

**DATA-DRIVEN AND ADAPTIVE ALGORITHMS FOR PRESSURE
CONTROL IN A MULTI-PHASE FLOW SYSTEM**

BY

AIMAN FARID BAWAZIR

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEM AND CONTROL ENGINEERING

DECEMBER 2023

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **AIMAN FARID BA WAZIR** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN SYSTEMS & CONTROL ENGINEERING**.



Dr. Mujahed Al-Dhaifallah
Department Chairman



Prof. Suliman S. Al-Homidan
Dean of Graduate Studies

Date



Dr. Nezar M. Al-Yazidi
(Advisor)



Prof. Abdul-Wahid A. Saif
(Member)



Prof. Sami El-Ferik
(Member)

© Aiman Farid Bawazir

2023

This work is dedicated to my beloved family

ACKNOWLEDGMENTS

I am grateful to Almighty Allah for giving me the patience, fortitude, wisdom, and health I needed to do this effort. My sincere appreciation to the control and instrumentation engineering department at King Fahd University of Petroleum and Minerals (KFUPM) for their support throughout the years as both an undergraduate and graduate student. I want to thank my supervisor, Dr. Nezar Al Yazidi, whose knowledge and experience were crucial in improving the research objectives and methods. He also has provided me with tremendous encouragement, advice, insightful comments, and mentoring. I remain grateful to Prof. Abdulwahed Saif and Prof. Sami El-Ferik, members of the committee, for their incredibly useful assistance to this project and who were always available when I needed them and contributed to my academic and professional achievements. I want to sincerely thank Dr. Mustafa AL Nasser, from Yokogawa Saudi Arabia, Dr. Rahul N. Gajbhiye and Mr. Ala S. AL-Dogail, from College of Petroleum Engineering and Geosciences, for their generous contribution in sharing valuable data for this research. Their willingness to collaborate and share insights has significantly enriched the depth and scope of this work.

Finally, I owe a debt of appreciation to my family: my parents for establishing in me a love of creativity and hardworking, and my brothers and sisters for always being there for me. My wife and son deserve special mention for their unfailing care and constant support. I give everything, including this, to everyone who supports me. |

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
ABSTRACT	xiv
ملخص الرسالة.....	xv
CHAPTER 1 INTRODUCTION	1
1.1. Background	1
1.2. Motivation.....	1
1.3. Objectives.....	2
1.4. Methodology	2
1.5. Thesis Organization.....	3
CHAPTER 2 LITERATURE REVIEW	5
2.1. Introduction	5
2.2. Multi-Phase Flow Experiment	10
CHAPTER 3 SYSTEM IDENTIFICATION	12

3.1. Introduction	12
3.2. Neural Network-Based Model.....	14
3.2.1. Introduction.....	14
3.2.2. Methodology	15
3.2.3. Results and Discussion	16
3.3. Nonlinear ARX Model	19
3.3.1. Introduction.....	19
3.3.2. Nonlinear ARX Model Structure	19
3.3.3. Dependency of ARX Parameters on Output	21
3.3.4. ARX Model Parameters Variation with Input and Output.....	23
3.3.5. Results of Nonlinear ARX Model Identification	24
3.4. Nonlinear Hammerstein-Wiener Model	26
3.4.1. Introduction.....	26
3.4.2. Nonlinear Hammerstein-Wiener Model Structure	26
3.4.3. Results of Nonlinear Hammerstein Model Identification.....	27
3.4.4. Identification of SISO Hammerstein Model	29
3.5. Summary	35
CHAPTER 4 NONLINEAR CONTROL DESIGN.....	37

4.1.	Introduction	37
4.2.	Adaptive PID Control Using Neural Network.....	37
4.2.1.	Introduction.....	37
4.2.2.	Architecture of the Control System.....	38
4.2.3.	Adaptation Process of PID Controller.....	40
4.2.4.	Results of the Adaptive PID Control.....	42
4.3.	Sliding Mode Controller for The Hammerstein Model.....	45
4.3.1.	Introduction.....	45
4.3.2.	Sliding Mode Control Results.....	48
CHAPTER 5 LINEARIZED MODEL CONTROL DESIGN		51
5.1.	Introduction	51
5.2.	Tangent Linearization	52
5.3.	Discrete PID with Linearized Model.....	55
5.4.	Linear Quadratic Regulator	62
5.5.	Sliding Mode Control (SMC)	66
CHAPTER 6 CONCLUSION AND FUTURE WORK		72
6.1.	Conclusion	72
6.2.	Future Work	73

REFERENCES.....	75
Vitae	85

LIST OF FIGURES

Figure 1	Horizontal Flow Loop[1].....	10
Figure 2	The Horizontal Flow Loop Diagram	11
Figure 3	Neural Network Training Performance (Plot Performance).....	17
Figure 4	Regression Plot	18
Figure 5	Nonlinear ARX Model Structure.....	20
Figure 6	Identification around the designated level of output [54].....	22
Figure 7	Model Estimation Using NLARX Model with Support Vector Machine as Output Function	24
Figure 8	Model Estimation Using NLARX Model with Sigmoid network as Output Function	25
Figure 9	Hammerstein-Wiener Model Structure	27
Figure 10	The Hammerstein model.....	27
Figure 11	Model Estimation Using Nonlinear Hammerstein Model with Sigmoid Network of 10 units and order 4 TF.....	28
Figure 12	Model Estimation Using Nonlinear Hammerstein Model with Sigmoid Network of 15 units and order 3TF.....	29
Figure 13	Output (the upper figure) and Input (the lower figure) Signals	34
Figure 14	Measured (Pressure Drop)) and Simulated Model Output	34
Figure 15	Adaptive PID Control using Neural Network Architecture.....	39

Figure 16	The Online NN Model Training	40
Figure 17	Step Response of Adaptive PID.....	43
Figure 18	The Control Signal of Adaptive PID	44
Figure 19	The error Signal for Step Response	44
Figure 20	PID Parameters Gain.....	45
Figure 21	Output vs. Estimated Output Using SMC.....	49
Figure 22	Absolute Error between Output and Estimated Output	50
Figure 23	Hammerstein Model as a Static Nonlinear Input and Linear State Space System.....	53
Figure 24	Step Response of the Linearized Model (Open Loop).....	55
Figure 25	Closed-Loop Step Response of The Linearized Model with PID Controller	58
Figure 26	The Control Signal u of PID	59
Figure 27	The Error Signal e Between the Step Response and The Linearized Response with PID	59
Figure 28	Closed-Loop Output Response vs Estimated Output Response of The Linearized Model with PID Controller	60
Figure 29	The Control Signal u of PID	60
Figure 30	The Absolute Error Signal e (Output- Estimated Output Responses) with PID	61

Figure 31	Closed-loop Step Response With LQR.....	63
Figure 32	The Control Signal u of LQR.....	64
Figure 33	The error signal e of Linearized response with LQR.....	64
Figure 34	Output Response vs Estimated Output Response of The Linearized Model with LQR	65
Figure 35	The Control Signal u of SMC	65
Figure 36	The Absolute Error Signal e Between the Output Response and The Estimated Output Response with LQR.....	66
Figure 37	Closed-Loop Step Response of Linearized Model With SMC.....	68
Figure 38	The Control signal u of SMC.....	69
Figure 39	The Error Signal e of The Linearized Response With SMC	69
Figure 40	Closed-Loop Output Response vs Estimated Output Response of The Linearized Model with SMC Controller	70
Figure 41	The Control Signal u of SMC	70
Figure 42	The Absolute Error Signal e Between the Output Response and The Estimated Output Response with SMC.....	71

LIST OF ABBREVIATIONS

PID	:	Proportional-Integral-Derivative
SMC	:	Sliding Mode Controller
EIV	:	Errors-In-Variables
ANN	:	Artificial Neural Network
ARX	:	Auto Regression eXogenous
NARX	:	Nonlinear Auto Regression eXogenous
H-W	:	Hammerstein -Weiner
N-L-N	:	Nonlinear-linear-Nonlinear
SISO	:	Single Input and Single Output
LQR	:	Linear Quadratic Regulator
MPC	:	Model Predictive Controller

ABSATTRACT

Full Name : Aiman Farid Bawazir

Thesis Title : Data-Driven and Adaptive Algorithms for Pressure Control in A Multi-Phase Flow System

Major Field : Systems and Control Engineering

Date of Degree : December, 2023

Investigating experimental data of process control systems is a major task in industrial applications. Multi-phase flow in pipes commonly exists in building petroleum and chemical systems. Accordingly, pressure drop must be controlled in a good way to design such systems. This study has examined the phenomenon of pressure drop in a flow consisting of both gas and liquid phases from a control point of view. Estimation of the model is performed by using system identification methods. It includes using neural network-based model, Auto Regression eXogenous (ARX) model and Hammerstein model. Based on the characteristics and responses, the best model will be selected. The PID control mechanism continuously modifies the model parameters of the system to consider any changes in the system's behavior and Linear Quadratic Regulator (LQR) is used to derive the optimal control while in the Sliding Mode Controller (SMC), the algorithm generates the system's optimal control signals. This approach has the potential to operate more effectively and efficiently for multi-phase systems, which are commonly employed in industries such as oil and gas, chemical processing, and wastewater treatment.

ملخص الرسالة

الاسم:

أيمن فريد احمد باوزير

عنوان الرسالة:

الخوارزميات المبنية على البيانات وقابلية التكيف للتحكم بالضغط في نظام تدفق متعدد الحالات

التخصص:

هندسة الأنظمة والتحكم

تاريخ الدرجة العلمية: ديسمبر 2023

تعتبر استكشاف البيانات التجريبية لأنظمة التحكم في العمليات مهمة رئيسية في التطبيقات الصناعية. يوجد التدفق متعدد الحالات في الأنابيب بشكل شائع في بناء أنظمة النفط والكيماويات. وبناءً على ذلك، يجب التحكم في انخفاض الضغط بطريقة جيدة لتصميم مثل هذه الأنظمة. قامت هذه الدراسة بفحص ظاهرة انخفاض الضغط في تدفق يتألف من مراحل الغاز والسائل من وجهة نظر التحكم. يتم أداء تقدير النموذج باستخدام أساليب تحديد النظام. يشمل ذلك استخدام نموذج قائم على الشبكة العصبية، ونموذج الانحدار التلقائي الخارجي، ونموذج هامر شتين استنادًا إلى الخصائص والاستجابات، سيتم اختيار أفضل نموذج.

تعد آلية التحكم PID تعديل معلمات النظام بشكل مستمر للاعتبار في أي تغييرات في سلوك النظام، بينما في وحدة التحكم بالوضع المنزلق (SMC)، تقوم الخوارزمية بتوليد إشارات التحكم الأمثل للنظام. تحمل هذه الطريقة إمكانية للعمل بشكل أكثر فعالية وكفاءة لأنظمة متعددة الحالات، والتي تُستخدم عادة في صناعات مثل صناعة النفط والغاز، ومعالجة المواد الكيميائية، ومعالجة مياه الصرف الصحي

CHAPTER 1 INTRODUCTION

1.1. Background

Multi-phase flow refers to the concurrent motion of multiple phases, including natural phenomena like rain, snow, and fog, as well as the flow of gas and liquid within pipelines. Nevertheless, the dynamics of multi-phase systems are far more complex than those of single-phase systems because of the differences in phase distribution, resulting in unique flow patterns or regimes [1]. Many factors and forces affecting fluids determine the class of flow pattern at any point in the pipe.

Many variables influence the flow patterns and pressure drops that occur at specific points in the pipe. The flow factors considered, in addition to pipe diameter, deviation angle, and volume fluxes, include turbulence, surface tension forces, buoyancy, and inertia. Accurate multiphase flow pattern estimation can lead to exact pressure drop forecasting in pipelines. Visual inspection is commonly used to recognize flow patterns; however, using visual inspection in industrial applications to monitor flow patterns and recognition during unstable flow conditions is complex and impractical[2].

1.2. Motivation

The flow pattern maps and equations lack enough accuracy with high relative error (20 to 30%) [3], which raises the need for an accurate and reliable method for identification of the flow patterns and pressure drop. In this manner, developing precise drop predictions of pressure is vital in transporting, processing and production of hydrocarbons from oil and gas fields.

1.3. Objectives

The primary objective of this research is to model and control the pressure drop in a system that involves the flow of multiple phases. The ultimate purpose is to establish a dependable and effective network for extracting, transporting, and treating hydrocarbons derived from oil and gas reservoirs.

The specific goals of the proposed research direction are as follows:

- To maintain a stable and optimal pressure drop within the system while adapting to changes in the process dynamics. This can aid in the improvement of process performance, the reduction of variability, and obtaining consistent product quality.
- Ensure that the process operates within the desired range, even under changing operating conditions or disturbances.
- To analyze the impact of multiphase flow on the dynamic behavior of the process.
- To apply contemporary control techniques in developing an enhanced method for ensuring system reliability.

1.4. Methodology

To accomplish the objective of the proposed work, the techniques are given as the following main steps:

- Cleaning and analyzing pressure drop raw data. Eliminating undesirable information from the dataset, including duplicate or irrelevant observations.
- Applying system identification techniques to come up with different system models and comparing them to choose the best fit to our data.

- Carrying on advanced control methods to the system to check stability, controllability, and observability.
- Utilizing Nonlinear PID (NPID) and sliding mode controllers (SMC) to ensure optimal performance and product quality.
- Validating the efficacy of linear controllers in regulating the linearized system model.
- Simulating the outcomes with simulation platforms such as MATLAB.

1.5. Thesis Organization

The thesis is structured into several chapters, each with a specific focus and purpose.

The initial chapter serves as an introduction, offering a summary of the research subject and delineating the study's objectives. It encompasses a concise background of the research issue and its importance, along with an exploration of the motivation behind the study, An overview of the research methodology and a summary of the study's scope.

The second chapter comprises a literature review, presenting an in-depth examination of the current body of literature concerning pressure control in a multi-phase flow system. This chapter critically evaluates pertinent literature, illuminating strengths and weaknesses in prior studies and identifying gaps that the current research seeks to fill.

The third chapter is the system identification, which outlines some identification techniques that deriving dynamic process models by using process test data. This chapter also includes a discussion of the methods used and provides high model precision within the examined operational span.

The fourth chapter centers on the design of a nonlinear controller. This chapter offers a thorough elucidation of the construction and execution of an adaptable PID Controller and Sliding Mode Controller (SMC) for the nonlinear Hammerstein model. Furthermore, it encompasses an evaluation of their accomplishments and constraints.

The fifth chapter of the book delves into the process of designing a linearized control, with a specific focus on the linearized model and its corresponding controller architecture. The chapter discusses the utilization of linear PID, Linear Quadratic Regulator (LQR), and Sliding Mode Controller (SMC) for designing control systems for the linearized model. It also explores the distinctive characteristics of each controller. The linearized model is a simplified representation of a nonlinear system that is utilized for the purpose of control design.

The concluding chapter serves as a summary, encapsulating the key discoveries of the study and exploring their implications for subsequent research. It additionally offers recommendations for future research endeavors and a discussion on the limitations.

CHAPTER 2 LITERATURE REVIEW

2. Literature Review

2.1. Introduction

In various industries, including oil well drilling [4], [5], multistage fracturing[6], and transportation of hydrocarbons [7], pipes with circular cross-sections are commonly used for multiphase flows. Two or more phases, such as liquid, gas, or solid, are moving in these fluxes. The distribution of phases at each pipe cross-section determines the flow patterns, and they are differentiated by the proportion of each phase and the assumption of equal pressure throughout phases [8]. Within a specific cross-section of a pipe when many phases are flowing, it is commonly assumed that the pressure is consistent throughout all phases. However, the flow pattern is influenced by the physical configuration of the phases. Liquid holdup is the term used to describe the quantity of liquid present in the combination. The forces exerted by other fluids or the conduit wall determine the flow regime, and the flow pattern has a substantial effect on the local pressure gradient. Therefore, to determine the pressure gradient and overall pressure reduction, it is critical to assess the flow regimes and volume fractions across the pipeline. Numerous investigations have investigated pipeline modeling of multiphase flow [7].

A segmentation method and numerical algorithm are employed to ascertain the pressure distribution within a pipe. In multiphase flow, both the flow pattern and pressure gradient exhibit variations over the pipe's length [8]. In order to account for this unpredictability, it is crucial to partition the entire pipe into distinct segments. In each segment, the flow

regime can be regarded as uniform, with a roughly consistent pressure gradient. The numerical technique computes the pressure gradient within each segment at every step, employing multiphase flow correlations and mechanistic models as standard tools.

Correlations derived from laboratory experiments, notably are [9]–[11], and others, are extensively employed. Several studies examine the constraints of these multiphase flow correlations, affirming that each correlation is only applicable within particular ranges of input parameters in order to obtain findings of satisfactory precision. In addition, a multitude of mechanistic models that integrate semi-empirical closure relations are frequently employed in order to forecast diverse characteristics of multiphase flow. with general examples being [12]–[14], and others.

The Leda Flow Point model [15] and OLGAS [16] are two very accurate multiphase flow models that are widely recognized for their ability to calculate pressure drop. These models serve as steady-state equivalents to relevant transient simulators. It is crucial to emphasize that this inquiry excludes the analysis of temporary occurrences in the flow of many phases in pipes, such as the slug flow regime. Transient multiphase flow models, including OLGA, Leda Flow, and some proprietary research codes(e.g., [17]), are effective in capturing such phenomena.

Simulating transient flow regimes, such as slug flows, which are characterized by moving boundaries on dense meshes, is computationally demanding. This creates an opportunity for acceleration through meta-modeling and machine learning. However, it is important to note that pursuing this approach would require a separate and dedicated study.

The review of literature reveals the utilization of machine learning (ML) algorithms in predicting pressure gradients and directly estimating pressure outputs. In [18] the authors employed an Artificial Neural Network (ANN) to predict bottomhole pressure. Conversely, [19] utilized an ANN to forecast bottomhole pressure but recommended well segmentation and identification of flow regimes within each segment for improved accuracy.

During drilling operations, the drilling process takes place within a specific pressure range known as the confined pressure window. This pressure window is determined by the fracture pressure (P_{frac}) and the reservoir pressure (P_{res}). Ensuring the bottomhole pressure (P_{bh}) remains within this range is essential for both safety and efficiency [20]. To achieve Managed Pressure Drilling (MPD), a small excess pressure is maintained to ensure that the bottomhole pressure remains within the desired range of confined pressure. MPD controllers encompass a spectrum of complexity, ranging from basic PID controllers to more sophisticated models such as the nonlinear model predictive controller (NMPC). An ODE-based PI choke pressure controller was introduced in a study [21]. This controller successfully controls system states throughout routine operations such as drilling, pipe extension, surge, and swab. In a separate study [22], utilizing a more straightforward nonlinear system of equations, the pressure at the well's bottom was predicted.. In addition, a choke controller utilizing feedback linearization was created to precisely regulate the choke pressure to a preset setpoint value. Precise estimation of the frictional pressure is critical for attaining the intended bottomhole pressure setpoint within the system. Experimental implementation of the controller has demonstrated superior performance compared to conventional PID controllers. In [23], a PID controller is suggested, with tuning parameters chosen dynamically based on data and a cost function. By employing

feedback linearization, [24] devised a choke controller that integrates an adaptive observer to estimate downhole measurements. [25] Proposed a resilient gain switching method to enable automated operation of Managed Pressure Drilling (MPD) by maintaining a consistent bottomhole pressure. In addition, [20] utilized a Nonlinear Model Predictive Controller (NMPC) that was based on a Hammerstein Wiener model. This approach took advantage of the constraint handling capabilities of NMPC.

An analysis of the process dynamics, system response, and system stability, which includes the step response as well as the poles and zeros, has been conducted utilizing a data experiment. The investigation primarily aimed to investigate the stability and responsiveness of the suggested model, which utilized Auto Regression eXogenous (ARX) and Auto Regression Moving Average eXogenous (ARMAX) models [26], [27]. Additionally, characterizing the dynamic behavior of a system using system identification from data collected in Excel spreadsheet is done in [28]. This model was created using a number of approaches, including the discrete time method[29] gain parameter[30], Laplace transform[31], nonparametric identification[32], Hammerstein-wiener[33], ARX identification[34], Heat transport and flow regimes during boiling [35], transient analysis[36], bioinspired algorithm[37], Pressure drop in a two-phase flow [38], approximating flow of two-phase patterns in horizontal pipes using ANN with nonlinear normalization (namely logarithmic normalization) to appropriately measure data, which improved prediction capability meaningfully[39], generalized poisons momentum functional[40].

