공학박사 학위논문

Construction and Online Adaptation of Nonlinear Semi-batch Process Model for Digital Twin Under Limited Data

제한적 데이터를 갖는 비선형 반회분식 공정의 디지털 트윈 구현을 위한 모델 구축 및 실시간 적응법

2021년 2월

서울대학교 대학원
화학생물공학부
배 재 한

Construction and Online Adaptation of Nonlinear Semi-batch Process Model for Digital Twin Under Limited Data

지도교수 이 종 민

이 논문을 공학박사 학위논문으로 제출함
2020년 12월

서울대학교 대학원
화학생물공학부
배 재 한

배 재 한의 공학박사 학위논문을 인준함
2020년 12월

위 원 장 _____ 이 원 보 _____ (인)

부위원장 _____ 이 종 민 _____

위 원 _____ 남 재 욱 _____ (인)

위 원 _____ 정 동 휘 _____

위 원 _____ 김 연 수 _____

Abstract

Construction and Online Adaptation of Nonlinear Semi-batch Process Model for Digital Twin Under Limited Data

Jaehan Bae
School of Chemical and Biological Engineering
The Graduate School
Seoul National University

A batch process in the chemical and biological industry can be characterized by its non-stationary operation, tracking certain pre-determined input trajectories for uniform production. This property of the batch process causes many difficulties in mathematical modeling procedure, which is an essential step for designing a digital twin of the target process. For modeling the process, it requires nonlinear models to simulate the complex system dynamics, and a parameter estimation (PE) problem has to be solved to complete the modeling procedure. Particularly, the PE becomes tricky, owing to limited information in the measured data of fixed operating conditions. Regardless of the amount of given data, the lack of information results in ill-conditioned PE, and not all model parameters can be estimated under this condition. Furthermore, the completion of the process model does not always lead to a successful application of model-based techniques to the batch processes. This is because unknown disturbances

are constantly affecting the actual process, resulting in a model-plant mismatch. For nonlinear models simulating the batch process, this problem is further highlighted by the structural uncertainty of the model itself. These problems occur frequently when modeling processes of production-scale and should be resolved to construct a digital twin. This thesis proposes model-based approaches that can manage these characteristics of the industrial batch processes.

First, we present the modeling and dynamic optimization techniques using a dynamic hybrid model that can avoid the ill-conditioned PE problem. The hybrid models are useful in situations where the available data is limited, and there is little confidence in the model structure owing to a lack of prior information about the process. The first part of this thesis defines a valid domain of dynamic hybrid model and proposes a method to utilize it as a constraint on dynamic optimization. It also provides an iterative algorithm to overcome the limitations of the narrow valid domain.

The second method examines the first-principle model, which can be used when prior information about the process is available. Initially, a stochastic model with additive stochastic terms is constructed based on the first-principle model. Then, a parameter subset selection (PSS) algorithm, which manages the ill-conditioned PE problem caused by insufficient data, is suggested for the stochastic model. A PE method for the stochastic model, estimating both the model parameters and the magnitudes of additive stochastic terms, is also suggested to support the PSS algorithm.

Finally, using a nonlinear first-principle model, we present an optimization-based online state and parameter estimation technique that can manage the model-plant mismatch. If a large number of pa-

rameters should be updated online with a limited amount of data, the online state and parameter estimation also encounters the ill-conditioned PE problem. To address this problem, we introduce the online PSS method into a moving horizon estimation, presenting an estimation algorithm that can solve the ill-conditioning problem of the online PE problem.

Illustrative examples are included at the end of each chapter to verify the performance of the proposed methods. Each example uses a virtual plant simulating a fed-batch bioreactor, which has the characteristics of nonlinear behaviors, non-stationary operation, and limited information in the measured data.

Keywords: Digital twin, Semi-batch process, Hybrid model, Dynamic optimization, Parameter estimation, State estimation
Student Number: 2015-21066

Contents

Abstract . . . . . **i**

1. Introduction **1**

1.1 Motivation 1

1.2 Scope of the thesis 6

1.3 Outline of the thesis 8

2. Background and preliminaries **9**

2.1 Parameter and state estimation for nonlinear dynamic models 9

2.1.1 Parameter estimation for nonlinear models . . 9

2.1.2 Moving horizon state and parameter estimation 12

2.2 Estimability analysis 17

2.2.1 Parameter ranking and subset selection 18

2.2.2 Sensitivity-based methods 19

3. Hybrid modeling of semi-batch process and construction of valid domain constraints **25**

3.1 Introduction 25

3.2 Hybrid modeling of dynamic system 27

3.3 Valid domain for dynamic hybrid model 32

3.3.1 Valid domain for static hybrid model 33

3.3.2 Convex hull criterion for dynamic hybrid model 35

3.3.3 Confidence interval criterion for dynamic hybrid model 37

3.3.4 Valid domain constraints for dynamic hybrid model 43
3.4 Dynamic optimization with valid domain constraints . 45
3.4.1 Problem formulation 45
3.4.2 Iterative application of overall scheme 47
3.5 Illustrative example 49
3.5.1 Hybrid model structure and problem statements 50
3.5.2 Formulate discretized dynamic hybrid model of target system 55
3.5.3 Construction of valid domain for dynamic hybrid model 62
3.5.4 Dynamic optimization with valid domain constraints 66
3.5.5 Iterative model update and dynamic optimization 74

4. Ranking-based Parameter Subset Selection for Nonlinear Dynamics with Stochastic Disturbances under Limited Data 80
4.1 Introduction 80
4.2 Parameter estimation for SDE models with additive stochastic terms 82
4.2.1 System modeled with stochastic differential equations 82
4.2.2 Laplace Approximation Maximum Likelihood Estimation (LAMLE) 84
4.3 Ranking-based PSS for simple SDE model under limited data 87

4.3.1 Improvement of initial parameter guesses via sampling-based optimization 87
4.3.2 Calculation of scaled sensitivity matrix and ranked list of parameters 89
4.3.3 Modified LAMLE algorithm for parameter subset selection 90
4.3.4 MSE-based PSS of SDEs model using gLAMLE 104
4.4 Numerical illustration: Modeling a fed-batch bioreactor under limited data 107
4.4.1 Description on fed-batch bioreactor model . . 107
4.4.2 Parameter Ranking with θ_o^* 114
4.4.3 MSE-based PSS with gLAMLE subroutine . . 118

5. Multirate Moving Horizon Estimation Combined with Parameter Subset Selection 135
5.1 Introduction 135
5.2 Multi-rate moving horizon estimation 137
5.3 Parameter ranking-based moving horizon estimation . 139
5.3.1 Online parameter subset selection in k^{th} horizon 140
5.3.2 Modification of stage cost to consider multi-rate measurements 142
5.3.3 Parameters ranking-based moving horizon state and parameter estimation 146
5.4 Numerical Example: A fed-batch bioreactor 150
5.4.1 Problem statement 151
5.4.2 Benchmark problems: basic moving horizon state and parameter estimations 155

5.4.3 Parameter ranking-based moving horizon state and parameter estimation 165

6. Concluding remarks **171**

6.1 Summary of the contributions 172

6.2 Limitations and Future works 173

A. Appendix **178**

A.1 Calculate Hessian of $\psi_k(\theta)$ 178

A.2 Minimizing condition number of matrix by scaling [1] 180

A.3 Simulation of SDE 181

A.4 Laplace approximation 183

Bibliography **184**

List of Tables

Table 3.1. Initial conditions and physical constraints for simulation 52
Table 3.2. Hyper-parameters of the black-box models . . 57
Table 4.1. Parameters to be estimated: true values and initial range 109
Table 4.2. Table for known parameters 112
Table 4.3. Table for operating conditions 113
Table 5.1. Initial conditions and process constants for simulation 153
Table 5.2. Model parameters for virtual plant and nonlinear model for the MHE 154
Table 5.3. Sum of NRMSE for state estimation and prediction results- MHE(x), MHE(x, θ), and prMHE 162

List of Figures

Figure 2.1. Graphical description for moving horizon estimation 14

Figure 3.1. Structures of a serial and an embedded hybrid model. (a) Parallel type. (b) Serial type. (c) Embedded type. 29

Figure 3.2. Example for convex hull criterion, where $x_b = [x_1, x_2]^\top \in \mathbb{R}^2$ 36

Figure 3.3. Rearranging multi-rate measurements 42

Figure 3.4. A flow diagram of an iterative application of overall scheme 48

Figure 3.5. Randomly perturbed input trajectories for the data generation 54

Figure 3.6. Comparision between true value and predicted value. (a) Traning result - μ, (b) Traning result - π 59

Figure 3.7. Hybrid model prediction results. (a) case 1 : Random input in the convex hull of ID data, (b)case 2 : Random input out of the convex hull of ID data 61

Figure 3.8. Calculated convex hull interval criterion. (a) convex hull of [X,S,P], (b) on X axis, (c) on S axis, (d) on P axis 63

Figure 3.9. Calculated confidence interval criterion for each output variables. (a) Φ_X plot, (b) Φ_S plot, (c) Φ_P plot, (d) Φ_V plot 65

Figure 3.10. Optimized state trajectories on the confidence interval criterion compared with the reference. (a) optimized X [g/L], (b) optimized S [g/L], (c) optimized P [g/L], (d) optimized V [kL] . . 69
Figure 3.11. Optimal state trajectories on the convex hull criterion compared with the reference. (a) optimized trajectory, (b) on X axis, (c) on S axis, (d) on P axis 71
Figure 3.12. Checking the validity of the optimized input trajectory : comparison of the state trajectories of the hybrid model and the plant model under the optimized input trajectory 73
Figure 3.13. Optimized state trajectories on the confidence interval criterion compared with the reference (iteration 2). (a) optimized X [g/L], (b) optimized S [g/L], (c) optimized P [g/L], (d) optimized V [kL] 75
Figure 3.14. Optimal state trajectories on the convex hull criterion compared with the reference (iteration 2). (a) optimized trajectory, (b) on X axis, (c) on S axis, (d) on P axis 76
Figure 3.15. Checking the validity of the optimized input trajectory (iteration 2) : compare the state trajectories of the hybrid model and the plant model under the optimized input trajectory 78

Figure 4.1. B-spline curve fitting with de Boor's algorithm: (a) testing a various number of knots and orders, (b) testing a variation of the value of B-spline coefficients. 92
Figure 4.2. Examples of convergence of Q, Σ iteration in LAMLE algorithm. (a) converged, (b) failed . 100
Figure 4.3. Flow-chart for the proposed algorithm 106
Figure 4.4. Boxplot for the distribution of θ_o^*. o : the true value, — : the median, — : the 25th and 75th percentiles, - - : whisker, + : outliers 115
Figure 4.5. Rank estimation results: (a) PR result with θ_o^o, (b) PR result with θ_o^* 117
Figure 4.6. MSE plots for the data set used in the modeling step (a) J_{k_r} plot, (b) MSE_{k_r} plot 119
Figure 4.7. Calculated indices for MSE-based parameter subset selection. (a)r_{C,k_r} (b)r_{CKub,k_r} ,(c)r_{CC,k_r} 120
Figure 4.8. Boxplot of the estimated parameters in $\hat{\Theta_{k_r}}$ (a) $k_r = 3$, (b) $k_r = 5$ 122
Figure 4.9. Boxplot of the estimated parameters in $\hat{\Theta_{k_r}}$ (a) $k_r = 7$, (b) $k_r = 12$ 123
Figure 4.10. Boxplots for estimated standard deviations of (a) constant state disturbances $\hat{Q}$ ($\sqrt{\hat{q}_i}$), (b) measurement errors $\hat{\Sigma}$ ($\hat{\sigma}_i$) 126
Figure 4.11. Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 3$, $\hat{\Theta}_3 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}\}$ 128

Figure 4.12. Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 5$, $\hat{\Theta}_5 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}, \hat{\theta}^{8}, \hat{\theta}^{1}\}$ 129
Figure 4.13. Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 7$, $\hat{\Theta}_7 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}, \hat{\theta}^{8}, \hat{\theta}^{1}, \hat{\theta}^{12}, \hat{\theta}^{9}\}$ 130
Figure 4.14. Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 12$, $\hat{\Theta}_{12} =$ all model parameters 131
Figure 4.15. Normalized mean squared errors for $k_r = 1, 2, \ldots, 12$ (a) training datasets (b) testing datasets 133
Figure 5.1. Multi-rate measurements indexed based on their sampling rates 138
Figure 5.2. Multi-rate measurements aligned along with the sampling points of the model discretization . . 144
Figure 5.3. Estimated states and model prediction results of the basic MMHE-aUKF: Measured outputs (black $\triangle$ marker), true state of the virtual plant (black solid line), estimated state (red dotted line), predicted state (green o marker) 157
Figure 5.4. Estimated states and model prediction results of the MMHE-aUKF estimating the states and the model parameters altogether 160
Figure 5.5. Trajectories of the model parameters updated with the MMHE-aUKF (red dotted line) and prMHE (blue dotted line) compared to the true trajectories (black solid line) 164

Figure 5.6. The number of the selected model parameters over time 167
Figure 5.7. Estimated states and model prediction results of prMHE 169
Figure 6.1. Effect of the threshold on the size of selected parameter subset calculated by the orthogonalization method 175

List of Algorithms

.1 Orthogonalization Algorithm [2] 21

.2 Wu's PSS algorithm [3] 23

.3 $gLAMLE$ algorithm 103

.4 prMHE algorithm 149

Chapter 1

Introduction

1.1 Motivation

In manufacturing industries, many processes are managed as a batch process, despite the recent attention drawn to a highly efficient batch-to-continuous concept [4, 5]. Higher value-added products, for example, polymers, semiconductors, and bio-pharmaceutics, are produced in a batch process to guarantee product quality [4]. A semi-batch process is one type of batch process that enables the operator to supply additional materials during the operation. The batch process often involves complicated phenomena, such as phase equilibria, heat transfer, reaction equilibria. Another key feature of the batch process is that it is operated in a non-stationary condition. States of the process keep changing throughout the operation, until reaching a specific condition; by the time that the states attain the target condition, the operation terminates, and the next batch starts after some preparation periods. As a consequence, a nonlinear model is often the first consideration when constructing a digital twin of the batch processes [4].

A digital twin refers to a digital counterpart of the physical processes [6]. The digital twin is formulated with the prior knowledge

and the measured data of its physical counterpart. It also requires a constant update with online measurements to reflect the changing characteristics of the target process. A well-designed digital twin allows simulation of various operational conditions. It also helps in understanding the effects of user-defined scenarios before performing further experiments on the real-world process. Moreover, as an accurate model of the target process, it can be used in many model-based approaches, including optimization, monitoring, and control.

As a primary option for constructing the digital twin of the semi-batch processes, nonlinear models can be classified by many criteria, including a priori information on the target system, stochastic/deterministic, lumped/distributed parameter, continuous/discrete [7]. Among them, the availability of a priori information is used as a criterion throughout this thesis, resulting in two categories: a first-principle model and an empirical model.

A first-principle model, also called a white-box model, is the best choice if we have plenty of prior information on the target system [7]. Model structures and parameters of a fully first-principle model are physically meaningful because all characteristics of the system are understood and modeled mechanistically. However, as a perfect understanding of the target system is often unrealistic for real-world processes, it often uses empirical relations to simulate the complex dynamics of the target system. Considering the modeling of a commercial fed-batch bioreactor, for example, because not all intra-or extra-cellular mechanisms are examined, some part of the system dynamics, such as cell growth or product formation, are modeled with the empirical relations, such as the Monod equation or Luedeking-Piret equation [5, 8, 9, 10]. Based on the selected model structure,

parameter estimation (PE) is required to fit the first-principle model to observations [4, 11].

An empirical model, or black-box model, constructs an input-output relationship using the experimental data [7]. The empirical modeling is preferred when a priori information on the target system is insufficient, because it explains the observed phenomena by using various functions instead. However, it has some disadvantages; it requires a large amount of data of sufficient quality, and it is not reliable to extrapolate outside the explored domain. Frequently used empirical models are as follows: time series models, such as Nonlinear Auto Regressive Moving Average model with eXogenous inputs (NARMAX); Fuzzy models; partial least square (PLS) models and neural networks models. Contrary to the first-principle models, model parameters of the empirical models have no physical meaning in general.

In some literature, a hybrid model, also called a grey-box model, is used to take the advantages of the first-principle and empirical models [12, 13, 14, 15]. As it uses multiple models simultaneously, complex systems showing unknown dynamics can be simulated, such as a slugging phenomenon [16] and oil field operation [17] in the oil and gas industry, hydraulic fracturing [18], polymerization processes [19, 20], bioreactors [21, 22, 23], and pressure swing adsorption [24].

Meanwhile, many batch processes are operated under a specific control policy calculated in advance by an upper-level optimizer or master controller to manage the product specifications [4]. However, this open-loop control strategy is disadvantageous in terms of process modeling, especially for the first-principle model. As it forces the system to follow a certain trajectory, it restricts the amount of informa-

tion gathered from the operation. Moreover, techniques such as the design of experiments are often difficult to implement in production-scale processes. The reason is that they require additional experiments or perturbations, which temporarily deteriorate the product quality [25]. Owing to these limitations, the modeling of a production-scale batch process often encounters an ill-conditioned PE problem, which makes it impossible to estimate all model parameters [26, 27, 28].

It is possible to avoid the ill-conditioned PE by using the hybrid model, as it does not require PE procedures. If the black-box parts of the hybrid model are trained with a sufficient amount of data, the hybrid model can simulate the target system accurately, at least in the domains explored by the observations [29, 30]. However, a detailed analysis of the domains on which the model is valid should be preceded before the practical use of the hybrid model [31, 32].

When modeling the target system with the first-principle models, analyzing estimability or identifiability can be helpful for solving the ill-conditioned PE problem. Estimability refers to the possibility of a unique estimation of model parameters using the measured data [33, 34]. On the contrary, identifiability refers to the possibility of a unique parameter estimation based only on the model structure itself, and therefore, it does not require experimental data [35, 36]. Through these analyses, we can determine a set of estimable parameters. By estimating only the selected parameters, it becomes possible to handle the ill-conditioning problem in the PE process. Then, various model-based methods can be applied to the target systems based on the constructed models.

However, when we want to apply online model-based techniques,

a model-plant mismatch should be considered owing to unknown disturbances and uncertainties [4]. Among various types of possible disturbances, a state disturbance and measurement noise are considered the most frequent. For these disturbances, extended Kalman filter (eKF) and unscented Kalman filter (uKF) have been actively used for the state estimation of the nonlinear models [37, 38, 39]. Besides these well-known disturbances, parametric uncertainties have to be considered when using the nonlinear models, as perfect modeling of the target system is impossible in general. Since the model structure does not reflect all possible dynamics of the system, the model parameters may have drifting values when the system is affected by the unknown disturbances [40]. Except for a few well-studied processes, all the systems simulated with the nonlinear model can have parametric uncertainties for the same reason. The Kalman filter-based estimators can cover this problem by augmenting the model parameters to the states. However, its accuracy drops significantly when the behavior of the drifting parameters is highly nonlinear [41, 42].

Optimization-based estimation methods, such as a moving horizon estimation (MHE), can be a way to address the model-plant mismatch of the online model-based approaches [43, 44]. Given the measured data, these methods generate the optimization problem using the augmented state as the decision variable. Then, the optimization problem is solved at each time step, and the model parameters are updated with the optimal solution to capture the parametric drifts. However, although the optimization-based methods can manage the parameter drift by the iterative update, it may also suffer from limited information. For example, as MHE solves the optimization problem with a moving horizon concept, only the limited number of data

points in the moving window of fixed size are available for each optimization problem, resulting in the ill-conditioned PE problem.

1.2 Scope of the thesis

The main objective of this thesis is to suggest the model-based approaches for the non-stationary semi-batch process under the limited data, which is often encountered in real-world problems. Several types of nonlinear models are tested to handle the ill-conditioning problem in offline and online applications. Proposed methods for each problem are verified with illustrative examples of virtual plants. The summary of the three chapters are below:

- Derivation of the valid domain of dynamic hybrid model and its application in model-based dynamic optimization.
- Parameter estimation method for a simple SDE model and parameter subset selection algorithm using the proposed parameter estimation method.
- Algorithm for multi-rate moving horizon state and parameter estimation, managing the overfitting problem with online parameter subset selection.

The first work proposes a method to identify the valid domain of a dynamic hybrid model and suggests its application to a dynamic optimization problem. The resulting valid domain can be easily introduced into the dynamic optimization problem by reformulating the valid domain into sets of inequality constraints. As the additional constraints restrict the range of exploration by the optimizer, an iterative

framework of model updates is presented to compensate for the limited improvements in the objective function.

The second part suggests an algorithm for selecting and estimating a parameter subset of a stochastic model when only a limited amount of data are available. A target system is represented by stochastic differential equations (SDE) with additive stochastic terms. State disturbances and measurement errors are estimated simultaneously with the model parameters to reduce the effects of uncertainties on the PE. Using the suggested PE method, an algorithm for parameter subset selection (PSS) is proposed based on a mean-squared-error-based PSS method. A virtual fed-batch bioreactor, with 12 model parameters to be estimated, is selected for a numerical illustration. The simulation results show that the proposed method effectively manages the overfitting problem owing to the ill-conditioned PE and improves the model prediction accuracy compared to cases where all of the model parameters are estimated.

The final part is about the online state and parameter estimation problem, incorporating the ill-conditioned PE owing to the limited measurements. An MHE formulation combined with the PSS is suggested in this part. For each horizon of MHE, a scaled parametric sensitivity is calculated online, and a subset of estimable parameters is determined. Then, the selected model parameters are estimated along with the state variables. The proposed MHE formulation handles multi-rate measurements to use as much information as possible from the measured data. The proposed method is illustrated by a virtual fed-batch bioreactor example. The result shows that the proposed method improves the accuracy of model prediction compared to the conventional MHE while maintaining the state estimation per-

formance.

1.3 Outline of the thesis

The remainder of the thesis is organized as follows. In Chapter 2, the backgrounds on parameter and state estimation are introduced, and the overview of estimability analysis is provided. A valid domain for the dynamic hybrid model and its usage in dynamic optimization is suggested in Chapter 3. Chapter 4 proposes an algorithm of PSS for the SDE model of additive stochastic terms. In Chapter 5, an algorithm for multi-rate MHE combined with PSS is suggested for the online state and parameter estimation. Finally, concluding remarks and possible directions for further study are given in Chapter 6.

Chapter 2

Background and preliminaries

2.1 Parameter and state estimation for nonlinear dynamic models

2.1.1 Parameter estimation for nonlinear models

When modeling chemical and biological systems, the target system is defined with specific dynamics, being represented by a collection of differential and algebraic equations. Assuming the system is modeled with a set of ordinary differential equations (ODE), the model can be written in the form

$$\frac{dx(t)}{dt} = f(x(t), u, p, \theta), \quad x(t_0) = x_o \tag{2.1}$$

$$y = h(x, u, p, \theta) \tag{2.2}$$

In the above, $x \in \mathbb{R}^{n_x}$ is the vector of state variables, $x_o \in \mathbb{R}^{n_x}$ is the vector of initial state conditions, $u \in \mathbb{R}^{n_u}$ is the vector of input variables, $p \in \mathbb{R}^{n_p}$ is the vector of known parameters, $\theta \in \mathbb{R}^{n_{theta}}$ is the vector of model parameters to be estimated, and $y \in \mathbb{R}^{n_y}$ is the vector of output variables, respectively. f and h are nonlinear functions that determine the model structure. Given this model, the

vector of measured data at time t_i, denoted as $\hat{y}_i$, can be related to the value obtained from the model of Eq. (2.1) as follows:

$$\hat{y}_i = y(t_i) + \epsilon_i, \quad i = 1, 2, \ldots, n \tag{2.3}$$

where n is the number of measuring times and ϵ_i is the vector of the measurement error.

When a model structure for the dynamics of the system is selected, model parameters should be calculated to complete the modeling procedure. Parameter estimation (PE) refers to the process of determining values of the model parameters by matching the model-calculated outputs to the set of measurements [11]. This problem can be formulated into an optimization problem, minimizing the difference between the measured output and calculated output at each measuring time t_i. The difference is represented by the residual e_i,

$$e_i = \hat{y}_i - y(t_i) \tag{2.4}$$

and the objective function of the optimization problem is defined based on the residuals.

The choice of the objective function is crucial for the result of PE, as it affects both the accuracy of the estimated parameters and their statistical properties. When the model output can be expressed as an explicit function of the variables and parameters, as in Eq. (2.1), there are three most popular objective functions for the PE problem: the Least Squares (LS), maximum-likelihood (ML), and determinant criterion [11].

For 1 data set of n measurements, the objective function for the

LS estimation is the weighted sum of squares of the residuals:

$$C_{LS} = \sum_{i=1}^{n} e_i^\top Q e_i \tag{2.5}$$

where Q_i is an $n_y \times n_y$ weighting matrix for the ith measurement. Selecting a proper Q_i matrix affect the accuracy of PE result. If the covariance matrix of the measured output is available, one can use $Q_i = \Sigma_i^{-1}$.

Assuming that the measured output of the ith measurement is normally distributed and all the experiments are independent, the log-likelihood function is given as a function of the model parameters θ and the measurement covariance matrices Σ_i, conditional on the measured outputs.

$$\begin{aligned} &L(\theta, \Sigma_1, \Sigma_2, \ldots \Sigma_n | \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n) \\ &\qquad = A - \frac{1}{2}\sum_{i}^{n} \log\det(\Sigma_i) - \frac{1}{2}\sum_{i}^{n} e_i^\top \Sigma_i^{-1} e_i \end{aligned} \tag{2.6}$$

where A is constant. The ML estimation is formulated as an optimization problem maximizing the log-likelihood function of Eq. (2.6) over θ and Σ_i. When the covariance matrices are known, Eq.(2.6) can be reduced to be the generalized LS objective function.

$$C_{ML} = \sum_{i=1}^{n} e_i^\top \Sigma_i^{-1} e_i \tag{2.7}$$

Furthermore, assuming $\Sigma_1 = \Sigma_2 = \ldots \Sigma_n = \Sigma$, the ML objective function can be obtained by minimizing the determinant criterion

[45],

$$C_{det} = \det\left(\sum_{i}^{n} e_i e_i^\top\right) \tag{2.8}$$

With the selected objective function, the PE problem is formulated into a nonlinear programming (NLP) constrained by the model equations of Eq. (2.1) and some physical constraints. By solving the NLP, the estimate of model parameters is obtained, which minimizes the objective function.

2.1.2 Moving horizon state and parameter estimation

A moving horizon estimation (MHE) is an optimization-based state estimation method, often utilized to estimate state variables and model parameters of the system online [46]. Unlike full-information estimators (FIE), the MHE formulates optimization problems with a moving horizon assumption, which uses only the measurements within a moving data window of fixed size [43], as shown in Figure 2.1. Therefore, this method is free from the typical "curse of dimensionality" of the FIE. To implement the estimator, one should discretize the continuous model with a proper sampling rate (sr),

$$x_{k+1} = f(x_k, u_k, p, \theta_k) + w_k, \quad x_o = x(t_o) \tag{2.9a}$$

$$y_k = h(x_k, u_k, p, \theta_k) + v_k, \quad k = 0, 1, ..., K. \tag{2.9b}$$

where x_k is a state variable, u_k is a input k, p is the vector of known parameters, and $theta_k$ is a set of model parameters, respectively. In Eq. (2.9a), the model parameters are assumed to change over time,

reflecting the drifting nature of the model parameters. w_i and v_i are additive noise terms assumed to follow zero-mean normal distributions. k is a sampling index, where $sr = t_{k+1} - t_k$, $t_K = t_f$.

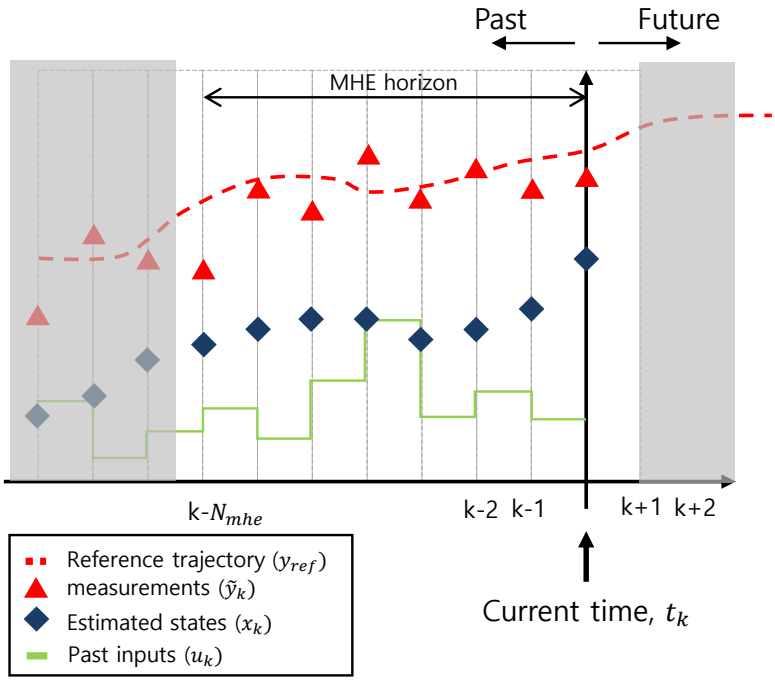


Figure 2.1: Graphical description for moving horizon estimation

2.1.2.1 Formulation of MHE problem

With the discretized model, the online estimation procedure of MHE is equivalent to solving a dynamic optimization problem iteratively with a moving window strategy. At each time step, the dynamic optimization problem of the current data window (horizon) is transformed into nonlinear programming (NLP) and solved with various optimization algorithms [47]. At the kth sampling time, MHE estimating trajectories of the state and model parameters in the current horizon of the size $(N+1)$, $x_{k-N:k|k}$ and $\theta_{k-N:k|k}$, is formulated into the NLP as

$$\min_{x_i,\theta_i,w_i,v_i} \quad C_k(x_i,\theta_i,w_i,v_i) \tag{2.10a}$$

$$s.t. \quad x_{i+1} = f(x_i,u_i,p,\theta_i) + w_i, \tag{2.10b}$$

$$i = L,\ldots,k-1,$$

$$y_i = h(x_i,u_i,p,\theta_i) + v_i, \quad i = L,\ldots,k \tag{2.10c}$$

$$g(x_i,u_i,p,\theta_i) = 0, \quad i = L,\ldots,k \tag{2.10d}$$

$$x_i^{min} \le x_i \le x_i^{max}, \quad \theta_i^{min} \le \theta_i \le \theta_i^{max}, \tag{2.10e}$$

$$w_i^{min} \le w_i \le w_i^{max}, \quad v_i^{min} \le v_i \le v_i^{max}. \tag{2.10f}$$

where subscript $\{i|k\}$ is simplified to i and $L = k-N$. In the kth horizon, the sequences of the measurements ($Y_{L:k|k}$) and the known inputs ($U_{L:k|k}$) are used to estimate a sequence of the states and parameters. Since all the state variables in the kth horizon can be calculated with $x_{L|k}$, $\theta_{L:k|k}$, Eqs. (2.9a), and $U_{L:k|k}$, the NLP becomes a problem of estimating $x_{L|k}$, $\theta_{L:k|k}$, $v_{L:k|k}$, and $w_{L:k|k}$.

The cost fucntion of MHE at the kth sampling time, $C_k(x_i,\theta_i,w_i,v_i)$

consists of two major cost terms, a stage cost and an arrival cost [41, 44]. C_k is often written as

$$C_k(x_i, \theta_i, w_i, v_i) = \left\| \begin{matrix} x_L - \bar{x}_L \\ \theta_L - \bar{\theta}_L \end{matrix} \right\|_{P_L}^2 + \sum_{i=L}^{k} \left\| v_i \right\|_{Q_i}^2 + \sum_{i=L}^{k-1} \left\| w_i \right\|_{R_i}^2 \tag{2.11}$$

where the first term in Eq. (2.11) is the arrival cost and the last two terms are the stage costs. The state costs minimize the errors in the output and the state transition. Inverse of the covariance matrices are often used as weighting factors for the quadratic terms in the stage costs [41]. Meanwhile, the arrival cost summarizes the information of the decision variables of the past sampling times, $i = 0, 1, \ldots, k - N - 1$ [41]. This is necessary for the MHE formaultion since it takes the moving horizon assumption to approximate the full-information estiator (FIE) [44].

2.1.2.2 Calculating arrival cost

Updating the weighting matrix of the arrival cost (P_L) is one of the major topics of the MHE formulation [48, 49, 50, 44, 51]. Some studies use the NLP sensitivity, extracting the covariance information from the linearized KKT conditions of the NLP by using the optimal sensitivity of the solutions [48, 49]. Unscented Kalman filtering (UKF) can be used to calculate the covariance matrix, considering the presence of active constraints for choosing the set of sigma points in the UKF step [50]. An update method for the covariance matrix can also be derived by approximating an ideal arrival cost and linearizing

the discrete model to get an analytic expression for the solution of the DAE consisting of the model [44]. While suggesting an update method based on sampling-based filters, various updating methods are summarized in [51].

In this thesis, the covariance term for the arrival cost is updated via augmented unscented Kalman filtering (aUKF) [50]. Assuming that the initial covariance matrix for the model parameters is calculated in the offline parameter estimation procedure, the model parameters at $i = L$ are augmented to x_L as additional states. Then, the state transition models for the model parameters are assumed to be $\theta_{i+1} = \theta_i + w_{\theta,i}$, and these are put together with the discretized model of Eqs. (2.9a). A sum of quadratic regulation terms for $w_{\theta,i}$ is appended to the stage cost. With this setting, UKF is used to update P_L at each horizon.

2.2 Estimability analysis

Estimability refers to the possibility of obtaining a unique estimation of model parameters by using the existing data on ODEs, differential algebraic equations, algebraic equations, and partial differential equations [33, 34]. It is also denoted as a practical, quantitative, or a posteriori identifiability of model parameters. A sensitivity-based analysis is performed with the measured data, so that an evaluation of the estimability is performed even when a large number of parameters and state variables are involved [33, 34]. However, there are several disadvantages; the performance of the analysis depends severely on the initial estimates of the parameters, and the computation increases rapidly as the number of involved parameters increases [34, 52].

2.2.1 Parameter ranking and subset selection

Parameter ranking (PR) is a technique for evaluating the estimability of model parameters based on measured data [53, 34, 2, 54]. Based on a scaled sensitivity matrix (SSM) [54, 53, 55, 56] or Fisher information matrix [57, 58], various approaches have been suggested for determining the priority of the model parameters in estimation problems. An orthogonalization method applies the Gram-Schmidt algorithm to an SSM [54, 27], and a principal component analysis is also applied to prioritize the model parameters [59, 53]. In some studies, a condition number, also referred to as a collinearity index, is used to determine the rank [55, 56].

Based on the ranked list of model parameters, the parameter subset selection (PSS) creates a reduced model by estimating only the estimable parameters, while maintaining the non-selected parameters at their initial values [34, 60, 3, 61]. As a consequence, the PSS makes it possible to manage the ill-conditioned PE problem originating from the limited data. Insofar as nonlinear models, various approaches have been attempted [3, 61, 62, 63, 64, 60, 27]. In some studies, the optimal parameter subset is selected from the list of ranked parameters based on the mean-squared error (MSE) and MSE-based indices [3, 61, 27, 60]. Methods based on singular value decomposition [62, 63] or QR decomposition [64] first determine an estimable parameter subset, and then repetitively estimate the model parameters and check whether the problem is well-posed.

2.2.2 Sensitivity-based methods

Among the various methods, the sensitivity-based methods is selected in this thesis. These methods use the scaled sensitivity matrix for assessing the estimability of the model parameters.

2.2.2.1 Scaled sensitivity matrix

A sensitivity matrix, denoted by S, of the output variables over the model parameters is calculated by

$$S_{ilm,j} = \frac{\partial y_{ilm}}{\partial \theta_j}, \quad S \in \mathbb{R}^{N_{data} \times p} \tag{2.12}$$

with the measurement (y_{ilm}) and model parameter (θ_j), for $i = 1 \ldots d$ output variables, $l = 1 \ldots n$ measurements, $m = 1 \ldots r$ data sets, $N_{data} = dnr$, and $j = 1 \ldots p$ parameters [34, 11]. This matrix summarizes the influences of the parameters on the model predictions. The sensitivity matrix (S) can be calculated using difference approximations with perturbed parameter values [65], or by solving sensitivity equations [66].

