Identifying Linear Parameter-Varying State Space Models

Estimating System Dynamics and Scheduling Variables From State Sequences and Input-Output Measurements

Knut Roar Lende

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Identifying Linear Parameter-Varying State Space Models

Estimating System Dynamics and Scheduling Variables From State Sequences and Input-Output Measurements

Master of Science Thesis

Knut Roar Lende

August 14, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Identifying Linear Parameter-Varying State Space Models by

Knut Roar Lende

in partial fulfillment of the requirements for the degree of

Master of Science Systems & Control

Dated: August 14, 2023

Supervisor(s):

Prof.dr.ir. Michel Verhaegen

Ir. Jacques Noom

Reader(s):

Dr. Raf van de Plas

Dr. Riccardo Ferrari

Abstract

Linear Parameter-Varying (LPV) systems can be used as a bridge to extend the well studied model based control methods of Linear Time-Invariant systems to certain nonlinear systems. Despite significant attention in literature over the last two decades, finding an efficient global state space identification algorithm remains an open problem. Furthermore, a common assumption has been that the scheduling signals governing the time-varying dynamics are known or measured exactly. These drawbacks are found to inhibit the number of use cases for model based LPV control. This thesis explores new ways of identifying LPV systems for more general nonlinear systems with limited information on the optimal scheduling variable and scheduling dimension.

To this end, two novel rank constrained least squares problems are presented to identify system matrices and scheduling signals from input and full state measurements, and a dictionary of possible scheduling signals. The use of the methods is demonstrated in a simulation experiment, and the results are compared to existing full state measurement methods.

At last, the full state measurement methods are coupled with a state sequence estimation method from literature in order to obtain a method for identification of quasi-LPV systems from Input-Output measurements. Here "quasi" indicates that the scheduling signal is dependent on the system state and input. The scheme is flexible, and a proof of concept is given on a small nonlinear system.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Before formulating the research question of this thesis, I was challenged with finding a problem I would like to solve, a way in which I could leave my mark on the scientific community in the field of systems and control. The freedom given by that question I see in the rest of the project, and I would like to thank my supervisors prof.dr.ir. Michel Verhaegen and ir. Jacques Noom for the opportunity. Not all master students have supervisors that communicate more enthusiasm about the project than the student sometimes feels himself. Their way of pointing out directions, suggesting papers with seemingly limited relevance, only for me to integrate them at the center of my research a month later has taught me a lot about combining information from different fields, and searching for underlying principles.

Further, I would like to thank my friends Liam, Sachin, Ali and Lucas. Thank you for being with me in early mornings and late evenings, through simple and difficult assignments throughout my Bachelors and Masters. I am grateful for everything I have learned from each and every one of you, and how we have found ways to play on each others strengths when collaborating.

I would also like to thank DSMG Krashna Musika and Stijn for five years of wonderful music, exciting challenges, and for continuously believing that I can do better.

Coming from abroad is difficult, as I have had a sea separate me from my immediate family. A special thanks therefore goes to the members of Redeemer International Church Delft, for the way I have been received and the home I have had among you.

At last, I would like to thank my family. Specifically I would like to thank my mother Tone, who fights for me and encourages me to go on when facing opposition, my sister Ane, who lights up my world with her wonderful smile, amazing hugs and great compassion for the people around her, my brother Sondre, who inspires me by his dreams and wild ideas, and my brother Magnus, whose passion for music motivates me to do my best. At last, I would like to thank my father Sigurd Olav, who has taught and demonstrated the importance and dignity of work to me.

Delft, University of Technology
August 14, 2023

Knut Roar Lende

"Look at the norm, but don't follow it"
— *Pranav Prasad*

Chapter 1

Introduction

The toddler learning how to grab stuff with his hands, the six year old learning how to ride a bike and the young adult learning how to cook or drive a car all have something in common. They use their bodies act on their surrounding environments, and their senses to observe the consequent change of affairs. It involves exploration and gathering of information, and once sufficient knowledge has been gathered, one can predict the consequences of current manipulations on the future state of affairs. In engineering, one prefers to describe future trajectories using mathematical models, and the building of these models from Input-Output (IO) data is referred to as system identification [4]. With the ability to predict future trajectories comes the possibility of evaluating the consequences of current actions, and thus enabling what is known as model-based control [5].

The closer the mathematical model is to reality, the better one can understand the consequences of current actions on future states. From a social perspective, utilizing the limited resources available on the earth in an optimal way is a core element of 9 out of 17 goals for sustainable development[1]. Engineering processes are especially involved in goal 12, Ensuring Sustainable Consumption and Production Patterns, a call to maximize the ratio between social benefits over physical resource use. Coupled with an increased desire for automation of industrial processes [6], the need for models with higher accuracy to facilitate more efficient control strategies is evident.

Linear Time-Invariant (LTI) systems are well studied and commonly used in industry. Unfortunately, most real world processes display some sort of nonlinear or time-varying behaviour, such as saturation, temperature dependence, etc. For several applications in mechanical and chemical engineering a linear approximation of these nonlinear processes is sufficient, see examples in [7]. However, the region of validity of the linear model might be exceeded with changes in the operating conditions, resulting in poor performance [8].

On the other hand, nonlinear system identification often relies on physical knowledge and other first-principles information. A notable exception is neural state-space models, [9], which are nonlinear black-box models capable of identifying nonlinear systems from IO measurements. Unfortunately, they can be difficult to train as the optimization problem is nonlinear and non-convex. Often large

[1] United Nations, The 17 Goals, Accessed online 04.08.2023 URL:https://sdgs.un.org/goals

amounts of data are required, or multiple runs with random initialization. Training times can be in the order of hours for even small networks as demonstrated by [9] and [10].

In all this, one should not forget that there exist sound, proven control methods for LTI systems. These include pole placement, Linear Quadratic Regulator (LQR), Model Predictive Control (MPC) and Robust controllers [7]. They are well studied, with corresponding proofs of stability, and often offer some intuition on how to choose the control parameters. The question then becomes whether it becomes possible to utilize these methods in systems that capture nonlinear dynamic behaviour. Linear Parameter-Varying (LPV) systems were developed to provide such an opportunity [11], but as of yet, finding a global, curse of dimensionality free state space identification algorithm remains an open problem.

This thesis presents novel ideas on how to estimate a quasi-LPV (qLPV) model from IO data. The report is started with an introduction to the existing methods for LPV state space identification in Chapter 2 along with a formulation of the research question. The theory and derivation of two full state measurement methods for identifying general LPV state space models developed as part of this thesis is then presented in Chapter 3. These full state measurement methods are put to the test by identifying a Continuously Stirred Tank Reactor (CSTR) and compared to other full state measurement methods in Chapter 4. At last, Chapter 5 gives a proof of concept on how one can estimate a state sequence from IO measurements, and use the estimate along with a full state method to identify an qLPV model of a nonlinear system. This approach would be a novel framework for qLPV state space identification. The conclusions follow in Chapter 6.

A last note in this introduction is regarding the mathematical notation utilised in this thesis. Due to a lot of different variable definitions, the superscript, ^{i}, is most often used to indicate the i-th element of a vector or block matrix. This should therefore not be interpreted as an exponent. Exceptions are the example in Chapter 4, and the variance, σ^2. To limit confusion, multiplications in other chapters are written out in their factors.

Chapter 2

Linear Parameter-Varying State-Space Identification

This chapter presents existing methods of Linear Parameter-Varying (LPV) system identification. The objective of LPV system identification is outlined in Section 2-1 before working through the different methods. First among them are the gain scheduling methods in Section 2-2, followed by full state measurement methods in Section 2-3. Approaches for estimating the state sequences are briefly presented in Section 2-4 before some novel neural network approaches are given in Section 2-5. At last, an overview of the open research areas and the research question of this thesis are given in Section 2-6

2-1 Modeling of Linear Parameter-Varying Systems

Identification of LPV state-space models is concerned with the building mathematical models describing the system dynamics by leveraging the relationship between the system matrices, the input, the output, the state and the scheduling variable. The different LPV identification methods assume different signals to be known, and identify the remaining signals as part of the identification.

A discrete time LPV system in state-space form evolves according to Equation 2-1, with the output governed by Equation 2-2. Here x_k, u_k, y_k and p_k are the state, input, output and scheduling variable at time instance k, respectively. The system matrices, A, B, C and D are functions of the scheduling variable p_k. For all system identification methods considered here, the functions relating the system matrices to the scheduling variable are unknown, and obtaining them is a key objective of the identification.

$$x_{k+1} = A(p_k)x_k + B(p_k)u_k \tag{2-1}$$
$$y_k = C(p_k)x_k + D(p_k)u_k \tag{2-2}$$

Often, the LPV model is assumed to have an affine structure. That means it can be written according to Equations 2-3 and 2-4, where $A(p_k)$ is a sum of constant matrices A_i multiplied by time-varying weights p_k.

$$x_{k+1} = \left(A_0 + \sum_{i=1}^{r} p_k^i A_i\right) x_k + \left(B_0 + \sum_{i=1}^{r} p_k^i B_i\right) u_k \tag{2-3}$$

$$y_k = \left(C_0 + \sum_{i=1}^{r} p_k^i C_i\right) x_k + \left(D_0 + \sum_{i=1}^{r} p_k^i D_i\right) u_k \tag{2-4}$$

The scheduling variable p_k governs the time-varying and nonlinear dynamics of the system. It can be a function of the input, output, time, or an external signal. In some cases, it is also taken as a function of the system state. The corresponding LPV model is then referred to as a quasi-LPV (qLPV) system [1]. At any time step k it is a vector with r entries, and r is frequently referred to as the scheduling dimension. Historically, the scheduling signal has been assumed known exactly.

A system state can be used to formulate continuous time dynamics as a set of first order differential equations, and determines the future evolution along with the future inputs [7]. In discrete time, the state evolution can be described by a set of difference equations [5].

The inputs and the outputs are measured signals relating the system to the surrounding environment. The inputs can be manipulated by the user, while the output is the response of the system based on it's state and received input. In engineering systems, inputs are supplied through actuators, while outputs are measured using sensors.

2-2 Gain Scheduling Methods

The concept of LPV systems was conceived based on gain scheduling controllers, where a controller was designed to operate in every operating point [11]. The gain scheduling identification methods assume the scheduling variable, the input and the output to be fully known, and identify the states and dynamics. It became common to identify linear systems in each operating point, that is, linearizing and identifying a model for a given condition, and design a suitable controller for this condition. When the system would be in between operating points, one would use interpolation to determine the active controller. This principle is still used when designing model based controllers for industrial applications, see among others [12], [13], [14] and [15].

Despite their initial popularity, a number of drawbacks have been identified for these methods. First of all, one would need to gather enough data in a given operating condition to successfully identify a model. This is not guaranteed, consider for example the out of plane flapping motion of a wind turbine that can be modeled as an LPV system with a periodic scheduling signal dependent on time [16]. Second, one runs the risk of obtaining different state definitions if one uses data-driven Linear Time-Invariant (LTI) methods for the different operating points. As a consequence, most of the systems are identified using first-principles in combination with a Prediction Error Method (PEM). These methods aims to minimize the difference between the simulated and measured trajectories of the system,, see for example [12]. A last drawback is that the number of models needed might be large for highly nonlinear and high dimensional systems [11], or in case the scheduling variable is inappropriately chosen.

2-3 Full State Measurement Methods

The full state measurement methods assumes the full state to be measured. The output equation thus becomes trivial, as $y_k = x_k + \epsilon_k$, with ϵ_k the measurement error. The input is known, and a common assumption is that the scheduling signal is known and noise free [2]. This section outlines the identification procedure of the two full state measurement functions encountered in literature.

The method of Verdult et al. formulates the identification as a least squares problem, minimizing the difference between a one step ahead predicted state, and the state measurement, and identifies an affine LPV state-space representation. The the authors knowledge, results have not been reported on applications. The method is worked out in Chapter 3, and used as a base for the two novel full state measurement methods.

Rizvi et al. [2] presents a nonparametric kernel method. The method uses Radial Basis Functions (RBFs) as the kernel. The weights of the respective RBFs are found by solving n linear systems of equations with N unknowns, where n again is the state dimension and N is the number of data points used for model training. The identified model has an affine representation with scheduling dimension r equal to the number of unique values in the scheduling function encountered during training. The method is further discussed in Chapter A and the performance is compared with the novel full state measurement methods in Chapter 4.

2-4 State Sequence Estimation Methods

As the state can seldom be assumed to be measured exactly, there has been developed methods for determining a state sequence along with the system dynamics. Common for most of these approaches is that they assume the input, output and scheduling variable to be known. Some also assume an affine LPV state-space representation.

Common for most attempts to reformulate the problem of finding an optimal state sequence as a linear problem is that they run into the curse of dimensionality. That is, the problem size grows exponentially with the past and future windows of inputs and measurements considered. The reason for this is closely associated with the computational complexity of LPV state propagation. Propagating an autonomous LTI system two time steps, e.g. from x_0 to x_2 can be done by multiplying by $A \cdot A$. If one is to propagate an autonomous LPV system with a scheduling dimension of one in a similar way, one would have to follow Equation 2-5, which include four unique matrix terms.

$$x_2 = (A_0 + A_1 p_1)\, x_1 = (A_0 A_0 + A_1 p_1 A_0 + A_0 A_1 p_0 + A_1 p_1 A_1 p_0)\, x_0 \tag{2-5}$$

One way of obtaining the state sequence is through extending known LTI subspace methods to LPV. Examples of this are found in [16], where the $PBSID$ algorithm from [17] is extended to the LPV case. The size of the least squares problem for this LPV identification algorithm scales according to Equation 2-6, where m and l are the number of inputs and outputs, r is the scheduling dimension, and s_p and s_f the size of the past and future windows of inputs and outputs, respectively. Other methods, assuming the input to be uncorrelated with the system and measurement noise are presented in [18]. When using these methods, care has to be taken that the state definition is the same across the different models. Van Wingerden [16] developed a kernel method to ensure this. Furthermore, the curse of dimensionality is shown to disappear if one can assume a periodic scheduling or piecewise

constant scheduling signal [16].

$$\rho_{PBSID} = (m+l) \sum_{j=1}^{s_p+s_f} r^j \tag{2-6}$$

Another approach is to identify an LPV Input-Output (IO) model, and find a realization of this model to get it in state-space form. Examples are presented in [19]. However, the realization of an affine IO model does in general not have a corresponding affine state-space model, introducing nontrivial dependencies on the scheduling variable. Furthermore, it is recommended to include a third step with the aim of reducing the state dimension resulting from the realization.

Research has also been done into kernel methods [11] and tensor trains [20]. Although some of these are able to avoid the exponential growth of the problem size with the number of past data points, their computational overhead is large compared to other methods for relatively simple systems.

2-5 Neural Network Approaches

The recent popularity of neural networks has inspired researchers to develop neural network based LPV state-space identification approaches. This is accomplished through structuring the network and cost function in a certain way. The individual sub-networks utilized in the structures are quite small, with up to 15 nodes divided over three layers. About six sub-networks are used per method. As the networks are optimized using gradient-descent based optimizers, they are considered curse of dimensionality free, yet the iterative manner of the optimizer could cause significantly longer computational times for simple systems.

Some of the approaches, notably [21],[22] and [23], assumes the scheduling variable to be known, along with the inputs and outputs, while others, such as LPV-SUBNET [1], allow the optimal scheduling variable to be co-optimized with the system matrices. The structure of the latter method is outlined in Figure 2-1, where there is a state encoder ξ_p mapping the past inputs and outputs to an estimated current state. This estimate is passed on along with the input to the network ζ, computing the optimal scheduling variable. A new state estimate is computed using the estimated system matrices, the estimated state and the estimated scheduling variable. This estimated state is again used to compute the next state estimate. The method has a recurrent structure, resulting in less efficient gradient computations.

A drawback of the neural-network methods is that the state dimension is a hyperparameter, and is not identified as part of the identification procedure. Consequently, it must either be determined through expert knowledge of the system or by multiple identification runs. The reported training time of LPV-SUBNET applied to a control moment gyroscope was about ten hours [1], yielding a significant identification time if multiple runs has to be conducted. A lower state dimension yields a simpler model, that may be expected to generalize better to unseen trajectories [24].

