Model Reduction and Remote Sensing for Precision Irrigation Applications

by

Soumya Ranjan Sahoo

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

Abstract

Two-thirds of the world population faces severe water stress at least once per month in a year. The rapidly growing population exacerbates freshwater scarcity. Additionally, climate change, pollution, and bio-energy demands amplify the water demand problem. Since a large percentage of freshwater withdrawals are for agricultural irrigation, increasing water-use efficiency in irrigation becomes extremely important. Closed-loop irrigation is one of the precision irrigation techniques which has the potential to improve water-use efficiency than the traditional approaches. Due to the large-scale features of agricultural fields and significant uncertainty, there are significant challenges associated with closed-loop irrigation including sensor placement, data assimilation, and controller design. This thesis addresses some of the major challenges of applying closed-loop irrigation in the actual fields by proposing novel methods.

Due to the availability of a limited number of sensors, in a typical agricultural field, it is essential to know the minimum number of sensors and optimal location of the sensors to estimate the field's soil moisture. The structure-preserving graph-based approach is used to reduce the order of a large-scale system model. A systematic approach has been developed to find the minimum number and best location of the sensors using observability and degree of observability analysis. In some irrigation implementing systems, the irrigation amount is non-uniform in spatial directions. In these scenarios, one reduced model for the whole time may not capture all the dynamics of a large-scale field or may increment the order of the resulting reduced model. Dynamic model reduction is proposed to handle these scenarios, where different reduced models are computed at different periods of time. Further, the sensor placement using dynamic reduced order is developed.

Next, the framework for the state estimation of the large-scale field using the

advanced optimization-based moving horizon estimation (MHE) is developed. The trajectory-based unsupervised machine learning method is proposed for adaptive model reduction of very large agricultural fields. Further, the algorithm of the existing MHE is modified to handle the adaptive reduced model. The proposed approach is applied to estimate the states of a real-agricultural field scenario located in Lethbridge, Canada.

Afterward, an optimization-based closed-loop scheduler for large agricultural fields to provide optimal irrigation time and amount is developed. The structure-preserving model reduction is used to decrease the dimension of the three-dimensional model. The scheduler has an objective similar to the economic zone MPC. In addition to that, time is also a decision variable to the optimization problem. The final objective is to maximize the yield while minimizing the water consumption and maximizing the time between the irrigation events.

Further, the algorithm for surface soil moisture estimation using the thermal and optical remote sensing method is developed. The machine learning based Long Short-Term Memory (LSTM) model is used to estimate the surface soil moisture. Due to the time-varying nature of the agro-hydrological model, the LSTM model is more preferred than the static neural network model. The LSTM based model is trained to obtain the surface soil moisture from the remote sensing images and the weather conditions. The developed method is applied to a real-agricultural farm located in Lethbridge, Canada using the experimental data collected in summer 2019.

In this thesis, the details of the study area and experimental data collection procedure for experiments conducted in summer 2019 and 2020 at an agricultural field and a golf club are provided.

Preface

The materials presented in this thesis are part of the research project under the supervision of Dr. Jinfeng Liu, and is funded by Alberta Innovates Technology Futures (AITF) and Natural Sciences and Engineering Research Council (NSERC) of Canada.

Chapter 3 of this thesis is a revised version of Soumya R. Sahoo, Xunyuan Yi, Jinfeng Liu, Optimal Sensor Placement for Agro-hydrological Systems, *AIChE*, 73:180-192, 2019.

Chapter 4 of this thesis is a revised version of Soumya R. Sahoo, Xunyuan Yin, Jinfeng Liu, Sirish L Shah, Dynamic Model Reduction and Optimal Sensor Placement for Agro-hydrological Systems, *21st IFAC World Congress*, 53(2), 11669-11674, Berlin, Germany, 2020

Chapter 5 has been submitted as: Soumya R. Sahoo and Jinfeng Liu, Adaptive Model Reduction and State Estimation of Agro-hydrological Systems, *Computers and Electronics in Agriculture* (minor revision). A short version has been accepted as Soumya R. Sahoo and Jinfeng Liu, Adaptive Model Reduction and State Estimation of Agro-hydrological Systems, *American Control Conference, Atlanta, GA, 2022.*

Chapter 6 has been submitted as: Soumya R. Sahoo, Bernard Ageyaman, Sarupa Debnath, Jinfeng Liu, Knowledge-based Optimal Irrigation Scheduling of Agro hydrological Systems, *Sustainability*, 14(3), 1304, 2022. A short version has been submitted as Soumya R. Sahoo, Bernard Ageyaman, Sarupa Debnath, Jinfeng Liu, Knowledge-based Optimal Irrigation Scheduling of Three-dimensional Agro-hydrological Systems, *IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems*, Busan, Republic of Korea, 2022.

Acknowledgements

First, I would like to express my profound gratitude to my supervisor Professor Jinfeng Liu for providing me with excellent guidance throughout my Ph.D. journey. His devotion towards research, patience, and enthusiasm made him my role model. His constant wholehearted support and encouragement significantly boosted my self-confidence and helped me build a path towards the achievement of my career goals. This thesis would not have been possible without his support and expert guidance.

I gratefully acknowledge Professor Stevan Dubljevic, Professor Zukui Li, Professor Qing Zhao, Professor Xiang Li for serving on my doctoral examination, and Professor Sirish L Shah for helping me in my thesis work.

I would also like to thank my PSACE group members who have helped and supported me during my Ph.D. journey, including Sarupa Debnath, Song Bo, Bernard Agyeman, Xunyuan Yin, Benjamin Decardi-Nelson, Jannatun Nahar, Su Liu, Yi Zhang, Jianbang Liu, Erfan Orouskhani, Aristarchus Gnanasekar, Rui Nian, An Zhang, Nirwair Bajwa, Nirav Raiyani, Zhiyinan Huang and Guoyang Yan. I am grateful to my friends and family in Edmonton who supported me in my personal life, including Dr. Jagannath Prasad Das, Gita Das, Shammy Raj, Shilpi Rathi, Vineet Rathi, Anuja Tripathi, and Aishwarya Rath.

I gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Alberta Innovative Technology Futures (AITF).

I am incredibly thankful to my father, Mayadhar Sahoo, my brother Manas Ranjan Sahoo and my sister-in-law Annapurna Sahoo for their constant support in my life. Last but not least, this thesis is dedicated to my mother, Late. Sanjukta Sahoo, to whom I owe everything.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Freshwater scarcity is one of the top global risks due to significant growth in population, change in climate, and the increase in pollution. [3, 4]. From the total amount of freshwater, around 70% is used in agriculture, 20% is used in industry, and 10% is used for domestic purposes [1] (Figure 1.1).

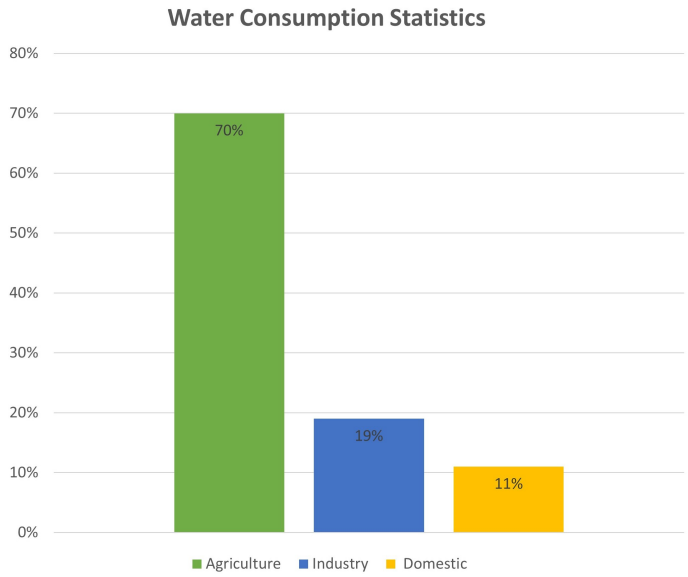


Figure 1.1: World wide water consumption statistics [1]

As can be seen, the agriculture sector requires a large portion of freshwater, and the demand is indeed increasing. One of the biggest reasons for the increase in demand is population growth. The global population is expected to be nine billion by the year 2050, and the food demand is also expected to be increased by 70% [5].

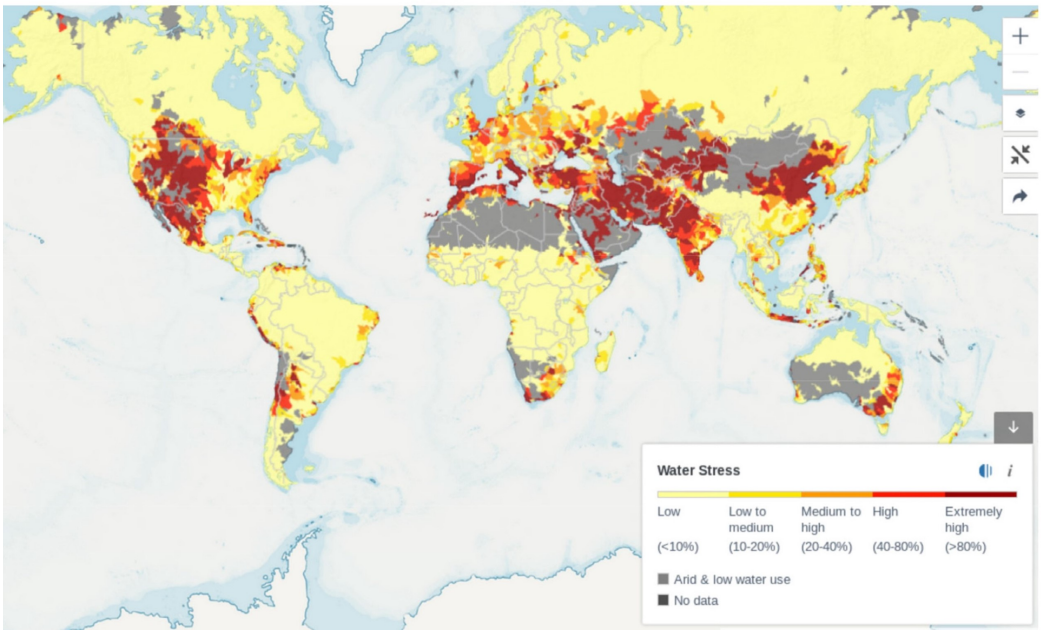


Figure 1.2: Projected water scarcity in 2030 [2]

About 40% of the world's food comes from irrigated agriculture. To match the food demand, the water required for irrigation has to increase by 11% [5]. In the present situation, three billion people live in a chronic water shortage area [6]. Figure 1.2 shows the water stress area projected in 2030, indicating most of the world population is going to face the water scarcity. Climate change is also one of the biggest factors in water management in agriculture. Due to global warming, the temperature starts to increase, which results in more evaporation from the land and sea, which makes less rainfall in the semi-arid and mid-latitude area [5]. In the irrigation practice, the water-use efficiency is around 50% to 60% due to poor irrigation strategy [7]. Furthermore, irrigation has a direct impact on the greenhouse gas emissions (GHG) [8] due to electricity usage. Previous studies have shown a huge amount of GHG release due to irrigation. In the US, the pumping requires around 23% of total farm energy [8]. Thus an important step towards managing the water crisis and to reduce GHG is to increase the water-use efficiency.

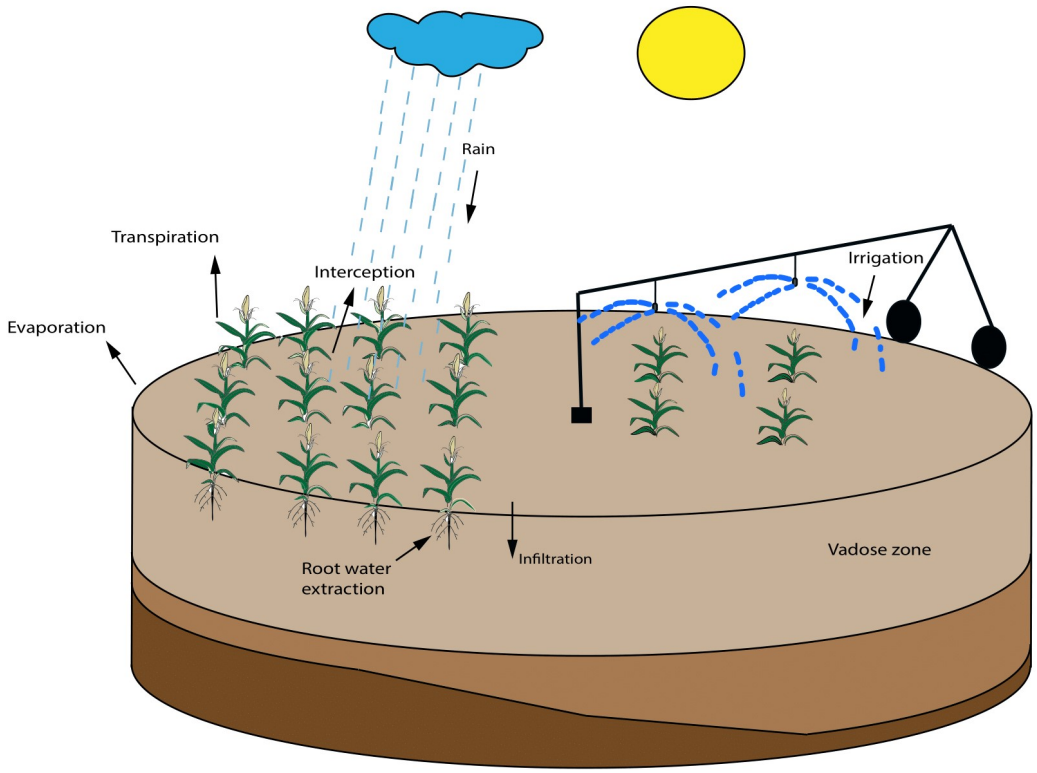


Figure 1.3: Schematic of the agro-hydrological system

1.2 Background

The hydrological cycle among soil, water, crop, and atmosphere is characterized by agro-hydrological systems. The schematic of the agro-hydrological system representing a simple hydrological cycle is presented in Figure 1.3. The inflow of the agro-hydrological system is precipitation, irrigation while the outflows consist of transpiration, root water extraction, evaporation, drainage, and runoff. The soil saturation condition plays a vital role in the precipitation and irrigation water entering the soil. In the case of unsaturated soil, water infiltration occurs, but for saturated soil, the water infiltration rate starts to decrease, and ponding occurs. The runoff happens after a certain threshold of ponding. Root water extraction happens from the root of the crop, which acts as a water sink. The evaporation and transpiration occur from the soil and the crop canopy.

The traditional irrigation is based on the uniform supply of water which ignores the spatial and temporal variability which often causes the over or under irrigation [9]. Different types of control techniques have been implemented for irrigation manage-

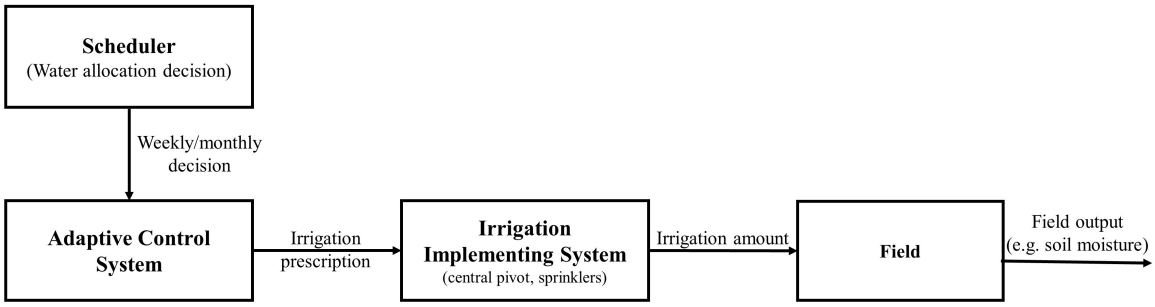


Figure 1.4: Current open-loop irrigation practice

ment and scheduling. The most popular current irrigation techniques are essentially in open-loop, in which any form of system measurements are not used to modify the inputs. The decisions are based on the historical data, farmers knowledge, heuristics or a steady-state model of the system. The traditional open-loop irrigation system is presented in Figure 1.4.

The main drawbacks of the open-loop irrigation system are that there is no direct connection between system outputs and inputs, the system is not able to compensate for disturbances and the irrigation is not precise which often leads to over irrigation [10]. Precision irrigation is required to increase the water-use efficiency of irrigation.

One promising precision irrigation solution is to take real-time feedback information into account in the irrigation system design and form a closed-loop irrigation system. The efficiency of water use is expected to be increased significantly by implementing a closed-loop system where the controller makes irrigation decisions based on real-time field conditions [11]. The real-time field information can be collected using different sensing instruments like soil moisture sensors, evapotranspiration (ET) sensors, remote sensing cameras, or satellite images. The entire condition of the field can be obtained by the information fusion from the sensors using the data assimilation methods. Based on the real-time field conditions, a feedback controller may be used to make optimal irrigation decisions for the next few hours and days. The control system may also take into account the local weather forecast, energy and water minimization, and other pre-defined crop and irrigation requirements. The schematic of a basic closed-loop irrigation system is shown in Figure 1.5. To implement such a control system, significant challenges are present in modeling, data assimilation, controller design due to the large scale of agricultural fields, complex water dynamics,

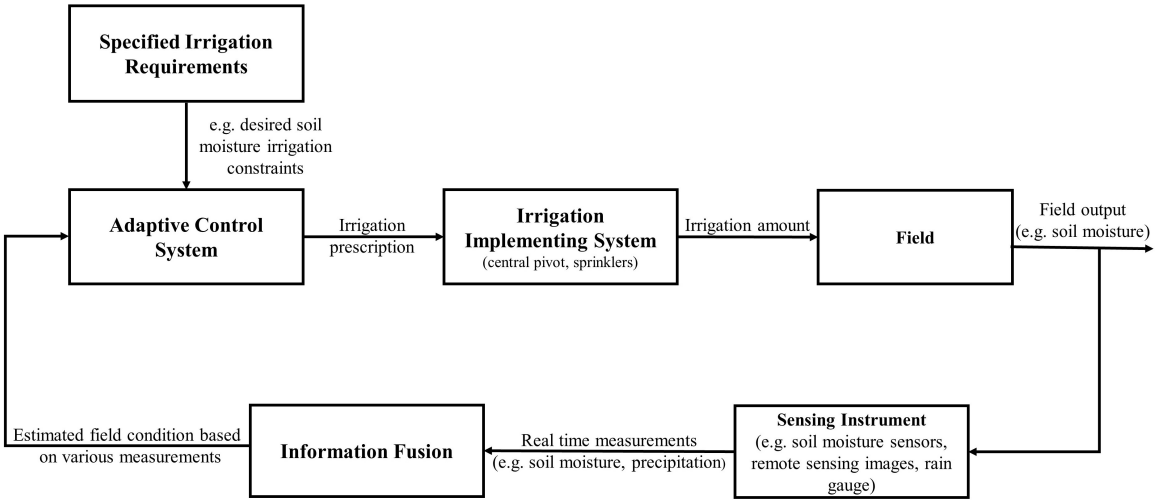


Figure 1.5: A schematic of the closed-loop irrigation system

and significant uncertainty.

1.3 Literature review

1.3.1 Model development of agro-hydrological systems

One of the most important aspects of precision irrigation is the modeling of the agro-hydrological system. In the last decade, much progress has been made in developing soil-crop-atmosphere models. The models evaluate the water movement through the soils and in the root zone considering the soil, crop, and atmospheric conditions. In recent years, commercially available one-dimensional models include HYDRUS, SWAP, SPOTPY, CRITERIA, WAVE, and STEMMUS and three-dimensional models include HYDRUS 3D, Saltirsoil-M, Coup Model, and VSoil Platform [12]. Advanced soil models like HYDRUS, SWAP, VSoil Platform, VS2D, WAVE, RichardsFOAM, FEFLOW use the Richards equation [13], which combines the Darcy equation with the sink term which models the evapotranspiration and root water extraction. The Richards equation is challenging to solve in hydrology because of the presence of dramatic nonlinearity [14]. The solution depends upon two highly nonlinear coefficients: soil water capacity (C) and hydraulic conductivity (K). The hydraulic conductivity (K) is very small in the dry condition and rapidly changes at near saturation. The pressure head and water capacity approach zero near saturation, but the diffusivity ($K\partial h/\partial\theta$) becomes extremely large. So at near full saturation, the coefficients

change rapidly, making the degeneracy of the system solution. Some models (e.g. AQUACROP) use simple mass, energy balance equation to calculate the soil moisture content instead of the Richards equation because of the above-discussed challenges.

Many researchers attempted to solve Richards equation by using different discretization schemes such as finite difference, finite volume, and finite element methods. Camporse and co-authors developed the CATHY model for surface and subsurface water flow using the continuous Galerkin method [15]. In [16, 17], the variably saturated flow models were developed using finite element method. Finsterle and co-authors developed the TOUGH model for the solution of vadose zone transport problems using the integrated finite difference method [18]. The VS2DI package was developed for simulation of water solute transportation in variably saturated soil using cell-centered differences [19]. Simunek and co-authors developed different numerical models for variably saturated transport (HYDRUS-1D and HYDRUS-2D), solute transport using analytical methods in soil (CXTFIT and STANMOD), and databases for the prediction of soil properties (RETC and UNSODA) [20]. The finite element method was used for the spatial discretization in the HYDRUS model. Most of the above-discussed models are basically black-box model, so using the black-box models to optimize automatic control, parameter, and state estimation is difficult. Orgogozo and co-authors proposed the RichardsFOAM solver for the solution of Richards equation [21]. The solver was developed in the open-source generalist computational fluid dynamics toolbox OpenFOAM framework, which uses the finite volume method. The authors discussed that the RichardsFOAM could handle large-scale problems in space and time with high ($\sim$90%) parallel efficiency using parallel computation. The main drawbacks of this model are the absence of the crop model, and the meshes can only handle constant cell volume. The most popular central pivot irrigation system works in a circular manner. In order to account for the circular movement, the polar coordinates are more preferred than the cartesian coordinates. In [22], a finite-difference model based on cylindrical coordinates is developed. More complete reviews on modeling and challenges in Richards equation can be found in [14, 23].

In Chapter 2, the numerical solution of cylindrical coordinates based Richards equation and how to improve the computational speed of the developed model is discussed.

1.3.2 State estimation and sensor placement

One of the major challenges of closed-loop irrigation is to obtain the field condition for decision making. The field condition can be measured based on soil moisture sensors, but it is unaffordable and unrealistic to deploy sensors everywhere on a farm. For a large agricultural field, one of the feasible solutions is to estimate the field condition or system states using a limited number of sensors. Two major questions arise: what the minimum number of sensors is required for estimating the soil moisture of the entire field, and where the best locations are to put the sensors.

The data assimilation or estimation of the soil moisture has been extensively studied in the past using popular methods like extended Kalman filter (EKF) [24, 25], ensemble Kalman filter (EnKF) [26, 27, 28, 29, 30, 31], particle filter [32], and moving horizon estimation (MHE) [33, 34, 35]. The major challenge in estimation using PF, EKF, and EnKF methods is that they cannot explicitly handle the constraints. In general, we do not have enough information about actual initial system states, so while designing the EKF and EnKF, we may need to provide a high initial covariance matrix. The high covariance may take some states to the extreme pressure head condition (dry or saturated region) and solve the agro-hydrological system using Richards equation at extreme pressure head conditions faces numerical challenges [14]. In Erdal and co-authors [29], it has also been shown that under extreme pressure head conditions, the EnKF becomes unstable. Another limitation of EKF and EnKF methods is the assumption of Gaussian noise. In the presence of non-Gaussian noise, the estimation performance may deteriorate. MHE can handle the above-discussed problems. In [33, 34, 35], MHE has been used for state and parameter estimation for both 1D and 3D system . But there are still some limitations, such as it applies to a 1D system or a small 3D system, which can be approximated to a 1D system. The challenges arise when solving an optimization problem for a large agricultural field in presence of heterogeneous soil types with non-uniform irrigation amounts.

In [36], the minimum required sensors were calculated using sensor failure approach by gradually removing the sensors. In this approach, brute force was used to determine the minimum sensor set, but the optimal placement of sensor was not discussed. Jannatun and co-authors [37] proposed a method to find the optimal po-

sitions of the sensor nodes using observability analysis but the applicability of this method is restricted to small one-dimensional systems. The classical approaches discussed in this paper for the observability analysis method may suffer from numerical issues.

Chapters 3 and 4 address the above problems and propose a systematic sensor placement method for large-scale agricultural fields. Chapter 5 addresses the above-discussed issue of using MHE for large agricultural fields.

1.3.3 Recent developments of closed-loop irrigation system

The variables in the feedback of the closed-loop irrigation are soil, plant, or atmosphere measurements [10]. The soil measurements include soil moisture content and soil water potential, whereas plant-based measurements include the tissue water content, canopy temperature, stomatal conductance, and growth rate [38]. The control objective for the soil moisture feedback is to keep soil moisture within certain zones. The controller starts irrigation if the level of soil moisture drops below lower threshold and stops when the high threshold is reached [39, 40, 41, 42]. The plant measurement-based feedback controller tries to maintain a certain stress threshold. In [43] and [44], the crop water stress index (CWSI) was used for plant feedback irrigation scheduling. In [45], the irrigation schedule was determined by the temperature threshold method, which used the infrared thermometer mounted on a central pivot. The crop water stress is determined by the leaf temperature. Plant feedback-based irrigation only provides information when the system requires the irrigation but does not provide the exact amount of water to be supplied, and it also faces the difficulty of implementation in real fields [38].

In [46], a setpoint-based wireless sensor network was implemented for the commercial pot-in-pot nursery. The authors reported 63% water saving compared to the water irrigated by an experienced manager. The annual saving was $5263, which considered the pumping, irrigation, and management cost-saving and the sensor cost. In [47], a wireless distributed network was developed, which consists of soil temperature and soil moisture sensors. The irrigation system was activated as an on-off controller when the soil moisture fell below a certain range, or the temperature was above a certain range. The authors claimed 90% of water-saving compared with the

traditional practices. In [48], a hybrid wireless and actuator network was deployed for closed-loop drip irrigation, and the system was activated when the soil moisture was out of the predefined threshold. The hybrid wireless system worked with a greater than 97% data accuracy rate. In [49], the automated closed-loop variable rate sprinkler system was developed, and the control strategy was based on the feedback from the distributed in-field soil moisture sensors. In [50], the sensor-based irrigation increased the revenue by 62% and profit by 65% per year. The authors also reported several advantages of precise irrigation such as the increase of quality and quantity of the crops, less production time period, decrease in electricity usage, and reduction of greenhouse gas emission because of the reduction of energy and fertilizer use. In [51], the sensor-based techniques (Iterative Hill Climbing Control and Iterative Learning Control) were used in the software VARIwise [52] to improve the yield and the irrigation efficiency. The field measurements were compared with the desired system output, and irrigation volume and irrigation were applied according to the difference. The VARIwise is an adaptive decision simulation framework that considers the spatial and temporally varied simulation [52]. These sensor-based approaches have reported several advantages, but they only considered the point-based measurements, and there is no model to describe the system dynamics. Thus, numerous sensors are necessary to capture the system's whole dynamics, which is practically nearly impossible for large fields. The designed controller also cannot handle the sudden changes in the system and the disturbances. The sensor-based irrigation might not optimize the overall system performance.

In [53], the automated irrigation scheduling techniques were developed and implemented using the real-time prediction from MAESTRA (Multi-Array Evaporation Stand Tree Radiation A) model. The irrigation efficiency, tree growth and the amount of water supplied were compared with the model-based irrigation (considers both model and sensors) to the sensing-based irrigation (only sensors). The author reported the growth of a tree in the method considering model is better than only sensor based method. In [54], the automatic neuro-controller was designed to keep the soil moisture in a certain range of the root zone. The neural network model was identified using the input of irrigation amount, soil property, and crop consumption, and root area soil moisture as output. The network was also retrained to adapt to

the soil properties, and crop water needs change. In [55], the artificial neural network was used to determine the irrigation amount by using the soil moisture, soil type, plant type, and irrigation period. The authors reported 20.5% of water-saving as well as 23.9% of energy saving. In [56], the Smart Irrigation Decision Support System (SIDSS) was implemented for weekly irrigation decision-based feedback from the soil moisture sensor and climatic variables. The PLSR and ANFIS machine learning techniques were used for SIDSS. In [57], the net irrigation was based on the inputs, which were evapotranspiration and percentage ratio of depletion in soil moisture using a fuzzy method. In [58], the fuzzy decision support system (FDSS) was developed to enhance irrigation. The FDSS combined with the soil moisture predictive model kept the soil moisture at a safe level. The soil moisture predictive model inputs were various growing degree days, irrigation amount to crop, and evapotranspiration. However, both the neural network and the fuzzy logic system require a huge amount of data for model training, which is difficult to get in actual applications and cannot sufficiently capture the physical dynamics of the system.

Park and co-authors used the receding horizon control (RHC) to maintain soil moisture and chemical concentration at a certain threshold [59]. The unsaturated, energy and solute model was considered to optimize the RHC method. The model parameters were estimated based on minimizing the error between sensor data and model estimates. However, the authors have used a one-dimensional model that cannot represent the horizontal diversity of the soil and crops. The authors also did not consider the crop growth model. In [60], the Model Predictive Control (MPC) was developed for irrigation scheduling. The state-space model was developed using the water balance method. The evapotranspiration and precipitation data were used as disturbances, irrigation amount was used as input, and root-zone moisture content was used as a state variable. Delgoda and co-authors developed MPC to minimize irrigation amount and root-zone soil moisture deficit subjected to the availability of irrigation water and threshold [61]. The system identification method was used to determine the variables in the water balance method and was compared with the AQUACROP model. The uncertainty in the weather forecast was considered in the MPC formulation. In [62], the predicted model was used to obtain the optimal irrigation amount by using MPC. The authors implemented the MPC framework in

the crop field. They reported 67.4% saving of water than the time-based open-loop irrigation, 40% saving of water than the feed-forward open-loop irrigation, and 14.9% water-saving than the sensor-based irrigation. Mccarthy and co-authors designed MPC using the variable rate irrigation simulation framework VARIwise [51]. The plant, soil, or the combinations of the plant, soil, or weather data were used as input variables. The soil water minimization or the square/boll count maximization was used as an objective function. However, the drawbacks are that the spatial variability (eg. soil moisture) was calculated based on the ordinary kriging-based interpolation method, which in real conditions might not be able to capture the dynamics of the system, the optimal location of the sensor placement is not described. More complete reviews of the automatic irrigation management and advanced controllers can be found in [10, 63, 64].

In Chapter 6, the challenges faced in scheduler design for large agricultural fields are addressed using a closed-loop approach.

1.3.4 Model reduction

An agro-hydrological system can be modeled using the Richards equation, a three-dimensional PDE equation. The Richards equation assumes the local equilibrium present in soil moisture content and soil water potential. The finer resolution grid (a few centimeters to a few meters) is often required to solve the Richards equation not only to ensure numerical stability but also to satisfy the local equilibrium assumption [65]. The finer resolution makes the number of states very high in the range of $(10^4 - 10^8)$. The increased system order makes the estimation and controller design very challenging, and it makes the optimization problem intractable. Model reduction is one of the widely accepted techniques to handle such types of high-dimensional systems. Model order reduction is a widely accepted technique to handle complex processes like in chemical plants. For example, Sahraei and co-authors developed a dynamic reduced-order model to capture the transient behavior of an entrained-flow gasifier using reactor network modeling approach [66].

A few popular model reduction techniques are proper orthogonal decomposition (POD), optimal Hankel norm reduction, balanced truncation methods [67]. However, these methods do not preserve the states' network topology or physical mean-

ing. Thus, these methods may not be useful while applying state constraints in the optimization-based state estimation and control design. The above methods are also not appropriate for sensor placement problems, distributed state, and controller design. In [68] and [69, 70], graph clustering-based model reduction methods were proposed. In these methods, the states in each cluster are aggregated into one single state, and the reducible clusters are merged to obtain a reduced model. In each cluster, the dynamic properties of all the states are almost the same. So the essential dynamic characteristics of the full-order system, as well as the physical topology are preserved by the reduced model. However, these graph clustering-based methods either only consider the connections between states, ignore the strength of connection, or only consider the strength of connection for linear systems. This makes the results obtained by the above methods sometimes not appropriate.

In general, the reduced model captures the dynamics of the actual system by a low-dimensional manifold. In some cases, the solution of the actual system is scattered all over the high dimensional space, or it changes over time. In such cases, one reduced model may not represent the actual system's high dimensional manifold or require a very high dimensional reduced space. In such scenarios, adaptive model reduction is one of the promising solutions. In [71, 72], the POD based adaptive model reduction is proposed.

In Chapters 3, 4, 5, 6, different structure-preserving model reduction techniques are proposed based on the various applications.

1.3.5 Remote sensing techniques for agriculture

As discussed before, it is challenging to put sensors everywhere in the field. The remote sensing images can cover a large region of the agricultural field. The estimation of surface soil moisture using remote sensing methods has been considered in many studies.

The surface soil moisture can be obtained using optical remote sensing. The optical remote sensing-based method deals with the relationship between soil moisture and spectral reflectance. Angstrom's showed the reflectance started to decrease when the soil moisture increased [73]. In [74], the relationship between relative reflectance and soil moisture was studied. The results showed that the reflectance was

inversely proportional to the soil moisture at low soil moisture content, but after a certain critical point, the reflectance was directly proportional to the soil moisture content. In [75], the results showed the soil moisture was an exponential function of reflectance. Liu and co-authors, estimated soil moisture content by different approaches such as relative reflectance method, derivative method, and difference of absorbance and reflectance method and showed that the soil moisture is a nonlinear function of reflectance [76]. In [77], the soil moisture was estimated from the reflectance in near-infrared and shortwave infrared wavelength using the inverted Gaussian function. Gao and co-authors obtained soil reflectance from the pixel decomposition of the NIR-red band, vegetation cover, and soil line and then used the soil reflectance to calculate the soil moisture [78].

The thermal remote sensing soil moisture is broadly estimated based on the thermal inertia method and temperature index method [79]. The land surface temperature represents the soil surface temperature for bare soil but represents the vegetation canopy temperature in the presence of vegetation cover. In [80], the normalized difference temperature index method (NDTI) was used to calculate the soil moisture. The Crop Water Stress Index (CWSI) indirectly reflects soil moisture. The CWSI can be calculated by following [81]:

$$CWSI = \frac{(T_c - T_a) - (T_{nws} - T_a)}{(T_c - T_a)_{max} - (T_{dry} - T_a)}$$

where T_c is the canopy temperature, T_a is the air temperature. T_{nws} is the non-water stressed temperature of canopy and T_{dry} is the water-stressed temperature of canopy.

Many studies have investigated the estimation of surface soil moisture from the combination of optical and thermal infrared remote sensing data. The surface soil moisture can be estimated from the two-dimensional feature space constructed by the vegetation index (VI)/fractional vegetation cover (FVC) and land surface temperature (LST). There are two types of methods to plot the LST and VI/FVC which are triangular method [82] and trapezoidal method [83, 84]. In a triangular method, the lowest surface temperature corresponds to the full vegetation area, but in a trapezoidal method, the lowest surface temperature corresponds to the well-watered vegetation [79]. Numerous methods have been developed to construct the feature space based on a different combination of LST and VI/FVC. In [85], five different combina-

tions of thermal and visible range data were reviewed to plot the feature space, which were 1) LST vs. VI [86, 82, 87]; 2) LST vs. albedo; 3) day-night surface temperature difference vs. VI; 4) LST and air temperature difference vs. VI; and 5) the output from the Soil Vegetation Atmosphere Transfer Model (SVAT) coupled with the LST-VI feature space. In [82], the author presented the triangular space based on LST-VI and identified the bare soil, partial vegetable cover, and full vegetable cover region. In [88], the temperature-vegetation dryness index (TVDI) was defined as:

$$TVDI = \frac{T_s - T_{smin}}{a + bNDVI - T_{smin}}$$

where T_{smin} is the minimum surface temperature, T_s is the surface temperature at the given pixel,, NDVI is normalized difference vegetation index, a and b are parameters to fit the data of $T_{smax} = a + bNDVI$. The TVDI of value 1 represents the dry edge, and value 0 represents the wet edge. The soil moisture can be directly estimated from the TDVI, which is a linear function in [89, 90] but in [91, 92], soil moisture was estimated by the nonlinear function of TVDI and FVC. Zheng and co-authors derived the dry and wet edge theoretically in place of the regression [92]. Holzman and co-authors [93] derived the soil moisture and crop yield using TVDI method and showed the strong correlation between soil moisture and TVDI ($R^2 = 0.61$ to 0.83). In [94], the Temperature Rising Rate Vegetation Index (TRRVDI) was proposed to estimate surface soil moisture. The TRRVDI was derived from the triangular method constructed using the mid-morning land surface temperature rate vs. the vegetation index. In [95], the soil moisture was estimated based on the evaporation fraction (EF), which was derived from the difference of surface temperature and air temperature vs. the vegetation index triangular plot method. In [96], the surface soil moisture was estimated based on the TVDI method for the heterogeneous regions. The drawbacks of these above-mentioned methods are the following: 1) poor spatial and temporal resolution, 2) influenced by surface roughness and vegetation cover, 3) a large amount of pixel required to estimate the extreme boundary of a wet and dry region, 4) the data obtained is not sufficient for model parameter estimation and state estimation, 5) difficult to calibrate and validate with the ground soil moisture data due to low resolution, 6) difficult to incorporate different types of uncertainty in the soil moisture models. Hassan-Esfahani and co-authors estimated surface soil moisture

using the high resolution thermal and optical imagery [97]. They developed artificial neural networks using optical data, thermal data, different types of vegetation index, and field capacity as input to estimate the soil moisture. However, the drawback of this method is the low temporal resolution which cannot provide enough information for real-time irrigation management.

Chapter 7 provides machine learning based model to estimate surface soil moisture.

1.4 Contributions and thesis outline

In this thesis, different elements of closed-loop irrigation for the large agricultural field are discussed. The thesis is arranged as follows:

In Chapter 2, the agro-hydrological system model is described. The cylindrical coordinate model is considered to capture the central pivot circular motion. The extended Kalman filter (EKF) steps are discussed. Theories of observability and degree of observability is discussed. Further, the details of cluster set and projection matrix is discussed.

In Chapter 3, the problem of optimal sensor placement for state estimation of agro-hydrological systems is addressed. A systematic approach is proposed to find the minimum number of sensors that ensure the entire system's observability and then to find the optimal locations of the sensors in terms of the degree of observability. In the proposed procedure, the key steps include order reduction of the large-scale system model, exploration of the minimum number of sensors needed for state estimation, and optimal placement of the sensors in the soil. Three different scenarios are considered, and optimal sensor placement is addressed for all the scenarios using the proposed procedure. Simulation results show the effectiveness of the proposed procedure and methods.

In Chapter 4, the optimal sensor placement for state estimation of agro-hydrological systems in the presence of a variable input system is proposed. A systematic approach is proposed to find the best locations of the sensors for agro-hydrological systems. In the proposed procedure, the key steps include order reduction of the large-scale system model, exploration of the minimum number of sensors needed for state estimation, and optimal placement of the sensors. Simulation results show the effectiveness of

the proposed approach.

In Chapter 5, a systematic approach for state estimation of large agricultural fields is proposed. A structure-preserving adaptive model reduction method using trajectory-based unsupervised machine learning techniques is designed. Furthermore, an adaptive MHE algorithm is developed based on an adaptive reduced model. The proposed algorithms are applied to a small simulated field to compare the performance of adaptive MHE over original MHE. Finally, the proposed approach is applied to a large-scale real agricultural field to test the effectiveness and superiority to address the current challenges. Extensive simulations are carried out to show the efficiency of the proposed approach.

In Chapter 6, the high dimensionality of the agricultural field and propose a systematic approach to provide optimum irrigation amount and irrigation time for three-dimensional agro-hydrological systems is proposed. First, a structure-preserving model reduction technique to decrease the dimension of the system model is introduced. Using the reduced model, the optimization-based closed-loop scheduler is designed in a model predictive control (MPC) environment. The closed-loop approach can handle weather disturbances and provide improved yield and water conservation. The primary objective of the proposed scheduler is to ensure maximum yield, minimum water consumption and maximize the time between the two irrigation events, which results in less electricity usage. The proposed approach is applied to three different scenarios to show the effectiveness and superiority of the proposed framework.

In Chapter 7, the surface soil moisture estimation using remote sensing images is proposed. The LSTM based machine learning model is developed to convert the multispectral images to soil moisture. The detailed image processing steps to use the remote sensing images in the framework of LSTM is provided. The proposed algorithm is applied to actual data collected in summer 2019 at Lethbridge. The results show the effectiveness of the proposed approach.

Chapter 8 summarizes the contributions of the work and proposes a few research directions.

Chapter 2

Preliminaries on modeling of agro-hydrological systems

2.1 System description

The dynamics of the water flows in the agro-hydrological system are governed by the Richards equation as follows [13]:

$$\frac{\partial \theta}{\partial t} = c(h, X)\frac{\partial h}{\partial t} = \nabla.(K(h, X).\nabla(h + z)) + S(h, X) \tag{2.1}$$

where X (cm) is the three-dimensional position vector, h (cm) is the field water pressure head, θ is the field soil moisture or soil water content, c (cm^{-1}) is the soil water capacity, K ($\mathrm{cm{\cdot}h}^{-1}$) is the hydraulic conductivity, z (cm) is the vertical coordinate, S (h^{-1}) is the source and sink term which includes the evapo-transpiration and can be calculated based on the Penmon-Moneith equation [98]. The relation between the hydraulic conductivity ($K(h, X)$) and the pressure head (h) is characterized as follows [99]:

$$K(h, X) = \begin{cases} K_{sat} S_e^{\lambda}[1 - (1 - (S_e^{\frac{1}{m}})^m]^2, & S_e < 1 \\ K_{sat}, & S_e \geq 1 \end{cases} \tag{2.2}$$

where $S_e = 1 + (-\alpha h)^n$, λ, α, m, n are the shape factors and K_{sat} is the saturated hydraulic conductivity. The capillary capacity ($c(h, X)$) can be computed as follows [100]:

$$c(h, X) = \begin{cases} (\theta_s - \theta_r)\alpha n(1 - (\frac{1}{n}))(-\alpha h)^{n-1}(1 + (-\alpha h)^n)^{-(2-(1/n)}, & h < 0 \\ S_r, & h \geq 0 \end{cases} \tag{2.3}$$

where θ_s denotes the saturated soil water content, θ_r denotes the residual water content and S_r is the specific storage coefficient of the porous medium under positive

pressure. The relation between moisture content (θ) and the pressure head can be expressed as follows [100]:

$$\theta = \begin{cases} (\theta_s - \theta_r)(1 + (-\alpha h)^n)^{-(1-(1/n))} + \theta_r, & h < 0 \\ \theta_s, & h \geq 0 \end{cases} \tag{2.4}$$

The sink term $S(h, z)$ in (2.1) characterizes the root water extraction rate. The total root water uptake depends upon transpiration rate, soil pressure head and root depth. The mathematical formulation of root-water uptake based on Feddes model is expressed as follows [101]:

$$S(h, z) = \alpha(h) S_{max}(z) \tag{2.5}$$

where $S_{max}(z)[\mathrm{m^3 m^{-3} s^{-1}}]$ is the maximum possible water extraction rate under optimal condition. $\alpha(h)$[-] is the dimensionless water stress factor. The water stress factor depends upon the pressure head values and can be expressed as follows:

$$\alpha(h) = \begin{cases} 0, & h > h_1 \\ \frac{h1-h}{h1-h2}, & h_1 > h \geq h_2 \\ 1, & h_2 > h \geq h_3 \\ \frac{h-h_4}{h_3-h_4}, & h_3 > h \geq h_4 \\ 0, & h_4 > h \end{cases} \tag{2.6}$$

where h_1 is the pressure head value above which the root doesn't extract any water, h_2 and h_3 are the upper and lower threshold values between which the root water extraction is maximum. h_4 is the permanent wilting point, below which the root water extraction is zero. The water stress factor graph is shown in Figure 2.1. The stress-free zone is in between h_2 (field capacity) and h_3 (limiting point).

The maximum possible water extraction rate $S_{max}(z)$ can be calculated using Feddes model [101]:

$$S_{\max}(h, z) = \frac{TP_p}{L} \tag{2.7}$$

where TP_p $[\mathrm{ms^{-1}}]$ is the potential transpiration rate and L [m] is the rooting depth. The potential transpiration rate TP_p is computed by:

$$TP_p = ET_p - EV \tag{2.8}$$

where ET_p $[\mathrm{ms^{-1}}]$ is the potential evapotranspiration rate and EV $[\mathrm{ms^{-1}}]$ is the potential evaporation rate. The EV can be expressed as follows:

$$EV = ET_p e^{(-0.623 LAI)} \tag{2.9}$$

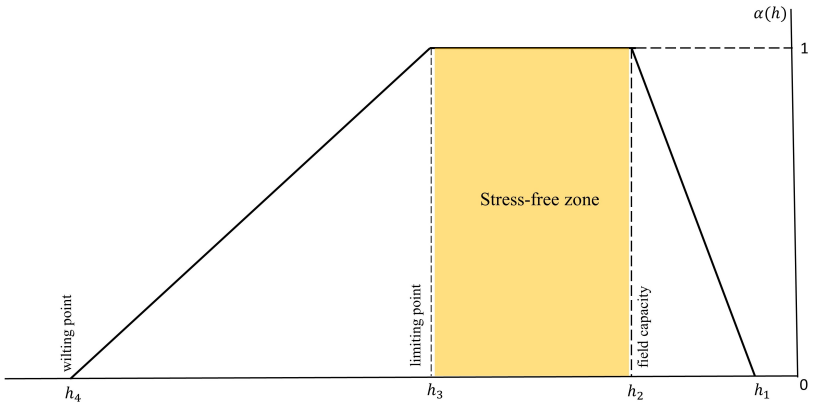


Figure 2.1: Water stress factor $\alpha(h)$ graph

where LAI is the leaf area index. The potential evapotranspiration ET_p is computed as follows:

$$ET_p = K_c PET \tag{2.10}$$

where PET [ms^{-1}] is the reference evaporation rate which can be calculated based on the Penmon-Moneith equation [98] and K_c [-] is the crop coefficient.

The crop yield model is also important along with the Richards equation. The crop yield model is modeled as a function of actual and potential evaporation and as follows:

$$\left(1 - \frac{Y_a}{Y_p}\right) = \sum_{k=1}^{T} K_y(k)\left(1 - \frac{ET_a}{ET_p}\right) \tag{2.11}$$

where Y_a is actual yield and Y_p is potential yield. $K_y(k)$ is the crop sensitivity factor at time k. T is the total time for growing seasons. ET_a is the actual evapotranspiration. ET_a can be represented as:

$$ET_a = \alpha(h) ET_p \tag{2.12}$$

where $\alpha(h)$ is calculated from (2.6) and ET_p is calculated from (2.10).

In (2.11), $(1 - \frac{Y_a}{Y_p})$ represents the yield deficiency and $(1 - \frac{ET_a}{ET_p})$ represents the ET deficiency. By combining (2.12) and (2.11), the relationship between the crop yield and soil pressure head is expressed as follows:

$$\left(1 - \frac{Y_a}{Y_p}\right) = \sum_{k=1}^{T} K_y(k)\left(1 - \alpha(h)\right) \tag{2.13}$$

In this thesis for some chapters, the agricultural field is equipped with a center pivot irrigation system is considered. A center pivot irrigation system rotates across

the field around a fixed pivot at the center of the field and irrigates in a circular manner. In order to account for the circular movement of the center pivot irrigation system, the Richards equation in (2.1) is expressed in the cylindrical coordinates as follows [22]:

$$c(h)\frac{\partial h}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right] + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\frac{K(h)}{r}\frac{\partial h}{\partial \theta}\right] + \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z}+1\right)\right] + S(h,z) \quad (2.14)$$

The explicit finite difference method to discretize the Richards equation in (2.1) and (2.14) is applied. Note that spatial discretization of the model is performed, such that a continuous-time state-space model is established as in the following form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= Cx(t) \end{aligned} \quad (2.15)$$

where x represents the pressure head and u denotes the irrigation amount. Each discretized node is one state variable in x. For a three-dimensional field, the number of discretized nodes can be very big. In (2.15), y denotes the measured outputs and it is considered that we have direct measurements of some of the state variables, which reflects a common case in agro-hydrological systems. The details of the disretization steps for cylindrical coordinates are discussed in the next section.

As the input (irrigation amount) is applied to each surface node, it is incorporated in the system surface boundary condition. The surface boundary condition is characterized by Neumann boundary condition as follows:

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

where h is the pressure head, $u(t)$ is the input to the system, $K(h)$ is the hydraulic conductivity and Z_s is the length of the soil column. The bottom boundary condition is specified as free drainage.