Generally, a sensitivity matrix should be normalized with proper scaling factors to consider differences in orders of magnitude and uncertainties in both measured outputs and the estimated model parameters. In this study, the scaling factors are selected as $s_y = y$ and $s_\theta = std_{\theta_o}$, where std_{θ_o} is the standard deviation calculated using the initial parameter estimate. It makes a scaled sensitivity matrix, Z as

$$Z_{ilm,j} = \frac{\partial y_{ilm}}{\partial \theta_j} \times \frac{s_{\theta,j}}{s_{y,ilm}} = std_{\theta_o} \times \frac{\partial \ln(y_{ilm})}{\partial \theta_j} \tag{2.13}$$

for $\theta_j \neq 0$ and $g_{ilm} \neq 0$.

The scaled sensitivity matrix is calculated numerically based on the model, measured data, and initial estimate of the model parameters. As the model structure is assumed to be fixed, the performances of the methods using the scaled sensitivity matrix depend heavily on initial parameter estimates However, it is difficult to obtain accurate initial parameter estimates in general. Therefore, the calibration process for initial parameter estimates may be necessary to ensure the performance of the estimability analysis, such as PR or PSS

2.2.2.2 Orthogonalization based parameter ranking method

An orthogonalization-based parameter ranking method is a heuristic method for prioritizing estimable parameters. It is the most commonly used method, owing to its simplicity and applicability [2, 54, 67]. The algorithm calculates the column-wise norm of the scaled sensitivity matrix obtained using Eq. (2.13) and selects the column (parameter) with the largest norm. Subsequently, it removes the part of the scaled sensitivity matrix that is linearly dependent on the selected column. The calculation continues until all the model parameters are ranked, or until the matrix norm of the residual matrix is below a user-defined threshold. The priority between the model parameters is determined by sequentially selecting the most estimable parameters and adding them to the ranked parameter list [2, 54]. The outline of the method is presented in Algorithm .1.

Algorithm .1: Orthogonalization Algorithm [2]

0. initialize matrices: $k = 1, Z_1 = Z, X_0 = \varnothing$
1. Calculate the Eulclidean norm of each column in the scaled sensitivity matrix, Z_k. Set the parameter corresponses to the column of the largest value,q_k as the first parameter of the ranked parameter list.
2. Set $X_k = [X_{k-1}, q_k]$
3. Using X_k, calculate the projection of Z_k onto X_k, $\hat{Z}_k$.

$$\hat{Z}_k = X_k(X_k^\top X_k)X_k^\top Z_k$$

and calculate the residual matrix, R_k, which implies that remove X_k all directional elements from Z_k.(R_k is now orthogonal to X_k)

$$R_k = Z_k - \hat{Z}_k$$

4. Set $Z_{k+1} = Rk$
5. Repeat step 1. to 4. until all parameters are ranked or until $X_k^\top X_k$ becomes nearly singular which makes its inverse matrix inaccurate.

2.2.2.3 MSE-based parameter subset selection

The MSE-based PSS methods calculate the MSEs of the model predictions while adding parameters to the selected subset one-by-one based on the list of ranked parameters. Then, the calculated MSEs are used to derive fitness indices for quantifying the improvements in the modeling accuracy corresponding to the size of the parameter subset [60, 3, 61]. The optimal number of parameters is determined based on these indices. The main procedure of this algorithm is summarized in Algorithm 1.

Algorithm .2: Wu's PSS algorithm [3]

0. A ranked parameter list is calculated from the parameter ranking procedure. The algorithm initialized with an empty set of selected parameters.
1. Starting from the top-ranked parameter, the parameters in a selected subset are estimated by solving a proper PE problem while setting unselected parameters to their initial estimates. Parameters are added to the subset one-by-one based on the ranked list, and the process is repeated until all parameters, $\theta \in R^p$, are estimated.
2. Calculate mean squared error, J_k of the model with the top k parameters estimated, $k = 1, 2, \ldots, p$, where J_k is

$$J_k = \sum_j \sum_{n=1}^{N} (\frac{y_{j,n} - \hat{y}_{j,n}}{s_{y_{j,n}}})^2, \quad n = 1, 2, \ldots, N = dnr.$$

3. Compute the critical ratio, $r_{C,k}$ for $k = 1, 2, \ldots, p-1$.

$$r_{C,k} = (J_k - J_p)/(p-k)$$

4. Compute the corrected critical ratio for each k using

$$r_{CKub,k} = max\left(r_{C,k} - 1, \frac{2}{p-k+2} r_{C,k}\right)$$

$$and$$

$$r_{CC,k} = \frac{p-k}{N}(r_{CKub,k} - 1)$$

5. Select the number of parameters to estimate corresponding to the lowest value of $r_{CC,k}$.

Various studies on MSE-based PSS algorithms cover different types of models, but most are deterministic models and lack a detailed description of the PE procedure. Therefore, even when using the existing PSS algorithms, if the target model for the PE contains stochastic terms, such as SDE models, a proper PE technique should be adopted to handle the stochastic models in the PSS procedure.

Chapter 3

Hybrid modeling of semi-batch process and construction of valid domain constraints [1]

3.1 Introduction

In this chapter, a hybrid modeling approach, which models a system with multiple types of models simultaneously [14, 68], is used for modeling the target system under limited data. This type of model is often called a hybrid semi-parametric model or a grey-box model because it comprises both a black-box and white-box model (a first-principles model) [12, 13]. Regarding the hybrid modeling of semi-batch processes, the kinetics are frequently simulated with the black-box model because it is often difficult to select the right structures for the kinetic models owing to various reasons, e.g., cell engineering, involving unknown reactions. Moreover, as the kinetic models involve many model parameters in general, estimating all the model parameters is often unavailable. As the nonlinear equations of unknown or poorly known kinetics can be replaced with a black-box model, the hybrid model accurately simulates the target system without an ad-

[1] This chapter is an adapted version of J. Bae, H. J. Lee, D. H. Jeong, and J. M. Lee, “Construction of valid domain for hybrid model and its application to dynamic optimization with controlled exploration," *Industrial & Engineering Chemistry Research*, vol. 59, no. 37, pp. 16380-16395, 2020. [32]

ditional parameter estimation procedure [69, 21, 70, 71]. In addition, because a user can choose the part to be replaced by the black-box model, the complexity of the model can be easily adjusted [30, 14].

However, the hybrid model suffers from certain limitations. A reliable black-box model cannot be constructed if the amount of data is insufficient [23, 72, 73, 74, 75]. Another disadvantage of this model is its limited range where the confidence is acceptable. This range is defined as a valid domain. The reason why the valid domain of the hybrid model is limited is that it consists of the white-box model, which allows extrapolation based on a prior knowledge about the system, and the black-box model that depends solely on the data used for training.

A valid domain analysis is suggested to define and to calculate the valid domain of the hybrid model with two complementary criteria: convex hull and confidence interval criteria [31]. However, this method defines and examines the valid domain only for static hybrid models. Furthermore, it is difficult to apply the method directly to a dynamic hybrid model because it requires an explicit solution of dynamic model equations. Moreover, the outputs of the dynamic system are often measured using multiple sampling rates. Therefore, the multi-rate measurement issues should also be addressed to determine the valid domain for the sampling time at which not all the output variables are measured.

A method to identify the valid domain of a multivariate dynamic system with nonlinearity is proposed in the following sections. The basic concepts of a dynamic hybrid model and the valid domain of a static hybrid model are described initially. A valid domain is extended to a dynamic hybrid model, and a transformation method is suggested

to recast the valid domain as a set of linear constraints that can be readily used in a dynamic optimization problem. Next, a structure of a dynamic optimization problem comprising valid domain constraints and an iterative modeling-optimization strategy is proposed in the following section. A fed-batch bioreactor is chosen as an illustrative example to demonstrate the efficacy of proposed method. A fed-batch bioreactor has complex nonlinear kinetics and is characterized by changing kinetics as the properties of the microorganisms are continuously modified for performance enhancement over successive operations [8, 9, 10]. These characteristics of the fed-batch bioreactor make it suitable for demonstrating the required performance of the proposed method.

3.2 Hybrid modeling of dynamic system

A hybrid model takes various forms depending on its building blocks, black-box and white-box models [69, 70, 76, 77], and the way these components are connected, in serial or parallel [69, 78, 15, 79]. When the process model (white-box model) is well-known and the prediction performance is limited by the unmodeled effects, the parallel-type hybrid model is used to improve the biased prediction of the process model [13]. In this type of a hybrid model, the black-box and white-box models are uncoupled and calculated in parallel. On the contrary, the serial-type hybrid model is suitable when no particular information about the fundamental mechanisms is available [76]. Instead of establishing a model representation for the unknown behaviors, the black-box model is trained and used to calculate the target behaviors in a serial-type hybrid model.

In this chapter, an embedded hybrid model, which is a serial-type hybrid model [69], is selected as the target model to simulate the semi-batch processes. In this type of a hybrid model, the black-box model also passes its output to the white-box model to compute the overall output. However, the embedded hybrid model differs from the conventional serial hybrid model in that the black and white box models interact in a bidirectional manner. Thus, these models act as a single combined model. Because the white-box model determines the main structure of the overall model in general, the black-box model is considered to be embedded in the white-box model, as illustrated in Figure 3.1c.

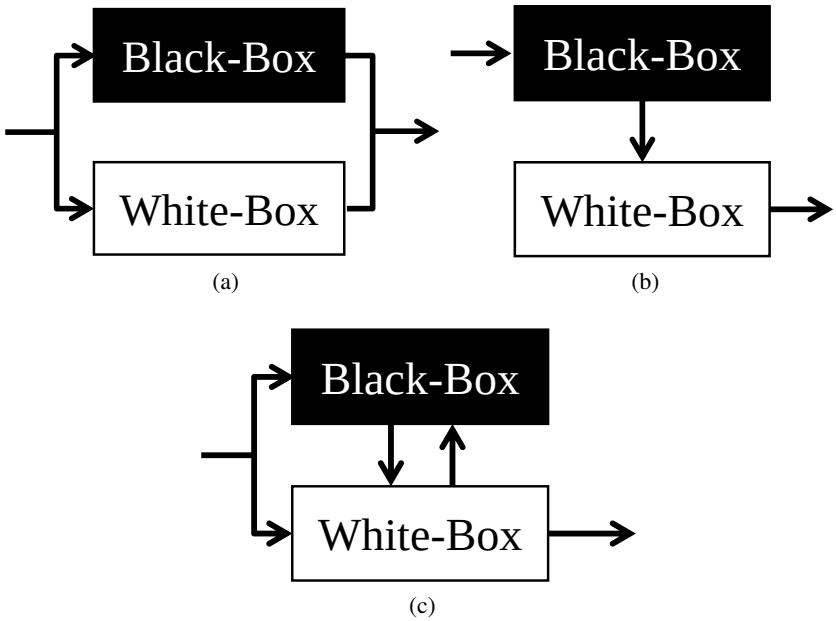


Figure 3.1: Structures of a serial and an embedded hybrid model. (a) Parallel type. (b) Serial type. (c) Embedded type.

Because the black-box model interacts with the white-box model, an embedded hybrid model can provide better extrapolation capabilities than an individual black-box model and better prediction accuracy than an individual white-box model. This study employs a neural networks (NN) for black-box modeling in the embedded structure, and the model for the state transition is represented as

$$\frac{dx}{dt} = f(x, u, p, B), \quad x(t_o) = x_o \tag{3.1a}$$

$$B = NN(x_b) = f_{NN}(x_b, \theta), \quad \theta = (W, b) \tag{3.1b}$$

In Eq. (3.1), x, u, and p represent the vector of state variables, input variables and known parameters of the model, respectively. B is the output of the black-box model which models unknown or poorly known parts of the system, and θ is the parameter vector of the black-box model. x_b is the vector of input variables for the black-box model, which can contain a subset of the state variables or the input variables. Since a neural networks is used as the structure of the black-box model, W and b represent the weights and biases of nodes, respectively.

The unknown part of the target system is generally modeled as a function of the state variables [14, 30]. If the unknown behavior of the system becomes highly nonlinear, the black-box model needs to have a significant number of hidden layers [80]. Accordingly, the number of nodes, layers, and hyperparameters of (deep) neural networks should also be carefully selected to avoid the overfitting problem [81].

The hybrid model can be trained in various ways. One method is to directly train the entire hybrid model. With this approach, the model parameters, i.e., the weights and biases of the black-box model,

are estimated by solving a parameter estimation problem [18, 80, 82, 69]. It is also possible to train the black-box model separately and combine it with the white-box model to generate the hybrid model. Although this is a direct and intuitive approach, it requires pairs of input-output data of the black-box model. The input variable of the black-box model are a set of state variables of the system. When the output of the black-box model is not available, a parameter estimation problem must be first solved to obtain the function values of the black-box model corresponding to the state variables. Because the target model has time-varying terms, the parameter estimation problem transforms into a dynamic optimization problem that minimizes the model prediction error (ϕ) by adjusting a sequence of the function values of the target black-box model ($B_k = B(x_b)|_{t=t_k}$)). Assuming that all the state variables are measurable, the parameter estimation problem can be written as,

$$\min_{B_k} \quad \phi(\tilde{y}_k, y_k) \tag{3.2a}$$

$$s.t. \quad \frac{dx(t)}{dt} = f(x(t), \tilde{u}_k, p, B_k), \quad x(t_0) = x_0 \tag{3.2b}$$

$$y_k = h(x_k, \tilde{u}_k, p) + \epsilon_k, \quad \epsilon_k \sim N(0, \Sigma_k) \tag{3.2c}$$

$$g(x(t), \tilde{u}_k, B_k) = 0, \quad h(x(t), \tilde{u}_k, B_k) \leq 0 \tag{3.2d}$$

$$x_L \leq x_k \leq x_U, \quad k = 0, 1, 2, \cdots, K-1. \tag{3.2e}$$

where $\tilde{y}_k$ is the vector of measured outputs, $\tilde{u}_k$ is the vector of measured values of the input variables, K is the number of sampling points, and ϕ is the cost function minimizing the difference between the measured state ($\tilde{x}$) and the predicted state (x) of each sampling time, respectively.

3.3 Valid domain for dynamic hybrid model

In this section, the concept of the valid domain for the static hybrid models is introduced and extended for the dynamic hybrid models. With regard to the dynamic hybrid model, managing the model-plant mismatch is also the main goal of the valid domain constraints. Two complementary valid domain criteria for the dynamic system, the convex hull criterion and the confidence interval criterion, are derived based on the measured output data and the discretized dynamic hybrid model.

Assuming the dynamic hybrid model is discretized as

$$x_{k+1} = f_d(x_k, u_k, p, B(x_b, \theta)), \tag{3.3}$$

$$y_k = h_d(x_k, u_k, p) + \epsilon_k, \quad \epsilon_k \sim N(0, \Sigma_k) \tag{3.4}$$

where$x_k \in \mathbb{R}^{n_x}$ is the vector of state variables, $u_k \in \mathbb{R}^{n_u}$ is the vector of input variables, $y_k \in \mathbb{R}^{n_y}$ is the vector of output variables at the kth sampling time, and p is a set of known model parameters. $B(x_b, \theta)$ is a black-box model, where x_b is the vector of input variables of the black-box model and θ is a set of parameters of the black-box model. Since the black-box models are used to model unknown dynamics of the system, e.g., reaction kinetics, x_b is often a subset of the state variables of the overall model. When a neural networks is used as the black-box model, $\theta = [W, b]$, weights and biases of nodes, respectively. ϵ_k follows i.i.d normal distribution, and Σ_k is assumed to be known.

3.3.1 Valid domain for static hybrid model

Compared to the pure black-box model, a hybrid model is known to have extrapolation capabilities to a certain degree owing to the use of the first-principle model [15, 83]. But still, the hybrid model is unable to simulate all the domain of the target process, and leaving some valid region in the domain may increase the mismatch between the model and the real process, as the black-box model in the hybrid model does not have prior knowledge about the target system. This makes it necessary to find a valid region of the hybrid model either in the state space or in the input space. A valid domain analysis for a static hybrid model consists of two criteria: a convex hull criterion and a confidence interval criterion [31]. Two complementary criteria are directly used as valid domain constraints in the form of either a nonlinear constraint or a Lagrangian penalty in the optimization problem [31].

3.3.1.1 Convex hull criterion for static system

Because black-box models in a hybrid model are used to simulate unknown or poorly known kinetics of the overall system, the input variables of the black-box models are often the state variables of the overall model. A convex hull criterion can be easily obtained by formulating a convex hull of the measured input variables of the black-box models [31]. Because a static system only requires a constant value for each input variable, the convex hull can be constructed by finding the vertices out of all the points. In addition, the convex hull can be represented with several half-spaces, which are represented as a set of linear inequality constraints on the input variables.

3.3.1.2 Confidence interval for static system

A confidence interval criterion is derived from the concept of inference region, especially a confidence band, in the parameter estimation theory of a hybrid model [26, 31]. It assumes that the measured data are normally distributed and finds the range of the output variables that satisfies a specific level of reliability, for example, an approximate highest posterior density (HPD) band with a significance of α [84].

$$\Phi_{ci,n}(u) = s\sqrt{2\nu_n^\top(u)\Omega^{-1}\nu_n(u)}\sqrt{PF(P, N-P; \alpha)} \tag{3.5}$$

$$s^2 = \frac{\psi(\hat{p})}{N-P} \tag{3.6}$$

where ν_n is the gradient of predicted output of the nth data set (y_n) with respect to $\theta \in \mathbb{R}^P$ evaluated at $\theta = \hat{\theta}$ and it is a function of the input variable (u). In Eqs. (3.5)-(3.6), P is the number of the model parameters, and N is the number of measured data set. $F(P, N - P; \alpha)$ is the upper α quantile of Fisher's F distribution with P and $N - P$ degrees of freedom [85]. $\Phi_{ci,n}$ is a half width of $(1 - \alpha)$ HPD band for the nth expected output of the model with a significance level of α. Ω is Hessian of the determinant criterion(ϕ) evaluated with $\theta = \hat{\theta}$ and the measured data.

In the optimization problem, $\Phi_{ci,n}$ can be used either as a set of inequality constraints or as a Lagrangian multiplier, by adding it to the objective function. When it is used as a nonlinear constraint, it is recast into a set of nonlinear inequalities as $\Phi_{ci,n} - \tilde{\Phi}_{ci,n} \leq 0$, where $\tilde{\Phi}_{ci,n} = \Phi_{ci,n}(\tilde{u})$. The resulting nonlinear inequalities are then added to the optimization problem.

3.3.2 Convex hull criterion for dynamic hybrid model

When one static black-box model is used to model each unknown portion of target dynamics in the dynamic hybrid model, the convex hull criterion for the dynamic hybrid model is derived using the same method as in the static hybrid model. Because the black-box models are static, the convex hull criterion can be formulated using the inputs of the black-box models, x_b, while disregarding the order of its sequences over time, as depicted in Figure 3.2a.

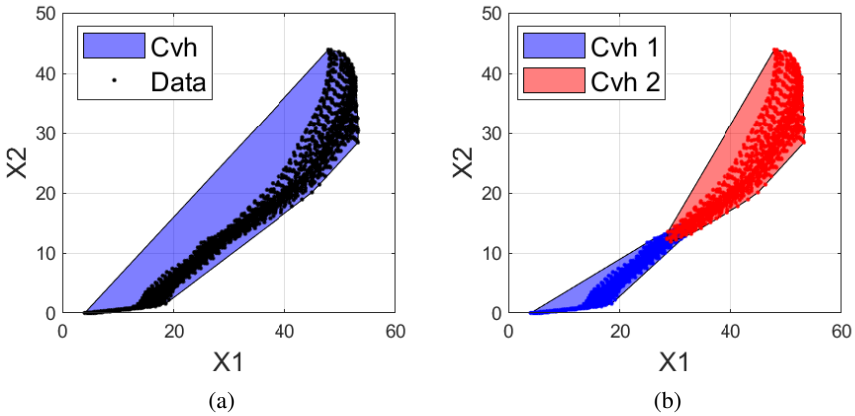


Figure 3.2: Example for convex hull criterion, where $x_b = [x_1, x_2]^\top \in \mathbb{R}^2$

If multiple static models are used for a set of target dynamics, e.g., switching black-box models in the hybrid model, multiple convex hulls are formulated corresponding to each black-box model as in Figure 3.2b. Then, the convex hulls can be used as a set of constraints on x_b in an alternative manner, while switching the black-box models in the dynamic hybrid model. Assuming that one target dynamics of the hybrid model is modeled with J black-box models, the indices of the sampling points, $k = 0, 1, \ldots, K$, are grouped into a set of indices, $I_j, j = 1, 2, \ldots, J$, based on which black-box model is used at the kth sampling point.

3.3.3 Confidence interval criterion for dynamic hybrid model

Unlike static systems, it is impossible to obtain an explicit expression of the output variables in dynamic systems because they are modeled using differential equations. Therefore, the confidence interval criterion of the dynamic hybrid model, $\Phi_{i,k}$, should be derived using implicit equations of the model. In this study, it is derived using the discretized model of Eq. (3.3) and measured data sets.

Assuming that the measurement error is an additive error and follows a normal distribution with zero mean, one can set the cost function, $\psi(\theta)$, for the model parameter estimation as a least-squares function weighted by the inverse of the known covariance matrix of

the measurement error, Σ_k [26].

$$\psi_k(\theta) = [\tilde{Y}_k - Y_k(\theta)]^\top \Sigma_k^{-1} [\tilde{Y}_k - Y_k(\theta)], \tag{3.7}$$

$$\psi(\theta) = \sum_k \psi_k(\theta), \quad k = 0, 1, \ldots, K \tag{3.8}$$

where $\tilde{Y}_k$ and $Y_k \in \mathbb{R}^{N_k \times n_y}$ is a matrix of the measured outputs at the kth sampling time, stacking y_k of each data set, and of the output predicted by the model, respectively. N_k is the number of the measured data sets at k and n_y is the number of the output variables. When the measurement error is unknown, a determinant criterion can be used as an alternative cost function for the parameter estimation problem [26].

$$\psi_k(\theta) = \det\left([\tilde{Y}_k - Y_k(\theta)]^\top [\tilde{Y}_k - Y_k(\theta)]\right) \tag{3.9}$$

If the scale of the measurements varies, $\tilde{Y}_k$ and Y_k should be scaled with proper scaling factors. When the measurement error is assumed to be known, the covariance matrix can be used as a scaling matrix for the measurements as in Eq. (3.7).

With the cost function of Eq. (3.8), the black-box models in the hybrid model are trained to get the best fitting model parameters, $\hat{\theta} \in \mathbb{R}^{N_\theta}$, where N_θ is the number of the model parameters. Since the model parameters to be estimated are the weights and bias of the nodes in the black-box model, N_θ is determined by the total number of layers and nodes of the black-box models.

$$N_\theta = (n_{in}+1)n_0 + \sum_{j=0}^{l+1}(n_j+1)n_{j+1} + (n_{l+1}+1)n_{out} \tag{3.10}$$

where n_{in} is the number of inputs, n_{out} is the number of outputs of the black-box model, n_j is the number of nodes of the jth layer, and $j = 0, \ldots, l+1$, where l is the number of hidden layers while $j = 0$ is an input layer and $j = l+1$ is an output layer, respectively.

With the estimated parameters ($\hat{\theta}$) and the discretized model, the confidence interval criterion at the kth sampling point is derived as a function of the state variables (x_{k-1}) and the input variables (u_{k-1}) of the $(k-1)^{th}$ sampling point.

$$\begin{aligned} &\Phi_{i,k}(x_{k-1}, u_{k-1}, \hat{\theta}) \\ &\quad = s_k\sqrt{2g_{i,k}^\top H_k^{-1} g_{i,k}}\sqrt{N_\theta F(N_\theta, N_k - N_\theta; \alpha)}, \end{aligned} \tag{3.11}$$

$$s_k = \frac{\psi_k(\hat{\theta})}{N_k - N_\theta} \tag{3.12}$$

where $g_{i,k} \in \mathbb{R}^{N_\theta}$ is a gradient of the ith output variables of the model, $y_{i,k}(\theta), i = 1, \ldots, n_y$, and $H_k \in \mathbb{R}^{N_\theta \times N_\theta}$ is a hessian of ψ_k with respect to θ calculated at $\theta = \hat{\theta}$, respectively. F is an upper α-quantile of f-distribution with N_θ and $(N_k - N_\theta)$ degree of freedom.

Assuming that $y_k = h_d(x_k) + \epsilon_k$, $g_k \in \mathbb{R}^{n_y \times N_\theta}$ can be derived

from the discretized model

$$g_k(x_{k-1}, u_{k-1}, \hat{\theta}) = \left.\frac{\partial y_k(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}} = \left.\frac{\partial h_d}{\partial x_k}\frac{\partial x_k(\theta)}{\partial \theta}\right|_{\theta=\hat{\theta}}$$
$$= \left.\frac{\partial h_d}{\partial x_k}\frac{\partial f_d(x_{k-1}, u_{k-1}, \theta)}{\partial \theta}\right|_{\theta=\hat{\theta}} \tag{3.13}$$

while H_k is calculated with the matrix differentiation and the chain rule as in A.1.

The resulting confidence interval criterion, $\Phi_k(\hat{\theta}) \in \mathbb{R}^{n_y}$, is a function of x_{k-1} and u_{k-1}. Since the discretized system is a Markov process, it is natural that assessing the validity of the current state (x_k) with the last state (x_{k-1}) and input (u_{k-1}) given the estimated parameters ($\hat{\theta}$). Except for $k = 0$, Φ_k is calculated for all the sampling points, resulting $K - 1$ nonlinear functions.

The number of parameters increases when a deep neural networks with an increased number of hidden layers is used as the black-box model. Furthermore, the size of the Hessian matrix, H_k, also increases. Consequently, if the a condition number of the matrix becomes large, the inverse calculation of the matrix yields inaccurate results. One way to solve this problem is to find a set of scaling matrices, a left scaling matrix L and a right scaling matrix R, which minimizes the condition number of the scaled matrix, $H_{s,k} = LH_kR$. In this method, L and R of H_k for each data set is formulated as a linear matrix inequality problem (LMIP), especially a generalized eigenvalue problem (GEVP), as in A.2 [1]. Using the scaled matrix, the inverse of H_k can be calculated as $H_k^{-1} = RH_{s,k}^{-1}L$.

With a simple modification of the proposed method, the confidence interval criterion of the dynamic system can be derived using

multi-rate measurements. For simplicity, let us assume that the target system has two types of measurements: slow-rate and fast-rate. Assuming that each measurement rate is a multiple of the sampling rate for the model discretization and considering the sampling rate of the discretization as a grid, all the measured data can be rearranged as depicted in Figure 3.3. The sampling points are classified into three categories based on the measurements available at the time of sampling: both measurements(t_{SF}), only fast measurement (t_F), and only slow measurement (t_S).

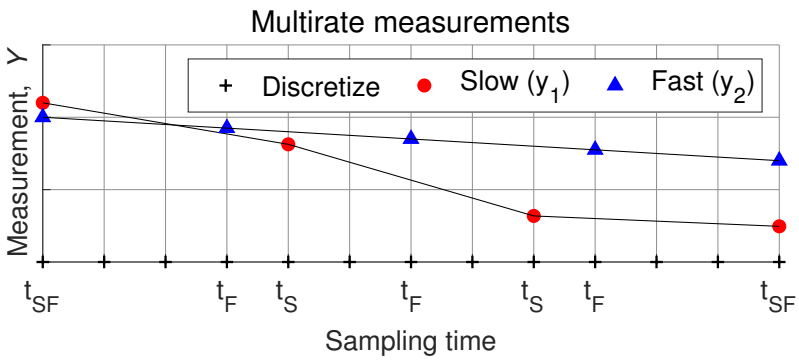


Figure 3.3: Rearranging multi-rate measurements

Let $\hat{Y}_k^n$ be a matrix of the rearranged measurement of the kth sampling time and the nth dataset. If $k \in t_{SF}$, the derivation of the confidence interval is the same as the system measured with a single rate measurement because all the measurements are available. In contrast, if $k \in t_S \cup t_F$, then $\hat{Y}_k^n$ has empty entries depending on the measurement rate of each output. Because the model is available, it is possible to estimate the empty entries in $\hat{Y}_k^n$ using Eq. (3.3) and measured data $\tilde{Y}_k^n$. Then, the confidence interval criterion of the proposed method is derived using the estimated output (Y_k^n) in the place of the measured output ($\tilde{Y}_k^n$). However, N_k should be reduced because some elements of the measurements in the rearranged measurement matrix ($\hat{Y}_k^n$) are not real measured values but are estimated. Therefore, $\hat{N}_k$ is defined as $\hat{N}_k = N_k \times r_k$ and used in Eq. (3.11) instead of N_k, where r_k is the ratio of the number of true measured data to the number of all the data entries in $\hat{Y}_k^n$. It causes the confidence interval to become wider at $k \in t_S \cup t_F$, which means that the predicted output at the sampling point is less accurate than that of $k \in t_{SF}$. Since the number of measured datasets at t_S or t_F should be larger than N_θ, more datasets are required in this case than the one in which all the output variables are measured.

3.3.4 Valid domain constraints for dynamic hybrid model

To apply the two valid domain criteria to the dynamic hybrid model, they are formulated as inequality constraints on $x_k, k = 0, 1, \ldots, K$. The convex hull criterion is explicitly formulated into a set of linear inequalities on x_b by finding half-spaces with facets of the convex hulls. The facets are determined by the convex hull algorithms, such

as the Qhull algorithm [86]. Because x_b is often a subset of the state variables of the hybrid model, the resulting linear inequalities become a set of linear constraints on x_k.

$$A_{ch,j}x_b \leq b_{ch,j} \Leftrightarrow A_{ch,j}x_k \leq b_{ch,j}, \quad k \in I_j \tag{3.14}$$

where $k = 0, 1, \ldots, K$ and $j = 1, 2, \ldots, J$. The number of linear inequality constraints due to the convex hull criterion grows exponentially with the state variables, which are the input for the black-box model. If it becomes computationally prohibitive to use all the linear inequality constraints, a set of linear inequalities at each sampling time can be approximated by an ellipsoid [1].

Meanwhile, the confidence interval criterion, $\Phi_k(x_{k-1}, u_{k-1}, \hat{\theta})$, is used as a nonlinear constraint on the state and input variables at each sampling point,

$$\Phi_k(x_{k-1}, u_{k-1}, \hat{\theta}) \leq \hat{\Phi}_k \tag{3.15}$$

where $\hat{\Phi}_k$ is a band width of the confidence interval calculated with the measured data sets. It means that the predicted output (y_k) is constrained to be in the set $\boldsymbol{Y}_k$,

$$\boldsymbol{Y}_k = \{y_k | y_k \in \cup([y_k^n - \hat{\phi}_k^n, y_k^n + \hat{\phi}_k^n])\} \tag{3.16}$$

$$\Leftrightarrow \quad min(\boldsymbol{Y}_k) \leq y_k \leq max(\boldsymbol{Y}_k) \tag{3.17}$$

where n denotes the nth dataset. Subsequently, Eq. (3.16) can be transformed into two linear inequalities on the output variables, as in Eq. (3.17), adding constraints to the input of the black-box model

indirectly. Therefore, there are two alternative ways to impose the confidence interval criterion onto the model: a nonlinear inequality constraint on the state and input variables at each sampling point, as in Eq. (3.15), or linear inequality constraints on the output variables at each sampling point, as in Eq. (3.17).

3.4 Dynamic optimization with valid domain constraints

3.4.1 Problem formulation

A dynamic optimization problem is formulated with the discretized hybrid model and the two valid domain constraints as

$$\min_{u_k} \quad C(x_k) \tag{3.18a}$$

$$s.t. \quad x_{k+1} = f_d(x_k, u_k, \hat{p}, B(x_b, \hat{\theta})), \quad x_o = x(t_0) \tag{3.18b}$$

$$y_k = h_d(x_k, u_k, \hat{p}) + \epsilon_k, \quad \epsilon_k \sim N(0, \Sigma_k) \tag{3.18c}$$

$$g_d(x_k, u_k, \hat{p}) = 0, \quad l_d(x_k, u_k, \hat{p}) \leq 0 \tag{3.18d}$$

$$x_k \in \Phi_{ch,k}, \quad (x_{k-1}, u_{k-1}) \in \Phi_{ci,k} \tag{3.18e}$$

$$x_{LB} \leq x_k \leq x_{UB}, \quad u_{LB} \leq u_k \leq u_{UB}. \tag{3.18f}$$

where g_d is an equality constraint and l_d is an inequality constraint. Eq.(3.18e) contains two valid domain constraints, where $\Phi_{ch,k}$ and $\Phi_{ci,k}$ denote the sets of variables satisfying the convex hull criterion in Eq. (3.14) and the confidence interval criteria in Eq. (3.15), respectively. As a result, the intersection of the two criteria becomes the actual valid domain of the dynamic hybrid model.

However, applying the intersection as a hard constraint can limit the improvement by optimization since the optimizer is only available

to explore the optimal solution within the small feasible region. To solve this problem while maintaining the validity of the hybrid model, two inequality constraints are defined: the intersection and union of $\Phi_{ch,k}$ and $\Phi_{ci,k}$.

$$\Phi_{\cap,k} = \{(x_k, u_k) | (x_k, u_k) \in \Phi_{ch,k} \cap \Phi_{ci,k}\} \tag{3.19a}$$

$$\Phi_{\cup,k} = \{(x_k, u_k) | (x_k, u_k) \in \Phi_{ch,k} \cup \Phi_{ci,k}\} \tag{3.19b}$$

Then, $\Phi_{\cap,k}$ is added to the dynamic optimization problem in Eq. (3.18) as a soft constraint and $\Phi_{\cup,k}$ replaces Eq. (3.18e) as a hard constraint.

$$\min_{u_k} \quad C'(x_k, u_k) = C(x_k) + W_k \sum_k C_{vd,k}(x_k, u_k) \tag{3.20a}$$

$$s.t. \quad x_{k+1} = f_d(x_k, u_k, \hat{p}, B(x_b, \hat{\theta})), \quad x_o = x(t_0) \tag{3.20b}$$

$$y_k = h_d(x_k, u_k, \hat{p}) + \epsilon_k, \quad \epsilon_k \sim N(0, \Sigma_k) \tag{3.20c}$$

$$g_d(x_k, u_k, \hat{p}) = 0, \quad l_d(x_k, u_k, \hat{p}) \leq 0 \tag{3.20d}$$

$$(x_k, u_k) \in \Phi_{\cup,k} \tag{3.20e}$$

$$x_{LB} \leq x_k \leq x_{UB}, \quad u_{LB} \leq u_k \leq u_{UB}. \tag{3.20f}$$

where $C_{vd,k}$ is a cost function of x_k and u_k, quantifying the violation of the soft constraint ($\Phi_{\cap,k}$). In Eq. (3.20), the optimizer minimizes the cost function $C'(x_k, u_k)$ while satisfying Eq. (3.20e). It allows the optimal solution to violate the soft constraint, $\Phi_{\cap,k}$, but the violation is also minimized by the optimizer. A trade-off between the cost function ($C(x_k)$) and the model validity ($C_{vd,k}$) is managed by adjusting W_k.

3.4.2 Iterative application of overall scheme

If the region defined by $\Phi_{\cup,k}(x_k, u_k) \leq 0$, which is determined by the hybrid model and the measured datasets, does not contain the unknown real optimal operating trajectory of the process, the dynamic optimization of Eq. (3.20) cannot find the real optimum. Furthermore, as the batch process is operated repeatedly, disturbances for which the current model cannot account may change the real optimal operating trajectory. Thus, it is necessary to consider the restricted (x_k, u_k)-domain and the time-varying nature of the actual process when the process is optimized using the hybrid model.