2-6 Research Question

As shown in this chapter, identification of LPV systems is an active research topic, and the pursuit of an identification procedure to be used in larger, complex systems is still ongoing. The characteristics and uses of the different methods are summarized in Table 2-I. This section analyses the open research problems, formulates the research question and outlines the main contributions of this thesis.

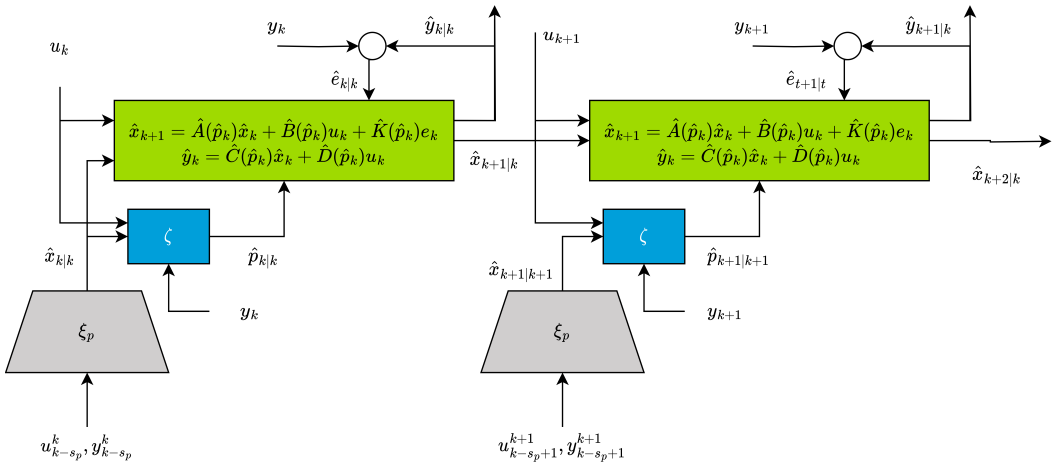

Figure 2-1: LPV-SUBNET, adapted from Figure 2 in [1] to the notation of this thesis

Table 2-I: The characteristics of the known LPV identification methods. ✓: method requires or is affected by the characteristic, *: an initial guess is required, but it is optimized within the given preset structure, †: iterative method, hence difficult to state computational cost.

	Gain Scheduling	Full State	State Sequence	Neural Network
Global Model		✓	✓	✓
State Sequence		✓		*
State Dimension		✓		✓
Scheduling Variable	✓	✓	✓	*
Scheduling Dimension	✓	✓	✓	✓
Curse of Dimensionality			✓	†

From Table 2-I, one can identify the following open research areas:

1. Most of the current LPV identification algorithms rely heavily on information about the scheduling variable, dictating the allowed nonlinear dependencies and the computational cost.
2. Of the encountered full state measurement methods, both depend on a predefined noise-free estimate of the scheduling signal.
3. All the methods depend on a predefined scheduling dimension.
4. The neural network approaches all depend on a predefined state dimension.
5. There are currently no methods allowing estimation of the state dimension and an optimal scheduling variable.
6. Obtaining a state sequence estimate remains computationally costly.

The observations above require the user of the algorithm to determine key aspects of the model before identifying it. In the extreme case of system identification, the starting point is a sequence of IO measurements from an unknown system. According to the author's view, the ideal algorithm should require as little information as possible and allow the user to make informed decisions based on the

available data. A short computational time would also be considered beneficial. If one were to include an ideal method in Table 2-I, it would have a check mark in the first row, thus identifying a global model, and be otherwise empty.

The research question investigated in this thesis was decided to be: How can one develop a curse of dimensionality free LPV state-space identification method allowing to estimate the scheduling parameter, state sequence and state dimension from IO measurements? The research question is considered from the perspective of a two-step scheme presented in Figure 2-2. First, a neural network based function is intended to estimate the state sequence from the IO data, before a full state measurement method with unknown scheduling would be used to estimate the scheduling variable and system dynamics.

The full state measurement method is discussed and developed in Chapter 3, where points 1, 2, and 3 are dealt with. To accomplish this, two novel rank constrained least squares problems are formulated, and two methods of relaxation are proposed. The performance of the formulations and relaxations are then evaluated on a simulation example and compared to existing full state LPV identification methods from literature.

At last, it is proposed to combine the novel full state measurement methods with a known approach for state sequence estimation. The approach is presented in Chapter 5. This combination would address points 4, 5 and 6 of the open research problems for qLPV systems, and could be a step towards an efficient identification scheme for identifying general LPV state space models from IO measurements.

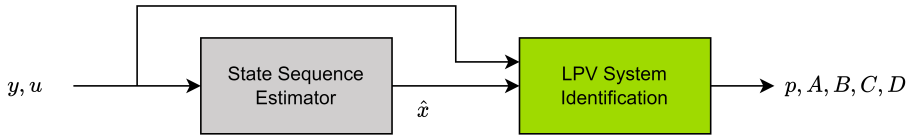

Figure 2-2: Two step approach for LPV state space identification

Chapter 3

Full State Measurement

Full state measurement approaches in Linear Parameter-Varying (LPV) system identification assumes that all the states relevant for the system dynamics are measured. The methods presented and developed in this section are applicable to general LPV systems as well as quasi-LPV (qLPV) systems, where the scheduling variable is allowed to depend on the system states and inputs. Although they are considered full state measurement methods, they remain applicable as long as an estimate of the state sequence is available. The methods will therefore also be presented with the output equation, even though that is trivial if the true system states are measured.

This chapter starts with introducing an example LPV system that will be used when deriving the formulations in the subsequent sections. The full state measurement method proposed by Verdult et al. [3] is then presented in Section 3-2. In Sections 3-3 and 3-4, two rank-constrained least squares problems are derived, in which the full state measurement method from Section 3-2 is extended to the case for unknown scheduling. The rank constrained problems are followed by two proposed ways of relaxing the rank constraints in order to obtain convex optimisation problems in Sections 3-5 and 3-6. At last, some notes are made on the appropriate model choice in Section 3-7.

3-1 Example

This chapter will be centered around the exaple LPV system whose state equation is given byEquation 3-1. It represents an autonomous, affine LPV system with a scheduling dimension r equal to 1. The output equation is given by Equation 3-2, and is trivial since the output is equal to the input. In other words, the full state is measured. Despite being simple, it contains the important complications of full state measurement affine LPV systems.

$$x_{k+1} = \left(\begin{bmatrix} 0 & 0.1 \\ -6.5 & -0.5 \end{bmatrix} + \begin{bmatrix} -0.1 & 0 \\ -0.1 & 0.5 \end{bmatrix} \cos k \right) x_k \tag{3-1}$$

$$y_k = x_k \tag{3-2}$$

An equivalent way to write the state equation using the Kronecker product $\otimes$ is given in Equation 3-3. This formulation is slightly more compact, and will be preferred in the remainder of the chapter.

$$x_{k+1} = \begin{bmatrix} 0 & 0.1 & -0.1 & 0 \\ -6.5 & -0.5 & -0.1 & 0.5 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \end{bmatrix} \otimes x_k \right) \tag{3-3}$$

The succeeding methods will be derived through this example, assuming that a full state measurement is available. After using the example, the general formulations are derived. This is done because the limited size of the example system makes it feasible to write out completely, making the derivations easier to follow.

3-2 Least Squares Formulation for Known Scheduling Parameter

Verdult et al. [3] introduces a least-squares based full state measurement method, where it is assumed that the states, the inputs and the scheduling variables are known. This section re-derives the method, as the least squares formulation forms a basis of the two novel formulations derived in the succeeding sections. At last it presents some requirements on the scheduling variable p, to ensure that the problem has a unique solution.

3-2-1 Example

Going back to the example, knowing the system states x_k, inputs u_k and scheduling variable p_k for $k = 0, \cdots, N+1$, the objective becomes to find the matrices A_0 and A_1 such that the state estimated by the model is as similar as possible to the measured state. To this end, one wants to minimize the squared difference between the left and right hand sides of Equation 3-1, defining the least squares problem given by Equation 3-4.

$$\min_{A_0, A_1} \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A_1 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \tag{3-4}$$

3-2-2 Generalization

For a more general affine LPV system with scheduling dimension r, one can rewrite Equations 2-3 and Equation 2-4 using the matrix Kronecker product to obtain Equations 3-5 and 3-6.

$$x_{k+1} = \begin{bmatrix} A_0 & A_1 & \cdots & A_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes x_k \right) + \begin{bmatrix} B_0 & B_1 & \cdots & B_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes u_k \right) \tag{3-5}$$

$$y_k = \begin{bmatrix} C_0 & C_1 & \cdots & C_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes x_k \right) + \begin{bmatrix} D_0 & D_1 & \cdots & D_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes u_k \right) \tag{3-6}$$

As both the state and scheduling signals are assumed to be known, one can formulate the following least squares minimization problem by subtracting the right hand sides from the left hand sides. Further stacking the equations for x_{k+1} and y_k allows one to write the least squares problem as Equation 3-7.

$$\min_{A_0,\cdots,D_r} \sum_{k=0}^{N} \left\| \begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} - \begin{bmatrix} A_0 & \cdots & A_r \\ C_0 & \cdots & C_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes x_k \right) - \begin{bmatrix} B_0 & \cdots & B_r \\ D_0 & \cdots & D_r \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes u_k \right) \right\|_2^2 \tag{3-7}$$

The total number of scalar unknowns is equal to the number of matrix entries. This number, ρ_{KS}, is given by Equation 3-8, where r is the scheduling dimension, n the state dimension, m the number of inputs and l the number of outputs.

$$\rho_{KS} = (r+1)(n+m)(n+l) \tag{3-8}$$

The optimisation problem defined by Equation 3-7 is equivalent to the full state measurement method proposed in [3]. Although it does not appear with an explicit vector of unknowns in its' current form, Equation 3-7 can be directly given to `cvxpy`[1] which will take care of the necessary book-keeping.

3-2-3 Condition Number

The condition number of an operator says something about the sensitivity of the output for a minor change in the input. For a matrix $\Phi \in \mathbb{R}^{\rho_a \times \rho_b}$, the condition number associated with the matrix 2-norm is defined according to Equation 3-9 [25].

$$\kappa_2(\Phi) = \frac{\sigma_{max}(\Phi)}{\sigma_{min}(\Phi)} \tag{3-9}$$

It is computed by dividing the maximum and minimum singular values, computed via the Singular Value Decomposition (SVD). An SVD is a matrix decomposition factorizing a matrix into the unitary matrices of left and right singular vectors U and V^T, respectively, and the matrix Σ with the singular values on the main diagonal [26]. Computing the SVD of Φ, one obtains Equation 3-10. As the minimum singular value is associated with the rank of the matrix, it says something about how difficult it is to compute the (pseudo-)inverse for a least squares problem.

$$\Phi = U\Sigma V^T \tag{3-10}$$

In the form of Equation 3-7, the objective function of the optimization problem is not explicitly available in the regular least-squares form of Equation 3-11. The latter states that a matrix of known entries, Φ, relates a vector of unknowns, γ, with a known vector ψ. Thus all the entries of the unknown matrices must be combined inside a vector γ. In this subsection, the matrix Φ will not be derived explicitly, but a matrix Ξ will be derived such that $\kappa_2(\Phi) = \kappa_2(\Xi)$.

$$\psi = \Phi\gamma \tag{3-11}$$

Again, looking at the example, the state equation is given by Equation 3-12 with the unknown coefficients a written out.

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} a_0^{1,1} & a_0^{1,2} & a_1^{1,1} & a_1^{1,2} \\ a_0^{2,1} & a_0^{2,2} & a_1^{2,1} & a_1^{2,2} \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k^1 \end{bmatrix} \otimes \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \right) \tag{3-12}$$

[1] `cvxpy` documentation, accessed online 07.02.2023, URL:https://www.cvxpy.org/index.html

One can then arrange the unknowns in a vector and the known entries in a matrix, according to Equation 3-13. This can be rewritten as Equation 3-14 through the use of the matrix Kronecker product. Here I_2 denotes the identity matrix with size 2×2. The least squares problem $\psi = \Phi\gamma$ can consequently be built by stacking the Equation 3-14 vertically for different values of k.

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} \left(\begin{bmatrix} 1 \\ p_k^1 \end{bmatrix} \otimes \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \right)^T & 0 \quad 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad 0 \quad 0 & \left(\begin{bmatrix} 1 \\ p_k^1 \end{bmatrix} \otimes \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \right)^T \end{bmatrix} \begin{bmatrix} a_0^{1,1} & a_0^{1,2} & a_1^{1,1} & a_1^{1,2} & a_0^{2,1} & a_0^{2,2} & a_1^{2,1} & a_1^{2,2} \end{bmatrix}^T \tag{3-13}$$

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} I_2 \otimes \left(\begin{bmatrix} 1 \\ p_k^1 \end{bmatrix} \otimes \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} \right)^T \end{bmatrix} \begin{bmatrix} a_0^{1,1} & a_0^{1,2} & a_1^{1,1} & a_1^{1,2} & a_0^{2,1} & a_0^{2,2} & a_1^{2,1} & a_1^{2,2} \end{bmatrix}^T \tag{3-14}$$

The condition number κ_2 is invariant under permutations P of Φ, since $PU\Sigma V^T$ is a valid SVD of $P\Phi$. This allows permuting Φ into a block matrix, as the unknowns associated with the first state never appear in the equation for the second, and vice versa. Furthermore, the kronecker product with I_2 does not change the condition number, as Equation 3-15 also is a valid SVD of $P\Phi$.

$$P\Phi = \begin{bmatrix} U_1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V_1^T & 0 \\ 0 & V_1^T \end{bmatrix} = \begin{bmatrix} \Xi & 0 \\ 0 & \Xi \end{bmatrix} \tag{3-15}$$

The condition number of Φ therefore equals the condition number of Ξ, where Ξ for a general least squares full state measurement LPV identification problem is given by Equation 3-16. Outputs and a higher state dimension will increase the number of blocks in Φ, while the Kronecker product between the scheduling variable and the inputs are added as additional columns of Ξ.

$$\Xi = \begin{bmatrix} p_0^T \otimes \begin{bmatrix} x_0^T & u_0^T \end{bmatrix} \\ p_1^T \otimes \begin{bmatrix} x_1^T & u_1^T \end{bmatrix} \\ \vdots \\ p_{N-1}^T \otimes \begin{bmatrix} x_{N-1}^T & u_{N-1}^T \end{bmatrix} \end{bmatrix} \tag{3-16}$$

From Equation 3-16, it is evident that the entries of the scheduling signals should be sufficiently different for the condition number to stay small, and the optimization problem defined by Equation 3-7 to remain well-posed.

3-3 Proposed Method: LPV-CONFUS

This section derives the LPV state space identification method using rank CONstrained least squares, assuming Full state measurement and Unknown Scheduling variable (LPV-CONFUS). It expands the least-squares method arrived at in the previous model by adding a rank constraint. This formulation is a novel contribution of this thesis, and the structure was inspired by the rank constraint presented in [27]. First, the least squares method is expanded for the example, before the general formulation is given. At last, a note is made concerning the solvability of the problem for higher scheduling dimensions.

