNN will be applicable since the vanishing/exploding gradient phenomenon is not expected to occur. Another model structure such as a deep neural network (DNN) can also be used to describe the system employed in the simulation example, after an initial manipulation of the dataset's inputs. However, such a model will not be flexible since a sequence length of two must be used during the model's implementation. This limitation is however absent in the RNN/LSTM since the sequence length used for the model development may be varied during its development.

In this work, the number of epochs (30) and batch size (100) hyperparameters were selected based on a number of simulation experiments. It was also observed that the default learning rate value set by the Keras library was suitable for the developed model. Similarly, the LSTM architecture (units = 50) was determined through a number of simulation experiments. While this brute force approach may be somewhat suitable for smaller LSTM models, a systematic approach can be considered to tune the hyperparameters of the identified LSTM model. The 'KerasTuner', a tool in the Keras library, can be used for selecting and tuning hyperparameters of the LSTM model

3.5.4 Results and discussion

In this subsection, we evaluate the performance of the above-developed reduced-order estimator. To evaluate the performance, we will use the average normalized estimation error of the target output X_{B3} as shown below:

$$\sigma_{X_{B3}} = \sqrt{\frac{1}{N_{sim}} \sum_{j=0}^{N_{sim}} \left(\frac{\hat{X}_{B3}(t_j) - X_{B3}(t_j)}{X_{B3}(t_j)} \right)^2} \tag{3.28}$$

where N_{sim} indicates the total simulation steps, $\hat{X}_{B3}$ denotes the estimated value, and X_{B3} denotes the actual value of the target variable. All the simulations were conducted in a desktop computer with an Intel i7 CPU at 3.2 GHz and 16 GB RAM. The LSTM models were trained using Keras and TensorFlow in Python programming language.

We design various simulation cases to test the performance of the proposed reduced-order estimation. Specifically, we consider three different schemes: (a) Scheme 1 - the proposed reduced-order estimator based on the LSTM model with $n_l = 2$, (b) Scheme 2 - a soft sensor that exploits the correlation between y, u and y_t, (c) Scheme 3, a full-order state estimator design based on a regular EKF based on the actual full-order process model, and Scheme 4, an LSTM model ($n_l = 2$) with all the measured variables (u, y) as well as the target output y_t as the inputs and the target output y_t as the single output. For these schemes, they are tuned to give their best performance for a fair comparison. Further, we note that the soft sensor in Scheme 2 was developed based on a dense neural network with y and u are the inputs and y_t as the output. Such a soft sensor explores the static correlation

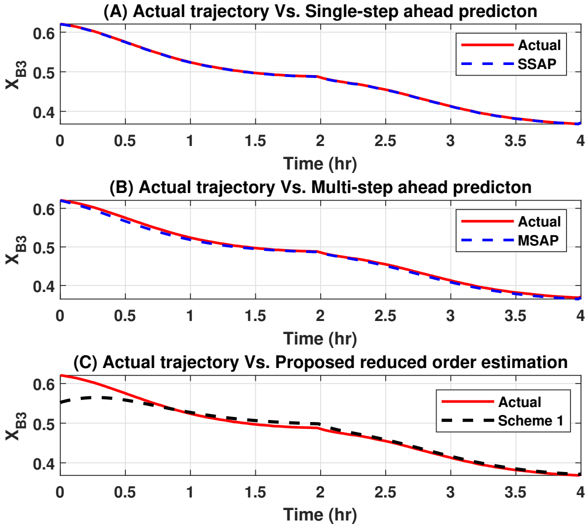


Figure 3.6: Trajectories of the actual states X_{B3}, the single-step ahead prediction (SSAP) (A), the multi-step ahead prediction (MSAP) using the LSTM model (B), and the estimated target variable using the proposed reduced-order estimator in Scheme 1 (C).

between the y, u, and y_t but does not consider the dynamics of the system. Scheme 4 explores only the dynamic relation between the measured variables and the target output. While in the proposed Scheme 1, the LSTM includes more state variables in its inputs and outputs to capture the dynamics that are essential for estimating the target output. Since there are unmeasured state variables in the LSTM model used in Scheme 1, the EKF is used together with the LSTM to estimate the target output based on the measured variables.

First, we show the LSTM modeling and reduced-order estimation performance. Figures 3.6A-B shows the actual target output X_{B3} and the one-step ahead and multi-step ahead open-loop predictions using the trained LSTM. From these plots, it can be seen that the trained LSTM model has a very good performance in predicting the evolution of X_{B3}. Based on many simulations with different initial conditions and noise realizations, the corresponding $\sigma_{X_{B3}}$ for the single-step and multi-step ahead

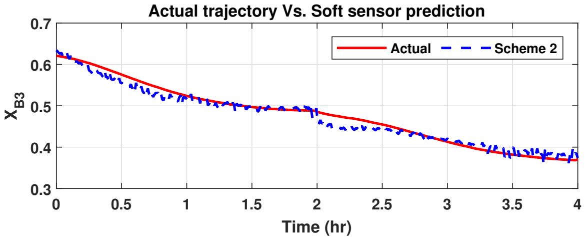


Figure 3.7: Trajectories of the actual states X_{B3} and the predicted target variable using the soft sensor in Scheme 2.

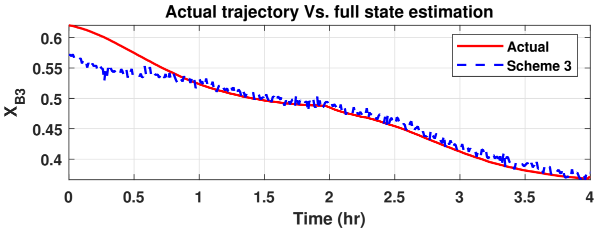


Figure 3.8: Trajectories of the actual states X_{B3} and the estimated target variable using the full-order estimation in Scheme 3.

predictions are 0.111% and 1.003%. These numbers further verify that the trained model has a good performance. Note that the results in Figures 3.6A-B are based on an initialization of the LSTM model from the actual initial X_{B3} value. It only shows the performance of the trained model.

Figures 3.6C shows the estimation performance of the proposed reduced-order estimation scheme. The estimator was initialized using a value that is different from the actual X_{B3} value. From the plot, it can be seen that the estimate of the proposed reduced-order estimation scheme (Scheme 1) can converge to the actual value quickly and then follow the actual value closely. The corresponding $\sigma_{X_{B3}}$ of the proposed reduced-order estimator is 1.43% from extensive simulations with different initial conditions and noise realizations. This demonstrates that the proposed approach is effective and applicable if only the target output is needed to be estimated.

Next, we present the performance of the soft sensor in Scheme 2. The inputs of

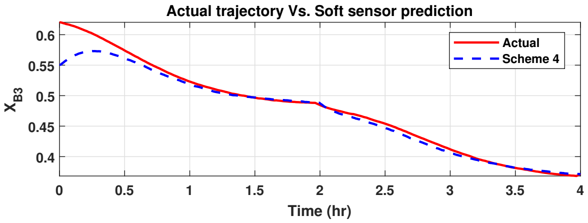


Figure 3.9: Trajectories of the actual states X_{B3} and the predicted target variable using the soft sensor in Scheme 4.

the soft sensor are T_1, T_2, T_3,Q_1, F_{20}, F_r, Q_3, Q_2 and F_{10} and the output is the target variable X_{B3}. A dense neural network is trained. Figure 3.7 shows the results on the same trajectory of X_{B3} as used in the previous simulation. From the figure, it can be seen that while the soft sensor can overall track the trend of X_{B3} but the prediction performance is much poorer compared with the estimator in Scheme 1. The $\sigma_{X_{B3}}$ of the soft sensor (Scheme 2) calculated from various simulations was 5.36%, which is much larger than the value for the proposed reduced-order estimator. This set of simulations illustrated that the proposed reduced-order estimator gives much-improved estimation performance compared with the soft sensor. The improvement in the estimation performance in Scheme 1 compared with Scheme 2 is from the explicit consideration of the dynamics of the system and the use of EKF. Note that the performance metrics reported are obtained from many simulations.

Then, we consider the full-order EKF based on the actual nonlinear model of the process. Figure 3.8 shows the estimation performance of the full-order EKF (Scheme 3). From Figure 3.8, it can be seen that the full-order EKF can also track the trend of X_{B3} but the estimate is much noisy compared with the estimated values in Scheme 1. The corresponding $\sigma_{X_{B3}}$ value of the full-order EKF is 4.61%, which is relatively higher than the value for Scheme 1.

The decreased performance of the full-order estimator in Scheme 3 compared with Scheme 1 may be explained by examining the degrees of the observability of the estimated variables in the two schemes. Let us consider the following criterion for measuring the degree of observability of a system [23, 89]:

$$\gamma(D_o) = \frac{min\{\lambda_i(D_o|i = 1, \cdots, n)\}}{max\{\lambda_i(D_o|i = 1, \cdots, n)\}} \tag{3.29}$$

where D_o is the observability matrix of the system and λ_i is the i^{th} singular value of D_o. For the nonlinear process and the nonlinear reduced-order model, we use linearization to find the corresponding linear system and then construct the corresponding observability matrix D_o. For the full-order nonlinear system, the degree of observability was found to be about 1.3×10^{-5} and for the reduced-order model, the degree of observability was 0.02. It is obvious that the reduced-order system has a much larger degree of observability compared with the full-order system. This makes sense since in the reduced-order model, the measured outputs are only used to estimate the 4 selected states but in the full-order estimator, the same number of measured outputs are used to estimate the entire state vector x which contains 9 elements. The much-improved degree of observability explains why Scheme 1 gives much-improved estimation performance compared with Scheme 3.

Now, we present the performance of the LSTM model in Scheme 4. Figure 3.9 shows the predicted target output and the actual trajectory of the target output. From the figure, it can be seen that the LSTM in Scheme 4 gives a relatively poor performance in predicting the target output compared with the proposed approach in Scheme 1. This can also be seen from the $\sigma_{X_{B3}}$ value, which is 2.00% for Scheme 4 and is 1.43% for the proposed Scheme 1. This is indeed expected since the LSTM in Scheme 4 only uses the measured variables and they cannot appropriately capture the essential dynamics that are needed to describe the target output X_{B3}. The proposed approach keeps all the necessary variables needed to capture the dynamics of X_{B3} in the LSTM and the EKF can be used to estimate the target output and the other unmeasured variables. It is also verified that the proposed approach is

Table 3.3: $\sigma_{X_{B3}}$ values for the trained LSTM model and the different schemes

Methods	$\sigma_{X_{B3}}$ in percentage (%)	Simulation time (s)
SSAP	0.111	-
MSAP	1.003	-
Scheme 1	1.43	15
Scheme 2	5.36	3
Scheme 3	4.61	20
Scheme 4	2.00	3

more robust to noise in the measured variables given the use of the EKF while the LSTM in Scheme 4 is more sensitive to measurement noise. When the variance of the measurement noise increases to $\sigma_v = 0.1$, the proposed approach gives $\sigma_{X_{B3}} = 2.02\%$ while the LSTM in Scheme 4 leads to a much worse $\sigma_{X_{B3}} = 6.12\%$.

Finally, we consider the computational complexity of the four schemes. Table 3.3 shows the simulation times of the four schemes. It can be seen that the soft sensor in Scheme 2 and the LSTM in Scheme 4 are the fastest. The proposed reduced-order estimator (Scheme 1) gives the best estimation performance and has a relatively smaller computational complexity (15 seconds for the entire simulation) compared with the full-order estimator (20 seconds for the entire simulation). The smaller computational times of Schemes 2 and 4 are due to the fact that they do not need to evaluate the EKF. Whereas, Scheme 1 and Scheme 3 calculate the Jacobians at every instant for predicting the covariance matrix, and also EKF has the update step which requires additional time. Scheme 3 evaluates the Jacobian matrix which is bigger than Scheme 1 and also updates for all the variables so comparatively, the computational cost is higher than Scheme 1. This further illustrates that the proposed reduced-order estimator can bring much-improved estimation performance using even less computational resources compared with a full-order estimator.

3.6 Summary

This chapter proposes an approach to select the appropriate inputs and outputs for data-driven reduced-order model development in the framework of LSTM neural network for reduced-order estimator design. A sensitivity-based approach was used in the reduced state and input vector selection. The LSTM neural network was used to develop the reduced-order model and the EKF was used to develop the reduced-order estimator. The application to a chemical process demonstrated the applicability and effectiveness of the proposed approach in achieving good target output estimation. In the simulations, the proposed approach was compared with a soft sensor design that did not consider the dynamics of the process, a full-order EKF, and an LSTM that only uses the measured variables. It was found that the proposed approach gives the best target output estimation performance whose $\sigma_{X_{B3}}$ is about 30% smaller than Scheme 4 which uses only measured variables in the training of the LSTM, and is more than 70% smaller than the full-order state estimator and the soft sensor that only uses the static relation between the measured variables and the target output.

Chapter 4

Adaptive model reduction and soil moisture estimation for agro-hydrological systems

This chapter presents a model reduction approach for state estimation of agro-hydrological systems. Specifically, we consider estimation of soil water content of agro-hydrological systems. Accurate soil moisture information is crucial for developing a precise irrigation control strategy to enhance water use efficiency. However, addressing the challenges associated with state estimation becomes necessary when dealing with large-scale fields. The major challenge in soil moisture estimation lies in the high dimensionality of the spatially discretized agro-hydrological models. We propose an error-triggered reduced state estimation approach to address this challenge. The proposed approach introduces a model reduction method that uses trajectory-based unsupervised machine learning techniques. Due to variations in soil dynamics, model mismatch is inevitable over time. Therefore, an error-trigger criterion is chosen to adaptively update the reduced model such that the prediction error between the reduced model and the original model over a prediction horizon is maintained below a predetermined threshold. An adaptive extended Kalman filter (EKF) is designed accordingly. The performance of the proposed algorithms is evaluated first using a

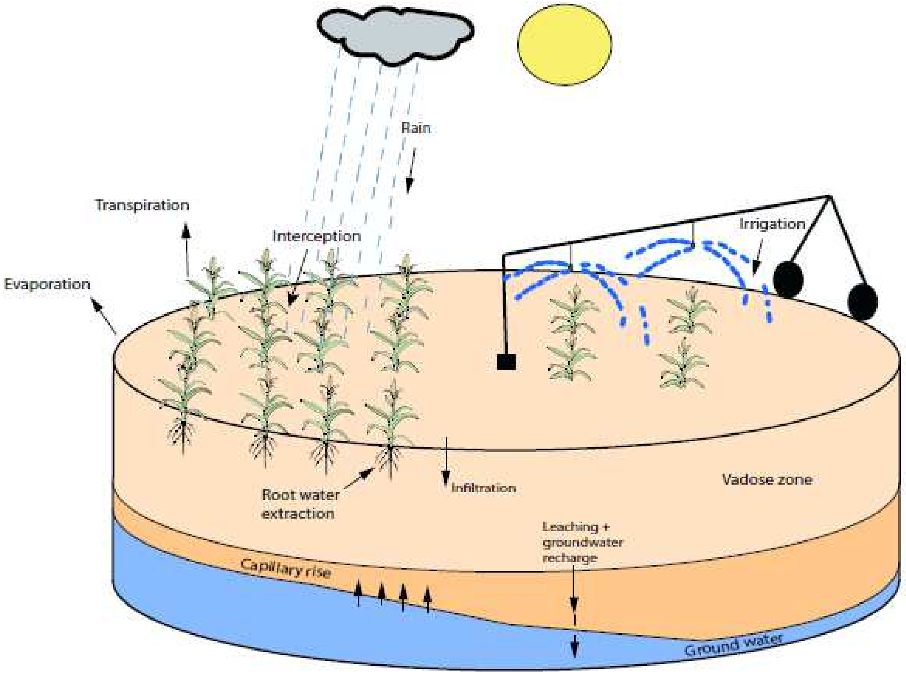


Figure 4.1: A diagram of an agro-hydrological system

small simulated field. Then, the proposed approach is also applied to a large-scale simulated agricultural field to evaluate its applicability.

4.1 Preliminaries

4.1.1 Description of agricultural systems

An agricultural field represents the complex dynamics involving soil, crops, the atmosphere, and water. A diagram of the agricultural system is shown in Figure 4.1. The water inputs into the system encompass external irrigation to the field and rain and other precipitation, while the main outputs include natural evaporation, transpiration, and groundwater drainage. The process of water infiltration into the soil, driven by both capillary and gravitational forces, is mathematically modeled by the

Richards equation, which was initially developed in [97] as follows:

$$\frac{\partial \theta_m}{\partial t} = c(h)\frac{\partial h}{\partial t} = \nabla \cdot (K(h)\nabla(h + z)) + S(h, z) \tag{4.1}$$

where h [m] is the pressure head as the length of the water column, θ_m [m^{3}m^{-3}] is the volumetric soil moisture content which is the volume of water to the unit volume of soil, $c(h)$ [m^{-1}] is the soil water capillary capacity, $K(h)$ [ms^{-1}] is the unsaturated hydraulic soil water conductivity which shows how readily water can flow through soil, z [m] is the axial coordinate, and $S(h, z)$ [m^{3}m^{-3}s^{-1}] is the sink term corresponds to the removal or decrease of water from the system. The equation between $K(h)$ and h is provided as below [98]:

$$K(h) = \begin{cases} K_s Q_s^{\lambda}[1 - (1 - (Q_s^{\frac{1}{m}})^m]^2, & Q_s < 1 \\ K_s, & Q_s \geq 1 \end{cases} \tag{4.2}$$

where K_s [ms^{-1}] is the saturated hydraulic conductivity, and Q_s can be expressed as $Q_s = 1 + (-h\alpha)^n$, $m, \lambda, \alpha,$ and n are the soil shape factors which are hydraulic soil properties and vary with soil type. The $c(h)$ is represented in following equation [99]:

$$c(h) = \begin{cases} (\theta_S - \theta_R)\alpha n(1 - (\frac{1}{n}))(-h\alpha)^{n-1}(1 + (-h\alpha)^n)^{-(2-(1/n))}, & h < 0 \\ S_R, & h \geq 0 \end{cases} \tag{4.3}$$

where θ_S [m^{3}m^{-3}], θ_R [m^{3}m^{-3}], and S_R [m^{-1}] correspond to the saturated soil moisture content, the residual soil moisture content, and the specific storage coefficient of the porous medium when subjected to positive pressure. The soil hydraulic function θ_m is defined as:

$$\theta_m = \theta_R + (\theta_S - \theta_R)[1 + (-\alpha h)^n]^{(\frac{1}{n} - 1)} \tag{4.4}$$

The $S(h, z)$ is the rate of root water uptake by plants. The optimum water extraction by the roots of the plants is considered and evaluated as defined below [100]:

$$S_{\max}(h, z) = \frac{ETP_p}{L_z} \tag{4.5}$$

where ETP_p [ms^{-1}] denotes the rates of potential evaporation and L_z [m] is the length of the roots in the soil. The rate of the potential evaporation ETP_p is found by:

$$ETP_p = K_c ET \tag{4.6}$$

where K_c [-] is the crop coefficient and ET [ms^{-1}] is the reference evaporation rate. ET can be evaluated using the Penmon-Moneith equation can be found in [100]. K_c [-] is an ambient temperature dependant variable. Additional insights and supporting equations can be found in [55].