2.2 Numerical method and discretization

The numerical discretization of the cylindrical version of Richards equation is developed in [22]. The Richards equation in (2.1) has been expressed in the cylindrical

coordinates. The gradient (∇) and divergence ($\nabla.$) operators are defined as below:

$$\nabla = \frac{\partial}{\partial r}\hat{r} \; + \; \frac{1}{r}\frac{\partial}{\partial \theta}\hat{\theta} \; + \; \frac{\partial}{\partial z}\hat{z} \tag{2.16}$$

$$\nabla.G = \frac{1}{r}\frac{\partial}{\partial r}(rG_r) + \frac{1}{r}\frac{\partial G_\theta}{\partial \theta} + \frac{\partial G_z}{\partial z} \tag{2.17}$$

where $\hat{\theta}$, $\hat{r}$, $\hat{z}$ are the unit vectors in the azimuthal, radial and axial directions respectively. G is the continuously differentiable vector field and the G_r, G_θ, and G_z are the components. Using the operators the Richards equation in (2.1) can be expressed in cylindrical coordinates as follows:

$$c(h)\frac{\partial h}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right] + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\frac{K(h)}{r}\frac{\partial h}{\partial \theta}\right] + \frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z} + 1\right)\right] + S(h,z) \tag{2.18}$$

The resulting cylindrical Richards equation is a non-linear elliptic and parabolic partial differential equation (PDE). The analytical solution of the Richards equation is difficult to solve because of its non-linearity and high dimension. The finite-difference numerical is used to solve the (2.18). The PDE is converted to ordinary differential equations (ODEs) by discretizing the spatial directions. The resulting ODEs can be solved using well-established numerical schemes. The central difference is employed to discretize the spatial directions, and the details is described below.

Discretization in the Radial Direction

$$\frac{1}{r}\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right]\Bigg|_{e_r,e_\theta,k} \approx \frac{1}{r_{e_r,e_\theta,k}\Delta r_i}\left[r_{e_r+\frac{1}{2},e_\theta,k}K_{e_r+\frac{1}{2},e_\theta,k}(h)\left(\frac{h_{e_r+1,e_\theta,k} - h_{e_r,e_\theta,k}}{\Delta r_E}\right) - r_{e_r-\frac{1}{2},e_\theta,k}K_{e_r-\frac{1}{2},e_\theta,k}(h)\left(\frac{h_{e_r,e_\theta,k} - h_{e_r-1,e_\theta,k}}{\Delta r_W}\right)\right] \tag{2.19}$$

where $e_r \in [0, N_r+1]$, $e_\theta \in [0, N_\theta+1]$ and $k \in [0, N_z+1]$ represent the position indices in the radial, azimuthal and axial directions respectively. N_r, N_θ and N_z are the total number of nodes in the r, θ and z direction respectively. $\Delta r_E = r_{e_r+1,e_\theta,k} - r_{e_r,e_\theta,k}$, $\Delta r_W = r_{e_r,e_\theta,k} - r_{e_r-1,e_\theta,k}$ and $\Delta r_i = \frac{1}{2}(\Delta r_E + \Delta r_W)$. $r_{e_r\pm\frac{1}{2},e_\theta,k} = \frac{1}{2}(r_{e_r,e_\theta,k} + r_{e_r\pm1,e_\theta,k})$ and $K_{e_r\pm\frac{1}{2},e_\theta,k}(h) \approx \frac{1}{2}(K(h_{e_r,e_\theta,k}) + K(h_{e_r\pm1,e_\theta,k}))$. All the nodes in the radial direction are in the center of the e_r^{th} compartment. The subscripts E, C, and W correspond to the coordinates $(e_r{+}1,\ e_\theta,\ k)$, $(e_r,\ e_\theta,\ k)$ and $(e_r{-}1,\ e_\theta,\ k)$ respectively.

Discretization in the Azimuthal Direction

$$\frac{1}{r}\frac{\partial}{\partial\theta}\left[\frac{K(h)}{r}\frac{\partial h}{\partial\theta}\right]\Bigg|_{e_r,e_\theta,k} \approx \frac{1}{r_{e_r,e_\theta,k}\Delta\theta_j}\left[\frac{K_{e_r,e_\theta+\frac{1}{2},k}(h)}{r_{e_r,e_\theta+\frac{1}{2},k}}\left(\frac{h_{e_r,e_\theta+1,k}-h_{e_r,e_\theta,k}}{\Delta\theta_T}\right)-\right.$$
$$\left.\frac{K_{e_r,e_\theta-\frac{1}{2},k}(h)}{r_{e_r,e_\theta-\frac{1}{2},k}}\left(\frac{h_{e_r,e_\theta,k}-h_{e_r,e_\theta-1,k}}{\Delta\theta_D}\right)\right] \quad (2.20)$$

where $\Delta\theta_T = \theta_{e_r,e_\theta+1,k} - \theta_{e_r,e_\theta,k}$, $\Delta\theta_D = \theta_{e_r,e_\theta,k} - \theta_{e_r,e_\theta-1,k}$ and $\Delta\theta_j = \frac{1}{2}(\Delta\theta_T + \Delta\theta_D)$. $r_{e_r,e_\theta\pm\frac{1}{2},k} = \frac{1}{2}(r_{e_r,e_\theta,k} + r_{e_r,e_\theta\pm1,k}) = r_{e_r,e_\theta,k}$ and $K_{e_r,e_\theta\pm\frac{1}{2},k}(h) \approx \frac{1}{2}(K(h_{e_r,e_\theta,k}) + K(h_{e_r,e_\theta\pm1,k}))$. All the nodes in the azimuthal direction are in the center of the e_θ^{th} compartment. The subscripts T, C, and D correspond to the coordinates $(e_r, e_\theta + 1, k)$, (e_r, e_θ, k) and $(e_r, e_\theta - 1, k)$ respectively.

Discretization in the Axial Direction

$$\frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z}+1\right)\right]\Bigg|_{e_r,e_\theta,k} \approx \frac{1}{\Delta z_k}\left[K_{e_r,e_\theta,k+\frac{1}{2}}(h)\left(\frac{h_{e_r,e_\theta,k+1}-h_{e_r,e_\theta,k}}{\Delta z_N}+1\right)\right.$$
$$\left.- K_{e_r,e_\theta,k-\frac{1}{2}}(h)\left(\frac{h_{e_r,e_\theta,k}-h_{e_r,e_\theta,k-1}}{\Delta z_S}+1\right)\right] \quad (2.21)$$

where $\Delta z_N = z_{e_r,e_\theta,k+1} - z_{e_r,e_\theta,k}$, $\Delta z_S = z_{e_r,e_\theta,k} - z_{e_r,e_\theta,k-1}$, $\Delta z_k = \frac{1}{2}(\Delta z_N + \Delta z_S)$ and $K_{e_r,e_\theta,k\pm\frac{1}{2}}(h) \approx \frac{1}{2}(K(h_{e_r,e_\theta,k}) + K(h_{e_r,e_\theta,k\pm1}))$. All the nodes in the axial direction are in the center of the k^{th} compartment. The subscripts N, C, and S correspond to the coordinates $(e_r,\ e_\theta,\ k+1)$, $(e_r,\ e_\theta,\ k)$ and $(e_r,\ e_\theta,\ k-1)$ respectively.

The resulting ODE, in terms of the temporal variable, is obtained by substituting (2.19), (2.20) and (2.21) into (2.18) as follows:

$$\frac{dh}{dt} = \frac{1}{C_{e_r,e_\theta,k}(h)}\left[\left(\frac{1}{r_{e_r,e_\theta,k}\Delta r_i}\left[r_{e_r+\frac{1}{2},e_\theta,k}K_{e_r+\frac{1}{2},e_\theta,k}(h)\left(\frac{h_{e_r+1,e_\theta,k}-h_{e_r,e_\theta,k}}{\Delta r_E}\right)-\right.\right.\right.$$
$$\left.\left.\left. r_{e_r-\frac{1}{2},e_\theta,k}K_{e_r-\frac{1}{2},e_\theta,k}(h)\left(\frac{h_{e_r,e_\theta,k}-h_{e_r-1,e_\theta,k}}{\Delta r_W}\right)\right]\right)+$$
$$\left(\frac{1}{r_{e_r,e_\theta,k}\Delta\theta_j}\left[\frac{K_{e_r,e_\theta+\frac{1}{2},k}(h)}{r_{e_r,e_\theta+\frac{1}{2},k}}\left(\frac{h_{e_r,e_\theta+1,k}-h_{e_r,e_\theta,k}}{\Delta\theta_T}\right)-\right.\right.$$
$$\left.\left.\frac{K_{e_r,e_\theta-\frac{1}{2},k}(h)}{r_{e_r,e_\theta-\frac{1}{2},k}}\left(\frac{h_{e_r,e_\theta,k}-h_{e_r,e_\theta-1,k}}{\Delta\theta_D}\right)\right]\right)+$$

$$\left(\frac{1}{\Delta z_k}\left[K_{e_r,e_\theta,k+\frac{1}{2}}(h)\left(\frac{h_{e_r,e_\theta,k+1}-h_{e_r,e_\theta,k}}{\Delta z_N}+1\right)\right.\right.$$
$$\left.\left.-K_{e_r,e_\theta,k-\frac{1}{2}}(h)\left(\frac{h_{e_r,e_\theta,k}-h_{e_r,e_\theta,k-1}}{\Delta z_S}+1\right)\right]\right)-S(h,z)\Bigg] \quad (2.22)$$

Treatment of the Axis

At the axis ($r = 0$), the singularity occurs for the first and second term of the (2.18). The L'Hopital's rule is applied to deal with this issue. The resulting equations are expressed as follows:

$$\lim_{r\to 0}\frac{1}{r}\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right]=\lim_{r\to 0}\frac{\frac{\partial}{\partial r}\left(\frac{\partial}{\partial r}\left[rK(h)\frac{\partial h}{\partial r}\right]\right)}{\frac{\partial}{\partial r}(r)}=2\frac{\partial}{\partial r}\left(K(h)\frac{\partial h}{\partial r}\right) \quad (2.23)$$

$$\lim_{r\to 0}\frac{1}{r}\frac{\partial}{\partial \theta}\left[\frac{K(h)}{r}\frac{\partial h}{\partial \theta}\right]=\lim_{r\to 0}\frac{\frac{\partial}{\partial r}\left(\frac{\partial}{\partial \theta}\left[K(h)\frac{\partial h}{\partial \theta}\right]\right)}{\frac{\partial}{\partial r}(r^2)}=\frac{1}{2}\frac{\partial}{\partial r}\left(\frac{\partial}{\partial r}\left(\frac{\partial}{\partial \theta}\left[K(h)\frac{\partial h}{\partial \theta}\right]\right)\right) \quad (2.24)$$

Substitution of (2.23) and (2.24) into (2.18) yields the cylindrical coordinates version of the Richards equation at $r = 0$:

$$C(h)\frac{\partial h}{\partial t}=2\frac{\partial}{\partial r}\left(K(h)\frac{\partial h}{\partial r}\right)+\frac{1}{2}\frac{\partial}{\partial r}\left(\frac{\partial}{\partial r}\left(\frac{\partial}{\partial \theta}\left[K(h)\frac{\partial h}{\partial \theta}\right]\right)\right)+\frac{\partial}{\partial z}\left[K(h)\left(\frac{\partial h}{\partial z}+1\right)\right]$$
$$-S(h,z) \quad (2.25)$$

Boundary Conditions

The boundary condition required to solve the ((2.22)) is expressed as follows:

$$\frac{\partial h(r,\theta,z,t)}{\partial r}=0 \qquad \text{at} \quad (r=0,\ \theta,\ z) \quad (2.26)$$

$$\frac{\partial h(r,\theta,z,t)}{\partial r}=0 \qquad \text{at} \quad (r=H_r,\ \theta,\ z) \quad (2.27)$$

$$h(r=0,\ \theta,\ z,\ t)=h(r=0,\ \theta=0,\ z,\ t) \quad (2.28)$$

$$h(r,\ \theta = 0,\ z,\ t) = h(r,\ \theta = 2\pi,\ z,\ t) \tag{2.29}$$

$$\frac{\partial(h\,(r,\theta,z,t))}{\partial z} = 0 \qquad \text{at} \quad (r,\ \theta,\ z = 0) \tag{2.30}$$

$$\frac{\partial(h\,(r,\theta,z,t))}{\partial z} = -1 - \frac{u_{\text{irr}}}{K(h)} \qquad \text{at} \qquad (r,\ \theta,\ z = H_z) \tag{2.31}$$

The zero gradient boundary condition is imposed at $r = 0$ and $r = H_r$ (H_r is the total radius of the field) as shown in (2.26) and (2.27). Equation (2.28) describes the coordinates of the dependent variable, the pressure head (h), that coincide at $r = 0$. After one revolution, the starting point coincides with the ending point and this is represented by (2.29). Equation (2.30) represents the free drainage boundary condition that is applied at the bottom of the field ($z = 0$). The Nuemann boundary condition, shown by (2.31), is applied at the top of the field ($z = H_z,\ H_z$ is the depth of the field). u_{irr} (m/s) in (2.31) represents the irrigation rate which is considered as the input in this work.

2.3 Extended Kalman Filter (EKF)

The EKF is a popular state estimation method for nonlinear systems where the successive linearization is performed around the previous time state estimates [102]. The EKF comprises two important steps, the prediction, and update step. In the prediction step, the state and covariance are predicted using the model for the next time. In the update step, the state and covariance are updated using the gain matrix when the measurement arrives. The details of each step are expressed below.

Initialization

(1) The discrete time model is obtained using the continuous time system (2.15) and can be expressed as:

$$x_{k+1} = F(x_k, u_k) + \omega(k)$$
$$y_k = Cx_k + v(k)$$

(2) The filter is initialised as follows:

$$E[x_0] = \hat{x}_0$$
$$E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] = P(0|0)$$

Prediction Step

(1) The predicted of the system at time t_{k+1}

$$\hat{x}_{k+1|k} = F(\hat{x}_{k|k}, u_k)$$

(2) The state covariance matrix prediction:

$$P_{(k+1|k)} = A_k P_{(k|k)} A_k^T + Q$$

where $A_k = \frac{\partial F}{\partial x}\big|_{\hat{x}_{k|k}, u_k}$ and Q is the covariance matrix of the process disturbance ω.

Update Step

The observation y_{k+1} at time t_{k+1} is used to update the covariance matrix and predicted state.

(1) Kalman gain matrix, K_{k+1} calculation

$$K_{k+1} = P_{(k+1|k)} C_{k+1}^T [C_{k+1} P_{(k+1|k)} C_{k+1}^T + R]^{-1}$$

where R is measurement noise's covriance matrix v.

(2) State update

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}[y_{k+1} - C\hat{x}_{k+1|k}]$$

(3) Covariance update

$$P_{(k+1|k+1)} = [I - K_{k+1} C_{k+1}] P_{(k+1|k)}$$

2.4 Observability

A system is said to be observable if the system states can be recovered using the measured outputs. Observability analysis plays a vital role in state and parameter estimation as well as in feedback control. The observability of a system can be checked using observability matrix, Lie derivatives, and PBH test [103]. These methods can only give whether a system state is observable or not and cannot give any information about selecting the sensor node or the minimum number of sensor sets required for the system to be observable. For large-scale linear systems executing the rank test of observability matrix may be ill-conditioned and may suffer numerical issues.

For nonlinear systems, the calculation of the Lie derivative for a large system is computationally restricted and applicable for only small-scale systems. In this section, two methods are discussed to obtain the minimum number of sensors.

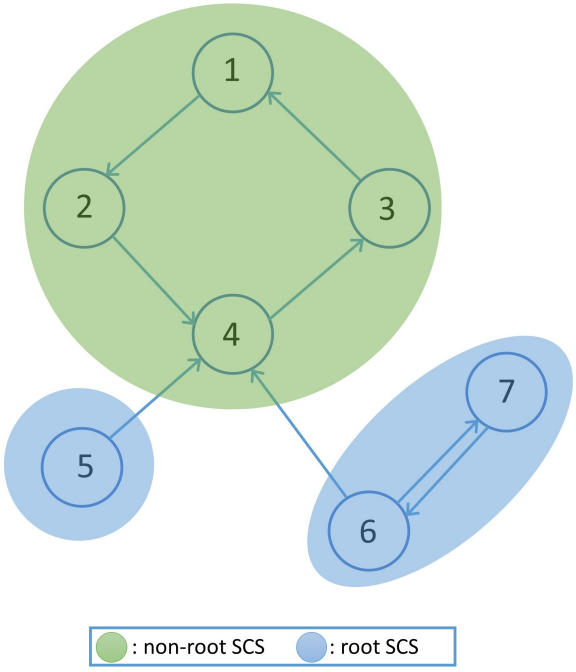


Figure 2.2: Structural observability

2.4.1 Structural observability

Liu and co-authors proposed an efficient algorithm to calculate the minimum number of sensor nodes using the graphical representation of the nonlinear system dynamics [103]. The procedure comprises the following steps:

1. Represent the nonlinear system as a directed graph. For a general nonlinear system like (2.15), if $\partial f_k(x)/\partial x_i \neq 0, k, i = 1, 2, \ldots n$, then there is a direct link from $x_k \rightarrow x_i$

2. Decompose the graph into strongly connected sets (SCSs). An SCS is a set which includes nodes such that each of them is reachable from other nodes in the set.

3. Determine the root SCSs that have no incoming links from other nodes (Figure 2.2).

4. The minimum number of sensors is the same as the number of root SCSs. A measurement node should be selected from each root SCS.

2.4.2 Maximum multiplicity theory

The maximum multiplicity theory is based on the linearized system matrix A of the nonlinear system [104] (supplementary information). The maximum multiplicity theory requires only A matrix information and does not require C matrix information to determine the minimum number of sensors. The PBH test suggests that the system is unobservable if the right eigenvector of A is in the null space of the output matrix C. The maximum multiplicity theory extended the PBH test and derived the minimum number of sensors must be more than or equal to the largest geometric multiplicity of eigenvalues of A. Specifically, the minimum number of sensors N_D can be computed as follows:

$$N_D = \max\{n - \text{rank}(\lambda_i I - A)\} \tag{2.32}$$

where λ_i are the eigenvalues of A and n is the order of the linearized system.

2.5 Degree of observability

Degree of observability quantifies the of observability of the system. When multiple sensor placement solutions exist, it is preferred to maximize the degree of observability. In this thesis, the use modal degree of observability is proposed. Modal degree of observability analyzes the ability of a sensor node to estimate each mode of the

system. The PBH test tells that the j^{th} mode is not observable by i^{th} node if the entry of the right eigenvector (v_{ij}) is zero. Gu et al. [105] extended the PBH test for controllability and suggested that if the entry of left eigenvector (w_{ij}) is small then the mode j is poorly controllable from the node i. Based on this approach, it is extended to the observability and it can be identified which mode is difficult to estimate from the sensor node. For a specific node i, if a sensor is placed, the normalized measure of the modal degree of observability is given below:

$$\mathscr{O}_i = \sum_{j=1}^{n} (1 - \lambda_j^2(A_d)) v_{ij}^2 \tag{2.33}$$

where A_d is the discretized model Jacobian matrix which is derived from $A_d = e^{AT}$ and T is the sampling time and λ_j $(j = 1, \ldots, r)$ are the eigenvalues of matrix A_d, n is the order of the system.

2.6 Cluster set and projection matrix

In this section, the definition of the structure preserving cluster sets from the graph or trajectories are defined. In this thesis, the reduced-order systems are constructed based on the Petrov-Garlerkin projection [67] and the projection matrix definition is expressed in this section. Inspired by the work of Cheng et al. [70] and Ishizaki et al. [68], the definitions of clusters and projection matrix are given below.

Definition 1: Let $\mathscr{C} = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_r\}$ be the collection of r cluster sets. A cluster is a non-empty set with the following properties: i) $\mathscr{C}_i \cap \mathscr{C}_j = \Phi$ and ii) $\mathscr{C}_1 \cup \mathscr{C}_2 \cup \ldots \cup \mathscr{C}_r = \mathscr{N}$ where $\mathscr{N}$ is the total number of states.

Definition 2: The projection matrix is defined as $\mathscr{U} \in \mathbb{R}^{n \times r}$, whose elements are expressed as follows:

$$\mathscr{U}_{i,j} = \begin{cases} w_i, & \text{if vertex } i \in \mathscr{C}_j \\ 0, & \text{otherwise} \end{cases} \tag{2.34}$$

and w_i is determined as follows:

$$w_i = 1/||\alpha_i||$$

$$\alpha_i = \mathbb{E}_i^T \alpha$$

where $\alpha = [1, \ldots, 1]^T \in \mathbb{R}^n$, $||\alpha_i||$ is the L_2 norm of α_i, $\mathbb{E}_i = e_{\mathscr{C}_i} \in \mathbb{R}^{n \times m}$ (m is the size of set $\mathscr{C}_i$) denotes a matrix whose columns are e_j's ($j \in \mathscr{C}_i$) and each e_j is the j-th column of the identity matrix of size $\mathbb{R}^{n \times n}$.

Chapter 3

Optimal sensor placement

3.1 Introduction

In this chapter, a systematic procedure to find the minimum number of sensors and the optimal sensor placement for three-dimensional agro-hydrological systems is proposed. The proposed procedure includes a few key steps: model reduction, a minimum number of sensor determination, and optimal sensor placement. First, a structure-preserving weighted directed graph based model reduction method is proposed. For the weighted directed graph, the directed edges are calculated using the connection between states, and the weights are calculated using a Jacobian matrix of the nonlinear system at a steady-state value. The similarities between the states are calculated using the weights from the graph, and cluster sets are created. The original nonlinear system is projected to a nonlinear reduced-order system using the projection matrix based on cluster sets. Subsequently, an effective approach to perform observability analysis for agro-hydrological systems is introduced. In the proposed approach, structural observability and the maximum multiplicity theory are used together to find the minimum number of sensors. Once the minimum number of sensors is determined, we propose to use the modal degree of observability further to determine the optimal sensor placement. Under three scenarios, the proposed procedure and methods are applied to the agro-hydrological system. The simulation results illustrate the applicability and effectiveness of the proposed procedure and methods.

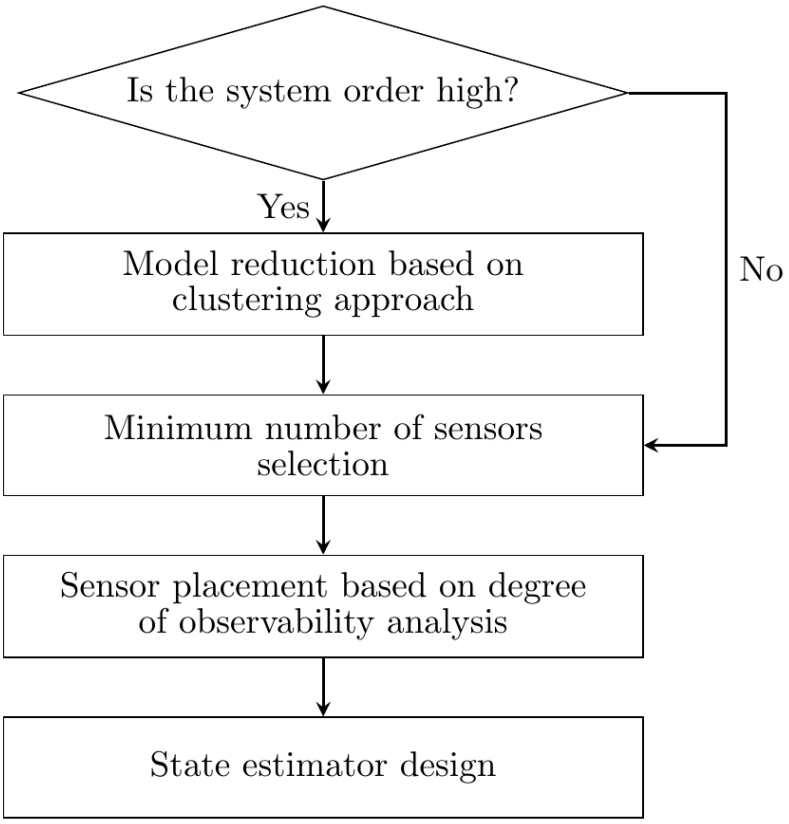


Figure 3.1: Flow diagram of the proposed sensor placement procedure

3.2 Proposed sensor placement procedure

In this section, a systematic procedure is proposed that can be used to determine the minimum number of soil moisture sensors that ensure the observability of the entire system and then to determine the optimal placement of the sensors in the field so that the degree of observability is maximized for the purpose of state estimation. A flow diagram of the proposed procedure is shown in Figure 3.1. The key steps involved in this procedure will be described in this section.

3.2.1 Model reduction

Let us consider the three-dimensional agro-hydrological system described in Chapter 2. The discretization of the Richards equation in (2.1) may lead to a model in the form of (2.15) with a very large number of state variables; that is, the dimension of x is very high. This makes it very challenging to perform sensor placement and state estimation directly. Model reduction is required to handle such large-scale systems.

In this section, we propose a graph-based clustering method that takes into account the weights on the graph edges explicitly.

Given that the objective of model reduction is to facilitate the sensor placement, it is desired that the reduced-order model preserves the spatial distribution of soil types (i.e., the network topology) of the original system. Different types of soil have different soil properties. In an agro-hydrological system, soil properties affect the strength of connection but not the connectivity between states given that all the states are obtained through discretization of the same Richards equation. A clustering approach based only on the connectivity between states is not able to capture the soil properties. In the proposed approach, weights that reflect the strength of connections between states are considered in clustering to capture the soil properties. The proposed method is very suitable for sensor placement problems, which will be demonstrated in the simulations section.

Let us represent the system in (2.15) as a directed graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N} = \{1, 2, \ldots, n\}$ denotes the states and $\mathscr{E} \subset \mathscr{N} \times \mathscr{N}$ denotes the directed edges. Let us also use $a_{i,j}$ to denote the weight from state j to state i in the directed graph. In this chapter, we propose to calculate the weights of edges based on the Jacobian of the nonlinear system (2.15) obtained at a steady-state; that is, $a_{i,j}$ is the corresponding element in the matrix $A = \frac{\partial f}{\partial x}\big|_{(x_s,u_s)}$ with (x_s, u_s) being a steady-state.

Next, the cluster sets of the graph is computed. We propose a weight based approach to cluster the states. Specifically, we propose to measure the similarity of two states based on the difference between the edge weights. For example, the similarity of state i and state j is measured by the difference between the two weights $a_{i,j}$ and $a_{j,i}$:

$$\mathscr{D}_{i,j} = |a_{i,j} - a_{j,i}| \tag{3.1}$$

When $\mathscr{D}_{i,j}$ is small, it implies that the two states (i and j) are similar and may be clustered together. Based on $\mathscr{D}_{i,j}$ for all the states, we can construct a dissimilarity matrix $\mathscr{D}$. Algorithm 1 summarizes the proposed clustering approach. The properties of the cluster sets are defined in Chapter 2.

Finally, the reduced-order system is constructed based on the Petrov-Galerkin projection framework as discussed in Chapter 2 and the reduced model for system

(2.15) is expressed below:

$$\begin{aligned} \dot{\xi}(t) &= f_r(\xi(t), u(t)) \\ y(t) &= C\mathscr{U}\xi(t) \end{aligned} \tag{3.2}$$

where $f_r(\xi(t), u(t)) = \mathscr{U}^T f(\mathscr{U}\xi(t), u(t))$ and $\xi(t) = \mathscr{U}^T x(t)$. Note that the actual state x can be approximated based on the mapping $x(t) \approx \mathscr{U}\xi(t)$. In the reduced model (3.2), the dimension of the reduced state vector ξ is a tuning parameter and determines the size of the reduced model.

Algorithm 1 Proposed clustering approach

Require: Creation of projection matrix $\mathscr{U}$
Input: A matrix, reduced model order r, model order n
Output: Reduced $\hat{A}$ matrix, Reduce order nonlinear system.

Initialization $i \leftarrow n, \hat{A} \leftarrow A$,
while $i > r$ **do**
  Compute $\mathscr{D}$ based on (3.1)
  Find the smallest element δ in $\mathscr{D}$
  Find the two states (p, q) corresponding to value δ
  Merge state p and state q into a single cluster
  Compute the projection matrix at the current iteration $\mathscr{U}^{(i)}$ based on *Definition 2* from Chapter 2
  Update $\hat{A}$ using the projection matrix and current $\hat{A}$: $\hat{A} \leftarrow \mathscr{U}^{(i),T}\hat{A}\mathscr{U}^{(i)}$
  Save $\mathscr{U}^{(i)}$ matrix at each iteration
  $i = i - 1$
end while
Compute final projection matrix $\mathscr{U} = \Pi_{i=r+1}^{n}\mathscr{U}^{(i)}$,
Compute reduced nonlinear model $f_r(\xi(t), u(t)) = \mathscr{U}^T f(\mathscr{U}\xi(t), u(t))$ and $\xi(t) = \mathscr{U}^T x(t)$

Remark 1 *Note that for an agro-hydrological system described by the Richards equation, if the boundary conditions ensure that the system has a steady-state, then the agro-hydrological system (2.15) is asymptotically stable [106]. Following similar arguments as in Cheng et al. [70] and Ishizaki et al. [68], it can be derived that the reduced model (3.2) preserves the stability of the original nonlinear system (2.15). According to Cheng et al. [70] and Ishizaki et al. [68], this further implies that the mismatch between the reduced model and the original nonlinear model is bounded for all time if the two systems have the same inputs. In order to pick an appropriate order of the reduced model, it is recommended to perform extensive simulations to study*

the mismatch between reduced models of different orders and the original nonlinear system and pick the order according to the accuracy needed for a specific application.

3.2.2 Minimum number of sensors

The objective of this step is to determine the minimum number of sensors that ensures the observability of the entire state vector of the agro-hydrological system (2.15). This will be done indirectly based on the reduced model (3.2).

In this section, we propose to use both the graphical approach introduced by Liu et al. [103] to check the structural observability and the maximum multiplicity theory [104] to further check the results given by the graphical approach. Details of both the methods are discussed in Chapter 2. The graphical approach for structural observability can handle nonlinear systems directly and can also handle very large-scale systems. This makes it very suitable for agro-hydrological systems considered. While the structural observability is effective and can be used to identify the minimum number of sensors, it does not consider the strength of connection between nodes. For example, when the connection between two nodes is insignificant or the weights on the links of two nodes (e.g., $a_{i,j}$ and $a_{j,i}$) are similar, the results given by the structural observability may be not sufficient. Also, if there are self-loops in the graph, the graphical method may not work appropriately [107].

In these cases, it is necessary to take into account the strength of connections in determining the minimum number of sensors. The maximum multiplicity theory can be used to estimate the minimum number of sensors taking into account the weights information in a graph. Things to note are that the rank calculation in (2.32) is based on the reduced linear matrix, which makes the rank calculation less expensive and less numerically challenging.

Example. Next, we use one example to illustrate the above discussed methods. We consider system (2.15) with two discretized nodes in each of the three directions. This gives a system with 8 nodes. For this very simple discretized agro-hydrological system, Figure 3.2 shows one possible weighted directed graph of the system. It can be observed that there is a connected path from any of the nodes to the rest of the nodes. This implies that the entire graph forms one SCS. Since there is no incoming edges

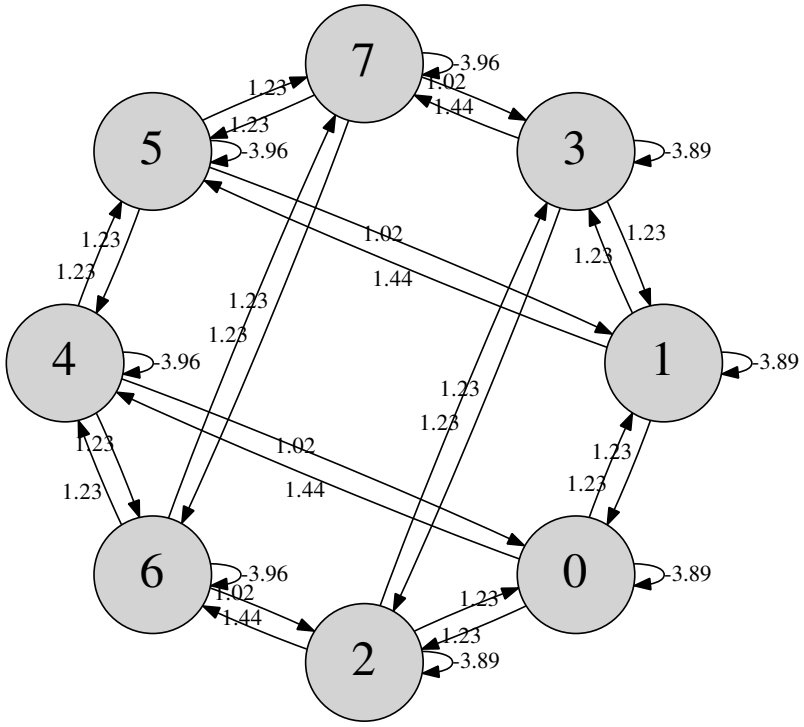


Figure 3.2: Graph representation of the example system with edge weights ($\times 10^{-4}$)

to the SCS, it is also a root SCS. By the structural observability method, one sensor should be sufficient to ensure the observability. However, due to the presence of self-looping and symmetries in the graph, the result given by the structural observability is not reliable [107]. When checking the observability of the system with one measured output using, for example, the PBH method at a steady-state based on the linearized model, it indeed shows that the system is not observable. It is recommended to further use the maximum multiplicity theory to check the result of the structural observability method. Moreover, it is better to linearize the nonlinear system at different sampled points along typical trajectories and use the maximum multiplicity theory for the linearized systems at all these sampled points to find the minimum number of sensors. For the system considered in this example, it was found using the maximum multiplicity theory that the minimum number of sensors is actually two.

3.2.3 Sensor placement based on degree of observability

The solution to the above discussed minimum number of sensors problems may not be unique. In order to find the optimal sensor placement, we propose to use modal

degree of observability as discussed in Chapter 2. This is performed based on the reduced linear matrix ($\hat{A}$) with order r.

A node with a higher degree of observability has the ability to estimate the difficult to reach states. The selection of optimal sensor nodes that maximizes the degree of observability consists of the following steps:

1. Find the minimum number of sensors N_D from the graphical method and the maximum multiplicity theory (2.32).

2. For each of the state/node i, $i = 1, \ldots, r$, in the reduced model, find the corresponding measure $\mathscr{O}_i$ from (2.33)

3. Order the r measures ($\mathscr{O}_i$, $i = 1, \ldots, r$) according to their values. The nodes corresponding to the N_d largest $\mathscr{O}_i$ values are the optimal locations to place the sensors.

Note that in the above method, we assume that we have direct measurement of the state variables. This is indeed the case for the agro-hydrological systems considered in this chapter. One advantage of this method is that it is not needed to check all the possible combinations of sensor sets and we only have to find the degree of observability measures for all the states and then order them to find the optimal sensor location. This is computationally very efficient for large-scale systems with many measurements.

3.2.4 State estimation based on reduced model

Once the optimal sensor placement is found, state estimation can be performed. In the following, we describe how we can get an estimate of the original system's state based on the reduced model.

Let us consider the reduced model (3.2) with additive process and measurement noise as follows:

$$\begin{aligned} \dot{\xi}(t) = &\quad f_r(\xi(t), u(t)) + w_r(t) \\ y(t) = &\quad C\mathscr{U}\xi(t) + v(t) \end{aligned} \tag{3.3}$$

where $w_r(t)$ denotes the process noise of the reduced model and $v(t)$ is the measurement noise. Note that if the original nonlinear system also has additive process noise

Table 3.1: Soil properties of four different types of soils

Soil type	K_s (cm/h)	θ_s (cm/cm)	θ_r (cm/cm)	α	n
Silt loam	0.45	0.45	0.067	0.020	1.41
Sandy clay loam	1.31	0.39	0.1	0.059	1.48
Loam	1.04	0.43	0.078	0.036	1.56
Sandy loam	5	0.41	0.067	0.08	2

and is $w(t)$, then

$$w_r(t) = \mathscr{U}^T w(t). \tag{3.4}$$

For the reduced model, we can use any state estimation techniques to estimate the state ξ. Popular techniques include the extended Kalman filter, moving horizon estimation. Once we have the estimate of ξ, $\hat{\xi}$, an estimate of the original system's state $\hat{x}$ can be obtained as follows:

$$\hat{x}(t) = \mathscr{U}\hat{\xi}(t). \tag{3.5}$$

3.3 Simulation results

In this section, the proposed procedure and algorithms are implemented to perform model reduction and to determine the minimum number of sensors and the optimal sensor placement for agro-hydrological systems under different scenarios. In the simulations, four different types of soil (silt loam, sandy clay loam, loam, sandy loam) are considered. The properties of these types of soil are listed in Table 3.1. Three scenarios are constructed based on the arrangement of the four soil types and the total size of the field.

3.3.1 Scenario 1: a small field with simple soil arrangement

In the first scenario, we consider a small field whose length, width and depth are 1 m, 1 m and 0.3 m, respectively. Four different homogeneous soil layers are considered. The entire system is discretized into 160 nodes with 4 nodes in both the length (X) and width (Y) directions and 10 nodes in the depth (Z) direction. These discretized nodes correspond to the pressure heads of the soil at different locations which are the

states of the system. The arrangement of different types of soil is shown in Figure 3.3. In this system, the bottom 80 nodes consist of silt loam and sandy clay loam and the top 80 nodes consist of sandy loam and loam. In the following, we illustrate how the proposed procedure may be used to find a reduced-order model for the system and to determine the optimal sensor placement for state estimation.

Model reduction. We first apply the proposed model reduction algorithm to the system. Figure 3.4b shows the reduced model with 40 nodes. From Figure 3.4a, it can be seen that the nodes within the same type of soil in the horizontal X direction are clustered while the nodes in the horizontal Y direction and the vertical Z direction are not clustered. The resulting reduced model overall keeps the network connectivity of the original system and can be interpreted as follows. First, due to the presence of the gravity, there is a strong tendency of water flow in the Z direction, which is much more significant than the flow tendency in either X or Y direction. This also implies that we may see more dynamics in the Z direction. This makes the similarity of the nodes in the Z direction relatively smaller compared with nodes in the other two directions. Therefore, the nodes in the Z direction are not clustered together. Second, from the soil properties shown in Table 3.1, it can be seen that the hydraulic conductivity of sandy clay loam is great than that of silt loam in the bottom 80 nodes. This implies that the tendency of flow is from the sandy clay loam to silt loam in the Y direction. This further implies that we may see more dynamics in the Y direction. That is, the similarity between the sandy clay loam nodes and the silt loam nodes is smaller than the similarity between the nodes in the X direction (i.e., nodes within the same type of soil). This explains why the nodes in the Y directions are not clustered together. Furthermore, given that the soil type does not change in the X direction, there is a small tendency of flow in the X direction within the same type of soil. This implies that we may see much less dynamics in the X direction (the similarity is high). This explains why the nodes in the X direction are clustered into one node. Similarly, the clustering for the top nodes can be explained.

If we further reduce the number of nodes of the system to 20, the resulting model is shown in Figure 3.4b. If we compare the two reduced models shown in Figures 3.4a and 3.4b, it can be seen that the horizontal nodes within each soil are further clustered

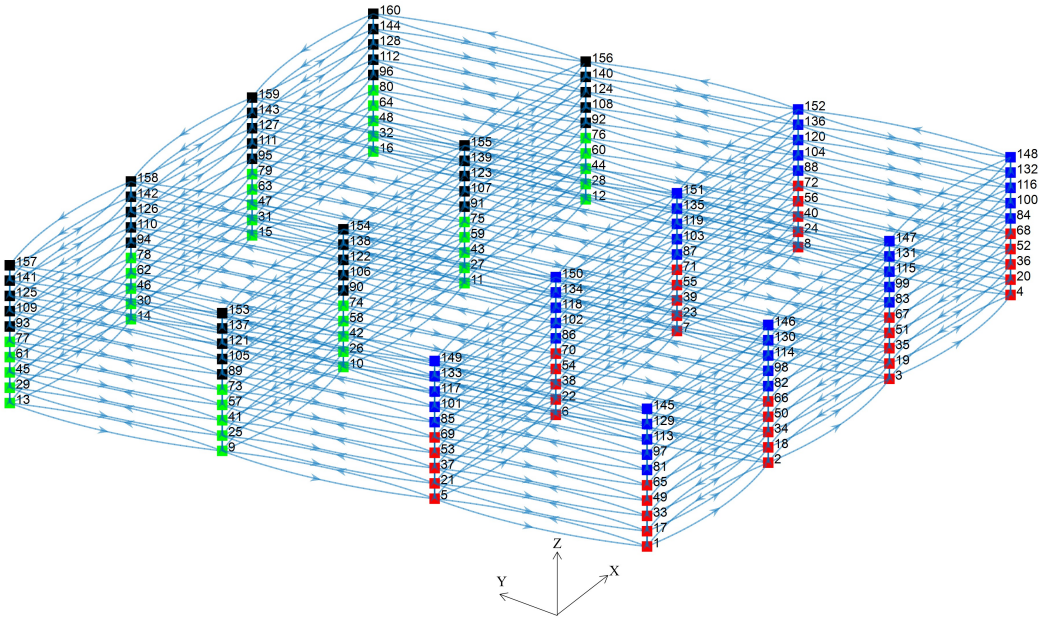


Figure 3.3: Graph representation of the system considered in Scenario 1. Four different types of soil are considered: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

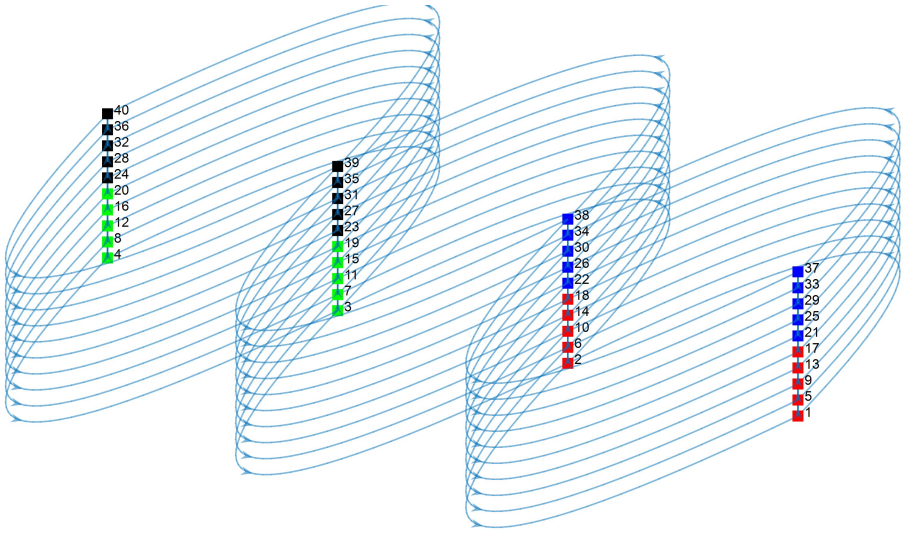
together while the nodes of the vertical direction are not clustered. This makes sense since the tendency of flow in the Z direction is much more significant due to the presence of gravity.

To evaluate the accuracy of a reduced model, we simulate the original system and the reduced model under the same conditions and evaluate the mismatch between sampled points of the trajectories. let us consider the following performance indicator that measures the degree of mismatch between a trajectory of the original system and the corresponding trajectory of a reduced model:

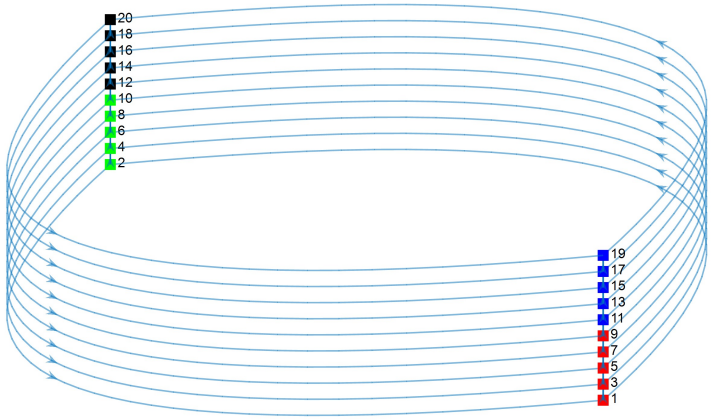
$$E = \sum_{k=0}^{T_f} \sum_{i=1}^{N} \left(\frac{(x_i(t_k) - \hat{x}_i(t_k))^T (x_i(t_k) - \hat{x}_i(t_k))}{T_f} \right) \tag{3.6}$$

where $T_f + 1$ is the total number of samples considered, N denotes the total number of nodes in the original discretized system, $x_i(t_k)$ represents the i^{th} state of the original system at sampling time t_k, $\hat{x}_i$ is the i^{th} element of the corresponding approximated state $\hat{x}$ from the reduced model.

A set of simulations are carried out to evaluate the performance of the reduced order models in terms of E value. In the simulations, the initial conditions of the



(a) Reduced model with 40 states



(b) Reduced model with 20 states

Figure 3.4: Reduced models of the system in Scenario 1 based on the proposed approach. (a) reduced model with 40 states; (b) reduced model with 20 states. In the plots, the four different types of soil are indicated as follows: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

original system states are selected to be -0.24 m and the reduced model initial conditions are calculated using the mapping matrix ($\xi(t_0) = \mathscr{U}^T x(t_0)$). The total simulation time (T_f) is 30 days (720 hours) and the total number of nodes (N) is 160. A random Gaussian signal (RGS) is used as input excitation in the range of $[5 \times 10^{-8} \quad 4 \times 10^{-7}]$ m $\cdot$ hr^{-1} in the simulations.

Figure 3.5 shows the values of E for reduced models with different orders (from 11 to 160). From the figure, it can be seen that the degree of mismatch decreases when

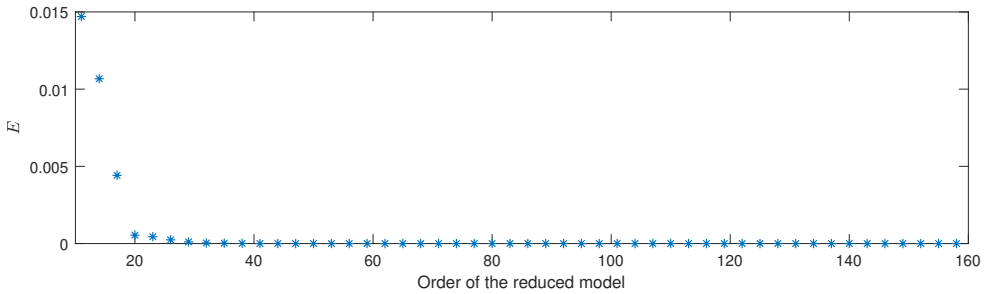


Figure 3.5: Values of E for reduced model with different orders for Scenario 1

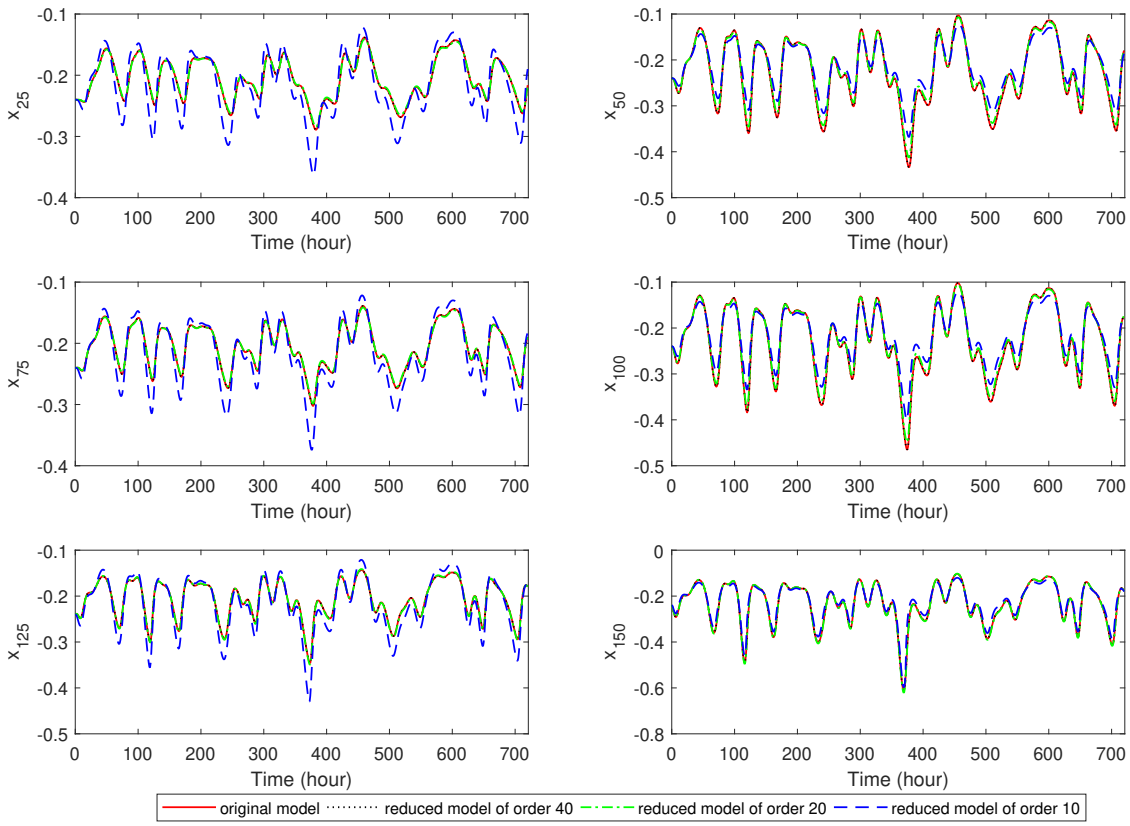


Figure 3.6: Trajectories of some states of the original system (red solid line), the reduced model of order 40 (black dotted line), the reduced model of order 20 (green dash-dot line) and the reduced model of order 10 (blue dashed line) for Scenario 1

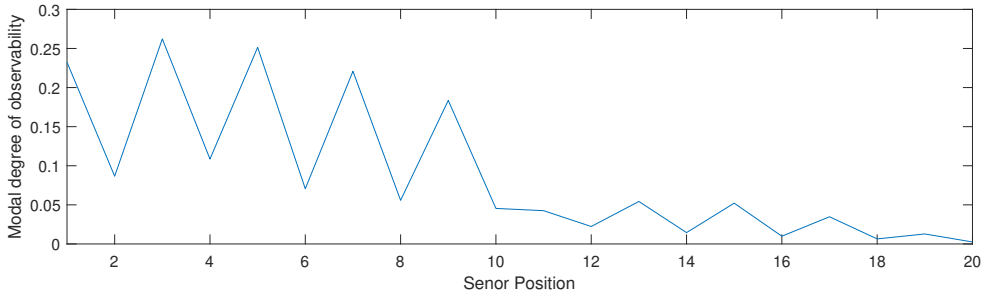
the order of the system increases and the error remains at a low level when the order of the reduced model is higher than 20. Figure 3.6 shows some of the state trajectories of the original system and the reduced models of order 10, 20 and 40, respectively. From this figure, we can see that the reduced models of order 20 and 40 give very close trajectories to the original system while the trajectories of the reduced model of

order 10 are relatively less close to the original trajectories. This further verifies the accuracy of the reduced models with order greater than 20. In the following analysis, we will use the reduced model of order 20.