3-3-1 Example

By replacing the known scheduling function $\cos k$ in Equation 3-4 with an unknown variable p_k, one obtains Equation 3-17. In this form, the full state measurement method with unknown scheduling is a nonlinear, nonconvex problem due to the multiplication of A_1 and p_k. The number of unknowns increases linearly with the number of datapoints, and without restrictions on p it becomes difficult to find an estimate for new trajectories.

$$\min_{A_0,A_1,p_0\cdots p_N} \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A_1 \end{bmatrix} \left(\begin{bmatrix} 1 \\ p_k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \tag{3-17}$$

An alternative could be to assume the scheduling variable p to be a linear combination of signals that can be found in a dictionary of signals, θ. The number of entries in θ is denoted by q. This is inspired by a similar approach some times used in blind system identification, for example [28]. For the sake of the example, one might choose a scheduling dictionary $\theta_k^T = \begin{bmatrix} \cos k & \cos 2k & \cos 3k \end{bmatrix}$. Introducing the scheduling dictionary into Equation 3-17 and rewriting to gather the unknowns, one obtains Equation 3-18.

$$\min_{\substack{A_0,A_1 \\ B_0,B_1 \\ \alpha_1^1,\alpha_1^2,\alpha_1^3}} \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A_1\alpha_1^1 & A_1\alpha_1^2 & A_1\alpha_1^3 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \\ \cos 2k \\ \cos 3k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \tag{3-18}$$

This optimization problem is still nonlinear and nonconvex, but has a lot fewer unknowns than Equation 3-17. However, by introducing the variables $A_1^1 = A_1\alpha_1^1$, $A_1^2 = A_1\alpha_1^2$ and $A_1^3 = A_1\alpha_1^3$, one can obtain an equivalent optimization problem with a linear least squares objective function and a rank constraint as given in Equation 3-19.

$$\begin{aligned} \min_{\substack{A_0,A_1,A_1^1,A_1^2,A_1^3 \\ \alpha_1^1,\alpha_1^2,\alpha_1^3}} & \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A_1^1 & A_1^2 & A_1^3 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \\ \cos 2k \\ \cos 3k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \\ \text{s.t.} \quad & \operatorname{rank} \begin{bmatrix} \operatorname{vec}(A_1) & \operatorname{vec}(A_1^1) & \operatorname{vec}(A_1^2) & \operatorname{vec}(A_1^3) \\ 1 & \alpha_1^1 & \alpha_1^2 & \alpha_1^3 \end{bmatrix} = 1 \end{aligned} \tag{3-19}$$

An advantage of this formulation is that the relative influence of the dictionary entries on the scheduling variable are directly available as optimization parameters. Placing a threshold on the α_1^i values when constructing the scheduling variable after the optimization, or adding a 1-norm constraint to drive the weight of less relevant dictionary entries to zero is straight forward.

3-3-2 Generalization

Removing knowledge of the scheduling signal from the general least squares formulation would result in the number of unknowns growing according to Equation 3-20, with N the number of data points. If the scheduling dimension r is larger than or equal to the sum of the number of states and the number of outputs per time step, i.e. $l + n \leq r$, the number of unknowns will always remain larger than

the number of equations leaving an ill-posed problem. Otherwise, the problem of estimating future trajectories would remain.

$$\rho_{IP} = \rho_{KS} + rN \tag{3-20}$$

Similar to the previous subsection, one introduces a dictionary of possible scheduling signals, θ. A general dictioary with entries θ^1 to θ^q would at timestep k be a vector defined according to Equation 3-21.

$$\theta_k^T = \begin{bmatrix}\theta_k^1 & \theta_k^2 & \cdots \theta_k^q\end{bmatrix} \tag{3-21}$$

Forcing each entry of the scheduling signal, p_k^i, to be a linear combination of the signals in the dictionary, one can express the i-th scheduling signal according to Equation 3-22. Here α_i is a row-vector.

$$p_k^i = \alpha_i \theta_k = \sum_{j=1}^{q} \alpha_i^j \theta_k^j \tag{3-22}$$

Inserting Equation 3-22 into Equation 3-7 would yield a nonlinear problem. As for the example, one can circumvent this by introducing new matrix variables, A_i^j such that $A_i \alpha_i^j = A_i^j$ for $j = 1, ..., q$ and $i = 1, ..., r$. This is displayed in Equation 3-23, and would be repeated for B, C and D. Again, q is the number of entries in the dictionary, while r is the scheduling dimension.

$$A_i p_k^i = A_i \alpha_i \theta_k = \sum_{j=1}^{q} A_i \alpha_i^j \theta_k^j = \sum_{j=1}^{q} A_i^j \theta_k^j \tag{3-23}$$

Although A_i^j are scalar multiples of each other if they have the same subscript i, the matrix variables with the same superscript, j, will be multiplying the same entry in the scheduling dictionary. It thus becomes convenient to define the matrices $\bar{A}, \bar{B}, \bar{C}$ and $\bar{D}$ according to Equations 3-24 to 3-27.

$$\bar{A} = \begin{bmatrix}A_0 & \sum_{i=1}^r A_i^1 & \cdots & \sum_{i=1}^r A_i^q\end{bmatrix} \tag{3-24}$$
$$\bar{B} = \begin{bmatrix}B_0 & \sum_{i=1}^r B_i^1 & \cdots & \sum_{i=1}^r B_i^q\end{bmatrix} \tag{3-25}$$
$$\bar{C} = \begin{bmatrix}C_0 & \sum_{i=1}^r C_i^1 & \cdots & \sum_{i=1}^r C_i^q\end{bmatrix} \tag{3-26}$$
$$\bar{D} = \begin{bmatrix}D_0 & \sum_{i=1}^r D_i^1 & \cdots & \sum_{i=1}^r D_i^q\end{bmatrix} \tag{3-27}$$

Equation 3-7 can then be extended to the rank-constrained least squares problem defined by Equation 3-28. Note that one obtains a rank constraint for every entry in the scheduling variable. In total one gets r rank constraints on matrices of the form given in Equation 3-29.

$$\min_{\bar{A},\bar{B},\bar{C},\bar{D}} \sum_{k=0}^{N} \left\| \begin{bmatrix}x_{k+1}\\ y_k\end{bmatrix} - \begin{bmatrix}\bar{A}\\ \bar{C}\end{bmatrix} \left(\begin{bmatrix}1\\ \theta_k\end{bmatrix} \otimes x_k \right) - \begin{bmatrix}\bar{B}\\ \bar{D}\end{bmatrix} \left(\begin{bmatrix}1\\ \theta_k\end{bmatrix} \otimes u_k \right) \right\|_2^2 \tag{3-28}$$
$$\text{s.t.} \quad \text{rank}(R_i) = 1, \qquad i = 1, \cdots, r$$

$$R_i = \begin{bmatrix} \text{vec}(A_i^1) & \text{vec}(A_i^2) & \cdots & \text{vec}(A_i^q) & \text{vec}(A_i) \\ \text{vec}(B_i^1) & \text{vec}(B_i^2) & \cdots & \text{vec}(B_i^q) & \text{vec}(B_i) \\ \text{vec}(C_i^1) & \text{vec}(C_i^2) & \cdots & \text{vec}(C_i^q) & \text{vec}(C_i) \\ \text{vec}(D_i^1) & \text{vec}(D_i^2) & \cdots & \text{vec}(D_i^q) & \text{vec}(D_i) \\ \alpha_i^1 & \alpha_i^2 & \cdots & \alpha_i^q & 1 \end{bmatrix} \tag{3-29}$$

The number of unknowns is now given by Equation 3-30, as one has to identify qr systems associated with the scheduling dictionary, r systems associated with the rank constraints, and the zeroth, time-invariant system. All of these have $(n+m)(n+m)$ unknowns. At last, one must identify qr of the α_i^j-weights. Although the number of unknowns does not grow with the number of datapoints N, it grows significantly for every increase in r or q.

$$\rho_{LPV-CONFUS} = (r(q+1)+1)(n+m)(n+l) + qr \tag{3-30}$$

3-3-3 Solvability For Higher Scheduling Dimensions

LPV-REFUS when used for a higher scheduling dimension could be regarded as having multiple equal dictionary entries. Instead of summing the matrices in $\bar{A}$, one could put them side by side and stack the dictionary θ r times, once for all of the scheduling dimensions. As a consequence, Ξ defined in Equation 3-16 will be rank-deficient and the minimum singular value of Ξ will be zero. This would mean an undefined condition number, which could be used as an argument in favour of an alternative formulation for higher scheduling dimensions. In this subsection, it will be showed how a numerical solver still can converge to an optimal solution, despite the ill-posed objective function. This subsection assumes the rank constraints to be relaxed using the nuclear norm $\|\cdot\|_*$.

The nuclear norm is minimized when the rank constrained variables are scalar multiples of each other, helping the problem to converge although the condition number is large. This can be seen by solving the convex optimization problem given in Equation 3-31. Here R^1 would solve the least squares part of the optimization problem without extending the dictionary to incorporate a higher scheduling dimension.

$$\begin{aligned} &\min_{R_1,R_2} \|R_1\|_* + \|R_2\|_* \\ &\text{s.t.} \qquad R_1 + R_2 = R^1 \end{aligned} \tag{3-31}$$

Through the triangle inequality for norms, one knows that Equation 3-32 holds [25].

$$\|R_1\| + \|R_2\| \geq \|R_1 + R_2\| = \left\|R^1\right\| \tag{3-32}$$

Furthermore, $\|R_1\|_*$ can be computed through Equation 3-33. Here $(R_1)^*$ denotes the conjugate transpose, which is equal to the transpose, $(R_1)^T$, as R_1 is assumed to be a real valued matrix.

$$\|R_1\|_* = \text{tr}(\sqrt{(R_1)^* R_1}) \tag{3-33}$$

Now, suppose one chooses a scalar $\beta \neq 0$ such that $\hat{R}_1 = \frac{1}{\beta} R^1$ and $\hat{R}_2 = \frac{\beta-1}{\beta} R^1$. Since the trace is a linear mapping and the square root of a product is equal to the product of the square roots, the following equalities hold:

$$\left\|\hat{R}_1\right\|_* + \left\|\hat{R}_2\right\|_* = \text{tr}\left(\sqrt{\frac{1}{\beta^2}(R^1)^* R^1}\right) + \text{tr}\left(\sqrt{\frac{(\beta-1)^2}{\beta^2}(R^1)^* R^1}\right) = \frac{1+(\beta-1)}{\beta}\text{tr}\left(\sqrt{(R^1)^* R^1}\right)$$

The proposed solution is thus a valid solution for the problem defined in Equation 3-31.

If one were to consider other possible solutions of the form $\hat{R}_1 = \frac{1}{\beta}R^1 + \Delta$ and $\hat{R}_2 = \frac{1-\beta}{\beta}R^1 - \Delta$, where Δ is anything but a scalar multiple of R^1, one would obtain the following inequality:

$$\left\|R^1\right\|_* \leq \left\|\hat{R}_1\right\|_* + \left\|\hat{R}_2\right\|_* = \left\|\frac{1}{\beta}R^1 + \Delta\right\|_* + \left\|\frac{\beta-1}{\beta}R^1 - \Delta\right\|_* \leq \left\|R^1\right\|_* + 2\left\|\Delta\right\|_*$$

The expressions would hold with equality only if Δ is a zero-matrix or a scalar multiple of R^1. Even though there are infinitely many solutions, the numerical methods later implemented seem to prefer the solution $\beta = r$, such that $R_1 = \cdots = R_r$. Finding a relaxation that allows one to move away from this solution is the topic of Section 3-6. On the other hand, this subsection shows that one can solve the least squares problem for the sum of all the systems multiplying the same entry in the scheduling dictionary.

3-4 Proposed Method: LPV-REFUS

As can be seen by Equation 3-30, the problem formulation for LPV-CONFUS scales with the product of r and q. This means the cost grows significantly when increasing the scheduling dimension and the size of the scheduling dictionary. Furthermore, if the scheduling dictionary is the same for all scheduling signals, only a sum of matrices multiplying the same dictionary entry appears in the least squares formulation. The discovery begs the question of whether it is possible to formulate a similar regularization with fewer parameters, growing according to Equation 3-34. To this end, the current section derives the LPV state space identification method using REduced rank constrained least squares, assuming Full state measurement with Unknown Scheduling variable (LPV-REFUS).

$$\rho_{LPV-REFUS} = (q+1)(n+m)(n+l) \tag{3-34}$$

3-4-1 Example

Equation 3-35 results from assuming a scheduling dimension of 2 instead of 1 and writing out the optimization problem of Equation 3-19 with the same three dictionary entries. It is evident one gets several variables multiplying the same signals.

$$\begin{aligned}\min_{\substack{A_0,A_1,A_2\\A_1^1,A_1^2,A_1^3\\A_2^1,A_2^2,A_2^3\\\alpha_1^1,\alpha_1^2,\alpha_1^3\\\alpha_2^1,\alpha_2^2,\alpha_2^3}} \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A_1^1 & A_1^2 & A_1^3 & A_2^1 & A_2^2 & A_2^3 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \\ \cos 2k \\ \cos 3k \\ \cos k \\ \cos 2k \\ \cos 3k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \\ \text{s.t.} \quad \operatorname{rank}(R_1) = 1 \\ \operatorname{rank}(R_2) = 1\end{aligned} \tag{3-35}$$

$$R_1 = \begin{bmatrix} \text{vec}(A_1) & \text{vec}(A_1^1) & \text{vec}(A_1^2) & \text{vec}(A_1^3) \\ 1 & \alpha_1^1 & \alpha_1^2 & \alpha_1^3 \end{bmatrix} \tag{3-36}$$

$$R_2 = \begin{bmatrix} \text{vec}(A_2) & \text{vec}(A_2^1) & \text{vec}(A_2^2) & \text{vec}(A_2^3) \\ 1 & \alpha_2^1 & \alpha_2^2 & \alpha_2^3 \end{bmatrix} \tag{3-37}$$

As the rank of a matrix is subadditive [26], the rank of the sum of two matrices is less than or equal to the sum of their individual ranks. Thus Equation 3-38 becomes Equation 3-39 for the example. Here all the entries are occurring as sums, and only A_i^j appear directly in the objective function. Recall A_i^j is defined as $A_i \alpha_i^j$. There exist no unique solution for α_i^j and A_i since for any scalar β different from zero, $\hat{\alpha}_i^j \beta$ and $\hat{A}_i \frac{1}{\beta}$ would also satisfy the constraint. Now that α_i^j and A_i no longer appear explicitly in the rank constraint, one might want to remove them from the optimization problem all together.

$$\text{rank}(R_1 + R_2) \leq \text{rank}(R_1) + \text{rank}(R_2) = 2 \tag{3-38}$$

$$\text{rank} \begin{bmatrix} \text{vec}(A_1) + \text{vec}(A_2) & \text{vec}(A_1^1) + \text{vec}(A_2^1) & \text{vec}(A_1^2) + \text{vec}(A_2^2) & \text{vec}(A_1^3) + \text{vec}(A_2^3) \\ 2 & \alpha_1^1 + \alpha_2^1 & \alpha_1^2 + \alpha_2^2 & \alpha_1^3 + \alpha_2^3 \end{bmatrix} \leq 2 \tag{3-39}$$

It is noted that if the original rank constraints on R_1 and R_2 are satisfied, any column of the matrix is a scalar multiple of either of the other columns, and any row is a scalar multiple of any of the other rows. This gives rise to the equivalence relation given in Equation 3-40. The less than or equal to one on the right hand side applies as it here is possible to obtain a zero matrix. This is unlikely to happen due to numerical solvers and noisy measurements, but is possible in case one has a noise free linear system or a dictionary that is independent from the nonlinear dynamics.

$$\text{rank} \begin{bmatrix} \text{vec}(A_1) & \text{vec}(A_1^1) & \text{vec}(A_1^2) & \text{vec}(A_1^3) \\ 1 & \alpha_1^1 & \alpha_1^2 & \alpha_1^3 \end{bmatrix} = 1 \Leftrightarrow \text{rank} \begin{bmatrix} \text{vec}(A_1^1) & \text{vec}(A_1^2) & \text{vec}(A_1^3) \end{bmatrix} \leq 1 \tag{3-40}$$

In a similar manner, one could remove the first column and last row form Equation 3-39 in order to obtain Equation 3-41.

$$\text{rank} \begin{bmatrix} \text{vec}(A_1^1) + \text{vec}(A_2^1) & \text{vec}(A_1^2) + \text{vec}(A_2^2) & \text{vec}(A_1^3) + \text{vec}(A_2^3) \end{bmatrix} \leq 2 \tag{3-41}$$

By using the rank constraint of Equation 3-41, neither the objective function nor the rank constraint distinguish between A_1^1 and A_2^1, A_1^2 and A_2^2 etc. They always appear in a sum, encouraging one to write the optimization problem in terms of the variables $A^1 = A_1^1 + A_2^1$, etc. This would result in Equation 3-42.