4.1.2 Problem formulation and state-space model

We consider agricultural fields equipped with a center-pivot irrigation, as depicted in Figure 4.1. This center-pivot rotates in a circular pattern. To capture the motion of the center-pivot irrigation system, the Richards equation in (4.1) is reformulated in 3D cylindrical form, as detailed in [57] as follows:

$$c(h)\frac{\partial h}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right] + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\frac{K(h)}{r}\frac{\partial h}{\partial \theta}\right] + \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z} + 1\right)\right] + S(h, z) \tag{4.7}$$

The Neumann boundary condition characterizes the surface boundary as follows:

$$\left.\frac{\partial h}{\partial z}\right|_{r,\theta,z=z_s} = -1 - \frac{u(t)}{K(h)}$$

where r, θ, and z represent the spacial variables for radial, azimuthal, and axial directions, respectively, $u(t)$ is the input that includes irrigation and precipitation at the surface of the field, and z_s is the soil depth. The bottom boundary condition of the soil is specified as free discharge. The Richards equation (4.7), which models agro-hydrological processes in three dimensions, is a nonlinear PDE that poses challenges for analytical solutions. For numerical analysis, we adopt the explicit centralized finite difference method to discretize Equation (4.7) to tackle this challenge. The

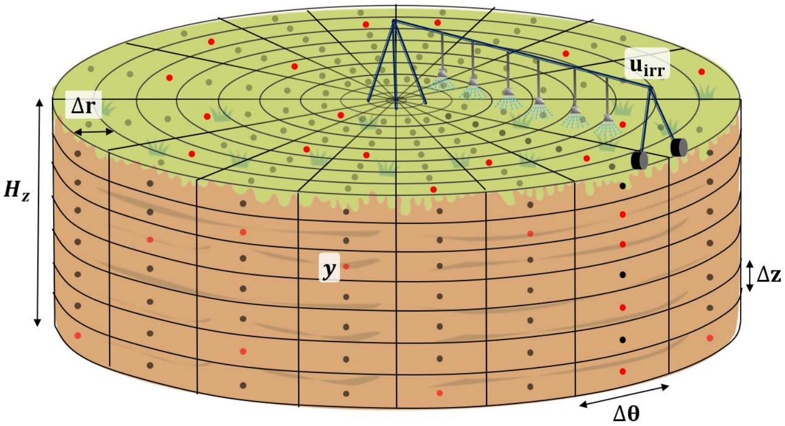


Figure 4.2: Discretization of the agricultural field where each dot denotes the discretized node and red dots indicate the point sensors

spatial discretization of the model establishes a continuous-time state-space model as follows:

$$\dot{x}(t) = f(x(t), u(t)) + w(t) \tag{4.8a}$$

$$y(t) = Cx(t) + v(t) \tag{4.8b}$$

where $x(t) \in \mathbb{R}^{N_x}$ denotes the soil pressure head value (state vector) of size N_x and $u \in \mathbb{R}^{N_u}$ represents the irrigation at the surface (input vector) with dimension N_u. $y(t) \in \mathbb{R}^{N_y}$ denotes the measurements at each sensor node in pressure head (observation vector), $w(t) \in \mathbb{R}^{N_x}$ is the additive process disturbance, C is a matrix indicating the relation between x and y, and $v(t) \in \mathbb{R}^{N_y}$ denotes the noise associated with measurements.

A discretized diagram of the agricultural field is provided in Figure 4.2. The model is discretized into total N_x nodes with N_r, N_θ, and N_z nodes in the radial, axial, and azimuthal directions, respectively. Noted that the total number of soil moisture nodes (states) is the discretized nodes N_x. Therefore, the dimension of the irrigation input u is the same as the radial N_r nodes shown in the diagram 4.2. That means, at any moment, the central pivot can irrigate radial nodes N_r for a

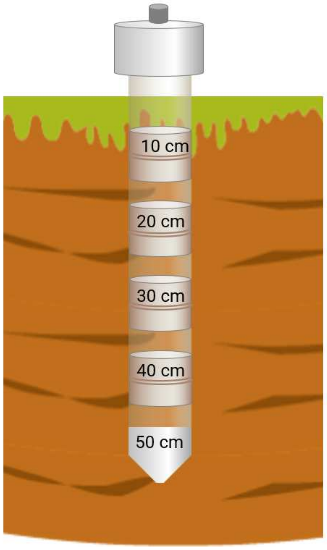


Figure 4.3: A schematic of a point sensor

particular axial direction, leaving the rest of the field unirrigated. However, the field experiences rainfall that is evenly distributed across its entirety.

In this work, we assume that point sensors (as shown in Figure 4.3) are deployed over the field. The principle and details of the point measurements in soil are provided in [66]. These sensors measure the soil pressure head at specific locations and depths within the field, such as the surface, the rooting depth, or at various depths within the soil. During each sampling period, the sensors record the soil pressure head measurements y at a total of N_y locations and the equation (4.8b) shows the relation between x and y. This system possesses inherent stability, rendering it easily detectable. Moreover, it is presumed that $N_y \ll N_x$, yet it still enables the observability of the complete set of states x.

Since this system is inherently high-dimensional, the state-space model as shown in (4.8), can be challenging to implement in any state estimation framework. To address this issue, one possible approach is to use model reduction. A reduced-order model that approximates the Richards equation can be obtained and used in state

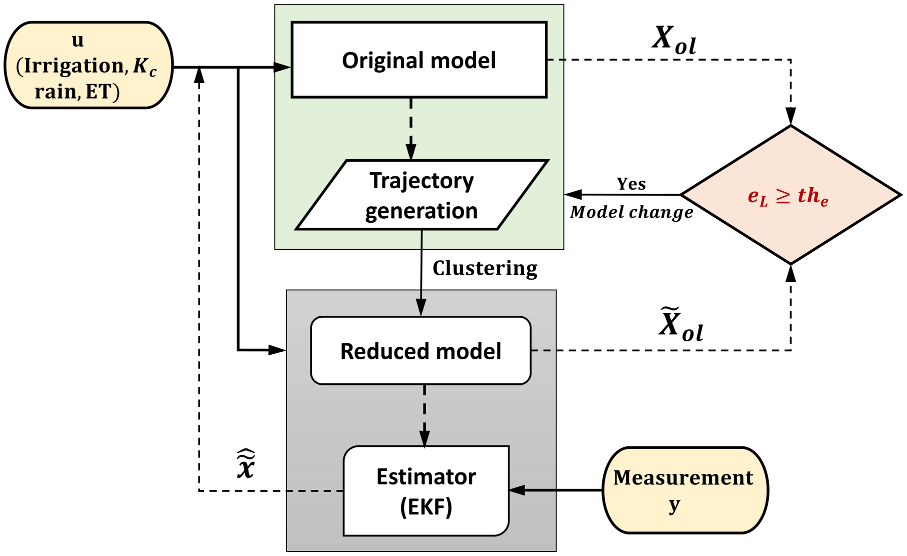


Figure 4.4: Proposed error-triggered adaptive model reduction and state estimation scheme

estimation. Over a growing season, soil properties may change over time especially when the soil is too dry or too wet. Therefore, it is crucial to adapt the reduced model to different conditions.

Our goal is to obtain information on soil moisture in terms of pressure head x at each discrete node throughout the field by utilizing the measurement y. It is assumed that y is sampled with a sampling time Δ; that is, $y(t_k)$ with $t_k = k\Delta$, $k = 0, 1, \ldots$, are available. This is a standard state estimation problem except that the size of x can be very large. The large dimensionality typically leads to two challenges: (a) computational complexity of the model and the associated estimation schemes and (b) low degree of observability of soil moisture pressure head (x) when the number of measured outputs (y) is small. To address this challenge, an error-triggered adaptive model reduction is designed by using a clustering technique inspired by the work [101]. Finally, we propose a state estimator for soil moisture based on this adaptive model reduction approach.

4.2 Proposed model reduction and state estimation

In this section, we present the error-triggered adaptive model reduction method and the associated extended Kalman filter design. Figure 4.4 illustrates the essential steps involved in the proposed approach. The state-space model of Richards equation (4.8) serves as the original model, and an error metric e_L is evaluated at each sampling time t_k to observe the prediction accuracy of the reduced model against the original model. In Figure 4.4, X_{ol} and $\tilde{X}_{ol}$ are the open loop predictions from the original model and reduced model respectively and $\hat{\tilde{x}}$ denotes the estimated state using model reduction. The error metric e_L will be introduced later in equation (4.18). If the current reduced-order model begins to fail in describing the soil water dynamics, for instance, due to variations in the soil moisture dynamics, e_L starts to increase, indicating a deviation from the original model. When e_L exceeds a pre-determined threshold th_e, the scheduled irrigation input and the weather forecast are used to collect the state trajectories based on simulating the original model, and a new reduced model is created. Soil moisture estimation of the entire field is performed based on the reduced model and field measurements. The main components of the proposed approach are explained in the remainder of this section.

4.2.1 Adaptive model reduction

The implementation involves identifying a reduced model used for soil moisture estimation. At each sampling time, the metric e_L is computed, and if it surpasses the threshold th_e, the reduced model is re-identified. For model identification initially or upon e_L surpassing the threshold, identical steps are followed, outlined as follows:

Step 1: State trajectory generation

At a sampling time t_k, when the identification of the reduced-order model is triggered, the soil moisture head or state trajectory of the original model (4.7) is generated. Specifically, the currently estimated state at t_k is used as the initial condition, and the equation (4.8) is simulated with prescribed irrigation actions for total sampling intervals of N_{f_d}. The trajectory of the state over the N_{f_d} steps is denoted as $\mathscr{X}_m$ as follows:

$$\mathscr{X}_m = [x(t_k) \; x(t_{k+1}) \; \ldots \; x(t_{k+N_{f_d}})]^T$$

where $\mathscr{X}_m \in \mathbb{R}^{N_{fd} \times N_x}$ is the state snapshot matrix for the m^{th} model reduction assuming that there were $m-1$ model reductions performed before t_k. The reduced models that are generated during this process are expected to perform well for at least N_{f_d} sampling time intervals.

Step 2: Clustering and reduced model creation

The new reduced-order model is created using the snapshot matrix $\mathscr{X}_m$. Each column in $\mathscr{X}_m$ indicates the trajectory of a state element or a node x_i where i $(i = 1, \ldots, N_x)$. The purpose of the clustering is to merge similar trajectories into one cluster. Instead of the state element x_i, a cluster will be considered as a state element of the reduced model. In this study, we used an agglomerative hierarchical clustering [102] technique to find the clusters for the trajectories. Initially, individual state elements are treated as clusters, and subsequently, the distances between these clusters are computed. Then, the clusters are merged such that the average distance is smaller than a threshold th_C. The threshold of the distance between the clusters plays a crucial role as a tunable factor in determining the performance of the reduced-order model. In assessing accuracy, the state elements' similarity is quantified commonly by the Euclidean distance between trajectories or the state elements. The average distance between two clusters is calculated as follows:

$$\mathcal{D}(a,b) = \frac{1}{n_a n_b} \sum_{m=1}^{n_a} \sum_{n=1}^{n_b} d(x_{am}, x_{bn})$$

where a and b denote the two distinct clusters, n_a, n_b are the sizes of the clusters of a and b respectively, x_{am} and x_{bn} denote data points within clusters a and b respectively.

Consider that after clustering, there are r_m clusters. Denote $C^{(m)} = \{C_1^{(m)}, C_2^{(m)}, \ldots, C_{r_m}^{(m)}\}$ as the collection of clusters for the m^{th} model reduction. The clusters adhere to the important criteria: i) $C_i^{(m)} \cap C_j^{(m)} = \Phi$ and ii) $C_1^{(m)} \cup C_2^{(m)} \cup \ldots \cup C_{r_m}^{(m)} = \mathscr{X}_m$.

The creation of the m^{th} reduced system relies on the utilization of the Petrov-Galerkin projection methodology, as elaborated in the work by Antoulas (2005) [1]. Within this Petrov-Galerkin projection approach, the fundamental component is the projection matrix, denoted as $U^{(m)} \in \mathbb{R}^{N_x \times r_m}$. This matrix is systematically crafted based on the structure of the clusters ($C^{(m)}$). The individual elements of $U^{(m)}$ are mathematically represented as follows:

$$U_{i,j}^{(m)} = \begin{cases} w_i, & \text{if point } i \in C_j^{(m)} \\ 0, & \text{else} \end{cases}$$

where w_i is the weight of each state element i in a cluster C_j during the m^{th} model reduction and can be found in the following equation:

$$w_i = 1/||\alpha_i||, \;\; \alpha_i = \mathbb{E}_i^T \alpha$$

where α denotes the inclusion of the state element in the cluster can be defined as $\alpha = [1, \ldots, 1]^T \in \mathbb{R}^{N_x}$, $||\alpha_i||$ denotes the L_2 norm of α_i, $\mathbb{E}_i = e_{C_i} \in \mathbb{R}^{N_x \times N_{C_i}}$ (N_{C_i} denied as the size of cluster i C_i) is a matrix with columns of e_j's and each e_j is the j-th column of the identity matrix $\mathbb{R}^{N_x \times N_x}$.

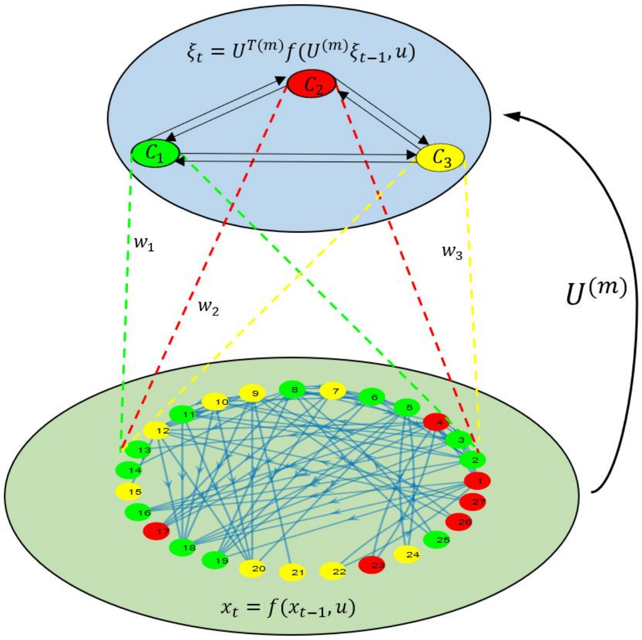


Figure 4.5: A representation of m^{th} model reduction

The m^{th} reduced model of (4.8) is illustrated in Figure 4.5 and the reduced state space expressed as follows:

$$\dot{\xi}^{(m)}(t) = f_r^{(m)}(\xi^{(m)}(t), u(t), w(t)) \tag{4.9}$$

where $f_r^{(m)} = U^{(m)T} f$ and $\xi^{(m)}(t) = U^{(m)T} x(t)$. It is worth noting that the predicted state from (4.8), x can be found by approximation of the original state through this equation $x(t) = U^{(m)} \xi^{(m)}$. After carrying out the numerical discretization, we found a discrete-time reduced model is shown below:

$$\xi^{(m)}(t_{k+1}) = f_{rd}^{(m)}(\xi^{(m)}(t_k), u(t_k), w(t_k)) \tag{4.10}$$

where f_{rd} is the discrete-time function of the reduced-order model. It's essential to highlight that during the transition from one reduced model to another, the dimension of the new reduced-order model may not align with the dimension of the previous one. To facilitate a seamless transition between these reduced models, a

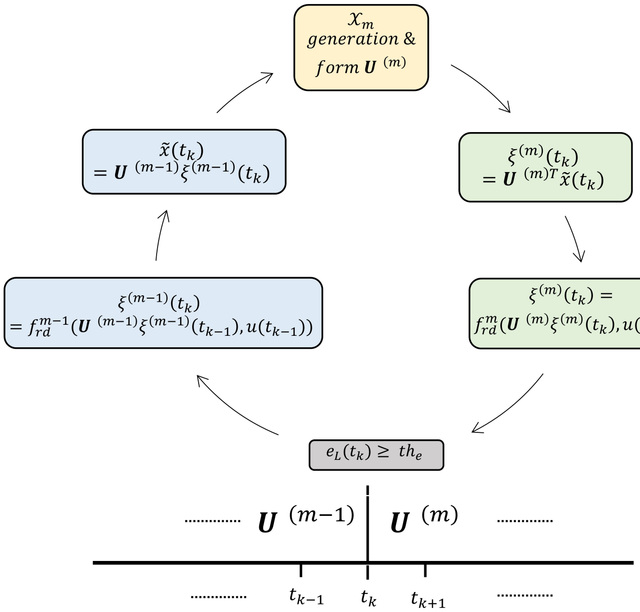


Figure 4.6: Information transformation from one reduced model to another

two-step process is employed. Initially, the state information of the previous reduced model is mapped back to the full state space. Subsequently, the entire state space is projected onto the new reduced model using the recently computed projection matrix. Figure 4.6 illustrates how information is transferred between models.

4.2.2 Adaptive extended Kalman filter

We propose an adaptive reduced state estimator in the framework of an EKF. EKF is a commonly utilized method for state estimation in nonlinear systems, characterized by its process of consecutively linearizing the nonlinear system at each step. The advantage of using EKF is its computational efficiency [25].

As explained earlier, when there is a model update or re-identification, the dimension of the new reduced-order model may not remain the same. Therefore, the standard EKF cannot be applied. To address this issue, we develop an adaptive EKF based on the adaptive model. Let us define the discrete reduced-order model (4.10) with the corresponding output equation as follows:

$$\begin{aligned} \xi^{(m)}(t_{k+1}) &= f_{rd}^{(m)}(\xi^{(m)}(t_k), u(t_k), w(t_k)) \\ y(t_k) &= C_r^{(m)}\xi^{(m)}(t_k) + v(t_k) \end{aligned} \tag{4.11}$$

where $v(t_k)$ denotes the measurement noise at time t_k, $C_r^{(m)} = CU^{(m)}$.

Adaptive EKF design

There are two steps in EKF: the prediction and the update of the states. At the sampling time t_k, in the prediction step, the adaptive EKF first predicts ξ^m at t_k based on the state estimate at t_{k-1} and the reduced model as follows:

$$\hat{\xi}^{(m)}(t_{k|k-1}) = f_{rd}^{(m)}(\hat{\xi}^{(m)}(t_{k-1}), u(t_{k-1}), w(t_{k-1})) \tag{4.12}$$

where $\hat{\xi}^{(m)}(t_{k|k-1})$ represents the reduced state prediction at time instant t_k using an initial guess $\hat{\xi}^{(m)}(t_0)$ or the previously estimated reduced state $\hat{\xi}^{(m)}(t_{k-1})$. The evolution of the variance of the reduced states is also propagated based on the reduced model:

$$P_r^{(m)}(t_{k|k-1}) = A_d^{(m)}(t_{k-1})P_r^{(m)}(t_{k-1})A_d^{(m)}(t_{k-1}) + Q_r^{(m)} \tag{4.13}$$

where $P_r^{(m)}$ and $Q_r^{(m)}$ denote covariance matrices for the state and process disturbance in reduced form, respectively, and $A_d^{(m)}(t_{k-1}) = \left.\frac{\partial f_{rd}^{(m)}}{\partial \xi^{(m)}}\right|_{\hat{\xi}^{(m)}(t_{k-1})}$ is the state-transition matrix obtained by linearizing the nonlinear reduced model at the estimated state at t_{k-1}. Note that if P and Q are the covariance matrices of the state and process disturbance for the original system, $P_r^{(m)} = U^{(m)T}PU^{(m)}$ and $Q_r^{(m)} = U^{(m)T}QU^{(m)}$.

At the sampling instant t_k, once the measurement $y(t_k)$ is available, it is used to update the predictions generated in the prediction step. In the update step of EKF, the current reduced state estimates $\hat{\xi}^{(m)}(t_k)$ is calculated based on the predicted value $\hat{\xi}^{(m)}(t_{k|k-1})$ as follows:

$$\hat{\xi}^{(m)}(t_k) = \hat{\xi}^{(m)}(t_{k|k-1}) + K_r^{(m)}(t_k)(y(t_k) - C_r^{(m)}\hat{\xi}^{(m)}(t_{k|k-1})) \tag{4.14}$$

where $\hat{\xi}^{(m)}(t_k)$ is the estimated reduced state at time t_k, and $K_r^{(m)}(t_k)$ is the correction gain which minimizes a *posteriori* error covariance using the observation innovation $y(t_k) - C_r^{(m)}\hat{\xi}^{(m)}(t_{k|k-1})$. The correction gain can be determined as below:

$$K_r^{(m)}(t_k) = P_r^{(m)}(t_{k|k-1})C_r^{(m)T}(R + C_r^{(m)}P_r^{(m)}(t_{k|k-1})C_r^{(m)T})^{-1} \tag{4.15}$$

Here, R represents the covariance matrix for observation noise. Additionally, the process involves updating the covariance matrix of the system state in the following equation:

$$P_r^{(m)}(t_k) = (I_{r_m} - K_r^{(m)}(t_k)C_r^{(m)})P_r^{(m)}(t_{k|k-1}) \tag{4.16}$$

where $P_r^{(m)}(t_k)$ denotes the *posteriori* error covariance matrix related to the state estimation error at time t_k and I_{r_m} is denoted an identity matrix specific to the m^{th} reduced model with size r_m. It is noted that $P(t_0)$, Q, and R are tuning parameters for the EKF. The state estimate at the time t_k is $\hat{\hat{x}}$ and evaluated from:

$$\hat{\hat{x}}(t_k) = U^{(m)}\hat{\xi}^{(m)}(t_k) \tag{4.17}$$

Information exchange during model transition

When there is a model update, the information in the EKF estimator of the previous model should be smoothly transferred to the EKF built on the new reduced model. The information transfer is performed by mapping all the information back to the full state system and then projecting it to the new reduced model. Consider that we need to transfer the information of the EKF based on the m^{th} reduced model to the EKF built on the $(m+1)^{th}$ model. The following steps are performed:

- Mapping the estimated reduced state and state covariance to the full order state and covariance: $\hat{\hat{x}} = U^{(m)}\hat{\xi}^{(m)}$, $P = U^{(m)}P_r^{(m)}U^{(m)T}$.
- Projecting the full order information to the new reduced model using the new projection matrix: $\hat{\xi}^{(m+1)} = U^{(m+1)T}\hat{\hat{x}}$, $P_r^{(m+1)} = U^{(m+1)T}PU^{(m+1)}$.