One way to resolve these difficulties is to update the hybrid model and perform model-based optimization in an iterative manner. In this method, the valid domain of each iteration is updated with the newly collected data based on the last optimization result. Therefore, (x_k, u_k)-domains where the past iterations have not been visited can be explored. Moreover, by updating the hybrid model over the batch-to-batch operation, the time-varying nature of the target plant can be considered. Figure 3.4 presents the overall procedure of this method.

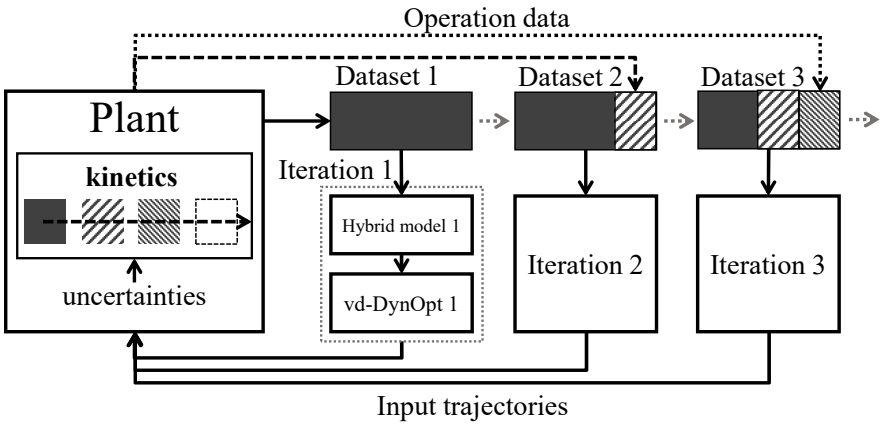


Figure 3.4: A flow diagram of an iterative application of overall scheme

First, it builds a hybrid model with the data collected from a real plant (Dataset 1). Subsequently, the valid domain-constrained dynamic optimization is solved to calculate a new input trajectory. While applying the calculated input trajectory to the real plant, new data are collected for the next iteration. Next, the newly collected data replace the previous dataset from the oldest one. The ratio of the replacement is adjusted (Dataset 2) to update the black-box part in the hybrid model. Since the resulting model is based on a specific dataset used at each iteration, it is possible to make the hybrid model account for the changes in the kinetics by updating the black-box models at each iteration.

3.5 Illustrative example

In this section, a numerical example is presented to demonstrate the applicability of the proposed method. By imposing the valid domain constraints on the hybrid model-based dynamic optimization problem, the proposed method optimizes the objective function while guaranteeing the validity of the predicted result. A virtual plant of a simple fed-batch bioreactor is selected as an example system. This virtual plant is assumed to have the following characteristics: cell growth in the target system is inhibited by high cell and product concentrations, and product generation by the cell is inhibited by a high substrate concentration. Since the growth of cells affects product generation, the concentrations of the cell and product also affect the production rate in the cell. The target of this example problem is to maximize the amount of the final product.

3.5.1 Hybrid model structure and problem statements

The target system is composed of ordinary differential equations (ODEs) describing the dynamics of state variables of the system: cell concentration($X[g/L]$), substrate concentration ($S[g/L]$), product concentration ($P[g/L]$), and reactor volume ($V[kL]$).

$$\frac{dX}{dt} = \mu(X, S, P)X - DX \tag{3.21a}$$

$$\frac{dS}{dt} = -r_{subs}X + (S_{in} - S)D, \quad r_{subs} = Y_{SX}\mu + m_S \tag{3.21b}$$

$$\frac{dP}{dt} = \pi(X, S, P)X - DX \tag{3.21c}$$

$$\frac{dV}{dt} = F_{in} - F_{out} - F_{evap} \tag{3.21d}$$

$$D = F_{in}/V \tag{3.21e}$$

$$F_{evap} = S\alpha_{evap}(\exp(2.5T/100) - 1) \tag{3.21f}$$

where μ is the cell growth kinetics, r_{subs} is the substrate consumption rate, and π is the product formation rate, respectively. The system has two input variables, the inlet and outlet flowrates ($F_{in}[L/hr]$ and $F_{out}[L/hr]$). F_{out} is added to the system to manage the volume of the broth in the reactor. The output model for this system is assumed to be

$$y_i = x_i + \epsilon_i, \quad i = 1, \ldots, 4, \quad \epsilon_i \sim N(0, \sigma_i^2) \tag{3.22}$$

where $[x_1, x_2, x_3, x_4] = [X, S, P, V]$.

All the kinetic equations, μ, r_{subs}, and π, depend nonlinearly on the state variables: X, S, and P. To test the situation where the kinet-

ics are partly known, r_{subs} is assumed to be known as a function of μ as in Eq. (3.21b). Since the kinetic models for μ and π are unknown, neural networks (NN) are used to model these kinetics. Therefore, two NN models are formulated with X, S, and P as the inputs, modeling μ and π as the outputs, respectively. Initial values of the state variables, known model parameters, σ_is, and physical constraints on the input and state variables are listed in Table 3.1.

Parameter	Value[units]	Description
X_o	5[g/L]	cell initial conc.
S_o	20[g/L]	substrate initial conc.
P_o	0[g/L]	product initial conc.
V_o	5.8[kL]	initial broth volume
F^{LB}	0[L/hr]	min. flow rate
F_{in}^{UB}	400[L/hr]	max. flow rate (F_{in})
F_{out}^{UB}	50[L/hr]	max. flow rate (F_{out})
V_{max}	10[kL]	max. volume
T	303.15[K]	reactor Temp.
V_{sp}	0.1[L]	sample volume
S_{in}	300[g/L]	Subs. conc. in feed stock
Y_{SX}	1.85[g S/g X]	yield coeff.
m_S	0.029 [g/g hr]	maintenance (subs.)
α_{evap}	5.24e-4 [L/hr]	evaporation constant
sr_1	0.2 [hr]	sampling rate 1(X,V)
sr_2	0.4 [hr]	sampling rate 2(S,P)
T_{batch}	20 [hr]	batch time
h	0.2 [hr]	sampling rate of discretization
σ_X	0.1 [g/L]	measurement error (X)
σ_S	0.1 [g/L]	measurement error (S)
σ_P	0.1 [g/L]	measurement error (P)
σ_V	0.05 [kL]	measurement error (V)

Table 3.1: Initial conditions and physical constraints for simulation

Fifty sets of simulation data are generated with the randomly perturbed initial states, normally distributed with $\pm$ 20% magnitude of the given initial conditions in Table 3.1. The input trajectories used for generating the fifty simulation data sets are also randomly created based on the fixed reference input trajectory, perturbed by 30% of its magnitude, as shown in Figure 3.5.

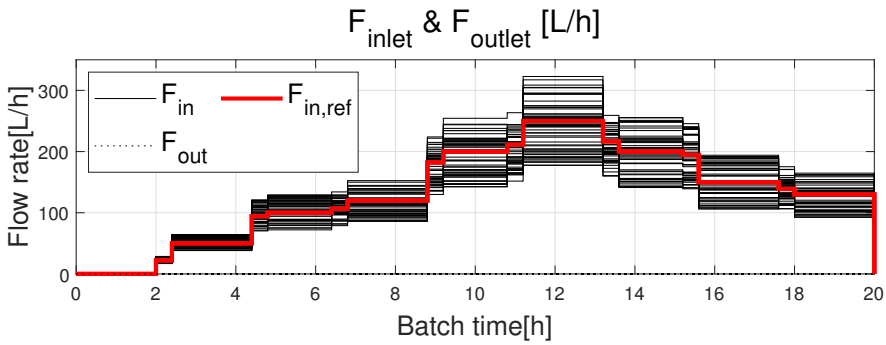


Figure 3.5: Randomly perturbed input trajectories for the data generation

Using the generated simulation data sets, measured responses for each data are calculated via the output model of Eq. (3.22). The batch and sampling times are often different from batch to batch in real processes, but it is assumed that each batch data has already been synchronized for this numerical example. X, S, P, and V are assumed to be measured at different sampling rates: X and V are sampled with a fast measurement (sr_1) while S and P are sampled with a slow measurement (sr_2). The overall simulation was performed using MATLAB 2019a.

3.5.2 Formulate discretized dynamic hybrid model of target system

The first step is to build a discretized dynamic hybrid model of the target system. In this numerical example, it is assumed that μ and π are measured together with the output variables at each sampling time. The assumption on measuring the kinetics of the system, μ and π in this example, is often unrealistic for the modeling problems in reality. However, as mentioned in Section 1.1, there are various methods to directly train the entire hybrid model, not requiring the target value estimates of the black-box model. Moreover, the trajectories of μ and π can also be estimated by solving the trajectory optimization of Eq. (3.2) and used for training the black-box parts of the hybrid model. In this example, however, the detailed process for modeling the dynamic hybrid model is omitted to focus on the valid domain constraints and its application in the dynamic optimization problems.

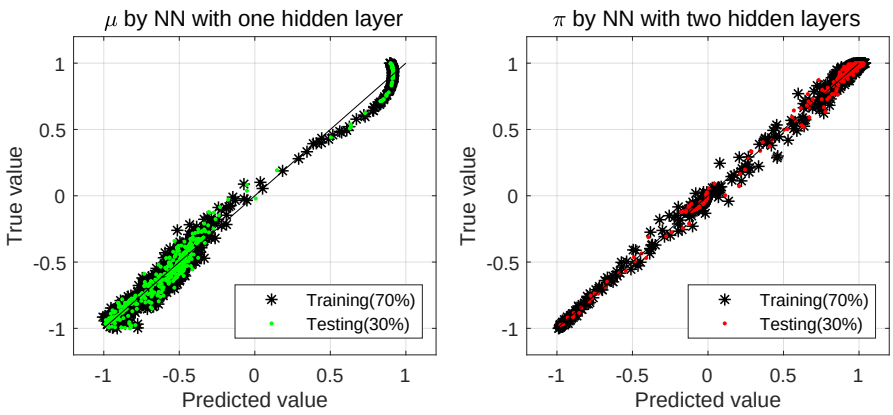
Seventy percent of the measured data were used as the training set, and validation was performed with the rest of the data. Two black-

box models of μ and π are trained as functions of X, S, and P. Two NN models of 3-layers (one hidden layer) for μ and 4-layer (two hidden layers) for π were developed. A *sigmoid* function, $f(x) = 1/(1+\exp(-x))$, was employed as an activation function. The hyper-parameters used throughout the training steps are listed in Table 3.2.

Hyper-parameter	NN for μ	NN for π
# of layers	3(1-1-1)	4(1-2-1)
# of nodes	3-3-1	3-3-2-1
Learning rate, α	0.01	0.02
Activation function	'sigmoid'	'sigmoid'

Table 3.2: Hyper-parameters of the black-box models

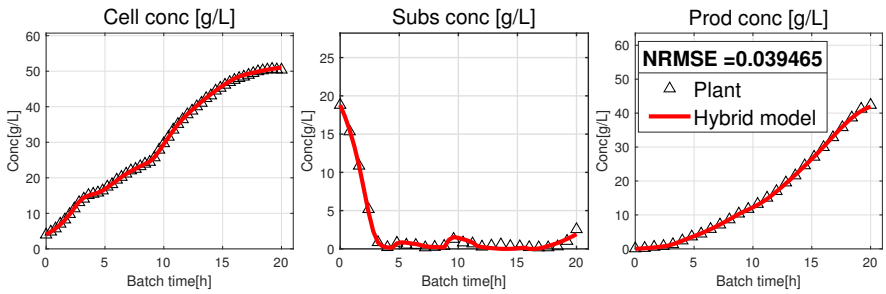
The number of the model parameters ($\theta = [W, b]$) is determined by the number of layers and nodes of each model. For this example, there are 39 parameters of the weights and biases of two black-box models. 16 parameters are required for the μ model: layer 1 ($W_1 \in \mathbb{R}^{3\times3}$ and $b_1 \in \mathbb{R}^3$, 12 parameters in total), layer 2 ($W_2 \in \mathbb{R}^3$ and $b_1 \in \mathbb{R}^1$, 4 parameters in total). The π model requires 23 parameters: 12 for layer 1, 8 for layer 2, 3 for layer 3. Figure 3.6 shows the training results of μ and π model.



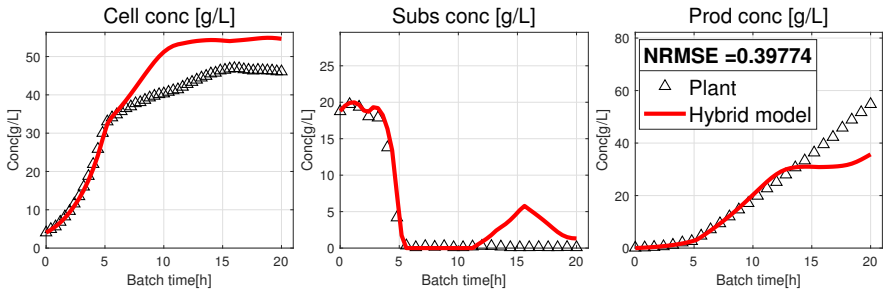
(a) (b)

Figure 3.6: Comparision between true value and predicted value. (a) Traning result - μ, (b) Traning result - π

The continuous model of Eq. (3.21) is discretized with a sampling rate of $h = 0.2\ [hr]$. A complete dynamic hybrid model can be achieved with the discretized model and two black-box models. To verify the prediction performance of the hybrid model, two different cases of randomly generated input trajectories were tested on the plant model and the hybrid model. The two different cases are: 1) an arbitrary input trajectory inside the convex hull of the training data set, and 2) an arbitrary input trajectory outside the convex hull of the given input data. Figure 3.7 shows the simulation results for both cases. For the first case, the hybrid model accurately predicts plant dynamics, as seen in Figure 3.7a. However, the hybrid model shows a poor prediction performance for the second case, as shown in Figure 3.7b. The normalized root mean-squared error (NRMSE) of model prediction is 0.0395 in case 1 and 0.3977 in case 2. This result is due to the extrapolation issue of the black-box models, which demonstrates the necessity for a valid domain analysis of a hybrid model.



(a)



(b)

Figure 3.7: Hybrid model prediction results. (a) case 1 : Random input in the convex hull of ID data, (b)case 2 : Random input out of the convex hull of ID data

3.5.3 Construction of valid domain for dynamic hybrid model

Two valid domain criteria are derived using the dynamic hybrid model. The confidence interval criterion is calculated at all the sampling points, $k = 0, 1, \ldots, 50$. The calculated convex hull criterion is shown in Figure 3.8. Because x_b is $x_b = [X, S, P] \in \mathbb{R}^3$ for both black-box models, the convex hull criterion is plotted in a three-dimensional graphic as in Figure 3.8a. Figures 3.8 (b-d) are two dimensional figures of the convex hull criterion, fixing the X, S, and P axes, respectively. The green lines in the figures indicate the trajectories of the measured data.

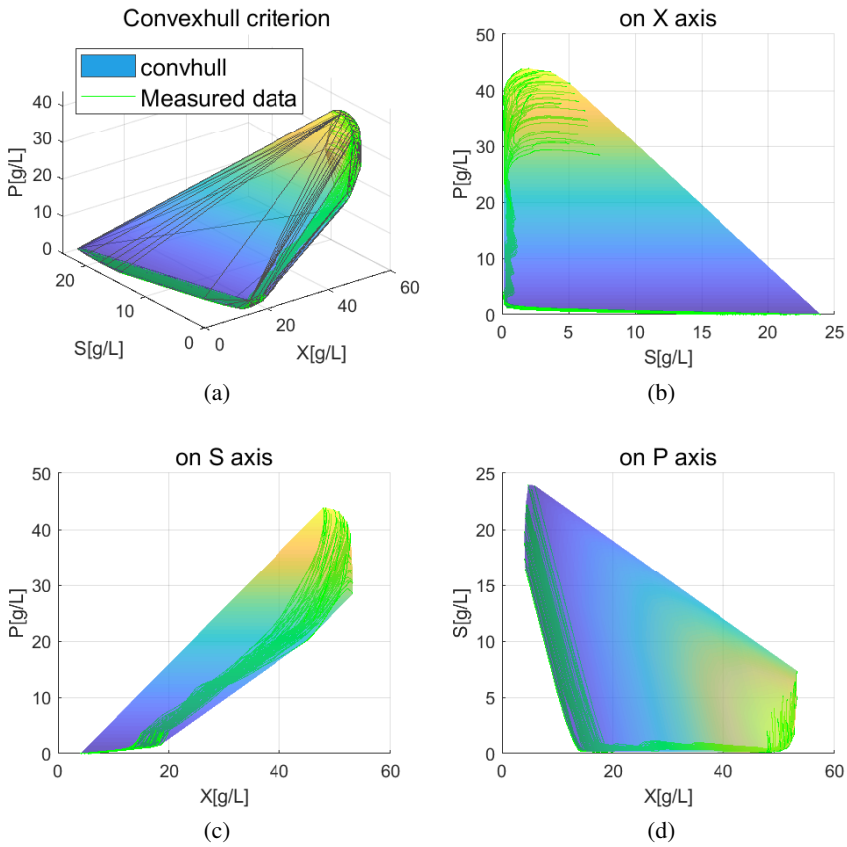


Figure 3.8: Calculated convex hull interval criterion. (a) convex hull of $[X,S,P]$, (b) on X axis, (c) on S axis, (d) on P axis

Unlike the convex hull criterion, the confidence interval criterion (Φ_k) is calculated only at $k = 1, 2, \ldots, 50$ since the kth confidence interval criterion is a function of x_{k-1} and u_{k-1}. Because the problem was assumed to have a multi-rate measurement, the same assumption was considered in the calculation of the confidence interval criterion. The calculated confidence interval criterion is depicted in Figure 3.9.

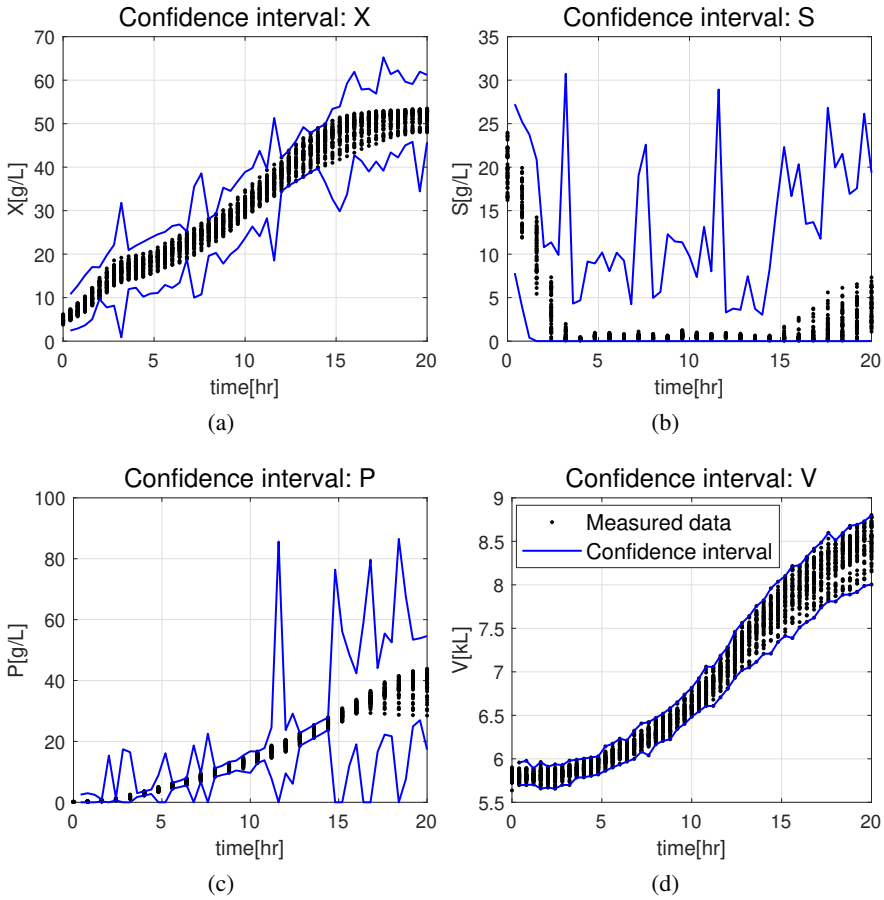


Figure 3.9: Calculated confidence interval criterion for each output variables. (a) Φ_X plot, (b) Φ_S plot, (c) Φ_P plot, (d) Φ_V plot

The blue lines in Figure 3.9 denote the calculated confidence interval criterion (Φ_k) and the black dots indicate the measured variables. The calculated confidence interval of the reactor volume (V) in Figure 3.9d shows that it has a very small Φ_ks compared to other state variables; this is because no black-box model is directly involved in calculating V in the discretized model.

Comparing Figures 3.9 (a-c), one can find out that Φ_k of S and P is wider than that of X. Moreover, Φ_ks, of which sampling points where S and P were not measured, have a larger values than Φ_ks of the other sampling points. One reason is that a smaller N_k is used to calculate Eq. (3.11) and Eq. (3.12) at these points. Moreover, the actual amount of data used to calculate H_k in Eq. (3.11) is less at these sampling points because the unmeasured variables are estimated using the model. This also affects the calculated values of Φ_k at the sampling times of sr_2.

3.5.4 Dynamic optimization with valid domain constraints

Two valid domain criteria of the dynamic hybrid model can be formulated as the inequality constraints as in Eq. (3.19). During optimization, the convex hull criterion results in 300 linear inequalities on X,S, and P at each sampling point, and a total of 15,300 constraints are added to the nonlinear programming (NLP). Using the method of Eq. (3.17), the confidence interval criterion is formulated as eight linear inequalities at each sampling point except $k = 0$, thereby resulting in 400 linear inequalities. Subsequently, two valid domain constraints, $\Phi_\cap$ and $\Phi_\cup$, are calculated from the inequalities. A dynamic optimization problem maximizing the amount of the product is solved

using the dynamic hybrid model and the valid domain constraints as in Eq. (3.20).

Dynamic optimization problems are often solved by transforming them into NLP [47]. Among the various methods, a direct multiple shooting method is used in this study [47]. The resulting NLP is written as

$$\max_{x_k,u_k} \quad P_{end}V_{end} + \sum_k P_k V_{sp} - \sum_k C_{vd,k} \tag{3.23a}$$

$$s.t. \quad x_{k+1} - f_d(x_k, u_k, \hat{p}, \mu(\hat{\theta}), \pi(\hat{\theta})) = 0, \tag{3.23b}$$

$$x_o = x(t_0) \tag{3.23c}$$

$$y_{i,k} = x_{i,k} + \epsilon_{i,k}, \quad \epsilon_{i,k} \sim N(0, \Sigma_{i,k}) \tag{3.23d}$$

$$i = x, s, p, v$$

$$\Phi_{\cup,k}(x_k, u_k) \leq 0 \tag{3.23e}$$

$$x_{LB} \leq x_k \leq x_{UB}, \quad u_{LB} \leq u_k \leq u_{UB} \tag{3.23f}$$

$$F_{in,LB} \leq F_{in} \leq F_{in,UB}, \tag{3.23g}$$

$$F_{out,LB} \leq F_{out} \leq F_{out,UB} \tag{3.23h}$$

where $C_{vd,k} = (\Phi_{\cap,k} - x_k)^\top W_k(\Phi_{\cap,k} - x_k)$ and W_k is assumed as $W_k = diag([0.1, 0.1, 0.1, 0.05])$. The resulting NLP contains 250 decision variables with 16400 inequality constraints, which consist of 15300 convex hull constraints and 400 confidence interval constraints on y_k, 500 upper and lower bounds for decision variables, and 200 continuity constraints for the multiple shooting, respectively. The dynamic optimization problem of Eq. (3.23) is solved on matlab-casADi environment on Matlab 2019a [87], taking 1.9903 sec to be solved

with Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz.

Figure 3.10 depicts the optimized trajectories of the state variables compared with the one reference trajectory from the measured data set.

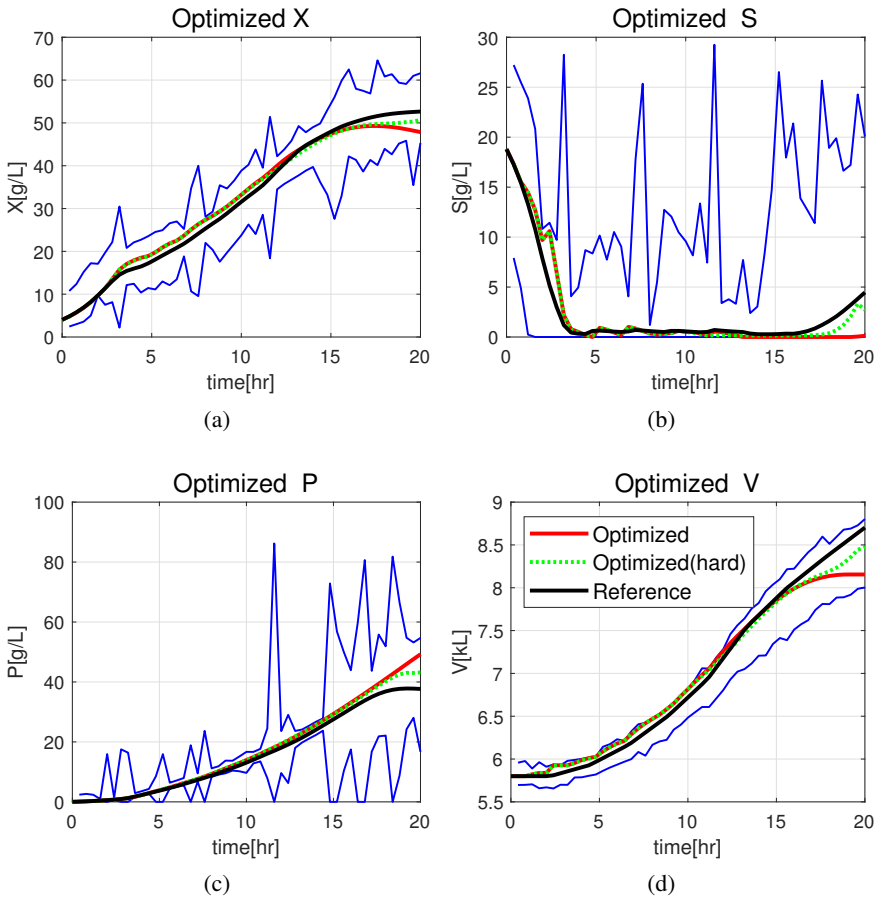


Figure 3.10: Optimized state trajectories on the confidence interval criterion compared with the reference. (a) optimized X [g/L], (b) optimized S [g/L], (c) optimized P [g/L], (d) optimized V [kL]

The final weight of product with the optimal trajectory is 401.63 $[kg]$ while that with the reference is 327.85 $[kg]$. The final volume in the optimal case is therefore less than that of the reference, which means that the optimized input trajectory uses less amount of the feed stock as compared to the reference case. Figure 3.11 depicts the optimal trajectories for the convex hull criterion.

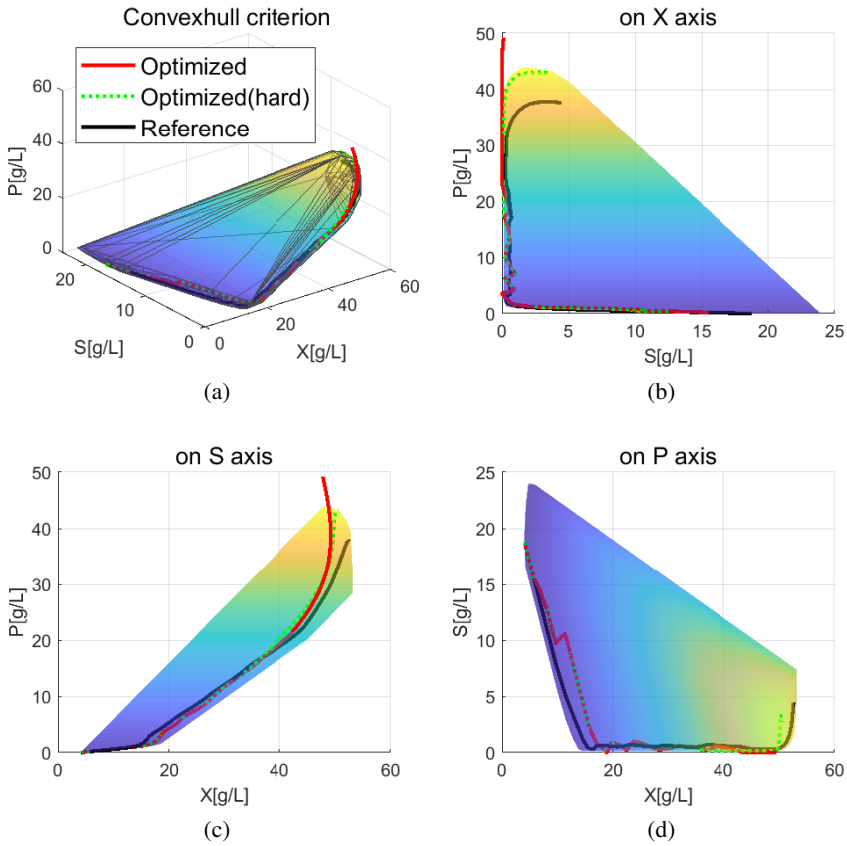


Figure 3.11: Optimal state trajectories on the convex hull criterion compared with the reference. (a) optimized trajectory, (b) on X axis, (c) on S axis, (d) on P axis

Because the constraints of Eq. (3.19a) are added to the optimization problem as a soft constraint, the optimal state trajectory could slightly escapes the convex hull criterion when P approaches 50 $[g/L]$.

The green dotted lines in Figures 3.10-3.11 represent the state trajectories optimized using Eq. (3.18). In this formulation, the constraint $Phi_{\cap,k} \leq 0$ is added to the optimization problem as a hard constraint. Because the optimal solution under this condition cannot leave the intersection of the convex hull and the confidence interval criterion, the final weight of product with the optimal trajectory is 365.65 $[kg]$, which is less than that of the optimal trajectory calculated by the proposed dynamic optimization of Eq. (3.23), i.e. 401.63 $[kg]$.

If the proposed method is effective for managing the model-plant mismatch while performing the input trajectory optimization, the optimized input trajectory, which is calculated with the valid domain constrained dynamic optimization based on the hybrid model, should be valid for the plant model, resulting in the same simulation results in both models. To verify whether the optimized state trajectory of Eq. (3.23) is valid for the plant model, the hybrid model and the plant model are simulated using the optimized input trajectories and the prediction results of the three state variables, X, S, and P, are compared in Figure 3.12.

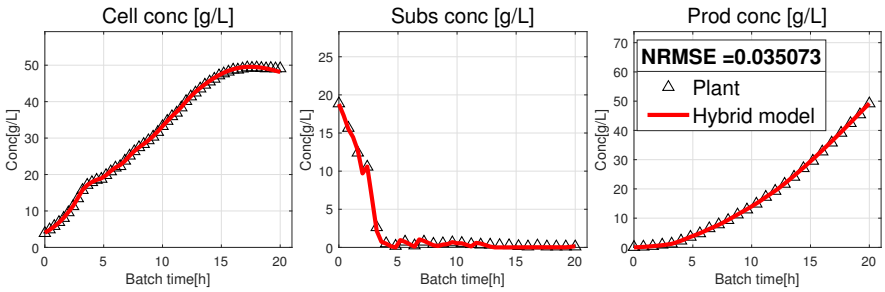


Figure 3.12: Checking the validity of the optimized input trajectory : comparison of the state trajectories of the hybrid model and the plant model under the optimized input trajectory

The simulation result shows that the dynamic hybrid model accurately simulates the target plant of interest, as seen in Figure 3.12. Therefore, we can conclude that the valid domain-constrained dynamic optimization of Eq. (3.23) provides an optimal solution while maintaining the validity of the dynamic hybrid model.

3.5.5 Iterative model update and dynamic optimization

Based on the optimized input trajectory from the first iteration, a new set of operational data is collected in the same manner for the next iteration. For this example, we assumed that all the other conditions are the same as in the iteration 1, except for the input trajectories. The same procedure is applied to the second dataset. The valid domains for each time interval are also derived similar to iteration 1 and used as the constraints in the optimization problem. Figures 3.13-3.14 shows the optimized state trajectories both for the confidence interval and the convex hull criterion.

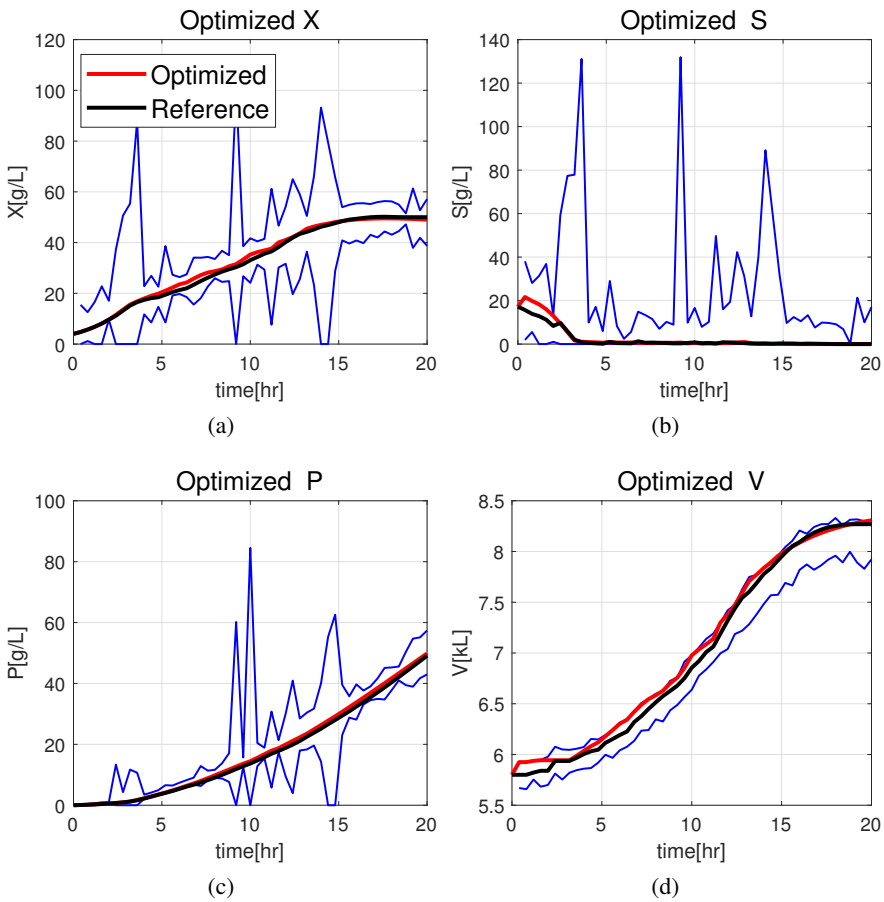


Figure 3.13: Optimized state trajectories on the confidence interval criterion compared with the reference (iteration 2). (a) optimized X [g/L], (b) optimized S [g/L], (c) optimized P [g/L], (d) optimized V [kL]

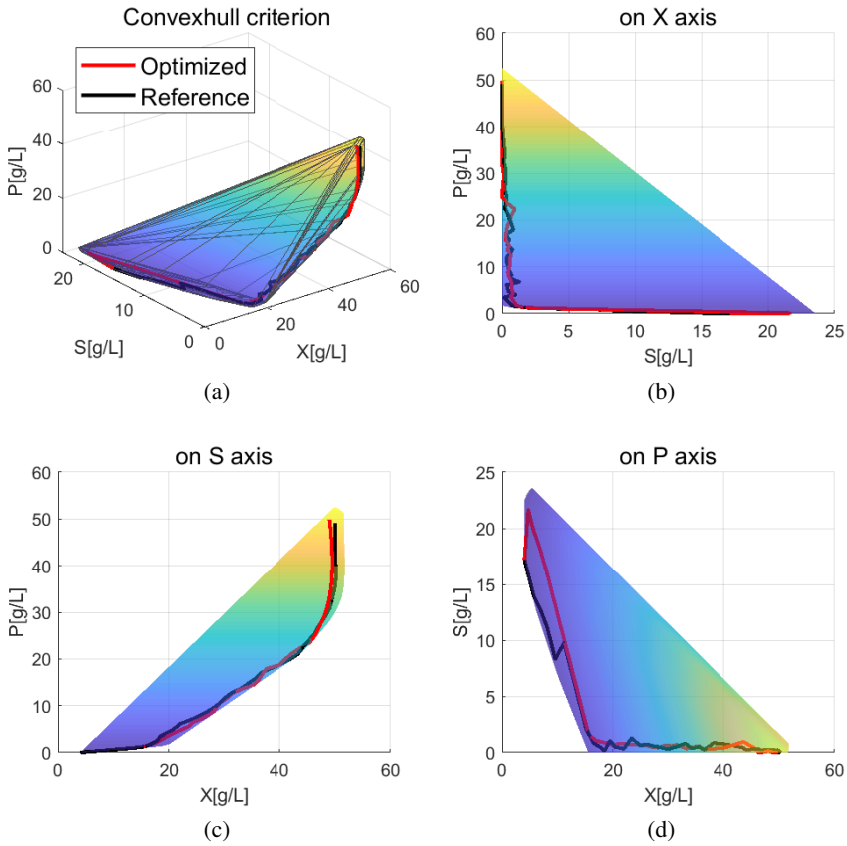


Figure 3.14: Optimal state trajectories on the convex hull criterion compared with the reference (iteration 2). (a) optimized trajectory, (b) on X axis, (c) on S axis, (d) on P axis

The final weight of product with the optimal trajectory of iteration 2 is 414.31 $[kg]$. Even if the amount of improvement in the cost function by iteration 2 is less than that by iteration 1, the iterative application of the proposed method improves the target cost while maintaining the validity of the dynamic hybrid model. Figure 3.15 depicts the accurate simulation of the virtual plant by the dynamic hybrid model and the validity of the optimized input trajectory of iteration 2, similar to iteration 1.