$$\begin{aligned} \min_{A_0, A^1, A^2, A^3} \sum_{k=0}^{N} \left\| x_{k+1} - \begin{bmatrix} A_0 & A^1 & A^2 & A^3 \end{bmatrix} \left(\begin{bmatrix} 1 \\ \cos k \\ \cos 2k \\ \cos 3k \end{bmatrix} \otimes x_k \right) \right\|_2^2 \\ \text{s.t.} \quad \text{rank}(R) \leq 2 \end{aligned} \tag{3-42}$$

$$R = \begin{bmatrix} \text{vec}(A^1) & \text{vec}(A^2) & \text{vec}(A^3) \end{bmatrix} \tag{3-43}$$

Reducing the size of the optimization problem is deemed beneficial when solving the optimization problem, however, the structure enabling the construction of an LPV system with a scheduling dimension of 2 has been lost and should to be recovered. This will be accomplished through computing the SVD of R. Recall Equation 3-10 and replace Φ by R. The number of non-zero singular values is equal to the rank of the matrix. Writing out R, U and Σ in Equation 3-44, it is evident that only the first two columns of U matters for the matrix multiplication $U\Sigma$ while only the first two rows of V^T matter for the matrix multiplication ΣV^T. All other entries are multiplied by zero. Here $U[:,1]$ denotes the first column of U, and $V^T[1.:]$ the first row of V^T.

$$\begin{bmatrix}\text{vec}(A^1) & \text{vec}(A^2) & \text{vec}(A^3)\end{bmatrix} = \begin{bmatrix}U[:,1] & U[:,2] & U[:,3] & \cdots\end{bmatrix} \begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & 0\\ \vdots & \vdots & \vdots\end{bmatrix} \begin{bmatrix}V^T[1.:]\\ V^T[2,:]\\ V^T[3,:]\end{bmatrix} \tag{3-44}$$

Carrying out the multiplication between Σ and V^T, and removing the columns and rows of U and ΣV^T that would be multiplied by or equal to zero, one obtains Equation 3-45.

$$\begin{bmatrix}\text{vec}(A^1) & \text{vec}(A^2) & \text{vec}(A^3)\end{bmatrix} = \begin{bmatrix}U[:,1] & U[:,2]\end{bmatrix} \begin{bmatrix}\sigma_1 V^T[1,:]\\ \sigma_2 V^T[2,:]\end{bmatrix} \tag{3-45}$$

The number of elements in a column of U equals the number of elements in a column of R, and the number of elements in a row of V^T equals the number of elements in a row of R. Furthermore, recall the relationship between A^1, A_1 and A_2 rewritten in Equation 3-46. It is evident that choosing $\text{vec}(A_1) = U[:1], \text{vec}(A_2) = U[:,2], \begin{bmatrix}\alpha_1^1 & \alpha_1^2 & \alpha_1^3\end{bmatrix} = \sigma_1 V[1,:]^T$ and $\begin{bmatrix}\alpha_2^1 & \alpha_2^2 & \alpha_2^3\end{bmatrix} = \sigma_2 V[2,:]^T$ satisfies Equation 3-46. Thus, the system with scheduling dimension 2 is recovered.

$$\begin{bmatrix}\text{vec}(A^1) & \text{vec}(A^2) & \text{vec}(A^3)\end{bmatrix} = \begin{bmatrix}\text{vec}(A_1) & \text{vec}(A_2)\end{bmatrix} \begin{bmatrix}\alpha_1^1 & \alpha_1^2 & \alpha_1^3\\ \alpha_2^1 & \alpha_2^2 & \alpha_2^3\end{bmatrix} \tag{3-46}$$

3-4-2 Generalization

The generalization of Equation 3-42 to a general LPV system with states, outputs, inputs and a general scheduling variable is straight forward, and given by Equation 3-47. In order to obtain Equation 3-47 from Equation 3-28, one considers the subadditive property of the matrix rank and removing the weights α_i^j and matrices $A_1 \cdots D_r$ from the rank constraint, as demonstrated in the previous subsection. The structure of the matrix defining the rank constraint is given in Equation 3-48.

$$\min_{A_0, A^1, \cdots, D^q} \sum_{k=0}^{N} \left\| \begin{bmatrix}x_{k+1}\\ y_k\end{bmatrix} - \begin{bmatrix}A_0 & \cdots & A^q\\ C_0 & \cdots & C^q\end{bmatrix} \left(\begin{bmatrix}1\\ \theta_k\end{bmatrix} \otimes x_k \right) - \begin{bmatrix}B_0 & \cdots & B^q\\ D_0 & \cdots & D^q\end{bmatrix} \left(\begin{bmatrix}1\\ \theta_k\end{bmatrix} \otimes u_k \right) \right\|_2^2 \tag{3-47}$$

$$\text{s.t.} \quad \text{rank}(R) \leq r$$

$$R = \begin{bmatrix}\text{vec}(A^1) & \text{vec}(A^2) & \cdots & \text{vec}(A^q)\\ \text{vec}(B^1) & \text{vec}(B^2) & \cdots & \text{vec}(B^q)\\ \text{vec}(C^1) & \text{vec}(C^2) & \cdots & \text{vec}(C^q)\\ \text{vec}(D^1) & \text{vec}(D^2) & \cdots & \text{vec}(D^q)\end{bmatrix} \tag{3-48}$$

Once a solution is found, the scheduling dimension could be reduced and optimal scheduling variables found using an SVD yielding a system of lower complexity. The approach similar to the one in the previous subsection. The new system with scheduling dimension r can again be read from the first r columns of U, given by Equation 3-49.

$$U[:, 1:r] = \begin{bmatrix} \text{vec}(A_1) & \text{vec}(A_2) & \cdots & \text{vec}(A_r) \\ \text{vec}(B_1) & \text{vec}(B_2) & \cdots & \text{vec}(B_r) \\ \text{vec}(C_1) & \text{vec}(C_2) & \cdots & \text{vec}(C_r) \\ \text{vec}(D_1) & \text{vec}(D_2) & \cdots & \text{vec}(D_r) \end{bmatrix} \tag{3-49}$$

The corresponding scheduling variables can be reconstructed from the dictionary according to Equation 3-50 using the row-vectors α_i containing the weights. The weights are again taken from the product of ΣV^T, given in Equation 3-51.

$$p_k^i = \alpha_i \theta_k \tag{3-50}$$

$$\Sigma V^T = \begin{bmatrix} \alpha_1^1 & \cdots & \alpha_1^q \\ \vdots & & \vdots \\ \alpha_r^1 & \cdots & \alpha_r^q \\ 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_r \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3-51}$$

3-5 Direct Relaxation

As rank constrained optimizations in general are NP-hard [27], the rank constraints present in LPV-CONFUS and LPV-REFUS should be relaxed to allow lower computational times. This could be accomplished by adding a term to the least squares objective function that will have a large magnitude when the rank constraint is violated, and close to zero when it is fulfilled. Furthermore, convex terms would be preferred.

One common approach is to add a scalar multiple of the nuclear norm of the rank constrained matrix to the objective function [27]. The nuclear norm is defined as the sum of the singular values. As the number of nonzero singular values equals the rank of the matrix, and the nuclear norm encourages driving small singular values towards zero, the relaxation could be justified. Furthermore, when coupled with the method for extracting a system with lower scheduling dimension through an SVD, one could use the magnitude of the singular values of the resulting R-matrix to determine the desired scheduling dimension.

Unfortunately, the relaxation will not work with the rank constraint of LPV-CONFUS, as the nuclear norm is minimized when setting the terms that don't appear explicitly in the leas squares part equal to zero. The affected terms are α_i and $\text{vec}(A_i)$, $\text{vec}(B_i)$, $\text{vec}(C_i)$ and $\text{vec}(D_i)$, making it difficult to reconstruct the LPV system with low scheduling dimension. In order to counter this, an iterative relaxation algorithm is defined in Section 3-6.

A second alternative for LPV-REFUS could be to use a composite matrix norm, applying a vector norm on each of the columns of the matrix R, and another on the resulting row-vector. A candidate for

this would be to apply the infinity norm as the inner norm, and the one norm as the outer. The infinity norm would only punish the largest value occurring in a system, while the one norm would drive the maximum values of systems towards zero. The unnecessary scheduling signals could be removed based on a threshold. By doing it this way, the results would be biased towards zero, encouraging a second run with only the relevant scheduling signals and no constraints. It would not provide a guarantee that the rank constraint is satisfied.

3-6 Iterative Relaxation

As discussed in Section 3-5, applying a direct relaxation of the rank constraints in LPV-CONFUS using a nuclear norm will result in difficulties when trying to extract the low scheduling dimension LPV system. The reason for this is that the nuclear norm of R_i with α_i, A_i, B_i, C_i and D_i equal to zero is less than for the rank one solution. The magnitude of the largest singular value increases if the parameters are different from zero. Thus one would have to find a way penalizing all but the largest singular value for the rank constraints in LPV-CONFUS, and all but the largest r singular values of LPV-REFUS.

In their paper, Yu et al. [27] utilize a similar rank constrained formulation to LPV-CONFUS. This they propose to relax by subtracting the Ky-Fan norm from the nuclear norm. The Ky-Fan r-norm is defined as the sum of the r largest singular values. Subtracting the Ky-Fan 1-norm from the nuclear norm of R_i in LPV-CONFUS would allow the maximum singular value to be as large as necessary to capture the dynamics, and would strongly favour the rank one solution.

An issue arising from subtracting the Ky-Fan norm is that the relaxed objective function is nonconvex. This latter part is solved by using an algorithm solving a sequence of convex problems proposed in [27]. Algorithm 1 presents a way to solve a convex objective function f for the parameters Θ subject to a rank constraint on the matrix function $H(\Theta)$ such that the latter has a rank less than or equal to r. U_1 represents the first r columns of U, and V_1^T the first r rows of V^T, where U and V^T are the matrices resulting from the singular value decomposition $H(\Theta) = U\Sigma V^T$. The Ky-Fan norm is linearized using Equation 3-52, and the current estimate Θ^{j+1} is optimized with the matrices resulting from an SVD of the previous optimum. As the weight ϕ goes towards infinity, the matrix is forced to have rank r [27].

$$\sum_{i=1}^{r} \sigma_i \left(H(\Theta)\right) = \operatorname{tr}\left[U_1^T H(\Theta) V_1\right] \tag{3-52}$$

The iterative relaxation procedure can be used with both LPV-CONFUS and LPV-REFUS. For the former, it is required, while the latter also can use a direct relaxation. Furthermore, the iterative relaxation scheme is observed to drive the solutions away from $R_1 = \cdots = R_r$ for LPV-CONFUS. This is assumed to be an attribute of the linearization of the Ky-Fan norm along with inaccuracies introduced by the numerical solver. Exactly how the division is determined is not known at the present time.

Algorithm 1 Convex-Concave Procedure [27]

$j \leftarrow 0$
$U_1^j \leftarrow 0$
$V_1^j \leftarrow 0$
$\hat{\Theta}^j \leftarrow 0$
$\phi \leftarrow \phi_0$
while $\left(f(\hat{\Theta}^{j-1}) - f(\hat{\Theta}^j) > \epsilon \textbf{ or } j = 0\right)$ **and** $\phi < \phi_{max}$ **do**
$\hat{\Theta}^{j+1} \leftarrow \arg\min_\Theta f(\Theta) + \phi\left(\|H(\Theta)\|_* - \operatorname{tr}\left[U_1^{j,T} H(\Theta) V_1^j\right]\right)$
$\begin{bmatrix} U_1^{j+1} & U_2^{j+1} \end{bmatrix} \begin{bmatrix} \Sigma_1^{j+1} & 0 \\ 0 & \Sigma_2^{j+1} \end{bmatrix} \begin{bmatrix} V_1^{j+1,T} \\ V_2^{j+1,T} \end{bmatrix} \leftarrow \text{SVD}(H(\hat{\Theta}^{j+1}))$
$\phi \leftarrow \mu\phi$ where $\mu > 1$
$j \leftarrow j + 1$
end while

3-7 Overview of Methods and Relaxations

The three different full state measurement methods and the two relaxations might not all be compatible with a specific problem setting. To aid in selecting a suitable path, Figure 3-1 is provided as a guideline. It could be seen as an expansion of the green block of Figure 2-2. If the scheduling variable is known, the unconstrained least squares formulation of Section 3-2 is the simplest to implement and fastest to execute. If one have a scheduling dimension equal to one, one might choose LPV-CONFUS, and otherwise LPV-REFUS is recommended. The last has direct relaxation as an alternative, while both LPV-CONFUS and LPV-REFUS are compatible with the iterative relaxation of Algorithm 1. LPV-REFUS further requires an additional SVD to reduce the scheduling dimension. Figure 3-1 should be considered as a roadmap, and the methods will be tested using the same data and scheduling dictionary in Chapter 4.

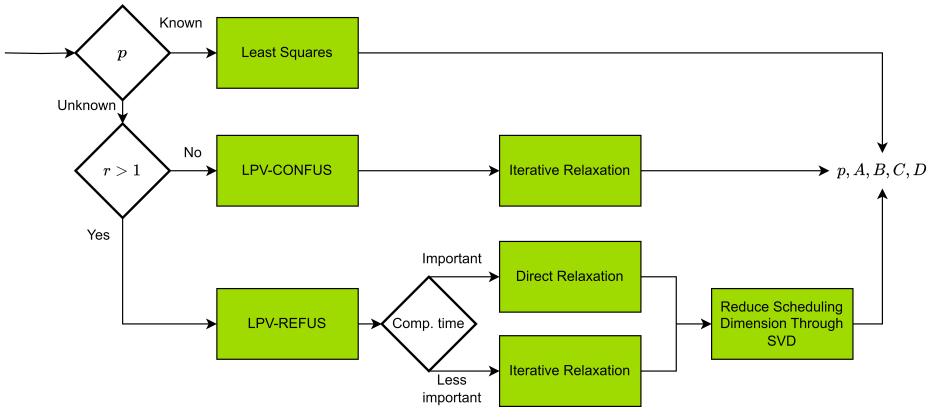

Figure 3-1: Overview of use cases for the different full state measurement methods and relaxations

Chapter 4

Full State Measurement Simulation Experiment

This chapter demonstrates the use of the novel full state measurement algorithms of Chapter 3, and presents their application to a simulated Continuously Stirred Tank Reactor (CSTR). A CSTR is an industrial application where it is reasonable to have access to the full state through measurements [2]. Further, it's dynamics are governed by nonlinear functions and an external signal that neither is regarded an input, nor a state. At last, it has been used in [2] for demonstrating the performance of their kernel based full state measurement method, and thus invites for comparison. First the analytical model with corresponding parameters is presented in Section 4-1, before the simulation settings are given in Section 4-2. Section 4-3 presents the dictionary of possible scheduling signals, while Sections 4-4 to 4-6 describes how the model performance is evaluated, presents the results, and compares them to known other full state measurement methods.

4-1 Problem Description

The CSTR system is a common reactor unit in chemical process engineering, and it facilitates a chemical reaction converting an inflowing liquid to a product [2]. The number of chemical reactions happening inside the tank depends on the concentration of reactants and the reactor temperature. In turn, the amount of reactions influence the reactant concentration and reactor temperature. A schematic is given in Figure 4-1, and the reactor has an inflow, Q_{in}, and an outflow, Q_{out}. Furthermore, the inflowing liquid has a concentration of reactants given by C_{in} and temperature T_{in}. T_c is the temperature of the cooling liquid, and h the liquid level inside the tank.

The analytical model used for simulation is provided in Equations 4-1 and 4-2 [8], while the coefficients and parameters for a steady-state solution are given in Table 4-I [2]. The temperature and concentration inside the tank, T_{rc} and C_{rc}, are the system states, while the cooling temperature, T_c, and inflow, Q_{in} are the inputs. Due to improper mixing, different manufacturers etc., the reactant concentration of the inflowing liquid, C_{in}, is assumed to vary over time and, allowed to deviate 50%

from the nominal value [8]. This significant change in operating condition motivated the autors of [29] and [8] to model the CSTR as an Linear Parameter-Varying (LPV) system.