4.2.3 Design of the error metric e_L and implementation algorithm

The reduced model update or re-identification is triggered by an error metric e_L, which is evaluated every sampling time. Specifically, at t_k, the estimated state $\hat{\tilde{x}}(t_k)$ is considered as the initial condition. The trajectory of the system state over the next N_{f_d} steps is predicted both based on the original model of (4.8) and the current reduced model. It is assumed that the irrigation amounts of the next N_{f_d} steps are known, which is typically the case in agricultural irrigation. After generating the predictions, the deviation between the trajectory generated by the reduced model and the trajectory of the original model is calculated by percentage mean absolute error (% MAE) and used as the error metric e_L. The design of e_L is inspired by the work of [101] and is shown below:

$$e_L(t_k) = \frac{100}{N_x} \sum_{j=1}^{N_{f_d}} \sum_{i=1}^{N_x} |\tilde{x}_i(t_{k+j}) - x_i(t_{k+j})| \tag{4.18}$$

where $\tilde{x}_i$ and x_i denote the predictions of i^{th} state element using the reduced model and the original model respectively. A prediction horizon N_{f_d} is considered. A model reduction is triggered if $e_L(t_k)$ exceeds the predefined threshold th_e. A new reduced model is generated as discussed earlier.

In order to optimize computational resources and minimize the need for frequent model generation, the Algorithm 2 employs the metric e_L and $\dot{e}_L$ as the criteria for model adjustments. The current value of $\dot{e}_L(k)$ is derived from the moving average calculated over the preceding ten consecutive differences of e_L when e_L is on the rise. This is used to filter out model disturbances that could affect model change decisions. To generate a snapshot matrix for cluster formation, the algorithm applies an initial soil content, irrigation u_{ir}, rain, evapotranspiration ET, crop coefficient (K_c) to Richards equation (4.8). The estimation process takes place in a reduced space, with the necessary variables transformed using the projection matrix. By utilizing the EKF design, from (4.12) to (4.17), the algorithm computes estimates for soil moisture. By incorporating these strategies, the algorithm achieves efficient computation, reduces model updates, and provides accurate estimations of soil moisture.

The proposed reduced-order state estimation is summarized in the following Algorithm (2):

Algorithm 2 Error-triggered reduced EKF algorithm

```
Initialization Initial guess $\hat{x}_0$, $P(0) > 0$, $Q, R > 0$, $th_e$ ,$th_C$, $m = 0$, and $N_{f_d}$
for $k = 0 \dots n$ do
    if $\big(e_L(k) > th_e \wedge \min \dot{e}_L(k) \geq 0.05 \vee k == 0\big)$ then
        $m \leftarrow m + 1$
        Apply $\hat{\bar{x}}_{k|k}$ and input to generate $\mathscr{X}_m$ and obtain $U^{(m)}$
        Convert $\hat{\xi}^{k|k} = U^{(m)}\hat{\bar{x}}_{k|k}$, $P_r^{(m)}(k) = U^{(m)T}P(k)U^{(m)}$ and $Q_r^{(m)} = U^{(m)T}QU^{(m)}$
    end if
    Obtain measurements $y(k)$
    Calculate current reduced estimates $\hat{\xi}^{(m)}(k|k)$
    Convert to actual state $\hat{\bar{x}}_{k|k} = U^{(m)}\hat{\xi}^{(m)}(k|k)$
    Compute $e_L(k)$ and $\dot{e}_L(k)$
end for
```

4.3 Simulation results for a small agricultural field

In this section, we demonstrate the effectiveness of the proposed adaptive model reduction and estimation in the state estimation of a small demo farm. The field is located to the east of Lethbridge, Alberta, Canada, with geographical coordinates latitude 49.68 N and longitude -112.73 W. The field consists of a central pivot irrigation system, and the crop being grown is sugar beet in its growing stage.

The soil depth considered in this study is 0.4 m, which is discretized evenly into 12 nodes. The field has a radius of 50 m, which is discretized radially into 20 nodes

Figure 4.7: Demo farm in Lethbridge

and azimuthally into 40 nodes. The surface boundary condition is characterized by the Neumann boundary condition, which specifies the flow of water out of the system, and the bottom boundary condition is specified as free drainage. In this work, we assume the use of point sensors to measure the soil pressure head values at 90 selected nodes of the field at each sampling time, including surface nodes and nodes at various depths. The total number of nodes in the field is 9600. The discretized diagram of the research farm and the farm itself are shown in Figure 4.2 and Figure 4.7 respectively.

The soil parameters in the field are known to vary across the field, as shown in Figure 4.8. In order to simulate the system, information about evapotranspiration (ET), crop coefficient (K_c), irrigation, and rain are used, as shown in Figure 4.9. The initial actual soil moisture x_0 in pressure head is simulated to be a distinct value for each quadrant of the field, namely, -3.5, -4.0, -2.7, and -1.5 m.

We consider three different estimation schemes to illustrate the proposed approach. Scheme I involves using the proposed adaptive modeling and EKF. In Scheme II, a reduced EKF is designed based on a single non-adaptive reduced model. This approach assumes that all information, such as irrigation, Evapotranspiration (ET),

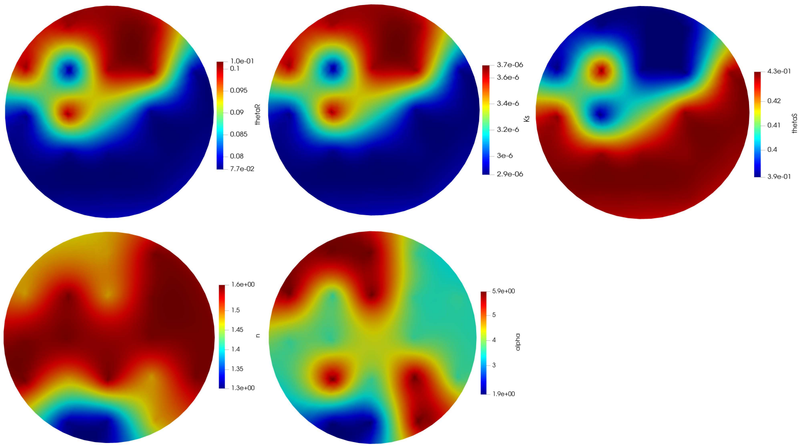


Figure 4.8: Different soil parameters used in the simulation

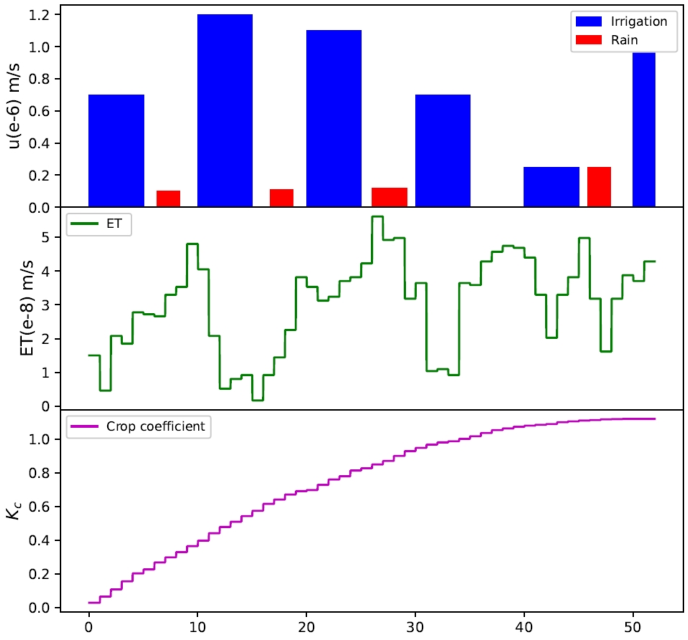


Figure 4.9: Input (irrigation, ET and rain) and K_c to the system

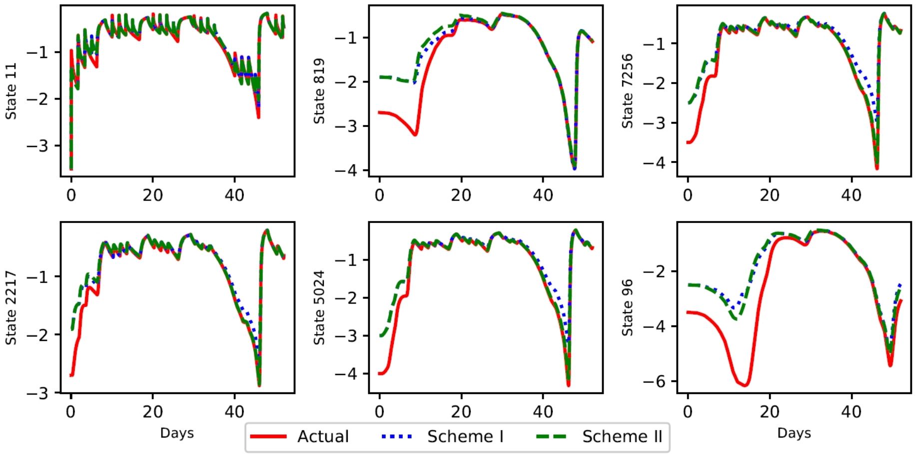


Figure 4.10: Actual state trajectories and state prediction of schemes I and II

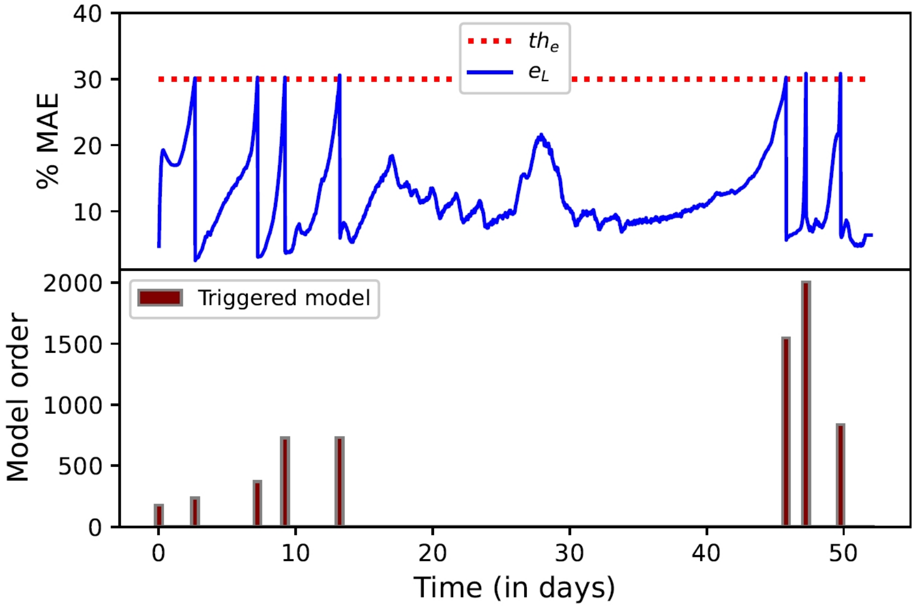


Figure 4.11: The proposed error-triggered adaptive EKF Scheme I

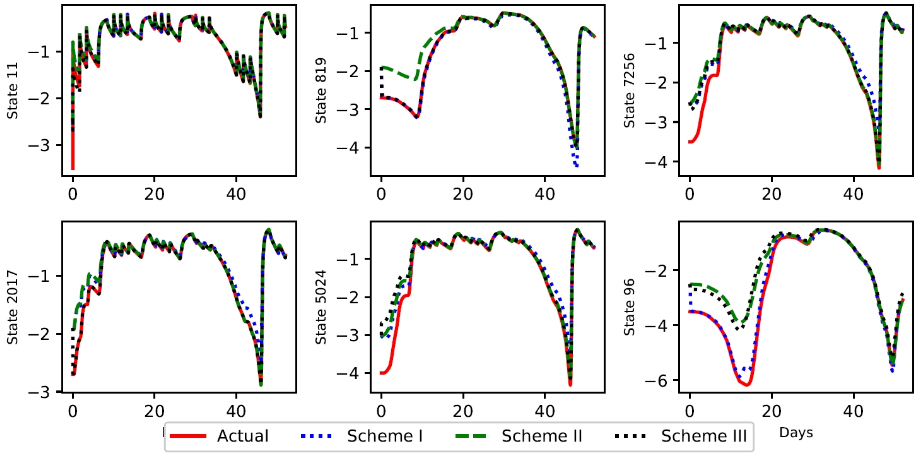


Figure 4.12: Actual state trajectories and the state estimation of all schemes

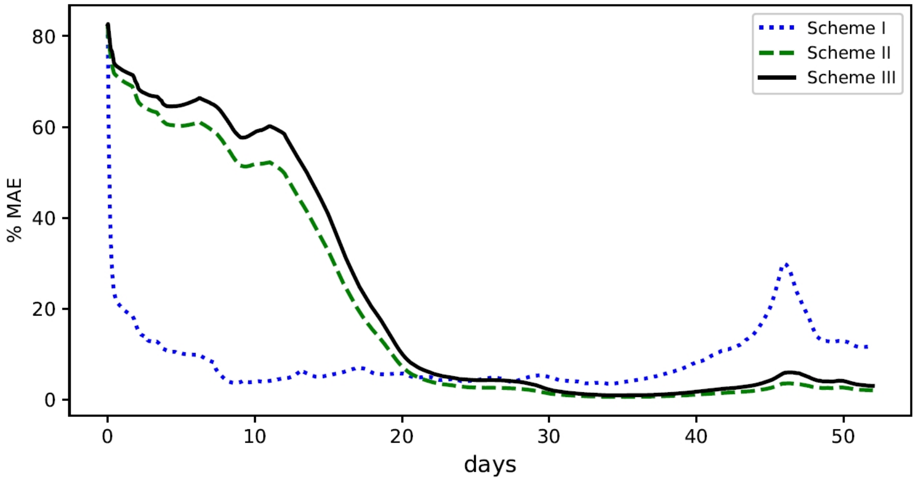


Figure 4.13: State estimation performance of the schemes

and rain, is available for the entire growing season and is used to generate trajectories for the non-adaptive model creation. Finally, Scheme III involves a full-order estimator design based on a regular EKF, in which the original model is used. These three schemes will be compared to evaluate the proposed adaptive EKF's performance and determine its effectiveness in accurately estimating the entire field's soil moisture.

The simulation runs for a total of 52 days, and the initial guess for the soil moisture $\hat{x}(0)$ is set to -2.5, -3.0, -1.9, and -2.0 m for the four quadrants, which differ from the actual values of soil moisture $x(0)$. The units for the soil pressure head values are measured in meters (m). The covariance matrices Q and R are defined as identity matrices with diagonal elements of $Q = 1.0 \times I_{Nx}$ and $R = 0.08 \times I_{Ny}$, respectively. The initial state covariance matrix P has entries of 5×10^{-5} for all off-diagonal elements and a value of 1.0 for all diagonal elements. The measurement noise is considered normally distributed with a zero mean and a standard deviation of 0.1. The sampling time is $\Delta = 30$ min.

Note that the initial reduced model is generated based on the $\hat{x}(0)$ as the initial state and not the actual initial state $x(0)$. The $x(0)$ is only utilized to generate the actual trajectories of the system. In Figure 4.10, the trajectories of actual states and predicted states of Schemes I and II are shown for a few selected states. The constant thresholds th_C for cluster generation are 0.2 and 0.5 in creating the new reduced model for Scheme I and Scheme II, respectively. Additionally, the non-adaptive reduced model used in Scheme II is found to be accurate enough at the order of 4708.

The proposed estimation approach, scheme I, employs a threshold $th_e = 0.3$ for the error-triggered criterion with $N_{f_d} = 48$. As depicted in Figure 4.11, the bottom plot displays the instances where error-triggered model re-identification occurred, with varying model orders. The re-identification is prompted by the error indicator e_L exceeding th_e. At the start and end of the simulation, frequent model changes were observed, with the highest model order of 2005 corresponding to high values of K_c and ET and a rapid decrease in the soil moisture.

In Figure 4.12, the state estimates for all schemes are shown to have excellent agreement with the system's actual trajectories. Figure 4.13 further highlights the percentage of the mean average error (% MAE) between the actual and EKF estimated states for all schemes. Note that the proposed scheme I converge much faster compared with the other two schemes may be due to the increased degree of observability of the estimation problem. In the proposed scheme, the number of measurements is kept the same (90) but the number of states that need to be estimated is significantly less. This helps the estimation scheme to converge faster. But at the same time, in the proposed approach, since each reduced model uses fewer nodes and the model mismatch error accumulates, the accuracy after the convergence is slightly poorer as can be seen from Figure 4.13. But with the triggered model adaptation, scheme I can maintain the estimation error within the pre-determined threshold through model adaptation. This can also be seen in Figure 4.13. The tuning parameters for re-identification of the reduced models include the fixed time N_{f_d}, threshold for cluster generation th_C, and error threshold th_e. These also provide more flexibility in tuning the estimation performance of the proposed approach.

The proposed approach is also much more computationally efficient. The error-triggered adaptive EKF scheme I, which includes adaptive clustering, model reduction, and recursive calculation, takes approximately 3 seconds to evaluate at each sampling time. Whereas, the non-adaptive reduced EKF scheme II utilizes a high number of reduced states, and takes longer for estimation, around 9 seconds per sampling time. It is worth noting that the assumption of having all information beforehand in the non-adaptive reduced EKF is not realistic, and subsequently, the system is prone to process and weather disturbances. Implementing the classical EKF based on the actual full-scale nonlinear model scheme III with 9600 discretization nodes is computationally challenging, and it takes approximately 35 seconds for each step. With an increase in the number of system states, the evaluation time for calculating the large state transition matrix A_d also experiences a substantial rise, rendering the estimation process computationally intractable, as indicated in [51].

The small field is considered to provide a comparison with the full estimator which is intractable for a large system. The conference paper [103] has succeeded in establishing that the reduced approach converges much faster compared to the full estimator, primarily due to the enhanced degree of observability in the estimation problem.

4.4 Simulation results for a large agricultural field

In this section, we present a demonstration of the efficacy of the proposed adaptive model reduction and state estimation in the soil moisture estimation of a large-scale agricultural farm.

4.4.1 Simulation settings

The investigated field is a circular field measuring 26.4 hectares, situated at the Alberta Irrigation Technology Center in Lethbridge, southern Alberta at latitude 49.72 N and longitude 112.80 W. The research farm, highlighted in green in Figure 4.14, serves as the study area. Within this expansive field, a five-span center-pivot irrigation system is deployed, covering a lateral distance of about 290 meters. Furthermore, the field is facilitated with a commercially viable irrigation system with variable rates. The soil hydraulic parameters used in this study are provided in Figure 4.15.