Al-Naser et al. [41] proposed a novel approach that employs artificial neural networks (ANNs) and three dimensionless parameters, specifically liquid Reynolds number, gas Reynolds number, and pressure drop multiplier, to enhance the versatility of the model in forecasting flow patterns in horizontal pipes. They also employed a preprocessing phase that involved natural logarithmic normalization of a large dataset, resulting in significant improvements in the accuracy of the predictions. In a separate study, Shanthi and Pappa aimed to improve the precision of identifying flow patterns of two-phase through the use of artificial intelligence techniques that rely on the characteristics extracted from images of the flowing material. To achieve this objective, they utilized fuzzy logic and support vector machines (SVMs) in conjunction with principal component analysis (PCA) [42]. Xiao et al. [43] formulated an artificial intelligence system that predicts the flow regime of gas and liquid nanofluid in vertical mini-channels by combining fuzzy logic and support vector machines with principal component analysis. Ribeiro et al. [44] conducted a study utilizing artificial intelligence to predict pressure gradients in vertical pipes that involve two or three-phase flows.

The classification of flow patterns into major and subcategories by numerous academics led to the development of multiple flow pattern maps according to factors such as pipe orientation, pipe diameters, and gas/liquid velocities. The four primary categories of horizontal flow patterns in pipelines are stratified, intermittent, annular, and dispersed bubble.

2.2. Multi-Phase Flow Experiment

According to [1], the researchers conducted experiments using an air/liquid system in a horizontal flow loop that consisted of a 9.15 m (30 ft) pipe with a 0.0254 m (1 inch) diameter to collect data. Figure 1 and 2 shows a picture and diagram of the experiment used to collect the data. Surface tension was altered by adjusting the surfactant solution's surface tension; calcium bromide was used to change density; and glycerin was used to adjust viscosity. The range of superficial gas velocity is 0 to 18.288 m/sec (0-60 ft/sec), while the range of superficial liquid velocity is 0 to 3.048 m/sec (0-10 ft/s).

To examine the existence of the hysteresis phenomenon, the gas velocity was altered during the flow experiments by first increasing it from minimum to maximum value and then decreasing it from maximum to minimum with fixing the liquid rate (constant liquid flow rate). The steady-state condition was observed at each stage of the experiment based on the pressure response and flow rate. Data was acquired in order to obtain sufficient data points.



Figure 1 Horizontal Flow Loop[1]

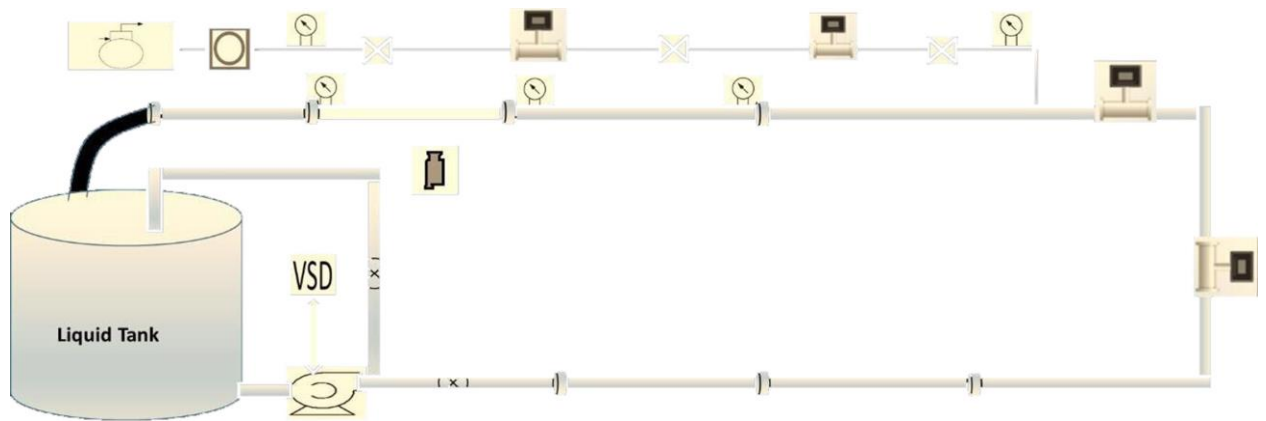


Figure 2 The Horizontal Flow Loop Diagram

CHAPTER 3 SYSTEM IDENTIFICATION

3. System Identification

3.1. Introduction

The prevalent approach in controller design involves the utilization of linear models. Nevertheless, it is imperative to acknowledge that a significant proportion of chemical processes exhibit nonlinearity. Consequently, it is essential to create a mathematical model in order to simulate and regulate the system. This mathematical model serves as a formalized representation, elucidating the intrinsic association between the process's input and its corresponding output. The establishment of process models can be realized through two primary methodologies: physical modeling, which involves the derivation of process models grounded in fundamental principles, encompassing physical, chemical, and biological laws. Process identification entails the derivation of dynamic process models through the utilization of process test data. This method is frequently referred to as black-box modeling, as it does not explicitly invoke physical principles in its approach. When compared with physical modeling, the advantages of identification are:

- Encompassing cost-effectiveness in modeling
- High model precision within the examined operational span.
- The simplicity of a model structure.

Nonetheless, one drawback associated with identification is its limitation in characterizing process behavior solely within the tested operational range.

System identification is a powerful scientific tool used to understand complex systems and their behavior. It involves the analysis and quantification of data gathered from a system in order to build mathematical models that accurately represent the system's behavior. This process enables scientists and engineers to better understand the underlying physical and mathematical principles that govern the system and use this knowledge to make predictions and improve its performance.

System identification finds extensive applications in various domains, including control engineering, signal processing, robotics, and numerous other fields. By providing a deeper understanding of complex systems, system identification plays a crucial role in advancing scientific research and technological development.

It seems like many identification methods presume that input data can always be precisely measured, but in reality, input data can be affected by noise, such as measurement noise. Unfortunately, classical identification methods often do not take noise into account, which can lead to invalid parameter estimation for identified models. This is especially valid when both the input and output data are affected by noise. The errors-in-variables (EIV) model acknowledges the presence of noise in both the input and output variables [45], [46].

In many chemical processes, it is commonly noted that nonlinearity is more prominent in the steady-state gain rather than in the time constants. This characteristic renders such nonlinear models particularly advantageous for the representation of chemical processes. A notable benefit inherent to these models lies in their compatibility with standard linear controller design methodologies. The capability of this methodology stems from its strategic integration of its nonlinear inverse element into the control loop, which has the

potential to mitigate the static nonlinearity that is inherent in the process. The inverse nonlinear function is situated at the process output in the Hammerstein model, whereas it is at the controller output in the Wiener model. As a result, the controller functions solely on the linear dynamic component of the process, allowing for the implementation of traditional linear controller methods. A linear (PID) controller is utilized to regulate the linear component, and its parameters are precisely modified utilizing the Internal Model Control (IMC) methodology.

3.2. Neural Network-Based Model

3.2.1. Introduction

In recent years, the application of neural networks in various fields has expanded exponentially, and one area that has witnessed significant progress is model identification. Traditional model identification methods often rely on mathematical formulations and assumptions that may not capture the complexities of real-world systems. Neural networks, with their ability to learn complex patterns from data, offer a promising alternative for model identification. This section delves into the creation and utilization of a model identification technique based on neural networks. It outlines the outcomes of implementing this method in a real-world scenario. Neural networks possess the capability to accurately approximate nonlinear functions with a high degree of precision [47]–[50]. Different nonlinearities are introduced into a neural network when parameters are altered; these nonlinearities are implemented on an error function via gradient descent. The aforementioned function computes the disparity between the predicted output of a neural network and the real output of a system with respect to specific pairs of input data or output data, which are commonly known as training data. [51]–[53].

3.2.2. Methodology

3.2.2.1. Collection of Data

In order to train the neural network for model identification, a dataset comprising input-output pairs from the target system was collected. The dataset included 5 inputs and 1 output as well as a diverse range of operating conditions and disturbances to ensure the neural network's robustness.

3.2.2.2. Neural Network Architecture

A feedforward neural network structure was selected due to its simplicity and efficiency in capturing nonlinear connections. The structure consisted of an input layer, one or more hidden levels utilizing activation functions, specifically the rectified linear unit (ReLU), and an output layer. The determination of the number of neurons controlled in each layer and the depth of the network was achieved through a systematic process of experimentation and optimization.

3.2.2.3. Training Procedure

The neural network was trained using a supervised learning approach. The dataset was partitioned into training and validation sets using a random selection process in order to assess the model's performance during the training process. The training process involved minimizing a suitable loss function through backpropagation and gradient descent optimization.

3.2.3. Results and Discussion

MATLAB is used to create and train a feedforward neural network. The network design comprises three hidden layers, each with 32, 64, and 128 neurons, respectively, each employing the hyperbolic tangent sigmoid transfer function. The chosen training algorithm is the scaled conjugate gradient descent (`trainscg`), although the Levenberg-Marquardt algorithm (`trainlm`) is offered as an alternative. A training set containing 70% of the data, a validation set comprising 15% of the data, and a testing set containing the remaining 15% of the data comprise the dataset. Training parameters, such as the number of epochs, minimum gradient, and maximum validation failures, are meticulously specified. Following the training phase, the network is evaluated using the provided input data and the performance is assessed. Additionally, there are commented-out sections for viewing the network architecture, saving the trained model, and generating a Simulink model. These functionalities can be uncommented and utilized based on specific needs, providing a comprehensive framework for building and training a neural network in MATLAB.

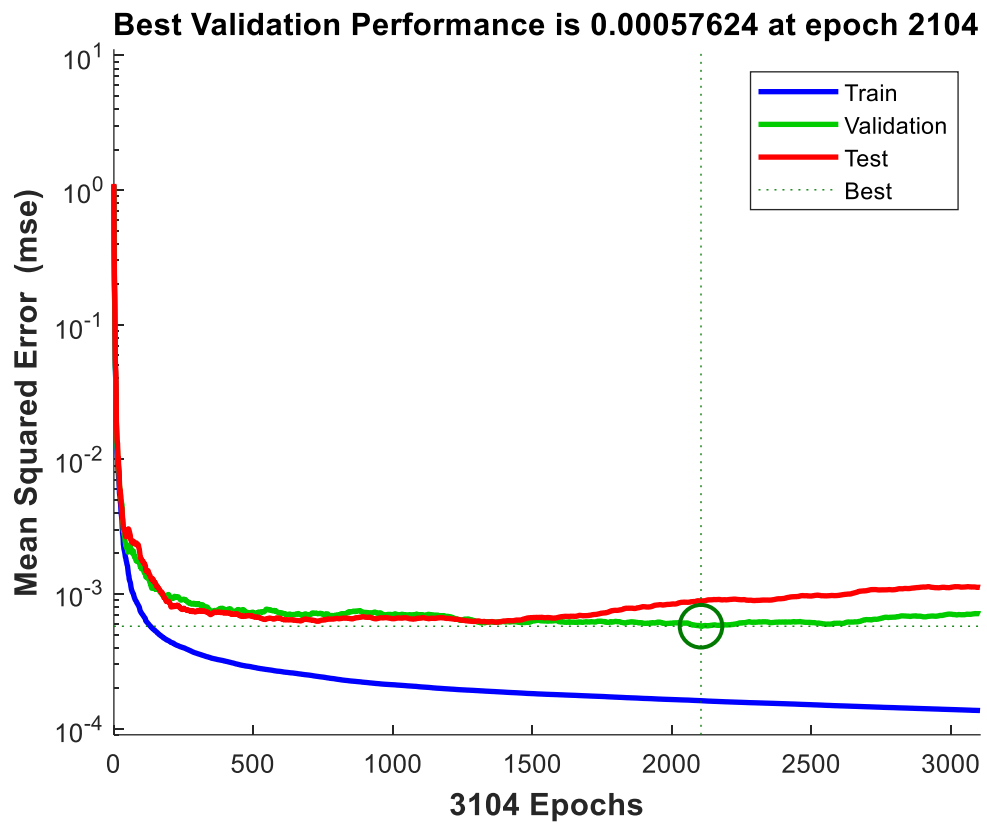


Figure 3 Neural Network Training Performance (Plot Performance)

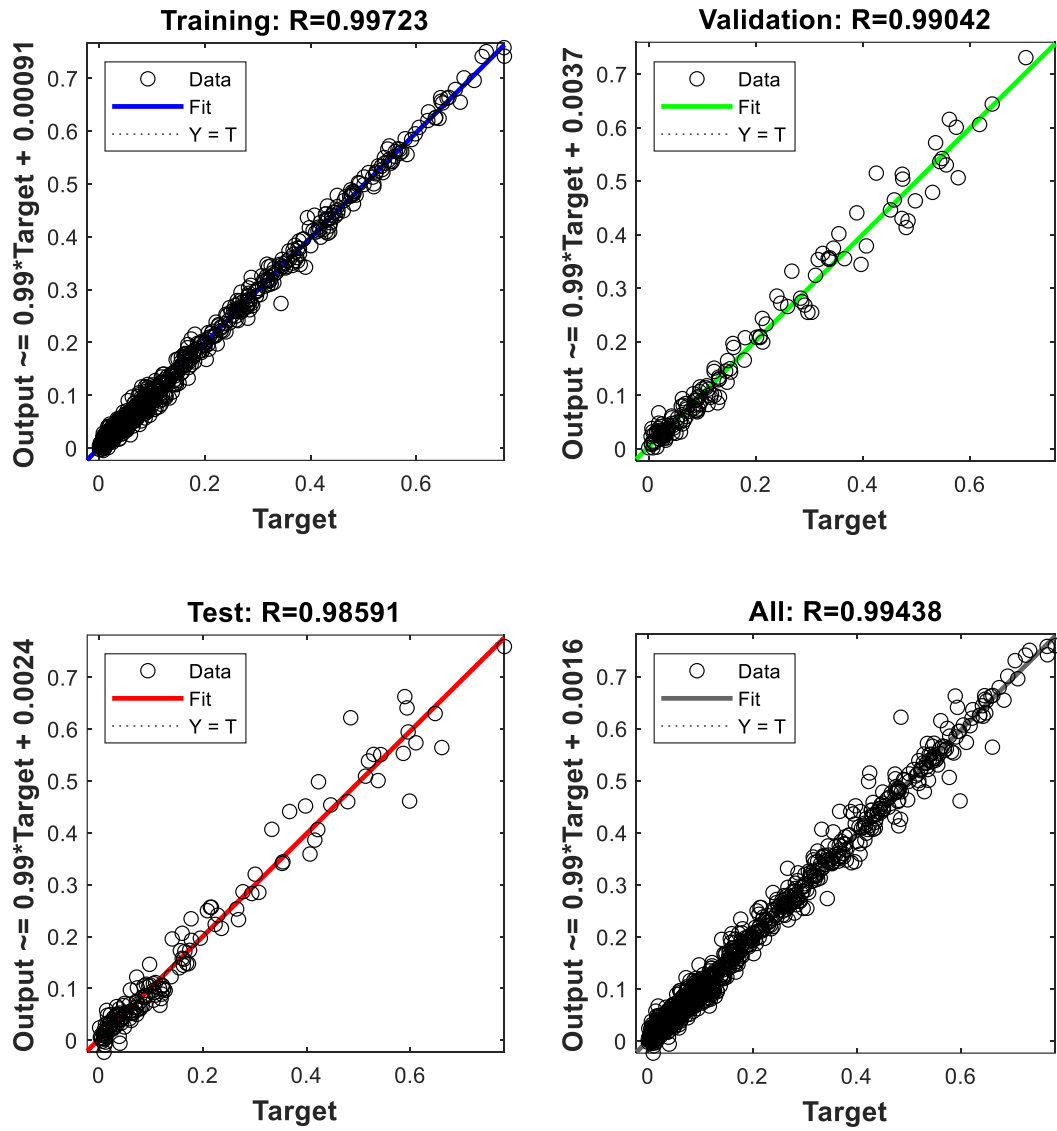


Figure 4 Regression Plot

The neural network-based model identification method offers several advantages over traditional approaches. It can handle complex, nonlinear relationships without relying on explicit mathematical formulations, making it suitable for systems with unknown or highly nonlinear dynamics. Additionally, the ability of neural network to learn from data enables

the identification of intricate patterns that may be challenging to capture using conventional methods.

Despite its advantages, the neural network-based approach poses challenges such as the need for substantial amounts of training data and the potential for overfitting. Careful attention must be paid to the selection of network architecture and hyperparameters to ensure robust performance across various scenarios.

3.3. Nonlinear ARX Model

3.3.1. Introduction

Nonlinear Auto Regressive eXogenous (ARX) models go beyond linear ARX models to handle nonlinear situations. These models employ diverse functions like wavelet and sigmoid networks to capture intricate nonlinear behavior. Unlike to traditional approaches in identifying nonlinear systems, which typically rely on pre-determined model structures, these models provide flexibility by accommodating different designs. As demonstrated by the Hammerstein-Wiener model, they may, for instance, integrate a fusion of linear and nonlinear memoryless dynamical systems.

3.3.2. Nonlinear ARX Model Structure

A nonlinear ARX model is composed of an output function and model regressors. The mapping objects associated with model outputs are incorporated into the output function, which also includes nonlinear and linear functions. The model output is generated through the interaction of these functions with model regressors, in addition to a constant

offset for that particular output. The provided diagram depicts the arrangement of a solitary output nonlinear ARX model within a simulation scenario.

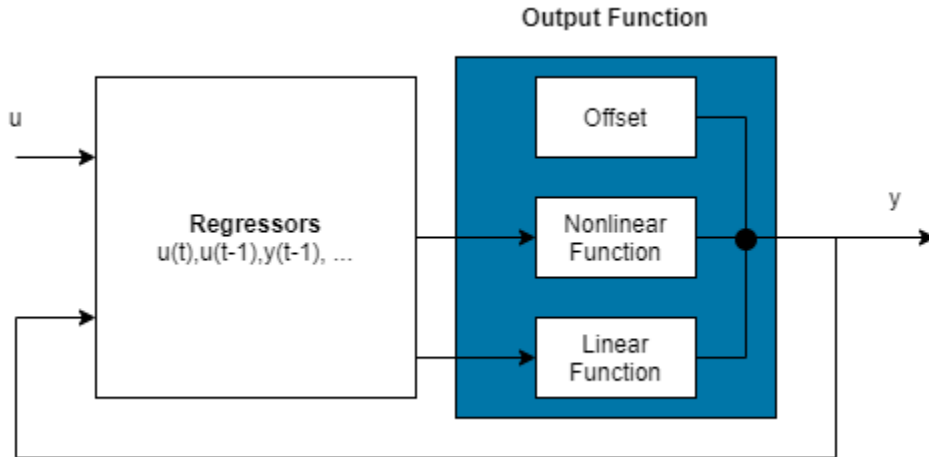


Figure 5 Nonlinear ARX Model Structure

This section outlines a method to identify nonlinear systems by utilizing operational data in the ARX format, where the parameters exhibit nonlinearity in relation to the input and output. The parameters for precise input and output levels are derived from corresponding databases. By iteratively identifying at different levels, one can establish the nonlinear connections among parameters within the specified input and output range. When there is uncertainty about the model order, the locally identifiable model is minimized, and the parameters are expressed as polynomial functions of the input and output variables to construct the nonlinear ARX model. This method is beneficial for compressing intricate, steady nonlinear systems into a succinct nonlinear ARX model that is appropriate for controller design. [54].

This part outlines the proposed method for detecting a nonlinear ARX model, where the parameters exhibit nonlinear relationships with both the input and output [55]. The equation representing the ARX model for the discrete-time linear system is given.

$$\begin{aligned} y[k] + a_1y[k - 1] + a_2y[k - 2] + \dots + a_ny[k - n] \\ = b_1u[k - 1] + b_2u[k - 2] + \dots + b_nu[k - n] + c \end{aligned} \quad (1)$$

In the given equation, $y[k]$ represents the output and $u[k]$ represents the input at time k . The coefficients a_i and b_i (where $i=1, 2, 3, \dots, n$) reflect the values assigned to the variables, while n indicates the system's order. The values of the parameters of the model are determined by the output $y[j]$ and the input $u[j]$ at time k , where j is equal to $k-n, k-n+1, k-n+2, \dots, k$.