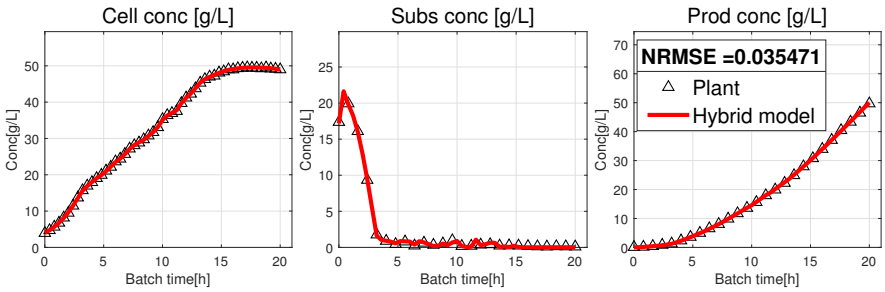


Figure 3.15: Checking the validity of the optimized input trajectory (iteration 2) : compare the state trajectories of the hybrid model and the plant model under the optimized input trajectory

Different results may be obtained when the unknown kinetics of the system modeled by the black-boxes change due to the repetitive batch operations. Even in this situation, we can expect an improved, or at least a similar, performance in a new operation by using the proposed method. Since the model and constraints are updated using the newly collected datasets in every iteration, the proposed method will return an improved result if a better solution is found in a newly explored valid domain defined with the updated model and constraints. Even in the opposite situation, since the proposed method optimizes the input trajectories based on the last optimized input trajectories, it provides a similar result to the last operation. Thus, the proposed methods can find the improved input trajectories for subsequent operations under the changing dynamics of the system using the iterative updates of the hybrid dynamic model and validity constraints.

Chapter 4

Ranking-based Parameter Subset Selection for Nonlinear Dynamics with Stochastic Disturbances under Limited Data [2]

4.1 Introduction

When modeling non-stationary batch processes, we often use a nonlinear model, expecting its capability of simulating complex behaviors of the target system. For example, first-principle models of various systems are nonlinear models, providing a profound analysis when the target system is well-known. However, there exist several issues on the usage of the nonlinear model from a practical perspective; the limited information and difficulty in designing an accurate model structure.

Besides the well-known problem of an ill-conditioned parameter estimation owing to the limited data, selecting a proper model is challenging, except for some well-studied systems [89, 90]. If the selected model structure is not suitable for capturing the characteristics of the system, the model may not simulate the system accurately,

[2]This chapter is an adapted version of J. Bae, D. H. Jeong, and J. M. Lee, "Ranking-based Parameter Subset Selection for Nonlinear Dynamics with Stochastic Disturbances under Limited Data," *Industrial & Engineering Chemistry Research*, vol. 59, no. 50, pp. 21854-21868, 2020. [88]

even if the parameters are estimated properly. Moreover, there also exist uncertainties that are inexplicable by the deterministic model. To overcome this limitation, some have considered using stochastic differential equations (SDEs) that can explain the stochastic effects in the target system [91, 92, 93].

There are many studies on PE methods for the SDE model [92], including a maximum-likelihood estimation (MLE)-based approach [94, 95, 96, 97], expectation-maximization (EM)-based approach [98, 99, 100], and Bayesian applications [101, 102]. Both the MLE and EM-based approaches estimate the model parameters by formulating a cost function based on a likelihood function, and then solving optimization problems. The likelihood functions are calculated by integrating a probability density function (PDF), which depends on the model of the target system. However, it is difficult to calculate the likelihood functions, as a multidimensional integration needs to be performed over the PDF [103, 97]. One way to solve this problem is to apply a Markov chain Monte Carlo (MCMC)-based method to approximate the likelihood function by repetitive sampling [104, 105, 102]. This approach is intuitive, but incurs a massive computational burden.

To avoid this problem, several techniques derive the likelihood function in a closed-form through appropriate approximations [96, 106, 107, 100, 97]. A closed-form expansion for the log-likelihood function is derived by using a multivariate Hermite expansion and its relation to Kolmogorov equations [107]. A quasi-maximum-likelihood estimator is introduced based on the estimation equation [108], and compares it to an infeasible MLE. A Laplace approximation (LA) is used to calculate analytic expressions of the log-likelihood func-

tions. Using the derived log-likelihood functions, a framework for PE of stochastic models is suggested, successfully estimating the model parameters and unknown additive stochastic terms in the model simultaneously [106, 100, 97]. However, the PE problem can still be ill-conditioned depending on the amount and quality of the data, regardless of the suitability of the model structures and corresponding PE methods.

This chapter suggests a method for deriving a model with reduced SDEs by using a ranking-based parameter subset selection to manage the ill-conditioned PE problem under quantitatively rich, but qualitatively limited data collected from a production-scale plant. First, the PE technique for the stochastic models is introduced. The overall PSS algorithm, including the modified PE method for SDE models with additive stochastic terms, is presented in the next section. Finally, the efficacy of the proposed method is illustrated and described based on application to a virtual plant representing a fed-batch bioreactor, followed by concluding remarks in the final section.

4.2 Parameter estimation for SDE models with additive stochastic terms

4.2.1 System modeled with stochastic differential equations

A SDE is a differential equation that contains stochastic processes; its solution, X_t, is also a stochastic process [91]. SDE models are directed to systems influenced by stochastic effects.

A SDE is written as

$$X_t = x + \int_0^t b(s, X_s)ds + \int_0^t \sigma(s, X_s)dW_s, \tag{4.1}$$

or

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t, \quad X_0 = x. \tag{4.2}$$

In both equations, X_t is a stochastic process, b and σ are functions of t and X_t, and W_t is a Wiener process, respectively.

A simple SDE model used in this study takes a constant coefficient for the Wiener process term, $\sigma(t, X_t) = Q$.

$$dx(t) = f(x(t), u(t), \theta)dt + QdW_t, \quad x(0) = x_o \tag{4.3}$$

In the above x is the state variable, u is the input variable, θ is the vector of the model parameters, and Q is the constant covariance matrix for determining the size of the random state disturbances. Assuming that the random state disturbances do not affect each other, Q is reduced to a diagonal matrix.

As the state variables are stochastic processes, it is impossible to obtain exact trajectories for the state variables by solving Eq. (4.3) [91]. Instead, the integrator results in a sample path, i.e., a realization of the stochastic process. The sample paths can be calculated based on the theorem shown in A.3 [91].

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta) + \eta(t), \tag{4.4}$$

$$\mathbb{E}[\eta(t_1)\eta(t_2)] = Q\delta(t_2 - t_1), \quad x(0) = x_o \tag{4.5}$$

where η is a zero-mean white-noise process with constant covariance matrix (Q).

4.2.2 Laplace Approximation Maximum Likelihood Estimation (LAMLE)

A MLE requires solving a multidimensional integral to calculate the cost function,

$$P(Y_m|\xi) = \int P(X_q, Y_m|\xi) dX_q. \tag{4.6}$$

In the LAMLE, the Laplace approximation (LA) is applied to the multidimensional integral of Eq. (4.6) to approximate an analytic solution [109, 110] (A.4).

Assuming that the state disturbances of the system follow Eq. 4.5, the measurement model of the target systems is modeled as

$$Y_m = g(X, U_m, \theta) + \epsilon_m, \quad \epsilon_m \sim N(0, \sigma_j^2) \tag{4.7}$$

where $Y_m \in \mathbb{R}^{n_y}$, $X_m \in \mathbb{R}^{n_x}$, and $U_m \in \mathbb{R}^{n_u}$ are the matrices of measured outputs, state and input variables, respectively. X includes both measured and unmeasured states. The measurement noise matrix, ϵ_m, is also assumed to be a diagonal matrix, meaning that all the measurement noises are independent and identically distributed for $j = 1, \ldots, n_y$.

With these assumptions, the LAMLE algorithm [97] is a MLE-based estimation technique for SDE models that can simultaneously estimate the disturbance intensity (Q) in Eq. (4.3), the measurement noise (Σ) in the output model, and the model parameters (θ). Using an

analytic expression of the integrand of Eq. (4.6) ($P(X_q, Y_m|\xi)$) suggested in [100], solving Eq. (4.6) with LA results in the cost function ($P(Y_m|\xi) = J_{LAMLE}$) as a function of $[\theta, \beta, Q, \Sigma]$.

$$
\begin{aligned}
& J_{LAMLE}(\theta, \beta, Q, \Sigma) \quad = \\
& [Y_m - g(\tilde{X}_m, U_m, \theta)]^\top \Sigma^{-1} [Y_m - g(\tilde{X}_m, U_m, \theta)] \\
+ & (x_{m0} - \tilde{x}_0)^\top S_{m0}^{-1} (x_{m0} - \tilde{x}_0) \\
+ & \int_{t_0}^{t_{end}} [\dot{\tilde{x}}(t) - f(\tilde{x}(t), u(t), \theta)]^\top Q^{-1} [\dot{\tilde{x}}(t) - f(\tilde{x}(t), u(t), \theta)] dt \\
+ & \sum_{l=1}^{n_y} N_l \ln(\sigma_l^2) + q \ln[det(Q)] + \ln[det(H_{\tilde{X}})]
\end{aligned}
\tag{4.8}
$$

assuming that N_l measurements are available for the lth output variable. In Eq. (4.8), θ is a set of the model parameters, β represents the B-spline coefficients, σ_l^2 is the estimated measurement error of the lth output variable, i.e., the lth diagonal element of Σ, and $H_{\tilde{X}}$ is the Hessian of J_{LAMLE} with respect to $\tilde{X}$ that is approximated with Eq. (4.9), respectively. $\tilde{X}_m$ is a matrix of the measured state variables, and $\tilde{x}$ is the vector of the state variables. The symbol $\sim$ over the letters denotes that the variables are approximated by B-spline curves.

In Eq. (4.8), a closed-form solution is required to calculate J_{LAMLE}, as approximated by a b-spline function [111, 103, 97]. Assuming that the deterministic part of the SDE model has a unique solution, the calculation is as follows:

$$
\tilde{x}_i(t) = \sum_{j=1}^{c_i} \beta_{i,j} \phi_{i,j}(t), \quad for \quad i = 1, \ldots, n_x \tag{4.9}
$$

where $\tilde{x}_i(t)$ is the approximated state trajectory of the ith state variable. For the ith state variable, c_i is the number of B-spline basis functions determined by the order of B-spline basis and the number of knots, $\beta_{i,j}$ is the jth B-spline coefficient, and $\phi_{i,j}$ is the corresponding B-spline basis function for $j = 1, \ldots, c_i$ [112]. In a matrix form,

$$\tilde{x}(t) = \Phi(t)\boldsymbol{B} \tag{4.10}$$

where Φ is a matrix of the spline basis functions. $\boldsymbol{B}$ is a matrix of the B-spline coefficients, where each column contains $\beta_{i,j}$, the B-spline coefficients for the ith state variable. The number of knots and order of basis functions need to be selected properly. The hyper-parameters affect the overall precision of the LAMLE algorithm, as they determine the accuracy of the approximated state trajectories.

While estimating the model parameters, the LAMLE algorithm updates Q and Σ iteratively until they satisfy the stopping criterion, i.e., a mean value of the squared relative difference (e_k).

$$e_k = \sum_{i=1}^{n_x} \left(\frac{q_{i,k} - q_{i,k-1}}{q_{i,k}} \right)^2 + \sum_{j=1}^{n_y} \left(\frac{\sigma_{j,k} - \sigma_{j,k-1}}{\sigma_{j,k}} \right)^2 \tag{4.11}$$

The update rules for Q and Σ in the LAMLE algorithm are derived by taking derivatives of J_{LAMLE} with respect to Q and Σ and equating them to zero, respectively. Because the iterative update rule does not guarantee the convergence of Q_k and Σ_k, it might fail to satisfy the stopping criterion in Eq. (4.11).

4.3 Ranking-based PSS for simple SDE model under limited data

A ranking-based PSS method for stochastic models is suggested in this section. Based on the ranked list of model parameters, the proposed method determines the optimal size of the parameter subset while simultaneously estimating the selected parameters, intensity of state disturbances (Q), and measurement errors (Σ).

As the main building block of the proposed PSS method, a PE method for the stochastic models is also proposed in this section. Based on the LAMLE algorithm, the proposed PE method relieves the dependency of the PE algorithm on the b-spline approximation. It also reduces the effects of the batchwise uncertainties on the estimation results and improves the convergence of the algorithm by adopting a learning rate.

4.3.1 Improvement of initial parameter guesses via sampling-based optimization

It is difficult to provide accurate initial estimates for the model parameters in the early stage of the modeling process. As the techniques used in this study, for example PR and PSS, rely heavily on the initial estimates of model parameters, poor initial estimates deteriorate the overall performance. To solve this problem, an additional PE problem of Eq. (4.12) is suggested and solved to provide the improved initial parameter guesses (θ_o^*).

This preliminary PE problem is formulated with the deterministic part of Eq. (4.3) and is solved with a sampling-based optimization

method, e.g., a genetic algorithm (GA) [113], with a weighted least-square (WLS) objective function. Rather than simply falling into local optima near the initial estimates, the sampling-based optimization method explores the decision variable domain with guided sampling strategies to find the optimum (θ_o^*) starting from the poor initial parameter estimates (θ_o^o). The improved initial parameters are also used as the initial parameter for the main PE algorithm.

$$\theta_o^* = arg \min_{\theta_o} J_{\theta_o}, \quad s.t. \quad \theta_o \in [\theta_{o,LB}, \theta_{o,UB}] \tag{4.12}$$

where the range of θ_o is determined based on the distribution of θ_o^o, assuming that θ_o^o is uniformly distributed over predetermined ranges. J_{θ_o} is defined as the WLS of deviations between predictions and measured output variables from the randomly selected data sets, denoted by K. The set K is generated based on a user-defined sampling rate to avoid over-fitting and regulate the computational load.

$$J_{\theta_o} = \sum_{k \in K} \left[\sum_{l=1}^{n_y} \left(\frac{y_{ml}^k - \hat{y}_l^k(\theta_o)}{s_{y_l}} \right)^2 \right] \tag{4.13}$$

where y_{ml}^k is the measured value of the l_{th} output variable of the kth dataset and $\hat{y}_l^k(\theta_o)$ is the predicted output variable using the model and θ_o. The additional PE problem solves the optimization problem in Eq. (4.12) repetitively and collects θ_o^*s, which are used as the initial parameter estimates for all of the techniques to be applied later in this study, including PR and PSS.

As this technique requires an excessive computational load, it is not suitable as the main PE method for the PSS procedure. Moreover,

as it tries to estimate all of the model parameters based on the insufficient information in the limited data, it cannot guarantee the accuracy of the initial parameter estimates (θ_o^*). In the overall algorithm, however, it plays a significant role, as it narrows down the range of the initial parameter estimates.

4.3.2 Calculation of scaled sensitivity matrix and ranked list of parameters

The SDE model in this study is assumed to have a constant coefficient for the Wiener process term, that is, the stochastic term of the model is independent of the model parameters. This assumption makes it possible to calculate a sensitivity matrix for the model parameters of the target models in the same way as calculating the sensitivity matrix for the deterministic ODE models. Eq. (4.3) can therefore be rewritten as

$$X_t = X_o + \int_0^t f(X_s, \theta) ds + \int_0^t \sigma dW_s \tag{4.14}$$

If the output model g is assumed as $g(X_t, U)$, then the sensitivity matrix is calculated as

$$\begin{aligned} \frac{\partial g}{\partial \theta} = \frac{\partial g}{\partial X_t}\frac{\partial X_t}{\partial \theta} &= \frac{\partial g}{\partial X_t}\left[\int_0^t \frac{\partial}{\partial \theta}\left(f(X_s, U_s, \theta)\right) ds + \int_0^t \frac{\partial \sigma}{\partial \theta} dW_s\right] \\ &= \frac{\partial g}{\partial X_t}\int_0^t \left(\frac{\partial f}{\partial \theta} + \frac{\partial f}{\partial X_s}\frac{\partial X_s}{\partial \theta}\right) ds \end{aligned} \tag{4.15}$$

which is the same as the method used in the deterministic models [114]. Then, a ranked parameter list, $\hat{\Theta} = [\theta^1, \theta^2, \ldots, \theta^p]$, is calcu-

lated by applying the orthogonalization algorithm to Z as described in Algorithm 1.

4.3.3 Modified LAMLE algorithm for parameter subset selection

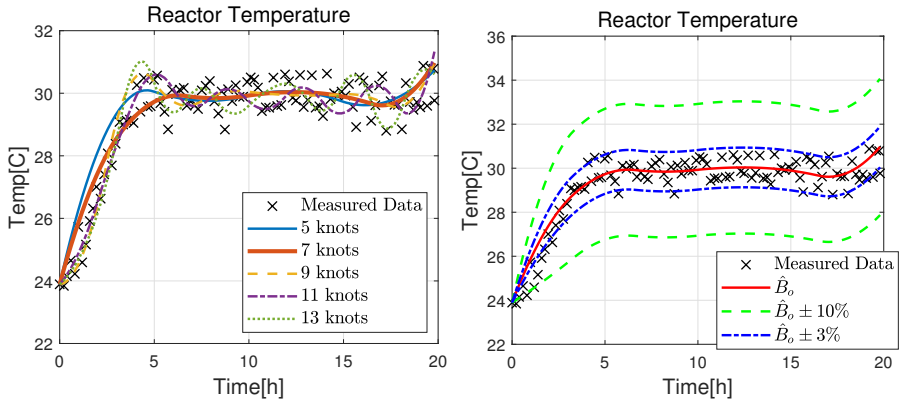
As the PE method for the overall PSS algorithm, a modified LAMLE algorithm is suggested in this section. Based on the LAMLE framework, the proposed PE algorithm is modified to manage the size of the optimization problem while considering multiple datasets simultaneously. A learning rate is also adopted to improve the convergence of the proposed algorithm numerically.

4.3.3.1 Selecting hyper-parameters of B-spline curves

The hyper-parameters of the B-spline curves, the number of knots and order of basis functions in the B-spline curves, critically affects the performance of the PE algorithm. If excessive knots are placed with high order of basis functions, the b-spline curves will approximate the state trajectories very accurately [111]. However, the computational burden for solving the LAMLE-based PE problem becomes intractable, because the number of decision variables of the optimization problem increases in proportion to the number of b-spline coefficients ($\boldsymbol{B}$), which is determined by the number of knots and order of basis functions [115]. The additional decision variables can be considered as dummy variables for the optimization problem, as the B-spline coefficients are not the target of the PE problem. Under this condition, if the number of B-spline coefficients dominates the number of real model parameters to be estimated, the optimal solutions

of the PE problem may fall into the local minima generated by the dummy variables.

In contrast, if a relatively small number of b-spline coefficients are used to adjust the size of optimization problem, the B-spline curves may fail to approximate the state trajectories. If the measured data are overfitted by the b-spline curves, the algorithm may simulate uncertainties as if they were real behaviors of the system dynamics, as shown in the fitting results for 11 and 13 knots in Figure 4.1a.



(a) (b)

Figure 4.1: B-spline curve fitting with de Boor's algorithm: (a) testing a various number of knots and orders, (b) testing a variation of the value of B-spline coefficients.

When solving the optimization problem, the overfitting issue may not occur owing to the state transition models that are added to the optimization problem as constraints. Nonetheless, the number of knots and order should be properly determined to ensure the performance of the PE algorithm and to manage the size of the optimization problem. To find the best hyper-parameter setting, the number of knots and order of the basis functions are predetermined using de Boor's algorithm [112]. Before starting the PE procedure, various combinations of the hyper-parameters are tested offline for each trajectory, as the measured data are fitted with B-spline curves by de Boor's algorithm. While keeping the precision of the fitted curves high, it is preferable to minimize the number of B-spline coefficients.

Meanwhile, the B-spline coefficients obtained by the de Boor's algorithm ($\hat{\boldsymbol{B}}_o$) are used as the initial values for the dummy variables, $\boldsymbol{B}$. As the fitted B-spline curves accurately simulate the real trajectories, the optima of the dummy variables in LAMLE can be assumed to be nearly identical to the initial estimates, $\hat{\boldsymbol{B}}_o$. Therefore, it is reasonable to set the range of the dummy variables as a narrow interval centered on $\hat{\boldsymbol{B}}_o$, to minimize the effects of the dummy variables on the optimization problem. As shown in Figure 4.1b, it is possible to handle the entire range of the measured data, with only a small variation in $\hat{\boldsymbol{B}}_o$.

4.3.3.2 Generalization of Objective functions, J_{gLAMLE}

When using the N datasets to estimate model parameters, the results can be vulnerable to measurement noise if solving the PE problem N times only with a single dataset each time. In this case,

the algorithm may recognize the batchwise measurement noises as real process dynamics, deteriorating the PE accuracy. Using multiple datasets for each PE is a simple way to resolve this problem. By considering the entire collection of datasets as a replay buffer, the random perturbations owing to the measurement noises cancel out each other based on a modified objective function, defined with a set of datasets randomly selected for the PE problem. In this method, the fluctuations introduced by the B-spline assumption are also repressed, as they are normalized when multiple data sets are involved in each PE procedure. Therefore, the parameter estimates are less affected by the measurement errors and B-spline setting. Consequently, the accuracy of the Q and Σ estimates is also improved. Initially, J_{AMLE} and J_{LAMLE} are re-defined for the kth dataset.

$$\begin{aligned} J^k_{AMLE} =& [Y^k_m - g(\tilde{X}^k_m, U^k_m, \theta)]^\top (\Sigma^k)^{-1} [Y^k_m - g(\tilde{X}^k_m, U^k_m, \theta)] \\ &+ (x^k_{m0} - \tilde{x}^k_0)^\top (S^k_{m0})^{-1} (x^k_{m0} - \tilde{x}^k_0) \\ &+ \int_{t_0}^{t^k_{end}} [\dot{\tilde{x}}^k(t) - f(\tilde{x}^k(t), u^k(t), \theta)]^\top \\ &(Q^k)^{-1} [\dot{\tilde{x}}^k(t) - f(\tilde{x}^k(t), u^k(t), \theta)] dt \end{aligned} \tag{4.16}$$

$$
\begin{aligned}
J^k_{LAMLE} =& [Y^k_m - g(\tilde{X}^k_m, U^k_m, \theta)]^\top (\Sigma^k)^{-1} [Y^k_m - g(\tilde{X}^k_m, U^k_m, \theta)] \\
&+ (x^k_{m0} - \tilde{x}^k_0)^\top (S^k_{m0})^{-1} (x^k_{m0} - \tilde{x}^k_0) \\
&+ \int_{t_0}^{t^k_{end}} [\dot{\tilde{x}}^k(t) - f(\tilde{x}^k(t), u^k(t), \theta)]^\top \\
&(Q^k)^{-1} [\dot{\tilde{x}}^k(t) - f(\tilde{x}^k(t), u^k(t), \theta)] dt \\
&+ \sum_{l=1}^{n^k_y} N^k_l \ln((\sigma^k_l)^2) + q^k \ln[det(Q^k)] \\
&+ \ln[det(H^k_{\tilde{X}})]
\end{aligned} \tag{4.17}
$$

where the superscript k indicates that the variables are calculated using the kth data set.

With J^k_{AMLE} and J^k_{LAMLE} of the kth data set in Eq. (4.16) and Eq. (4.17), i.e., the set-wise objective functions, the suggested generalization of the objective functions of the ith data set is given by

$$
\begin{aligned}
J^i_{gAMLE} &= J^i_{AMLE} + \sum_{j \in S^i_{rs}} J^j_{AMLE}\big|_{Q^j = Q^i, \Sigma^j = \Sigma^i} \\
&= f(\boldsymbol{B}^i, \boldsymbol{B}^{j_1}, \ldots, \boldsymbol{B}^{j_{N^i_{rs}}}, \theta)
\end{aligned} \tag{4.18}
$$

$$
\begin{aligned}
J^i_{gLAMLE} &= J^i_{LAMLE} + \sum_{j \in S^i_{rs}} J^j_{LAMLE}\big|_{Q^j = Q^i, \Sigma^j = \Sigma^i} \\
&= f(\boldsymbol{B}^i, \boldsymbol{B}^{j_1}, \ldots, \boldsymbol{B}^{j_{N^i_{rs}}}, \theta, Q^i, \Sigma^i)
\end{aligned} \tag{4.19}
$$

where S^i_{rs} is a set of indices of randomly selected data sets, $j_1, \ldots, j_{N^i_{rs}} \in S^i_{rs}$, not including the ith data set, N^i_{rs} is the size of S^i_{rs} and it is determined by a user-selected sampling rate. $\boldsymbol{B}^l$ is the B-spline coefficients related to the lth data set. The number of overall data sets involved in

the ith iteration is $N_{rs}^i + 1$, the ith data set and N_{rs}^i data sets in S_{rs}^i. The generalized cost functions can be interpreted as a sum of log-likelihood functions,

$$J_{gLAMLE} = \sum_{S_{rs}} \ln p(X_{q,k}, Y_{m,k}|\theta), \quad k = 1, \ldots, N_{rs} \tag{4.20}$$

where N_{rs} is the number of randomly selected data sets.

4.3.3.3 Derivation of $\hat{J}_{gAMLE}$, $\hat{J}_{gLAMLE}$ and Q, Σ update rules

For the ith iteration, however, simply adding the objective functions for all $N_{rs}^i + 1$ data sets as in Eqs. (4.18) and (4.19) causes all of the B-spline coefficients, $\boldsymbol{B}^i$ and $\boldsymbol{B}^j$, $j \in S_{rs}^i$, to become decision variables for the PE problem. In other words, the number of decision variables increases in proportion to N_{rs}^i. This amplifies the disadvantage of the B-spline approximation-based method, causing the number of dummy variables to dominate the real model parameters, and potentially resulting in bad local optima. It also makes the overall computational load grow very quickly.

However, $\hat{\boldsymbol{B}}_o^k$, the initial value of the B-spline coefficients of the kth data set calculated with the de Boor's algorithm, is assumed to be very close to the optimal value of the B-spline coefficients of the kth data set ($\boldsymbol{B}_*^k$). Therefore, it is possible to construct the objective functions to be a function of only the b-spline coefficients of the ith data set ($\boldsymbol{B}_i$) and the model parameters (θ) by using the assumption.

Assuming $\boldsymbol{B}^j = \hat{\boldsymbol{B}}_o^j$ for $\forall j \in S_{rs}^i$, the objective functions become

$$\begin{aligned}\hat{J}_{gAMLE}^i &= J_{AMLE}^i + \sum_{j \in S_{rs}} J_{AMLE}^j \Big|_{\boldsymbol{B}^j = \hat{\boldsymbol{B}}_o^j, Q^j = Q^i, \Sigma^j = \Sigma^i} \\ &= f(\boldsymbol{B}^i, \theta)\end{aligned} \tag{4.21}$$

$$\begin{aligned}\hat{J}_{gLAMLE}^i &= J_{LAMLE}^i + \sum_{j \in S_{rs}} J_{LAMLE}^j \Big|_{\boldsymbol{B}^j = \hat{\boldsymbol{B}}_o^j, Q^j = Q^i, , \Sigma^j = \Sigma^i} \\ &= f(\boldsymbol{B}^i, \theta, Q^i, \Sigma^i).\end{aligned} \tag{4.22}$$

Iterative update rules for Q_n^i and Σ_n^i, where n is the iteration index for the Q and Σ iteration of the ith data set, are derived by differentiating $\hat{J}_{gLAMLE}^i$ with respect to Q^i and Σ^i and equating them to zero, respectively.

$$\begin{aligned}Q_{n+1}^i = &\left(\sum_{j \in S_{rs}} q^j + tr(\Psi^i H_{\beta^i}^{-1} {\Psi^i}^\top)(\Sigma_n^i)^{-1} \right)^{-1} \\ &\left(J_{Q,inner}^i(\boldsymbol{B}^i, \theta^i) + \sum_{j \in S_{rs}} J_{Q,inner}^j(\hat{\boldsymbol{B}}_o^j, \theta^i) \right)\end{aligned} \tag{4.23}$$

$$\begin{aligned}{\sigma_{l,n+1}^i}^2 = &(N_l^i + \sum_{j \in S_{rs}} N_l^j)^{-1} \Bigg(J_{\Sigma,inner}^i(\boldsymbol{B}^i, \theta^i) \\ &+ \sum_{j \in S_{rs}} J_{\Sigma,inner}^i(\hat{\boldsymbol{B}}_o^j, \theta^i) + tr(\Psi_l^i H_{\beta_l^i}^{-1} {\Psi_l^i}^\top)^{-1} \Bigg) \Bigg|_n\end{aligned} \tag{4.24}$$

where $\sigma_{l,n+1}^i$ is the lth diagonal element of Σ_n^i. $J_{Q,inner}^i$ and $J_{\Sigma,inner}^i$

in Eqs. (4.23)-(4.24) are defined as

$$J^i_{Q,inner}(\boldsymbol{B},\theta) = \int_{t^i_0}^{t^i_{end}} \left(\dot{\tilde{x}}^i(t)|_{\boldsymbol{B}} - f(\tilde{x}^i(t)|_{\boldsymbol{B}}, u^i(t), \theta)\right]^\top \left[\dot{\tilde{x}}^i(t)|_{\boldsymbol{B}} - f(\tilde{x}^i(t)|_{\boldsymbol{B}}, u^i(t), \theta)\right]\right) dt \tag{4.25}$$

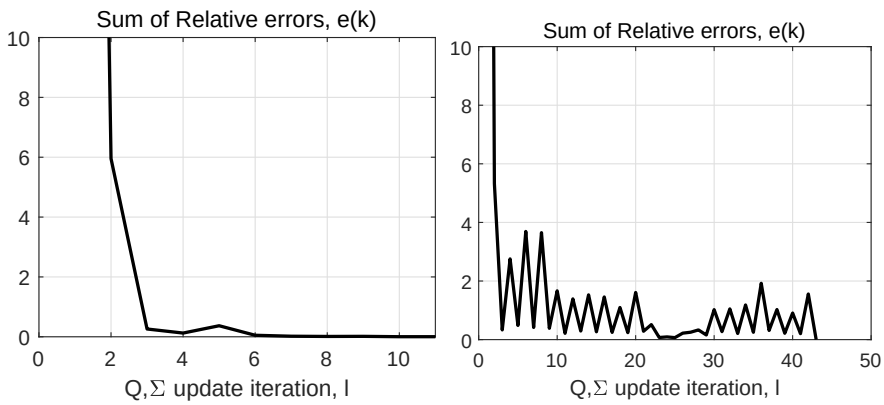
$$J^i_{\Sigma,inner}(\boldsymbol{B},\theta) = \sum_{s=1}^{N^i_l} [y^i_l(t^i_{ml,s}) - g_l(\tilde{x}^i(t^i_{ml,s}), u(t^i_{ml,s}), \theta)]^2 \tag{4.26}$$

As $\hat{J}_{gLAMLE}$ and $\hat{J}_{gAMLE}$ calculate J_{LAMLE} and J_{AMLE} $N^i_{rs}+1$ times, the generalization increases the overall computational load in proportion to the size of the sampled data sets. However, by the assumption of the B-spline coefficients, $\boldsymbol{B}^j = \hat{\boldsymbol{B}}^j_o$ for $\forall j \in S^i_{rs}$, it is possible to keep the number of decision variables at the original level, and it is therefore possible to regulate the increasing computational load while reducing the effects of the batchwise uncertainties the model parameters.

4.3.3.4 Adopting a learning rate (α) into Q, Σ update rules

With the existing Q update rule, it occasionally fails to converge Q, owing to the numerical instability originating from the update rule itself. In each iteration, the value of Q is directly affected by the optimization result. Then, the inverse of the calculated Q is used as weighting matrices in the objective function, and thus also affects the optimization result for the next iteration. Consequently, if the update rule changes the value of Q significantly in a single iteration, the stopping criterion in Eq. (4.11) may not be satisfied. In this case, simply

increasing the number of iterations does not guarantee convergence as the convergence of Q_k sequences has not been proven. Figure 4.2 shows examples of the stopping criterion in both converged and failed cases.



(a) (b)

Figure 4.2: Examples of convergence of Q, Σ iteration in LAMLE algorithm. (a) converged, (b) failed

The convergence of the stopping criterion can be improved by adopting a learning rate, $\alpha \in [0, 1]$, into the update rules.

$$\begin{aligned} \tilde{Q}_{n+1} &= \tilde{Q}_n + \alpha(Q_{n+1} - \tilde{Q}_n), \\ \tilde{\Sigma}_{n+1} &= \tilde{\Sigma}_n + \alpha(\Sigma_{n+1} - \tilde{\Sigma}_n), \quad \alpha \in (0, 1). \end{aligned} \tag{4.27}$$

where Q_{n+1} and Σ_{n+1} are the calculated values of Q and Σ in the nth iteration, and $\tilde{Q}_{n+1}$ and $\tilde{\Sigma}_{n+1}$ are the values of Q and Σ that will be used in the generalized objective functions for the next iteration. This update rule with the constant learning rate becomes an exponential recency-weighted average [116] of the calculated Q_ns and $\tilde{Q}_o$. For example, the Q update rule in Eq. (4.27) can be rewritten as

$$\tilde{Q}_{n+1} = \tilde{Q}_n + \alpha(Q_{n+1} - \tilde{Q}_n) \tag{4.28a}$$

$$= \alpha Q_{n+1} + (1 - \alpha)\tilde{Q}_n \tag{4.28b}$$

$$= \alpha Q_{n+1} + (1 - \alpha)[\alpha Q_n + (1 - \alpha)\tilde{Q}_{n-1}] \tag{4.28c}$$

$$= \cdots$$

$$= (1 - \alpha)^n \tilde{Q}_o + \sum_{i=1}^{n} \alpha(1 - \alpha)^{n-i} Q_{i+1} \tag{4.28d}$$

where more weights are given to recent Q_is than to past Q_is. Meanwhile, introducing α does not affect the value of converged Q and Σ. For Q, suppose $Q_n \to Q$ and $\tilde{Q}_n \to \tilde{Q}$ as $n \to \infty$. Then,

$$\tilde{Q} = \alpha Q + (1 - \alpha)\tilde{Q} \tag{4.29}$$

which can be reduced to $\tilde{Q} = Q$, as α is none zero. The same logic

can be applied to Σ. Adopting α still does not guarantee convergence and may slow the convergence rate, but we can manage the numerical instability owing to the rapid changes in Q and Σ. A detailed description for $gLAMLE$ is illustrated in Algorithm 2.