In this study, the soil depth is set at 0.4 m and is evenly divided into 12 discrete nodes. The field has a radius of 290 m, which is discretized radially into 30 nodes

Figure 4.14: Investigated area in Lethbridge, Alberta, Canada.

and azimuthally into 68 nodes. In this work, we assume the use of point sensors to provide measurements of the soil pressure head values at 90 selected nodes of the farm at each sampling time, including surface nodes and nodes at various depths. Readers are encouraged to explore the optimal placement of soil sensors for accurate state estimation [50] and sensor selection for continuous monoclonal antibodies production [104]. The total number of discretized nodes (states) in the research field is 20400. To simulate the system, information about ET, K_c, irrigation, and rain is used, as shown in Figure 4.16. The initial actual soil moisture x_0 in pressure head (m) is simulated to be a distinct value for each quadrant of the field, namely, -13.5, -14.0, -12.7, and -11.5 m.

The simulation runs for a total of 30 days with a sampling time $\Delta = 30$ min, and the initial soil moisture $x(0)$ is unknown and taken as an initial guess as -10.0, -12.0, -9.0, and -14.0 m for the four quadrants, which differ from the actual values of soil moisture $x(0)$. The units for the soil pressure head values are measured in meters (m). The covariance matrices R and Q are defined as diagonal matrices with

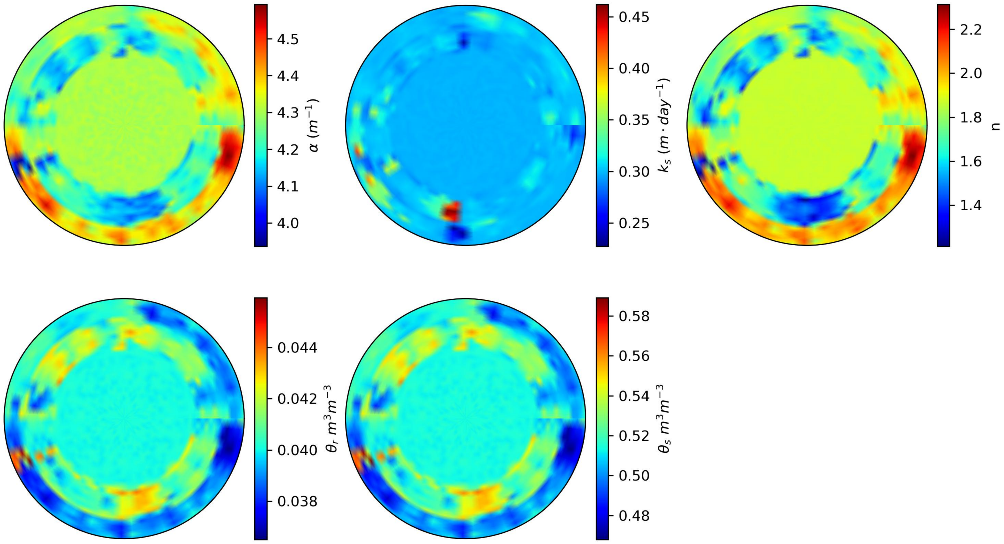


Figure 4.15: Different soil parameters (α, K_s, n, θ_r, and θ_s) used for the study

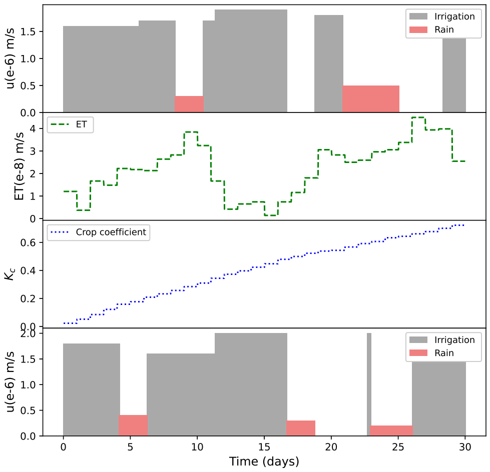


Figure 4.16: Input (irrigation, ET, K_c, rain) of the system: real-time irrigation and rain information (top) and scheduling and forecast error irrigation and rain disturbance (bottom)

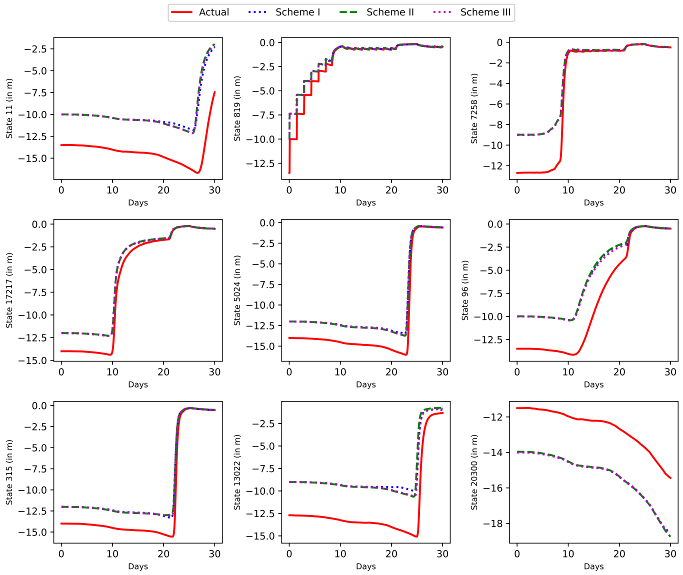


Figure 4.17: Actual state trajectories and state prediction of schemes I, II, and III for large field

diagonal elements of $R = 0.08 \times I_{N_y}$ and $Q = 1.0 \times I_{N_x}$, respectively. I_{N_y} and I_{N_x} are the identity matrices of dimension N_y and N_x respectively. The initial state covariance matrix P has entries of 5×10^{-5} for all off-diagonal elements and a value of 1.0 for all diagonal elements. The normal distribution of the measurement noise and process disturbance are considered with a mean of zero. Their respective standard deviations are 0.8 and 1×10^{-7}.

We explore three estimation schemes to demonstrate the effectiveness of the proposed approach. Scheme I involves using the proposed adaptive modeling and EKF. In Scheme II, a reduced EKF is designed based on a single non-adaptive reduced model for the total simulation time. To reduce input disturbance in Scheme II, a reduced EKF is designed based on a time-triggered adaptive reduced model in Scheme III. We assume that all the information, such as irrigation, ET, K_c, and rain, is available for the total operation duration accurately. It's important to highlight that the traditional EKF is not included in this study due to its computational infeasibility

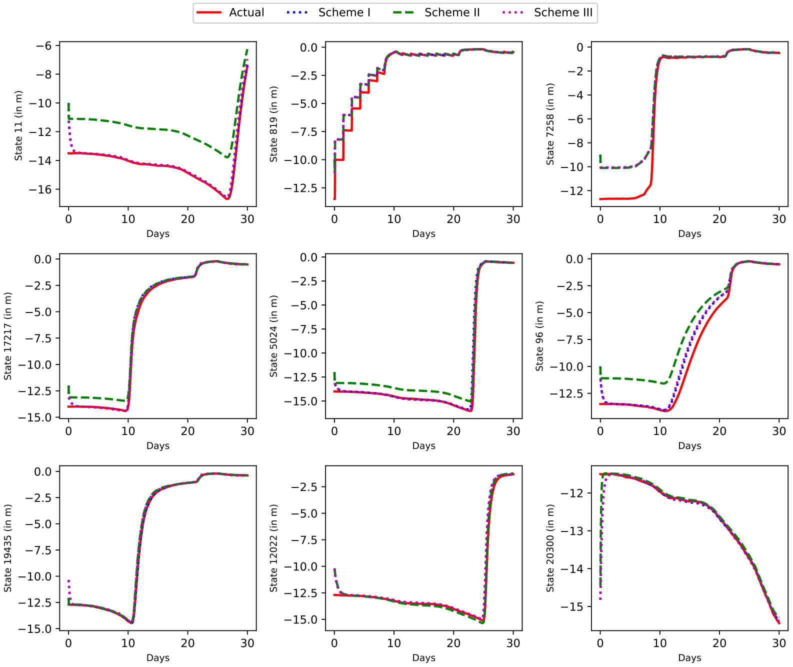
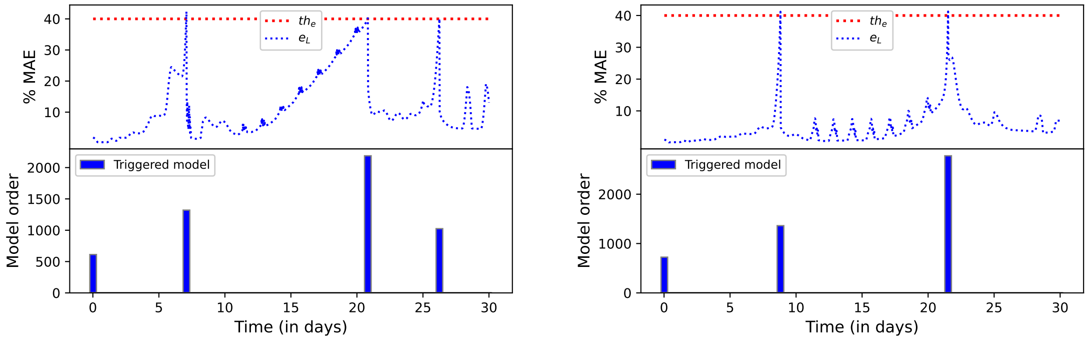
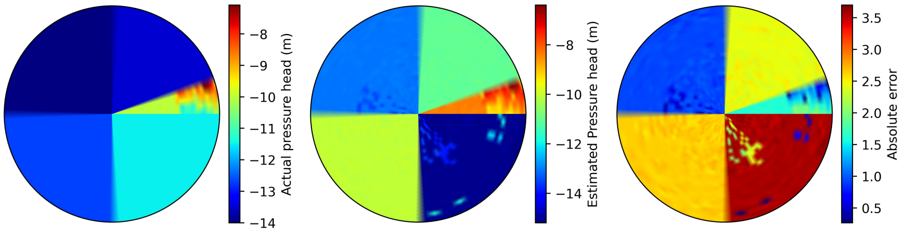


Figure 4.18: Actual state trajectories and the state estimation of schemes I, II, and III for large field

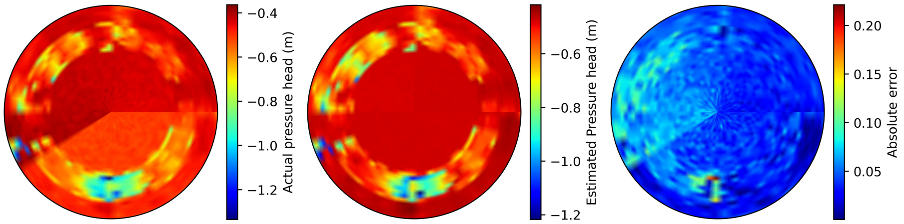


(a) $N_{f_d} = 250$ **(b)** $N_{f_d} = 350$

Figure 4.19: The proposed error-triggered adaptive EKF Scheme I: change of % MAE for the reduced model (top) and model re-identification instances with model orders (bottom)



(a) Initial soil water map on day one



(b) Final soil water map on the last day

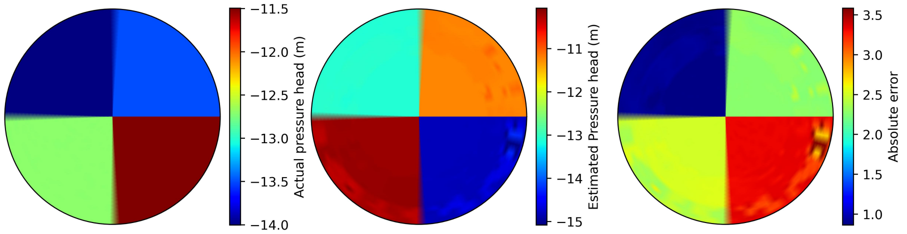
Figure 4.20: Soil moisture pressure head distribution for actual, estimated, and absolute error between actual and estimated states for surface (From left to right)

limitation for the large-scale system, as discussed in [103].

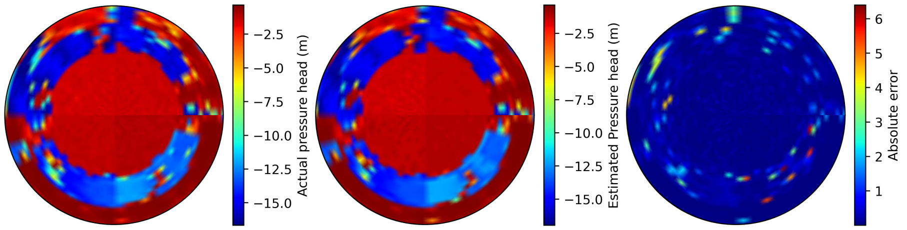
Note that the initial reduced model is created by applying the initial guess $\hat{x}(0)$ as the initial state and not the actual initial state $x(0)$ which is unknown for all the schemes. The purpose of the estimator is to estimate the initial soil moisture of the agricultural system. Considering $x(0)$ as the actual state at $t_0 = 0$, actual trajectories are generated to compare the reduced models' performance for the entire growing season. In Figure 4.17, the trajectories of actual states and predicted states of all the schemes are shown for a few selected states. Similarly, as shown in Figure 4.18, the actual and estimated state trajectories of all the schemes are shown for the same states.

4.4.2 Estimation accuracy

In Figure 4.19, Scheme I, the proposed approach, employs specific thresholds for the error-triggered criterion: $th_e = 40$ for the error threshold and $th_C = 1.0$ for

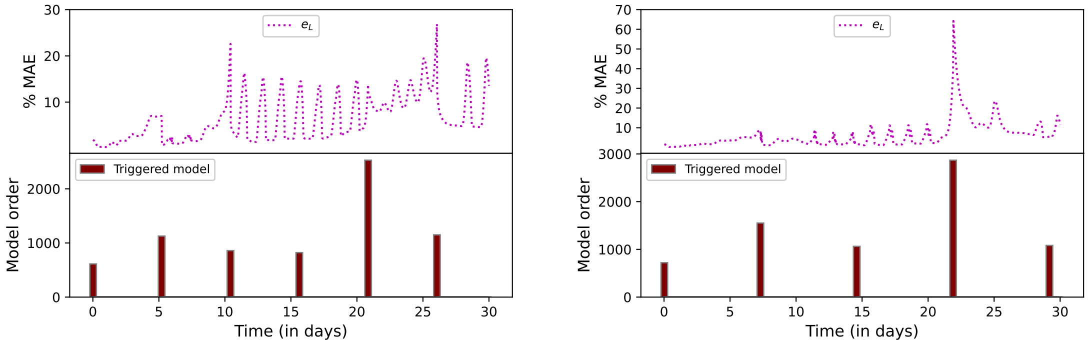


(a) Initial soil water map on day one



(b) Final soil water map on the last day

Figure 4.21: Soil moisture pressure head distribution at 30 cm for actual, estimated, and absolute estimation error between actual and estimated states (From left to right)



(a) $N_{f_d} = 250$ **(b)** $N_{f_d} = 350$

Figure 4.22: The time-triggered adaptive EKF Scheme III: change of % MAE for the reduced model (top) and model re-identification instances with model orders (bottom)

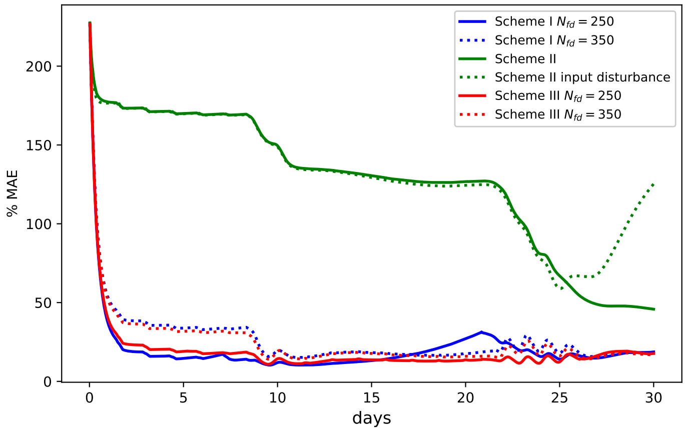


Figure 4.23: State estimation performance (% MAE) of all the schemes for large field

the cluster threshold. Two different horizon lengths of $N_{f_d} = 250$ and $N_{f_d} = 350$ are utilized in this scheme. The bottom plots in Figure 4.19 provide insights into instances of reduced model re-identification and the corresponding reduced model dimensions for the error-triggered approach. Simultaneously, the top plots show the variation of % MAE for the reduced model. The re-identification process is initiated when the error indicator e_L surpasses the threshold th_e. Initially, at the start of the season, the reduced-order model indicates a low model order, indicating a homogeneous field condition. However, as time progresses and the system receives input, the model order gradually increases, suggesting a shift in field dynamics. Figure 4.19a showcases an additional model identification due to considerations of a short horizon. Notably, the highest model order is observed for the longer horizon at 2778, while the short horizon exhibits a model order of 2188. Figure 4.20 presents the surface soil pressure head distribution of the actual, estimated, and absolute error between the actual and estimated states for both the 1^{st} and 30^{th} days, considering the longer horizon. From Figure 4.20(a) absolute error plot at each node, it is evident that the absolute estimation error on the first day is significantly reduced to the last day, as shown in Figure 4.20(b). Similarly, in Figure 4.21, the soil moisture maps at a depth of 0.3 m are presented, and the absolute error between actual and estimated states is decreased over the estimation period except for a few locations.

In the centralized model, Scheme II, the cluster generation threshold th_C is set to 5.0, resulting in a model order of 1974 which can capture the dynamics for the total simulation time. As shown in Figure 4.23 the solid green line, the estimation barely converges, with a long convergence delay leading to significant estimation errors. Furthermore, the centralized model is highly vulnerable to forecast and input disturbances, as the assumption that all information is perfectly known is unrealistic in real-world scenarios. Although it is assumed that information is exactly known, in real scenarios, this can be way deviated from the actual forecast and input information. Therefore, using input and rain with disturbance shown in the Figure 4.16 bottom plot, it is observed that the estimation diverges, provided as in the green dotted line in the Figure 4.23. Thus, the adaptation of system dynamics to the reduced model is necessary to address the problem. To improve this, we propose a time-triggered adaptive model reduction that can be used for comparison.

Scheme III, which demonstrates time-triggered model reduction, is illustrated in Figure 4.22 for two distinct values of N_{f_d}. To mitigate weather forecast and irrigation decision uncertainties, we explored the time-triggered algorithm with structural similarities to the centralized method. In this design, the generated trajectories are updated at shorter fixed intervals, differing from the centralized method, which takes into account the entire operational dataset. The performance of the time-triggered estimator is depicted in red in Figure 4.23. For $N_{f_d} = 250$ and $N_{f_d} = 350$, the highest model orders are 2528 and 2868, respectively. When comparing the two triggered EKF designs, error-triggered and time-triggered, the estimation error remains similar for a specific horizon N_{f_d} considered; however, the time-triggered model reduction exhibits a higher number of model changes. To investigate the plotted results, Figure

4.23 displays how Scheme I with $N_{f_d} = 250$ achieves a reduction in estimation error around the 22^{nd} day, showcasing the benefits of model reduction.

In Figure 4.18, the actual trajectories and estimated trajectores for all schemes are presented to have excellent agreement with the system's actual trajectories. Figure 4.23 further highlights the percentage of MAE between the actual and EKF estimated states for all schemes. Note that the proposed triggered methods converge much faster compared with the centralized model due to the increased degree of observability of the estimation problem. In the proposed triggered schemes, the number of measurements is kept the same (90) but the number of states that need to be estimated is significantly less. This helps the estimation scheme to converge faster. However, in the proposed approach, since each reduced model uses fewer nodes and the model mismatch error accumulates, the accuracy after the convergence is slightly poorer as can be seen from Figure 4.23. With the adoption of the time-triggered model scheme, Scheme I effectively keeps the estimation error within the predefined threshold, ensuring reliable performance. The tuning parameters for re-identification of the reduced models include the fixed time N_{f_d}, threshold for cluster generation th_C, and error threshold th_e. These also provide more flexibility in tuning the performance of the adaptive estimator in the proposed approach.