3.3.3. Dependency of ARX Parameters on Output

This subsection provides a comprehensive discussion of the identification technique for the nonlinear ARX model, with a specific focus on the output aspect. In this particular instance, the process of identifying the nonlinear system requires an iterative approach to determine the parameters in the ARX model. This is done by using input-output data from various output levels, as shown in Figure 6. The ARX model obtained corresponds to the localized linear system centered around the selected output tier. The parameters assigned to different output tiers can be represented as functions that depend on the output. The process of determining the relevant local linear ARX model is illustrated in Figure 6. The parameters are determined through the least square criterion, defining the linear ARX model around y_o as:

$$\begin{aligned}
& y[k_1] + a_1 y[k_1 - 1] + \cdots + a_n y[k_1 - n] \\
& = b_1 u[k_1 - 1] + \cdots + b_n u[k_1 - n] + c
\end{aligned} \tag{2}$$

$$\begin{aligned}
& y[k_2] + a_1 y[k_2 - 1] + \cdots + a_n y[k_2 - n] \\
& = b_1 u[k_2 - 1] + \cdots + b_n u[k_2 - n] + c
\end{aligned} \tag{3}$$

$$\begin{aligned}
& y[k_3] + a_1 y[k_3 - 1] + \cdots + a_n y[k_3 - n] \\
& = b_1 u[k_3 - 1] + \cdots + b_n u[k_3 - n] + c
\end{aligned} \tag{4}$$

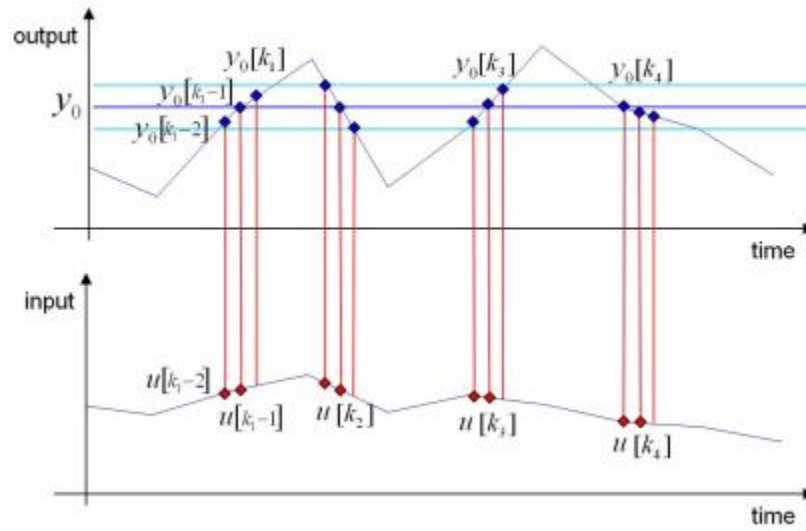


Figure 6 Identification around the designated level of output [54]

As indicated in equation (5), the coefficients of the ARX model are expressed as functions of the output y_o .

$$\begin{aligned} y[k] + a_1(y_o)y[k - 1] + \dots + a_n(y_o)y[k - n] \\ = b_1(y_o)u[k - 1] + \dots + b_n(y_o)u[k - n] + c(y_o) \end{aligned} \quad (5)$$

To avoid the use of output data across multiple levels, the process of identifying a local ARX model can be carried out by choosing the output to stay within a specific range centered around a particular output level. Through iterative identification across the defined output range and combining the obtained models, the successful identification of the nonlinear ARX model is achievable.

3.3.4. ARX Model Parameters Variation with Input and Output

We explored the process of identifying ARX model parameters associated with the output in the preceding section. In this section, parameters that are impacted by both the input variable (u) and the output variable (y) are introduced via an iterative process. This involves the detection of local autoregressive exogenous (ARX) models where the input and output values fall within a predetermined range. The establishment of an ARX model involves the methodical selection of various input and output levels, with the parameters that are affected by both the input and output variables being determined using Equation (6).

$$\begin{aligned}
y[k] + a_1(u, y)y[k - 1] + \dots + a_n(u, y)y[k - n] \\
= b_1(u, y)u[k - 1] + \dots + b_n(u, y)u[k - n] \\
+ c(u, y)
\end{aligned} \tag{6}$$

The synthesis of local ARX models recognized at various stages of both input and output throughout the designated range achieves the determination of the nonlinear ARX model.

3.3.5. Results of Nonlinear ARX Model Identification

After preparing the data, they are fed into MATLAB for identification process. In this subsection, the results of different setup are presented and the impact of data loss and fitting to estimation is discussed in order to validate the proposed algorithm.

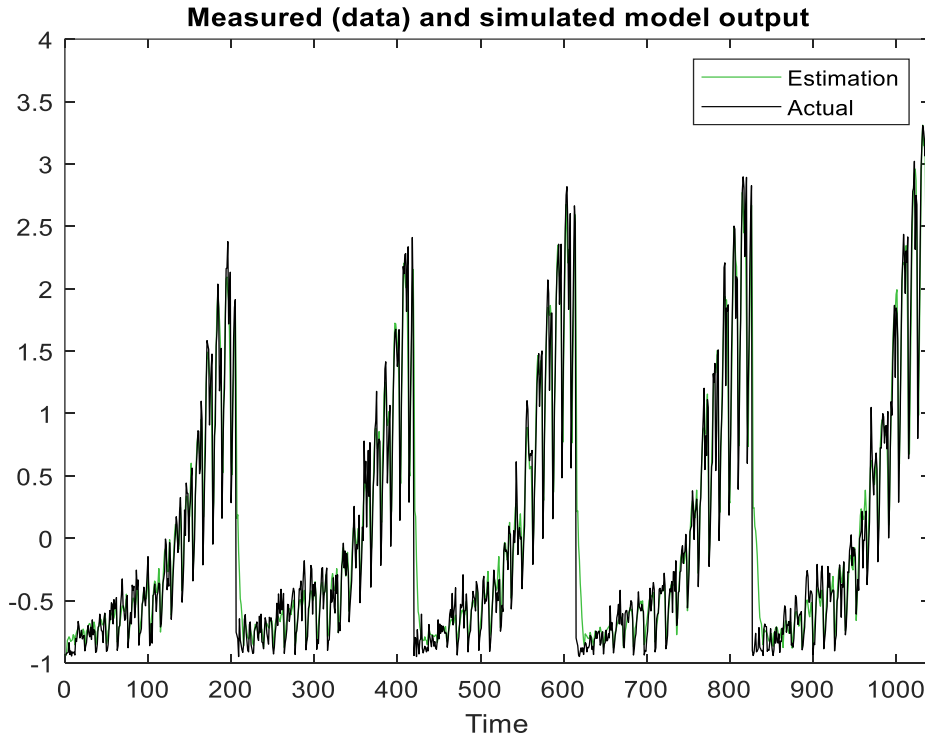


Figure 7 Model Estimation Using NLARX Model with Support Vector Machine as Output Function

Figure 7 was estimated using Nonlinear ARX model with 1 output and 5 inputs on time domain data of sampling time 1 second. The regressors are linear regressor and polynomial order 2 regressor. The output function is support vector machine function using a Gaussian kernel. The fit to estimation data is 78.77% and the mean square error is 0.04504.

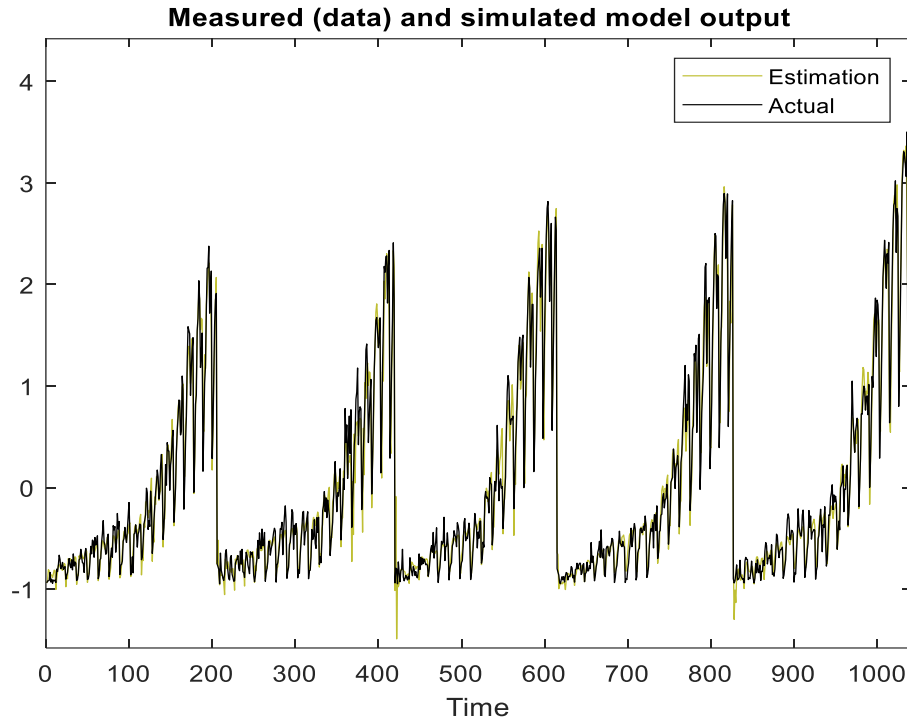


Figure 8 Model Estimation Using NLARX Model with Sigmoid network as Output Function

Figure was estimated using Nonlinear ARX model of 5 inputs and 1 output on time domain data of sampling time 1 second. The regressors are linear regressor and polynomial order 2 regressor. The output function is sigmoid network with 25 units. The fit to estimation data is 86.52% and the mean square error is 0.01816.

3.4. Nonlinear Hammerstein-Wiener Model

3.4.1. Introduction

N-L-N models, also referred to as nonlinear block-oriented models, belong to a distinct category of models. These models incorporate a Linear (L) subsystem that is situated between two subsystems of static Nonlinearity (N) [56]. By incorporating two nonlinear components instead of just one, the modeling capabilities of H-W models are enhanced, particularly for nonlinear systems that display both actuator and sensor nonlinearities [57]. The recognition of H-W models has garnered significant interest in recent years, with multiple studies exploring various frameworks.

3.4.2. Nonlinear Hammerstein-Wiener Model Structure

The structure of the model is formulated by dividing the system into blocks. These blocks consist of static nonlinear components combined with dynamic linear terms to form the complete system. The H-W model operates in discrete-time. Figure 9 illustrates the structural depiction of the H-W model. Specifically, if the model integrates solely the input nonlinearity f , it is denoted as a Hammerstein model (see Figure 10). Likewise, if the model encompasses nonlinearity h solely at the output, it is termed a Wiener model.

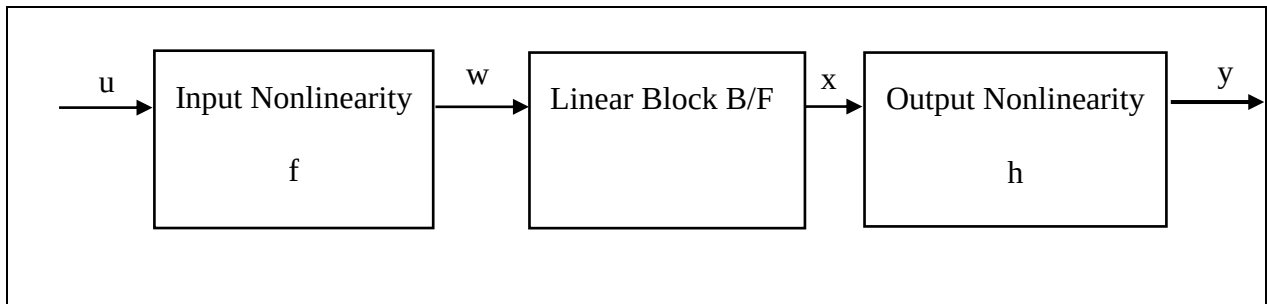


Figure 9 Hammerstein-Wiener Model Structure

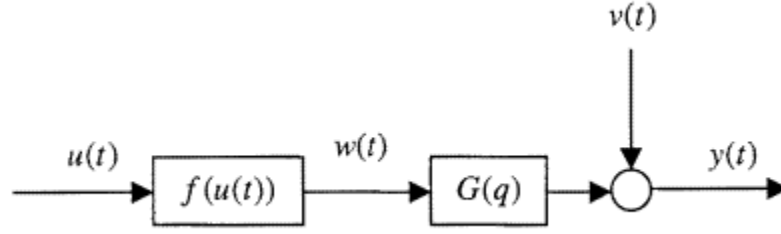


Figure 10The Hammerstein model

3.4.3. Results of Nonlinear Hammerstein Model Identification

In the context of identifying a Hammerstein model using MATLAB's System Identification Toolbox, the static nonlinearity component can be specified as 5 sigmoid networks (as the number of the inputs) with 10 units each. This configuration enables the model to capture intricate and non-linear connections between the input and output signals. The Levenberg-Marquardt (LM) searching method, a widely used optimization algorithm, is employed in the identification process. The LM approach enables the adjustment of the network's parameters, leading to enhanced accuracy in approximating nonlinearities. The dynamic linear part is estimated using a transfer function with 3 poles and 4 zeros. The specific architecture and the LM searching method together enable a robust and efficient identification of the model. The fitting to estimation is 66.9%, and the mean square error is 0.1116. The representation of the underlying system dynamics is illustrated in the figure 11.

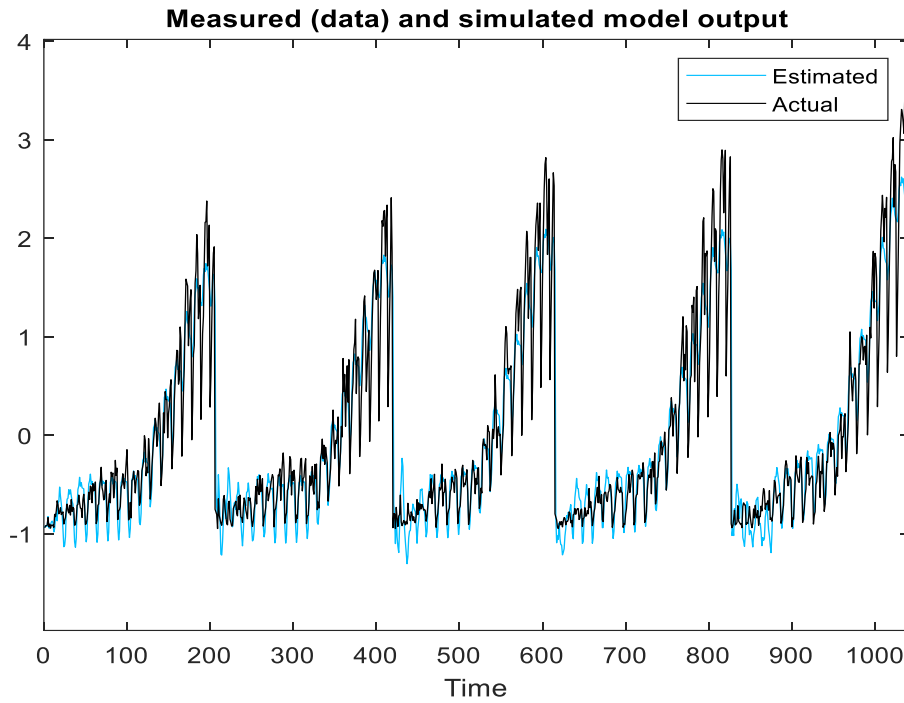


Figure 11 Model Estimation Using Nonlinear Hammerstein Model with Sigmoid Network of 10 units and order 4 TF

Repeating the same process and increasing the number of units in the static function to 15 and reduce the order of the linear dynamic function to 3, the fitting to estimation is increased to 80.12% and the mean square error reduced to 0.03956. See figure 12.

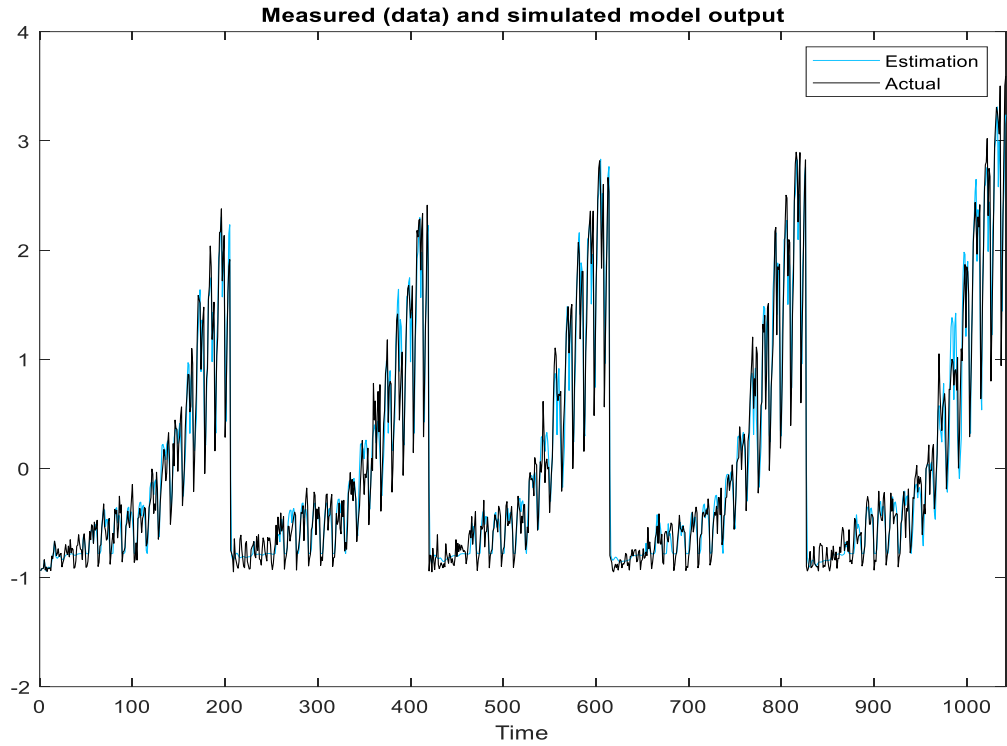


Figure 12 Model Estimation Using Nonlinear Hammerstein Model with Sigmoid Network of 15 units and order 3TF

3.4.4. Identification of SISO Hammerstein Model

Within this section, we have applied the identical approach as in the prior section to identify the Hammerstein model. This model represents a system that operates in discrete time and has one input and one output. The system is linear and time-invariant. Here, the input is defined as the liquid speed to gas speed ratio, and the output corresponds to the measured pressure drop in psi.

The Hammerstein model is composed of a sequence of static and dynamic blocks. Nonlinearity in the inputs is captured and transformed into $w(t)$ by the static input block, as described in Equation (8a). Following the transformation, the unmeasured term, denoted

as $w(t)$, is subjected to a linear function block in which $G(q)$ symbolizes a time-invariant linear transfer function. Equation (8b) [20] is utilized by this linear function to transform the linear input into the output $x(t)$. The $v(t)$ noise term is a stationary, bounded stochastic signal. It is imperative to underscore that the model's output does not incorporate nonlinearity. The degrees of the numerator and denominator in the linear functional blocks are denoted by the parameters m and n , respectively. The index α serves as a measure of nonlinearity.

The nonlinear function, parameterized through a polynomial function, can be expressed as:

$$f(u(t)) = \beta_1 u(t) + \beta_2 u^2(t) + \dots + \beta_m u^m(t) \quad (7)$$

$$w(t) = f(\alpha, u) \quad (8a)$$

$$x(t) = \frac{B(q^{-1})}{A(q^{-1})} w(t) \quad (8b)$$

$$G(q) = \frac{B(q)}{A(q)} = \frac{b_o + b_1 q^{-1} + \dots + b_m q^{-m}}{1 + a_1 q^{-1} + \dots + a_n q^{-n}} \quad (8c)$$

$$v(t) = H(q)e(t) = \frac{1 + c_1 q^{-1} + \dots + c_n q^{-l}}{1 + d_1 q^{-1} + \dots + d_n q^{-l}} = \frac{C(q)}{D(q)} e(t) \quad (8d)$$

Upon applying these equations to the (SISO) Hammerstein model, the resulting structure is as follows:

$$y(t) = \frac{B(q)}{A(q)} \sum_{k=1}^m \beta_k u^k(t) + \frac{C(q)}{B(q)} e(t) \quad (9)$$

where:

- l denotes the linear component order in the Hammerstein model.
- q^{-l} is the delay operator symbolic representation and the complex variable z^{-1} .
- $H(q)$ represents the linear filter transfer function.
- $e(t)$ is a zero mean value and variance of λ^2 white noise sequence.
- m signifies the polynomials order.
- $[\beta_1, \beta_2, \dots, \beta_m, a_1, a_2, \dots, a_n, b_0, b_1, \dots, b_m, c_1, c_2, \dots, c_l, d_1, d_2, \dots, d_l]$ are the parameters to be estimated.