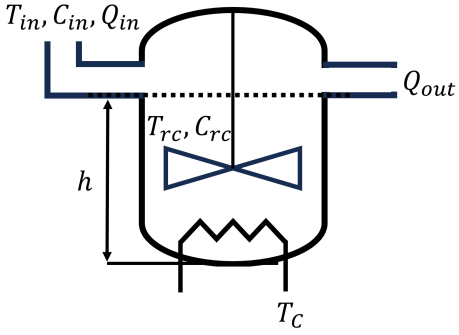

Figure 4-1: CSTR, adapted from Figure 1 in [2]

$$\dot{C}_{rc} = \frac{Q_{in}}{V_{rc}}(C_{in} - C_{rc}) - k_0 e^{\frac{-E_A}{R_{rc}T_{rc}}} C_{rc} \tag{4-1}$$

$$\dot{T}_{rc} = \frac{Q_{in}}{V_{rc}}(T_{in} - T_{rc}) - \frac{U_{HE}A_{HE}}{\rho_{rc}V_{rc}c_\rho}(T_{rc} - T_c) + \frac{\Delta H k_0}{\rho_{rc}c_\rho} e^{\frac{-E_A}{R_{rc}T_{rc}}} C_{rc} \tag{4-2}$$

Inspecting the analytical equations, one observes three causes of nonlinearities, and one time-varying signal. The term $e^{\frac{-E_A}{RT_{rc}}}C_{rc}$ appears in both equations, while Q_{in} multiplies C_{rc} and T_{rc} in Equation 4-1 and Equation 4-2, respectively. At last, Q_{in} multiplies the time-varying C_{in}. It is thus expected that functions of both states and C_{in} should be included in the scheduling dictionary.

The assumptions are taken from [8], and are:

- The inflow rate, Q_{in}, and outflow rate, Q_{out}, are kept equal, resulting in a constant liquid level h and liquid volume in the reactor V_{rc}.
- The reactor is mixed ideally, thus the concentration C_{rc} and temperature T_{rc} are uniform at any moment in time.
- The physical properties of the tank and reaction remain constant over time.
- The coolant temperature T_c is equal over the heat exchanger.
- The temperature of the inflowing liquid T_{in} remains constant.
- The heat generated by the stirring can be neglected.

4-2 Simulation Description

In order to generate training and validation data, two different simulation runs were set up. The inputs, outputs, initial states and inflow concentration variations were designed to replicate the simulations

Table 4-I: Steady state operation point of the Continuously Stirred Tank Reactor model [2]

V_{rc}	Reactor Volume	5 m^3
C_{in}	Concentration of the inflow liquid	800 kg/m^3
C_{rc}	Concentration of the reactor	213.69 kg/m^3
Q_{in}	Inflow rate	0.01 m^3/s
Q_{out}	Outflow rate	0.01 m^3/s
k_0	Pre-exponential term	25 s^{-1}
E_A	Activation energy of the reaction	30 kJ/kg
T_{in}	Temperature of inflowing liquid	353 K
T_{rc}	Temperature in the reactor	428.5 K
T_c	Coolant temperature	300 K
ρ_{rc}	Density	800 kg/m^3
c_ρ	Specific heat	1 kJ/(kg·s)
ΔH	Heat of reaction	125 kJ/kg
U_{HE}	Heat transfer coefficient	1 kJ/(kg·s)
A_{HE}	Surface area of the heat exchanger	1 m^2
h	Liquid level (height)	5 m
R_{rc}	Gas constant	8.31 J/(mol·K)

performed in [2] as closely as possible. Unfortunately, not all of the necessary information is available, and where possible, [8] was consulted. The time-varying trajectory C_{in} was attempted reconstructed from a Piece-Wise Affine (PWA) graph provided in [2]. The axes of the plot have low resolution, and the reconstructed graph does not match the provided graph exactly. The first six hours of the replicated time-varying trajectories for the training and validation runs are displayed in Figure 4-2.

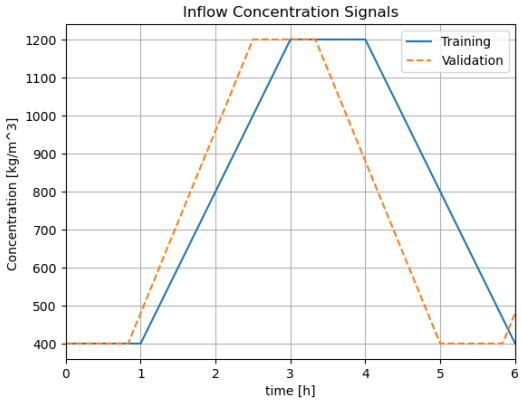

Figure 4-2: Inflow concentration trajectory C_{in} for training and validation, first six hours

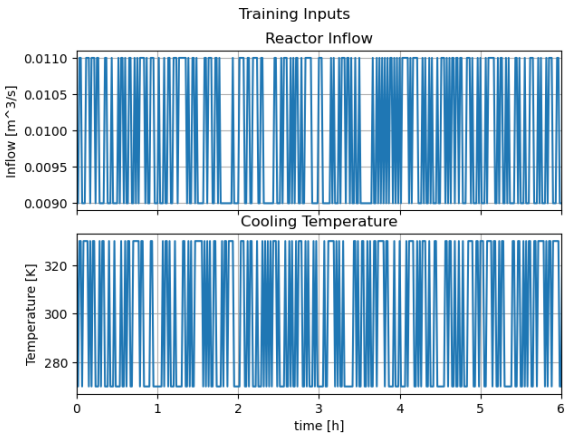

Figure 4-3: Training inputs, first six hours

The system was simulated in continuous time using the Runge-Kutta 5(4) method [30] as implemented by scipy [1]. The simulation interval was from zero to 24 hours. The input signals were Pseudo Random Binary Sequences (PRBSs) with 10% deviation from the steady state values of Q_{in} and T_C, consistent

[1]Scipy documentation, accessed 13.05.2023 online, URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html#r1793483225751

with [8].The first six hours of training inputs are plotted in Figure 4-3. The input value is allowed to change every 60 seconds, and the states are measured at the same time instance as the input updates.

Gaussian white noise signals with zero mean and variances $\sigma^2_{C_{rc}} = 1.5$ and $\sigma^2_{T_{rc}} = 0.5$ were added to the tank concentration and temperature measurements. Tóth et al. reports that the variances yield a Signal to Noise Ratio (SNR) of 30dB and 20dB for C_{rc} and T_{rc}, respectively [8]. In [8], the training data is obtained through multiple runs with constant values of C_{in}. The inflow concentration signal, Figure 4-2 reconstructed from Figure 5 of [2], results in larger variations in the temperature and concentration measurements, resulting in far higher SNRs. The first six hours of training data with added noise is plotted in Figure 4-4.

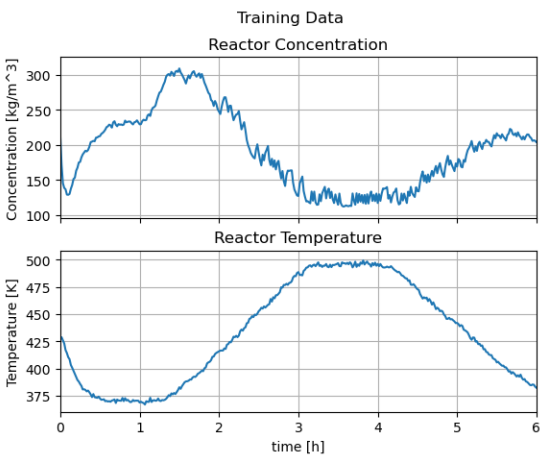

Figure 4-4: First six hours of the data used for training

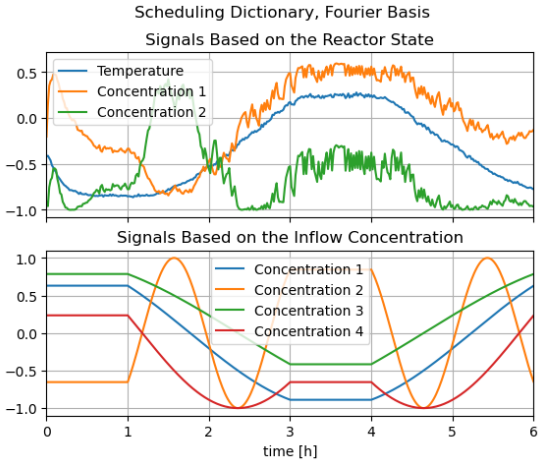

Figure 4-5: The seven signals of the scheduling dictionary, first six hours

4-3 Model Identification

LPV state-space models were identified using the least squares approach for known scheduling from Section 3-2, LPV-CONFUS from Section 3-3 and LPV-REFUS from Section 3-4. It was decided to use a Fourier basis for the scheduling dictionary. The frequency factors were chosen such that the entire dynamic range of the signal would be about $(0, \pi)$ in order to make them roughly one to one with the original functions. It was decided not to include signals depending on the inputs in the dictionary, as both input channels only take two values each. Furthermore, the analytical equations do not include nonlinearities introduced by only the inputs. It was decided to include functions of the states, which makes the identified models quasi-LPV (qLPV) models.

The functions composing the dictionary are given in Table 4-II, and plotted in Figure 4-5. In the remainder of this chapter, the scheduling signals keep their position in the dictionary, i.e. the first entry is dependent on T_{rc}, etc. The condition number of the resulting Ξ-matrix was found to be 40506, indicating a difference of 5 in the order of magnitude between the largest and smallest singular values.

On a last note, the input Q_{in} was multiplied by a factor 1000 before being passed to the identification algorithm. This was done in order to reduce computational time and avoid excessively large entries in the input matrices. This was found beneficial for the condition number of Ξ, as it used to be 40484474. The effective decrease was a factor of about one thousand.

Table 4-II: Scheduling dictionary entries, Fourier Basis

Dependency	Variable	Function
Temperature Reactor	T_{rc}	$\cos\left(\frac{T_{rc}}{100}\right)$
Concentration Reactor	C_{rc}	$\cos\left(\frac{C_{rc}}{120}\right)$
Concentration Reactor	C_{rc}	$\cos\left(\frac{C_{rc}}{60}\right)$
Concentration inflow	C_{in}	$\cos\left(\frac{C_{in}}{450}\right)$
Concentration inflow	C_{in}	$\cos\left(\frac{C_{in}}{100}\right)$
Concentration inflow	C_{in}	$\cos\left(\frac{C_{in}}{600}\right)$
Concentration inflow	C_{in}	$\cos\left(\frac{C_{in}}{300}\right)$

4-4 Performance Evaluation

Once a model was obtained using the algorithms of Chapter 3, the model performance was evaluated. This was done by simulating the model from the nominal, true, initial condition, providing the same input and C_{in} signal as when generating the validation data. The scheduling functions depending on the system state was evaluated at every time step using the obtained simulated state.

The obtained LPV models were evaluated using the metric Best Fit Rate (BFR) given by Equation 4-3. Here x_k denotes the true state, $\hat{x}_k$ the estimated state, and $\bar{x}$ the entry-wise mean of the true states. This metric was chosen, because it allows comparison with the results presented in [8] and [2]. The true states are noise free.

$$\text{BFR} = 100\% \cdot \max\left(1 - \frac{\sum_{k=1}^{N+1} \|x_k - \hat{x}_k\|_2}{\sum_{k=1}^{N+1} \|x_k - \bar{x}\|_2}, 0\right) \tag{4-3}$$

4-5 Results

This section presents the results and performance of the identified systems using the data found using the settings described in Section 4-2. In total, four configurations were tested from the possible combinations of problem formulations and relaxation methods described in Chapter 3. First, the unregularized least squares formulation of [3]. Then LPV-CONFUS presented in Section 3-3 used with Algorithm 1. LPV-REFUS was first used along with Algorithm 1, and then used in combination with a direct relaxation. The identification was performed on a laptop with a intel(r) core(tm) i7-10750h processor.

4-5-1 Least Squares Formulation

This subsection presents the results of using the unregularized least squares formulation presented in [3] and replicated in Section 3-2 using all dictionary entries as scheduling variables. A plot of the validation run is provided in Figure 4-6, and shows a quite accurate validation run. The largest inaccuracies seems to occur between 1 and 2 hours, and between 6 and 7 hours. In both of these regions, the peaks in concentration are underestimated.

The total computational time was 0.3 seconds, which is expected to be a lower limit for the methods considered in this section. In the end, the BFR ends at 94% and the scheduling dimension remains

unchanged at 7. Although exceeding the validation performance would be difficult, the objective of LPV-CONFUS and LPV-REFUS would be to decrease the scheduling dimension, yielding models with lower complexity.

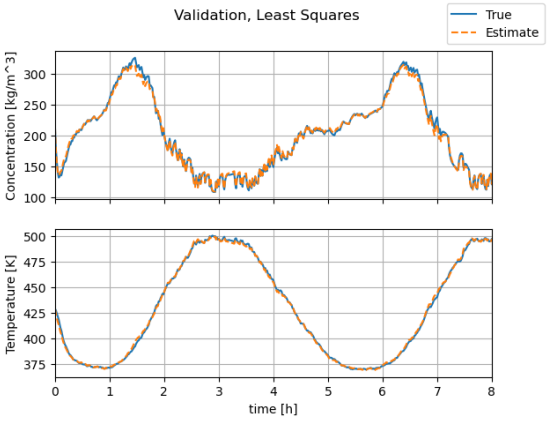

Figure 4-6: Validation run after solving the Least Squares problem from [3]

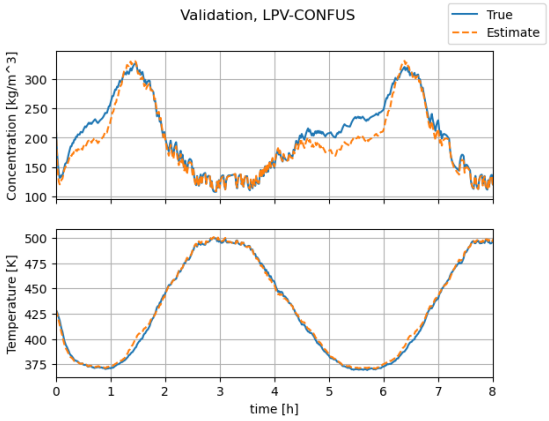

Figure 4-7: Validation run using LPV-CONFUS with scheduling dimension $r = 3$

4-5-2 LPV-CONFUS

This subsection presents the results using LPV-CONFUS presented in Section 3-3, and utilizing Algorithm 1 for optimization.

First, identification was attempted using a scheduling dimension $r = 3$. The algorithm was run using $\phi = 1, 1, 5, 10, 100, 500, 700, 900, 1100, 1300$, and the validation run can be seen in Figure 4-7. Here, the discrepancies appear to be largest during the phases of increasing concentration, visible in the intervals 0-1 and 5-6. In these phases, the smaller oscillations remain rather accurate. In the temperature plot, the changes appear to be predicted before they are measured. The BFR is 76%, and the computational time 355.3 seconds.

An additional attempt was made, using a scheduling dimension $r = 4$. Again ϕ was set to take the values $1, 1, 5, 10, 100, 500, 700, 900, 1100$ and 1300. The resulting validation run is plotted in Figure 4-8. The estimate is close in the rising phases, and only the peaks in concentration between 1 and 2, 4 and 5, and 6 and 7 are underestimated. The BFR is calculated to be 88%, and the computational time 128.5 seconds. As the results are more accurate and faster, these results are the ones that will be considered in the final comparison.

At last, a closer look is taken on the identified optimal scheduling signals. They are plotted in Figure 4-9, and the weights of the components in the original dictionary are given in Equation 4-4. The dictionary entry with the lowest significance appears to be the fifth. This is understandable, as it oscillates multiple times between the largest and smallest value of the inflow concentration, and thus poorly represents the physical signal it is based on. The largest individual weights are given to the sixth entry in the second scheduling signal. This is also visible in the plotted second entry of the scheduling variable, seen in Figure 4-9.