4.4.3 Computation times

The proposed approach in this study offers significant improvements in computational efficiency. All the computational simulations were performed on a computer loaded with Intel(R) Core(TM) $i7 - 8700$ CPU operating at 3.2GHz and 24.0 GB RAM. The proposed error-triggered adaptive EKF, scheme I, which incorporates adaptive clustering, model reduction, and recursive calculation, requires approximately 5 seconds to evaluate at each sampling time in Table 4.1. On the other hand, the time-triggered EKF scheme II, which experiences frequent model changes, takes

Scheme	Simulation time (Sec)
Scheme I $N_{f_d} = 250$	5.01
Scheme I $N_{f_d} = 350$	4.93
Scheme II	2.20
Scheme III $N_{f_d} = 250$	5.52
Scheme III $N_{f_d} = 350$	5.48

Table 4.1: Average computational speed per iteration comparison of different schemes

a slightly longer estimation time of around 5.5 seconds per sampling time considering the total simulation time. It is important to note that the assumption of having all information beforehand in the non-adaptive reduced EKF is unrealistic, leaving the system vulnerable to process and weather disturbances. When the field dynamics exhibit variations, the centralized model ends up with a high reduced-order model. Consequently, as the number of reduced system states increases, the evaluation time for calculating the large state transition matrix, denoted as A_d, also increases exponentially. While Scheme II may require slightly less time, using a centralized model is not practical due to its limitations.

4.5 Summary

In this chapter, we addressed the problem of state estimation of a large-scale agro-hydrological system equipped with a central pivot. We employed a finite difference method to discretize the Richards equation, which describes the dynamics of the agro-hydrological system within a cylindrical coordinate framework. We designed a reduced state estimator using an error-triggered reduction method approach. To validate the effectiveness of our approach, we implemented the proposed adaptive state estimation in a small demo farm and a large agricultural field. The reduced state estimator demonstrated a satisfactory performance. In particular, accurate estimates of the soil moisture were obtained and improved computational speed was achieved as compared to the centralized method.

Chapter 5

An application of adaptive model reduction to soil moisture and its hydraulic parameter estimation via remote sensing

In this chapter, the work presents a real case study of the adaptive model reduction in data assimilation of an agro-hydrological system. While the preceding chapter showcased the effectiveness of the triggered adaptive model reduction approach in state estimation, this chapter delves into its application with rotating measurements, exploring the simultaneous estimation of both state and parameters. An information fusion system for rotating microwave measurements was considered in [72]. In [72], the cylindrical coordinate version of the Richards equation, the EKF, and measurements from microwave remote sensors were used to estimate soil moisture and hydraulic parameters of fields equipped with center-pivots. The current study aims to improve the computational efficiency and accuracy of the soil moisture estimates by implementing a reduced-order approach. In Chapter 4, we proposed an error-triggered model reduction for state estimation which is designed specifically for point sensors that are continuously available [105]. However, in this chapter, we discuss the challenges associated with the discontinuous data acquisition and the benefits of the proposed model reduction. The intermittent nature of remote sensors necessitates adaptive model reduction, triggered upon measurement availability. Simultaneously, we only develop new reduced models when the prediction performance falls short of the existing reduced model, minimizing the need for frequent model adjustments. The results demonstrate the effectiveness of the proposed model reduction method in improving the computational efficiency and accuracy of soil moisture estimation in a real large-scale agricultural field.

5.1 Model description

As discussed in Chapter 4, the spatial discretization of the agro-hydrological model establishes a continuous-time state-space model in the following form:

$$\dot{x}(t) = f(x(t), p(t), u(t)) + w(t) \tag{5.1a}$$

$$y(t) = C(t)x(t) + v(t) \tag{5.1b}$$

When the soil moisture measurements are obtained from the microwave radiometers mounted on the center-pivot, the C matrix in the output equation changes over time. As the center-pivot irrigates the field, the microwave radiometer rotates and provides the measure of the soil moisture content shown in Figure 5.1. Generally, the radiometers measure the moisture content of areas that have not yet been irrigated. Thus, we receive the measurements of different locations of the field which are changing during the rotation cycle of the center-pivot. At any particular sampling time, the measurements are available for N_r nodes at maximum according to the location of the center-pivot. In this work, we also consider estimating soil hydraulic parameters. Owing to the heterogeneity of soil texture, each discrete node is assigned its unique set of parameters. In this study, we consider five soil hydraulic parameters: K_s, θ_s,

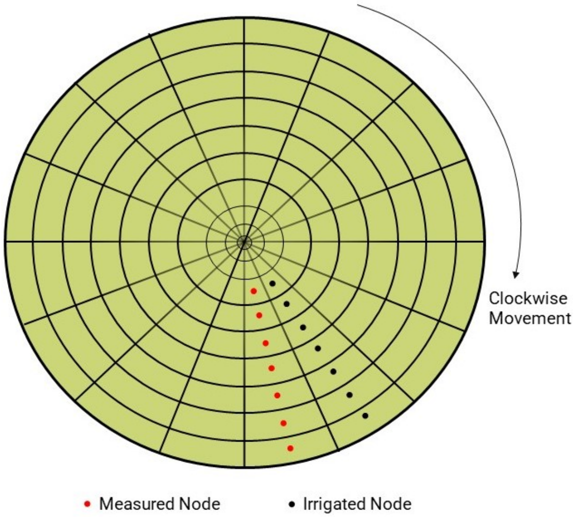


Figure 5.1: A top view of microwave remote sensors on center-pivots

θ_r, α, and η, for each node. We also assume that the same surface soil properties exist at various depths within the soil.

5.1.1 Problem formulation

Microwave sensors can operate round-the-clock, weather-independent, ensuring a continuous stream of real-time data for farmers. While these remote sensors play a significant role, they have limitations in providing comprehensive and precise soil moisture information, mainly due to their constrained spatial and temporal coverage of the field. The time required for a center-pivot to complete one rotation cycle is typically two to three days which delays the generation of a single water content map. That's why leveraging an agro-hydrological model becomes essential to generate an accurate soil water map across the entire area.

Data assimilation methods serve as valuable tools for the continuous estimation of soil moisture through the fusion of information, delivering essential information for efficient field management [57, 72, 106]. These studies explore the application of microwave remote sensing-based data assimilation methods, specifically focusing on constructing water content maps suitable for closed-loop irrigation implementation. In [52], it performs simultaneous state and parameter estimation of the Richards equation by identifying and selecting a subset of the most estimable parameters. The selected parameters that have the greatest impact on the system behavior or model predictions, along with system states, are augmented for simultaneous estimation to maximize information from observations. Nevertheless, given the high dimensionality of the system, these methods, while promising, require substantial computational resources and simulation time. Additionally, the augmentation of discrete states and parameters leads to a considerably large system that poses numerical challenges. Scaling these methods for use in large-scale fields presents a significant challenge, necessitating further studies to make them more practical and efficient for broader agricultural applications.

5.2 Proposed approach

In the previous chapter 4, we introduce error-triggered adaptive model reduction aiming at enhancing soil moisture estimation accuracy. It is focused on the point sensors which provide the soil water head for the corresponding sensor locations and also at different soil depths continuously at each sampling time. Additionally, the work assumes a priori knowledge of parameter values. Drawing the inspiration, we expand our investigation to incorporate microwave remote sensing and simultaneous estimation to further refine the field soil moisture estimation.

In Figure 5.2, we depict the essential steps of the data assimilation process within an adaptive reduced model framework, aiming to acquire a comprehensive understanding of the proposed method. Initial soil moisture and hydraulic parameters are determined by interpolating soil moisture observations and prior parameter knowledge. The availability of measurements, acting as an event-trigger, guides

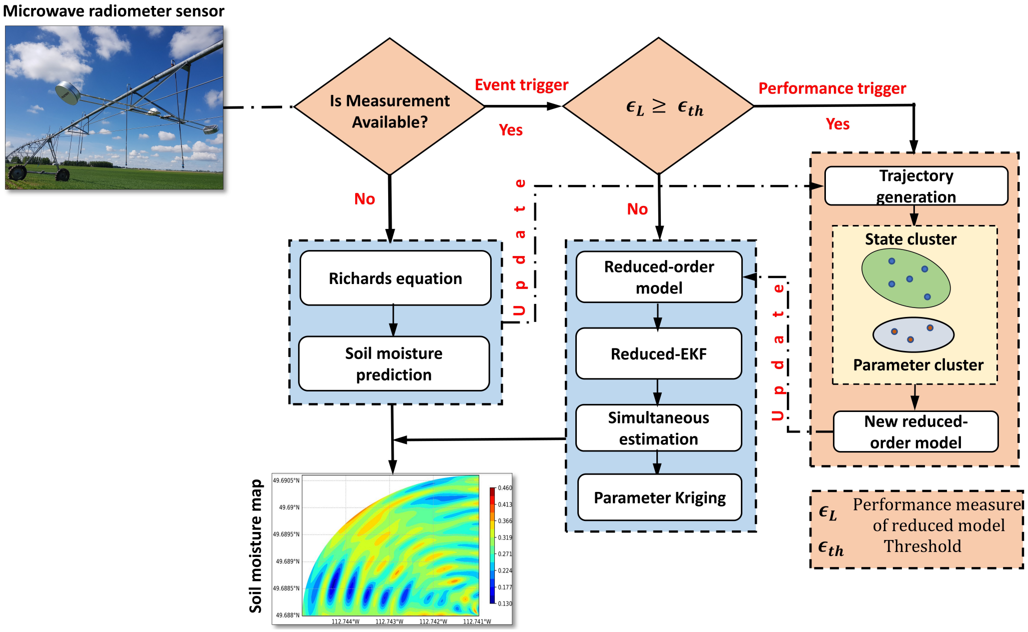


Figure 5.2: The proposed model reduction and simultaneous state and parameter estimation

the decision-making process, determining whether to pursue open-loop predictions employing the Richards equation or opt for simultaneous estimation using a reduced-order model. Sequentially, the prediction performance of the existing reduced model is assessed, and only the reduced-order model exhibiting inadequate performance is replaced with a new model that is evaluated in real-time. The newly generated trajectory is initiated based on the current estimates or predicted soil moisture information. Subsequently, clusters are formed using the hierarchical clustering method, separately identifying clusters for both states and parameters. These trajectories are developed with consideration of anticipated irrigation, rainfall, evapotranspiration (ET), and crop coefficient (K_c) over three days. The reduced model performance is compared with the Richards equation. Within the figure, ϵ_L serves as the performance trigger metric, while ϵ_{th} represents the allowable performance threshold, which will be discussed in more detail later. Utilizing the reduced-order model, we construct a reduced-order estimator through an EKF algorithm, enabling simultaneous soil moisture and parameter estimation. Parameters chosen for estimation are those associated with nodes where measurements are available during the sampling period, as discussed in [52]. To update the unselected parameters, estimated parameters are interpolated using the Kriging method. In this section, we explain the methodology of the adaptive model reduction method and the design of the estimator.

Remark 4 *This work is mainly application-oriented and addresses a specific problem of rotating and intermittent measurements, drawing inspiration from our prior research outlined in chapter 4. The model reduction employs the same clustering technique, with distinct clusters generated for soil moisture and parameters, which are subsequently combined to form a reduced augmented system. The key deviation in the methodology lies in the incorporation of parameter clusters.*

We briefly discuss the methodology of the adaptive model reduction. We assess measurement availability and calculate ϵ_L. If the metric ϵ_L exceeds the threshold ϵ_{th}, the new reduced model is identified and replaced with the existing reduced model. To identify the reduced model at the beginning or to re-identify a reduced-order model, the same steps are followed and explained briefly.

The open-loop simulation is performed using the current state estimates at t_k, $\hat{x}(t_k)$, and the system equation (5.1) with prescribed irrigation actions for the following N_{f_d} steps.

$$\mathscr{X}_m = [x(t_k)\ x(t_{k+1})\ \dots\ x(t_{k+N_{f_d}})]$$

where $\mathscr{X}_m \in \mathbb{R}^{N_x \times N_{f_d}}$ is the state snapshot matrix for the m^{th} model reduction assuming that there were $(m-1)$ number of model reductions performed before t_k. The set of parameters for generating clusters is as follows:

$$\mathscr{P}_m = [K_s(t_k), \theta_s(t_k), \theta_r(t_k), \alpha(t_k), \eta(t_k)]$$

where $\mathscr{P}_m$ is the current parameter estimate at t_k.

Once the sets $\mathscr{X}_m$ and $\mathscr{P}_m$ are established, clustering is independently applied to each. Following the methodology outlined in Chapter 4, we implement the agglomerative hierarchical clustering technique to group similar trajectories in $\mathscr{X}_m$ and parameters in $\mathscr{P}_m$ into clusters. Parameter clustering follows similar steps as state clustering. The thresholds for these clusters, which are critical for maintaining the accuracy of the reduced model, are denoted as th_x for states and th_p for parameters. For each variable set, the projection matrices are constructed using the Petrov-Galerkin projection method, resulting in $\mathscr{U}^{(m)} \in \mathbb{R}^{N_x \times r_m}$ for state variables and $\mathscr{V}^{(m)} \in \mathbb{R}^{N_p \times p_m}$ for parameter variables, where r_m and p_m represent the dimensions of the reduced state and parameter spaces, respectively.

The parameters have a different range, variability, and influence on the model compared to the states. This separate clustering ensures that it is performed for the type of data it processes, potentially leading to more accurate and topology-preserved clusters. The clustering-based model reduction employs projection methods to create cost-effective, simplified models of the nonlinear state space (5.1). For a nonlinear system, the projection matrices cannot be explicitly computed without going back to the original model (5.1). This contrasts with linear systems, where reduced models often feature a more compact and computationally straightforward representation. Consequently, in (5.1), the correlation between reduced parameters and states is inherent within the system itself, eliminating the need for a separate characterization of the correlation.

As previously mentioned, the microwave radiometers measure soil moisture content during the center-pivot rotation. Consequently, the measured soil moisture content locations of the field move along with the pivot. The potentially estimable parameters corresponding to the measured states may change as the pivot rotates. When the center-pivot rotates, the estimable parameters are grouped in one vector and assigned to $p^e \in \mathbb{R}^{N_{pe}}$ and the rest of the parameters of the nodes not measured are collected in $p^{ne} \in \mathbb{R}^{N_{pn}}$ such as total parameters $p = [p^e \;\; p^{ne}]$. The states and parameters are augmented in order to estimate simultaneously and it takes the form: $x_a(t) = [x(t) \;\; p(t)]$. The augmentation of the simultaneous state and parameter model is performed by adding the state space equation (4.8) with the following equation for the soil parameters:

$$\dot{p}(t) = 0 \tag{5.2}$$

where the soil hydraulic parameters are assumed to be constant during the study. The augmented form for simultaneous parameter and state estimation is shown below:

$$\dot{x}_a(t) = f_a(x_a(t), u(t)) + w_a(t) \tag{5.3a}$$

$$y(t) = C_a(t)x_a(t) + v_a(t) \tag{5.3b}$$

where subscript a to each variable and function denotes the augmentation. The soil parameters p^{ne} which are not estimable are replaced with the soil nominal values during the state estimation of the augmented state x_a. Consequently, the primarily estimable variables are $x(t)$ and $p^e(t)$. The total augmentation is necessary for updating the covariance matrix in EKF, simplifying the calculation of the simultaneous estimation. At last, the Kriging interpolation approach is applied to evaluate the entries of p^{ne} after obtaining p^e from the simultaneous estimation. For the augmented reduced model we define the augmented projection $U \in \mathbb{R}^{(N_x+N_p)\times(r_m+p_m)}$ after m^{th} model reduction as a block diagonal matrix (*blkdiag*) as shown below:

$$U^m = blkdiag\{[\mathscr{U}^m], [\mathscr{V}^m]\} \tag{5.4}$$

where a block diagonal matrix is defined as a matrix where the diagonal contains sub-matrices, and the off-diagonal blocks are filled with zeros. The continuous time m^{th} reduced model is represented as follows:

$$\dot{\xi}^{(m)}(t) = f_r^{(m)}(\xi^{(m)}(t), u(t), w_a(t)) \tag{5.5}$$

where ξ is the reduced state and f_r is the reduced-order model expressed as $\xi^{(m)}(t) = U^{(m)T}x_a(t)$ and $f_r^{(m)} = U^{(m)T}f_a$ respectively. Note that the predicted state x_a from the state space (5.3) can be approximated based on mapping $\tilde{x}_a(t) = U^{(m)}\xi^{(m)}$. The continuous-time reduced state space is discretized to implement a discrete-time EKF. Hence, the discrete-time reduced-order model takes the form as follows:

$$\xi^{(m)}(t_{k+1}) = f_{rd}^{(m)}(\xi^{(m)}(t_k), u(t_k), w_a(t_k)) \tag{5.6a}$$

$$y(t_k) = C_r^{(m)}(t_k)\xi^{(m)}(t_k) + v_a(t_k) \tag{5.6b}$$

where f_{rd} is the discrete-time function of the reduced model and $C_r^{(m)} = C_aU^{(m)T}$. The state information of the former is mapped to the full-order system and then projected onto the latter using an updated matrix, ensuring consistency between model transitions.

EKF is designed based on the reduced augmented model for the data assimilation design. CasADi [107], an open-source library for numerical optimization, is used to symbolically solve the reduced model and the required Jacobian matrices. The EKF algorithm needs to initialize with a prior guess of the unknown initial augmented state $\hat{x}_a(t_0)$ and its augmented covariance matrix $P_a(t_0)$. Q_a is the augmented covariance matrix for the system disturbance w_a. R is the covariance matrix of the measurement noise. It is noted that $P_a(t_0)$, Q_a, and R are the tuning parameters for the algorithm. The conversion to reduced space and further details on this process can be found in Chapter 4.

5.2.1 Design of the triggered metrics

The event-trigger is a binary decision: when measurements are available, the EKF will be executed; otherwise, the open-loop prediction will be computed. This event is defined by the availability of measurements, which triggers the use of the EKF. When the EKF is active, it's essential to assess the performance of the reduced model to minimize any model-plant mismatch. A sequential triggering approach is particularly well-suited for microwave measurements. The reduced model update or re-identification is triggered by an error metric ϵ_L, which is evaluated for every event-trigger. While investigating the model performance, the reduced model is compared with the Richards equation by checking the prediction accuracy.

$$\epsilon_L(t_k) = \frac{1}{N_d} \frac{100}{N_x + N_p} \sum_{j=1}^{N_d} |\tilde{x}_a(t_{k+j}) - x_a(t_{k+j})| \tag{5.7}$$

where $\tilde{x}_a$ and x_a denote the augmented state and parameters from the open-loop predictions from the reduced model and the Richards equation respectively. The term N_d denotes the prediction horizon.

5.3 Application to real-data: Case study analysis

We showcase the effectiveness and performance of the adaptive model reduction, and simultaneous state and parameter estimation method by implementing to the microwave remote sensor data acquired from an agricultural field with a center-pivot irrigation system.

5.3.1 Study area

The investigated agricultural field is located at lat 49.72 N and long 112.80 W in Alberta, Canada with an average elevation of approximately 888 m, and the study area is shown in Figure 5.3. The field primarily consists of clay, loam soil, and sand. The field is a circular field facilitated with a five-span center-pivot system. The microwave radiometers are installed on the center-pivot system to obtain the measurement of soil moisture at various points during the rotation cycle and the soil moisture observations utilized in the work are collected in the summer of 2022. The soil parameters in the field are known to vary across the field as shown in Figure 5.5.