The process of identifying the Hammerstein model encompasses of four steps:

i. Test of The Design

To ensure the identifiability of the parameters, an identification test needs to be formulated using appropriate excitation signals. Ideally, the test should be tailored to yield an identified model well-suited for control applications. Let the data that has been obtained as input/output be represented as:

$$\{u(1), y(1), u(2), y(2), \dots, u(N), y(N)\}$$

where N denotes the samples number.

ii. Estimating the Parameters

Creating the model parameters involves utilizing input/output data collected during testing, accomplished through the minimization of the loss function.:

$$V_{PE} = \sum_{t=1}^N \varepsilon^2(t) \quad (10)$$

where

$$\varepsilon(t) = H^{-1}(q)[y(t) - G(q)f(u(t))] = \frac{D(q)}{C(q)} \left[y(t) - \frac{B(q)}{A(q)} \sum_{k=1}^m \beta_k u^k(t) \right] \quad (11)$$

iii. Selecting the Order

The values for l and m should be selected to achieve the highest accuracy for control purposes in the obtained model.

iv. Validating of The Model.

The identified model undergoes verification to assess its suitability for control; if not deemed suitable, another identification test is designed. The linear component upper error bound is utilized for this evaluation.

The Hammerstein model is determined through the utilization of actual data acquired from experimental tests conducted on the system. The data is entered into the system identification toolbox in MATLAB. [58]. The investigations involve regulating the air and water velocities to fall within predetermined ranges. The gas exhibits a velocity range of 0 to 18.288 m/s (0-60 ft/s), while the liquid demonstrates a velocity range of 0 to 3.048 m/s

(0-10 ft/s). The input is specified as the ratio of the velocities of the liquid and gas. Input signals and a comparison of pressure values between measured and projected data are illustrated in Figures 13 and 14. The parameters of the linear model are denoted by the equations that follow, in which F denotes the nonlinear function that involves the unit delay operator q^{-1} . A satisfactory fitting = 92.07% is attained between the measured and predicted signals. However, simulation errors lead to results that are noisy when there are sudden changes, in contrast to the actual measured values. The nonlinearity parameters are determined by employing a 1×1 array of Sigmoid Network consisting of 200 units in MATLAB, which is then transformed into a polynomial of order 3. Equations (8a) to (8e) delineate the linear component of the Hammerstein model. Parameters for output nonlinearity are set with a unit gain, consistent with the Hammerstein model. The system regards temperature as a disturbance.

$$w(t) = f(u(t))$$

$$y(t) = \frac{B(z)}{F(z)} w(t) + e(t)$$

$$B = -0.0943z^{-1} + z^{-2}$$

$$F = 1 - 1.8996z^{-1} + 1.3422z^{-2} - 0.4371z^{-3}$$

$\frac{B(z)}{F(z)}$ is the transfer function represents the linear component of the Hammerstien model

while w represents the nonlinear function.

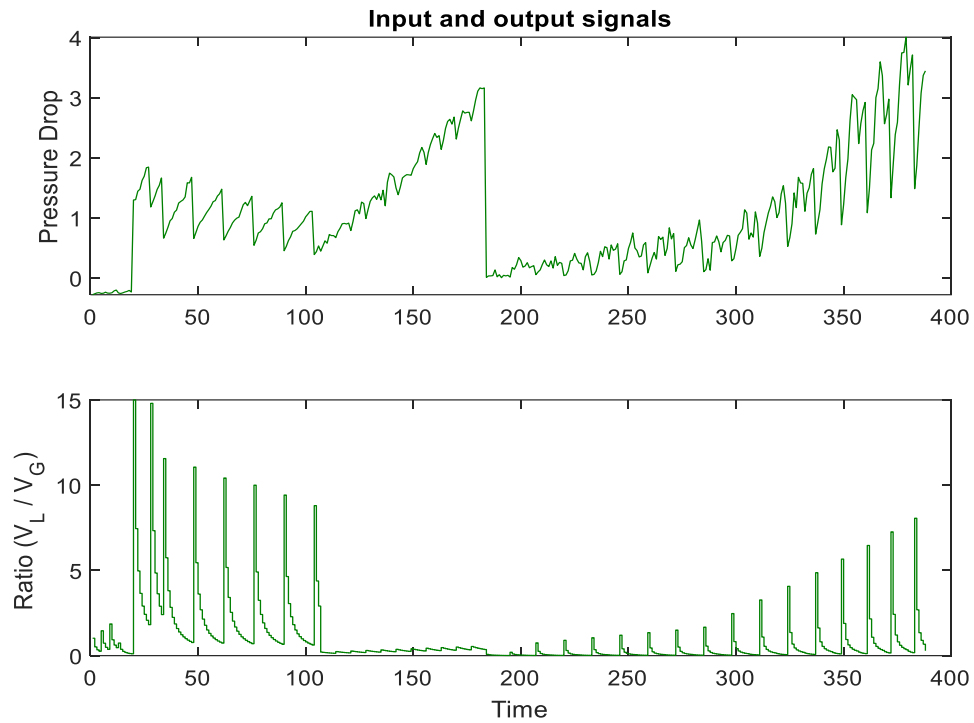


Figure 13 Output (the upper figure) and Input (the lower figure) Signals

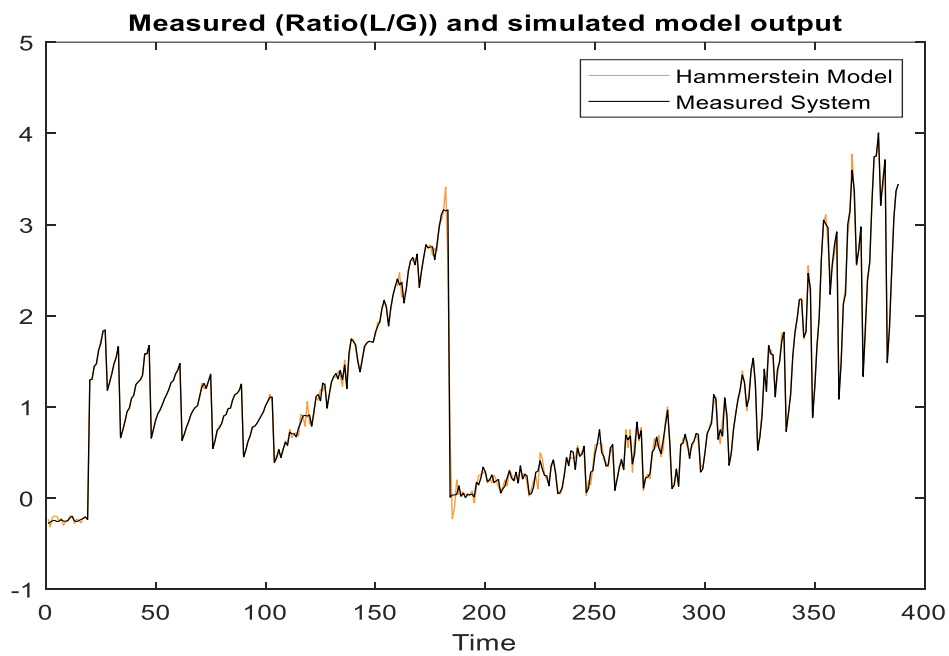


Figure 14 Measured (Pressure Drop) and Simulated Model Output

3.5. Summary

This section summarizes different identification techniques employed for system modeling, showcasing their associated structures and corresponding fitting percentages. The neural network-based model for MISO systems, featuring three hidden layers with a Rectified Linear Unit (ReLU) activation function, exhibits a high fitting percentage of 99.4%. In the nonlinear ARX category for MISO systems, two distinct structures are presented: one utilizing an Order 2 polynomial combined with Support Vector Machines (SVM) using Gaussian kernel achieving a fitting of 78.77%, and another employing an Order 2 polynomial with a 25-unit sigmoid network yielding an 86.52% fitting. The nonlinear Hammerstein models for MISO systems involve diverse structures, including Order 4 transfer function with a 10-unit sigmoid network (66.9% fitting) and Order 3 transfer function with a 15-unit sigmoid network (80.12% fitting). Lastly, a Nonlinear Hammerstein SISO model is specified, featuring an Order 3 transfer function with a 200-unit sigmoid network and achieving a fitting percentage of 92.07%. These models are designed for system identification, with fitting percentages serving as indicators of their effectiveness in capturing the underlying dynamics of the respective systems. The selection of a particular model and its structure is influenced by the unique characteristics and requirements of the system under consideration.

Identification Technique	System	Structure	% fitting
Neural Network-Based Model	MISO	3 hidden layers with ReLU activation function	99.4
Nonlinear ARX model	MISO	Order 2 polynomial + SVM	78.77
		Order 2 polynomial + 25 units sigmoid network	86.52
Nonlinear Hammerstein model	MISO	Order 4 transfer function + 10 units sigmoid network	66.9
		Order 3 transfer function + 15 units sigmoid network	80.12
Nonlinear Hammerstein SISO model	SISO	Order 3 transfer function + 200 units sigmoid network	92.07

CHAPTER 4 NONLINEAR CONTROL DESIGN

4. Nonlinear Controller Design

4.1. Introduction

Linear controllers, such as PID, are often employed to effectively regulate various nonlinear processes by tuning them in proximity to an operational point. However, these control algorithms typically neglect the inherent nonlinearities of the process, leading to suboptimal performance in specific instances [59]. Enhanced performance for strongly nonlinear processes can be achieved through the utilization of nonlinear PID controllers. These controllers leverage a nonlinear function to capture the intricacies of the process, drawing upon a priori knowledge. Despite attempts to represent nonlinearities accurately using cubic splines with knots, the inherent nonlinearity of the functions cannot be precisely captured, resulting in a nonlinearity mismatch error. The extent of this error is contingent upon the number and placement of the knots.

4.2. Adaptive PID Control Using Neural Network

4.2.1. Introduction

The PID control strategy stands out as a sophisticated and widely employed method in various industries. A common approach to derive a discrete-time PID controller involves obtaining the time-derivative and discretizing the resultant equation., This process yields a representative expression for a typical discrete-time PID controller:

$$\begin{aligned}
u(k) = & u(k-1) + K_p (e(k) - e(k-1)) \\
& + \frac{K_I T}{2} (e(k) + e(k-1)) + \frac{K_D}{T} (e(k) - 2e(k-1) \\
& + e(k-2))
\end{aligned} \tag{12}$$

Here $u(k)$ is the control effort at time k . K_p, K_I and K_D are proportional, integral and derivative gains in a discrete-time PID controller, respectively along with a specified sampling period T . $e(k)$ represents the tracking error at time k defined as $e(k) = y_d(k) - y(k)$; $y_d(k)$ signifies the desired plant output, $y(k)$ represents the plant output.

$$\Delta u(k) = u(k) - u(k-1) = K_p e_p(k) + K_I e_I(k) + K_D e_D(k) \tag{13}$$

where

$$e_p(k) = e(k) - e(k-1)$$

$$e_I(k) = \frac{T}{2} (e(k) + e(k-1)) \tag{14}$$

$$e_D(k) = \frac{1}{T} (e(k) - 2e(k-1) + e(k-2))$$

4.2.2. Architecture of the Control System

Figure 17 depicts the arrangement of the newly implemented control algorithm, which combines neural networks with a PID controller. In this configuration, the PID controller is cascaded with the model that represents the plant. The error signal, obtained by subtracting the desired plant output from the plant model, is exclusively utilized for

adjusting the PID settings. In this inventive adaptive control approach, modifications to the model plant weights are executed offline, and afterward, the fixed weights are exclusively employed for real-time adjustments of the PID controller gains. This control algorithm, as shown, features a simple structure and provides a computational time benefit.

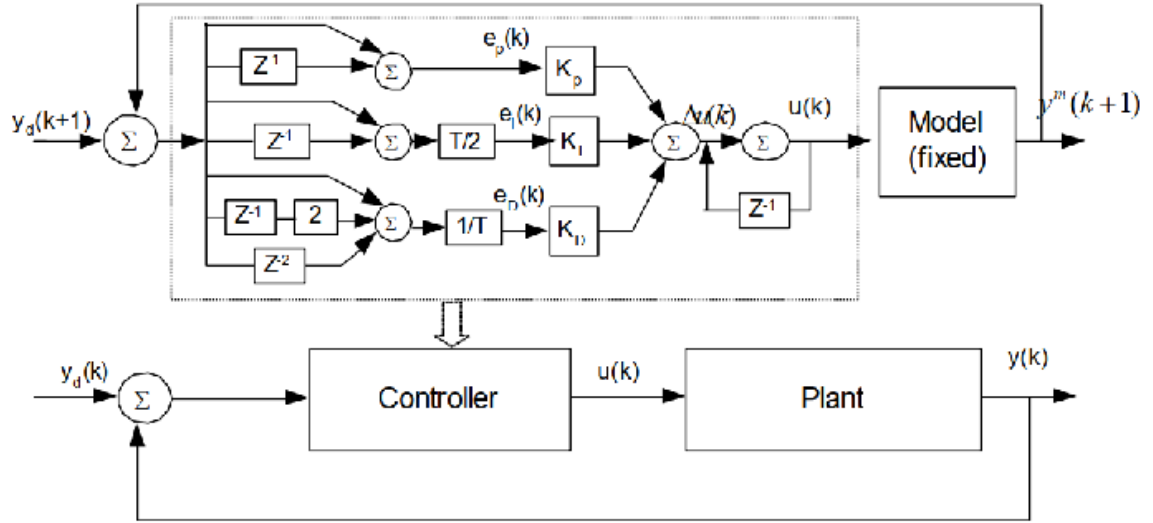


Figure 15 Adaptive PID Control using Neural Network Architecture

In order to acquire the neural network model for the plant, a feedforward neural network is employed to capture the system dynamics, and the weights are trained using the back-propagation algorithm. Figure 18 displays the block diagram illustrating the identification system.

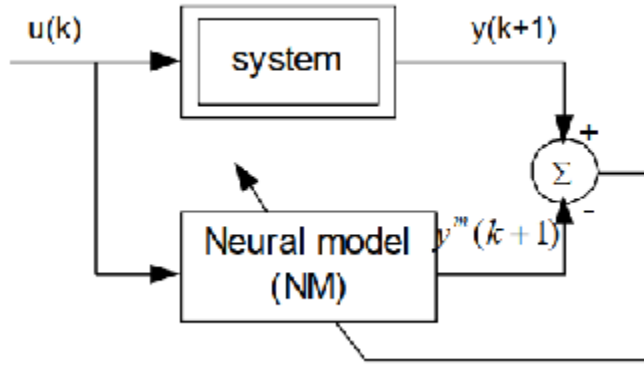


Figure 16 The Online NN Model Training

The weights of the neural model are modified in order to minimize the cost function specified by:

$$E = \frac{1}{2} (y(k+1) - y^m(k+1))^2 \quad (15)$$

The scalar output of the system is denoted as $y(k)$, whereas the output of the neural model is represented as $y^m(k)$.

4.2.3. Adaptation Process of PID Controller

In every control cycle, modifications are implemented according to the cost function and the Neural Model (NM) featuring fixed weights. To accelerate the learning algorithm's convergence and guarantee stability in the closed-loop system, stability properties are considered in updating the weight. This includes integrating the influence of past iterations through momentum, the magnitude of which depends on the stability properties of the closed-loop system. Notably, the impact of momentum becomes pronounced when the

desired stability response is not satisfactorily attained, as indicated by a small margin of stability.

The update for tuning the weights of the gains, employing a descent method, is expressed as:

$$\Delta k_{PID}(t) = -\alpha \frac{\partial E(t)}{\partial k_{PID}} + \beta(t) \Delta k_{PID}(t-1) \quad (16)$$

where α denotes the learning rate and β is the momentum rate, adjusted according to the following equation:

$$\beta(t) = \beta_0 \exp(-b_{p,c})$$

where β_0 is its initial value and $b_{p,c}$ is the exponential decay factor.

The learning rate α should be bounded ($0 < \alpha < 0.21$) to insure the system stability and the introduced method for adaptive momentum tuning aims to expedite the adjustment of controller parameters when the controller's robustness performance is insufficient. Conversely, it slows down the adaptation when the controller exhibits satisfactory robustness performance.

The derivatives in Equation 16 are calculated as:

$$\frac{\partial E(t)}{\partial k_p} = \frac{\partial E(t)}{\partial y^m} \frac{\partial y^m}{\partial u(t)} \frac{\partial u(t)}{\partial K_p} = -(y - y^m) \frac{\partial y^m}{\partial u(t)} e_p(t) \quad (17)$$

$$\frac{\partial E(t)}{\partial k_I} = \frac{\partial E(t)}{\partial y^m} \frac{\partial y^m}{\partial u(t)} \frac{\partial u(t)}{\partial K_I} = -(y - y^m) \frac{\partial y^m}{\partial u(t)} e_I(t)$$

$$\frac{\partial E(t)}{\partial k_D} = \frac{\partial E(t)}{\partial y^m} \frac{\partial y^m}{\partial u(t)} \frac{\partial u(t)}{\partial K_D} = -(y - y^m) \frac{\partial y^m}{\partial u(t)} e_D(t)$$

Given that the functions for both hidden and output neurons were characterized by the logistic sigmoid function

$$f(x) = \frac{1}{1 + \exp(-x)}$$

Then

$$\frac{\partial y^m}{\partial u(t)} = y^m(1 - y^m) \sum_j W_j^m O_j^m (1 - O_j^m) W_{1j}^m \quad (18)$$

O_j^m indicates the output of the j -th neuron in the hidden layer of the Neural Model (NM), whereas W_{1j}^m denotes the weights of the NM from the input neurons to the intermediate layer and W_j^m denotes the weights from the intermediate layer to the output.

4.2.4. Results of the Adaptive PID Control

MATLAB is used to implements a dynamic control system for a nonlinear plant using a combination of PID control and neural network-based compensation. State variables are updated based on nonlinear differential equations. The PID controller is employed to calculate control signals, with gains dynamically adjusted over time. The neural network

is utilized to approximate the unfamiliar dynamics of the system, allowing the controller to adjust to varying circumstances.