Algorithm .3: $gLAMLE$ algorithm

initialize: for all N datasets, $i \in \{1, 2, \ldots, N\} = S$

- Determine the number of knots and order of B-spline curves and find $\hat{\boldsymbol{B}}_o$ by using deBoor's algorithm.
- Set the sample size N_{rs}, learning rate α, stopping criterion e_*
- initialize $\bar{Q}_o^i = Q_o^i = Q_o$ and e_o^i

for $i = 1, \ldots, N$ **do**

  (0) Set $n = 0$

  (1) Set S_{rs}^i by randomly selecting N_{rs} datasets from $S - \{i\}$

  while $e_n > e_*$ **do**

    (2) With $\bar{Q}_n^i$ and $\hat{\boldsymbol{B}}_o^j$ of $j \in S_{rs}^i$, formulate $\hat{J}_{gAMLE,n}^i$ with Eq. (4.21)

    (3) Find $\hat{\theta}_n^i$, $\hat{\boldsymbol{B}}_n^i$ by minimizing $\hat{J}_{gAMLE,n}^i$ and get H_{β^i}

    (4) Calculate Q_{n+1}^i with $\hat{\theta}_n^i$, $\hat{\boldsymbol{B}}_n^i$, H_{β^i}, and Eq. (4.23)

    (5) Update $\bar{Q}_{n+1}^i$ with Eq.(4.27) and check the stopping criterion e_n with Eq.(4.11)

  end

end

4.3.4 MSE-based PSS of SDEs model using gLAMLE

The MSE-based PSS method consists of two steps: 1) solving PE problems iteratively with the $gLAMLE$ algorithm based on the ranked list of parameters; and 2) calculating the MSE-based indices. Let $\hat{\Theta} = [\theta^1, \theta^2, \ldots, \theta^p]$ be the ranked list calculated using the PR method. The PE problem for the k_r^{th} iteration, where $k_r = 1, 2, \ldots, p$, is solved with the first k_r parameters of $\hat{\Theta}$ and the b-spline coefficients as the decision variables of the optimization problem while fixing the non-selected parameters to their initial values, θ_o^*. Therefore, the k_r^{th} parameter subset is constructed as

$$\Theta_{k_r} = \left[\theta^1, \theta^2, \ldots, \theta^{k_r}, \theta_o^{k_r+1}, \ldots, \theta_o^p\right]. \tag{4.30}$$

where $k_r = 1, 2, \ldots, p$. Through this procedure, the estimated parameter subset, $\hat{\Theta}_{k_r} = \left[\hat{\theta}^1, \hat{\theta}^2, \ldots, \hat{\theta}^{k_r}, \theta_o^{k_r+1}, \ldots, \theta_o^p\right]$, is sequentially calculated for all k_r. Q_{k_r} and Σ_{k_r} are calculated simultaneously using Q_{k_r-1} and Σ_{k_r-1} as the initial values for the k_r^{th} iteration.

J_{k_r} is defined as the WLS of deviations between predictions and true measured outputs and used as a performance criterion to evaluate the prediction accuracy of the model corresponding to each $\hat{\Theta}_{k_r}$.

$$J_{k_r} = \sum_{k=1}^{N_{tot}} \left[\sum_{l=1}^{n_y} \left(\frac{\bar{y}_{ml}^k - \bar{y}_l^k(\hat{\Theta}_{k_r})}{s_{y_l}} \right)^2 \right] \tag{4.31}$$

where N_{tot} is the total number of data sets. As in Eq. (4.13), a proper scaling factor, s_{y_l}, is essential for each state variable, to cancel out the differences originating from the magnitude of each variable..

With J_{k_r} and $\hat{\Theta}_{k_r}$, the indices, r_{C,k_r}, r_{CKub,k_r}, and, r_{CC,k_r}, are

calculated to conduct the MSE-based PSS. The optimal size of the parameter subset, r_k^*, is determined using the r_{CC,k_r}: the optimum of r_k minimizes r_{CC,k_r}. Then, $\hat{\Theta}_{k_r^*}$, $Q_{k_r^*}$, and $\Sigma_{k_r^*}$ are considered as the optimal estimates of the model parameters, state disturbance intensities, and measurement noises, respectively. Figure 4.3 summarizes the overall procedure of the proposed PSS method.

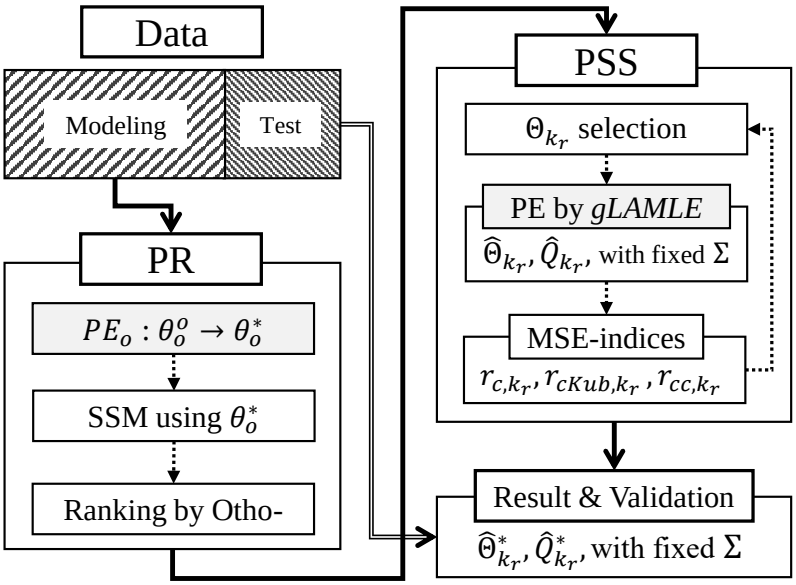


Figure 4.3: Flow-chart for the proposed algorithm

4.4 Numerical illustration: Modeling a fed-batch bioreactor under limited data

The proposed method is applied to a virtual plant of a fed-batch bioreactor to illustrate the efficacy of the proposed parameter subset selection method. The target system is simulated with a simple SDE model consisting of a deterministic unstructured model part and stochastic parts of random disturbances on the state variables (Q) and output variables (Σ). As the target system is a production-scale plant, the available data are assumed to be obtained from noisy measurements of the virtual plant, as operated with fixed input trajectories and fixed initial states. For each batch, the input trajectories and initial states are randomly perturbed from the fixed values with zero-mean Gaussian errors whose standard deviations are assumed to be 20% and 10% of the magnitudes, respectively, to simulate the deviations between the batches. As the datasets are generated based on similar input trajectories, being slightly perturbed from the fixed input trajectories, the PE problem becomes ill-conditioned due to the insufficient information in the datasets, even though the quantity of datasets is assumed to be sufficient. Moreover, the initial values of the parameters (θ_o^o) are assumed to be uniformly distributed over 10^{-1} to 10 times the scale of the variables to simulate a situation where relatively little information regarding the parameter initial guesses is available. For example, if the true value of θ_1 is 0.31, then $\theta_{1,o}^o \in [0.01, 1]$.

4.4.1 Description on fed-batch bioreactor model

There are four state variables, cell concentration (X $[g/L]$), substrate concentration (S $[g/L]$), broth volume (V $[kL]$), and tempera-

ture (T $[C]$), and there are 12 model parameters to be estimated. The true values and initial distribution of the model parameters are listed in Table 4.1.

#	Parameter	True value	Initial range	Unit
1	$C_{\rho b}$	4.2	0.1-10	kJ/kgK
2	$C_{\rho s}$	5.9	0.1-10	kJ/kgK
3	α_{evap}	$5.24 \cdot 10^{-4}$	10^{-5}-10^{-3}	L/hr
4	U_{Jacket}	36	1-100	kWm^2/K
5	Y_{qx}	25	1-100	kJ/g
6	U_h	12	1-100	kWm^2/K
7	U_c	12	1-100	kWm^2/K
8	μ_{max}	180	10-1000	h^-1
9	K_s	1.5	0.1-10	g/L
10	E_g	$1.488 \cdot 10^4$	10^3-10^5	J/mol
11	Y_{xs}	0.54	0.01-1	g/g
12	m_s	0.029	10^{-3}-0.1	g/ghr

Table 4.1: Parameters to be estimated: true values and initial range

The target system is controlled by four inputs, the inlet (F_{in})/outlet (F_{out}) and hot (F_h)/cold (F_c) feed rates $[L/hr]$, respectively. F_{in} and F_{out} are assumed to follow a certain pattern with $\pm$ 20% random perturbations. Meanwhile, F_h and F_c are assumed to be controlled by a PID temperature controller. The SDE model of the target system is

$$dX(t) = [\mu_g X(t) - DX(t)]\, dt + q_X dW \tag{4.32}$$

$$dS(t) = \left[-\left(\frac{1}{Y_{xs}} \cdot \mu + m_s \right) X(t) + D\left(S_{in} - S(t)\right) \right] dt + q_S dW \tag{4.33}$$

$$dV(t) = 1/1000 \cdot [F_{in} - F_{out} - F_{evap}]\, dt + q_V dW \tag{4.34}$$

$$\begin{aligned} dT(t) = \Bigg[& \frac{1}{1000 \cdot V(t)} \frac{1}{C_{\rho_b}\rho_b} \left(F_{in} C_{\rho_s} \rho_f (T_s - T(t))\right. \\ & - \Delta H_{evap} \rho_w F_{evap} + F_h \alpha_h U_h (T_h - T(t)) \\ & + F_c \alpha_c U_c (T_c - T(t)) + U_{Jacket} A_{Jacket} (T_{air} - T(t)) \\ & \left. + \mu X(t) Y_{qx} \cdot 1000 \cdot V(t) \right] dt + q_T dW \end{aligned} \tag{4.35}$$

$$\mu_g = \mu_{max} \frac{S(t)}{S(t) + K_s} \exp\left(\frac{-E_g}{R \cdot (T(t) + 273.15)} \right) \tag{4.36}$$

$$F_{evap} = S(t) \alpha_{evap} \exp\left(\frac{2.5T(t)}{100} - 1 \right) \tag{4.37}$$

$$D = \frac{F_{in}}{1000 \cdot V(t)} \tag{4.38}$$

where $Q = diag[q_X, q_S, q_V, q_T]$. Assuming that all the state variables are measurable with additive measurement noises, the output model

is

$$
\begin{aligned}
Y_{X,m_1} = X_{m_1} + \epsilon_{X,m_1}, \quad Y_{S,m_2} = S_{m_2} + \epsilon_{S,m_2} \\
Y_{V,m_3} = V_{m_3} + \epsilon_{V,m_3}, \quad Y_{T,m_4} = T_{m_4} + \epsilon_{T,m_4}
\end{aligned}
\tag{4.39}
$$

where the subscript $m_.$ indicates a sampling point, $\epsilon_{.,m_.}$ is a zero-mean measurement error with a covariance $\sigma_.^2$, and $\Sigma = diag\,[\sigma_X^2, \sigma_S^2, \sigma_V^2, \sigma_T^2]$. Tables 4.2, 4.3 list the known parameters and operation conditions of the target system, respectively.

#	Name	Value	Unit
1	ΔH_{evap}	2430.7	kJ/kg
2	ρ_w	5.9	kg/m^3
3	ρ_s	$1.54 \cdot 10^4$	kg/m^3
4	R	8.314	$J/Kmol$
5	A_{Jacket}	105	m^2
6	α_h	2451.8	kJ/m^3
7	α_c	2451.8	kJ/m^3

Table 4.2: Table for known parameters

#	Name	Value	Unit
1	S_{in}	200	g/L
2	T_h	40	C
3	T_c	20	C
4	T_s	25	C
5	T_{air}	20	C
6	T_{sp}	30	C

Table 4.3: Table for operating conditions

50 data sets were generated using the SDE model. To simulate a general situation, the sampling rates for the state variables are assumed to be $[X, S, V, T] = [0.2, 0.5, 0.2, 0.2][hr]$, resulting in multi-rate measurements.

4.4.2 Parameter Ranking with θ_o^*

4.4.2.1 Sampling-based PE to improve initial parameter guesses

Given the parameter initial guesses (θ_o^o) in Table 4.1, θ_o^* is calculated via the sampling-based optimization problem in Eq. (4.12) with the deterministic part of the model. GA is used as the optimization method, with 100 generations and 200 populations. Figure 4.4 shows a boxplot of the resulting θ_o^* values calculated for all of the datasets.

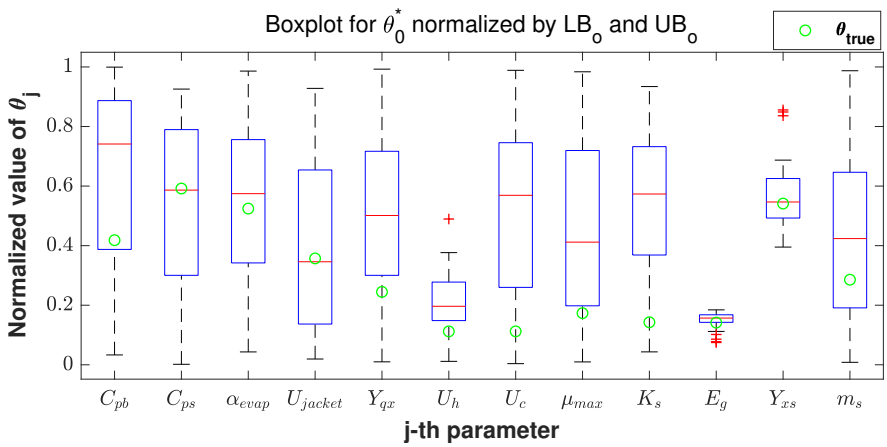


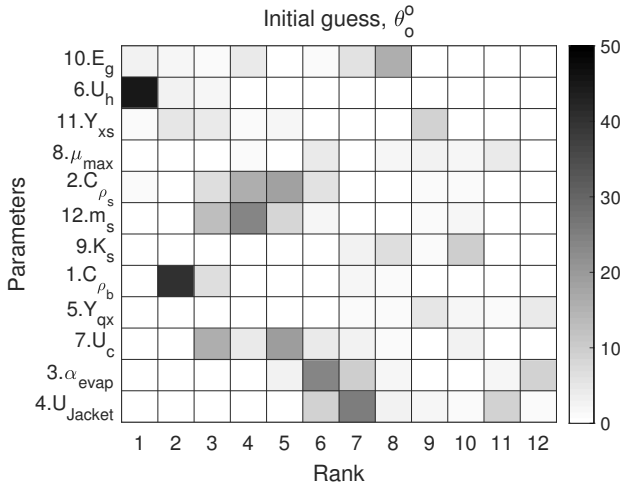
Figure 4.4: Boxplot for the distribution of θ_o^*. o : the true value, — : the median, — : the 25th and 75th percentiles, - - : whisker, + : outliers

Although it is not effective for all the model parameters, the GA algorithm improves the initial guesses for three parameters, i.e., θ_6, θ_{10}, and, θ_{11}. The calculated sets of θ_o^* are used as the initial estimates of the model parameters for the rest of the PSS procedure.

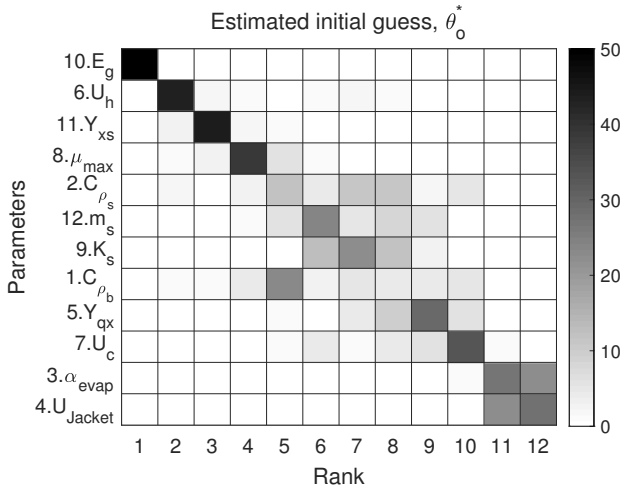
4.4.2.2 Applying Orthogonalization algorithm and the resulting ranked list of the model parameters

The scaled sensitivity matrix in Eq. (4.15) is calculated numerically, using a difference approximation with perturbed parameter values. While perturbing the ith column corresponding to the ith model parameter, the other parameters are fixed at their initial values, $\theta_{o,j}^*$. Two rank estimations are performed with θ_o^o and θ_o^*, to show the effectiveness of the improved initial parameter estimates, θ_o^*, on the PR result.

As the true values of the model parameters are available for this numerical example, it is possible to calculate the true rank of the model parameters. The results of the two rank estimations are rearranged in order of the true rank and plotted into heatmap graphs to compare the accuracy of each rank estimation. If the rank estimation is accurate, relatively more data will be located at the diagonal of the heatmap graph.



(a)



(b)

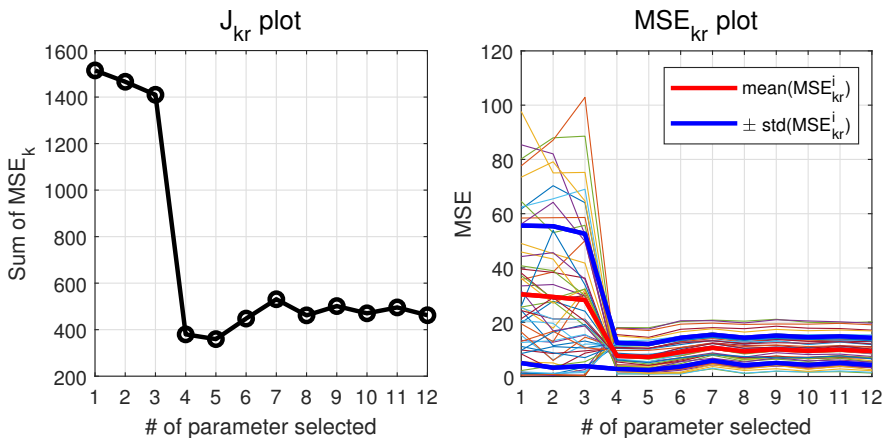
Figure 4.5: Rank estimation results: (a) PR result with θ_o^o, (b) PR result with θ_o^*

Comparing the two heatmap graphs in Figure 4.5, the rank estimation with θ_o^* shows better accuracy than that with θ_o^o. The ranked list calculated with θ_o^* is used in the following PSS step.

4.4.3 MSE-based PSS with gLAMLE subroutine

The hyper-parameters, i.e., the order of basis functions and number of knots for the B-spline curves, should be determined before applying the proposed PSS method. Several combinations of the order of the B-spline basis and number of knots for each state variable are tested by fitting the curves to the data using de Boor's algorithm as depicted in Figure 4.1a. For this numerical example, the order of the basis function and the number of knots are selected for the state variables $[X, S, V, T]$ as $Order = [4, 4, 3, 3]$ and $Knots = [8, 8, 7, 7]$. The learning rate for the gLAMLE algorithm, α is set to $\alpha = 0.15$. Given the hyper-parameters and ranked list of the model parameters, the optimal size of the model parameter subset is determined by the MSE-based PSS method, while simultaneously estimating the selected model parameters.

First, a sequence of parameter subsets, each with k_r elements, is determined based on the ranked list, for example, $\Theta_1 = \{10\} \rightarrow \Theta_2 = \{10, 6\} \rightarrow \Theta_3 = \{10, 6, 11\} \rightarrow \cdots \rightarrow \Theta_{12} = \{10, 6, 11, \ldots, 3, 4\}$. With these subsets, $gLAMLE$ algorithm is applied repetitively to estimate $\hat{\Theta}_{k_r}$ for $k_r = 1, 2, \ldots, 12$. Using $\hat{\Theta}_{k_r}$, the sum of scaled mean-squared-error (J_{k_r}) and the MSE-indices (r_{C,k_r}, r_{CKub,k_r}, and, r_{CC,k_r}) are calculated to perform the MSE-based PSS. The calculated MSEs and the indices are plotted in Figures 4.6-4.7.



(a) (b)

Figure 4.6: MSE plots for the data set used in the modeling step (a) J_{k_r} plot, (b) MSE_{k_r} plot

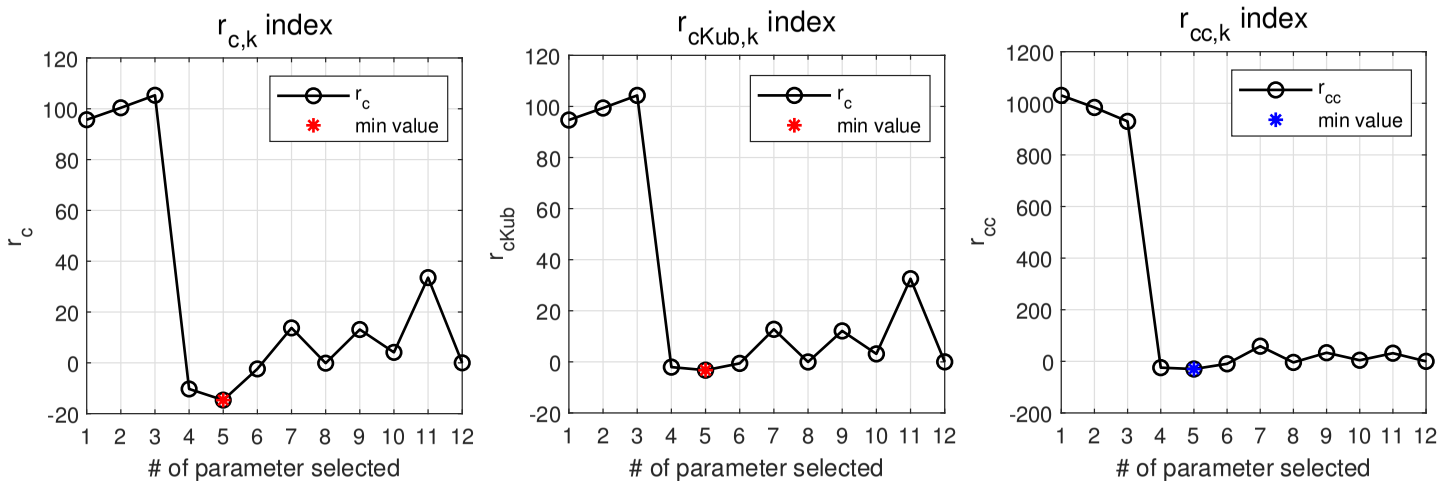
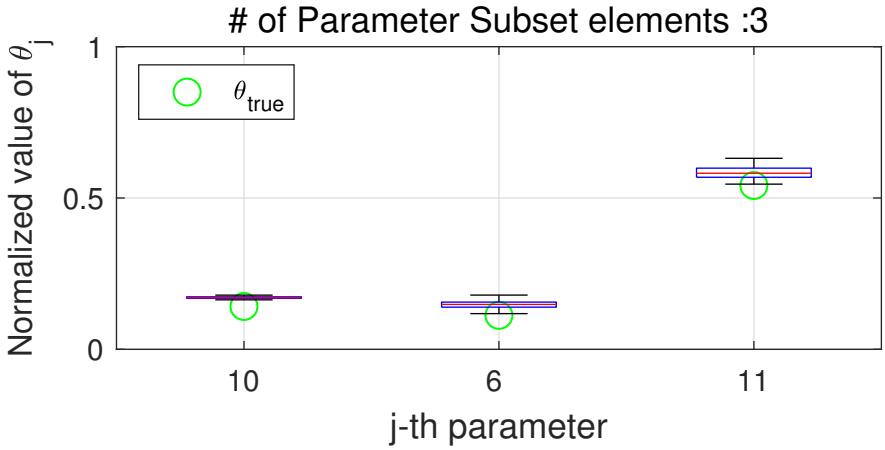


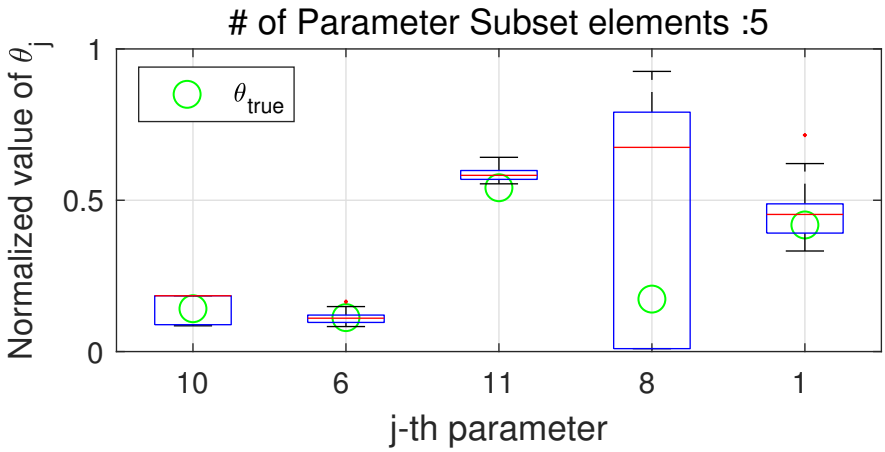
Figure 4.7: Calculated indices for MSE-based parameter subset selection. (a)r_{C,k_r} (b)r_{CKub,k_r} ,(c)r_{CC,k_r}

Based on the calculated index, r_{CC,k_r}, the optimal size of the subset under the given datasets is determined as $k_r^* = 5$, as shown in Figure 4.7c. Subsequently, the PE result, solved for $\Theta_5 = \{10, 6, 11, 8, 2\}$, becomes the optimal value of the selected model parameters (Θ_5^*), while keeping the other parameters at their initial estimated values (θ_o^*). The best estimates for Q and Σ are also obtained while selecting and estimating the best parameter subset. However, the result of PSS, including the ranked list of parameters, k_r^*, and estimated $\Theta_{k_r^*}$, Q^*, can be different when the proposed algorithm is performed with other datasets because the estimability analysis using the sensitivity matrix depends heavily on the model structure and datasets used for modeling.

When estimating the model parameters of deterministic models, the MSE tends to decrease as the number of parameters to be estimated increases. This is owing to the fact that more manipulated variables can be adjusted to explain the given data. In this numerical example, however, the value of the MSE increases when more than five parameters are estimated as shown in Figure 4.6a. The increasing MSE can be explained by the results of the PE. Figures 4.8a-4.9b show the estimation results of $\hat{\Theta}_{k_r}$ for $k_r = 3, 5, 7, 12$ with boxplots.

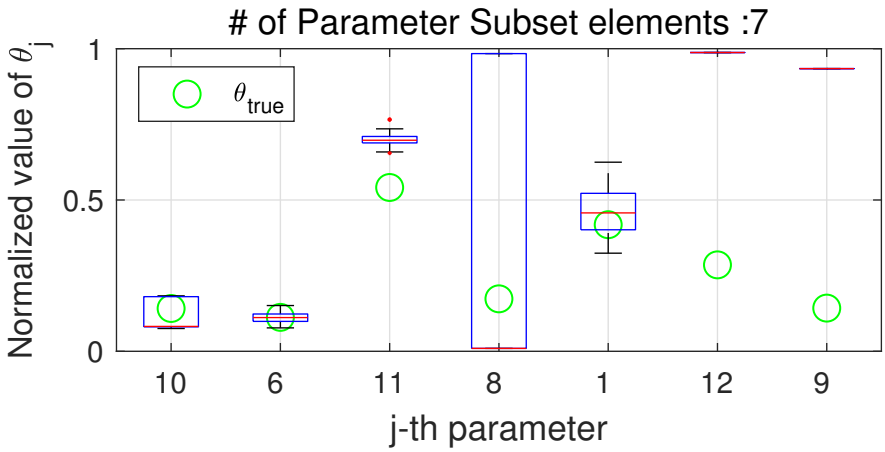


(a)

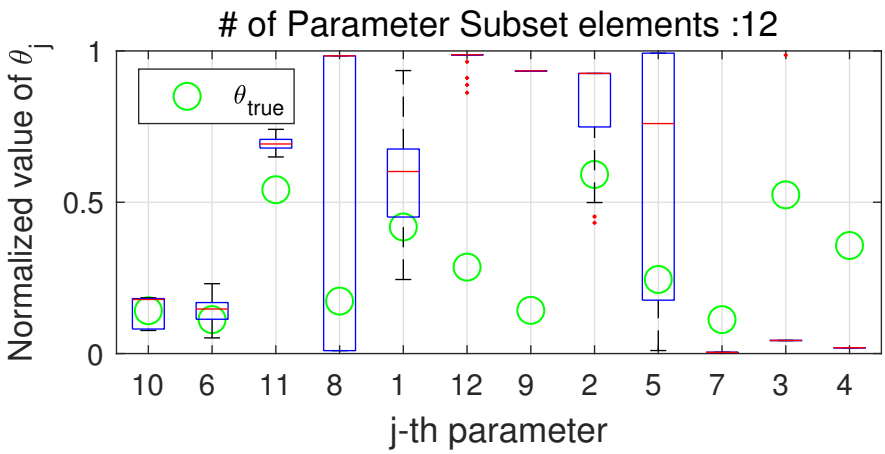


(b)

Figure 4.8: Boxplot of the estimated parameters in $\hat{\Theta_{k_r}}$ (a) $k_r = 3$, (b) $k_r = 5$



(a)



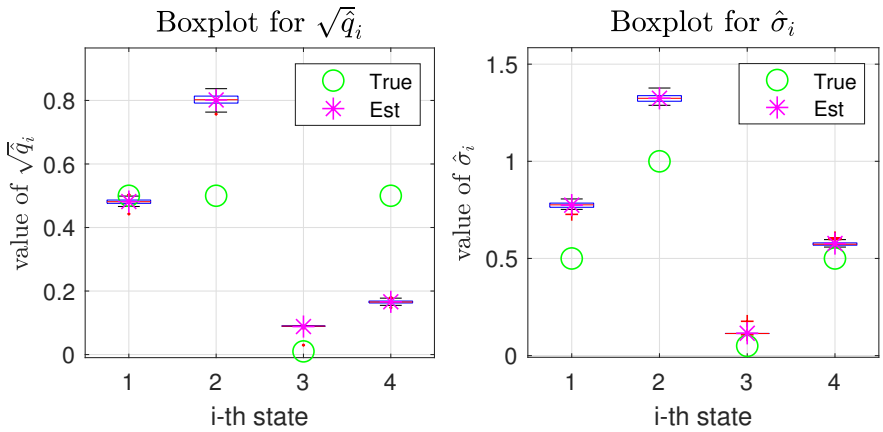
(b)

Figure 4.9: Boxplot of the estimated parameters in $\hat{\Theta_{k_r}}$ (a) $k_r = 7$, (b) $k_r = 12$

When the number of selected parameters is five or less, as shown in Figures 4.8a and 4.8b, the model parameters in Θ_{k_r} are estimated accurately except for θ_8, the fourth parameter in the ranked list. Moreover, θ_10 lose its accuracy even though it is estimated accurately in $\hat{\Theta}_3$, as shown in Figures 4.8b-4.9b. This inaccuracy occurs owing to the correlation between θ_8 and θ_{10}. In Eq. (4.37), θ_8 and θ_{10} are multiplied together as $\theta_8 \times \exp(-\theta_{10})$. Therefore, they become indistinguishable when the PE algorithm tries to estimate these two parameters simultaneously.

When estimating more than five parameters, it is impossible to estimate the 6th to 12th parameters, $\theta^6(12), \ldots, \theta^{12}(4)$, as shown in Figures4.9a and 4.9b. Moreover, when comparing the estimation results of the third and fifth parameters in Figure 4.8a to 4.9b, we find that the third and fifth parameters are accurately estimated when k_r is less than five, whereas they lose their accuracy when k_r is over six. This is due to the correlations between the parameters. The parameter estimates for the lower-ranked parameters are inaccurate as they have low sensitivity on the objective function and the initial guesses for these parameters are inaccurate, as shown in Figure 4.4. The poor estimates of the lower-ranked parameters affect the accuracy of higher-ranked parameter estimates through the correlation, resulting in biases in the parameter estimates. For instance, θ_{11} and θ_{12} are correlated through Eq. (4.33) in this numerical example. As shown in Figure 4.9a, the estimate of θ_{11} is biased when $k_r = 7$ owing to the inaccuracy in the estimate of θ_{12}, comparing with the θ_{11} estimate of $k_r = 5$ in Figure 4.8b. The inaccurate estimates affect the estimates of Q and Σ, and as a result, the overall prediction accuracy is deteriorated. Consequently, the value of the MSE increases when es-

timating more parameters than the optimal number, $k_r^* = 5$, as shown in Figure 4.6a. In other words, estimating all the parameters worsens the estimation results, and proves that the PSS procedure is necessary and possible for the ill-conditioned PE problem of the SDE models. Figure 4.10 shows the estimated state disturbance intensity ($\hat{Q}$) and measurement error ($\hat{\Sigma}$).



(a) (b)

Figure 4.10: Boxplots for estimated standard deviations of (a) constant state disturbances $\hat{Q}$ ($\sqrt{\hat{q}_i}$), (b) measurement errors $\hat{\Sigma}$ ($\hat{\sigma}_i$)

As shown in Figure 4.10, most of the results estimate the true values correctly or slightly overestimate them except for $\sqrt{\hat{q}_2}$ and $\sqrt{\hat{q}_4}$. As the accuracy of the Q and Σ estimation depends on how the B-spline curve is designed, it can be argued that these inaccurate estimates are caused by the over-fitted B-spline curves for the second and fourth state variables (S and T).

Using $\hat{\Theta}_{k_r}$, $\hat{Q}$, and $\hat{\Sigma}$, simulations are performed with additional 20 datasets to validate the PSS results. Figures 4.11-4.14 show one sample of the simulation results for $k_r = 3, 5, 7, 12$.