Figure 5.3: Investigated area in Lethbridge, Alberta, Canada

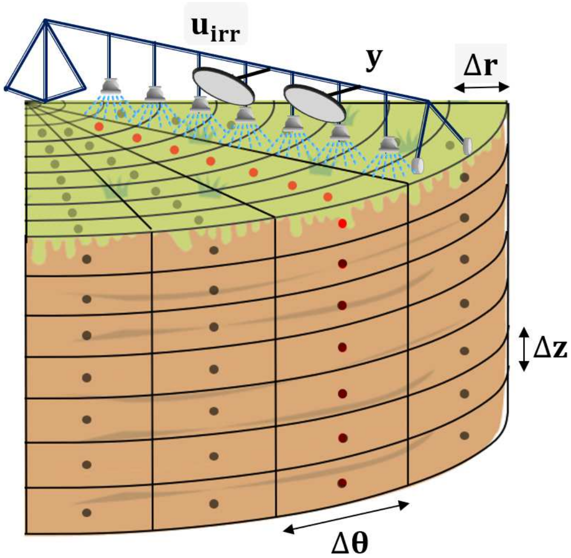
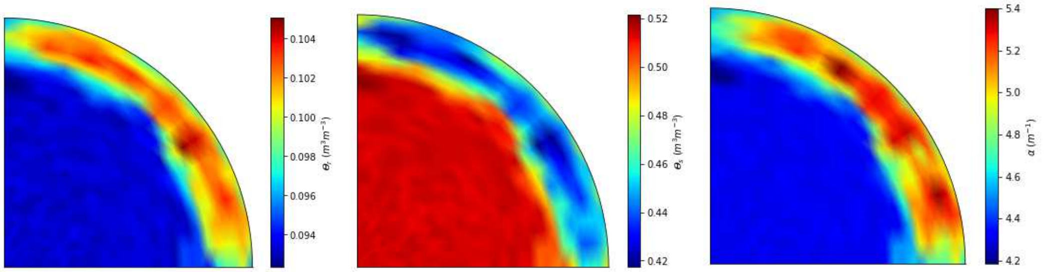
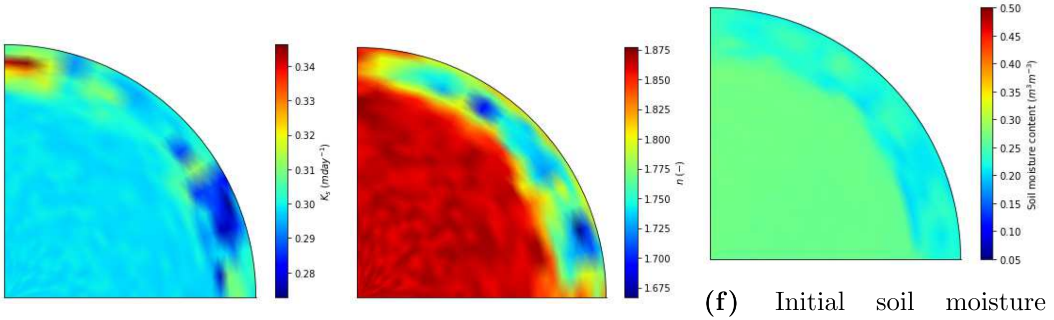


Figure 5.4: A schematic diagram of the quadrant 1 where red nodes have the soil moisture measurements

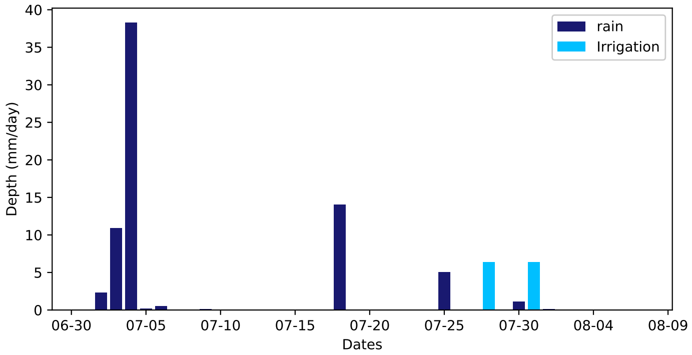


(a) $\theta_r(m^3m^{-3})$ **(b)** $\theta_s(m^3m^{-3})$ **(c)** $\alpha(m^{-1})$

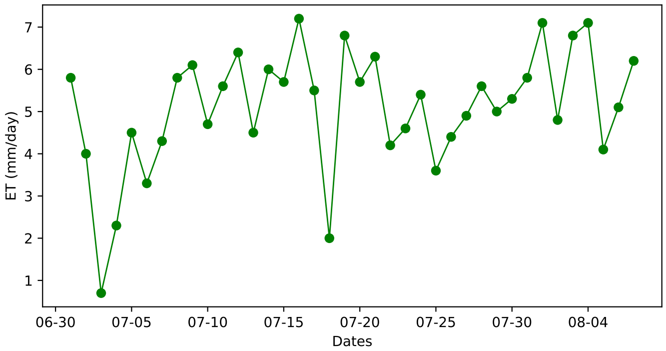


(d) $K_s(mday^{-1})$ **(e)** $n(-)$ **(f)** Initial soil moisture $\theta_m(m^3m^{-3})$

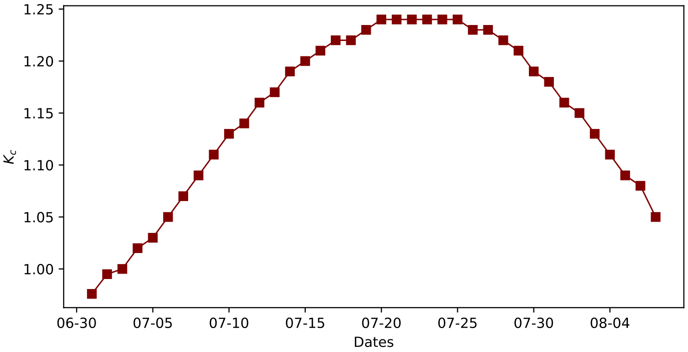
Figure 5.5: Spatial distribution of the nominal soil hydraulic parameters (θ_r, θ_s, α, K_s, and n) and initial guess for soil moisture content for quadrant 1



(a) Daily precipitation rates



(b) Daily reference evapotranspiration (ET)



(c) Daily crop coefficient

Figure 5.6: Inputs to the model for the entire simulation period

A quadrant of the field has been chosen, delineated by a solid blue line in Figure 5.3, where wheat was in the growth stage.

5.3.2 Numerical modeling of Richards equation

The quadrant of the agricultural field under investigation encompasses approximately a radius of 290 m, a depth of 0.32 m, and spans an angle of 0.5π rad. The boundary conditions used in Richards equation for a quadrant can be found in [57]. For discretizing the quadrant, we have divided the radius, angle, and depth into 30, 17, and 10 equally spaced sectors, respectively, as illustrated in Figure 5.4. Hence, the total discretized nodes are $N_x = 5100$ (number of states) and the total number of parameters is $N_p = 2550$ giving in total of 7650 estimates. The spatial discretization is selected based on the resolution of the microwave radiometer and additional discretization to the axial nodes is required due to having a minimal effect on the system trajectories. We consider a timestep of 30 min the temporal discretization of the reduced and Richards model. An explicit numerical method is utilized to approximate the time and space derivative. The system trajectories are assessed for different timesteps, and a suitable timestep is chosen. In solving the Richards equation, we employ a symbolic CasADi integrator, specifically the Runge-Kutta (RK4) method, which is an explicit technique featuring third-order finite difference [107]. This symbolic approach with CasADi simplifies the computation of the necessary Jacobian matrices for implementing the EKF. All the simulations are performed on a computer equipped with an Intel(R) Core(TM) $i7-8700$ CPU at 3.2GHz and 24.0 GB RAM.

5.3.3 Data preparation

The soil moisture content measurements are considered from July 1^{st}, 2022 to August 8^{th}, 2022 in the investigated quadrant in this study. On average, the microwave radiometers provide soil moisture measurements twice in an hour. The simultaneous state and parameter estimation is trained by $\frac{4}{5}^{th}$ portion of the measurements and the rest of the measurements are utilized for validation. For the data preparation, the initial raw soil moisture measurements undergo a sequence of data preprocessing procedures. These steps include sorting the measurements by date and time, grouping them by quadrants, removing outliers, and mapping the measurements to the nodes of the model. The data pre-processing and preparation are explained explicitly in our previous work [72].

Daily reference evapotranspiration (Figure 5.6b), and daily precipitation data (Figure 5.6a) are crucial model inputs for the Richards equation and reduced model and are acquired from the website of Alberta Information Service. Also, the irrigation information for the entire crop season (Figure 5.6a) is sourced from the Alberta Irrigation Center. To determine the crop coefficient values (Figure 5.6c) for wheat throughout the simulation period, we employ a daily temperature-dependent equation as outlined in [72].

5.3.4 Adaptive model reduction

The proposed adaptive model reduction strategy utilizes distinct thresholds for cluster generation. Specifically, we set th_x to 0.01 for the state and th_p to 0.0005 for the parameters. As the model undergoes a change, a trajectory spanning three days ($N_{fd} = 144$) is generated to account for the extended prediction horizon. The criteria for performance assessment consider the prediction for the following day ($N_d = 48$). To set the error threshold, we fixed a threshold for the reduced model's performance at $\epsilon_{th} = 3$. To determine these parameters, several values are assessed, as their selection significantly influences the frequency of model reductions.

5.3.5 Reduced simultaneous estimation

The reduced estimator is initialized with initial guess $\hat{x}_a(t_0)$ which is set with limited knowledge about the actual condition $x_a(t_0)$ shown in Figure 5.5f. The initial condition is interpolated using the Kriging interpolation method based on the available observation. The initial states and nominal hydraulic parameters map are depicted in Figure 4.15 for quadrant 1. In our prior study [72], we determined the parameter values for quadrant 4. In this current work, we make an informed initial guess for these parameters by leveraging the similarities in soil type. The initial covariance matrix $P_a(t_0)$ is a matrix with higher diagonal elements than the off-diagonal elements to accommodate the uncertainty in the unknown initial guess $\hat{x}_a(t_0)$. In particular, the state elements are initialized in a range of $0.1 - 1700$, while for the parameters, 5 is assigned to $P_a(t_0)$. The EKF is typically capable of handling this selection of large initial variance. The covariance matrices of process disturbance $Q = 0.008 I_{N_x+N_p}$ and measurement noise $R = 0.01 I_{N_y}$ are tunable where I is the identity matrix. However, our experience reveals that dynamically adjusting these matrices significantly improves the performance of the EKF. We designed Q and R varying with the sampling time i, which ranges from 1 to 48 for a day, depending on the poor performance of cross-validation. We set $Q = 0.008 \times (\frac{i}{2}) \times I_{N_x+N_p}$ and $R = (\frac{0.01}{i}) \times I_{N_y}$ when the mean absolute error between the validation data and the estimates is greater than 0.02. Notably, it places increasing importance on measurements as time progresses, enhancing the ability of the estimators.

5.3.6 Performance evaluation criteria

The proposed reduced simultaneous estimation employs different ways of cross-validation. First of all, the available measurements are randomly divided into assigned portions of training and validation sets to evaluate the accuracy of state estimates. The state estimates are cross-referenced with the measurements from the validation dataset. Essentially, this validation approach aims to assess the accuracy of the soil moisture estimates. The performance of the proposed approach is quantified using the Normalized Root Mean Square Error (NRMSE). It is calculated comparing between the soil moisture content $\hat{y}(t_k)$ obtained from estimation and sensor measured soil moisture content $y(t_k)$ and defined as follows:

$$NRMSE = \frac{1}{(y_{max} - y_{min})} \sqrt{\frac{\sum\limits_{k=1}^{N_y^v} (y(t_k) - \hat{y}(t_k))}{N_y^v}} \tag{5.8}$$

where N_y^v represents the number of measurements within the validation dataset, and y_{max} and y_{min} are the highest and lowest soil moisture content present within the validation set, respectively. In addition to the overall NRMSE, we also use the performance measure to monitor the performance of the estimation whenever the measurement is available. Achieving a smaller NRMSE value reflects an improved alignment between the estimated and measured values.

Furthermore, we conduct a thorough comparative analysis in the context of the soil moisture observations. We utilize the estimated variables to evaluate model predictions up to the time when the latest measurements become available, allowing the model predictions to be directly compared with the actual measurements in both cases. Since we trust the actual measurements more than the models, this analysis provides valuable insights into the performance of each case when aligned with the true values. We assess this performance using NRMSE daily.

Lastly, the Absolute Difference ($AD = |\hat{x}_a(t_k) - \hat{x}_r(t_k)|$) is a metric used to quantify the discrepancy between estimated variables at the end of the season. The estimated variables $\hat{x}_a$ and $\hat{x}_r$ are obtained from full estimation and reduced estimation, respectively. Note that the AD measures the difference of an estimated variable, which is evaluated for each node in quadrant 1. By introducing both AD for individual nodes and NRMSE for the overall performance, we can have a more comprehensive assessment of the proposed method.

5.3.7 Results and performance evaluation

Figure 5.7 provides a comprehensive summary of the results found through the proposed adaptive model reduction approach. In Figure 5.7a, we represent instances where measurements are available i.e. events are triggered with solid circles, followed by error-triggers depicted as solid squares with the measurement count for training the EKF. The first trigger corresponds to the initial reduced model. The occurrence of an event-trigger, followed by an error-trigger, signals the necessity for a model adjustment. Out of fifteen available days with soil measurements, error-triggers are activated seven times.

Figure 5.7b shows the error metric that triggers the re-identification of the new model. It is evident that the threshold is violated six times, resulting in the re-identification of reduced model on each occasion. The tuning parameters for the cluster generation are set to ensure that the new model's metrics fall within the specified threshold. The same cluster configuration may not be universally effective for different scenarios. However, it is important to ensure that the new model's performance consistently aligns with the threshold set to maintain the plant model mismatch within acceptable limits.

Notably, we observe frequent model changes, especially when the system incorporates feedback from microwave sensors in Figure 5.7c. The maximum model order is recorded as 3507 (3260, and 247), while the minimum model order stands at 1969 (1619, and 350). The initial model order appears lower in comparison to the end-season, which is a reflection of the different soil moisture conditions. This difference is most evident when examining 3D soil moisture estimation data further emphasizing the importance of conducting a 3D analysis. This variation in the field corresponds to changes in system dynamics due to different environmental conditions, soil properties, and crop water intake, leading to alterations in the infiltration process.

In Table 4.1, the simulation time required for estimation at each sampling point varies from 30 seconds for model orders around 2000 to 100 seconds for model orders around 3500. However, the full-order system with 7650 variables takes more than twice as long compared to the highest-ordered reduced model. As the number of the system states increases, the evaluation time for the large state-transition matrix, A_d, also grows exponentially. For the same reason, employing the EKF in the large-scale field becomes nearly intractable with the available computational resources. While comparing the computation time, as shown in Table 4.1, we can observe a significant improvement in computational efficiency of over 50% attributed to the reduced estimator. On average, the reduced estimator can provide assimilated soil moisture in just 3.5 s, while the full estimation takes approximately 7.4 s. Despite the additional online simulation time required for model changes and performance metrics, this extra time remains significantly lower compared to the estimation time needed for the full estimator. This is why the reduced approach highly realistic and feasible approach for application in the large-scale field.

In the context of assessing computational performance, it is essential to consider the accuracy in comparison to the actual measurements, as shown in Table 5.2. The cross-validation Figures 5.8a and 5.9a compare the actual measurements with the estimated states derived from both the reduced estimator and the full-state estimator. The results clearly indicate that both state estimates are in good agreement with the actual measurements and a closer alignment to the actual data towards the end of the season. Following the convergence of the variables, it is anticipated that the estimation accuracy will surpass the predictive accuracy of the field model. Noted, these plots do not represent the trajectory instead compare the actual samples during the period. We achieve overall NRMSE values of 0.1385 and 0.1244 for the reduced and full order estimations, respectively. However, it highlights a discrepancy of approximately 12 % in the cross-validation results, which corresponds to model approximation errors. When examining the daily NRMSE in Figures 5.8b and 5.9b, we realize that a limited number of measurements can result in decreased accuracy. Also, a similar error pattern is observed on most days, except for the 21^{st} and the following day, which could be due to a potential model-plant mismatch. Further, in Table 5.2, the narrower range of reduced estimates compared to the full estimation highlights the relatively consistent and reliable results offered by the reduced estimator.

Figure 5.10, the plot illustrates the NRMSE between the open loop predictions obtained using the estimated variables and the actual measurements. For example, we utilize observations on 22^{nd} July to estimate the variables and perform the open loop simulation based on those estimates until the next available measurement on 25^{th} July. The comparison shows the collective performance of the models and estimators and both are closely aligned towards the end of the season. Also, the error consistently fall within a similar range, affirming the consistency and reliability of the reduced model.

Finally, in Figures 5.11, 5.12 and 5.13, the spatial distribution of soil moisture yield consistently within the same range, with an average absolute difference of approximately 0.07 m^3m^{-3}. Similarly, in Figure 5.14 and 5.15, the estimated parameter K_s is converged, and average values are in the same range around 0.305 $mday^{-1}$ [72].

While the results are promising, it's worthwhile to address the demanding requirements and explore other potential modifications. This proposed method involves tuning various model parameters, such as th_x, th_p, selecting the type of error measure, setting the error threshold ϵ_{th}, and fine-tuning parameters associated with EKF. Uncertainty stems from the initial soil moisture condition which is often not fully known and can affect the reliability of the estimation process. In extreme soil moisture conditions, such as excessively dry or wet field conditions, numerical issues

Table 5.1: Computational speed per iteration on an average for Data assimilation and Estimation steps

	Data assimilation (sec)	Estimation (sec)
Reduced estimation	3.5 ± 0.4	$30 - 100$
Full estimation	7.4 ± 0.6	220 ± 40

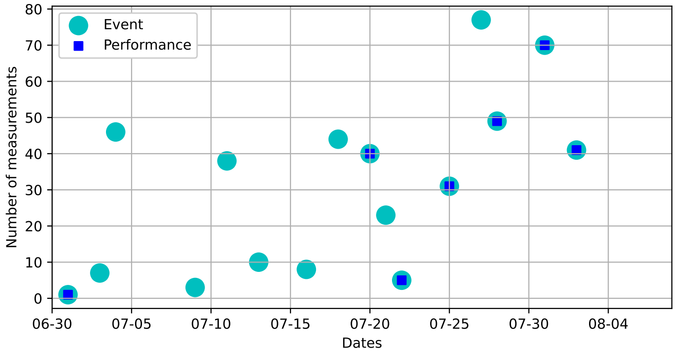
Table 5.2: Overall NRMSE comparison on the validation sets

	Overall NRMSE	Max value	Min value
Reduced estimation	0.1385	0.465	0.121
Full estimation	0.1244	0.557	0.102

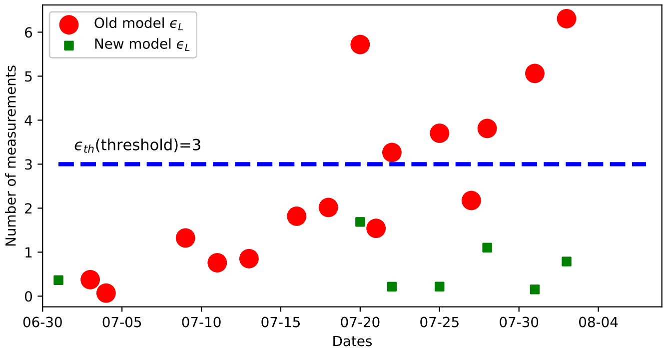
can surface, necessitating the use of implicit solvers to ensure stable solutions. To extend the reliable estimation of hydraulic parameters, integrating data from point sensors into the proposed framework is crucial.