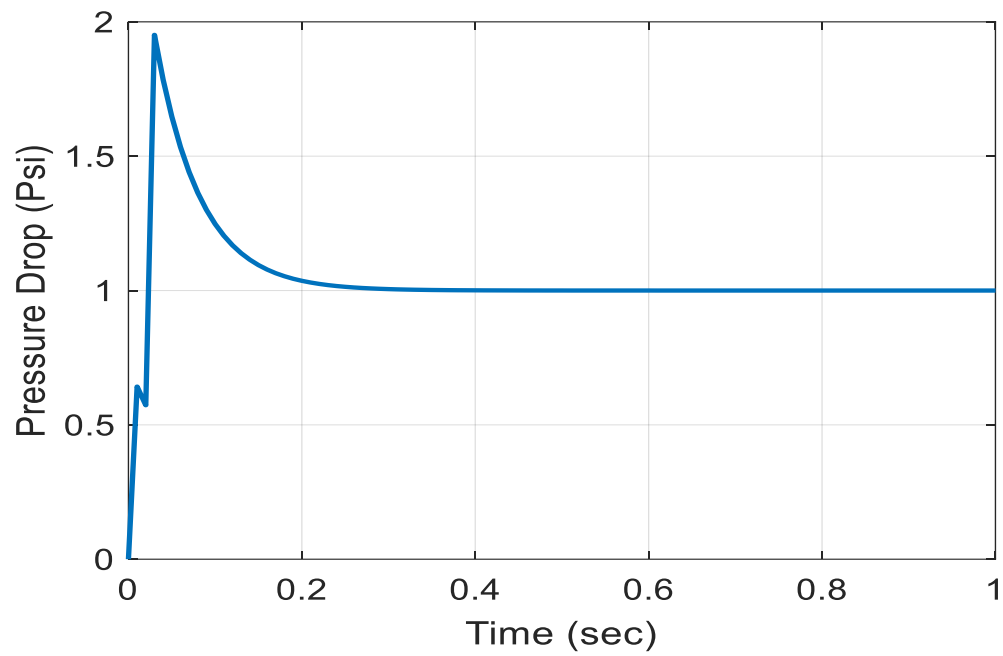


Figure 17 Step Response of Adaptive PID

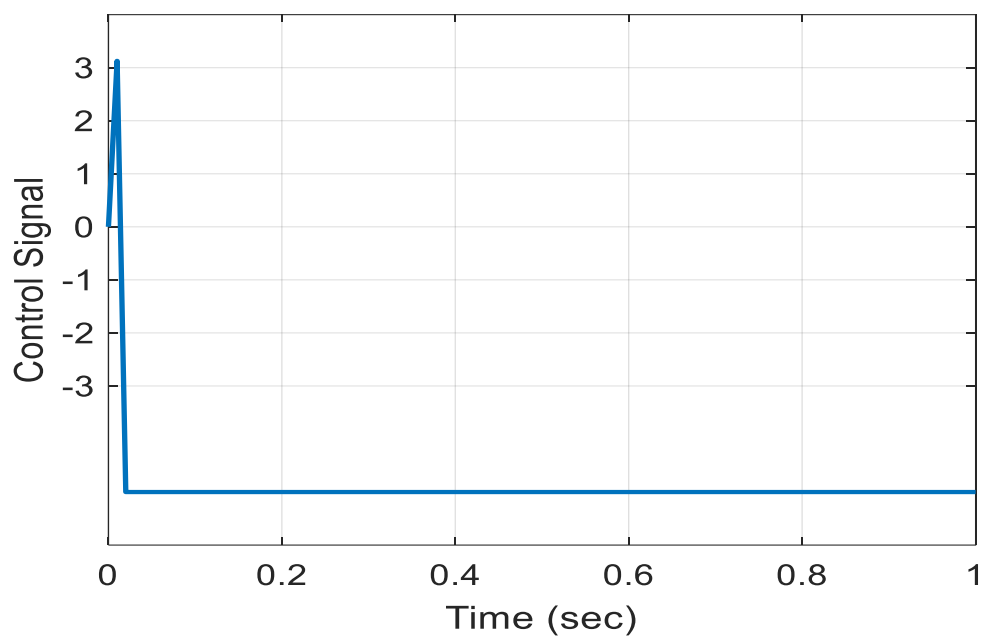


Figure 18 The Control Signal of Adaptive PID

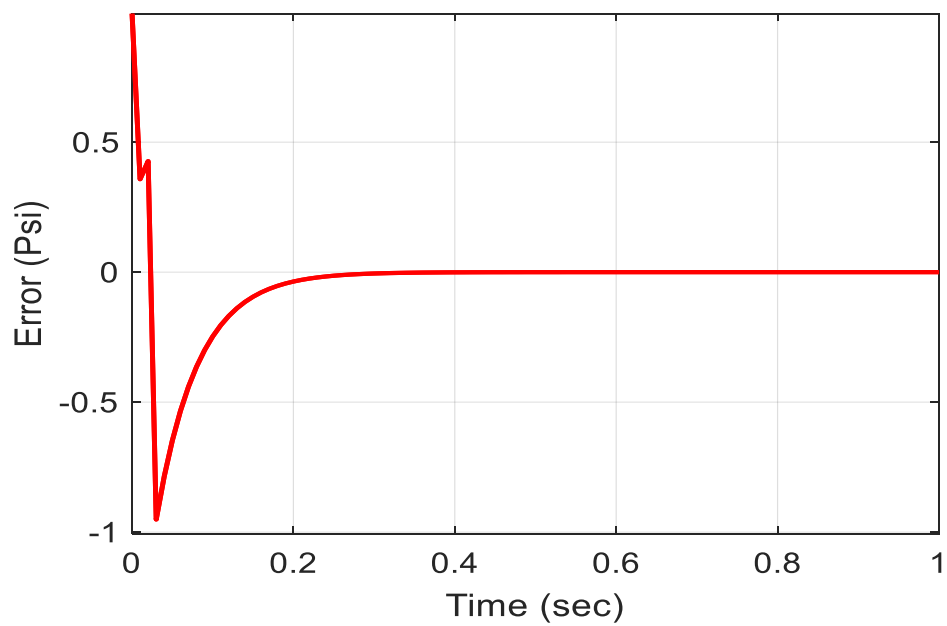


Figure 19 The error Signal for Step Response

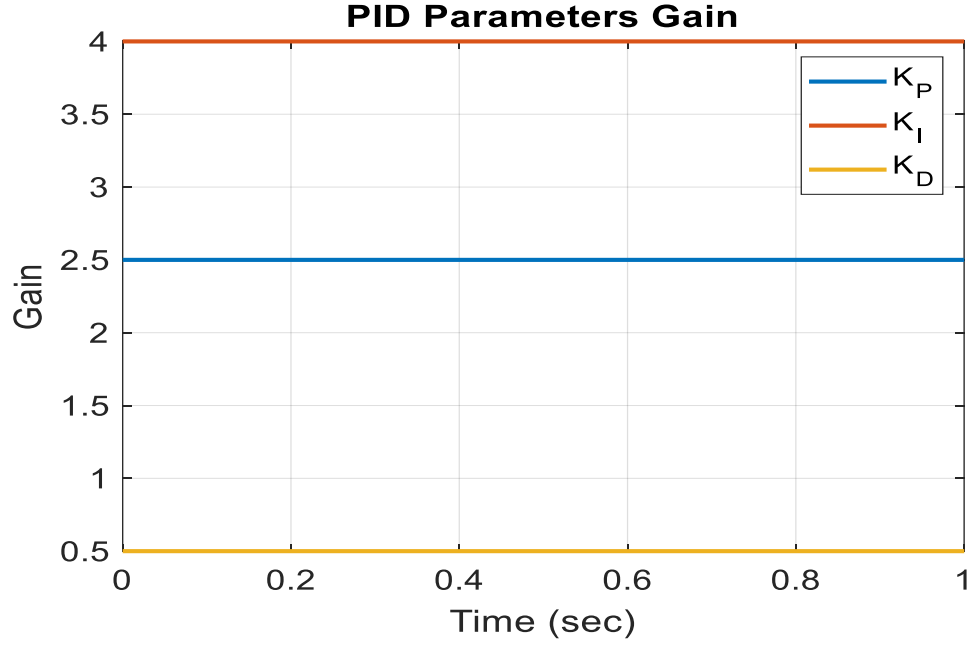


Figure 20 PID Parameters Gain

Notably, the response shows overshoot before settling in steady-state value, this due to the minimum phase in the system. The incorporation of constraints on control signals ensures stability by preventing them from exceeding predefined limits. Overall, this technique demonstrates a comprehensive approach to dynamic control and adaptation in a nonlinear system, combining classical PID control with the flexibility of a neural network-based compensator.

4.3. Sliding Mode Controller for The Hammerstein Model

4.3.1. Introduction

The SMC methodology is recognized as a reliable control strategy that is particularly effective in managing complex systems with higher-order nonlinearity, which are vulnerable to uncertainty in parameters and external disturbances [60]. The notion of

sliding mode operation was initially developed in the 1950s in the former Soviet Union as part of the variable structure systems framework [61], [62]. In its early stages of development, the variable structure system did not receive much attention in the field of control, mostly because of its inherent complexity. The intricacy was most apparent when dealing with the difficulties related to chattering, which encompassed actuators, sensors, and switching systems.

Nevertheless, there are notable advantages associated with this approach. By selecting an appropriate switching function, the system's dynamic behavior is altered, resulting in a closed-loop response that is unaffected by uncertainties that are compatible with the system [60]. The intriguing attribute has captured the attention of academics, motivating their contributions to the advancement of Sliding Mode Control (SMC).

Recently, advancements in design techniques for Sliding Mode Control (SMC) have successfully tackled significant challenges, including reducing chattering, adjusting for unmeasured dynamics, adapting to noisy systems, and improving the dynamic performance of closed-loop systems. An advantageous aspect of choosing an SMC controller instead of linear control methods such as PID is its capacity to guarantee stability and resilient performance in diverse unpredictable systems, whereas PID may encounter difficulties in maintaining stability in uncertain situations [63].

The sliding surface is the boundary at which the state trajectories of the system intersect. A control input is implemented at this juncture in order to preserve the state of the system on the sliding surface. This operation mandates that the system conform to a pre-established trajectory [64].

The sliding surface is often characterized mathematically as:

$$S(x) = 0$$

Where:

- $S(x)$ is the Controller's sliding surface.
- x represents the system's state.

The selection of the sliding surface function is determined by the system's dynamics and the control objectives. It is critical to choose the appropriate sliding surface function so as to achieve the desired control performance.

The sliding surface is calculated as:

$$S(k) = C_1 e(k) + e(k + 1) \quad (19)$$

The sliding surface is defined by this equation. C_1 is a sliding mode gain which derived from the system matrices. the sliding mode control gain λ . In this particular scenario, the value of $\lambda = 10$. The design of the sliding surface function must ensure that the system states are driven towards the intended reference or equilibrium point in a regulated manner.

The control signal is engineered to modify the state or output of the system with the purpose of attaining a desired performance or fulfilling certain control objectives.

$$u = -L_1' \cdot L_2 \quad (20)$$

$$L_1 = (A - \lambda BC)B^{-1} \quad (21)$$

$$L_2 = x - Sy \quad (22)$$

u denotes the control signal. L_1 defines the control law, while L_2 is a component of the sliding mode control. The value of u is determined by taking the negative of the inner product between L_1' (the transpose of L_1) and L_2 . The objective of this control signal is to steer the system's state or output towards the designated reference or equilibrium point, as shown by the sliding surface.

4.3.2. Sliding Mode Control Results

The control signal is transmitted to the actuator of the system, and its impact on the system is contingent upon the unique dynamics and characteristics of the system. The objective is to use the control signal to regulate the system and achieve the desired control performance, such as tracking a reference trajectory, maintaining stability, or rejecting disturbances.

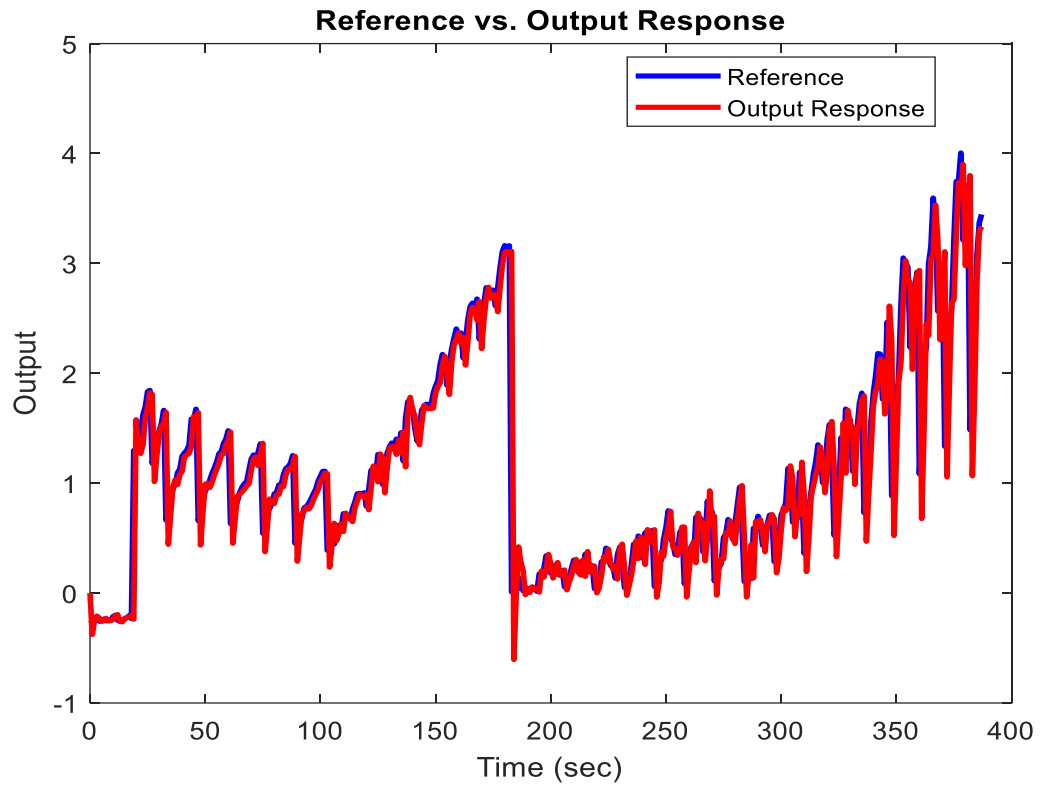


Figure 21 Output vs. Estimated Output Using SMC

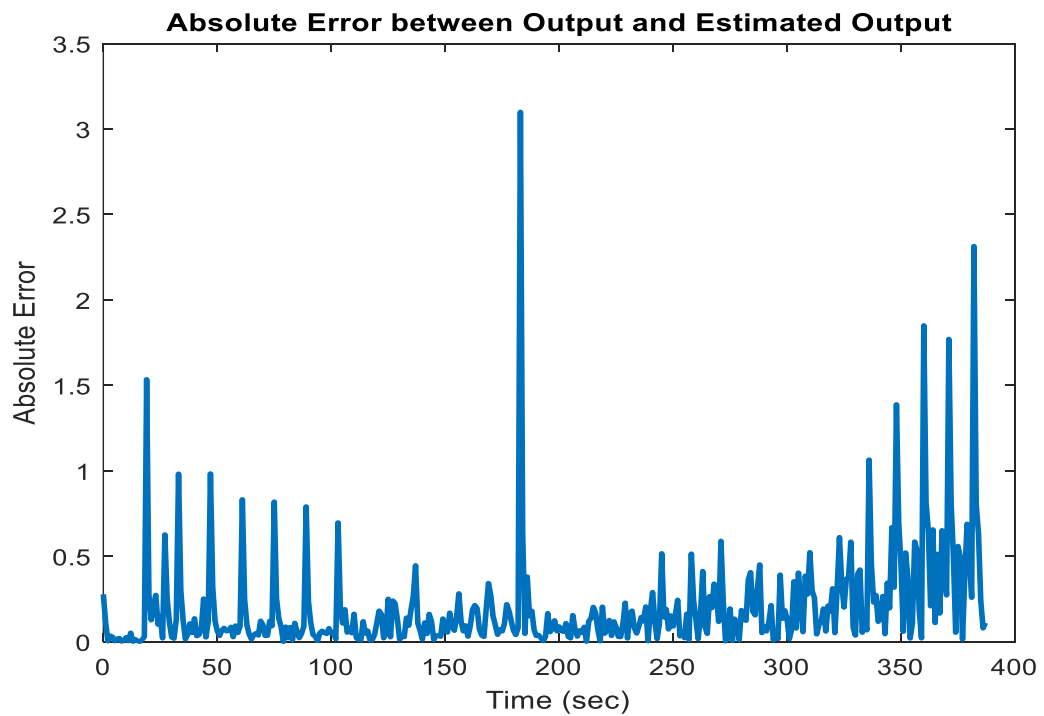


Figure 22 Absolute Error between Output and Estimated Output

The control strategy can guide the system to the intended sliding mode and sustain it, even when disturbances are present. Employing the reaching law and sliding mode control law, a sliding mode controller has been developed for the Hammerstein model. Through MATLAB simulations, the control law was assessed, revealing that the system converges to the sliding mode and persists in that state throughout the simulation period.

Furthermore, this robust control approach demonstrates its effectiveness in handling disturbances and maintaining the desired sliding mode, showcasing its applicability in real-world scenarios. The MATLAB simulations validate the controller's ability to consistently achieve and uphold the designated sliding mode, highlighting its stability and reliability in dynamic systems

CHAPTER 5 LINEARIZED MODEL CONTROL DESIGN

5. Linearized Model Control Design

5.1. Introduction

In the area of control design, addressing various forms of nonlinearities in the system has long presented challenging issues. Depending on the nature of the nonlinearity, diverse approaches in control design can be applied to mitigate its impact on the control system. If the system exhibits only mild nonlinearity, approximating it with a linear system may suffice, allowing the application of standard linear theory when developing a control law [65]. However, in the presence of more pronounced nonlinearities, it is advisable to explicitly account for the nonlinearity during the control design process. A commonly utilized approach when dealing with dynamic nonlinearities is the method of exact linearization through internal feedback. This technique facilitates the linearization of the input-output relationships of a nonlinear system and has undergone extensive research.

In the domain of the process industry, a prevalent type of nonlinearity arises when a nonlinearity influences the input signal to a process. This situation often occurs due to nonlinear actuators, such as nonlinear valves. It is frequently reasonable to neglect the dynamics of the nonlinear actuator compared to the overall process, indicating that the nonlinearity is predominantly static. A prevalent strategy for representing the input-output relationship in such situations is to employ a static input nonlinearity alongside a linear system. This model is known as a Hammerstein model. To provide concrete instances of Hammerstein models, consider the following examples [66]. One commonly employed

approach to handle systems that exhibit static nonlinearity is to utilize a model that represents the antithesis of the static nonlinearity in order to mitigate its impacts. In the subsequent discourse, this approach shall be denoted as the conventional method. Alternatively, one can opt for a gain scheduling technique. A detailed examination of these approaches can be found in sources such as [67]. Further applications are detailed in [68], [69]. Another method to tackle the issue revolves around implicitly approximating the inverse using high-gain feedback, as discussed in [70]. However, a significant drawback of this approach is that the sensitivity to noise increases due to the application of high-gain feedback

5.2. Tangent Linearization

Within this section, we undertake the process of linearizing the Hammerstein model, focusing on a certain operating point that is determined by the input U_0 and the state values X_0 . It is important to emphasize that X_0 does not necessarily include values of the equilibrium state. The resulting output yields a linear model that serves as the most favorable linear approximation for inputs that display minor fluctuations around a constant input $u(t) = U$. The fundamental principle underlying the linearization technique is tangent linearization. By integrating a linear system with a static nonlinear system, the model structure visually represents a nonlinear system. By connecting a linear system in series with a static non-linear system, this configuration enables the representation of a non-linear system. For example, a fixed nonlinearity can be used to replicate phenomena such as saturation or dead-zone impacts. The diagram illustrates a nonlinear system that has been transformed into a modified linear system due to the presence of static input nonlinearity.

The function f represents the input nonlinearity, while $[A, B, C, D]$ represents the state-space parameters of the linear model.

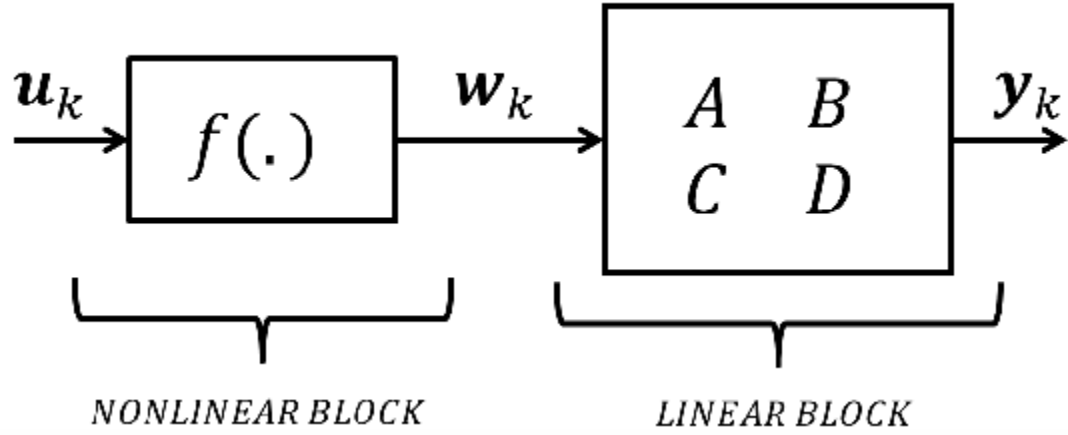


Figure 23 Hammerstein Model as a Static Nonlinear Input and Linear State Space System

The model's behavior is determined by the following set of equations.

$$w_k = f(u_k) \quad (23)$$

$$x_{k+1} = Ax_k + Bw_k \quad (24)$$

$$y = Cx_k + Dw_k \quad (25)$$

In these equations:

- u represents the input signal,
- w signifies the intermediate signal (output of the input nonlinearity),

- y denotes the output of the model.