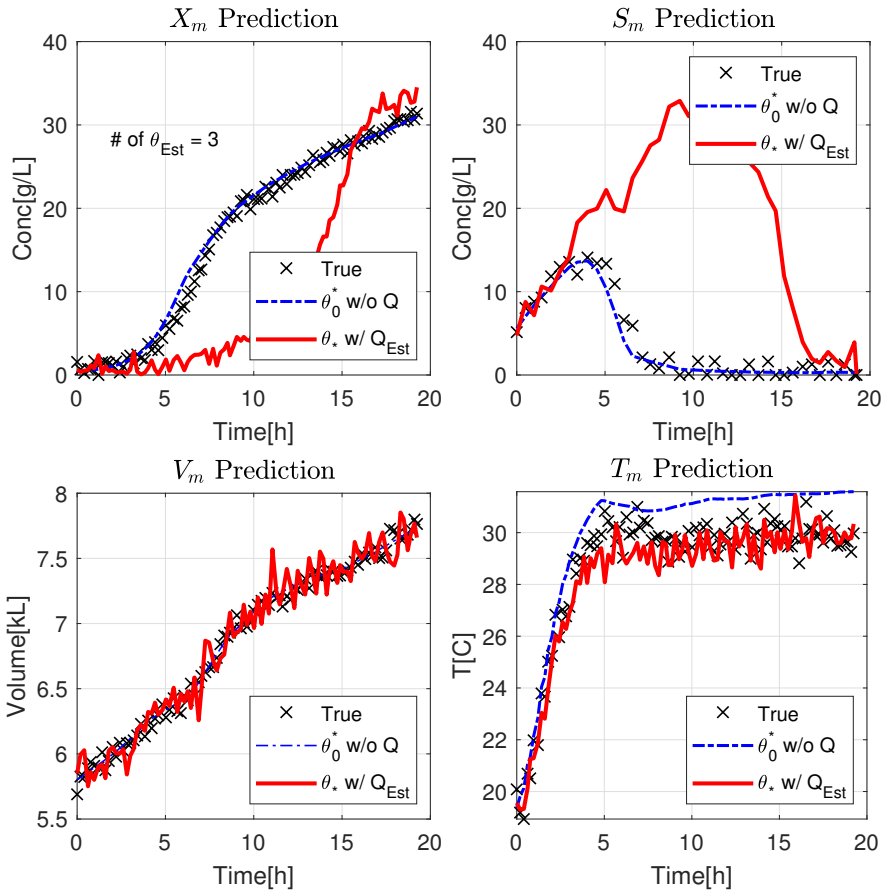


Figure 4.11: Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 3$, $\hat{\Theta}_3 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}\}$

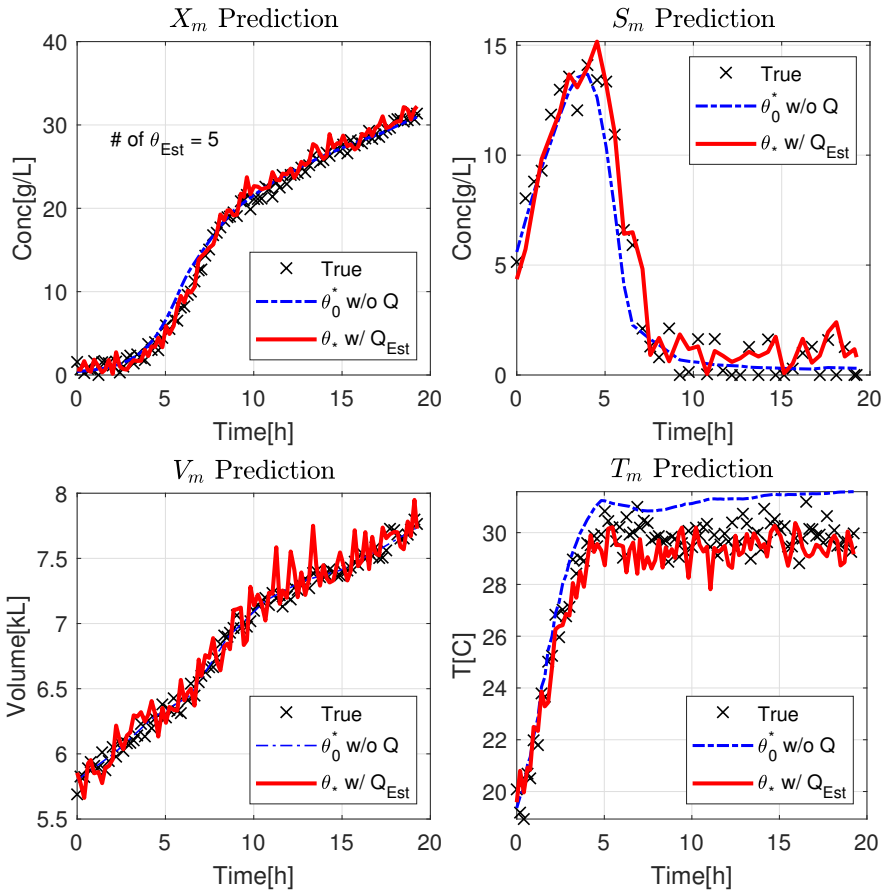


Figure 4.12: Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 5$, $\hat{\Theta}_5 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}, \hat{\theta}^{8}, \hat{\theta}^{1}\}$

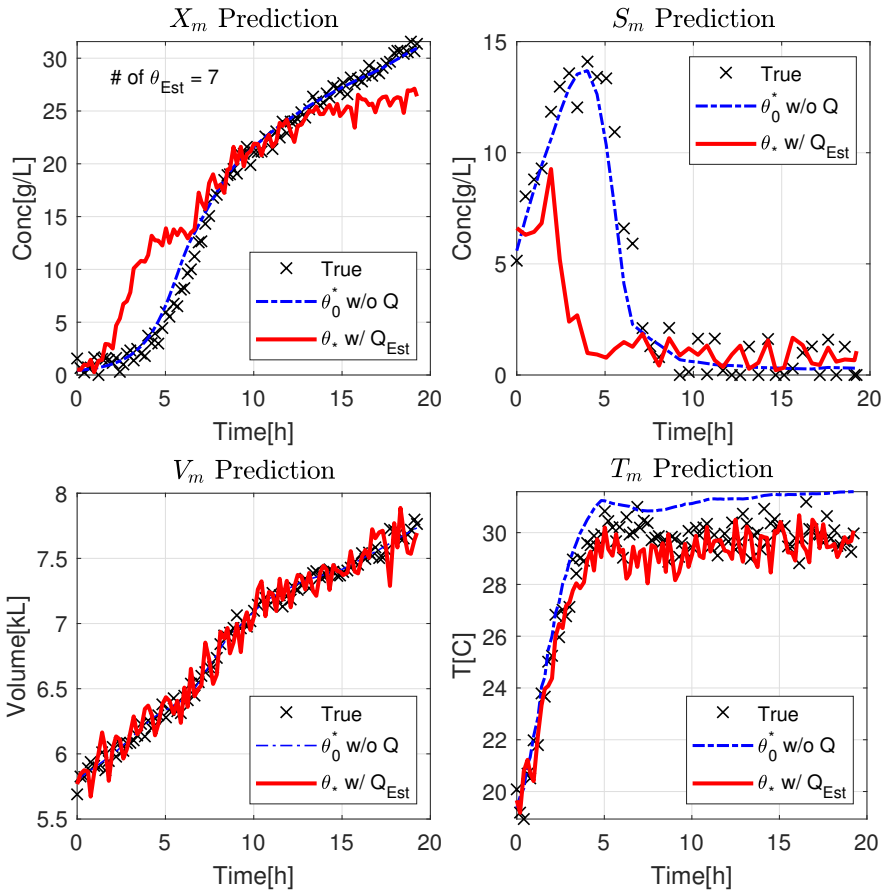


Figure 4.13: Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 7$, $\hat{\Theta}_7 = \{\hat{\theta}^{10}, \hat{\theta}^{6}, \hat{\theta}^{11}, \hat{\theta}^{8}, \hat{\theta}^{1}, \hat{\theta}^{12}, \hat{\theta}^{9}\}$

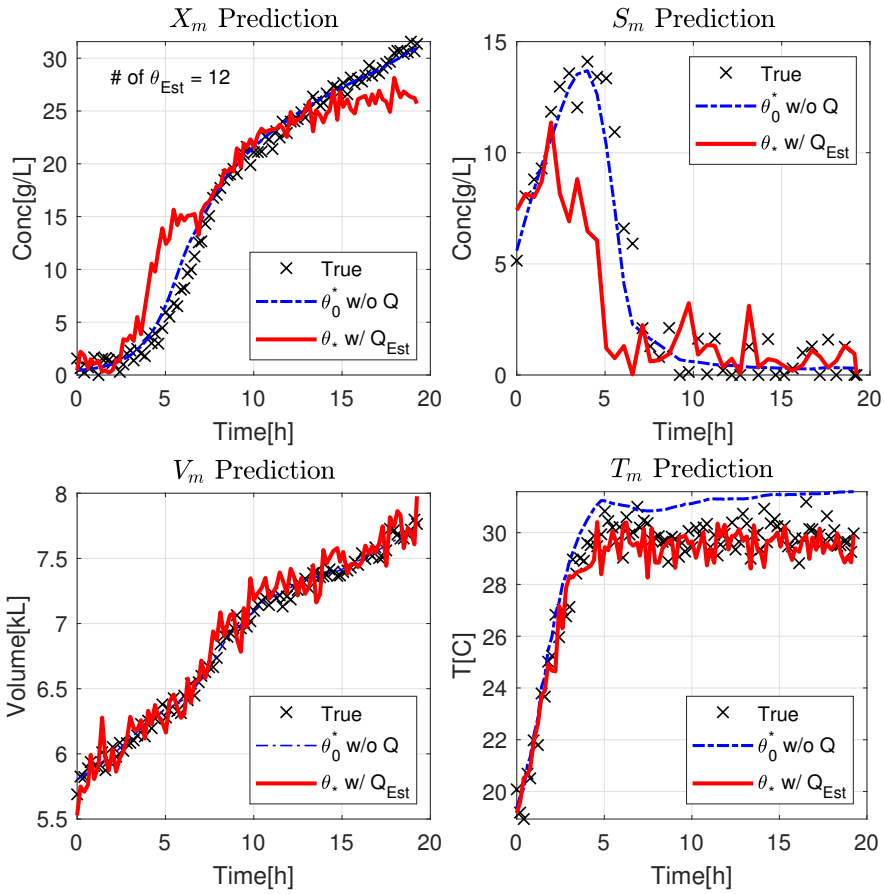
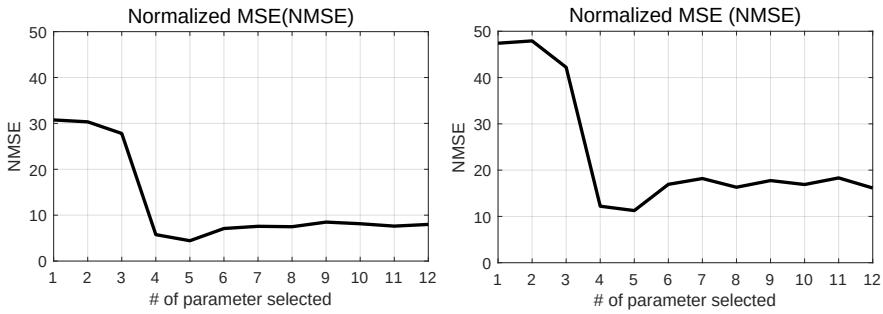


Figure 4.14: Simulation results using one sample out of the validation data sets, $\hat{\Theta}_{k_r}$ for $k_r = 12$, $\hat{\Theta}_{12}$ = all model parameters

When $k_r = 3$, $\hat{\Theta}_3$ can accurately estimate the true parameter values, as in shown Figure 4.8a. However, considering the low prediction accuracy shown in Figure 4.11, it is not sufficient to simulate the measurement data with only three updated parameters. The simulation results with $\hat{\Theta}_5$-$\hat{\Theta}_{12}$ show a more accurate prediction for the measurement data, as shown in Figures 4.12-4.14. The normalized mean squared errors (NMSE) for the testing simulations of $\hat{\Theta}_3$, $\hat{\Theta}_5$, $\hat{\Theta}_7$, and $\hat{\Theta}_{12}$ are 42.21, 11.26, 18.19, and 16.14, respectively.

The NMSE increases when k_r is 6 or more as the selected parameter subsets overfit the training datasets. The overfitting occurs as more parameters were involved in the parameter estimation procedure that can be estimated. The correlation between the model parameters also affects the NMSE of both training and testing simulations, making the values of NMSE increase when k_r is more than 5 in both results as illustrated in Figure 4.15.



(a) (b)

Figure 4.15: Normalized mean squared errors for $k_r = 1, 2, \ldots, 12$ (a) training datasets (b) testing datasets

Consequently, the simulation performed with $\hat{\Theta}_7$ and $\hat{\Theta}_{12}$ result in inaccurate predictions for the test data, as shown in Figures 4.13 and 4.14. The validation results in Figures 4.11-4.14 show that the proposed method successfully selects the optimal parameter subset while simultaneously estimating the values of the model parameters in the selected parameter subset, Q, and Σ.

Chapter 5

Multirate Moving Horizon Estimation Combined with Parameter Subset Selection [3]

5.1 Introduction

Even when the model parameters have been successfully estimated, it is hard to guarantee the accuracy of model-based online state estimation and prediction due to uncertainties in the model parameters. The uncertainties in the model parameters arise from various causes. In some cases, the parametric uncertainties occur due to the lack of information provided by the available measured data. When the number of unknown parameters is greater than the number of estimable parameters, the parameter estimation becomes ill-conditioned [34, 117]. The model parameters estimated by the ill-conditioned parameter estimation problem often have large variances, resulting in an over-fitted model of poor estimation and prediction accuracy.

The drifting nature of the system dynamics also causes uncertainties in the model parameters. As the target system operates, the

[3] This chapter is an adapted version of J. Bae, Y. Kim, and J. M. Lee, “Multirate Moving Horizon Estimation Combined with Parameter Subset Selection," *Computer & Chemical Engineering*. Accepted

system dynamics often change. Such alterations may occur within a single batch or in batch-to-batch operations [40]. Since the system dynamics are represented with the model parameters based on the predetermined model structure, the changes in the dynamics affect the model parameters. Due to these uncertainties, one needs to implement an estimation technique that can deal with the parametric uncertainties to achieve more accurate state estimates and model predictions.

A moving horizon estimation (MHE) is an optimization-based method that estimates the states and parameters of the system using a limited number of past measurements [46]. The MHE can estimate the model parameters together with the state variables by adding the model parameters to the decision variables of the optimization problem in each horizon [44, 118]. However, the MHE that estimates both states and parameters may be vulnerable to systems where the parameter uncertainties are caused by both the ill-conditioned parameter estimation and the parameter drifts. Since it determines all the states and parameters only with the measured data in the current horizon, it also becomes ill-conditioned to estimate all the model parameters online. Then, the accuracy of the online model predictions can be unsatisfactory even though it estimates the states of the system accurately.

This chapter aims to propose a MHE formulation that can improve model prediction accuracy while maintaining state estimation performance. To deal with the ill-conditioning problem, the PSS method is introduced to the online calculation of MHE. Through the PSS procedure, estimable parameters are determined based on the measured data in each horizon, and the optimizer of each horizon only

updates these parameters. To apply the PSS procedure to the online calculation of the MHE, it has to be solved quickly on each horizon. Among various methods for the PSS, a heuristic method based on the parametric sensitivity matrix is used in this study. The heuristic PSS methods are more preferable for online calculation compared to optimization-based PSS methods since they require fewer computations. The proposed MHE is developed based on a multi-rate MHE formulation to make the best use of the measured data. The efficacy of the proposed method is demonstrated using simulation studies on a virtual plant of a fed-batch bioreactor system.

5.2 Multi-rate moving horizon estimation

In practice, many processes have measurements sampled with multiple sampling rates. When applying MHE to these processes, calculating the stage cost can be confusing since measurements of some sampling time in the current horizon are missing. It is one solution to use only the data of the time at which all the outputs are measured, but it is not desirable since some of the measurements may be omitted while calculating the stage cost.

To overcome this problem, a concept of multi-rate MHE (MMHE) is suggested [119, 49, 120]. The key idea of the MMHE formulations suggested is classifying the time indices into two categories: only the fast measurement is available ($k \in F$) and both measurements are available ($k \in SF$) as in figure 5.1, under the assumption that the fast measurement rate is an integer multiple of the slow measurement rate.

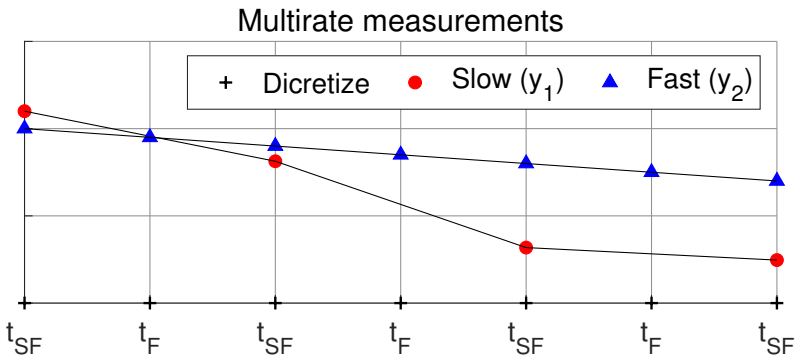


Figure 5.1: Multi-rate measurements indexed based on their sampling rates

Based on the classified indices, they developed NLP including two output models, one for $y_{k \in F}$ and the other for $y_{k \in SF}$. However, it becomes puzzling to classify all the measurements when the system involves various measurements with different sampling rates or non-periodic measurements.

5.3 Parameter ranking-based moving horizon estimation

In the MHE formulation in Eqs. (2.10), the model parameters are updated together with the state variables to improve the state estimation accuracy. However, it can be impossible to estimate all the model parameters accurately since a limited number of measurements in a fixed window are used to solve NLP. It results in an ill-conditioned parameter estimation problem. It becomes more problematic when only a part of the measurable data is available, as in the system with multi-rate measurements since the same length of the MHE horizon contains fewer number of data. Moreover, it affects the accuracy of the model prediction. Since the model parameters are easily overfitted under this condition, the model with the updated parameters may produce worse predictions than the initial model.

In this section, I propose a modified MHE formulation, introducing the parameter subset selection method to the multi-rate MHE. In each horizon, the scaled parametric sensitivity matrix is calculated with the last estimate of the model parameters, then a subset of the model parameters is selected via the orthogonalization-based PSS method. The parameter subset contains the model parameters of high sensitivity. Then, a multi-rate MHE is formulated, which esti-

mates the state variables and the selected model parameters simultaneously. The proposed MHE formulation involves a reduced number of decision variables, updating the most sensitive parameters in each horizon. Since only the estimable parameters are updated, the proposed method is less vulnerable to the over-fitting problem, resulting in better model prediction while maintaining the state estimation accuracy.

5.3.1 Online parameter subset selection in k^{th} horizon

While estimating all the decision variables in the kth horizon, the parametric sensitivity matrix of the state variables at each sampling time can be calculated with the estimated states ($\hat{X}_{L_k:k}$), model parameters ($\hat{\theta}_{L_k:k}$), the known input sequence ($U_{L_k:k}$), and analytic expressions of the parametric sensitivity, where $L_k = k - N_{mhe}$. Since the purpose of the online parameter subset selection is to select a set of model parameters that have significant effects on the state variables, the parametric sensitivity matrices are calculated for the state variables instead of the output variables.

The analytic expression for the parametric sensitivity of the state variables is derived by differentiating Eqs. (2.9a) with each model parameters. As the discretized model describes, the $i+1^{th}$ states are affected by the model parameters of the ith sampling time.

$$S_{i|k} = \frac{\partial x_{i+1}}{\partial \theta_i} = \left.\frac{\partial f(x_i, u_i, p, \theta_i)}{\theta_i}\right|_{\substack{x_i=\hat{x}_{i|k} \\ \theta_i=\hat{\theta}_{i|k}}}, \tag{5.1}$$

$$Z_{i|k} = \sigma_{i|k} \circ S_{i|k}, \quad \sigma_{i|k,lm} = s^m_{\theta,i|k}/s^l_{y,i|k} \tag{5.2}$$

$$i = L_k, ..., k-1, k.$$

where $S_{i|k}, \sigma_{i|k} \in \mathbb{R}^{n_x \times n_p}$. $\sigma_{i|k}$ is a scaling matrix and its $(l, m)^{th}$ component is $s^m_{\theta,i|k}/s^l_{y,i|k}$, the scaling factor of mth parameter over the scaling factor of lth state variable. $A \circ B$ implies an element-wise multiplication between A and B. In this paper, both scaling factors are set to be standard deviations of the parameters and the state variables. Both scaling factors are iteratively updated by the estimator using the covariance matrix calculated while updating the arrival cost term.

A scaled parametric sensitivity matrix of the kth horizon is then formulated by stacking the scaled sensitivity matrices of each sampling point in the horizon.

$$\boldsymbol{Z}_k = [Z^\top_{L_k|k}, Z^\top_{L_k+1|k}, ..., Z^\top_{k|k}]^\top. \tag{5.3}$$

where $\boldsymbol{Z}_k \in \mathbb{R}^{(n_x \cdot (N_{mhe}+1)) \times n_p}$. With $\boldsymbol{Z}_k$ calculated, a ranked list of the model parameters on the current horizon is calculated by using the orthogonalization algorithm [54],

$$\Theta_k = [\theta^1_k, \theta^2_k, ..., \theta^{n_p}_k] \tag{5.4}$$

where n_p is the number of the model parameters. Then, the subset of the model parameters is determined by the residual values, which are also calculated through the orthogonalization procedure. Given a threshold, only the first r_k parameters are selected with their residual values are higher than the threshold. The parameter subset can also be determined by mean-squared error (MSE)-based subset selection methods [3]. Evaluating the MSE while estimating the model parameters in the order of the ranked list, e.g., $[\theta^1] \rightarrow [\theta^1, \theta^2] \rightarrow [\theta^1, \theta^2, \theta^3]$

$\rightarrow \ldots \rightarrow \Theta$, the indices for subset selection are calculated with the resulting MSE values. The size of the parameter subset is determined with the indices and the ranked list of the model parameters. Since the series of the parameter subsets should be estimated by optimization during the subset selection procedure, the MSE-based method requires more computation than the residual-based selection, while it does not require the user-defined threshold.

As a result, the first r_k parameters (Θ_k^s) become a part of the decision variables of NLP at the kth horizon together with the state variables. As a consequence, the number of decision variables at each sampling time in the kth horizon becomes $(n_x + r_k)$. A set of non-selected parameters (Θ_k^{ns}) are set to the values that were calculated at the previous horizon. For the kth horizon, the resulting subsets of the model parameters are

$$\Theta_{i|k}^s = [\theta_{i|k}^1, ..., \theta_{i|k}^{r_k}], \tag{5.5}$$

$$\Theta_{i|k}^{ns} = [\theta_{i|k}^{r_k+1} ..., \theta_{i|k}^{n_p}] = [\hat{\theta}_{i|k-1}^{r_k+1} ..., \hat{\theta}_{i|k-1}^{n_p}] \tag{5.6}$$

,where $i = L_k, ..., k-1, k$.

5.3.2 Modification of stage cost to consider multi-rate measurements

To perform the parameter subset selection online on every horizon, the parametric sensitivity matrix should be calculated for each horizon. It requires to calculate the output variables at every sampling point in the current horizon. Therefore, a modified MMHE formulation is proposed to handle this problem.

Let us assume that the sampling interval h for the model discretization is always shorter than all the measurements, and the sampling intervals of the measurement are integer multiples of h. Then, the measured data can be aligned along with the sampling points of the model discretization, as in figure 5.2.

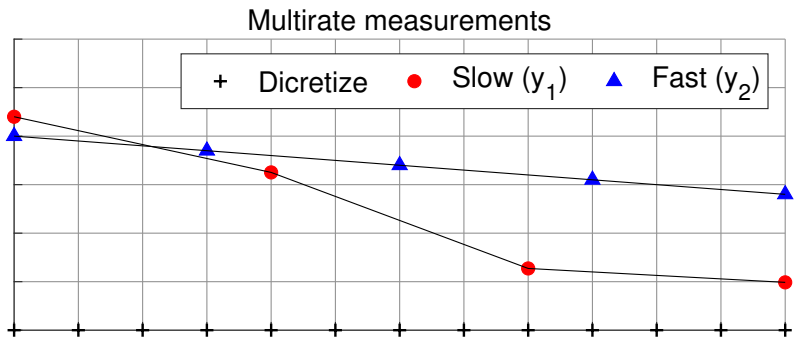


Figure 5.2: Multi-rate measurements aligned along with the sampling points of the model discretization

Let $Y_{L_k:k}$ be the aligned multi-rate measurements of the kth horizon. To solve NLP of the kth horizon, the stage cost has to be modified to consider the multi-rate measurements. One can achieve it by using the structure of $Y_{L_k:k}$ rather than classifying the measurement indices. Following the alignment in Figure 5.2, $Y_{L_k:k}$ only contains the data where the measurements exist. Let $\tilde{Y}_{L_k:k} \in \mathbb{R}^{n_y \times (N_{mhe}+1)}$ be a matrix that inherits the structure of $Y_{L_k:k}$,

$$\tilde{Y}_{L_k:k} = [\tilde{y}_{ij}] \in \mathbb{R}^{n_y \times (N_{mhe}+1)}, \tag{5.7}$$
$$s.t. \quad \tilde{y}_{ij} = \begin{cases} 1 & \text{if } y_{ij} \text{ exist} \\ 0 & \text{otherwise} \end{cases}$$

where y_{ij} is the $(i,j)^{th}$ component of $Y_{L_k:k}$ for $i = L_k, ..., k-1, k$ and $j = 1, 2, \ldots, n_y$. It implies that its element is 1 if the corresponding element of Y_i exist, or 0 otherwise. By simply multiplying $\tilde{Y}_{i|k}$ to the $v_i = y_i - h(x_i, u_i, p, \theta_i)$ component-wisely, the multi-rate measurements can be taken into the MHE formulation since the stage cost terms become 0 at which no measurements exist. Then, the cost function of the NLP for the kth horizon becomes,

$$\begin{aligned} C_{k,modi}(x_i, \theta_i, w_i, v_i) = & \left\| \begin{matrix} x_{L_k} - \bar{x}_{L_k} \\ \theta_{L_k} - \bar{\theta}_{L_k} \end{matrix} \right\|_{P_L}^2 + \sum_{i=L_k}^{k} \left\| v_i \circ \tilde{Y}_{i|k} \right\|_{Q_i}^2 \\ & + \sum_{i=L_k}^{k-1} \left(\left\| w_i \right\|_{R_x}^2 + \left\| w_{\theta,i} \right\|_{R_\theta}^2 \right) \end{aligned} \tag{5.8}$$

where $v_i = y_i - h(x_i, u_i, p, \theta_i)$ and $w_{\theta,i} = \theta_{i+1} - \theta_i$ for $i = L_k, \ldots, k$. Except for $\tilde{Y}_{i|k}$, the MHE formulation of the modified MMHE is equal to that of Eqs. (2.10).

In MHE-aUKF framework, the covariance matrix of the arrival cost (P_L) is updated by augmented unscented Kalman filter (aUKF), requiring all the output values and the estimates of the model parameters at every sampling point. Since the model is available, it is possible to estimate the value of the measurements at which the outputs are not measured. Let us define the modified measurements matrix $\hat{Y}_{L_k:k}$ as the matrix $Y_{L_k:k}$ with the values of the unmeasured output estimated with the model. With $\hat{Y}_{L_k:k}$, the covariance matrix of the arrival cost can be updated by aUKF while using the modified MMHE formulation. $\hat{Y}_{L_k:k}$ can also be used for the state and parameter estimation via aUKF at $k = 0, 1, \ldots, N_{mhe} - 1$ until the MHE starts at $k = N_{mhe}$. The part for the model parameters in the updated covariance matrix is used to calculate the scaling factor for the model parameters for the next horizon.

5.3.3 Parameters ranking-based moving horizon state and parameter estimation

Updating the selected model parameters, it is possible to overcome the over-fitting problem while maintaining state estimation accuracy. Based on the modified MMHE formulation, a parameter ranking-based MHE (prMHE) is formulated by introducing the online parameter subset selection procedure into each horizon of the MHE. For $k = 0, 1, \ldots, N_{mhe} - 1$, the kth MHE horizon expands one by one until $k = N_{mhe}$. prMHE of full horizon of $N_{mhe} + 1$ sampling points starts from $k = N_{mhe}$, containing the sampling points of $k = 0, 1, \ldots, N_{mhe}$ in the horizon.

In the kth horizon of prMHE, the parametric sensitivity matrix

$\boldsymbol{Z}_k$ of Eq. (5.3) is calculated with the kth initial trajectories of the model parameters $\Theta^o_{L_k:k|k}$, the state variables $X^o_{L_k:k|k}$, and the known input trajectories $U_{L_k:k}$. $\theta^o_{L_k:k|k}$ and $X^o_{L_k:k|k}$ are updated from the estimated trajectories of the $(k-1)^{th}$ horizon. Then, the subset of the model parameters is selected by performing the parameter subset selection. Let us assume the ranked list of the parameters is derived as in Eq. (5.4) and the first r_k parameters are selected as in Eq. (5.5). Then, the NLP formulation of prMHE at the kth horizon is written as,

$$\min_{x_i, w_i, v_i, \boldsymbol{\Theta}^s_k} \quad C_{k,modi}(x_i, \theta_i, w_i, v_i) \tag{5.9a}$$

$$s.t. \quad x_{i+1} = f(x_i, u_i, p, \theta_i) + w_i, \tag{5.9b}$$

$$w_{\theta,i} = \theta_{i+1} - \theta_i, \quad i = L_k, \ldots, k-1, \tag{5.9c}$$

$$y_i = h(x_i, u_i, p, \theta_i) + v_i, \quad i = L_k, \ldots, k \tag{5.9d}$$

$$g(x_i, u_i, p, \theta_i) = 0, \quad i = L_k, \ldots, k \tag{5.9e}$$

$$x_i^{min} \leq x_i \leq x_i^{max}, \quad v_i^{min} \leq v_i \leq v_i^{max}, \tag{5.9f}$$

$$w_i^{min} \leq w_i \leq w_i^{max}, \quad w_{\theta,i}^{min} \leq w_{\theta,i} \leq w_{\theta,i}^{max}, \tag{5.9g}$$

$$\theta_i^{j,min} \leq \theta_i^j \leq \theta_i^{j,max}, \quad \theta_i^j \in \Theta_k^s \tag{5.9h}$$

$$\theta_i^j = \hat{\theta}^j_{i|k-1}, \quad \theta_i^j \in \Theta_k^{ns} \tag{5.9i}$$

where $\boldsymbol{\Theta}^s_k = [\Theta^s_{L_k|k}, \Theta^s_{L_k+1|k}, \ldots, \Theta^s_{k|k}] \in \mathbb{R}^{r_k \times (N_{mhe}+1)}$ and θ_i^j is the jth model parameter of the ith sampling point in the kth horizon. The non-selected parameters are set to the values calculated at the previous horizon. While the NLP of MHE in Eq.(2.10) needs to estimate $(n_p \times (N_{mhe}+1))$ model parameters for each horizon, the NLP of the proposed method in Eq. (5.9) estimates only $(r_k \times (N_{mhe}+1))$ model

parameters. If r_k is small enough compared to n_p, which means that less amount of data is required, it is possible to adjust the size of window for each horizon.

Algorithm .4: prMHE algorithm

Result:

initialize: $X_0^o = X_o^o$, $\theta_0^o = \theta_o$, $P_{L,0|0}^o = P_L^o$

for $k = 0, 1, \ldots, N$ **do**

(1) Get measured data Y_k from plant

if $k < N_{mhe}$ **then**

$L_k = 0$, (Expanding horizon)

else

$L_k = k - N_{mhe}$, (Moving horizon)

end

(2) Set measurement matrix $Y_{L_k:k|k}$ and index matrix $\tilde{Y}_{L_k:k|k}$.

(3) Calculate Z_k matrix Z_k = $[Z_{L_k|k}^\top, Z_{L_k-1|k}^\top, \ldots, Z_{k|k}^\top]$, and determine Θ_k^s and Θ_k^{ns}.

(4) Formulate NLP of Eq. (5.9) with $i = 0, 1, \ldots, k$ and solve NLP with the initial variables: $X_{L_k:k|k}^o$, $\theta_{L_k:k|k}^o$, $P_{L,L_k|k}^o$.

if $k < N_{mhe}$ **then**

(Expanding horizon)

(5)' Update P_L matrix using aUKF $[\hat{X}_{0|k}, \hat{\theta}_{0|k}, P_{L,0|k}^o] \rightarrow P_{L,0|k}$.

(6)' Update the matrices of initial variables $X_{0:k+1}^o$=$[\hat{X}_{0:k|k}$,$\hat{X}_{k|k}]$, $\theta_{0:k+1}^o$=$[\hat{\theta}_{0:k|k}$,$\hat{\theta}_{k|k}]$, $P_{L,0|k+1}^o$ = $P_{L,0|k}$

else

(Moving horizon)

(5) Update P_L matrix using aUKF, with $[\hat{X}_{L_k|k}, \hat{\theta}_{L_k|k}, P_{L,L_k|k}^o] \rightarrow P_{L,L_k+1|k}$

(6) Update the matrices of initial variables $X_{L_k+1:k+1}^o$=$[\hat{X}_{L_k+1:k|k}$,$\hat{X}_{k|k}]$, $\theta_{L_k+1:k+1}^o$=$[\hat{\theta}_{L_k+1:k|k}$,$\hat{\theta}_{k|k}]$, $P_{L,L_k+1|k+1}^o$ = $P_{L,L_k+1|k}$

end

(7) Return the current estimates for model prediction: $\hat{X}_{k|k}$, $\hat{\theta}_{k|k}$

end

It is also possible to fix the model parameters over the kth horizon, as $\Theta_k^s = \Theta_{L_k|k}^s = \Theta_{L_k+1|k}^s = \ldots = \Theta_{k|k}^s$. It reduces the online computational load since it reduces the number of the decision variables of the NLP by $r_k \times N_{mhe}$. The prMHE problem with the fixed model parameters is formulated by the same NLP of Eqs. (5.9) with the vector, Θ_k^s, instead of the matrix, $\mathbf{\Theta}_k^s$.

If the model parameters fluctuate or drift fast in the horizon, prMHE of Eq.(5.9) with the free parameters is more suitable than prMHE with the fixed model parameters since it reflects the behaviors of the model parameters at every sampling time in each horizon. In the opposite case, prMHE with the fixed parameters is preferable since it involves fewer decision variables, being less vulnerable to the over-fitting issues, and reducing the computational burden.

5.4 Numerical Example: A fed-batch bioreactor

In this section, the proposed MHE algorithm (prMHE) is applied to a virtual plant of a fed-batch bioreactor to show its efficacy. A state estimation and model prediction is performed and the result is compared with the results of two benchmark MHEs, including MHE estimating only the states (MHE(x)) and MHE estimating the states and all the model parameters (MHE(x, θ)), respectively. The overall simulations are carried out on Matlab 2019a environment. Dynamic optimizations in each horizon of the MHE problems are transformed into the NLP by a collocation method [47] with the Legendre collocation points, and they are solved with Matlab-CasADi environment on Matlab 19a [87].

5.4.1 Problem statement

A virtual plant of a fed-batch bioreactor is selected as the target nonlinear system in this example. The plant model of 4 state variables (X, S, V, T), 4 input variables (F_{in}, F_{out}, F_c, F_h), and 12 model parameters is assumed to operate for $\tau_{batch} = 20[hr]$. The plant model equations are

$$\frac{dX}{dt} = \mu(X,S,T)X - DX + \eta_X \tag{5.10a}$$

$$\frac{dS}{dt} = -\sigma(X,S,T)X + D(S_{in} - S) + \eta_S \tag{5.10b}$$

$$\frac{dV}{dt} = (Fin - Fout - Fevap)/1000 + \eta_V \tag{5.10c}$$

$$\frac{dT}{dt} = \frac{1}{1000V}\frac{1}{C_{\rho_b}\rho_b}\Bigg(F_{in}C_{\rho_s}\rho_f(T_s - T) \tag{5.10d}$$

$$- \Delta H_{evap}\rho_w F_{evap} + U_{jkt}A_{jkt}(T_{air} - T) \tag{5.10e}$$

$$+ F_h\alpha_h U_h(T_h - T) + F_c\alpha_c U_c(T_c - T) \tag{5.10f}$$

$$+ \mu(X,S,T)XY_{qX} \times 1000V\Bigg) + \eta_T \tag{5.10g}$$

$$\mu(X,S,T) = \frac{\mu_{max}S}{S + K_s}\exp\left(\frac{-Eg}{R(T + 273.15)}\right) \tag{5.10h}$$

$$\sigma(X,S,T) = Y_{sx}X\mu(X,S,T) + m_s \tag{5.10i}$$

$$\rho_f = (S/1000)\rho_s + (1 - S_{in}/1000)\rho_w \tag{5.10j}$$

$$\rho_b = (S/1000)\rho_s + (1 - S/1000)\rho_w \tag{5.10k}$$

$$F_{evap} = S\alpha_{evap}\left(\exp\left(\frac{2.5T}{100}\right) - 1\right) \tag{5.10l}$$

$$D = F_{in}/(1000V) \tag{5.10m}$$

where $X[g/L]$ is a cell concentration, $S[g/L]$ is a substrate concentration, $V[kL]$ is a broth volume, and $T[C]$ is a reactor temperature. η_j

is an additive state disturbance for $j = X, S, V, T$, assumed to follow $N(0, \eta_j)$. The output model for this system is

$$y_j = x_j + \epsilon_j, \quad \epsilon_j \sim N(0, \sigma_j^2) \tag{5.11}$$

where $j = X, S, V, T$ and ϵ_j is an additive measurement error, assumed to follow $N(0, \sigma_j)$. The output variables are measured with multiple measurement rates: a fast measurement $sr_1 = 0.2[hr]$ for X, V, T and a slow measurement $sr_2 = 0.5[hr]$ for S. The temperature is controlled by PI-controller with a set-point, $T_{sp} = 30[C]$. The initial state and constants are tabulated in Table 5.1. The model parameters of the plant are also tabulated in Table 5.2.