5.4 Summary

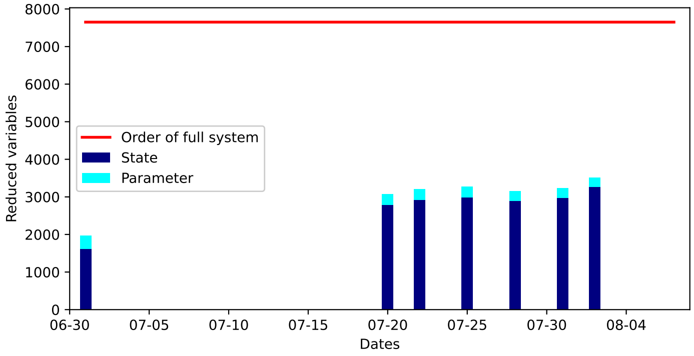
In this chapter, we address the challenge of higher dimensionality in the agro-hydrological system concerning state and parameter estimation. We apply a reduction approach to simultaneously estimate soil moisture and soil hydraulic parameters for large-scale agricultural systems with varying soil conditions, using soil moisture measurements collected from microwave radiometers mounted on center-pivot irrigation systems. We leverage a sequential trigger approach, an event-trigger, and an error-trigger to systematically deduce the reduced model. The proposed approach presents adaptive modeling of the investigated field using the cylindrical form of the Richards equation and integrating remote sensor-provided soil moisture data to the extended Kalman filtering technique. The EKF is performed in a reduced space, and



(a) Triggered error corresponds to event-trigger with measurements for training of EKF

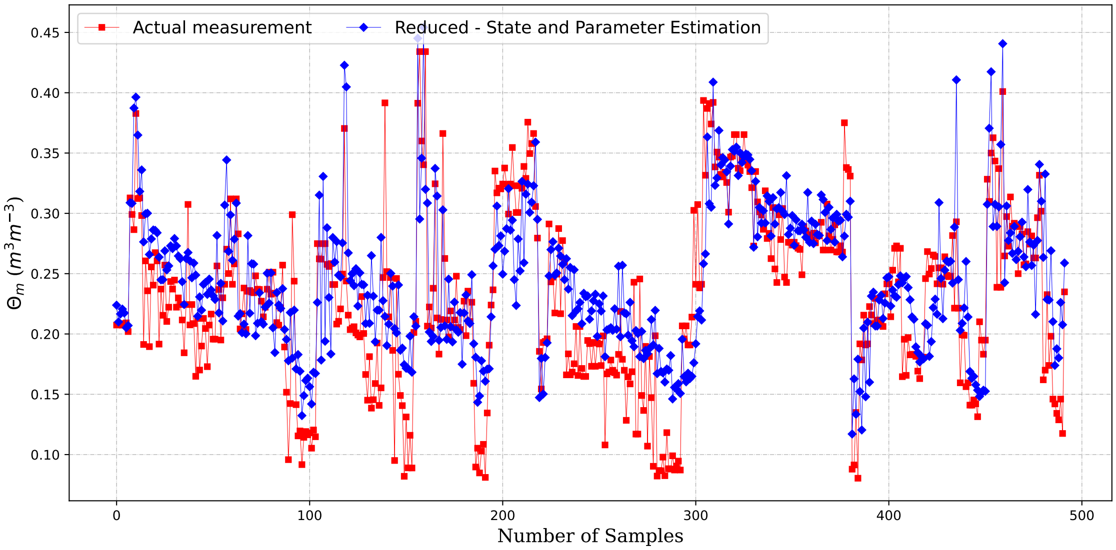


(b) Performance measure for existing model and newly generated model

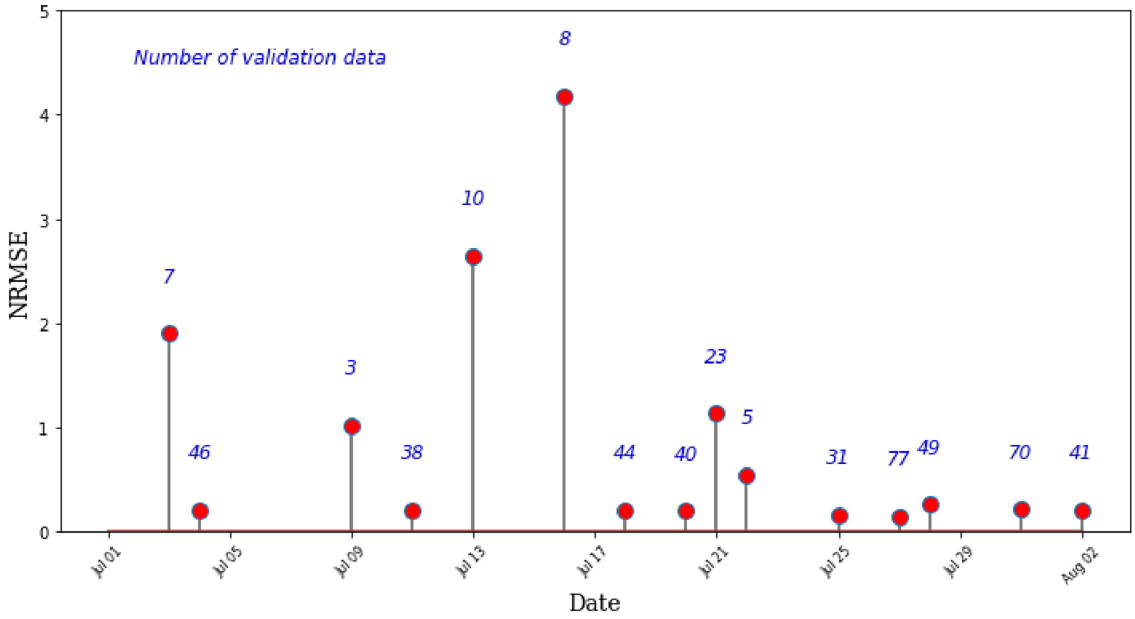


(c) Reduced state and parameter order of adaptive models

Figure 5.7: Proposed sequential triggered approach

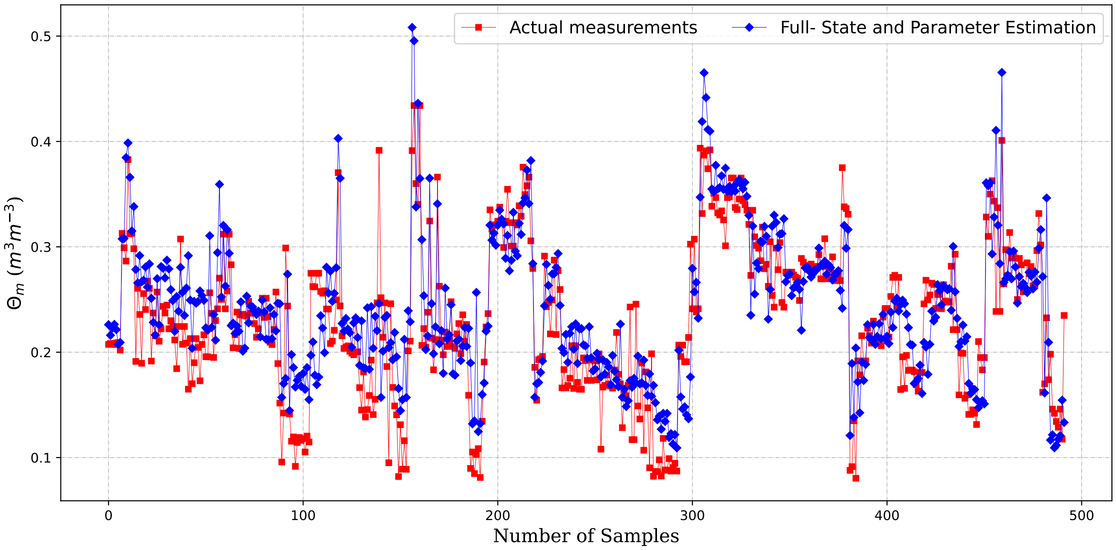


(a) Actual measurements vs reduced estimation for the entire season with overall NRMSE= 0.1385

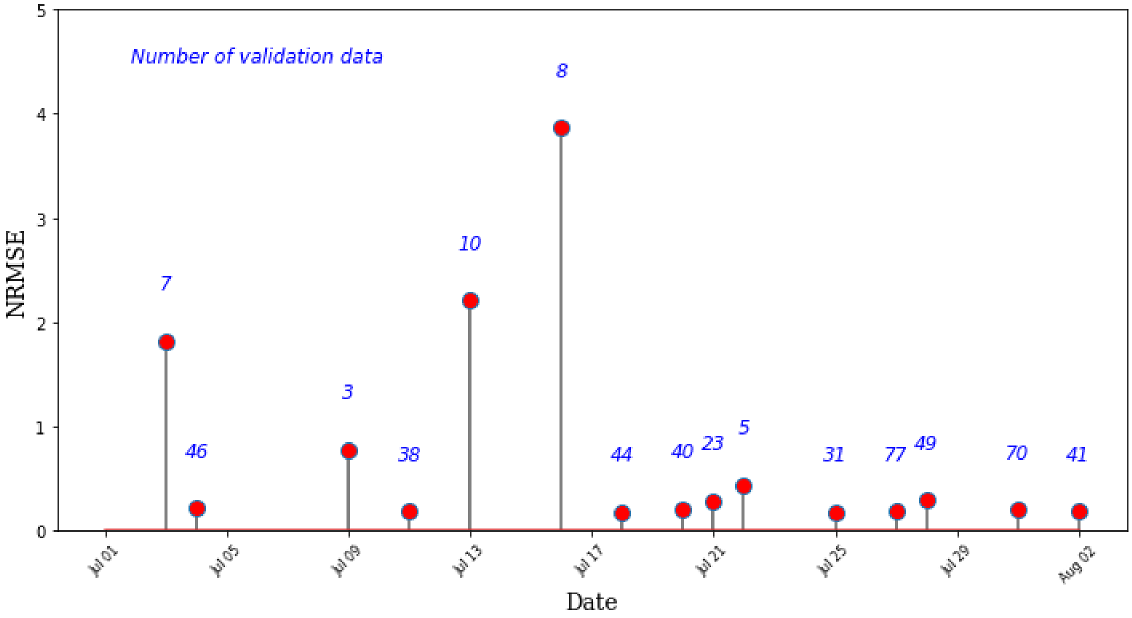


(b) Day-wise NRMSE between actual data and reduced estimation

Figure 5.8: Proposed reduced estimation cross-validation result



(a) Actual measurements vs full estimation for the entire season with overall NRMSE= 0.1244



(b) Day-wise NRMSE between actual data and full estimation

Figure 5.9: Full state and parameter estimation cross-validation result

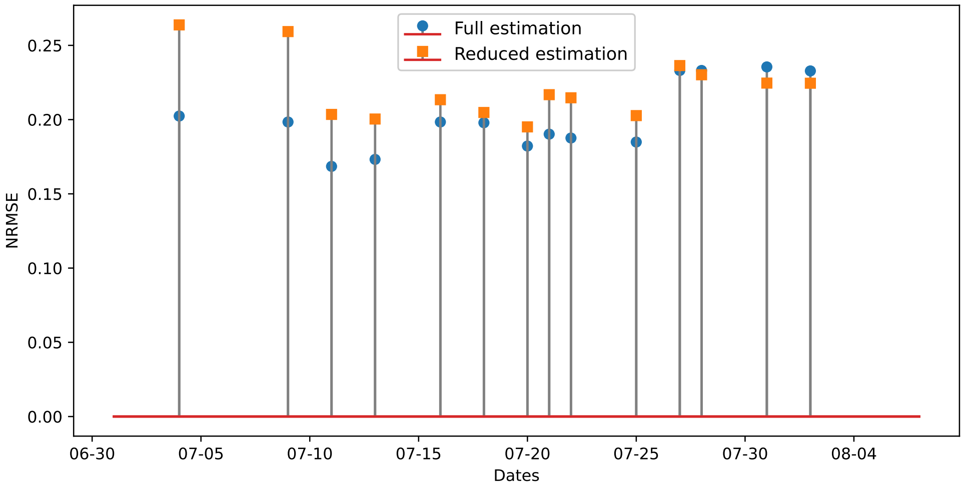
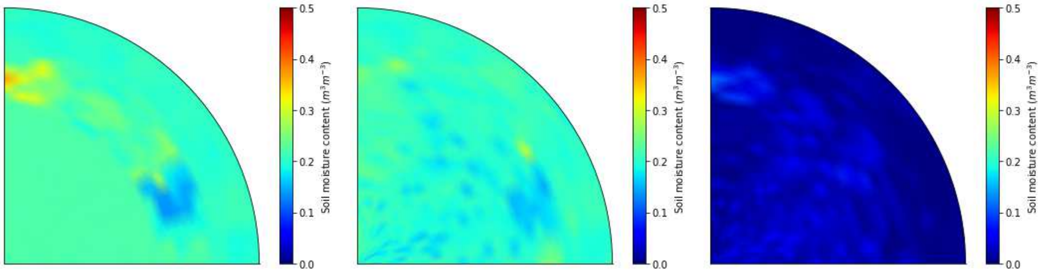
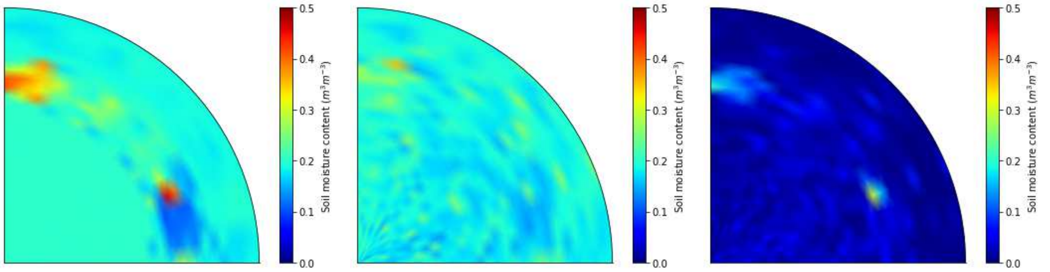


Figure 5.10: Measured soil moisture for cross-validation with day-wise NRMSE



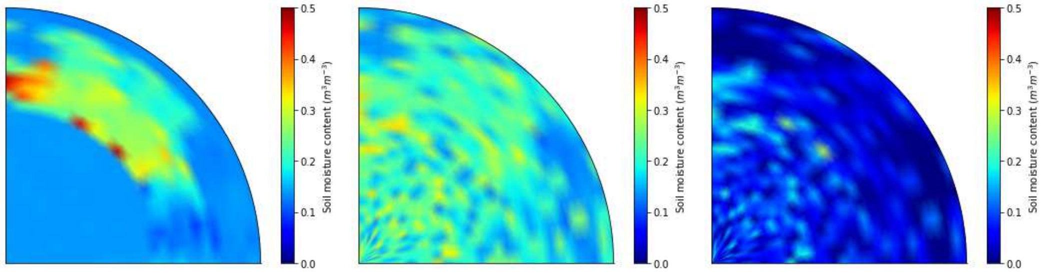
(a) Full estimation **(b)** Reduced estimation **(c)** AD between the two cases

Figure 5.11: Spatial soil moisture and AD on 11^{th} July



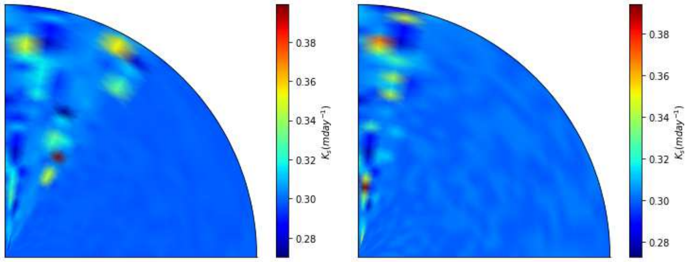
(a) Full estimation **(b)** Reduced estimation **(c)** AD between the two cases

Figure 5.12: Spatial soil moisture and AD on 20^{th} July



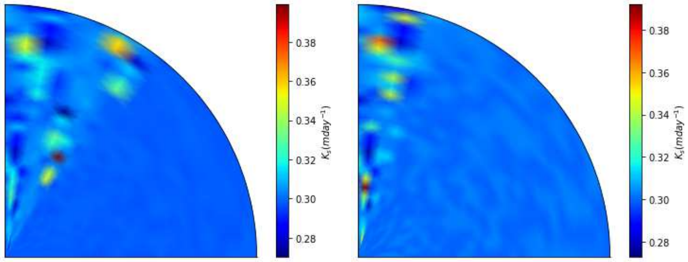
(a) Full estimation **(b)** Reduced estimation **(c)** AD between the two cases

Figure 5.13: Spatial soil moisture and AD on 2^{nd} August



(a) Full estimation **(b)** Reduced estimation

Figure 5.14: Estimated $K_s(mday^{-1})$ on 31^{st} July for both cases



(a) Full estimation **(b)** Reduced estimation

Figure 5.15: Estimated $K_s(mday^{-1})$ on 2^{th} August for both cases

the Richards equation is considered for soil moisture prediction in the absence of observations. The reduced state estimator demonstrates satisfactory performance, providing both high accuracy and improved computational speed. Thus, it concludes that the reduced approach can deliver sufficiently accurate soil moisture information which can be used to design of closed-loop irrigation systems. This proposed approach shows a greater potential for scalability in large-scale agricultural fields. Our future work will focus on implementing adaptive model reduction methods to design a controller for the field.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we have proposed systematic approaches to reduced-order models and state estimation applied to various applications.

In Chapter 2, an implicit two-time-scale nonlinear system is considered and a distributed state estimation method is proposed. Specifically, the nonlinear system is decomposed into a fast system and a slow subsystem. The solution of the system is approximated to a composite solution by combining the solution from fast and slow subsystems separately. A local MHE estimator is designed for the slow subsystem and and an EKF for the fast system. The estimator for the fast system is not required to send out any information. The effectiveness of the proposed approach was illustrated using a jacketed CSTR example.

In Chapter 3, a sensitivity-based method is proposed to determine how to select the appropriate inputs and outputs for data collection and data-driven model development to estimate the desired outputs accurately. Although a mechanistic model is available for the process but is too computationally complex for estimator design and only some target outputs are of interest and should be estimated. A systematic method to design a reduced state estimation framework is proposed for a target output. The effectiveness of the proposed approach is illustrated using a separator-reactor process example.

In Chapter 4, the problem of higher dimensionality of the agro-hydrological system defined by PDE is addressed. For a small field, the full estimator performance is compared with the reduced estimator method. The results show that the adaptive estimator has a better performance than the full estimator in terms of estimation error and computational cost [103]. The proposed approach has been applied to a large-scale agricultural field using an error-triggered and time-triggering model reduction approach. In both cases, the state estimation showed a satisfactory result and the effectiveness of the proposed adaptive model reduction methods.

In Chapter 5, soil moisture measurements of a field at the Alberta Irrigation Center, Lethbridge, obtained from microwave sensors mounted on a center pivot are used as the observations. An event-triggered followed by error-triggered model reduction is apt to reduce frequent model reduction. Frequently updated moisture content maps are constructed to demonstrate the ability of the information fusion system to eliminate the time delay associated with the current microwave remote sensing approach. A strong agreement between the estimated moisture content maps and the actual moisture content maps is also evident from the simulation results. The information fusion system was able to provide soil moisture estimates that are within reasonable error bounds from the actual soil moisture values.

6.2 Future research directions

- *Controller design :* As outlined in this thesis, there has been a lot of focus on reduced-order state estimation, which is seen as a practical and effective solution for tackling state estimation issues in large-scale processes. In the thesis, structure and topology-preserving adaptive model reduction techniques are explored using the triggering criterion. State estimation can be used both for process monitoring and control applications. The structure-preserving quality of the reduced model can be beneficial to constructing an observer-based controller for example a set point tracking MPC design integrated with a scheduler. The implementation of dynamic optimization will be highly useful due to the reduction of the states of the system.

- *Distributed state estimation and controller design :* When designing controllers for large-scale systems, employing a distributed estimation framework becomes beneficial for estimating the unmeasured states necessary for state feedback controls. The adaptive model reduction method will greatly help to construct the distributed framework. The distributed estimation and control will improve the computational cost.
- *Parameter estimation :* To expand the set of estimable hydraulic parameters, integrating measurements from fixed-point sensors into the proposed framework is crucial. This adjustment affects the spatial distribution of estimable hydraulic parameters, necessitating comprehensive simulations using sensitivity analysis and orthogonal projection methods. Future directions should prioritize the development of alternative methods for constructing the output sensitivity matrix in the context of incorporating fixed-point sensor measurements along with microwave sensors. Finally, considering the potential, a reduced-order model can be a valuable approximation in this context.

Bibliography

[1] A. C. Antoulas, *Approximation of large-scale dynamical systems.* SIAM, 2005.

[2] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, vol. 57, no. 4, pp. 483–531, 2015.

[3] X. Cheng and J. Scherpen, "Model reduction methods for complex network systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 425–453, 2021.

[4] J. A. Lee, M. Verleysen, *et al.*, *Nonlinear dimensionality reduction*, vol. 1. Springer, 2007.

[5] J. Liu, D. Muñoz de la Peña, and P. D. Christofides, "Distributed model predictive control of nonlinear process systems," *AIChE journal*, vol. 55, no. 5, pp. 1171–1184, 2009.

[6] S. A. Obiri, S. Bo, B. T. Agyeman, S. Debnath, B. Decardi-Nelson, and J. Liu, "Optimizing the switching operation in monoclonal antibody production: Economic mpc and reinforcement learning," *Chemical Engineering Research and Design*, vol. 199, pp. 61–73, 2023.