The linearized representation of the Hammerstein model at a specific operating point (X^*, u^*) is expressed as follows:

$$\begin{aligned}\Delta X(k+1) &= A\Delta x(k) + Bf_u\Delta u(k) \\ \Delta y(k) &\approx C\Delta x(k) + Df_u\Delta u(k)\end{aligned}\tag{26}$$

where

- $\Delta X(k) = X(k) - X^*(k)$
- $\Delta u(k) = u(k) - u^*(k)$
- $\Delta y(k) = y(k) - y^*(k)$
- $f_u = \left. \frac{\partial}{\partial u} f(u) \right|_{u=u^*}$

In this context, y^* represents the model output corresponding to the input u^* and state x^* , where $w^* = f(u^*)$.

After linearization, the state space matrices of the system becomes:

$$\begin{aligned}A &= \begin{bmatrix} 1.8996 & -1.3422 & 0.4371 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, & B &= \begin{bmatrix} 0.7902 \\ 0 \\ 0 \end{bmatrix} \\ C &= [0.8209 \quad 0.1265 \quad -0.0412], & D &= [-0.0745]\end{aligned}$$

Sample time: 1 second.

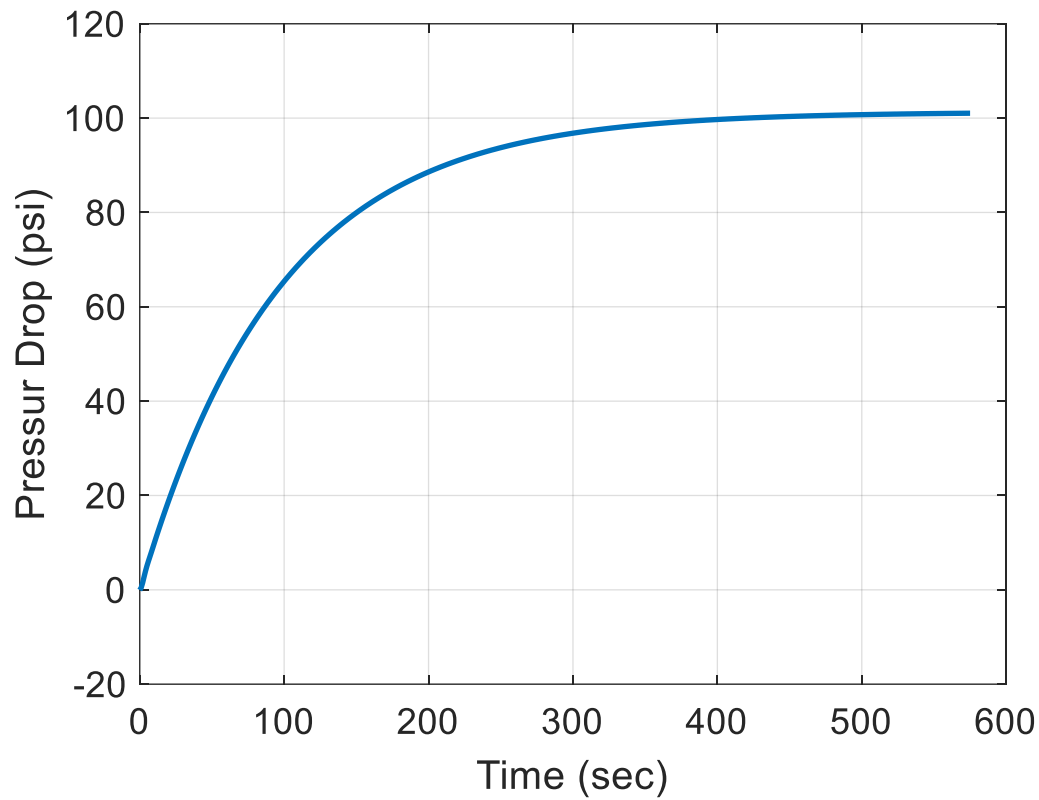


Figure 24 Step Response of the Linearized Model (Open Loop)

5.3. Discrete PID with Linearized Model

The implementation of a Proportional-Integral-Derivative (PID) controller in the context of a linearized Hammerstein model represents a promising approach to address the control challenges associated with nonlinear systems. The Hammerstein model, characterized by a static nonlinearity followed by a linear dynamic system, is a common representation for a wide range of industrial processes. When dealing with such systems, it is essential to

develop control strategies that can mitigate the nonlinearities while maintaining system stability and performance.

In a recent research study, a PID controller was implemented to regulate a linearized Hammerstein model that closely approximated the behavior of a practical nonlinear process. The results of this investigation demonstrated the PID controller's ability to effectively manage the system's behavior. The proportional component of the PID controller enabled the system to respond promptly to setpoint changes, ensuring that the temperature was regulated within the desired range. The integral term played a critical role in eliminating any steady-state errors that could result from the nonlinear behavior. In the context of the multi-phase flow system, this helped maintain the pressure drop at the setpoint value over extended operating periods. The derivative term of the controller aided in damping oscillations, preventing overshoot, and ensuring a smooth transition during temperature changes.

However, it is essential to consider certain aspects in the discussion of these results. The performance of the PID controller is inherently linked to the accuracy of the linearization process. Accurate linearization requires careful selection of the operating conditions and the choice of the linearization point. Variations in operating conditions, nonlinearities outside the chosen linearization range, or significant changes in process parameters can potentially impact the controller's performance. Therefore, to successfully apply a PID controller to a linearized Hammerstein model, engineers and researchers must have a deep understanding of the underlying system dynamics and should carry out diligent parameter tuning for optimal control.

Utilizing a PID controller with a linearized Hammerstein model is an effective approach for managing nonlinear systems. However, it should be noted that the success of this approach hinges on the accuracy of the linearization process and the careful consideration of various factors such as operating conditions and system dynamics. By employing the PID controller in such scenarios, engineers can attain efficient and robust control, contributing to improved process performance and safety in industrial applications.

The genetic algorithm (GA) is employed to optimize the settings of the linear PID controller. The algorithm's objective is to minimize a fitness function, namely the Mean Square Error, by utilizing the provided linearized system. The genetic algorithm configuration entails specifying variables, constraints, and settings. Once the process is completed, the refined PID parameters are utilized to create a transfer function for the PID controller. The closed-loop system, which includes the PID controller, is defined.

$$K_p = 0.3735$$

$$K_I = 0.0438 \tag{27}$$

$$K_D = 0$$

The linear PID controller is simulated with MATLAB. A plot of the step response of magnitude 1 is applied to the linear PID controller structure, providing a visual representation of the optimized system's performance. The response can be shown as follows.

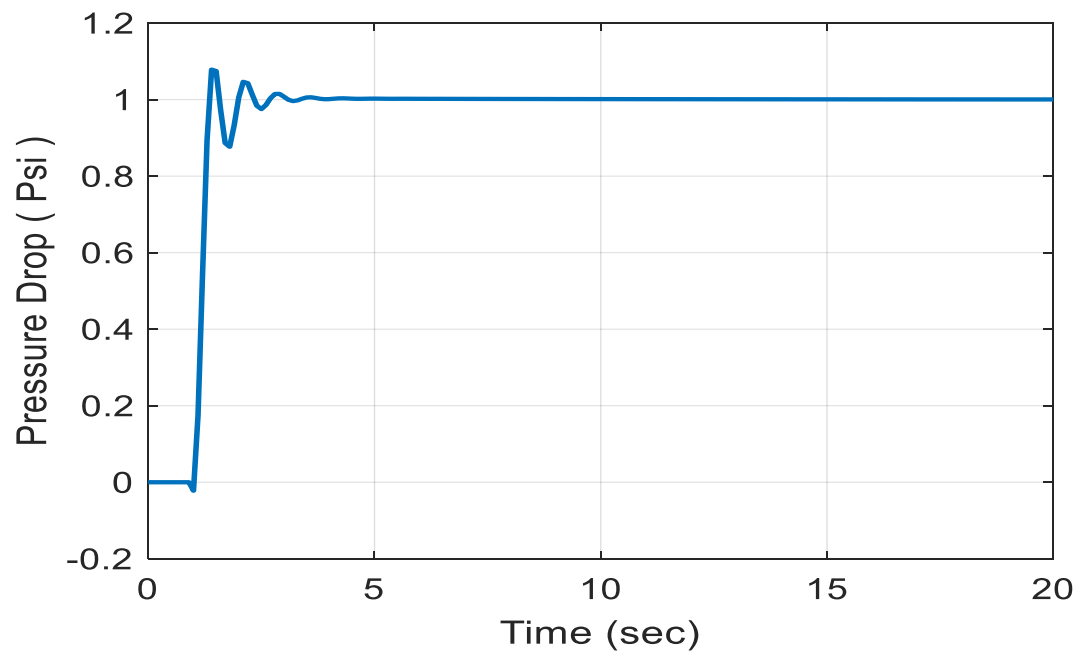


Figure 25 Closed-Loop Step Response of The Linearized Model with PID Controller

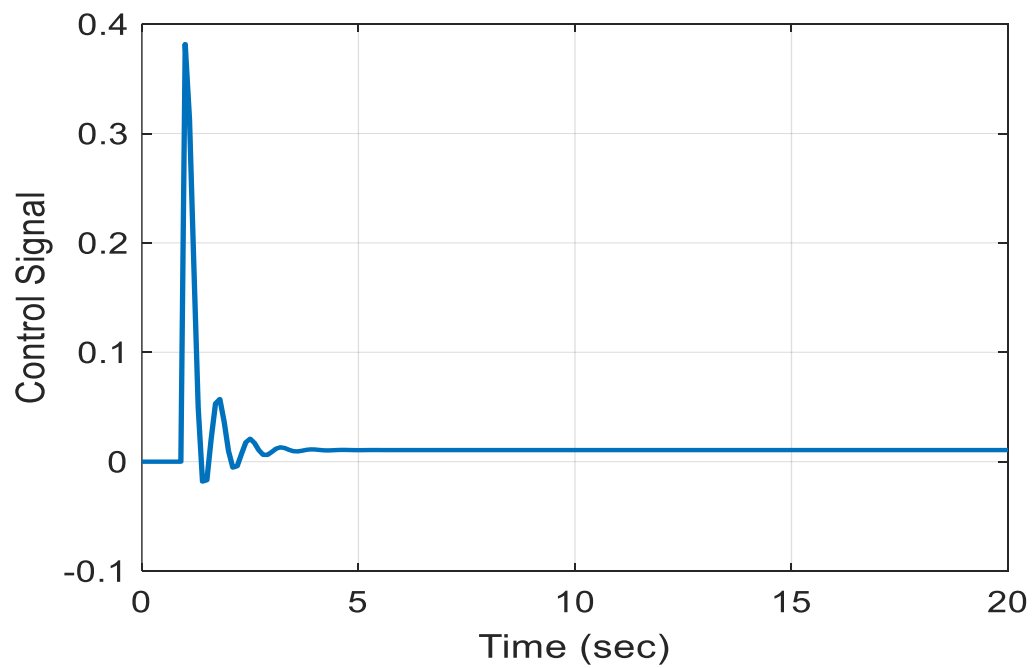


Figure 26 The Control Signal u of PID

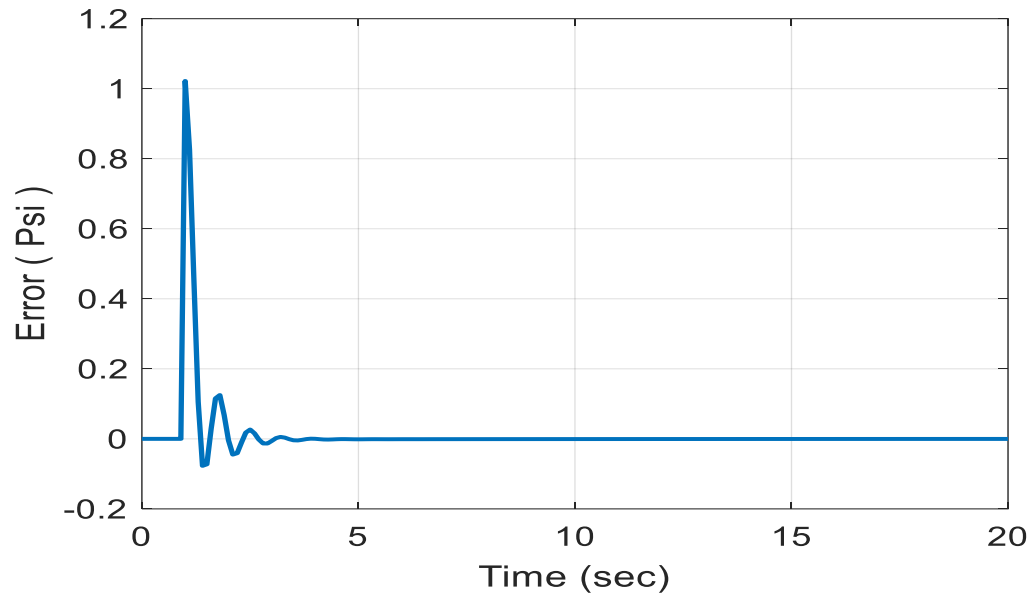


Figure 27 The Error Signal e Between the Step Response and The Linearized Response with PID

In the figures below, we test the response with the actual data.

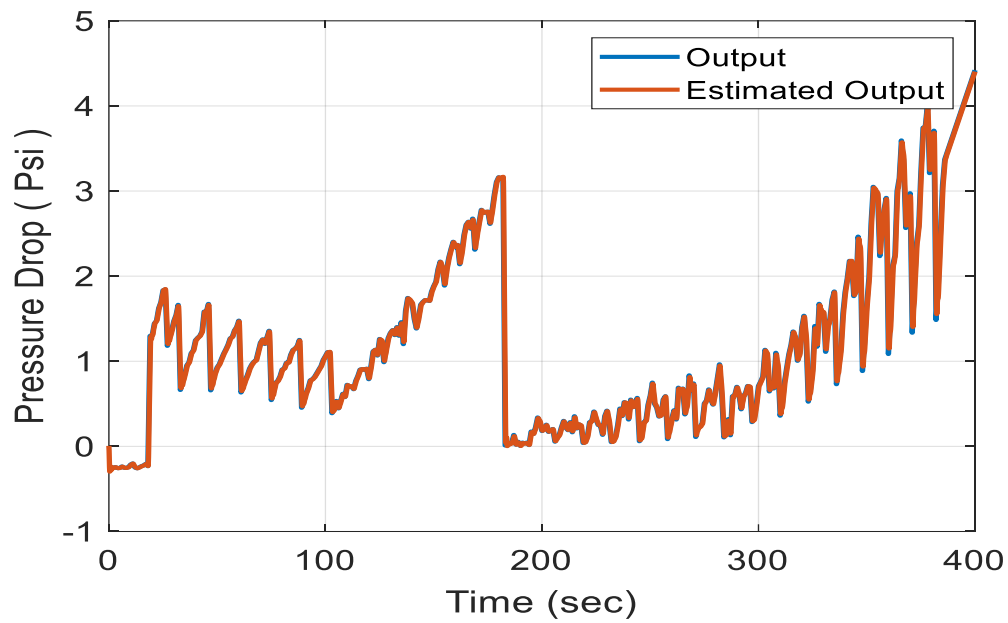


Figure 28 Closed-Loop Output Response vs Estimated Output Response of The Linearized Model with PID Controller

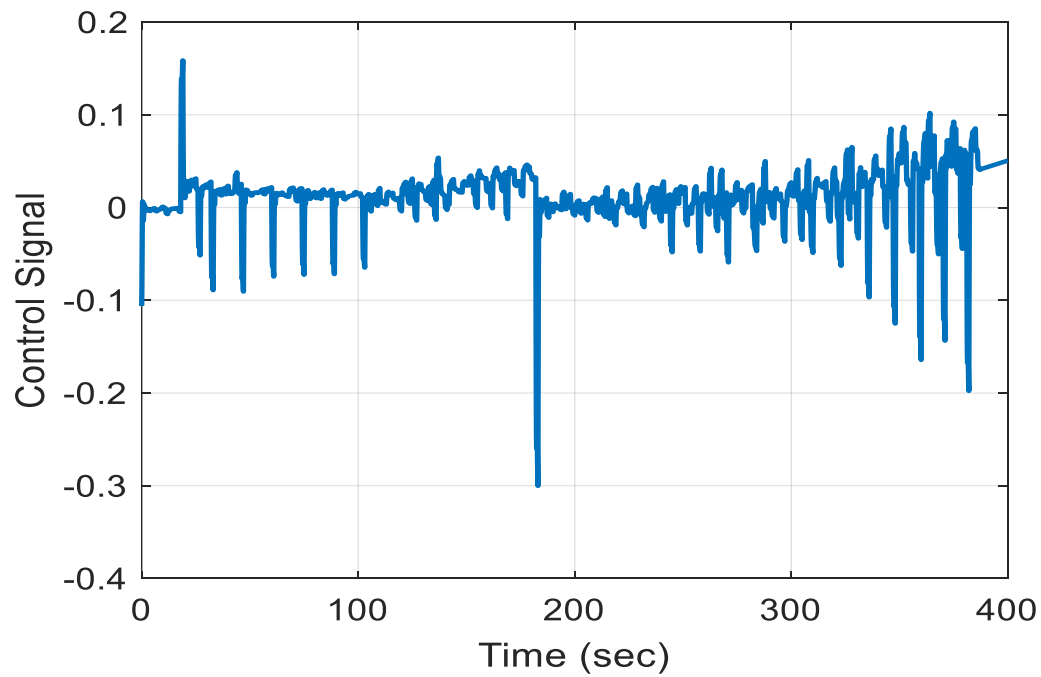


Figure 29 The Control Signal u of PID

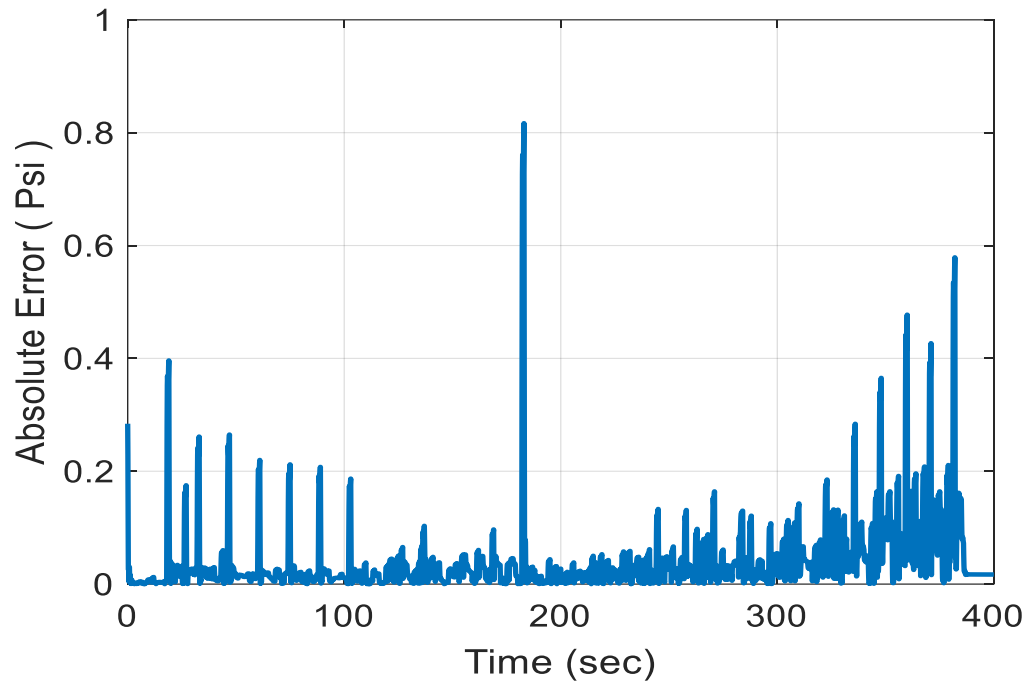


Figure 30 The Absolute Error Signal e (Output- Estimated Output Responses) with PID

The computer hosts the simulated PID controller, which is tested through experiments involving the pressure drop process. Step inputs are employed to assess the PID controller's effectiveness at various gains. The experiments utilize a consistent linearized model for result comparison. Robustness evaluation of the linear PID controller is carried out, with an experimental analysis involving varied closed-loop time constant values. Notably, an increase in the closed-loop time constant results in a gradual slowing down of the controllers.

To illustrate the performance of the proposed PID with linearized model method, figures above show experimental results with PID controller. PID controller has been tuned such that a fast-rising time, a fast-settling time and small overshoot are obtained.