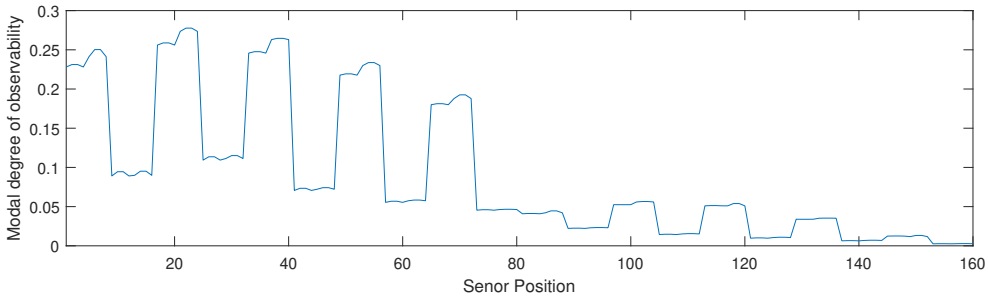
Minimum number of sensors selection. Once an accurate enough reduced model is developed for the original model, we can apply the graphical method and the maximum multiplicity theory to determine the minimum number of sensors that gives observability of the entire state vector. This can be done based on the reduced model. For the system considered in Scenario 1, when the graphical method is used, it can be found that the minimum number of sensors that ensures system observability is one. When the maximum multiplicity theory is used, the resulting minimum number of sensors is also one.

Senor placement. Now, we have determined the minimum number of sensors; however, it does not provide any information on where to optimally place the sensor. Following the proposed procedure, we use the degree of observability to determine where to place the sensor to minimize the total effort required to estimate the states of the system. This analysis can also be done based on the reduced model. Figure 3.7a shows the modal degree of observability obtained based on the reduced model of order 20 when the sensor is placed at different nodes. From Figure 3.7a, it can be seen that the degree of observability is relatively higher if the sensor is placed at node 1 to node 10. These nodes correspond to the soil of loam and silt loam at the bottom layer. The degree of observability is the highest when the sensor is placed at node 1, 3, 5, 7, or 9. These nodes correspond to the silt loam at the bottom layer. This result implies that it is favorable to place the sensor at the downstream of the potential water flows.

In order to verify the results obtained based on the reduced model, we also apply the modal degree of observability method to the original system considered in this scenario. Figure 3.7b shows the degree of observability when the sensor is placed at different nodes. From Figures 3.7a and 3.7b, it can be seen that the reduced model captures the model degree of observability trend of the original system. This indeed is important and makes the reduced model useful in sensor placement.



(a) Reduced model of order 20.



(b) Original system.

Figure 3.7: Modal degree of observability of the original system and the reduced model of order 20 for Scenario 1 when the sensor is placed at different nodes

Based on the above results, we can place the sensor at the location corresponding to node 3 of the reduced model which gives the highest degree of observability. This corresponds to nodes $\{17 \dots 24\}$ of the original system.

3.3.2 Scenario 2: the small field with a different soil arrangement

In the second scenario, we consider a small field with the same size as the one considered in Scenario 1 but with a different soil arrangement. The arrangement of the different types of soil is shown in Figure 3.8. Similar to the system considered in Scenario 1, in the system considered in Scenario 2, the bottom 80 nodes consist of silt loam and sandy clay loam and the top 80 nodes consist of sandy loam and loam. The difference is in the relative arrangement of these types of soil. The different soil arrangement makes the system considered in Scenario 2 more complicated.

Model reduction. In this scenario, the tendency of flow in the Y direction is from sandy clay loam to silt loam in the negative Y direction for the bottom 80 nodes and the tendency of flow for the top 80 nodes is from sandy loam and loam in the negative

X direction. Figures 3.9a, 3.9b and 3.9c show the corresponding reduced models with 60, 40, and 20 nodes, respectively. From Figure 3.9a, it can be seen that the nodes in the top layer within the same soil type in the Y direction are clustered together first. This is due to the fact that the tendency of flow within these nodes is relatively minor. Then from Figure 3.9b, it can be observed that when the number of nodes is reduced, the nodes of sandy clay loam and sandy loam are clustered because the tendency of flow is generated from these nodes which implies that these nodes are in the upstream and contain less information than the nodes of loam and silt loam that are in the downstream. Finally, Figure 3.9c shows that if the number of nodes further reduces, the top 80 nodes are clustered in the Y direction and the bottom 80 nodes are clustered in the X direction and in the Z direction no nodes are clustered because of again the presence of gravity which makes the tendency of flow of Z direction is much more significant than that of X and Y directions.

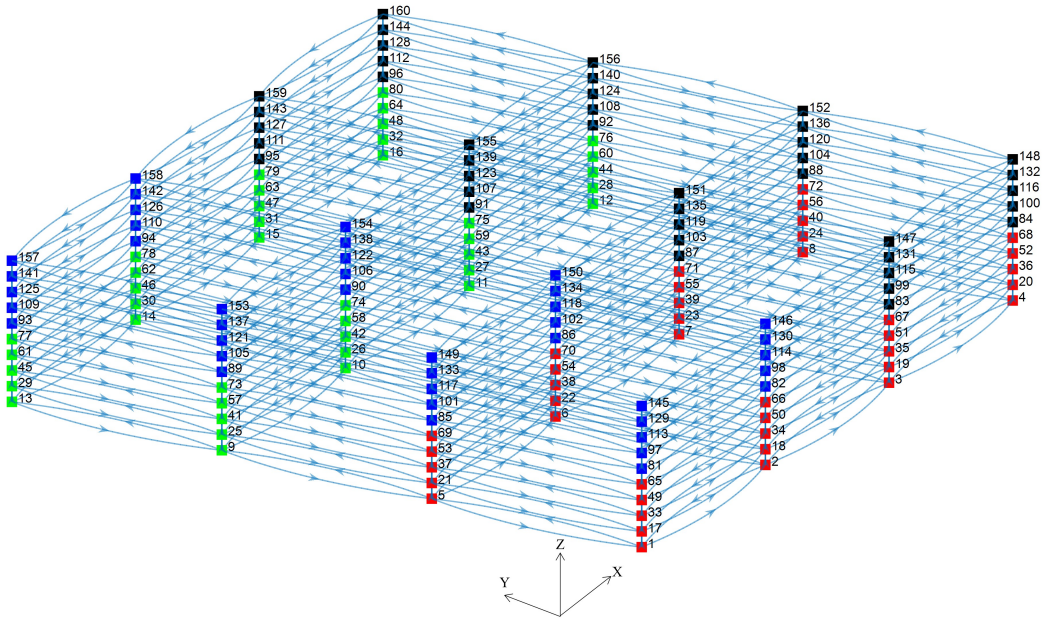


Figure 3.8: Graph representation of the system considered in Scenario 2. A different arrangement of the four different types of soil (silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)) is considered

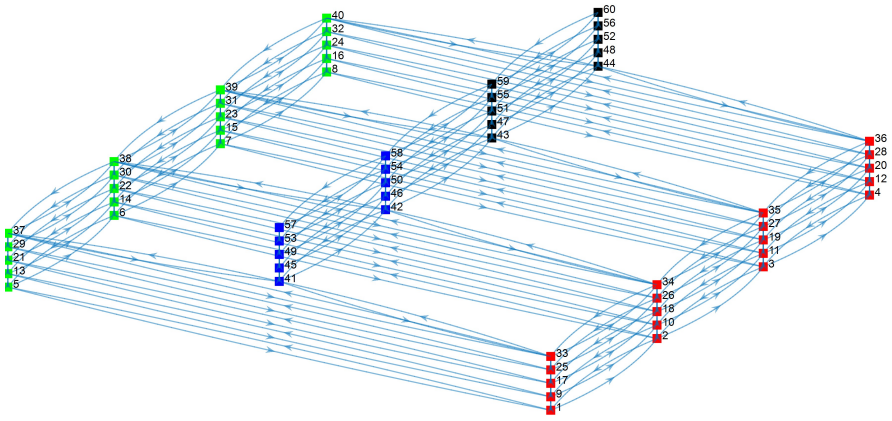
To evaluate the accuracy of the reduced models, the indicator in (3.6) is again used. Figure 3.10 shows the values of E for reduced models with different orders. From the figure, we can observe that the value of E decreases with the increase of

the order of the reduced model and when the order is about 40 and higher, the value of E goes to a relatively low level. If we compare the results shown in Figure 3.10 for Scenario 2 and Figure 3.5 for Scenario 1, it can be seen that a reduced model of order 20 is sufficient accurate for the system in Scenario 1 while more nodes (about 40) are needed to accurately approximate the system in Scenario 2. This is because for the system in Scenario 2, the soil arrangement is more complex and we have significant flow tendencies in all the three directions whereas in the system of Scenario 1, we only have significant flow tendencies in X and Z directions.

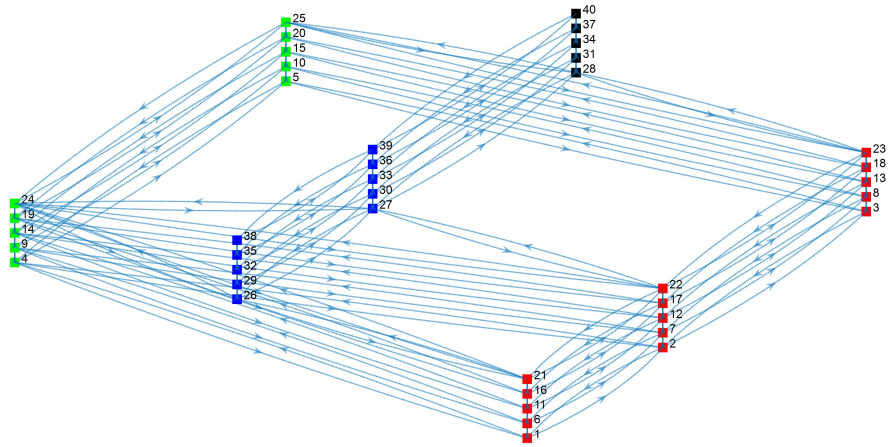
Figure 3.11 shows the comparison of a few state trajectories of the original system and the corresponding state trajectories of the reduced model of order 40, 20, and 10, respectively. The figure further verifies that the reduced model of order 40 gives very accurate approximation of the original system while the reduced models of order 20 and 10 give less accurate approximation of the original system. Let us consider the reduced model of order 40 for the following analysis of Scenario 2.

Minimum number of sensors and sensor placement. For the system considered in Scenario 2, it can be found that the minimum number of sensor that ensures the observability of all the states is also one. To determine the optimal sensor placement, similar to Scenario 1, we use the degree of observability obtained based on the reduced model. Figure 3.12a shows the modal degree of observability of the reduced model of order 40 when the sensor is placed at different nodes. We also present the modal degree of observability of the original system in Figure 3.12b. From the two figures, it is observed that the reduced model is able to capture the model degree of observability trend of the original system as in Scenario 1. The results again can be explained using the tendency of flow among the nodes. Based on the results shown in Figure 3.12a, we can place the sensor at the location corresponding to node 8 of the reduced model of order 40 to obtain the maximum degree of observability. Node 8 of the reduced model corresponding to nodes {19, 20, 23, 24} of the original system.

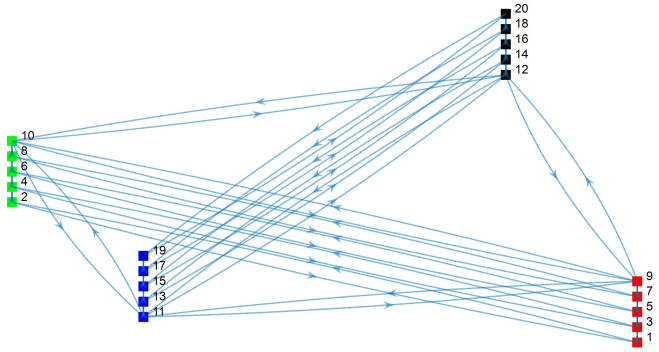
Robustness. More simulations are carried out to study the robustness of the proposed procedure and algorithms. In this set of simulations, uncertainty in model parameters is considered. Specifically, random noise is introduced to the system pa-



(a) Reduced model with 60 states



(b) Reduced model with 40 states



(c) Reduced model with 20 states

Figure 3.9: Reduced models of the system considered in Scenario 2. (a) reduced model with 60 states; (b) reduced model with 40 states; (c) reduced model with 20 states. In the plots, the four different types of soil are indicated as follows: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

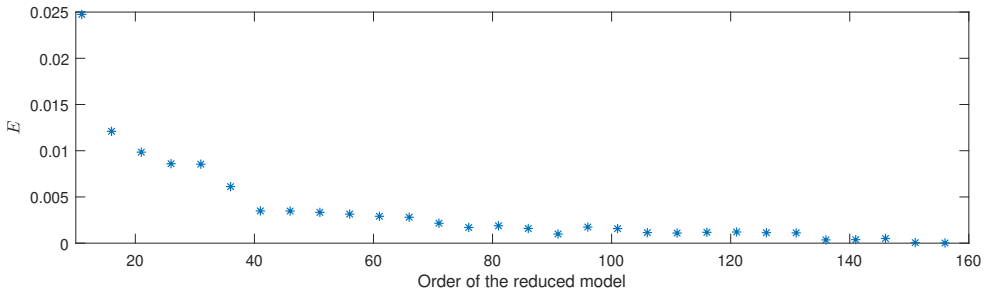


Figure 3.10: Values of E for reduced model with different orders for Scenario 2

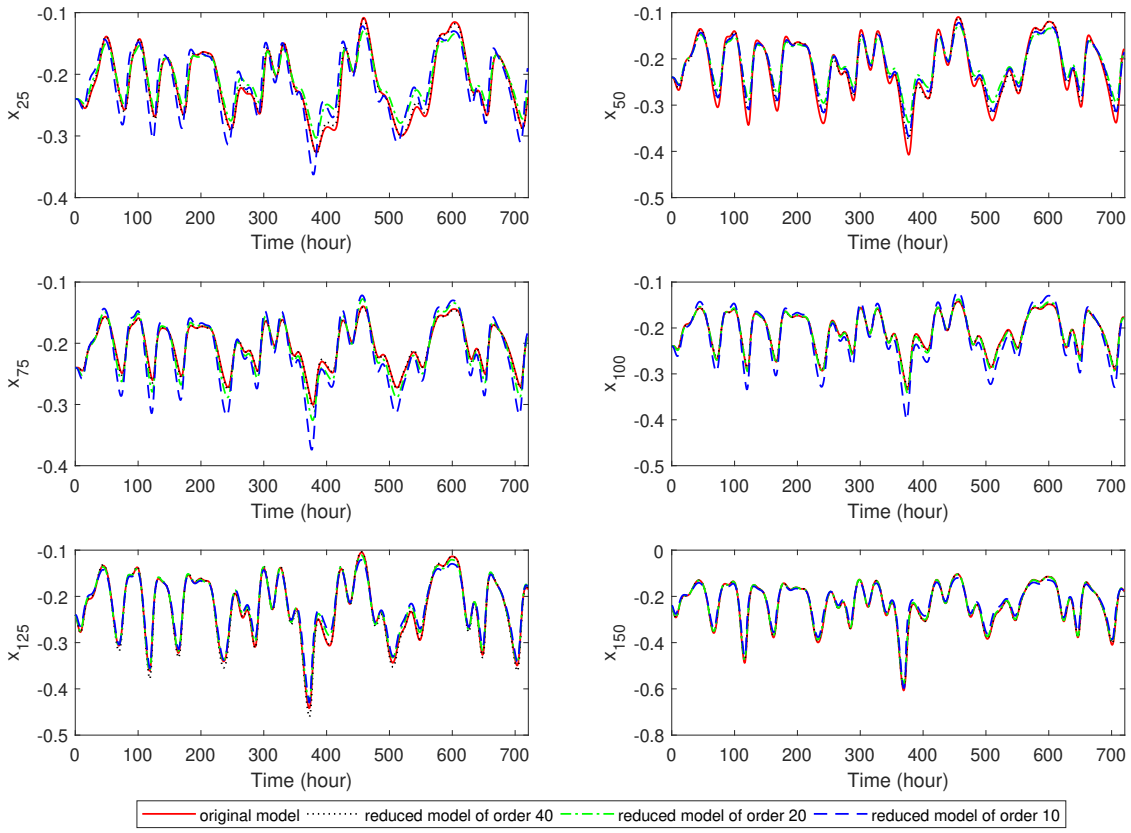
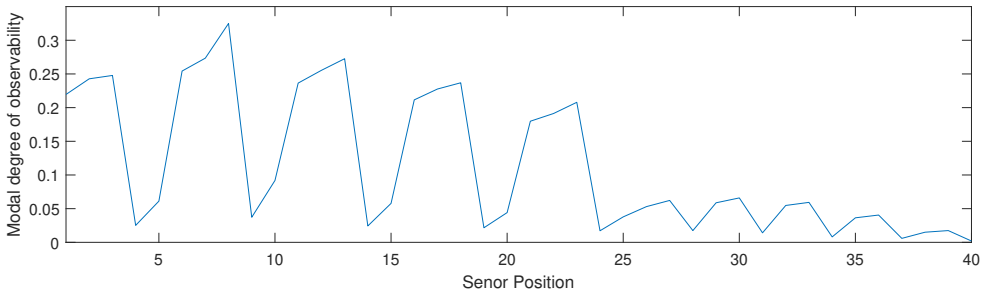


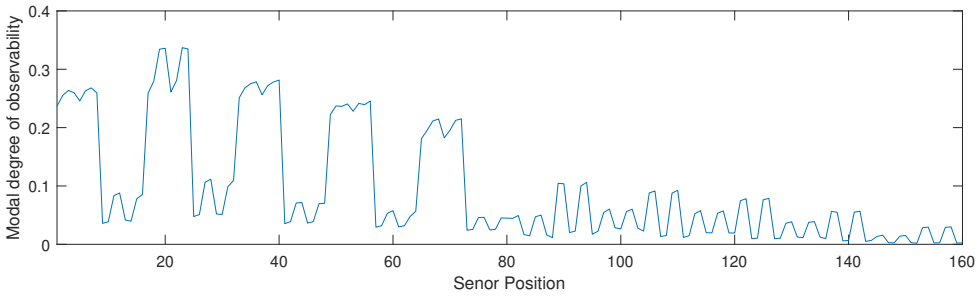
Figure 3.11: Trajectories of a few states of original model (red solid line), the reduced model of order 40 (black dotted line), the reduced model of order 20 (green dash-dot line) and the reduced model of order 10 (blue dashed line) for Scenario 2

rameters. The random noise added to K_s is with zero mean and variance of 3% of the nominal value and for other parameters $(\alpha, n, \theta_s, \theta_r)$ is with zero mean and variance of 0.1% of the respective nominal values.

Figure 3.13 shows the model reduction results of order 60, 40 and 20. It can be seen from Figure 3.13a that there is more randomness in the nodes of silt loam



(a) Reduced system of order 40.



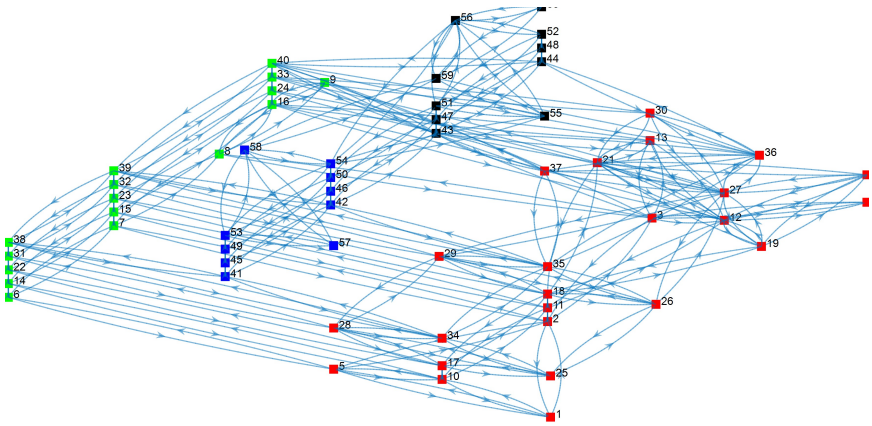
(b) Original system.

Figure 3.12: Modal degree of observability of the original system and the reduced model of order 40 for Scenario 2 when the sensor is placed at different nodes

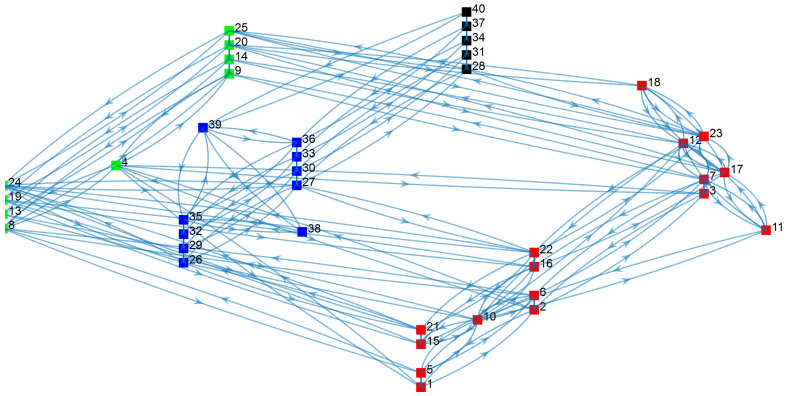
because theses nodes are in downstream so the uncertainty in the parameters affects these nodes most. Then from Figure 3.13b, it can be observed that the randomness is less in the sandy loam and sandy clay loam because these nodes are in the upstream and contain less information of the system so that the uncertainty in the parameters does not affect that much compared to other soil nodes. Finally, Figure 3.13c shows that the reduced system of order 20 with uncertainty is the same as the case without uncertainty because the nodes within the same soil types are affected by the uncertainty but the overall tendency of flow is not affected that much by the uncertainty in parameters.

The accuracy of the reduced-order models has also been verified. Figure 3.14 shows the trajectories of the original system and the reduced models of order 40, 20 and 10. From the figure, it can be observed that the reduced models of order 40 and 20 both give good approximation of the original nonlinear system.

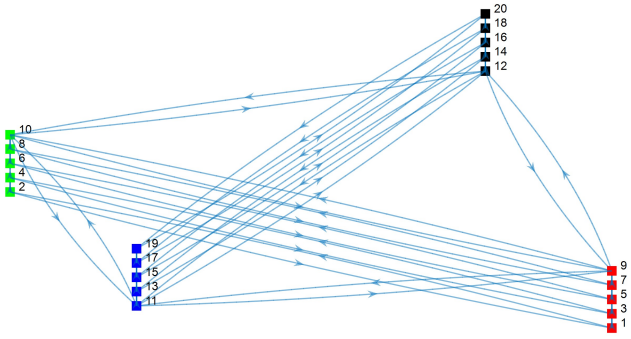
For this system, it can still be found that the minimum number of the sensor is one. Figure 3.15 shows the modal degree of observability of the reduced system of order 40 and the original system. Similar to the case without uncertainty, the



(a) Reduced model with 60 states



(b) Reduced model with 40 states



(c) Reduced model with 20 states

Figure 3.13: Reduced models of the system considered in Scenario 2 with parameter uncertainty. (a) reduced model with 60 states; (b) reduced model with 40 states; (c) reduced model with 20 states. In the plots, the four different types of soil are indicated as follows: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

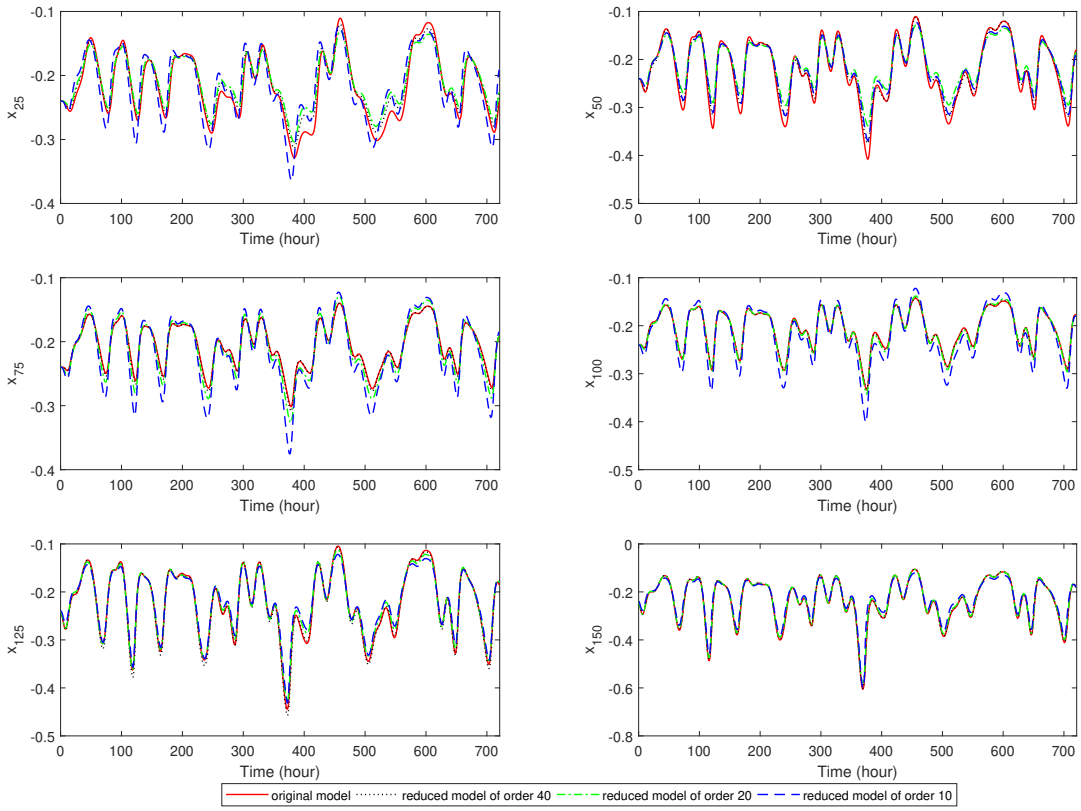
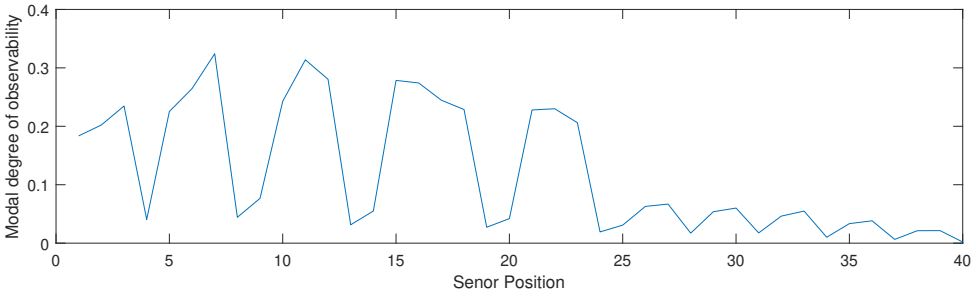


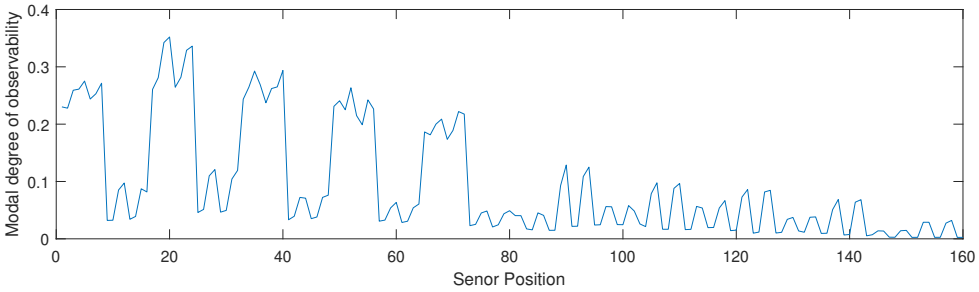
Figure 3.14: Trajectories of a few states of original model (red solid line), the reduced model of order 40 (black dotted line), the reduced model of order 20 (green dash-dot line) and the reduced model of order 10 (blue dashed line) for Scenario 2 with parameter uncertainty

reduced model can capture the modal degree of the trend of the original system. Based on Figure 3.15a, the maximum degree of observability is at node 7 while for the case without uncertainty the maximum degree of observability is at node 8. But the corresponding original nodes of node 7 is the same as the corresponding original nodes of node 8 of the case without uncertainty which is {19, 20, 23, 24}.

Remark 2 *Note that the field considered in Scenario 2 is very small. For such a small field, the soil properties in general have very minor variations. Therefore, the variances considered in the noise added to the parameters are small as well. The results of the simulations with parameter uncertainty illustrates that the proposed procedure and algorithms can still give good reduced models and may be used in sensor placement. A more systematic study is needed to investigate the robustness of the*



(a) Reduced system of order 40.



(b) Original system.

Figure 3.15: Modal degree of observability of the original system and the reduced model of order 40 for Scenario 2 with parameter uncertainty when the sensor is placed at different nodes

proposed procedure and algorithms, which will be considered in our future work.

3.3.3 Scenario 3: a larger field

In the third scenario, we consider a comparatively bigger field than the first two scenarios. The length, width and depth of the field are 10 m, 10 m and 0.6 m, respectively. The system is discretized into 8000 nodes with 20 discretized nodes in each of the three directions. The arrangement of the different types of soil is similar to that in Scenario 1 and is shown in Figure 3.16. Based on the system with 8000 nodes directly, it is very challenging to consider sensor placement.

To find the optimal sensor placement, the system is reduced from 8000 states to 40 states based on the proposed approach. The reduced model of order 40 is shown in Figure 3.17. In the reduced model, all the nodes within the same type of soil in both X and Y directions are clustered and the nodes in Z direction are kept. The results again can be explained based on the tendency of flow.

To verify the accuracy of the reduced model, we have compared the trajectories of the original system and the reduced model. Figure 3.18 shows some representative

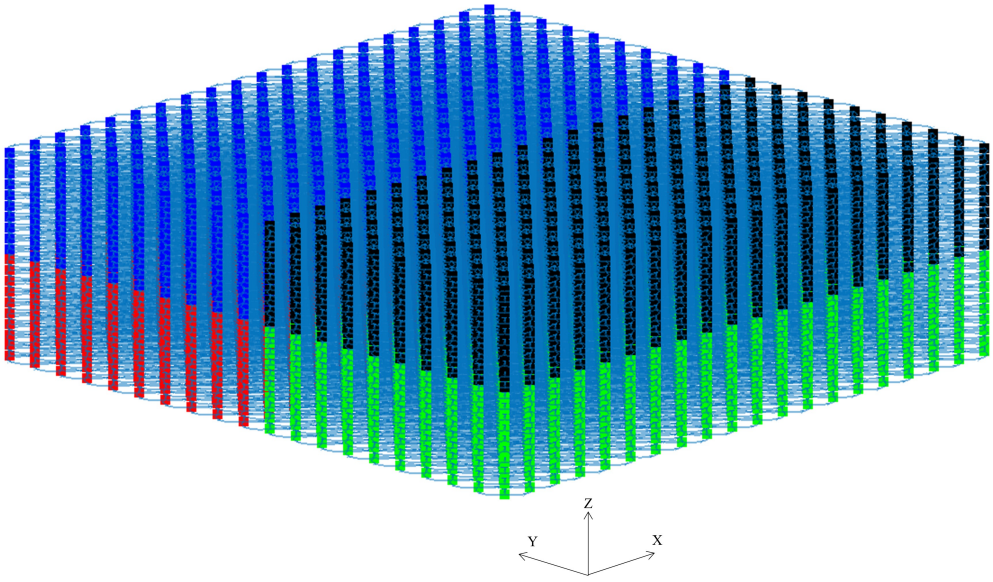


Figure 3.16: System considered in Scenario 3. Four different types of soil are considered: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

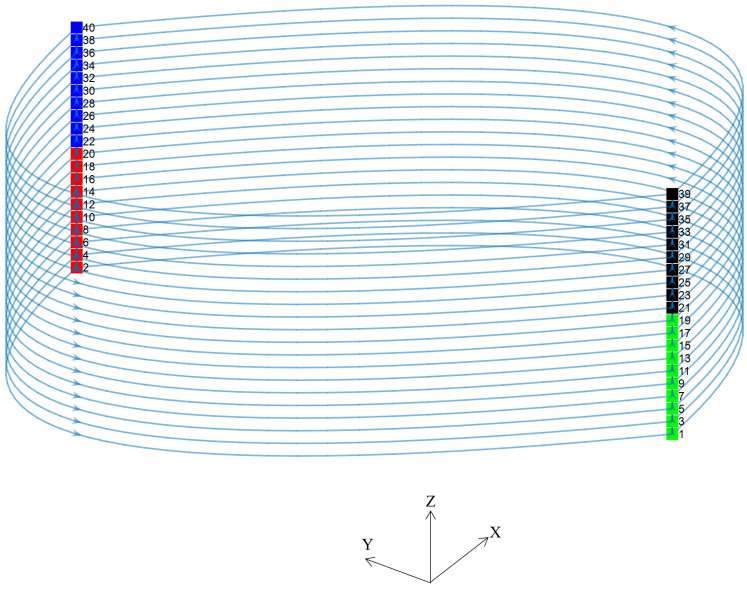


Figure 3.17: Reduced model of order 40 for the system considered in Scenario 3. In the plot, the four different types of soil are indicated as follows: silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)

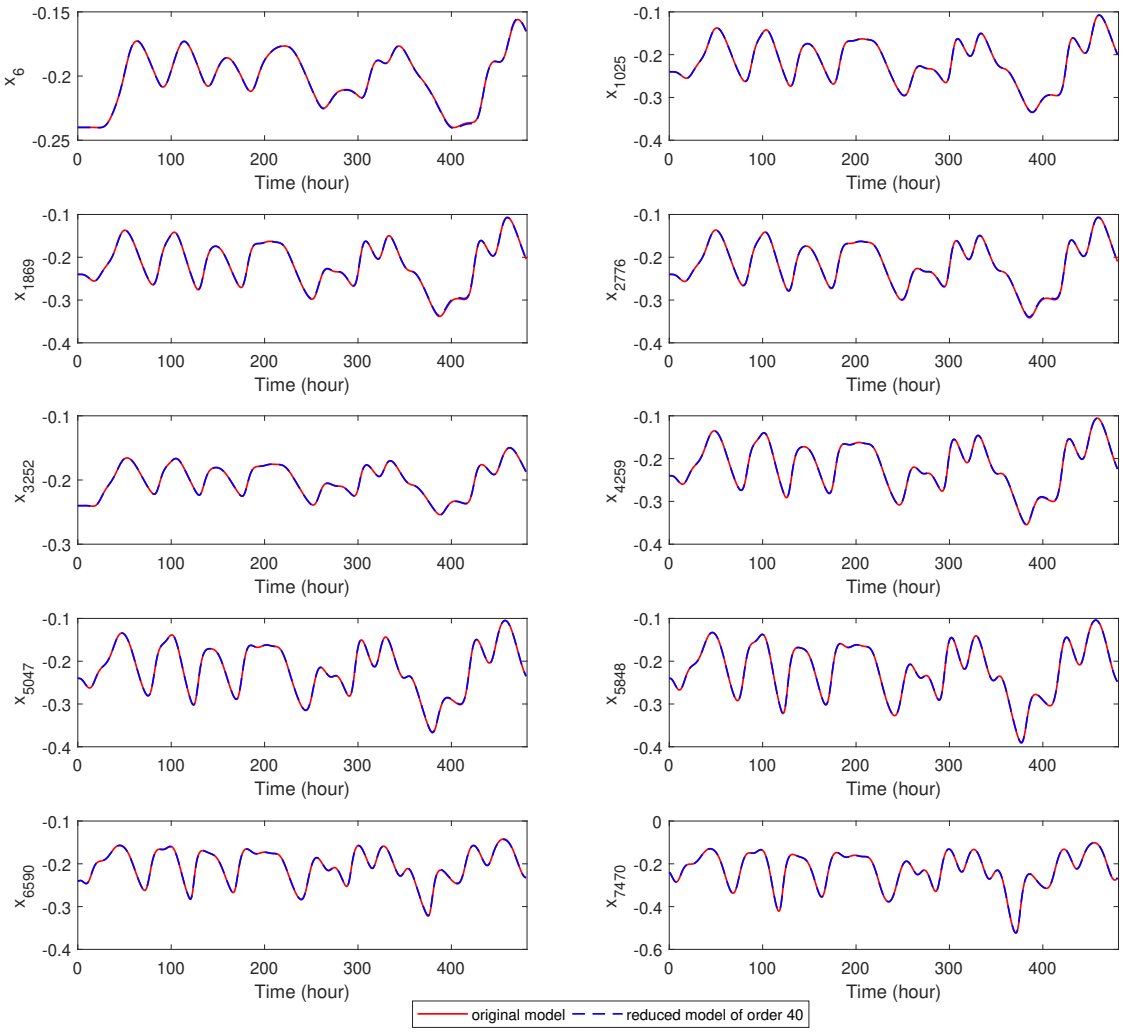


Figure 3.18: Trajectories of some of the states of original model (red solid line) and the corresponding states of the reduced model of order 40 (black dashed line) for Scenario 3

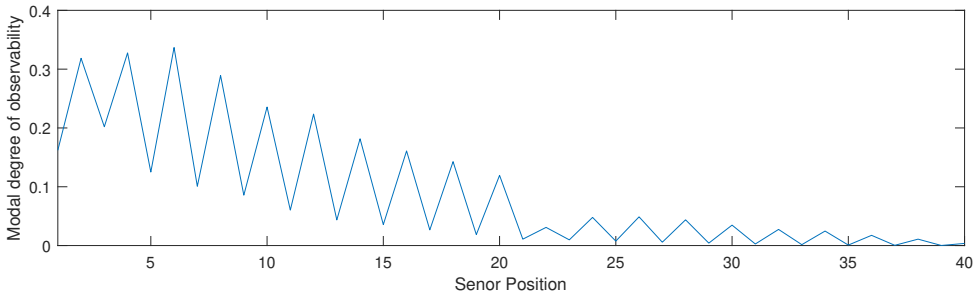


Figure 3.19: Modal degree of observability of the reduced model of order 40 for Scenario 3

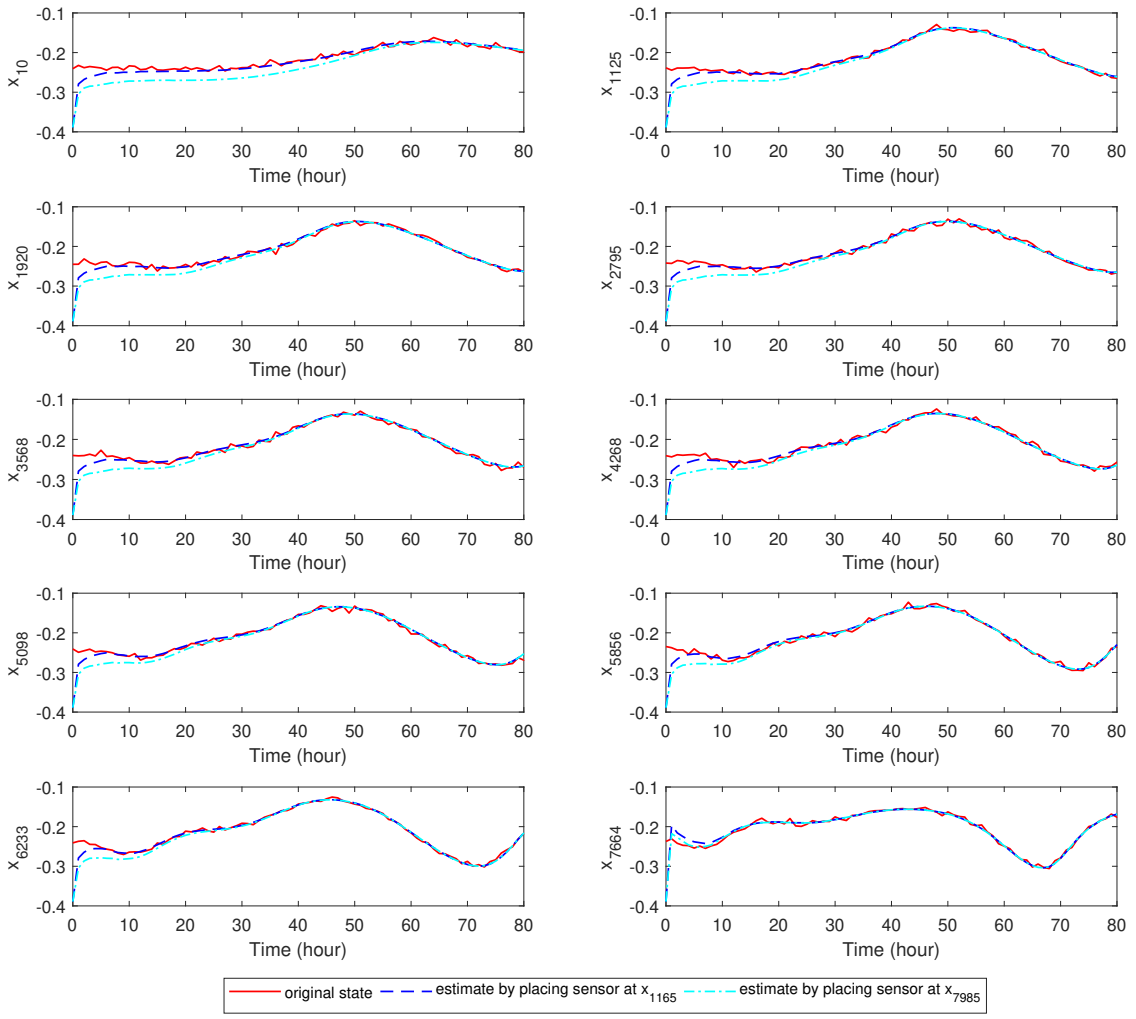


Figure 3.20: Trajectories of the actual states (red solid line), the state estimates with sensor placed at x_{1165} (blue dashed line) and the sensor placed at x_{7985} (cyan dash-dot line)

trajectories of the original system and the reduced model. From the figure, it can be observed that the reduced model trajectories have a very good agreement with the original system trajectories.

Next, let us use the reduced model to determine the minimum number of sensors and to find the optimal sensor placement. Based on the reduced model, if we apply the graphical method and the maximum multiplicity theory, we can find that the minimum number of sensors for the system is still one. The modal degree of observability of the reduced model is shown in Figure 3.19. Similar to Scenario 1 and Scenario 2, the top nodes have less degree of observability than the bottom nodes and the nodes consists of loam and silt loam have more degree of observability than nodes of the

sandy loam and sandy clay loam.

Based on Figure 3.19, the optimal sensor placement is to place the sensor at a place corresponding to node 6 of the reduced model. Let us consider that we have the sensor placed at node 1165 of the original system (i.e., we have measurement x_{1165}), which is clustered to node 6 of the reduced model. Figure 3.20 shows the actual state and the estimates based on the optimal sensor placement at x_{1165} and another case with the sensor placed at x_{7985} (which corresponds to node 40 of the reduced model). Note that in this set of simulations, the estimators are extended Kalman filters and the two filters are optimally tuned following the same procedure. We consider there is no communication delay for the availability of the measurements. The process noise $w(t)$ for all the states are generated by normal distribution with zero mean and the standard deviation of 4.6×10^{-3}. The measurement noise is a Gaussian white noise with zero mean and standard deviation of 4.5×10^{-3}. From the figure, we can see that with the optimally placed sensor, the estimates converge to the actual system states much faster.

3.4 Conclusions

This chapter proposed the schematic procedure of optimal placement for three dimensional agro-hydrological systems. The first step was to reduce the order of the system using a graph-based similarity matrix between the states. In the second step, the minimum number of sensors was calculated using the structural observability method and maximum multiplicity theory. The final step calculates the best sensor position using the modal degree of observability analysis. Three different scenarios were considered. A small field was considered for the first two scenarios with different soil arrangements. In the third scenario, a relatively bigger field was considered. The estimation result showed that the optimally placed sensor lead to improved estimation performance.

Chapter 4

Dynamic model reduction and optimal sensor placement

4.1 Introduction

In this chapter, a systematic procedure for optimal sensor placement for three-dimensional agro-hydrological systems in the presence of non-uniform inputs and heterogeneous soil types is proposed. The proposed sensor placement method follows similar ideas as in Chapter 3. As discussed in Chapter 3, the model reduction is based on weighted directed graphs, and the weights are calculated using the Jacobin matrix at a steady state. However, the input amount is periodic and non-uniform in spatial directions in some popular irrigation techniques. So the weights of the graph obtained at one steady-state point may not represent the whole system dynamics which motivated us to propose a dynamic model reduction method. In the proposed dynamic model reduction method, the operating regions are first selected, and one nonlinear reduced model is constructed at each operating region. The model switch algorithm is proposed at the boundary of each operating region. Next, the minimum number of sensors using structural observability and maximum multiplicity theorem is proposed. As the reduced model changes over time, calculating the minimum number of sensors for each operating region and picking the maximum value from all the operating regions is proposed. The maximum value is the minimum number of sensors for the entire system. Afterward, the senor placement using the modal degree of observability is proposed. The reduced-order model degree of observability is calculated at each operating point and convert it to the original system order using a projection matrix.

The final model degree of observability matrix is obtained using the average value at all the operating regions. The proposed procedure and methods are applied to the agro-hydrological system with four different sprinklers and four different soil types. The simulation results illustrate the applicability and effectiveness of the proposed procedure and methods.

4.2 Proposed sensor placement procedure

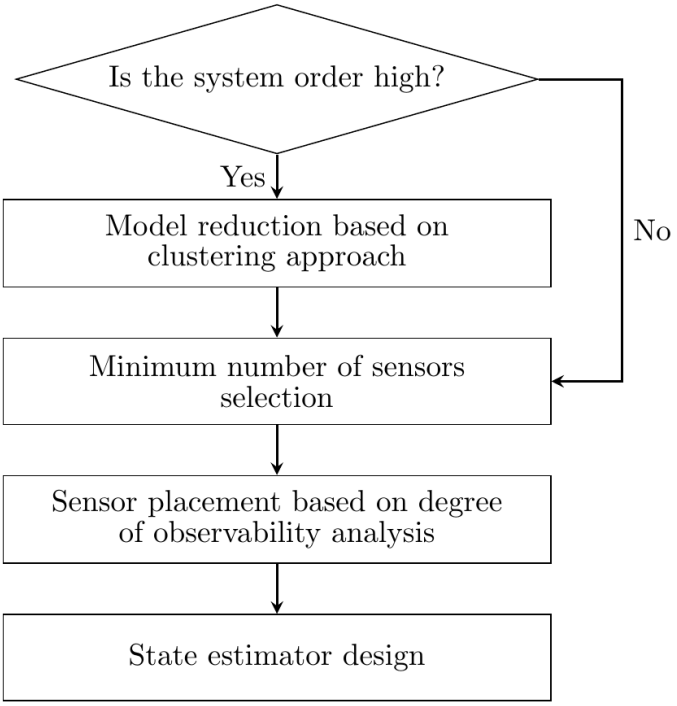


Figure 4.1: Flow diagram of the proposed sensor placement procedure

In this section, we propose the systematic method to obtain the optimal sensor placement. The key steps include model reduction, the minimum number of sensor selection, optimal sensor placement and the state estimation. A flow diagram of the proposed procedure is shown in Figure. 4.1.