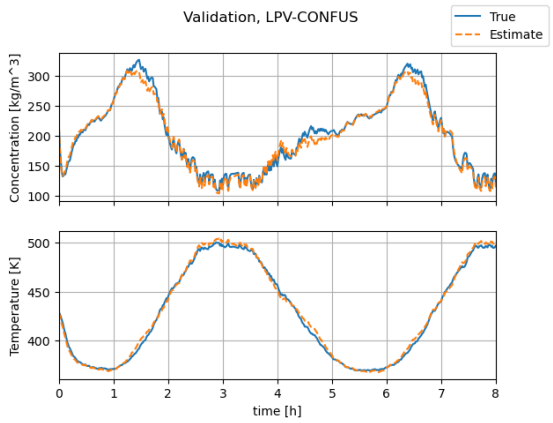

Figure 4-8: Validation run using the original formulation with scheduling dimension $r = 4$

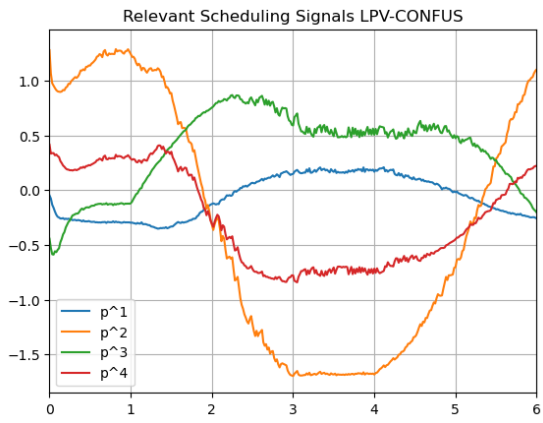

Figure 4-9: The relevant scheduling signals using LPV-CONFUS with scheduling dimension $r = 4$

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 0.576 & -0.106 & -0.097 & -0.214 & -0.010 & 0.258 & 0.037 \\ 0.330 & -0.599 & 0.219 & 0.374 & -0.033 & 1.390 & 0.628 \\ -0.704 & -0.248 & -0.111 & -0.110 & 0.015 & -0.898 & -0.447 \\ 0.452 & -0.211 & 0.293 & 0.0365 & -0.015 & 0.942 & 0.249 \end{bmatrix} \tag{4-4}$$

4-5-3 LPV-REFUS

This subsection displays the performance of the model obtained when using LPV-REFUS, presented in Section 3-4. The algorithm was run with $\phi = 1, 1, 5, 10, 100, 500, 1000, 5000, 8000, 10000, 10500, 11000, 11500, 12000$ and scheduling dimension $r = 3$. The validation run is displayed in Figure 4-10. From the plot, it is evident that the model closely resembles the original system, and even the peaks are well captured. The obtained BFR was 95%, achieved after a computational time of 71.1 seconds.

In order to gain insight in the determined scheduling variables, their time-evolution and weights are presented in Figure 4-11 and Equation 4-5. As for LPV-CONFUS, the fifth entry appears to be quite insignificant. The first entry, related to the reactor temperature, has a quite large influence in all three scheduling variable entries, while the fourth, sixth and seventh entries vary quite significantly. The moderate significance of the second and third entries could be explained by the fact that the only nonlinear term in Equations 4-1 and 4-2 where neither the temperature, nor the inflow concentration are involved, the reactor concentration multiplies the inflow. As the inflow was multiplied by a thousand prior to identification in order to improve the numerical stability of the algorithms, the term requires a lower weight.

$$\bar{\Sigma} V^T[0:3,:] = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 0.856 & -0.622 & 0.660 & 1.348 & -0.082 & 2.999 & 1.601 \\ -0.597 & -0.172 & -0.032 & -0.598 & 0.011 & 0.443 & -0.060 \\ -0.433 & -0.223 & 0.058 & 0.382 & -0.005 & -0.141 & 0.063 \end{bmatrix} \tag{4-5}$$

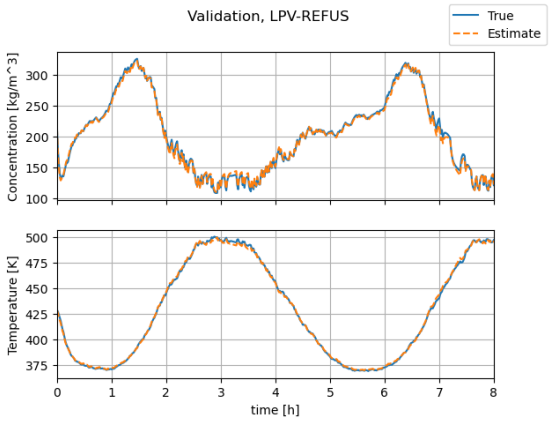

Figure 4-10: Validation run after identification using the LPV-REFUS formulation

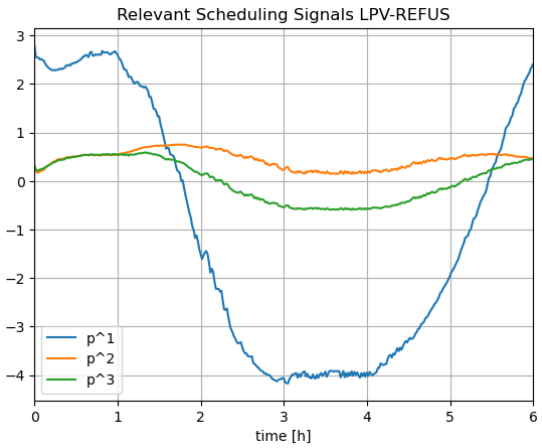

Figure 4-11: The relevant scheduling signals using LPV-REFUS

4-5-4 Direct Relaxation of LPV-REFUS

This section presents the result of using a non-iterative relaxation with LPV-REFUS. Specifically, a composite matrix norm is used on R, taking the one norm across the infinity norms of the individual columns. Consequently, one only penalizes the largest entry of a column of R, where a column of R corresponds to all the matrix entries of the dynamical system multiplying a signal in the scheduling dictionary. The relevant signals can then be filtered out based on the magnitude of the largest entry in it's corresponding system.

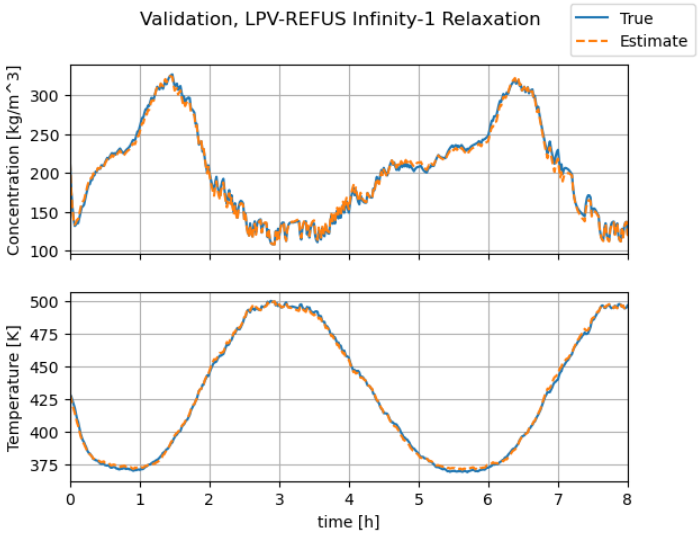

Figure 4-12: Validation run after identification using the composite $1-\infty$ norm presented in Section 3-5 and LPV-REFUS

The plot of the validation run is given in Figure 4-12. The peaks in concentration happening between

one and two appear accurately estimated, as are the spikes around seven. There is little to distinguish this result from the other methods. The BFR is computed to be 94%, with a computational time of 7.7 seconds and a scheduling dimension equal to 6. The weight on the relaxation was set equal to 30, and the threshold to 0.1. The dictionary not included corresponded to entry number 5, concentration inflow 2. This is consistent with the other results using LPV-CONFUS and LPV-REFUS.

4-6 Comparison

This section compares the results obtained in Section 4-5 with each other, and results from other papers using different methods to identify the same system.

The CSTR example has been studied for other LPV identification approaches, notably [2] and [8]. To enable a fair comparison, care has been taken to reconstruct the simulations from these papers as accurately as possible. As no analytical expressions for the scheduling and input signal were provided, they have been attempted reconstructed from the provided graphs. Some discrepancies still occur, for example in the PWA signal for the inflow concentration.

The method used for identification in [2] is briefly discussed in Appendix A. A different approach by Tóth et al. [8] identifies an LPV Input-Output (IO) model based on Orthonormal Basis Functions (OBFs). The IO model mapped the past eight inputs and scheduling variables to the next output. As such, the model is severely different from a state-space model, and the resulting scheduling dimension remains unknown. The computational time needed for training is also not reported.

Table 4-III: Performance of different full state measurement methods on the Continuously Stirred Tank Reactor. *The method was implemented and evaluated as part of this thesis. †The number is estimated as part of this thesis.

Method	[8]	[2]	Least Squares[3]*	LPV-CONFUS	LPV-REFUS	Direct
BFR	97%	85%	94%	88%	95%	94%
Runtime [s]	N/A	N/A	0.3	128.5	71.1	7.7
Resulting r	N/A	101†	7	4	3	6
Reduces r	×	×	×	✓	✓	✓

The identification results are given in Table 4-III, along with some key parameters. In terms of BFR, the IO model from [8] is found to perform the best of the considered methods. It is followed by LPV-REFUS, first using Algorithm 1 for relaxation, and then using a direct relaxation. The regular least squares formulation from [3] performs similar to the latter with a BFR of 94%. The last two spots are occupied by LPV-CONFUS, and the kernel method from [2]. The comparatively poor performance of the latter has been ascribed to the limited information available in the Radial Basis Functions (RBFs) with the reported kernel parameters. Furthermore, significant nonlinear contributions come from interactions between the states. More details are provided in Appendix A.

In terms of computational time, it is evident that the simpler formulations are quicker. The least squares formulation only takes 0.4 seconds to converge, and the direct relaxation of LPV-REFUS converges in 7.7 seconds. From there, the computational time increases drastically as the iterative relaxation is utilized. Using iterative relaxation, one is forced to solve multiple optimization problems. First, LPV-REFUS taking 71.1 seconds, before LPV-CONFUS needing 128.5 seconds. This last increase is ascribed to the increased number of variables, and more complex problem composition.

From the perspective of scheduling dimension, LPV-REFUS with iterative relaxation is able to reduce it from seven to three, including a small increase in validation performance. The iterative relaxations

here perform better than the direct and non-regularized methods, resulting in simpler models with lower scheduling dimensions. The main benefit of a lower scheduling dimension is that it reduces the effort of designing an LPV controller to regulate the system [15].

Chapter 5

State Sequence Estimation

The number of use cases where one knows a full state measurement is available is limited. Estimating the state sequence has been one of the main research areas in the past decades, and is at the core of [16], [19], [20] and others presented in Sections 2-4 and 2-5. However, circumventing the curse of dimensionality without obtaining an excessive overhead remains an open problem. Furthermore, the neural network approaches referred to in Section 2-5 all assume a known state dimension. This chapter presents a method from literature that might be coupled with the full state measurement approaches of Chapter 3 in order to obtain an quasi-Linear Parameter-Varying (LPV) (qLPV) state space model in a curse of dimensionality free way. The state sequence estimation approach was first presented in [31]. First a definition of the system state is given in Section 5-1. Then, the method for state estimation and state dimension determination from [31] is presented in Section 5-2 and Section 5-3, respectively. The proposed framework is then tested on the simulation example from [31] and a qLPV model is identified in Section 5-4. At last, the obtained results are briefly discussed.

5-1 State Definition

At the core of identifying a state sequence from Input-Output (IO) measurements is the assumption that one can reconstruct the relation between them. In continuous time, the definition of a state is the variables whose dynamics can be described exactly by first order differential equations [7]. Friedland [7] further stresses the importance of the future trajectory to be determined by the current system state and future inputs. Knowledge of the state would thus remove any need to remember past states, inputs and outputs in order to predict the future.

LPV-SUBNET [1] uses a neural network to estimate the current state from the past s inputs and outputs. This state encoder is represented by the gray trapezoid ψ_ξ in Figure 2-1. Verhoek et al. [1] argue the existence of such a map through the state reconstruction theory, and refers to [10], who in turn refers to [32].

Verdult et al. [31] restates the theory of embedding of nonautonomous dynamics. The key aspect is that the nonlinear map from the current state and next $2n$ inputs to the current and consecutive $2n$

measurements is an embedding under mild assumptions [33], allowing reconstruction of the dynamics from a finite number of input and output measurements. Here n represent the original state dimension, which in practice is unknown. The requirement of $2n+1$ delayed measurements is an upper bound [33]. In other words, one could use a subset of $2n+1$ consecutive measurements as the system state, and identify a corresponding LPV state space model. However, as general dynamical systems allow inputs, this would entail the current state being dependent on future inputs. The following section presents how this influence could be removed.

5-2 State Estimation

In the paper by Verdult et al. [31], an approach for identifying nonlinear state space models from IO data is proposed. The core idea of this chapter is to use the same approach up until the last step. Instead of identifying a general nonlinear state space, one would identify a qLPV system. An overview for the approach for state estimation is given by the gray boxes of Figure 5-1.

A central idea in [31] is the proposal to remove the influence of future inputs on the current state by using a nonlinear IO model, G_{IO}, in order to simulate $2n+1$ time steps into the future with zero input. This simulation is described by Equations 5-1 and 5-2. The current state $\hat{\mathbf{x}}_k$ is then taken equal to the current measurement and the simulated next $2n+1$ outputs as described in Equation 5-3. If the dynamics are slow relative to the sampling frequency, one could consider projecting further into the future, and scaling the index of the past inputs and outputs accordingly.

$$\hat{y}_{k+1} = G_{IO}(y_k, y_{k-1}, \cdots, y_{k-s_p}, u_k, u_{k-1} \cdots, u_{k-s_p}) \tag{5-1}$$

$$\hat{y}_{k+2} = G_{IO}(\hat{y}_{k+1}, y_k, \cdots, y_{k-s_p+1}, 0, u_k, \cdots, u_{k-s_p+1}) \tag{5-2}$$

$$\hat{\mathbf{x}}_k = \begin{bmatrix} y_k & \hat{y}_{k+1} & \cdots & \hat{y}_{k+2n} \end{bmatrix} \tag{5-3}$$

An advantage of training a nonlinear IO compared to the recurrent neural network structures, is that the IO model has a simpler overall structure. This leads to simpler backpropagation, and quicker training times.

5-3 Determining the State Dimension

Once again, as $2n+1$ is an upper limit on the number of delayed measurements, and thus it might be possible to accomplish a lower state dimension. Verdult et al. [31] suggests doing this based on the method of false nearest neighbours [24]. Two neighbours are considered false if the square root of the relative euclidean distance between two vectors $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_1$ increases more than a preset threshold, δ, when including an additional time-delayed measurement. This criterion is restated in Equation 5-4, where $\hat{\mathbf{x}}_1[\tau]$ denotes the τ-th element of $\hat{\mathbf{x}}_1$, and $\hat{\mathbf{x}}_1[1:\tau-1]$ means the sub-vector consisting of the first $\tau-1$ entries of $\hat{\mathbf{x}}_1$.

$$\frac{|\hat{\mathbf{x}}_1[\tau] - \hat{\mathbf{x}}_2[\tau]|}{\|\hat{\mathbf{x}}_1[1:\tau-1] - \hat{\mathbf{x}}_2[1:\tau-1]\|_2} > \delta \tag{5-4}$$

The estimated state dimension $\hat{n}$ is determined by increasing τ in increments of 1 starting from 1, and when the total number of false nearest neighbours is close to zero, $\hat{n} = \tau$. Every combination of $\hat{\mathbf{x}}_i$,

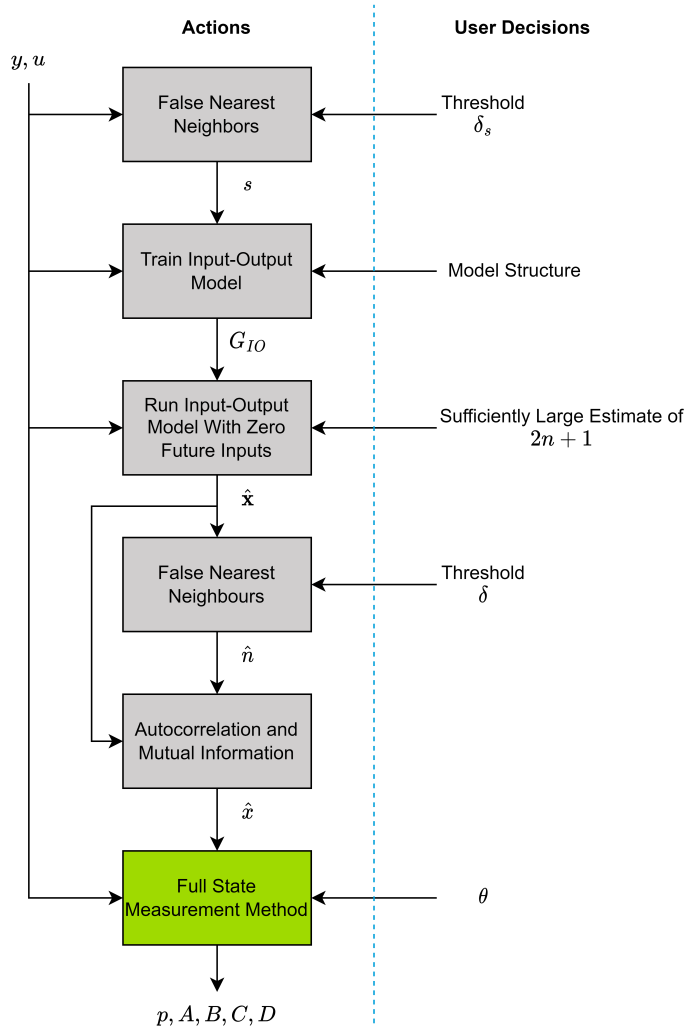

Figure 5-1: Proposed scheme for LPV system identification from IO measurements

$\hat{\mathbf{x}}_j$, such that $i, j = 1, \cdots N$ and $i \neq j$ is evaluated for every τ. The method is also used to determine the number of past inputs and outputs to use in the nonlinear IO model. From Figure 5-1, one can see that the only unaided user decision associated with determining the state dimension is the threshold δ. Once δ is decided, $\hat{n}$ can be read from a graph.