5.4. Linear Quadratic Regulator

In control theory, the Linear Quadratic Regulator (LQR) is employed for the analysis of systems represented in state-space. The utilization of the state-space approach proves particularly straightforward, especially in the context of (SISO) systems. The LQR controller is instrumental in deriving the optimal control sequence, aiming to minimize a quadratic cost function expressed as:

$$J = \int_0^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \quad (28)$$

The control energy is represented by $u(t)^T R u(t)$ while the transient energy is expressed as $x(t)^T Q x(t)$. Where R and Q are a positive definite matrix, and weighting matrix, respectively. Q may take on positive semi-definite values. The control signal $u(t)$ signifies the optimal control, governing the system signal. [71], which is:

$$u(t) = -Kx(t) = -R^{-1}B^T P x(t) \quad (29)$$

K denotes the linear optimal feedback matrix, while the solution to the Riccati equation can be found by P , which can be determined through:

$$PA + A^T - PBR^{-1}P + Q = 0 \quad (30)$$

The weighting matrices Q and R are determined as follows:

$$Q = 0.270, \quad R = 0.265$$

Once the optimal values for the weighting matrices Q and R are determined offline, the controller gain matrix can be computed using the LQR MATLAB function:

$$K = dlqr[A, B, Q, R]$$

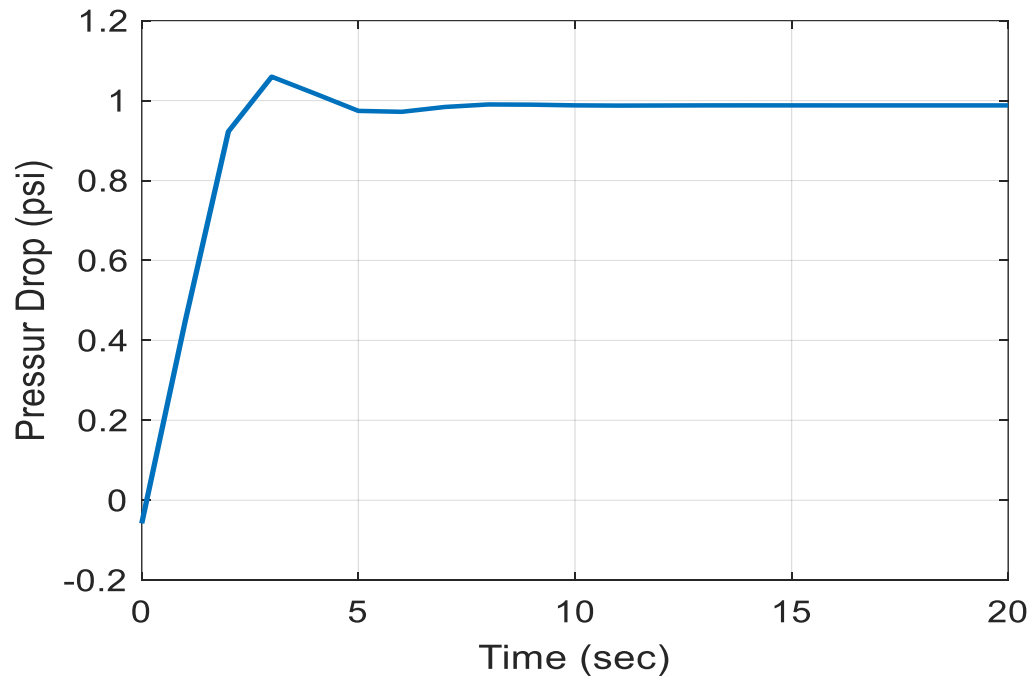


Figure 31 Closed-loop Step Response With LQR

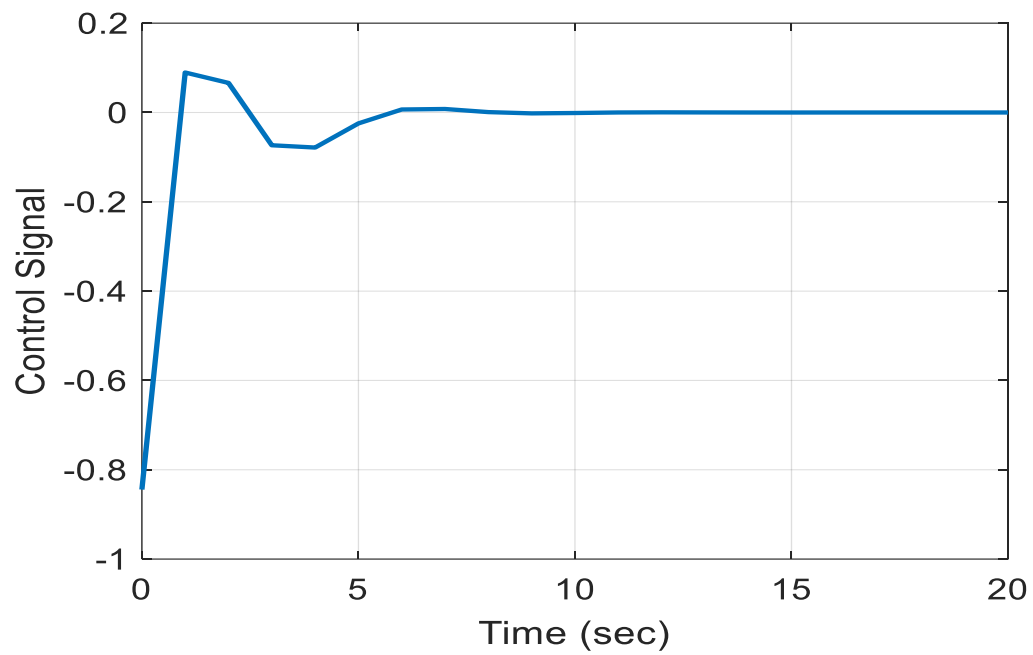


Figure 32 The Control Signal u of LQR

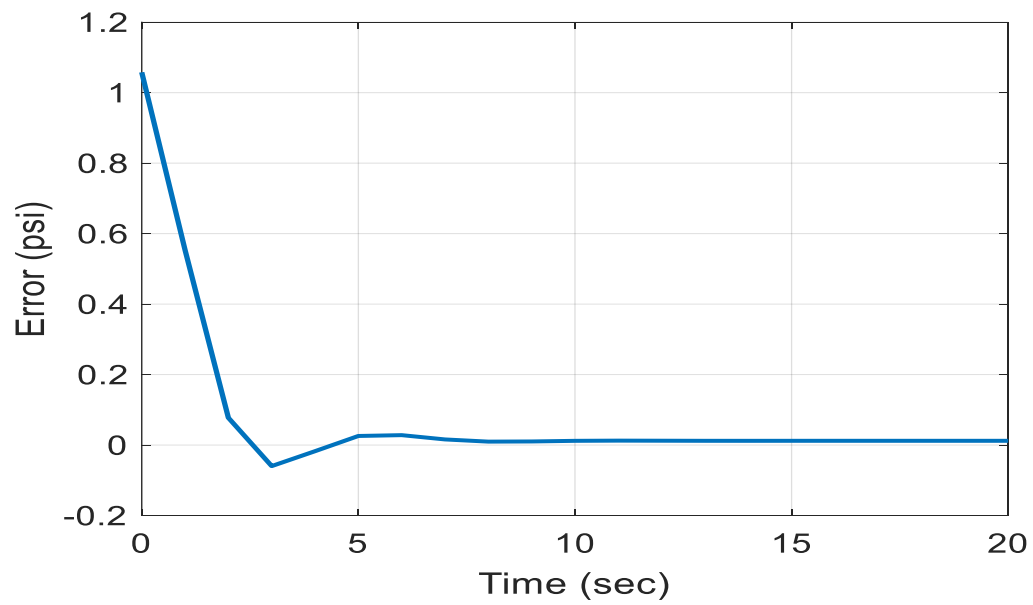


Figure 33 The error signal e of Linearized response with LQR

Now, the response is tested by the actual data collected from the experiment.

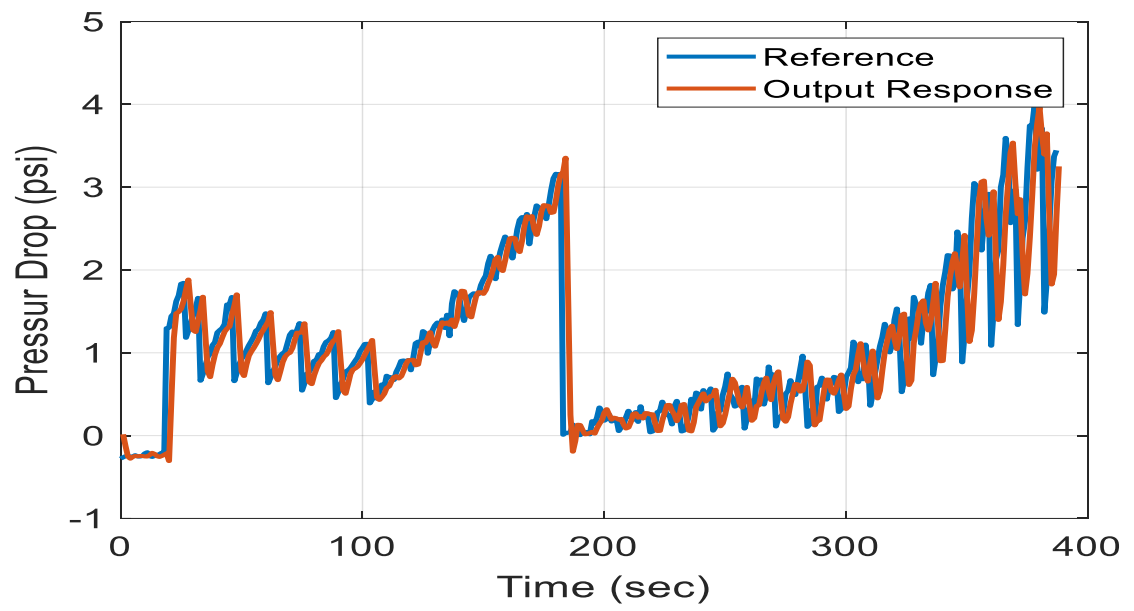


Figure 34 Output Response vs Estimated Output Response of The Linearized Model with LQR

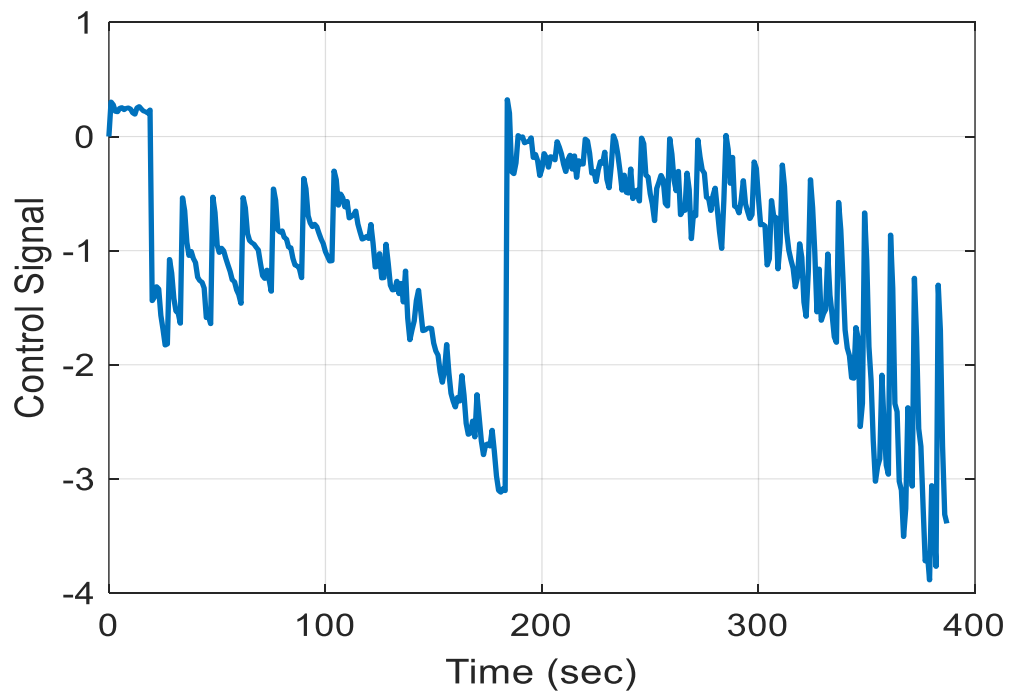


Figure 35 The Control Signal u of SMC

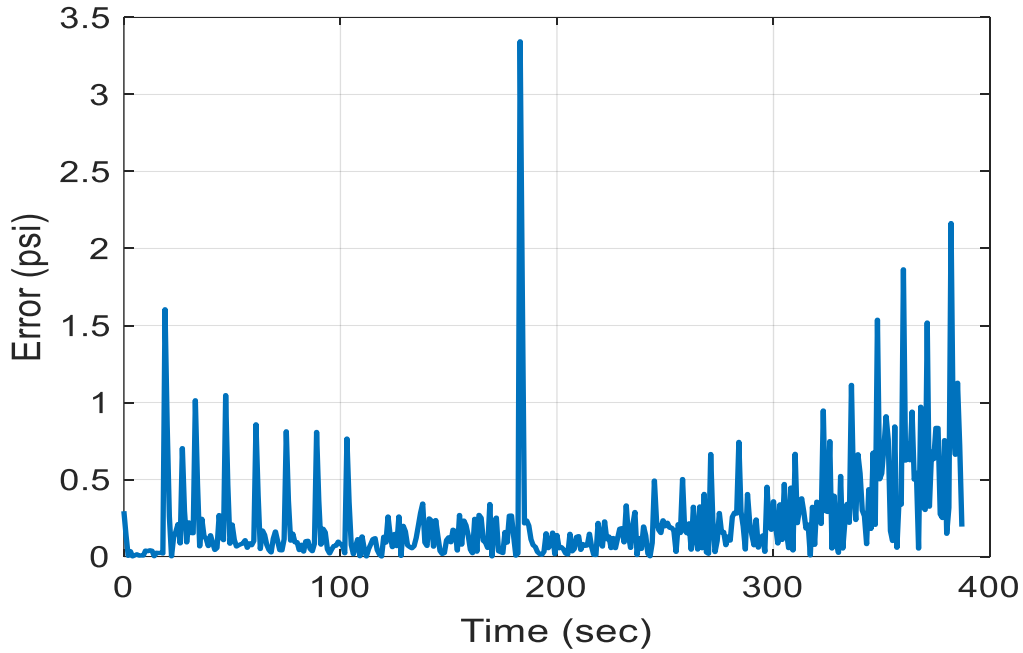


Figure 36 The Absolute Error Signal e Between the Output Response and The Estimated Output Response with LQR

Figures 34-36 illustrate the responses of the output. Step reference with magnitude 1 is presented.

5.5. Sliding Mode Control (SMC)

The first step in developing a sliding mode controller is to establish the sliding surface, as described in reference [72]. The sliding surface is shown or defined as:

$$s(k) = e(k + 1) + C \cdot e(k) \quad (31)$$

The equation for the sliding surface, denoted as s , is defined as follows: C is a positive constant that determines the system bandwidth (also known as λ in some literature), and e is the error signal, which is the difference between the reference signal and the process variable. Once the sliding surface has been established, the next step is to generate a control

signal that can effectively reach and maintain the sliding surface. This process is subject to:

$$s(k) \cdot s(k + 1) > 0 \quad (32)$$

In order to satisfy this requirement, the output signal of the discontinuous controller is determined by applying the sign function to the variable that represents the current value of the sliding surface. Additionally, K is a positive constant.

$$u = K \cdot \text{sgn}(s) \quad (33)$$

[73] defines the sign function (sgn) as

$$s = \begin{cases} -1, & s < 0 \\ +1, & s \geq 0 \end{cases} \quad (34)$$

Utilizing the sign function may induce chattering, potentially impacting the system. Mitigating this phenomenon is crucial for Sliding Mode Control (SMC). To prevent chattering, one possible method is to replace the sign function with the pseudo function:

$$u = K \cdot \frac{s}{|s| + \delta} \quad (35)$$

Here, δ the tuning parameter is a modest positive constant that is intended to reduce chattering. [74]. The selection of δ is pivotal, as opting for a value too small may not entirely eliminate chattering, while choosing a value too large may pose challenges for the controller when reaching the reference value [75].

The values of SMC parameters are as the following:

$$C = 5, \quad K = 0.1, \quad \delta = 0.3$$

The system underwent testing with a step function, wherein the setpoint went from 0 to 1 Psi within the initial second of the simulation. The corresponding output response is showed in figure 37.

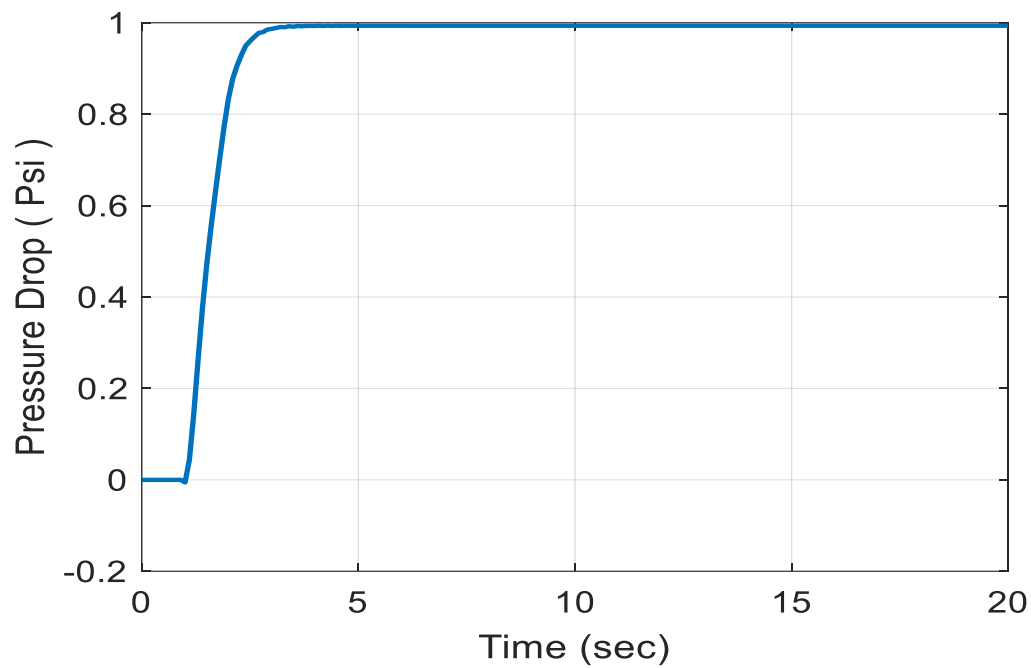


Figure 37 Closed-Loop Step Response of Linearized Model With SMC

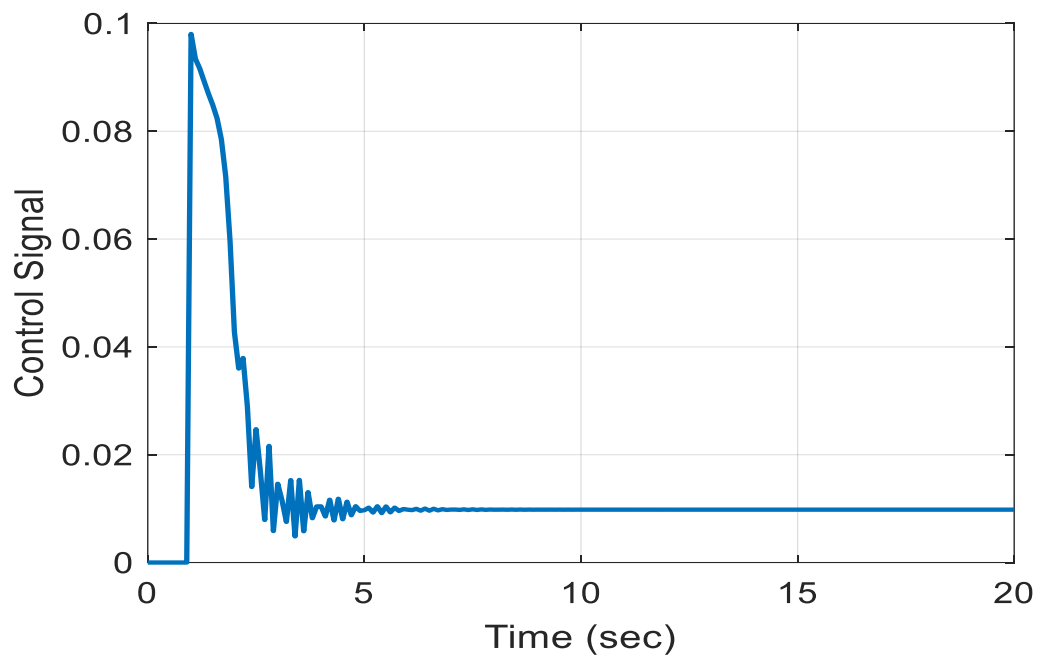


Figure 38 The Control signal u of SMC

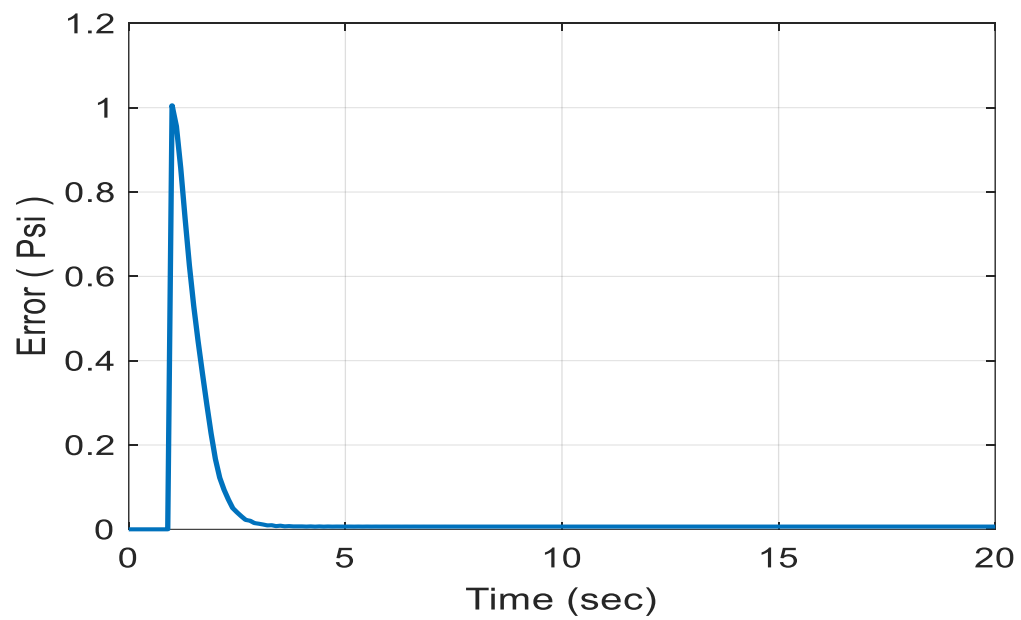


Figure 39 The Error Signal e of The Linearized Response With SMC

Below, the output response is tested with the actual data.