4.2.1 Dynamic model reduction

We propose a new dynamic graph-based clustering method for model reduction. The steps to calculate the dynamic model reduction as follows:

Step 1: calculation of operating points

Simulate the original non-linear model and obtain the state trajectory snapshot from initial time (t_0) to final time (t_N)

$$\mathscr{X} = [x(t_0)\ x(t_1)\ \ldots\ x(t_N)]$$

where $\mathscr{X}$: snapshot matrix $\in \mathbb{R}^{n\times(N+1)}$, n: number of states. Based on the state snapshot and input, obtain different operating point sets ($\mathscr{P} \in \mathbb{R}^s$). The $\mathscr{P}$ contains the information of time at which the system should be linearized. The linearization points at the operating points are $[(x_{\mathscr{P}(1)}, u_{\mathscr{P}(1)}), \ldots (x_{\mathscr{P}(s)}, u_{\mathscr{P}(s)})]$. In this work, the operating points are selected at the endpoint of each peak of the periodic input sequence.

Step 2: operating region calculation

The total time of the system has been divided into different operating regions. The operating regions are selected based on the operating points, in other words, each operating points have their operating region. We have proposed that for each operating region one projection matrix will be constructed which will lead to one reduced model for each operating region. Lets consider $\mathscr{R} = \{\mathscr{R}_1, \mathscr{R}_2, \ldots, \mathscr{R}_s\}$ as the collection of operating regions, which has the following properties: i) $\mathscr{R}_i \cap \mathscr{R}_j = \Phi$ and ii) $\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots \cup \mathscr{R}_s = \mathscr{T}$, where $\mathscr{T} = [0, T_f]$, T_f is total operating time.

Step 3: graph construction

Let us represent the system in (2.15) as a directed graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N} = \{1, 2, \ldots, n\}$ denotes the states and $\mathscr{E} \subset \mathscr{N} \times \mathscr{N}$ denotes the directed edges. Let us also use $a_{i,j}$ to denote the weight from state j to state i in the directed graph. In this work, we have considered the time varying weights for the directed graph. In other words, the weights of the directed graph may change for each operating regions. The weights of the edges are calculated based on the Jacobian of the nonlinear system at different linearization points. For one operating region ($\mathscr{R}_k$), the jacobian matrix is calculated as follows:

$$A^{(k)} = \left.\frac{\partial f}{\partial x}\right|_{(x_{\mathscr{P}}(k), u_{\mathscr{P}}(k))}$$

where $x_{\mathscr{P}}(k)$ and $u_{\mathscr{P}}(k)$) are the selected linearized points.

Step 4: reduced order system calculation

In Chapter 3, the difference of the edge weights can be used to measure the similarity of the two states which can be used to cluster the states. The proposed method calculates the edge weight at one steady-state point. But when there are both different types of soil and different inputs, it may show model approximation error. This leads us to propose an approach towards considering the dynamic weights of the system at different time instances. The idea is to evaluate the edge weights at different linearization points. For example, the similarity of state i and state j is measured at each operating region ($\mathscr{R}_k$) by calculating:

$$\mathscr{D}_{i,j}^{(k)} = |a_{i,j}^{(k)} - a_{j,i}^{(k)}| \tag{4.1}$$

where $a_{i,j}^{(k)}$ and $a_{j,i}^{(k)}$ are the corresponding elements in the matrix of $A^{(k)}$. When $\mathscr{D}_{i,j}^{(k)}$ is small, it implies that the two states (i and j) are similar at $\mathscr{R}_k$ and may be clustered together.

The dynamic reduced order system is constructed based on the Petrov-Galerkin projection framework as discused in Chapter 2. The dynamic reduced model at each operating region ($\mathscr{R}_k$) is expressed as:

$$\dot{\xi}^{(k)}(t) = f_r^{(k)}(\xi^{(k)}(t), u(t)) \tag{4.2}$$

where $f_r^{(k)}(\xi^{(k)}(t), u(t)) = \mathscr{U}^{(k),T} f(\mathscr{U}^{(k),T}\xi^{(k)}(t), u(t))$ and $\xi^{(k)}(t) = \mathscr{U}^{(k),T} x(t)$. Note that the actual state x can be approximated based on mapping $\tilde{x}(t) = \mathscr{U}^{(k),T}\xi^{(k)}(t)$. In the reduced model (4.2), the dimension of the reduced state vector ξ is a tuning parameter and determines the size of the reduced model. Algorithm 2 summarizes the proposed dynamic clustering approach.

The discrete model of (4.2) is expressed as follows:

$$\xi^{(k)}(i+1) = f_{rd}^{(k)}(\xi^{(k)}(i), u(i)) \tag{4.3}$$

Step 5: switching of reduced model

When the system moves from one operating region to another operating region, the corresponding reduced models will also be switched. To handle the impact of changing

reduced-order models from one operating region to other operating regions, we consider information exchange between reduced models at the boundary of the operating regions. The switching algorithm between reduced model are as follows:

1. Let us consider two operating regions $\mathscr{R}_k$ and $\mathscr{R}_{k+1}$ at boundary of time i. Compute the states of the reduced system at i using the projection matrix $\mathscr{U}^{(k)}$ which is generated for the operating region $\mathscr{R}_k$ and, the past reduced state and input information.

$$\xi^{(k)}(i) = \mathscr{U}^{(k)T} f_d(\mathscr{U}^{(k)}\xi^{(k)}(i-1), u(i-1))$$

2. Compute the approximated states of original system ($\tilde{x}(i)$) at i using: $\tilde{x}(i) = \mathscr{U}^{(k)}\xi^{(k)}(i)$

3. Compute the initial value for next operating region reduced model as $\xi^{(k+1)}(i) = \mathscr{U}^{(k+1),T}\tilde{x}(i)$

4.2.2 Minimum number of sensors

As discussed in Chapters 2 and 3, the minimum number of sensors can be obtained using the structural observability and maximum multiplicity theory. The reduced model (4.2) can be used indirectly to determine the minimum number of sensors.

The reduced model structure may change for every operating region because the cluster sets are different. So the structural observability analysis is performed at every operating point, and the minimum sensors values for each operating region are obtained. The minimum number of sensors for the entire system is calculated, picking the maximum value among the values collected for each operating region. The main idea is that even if we may use extra sensors for some operating regions, the entire system will be observable using the calculated number of sensors.

Similar to the idea of performing structural observability at each operating point, the maximum multiplicity theory is also calculated using at each operating point, and the maximum value is considered as follows:

$$N_D = \max_k\{\max_i\{r - \text{rank}(\lambda_i I_r - A_r^{(k)}\}\} \tag{4.4}$$

Algorithm 2 Dynamic model reduction based on clustering

Require: Creation of projection matrix $\mathscr{U}(t)$

Input: Operating points $\mathscr{P} \in \mathbb{R}^s$, linearization points $[(x_{\mathscr{P}(1)}, u_{\mathscr{P}(1)}), \ldots (x_{\mathscr{P}(s)}, u_{\mathscr{P}(s)})]$, linearized matrices $A^{(k)}$ at linearization points, reduced model order r, model order n

Output: Reduce order nonlinear system at different operating points

1: **for** $k = 1 \ldots s+1$ **do**
2: $A^{(k)} = \frac{\partial f}{\partial x}\big|_{(x_{\mathscr{P}(k)}, u_{\mathscr{P}(k)})}$
3: **Initialization** $i \leftarrow n, \hat{A}^{(k)} \leftarrow A^{(k)}$,
4: **while** $i > r$ **do**
5: Compute $\mathscr{D}^{(k)}$ based on (4.1) and find the smallest element δ
6: Find the states (p, q) corresponding to value δ and merge the states into a single cluster
7: Compute the projection matrix at the current iteration $\mathscr{U}^{(k,i)}$ based on *Definition 2*
8: Update $\hat{A}^{(k)}$ using the projection matrix and current $\hat{A}^{(k)}$: $\hat{A}^{(k)} \leftarrow \mathscr{U}^{(k,i),T}\hat{A}^{(k)}\mathscr{U}^{(k,i)}$
9: Save $\mathscr{U}^{(k,i)}$ matrix at each iteration
10: $i = i - 1$
11: **end while**
12: Compute final projection matrix $\mathscr{U}^{(k)} = \Pi_{i=r+1}^{n}\mathscr{U}^{(k,i)}$,
13: $k = k + 1$
14: Compute reduced nonlinear model $f_r^{(k)}(\xi(t), u(t)) = \mathscr{U}^{(k),T} f(\mathscr{U}^{(k)}\xi(t), u(t))$ and $\xi(t) = \mathscr{U}^{(k),T}x(t)$
15: Save $\mathscr{U}^{(k)}, f_r^{(k)}(\xi(t), u(t))$ at each iteration
16: **end for**

where $A_r^{(k)}$ is the Jacobian of the reduced nonlinear system at operating point (k), λ_i, $i = 1, \ldots, r$, are the eigenvalues of $A_r^{(k)}$ and r is the order of the linearized system.

4.2.3 Optimal sensor placement

The modal degree of observability as discussed in Chapter 2 is used to determine the optimal location of sensors. The modal degree of observability at operating region $\mathscr{R}_k$ is calculated as follows:

$$\mathscr{O}_{r_i}^{(k)} = \sum_{j=1}^{r}(1 - \lambda_j^2(A_{dr}^{(k)}))v_{ij}^2 \tag{4.5}$$

where $A_{dr}^{(k)}$ is the discretized reduced model Jacobian matrix at operating region ($\mathscr{R}_k$), r is the order of reduced model, λ_j ($j = 1, \ldots, r$) are the eigenvalues of matrix $A_{dr}^{(k)}$ and v_{ij} are the right eigen vectors.

The optimal sensor node sets can be obtained by maximizing the degree of observability and the method consists of following steps:

1. Using the graphical method and the maximum multiplicity theory find the minimum number of sets N_D

2. For each operating point, k, $k = 1, \ldots, s$, calculate the modal degree of observability matrix $\mathscr{O}_r^{(k)} = [\mathscr{O}_{r1}^{(k)}, \ldots, \mathscr{O}_{rr}^{(k)}] \in \mathbb{R}^{1\times r}$

3. Find the modal degree of observability of the original system at each operating point as $\mathscr{O}^{(k)} = \mathscr{O}_r^{(k)}\mathscr{U}^{(T)} \in \mathbb{R}^{1\times n}$

4. Calculate the final modal degree of observability ($\mathscr{O}$) of the original system as the average value of the modal degree of observability at each operating points

5. The nodes correspond to the first N_D biggest elements of $\mathscr{O}$ measures are the optimal sensor nodes.

The proposed algorithm of the modal degree of observability is a little different from Chapter 3 because, in this case, the modal degree of observability of the reduced-order system is converted back to the original system. The reason to do that is the nodes of the reduced-order system don't correspond to the same nodes at different operating points in the original system. After all, at each operating point, the cluster matrix may change.

4.2.4 State estimation based on reduced model

After finding the optimal sensor placement, we can perform the state estimation algorithm. Let us consider the reduced model (4.2) at each operating region (k) with measurement and process noise as follows:

$$\begin{aligned} \dot{\xi}(t) = &\quad f_r^{(k)}(\xi(t), u(t)) + w_r^{(k)}(t) \\ y(t) = &\quad C\mathscr{U}^{(k)}\xi(t) + v(t) \end{aligned} \tag{4.6}$$

where $w_r(t)$ and $v(t)$ denote the process noise of the reduced model and the measurement noise respectively. The original systems additive noise $w(t)$ can be converted to reduced model additive noise $w_r^{(k)}(t)$ using $w_r^{(k)}(t) = \mathscr{U}^{(k),T}w(t)$. For the reduced

model, we can use any state estimation techniques to estimate the state ξ. Once we have the estimate of ξ, $\hat{\xi}$, an estimate of the original system's state $\hat{x}$ can be obtained by switching the model at different operating regions.

4.3 Simulations

In this section, we have considered an agro-hydrological system and performed the above-proposed sensor placement procedure. We have considered a field with the length, width, and depth as 16 m, 4 m, and 0.3 m. The system is discretized into 640 nodes with 16 nodes in the X direction, 4 nodes in the Y direction and 10 nodes in the Z direction. The arrangement of soil is shown in Figure 4.2. The bottom 320 nodes consist of silt loam and sandy clay loam and the top 320 nodes consist of sandy loam and loam. We have applied four different inputs to four different sections on the surface nodes. Four inputs are considered for this system as shown in Figure 4.3.

It is assumed in this work that the irrigation prescription (input profiles) is known in advance and is used in model reduction. Note that this assumption is not that restrictive for state estimation. In the simulations, we consider non-periodic inputs with different amplitudes as presented in Figure 4.4 for illustration purpose. The sprinklers do not irrigate continuously and are turned on for a short period every some time. This represents the typical scenarios in agriculture irrigation.

4.3.1 Model reduction

We first apply the proposed dynamic model reduction algorithm to the system. The operating points and the operating regions are selected based on the input sequence. The operating points are the endpoint of each peak of the input sequence while each operating regions are from the starting point of each peak to the starting point of the next peak. In our simulation, we have selected 6 operating points corresponding to time instants $5, 35, 65, 125, 195, 239$ and the operating regions are determined according to time periods $(0-29), (30-59), (60-119), (120-189), (190-233), (234-240)$. In each operating region, we have used one reduced model based on the proposed dynamic model reduction algorithm.

To evaluate the accuracy of the reduced model, we simulate the original system

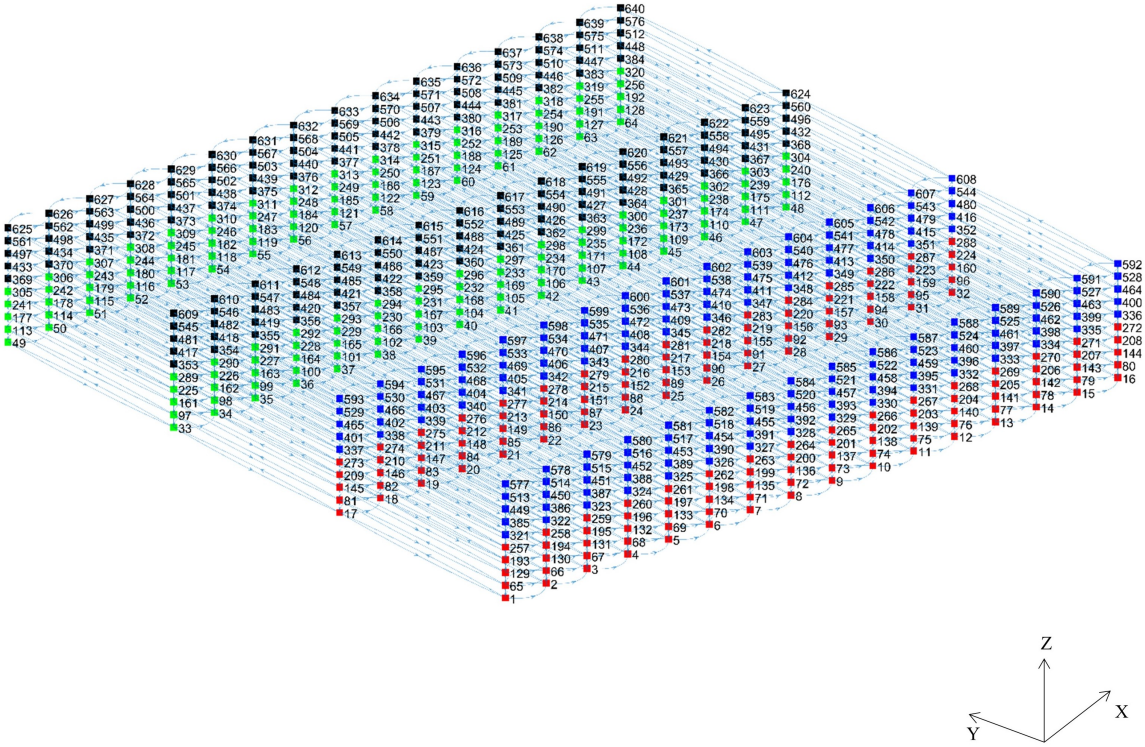


Figure 4.2: Graph representation of the system. A different arrangement of the four different types of soil (silt loam (red), loam (blue), sandy loam (black) and sandy clay loam (green)) is considered

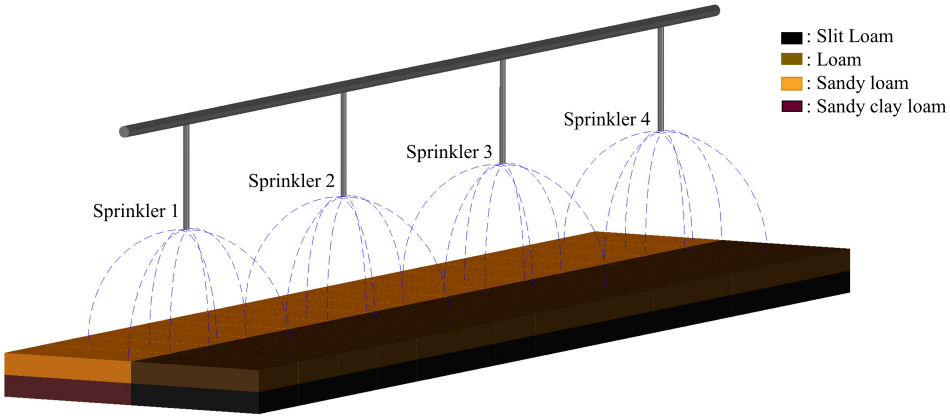


Figure 4.3: Schematic of input and soil types distribution

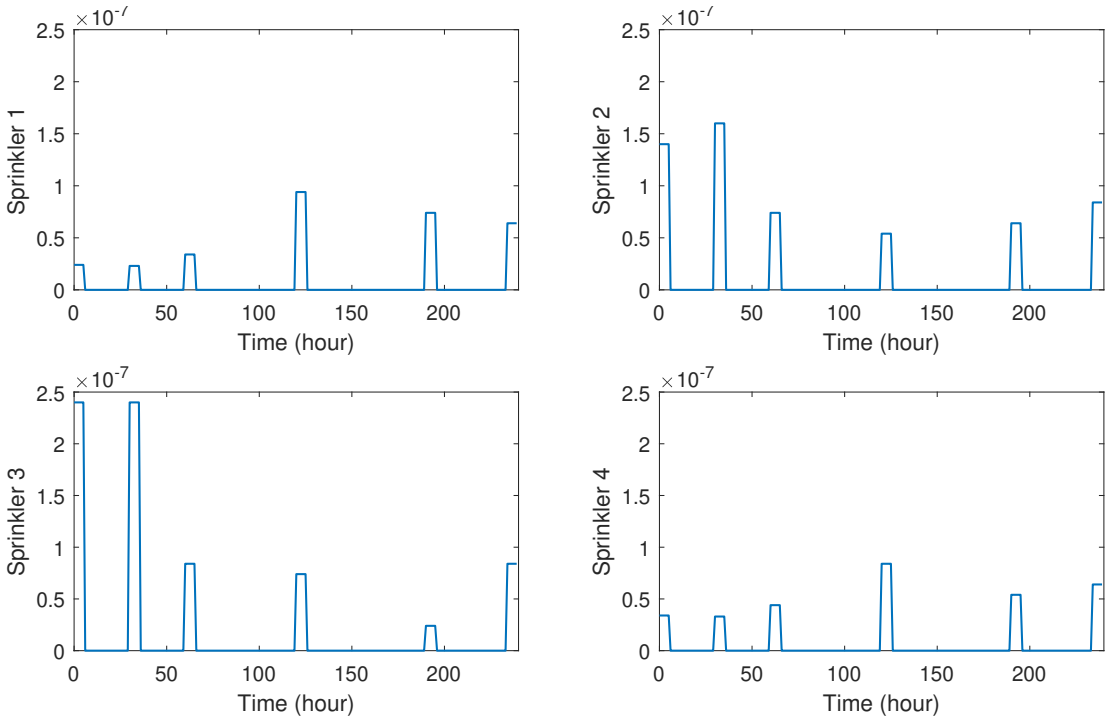


Figure 4.4: Input values of four different sprinklers

and the dynamic reduced-order system under the same conditions. To show the effectiveness of the proposed dynamic model reduction over the static model reduction method in Chapter 3, we obtain the static model reduction at the endpoint of the system trajectory. Figure 4.5 shows the trajectories of the actual system, the dynamic reduced-order system, and the reduced model based on the final time step trajectory. We can observe from Figure 4.5, the static model reduction trajectories are relatively less close to the original system model than the trajectories obtained from the dynamic model reduction. To quantify the error of the dynamic model reduction and static model reduction, he following performance indicator (E_1) is used by considering the trajectories of the original and the reduced system,

$$E_1(t_k) = \sum_{i=1}^{N} \left| \left(x_i(t_k) - \tilde{x}_i(t_k) \right) \right| \tag{4.7}$$

where N denotes the total number of nodes in the original discretized system, $x_i(t_k)$ represents the i^{th} state of the original system at sampling time t_k, $\tilde{x}_i$ is the i^{th} element of the corresponding approximated state $\tilde{x}$ from the reduced model. Figure 4.6 shows the comparison of the average error of all the states for each time step for dynamic

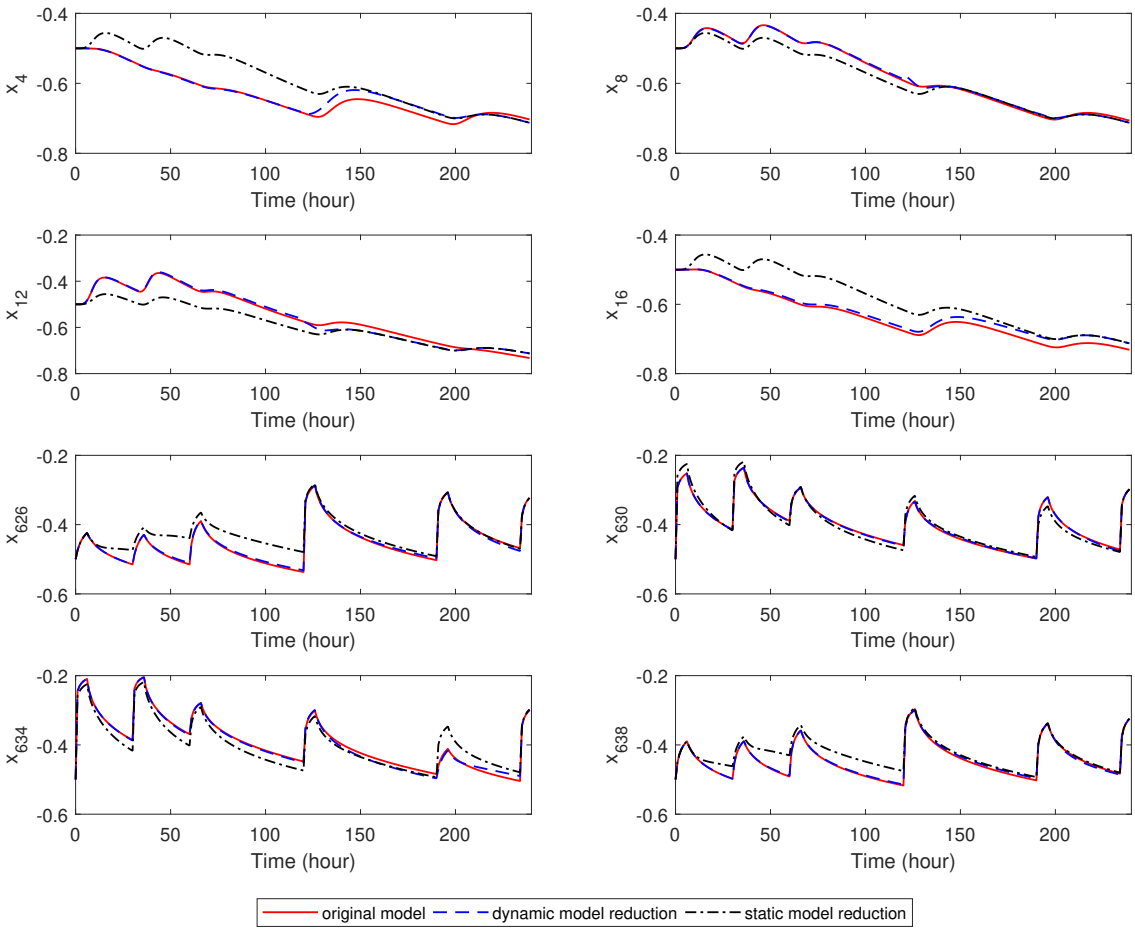


Figure 4.5: Trajectories of the actual states (red solid line), the dynamic reduced model of order 80 (blue dashed line), the reduced model based on final time step state trajectory of order 80 (black dash-dot line)

reduced-order system and static reduced-order system, which shows that the average error of the static model reduction is significantly higher than that of the dynamic reduced-order model.

4.3.2 Minimum number of sensors selection

After obtaining the reduced models at each operating point, we apply both the graphical method and the maximum multiplicity theory to determine the minimum number of sensors. For the system under consideration, the minimum number of sensors using the graphical method is one for all the reduced-order systems. By applying the maximum multiplicity theorem, the resulting minimum number of sensors may change depending upon the numerical threshold for zero. For the system considered, we

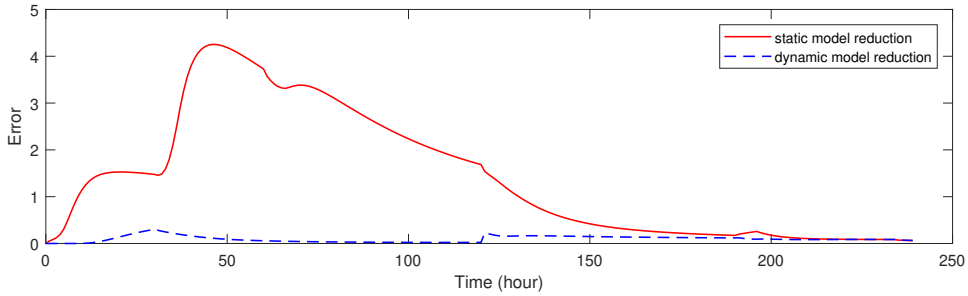


Figure 4.6: Comparison of average error of all states for each time step for dynamic model reduction (blue dashed line) and static model reduction (red solid line)

obtain the minimum number of sensors as one by using the maximum multiplicity theorem as well.

4.3.3 Sensor placement

Once the minimum number of sensors is determined, we apply a modal degree of observability as discussed in section 4.2.3 to determine the optimal sensor locations. Figure 4.7 shows the modal degree of observability of the original system at different selected operating points. We also present the average modal degree of observability of the original system in Figure 4.8. Based on the results shown in Figure 4.8, we can place the sensor at the location that corresponds to any node from $\{25, 26, 27, 28\}$.

4.3.4 State estimation

After obtaining the sensor position, we perform the state estimation using an extended Kalman filter (EKF). To verify the effectiveness of the proposed approach, we place the sensor at the location corresponds to node 25 and at node 609. Figure 4.9 shows the actual state and the estimates based on the optimal location and at node 609. Note that the two extended Kalman filter at two positions are optimally tuned with the same values and with the same noise sequence. From Figure 4.9, we can observe that the filter based on the optimally placed sensor converges to the actual system faster.

To quantify the error of the estimated states from EKF, the following error measure has been used,

$$E_2 = \sum_{i=1}^{N} \left| \left(\tilde{x}_i(t_k) - \hat{x}_i(t_k) \right) \right|$$

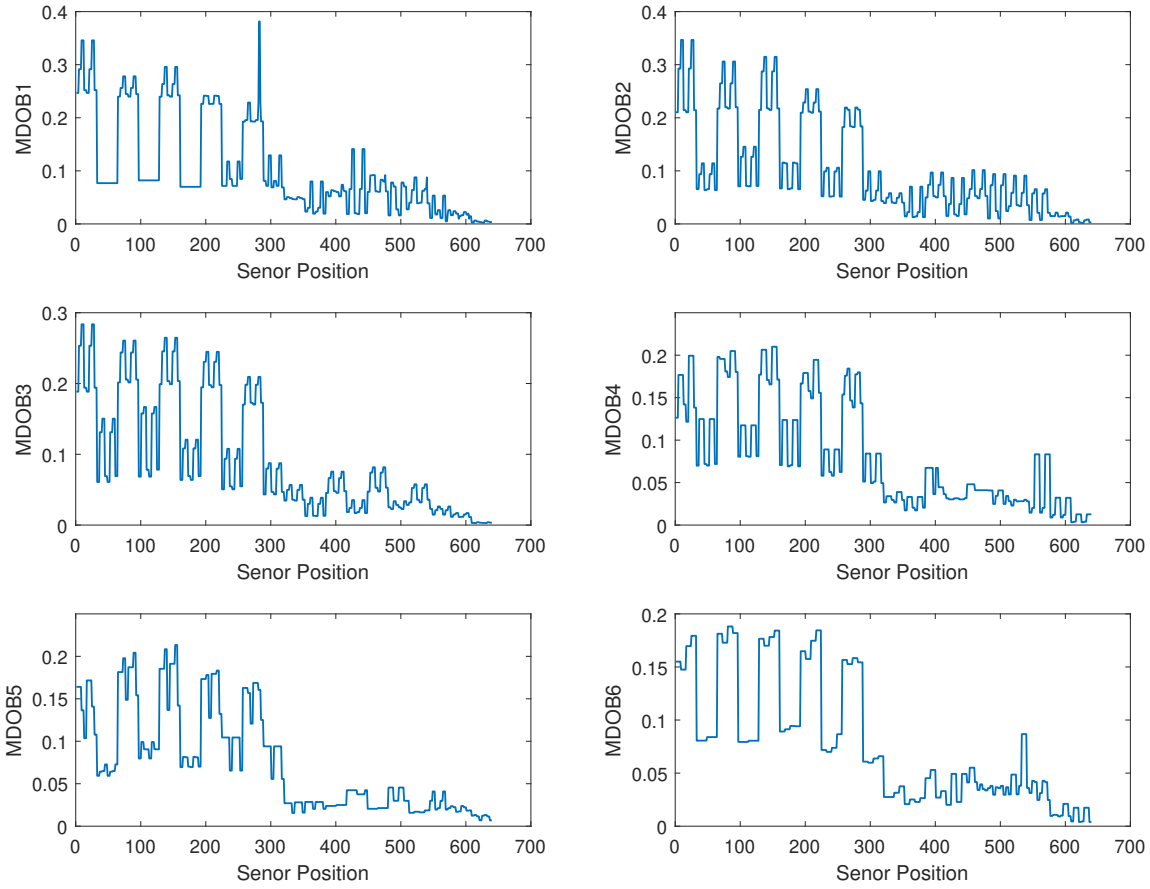


Figure 4.7: Modal degree of observability of the original system at different operating points

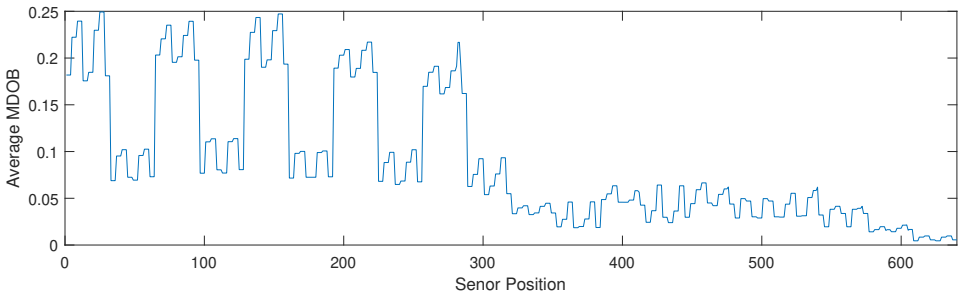


Figure 4.8: Average Modal degree of observability of the original system when the sensor is placed at different nodes

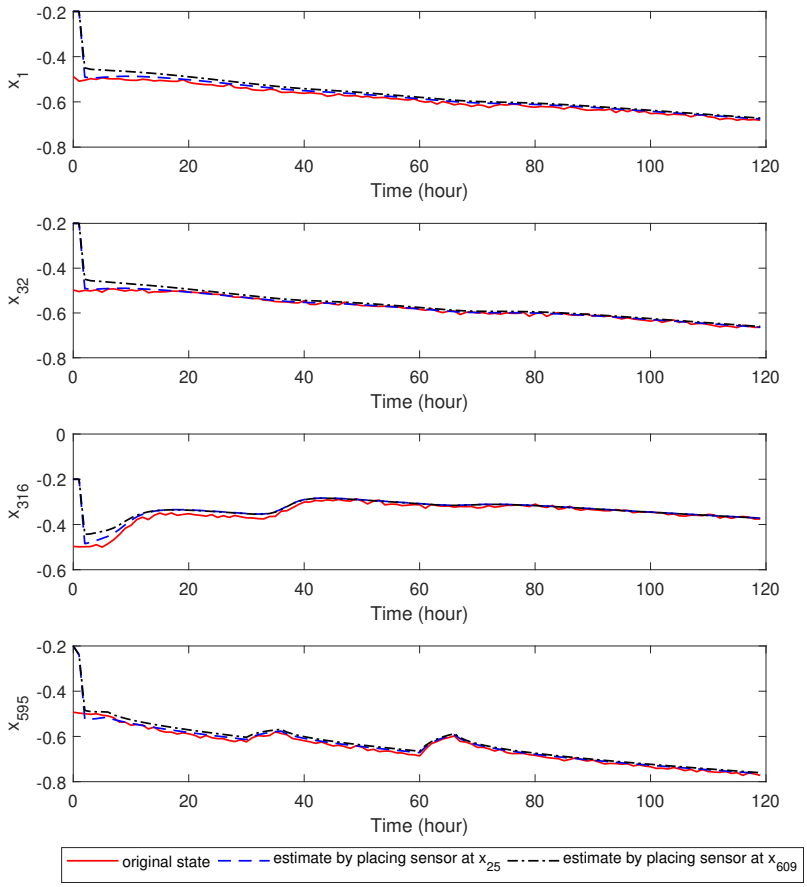


Figure 4.9: Trajectories of the actual states (red solid line), the state estimates with sensor placed at x_{25} (blue dashed line) and the sensor placed at x_{609} (black dash-dot line)

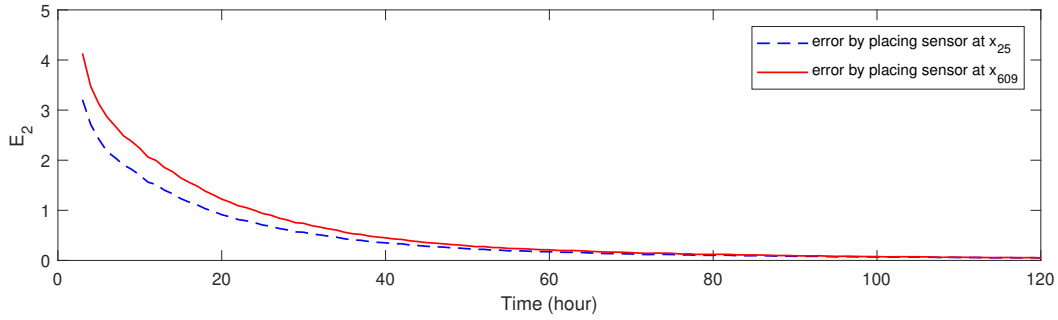


Figure 4.10: Comparison of average error of all states for each time step for sensor placed at x_{25} (blue dashed line) and the sensor placed at x_{609} (red solid line)

where $T_f + 1$ is the total number of samples considered, N denotes the total number of nodes in the original discretized system, $\tilde{x}_i(t_k)$ represents the i^{th} state of the approximated state from reduced model at sampling time t_k, $\hat{x}_i$ is the i^{th} element of the corresponding estimated state $\hat{x}$ from the estimated model. Figure 4.10 shows the comparison of the error measure (E_2) by placing the sensor at location 25 and placing the sensor at location 609, which shows that the estimation error of sensor placed at 609 is higher than that of the estimation error of sensor placed at the optimal location.

4.4 Conclusions

In this chapter, the problem of optimal sensor placement in the presence of non-uniform input was addressed. First, the different operating regions were identified. A different reduced model was constructed for each operating region, and the reduced models have exchanged information at the boundaries of the operating regions. Once the reduced models were created, the minimum sensor and the optimal locations were calculated using observability and modal degree of observability. One simulation case was considered, and the result showed the effectiveness of the proposed method.

Chapter 5

Adaptive model reduction and state estimation using moving horizon estimation

5.1 Introduction

In this chapter, the dynamic model reduction in Chapter 4 is generalized and a structure-preserving adaptive model reduction method for state estimation of large-scale agro-hydrological systems is proposed. In Chapters 3 and 4, the similarities between the states are obtained using the weights from the Jacobian of the nonlinear system. For very large-scale systems, the Jacobian calculation is computationally expensive. For an application like sensor placement, where we perform all the calculations offline, it is still manageable to conduct the computationally expensive calculations. However, the state estimation usually is performed online, so the Jacobian calculation at different points on the trajectories, like dynamic model reduction, is difficult. The above challenge motivates us to use the trajectories of the actual nonlinear system to calculate the similarity between the states. First, the operating regions are calculated using the input sequence. Then the system trajectories are generated using the original nonlinear model. Next, the unsupervised machine learning clustering approach is used to find the states having similar trajectories and put them in one cluster. Afterward, the projection-based nonlinear reduced model is constructed using the clustering information. For each operating region, different reduced models are constructed. As the model may change over time, the changing boundary condition is proposed to move from one model to another. Similar to the

idea of adaptive model reduction, an adaptive moving horizon estimation is proposed. The challenges arise because the model changes over time, and classical MHE cannot handle it. The proposed adaptive reduced-order MHE considers the information exchange of reduced models and estimates the original system's states. The effectiveness of the proposed adaptive MHE is compared with the original MHE for a small system. The proposed adaptive MHE is also applied to the real agricultural field located in Lethbridge, Alberta, with the heterogeneous soil type and with the central pivot irrigation system.

5.2 Problem formulation

As described in Chapter 2, the dimension of u is the same as the number of surface nodes. However, at a specific time, only the elements in u that align with the central pivot may not be zero, and the other elements are zero. This imposes a time-varying constraint on u as follows:

$$\mathscr{U}_{lb}(t) \leq u(t) \leq \mathscr{U}_{ub}(t) \tag{5.1}$$

We consider tensiometer sensors to measure the soil pressure head values (states) at a few selected locations on the field. The number of sensors (N_y) are significantly smaller than the number of states (N_x) of the system ($N_y << N_x$). At every sampling time, the tensiometer sensors provide the soil pressure head value for the corresponding sensor locations.

Therefore, a continuous time state-space model with measurements is considered as follows:

$$\dot{x}(t) = f(x(t), u(t)) + w(t) \tag{5.2a}$$

$$y(t) = Cx(t) + v(t) \tag{5.2b}$$

where $y(t) \in \mathbb{R}^{N_y}$ denotes the soil pressure head measurements at the sensor nodes, $w(t) \in \mathbb{R}^{N_x}$ is the additive system noise and $v(t) \in \mathbb{R}^{N_y}$ denotes the measurement noise.

The objective is to estimate the soil pressure head at each node in the field using sensor measurements for a three-dimensional field with a central pivot irrigation system and heterogeneous soil types.

5.3 Proposed adaptive model reduction

Let us consider the cylindrical three-dimensional Richards equation in (2.14). As discussed in the introduction part that the finer resolution discretization is required to solve the Richards equation. Thus the number of states in (5.2) becomes very high, and it is challenging to perform the optimization-based state estimation. Model reduction is one of the solutions to handle the problem. In this section, we propose a structure-preserving adaptive model reduction method to address this issue.

There are mainly two challenges associated with the classical model reduction algorithms: 1) The reduced model does not preserve the physical significance of the states. Given that the goal of the model reduction is to perform state estimation based on MHE, it is preferred that the reduced-model preserves the system network topology while applying state constraints to the MHE. 2) In general, the reduced model tries to capture the actual system trajectory using low-dimensional space. For an agricultural field with a center pivot irrigation system, irrigation water is not applied to the entire field simultaneously. At a specific time, only the small segment of the field that covered by the center pivot is irrigated. For the segment that is irrigated, the water soil moisture changes quickly due to irrigation. For the rest of the field (majority of the field), the water soil moisture changes much slower. To capture the dynamics of this specific time instant, a reduced model with more states kept for the segment been irrigated and relative less for the rest of the field is appropriate. If a single model is used to capture the dynamics of the field as the center pivot rotates, much more states are needed in the reduced model.

These two challenges motivate us to construct a structure-preserving adaptive model reduction method based on the unsupervised machine learning clustering technique. The motivation to use the adaptive model reduction in the agro-hydrological system is illustrated using a simple example in below section 5.3.1.

5.3.1 Motivating example

In the agro-hydrological system, the input usually follows a non-periodic way. For instance, the central pivot as an irrigation system circularly puts water (Figure 1.3). So the system dynamics may change substantially both in spatial and temporal di-

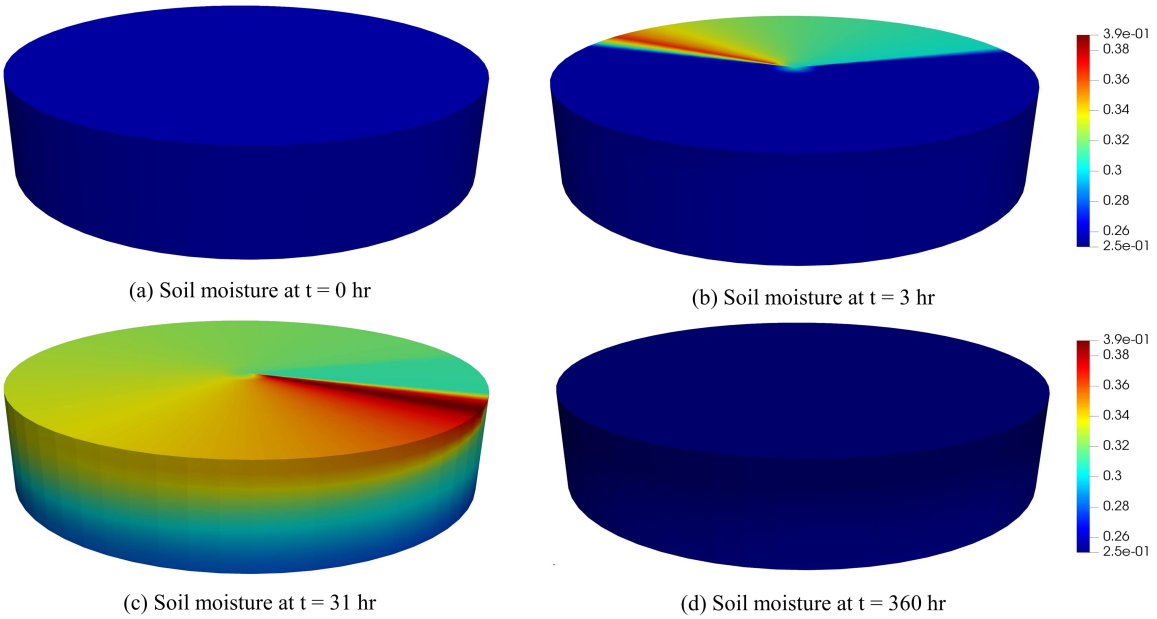


Figure 5.1: Motivational example for adaptive model reduction

rections from one node to another. For example, consider a field with uniform initial conditions and homogeneous soil types. The central pivot takes 8 hrs to complete one circle. The total simulation time period is 15 days. For the first two days, there is continuous irrigation, and the irrigation amount is zero for the other remaining days. From Figure 5.1(a) at $t = 0$ hr, we can observe that the soil moisture all over the field is constant, and the system can be represented by only one node. When the irrigation starts, the soil moisture changes spatially and temporally, more nodes in a reduced model are required to capture the variation in the system. From Figures 5.1(b) and 5.1(c), we notice that at the 3^{rd} hr, the variation of the soil moisture across the field is relatively smaller compared to the 31^{th} hr. After we stop the irrigation, the variation of the soil moisture across the field becomes small gradually, which implies that few number nodes are required to capture the dynamics in the reduced model. If we represent the system with only one global reduced model, we may need to use a higher-order reduced model to capture all these changes. It is preferable to change the reduced model over time based on the current situation of the system.

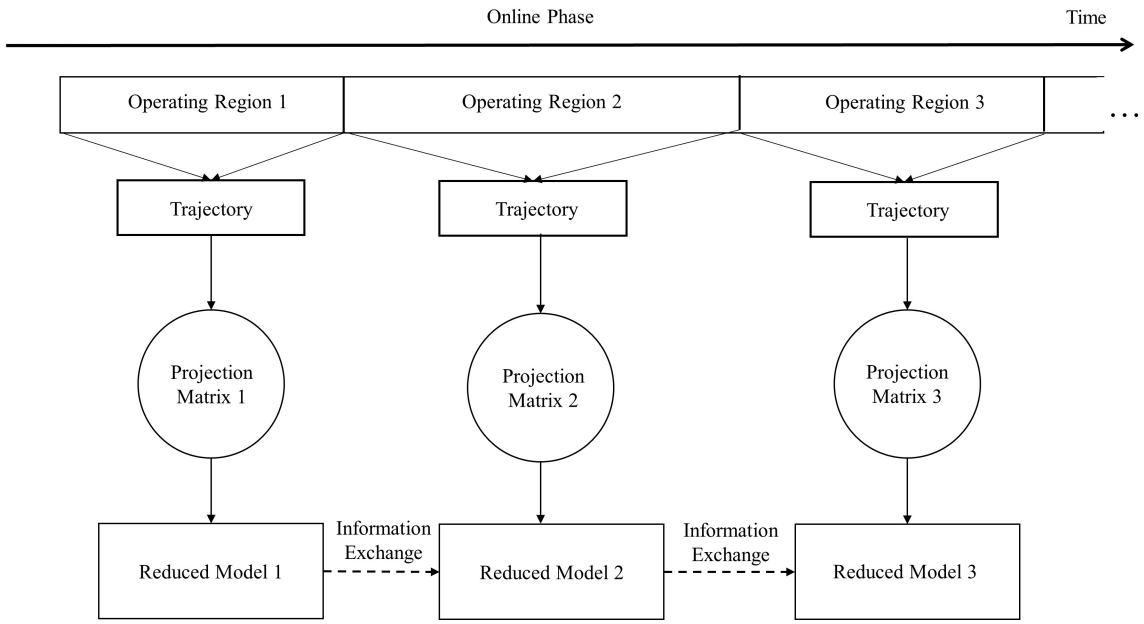


Figure 5.2: Steps to calculate adaptive reduced model

5.3.2 Proposed method

The key steps to calculate the adaptive model reduction are shown in Figure 5.2. The figure shows that the first step is to divide the total time into different operating regions. In the second step, the trajectories are generated for each operating region. In the third step, the clusters and the projection matrix are generated based on the similarity between the trajectories. In the same step, using the projection matrix, the reduced-order model is obtained. In the final stage, the information exchange between the reduced-order model is considered. The illustration of the proposed adaptive model is shown in Figure 5.3. The details of each step in the proposed algorithm are described as follows.

Step 1: operating region calculation

The idea in this step is to divide the total time period of the system into different operating regions based on input sequence or system trajectories. The operating region may be determined based upon the irrigation prescriptions (how the center pivot moves). It is more a heuristic-based approach. Let us consider the motivating example again. While there is irrigation for the first two days, the system dynamics change rapidly every sampling time. So in the first two days, we may divide the time with

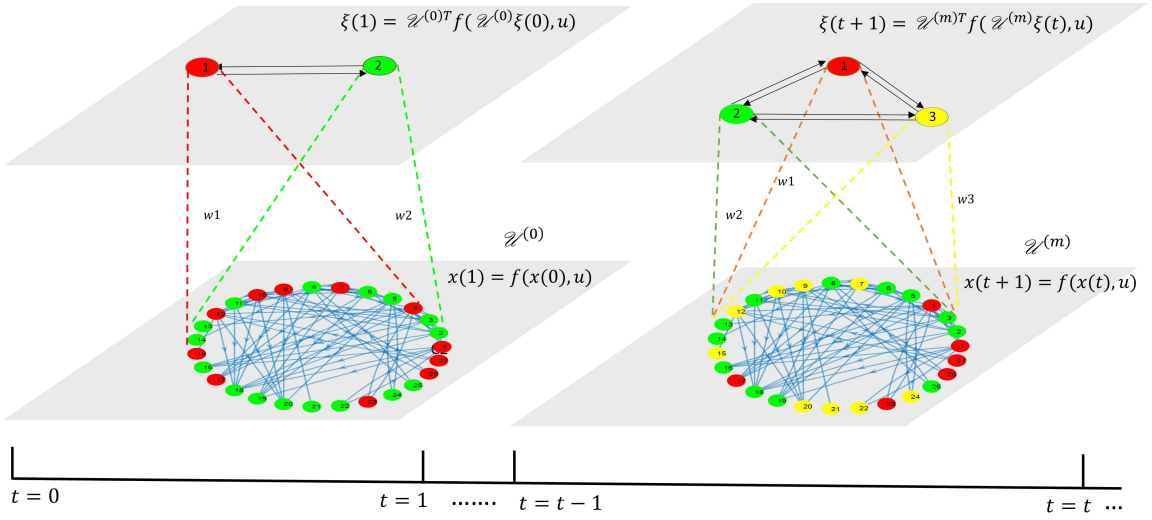


Figure 5.3: Illustration of adaptive model reduction

finer size operating regions. After the irrigation stops, the system dynamics decreases gradually, so the coarse size operating regions may be used. For each operating region, one reduced model is constructed. Let us consider $\mathscr{R} = \{\mathscr{R}_1, \mathscr{R}_2, \ldots, \mathscr{R}_s\}$ as the collection of operating regions, which has the following property: $\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots \cup \mathscr{R}_s = \mathscr{T}$, where $\mathscr{T} = [0, T_f]$, T_f is total operating time, s is the total number of operating regions and $\mathscr{R}_1 \cap \mathscr{R}_2 \cap \ldots \cap \mathscr{R}_s \neq \emptyset$ may hold. Things to note that the cardinality of each set of operating regions may be different.