The embedding delay, η, determines the entries from $\hat{\mathbf{x}}_k$ to be included in the final estimated state $\hat{x}_k$. It is proposed determined via the autocorrelation function of $\hat{\mathbf{x}}_k$ and the first minimum of the mutual information criterion [34]. The final state estimate is thus given by Equation 5-5.

$$\hat{x}_k = \begin{bmatrix} y_k & \hat{y}_{k+\eta} & \hat{y}_{k+2\eta} & \cdots & \hat{y}_{k+(\hat{n}-1)\eta} \end{bmatrix} \tag{5-5}$$

5-4 Simulation Example

This section discusses an attempt to recreate the results presented in [31]. Instead of a nonlinear state space model, a qLPV model is identified using the estimated state sequence. The parameters $s = 2$, $\hat{n} = 2$, $\eta = 5$ and the structure of the nonlinear IO model was taken equal to the ones presented in [31].

5-4-1 Simulation Setup

The identification example presented in [31] is centered around the system in Equation 5-6. A neural network with one hidden layer of five neurons was trained to map two past IO measurements to the consecutive output measurement. The embedding dimension of two was again determined through the Lipschitz numbers and false nearest neighbours. The activation functions of the network was not specified, and 600 datapoints were considered available for training. They were collected at a sampling frequency of 20Hz. The input is zero-order hold white noise with a uniform distribution between -0.5 and 0.5. To stay consistent with the rest of the thesis, (x^i) indicates the i-th state. The superscripts appearing outside brackets are actual exponents.

$$\dot{x}^1 = x^2 - 0.1\cos(x^1)(5x^1 - 4(x^1)^3 + (x^1)^5) - 0.5\cos(x^1)u \tag{5-6}$$

$$\dot{x}^2 = -65x^1 + 50(x^1)^3 - 15(x^1)^5 - x^2 - 100u$$

$$y = x^1 \tag{5-7}$$

5-4-2 Nonlinear Input-Output Model

The nonlinear IO model was attempted replicated using a neural network with one hidden layer of five neurons, mapping the two past IO measurements to the next output. The `tanh` function was used for activation, and the network was implemented using the sequential model from Tensorflow[1]. It was difficult to find a nonlinear IO model with acceptable performance. Despite assuming a noise free output, consistent with [31], no success was had training on only 600 datapoints. The number of datapoints were thus increased to 1200. A random initialization and 5000 epochs were utilized. The training time of the network was about two and a half minutes.

$$VAF(y, \hat{y}) = \max\left(0, \left(1 - \frac{\sum_{k=1}^{N} \|y_k - \hat{y}_k\|_2^2}{\sum_{k=1}^{N} \|y_k\|_2^2}\right) \cdot 100\%\right) \tag{5-8}$$

The IO model's performance on the training data is displayed in Figures 5-2 and 5-3. It is clearly well-fit to the training data. However, running the IO model in open loop, using only the first two measurement and inputs of a new dataset, one obtains a Variance Accounted For (VAF) of 60%. The VAF is defined in Equation 5-8 [5]. The simulation run is displayed in Figures 5-4 and 5-5. The poor validation performance indicates that the IO model is overfit. This could be avoided by decreasing the model complexity, for example reducing to four neurons, by increasing the amount of data available, as already done, or by employing some regularization, for example 1-norm constraints on the neuron weights. Based on plots in [31], the nonlinear IO model obtained by Verdult et al. achieved better

[1] Tensorflow documentation. Accessed online 10.05.2023 URL: https://www.tensorflow.org/guide/keras/sequential_model

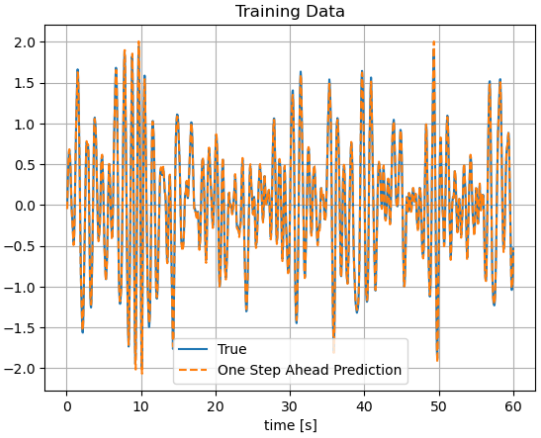

Figure 5-2: Using the IO-model to predict the next output of the training data

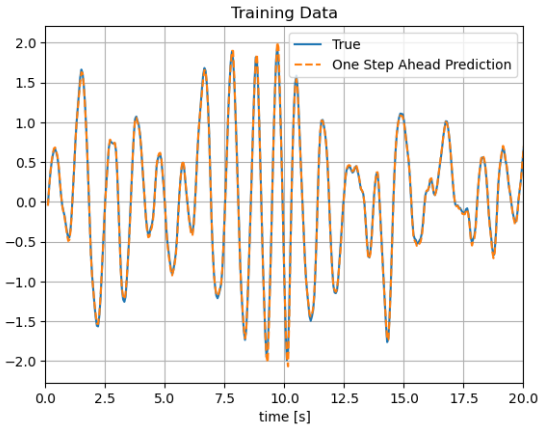

Figure 5-3: Closeup of Figure 5-2 showing the first 20 seconds

performance. No quantitative measure of performance was provided for validating the IO model. Despite the comparatively poor performance, the obtained IO model was nonetheless used to estimate a state sequence with two states. The true measurement, and the fifth forward prediction under zero input were utilized, consistent with [31].

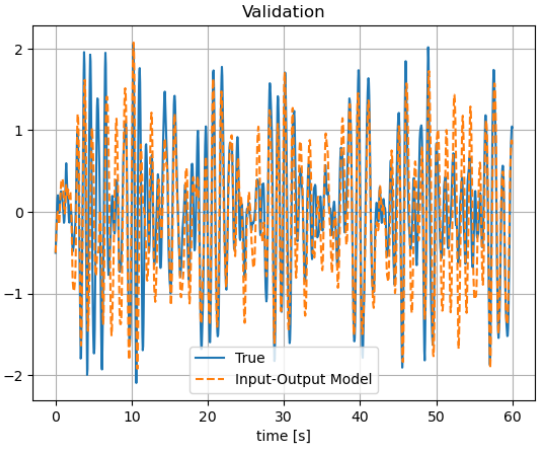

Figure 5-4: Validating the IO model on a new dataset

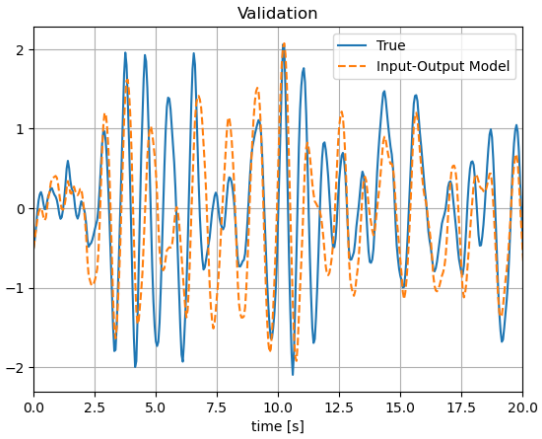

Figure 5-5: Closeup of Figure 5-4 showing the first 20 seconds

5-4-3 LPV Identification From Estimated States

The estimated state sequence was then used to estimate an LPV model. A dictionary with various polynomials of x^1 and x^2, and products between $\cos(x^1)$ and the polynomials of x^1 were used. The polynomials were normalized through dividing by the maximal value encountered to the order of the polynomial, i.e. $(\hat{x}^1)^3/(\max_k \hat{x}_k^1)^3$. In total, the dictionary had 20 entries. A non-iterative, nuclear norm was used for relaxation of the higher scheduling dimension formulation.

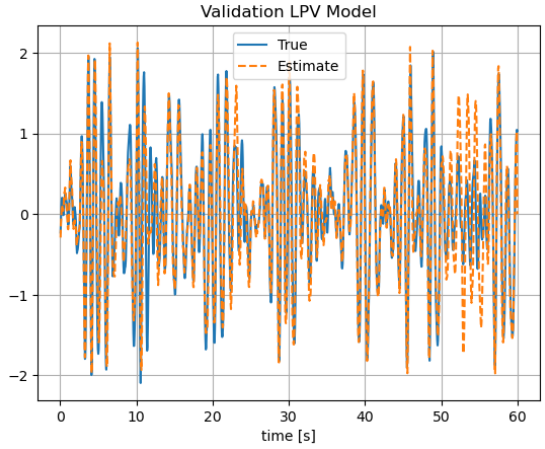

Figure 5-6: Validation run of the identified LPV model

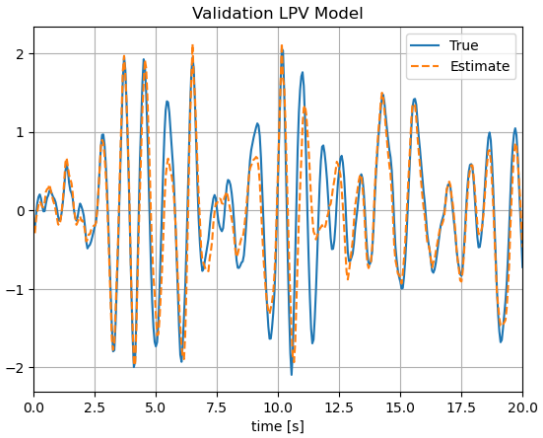

Figure 5-7: Closeup of Figure 5-6 showing the first 20 seconds

The identified LPV model was run on the validation data set, and the results are displayed in Figures 5-6 and 5-7. Here, the VAF ended on 73%, and the Best Fit Rate (BFR) was 52%. The higher fit compared to the nonlinear IO model is ascribed to the fact that the LPV algorithm has access to one true state, and only one state estimate is generated using the IO model. All the nonlinearities of the true model are associated with this one known state, which reduces the importance of the second state. Furthermore, comparatively few consecutive updates had to be performed by the nonlinear IO model in order to obtain the estimate of the second state.

5-5 Discussion

Although the simulation results presented in the previous section are relatively poor, the possible flexibility of the LPV identification algorithm is significant. It allows estimating the state dimension, state sequence, scheduling variable, and an affine LPV system in a curse of dimensionality free way, albeit with a significant computational overhead. Furthermore, the algorithm can be executed in smaller parts with feedback available in-between, allowing adjustments to be made underway.

When coupling the full state measurement methods to a state sequence estimate based entirely on IO data, new complexities arise with regards to creating the scheduling dictionary. If one have access to measurements, a suggestion would be to base it on those, but the parameter variation might not be captured in an explicit measurement for some Linear Time-Varying (LTV) systems. The ability of the nonlinear IO model to capture such kinds of time-varying dynamics should be thoroughly investigated.

Chapter 6

Conclusion and Outlook

This chapter presents the conclusions that can be drawn from the thesis, and give an overview of recommended future research areas.

6-1 Conclusion

This thesis has, presented the derivation of two novel full state measurement Linear Parameter-Varying (LPV) identification algorithms that estimate scheduling variables and system matrices from Input-Output (IO) data and a dictionary of possible scheduling signals. The reduced scheduling dimension makes it feasible to model nonlinear system as LPV models.

The full state measurement methods were demonstrated on a Continuously Stirred Tank Reactor (CSTR), and were in some formulations found to outperform current LPV state-space identification algorithms in terms of Best Fit Rate (BFR). This comes at the cost of an increased runtime. Furthermore, the proposed methods were capable of reducing the scheduling dimension, reducing the effort of designing controllers.

At last, a curse of dimensionality free way of estimating a state sequence for single output nonlinear systems was revisited and used to identify a quasi-LPV (qLPV) model using a full state measurement method. Although the identified qLPV model displayed low validation performance, the results work as a proof of concept for a curse of dimensionality free qLPV identification algorithm capable of determining both state dimension, scheduling variable and dynamics from IO measurements. Furhtermore, the simple structure of the nonlinear IO model allows lower training times.

6-2 Outlook

In the future, it would be interesting to see how the proposed full state measurement methods LPV-CONFUS and LPV-REFUS perform on noisy, real-world data. It is expected that introducing more

noisy measurements in the scheduling dictionary might cause longer computational times. Furthermore, one might find it more difficult to decrease the scheduling dimension. As an extension, one should also test the state sequence estimation algorithm in a similar setting.

It is also recommended to test the state sequence estimation scheme on a different set of systems. These would have larger state dimension and nonlinearities associated with more than one state.

Furthermore, one might consider extending the state sequence estimation algorithm to include more general LPV systems. This would include investigating the time variations that could be captured without passing them to the nonlinear IO model. One might also consider passing a dictionary to the nonlinear IO model.

At last, one might consider extending the method to blind LPV identification through adding an additional rank constraint in the state space identification step. This would also require determining convergence criteria on the combined rank constraints.

Appendix A

Continuously Stirred Tank Reactor Identification Using Radial Basis Functions

This appendix discusses the method developed in [2], and presents the results from using a full state measurement method with the basis functions used in the same paper. The experiment is the same as the one used in Chapter 4, and the details will not be restated here.

Instead of using the Fourier basis, attempts were made at using the Radial Basis Functions (RBFs) introduced in [2] together with the LPV-REFUS formulation. In [2] the authors argues the use of C_{in} based RBFs based on the change of the step response for different constant values of C_{in}. The equivalent affine representation presented in Equation A-1, where x_i is the state at the i-th time step, ω_i are unknown weights, in a row-vector, such that $\omega_i x_i$ defines a matrix. The weights are obtained during training of the model.

$$A(p_k) = \sum_{i=1}^{N} \omega_i x_i^T e^{\frac{(p_k - p_i)^2}{2\sigma^2}} \tag{A-1}$$

The term $e^{\frac{(p_k - p_i)^2}{2\sigma^2}}$ equals one when $p_k = p_i$, and approach zero as the difference between them grows. After a fine grid search, the authors of [2] decide to use $\sigma^2 = 620$, yielding rather wide RBFs. Furthermore, the affine representation has a scheduling dimension equal to the number of unique values of p_i encountered in the training data. For the reconstructed training signal in Figure 4-2, a scheduling dimension of 101 would be necessary.

The time variation of 17 of the 101 scheduling signals are plotted in Figure A-1, and one can observe that they are fairly similar. Computing the condition number of the least-squares part of the optimization problem, as described in Section 3-2, one finds it in the order of 10^{18}. Thus the RBFs based on C_{in} with the kernel parameter of 620^2 is not found informative enough to identify a meaningful

Linear Parameter-Varying (LPV) model. It was found that the algorithm devised for higher scheduling dimension struggles to converge with a dictionary of more than 17 entries. The corresponding results can be seen in Figure A-3.