[7] R. Alexander, G. Campani, S. Dinh, and F. V. Lima, "Challenges and opportunities on nonlinear state estimation of chemical and biochemical processes," *Processes*, vol. 8, no. 11, p. 1462, 2020.

[8] X. Hu, F. Feng, K. Liu, L. Zhang, J. Xie, and B. Liu, "State estimation for advanced battery management: Key challenges and future trends," *Renewable and Sustainable Energy Reviews*, vol. 114, p. 109334, 2019.

[9] A. Kumar and P. Daoutidis, "Nonlinear dynamics and control of process systems with recycle," *Journal of Process Control*, p. 10, 2002.

[10] A. Kumar, P. D. Christofides, and P. Daoutidis, "Singular perturbation modeling of nonlinear processes with nonexplicit time-scale multiplicity," *Chemical Engineering Science*, vol. 53, no. 8, pp. 1491–1504, 1998.

[11] X. Yin and J. Liu, "Distributed moving horizon state estimation of two-time-scale nonlinear systems," *Automatica*, vol. 79, pp. 152–161, 2017.

[12] P. Kokotović, H. K. Khalil, and J. O'reilly, *Singular perturbation methods in control: analysis and design*. SIAM, 1999.

[13] X. Chen, M. Heidarinejad, J. Liu, and P. D. Christofides, "Composite fast-slow MPC design for nonlinear singularly perturbed systems," *AIChE Journal*, vol. 58, no. 6, pp. 1802–1811, 2012.

[14] X. Chen, M. Heidarinejad, J. Liu, D. M. de la Peña, and P. D. Christofides, "Model predictive control of nonlinear singularly perturbed systems: Application to a large-scale process network," *Journal of Process Control*, vol. 21, no. 9, pp. 1296–1305, 2011.

[15] A. Kumar and P. Daoutidis, "Nonlinear Model Reduction and Control for High-Purity Distillation Columns," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4495–4505, 2003.

[16] Z. Duan and C. Kravaris, "Nonlinear observer design for two-time-scale systems," *AIChE Journal*, vol. 66, no. 6, p. e16956, 2020.

[17] C. Georgakis, "A quasi-modal approach to model reduction," *Joint Automatic Control Conference*, no. 14, pp. 639–644, 1977.

[18] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations," *Journal of Differential Equations*, vol. 31, no. 1, pp. 53–98, 1979.

[19] R. Marino and P. V. Kokotovic, "A geometric approach to nonlinear singularly perturbed control systems," *Automatica*, vol. 24, no. 1, pp. 31–41, 1988.

[20] H. Krishnan and N. McClamroch, "On the connection between nonlinear differential-algebraic equations and singularly perturbed control systems in nonstandard form," *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 1079–1084, 1994.

[21] M. Baldea, P. Daoutidis, and A. Kumar, "Dynamics and control of integrated networks with purge streams," *AIChE Journal*, vol. 52, no. 4, pp. 1460–1472, 2006.

[22] D. S. Naidu and A. J. Calise, "Singular Perturbations and Time Scales in Guidance and Control of Aerospace Systems: A Survey," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 6, pp. 1057–1078, 2001.

[23] X. Yin and J. Liu, "State estimation of wastewater treatment plants based on model approximation," *Computers & Chemical Engineering*, vol. 111, pp. 79–91, 2018.

[24] B. F. Farrell and P. J. Ioannou, “State Estimation Using a Reduced-Order Kalman Filter,” *Journal of the Atmospheric Sciences*, vol. 58, no. 23, pp. 3666–3680, 2001.

[25] S. Debnath, S. R. Sahoo, B. Decardi-Nelson, and J. Liu, “Subsystem decomposition and distributed state estimation of nonlinear processes with implicit time-scale multiplicity,” *AIChE Journal*, vol. 68, no. 5, p. e17661, 2022.

[26] S. R. Sahoo and J. Liu, “Adaptive model reduction and state estimation of agro-hydrological systems,” *Computers and Electronics in Agriculture*, vol. 195, p. 106825, 2022.

[27] A. K. Singh and J. Hahn, “State estimation for high-dimensional chemical processes,” *Computers & Chemical Engineering*, vol. 29, no. 11, pp. 2326–2334, 2005.

[28] A. Narasingam and J. S.-I. Kwon, “Data-driven identification of interpretable reduced-order models using sparse regression,” *Computers & Chemical Engineering*, vol. 119, pp. 101–111, 2018.

[29] K. Raviprakash, B. Huang, and V. Prasad, “A hybrid modelling approach to model process dynamics by the discovery of a system of partial differential equations,” *Computers & Chemical Engineering*, vol. 164, p. 107862, 2022.

[30] K. Raviprakash, A. Ganesh, B. Huang, and V. Prasad, “Modelling temperature dynamics of the sagd process in an oil reservoir by the discovery of parametric partial differential equations,” *The Canadian Journal of Chemical Engineering*, vol. 101, no. 7, pp. 3948–3962, 2023.

[31] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[32] S. H. Son, H.-K. Choi, J. Moon, and J. S.-I. Kwon, “Hybrid koopman model predictive control of nonlinear systems using multiple edmd models: An application to a batch pulp digester with feed fluctuation,” *Control Engineering Practice*, vol. 118, p. 104956, 2022.

[33] P. Shah, M. Z. Sheriff, M. S. F. Bangi, C. Kravaris, J. S.-I. Kwon, C. Botre, and J. Hirota, “Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters,” *Chemical Engineering Journal*, vol. 441, p. 135643, 2022.

[34] D. Lee, A. Jayaraman, and J. S. Kwon, “Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling,” *PLoS Computational Biology*, vol. 16, no. 12, p. e1008472, 2020.

[35] M. S. F. Bangi and J. S.-I. Kwon, “Deep hybrid model-based predictive control with guarantees on domain of applicability,” *AIChE Journal*, p. e18012, 2022.

[36] M. S. Alhajeri, Z. Wu, D. Rincon, F. Albalawi, and P. D. Christofides, “Machine-learning-based state estimation and predictive control of nonlinear processes,” *Chemical Engineering Research and Design*, vol. 167, pp. 268–280, 2021.

[37] D. Li, J. Zhou, and Y. Liu, “Recurrent-neural-network-based unscented Kalman filter for estimating and compensating the random drift of MEMS

gyroscopes in real time," *Mechanical Systems and Signal Processing*, vol. 147, p. 107057, 2021.

[38] Y. Xia and J. Wang, "Low-dimensional recurrent neural network-based Kalman filter for speech enhancement," *Neural Networks*, vol. 67, pp. 131–139, 2015.

[39] T. Fernando, H. Maier, and G. Dandy, "Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach," *Journal of Hydrology*, vol. 367, no. 3-4, pp. 165–176, 2009.

[40] P. V. Balachandran, D. Xue, J. Theiler, J. Hogden, J. E. Gubernatis, and T. Lookman, "Importance of feature selection in machine learning and adaptive design for materials," in *Materials discovery and design*, pp. 59–79, Springer, 2018.

[41] S. Vijaya Raghavan, T. Radhakrishnan, and K. Srinivasan, "Soft sensor based composition estimation and controller design for an ideal reactive distillation column," *ISA Transactions*, vol. 50, no. 1, pp. 61–70, 2011.

[42] W. Ke, D. Huang, F. Yang, and Y. Jiang, "Soft sensor development and applications based on LSTM in deep neural networks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, (Honolulu, HI), pp. 1–6, IEEE, 2017.

[43] R. Sharmin, U. Sundararaj, S. Shah, L. Vande Griend, and Y.-J. Sun, "Inferential sensors for estimation of polymer quality parameters: Industrial application of a PLS-based soft sensor for a LDPE plant," *Chemical Engineering Science*, vol. 61, no. 19, pp. 6372–6384, 2006.

[44] E. Zamprogna, M. Barolo, and D. E. Seborg, "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis," *Journal of Process Control*, vol. 15, no. 1, pp. 39–52, 2005.

[45] V. K. Puli, R. Raveendran, and B. Huang, "Complex probabilistic slow feature extraction with applications in process data analytics," *Computers & Chemical Engineering*, vol. 154, p. 107456, 2021.

[46] L. Xie, Y. Zhu, B. Huang, and Y. Zheng, "Kalman filtering approach to multirate information fusion for soft sensor development," in *2012 15th International Conference on Information Fusion, Singapore*, pp. 641–648, IEEE, 2012.

[47] "Global risks 2015," *World Economic Forum Report*, 2015.

[48] "Waste water the untapped resource," *The United Nations World Water Development Report*, 2017.

[49] C. Lozoya, C. Mendoza, L. Mejía, J. Quintana, G. Mendoza, M. Bustillos, O. Arras, and L. Solís, "Model predictive control for closed-loop irrigation," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4429–4434, 2014.

[50] S. R. Sahoo, X. Yin, and J. Liu, "Optimal sensor placement for agro-hydrological systems," *AIChE Journal*, vol. 65, no. 12, p. e16795, 2019.

[51] S. R. Sahoo and J. Liu, "Adaptive model reduction and state estimation of agro-hydrological systems," *Computers and Electronics in Agriculture*, vol. 195, p. 106825, 2022.

[52] E. Orouskhani, B. T. Agyeman, and J. Liu, "Simultaneous estimation of soil moisture and hydraulic parameters for precision agriculture. part a: Methodology," in *2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pp. 12–17, IEEE, 2022.

[53] X. Yin, S. Bo, J. Liu, and B. Huang, “Consensus-based approach for parameter and state estimation of agro-hydrological systems,” *AIChE Journal*, vol. 67, no. 2, p. e17096, 2021.

[54] Y. Mao, S. Liu, J. Nahar, J. Liu, and F. Ding, “Soil moisture regulation of agro-hydrological systems using zone model predictive control,” *Computers and Electronics in Agriculture*, vol. 154, pp. 239–247, 2018.

[55] S. R. Sahoo, B. T. Agyeman, S. Debnath, and J. Liu, “Knowledge-based optimal irrigation scheduling of agro-hydrological systems,” *Sustainability*, vol. 14, no. 3, p. 1304, 2022.

[56] B. T. Agyeman, S. R. Sahoo, J. Liu, and S. L. Shah, “An lstm-based mixed-integer model predictive control for irrigation scheduling,” *The Canadian Journal of Chemical Engineering*, vol. 101, no. 6, pp. 3362–3381, 2023.

[57] B. T. Agyeman, S. Bo, S. R. Sahoo, X. Yin, J. Liu, and S. L. Shah, “Soil moisture map construction by sequential data assimilation using an extended kalman filter,” *Journal of Hydrology*, vol. 598, p. 126425, 2021.

[58] S. Bo, S. R. Sahoo, X. Yin, J. Liu, and S. L. Shah, “Parameter and state estimation of one-dimensional infiltration processes: A simultaneous approach,” *Mathematics*, vol. 8, no. 1, p. 134, 2020.

[59] E. Babaeian, M. Sadeghi, S. B. Jones, C. Montzka, H. Vereecken, and M. Tuller, “Ground, proximal, and satellite remote sensing of soil moisture,” *Reviews of Geophysics*, vol. 57, no. 2, pp. 530–616, 2019.

[60] X. Yin and J. Liu, “State estimation of wastewater treatment plants based on model approximation,” *Computers & Chemical Engineering*, vol. 111, pp. 79–91, 2018.

[61] S. Kung and D. Lin, "Optimal hankel-norm model reductions: Multivariable systems," *IEEE Transactions on Automatic Control*, vol. 26, no. 4, pp. 832–852, 1981.

[62] V. K. Puli and B. Huang, "Variational bayesian approach to nonstationary and oscillatory slow feature analysis with applications in soft sensing and process monitoring," *IEEE Transactions on Control Systems Technology*, 2023.

[63] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, no. 8, pp. 748–766, 2004.

[64] X. Cheng and J. Scherpen, "Gramian-based model reduction of directed networks," *arXiv preprint arXiv:1901.01285*, 2019.

[65] S. R. Sahoo, B. T. Agyeman, S. Debnath, and J. Liu, "Knowledge-based optimal irrigation scheduling of three-dimensional agro-hydrological systems," *IFAC-PapersOnLine*, vol. 55, no. 7, pp. 441–446, 2022.

[66] M. W. Rasheed, J. Tang, A. Sarwar, S. Shah, N. Saddique, M. U. Khan, M. Imran Khan, S. Nawaz, R. R. Shamshiri, M. Aziz, *et al.*, "Soil moisture measuring techniques and factors affecting the moisture dynamics: A comprehensive review," *Sustainability*, vol. 14, no. 18, p. 11538, 2022.

[67] G. Petropoulos, *Remote sensing of energy fluxes and soil moisture content*. CRC press, 2013.

[68] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi, "Hydrologic data assimilation with the ensemble Kalman filter," *Monthly Weather Review*, vol. 130, no. 1, pp. 103–114, 2002.

[69] M. Pan, E. F. Wood, R. Wójcik, and M. F. McCabe, "Estimation of regional terrestrial water cycle using multi-sensor remote sensing observations and data assimilation," *Remote Sensing of Environment*, vol. 112, no. 4, pp. 1282–1294, 2008.

[70] S. Bo, S. R. Sahoo, X. Yin, J. Liu, and S. L. Shah, "Parameter and State Estimation of One-Dimensional Infiltration Processes: A Simultaneous Approach," *Mathematics*, vol. 8, no. 1, p. 134, 2020.

[71] J. Liu, A. Gnanasekar, Y. Zhang, S. Bo, J. Liu, J. Hu, and T. Zou, "Simultaneous State and Parameter Estimation: The Role of Sensitivity Analysis," *Industrial & Engineering Chemistry Research*, vol. 60, no. 7, pp. 2971–2982, 2021.

[72] B. T. Agyeman, E. Orouskhani, M. Naouri, W. Appels, M. Wolleben, J. Liu, and S. L. Shah, "Maximizing soil moisture estimation accuracy through simultaneous hydraulic parameter estimation using microwave remote sensing: Methodology and application," *arXiv preprint arXiv:2305.15549*, 2023.

[73] S. Bo and J. Liu, "A decentralized framework for parameter and state estimation of infiltration processes," *Mathematics*, vol. 8, no. 5, p. 681, 2020.

[74] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[75] J. Zhang and J. Liu, "Distributed moving horizon state estimation for nonlinear systems with bounded uncertainties," *Journal of Process Control*, vol. 23, no. 9, pp. 1281–1295, 2013.

[76] X. Yin, B. Decardi-Nelson, and J. Liu, "Subsystem decomposition and distributed moving horizon estimation of wastewater treatment plants," *Chemical Engineering Research and Design*, vol. 134, pp. 405–419, 2018.

[77] M. Baldea and P. Daoutidis, *Dynamics and Nonlinear Control of Integrated Process Systems.* Cambridge Series in Chemical Engineering, Cambridge: Cambridge University Press, 2012.

[78] F. Verhulst, *Methods and applications of singular perturbations: boundary layers and multiple timescale dynamics*, vol. 50. Springer Science & Business Media, 2005.

[79] J. K. Kevorkian and J. D. Cole, *Multiple scale and singular perturbation methods*, vol. 114. Springer Science & Business Media, 2012.

[80] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, pp. 246–258, Feb. 2003.

[81] M. Valipour and L. A. Ricardez-Sandoval, "Constrained abridged gaussian sum extended kalman filter: constrained nonlinear systems with non-gaussian noises and uncertainties," *Industrial & Engineering Chemistry Research*, vol. 60, no. 47, pp. 17110–17127, 2021.

[82] M. Valipour and L. A. Ricardez-Sandoval, "Assessing the impact of ekf as the arrival cost in the moving horizon estimation under nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 60, no. 7, pp. 2994–3012, 2021.

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

[84] S. Debnath, S. R. Sahoo, B. T. Agyeman, and J. Liu, "Input-output selection for lstm-based reduced-order state estimator design," in *22nd IFAC World Congress, Japan*, pp. 6940–6945, IFAC, 2023.

[85] S. Debnath, S. R. Sahoo, B. T. Agyeman, and J. Liu, "Input-output selection for lstm-based reduced-order state estimator design," *Mathematics*, vol. 11, no. 2, p. 400, 2023.

[86] T. Zhao, Y. Zheng, and Z. Wu, "Improving Computational Efficiency of Machine Learning Modeling of Nonlinear Processes Using Sensitivity Analysis and Active Learning," *Digital Chemical Engineering*, p. 100027, 2022.

[87] T. Zhao, Y. Zheng, J. Gong, and Z. Wu, "Machine learning-based reduced-order modeling and predictive control of nonlinear processes," *Chemical Engineering Research and Design*, vol. 179, pp. 435–451, 2022.

[88] T. Zhao, Y. Zheng, and Z. Wu, "Feature selection-based machine learning modeling for distributed model predictive control of nonlinear processes," *Computers & Chemical Engineering*, vol. 169, p. 108074, 2023.

[89] N. L. Grubben and K. J. Keesman, "Controllability and observability of 2D thermal flow in bulk storage facilities using sensitivity fields," *International Journal of Control*, vol. 91, no. 7, pp. 1554–1566, 2018.

[90] J. D. Stigter, D. Joubert, and J. Molenaar, "Observability of Complex Systems: Finding the Gap," *Scientific Reports*, vol. 7, no. 1, p. 16566, 2017.

[91] J. Stigter, L. van Willigenburg, and J. Molenaar, "An Efficient Method to Assess Local Controllability and Observability for Non-Linear Systems," *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 535–540, 2018.

[92] J. D. Stigter and J. Molenaar, “A fast algorithm to assess local structural identifiability,” *Automatica*, vol. 58, pp. 118–124, 2015.

[93] R. Li, M. Henson, and M. Kurtz, “Selection of model parameters for off-line parameter estimation,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 402–412, 2004.

[94] H. Wang and H. Ma, “Optimal investment portfolios for internet money funds based on LSTM and La-VaR: evidence from China,” *Mathematics*, vol. 10, p. 2864, 2022.

[95] I. Abuqaddom, B. A. Mahafzah, and H. Faris, “Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients,” *Knowledge-Based Systems*, vol. 230, p. 107391, 2021.

[96] A. Rauh, S. Wirtensohn, P. Hoher, J. Reuter, and L. Jaulin, “Reliability assessment of an unscented kalman filter by using ellipsoidal enclosure techniques,” *Mathematics*, vol. 10, p. 3011, 2022.

[97] L. A. Richards, “Capillary conduction of liquids through porous mediums,” *Physics*, vol. 1, no. 5, pp. 318–333, 1931.

[98] Y. Mualem, “A new model for predicting the hydraulic conductivity of unsaturated porous media,” *Water Resources Research*, vol. 12, no. 3, pp. 513–522, 1976.

[99] M. T. Van Genuchten, “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils,” *Soil science society of America journal*, vol. 44, no. 5, pp. 892–898, 1980.

[100] R. G. Allen, L. S. Pereira, D. Raes, M. Smith, *et al.*, "Crop evapotranspiration-guidelines for computing crop water requirements-fao irrigation and drainage paper 56," *Fao, Rome*, vol. 300, no. 9, p. D05109, 1998.

[101] A. Alanqar, H. Durand, and P. D. Christofides, "Error-triggered online model identification for model-based feedback control," *AIChE Journal*, vol. 63, no. 3, pp. 949–966, 2017.

[102] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," *CS&E Technical Reports*, 2000.

[103] S. Debnath, S. R. Sahoo, B. T. Agyeman, X. Yin, and J. Liu, "An error-triggered adaptive model reduction and soil moisture estimation for agro-hydrological system," in *62nd IEEE Conference on Decision and Control (CDC), Singapore*, pp. 7586–7591, IEEE, 2023.

[104] S. A. Obiri, B. T. Agyeman, S. Debnath, S. Liu, and J. Liu, "Sensor selection and state estimation of continuous mab production processes," *Mathematics*, vol. 11, no. 18, p. 3860, 2023.

[105] S. Debnath, B. T. Agyeman, S. R. Sahoo, X. Yin, and J. Liu, "Performance triggered adaptive model reduction for soil moisture estimation in precision irrigation *(Submitted)*," *International Journal of Adaptive Control and Signal Processing*, 2024.

[106] B. T. Agyeman, E. Orouskhani, and J. Liu, "Simultaneous estimation of soil moisture and hydraulic parameters for precision agriculture. part b: Application to a real field," in *2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pp. 18–23, IEEE, 2022.

[107] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.