Step 2: trajectory generation

In this step, the state snapshots are generated for each operating region. Let us consider the nonlinear system in (2.15). Based on a prescribed input, simulate the original nonlinear system and capture the trajectories in each operating region. For each operating region ($\mathscr{R}_m \in \mathscr{R}$), the trajectories can be generated as follows:

$$\mathscr{X}_m = [x(t_{0,m})\ x(t_{1,m})\ \ldots\ x(t_{N_m,m})]$$

where $\mathscr{X}_m \in \mathbb{R}^{n \times N_m}$ is the snapshot matrix of actual system for operating region $\mathscr{R}_m$, n is the number of states, N_m is number of sampling interval in the operating region ($|\mathscr{R}_m| = N_m$). Note that this step is performed online and the trajectories are generated at the beginning of the operating region at time $t_{0,m}$.

Step 3: projection matrix and reduced order model generation

After generating the operating regions and the corresponding trajectories, one reduced model is created for each operating region. The reduced model is generated based on the trajectory of the system. The main idea is to investigate the system trajectories and create state cluster sets for each operating region. The states having similar dynamics based on the state trajectories are put into the same clusters. In this chapter, the agglomerative hierarchical clustering [108] is used. We use the Euclidean distance between trajectories as the distance measure for states. The main reason to choose agglomerative hierarchical clustering is because of the capability to define the distance threshold between the clusters instead of predefining the number of cluster sets. The distance threshold is a tuning parameter for the accuracy of reduced model. There are three commonly used linkage methods present in agglomerative hierarchical clustering (e.g. single, average, complete linkage). In this chapter, we use the average linkage, and it considers the average distance between each point in one cluster to every point in other clusters and it is calculated as follows:

$$D(p,q) = \frac{1}{n_p n_q} \sum_{i=1}^{n_p} \sum_{j=1}^{n_q} d(x_{pi}, x_{qj})$$

and

$$d(x_{pi}, x_{qj})^2 = \sum_{k=1}^{n} (x_{pik} - x_{qjk})^2$$

where p and q are two clusters, i and j are data points within the clusters, d is the Euclidean distance between i and j, n_p, n_q are the size of the clusters of p and q respectively and n is the dimension of state vector.

Let us consider $\mathscr{C}^{(m)} = \{\mathscr{C}_1^{(m)}, \mathscr{C}_2^{(m)}, \ldots, \mathscr{C}_r^{(m)}\}$ be the collection of clusters for operating region ($\mathscr{R}_m$) after the hierarchical clustering and r is the order of the resulted reduced model. The properties of the resultant clusters are discussed in *Definition 1*. The adaptive reduced-order system is constructed based on the Petrov-Galerkin projection framework and the projection matrix ($\mathscr{U}^{(m)}$) can be obtained using *Definition 2* in Chapter 2.

The adaptive reduced model of (2.15) for each operating region ($\mathscr{R}_m \in \mathscr{R}$) is expressed as:

$$\dot{\xi}^{(m)}(t) = f_r^{(m)}(\xi^{(m)}(t), u(t)) \tag{5.3}$$

where $f_r^{(m)}(\xi^{(m)}(t), u(t)) = \mathscr{U}^{(m)T} f(\mathscr{U}^{(m)}\xi^{(m)}(t), u(t))$ and $\xi^{(m)}(t) = \mathscr{U}^{(m)T} x(t)$. Note that the actual state x can be approximated based on mapping $\tilde{x}(t) = \mathscr{U}^{(m)}\xi^{(m)}$ in operating region ($\mathscr{R}_m$). The discrete model of (5.3) is expressed as follows:

$$\xi^{(m)}(k+1) = f_{rd}^{(m)}(\xi^{(m)}(k), u(k)) \tag{5.4}$$

Step 4: information exchange between reduced models

For each operating region, we may get a different order reduced-order model. Thus, when the system moves from one operating region to another operating region, the corresponding reduced models are also required to be switched. To handle the impact of changing reduced-order models from one operating region to other operating regions, we consider information exchange between reduced models at the boundary of the operating regions. The steps to calculate the changing boundary is shown in Figure 5.4 and the algorithm as follows:

1. Let us consider two operating regions $\mathscr{R}_m$ and $\mathscr{R}_{m+1}$ at boundary of time k. Compute the states of the reduced system at k using the projection matrix $\mathscr{U}^{(m)}$ which is generated for the operating region $\mathscr{R}_m$ and, the past reduced state and input information.

$$\xi^{(m)}(k) = \mathscr{U}^{(m)T} f_d(\mathscr{U}^{(m)}\xi^{(m)}(k-1), u(k-1))$$

2. Covert the reduced system to approximated original system state ($\tilde{x}(k)$) at k using: $\tilde{x}(k) = \mathscr{U}^{(m)}\xi^{(m)}(k)$

3. Compute the initial value for next operating region reduced model as $\xi^{(m+1)}(k) = \mathscr{U}^{(m+1)}\tilde{x}(k)$

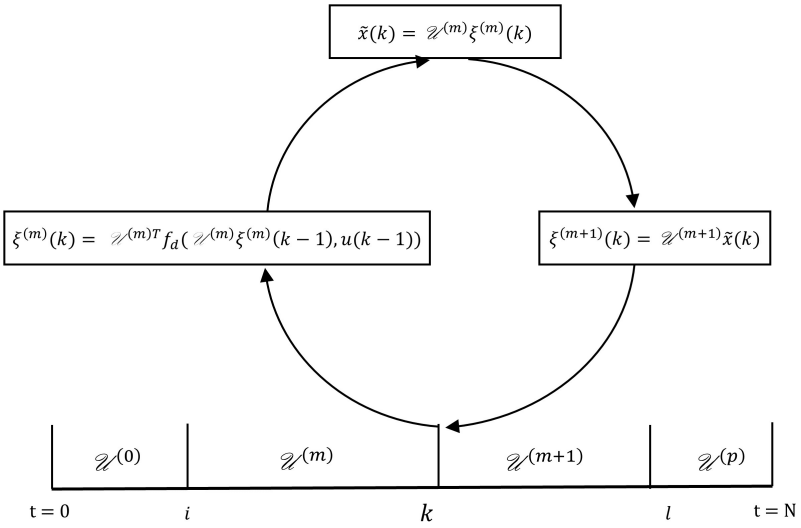


Figure 5.4: Changing boundary from one reduced model to another

The main advantages of using the adaptive reduced-order system are as follows:

1. The adaptive model reduction approach considers the local dynamics of the system, so the number of states of the adaptive reduced model is lesser than the global model reduction models.

2. In MHE, the number of decision variables is the number of states times the horizon length. For a large agricultural field, the number of states (x) is very high so as the decision variables and make the optimization problem intractable. In the proposed adaptive reduced model, the number of reduced states (ξ) is way lesser than the actual system states ($\xi << x$). Thus the number of decision variables in the adaptive MHE is lesser and makes the optimization problem tractable.

3. The degree of observability of the adaptive reduced model is higher than the original model, which signifies that using the same number of measurements, the adaptive MHE can estimates states faster than the original MHE.

4. The adaptive model reduction is also helpful while estimating using EKF or EnKF. For EKF, instead of calculating the huge jacobian matrix, the reduced-order jacobian matrix can be used. Similarly, the large matrix calculations can be avoided while using EnkF.

5. The adaptive reduced-order model is also helpful in other control problems like sensor placement and controller design.

5.4 Adaptive moving horizon estimation

This section proposes the adaptive moving horizon estimation technique based on the adaptive reduced model. As the reduced model order changes over time, using the classical moving horizon method is not feasible. In this section, we propose the adaptive MHE design. In the proposed design, the length of the operating period is considered to be the same as the horizon length of the MHE for simplicity. This also implies that the reduced model may change every sampling time because the horizon changes every time instant. The details of proposed adaptive MHE are described as follows.

Let us consider the discrete reduced model (5.4) at each operating region with additive process and measurement noise as follows:

$$\begin{aligned} \xi^{(k)}(k+1) = &\quad f_{rd}^{(k)}(\xi^{(k)}(k), u(k)) + w_r^{(k)}(k) \\ y(k) = &\quad C\mathscr{U}^{(k)}\xi^{(k)}(k) + v(k) \end{aligned} \tag{5.5}$$

where $w_r^{(k)}(k)$ denotes the process noise of the reduced model at time k and operating region k and $v(t)$ is the measurement noise. Note that if the original nonlinear system also has additive process noise and is $w(k)$, then $w_r^{(k)}(k) = \mathscr{U}^{(k),T} w(k)$. In MHE approach, for time (k) less than horizon length (N), the full information problem (5.6) is solved while the time (k) greater than horizon length (N), the MHE problem with arrival cost is solved with (5.7).

$$\phi_k = \min_{\substack{\hat{\xi}_{0|k}^{(k)}, \\ \{\hat{w}_r{}^{(k)}\}_{k=0}^{k-1}}} \sum_{j=0}^{k-1} ||\hat{w}_r{}^{(k)}(j)||^2_{\{Q_r^{(k)}\}^{-1}} + \sum_{j=0}^{k} ||\hat{v}(j)||^2_{R^{-1}} + ||\hat{\xi}_{0|k}^{(k)} - \bar{\xi}_0^{(k)}||^2_{\{P_r^{(k)}\}^{-1}} \tag{5.6a}$$

$$\text{s.t. } \hat{\xi}^{(k)}(j+1) = f_{rd}^{(k)}(\hat{\xi}^{(k)}(j), u(j)) + \hat{w}_r{}^{(k)}(j), \quad j\epsilon[0, k-1] \tag{5.6b}$$

$$\hat{v}(j) = y(j) - C_r^{(k)}\hat{\xi}^{(k)}(j), \quad j\epsilon[0, k] \tag{5.6c}$$

$$\hat{\xi}^{(k)}(j) \in \mathbb{E}^{(k)}, \ \hat{w}_r{}^{(k)}(j) \in \mathbb{W}^{(k)}, \ \hat{v}(j) \in \mathbb{V}_f \tag{5.6d}$$

$$\Phi_k = \min_{\substack{\hat{\xi}^{(k)}_{k-N|k}, \\ \{\hat{w_r}^{(k)}\}^{k-1}_{k=k-N}}} \sum_{j=k-N}^{k-1} ||\hat{w_r}^{(k)}(j)||^2_{\{Q_r^{(k)}\}^{-1}} + \sum_{j=k-N}^{k} ||\hat{v}(j)||^2_{R^{-1}} + ||\hat{\xi}^{(k)}_{k-N|k} - \bar{\xi}^{(k)}_{k-N}||^2_{\{P_r^{(k)}\}^{-1}} \tag{5.7a}$$

$$\text{s.t. } \hat{\xi}^{(k)}(j+1) = f_{rd}^{(k)}(\hat{\xi}^{(k)}(j), u(j)) + \hat{w_r}^{(k)}(j), \quad j\epsilon[k-N, k-1] \tag{5.7b}$$

$$\hat{v}(j) = y(j) - C_r^{(k)}\hat{\xi}^{(k)}(j), \quad j\epsilon[k-N, k] \tag{5.7c}$$

$$\hat{\xi}^{(k)}(j) \in \mathbb{E}^{(k)}, \ \hat{w_r}^{(k)}(j) \in \mathbb{W}^{(k)}, \ \hat{v}(j) \in \mathbb{V}_f \tag{5.7d}$$

where $\hat{w}_r^{(k)}$ denotes the estimated reduced order system disturbance, $\hat{v}_r^{(k)}$ denotes the estimated measurement noise and $\hat{\xi}_r^{(k)}$ denotes the estimated reduced order system states. In Equations (5.6a), (5.7a), the $Q_r^{(k)}$ and R are symmetric positive definite weight matrices which penalize the estimated system noise and output noise respectively. The weighting matrix $P_r^{(k)}$ makes sure that the estimated system state ($\hat{\xi}_0^{(k)}$) is not far away from the initial guess of the system ($\bar{\xi}_0^{(k)}$). The $Q_r^{(k)}$ is calculated based on the projection of full order system Q matrix ($Q_r^{(k)} = \mathscr{U}^{(k)} Q \mathscr{U}^{(k)T}$). Similarly the weighting matrix $P_r^{(k)}$ is calculated using full order system P matrix ($P_r^{(k)} = \mathscr{U}^{(k)} P \mathscr{U}^{(k)T}$). The Equations (5.6b), (5.7b) are the reduced order models with process noise and the Equations (5.6c), (5.7c) are the output models with measurement noise. Equations (5.6d), (5.7d) represent the constraints for the estimated system states, estimated process noise and estimated measurement noise.

Let us first discuss the steps for adaptive full information estimation problem. First we obtain the measurements ($y_{0:k}$) from time 0 to current time k. Then the snapshot matrix is generated based on simulating the original non-linear system from time $t = 0$ to $t = k+1$ using the initial guess $\bar{x}_0$ and input. After that the projection matrix ($\mathscr{U}^{(k)}$) is generated for current time k as discussed in Section 3. The reduced model is generated as per Equation (5.5) using the generated projection matrix. The initial guess of the full order system states is converted into reduced order guess using $\bar{\xi}_0^{(k)} = \mathscr{U}^{(k)T} \bar{x}_0$. Then the optimization problem Equation (5.6) is solved using the initial guess of the reduced order system and measurements collected from time zero. After solving the optimization problem, we obtain the current estimates of the reduced order system states $\hat{\xi}_{j|k}^{(k)}$ from time $j = 0$ to $j = k$. At the final step, the estimated reduced order system state at current time is projected back into full order system states using ($\hat{\hat{x}}_{k|k} = \mathscr{U}^{(k)} \hat{\xi}_{k|k}^{(k)}$) and then continue the loop till $k < N$.

The moving horizon estimation at ($k > N$) also follows the similar algorithm as the full information estimation problem. First the measurement is collected from $t = k - N$ to current time k. Then snapshot matrix is generated from time $k - N$ to $k+1$ based on initial condition $\bar{x}_{k-N}$ which is the estimated state $\hat{\bar{x}}_{k-N|k-N}$ obtained at $k - N$. Then the projection matrix followed by the reduced order non-linear model is calculated. In the next step, the reduced-order system initial guess for the optimization problem (5.7) is calculated using $\bar{\xi}^{(k)}_{k-N} = \mathscr{U}^{(k)T}\bar{x}_{k-N}$. After that the optimization problem Equation (5.7) is solved to obtain the reduced order estimates $\hat{\xi}^{(k)}_{j|k}$ for $j = k - N, \ldots, k$. Next, we obtain the estimate of approximated full order system states using the back projection $\hat{\bar{x}}_{k|k} = \mathscr{U}^{(k)}\hat{\xi}^{(k)}_{k|k}$. Finally increase the time, update the initial guess and continue the loop till final time. The details of the adaptive MHE algorithm is presented in Algorithm 3.

5.5 Application to a small field: simulation case

In this section, we apply the proposed algorithms to a small simulated field. First, the performance of the proposed adaptive MHE is compared with the original MHE results. The original MHE is designed based on the full-order system. Secondly, the robustness of the adaptive MHE is discussed based on the different initial guesses of the estimator. The main motive to use the small simulated field in this section is that it is convenient to simulate the original MHE for a small-scale system. As the number of nodes increases, the computational cost increases exponentially for the original MHE. The original MHE used in this simulation is formulated as follows:

$$\Gamma_k = \min_{\substack{\hat{x}_{k-N},\\ \{\hat{w}_k\}_{k=k-N}^{k-1}}} \sum_{j=k-N}^{k-1} ||\hat{w}(j)||^2_{Q^{-1}} + \sum_{j=k-N}^{k} ||\hat{v}(j)||^2_{R^{-1}} + ||\hat{x}_{k-N} - \bar{x}_{k-N|k-N}||^2_{P^{-1}} \quad (5.8a)$$

$$\text{s.t. } \hat{x}(j+1) = f(\hat{x}(j), u(j)) + \hat{w(j)}, \quad j\epsilon[k-N, k-1] \quad (5.8b)$$

$$\hat{v}(j) = y(j) - C\hat{x}(j), \quad j\epsilon[k-N, k] \quad (5.8c)$$

$$\mathbb{X}_{lb} \leq \hat{x}(j) \leq \mathbb{X}_{ub}, \ \mathbb{W}_{lb} \leq \hat{w}(j) \leq \mathbb{W}_{ub}, \ \hat{v}(j) \in \mathbb{V} \quad (5.8d)$$

where $\hat{x}_{K-N}$ and $\hat{w}_k$ are the decision variables to the optimization problem. N denotes the length of the optimization problem, and k is the current time instance. The

Algorithm 3 Adaptive moving horizon estimation algorithm

Input: Initial guess $\bar{x}_0$, covariance matrices ($P_0 > 0, Q, R > 0$), horizon length ($N \geq 1$), simulation length (T)

Initialization $\bar{x}_{k-N} \leftarrow \bar{x}_0$

Output: Estimated states in the order of actual system

for $k = 0 \dots T$ **do**
  if $k < N$ **then**
    Obtain measurements $Y = y_{0:k}$
    Generate snapshot matrix : $X = [\bar{x}_0 \dots \ \bar{x}_{k+1}]$
    Generate projection matrix ($\mathscr{U}^{(k)}$)
    Obtain reduced nonlinear model using Equation (6)
    Covert the initial guess to reduced order guess $\bar{\xi}_0^{(k)} = \mathscr{U}^{(k)T} \bar{x}_0$
    Solve ϕ_k with reduced nonlinear model, $\bar{\xi}_0^{(k)}$ and Y
    Obtain current estimates $\hat{\xi}_{j|k}^{(k)}$ for $j = 0, \dots, k$
    Convert the reduced order estimate to approximated original state $\hat{\bar{x}}_{k|k} = \mathscr{U}^{(k)} \hat{\xi}_{k|k}^{(k)}$
  else
    Obtain measurements $Y = y_{k-N:k}$
    Generate snapshot matrix : $X = [\bar{x}_{k-N} \dots \ \bar{x}_{k+1}]$ with $\bar{x}_{k-N}$ as initial condition
    Generate projection matrix ($\mathscr{U}^{(k)}$)
    Obtain reduced nonlinear model using Equation (6)
    Covert the initial guess to reduced order guess $\bar{\xi}_{k-N}^{(k)} = \mathscr{U}^{(k)T} \bar{x}_{k-N}$
    Solve Φ_k with reduced nonlinear model, $\bar{\xi}_{k-N}^{(k)}$ and Y
    Obtain current reduced order estimates $\hat{\xi}_{j|k}^{(k)}$ for $j = k - N, \dots, k$
    Convert the estimates to approximated original states $\hat{\bar{x}}_{k|k} = \mathscr{U}^{(k)} \hat{\xi}_{k|k}^{(k)}$
    $\bar{x}_{k-N} \leftarrow \hat{\bar{x}}_{k-N+1|k-N+1}$
  end if
end for

optimization problem objective is to minimize the error of predicted and measurements ($||\hat{v}||^2_{R^{-1}}$), the process disturbance ($||\hat{w}||^2_{Q^{-1}}$) and the arrival cost ($|.|^2_{P^{-1}}$). The arrival cost summarizes the cost from the initial time to the model until the beginning of the estimation window. The states and model uncertainty is bounded. P, Q, R are the constant positive define matrices and the tuning parameters of the optimization problem. P matrix is calculated based on the following formula:

$$P = l \times M^T M \tag{5.9}$$

where $M = 0.5(\mathbb{X}_{lb} - \mathbb{X}_{ub})$ and l is a tuning parameter.

Table 5.1: The parameters of loamy soil

K_s (m/s)	θ_s (m^3/m^3)	θ_r (m^3/m^3)	α (1/m)	n (-)
2.889×10^{-6}	0.430	0.0780	3.60	1.56

We consider a circular field of radius 10 m and depth 0.15 m with loam soil type. Table 5.1 shows the hydraulic properties of the loam soil used in the simulation. The field is discretized into 400 nodes with 5 nodes in radial, 8 nodes in azimuthal and 10 nodes in the axial direction. The soil pressure heads at these discretization nodes are the states of the system. Total simulation time is 80hrs, and the time step used for temporal discretization is 6 mins. The central pivot is considered as the irrigation system, and the central pivot's rotating speed is 0.01745 m/s. Non-uniform irrigation has been considered for over 100 hours, and the irrigation amount is shown in Figure 5.5. Sugar beet at its development stage is considered as the crop for the field. The reference evapotranspiration value is considered as 0.085 mm/hr, and the crop-coefficient value is 0.35. We consider 3 sensors at the depth of 13.5 cm (states 89, 185, 281). Process and measurement noise are considered for the simulation with zero mean and standard deviation of 0.15 m and 0.15 m, respectively.

5.5.1 Performance comparison of proposed adaptive MHE and original MHE

In this subsection, we compare the performance of the proposed adaptive MHE with the original MHE. The actual system is used for generating measurements and it is simulated with a homogeneous initial condition of -1.0 m pressure head. The initial guesses for both the adaptive MHE and original MHE is considered the 50% of the actual initial state. The estimation window size is 3. For the original MHE, the Q and R values are considered a diagonal matrix with 10^{-6} and 0.01 values, respectively and P matrix is designed based on Equation (5.9). For the adaptive MHE, Q_r, P_r in Equations (5.6a, 5.7a) are calculated based on ($P_r^{(k)} = \mathscr{U}^{(k)} P \mathscr{U}^{(k)T}$) and ($Q_r^{(k)} = \mathscr{U}^{(k)} Q \mathscr{U}^{(k)T}$).

First, the number of clusters each time for the adaptive MHE is shown in Figure 5.6. It shows that the number of clusters decrease and increase based upon the dynamics of the system. Figure 5.7 shows some of the selected state trajectories of

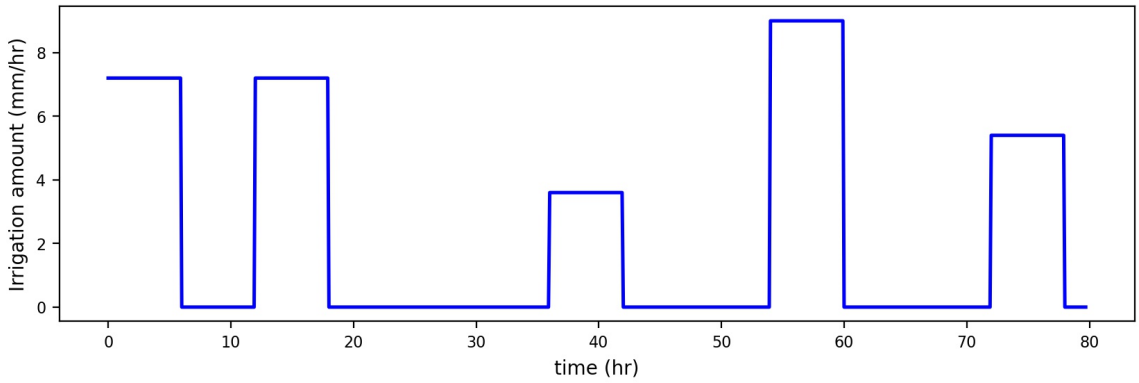


Figure 5.5: Irrigation amount for the small field

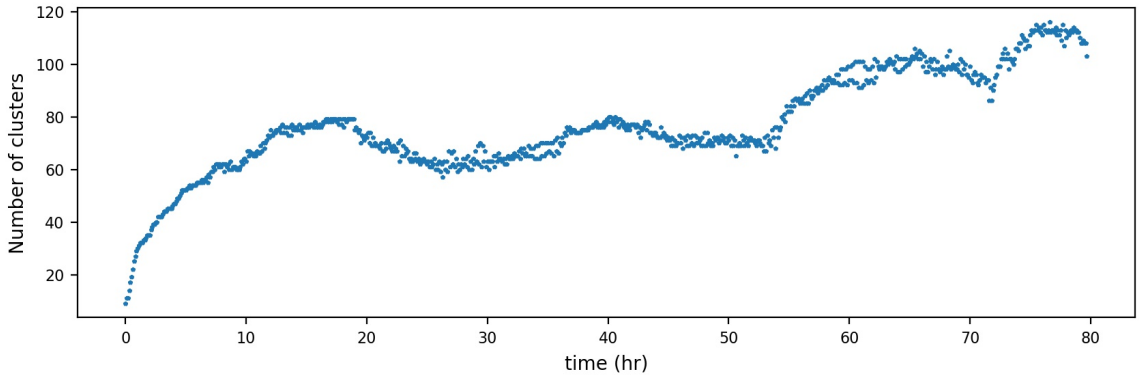


Figure 5.6: Number of clusters for the small field

the actual system and the estimated states by the original MHE and the proposed adaptive MHE. It can be observed that both the original MHE and the proposed adaptive MHE can track the actual process very well.

To further investigate the estimation performance, we consider the mean square error (MSE) of the estimation error as the performance indicator. Figure 5.8 shows the MSE of the original MHE and adaptive MHE. It can be observed that the MSE of the original MHE is higher than the MSE of adaptive MHE at the beginning, and then it converges to the nearly identical value. It can be explained based on the degree of observability. From Figure 5.6, we see that the number of clusters at the initial time is 9, which means the number of reduced states is 9. We have 3 measurements, the degree of observability to estimate 9 states in adaptive MHE is higher than that of estimating 400 states in original MHE. Therefore, in the beginning, the MSE of the adaptive MHE is lower than that of the original MHE. Another observation is that

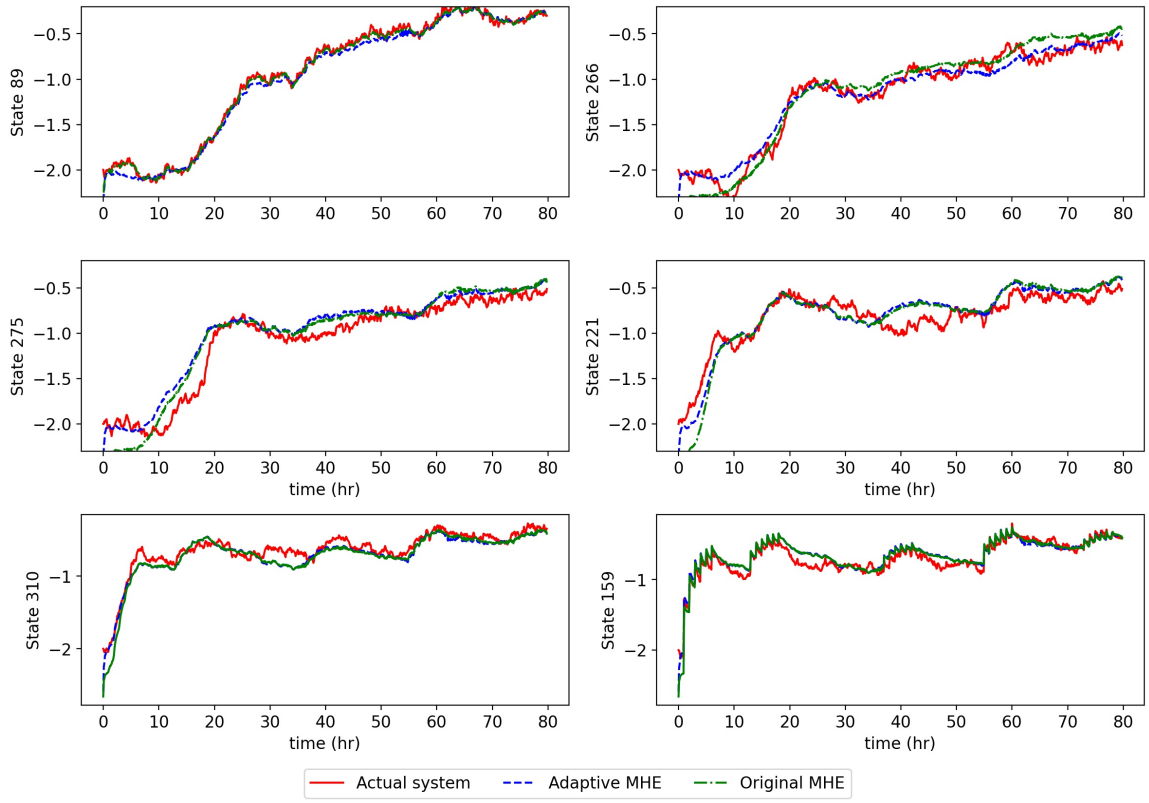


Figure 5.7: Selected state trajectories of the actual states (red solid line), estimated states using adaptive MHE (blue dotted line) and estimated states using original MHE (green dashed line)

Table 5.2: Computational speed comparison of original MHE and adaptive MHE

Original MHE	Adaptive MHE
7773 sec	546 sec

the system does not decrease to zero values. It is because of presence of noise in the process and measurement variables.

The above simulations were performed in a desktop computer with Intel i7-6700 CPU at 3.4GHz and 16.0GB RAM. The computation speed for entire simulation time of the original MHE and adaptive MHE (including clustering, model reduction and optimization) is shown in Table 5.2. It can be observed that the proposed adaptive speed is nearly 14 times faster than the original MHE.

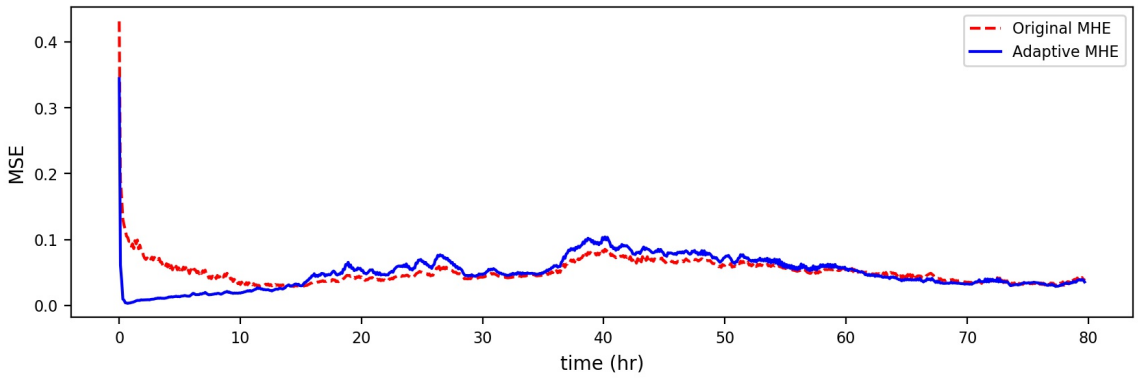


Figure 5.8: Mean square error of the original MHE (red solid line) and the adaptive MHE (blue solid line)

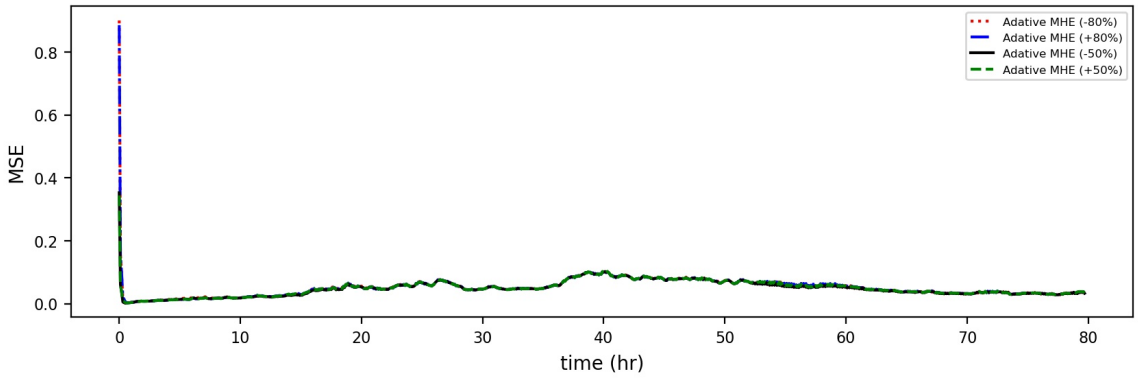


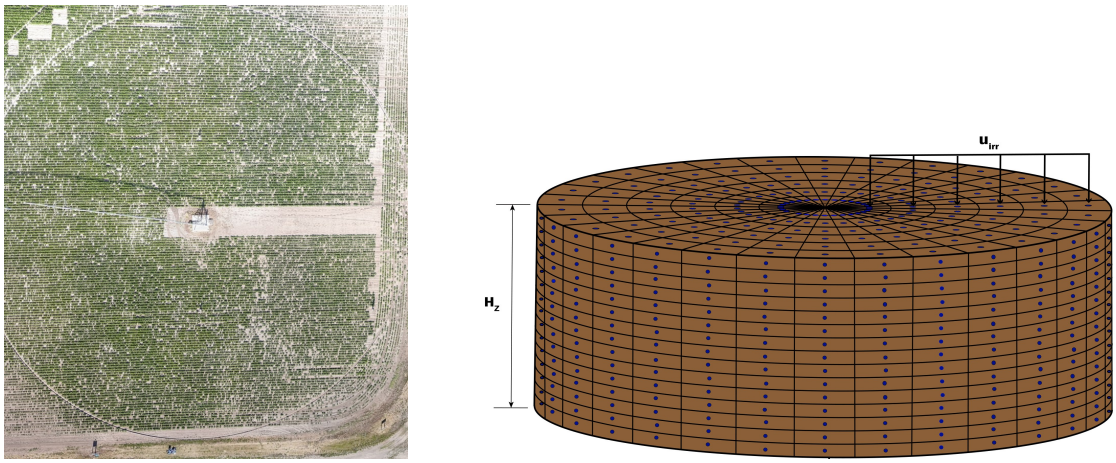
Figure 5.9: Mean square error of adaptive MHE staring from different initial guesses

5.5.2 Robustness of adaptive MHE

More simulations are carried out to check the robustness of the proposed adaptive MHE. Different initial guesses to the adaptive MHE are provided in this simulation, varying from $+80\%, +50\%, -50\%, -80\%$ of the actual initial condition. Figure 5.9 shows the MSE of all 4 cases. We can observe that at the beginning, the $+80\%, -80\%$ cases have more error than the $+50\%, -50\%$ cases, and after few time steps, all the cases converge to nearly same values. This result shows the robustness of the adaptive MHE.

5.6 Application to a real-agricultural field

In this section, the proposed procedure and algorithms are applied to perform model reduction and state estimation to a real-agricultural field. The agricultural field is



(a) **(b)**

Figure 5.10: a) Demo farm in Lethbridge, (b) A schematic diagram of the demo farm model

located east of the city Lethbridge, Alberta. The details of the experimental setup is discussed in Appendix A.

For this chapter, the data from July 15^{th}, 2019 is considered and at that period, the crop was sugar beet at its development stage. The crop coefficient (K_c) value 0.35 is used, and the reference evapotranspiration is obtained from the weather station to calculate the total evapotranspiration by using Equation (2.10). In this chapter, the cylindrical agro-hydrological model of the farm is constructed using finite difference discretization of Richards equation as discussed in Chapter 2. For the model, the depth till 0.5 m is considered. The model is discretized into 31500 nodes with 21, 60, 25 nodes in the radial, azimuthal, and axial directions, respectively. The demo farm and the schematic diagram of the demo farm is shown in Figure 5.10.

The initial condition of the field is obtained from the thermal and optical remote sensing images. First, the surface soil moisture is estimated from the thermal and Normalized difference vegetation index (NDVI) data. The k-means clustering approach is applied to find out relative wet and dry areas of the field, and the clustered soil moisture is used as an initial condition of the demo farm. In this chapter, it is assumed the initial condition of the surface is the same in the z-direction till 50cm. Figure 5.11 shows the clustered surface soil moisture. We consider 20 sensors at the depth of 48 cm and the position of the nodes are 1526, 3101, 4676, 6251, 7826, 9401, 10976, 12551, 14126, 15701, 17276, 18851, 20426, 22001, 23576, 25151, 26726, 28301,

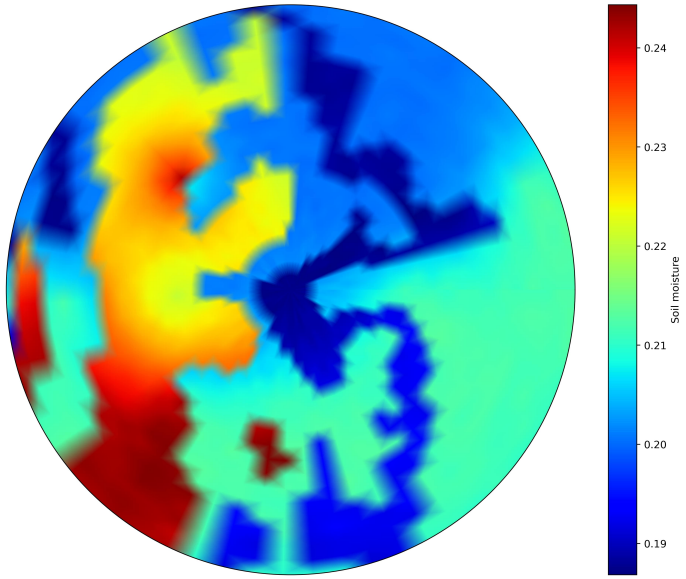


Figure 5.11: Surface soil moisture initial condition

29876, 31451.

In the next subsections, the proposed procedure and algorithms are implemented on the real field model to perform adaptive model reduction and state estimation based on adaptive MHE under different scenarios.

5.6.1 Results: adaptive model reduction

In this subsection, the proposed adaptive model reduction algorithm developed in Section 5.3 is applied to the real field. The main propose is to verify performance of the adaptive reduced order model for the large field. The initial condition, soil types and size of the field is considered as described above. For the simulation, the process noise is added. The process noise is normal distributed with zero mean and standard deviation of 6×10^{-3} m. The irrigation amount is 0.72 mm/hr, and the total simulation time is 24 hr.

Figure 5.12 shows the number of clusters each time after model reduction. It can be observed that based on the system dynamics, the number of clusters change each time. Figure 5.13 shows the state trajectories of the actual system and the adaptive reduced-order system. We can observe that the state trajectories of the actual system have an excellent agreement with the actual system trajectories. For further analysis of the performance, the MSE of the estimation error is considered and shown in Figure

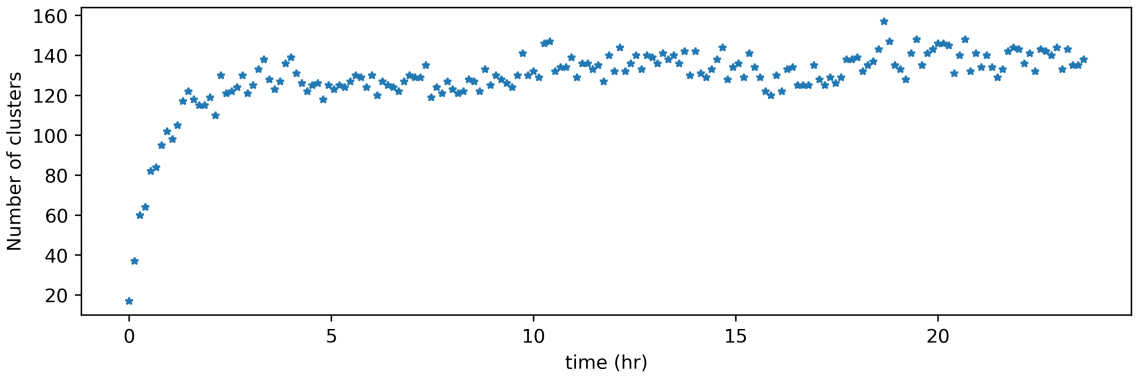


Figure 5.12: Number of clusters for adaptive model reduction

5.14. It is evident that the MSE value is very low in the range of 1e-05, and thus the adaptive model reduction can track the actual system's trajectories accurately in presence of the process noise.

5.6.2 Result: adaptive MHE

In this section, we apply the proposed adaptive MHE algorithms to the real field. The performance of the proposed adaptive MHE is demonstrated under the following two scenarios: (1) Scenario 1: field model without process and measurement noise; (2) Scenario 2: field model in presence of process and measurement noise.

Scenario 1: nominal case

In this scenario, we consider the field model without process and measurement noise. The adaptive moving horizon estimation method is applied to estimate the states of the field model. It is assumed that the actual process of the system is known, but the estimator only knows the actual measurement. The initial guess to the adaptive MHE is $\bar{x}_0 = 0.5 \times x_0$, where x_0 is the initial condition of the actual system and $\bar{x}_0$ is the initial condition of the adaptive MHE. The total simulation time is 24 hours. The Q and R values are considered as an identity matrix with diagonal elements as $Q = 2.5e-05 \times I_{Nx}$ and $R = 0.01 \times I_{Ny}$ respectively. The P matrix is designed based on Equation 5.9 using the lower and upper bound of the states. The irrigation amount is considered as 0.72 mm/hr. Figure 5.15 shows the number of clusters at each time interval and it shows that around 20-130 numbers of states are required each

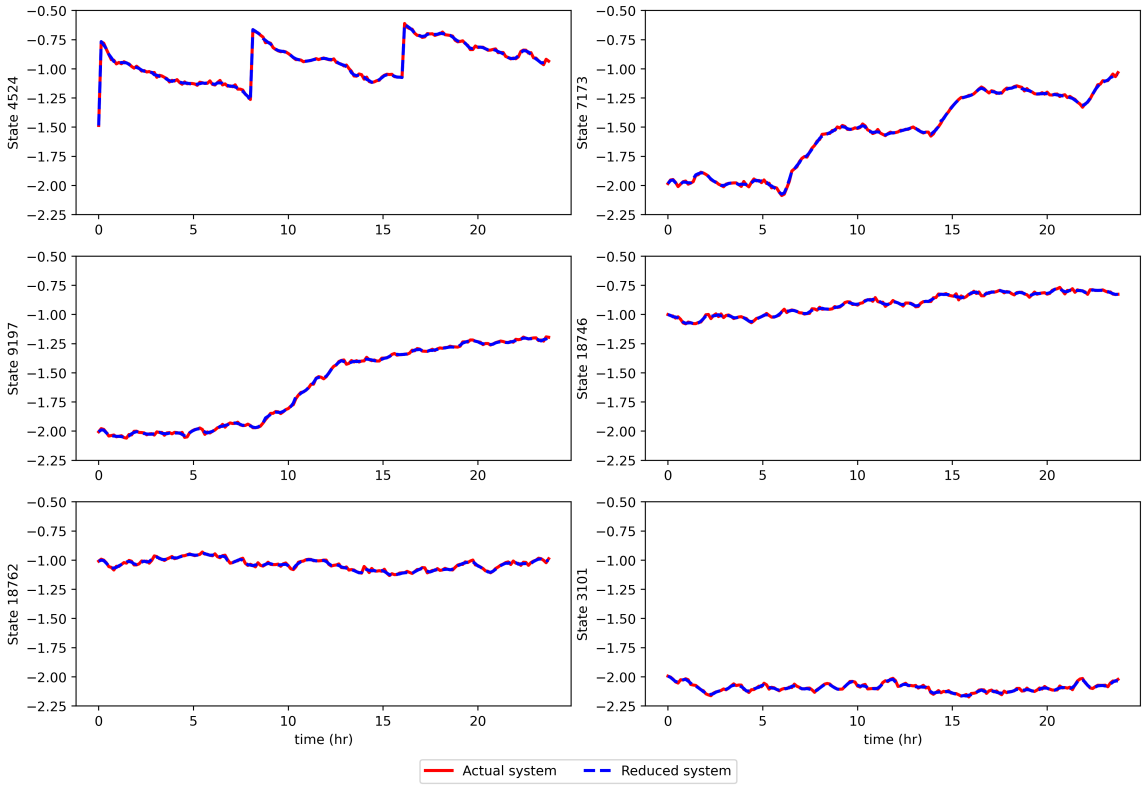


Figure 5.13: Selected state trajectories of the actual states (red solid line), reduced states (blue dotted line)

time to represent the system dynamics. Figure 5.16 presents the trajectories of some selected states of the actual states and the estimated states. It can be observed that the proposed adaptive moving horizon estimation can track the actual system states accurately. For further analysis, Figure 5.17 shows the MSE between the adaptive MHE and actual system. It can be observed that the MSE converges to a low value after few hours of simulation, which shows the superiority of the proposed adaptive MHE. Next, the surface soil pressure head for the actual system states, estimated system states, and absolute estimation error at time 1 hr and 24 hr are presented in Figure 5.18. From Figure 5.18(b), it can be observed that the absolute estimation error at time 24 hr is very much lower than the absolute error at time 1 hr (Figure 5.18(a)). In Figure 5.19, the soil moisture map for the actual system, estimated system states, and absolute estimation error at a depth of 0.5 m is presented. Similar to the surface layer, the estimation error decreases from time 1^{st} hr to 24^{th} hour.

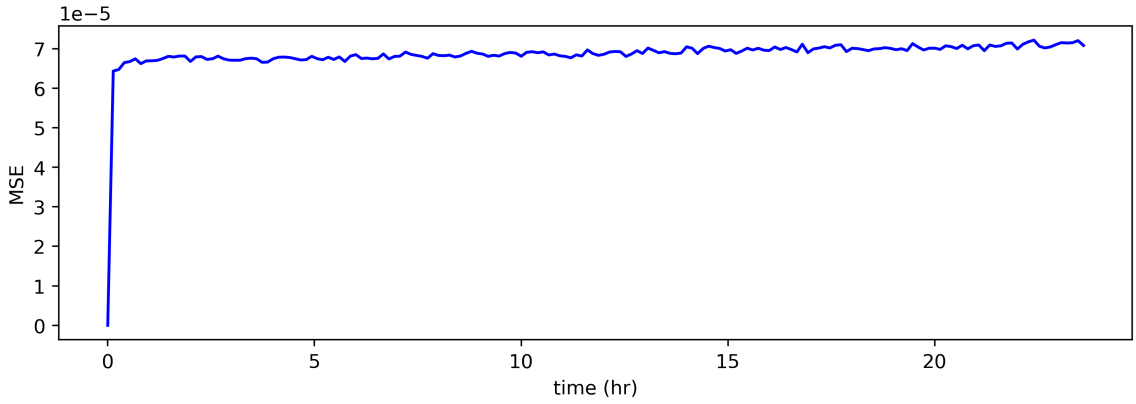


Figure 5.14: Trajectory mean square error of actual system and reduced system

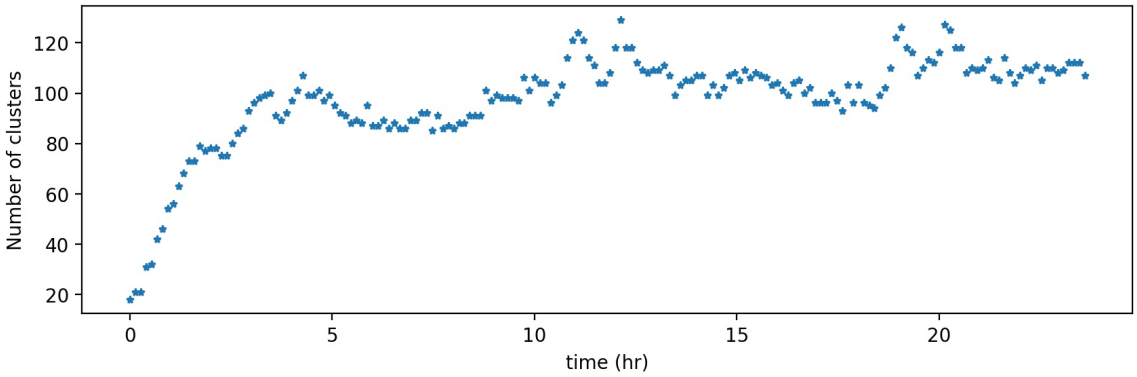


Figure 5.15: Number of clusters for scenario 1

5.6.3 Scenario 2: noisy case

In this scenario, the performance of the adaptive MHE is discussed in the presence of process and measurement noise. The total simulation time is 24 hours, and the initial guess is considered 0.5 times the actual soil moisture. The same P, Q, R values are used as the nominal case. The irrigation amount is 0.72 mm/hr. The process noise and measurement noise are considered normally distributed noise with zero mean and standard deviation of 6×10^{-3} m and 4×10^{-3} m.