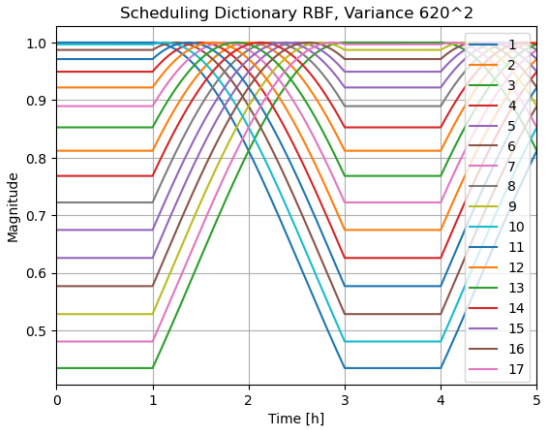

Figure A-1: RBF dictionary with a kernel parameter of $\sigma^2 = 620^2$

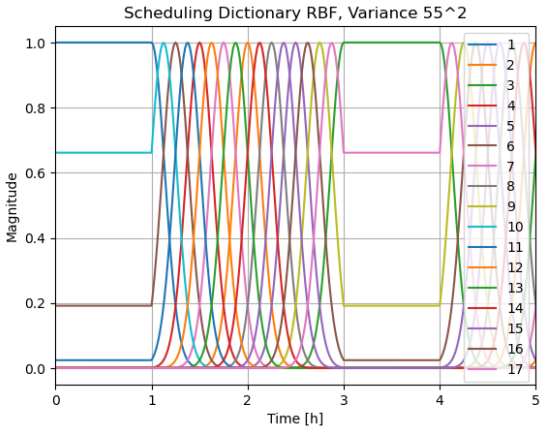

Figure A-2: RBF dictionary with a kernel parameter $\sigma^2 = 55^2$

Parallels were drawn between the gain-scheduling techniques mentioned in Section 2-2 and the nature of the RBFs. It was thus decided to run a second attempt with RBFs and a much lower kernel parameter, ensuring only a couple of the functions are active at the same time. The time evolution of the dictionary can be seen in Figure A-2 for the same 17 RBF centers as in Figure A-1. and the result is displayed in Figure A-4. Clearly the concentration increase between 4 and 6 hours is not well captured, neither is the temperature. It is visible from Table A-I that the Best Fit Rate (BFR) goes down from 77% to 71%. This could be ascribed to the combination of limited data points available in the middle range of C_{in} and a comparatively large influence of the neighbouring functions.

Table A-I: Attempts using a radial basis function basis, similar to [2]

Number of functions	kernel parameter	BFR	Computation Time [s]	Condition Number
3	620	79%	0.3	$3.3 \cdot 10^4$
9	620	78%	16.2	$2.7 \cdot 10^{15}$
17	620	79%	20.3	$9.9 \cdot 10^{17}$
9	55	77%	0.4	$2.1 \cdot 10^3$
17	55	71%	1.1	$1.3 \cdot 10^5$

Table A-I shows the numerical measure of the validation performance along with the computational time and the condition number for various dictionaries. One here sees a clear correlation between the condition number of the least squares problem and the computational time. Furthermore, the performance is significantly lower compared to the results reported in Chapter 4.

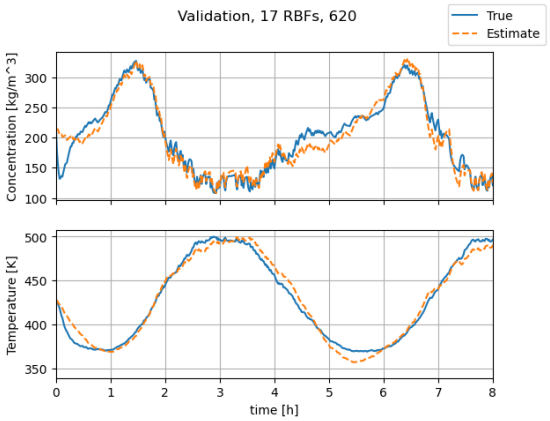

Figure A-3: Validation run using 17RBFs with kernel parameter $\sigma^2 = 620^2$

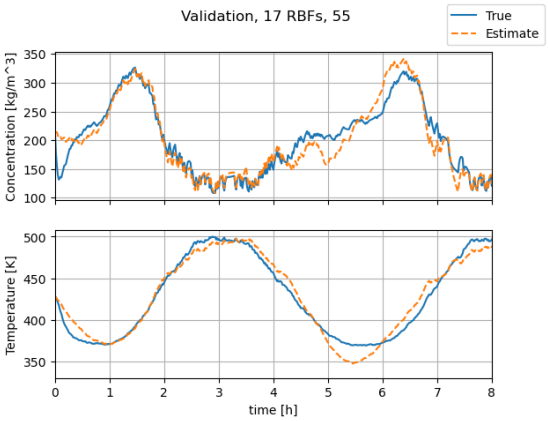

Figure A-4: Validation run using 17RBFs with kernel parameter $\sigma^2 = 55^2$

Bibliography

[1] C. Verhoek, G. I. Beintema, S. Haesaert, M. Schoukens, and R. Tóth, "Deep-learning-based identification of lpv models for nonlinear systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 3274–3280.

[2] S. Z. Rizvi, J. Mohammadpour, R. Tóth, and N. Meskin, "A kernel-based approach to mimo lpv state-space identification and application to a nonlinear process system," *IFAC-PapersOnLine*, vol. 48, no. 26, pp. 85–90, 2015.

[3] V. Verdult, M. Lovera, and M. Verhaegen, "Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics," *International Journal of Control*, vol. 77, no. 13, pp. 1149–1159, 2004.

[4] L. Ljung, *System Identification: An Overview*. London: Springer London, 2015, pp. 1443–1458. [Online]. Available: https://doi.org/10.1007/978-1-4471-5058-9_100

[5] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[6] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, pp. 239–242, 2014.

[7] B. Friedland, *Control System Design: An Introduction to State-Space Methods*. Dover Publications, Inc., 2005.

[8] R. Tóth, P. Van den Hof, J. Ludlage, and P. Heuberger, "Identification of nonlinear process models in an lpv framework," *IFAC Proceedings Volumes*, vol. 43, no. 5, pp. 877–882, 2010.

[9] J. A. Suykens, J. P. Vandewalle, and B. L. De Moor, *Artificial neural networks for modelling and control of non-linear systems*. Springer Science & Business Media, 1995.

[10] G. I. Beintema, M. Schoukens, and R. Tóth, "Deep subspace encoders for nonlinear system identification," *Automatica*, vol. 156, p. 111210, 2023.

[11] R. Tóth, *Modeling and identification of linear parameter-varying systems*. Springer, 2010, vol. 403.

[12] L. Zhang, J. Liu, C. Zhuang, M. Yao, F. Chen, and C. Zhang, "Gain scheduling control of ball screw feed drives based on linear parameter varying model," *The International Journal of Advanced Manufacturing Technology*, vol. 124, no. 11-12, pp. 4493–4510, 2023.

[13] F. R. López-Estrada, J.-C. Ponsart, D. Theilliol, Y. Zhang, and C.-M. Astorga-Zaragoza, "Lpv model-based tracking control and robust sensor fault diagnosis for a quadrotor uav," *Journal of Intelligent & Robotic Systems*, vol. 84, pp. 163–177, 2016.

[14] J. J. Yang, S. Zhang, R. Song, and G. G. Zhu, "Lpv model identification of an evvt system," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 4723–4728.

[15] A. K. Al-Jiboory and G. Zhu, "Static output-feedback robust gain-scheduling control with guaranteed h2 performance," *Journal of the Franklin Institute*, vol. 355, no. 5, pp. 2221–2242, 2018.

[16] J.-W. Van Wingerden, "Control of wind turbines with'smart'rotors: Proof of concept & lpv subspace identification," Ph.D. dissertation, PhD dissertation, Delft University of Technology, 2008.

[17] A. Chiuso and G. Picci, "Consistency analysis of some closed-loop subspace identification methods," *Automatica*, vol. 41, no. 3, pp. 377–391, 2005.

[18] V. Verdult, "Non linear system identification: a state-space approach," Ph.D. dissertation, PhD dissertation, Universiteit Twente, 2002.

[19] P. B. Cox, "Towards efficient identification of linear parameter-varying state-space models," Ph.D. dissertation, PhD dissertation, Eindhoven University of Technology, 2018.

[20] B. Gunes, "A tensor approach to linear parameter varying system identification," Ph.D. dissertation, Delft University of Technology, 2018.

[21] Y. Bao, J. M. Velni, A. Basina, and M. Shahbakhti, "Identification of state-space linear parameter-varying models using artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5286–5291, 2020.

[22] A. Rehmer and A. Kroll, "On affine quasi-lpv system identification with unknown state-scheduling using (deep) recurrent neural networks," in *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2022, pp. 446–451.

[23] ——, "A deep recurrent neural network model for affine quasi-lpv system identification," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 566–571.

[24] M. B. Kennel, R. Brown, and H. D. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical review A*, vol. 45, no. 6, p. 3403, 1992.

[25] C. Vuik and D. Lahaye, *Scientific computing : wi4201*. Delft Institute of Applied Mathematics, 2019.

[26] D. Lay, R. Lay, and J. McDonald, *Linear Algebra and its Applications*, 5th ed. Pearson, 2016.

[27] C. Yu, L. Ljung, A. Wills, and M. Verhaegen, "Constrained subspace method for the identification of structured state-space models (cosmos)," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4201–4214, 2019.

[28] J. Noom, O. Soloviev, and M. Verhaegen, "Data-driven fault diagnosis under sparseness assumption," 2022.

[29] S. Z. Rizvi, J. Mohammadpour, R. Tóth, and N. Meskin, "An iv-svm-based approach for identification of state-space lpv models under generic noise conditions," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 7380–7385.

[30] J. R. Dormand and P. J. Prince, "A family of embedded runge-kutta formulae," *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.

[31] V. Verdult, M. Verhaegen, and J. Scherpen, "Identification of nonlinear nonautonomous state space systems from input-output measurements," in *Proceedings of IEEE International Conference on Industrial Technology 2000 (IEEE Cat. No. 00TH8482)*, vol. 1. IEEE, 2000, pp. 410–414.

[32] A. Isidori, *Nonlinear control systems: an introduction*. Springer, 1985.

[33] J. Stark, "Delay embeddings for forced systems. i. deterministic forcing," *Journal of Nonlinear Science*, vol. 9, pp. 255–332, 1999.

[34] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physical review A*, vol. 33, no. 2, p. 1134, 1986.

Glossary

List of Acronyms

LPV	Linear Parameter-Varying
IO	Input-Output
VAF	Variance Accounted For
LTI	Linear Time-Invariant
CSTR	Continuously Stirred Tank Reactor
RBF	Radial Basis Function
PEM	Prediction Error Method
SVD	Singular Value Decomposition
qLPV	quasi-Linear Parameter-Varying (LPV)
PWA	Piece-Wise Affine
PRBS	Pseudo Random Binary Sequence
SNR	Signal to Noise Ratio
BFR	Best Fit Rate
LTV	Linear Time-Varying
LQR	Linear Quadratic Regulator
MPC	Model Predictive Control
OBF	Orthonormal Basis Function

List of Symbols

$A(\cdot)$	Matrix function governing the influence of the current state on the next state
A^i	Sum of all A^i_j with i in the superscript, multiplying the same entry in θ
A^j_i	The product of A_i and α^j_i
A_i	Matrix governing the influence of the current state on the next state, multiplies the i-th entry of the scheduling signal/vector
A_{HE}	Surface area of the heat exchanger
$B(\cdot)$	Matrix function governing the influence of the current input on the next state
B^i	Sum of all B^i_j with i in the superscript, multiplying the same entry in θ
B^j_i	The product of B_i and α^j_i
B_i	Matrix governing the influence of the current input on the next state, multiplies the i-th entry of the scheduling signal/vector
$C(\cdot)$	Matrix function governing the influence of the current state on the current output
C^i	Sum of all C^i_j with i in the superscript, multiplying the same entry in θ
C^j_i	The product of C_i and α^j_i
C_i	Matrix governing the influence of the current state on the next output, multiplies the i-th entry of the scheduling signal/vector
C_{in}	Inflow concentration
C_{rc}	Concentration in the reactor
$D(\cdot)$	Matrix function governing the influence of the current input on the current output
D^i	Sum of all D^i_j with i in the superscript, multiplying the same entry in θ
D^j_i	The product of D_i and α^j_i

D_i	Matrix governing the influence of the current input on the next output, multiplies the i-th entry of the scheduling signal/vector
E_A	Activation energy of the reaction
$H(\cdot)$	Matrix function
N	Number of data points
Q_{in}	Inflow rate
Q_{out}	Outflow rate
R	Matrix used to define the rank constraint for LPV-REFUS
R_i	Matrix used to define the rank constraint for the i-th entry of the scheduling variable using LPV-CONFUS
R_{rc}	Gas constant
T_c	Coolant temperature
T_{in}	Temperature of the inflowing liquid
T_{rc}	Temperature in the reactor
U	Unitary matrix containing the left-hand singular vectors of R
U_{HE}	Heat transfer coefficient of the heat exchanger
V^T	Unitary matrix containing the right-hand singular vectors of R
V_{rc}	Reactor Volume
$\bar{A}$	Composite matrix consisting of the sums over i of A_i^j stacked next to each other with increasing j
$\bar{B}$	Composite matrix consisting of the sums over i of B_i^j stacked next to each other with increasing j
$\bar{C}$	Composite matrix consisting of the sums over i of C_i^j stacked next to each other with increasing j
$\bar{D}$	Composite matrix consisting of the sums over i of D_i^j stacked next to each other with increasing j
$\bar{\Sigma}$	Matrix containing the r largest singular values of R
ΔH	Heat of the reaction
Δ	General matrix used in proofs
Φ	The matrix multiplying the vector of unknowns in the regular least squares formulation
Σ	Matrix containing the singular values of R
Θ	Generalized optimization variables

Ξ	Matrix with the same κ_2 as Φ
α_i	Row vector with the weights relating θ_k to p_k^i
α_i^j	Scalar weight relating θ_k^j to p_k^i
β	Scalar variable used in proofs
δ	Threshold for the false nearest neighbour approach
δ_s	Threshold, false nearest neighbour, number of past inputs and outputs considered by the nonlinear input-output model
ϵ_k	Error vector at time step k
η	Embedding delay used to determine the final entries in the state estimate
γ	Vector of unknowns in the regular least squares formulation
$\hat{\mathbf{x}}$	State estimate using a nonlinear Input-Output (IO) model with zero future inputs
κ_2	Condition number using the 2-norm
ω	Weight in the affine representation obtained by [2]
ϕ	Iterative weight in the convex-concave procedure
π	Ratio of the circumference to the diameter of a circle, 3.14....
ψ	Vector of knowns in the regular least squares formulation
ρ_{KS}	Number of unknowns for the full state measurement with Known Scheduling
$\rho_{LPV-CONFUS}$	Number of unknowns for the rank constrained full state measurement with unknown scheduling
$\rho_{LPV-REFUS}$	Number of unknowns for the reduced rank constrained full state measurement with unknown scheduling
ρ_{rc}	Density of the reactant for the continuously stirred tank reactor example
σ^2	Variance
σ_i	i-th largest singular value of R
τ	Iterator used to determine the state dimension
θ	Dictionary of scheduling signals
ξ_p	Nonlinear map from past inputs and outputs to a current state estimate
ζ	Nonlinear map computing the optimal scheduling variable based on the current system state estimate and input
c_ρ	Specific heat
h	Liquid level, height
i	Iterator or counter

j	Iterator or counter
k	time step indicator
k_0	Pre-exponential term in the analytical equations for the continuously stirred tank reactor
l	Output dimension
m	Input dimension
n	State dimension
p_k	Scheduling variable or vector at time step k
p_k^i	i-th entry of the scheduling variable or vector at time step k
q	Number of entries in the scheduling dictionary θ
r	Scheduling Dimension
u_k	Input at time step k
x_k	State at time step k
y_k	Output at time step k