Parameter	Value[units]	Description
X_o	0.5 [g/L]	cell initial conc.
S_o	5 [g/L]	substrate initial conc.
V_o	5.8 [kL]	initial broth volume
T_o	27 [C]	initial temperature
S_{in}	200 [g/L]	Feed subs. conc.
T_h	40 [C]	Heat source temperature
T_c	20 [C]	Cooling water temperature
T_s	25 [C]	Feed temperature
T_{air}	20 [C]	Air temperature
T_{sp}	30 [C]	T Set-point
η_X	0.5 [g/L]	state disturbance (X)
η_S	0.5 [g/L]	state disturbance (S)
η_V	0.01 [kL]	state disturbance (V)
η_T	0.5 [C]	state disturbance (T)
σ_X	1 [g/L]	measurement error (X)
σ_S	1 [g/L]	measurement error (S)
σ_V	0.05 [kL]	measurement error (V)
σ_T	0.5 [C]	measurement error (T)
τ_{batch}	20 [hr]	Batch time
h	0.1 [hr]	dicretization rate
sr_1	0.2 [hr]	fast measurement rate
sr_2	0.5 [hr]	slow measurement rate
ΔH_{evap}	2430.7 [kJ/kg]	Heat of evaporation
ρ_w	1000 [kg/m3]	water density
ρ_s	1540 [kg/m3]	substrate density
R	8.314e-03 [kJ/mol K]	gas coefficient
A_{jkt}	0.2 [hr]	Jacket surface area
α_h	2451.8 [kJ/m3]	Heat transfer coeff.
α_c	2451.8 [kJ/m3]	Heat transfer coeff.

Table 5.1: Initial conditions and process constants for simulation

Param.	Value[units]	Description
C_{ρ_b}	4.2 [kJ/(kg K)]	heat cap. of broth
C_{ρ_s}	5.9 [kJ/(kg K)]	heat cap. of subs.
α_{evap}	5.24e-04 [L/hr]	evaporation constant
U_{jkt}	36 [kW m2/K]	heat transfer coeff. of jacket
U_h	12 [kW m2/K]	heat transfer coeff.
U_c	12 [kW m2/K]	heat transfer coeff.
μ_{max}	180 [1/hr]	max cell growth coeff.
K_s	1.5 [g/L]	subs. coefficient
E_g	14.88 [kJ/mol]	energy coeff. of cell growth
Y_{sx}	1.85 [g/g]	yield coeff.
m_s	0.029 [g/g hr]	maintenance (subs.)
Y_{qx}	25 [kJ/g]	heat generation coeff. of cell

Table 5.2: Model parameters for virtual plant and nonlinear model for the MHE

A nonlinear model for the MHE is assumed to have the same structure and the model parameters with the plant model and be discretized with the sampling rate of $h = 0.1[hr]$. However, parameter drifts are introduced to the plant model to test the accuracy of state estimation and model prediction under the parametric uncertainties. Among the model parameters, U_c (θ_7), μ_{max} (θ_8), and E_g (θ_{10}) are assumed to change during the batch time. U_c decreases linearly to be 9.6 (-20%), while E_g increases linearly to be 16.37 (+10%), respectively. To simulate a sudden change in the model parameter, μ_{max} is assumed to decrease to be 90 (-50%) when the operation time reaches 10 hr. Trajectories of the changing model parameters are shown in Figure 5.5, drawn with black solid lines.

5.4.2 Benchmark problems: basic moving horizon state and parameter estimations

Using the cost function of Eq. (5.8), a basic MMHE-aUKF only estimating the states of the system (MHE(x)) is formulated as

$$\min_{x_i, w_i, v_i} \left\| x_{L_k} - \bar{x}_{L_k} \right\|^2_{P_{L_k}} + \sum_{i=L_k}^{k} \left\| v_i \circ \tilde{Y}_{i|k} \right\|^2_Q + \sum_{i=L_k}^{k-1} \left\| w_i \right\|^2_R \quad (5.12a)$$

$$s.t. \quad x_{i+1} = f(x_i, u_i, p, \theta_o) + w_i, \quad (5.12b)$$

$$i = L_k, \ldots, k-1, \quad (5.12c)$$

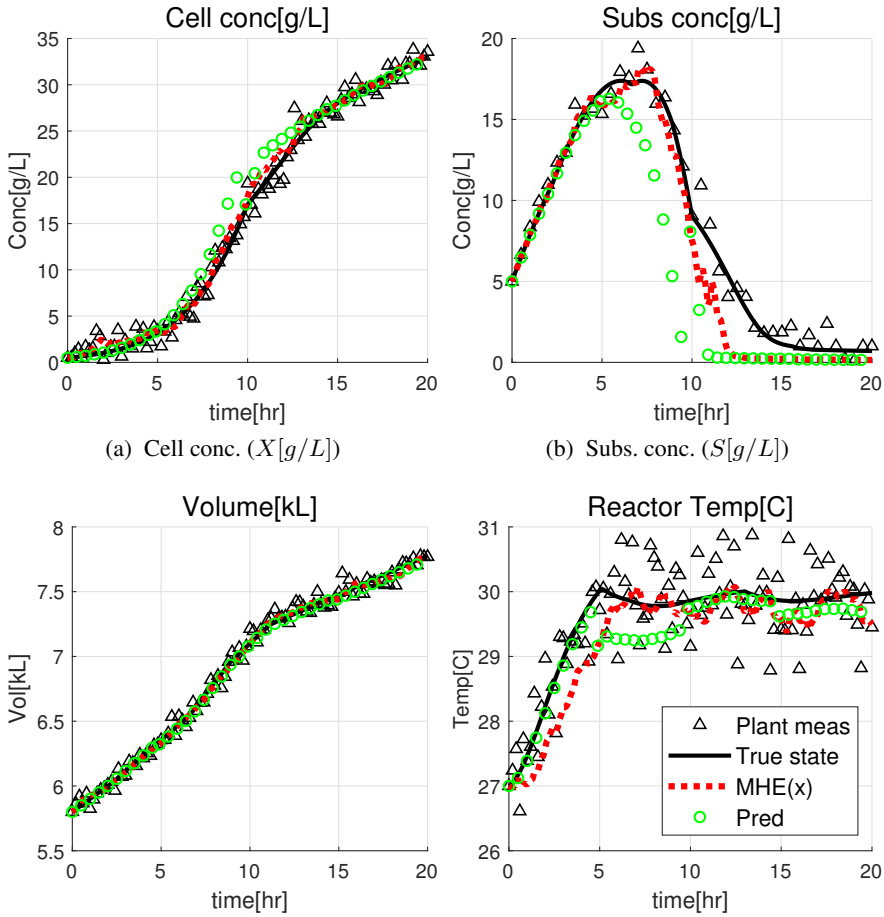
$$y_i = x_i + v_i, \quad i = L_k, \ldots, k \quad (5.12d)$$

$$g(x_i, u_i, p, \theta_o) = 0, \quad i = L_k, \ldots, k \quad (5.12e)$$

$$x_i^{min} \leq x_i \leq x_i^{max}, \quad v_i^{min} \leq v_i \leq v_i^{max}, \quad (5.12f)$$

$$w_i^{min} \leq w_i \leq w_i^{max}, \quad (5.12g)$$

where $N_{mhe} = 10$, $k = 0, 1, \ldots, 201$, $L_k = k - N_{mhe}$, $Q = diag([\eta_X^2, \eta_S^2, \eta_V^2, \eta_T^2])^{-1}$, and $R = diag([\sigma_X^2, \sigma_S^2, \sigma_V^2, \sigma_T^2])^{-1}$, respectively. At several sampling times, $t = 0, 5, 10, 15[hr]$, the future states over 50 steps, 5 hr, are predicted with the estimated states at each sampling time and the nonlinear model. Figure 5.3 shows the estimated state trajectories and the model prediction results.



(a) Cell conc. ($X[g/L]$)

(b) Subs. conc. ($S[g/L]$)

(c) Volume ($V[kL]$)

(d) Temp. ($T[C]$)

Figure 5.3: Estimated states and model prediction results of the basic MMHE-aUKF: Measured outputs (black △ marker), true state of the virtual plant (black solid line), estimated state (red dotted line), predicted state (green o marker)

As shown in Figures 5.3 (a) and (c-d), the state variables measured with the fast measurement (sr_1), X, V, and T, are estimated accurately. However, the state estimate of S shows off-sets after $t = 10hr$, when the sudden change is occurred to θ_{10}, as seen in Figure 5.3 (b). Green ring markers in Figure 5.3 indicate the model prediction results calculated at $t = 0, 5, 10, 15[hr]$. Since MHE(x) can not take the parameter drifts into account, the entire model prediction depends on the initial model. As a consequence, the model prediction becomes inaccurate over the batch time, while the model at the early stage of the operation ($t = 0$ to $t = 5$) predicts the true states accurately, as shown in Figure 5.3 (b). The model predicts V exactly, since V is less affected by the model parameters.

The accuracy of the state estimations can be improved by estimating the states and model parameters simultaneously as in Eqs. (2.10). The MMHE-aUKF estimating both the states and the model parameters (MHE(x, θ) is written as

$$\min_{x_i, w_i, v_i, \theta} \quad C(x_i, w_i, v_i, \theta) \tag{5.13a}$$

$$s.t. \quad x_{i+1} = f(x_i, u_i, p, \theta) + w_i, \tag{5.13b}$$

$$w_{\theta,i} = \theta_{i+1} - \theta_i, \quad i = L_k, \ldots, k-1, \tag{5.13c}$$

$$y_i = x_i + v_i, \quad i = L_k, \ldots, k \tag{5.13d}$$

$$g(x_i, u_i, p, \theta) = 0, \quad i = L_k, \ldots, k \tag{5.13e}$$

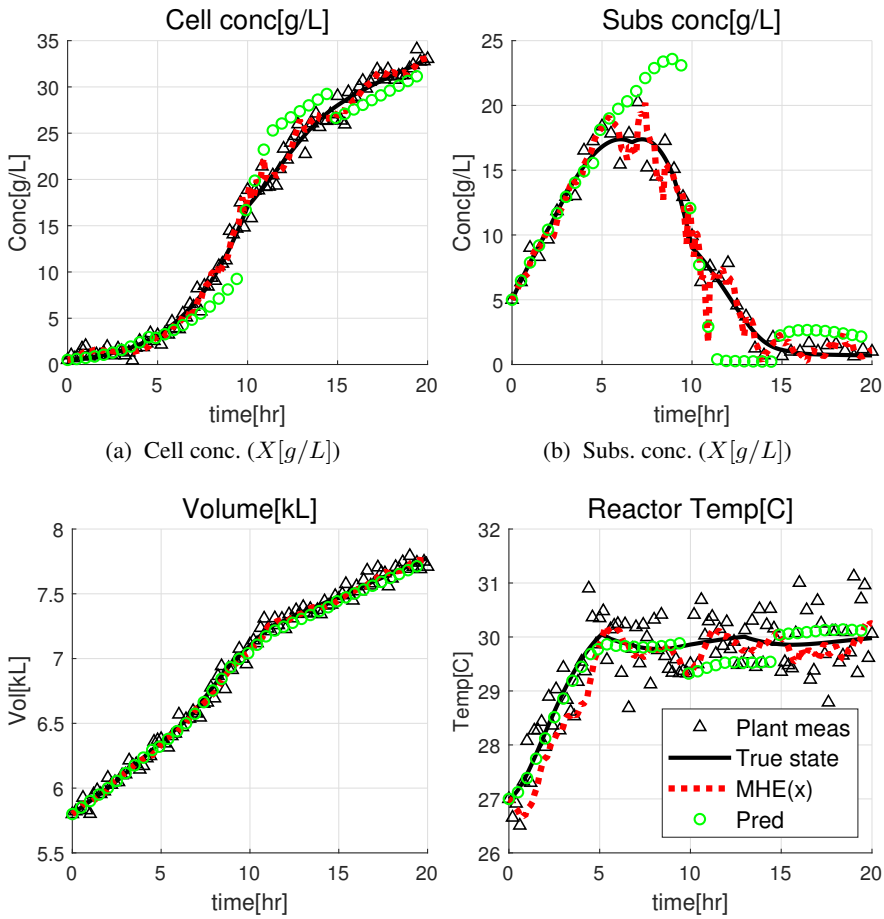
$$x_i^{min} \leq x_i \leq x_i^{max}, \quad v_i^{min} \leq v_i \leq v_i^{max}, \tag{5.13f}$$

$$w_i^{min} \leq w_i \leq w_i^{max}, \quad w_{\theta,i}^{min} \leq w_{\theta,i} \leq w_{\theta,i}^{max}, \tag{5.13g}$$

$$\theta_k^{LB} \leq \theta_{i|k} \leq \theta_k^{UB}. \tag{5.13h}$$

where Q_i in Eq. (5.8) is set to be $Q_i = Q$. In Eq.(5.13), Eq. (5.8)

is used for the cost function $C(x_i, w_i, v_i, \theta)$. The covariance matrix of the arrival cost, P_{L_k}, is updated by aUKF. The upper and lower bounds for $\theta_{i|k}$ are set to be $(1 \pm a)\ \hat{\theta}_{i|k-1}$ for $i = L_k, \ldots, k-1$ and $(1 \pm a)\ \hat{\theta}_{k-1|k-1}$ for $i = k$, where $\hat{\theta}_{i|k-1}$ is the estimated model parameters of the $(k-1)^{th}$ horizon. a is set to 0.01 for this example, meaning that 1% variations on the model parameters are allowed for each MHE horizon. Figure 5.4 shows the estimated state trajectories and the model prediction results performed at $t = 0, 5, 10, 15[hr]$.



(a) Cell conc. ($X[g/L]$)

(b) Subs. conc. ($X[g/L]$)

(c) Volume ($V[kL]$)

(d) Temp. ($T[C]$)

Figure 5.4: Estimated states and model prediction results of the MMHE-aUKF estimating the states and the model parameters altogether

Comparing Figure 5.3 and Figure 5.4, it can be found out that a simultaneous estimation of the states and the model parameters improves the accuracy of the state estimation, reflecting the drifting characteristics of the model parameters. A sum of normalized root mean-squared-errors (NRMSE) over four state variables for both results of the state estimation are 0.26 and 0.21, tabulated in Table 5.3.

Σ **NRMSE**	State Estimation	State Prediction (for 5 hr)			
		t = 0hr	t = 5hr	t = 10hr	t = 15hr
MHE(x)	0.26	0.06	0.66	0.35	0.13
MHE(x, θ)	0.21	0.06	0.50	0.49	0.23
prMHE	0.18	0.10	0.22	0.31	0.06

Table 5.3: Sum of NRMSE for state estimation and prediction results-MHE(x), MHE(x, θ), and prMHE

The state predictions are performed at $t = 0, 5, 10, 15[hr]$ and plotted in Figure 5.4 with green ring markers. The sums of NRMSE are also tabulated in Table 5.3. Unlike MHE(x), the model parameters are updated at each sampling time and the updated model is used for the state predictions in this case. Even though the model parameters have been updated throughout the estimation procedure, however, the model predictions are not improved and even become inaccurate at some time horizons. The poor model prediction is due to the ill-conditioning problem of the model parameter estimation in MHE(x, θ), resulting in the updated model to over-fit the measured outputs in each MHE horizon. The trajectories of updated model parameters are shown in Figure 5.5, together with the true parameter trajectories.

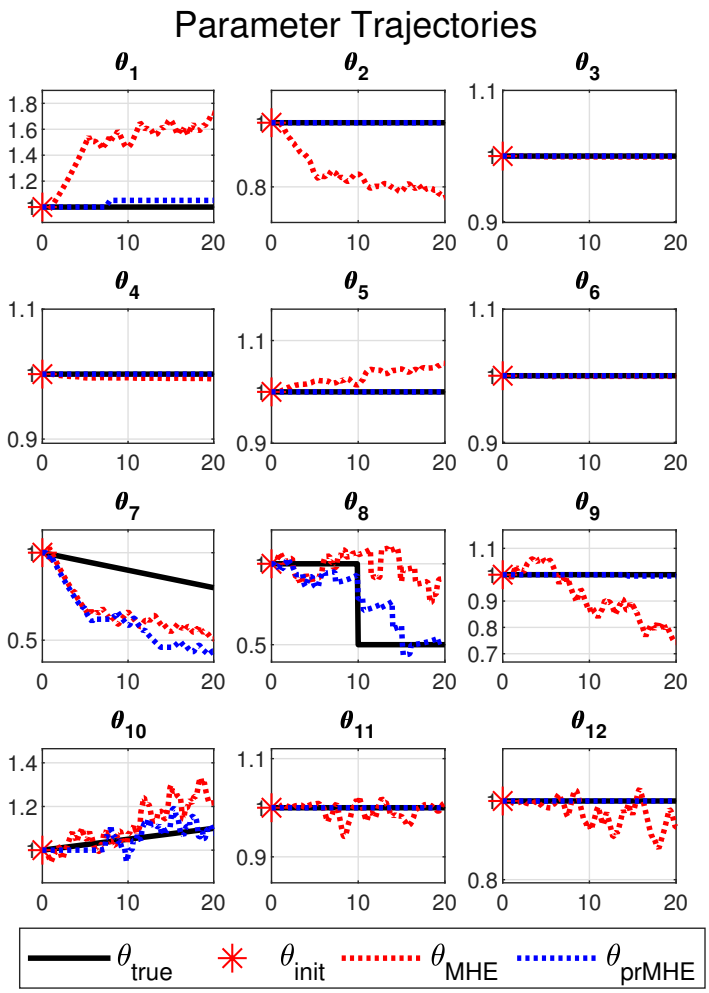


Figure 5.5: Trajectories of the model parameters updated with the MMHE-aUKF (red dotted line) and prMHE (blue dotted line) compared to the true trajectories (black solid line)

Red dotted lines in Figure 5.5 are the trajectories of the model parameters updated by MHE(x, θ). Compared with the true parameter trajectories (black solid line), it shows that MHE(x, θ) fails to follow the true parameter trajectories while updating the wrong model parameters, which are not perturbed by the parameter drifts.

5.4.3 Parameter ranking-based moving horizon state and parameter estimation

The proposed MHE algorithm, prMHE, is able to relieve the over-fitting problem while maintaining the improved state estimation accuracy by performing the parameter subset selection in each MHE horizon. It only updates the model parameters that have a large sensitivity on the state variables, reducing the number of model parameters to be estimated in each MHE horizon. Applying prMHE to the example problem, the resulting NLP for each horizon is

$$\min_{x_i, w_i, v_i, \boldsymbol{\Theta}_k^s} \quad C(x_i, w_i, v_i, \theta) \tag{5.14a}$$

$$s.t. \quad x_{i+1} = f(x_i, u_i, p, \theta) + w_i, \tag{5.14b}$$

$$w_{\theta,i} = \theta_{i+1} - \theta_i, \quad i = L_k, \ldots, k-1, \tag{5.14c}$$

$$y_i = x_i + v_i, \quad i = L_k, \ldots, k \tag{5.14d}$$

$$g(x_i, u_i, p, \theta) = 0, \quad i = L_k, \ldots, k \tag{5.14e}$$

$$x_i^{min} \leq x_i \leq x_i^{max}, \quad v_i^{min} \leq v_i \leq v_i^{max}, \tag{5.14f}$$

$$w_i^{min} \leq w_i \leq w_i^{max}, \quad w_{\theta,i}^{min} \leq w_{\theta,i} \leq w_{\theta,i}^{max}, \tag{5.14g}$$

$$\theta_k^{LB} \leq \theta_{i|k} \leq \theta_k^{UB}. \tag{5.14h}$$

where Q_i, R, R_θ, and the upper and lower bounds for the model parameters are the same as in Eqs. (5.13). As the orthogonalization method is used for selecting the parameter subset, it is essential to supply a proper threshold to the PSS algorithm because it determines the number of parameters to be updated in each time step. If the threshold is not selected properly, relatively lower ranked parameters can not be addressed adequately by the prMHE. In this example, the user-defined threshold for the orthogonalization-based subset selection is set to 0.01. Figure 5.6 shows the number of the selected parameters at each horizon.

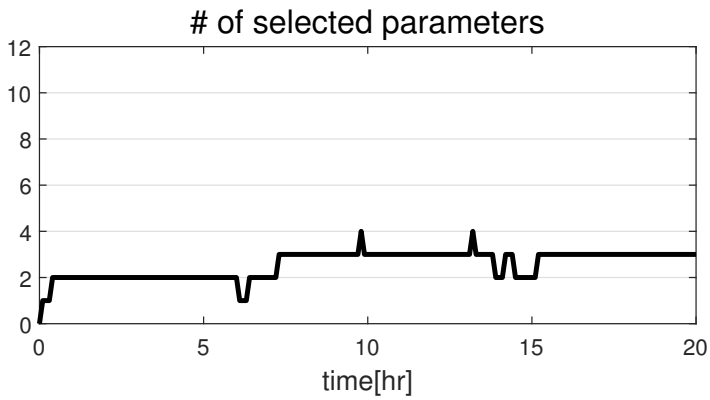
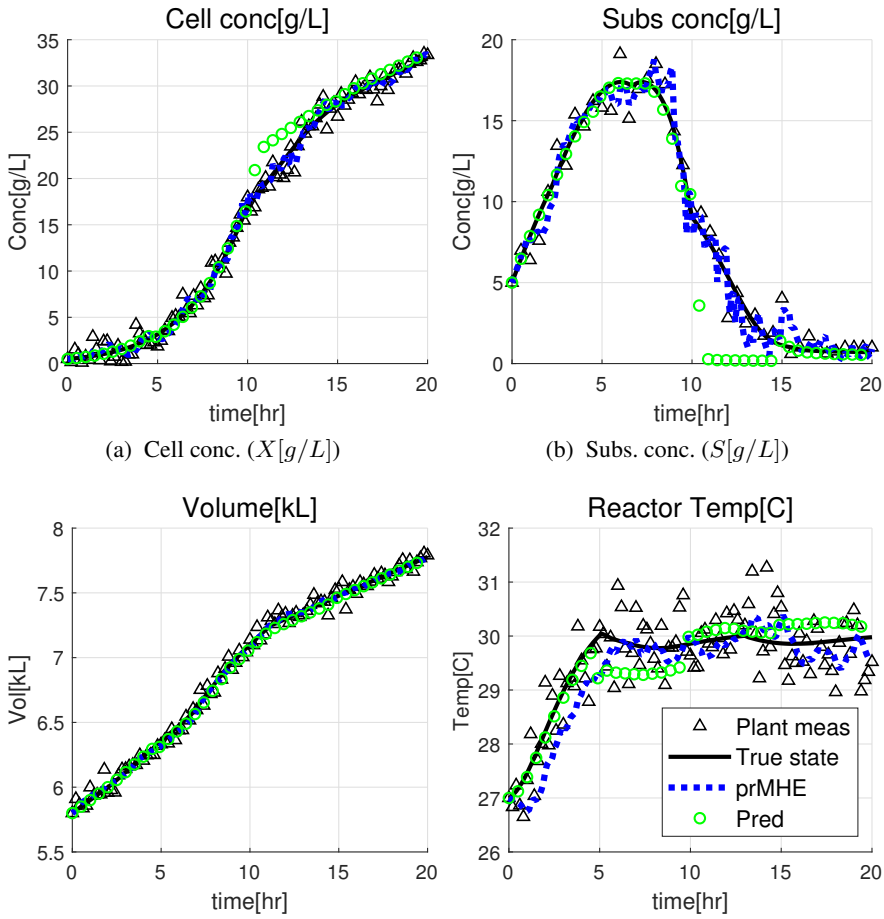


Figure 5.6: The number of the selected model parameters over time

As shown in Figure 5.6, only one to four model parameters are selected among the 12 model parameters. The number of selected model parameters tends to increase over the batch time, reflecting the growing intensity of the parameter drifts. Even if the parametric drifts are introduced to both U_c, μ_{max}, and E_g, these model parameters may not dominate the parametric sensitivity matrix when the state variables are not sensitive to these model parameters at some MHE horizon. In such cases, other model parameters can be updated, or even none of them are selected at all. The trajectories of the model parameters updated by prMHE are also shown in Figure 5.5.

As shown in Figure 5.5, prMHE (blue dotted line) is able to track the true parameter drifts while MHE(x, θ) (red dotted line) suffers from the over-fitting problem. However, prMHE updates non-perturbed parameters (θ_1 and θ_9) for some time steps and it is also unable to track the parameter trajectory of θ_7. These improper update problems occur to the lower-ranked parameters, owing to the improper setting of the threshold in the PSS procedure. Nevertheless, it is expected to provide better state predictions since prMHE is less affected by the over-fitting problem. Figure 5.7 shows the estimated state trajectories and the state prediction results of prMHE. Sums of NRMSEs for state estimation and prediction are also tabulated in Table 5.3.



(a) Cell conc. ($X[g/L]$)

(b) Subs. conc. ($S[g/L]$)

(c) Volume ($V[kL]$)

(d) Temp. ($T[C]$)

Figure 5.7: Estimated states and model prediction results of prMHE

As shown in Figure 5.7, prMHE maintains the accuracy of the state estimation, while updating less number of model parameters in each horizon. It even improves the state estimation accuracy for this example problem, with the sum of NRMSEs = 0.18. The accuracy of state predictions is also improved, especially at $t = 5[hr]$, where the two benchmark problems show poor prediction accuracy. However, as shown in Figures 5.7 (a) and (b), the accuracy of the model prediction at $t = 10[hr]$ is not improved over the results of two benchmarks. Since the prMHE updates the model parameters based on the set of past data, the change in θ_8 at $t = 10[hr]$ can not be used for updating the model parameters at $t = 10[hr]$, resulting in inaccurate state predictions at $t = 10[hr]$. Since it is impossible to predict the future changes in the model parameters, there exists some time lag until the parametric drifts of the process are captured by the prMHE. Even though prMHE can not track the true parameter drifts exactly to due the time lag, it provides improved state estimation and predictions by suppressing the ill-conditioning problem occurring in the model parameter estimation procedure of MHE(x, θ).

Chapter 6

Concluding remarks

When the target system is a non-stationary semi-batch process, a nonlinear model is often used to simulate the complex dynamics of the system. If the model parameters are properly estimated, various model-based methods can be successfully applied to the target system, appreciating the prediction performance of the nonlinear models. However, from a practical point of view, there are several difficulties in using nonlinear models. It is difficult to find an exact model structure for complex systems. Moreover, the available data are often limited, making it impossible to estimate all the model parameters. The limited data results in an ill-conditioning problem in parameter estimation (PE), deteriorating the prediction accuracy of the model. This situation is encountered quite frequently in a real-world problem. Under this condition, several types of nonlinear models, including a hybrid model, deterministic model, and simple stochastic model, are employed in this thesis to achieve various purposes: process modeling, dynamic optimization, and online state estimation.

In the first part of the thesis, we avoided the ill-conditioned PE problem by using the dynamic hybrid model, instead of using a common nonlinear model. The model-based dynamic optimization scheme is suggested for the hybrid model, together with the valid domain

constraints regulating the range of exploration. Meanwhile, the next two methods used nonlinear models that require PE to complete the modeling procedure. In the second part, the target system is modeled with simple stochastic differential equations (SDE) to manage the uncertainty in the model structure. We proposed a parameter subset selection (PSS) method for the stochastic model to resolve the ill-conditioned PE problem owing to the limited data. In the final chapter, the MHE estimator using the deterministic nonlinear model is suggested for online state and parameter estimation. PSS is applied to the basic formulation of the MHE, improving not only the accuracy of online state estimation but also the model prediction. Numerical examples are provided to demonstrate the efficacy of each method.

6.1 Summary of the contributions

The detailed summaries of each method are as follows:

In Chapter 3, we defined the valid domain for the dynamic hybrid model based on two complementary validity domain criteria of the static hybrid system; the convex hull criterion and the confidence interval criterion. Both valid domain criteria are transformed into sets of inequality constraints for a dynamic optimization problem. The iterative strategy for model update and optimization are also proposed to find improved optimal input trajectories under repetitive batch operations. It is also useful to handle the systems that have changing dynamics in batch-to-batch operations.

Chapter 4 proposed a ranking-based PSS method for stochastic models to solve the ill-conditioned PE problem, owing to the lack of information in the available data. By selecting an SDE model with

additive stochastic terms as a model form of the target system, the proposed method determines the optimal size of the parameter subset while simultaneously estimating the values of the selected parameters, magnitudes of the state disturbances, and measurement errors. As a key part of the proposed PSS method, the LAMLE-based PE method (gLAMLE) is proposed for the stochastic models. The proposed PE method manages the effects of batch-wise uncertainties on the estimation results and improves the convergence of the algorithm by adopting a learning rate.

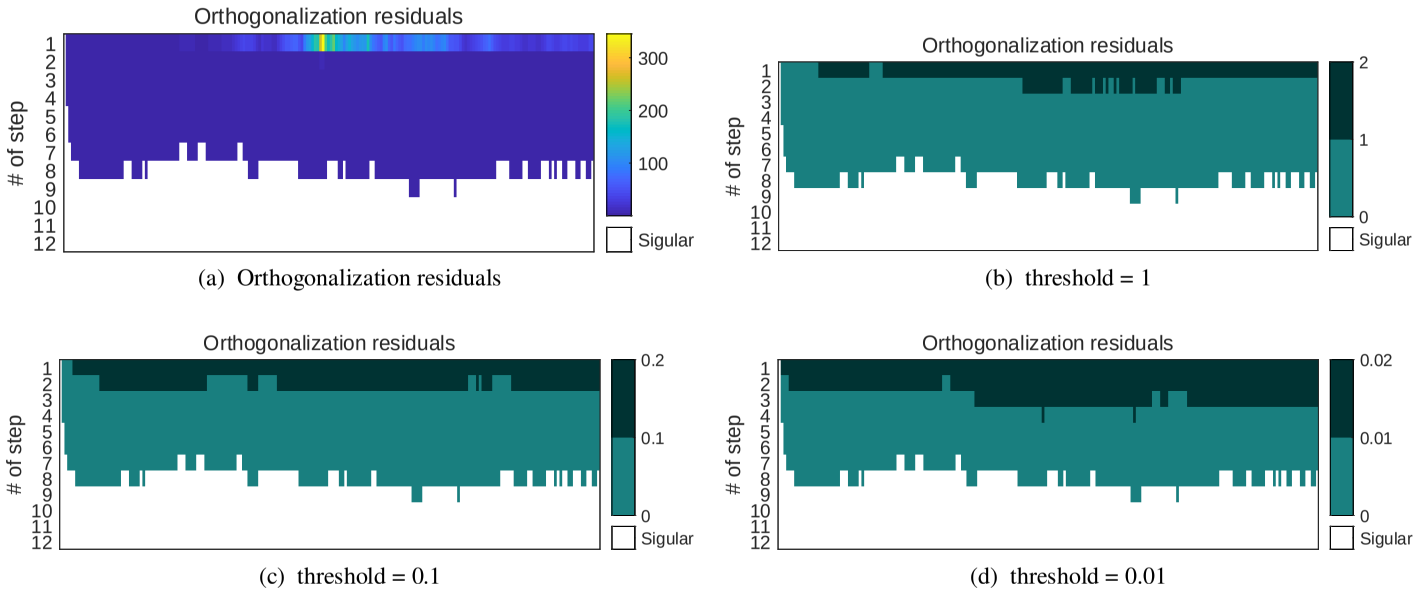
The ill-conditioned PE problem also occurs in the online estimation when using MHE, as the MHE estimates all the model parameters and state variables with the limited amount of measurements in the moving window of fixed size. In Chapter 5, a parameter ranking-based moving horizon estimation (prMHE) method was suggested to overcome the ill-conditioning problem in the MHE formulation. At each time step of prMHE, a ranked list of the parameters is calculated based on the scaled sensitivity matrix, and the subset of estimable parameters is determined. In the trajectory optimization of each time step, only the selected model parameters are updated along with the state variables. Consequently, it improves the accuracy of state estimation and model prediction by managing the ill-conditioning problem of online parameter estimation.

6.2 Limitations and Future works

For the successful application of the hybrid model-based methods, further considerations are required, including 1) when the model should be updated with the newly collected data, and 2) how much

data should be replaced at the start of each iteration. Without these considerations, we cannot expect accurate prediction performance of the hybrid model.

Throughout the thesis, the ranked list of the model parameters is determined by an orthogonalization-based method. Besides its simplicity, it has several drawbacks: it requires a user-defined threshold, and cannot consider the correlations between model parameters. Moreover, this method performs matrix calculation iteratively, making the overall algorithm computationally demanding. For example, Figure 6.1 shows that the size of the online selected parameter subset differs depending on the level of the pre-defined threshold. In Figures 6.1 (b)-(c), the number of selected parameters is indicated with the dark green cells, the light green cells indicate the number of ranked but not selected parameters, and the number of unranked parameters is drawn as the white cells.



(a) Orthogonalization residuals

(b) threshold = 1

(c) threshold = 0.1

(d) threshold = 0.01

Figure 6.1: Effect of the threshold on the size of selected parameter subset calculated by the orthogonalization method

However, as the proposed methods in the second and third part of the thesis do not restrict the type of parameter ranking methods, other parameter ranking methods could be used to improve the performance of the proposed PSS method. One way is to define a selection criterion based on statistical significance [62, 121]. Considering the characteristics of the target system, a user-defined selection criterion can be designed by using the parameter estimates and the estimated standard deviation of the model parameters, which are calculated during the online or offline parameter estimation procedure. It still requires the user-supplied thresholds on the significance level, but this method may reduce difficulty in choosing a proper threshold

The overall performance of the PSS procedure can decrease if the correlations between the model parameters are not negligible. In the second part, the numerical example shows that the correlation between the model parameters deteriorates the accuracy of parameter estimation. One way to resolve the correlation problem before starting the parameter estimation is to simplify the indistinguishable parts of the model into a single parameter, resulting in a lumped parameter model. First, structurally indistinguishable parameters should be lumped together with other parameters before performing the parameter estimation. After finishing the first parameter estimation, it is possible to calculate the estimated covariance matrix of the parameters. This matrix can be used to determine the correlations between parameters, which were not detected before performing the parameter estimation. However, simplifying the model structure to resolve the correlation problem may cause a structural uncertainty in the nonlinear model. It requires an online parameter update for accurate state estimation or model prediction, which is one of the problems that the

third part of the thesis tries to address.

For the MHE formulation, the overall performance of estimation can be improved by applying other methods for updating the covariance matrix of the arrival cost term. An NLP optimality-based method is one of the possible choices [48, 121]. Through this method, an exact covariance matrix of the model parameters is calculated based on the optimal KKT system of the formulated NLP. Moreover, the reduced Hessian matrix can also be used for the online PSS procedure in the prMHE algorithm. As the prMHE is defined with the augmented state, considering the state and model parameters simultaneously, the part for the model parameters in the whole reduced Hessian matrix can be easily separated and used as FIM in PSS procedures. Therefore, if the reduced Hessian of the augmented state can be extracted directly from the optimizer, it can be effective not only for improving the accuracy of state estimation but also for reducing the online computation of PSS used in the prMHE algorithm, not requiring further calculation for the sensitivity matrix at each time step.

Chapter A

Appendix

A.1 Calculate Hessian of $\psi_k(\theta)$

H_k is the Hessian of ψ_k evaluated at $\theta = \hat{\theta}$,

$$H_k = \left.\frac{\partial^2 \psi_k}{\partial\theta\partial\theta^\top}\right|_{\theta=\hat{\theta}}. \tag{A.1}$$

It can be derived by the matrix differentiation and the chain rule [122]. For $k = 1, \ldots, K$ sampling points and $n = 1, \ldots, N_k$ data sets, let, $\xi_{k,n} \in \mathbb{R}^{N_k}$, $y_{k,n} \in \mathbb{R}^{n_y}$, and $x_{k,n} \in \mathbb{R}^{n_x}$ where $n_y = n_x$ and $y = x + \epsilon$, respectively. Then, we can assume $x_{k,n} = f(\xi_{k,n})$ and $\xi_{k,n} = s(\theta)$ where $f : \mathbb{R}^{N_k} \mapsto \mathbb{R}^{N_x}$ and $s : \mathbb{R}^{N_p} \mapsto \mathbb{R}^{N_k}$. Given this, $g_{k,n}$ is calculated with

$$\frac{\partial y_{k,n}}{\partial\theta} = \frac{\partial \xi_{k,n}}{\partial\theta}\frac{\partial f(\xi_{k,n})}{\partial \xi_{i,n}} = \frac{\partial \xi_{k,n}}{\partial\theta}\frac{\partial y_{k,n}}{\partial \xi_{k,n}}, \tag{A.2}$$

$$\frac{\partial \xi_{k,n}}{\partial\theta} \in \mathbb{R}^{N_\theta \times N_k} \quad \frac{\partial f(\xi_{k,n})}{\partial \xi_{i,n}} \in \mathbb{R}^{N_k \times N_x} \quad \frac{\partial y_{k,n}}{\partial\theta} \in \mathbb{R}^{N_\theta \times N_x}. \tag{A.3}$$

Let $A_k = y_k^\top y_k \in \mathbb{R}^{n_y \times n_y}$, and $\psi_k = \det A_k \in \mathbb{R}$ for simplicity.