The number of clusters generated for the adaptive MHE is shown in Figure 5.20. Things to note that the tolerance used to generate the clusters is higher than that of the process noise so that the noise does not affect the number of clusters. Figure 5.22 shows the trajectories of actual states and estimated states for few selected states. The results show that the estimated states can track the actual trajectory in the

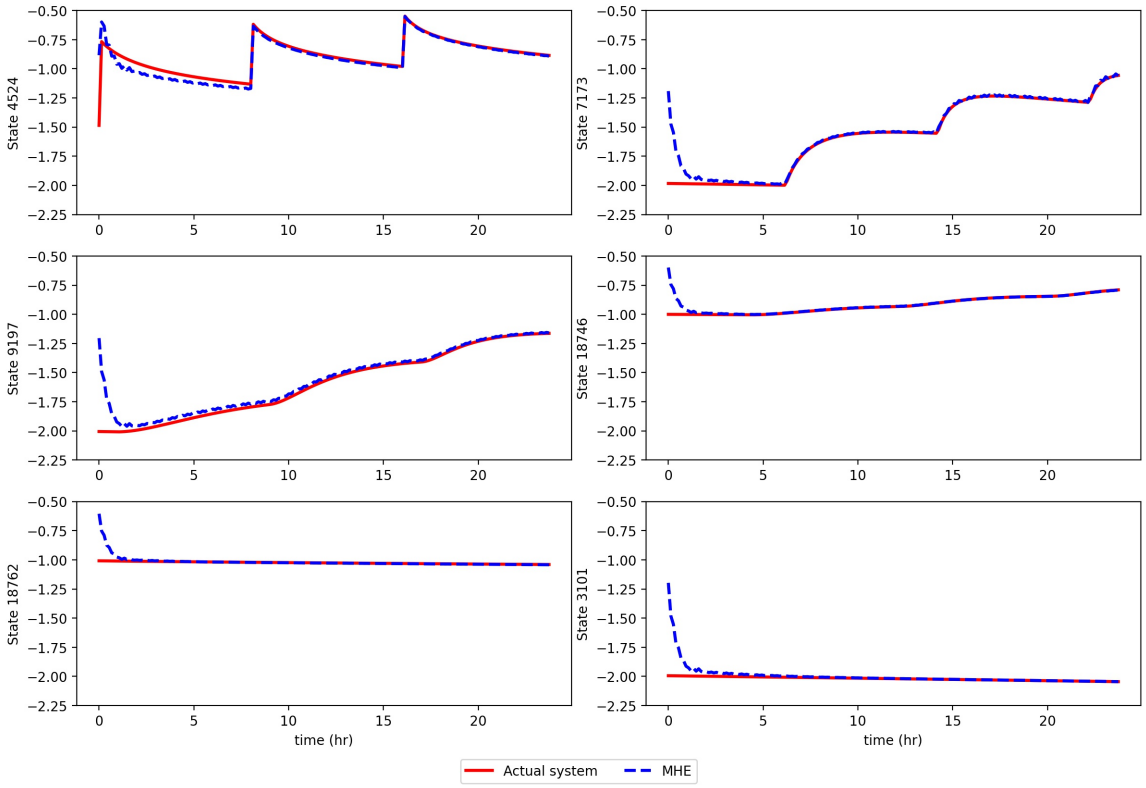


Figure 5.16: Selected state trajectories of the actual states (red solid line), estimated states (blue dotted line) for scenario 1

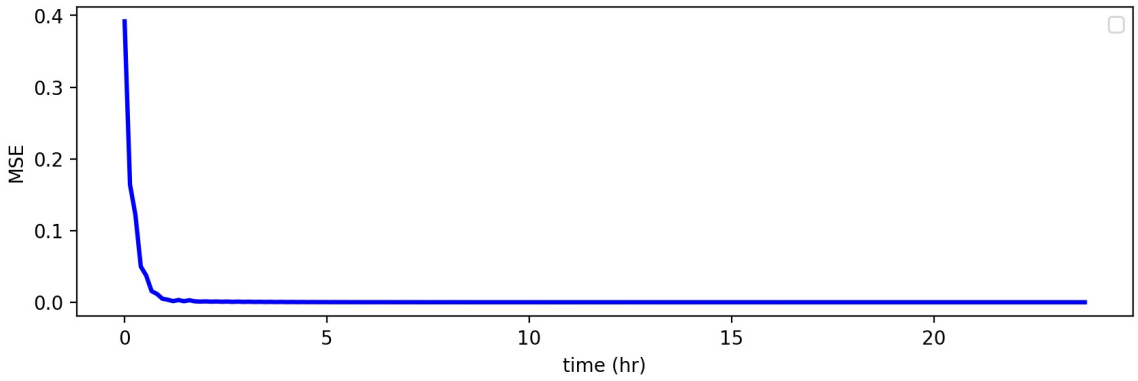


Figure 5.17: Trajectory mean square error of adaptive MHE for scenario 1

presence of noise. The MSE of estimation error is also presented in Figure 5.21. The results show that the estimation error converges to a very low value within a few hours of simulation.

For further analysis, the actual system states, estimated system states, and absolute estimation error for the surface nodes at time 1 hr and 24 hr are presented

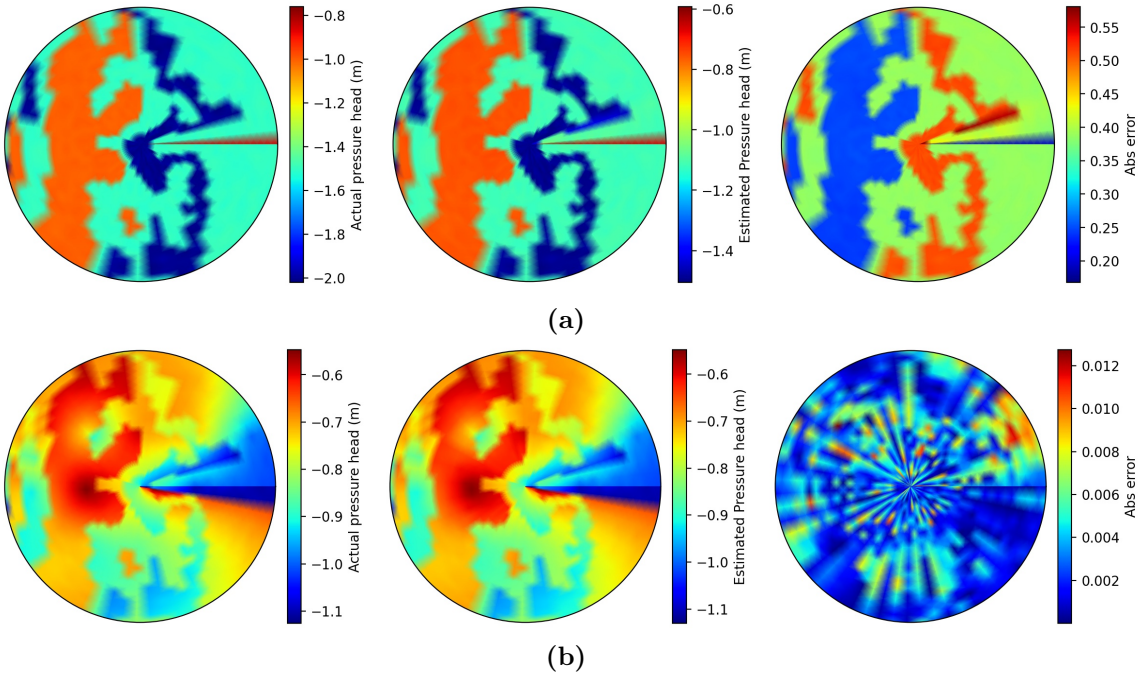
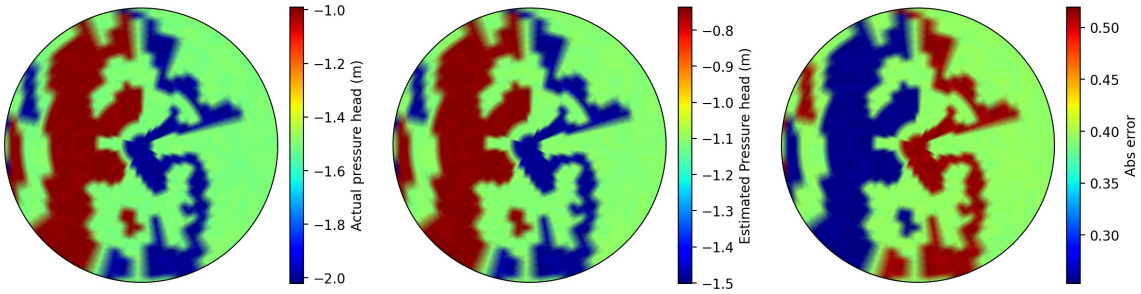


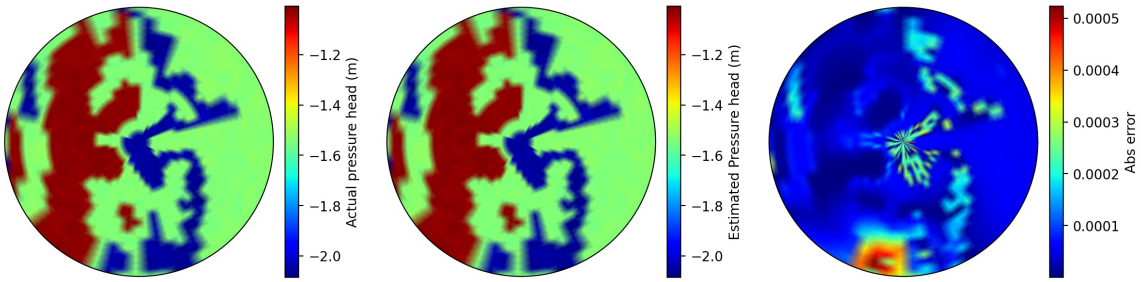
Figure 5.18: (a) Surface soil moisture map for actual states, estimated states and absolute estimation error at time $t = 1$ hr (Left to right) (b) Surface soil moisture map for actual states, estimated states and absolute estimation error at time $t = 24$ hr (Left to right) (scenario 1)

in Figure 5.23. The figure shows that the absolute error at time 24 hr is less than the absolute error at time 1 hr. Similarly, in Figure 5.24, the soil moisture map for the actual system states, estimated system state, and absolute estimation error at the bottom layer are presented. We can also see a similar trend as the surface layer where the estimation error at 24^{th} hr is lesser than the 1^{st} hour of simulation.

The above simulations were performed in a desktop computer with Intel i7-6700 CPU at 3.4GHz and 16.0GB RAM. The average time needed to evaluate the adaptive MHE (including clustering, model reduction and MHE optimization) every sampling time is about 90s. Note that a regular MHE based on the original nonlinear model with 31500 discretization nodes is very computationally challenging to implement. The size of the RAM of the above desktop computer was not sufficient in running the regular MHE.



(a)



(b)

Figure 5.19: (a) Bottom soil moisture map for actual states, estimated states and absolute estimation error at time $t = 1$ hr (Left to right) (b) Bottom soil moisture map for actual states, estimated states and absolute estimation error at time $t = 24$ hr (Left to right) (scenario 1)

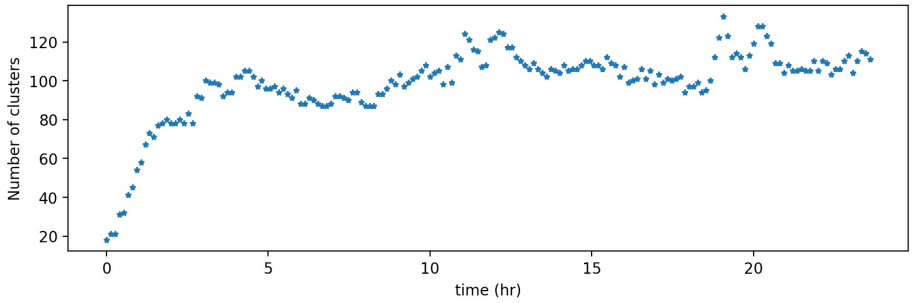


Figure 5.20: Number of clusters for scenario 2

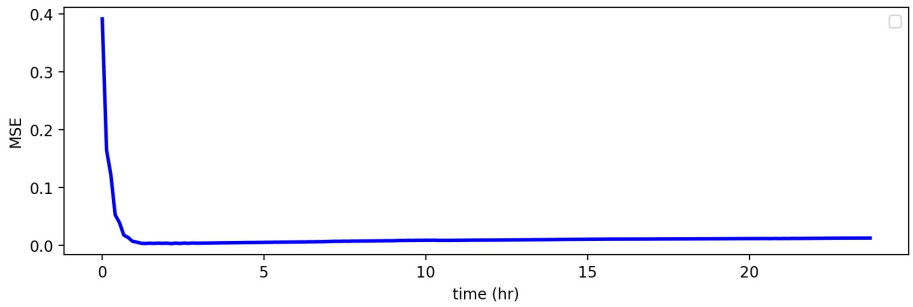


Figure 5.21: Trajectory mean square error for adaptive MHE for scenario 2

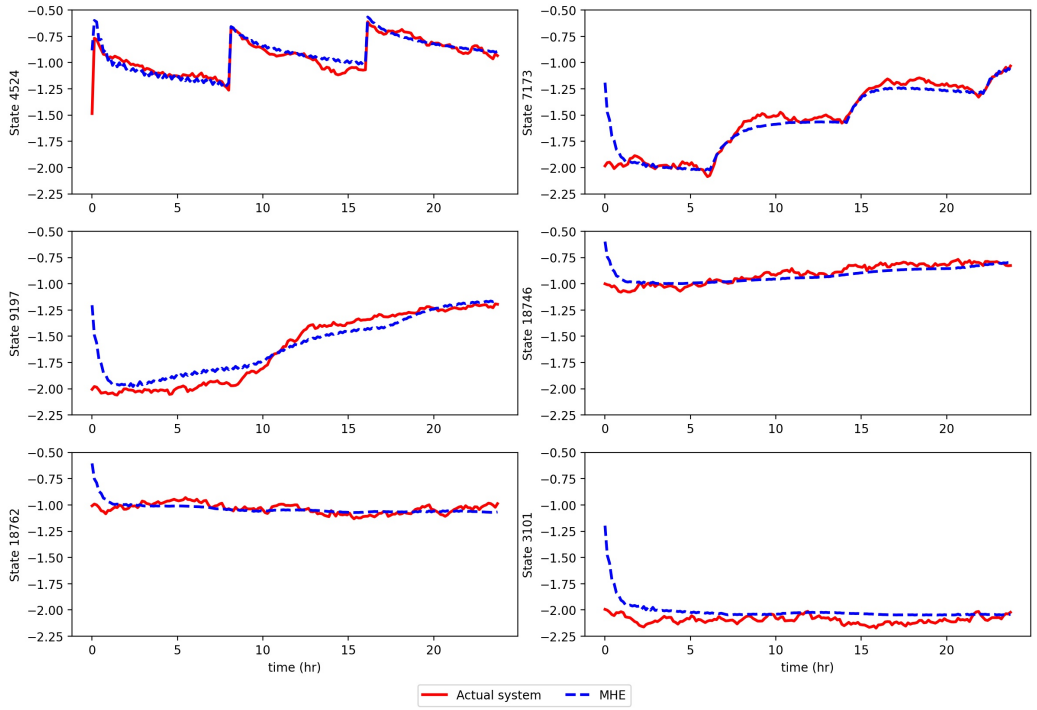


Figure 5.22: Selected state trajectories of the actual states (red solid line), estimated states (blue dotted line) for scenario 2

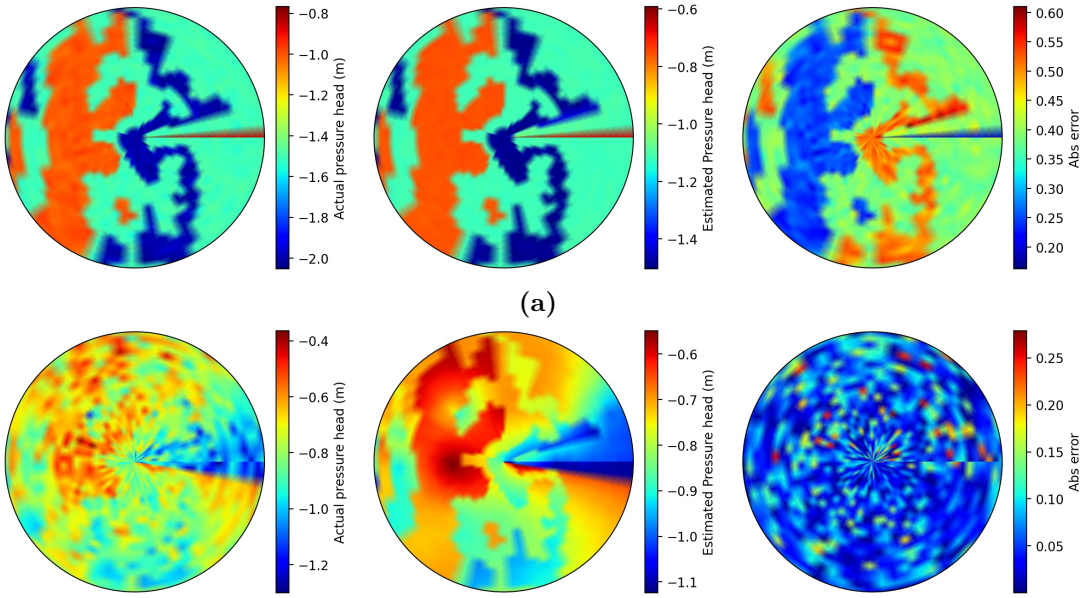


Figure 5.23: (a) Surface soil moisture map for actual states, estimated states and absolute estimation error at time $t = 1$ hr (Left to right) (b) Surface soil moisture map for actual states, estimated states and absolute estimation error at time $t = 24$ hr (Left to right) (scenario 2)

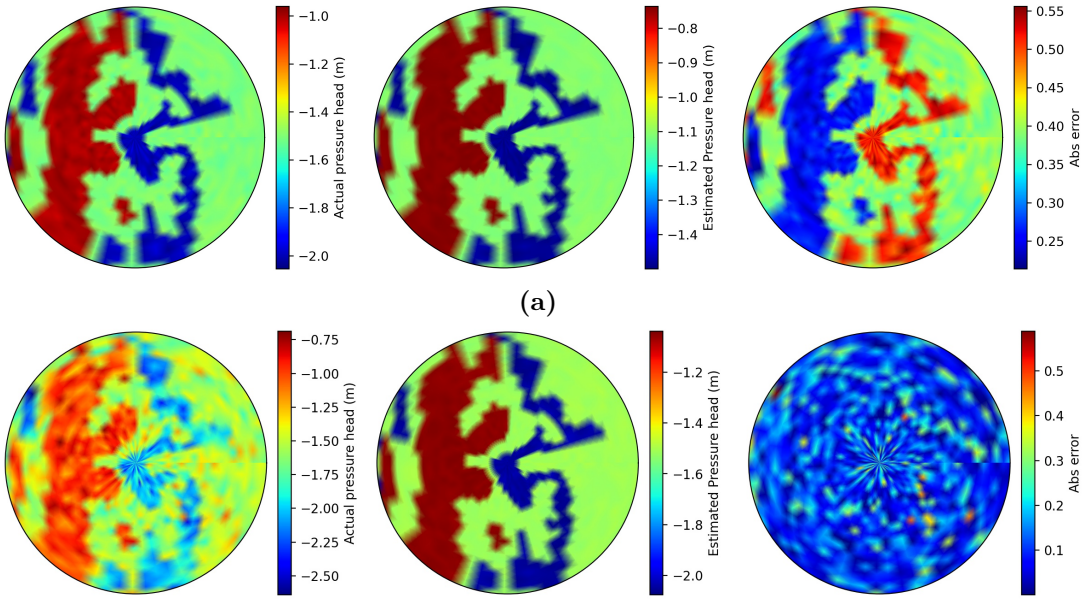


Figure 5.24: (a) Bottom soil moisture map for actual states, estimated states and absolute estimation error at time $t = 1$ hr (Left to right) (b) Bottom soil moisture map for actual states, estimated states and absolute estimation error at time $t = 24$ hr (Left to right) (scenario 2)

5.7 Conclusions

In this chapter, the state estimation for the high-dimensional agro-hydrological system was addressed. The structure-preserving adaptive model reduction algorithm was proposed using the nonlinear system trajectories and hierarchical clustering methods. The adaptive MHE was developed using the information exchange between different reduced models based on the adaptive reduced model. The performance of the adaptive MHE was compared with the actual MHE for a small simulated field. The result showed that the adaptive MHE has a better performance than the original MHE. The proposed approach was applied to a real-agricultural field using two different simulation scenarios located in Lethbridge, Canada, and the state estimation showed a satisfactory result.

Chapter 6

Knowledge based optimal irrigation scheduling

6.1 Introduction

The primary objective of the chapter is to study closed-loop scheduling in agricultural management. In this chapter, an optimization-based scheduler design that provides the optimal time for irrigation events and the corresponding optimal amount of irrigation is proposed. The objective of the scheduler is to maximize the crop yield while minimizing the water amount of irrigation.

First, the model reduction used in this chapter has a similar idea as the adaptive model reduction discussed in Chapter 5. However, for the scheduler application, only one global reduced model is used for the whole time period. The main reason to use one global reduced model is that both the input and time are unknown for the proposed scheduler approach, which makes it difficult to determine the operating regions and boundaries. Though using one global reduced model may increase the number of reduced nodes than the adaptive model reduction, still, the number of reduced states are much smaller than the original system states. Additionally, for state estimation, the states are the main decision variables, but for scheduler design, the inputs are primary decision variables, so the increase in the number of reduced state variables has less effect on optimization than the state estimation.

Based on the reduced-order model, the proposed closed-loop scheduling is built and investigated. In the design of the scheduler, empirical knowledge from the farmer is considered to reduce the computational complexity of the scheduler. The proposed

scheduler is designed in the framework of model predictive control. In this framework, both irrigation time and irrigation water are considered as the decision variables. The effectiveness of the designed scheduler is thoroughly investigated using four different scenarios.

6.1.1 Problem formulation

As described in Chapter 2, the input u is only present at the top surface nodes. When we consider agricultural fields equipped with central pivot irrigation system, the central pivot is not able to provide water to every plant at the same time. There is a rotational time gap for plant in receiving water. Therefore, at any instance, only the portion of the field under the central pivot has the access to irrigation water. Thus, this imposes a time-varying constraint on u as follows:

$$\mathscr{U}_{lb}(t) \leq u(t) \leq \mathscr{U}_{ub}(t) \tag{6.1}$$

The elements in U_{ub} which align to the central pivot may not be zero and other elements are zero.

A continuous time state-space model with measurements and disturbances is considered as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), d(t)) \\ y(t) &= Cx(t) \end{aligned} \tag{6.2}$$

where $y(t) \in \mathbb{R}^{N_y}$ denotes the soil pressure head measurements, $d(t) \in \mathbb{R}^{N_d}$ is the weather disturbances.

The objective is to calculate the optimal time and irrigation amount for maximum crop yield and water conservation for three-dimensional fields model with a central pivot irrigation system.

6.2 Model reduction

For a scheduler design, the optimization cost increases if the states are decision variables or necessary state constraints are present for the system. The model reduction can deal with these issues. One global reduced model is constructed for the whole scheduler design. Similar to the idea of trajectories clustering in the adaptive model

reduction in Chapter 5, the reduced model is created. So for the adaptive model reduction method, if we just consider that the operating region consists from the initial to the final time, then we don't need to switch the models at different operating regions, and one global reduced model is obtained.

The adaptive reduced model is expressed as:

$$\dot{\xi}(t) = f_r(\xi(t), u(t)) \tag{6.3}$$

where $f_r(\xi(t), u(t)) = \mathscr{U}^T f(\mathscr{U}\xi(t), u(t))$ and $\xi(t) = \mathscr{U}^T x(t)$. Note that the actual state $x(t)$ can be approximated based on mapping $\tilde{x}(t) = \mathscr{U}\xi$. The discrete model of (6.3) is expressed as follows:

$$\xi(k+1) = f_{rd}(\xi(k), u(k)) \tag{6.4}$$

6.3 Proposed closed-loop scheduling

This section proposes closed-loop scheduling to calculate the time intervals between irrigation events and the water amount for each event. The primary objective is to maximize the crop yield while reducing the total water use and equipment operating cost. The scheduler considers historical weather and weekly weather forecast, and soil moisture measurements. The main idea is to irrigate the field and calculate the time required to reach the lower stress-free zone. This is inspired by the practice of the farmers.

In this chapter, the iterative finite horizon optimization is considered like the classical MPC. However, the length of the horizon is not fixed because the time is also a decision variable in the optimization problem. Since the weather forecast over the horizon may not be accurate, the receding horizon strategy is implemented to handle the uncertainty in the weather forecast and in the modeling. The optimization problem is resolved after a few days interval with a more accurate weather forecast and recent measurements.

Each horizon consists of three separate segments. In the first segment, we irrigate the field, and the amount of water to be irrigated is the primary decision variable. In the second segment, the time is a decision variable that calculates the time for the next irrigation event. The time is calculated such that the plants will not experience

stress, and the yield will be maximized. The third segment calculates the irrigation amount for the next horizon. The third segment is added to the optimization problem to give the optimizer some flexibility to see a few more days further into the future to increase the performance and robustness of the scheduler. In all three segments, the yield and the constraints are considered to keep the pressure head within a stress-free zone. The slack variables are introduced to relax the target zone.

All three segments are optimized simultaneously. The reason to optimize everything together is that the amount of irrigation in the first and third segments will affect the total time in the second segment. The scenario may arise where we irrigate enough, but the total time for the next irrigation does not change much. This can be illustrated using a simple example. Let us consider the central pivot irrigation system. The irrigation amount is changed gradually from 0.6×10^{-6} m/s to 2.8×10^{-6} m/s as shown in Figure 6.1. First, we irrigate the system and calculate the time when it reaches the lower zone. From Figure 6.1, we can observe that with irrigation amount 0.6×10^{-6} m/s, the number of days it reaches the lower zone in 23 days, and with irrigation amount 2.8×10^{-6} m/s, it takes around 61 days. It can also be observed that after 1.6×10^{-6} m/s irrigation amount, the number of days to reach the lower zone does not change much. Even if putting double the amount of water, the next irrigation days do not change significantly. So if we optimize both the irrigation amount and the time together, we can save water and optimize the equipment functioning cost.

For each horizon, the optimization problem is formulated as follows:

$$\min_{\substack{u(j),\overline{\epsilon_r}(j),\\ \underline{\epsilon_r}(j),T}} Q_y\left(1-\frac{Y_a}{Y_p}\right)^2 + Q_u\sum_{j=k}^{N-1} u(j) - Q_T\sum_{j=k+N_1}^{N_1+N_2}\frac{T}{T_{ub}} + \sum_{j=k+1}^{k+N}(\overline{Q_r}\overline{\epsilon_r}(j)^2 + \underline{Q_r}\ \underline{\epsilon_r}(j)^2) \tag{6.5a}$$

$$\text{s.t. } \left(1-\frac{Y_a}{Y_p}\right) = \sum_{j=k+1}^{k+N} K_y(j)(1-K_s(y_r)) \tag{6.5b}$$

$$\tilde{\xi}(j+1) = \mathscr{U}^T f(\mathscr{U}\tilde{\xi}(j), u(j), d, \Delta T(j)), \quad j = k, ..., N-1 \tag{6.5c}$$

$$y_r(j) = C_r\tilde{\xi}(j) \tag{6.5d}$$

$$u_{lb} < u(j) < u_{ub},\ j = k, ..., k+N_1\ \&\ k+N_1+N_2, ...k+N \tag{6.5e}$$

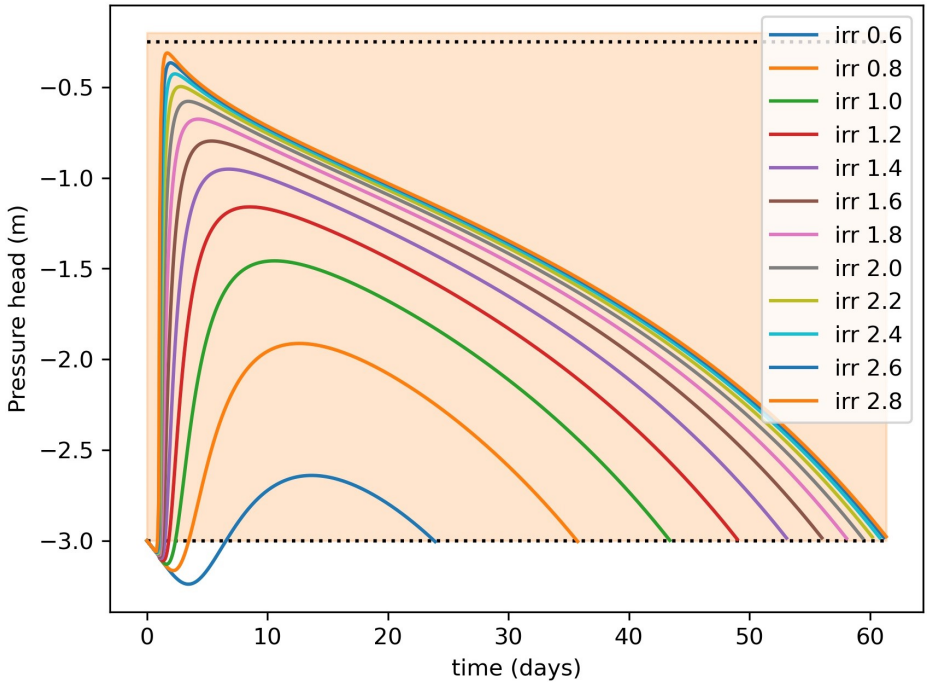


Figure 6.1: Motivation of optimizing both input and time together (irr represents the irrigation amount and the units are in $\times 10^{-6}$)

$$u(j) = 0, \quad j = k + N1, ..., k + N1 + N2 \tag{6.5f}$$

$$\tilde{\xi}(j) \in \mathbb{Z}, \quad j = k, ..., k + N - 1 \tag{6.5g}$$

$$\underline{V} - \underline{\epsilon_r}(j) < y_r(j) < \bar{V} + \bar{\epsilon_r}(j) \tag{6.5h}$$

$$\bar{\epsilon_r}(j) \geq 0, \underline{\epsilon_r}(j) \geq 0 \tag{6.5i}$$

$$T_{lb} < T < T_{ub} \tag{6.5j}$$

$$N_1 + N_2 + N_3 = N \tag{6.5k}$$

where (6.5a) defines the cost function to be minimized and the input (u), time (T) and slack variables ($\overline{\epsilon_r}, \underline{\epsilon_r}$) are the decision variables. In (6.5a), the first term is the crop yield deficiency cost, the second term considers the irrigation water cost, the third term denotes the normalized time cost which is active only in the second segment. The last term in (6.5a) is the cost term of non-negative slack variables ($\overline{\epsilon_r}, \underline{\epsilon_r}$) which is introduced to relax the bounds of target zones $\bar{V}, \underline{V}$ in (6.5h). $Q_y, Q_u, Q_T, \underline{Q_r}, \overline{Q_r}$ are the positive weighting factors. Equation (6.5b) is the model used to evaluate the yield deficiency. Equations (6.5c), (6.5d) represent the discrete time reduced-order model and the output function. $\tilde{\xi}(k)$ denotes current state estimates at time k. In this work,

we assume all the states can be measured. N_1 is the number of sampling time for first segment, N_2 is the number of sampling time for second segment ($\Delta T_2 = T/N_2$). Things to note that in segment 2, the time is unknown so the number of sampling points (N_2) is fixed and it is chosen based on upper bound of T_{ub} such that the model does not experience numerical issues. N_3 is the number of sampling time for third segment. Equation (6.5k) shows the total sampling time is N. Equation (6.5e) provides the bounds of input for first segment and third segment. Equation (6.5f) defines the input amount is zero for second segment. Equation (6.5h) imposes the zone constraints with the slack variables and Equation (6.5i) implies the slack variables are non-negative. Equation (6.5j) defines the constraints for the lower and upper bound of time.

As discussed before, the receding horizon strategy is implemented to handle the weather and modeling uncertainty and use the scheduler as a closed-loop system.

In general, the scheduler problem is solved using the mixed integer nonlinear programming (MINLP), where the daily irrigation decisions (yes/no) are designed as a binary variable. But the MINLP for a large-scale system is computationally expensive. In this work, the advantage of using time T as a decision variable is that the nonlinear programming (NLP) can be used instead of the MINLP because T does not have to be an integer and the number of sampling periods is fixed.

6.4 Results

In this section, the proposed algorithms are applied to demonstrate the performance of the reduced order model and the scheduler under different scenarios. A field of radius 50 m and depth 30 cm is considered. The 30 cm depth is selected because, for crops selected in this work, the maximum root-water extraction happens in between the depth of 30 cm. The field is equipped with central pivot irrigation system. The model of the farm is constructed using finite difference discretization of the Richards equation. The entire system is discretized into 1920 nodes with 5 in radial, 64 in azimuthal and 6 in axial direction. Each nodes corresponds to states of the system. The central pivot takes around 8 hours to irrigate the whole field. Different types of crops and soil types are considered in different scenarios.

Table 6.1: Soil properties of three different types of soil

Soil type	K_s (m/s)	θ_s (m^3/m^3)	θ_r (m^3/m^3)	α (1/m)	n (-)
Loam	2.889×10^{-6}	0.43	0.0780	3.6	1.56
Sandy clay loam	3.6388×10^{-6}	0.39	0.1	5.9	1.48
Clay loam	7.2223×10^{-7}	0.41	0.095	1.9	1.31

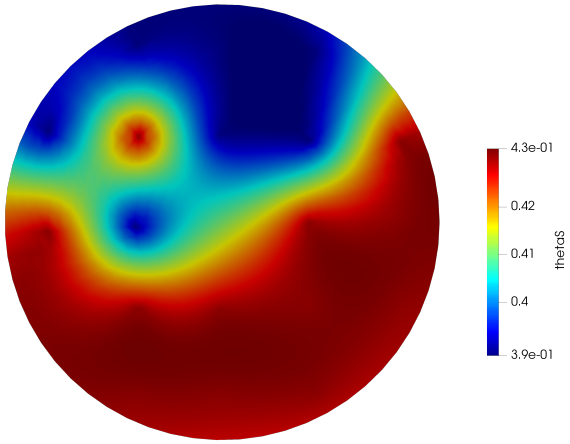


Figure 6.2: Soil parameter θ_s for the field

6.4.1 Results: model reduction

In this subsection, the proposed model reduction discussed in section 6.2 is applied to the system. First, the effect of reduced model order on the mean square error (MSE) of the reduced model is discussed. Then the robustness of the reduced model is discussed.

In this simulation, we consider the real soil properties of the field located at Lethbridge, Canada. In summer 2019, we collected soil samples at 20 points in the field and in the soil lab the soil types are estimated. We found three different soil types present in the field: loam, sandy clay loam, and clay loam. The hydraulic properties of the soil types are shown in Table 6.1. The kriging method is applied to get the soil properties for all other nodes of the field. Figure 6.2 shows one selected parameter (θ_s) of the surface nodes. The other parameters also follow the same trend.

Figure 6.3 shows the MSE values of the reduced model with the number of reduced states. The number of reduced states is obtained by changing the threshold values of the agglomerative hierarchical clustering method starting from 0.3 to 3.5 by increasing

0.2. As discussed in section 2, we can specify the threshold values in hierarchical clustering instead of the number of reduced models. The MSE values are calculated between the actual model and the reduced-order model. From Figure 6.3 it can be observed that after 56 reduced states, the error values are minimal. For this simulation, we consider 30 reduced states.

Next, we consider the robustness of the reduced model. In optimization-based controller design, the input amount is one of the important decision variables. So the reduced model should be robust enough to handle different input amounts. First, the projection matrix is calculated using the initial condition -4.0 m and input amount 2e-06 m/s. Then using the same projection matrix, we simulate the reduced-order model starting from different initial condition (-3.0 m) and different input amounts (1e-06 m/s, 0.5e-06 m/s, 0.1e-06 m/s, 0.05e-06 m/s). Figure 6.4 shows the state trajectories of the actual system and the reduced-order system randomly chosen state 77 (surface) and 1897 (depth 25 cm) for different input amounts. It can be observed that reduced order trajectories are very close to the actual order trajectories, which shows the robustness of the proposed model reduction method.

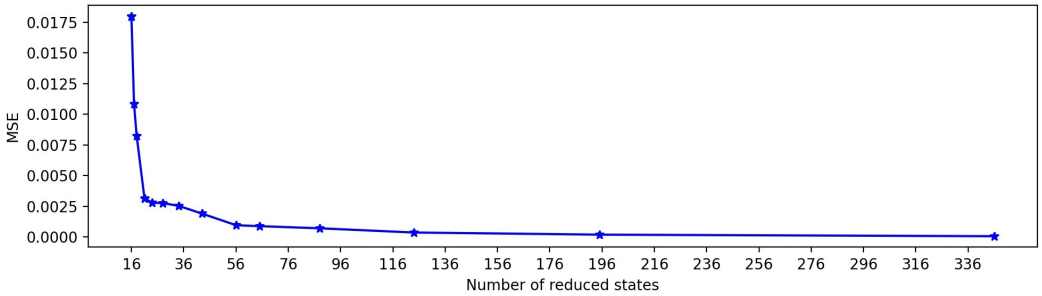


Figure 6.3: Values of MSE with different reduced order

6.4.2 Result: scheduler

In this subsection, the performance of the proposed scheduler design is demonstrated under the following three scenarios: (1) Scenario 1: uniform soil types, no rain, constant ET and grass; (2) Scenario 2: non-uniform soil types, no rain, constant ET and grass; (3) Scenario 3: uniform soil, lettuce crop type, variable ET and rain.

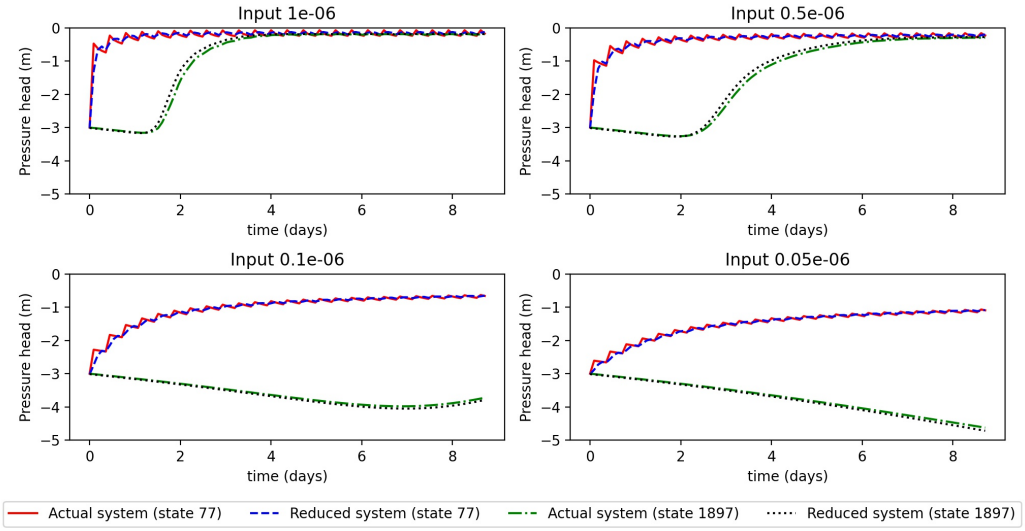


Figure 6.4: Selected state trajectories of the actual system and reduced system for four different inputs

Scenario 1

In this subsection, we consider a simple scenario of uniform soil type of loam. We further assumed that there are no disturbances present like rain and variable ET. We consider the crop as grass. The simple scenario is considered to show how the proposed scheduler algorithm works.

We consider the 2^{nd} layer from the top as the root zone of the grass, which is the output to the optimization problem (6.5d). The values of the tuning parameters $Q_y, Q_u, Q_T, \underline{Q_r}, \overline{Q_r}$ are 1, 1, 1, 100, 1 respectively. In this scenario, we put more weight on $\underline{Q_r}$ to make the system not violate the lower bound, which is more crucial for the zone control. As described in the previous sections, it is difficult to get back to normal again if the plant gets stressed once. The actual upper and lower bound of the system for the maximum yield and stress-free zone is -0.25 m and -3.1 m [109]. Things to note are that the stress-free zone depends upon different types of crop and ET values. To ensure the system does not experience the actual stress, a more conservative zone is considered with lower and upper zones -2.8 m and -1.0 m, respectively. So even if the system violates the more conservative zone, it does not experience the actual stress. The lower bound and upper bound of input are considered as 0 m/s and 4e-07 m/s, respectively. The lower bound of time is considered 30 min while the upper

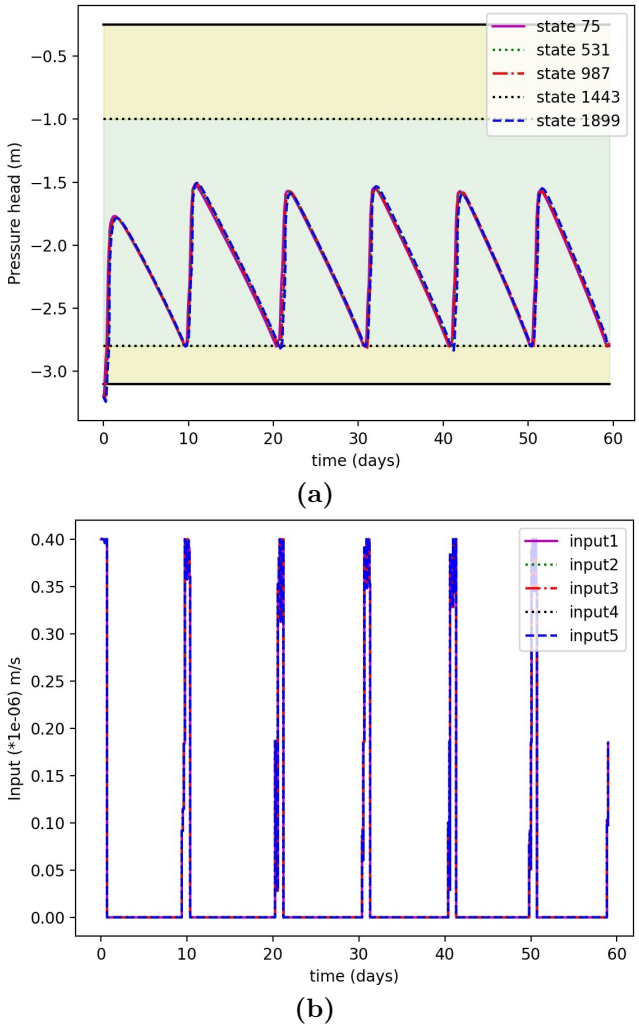


Figure 6.5: (a) Selected state trajectories under the proposed zone scheduler design for scenario 1; (b) Irrigation amount for 5 different sprinklers obtained from proposed zone scheduler

bound is kept as 16 days.

Figure 6.5(a) shows the states trajectories of five randomly selected states on 2^{nd} layer and Figure 6.5(b) shows the input amount. From Figure 6.5(a), we can observe that after it reaches the conservative lower zone, the system irrigates again. The root zone remains within the zone all the time so that the plants never get stressed. Figure 6.5(b) shows the input amount for all the five sprinklers. As the farm has a uniform soil type, all the sprinklers give nearly the same value. We can also observe that the irrigation event can be planned after nearly ten days interval.

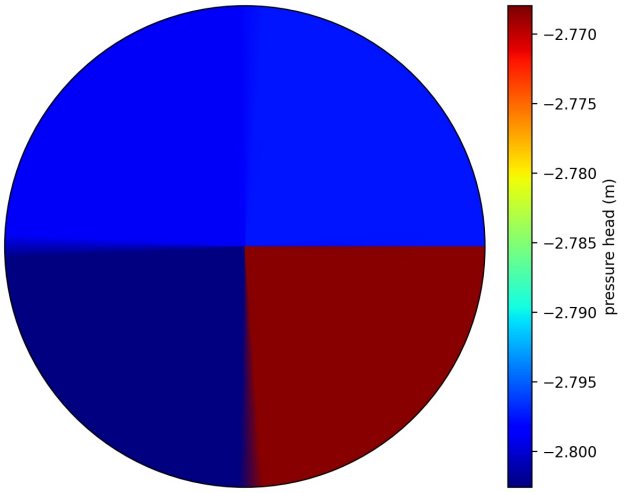


Figure 6.6: Pressure head values for root zone layer

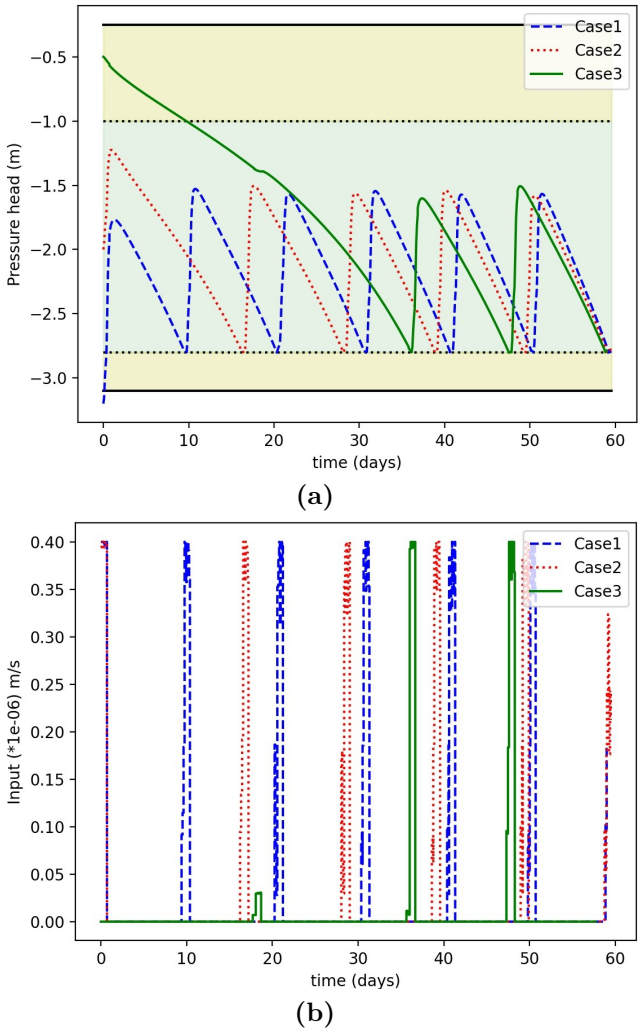


Figure 6.7: (a) Selected state trajectories for three cases (different initial conditions) (scenario 1); (b) Input trajectories for three cases

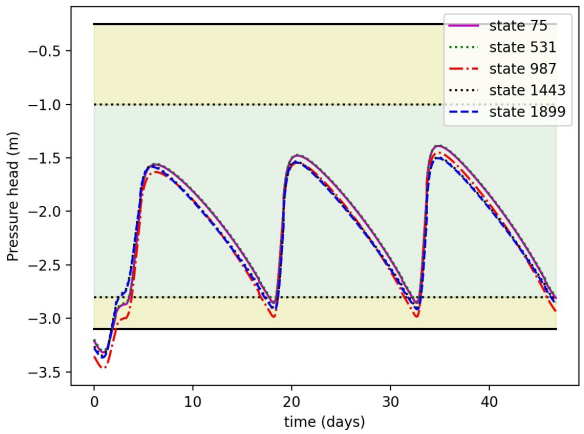
Figure 6.6 shows the 2^{nd} layer pressure head values. We can observe that all the nodes are around -2.8 m which is the conservative lower zone value. In all the quadrants, we see little bit different pressure head values because of the central pivot rotation.

Further, the robustness of the proposed scheduler is analyzed. We consider three different cases based on initial conditions (-0.5 m, -3.2 m, and -2.0 m). The initial conditions are chosen such that two initial conditions are outside of the zone and one is already inside the zone. Figure 6.7(a) presents the state trajectory of one randomly selected state 75 on the 2^{nd} layer. It can be observed that in all three cases, the scheduler works fine. Figure 6.7(a) shows the input profile for sprinkler 1 for all the three cases. We can see that in case 1 the irrigation amount is higher than the other two cases. This is because the initial condition is outside the lower zone and requires more and frequent irrigation to keep within the zone. In case 3, as the initial condition is already very wet, the scheduler prescribes very little water until it reaches the lower zone.

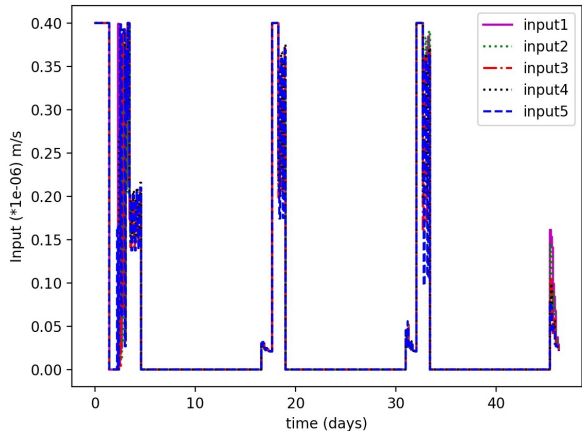
Scenario 2

In this scenario, we consider the non-uniform soil types, no rain, and constant ET. The motive is to show the efficiency of the scheduler in the presence of different soil types. The soil arrangement discussed in subsection 6.4.1 is considered in this scenario (Figure 6.2). We consider 3^{rd} layer from the top (10 cm depth) as the root zone of the crop-type grass. The scheduler tries to keep only the 3^{rd} layer inside the zone. The values of the upper and lower bounds of the actual and conservative zones are the same as in scenario 1. The lower and upper bounds of inputs are 0 m/s and 4e-07 m/s. The lower and upper bounds of the time are 30 mins and 12 days.