Then, each component of H_k in Eq. (A.1) is calculated by

$$\begin{aligned}
\left.\frac{\partial^2 \psi_k}{\partial \theta_{p_1} \partial \theta_{p_2}}\right|_{\hat{\theta}} &= \left.\frac{\partial^2 \det A_k}{\partial \theta_{p_1} \partial \theta_{p_2}}\right|_{\hat{\theta}, \tilde{u}_i}, \quad p_1, p_2 = 1, \ldots, N_\theta \\
&= \hat{\psi}_k \times \left[tr\left(A_k^{-1} \frac{\partial A_k}{\partial \theta_{p_1}} \right) tr\left(A_k^{-1} \frac{\partial A_k}{\partial \theta_{p_2}} \right) \right. \\
&\quad \left. + tr\left(A_k^{-1} \frac{\partial^2 A_k}{\partial \theta_{p_1} \partial \theta_{p_2}} \right) - tr\left(A_k^{-1} \frac{\partial A_k}{\partial \theta_{p_1}} A_k^{-1} \frac{\partial A_k}{\partial \theta_{p_2}} \right) \right] \Bigg|_{\hat{\theta}}.
\end{aligned} \tag{A.4}$$

Let $\tilde{\xi}_k = [\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,N_k}] \in \mathbb{R}^{K \times N_k}$. The first and second-order partial derivatives of A_k over θ are calculated by the chain rule as

$$\frac{\partial A_k}{\partial \theta_{p_1}} = \sum_{q=1}^{\tilde{N}_k} \frac{\partial A_k}{\partial \xi_q} \frac{\partial \xi_q}{\partial \theta_{p_1}} \in \mathbb{R}^{n_y \times n_y} \tag{A.5}$$

$$\frac{\partial^2 A_k}{\partial \theta_{p_1} \partial \theta_{p_2}} = \sum_{q=1}^{\tilde{N}_k} \left\{ \sum_{t=1}^{\tilde{N}_k} \frac{\partial^2 A_k}{\partial \xi_q \partial \xi_t} \frac{\partial \xi_t}{\partial \theta_{p_2}} \frac{\partial \xi_p}{\partial \theta_{p_1}} + \frac{\partial A_k}{\partial \xi_q} \frac{\partial^2 \xi_q}{\partial \theta_{p_1} \partial \theta_{p_2}} \right\} \tag{A.6}$$

$$\frac{\partial A_k}{\partial \xi_q} = \frac{\partial y_k^\top y_k}{\partial \xi_q} = 2 y_i^\top \frac{\partial y_i}{\partial \xi_q} \in \mathbb{R}^{N_x \times N_x} \tag{A.7}$$

$$\frac{\partial^2 A_k}{\partial \xi_q \partial \xi_t} = 2 \left(\frac{\partial y_k}{\partial \xi_q}^\top \frac{\partial y_k}{\partial \xi_t} + y_k^\top \frac{\partial^2 y_k}{\partial \xi_q \partial \xi_t} \right) \tag{A.8}$$

where $\partial \xi / \partial \theta$ and $\partial^2 \xi / \partial \theta^2$ can be calculated while training the black-box models.

A.2 Minimizing condition number of matrix by scaling [1]

For a full-rank matrix $H \in \mathbb{R}^{p\times q}$, with $p \leq q$, the problem of minimizing the condition number of H is

$$\min_{L,R} \quad \kappa(LHR) \tag{A.9a}$$

$$s.t. \quad L \in \mathbb{R}^{p\times p}, \quad R \in \mathbb{R}^{q\times q} \tag{A.9b}$$

where L and R are the optimization variables, assumed to be diagonal and nonsingular matrices. This problem can be transformed into a GEVP,

$$\min_{P,Q} \quad \gamma^2 \tag{A.10a}$$

$$s.t. \quad P > 0, \quad P \in \mathbb{R}^{p\times p}, \tag{A.10b}$$

$$Q > 0, \quad Q \in \mathbb{R}^{q\times q} \tag{A.10c}$$

$$Q \leq H^\top PH \leq \gamma^2 Q \tag{A.10d}$$

where $\gamma > 1$, $P = L^\top L$, and $Q = (RR^\top)^{-1}$. Since L and R are diagonal, nonsingular matrices, P and Q as well. Eq. (A.10) can be solved by LMI solvers, such as CVX [123] or a matlab function 'gevp' in Robust Control Toolbox. From the solution, $L = P^{1/2}$ and $R = Q^{-1/2}$, respectively.

A.3 Simulation of SDE

First of all, one thing to have in mind is that Eq. (4.2) is a convention in notation and does not mean a real differentiation [91].

With Eq. (4.2), let us first assume b and σ are time-independent, Lipshitz continuous and bounded. And also assume that the initial condition, X_0 is non-random. Then, we can write the ode whose solution is X_t^n,

$$\frac{d}{dt}X_t^n = b(X_t^n) + \sigma(X_t^n)\xi_t^n, \quad X_0^n = X_0, \tag{A.11}$$

,that is driven by an approximated white noise, ξ_t^n, which satisfies that

$$\sup_{t\in[0,T]} \|W_t - W_t^n\| \overset{n\to\infty}{\longrightarrow} 0 \quad a.s, \qquad W_t^n = \int_0^t \xi_s^n ds. \tag{A.12}$$

Meanwhile, let X_t be the solution of the SDE with the $Itô-correction$ term

$$dX_t = \tilde{b}(X_t)dt + \sigma(X_t)dW_t \tag{A.13}$$

$$\tilde{b}^i(x) = b^i(x) + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{m}\frac{\partial\sigma^{ik}(x)}{\partial x^j}\sigma^{jk}(x). \tag{A.14}$$

For the solution of each equation, X_t^n and X_t is derived by the Wong-Zakai theorem [91],

$$\mathbb{E}\left[\sup_{t\in[0,T]} |X_t^n - X_t|^2\right] \overset{n\to\infty}{\longrightarrow} 0 \tag{A.15}$$

for any $T < \infty$.

Since the form of SDE that is interested in this study is that the SDE with a constant σ, so let $\sigma(X_t) = \sigma$, then it can be described with simpler expressions, comparing to Eq. (A.14), as

$$dX_t = b(X_t)dt + \sigma dW_t, \quad \sup_{t\in[0,T]} \|X_t^n - X_t\| \overset{n\to\infty}{\longrightarrow} 0 \quad a.s. \tag{A.16}$$

where the $Itô - correction$ term does not need to be considered.

A SDE model that chosen to model systems takes the form of Eq. (A.11) with constant σ,

$$\frac{dx(t)}{dt} = f(x(t), u(t), \theta) + \eta(t), \tag{A.17}$$

$$\mathbb{E}\eta(t_1)\eta(t_2) = Q\delta(t_2 - t_1), \quad x(0) = x_o \tag{A.18}$$

where x is a state variable, u is a input variable, θ is the vector of model parameters, and η is zero-mean white-noise process with constant covariance matrix (Q). And this equation is rewritten, without any problem, into

$$dx(t) = f(x(t), u(t), \theta)dt + QdW_t, \quad x(0) = x_o \tag{A.19}$$

Stochastic differential equations seldom admit analytical soluiton, like their non-random counterparts. Therefore, numerical approaches to simulate such systems are also essential. Meaning of simulation of SDEs is to simulate (approximate) sample paths of the SDE with proper distribution. There are a lot of methods to numerically simulate the SDE, including the Euler-Maruyama method, the Milstein scheme, the stochastic Runge-Kutta methods, etc. As well as detailed

description of the SDE, check [91, 90] for further information.

A.4 Laplace approximation

Laplace approximation (LA) is one method of the asymptotic expansions of integrals, considering the integral of a 1-d random variable x (originally starts from much general integrand in [110], but let us use following simple integral directly)

$$\begin{aligned}\int_a^b \frac{1}{\sqrt{2\pi}} \exp\{-h(x)\}dx \\ = \exp\{-h(x_0)\} \int_{-\infty}^{\infty} \frac{x^{'}(t)}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt\end{aligned} \tag{A.20}$$

where x_0 is mode of $h(x)$, and writing $h(x) - h(x_0) = \frac{1}{2}t^2$. By expanding $h(x)$,

$$\frac{1}{2}t^2 = \frac{1}{2}(x - x_0)^2 h^{''}(x_0) + \frac{1}{6}(x - x_0)^3 h^{(3)}(x_0) + \ldots, \tag{A.21}$$

and let us assume that selecting only upto first term of Eq. (A.21). With Eq. (A.20), assumed Eq. (A.21) and let $p(x) = -h(x)$, it is possible to derive the LA form to approximate the multidimensional integral,

$$\begin{aligned}\ln \int e^{p(\boldsymbol{x})} d\boldsymbol{x} \approx & p(\hat{\boldsymbol{x}}) - \frac{1}{2} \ln det \left(\left\{ \frac{-\partial^2 p}{\partial \boldsymbol{x} \partial \boldsymbol{x}^\top} \right\} |_{\hat{\boldsymbol{x}}} \right) \\ & + \frac{1}{2} dim(\boldsymbol{x}) \ln(2\pi),\end{aligned} \tag{A.22}$$

where $\hat{\boldsymbol{x}}$ is the vector of realizations of random variables that maximizes $p(x)$.

Bibliography

[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15. Siam, 1994.

[2] K. Z. Yao, B. M. Shaw, B. Kou, K. B. McAuley, and D. Bacon, "Modeling ethylene/butene copolymerization with multi-site catalysts: parameter estimability and experimental design," *Polymer Reaction Engineering*, vol. 11, no. 3, pp. 563–588, 2003.

[3] S. Wu, K. A. McLean, T. J. Harris, and K. B. McAuley, "Selection of optimal parameter set using estimability analysis and mse-based model-selection criterion," *International Journal of Advanced Mechatronic Systems*, vol. 3, no. 3, pp. 188–197, 2011.

[4] P. Mhaskar, A. Garg, and B. Corbett, *Modeling and Control of Batch Processes: Theory and Applications*. Springer, 2018.

[5] J. Villadsen, J. Nielsen, and G. Lidén, *Bioreaction engineering principles*. Springer Science & Business Media, 2011.

[6] A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, 2019.

[7] M. Chidambaram, *Mathematical Modelling and Simulation in Chemical Engineering*. Cambridge University Press, 2018.

[8] M. Glacken, C. Huang, and A. Sinskey, "Mathematical descriptions of hybridoma culture kinetics. iii. simulation of fed-batch bioreactors," *Journal of biotechnology*, vol. 10, no. 1, pp. 39–65, 1989.

[9] A. Cruz, A. Silva, M. Araujo, R. Giordano, and C. Hokka, "Modelling and optimization of the cephalosporin c production bioprocess in a fed-batch bioreactor with invert sugar as substrate," *Chemical Engineering Science*, vol. 54, no. 15-16, pp. 3137–3142, 1999.

[10] M. Thilakavathi, T. Basak, and T. Panda, "Modeling of enzyme production kinetics," *Applied microbiology and biotechnology*, vol. 73, no. 5, pp. 991–1007, 2007.

[11] P. Englezos and N. Kalogerakis, *Applied parameter estimation for chemical engineers*. CRC Press, 2000.

[12] D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.

[13] M. L. Thompson and M. A. Kramer, "Modeling chemical processes using prior knowledge and neural networks," *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994.

[14] R. Oliveira, R. M. Portela, and M. von Stosch, "Hybrid model structures for knowledge integration," in *Hybrid Modeling in Process Industries*, pp. 13–35, CRC Press, 2018.

[15] S. Zendehboudi, N. Rezaei, and A. Lohi, "Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review," *Applied Energy*, vol. 228, pp. 2539–2566, 2018.

[16] J. K. Møller, G. Goranović, N. K. Poulsen, and H. Madsen, "Physical-stochastic (greybox) modeling of slugging," *Ifac-papersonline*, vol. 51, no. 8, pp. 197–202, 2018.

[17] B. Beykal, F. Boukouvala, C. A. Floudas, N. Sorek, H. Zalavadia, and E. Gildin, "Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations," *Computers & Chemical Engineering*, vol. 114, pp. 99–110, 2018.

[18] M. S. F. Bangi and J. S.-I. Kwon, "Deep hybrid modeling of chemical process: Application to hydraulic fracturing," *Computers & Chemical Engineering*, vol. 134, p. 106696, 2020.

[19] C. Ng and M. Hussain, "Hybrid neural network—prior knowledge model in temperature control of a semi-batch polymerization process," *Chemical Engineering and Processing: Process Intensification*, vol. 43, no. 4, pp. 559–570, 2004.

[20] J.-S. Chang, S.-C. Lu, and Y.-L. Chiu, "Dynamic modeling of batch polymerization reactors via the hybrid neural-network rate-function approach," *Chemical Engineering Journal*, vol. 130, no. 1, pp. 19–28, 2007.

[21] A. Teixeira, A. Cunha, J. Clemente, J. Moreira, H. Cruz, P. Alves, M. Carrondo, and R. Oliveira, "Modelling and optimization of a recombinant bhk-21 cultivation process using hybrid grey-box systems," *Journal of biotechnology*, vol. 118, no. 3, pp. 290–303, 2005.

[22] P. Setoodeh, A. Jahanmiri, and R. Eslamloueyan, "Hybrid neural modeling framework for simulation and optimization of diauxie-involved fed-batch fermentative succinate production," *Chemical engineering science*, vol. 81, pp. 57–76, 2012.

[23] S. James, R. Legge, and H. Budman, "Comparative study of black-box and hybrid estimation methods in fed-batch fermentation," *Journal of process control*, vol. 12, no. 1, pp. 113–121, 2002.

[24] F. Boukouvala, M. F. Hasan, and C. A. Floudas, "Global optimization of general constrained grey-box models: new method and its application to constrained pdes for pressure swing adsorption," *Journal of Global Optimization*, vol. 67, no. 1-2, pp. 3–42, 2017.

[25] J. R. Banga and E. Balsa-Canto, "Parameter estimation and optimal experimental design," *Essays in biochemistry*, vol. 45, pp. 195–210, 2008.

[26] P. Englezos and N. Kalogerakis, *Applied parameter estimation for chemical engineers*. CRC Press, 2000.

[27] B. Kim and J. H. Lee, "Parameter subset selection and biased estimation for a class of ill-conditioned estimation problems," *Journal of Process Control*, vol. 81, pp. 65–75, 2019.

[28] A. Gábor and J. R. Banga, "Robust and efficient parameter estimation in dynamic models of biological systems," *BMC systems biology*, vol. 9, no. 1, p. 74, 2015.

[29] G. Papadopoulos, P. J. Edwards, and A. F. Murray, "Confidence estimation methods for neural networks: A practical comparison," *IEEE transactions on neural networks*, vol. 12, no. 6, pp. 1278–1287, 2001.

[30] J. Glassey and M. von Stosch, "Benefits and challenges of hybrid modeling in the process industries: An introduction," in *Hybrid Modeling in Process Industries*, pp. 1–12, CRC Press, 2018.

[31] O. Kahrs and W. Marquardt, "The validity domain of hybrid models and its application in process optimization," *Chemical Engineering and Processing: Process Intensification*, vol. 46, no. 11, pp. 1054–1066, 2007.

[32] J. Bae, H. J. Lee, D. H. Jeong, and J. M. Lee, "Construction of valid domain for hybrid model and its application to dynamic optimization with controlled exploration," *Industrial & Engineering Chemistry Research*, 2020.

[33] A. Holmberg, "On the practical identifiability of microbial growth models incorporating michaelis-menten type nonlinearities," *Mathematical Biosciences*, vol. 62, no. 1, pp. 23–43, 1982.

[34] K. A. McLean and K. B. McAuley, "Mathematical modelling of chemical processes—obtaining the best model predictions and parameter estimates using identifiability and estimability procedures," *The Canadian Journal of Chemical Engineering*, vol. 90, no. 2, pp. 351–366, 2012.

[35] R. Bellman and K. J. Åström, "On structural identifiability," *Mathematical biosciences*, vol. 7, no. 3-4, pp. 329–339, 1970.

[36] J. E. Jiménez-Hornero, I. M. Santos-Dueñas, I. Garcı, *et al.*, "Structural identifiability of a model for the acetic acid fermentation process," *Mathematical biosciences*, vol. 216, no. 2, pp. 154–162, 2008.

[37] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice with MATLAB*. John Wiley & Sons, 2014.

[38] A. Gelb, *Applied optimal estimation*. MIT press, 1974.

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

[40] J. S.-I. Kwon, M. Nayhouse, G. Orkoulas, D. Ni, and P. D. Christofides, "A method for handling batch-to-batch parametric drift using moving horizon estimation: application to run-to-run mpc of batch crystallization," *Chemical Engineering Science*, vol. 127, pp. 210–219, 2015.

[41] D. A. Allan and J. B. Rawlings, "Moving horizon estimation," in *Handbook of Model Predictive Control*, pp. 99–124, Springer, 2019.

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

[43] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation—an introductory overview," in *Advances in control*, pp. 391–449, Springer, 1999.

[44] P. Kühl, M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock, "A real-time algorithm for moving horizon state and parameter estimation," *Computers & chemical engineering*, vol. 35, no. 1, pp. 71–83, 2011.

[45] G. E. Box and N. R. Draper, "The bayesian estimation of common parameters from several responses," *Biometrika*, vol. 52, no. 3-4, pp. 355–365, 1965.

[46] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[47] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[48] V. M. Zavala, C. D. Laird, and L. T. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *Journal of Process Control*, vol. 18, no. 9, pp. 876–884, 2008.

[49] R. López-Negrete and L. T. Biegler, "A moving horizon estimator for processes with multi-rate measurements: A nonlinear programming sensitivity approach," *Journal of Process Control*, vol. 22, no. 4, pp. 677–688, 2012.

[50] C. C. Qu and J. Hahn, "Computation of arrival cost for moving horizon estimation via unscented kalman filtering," *Journal of Process Control*, vol. 19, no. 2, pp. 358–363, 2009.

[51] S. Ungarala, "Computing arrival cost parameters in moving horizon estimation using sampling based filters," *Journal of Process Control*, vol. 19, no. 9, pp. 1576–1588, 2009.

[52] M. Rodriguez-Fernandez, J. A. Egea, and J. R. Banga, "Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems," *BMC bioinformatics*, vol. 7, no. 1, p. 483, 2006.

[53] R. Li, M. A. Henson, and M. J. Kurtz, "Selection of model parameters for off-line parameter estimation," *IEEE Transactions on control systems technology*, vol. 12, no. 3, pp. 402–412, 2004.

[54] B. F. Lund and B. A. Foss, "Parameter ranking by orthogonalization—applied to nonlinear mechanistic models," *Automatica*, vol. 44, no. 1, pp. 278–281, 2008.

[55] S. R. Weijers and P. A. Vanrolleghem, "A procedure for selecting best identifiable parameters in calibrating activated sludge model no. 1 to full-scale plant data," *Water science and technology*, vol. 36, no. 5, pp. 69–79, 1997.

[56] D. A. Belsley, *Conditioning diagnostics: Collinearity and weak data in regression*, vol. 396. Wiley New York, 1991.

[57] M. Sanayei and C. N. Javdekar, "Sensor placement for parameter estimation of structures using fisher information matrix," in *Applications of Advanced Technologies in Transportation (2002)*, pp. 385–386, 2002.

[58] V. C. Machado, G. Tapia, D. Gabriel, J. Lafuente, and J. A. Baeza, "Systematic identifiability study based on the fisher information matrix for reducing the number of parameters calibration of an activated sludge model," *Environmental Modelling & Software*, vol. 24, no. 11, pp. 1274–1284, 2009.

[59] A. Poyton, M. S. Varziri, K. B. McAuley, P. J. McLellan, and J. O. Ramsay, "Parameter estimation in continuous-time dynamic models using principal differential analysis," *Computers & chemical engineering*, vol. 30, no. 4, pp. 698–708, 2006.

[60] K. A. McLean, S. Wu, and K. B. McAuley, "Mean-squared-error methods for selecting optimal parameter subsets for estimation," *Industrial & Engineering Chemistry Research*, vol. 51, no. 17, pp. 6105–6115, 2012.

[61] Y. Chu, Z. Huang, and J. Hahn, "Improving prediction capabilities of complex dynamic models via parameter selection and estimation," *Chemical Engineering Science*, vol. 64, no. 19, pp. 4178–4185, 2009.

[62] T. Barz, M. Peñuela, A. Villegas, S. Ochoa, G. Wozny, *et al.*, "Model-based identifiable parameter determination applied to a simultaneous saccharification and fermentation process model for bio-ethanol production," *Biotechnology progress*, vol. 29, no. 4, pp. 1064–1082, 2013.

[63] P. Xu, "Truncated svd methods for discrete linear ill-posed problems," *Geophysical Journal International*, vol. 135, no. 2, pp. 505–514, 1998.

[64] A. Gábor, A. F. Villaverde, and J. R. Banga, "Parameter identifiability analysis and visualization in large-scale kinetic models of biosystems," *BMC systems biology*, vol. 11, no. 1, p. 54, 2017.

[65] A. Saltelli, K. Chan, and M. Scott, "Sensitivity analysis john wiley & sons publishers," *Probability and Statistics series*, 2000.

[66] J. R. Leis and M. A. Kramer, "The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 14, no. 1, pp. 45–60, 1988.

[67] D. E. Thompson, K. B. McAuley, and P. J. McLellan, "Parameter estimation in a simplified mwd model for hdpe produced by a ziegler-natta catalyst," *Macromolecular Reaction Engineering*, vol. 3, no. 4, pp. 160–177, 2009.

[68] J. Řehoř and V. Havlena, "A practical approach to grey-box model identification," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 10776–10781, 2011.

[69] M. Von Stosch, R. Oliveira, J. Peres, and S. F. de Azevedo, "Hybrid semi-parametric modeling in process systems engineering: Past, present and future," *Computers & Chemical Engineering*, vol. 60, pp. 86–101, 2014.

[70] S. Zendehboudi, N. Rezaei, and A. Lohi, "Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review," *Applied energy*, vol. 228, pp. 2539–2566, 2018.

[71] R. Bardini, G. Politano, A. Benso, and S. Di Carlo, "Multi-level and hybrid modelling approaches for systems biology," *Computational and structural biotechnology journal*, vol. 15, pp. 396–402, 2017.

[72] M. von Stosch, "Hybrid models and experimental design," in *Hybrid Modeling in Process Industries*, pp. 37–61, CRC Press, 2018.

[73] D. Kim, H.-S. Oh, and I.-C. Moon, "Black-box modeling for aircraft maneuver control with bayesian optimization," *International Journal of Control, Automation and Systems*, vol. 17, no. 6, pp. 1558–1568, 2019.

[74] S. Khadraoui and H. Nounou, "A nonparametric approach to design fixed-order controllers for systems with constrained input," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2870–2877, 2018.

[75] H. C. Lv and D. Y. Tian, "Modeling of the phase equilibria of aqueous two-phase systems using three-dimensional neural network," *Korean Journal of Chemical Engineering*, vol. 34, no. 1, pp. 170–178, 2017.

[76] M. von Stosch, R. Oliveira, J. Peres, and S. F. De Azevedo, "A novel identification method for hybrid (n) pls dynamical systems with application to bioprocesses," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10862–10874, 2011.

[77] R. Romijn, L. Özkan, S. Weiland, J. Ludlage, and W. Marquardt, "A grey-box modeling approach for the reduction of nonlinear systems," *Journal of Process Control*, vol. 18, no. 9, pp. 906–914, 2008.

[78] D. Solle, B. Hitzmann, C. Herwig, M. Pereira Remelhe, S. Ulonska, L. Wuerth, A. Prata, and T. Steckenreiter, "Between the poles of data-

driven and mechanistic modeling for process operation," *Chemie Ingenieur Technik*, vol. 89, no. 5, pp. 542–561, 2017.

[79] M. Agarwal, "Combining neural and conventional paradigms for modelling, prediction and control," *International Journal of Systems Science*, vol. 28, no. 1, pp. 65–81, 1997.

[80] T. Mrziglod and A. Schuppert, "Hybrid model identification and discrimination with practical examples from the chemical industry," in *Hybrid Modeling in Process Industries*, pp. 63–88, CRC Press, 2018.

[81] O. Kahrs and W. Marquardt, "Incremental identification of hybrid process models," *Computers & Chemical Engineering*, vol. 32, no. 4-5, pp. 694–705, 2008.

[82] M. von Stosch, J.-M. Hamelink, and R. Oliveira, "Hybrid modeling as a qbd/pat tool in process development: an industrial e. coli case study," *Bioprocess and biosystems engineering*, vol. 39, no. 5, pp. 773–784, 2016.

[83] M. von Stosch, S. Davy, K. Francois, V. Galvanauskas, J.-M. Hamelink, A. Luebbert, M. Mayer, R. Oliveira, R. O'Kennedy, P. Rice, *et al.*, "Hybrid modeling for quality by design and pat-benefits and challenges of applications in biopharmaceutical industry," *Biotechnology journal*, vol. 9, no. 6, pp. 719–726, 2014.

[84] D. Bates and D. Watts, *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons Inc, 1988.

[85] G. A. Seber and A. J. Lee, *Linear regression analysis*, vol. 329. John Wiley & Sons, 2012.

[86] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[87] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[88] J. Bae, D. H. Jeong, and J. M. Lee, "Ranking-based parameter subset selection for nonlinear dynamics with stochastic disturbances under limited data," *Industrial & Engineering Chemistry Research*, 2020.

[89] H. Verdejo, A. Awerkin, W. Kliemann, and C. Becker, "Modelling uncertainties in electrical power systems with stochastic differential equations," *International Journal of Electrical Power & Energy Systems*, vol. 113, pp. 322–332, 2019.

[90] S. Särkkä and A. Solin, *Applied stochastic differential equations*, vol. 10. Cambridge University Press, 2019.

[91] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[92] S. Donnet and A. Samson, "A review on estimation of stochastic differential equations for pharmacokinetic/pharmacodynamic models," *Advanced drug delivery reviews*, vol. 65, no. 7, pp. 929–939, 2013.

[93] L. Ferrante, S. Bompadre, and L. Leone, "A stochastic compartmental model with long lasting infusion," *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, vol. 45, no. 2, pp. 182–194, 2003.

[94] R. Kupferman and A. Stuart, "Fitting sde models to nonlinear kac–zwanzig heat bath models," *Physica D: Nonlinear Phenomena*, vol. 199, no. 3-4, pp. 279–316, 2004.

[95] L. Ferrante, S. Bompadre, and L. Leone, "A stochastic compartmental model with long lasting infusion," *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, vol. 45, no. 2, pp. 182–194, 2003.

[96] J. Timmer, "Parameter estimation in nonlinear stochastic differential equations," *Chaos, Solitons & Fractals*, vol. 11, no. 15, pp. 2571–2578, 2000.

[97] H. Karimi and K. B. McAuley, "A maximum-likelihood method for estimating parameters, stochastic disturbance intensities and measurement noise variances in nonlinear dynamic models with process disturbances," *Computers & Chemical Engineering*, vol. 67, pp. 178–198, 2014.

[98] S. Walker, "An em algorithm for nonlinear random effects models," *Biometrics*, pp. 934–944, 1996.

[99] E. Kuhn and M. Lavielle, "Maximum likelihood estimation in nonlinear mixed effects models," *Computational Statistics & Data Analysis*, vol. 49, no. 4, pp. 1020–1038, 2005.

[100] H. Karimi and K. B. McAuley, "An approximate expectation maximisation algorithm for estimating parameters in nonlinear dynamic models with process disturbances," *The Canadian Journal of Chemical Engineering*, vol. 92, no. 5, pp. 835–850, 2014.

[101] H. Karimi and K. B. McAuley, "Bayesian objective functions for estimating parameters in nonlinear stochastic differential equation models with limited data," *Industrial & Engineering Chemistry Research*, vol. 57, no. 27, pp. 8946–8961, 2018.

[102] A. Golightly and D. J. Wilkinson, "Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo," *Interface focus*, vol. 1, no. 6, pp. 807–820, 2011.

[103] M. Varziri, K. McAuley, and P. McLellan, "Parameter and state estimation in nonlinear stochastic continuous-time dynamic models with unknown disturbance intensity," *The Canadian Journal of Chemical Engineering*, vol. 86, no. 5, pp. 828–837, 2008.

[104] A. R. Pedersen, "A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations," *Scandinavian journal of statistics*, pp. 55–71, 1995.

[105] A. Golightly and D. J. Wilkinson, "Markov chain monte carlo algorithms for sde parameter estimation," *Learning and Inference for Computational Systems Biology*, pp. 253–276, 2010.

[106] U. Picchini and S. Ditlevsen, "Practical estimation of high dimensional stochastic differential mixed-effects models," *Computational Statistics & Data Analysis*, vol. 55, no. 3, pp. 1426–1444, 2011.

[107] Y. Ait-Sahalia *et al.*, "Closed-form likelihood expansions for multivariate diffusions," *The Annals of Statistics*, vol. 36, no. 2, pp. 906–937, 2008.

[108] B. M. Bibby and M. Sørensen, "Martingale estimation functions for discretely observed diffusion processes," *Bernoulli*, pp. 17–39, 1995.

[109] O. E. Barndorff-Nielsen, "Asymptotic techniques; for use in statistics," tech. rep., 1989.

[110] J. L. Jensen, *Saddlepoint approximations*. No. 16, Oxford University Press, 1995.

[111] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: a generalized smoothing approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 5, pp. 741–796, 2007.

[112] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*, vol. 27. springer-verlag New York, 1978.

[113] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[114] T. Turányi, "Sensitivity analysis of complex kinetic systems. tools and applications," *Journal of mathematical chemistry*, vol. 5, no. 3, pp. 203–248, 1990.

[115] A. Poyton, M. S. Varziri, K. B. McAuley, P. J. McLellan, and J. O. Ramsay, "Parameter estimation in continuous-time dynamic models using principal differential analysis," *Computers & chemical engineering*, vol. 30, no. 4, pp. 698–708, 2006.

[116] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[117] Y. Chu, Z. Huang, and J. Hahn, "Improving prediction capabilities of complex dynamic models via parameter selection and estimation," *Chemical Engineering Science*, vol. 64, no. 19, pp. 4178–4185, 2009.

[118] A. Küpper, M. Diehl, J. P. Schlöder, H. G. Bock, and S. Engell, "Efficient moving horizon state and parameter estimation for smb processes," *Journal of Process Control*, vol. 19, no. 5, pp. 785–802, 2009.

[119] S. Krämer and R. Gesthuisen, "Multirate state estimation using moving horizon estimation," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 1–6, 2005.

[120] A. Liu, W.-A. Zhang, M. Z. Chen, and L. Yu, "Moving horizon estimation for mobile robots with multirate sampling," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1457–1467, 2016.

[121] W. Chen and L. T. Biegler, "Reduced hessian based parameter selection and estimation with simultaneous collocation approach," *AIChE Journal*, p. e16242, 2020.

[122] K. B. Petersen *et al.*, "The matrix cookbook,"

[123] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." `http://cvxr.com/cvx`, Mar. 2014.

초 록

산업적으로 활용되는 회분식 공정은 균일한 제품 생산을 위하여 사전에 결정된 입력 궤적을 따라 비 정상상태로 운전된다는 특징이 있다. 회분식 공정의 이러한 특성은 공정의 디지털 트윈을 설계하는 데 필수적인 단계인 공정의 수학적 모델링 과정에 여러 어려움을 야기한다. 우선, 공정의 복잡한 거동을 모사하기 위해 비선형 모델을 사용해야 하며, 그에 해당하는 모델 매개변수를 추정하는 과정이 필요하다. 이때, 고정된 입력 궤적에 따라 운전된 공정의 데이터에는 파라미터 추정에 사용할 정보량이 충분하지 않기 때문에, 주어진 데이터의 양과 별개로 파라미터 추정에 불량 조건(ill-condition) 문제가 발생한다. 불량 조건 파라미터 추정 문제가 발생하면, 모든 모델 매개변수를 정확히 추정하는 것은 불가능하다. 더욱이, 완성된 모델 기반의 기법을 공정에 곧 바로 적용할 수 있는 것도 아니다. 실제 공정은 알려지지 않은 외란에 의하여 계속해서 영향을 받으며, 그로 인해 모델-공정 부조화(model-plant mismatch)가 발생하기 때문이다. 회분식 공정을 모사하는 비선형 모델의 경우, 모델 구조적 불확실성으로 인하여 이 문제가 더욱 부각된다. 이런 문제들은 생산 규모의 공정을 모델링할 때 자주 발생하며, 공정의 디지털 트윈을 구축하기 위해서 반드시 해결되어야 한다. 본 논문에서는, 이러한 회분식 공정의 특성을 고려할 수 있는 모델 기반의 접근법을 제안한다.

첫 번째로, 동적 복합 모델(dynamic hybrid model) 기반의 모델링 및 동적 최적화 기법을 제시한다. 복합 모델은 데이터가 부족한 상황에서 공정에 대한 사전 정보 또한 불완전하여 공정을 모사할 모델의 구조에 대한 확신이 없는 상황에서 유용하다. 본 논문의 첫 부분은 복합 모델을 사용한 공정 해석이 타당한 영역인 유효 영역

(valid domain)을 정의하고, 그것을 동적 최적화의 제약 조건으로 활용할 수 있는 방법 및 좁은 유효 영역의 한계를 극복하기 위한 반복 알고리즘을 제안한다.

두 번째 방법론은 제일원칙 모델(first-principle model)을 기반으로 모델의 불확실성에 대처하기 위하여, 가산적 통계항을 갖는 확률 미분 방정식(stochastic differential equation) 모델을 사용한다. 이 모델을 기반으로 하여, 부족한 데이터로 인하여 발생한 불량 조건 매개변수 추정 문제를 다룰 수 있는 매개변수 부분집합 선택 (parameter subset selection) 알고리즘을 제시한다. 이 과정에서, 비선형 모델의 모델 매개변수 뿐 만 아니라, 가산적 통계항의 크기를 함께 추정할 수 있는 매개변수 추정 기법도 함께 제시하며 이는 매개변수 부분집합 선택 알고리즘 내에서 반복적으로 사용된다.

마지막으로, 완성된 비선형 제일원칙 모델을 사용하여 모델-공정 부조화를 해결할 수 있는 최적화 기반의 실시간 상태 및 매개변수 추정 기법을 제시한다. 제한된 양의 데이터를 사용하여 많은 매개변수를 실시간으로 추정해야 한다면, 실시간 상태 및 매개변수 추정 기법을 사용하는 과정에서도 불량 조건 문제가 발생한다. 이 문제를 해결하기 위해서, 실시간 매개변수 부분집합 선택 기법을 실시간 추정 기법인 moving horizon estimation에 도입하여 실시간 매개변수 추정 문제에서 발생하는 불량 조건 문제를 해결할 수 있는 알고리즘을 제시한다.

제안된 기법들의 성능을 검증하기 위하여, 각 단원의 마지막에 예제가 수록되어 있다. 각 예제는 산업적 반 회분식 생물 반응기를 모사하는 가상 공정을 사용하며, 이 시스템은 비선형적 거동, 비 정상상태 운전, 그리고 측정 데이터에 포함된 제한적 정보라는 특성을

모두 가진다.

주요어 : 디지털 트윈, 반회분식 공정, 하이브리드 모델, 동적 최적화, 파라미터 추정, 상태 추정
학번 : 2015-21066