Figure 6.8(a) shows state trajectories of some selected states from the 3^{rd} layer. It shows that the states stay with in the zone and some of the states are little outside of the conservative zone. It may be caused because of the reduced model error and different soil types. All the states values for 3^{rd} layer at the end time is shown in Figure 6.9. We can observe that the all the pressure head values are above the lower zone value -3.1 m. Figure 6.8(b) shows the input trajectories of five different sprinklers obtained form the scheduler. As observed from the figure, the input amounts are not



(a)



(b)

Figure 6.8: (a) Selected state trajectories at 3^{rd} layer under the proposed zone scheduler design for scenario 2; (b) Irrigation amount for 5 different sprinklers obtained from proposed zone scheduler

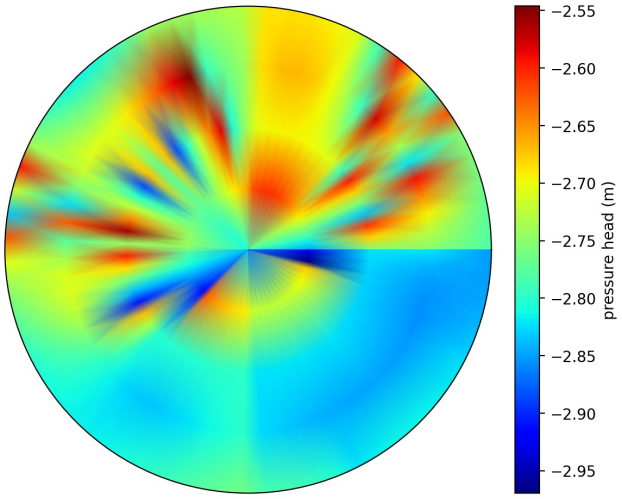


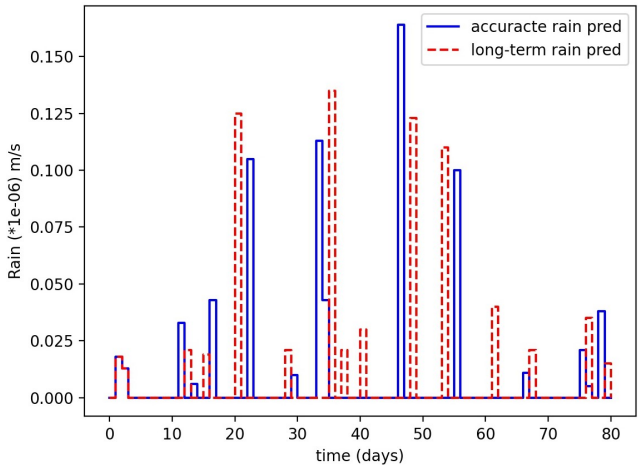
Figure 6.9: Pressure head values of depth 10cm for scenario 2

same for all the sprinklers. This shows for different soil types, the sprinklers may put different amount of water.

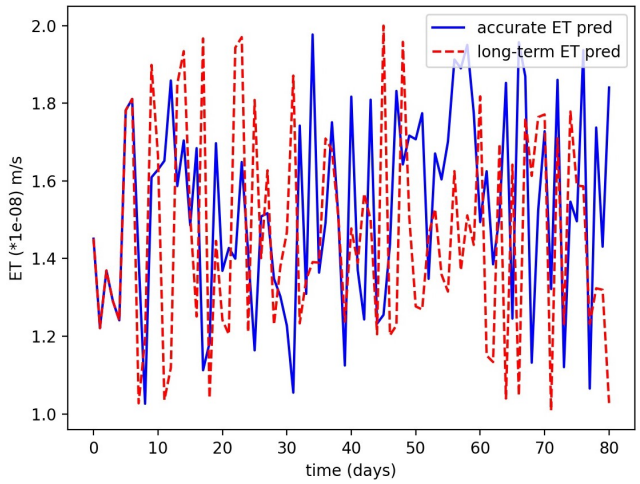
Scenario 3

In this scenario, the variable ET and rain uncertainties are considered. The uniform soil type of loam and crop type of lettuce is chosen for the simulation. This scenario is shown to check the efficacy of the scheduler in the presence of weather disturbances and crop growth stages. Total 85% root water is extracted from the top 30 cm soil. For the high yield and the crop not to experience any stress, keeping the top 30 cm in the stress-free zone is required. In this scenario, the objective of the scheduler is to keep all the layers in the zone. The values of the upper and lower bound for the actual zone are -0.25 m and -3.1 m. The upper and lower bound for the conservative zone is considered as -0.5 m and -2.3 m. The values of the tuning parameters $Q_y, Q_u, Q_T, \overline{Q_r}, \underline{Q_r}$ are 1, 100, 1, 100, 0.01, 1 respectively. Things to note are that the tuning parameter values can be adjusted depending on the root growth with time. The lower and upper bounds of the time are 30 mins and 12 days. The upper and lower bound of the input are 0 m/s and 4e-07 m/s. In general, seven days rain predictions can be 80% accurate. So for one horizon in the scheduler, the accurate weather prediction of 7 days is used, and for the rest of the days, the long-term prediction value is used. The values of accurate weather prediction and long-term weather prediction considered for this simulation are shown in Figure 6.10(a). Similarly, the ET value for accurate and long-term weather prediction is shown in Figure 6.10(b). The crop coefficient (K_c) for lettuce crop type for all the growing season is shown in 6.10(c). All growing season consists of the initial, crop development, mid-season, and late season.

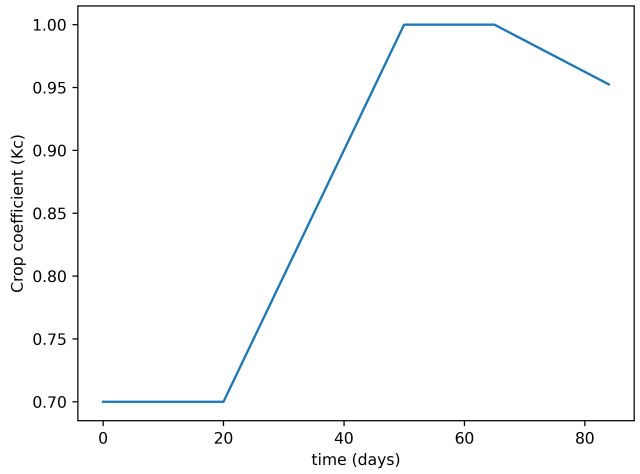
Figure 6.11(a) shows some selected state trajectories at different depths. We can observe that for all the depths, the states are in the stress-free zone. We can also see that around 50 days, state 60 (bottom node) goes outside of the conservative zone. This happens because around day 50, the crop is at the mature stage, and the ET values are high 6.10(c), and there is some delay between the water to reach the bottom node. That is why we choose a conservative zone of -2.3 m, So even if the bottom nodes go outside of the conservative zone, it may have less chance to go outside of the actual stress zone.



(a)



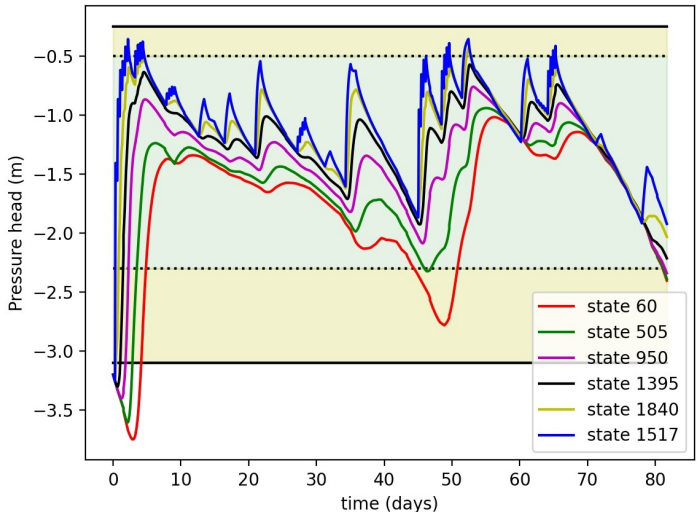
(b)



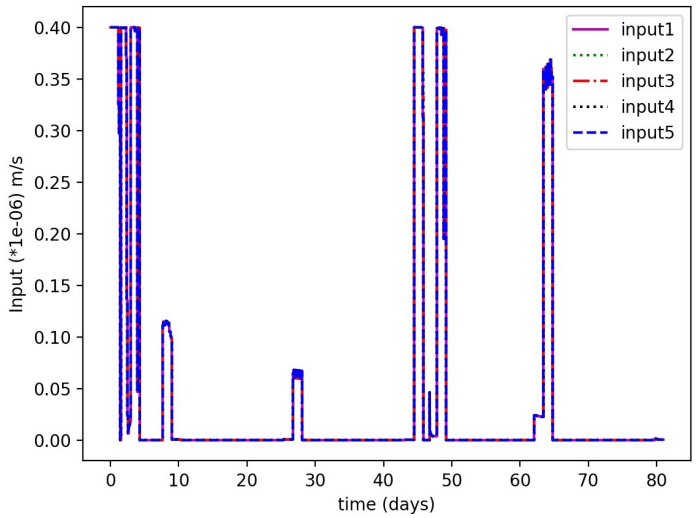
(c)

Figure 6.10: (a) Accurate rain forecast and long term forecast prediction; (b) Accurate ET forecast and long-term ET forecast, (c) Crop coefficient for total growing season for lettuce

Figure 6.11(b) shows the input amount for all five sprinklers. We can observe that there is frequent irrigation at the beginning because the states are outside of the zone. Moreover, around day 45-50, there is frequent irrigation because of high ET values, and the crop needs more water at that stage. For other days, because of rain and low ET values, the scheduler prescribes less frequent irrigation.



(a)



(b)

Figure 6.11: (a) Selected state trajectories for all layers under the proposed zone scheduler design for scenario 3; (b) Irrigation amount for 5 different sprinklers obtained from proposed scheduler for scenario 3

6.5 Conclusions

In this chapter, the closed-loop scheduler for a large-scale three-dimensional model was proposed. The scheduler was designed to provide optimal irrigation time and amount to ensure high yield, less water consumption, and less equipment functioning cost. The motivation to use both time and input as decision variables were discussed. The effectiveness of the proposed approach was shown using three different scenarios.

Chapter 7

Surface soil moisture remote sensing through Long Short-Term Memory (LSTM)

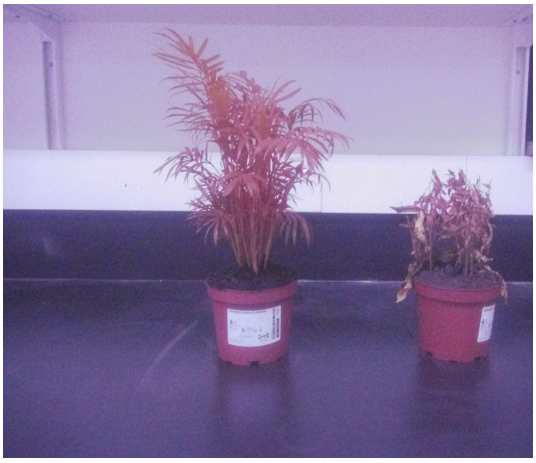
7.1 Introduction

Point soil moisture sensors provide soil moisture measurements for sparse point locations. It is expensive to deploy many point sensors to get soil moisture measurements of the entire field. Remote sensing provides an affordable and feasible way to obtain soil moisture measurements of the entire field. In this chapter, a machine learning based Long Short-Term Memory (LSTM) to estimate surface soil moisture using thermal and optical remote sensing images, weather conditions, and soil types as input to the system is proposed. The LSTM is a recurrent neural network (RNN) model which is mostly used for supervised learning of sequential data. First, the details of different vegetation indexes and model development are discussed. After that, the data prepossessing for the collected drone images from the 2019 demo-farm experiment is discussed. The data preprocessing step includes radiometric thermal image conversion, image stitching, image registration, and sensor location identification. Next the LSTM model is trained to provide the surface soil moisture. The efficacy of the proposed model is discussed using the actual data collected from the 2019 experiment in the result section.

7.2 Methods

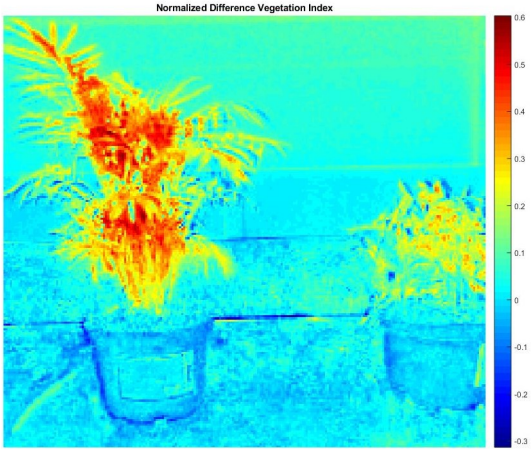
In this section, we discuss the different input indexes and model development for the soil moisture estimation.

7.2.1 Normalized Difference Vegetation Index (NDVI)

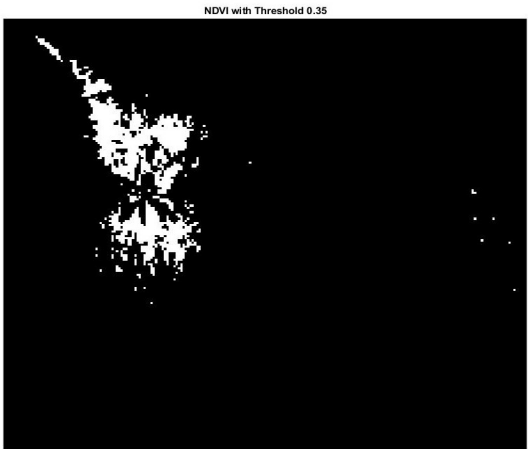


(a) Near infrared image

(b) RGB image



(c) NDVI



(d) NDVI with threshold 0.35

Figure 7.1: NDVI analysis of healthy and unhealthy plant

The NDVI is the most popular vegetation index to measure plant health. It can be measured by the amount of light that plant reflects at specific frequencies. The healthy plant leaves strongly reflect near-infrared light and strongly absorb red light, but when the plant is dehydrated and unhealthy, the spongy layer of the leaves starts

degrading, and the plant absorbs more near-infrared light. So NDVI can be calculated as follows:

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

where NIR is the near-infrared reflectivity and RED is the red reflectivity.

The NDVI value varies between -1 to 1. Where number between -1 to 0 represents dead plants or inanimate objects, 0 to 0.33 indicates the stressed plant, 0.33 to 0.66 is a moderately healthy plant, and 0.66 to 1 is a very healthy plant [110]. Experiments were conducted in the lab to verify the NDVI results. Figures 7.1a and 7.1b represent the near-infrared and RGB image of two plants, one is healthy and the other one is unhealthy. Figure 7.1c shows the NDVI analysis of the two plants. From Figure 7.1d, we can observe the healthy plant whose NDVI is more than 0.35. As the near-infrared image and visible image were captured in two different cameras, the image registration or image alignment method was required to align the two images. Here, the speeded up robust features (SURF) method was used to align the two images.

7.2.2 Temperature Vegetation Dryness Index (TVDI)

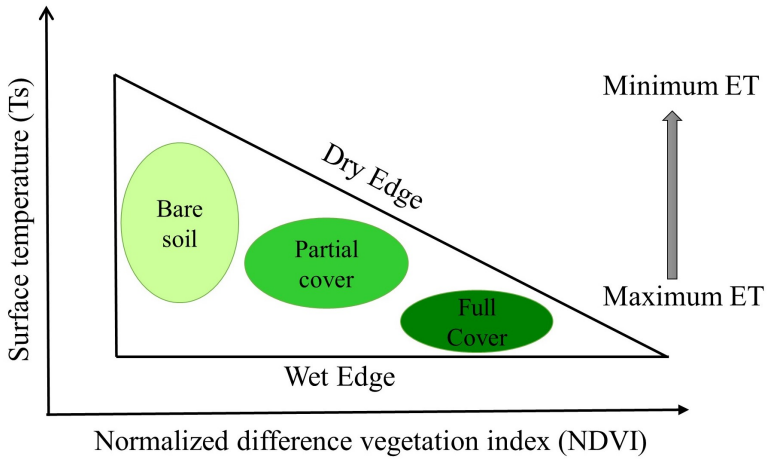


Figure 7.2: T_s-NDVI triangle feature space

The Temperature Vegetation Dryness Index (TVDI) combines the surface temperature and the vegetation index (NDVI). The mathematical expression for TVDI is:

$$TDVI = \frac{T_s - T_{s,min}}{T_{s,max} - T_{s,min}}$$

where T_s is the surface temperature, $T_{s,min}$ is the lowest surface temperature and $T_{s,max}$ is the maximum surface temperature which can be expressed as a function of NDVI ($T_{s,max} = a + b\ (NDVI)$), a, b are the coefficients of the dry edge. The T_s and $NDVI$ feature space can be plotted as a triangular area as shown in Figure 7.2. The dry edge is represented as the diagonal line of the triangle which depends linearly on the NDVI. The wet edge can be used as a constant bottom line of the triangle. The ET represents the evapotranspiration term. For full cover plant, the ET is maximum and for bare soil the ET is minimum. More details of the TVDI methods can be found in [88].

7.2.3 LSTM modeling

In this section, we briefly discuss the model approximation method long short-term memory recurrent neural networks (LSTM-RNNs).

RNNs are universal approximators, a class of nonlinear function which allows identification of dynamical systems in form of high dimensional nonlinear state space models. For sequential time series data, such as for chemical processes, LSTM-RNNs can capture a nonlinear dynamics much easier with a sequential data sets and no prior knowledge regarding the dynamics of the system. While the traditional RNNs suffers loosing error information pertaining to long sequences of data, LSTM is able to deal with this problem by protecting error information from decaying using learnable gates [111].

A brief step-by-step procedure for identifying a LSTM-RNN model is outlined as follows:

1. Normalise the data set from the original range so that all values are within the range of 0 and 1.

2. Reconstruct the dataset according to the predetermined input-output dimension

3. Select the number of LSTM layers, the number of nodes in each layer, the optimiser, the batch size and epochs for model fit. Note that there are no universal rules for determining these parameters.

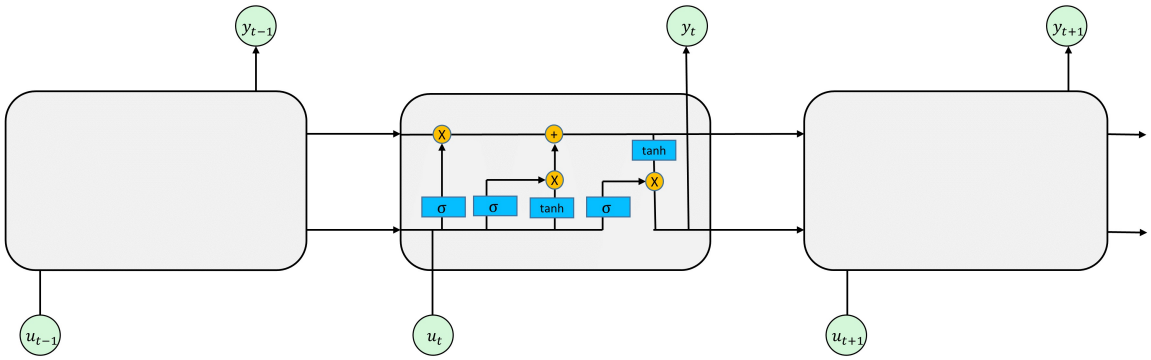


Figure 7.3: Information flow in LSTM units

The internal operation of t^{th} memory unit is shown in Fig. 7.3 which evaluates the mapping from the input sequence x_t output y_t using the given (7.1) to (7.5). The flow of information in each LSTM unit consists of hidden state vector h_t, cell memory state vector c_t, and three gates: forget gate f_t, input gate i_t and output gate o_t.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{7.1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{7.2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{7.3}$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7.4}$$

$$h_t = o_t * \tanh(c_t) \tag{7.5}$$

where h_t represents the output of the LSTM layer at t. W, U and b are the weight coefficients of input sequence, weight coefficients of hidden states and bias term of LSTM layer at time t respectively.

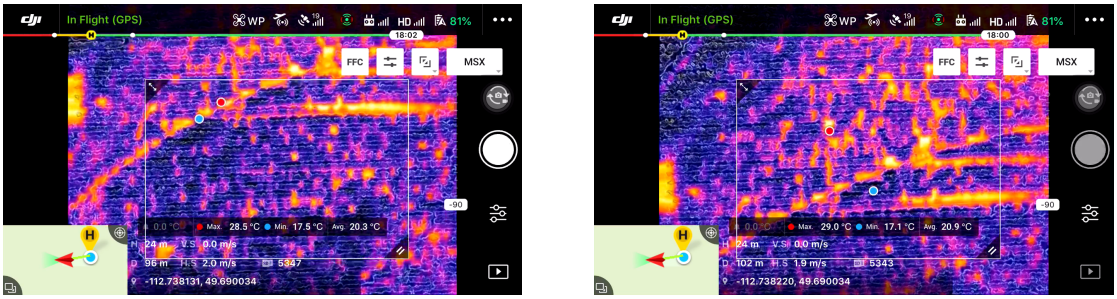
7.3 Data pre-processing

In this section, the images collected from the drone (Appendix A) and the soil moisture sensors are pre-processed before using in it the soil moisture estimation.

7.3.1 Convert thermal images to radiometric thermal images

As discussed in Appendix A, the main challenge with the used thermal cameras is that the thermal image lacks radiometric information, which means the thermal images give the dry and cold area but not accurate temperature information. The

radiometric thermal image contains the actual temperature information. In these thermal images, the actual temperature of the highest and lowest values of one image may be completely different from the other image's same highest and lowest values. However, while flying the drone, we can save the video of thermal images for the whole flight time. The video contains the highest, lowest, average temperature, and GPS locations of each time frame as shown in Figure 7.4.



(a) **(b)**

Figure 7.4: Extracted thermal images from video at different time

The following steps are employed to convert the thermal images into radiometric thermal images.

1. Convert the captured video into images using a predefined frame ratio. In this analysis, we convert the video into images every second.

2. The drone usually takes one image nearly every three seconds. In this step, we find those images from the converted images from the drone video. Figure 7.4(a) shows the image where drone is not taking images and Figure 7.4(b) shows where the drone is taking images. It can be noticed from 7.4(b) that the middle right side circular button becomes grey when the drone takes images (that means the home button gets pressed when the images are taken).

   - First, the home button region is selected on the images.
   - Then, the region is checked if grey or white. If it is grey, then the following images are selected.

3. The highest, lowest, and average temperatures for each selected image are calculated. First, the region of interest (ROI) around the temperature information

is selected. Then the temperature information is recognized using the optical character recognition (OCR) method and converted into machine-encoded text.

4. Convert the individual thermal gray images collected from drone to radiometric images using the highest and lowest temperature information.

$$T_{\text{radiometry}} = T_{\text{gray}} \times (T_{\max} - T_{\min}) + T_{\min}$$

7.3.2 Image registration and image stitching

The thermal map of the field is constructed by utilizing feature-based image processing techniques, image registration and image stitching. For image registration, the Scale-invariant Feature Transform (SIFT) is used for finding and matching SIFT descriptors [112]. An open-source MATLAB library VLFeat is used for implementing SIFT [113]. After successfully registering each image pair utilizing matched descriptors, images are stitched together to form a completed map of the field. The overlapped areas are processed by taking the average over the overlapped intensity values. The details of the algorithm is explained below.

The drone took images in a lawnmower flight pattern. The frontal overlap of two successive images in the same column was set as 50% and the side overlap of two neighbouring images in the adjoining columns was set as 70%. Therefore, for any two successive images taken by the drone, they contain a common area. The common area serves as the essential element for image registration and stitching. The following algorithm describes how to generate a thermal map from images.

1. Group images of the same column into one group, based on their GPS information.

2. Taking one group of images as an example

   - Detect and match features between two successive images, using SIFT. (Figure 7.5)
   - Among tentative matches obtained from SIFT, the inliers are selected using RANSAC. Based on selected inliers, the geometric transformation which maps one image to another one is estimated. (Figure 7.6)

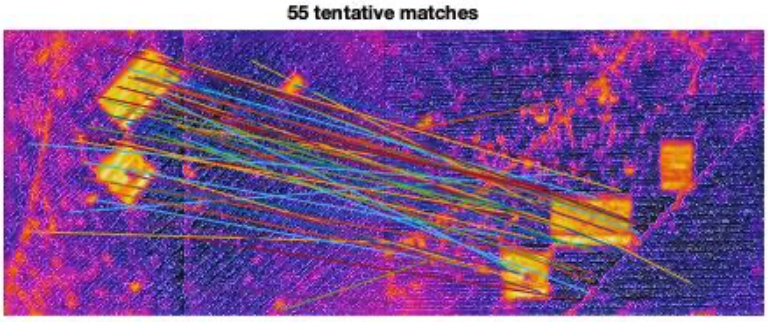


Figure 7.5: Tentative matches from between two images

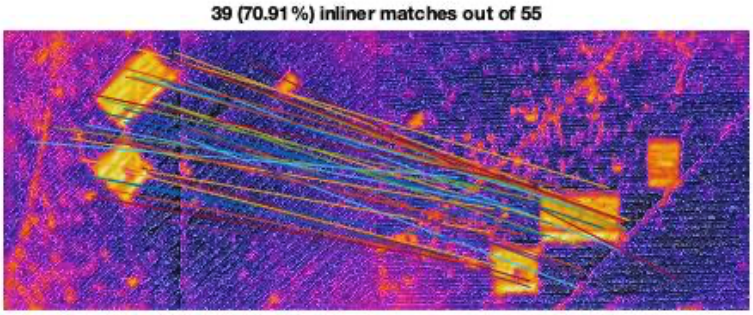


Figure 7.6: Selected inliner matches

- If only one pair of images is required for registration, the stitched image is shown in Figure 7.7.

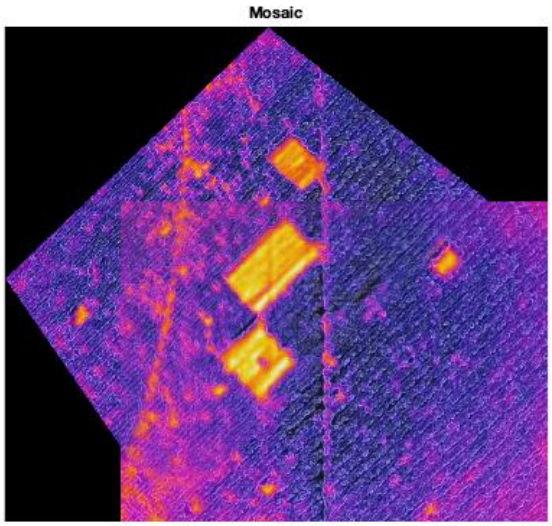
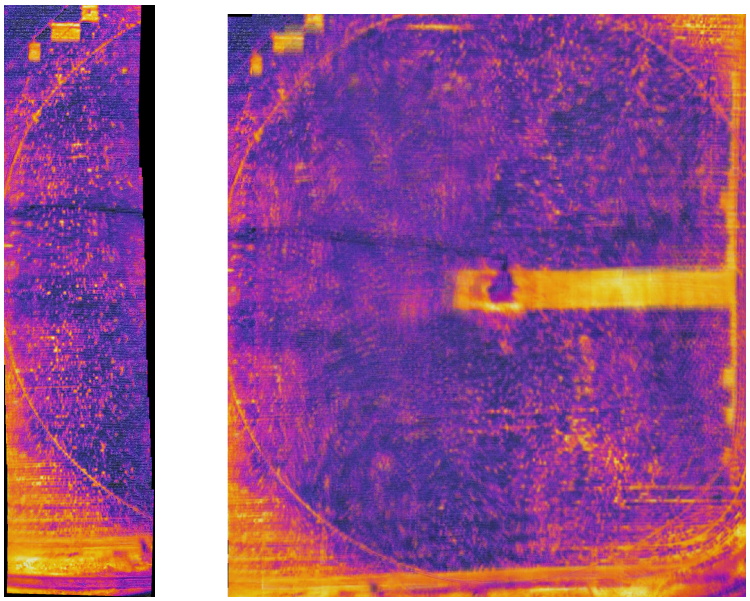


Figure 7.7: One pair image mosaic

- However, there are more than one pair of images within the same column. In order to stitch all of them together, first, estimate geometric transformations of all pairs of successive images. Then map images to the center of the image by multiplying transformation matrices.
- At last, the images within the column are stitched together. Pixels values over the overlapping area are calculated using the average of overlapping pixels at the same location (Figure 7.8 (a)).

3. Repeat Step 2 to stitch all columns

4. By treating each stitched column as an image, register each successive columns using the same methods in Step 2. Then stitch all columns into a completed map (Figure 7.8 (b)).



(a) **(b)**

Figure 7.8: (a) Stitched image of one column; (b) Completed stitched image

7.3.3 Identify the sensor locations

As discussed in Appendix A, we have 20 sensor locations marked on the field. We also collect the soil moisture simultaneously using handheld moisture probes at those marked sensor locations when we fly the drone. The GPS location of the marked

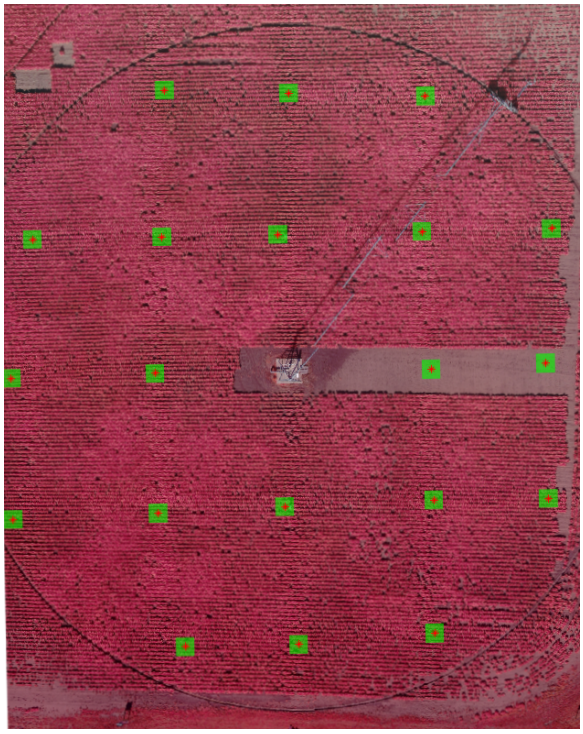


Figure 7.9: Identified sensor locations on NIR image of 31^{st} July, 2019

sensor position has been recorded. Each of the stitched NDVI image pixels is converted into GPS coordinates using two known GPS locations on the field and the same points on the image. The sensor locations on the image have been identified using the minimum distance between GPS coordinates of the marked sensor locations and the pixels of the stitched image. The image registration between different NIR, thermal and RGB images is not always perfect. So to handle the misregistration between the images, we used the averaging method around the sensor location. We select a small rectangle around the sensor pixels and average the values inside the rectangle. The average value has been used as an input to the LSTM instead of only one selected sensor pixel value. Figure 7.9 shows the identified sensor locations and averaging rectangle on NIR image of 31^{st} July, 2019. Since, every stitched images are registered with each other so the sensor pixel identification of one stitched image is sufficient.

7.4 Proposed soil moisture estimation method

Typically, remote sensing soil moisture estimation is performed based on static regression models. In this section, the time-varying soil moisture estimation using the remote sensing images and point sensors are proposed. LSTM model is mainly used to handle the sequential data. The agro-hydrological model based on the Richards equation (2.1) is a time-varying model. So it is more appropriate to use a time-varying model like LSTM to estimate the soil moisture than a static neural network model. The LSTM model uses the recent and past time step inputs as well as past time step outputs information to predict the current outputs of the system. Mathematically it can be written as follows:

$$y(t) = f(u(t), u(t-1), y(t-1)) \tag{7.6}$$

where $y(t)$: current time step outputs, $u(t)$: current time step inputs, $u(t-1)$: past time step inputs, $y(t-1)$: past time step output.

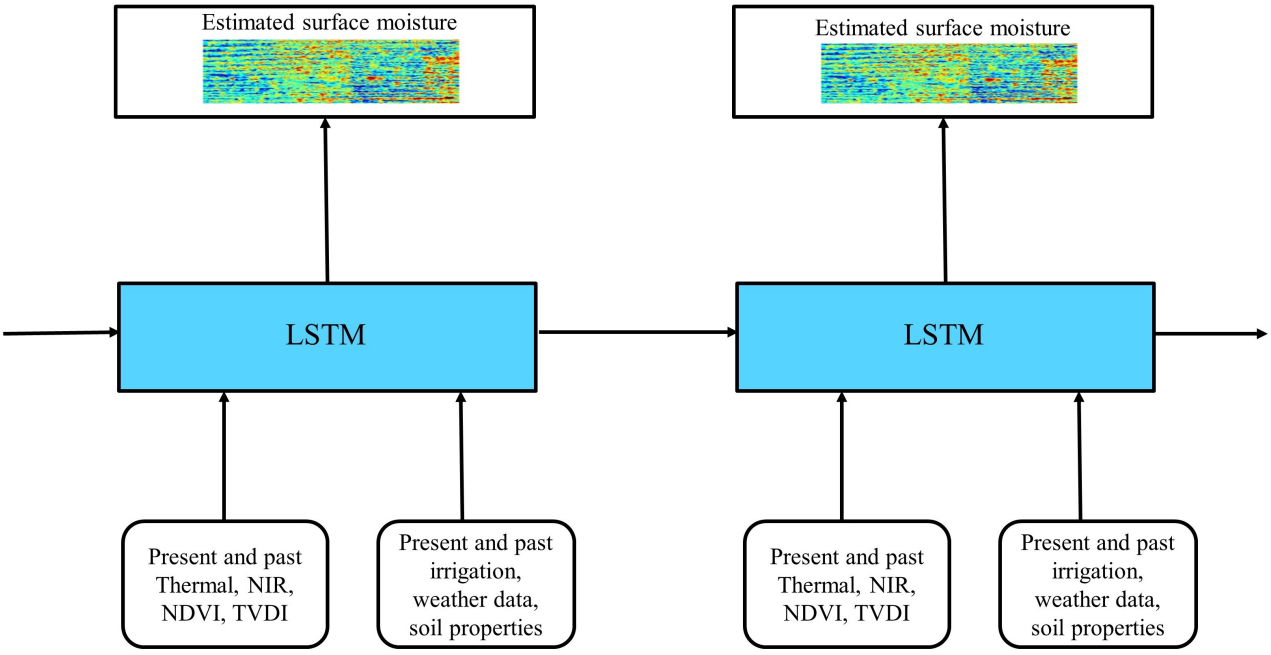


Figure 7.10: Flow diagram of proposed soil moisture estimation

Figure 7.10 shows the proposed method for entire farm's soil moisture estimation from the remote sensing images. For the remote sensing soil moisture estimation, the typical inputs to the system are thermal, NIR, NDVI, TVDI images, climate data

and geo-attributes of the farm. The climate data includes rain, ET, air temperature, and geo-attributes includes the soil field capacity. Let us divide the system input into two parts: 1) remote sensing images (U_1); 2) climate and geo-attributes (U_2). When we fly the drone, we usually do not obtain remote sensing images every day. So there may be irregular gaps present between $u(t)$ and $u(t-1)$. So for our proposed method, we use only recent past day climate and geo-attributes U_2 as a part of past inputs $u(t-1)$. Things to note are that we can obtain climate data every day.

The output of the LSTM is surface soil moisture. The LSTM is trained based on the surface soil moisture probe measurements at a few sensor locations. Then the surface soil moisture of the whole field is calculated from the LSTM model. In (7.6), we observe that we also use past day soil moisture ($y(t-1)$) as one of the inputs to the LSTM model.

7.5 Results and discussion

In this section, the proposed method is applied to real experimental data conducted in Summer 2019 at Lethbridge, Canada. The details and data collection procedure of the experiment are explained in Appendix A. The main purpose is to estimate the surface soil moisture of the field using remote sensing images. For this work, we consider the data from July 2^{th}, 2019 to July 31^{st}, 2019. For the month of July, 2019 on the date {2,4,9,11,15,17,19,22,23,26,29,30,31}, the remote sensing images are available. The weather data is obtained from the weather station near the demo farm. The input and output data at the sensor locations are obtained from the images, weather data, and soil moisture sensors. Figure 7.9 and A.9 shows the 20 sensors location from where the soil moisture data are collected from the hydraprobe sensor. The soil moisture data were collected at the same when the drone was flying. Out of 20 locations, 16 location data are used for training, and the other 4 locations {5,6,16,19} are used for validation. For the LSTM, two hidden layers of unit 300 are used. The epoch and batch size are 50 and 4, respectively. Figure 7.11 shows the Mean square error of the training and validation data with respect to the number of epochs. It can be seen that both the training and validation data converge and the MSE of the training is lower than the validation set. Figure 7.12 shows the training

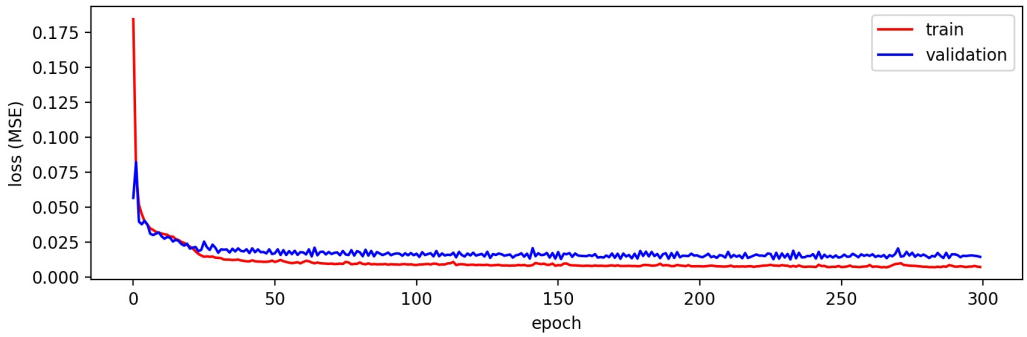


Figure 7.11: Performance plot of Mean Squared Error (MSE) with the epoch for training and validation

results of the predicted and measured soil moisture for four randomly selected sensor positions. We can observe that the predicted output from the LSTM model has good

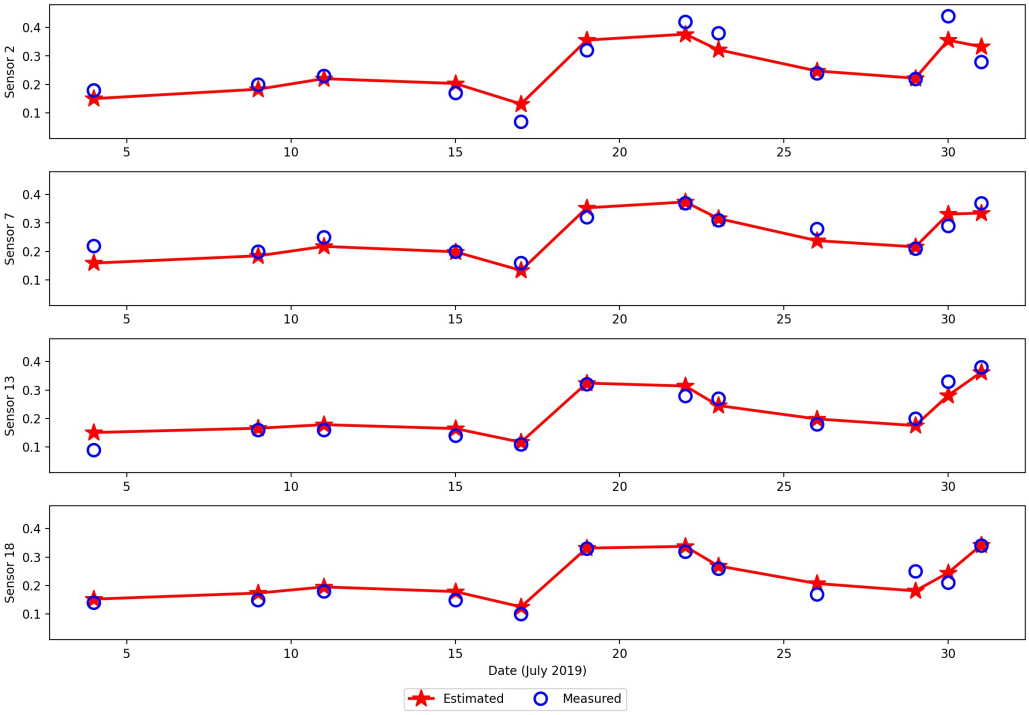


Figure 7.12: Training of measured and predicted soil moisture for some selected sensor positions {2,7,13,18}

agreement with the measured soil moisture. Figure 7.13 shows the validation results of predicted and measured soil moisture at the four validation sensor positions. We can see that the predicted values from the LSTM follow the trend of the measured soil moisture.

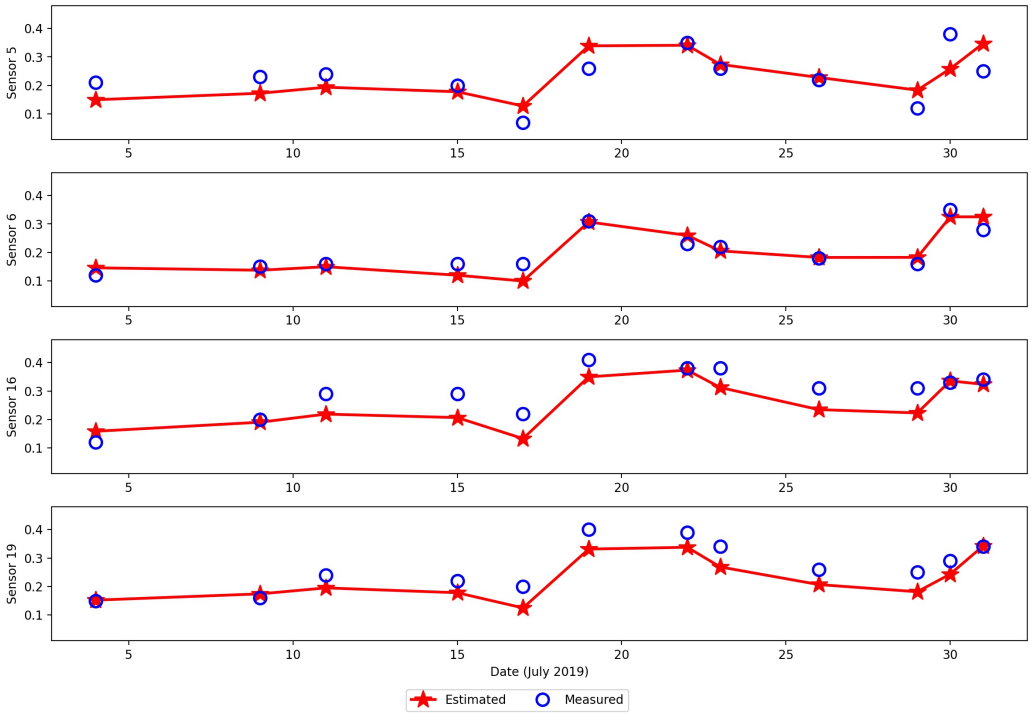


Figure 7.13: Validation of measured and predicted soil moisture for some selected sensor positions {5,6,16,19}

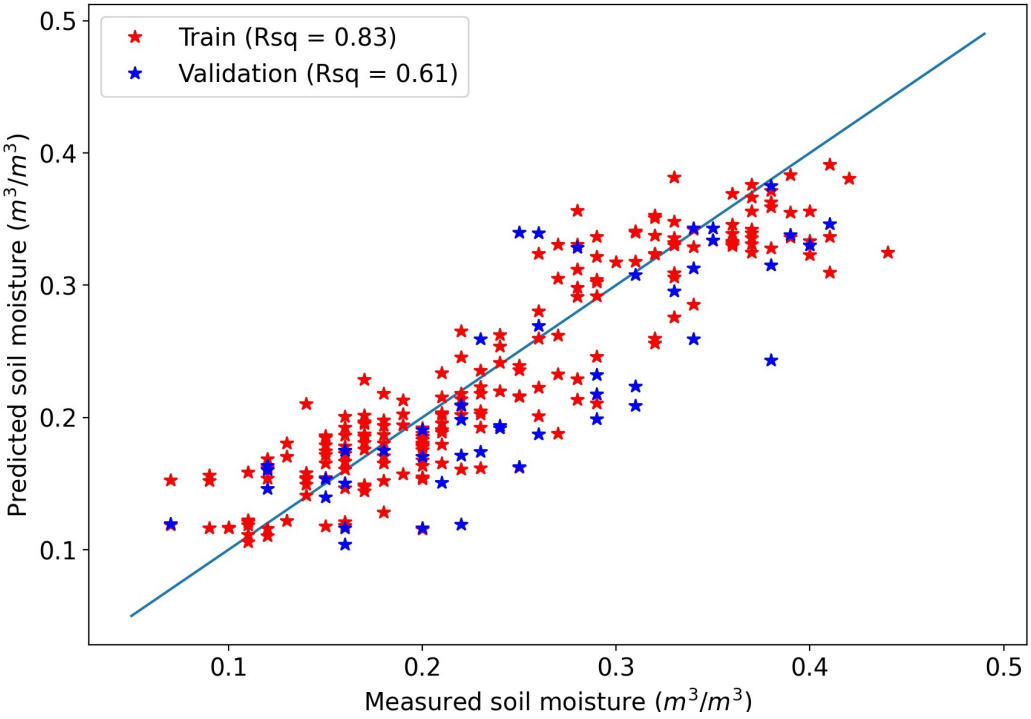


Figure 7.14: Measured and predicted soil moisture scatter plot

In Figure 7.15, for some selected dates, the surface soil moisture maps are shown.

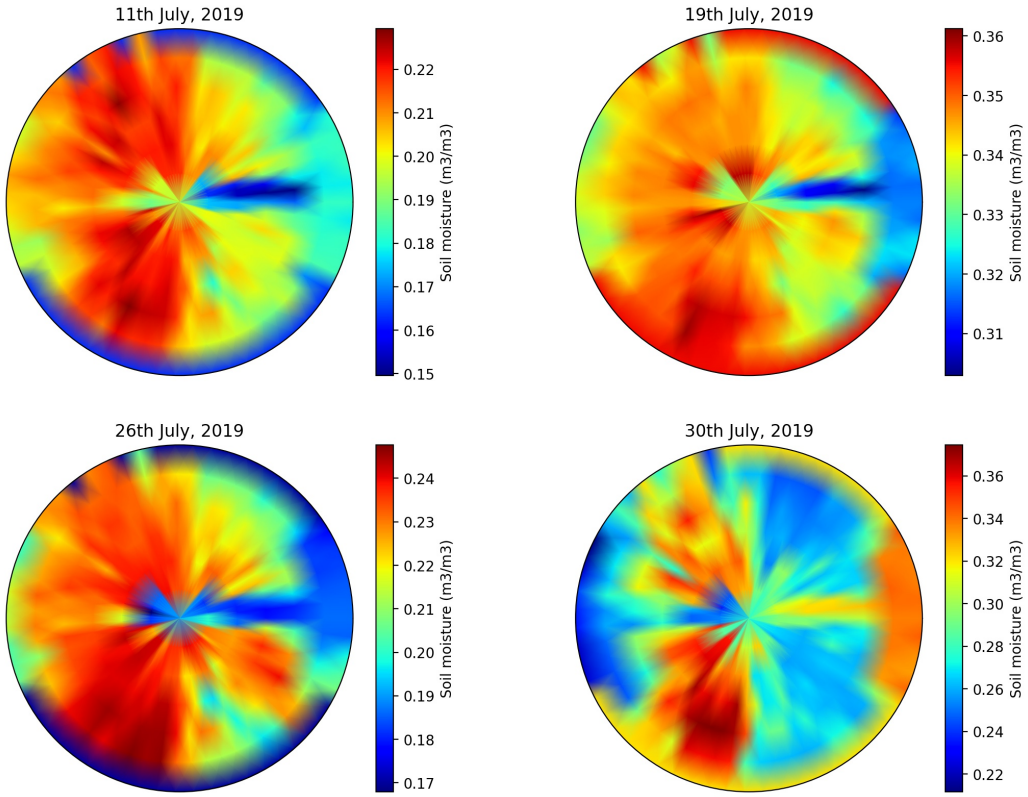


Figure 7.15: Surface soil moisture predicted by LSTM for the date 11^{th} July, 2019, 19^{th} July, 2019, 26^{th} July, 2019, 30^{th} July, 2019

For further analysis, all the validation and training measurement points vs. the predicted points at the same location and time are plotted in Figure 7.14. The result shows that most of the validation and training points are near the 45° line. The R^2 of the validation points are 0.61, and the training points are 0.83, which shows the superiority of the proposed algorithm.

7.6 Conclusions

In this chapter, we presented the algorithm to estimate the surface soil moisture from thermal and optical remote sensing images using the LSTM model. The surface soil moisture was estimated using the drone remote sensing images for a real field. The input to the LSTM was thermal, NDVI, TVDI, NIR images, rain, ET, irrigation

amount, and soil types. The LSTM was trained using the measured surface soil moisture collected using the hand-held hydroprobe sensors. The predicted output from the LSTM model at both validation and training sensor points showed a good agreement with the measured soil moisture with R^2 0.63 and 0.83, respectively.

Chapter 8

Conclusions and future work

8.1 Conclusions

In this thesis, we addressed major challenges in closed-loop irrigation for the application in large-scale agricultural fields. Different model reduction methods were proposed to handle the high dimensionality for sensor placement, state estimation, and scheduler design. Further, the remote sensing soil moisture estimation was discussed using the machine-learning based model.

In Chapter 3, the optimal sensor placement for three-dimensional agro-hydrological systems was addressed. A systematic procedure was proposed to address the problem. In the procedure, a clustering approach was proposed for model reduction. The graphical method for structural observability was used to find the minimum number of sensor for the nonlinear system. The maximum multiplicity theory was implemented to further investigate the observability results. Based on the observability results, model degree of observability was used to find the optimal sensor placement. The proposed procedure and methods were applied to the agro-hydrological systems under three scenarios. The results illustrated the effectiveness of the proposed procedure and methods.

In Chapter 4, the problem of optimal sensor placement for three-dimensional agro-hydrological systems in presence of different sprinklers was addressed. A systematic procedure was proposed to address the problem. The proposed procedure was applied to one agro-hydrological system. The results illustrated the effectiveness of the proposed procedure and methods.

In Chapter 5, the problem of higher dimensionality of the agro-hydrological system

was addressed. The algorithm for structure-preserving adaptive model reduction and adaptive state estimation was proposed. For a small simulated field, the original MHE performance was compared with the proposed MHE method. The results show that the adaptive MHE has a better performance than the original MHE in terms of estimation error and computational cost. The robustness of the adaptive MHE was also presented. The proposed approach has been applied to an real agricultural field under the noisy and nominal case. In both cases, the state estimation showed a satisfactory result and the effectiveness of the proposed adaptive model reduction methods.

In Chapter 6, the closed-loop scheduler for a large three-dimensional agro-hydrological system was proposed. The motivation and the algorithm of the proposed scheduler were presented. The scheduler optimizes the irrigation amount and time of the following irrigation event to assure maximum yield, water preservation, and electricity usage. The proposed approach has been applied to three different cases. In all the cases, the proposed scheduler showed a satisfactory result.

In Chapter 7, LSTM based model was proposed to estimate surface soil moisture using thermal and optical remote sensing images. This chapter used the data from the experiment conducted in summer 2019. First, the definitions of different vegetation indices were discussed. Then, the data pre-processing steps for the collected drone images were discussed. The LSTM was trained using the collected soil moisture. The training and validation of LSTM showed satisfactory results. The surface soil moisture map was constructed using the LSTM model.

8.2 Future research directions

8.2.1 Computationally efficient 3D model development

The computationally efficient large-scale agro-hydrological model still has some room to develop. The 3D model should be capable of handling detailed agricultural land information like slope, crop biology, etc. The complex field geometries can be solved efficiently using the adaptive meshing and immersogeometric finite element method.

8.2.2 Machine learning-based state estimation

The machine learning models like LSTM may improve the online computational speed of the estimation methods. It will be interesting to use the LSTM for a large-scale agricultural field. Most of the machine learning-based models experience the curse of dimensionality. More research may be devoted towards the model reduction-based machine learning methods.

8.2.3 Scheduler and controller design using reinforcement learning (RL)

In recent years, reinforcement learning is gaining popularity in the field of dynamic and adaptive control. It will be interesting to apply reinforcement learning in the field of scheduler and controller design. The model-based controller knowledge can be applied to train reinforcement learning which may result in faster online computational speed.

8.2.4 Efficient soil moisture estimation using different remote sensing methods

There are still many challenges associated with the thermal and optical remote sensing methods. The integration of microwave, thermal and optical remote sensing methods will be useful to handle many of the recent remote sensing challenges.

8.2.5 Distributed state and controller design

In the thesis, we proposed structure-preserving model reduction techniques. The structure-preserving quality of the reduced model can be beneficial to construct the distributed framework. The distributed estimation and control will improve the computational cost.

Bibliography

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

[2] "Aqueduct water risk atlas." `https://www.wri.org/aqueduct`.

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

[4] C. Moseley and A. Nicolas, "Water in a changing world," tech. rep., Unesco Pub, Paris, 2010.

[5] H. Turral, B. Jacob, and F. Jean-Marc, "Climate change, water and food security," *FAO Water Reports Rome*, 2011.

[6] A. Sarma, "Precision irrigation-a tool for sustainable management of irrigation water," *Proceedings of the Civil Engineering for Sustainable Development-Opportunities and Challenges, Guwahati, India*, pp. 19–21, 2016.

[7] 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.

[8] X. Zou, Y. Li, K. Li, R. Cremades, Q. Gao, Y. Wan, and X. Qin, "Greenhouse gas emissions from agricultural irrigation in china," *Mitigation and Adaptation Strategies for Global Change*, vol. 20, no. 2, pp. 295–315, 2015.

[9] A. Daccache, J. W. Knox, E. K. Weatherhead, A. Daneshkhah, and T. M. Hess, "Implementing precision irrigation in a humid climate – Recent experiences and on-going challenges," *Agricultural Water Management*, vol. 147, pp. 135–143, Jan. 2015.

[10] R. Romero, J. Muriel, I. García, and D. Muñoz de la Peña, "Research on automatic irrigation control: State of the art and recent results," *Agricultural Water Management*, vol. 114, pp. 59–66, Nov. 2012.

[11] S. L. Shah, B. R. Bakshi, J. Liu, C. Georgakis, B. Chachuat, R. D. Braatz, and B. R. Young, "Meeting the challenge of water sustainability: The role of process systems engineering," *AIChE Journal*, vol. 67, no. 2, p. e17113, 2021.

[12] "Ismc-the international soil modeling consortium." `https://soil-modeling.org/resources-links/model-portal`.

[13] L. A. Richards, "Capillary conduction of liquids through porous mediums," *Physics*, vol. 1, pp. 318–333, Nov. 1931.

[14] M. W. Farthing and F. L. Ogden, "Numerical Solution of Richards' Equation: A Review of Advances and Challenges," *Soil Science Society of America Journal*, vol. 81, no. 6, p. 1257, 2017.

[15] M. Camporese, C. Paniconi, M. Putti, and S. Orlandini, "Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data," *Water Resources Research*, vol. 46, Feb. 2010.

[16] P. Brunner and C. T. Simmons, "HydroGeoSphere: A Fully Integrated, Physically Based Hydrological Model," *Groundwater*, vol. 50, pp. 170–176, Mar. 2012.

[17] M. G. Trefry and C. Muffels, "FEFLOW: A Finite-Element Ground Water Flow and Transport Modeling Tool," *Groundwater*, vol. 45, pp. 525–528, Sept. 2007.

[18] S. Finsterle, C. Doughty, M. Kowalsky, G. J. Moridis, L. Pan, T. Xu, Y. Zhang, and K. Pruess, "Advanced Vadose Zone Simulations Using TOUGH," *Vadose Zone Journal*, vol. 7, no. 2, p. 601, 2008.

[19] R. W. Healy, "Simulating Water, Solute, and Heat Transport in the Subsurface with the VS2di Software Package," *Vadose Zone Journal*, vol. 7, pp. 632–639, May 2008.

[20] J. Šimůnek, V. Genuchten, M. Th, and M. Šejna, "Development and Applications of the HYDRUS and STANMOD Software Packages and Related Codes," *Vadose Zone Journal*, vol. 7, pp. 587–600, May 2008.

[21] L. Orgogozo, N. Renon, C. Soulaine, F. Hénon, S. Tomer, D. Labat, O. Pokrovsky, M. Sekhar, R. Ababou, and M. Quintard, "An open source massively parallel solver for Richards equation: Mechanistic modelling of water fluxes at the watershed scale," *Computer Physics Communications*, vol. 185, pp. 3358–3371, Dec. 2014.

[22] 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.

[23] H. Vereecken, A. Schnepf, J. Hopmans, M. Javaux, D. Or, T. Roose, J. Vanderborght, M. Young, W. Amelung, M. Aitkenhead, S. Allison, S. Assouline, P. Baveye, M. Berli, N. Brüggemann, P. Finke, M. Flury, T. Gaiser, G. Govers, T. Ghezzehei, P. Hallett, H. Hendricks Franssen, J. Heppell, R. Horn, J. Huisman, D. Jacques, F. Jonard, S. Kollet, F. Lafolie, K. Lamorski, D. Leitner, A. McBratney, B. Minasny, C. Montzka, W. Nowak, Y. Pachepsky, J. Padarian, N. Romano, K. Roth, Y. Rothfuss, E. Rowe, A. Schwen, J. Šimůnek, A. Tiktak, J. Van Dam, S. van der Zee, H. Vogel, J. Vrugt, T. Wöhling, and I. Young, "Modeling Soil Processes: Review, Key Challenges, and New Perspectives," *Vadose Zone Journal*, vol. 15, no. 5, p. 0, 2016.

[24] R. H. Reichle, J. P. Walker, R. D. Koster, and P. R. Houser, "Extended versus ensemble Kalman filtering for land data assimilation," *J. Hydrometeor.*, vol. 3, no. 6, pp. 728–740, 2002.

[25] H. Lü, Z. Yu, Y. Zhu, S. Drake, Z. Hao, and E. A. Sudicky, "Dual state-parameter estimation of root zone soil moisture by optimal parameter estimation and extended Kalman filter data assimilation," *Advances in Water Resources*, vol. 34, no. 3, pp. 395–406, 2011.

[26] G. J. M. D. Lannoy, P. R. Houser, V. R. N. Pauwels, and N. E. C. Verhoest, "State and bias estimation for soil moisture profiles by an ensemble Kalman filter: Effect of assimilation depth and frequency," *Water Resources Research*, vol. 43, no. 6, 2007.

[27] C.-C. Wu and S. A. Margulis, "Feasibility of real-time soil state and flux characterization for wastewater reuse using an embedded sensor network data assimilation approach," *Journal of Hydrology*, vol. 399, pp. 313–325, Mar. 2011.

[28] D. Pasetto, M. Camporese, and M. Putti, "Ensemble Kalman filter versus particle filter for a physically-based coupled surface–subsurface model," *Advances in Water Resources*, vol. 47, pp. 1–13, 2012.

[29] D. Erdal, M. Rahman, and I. Neuweiler, "The importance of state transformations when using the ensemble Kalman filter for unsaturated flow modeling: Dealing with strong nonlinearities," *Advances in Water Resources*, vol. 86, pp. 354–365, Dec. 2015.

[30] W. Kurtz, G. He, S. J. Kollet, R. M. Maxwell, H. Vereecken, and H.-J. Hendricks Franssen, "TerrSysMP–PDAF (version 1.0): a modular high-performance data assimilation framework for an integrated land surface–subsurface model," *Geoscientific Model Development*, vol. 9, no. 4, pp. 1341–1360, 2016.

[31] H. Zhang, H.-J. Hendricks Franssen, X. Han, J. A. Vrugt, and H. Vereecken, "State and parameter estimation of two land surface models using the ensemble Kalman filter and the particle filter," *Hydrol. Earth Syst. Sci.*, vol. 21, no. 9, pp. 4927–4958, 2017.

[32] C. Montzka, H. Moradkhani, L. Weihermüller, H.-J. H. Franssen, M. Canty, and H. Vereecken, "Hydraulic parameter estimation by remotely-sensed top soil moisture observations with the particle filter," *Journal of Hydrology*, vol. 399, no. 3, pp. 410–421, 2011.

[33] 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.

[34] S. Bo, S. R. Sahoo, X. Yin, J. Liu, and S. L. Shah, "Simultaneous parameter and state estimation of agro-hydrological systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11767–11772, 2020. 21st IFAC World Congress.

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

[36] D. Pasetto, G.-Y. Niu, L. Pangle, C. Paniconi, M. Putti, and P. A. Troch, "Impact of sensor failure on the observability of flow dynamics at the Biosphere 2 LEO hillslopes," *Advances in Water Resources*, vol. 86, pp. 327–339, 2015.

[37] J. Nahar, J. Liu, and S. L. Shah, "Parameter and state estimation of an agro-hydrological system based on system observability analysis," *Computers & Chemical Engineering*, vol. 121, pp. 450–464, 2019.

[38] H. G. Jones, "Irrigation scheduling: advantages and pitfalls of plant-based methods," *Journal of Experimental Botany*, vol. 55, pp. 2427–2436, Sept. 2004.

[39] G. Vellidis, M. Tucker, C. Perry, C. Kvien, and C. Bednarz, "A real-time wireless smart sensor array for scheduling irrigation," *Computers and Electronics in Agriculture*, vol. 61, pp. 44–50, Apr. 2008.

[40] B. Cardenas-Lailhacar, M. D. Dukes, and G. L. Miller, "Sensor-Based Automation of Irrigation on Bermudagrass, during Wet Weather Conditions," *Journal of Irrigation and Drainage Engineering*, vol. 134, pp. 120–128, Apr. 2008.

[41] B. Cardenas-Lailhacar, M. D. Dukes, and G. L. Miller, "Sensor-Based Automation of Irrigation on Bermudagrass during Dry Weather Conditions," *Journal of Irrigation and Drainage Engineering*, vol. 136, pp. 184–193, Mar. 2010.

[42] T. Boutraa, A. Akhkha, A. Alshuaibi, and R. Atta, "Evaluation of the effectiveness of an automated irrigation system using wheat crops," *Agriculture and Biology Journal of North America*, vol. 2, pp. 80–88, Jan. 2011.

[43] S. A. O'Shaughnessy, S. R. Evett, P. D. Colaizzi, and T. A. Howell, "A crop water stress index and time threshold for automatic irrigation scheduling of

grain sorghum," *Agricultural Water Management*, vol. 107, pp. 122–132, May 2012.

[44] Y. Erdem, L. Arin, T. Erdem, S. Polat, M. Deveci, H. Okursoy, and H. T. Gültaş, "Crop water stress index for assessing irrigation scheduling of drip irrigated broccoli (Brassica oleracea L. var. italica)," *Agricultural Water Management*, vol. 98, pp. 148–156, Dec. 2010.

[45] R. T. Peters and S. R. Evett, "Automation of a Center Pivot Using the Temperature-Time-Threshold Method of Irrigation Scheduling," *Journal of Irrigation and Drainage Engineering*, vol. 134, pp. 286–291, June 2008.

[46] B. E. Belayneh, J. D. Lea-Cox, and E. Lichtenberg, "Costs and Benefits of Implementing Sensor-controlled Irrigation in a Commercial Pot-in-Pot Container Nursery," *HortTechnology*, vol. 23, pp. 760–769, Dec. 2013.

[47] J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. A. Porta-Gandara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, pp. 166–176, Jan. 2014.

[48] Z. Li, N. Wang, T. Hong, A. Franzen, and J. Li, "Closed-loop drip irrigation control using a hybrid wireless sensor and actuator network," *Science China Information Sciences*, vol. 54, pp. 577–588, Mar. 2011.

[49] Y. Kim, R. G. Evans, and W. M. Iversen, "Evaluation of Closed-Loop Site-Specific Irrigation with Wireless Sensor Network," *Journal of Irrigation and Drainage Engineering*, vol. 135, pp. 25–31, Feb. 2009.

[50] M. Saavoss, J. Majsztrik, B. Belayneh, J. Lea-Cox, and E. Lichtenberg, "Yield, quality and profitability of sensor-controlled irrigation: a case study of snapdragon production," *Irrig Sci*, vol. 34, pp. 409–420, Sept. 2016.

[51] A. C. McCarthy, N. H. Hancock, and S. R. Raine, "Development and simulation of sensor-based irrigation control strategies for cotton using the VARIwise

simulation framework," *Computers and Electronics in Agriculture*, vol. 101, pp. 148–162, Feb. 2014.

[52] A. C. McCarthy, N. H. Hancock, and S. R. Raine, "VARIwise: A general-purpose adaptive control simulation framework for spatially and temporally varied irrigation at sub-field scale," *Computers and Electronics in Agriculture*, vol. 70, pp. 117–128, Jan. 2010.

[53] D. Barnard and W. Bauerle, "Species-specific irrigation scheduling with a spatially explicit biophysical model: A comparison to substrate moisture sensing with insight into simplified physiological parameterization," *Agricultural and Forest Meteorology*, vol. 214-215, pp. 48–59, Dec. 2015.

[54] F. Capraro, D. Patino, S. Tosetti, and C. Schugurensky, "Neural network-based irrigation control for precision agriculture," in *2008 IEEE International Conference on Networking, Sensing and Control*, pp. 357–362, IEEE, 2008.

[55] N. Karasekreter, F. Başçiftçi, and U. Fidan, "A new suggestion for an irrigation schedule with an artificial neural network," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, pp. 93–104, Mar. 2013.

[56] H. Navarro-Hellín, J. Martínez-del Rincon, R. Domingo-Miguel, F. Soto-Valles, and R. Torres-Sánchez, "A decision support system for managing irrigation in agriculture," *Computers and Electronics in Agriculture*, vol. 124, pp. 121–131, June 2016.

[57] A. K.Mousa, M. S. Croock, and M. N. Abdullah, "Fuzzy based Decision Support Model for Irrigation System Management," *International Journal of Computer Applications*, vol. 104, pp. 14–20, Oct. 2014.

[58] E. Giusti and S. Marsili-Libelli, "A Fuzzy Decision Support System for irrigation and water conservation in agriculture," *Environmental Modelling & Software*, vol. 63, pp. 73–86, Jan. 2015.

[59] Y. Park, J. S. Shamma, and T. C. Harmon, "A Receding Horizon Control algorithm for adaptive management of soil moisture and chemical levels during

irrigation," *Environmental Modelling & Software*, vol. 24, pp. 1112–1121, Sept. 2009.

[60] S. K. Saleem, D. K. Delgoda, S. K. Ooi, K. B. Dassanayake, L. Liu, M. N. Halgamuge, and H. Malano, "Model Predictive Control for Real-Time Irrigation Scheduling," *IFAC Proceedings Volumes*, vol. 46, pp. 299–304, Aug. 2013.

[61] D. Delgoda, H. Malano, S. K. Saleem, and M. N. Halgamuge, "Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AQUACROP model," *Environmental Modelling & Software*, vol. 78, pp. 40–53, Apr. 2016.

[62] C. Lozoya, C. Mendoza, A. Aguilar, A. Román, and R. Castelló, "Sensor-Based Model Driven Control Strategy for Precision Irrigation," *Journal of Sensors*, vol. 2016, pp. 1–12, 2016.

[63] A. C. McCarthy, N. H. Hancock, and S. R. Raine, "Advanced process control of irrigation: the current state and an analysis to aid future development," *Irrigation Science*, vol. 31, pp. 183–192, May 2013.

[64] O. Adeyemi, I. Grove, S. Peets, and T. Norton, "Advanced Monitoring and Management Systems for Improving Sustainability in Precision Irrigation," *Sustainability*, vol. 9, p. 353, Feb. 2017.

[65] 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.

[66] M. H. Sahraei, M. A. Duchesne, R. W. Hughes, and L. A. Ricardez-Sandoval, "Dynamic reduced order modeling of an entrained-flow slagging gasifier using a new recirculation ratio correlation," *Fuel*, vol. 196, pp. 520–531, 2017.

[67] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control, Society for Industrial and Applied Mathematics, 2005.

[68] T. Ishizaki, K. Kashima, J. Imura, and K. Aihara, "Model reduction and clusterization of large-scale bidirectional networks," *IEEE Transactions on Automatic Control*, vol. 59, pp. 48–63, 2014.

[69] X. Cheng, Y. Kawano, and J. M. A. Scherpen, "Graph structure-preserving model reduction of linear network systems," in *2016 European Control Conference (ECC)*, pp. 1970–1975, 2016.

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

[71] B. Peherstorfer and K. Willcox, "Dynamic data-driven reduced-order models," *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 21–41, 2015.

[72] B. Peherstorfer and K. Willcox, "Online adaptive model reduction for nonlinear systems via low-rank updates," *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. A2123–A2150, 2015.

[73] A. Ångström, "The albedo of various surfaces of ground," *Geografiska Annaler*, vol. 7, no. 4, pp. 323–342, 1925.

[74] L. Weidong, F. Baret, G. Xingfa, T. Qingxi, Z. Lanfen, and Z. Bing, "Relating soil surface moisture to reflectance," *Remote Sensing of Environment*, vol. 81, pp. 238–246, Aug. 2002.

[75] D. B. Lobell and G. P. Asner, "Moisture Effects on Soil Reflectance," *SOIL SCI. SOC. AM. J.*, vol. 66, p. 6, 2002.

[76] W. Liu, F. Baret, X. Gu, B. Zhang, Q. Tong, and L. Zheng, "Evaluation of methods for soil surface moisture estimation from reflectance data," *International Journal of Remote Sensing*, vol. 24, pp. 2069–2083, Jan. 2003.

[77] M. L. Whiting, L. Li, and S. L. Ustin, "Predicting water content using Gaussian model on soil spectra," *Remote Sensing of Environment*, vol. 89, pp. 535–552, Feb. 2004.

[78] Z. Gao, X. Xu, J. Wang, H. Yang, W. Huang, and H. Feng, "A method of estimating soil moisture based on the linear decomposition of mixture pixels," *Mathematical and Computer Modelling*, vol. 58, pp. 606–613, Aug. 2013.

[79] D. Zhang and G. Zhou, "Estimation of Soil Moisture from Optical and Thermal Remote Sensing: A Review," *Sensors*, vol. 16, p. 1308, Aug. 2016.

[80] J. Peng, Y. Liu, and A. Loew, "Uncertainties in Estimating Normalized Difference Temperature Index From TOA Radiances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 2487–2497, May 2013.

[81] S. O. Ihuoma and C. A. Madramootoo, "Recent advances in crop water stress detection," *Computers and Electronics in Agriculture*, vol. 141, pp. 267–275, Sept. 2017.

[82] J. Price, "Using spatial context in satellite data to infer regional scale evapotranspiration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28, pp. 940–948, Sept. 1990.

[83] M. S. Moran, T. R. Clarke, Y. Inoue, and A. Vidal, "Estimating crop water deficit using the relation between surface-air temperature and spectral vegetation index," *Remote Sensing of Environment*, vol. 49, pp. 246–263, Sept. 1994.

[84] Y. Yang, H. Guan, D. Long, B. Liu, G. Qin, J. Qin, and O. Batelaan, "Estimation of Surface Soil Moisture from Thermal Infrared Remote Sensing Using an Improved Trapezoid Method," *Remote Sensing*, vol. 7, pp. 8250–8270, June 2015.

[85] G. Petropoulos, T. Carlson, M. Wooster, and S. Islam, "A review of Ts/VI remote sensing based methods for the retrieval of land surface energy fluxes and soil surface moisture," *Progress in Physical Geography*, vol. 33, pp. 224–250, Apr. 2009.

[86] R. R. Gillies, W. P. Kustas, and K. S. Humes, "A verification of the 'triangle' method for obtaining surface soil water content and energy fluxes from remote

measurements of the Normalized Difference Vegetation Index (NDVI) and surface e," *International Journal of Remote Sensing*, vol. 18, pp. 3145–3166, Oct. 1997.

[87] J. Xin, G. Tian, Q. Liu, and L. Chen, "Combining vegetation index and remotely sensed temperature for estimation of soil moisture in China," *International Journal of Remote Sensing*, vol. 27, pp. 2071–2075, May 2006.

[88] I. Sandholt, K. Rasmussen, and J. Andersen, "A simple interpretation of the surface temperature/vegetation index space for assessment of surface moisture status," *Remote Sensing of Environment*, vol. 79, pp. 213–224, Feb. 2002.

[89] Yang Han, Yeqiao Wang, and Yunsheng Zhao, "Estimating Soil Moisture Conditions of the Greater Changbai Mountains by Land Surface Temperature and NDVI," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, pp. 2509–2515, June 2010.

[90] J. Chen, C. Wang, H. Jiang, L. Mao, and Z. Yu, "Estimating soil moisture using Temperature–Vegetation Dryness Index (TVDI) in the Huang-huai-hai (HHH) plain," *International Journal of Remote Sensing*, vol. 32, pp. 1165–1177, Feb. 2011.

[91] T. Carlson, "An Overview of the "Triangle Method" for Estimating Surface Evapotranspiration and Soil Moisture from Satellite Imagery," *Sensors*, vol. 7, pp. 1612–1629, Aug. 2007.

[92] D. Zhang, R. Tang, B.-H. Tang, H. Wu, and Z.-L. Li, "A Simple Method for Soil Moisture Determination From LST–VI Feature Space Using Nonlinear Interpolation Based on Thermal Infrared Remotely Sensed Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, pp. 638–648, Feb. 2015.

[93] M. E. Holzman, R. Rivas, and M. C. Piccolo, "Estimating soil moisture and the relationship with crop yield using surface temperature and vegetation index," *International Journal of Applied Earth Observation and Geoinformation*, vol. 28, pp. 181–192, May 2014.

[94] D. Zhang, R. Tang, W. Zhao, B. Tang, H. Wu, K. Shao, and Z.-L. Li, "Surface Soil Water Content Estimation from Thermal Remote Sensing based on the Temporal Variation of Land Surface Temperature," *Remote Sensing*, vol. 6, pp. 3170–3187, Apr. 2014.

[95] P. Rahimzadeh-Bajgiran, A. A. Berg, C. Champagne, and K. Omasa, "Estimation of soil moisture using optical/thermal infrared remote sensing in the Canadian Prairies," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 94–103, Sept. 2013.

[96] Z. Liu, Z. Yao, and R. Wang, "Evaluating the surface temperature and vegetation index method for estimating surface soil moisture in heterogeneous regions," *Hydrology Research*, vol. 49, pp. 689–699, June 2018.

[97] L. Hassan-Esfahani, A. Torres-Rua, A. Jensen, and M. McKee, "Assessment of Surface Soil Moisture Using High-Resolution Multi-Spectral Imagery and Artificial Neural Networks," *Remote Sensing*, vol. 7, pp. 2627–2646, Mar. 2015.

[98] 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.

[99] Y. Mualem, "A new model for predicting the hydraulic conductivity of unsaturated porous media," *Water Resources Research*, vol. 12, pp. 513–522, June 1976.

[100] M. T. van Genuchten, "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils1," *Soil Science Society of America Journal*, vol. 44, no. 5, p. 892, 1980.

[101] R. A. Feddes, *Simulation of field water use and crop yield*. Pudoc, 1982.

[102] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[103] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, "Observability of complex systems," *Proceedings of the National Academy of Sciences*, vol. 110, pp. 2460–2465, Feb. 2013.

[104] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Communications*, vol. 4, no. 1, 2013.

[105] S. Gu, F. Pasqualetti, M. Cieslak, Q. K. Telesford, A. B. Yu, A. E. Kahn, J. D. Medaglia, J. M. Vettel, M. B. Miller, S. T. Grafton, and D. S. Bassett, "Controllability of structural brain networks," *Nature Communications*, vol. 6, p. 8414, 2015.

[106] A. G. Egorov, R. Z. Dautov, J. L. Nieber, and A. Y. Sheshukov, "Stability analysis of gravity-driven infiltrating flow," *Water resources research*, vol. 39, no. 9, 2003.

[107] C. Zhao, A. Zeng, R. Jiang, Z. Yuan, and W.-X. Wang, "Controllability of flow-conservation networks," *Physical Review E*, vol. 96, July 2017.

[108] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," 2000.

[109] R. Feddes and P. Raats, "Parameterizing the soil-water-plant root system," *Unsaturated-zone modeling: Progress, challenges, applications*, vol. 6, pp. 95–141, 2004.

[110] "Ndvi faq: All you need to know about index." `https://eos.com/blog/ndvi-faq-all-you-need-to-know-about-ndvi/`.

[111] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[112] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[113] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." `http://www.vlfeat.org/`, 2008.

[114] "Alberta climate information service website." `https://agriculture.alberta.ca/acis/`.

[115] R. F. Carsel and R. S. Parrish, "Developing joint probability distributions of soil water retention characteristics," *Water resources research*, vol. 24, no. 5, pp. 755–769, 1988.

[116] "Dji mavic 2 enterprise dual." https://www.dji.com/ca/.

[117] "Agrocam." https://www.agrocam.eu.

[118] "Watermark sensors." https://www.campbellsci.ca/253-l.

[119] "Hydrago sensors." https://stevenswater.com/products/hydrago-s/.

[120] "Data logger." https://www.campbellsci.ca/cr800.

[121] "Multiplexer." https://www.campbellsci.ca/am16-32b.

[122] "Flir lepton radiometric camera." https://www.flir.ca/products/lepton/.

[123] "Arducam ptz camera." https://www.arducam.com.

Appendix A

Remote sensing data collection procedure using drone

A.1 Introduction

This Appendix discusses the data collection procedure and analysis of the remote sensing experiment conducted in summer 2019. The experiment has three main steps: 1) Soil sample collection for texture analysis; 2) Thermal and optical remote sensing image collection by a flying drone; 3) Soil moisture data collection for ground truth First, we discuss the details of the study area. Then the soil sample collection procedure and the soil texture analysis method are discussed. Afterward, we discuss the image collection procedure from the drone. Finally, we discuss the soil moisture data collection procedure.

A.2 Study area

The agricultural field is located east of the city Lethbridge, Alberta, Canada (Lon: $-112.7385 : -112.7365$, Lat: $49.6896 : 49.6908$). The field is equipped with a central pivot irrigation system and consists of one pivot arm with 21 sprinklers. The radius of the field is 50m. One weather station is located near the agricultural field managed by Lethbridge Demo Farm Irrigation Management Climate Information Network (IMCIN). The weather station's data is obtained from the Alberta Climate Information Service website [114]. The crop was sugar beet. The reference evapotranspiration, rain, wind speed, air temperature is obtained from the weather station. The central pivot usually takes 8 hours to irrigate the whole field.

(a)

(b)

Figure A.1: a) Demo farm in Lethbridge, (b) Satellite view of the demo farm

A.3 Soil texture analysis

The soil parameters of the model are obtained from the soil texture experiment. The five parameters value $(K_s, \alpha, n, \theta_s, \theta_r)$ in Richards equation (2.1) depend upon the types of soil. The soil samples are collected at 40 points in the field (20 points from surface to depth 0.25 cm, 20 points at 0.25 cm to 0.5 cm) using auger A.2. After collecting the soil sample, the types of soil are estimated in the soil lab. The five parameters of the soil are calculated based on types of soil [115]. However, there are soil samples at only 40 points in the field, the soil parameters for the other nodes are unknown. The traditional kriging approach is used to map all the parameters for the whole field. For the initial analysis, the spherical kriging has been used to map all the parameters for the whole field. Figure A.3 shows some of the selected parameters of the entire Demo farm after kriging interpolation.

A.4 Remote sensing image collection using drone

For the experiment, the DJI Mavic 2 Enterprise Dual [116] is used to capture the RGB and thermal images. The AgroCam NDVI [117] camera is used to obtain the NIR image and the blue NDVI. The thermal camera is already integrated into the drone. The Agrocam NDVI camera is mounted on the bottom of the DJI Mavic drone

Figure A.2: Soil sample collection

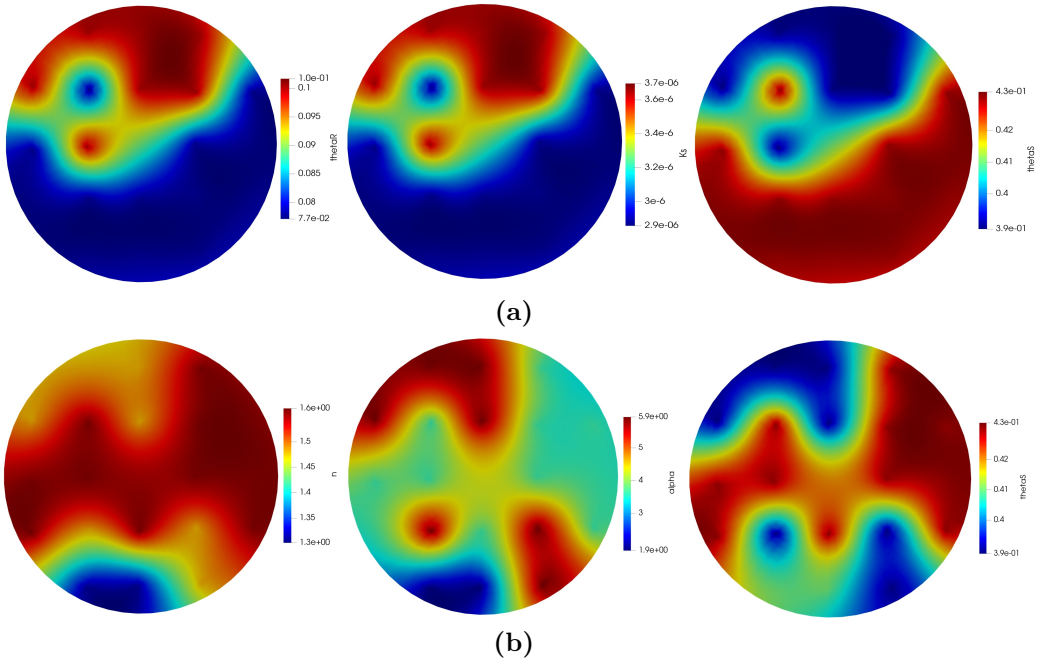


Figure A.3: (a) soil parameters $(\theta_R, K_S, \theta_S)$ from top to 25 cm depth, (b) soil parameters (n, α, θ_S) from 25 cm to 50 cm depth

(a) (b)

Figure A.4: a) DJI Mavic 2 Enterprise Dual, (b) AgroCam

using the GPS module and integration module. Thus we get all three spectral images simultaneously. The drone and the Agrocam is shown in Figure A.4 [116, 117].

Some of the essential flight details are; 1) flight altitude: 80 ft, 2) front and side overlap: 80% and 75%, flight speed: 7 mph. The issue with the thermal camera in DJI Mavic is that it does not provide radiometric thermal images. It only gives the cold and hot area for the particular image. Thus we take the video of the thermal images and later convert the thermal image to a radiometric thermal image. The details of conversion are discussed in chapter 7. The procedure to take the video is discussed as follows:

- Download the DJI PILOT app.
- Connect your phone to the remote controller of the drone. Open the DJI App to make sure it is connected.
- After the drone has taken off, open the app and turn on the screen recording. Sometimes the app fails to provide a video recording as the drone flies. You may have to close the app and open it again and then start the screen recording.
- Drag the box to cover more area in the image as shown in Figure 3.
- We need to make sure that the shutter button should not be covered by the phone's onscreen function button as shown in Figure A.5.
- The app should remain open as the drone flies and during the screen recording process.

- Stop the screen recording when the drone returns home.

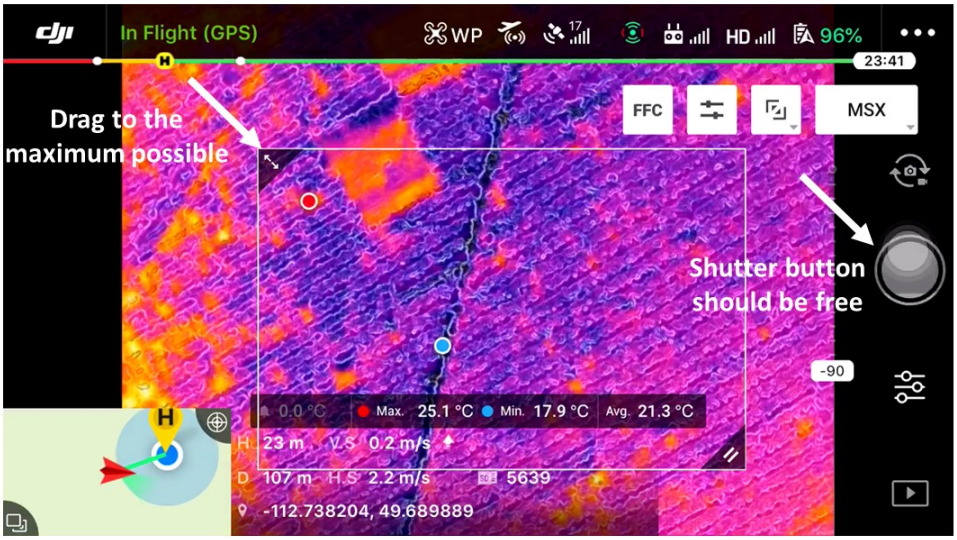


Figure A.5: Video taking procedure

The checklist for the flight operation is as follows:

A.4.1 Checklist

- Pre-flight procedure check list:
  - Check the weather condition is suitable for operation
  - Check NOTAMS
  - Check if the UAV battery has full charge
  - Check if the remote controller is fully charged
  - Check if the mobile battery has full charge
  - Check the flight path
  - Verify the takeoff/ landing area is clear and there is no obstacle
  - Check if the NDVI camera are turned on
  - Check SD card is inserted in both drone and NDVI camera

  - Check if the SD card has enough space
  - Check if the propeller is mounted correctly

- Take-off check list:
  - Turn on the controller
  - Turn on the UAV
  - Check if the return to home setting is correct
  - Check GPS signal of NDVI camera is green
  - Check if the camera is pointing downwards
  - Verify if the motor is working normally by hovering for 15 sec
  - Check if the pre-flight check list is green in drone deploy app

- During flight checklist:
  - Check the battery level of the UAV by the remote controller
  - Continuous monitor for any obstacles like birds, persons
  - Go back to the DJI pilot app and start recording the screen to get the maximum and minimum temperature of the field
  - Check if the landing area is clear

- Post flight checklist:
  - Power off the UAV
  - Power off the payload cameras
  - Visually check if the aircraft is free of damage
  - Check if the NDVI image and thermal image are captured

A.5 Soil moisture data collection

The soil moisture collection is important to train and validate the first principle and machine learning based model. In the experiment, two different types of soil moisture sensors are used (Figure A.6):

- Watermark sensor [118]
- Stevan hydrago sensor [119]

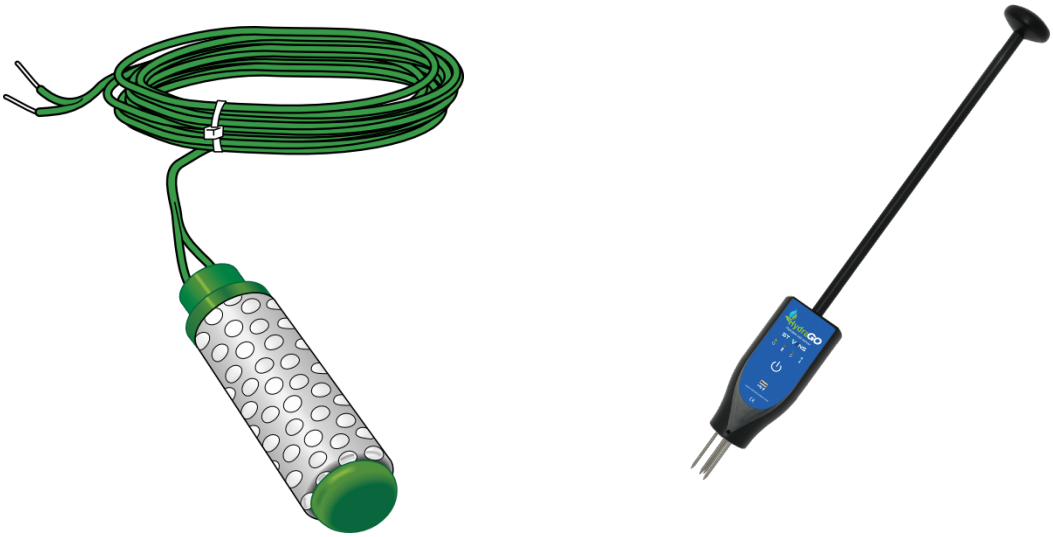


Figure A.6: a) Watermark sensor, b) Hydrago sensor

The watermark sensor is used measure soil water tension and it is placed at different depths. The hydrago sensor is used to measure the surface soil moisture, soil temperature and electric conductivity. Forty-two watermark sensors are installed at 14 sites in the field at a depth of 25 cm, 50 cm and 75 cm. The watermark sensors are first connected to the multiplexer. The multiplexers are used to connect many sensors to a single data logger. The data loggers collect the data from the water mark sensor. As the data logger does not have enough space to connect all the 42 sensors, the multiplexer is used to connect multiple sensors to the data logger. Two data loggers (CR 800) [120] and two multiplexers [121] are used for the whole field (Figure A.7). The CRBasic program is used to connect the water mark sensor to the data logger. Two solar panels are installed to charge the data loggers. The data logger, multiplexer and solar panel are shown in Figure A.8.

The Steven hydrago sensor is used to collect surface soil moisture at specific locations in the field while the drone is flying. The location and field layout of the watermark sensors and the hydrogo sensors are shown in Figure A.9. The soil moisture from the hydrago sensors are collected from all the 20 locations. While the water

Figure A.7: a) CR-800 data logger , b) Multiplexer

Figure A.8: Data logger, Multiplexer, Solar panel and Sensors

mark sensors are located at the 1, 2, 3, 5, 6, 7, 10, 11, 14, 15, 16, 18, 19, 20 sites.

A.6 Data collection procedure

Daily data collection procedure

- Fly the drone between the time 8.00AM to 10AM

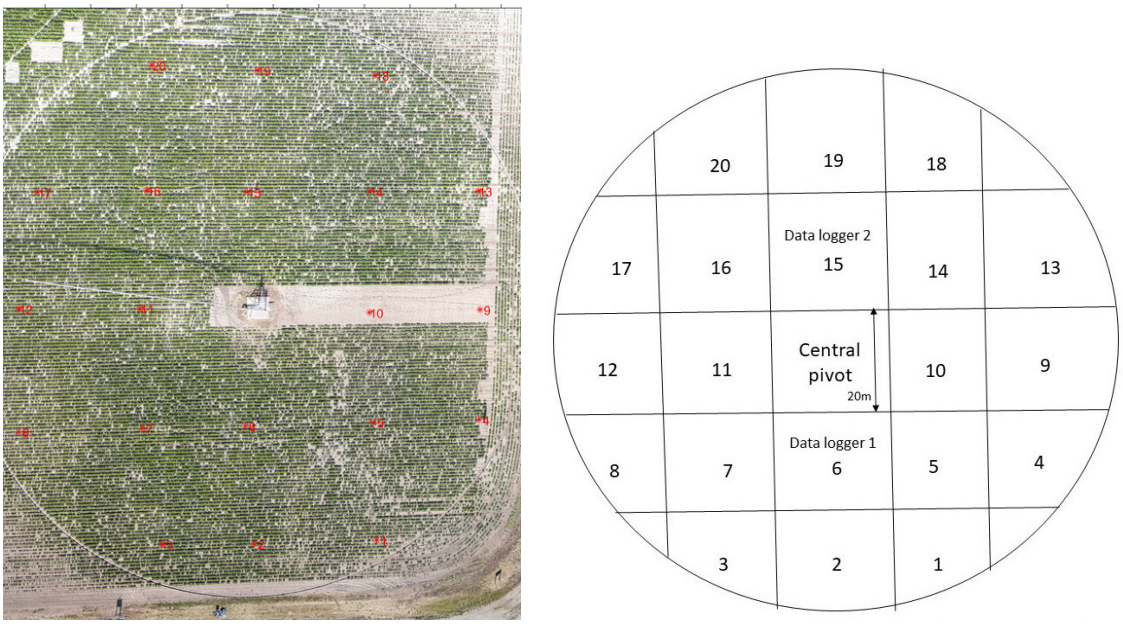


Figure A.9: a) Sensor positions, b) Field layout

- Collect data from steven probe during the flying of drone
- Record the weather data from the weather information website
- Take the picture of the leaf and monitor the growth rate of sugar beet

Weekly data collection procedure

- Collect the data from the watermark sensor

Appendix B

Remote sensing methods by mounting cameras on mower

B.1 Introduction

In this appendix, the remote sensing data collection procedure and analysis of the experiment conducted at the Royal Mayfair Golf Club, Edmonton, in the summer of 2020 is discussed. In this experiment, the cameras are mounted on the mower, and when the mower moves, the multi-spectral images are captured automatically. In this appendix, first, we discuss the study area, equipment design, and the data collection procedure. Afterward, the algorithm for the automatic image collection procedure is discussed. In the end, some of the preliminary results are discussed.

B.2 Equipment design

The Royal Mayfair Golf Club is located at 9450 Groat Rd NW, Edmonton, AB T6G 2T5. For the setup, the multispectral camera has been designed. The multispectral camera consists of a thermal camera (FLIR Lepton radiometric camera) [122] and an optical camera (Arducam PTZ Camera) [123]. In the Arducam PTZ camera, there is a switchable IR cut which means the camera first takes the visible range image and then it switches back to capture near the infrared image. The whole setup consists of the multi-spectral camera set, a GPS module, Raspberry pi computer module. In the golf club every morning the mower cuts the grass. The multispectral cameras are mounted on the mower as shown in Figure B.1.

Figure B.1: Cameras mounted on mower

The code has been written on the raspberry pi in such a way that it detects the movement of the mower using the GPS location and starts taking pictures and stop taking pictures when the movement stops. For some days, the actual soil moisture has been collected using the handheld hydroporobe sensor.

B.3 Results

The mower path is shown in Figure B.2. The surface temperature map is constructed using the thermal images (Figure B.3 (a)). Using the soil moisture collected and the temperature images, the surface soil moisture is estimated using simple regression analysis. The Figure B.3 (b) shows the surface soil moisture map.

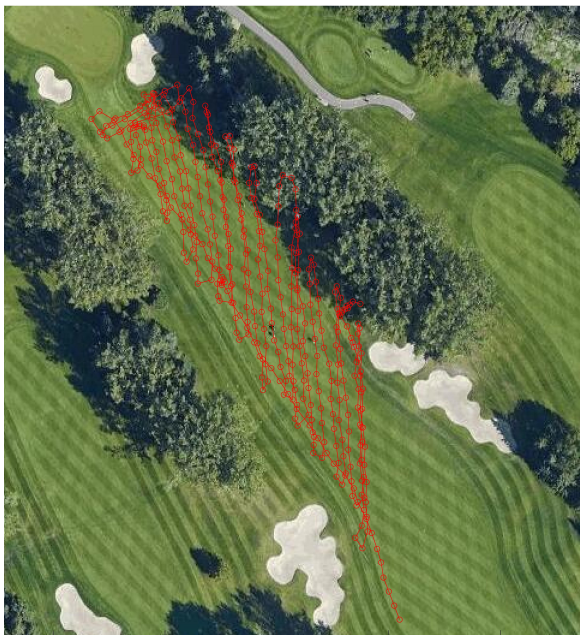


Figure B.2: Path of mower at the Royal Mayfair Golf Club

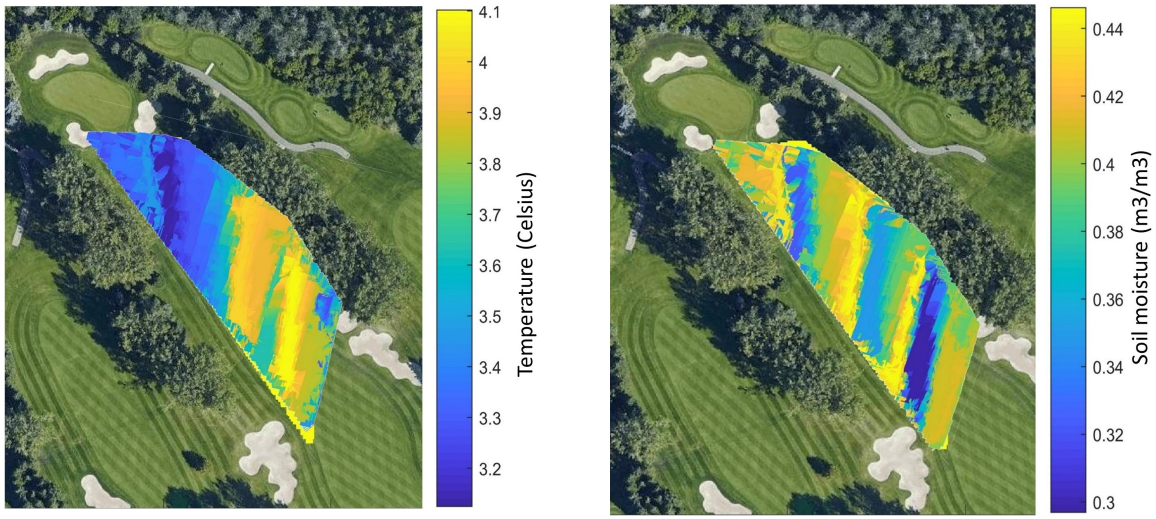


Figure B.3: a) Surface temperature, b) Surface soil moisture