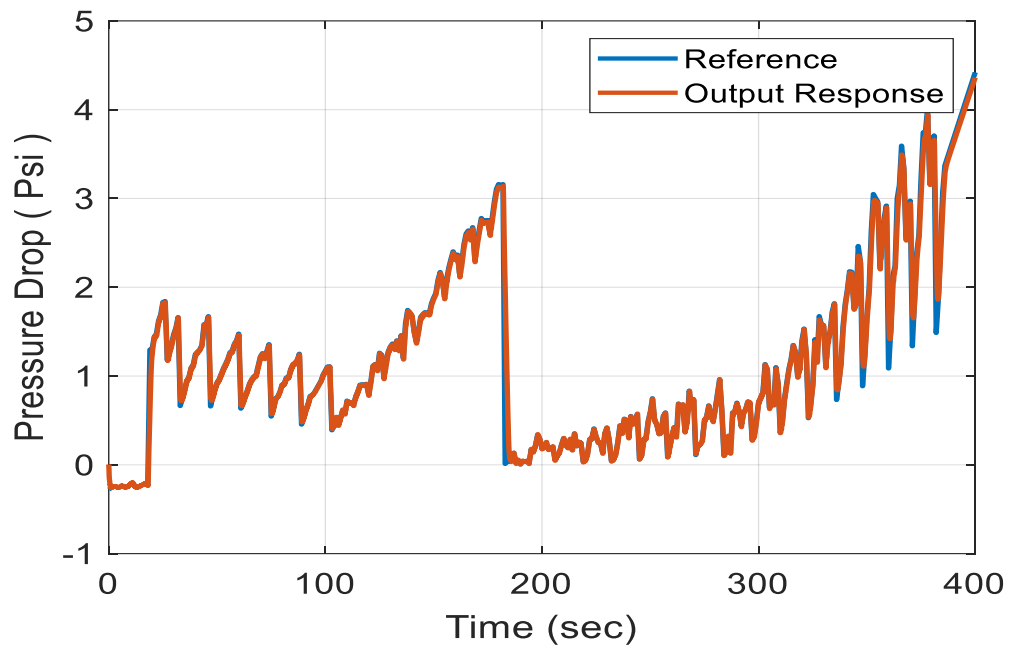


Figure 40 Closed-Loop Output Response vs Estimated Output Response of The Linearized Model with SMC Controller

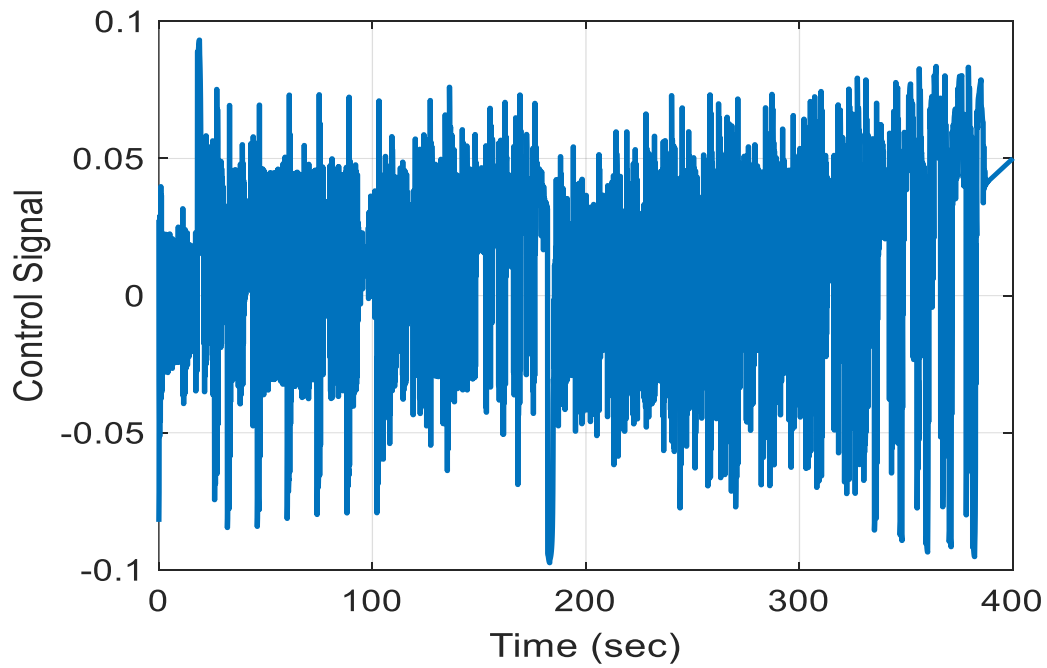


Figure 41 The Control Signal u of SMC

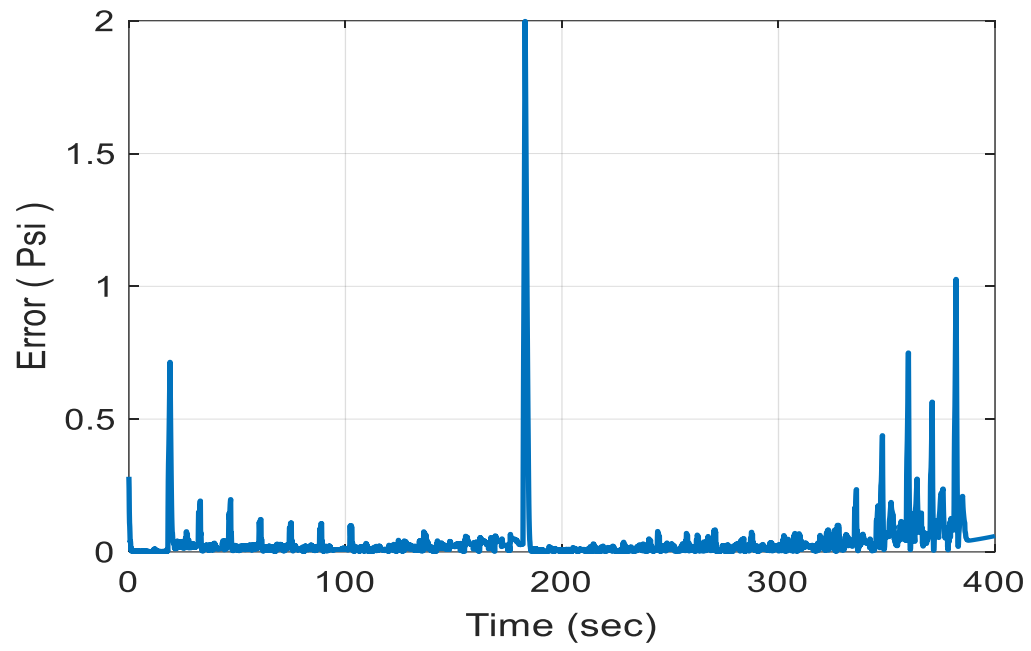


Figure 42 The Absolute Error Signal e Between the Output Response and The Estimated Output Response with SMC

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1. Conclusion

The purpose of this study was to use different identification techniques to derive an accurate mathematical model for pressure drop in multiphase system, as well as to design linear and nonlinear control algorithms to stabilize and evaluate the results of these control systems using simulation software (MATLAB).

In conclusion, this research has demonstrated the importance of investigating experimental data of process control systems, particularly in the context of multi-phase flow in pipes commonly found in petroleum and chemical systems. The study has focused on pressure drop in two-phase gas/liquid flow data from a control point of view, utilizing system identification methods such as neural network-based model, Auto Regression eXogenous (ARX) model, and Hammerstein model. By comparing the characteristics and responses of these models, the best fit to the data with simplicity was selected.

The PID control mechanism continuously modifies the model parameters of the system to consider any changes in the system's behavior, while the SMC algorithm generates the system's optimal control signals. The study has shown that these approaches have the potential to operate more effectively and efficiently for multi-phase systems, which are commonly utilized in sectors such as oil and gas, chemical processing, and wastewater treatment.

The research findings have profound implications for the design and operation of multi-phase flow systems. By employing data-driven and adaptive algorithms, pressure control can be enhanced, leading to improved reliability and efficiency of these systems.

6.2. Future Work

Future work in this research area could focus on several areas to further improve the understanding and control multi-phase flow systems pressure drop. The application of the techniques studied in this research could be extended to other industries and systems beyond petroleum and chemical processing. This could involve investigating the use of these techniques in other sectors such as food processing, pharmaceuticals, and water treatment, among others.

- Studying the influence of liquid characteristics, such as density, viscosity, and surface tension, on pressure drop enables a more thorough comprehension of the model. Conducting experiments to quantify the impact on pressure drop and integrating such findings into the model would enhance its accuracy. This refinement could extend the model's applicability to a broader spectrum of multi-phase flow systems.
- The development of adaptive control algorithms to account for different temperatures could enhance the accuracy of the estimation when operating in a real environment. This could involve developing algorithms that can adapt to changes in temperature and adjust the control signals accordingly. This could improve the reliability and efficiency of multi-phase flow systems and enable them to operate more effectively in real-world environments.

- An optimal controller such as Model Predictive Control (MPC) could be proposed and compared with the controllers proposed in this thesis. This could involve developing an MPC controller and comparing its performance with the controllers proposed in this research. This could provide insights into the effectiveness of different control mechanisms and enable the development of more effective and efficient control strategies for multi-phase flow systems.
- The work could be extended to inclined pipes in multi-phase systems and tested with the proposed control techniques. This could involve conducting experiments to measure pressure drop in inclined pipes and developing control strategies that can account for the effects of gravity. This could improve the understanding and control of multi-phase flow systems in a wider range of applications.

Overall, these areas of future work have the potential to further improve the understanding and control of pressure drop in multi-phase flow systems, and could have significant implications for a range of industrial applications.

REFERENCES

- [1] A. S. AL-Dogail and R. N. Gajbhiye, “Effects of density, viscosity and surface tension on flow regimes and pressure drop of two-phase flow in horizontal pipes,” *J. Pet. Sci. Eng.*, vol. 205, Oct. 2021, doi: 10.1016/j.petrol.2021.108719.
- [2] O. C. Jones and N. Zuber, “STATISTICAL METHODS FOR MEASUREMENT AND ANALYSIS IN TWO-PHASE FLOW,” *Int. Heat Transf. Conf. 5*, pp. 200–204, 1974, doi: 10.1615/IHTC5.480.
- [3] A. Al-Dogail, R. Gajbhiye, A. Alnajim, and M. Alnaser, “Dimensionless Artificial Intelligence-Based Model for Multiphase Flow Pattern Recognition in Horizontal Pipe,” 2022. [Online]. Available: <http://onepetro.org/PO/article-pdf/37/02/244/2687421/spe-209198-pa.pdf/1>
- [4] H. Ladva, P. Tardy, P. Howard, and E. Dussan V, “Multiphase flow and drilling-fluid filtrate effects on the onset of production,” *SPE J.*, vol. 6, no. 04, pp. 425–432, 2001.
- [5] B. Sun *et al.*, “Multiphase flow behavior for acid-gas mixture and drilling fluid flow in vertical wellbore,” *J. Pet. Sci. Eng.*, vol. 165, pp. 388–396, 2018.
- [6] A. A. Osipov, “Fluid mechanics of hydraulic fracturing: a review,” *J. Pet. Sci. Eng.*, vol. 156, pp. 513–535, 2017.
- [7] O. Bratland, “Pipe flow 2: multi-phase flow assurance.” Ove Bratland, 2010.
- [8] E. Kanin, A. Osipov, A. L. Vainshtein, and E. Burnaev, “A predictive model for steady-state multiphase pipe flow: Machine learning on lab data,” *J. Pet. Sci. Eng.*, vol. 180, pp. 727–746, 2019.

- [9] K. Aziz and G. W. Govier, "Pressure drop in wells producing oil and gas," *J. Can. Pet. Technol.*, vol. 11, no. 03, 1972.
- [10] D. H. Beggs and J. P. Brill, "A study of two-phase flow in inclined pipes," *J. Pet. Technol.*, vol. 25, no. 05, pp. 607–617, 1973.
- [11] H. Mukherjee and J. P. Brill, "Liquid holdup correlations for inclined two-phase flow," *J. Pet. Technol.*, vol. 35, no. 05, pp. 1003–1008, 1983.
- [12] A. R. Hasan and C. S. Kabir, "A study of multiphase flow behavior in vertical wells," *SPE Prod. Eng.*, vol. 3, no. 02, pp. 263–272, 1988.
- [13] A. Ansari, N. Sylvester, C. Sarica, O. Shoham, and J. Brill, "A comprehensive mechanistic model for upward two-phase flow in wellbores," *SPE Prod. Facil.*, vol. 9, no. 02, pp. 143–151, 1994.
- [14] H.-Q. Zhang, Q. Wang, C. Sarica, and J. P. Brill, "Unified model for gas-liquid pipe flow via slug dynamics—part 1: model development," *J Energy Resour Technol*, vol. 125, no. 4, pp. 266–273, 2003.
- [15] T. J. Danielson, K. M. Bansal, R. Hansen, and E. Leporcher, "LEDA: the next multiphase flow performance simulator," in *BHR International Conference on Multiphase Production Technology*, BHR, 2005, p. BHR-2005.
- [16] K. H. Bendiksen, D. Maines, R. Moe, and S. Nuland, "The dynamic two-fluid model OLGA: Theory and application," *SPE Repr. Ser.*, no. 1, pp. 52–61, 2004.
- [17] B. Krasnopolsky, A. Starostin, and A. Osipov, "Unified graph-based multi-fluid model for gas-liquid pipeline flows," *Comput. Math. Appl.*, vol. 72, no. 5, pp. 1244–1262, 2016.

- [18] E. Osman, M. A. Ayoub, and M. A. Aggour, “Artificial neural network model for predicting bottomhole flowing pressure in vertical multiphase flow,” in *SPE middle east oil and gas show and conference*, SPE, 2005, p. SPE-93632.
- [19] X. Li, J. Miskimins, and B. T. Hoffman, “A combined bottom-hole pressure calculation procedure using multiphase correlations and artificial neural network models,” in *SPE Annual Technical Conference and Exhibition?*, SPE, 2014, p. SPE-170683.
- [20] A. Amin, S. Imtiaz, A. Rahman, and F. Khan, “Nonlinear model predictive control of a Hammerstein Wiener model based experimental managed pressure drilling setup,” *ISA Trans.*, vol. 88, pp. 225–232, 2019.
- [21] J.-M. Godhavn, “Control requirements for automatic managed pressure drilling system,” *SPE Drill. Complet.*, vol. 25, no. 03, pp. 336–345, 2010.
- [22] J.-M. Godhavn, A. Pavlov, G.-O. Kaasa, and N. L. Rolland, “Drilling seeking automatic control solutions,” *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 10842–10850, 2011.
- [23] H. B. Siahaan, H. Jin, and M. G. Safonov, “An adaptive PID switching controller for pressure regulation in drilling,” *IFAC Proc. Vol.*, vol. 45, no. 8, pp. 90–94, 2012.
- [24] E. Hauge, O. M. Aamo, and J.-M. Godhavn, “Model-based estimation and control of in/out-flux during drilling,” in *2012 American Control Conference (ACC)*, IEEE, 2012, pp. 4909–4914.
- [25] A. Nandan, S. Imtiaz, and S. Butt, “Robust control of managed pressure drilling,” in *2014 Oceans-St. John’s*, IEEE, 2014, pp. 1–8.

- [26] T. Mulyana, J. Alhilman, and E. Kurniawan, "Data analysis using system identification toolbox of heat exchanger process control training system," *2016 4th Int. Conf. Inf. Commun. Technol. ICoICT 2016*, Sep. 2016, doi: 10.1109/ICOICT.2016.7571958.
- [27] A. Simpkins, "System Identification: Theory for the User, 2nd Edition (Ljung, L.; 1999) [On the Shelf]," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 95–96, Jun. 2012, doi: 10.1109/MRA.2012.2192817.
- [28] A. Gilat, "MATLAB : an introduction with applications," p. 343, 2005.
- [29] M. F. Rukandi, S. H. Pratama, Suprijanto, and F. I. Muchtadi, "Movement Imagery Process in Brain-Computer Interface Based on Mu Rhythm," *2nd Int. Conf. Instrum. Control Autom.*, pp. 294–297, 2009.
- [30] D. Hanafi and A. A. A. Emhemed, "Control Modes Comparison Results Focused on Real Heat Exchanger System Response," *Int. J. Integr. Eng.*, vol. 3, no. 2, pp. 1–4, 2011.
- [31] T. Mulyana, M. N. M. Than, and D. Hanafi, "A discrete time model of boiler drum and heat exchanger QAD Model BDT 921," in *Proceedings of the International Conference on Instrumentation, Control & Automation (ICA2009)*, 2009, pp. 154–159.
- [32] D. Hanafi, M. N. M. Than, A. A. A. Emhemed, T. Mulyana, A. M. Zaid, and A. H. Johari, "Heat exchanger's shell and tube modeling for intelligent control design," *2011 IEEE 3rd Int. Conf. Commun. Softw. Netw. ICCSN 2011*, pp. 37–41, 2011, doi: 10.1109/ICCSN.2011.6014844.

- [33] T. Mulyana, M. N. M. Than, and N. A. Mustapha, "Identification of heat exchanger qad model BDT 921 based on hammerstein-wiener model," in *International Seminar on the Application of Science & Mathematics*, 2011.
- [34] T. Mulyana, F. N. Suhaimi, D. Hanafi, and M. N. M. Than, "Arx model of four types heat exchanger identification," in *Proceedings of Malaysian Technical Universities International Conference on Engineering and Technology (MUiCET'11)*, 2011.
- [35] A. Sadikin, M. A. Jamil, and N. M. Isa, "Investigation of Boiling Heat Transfer and Flow Regimes in a Heat Exchanger," *Appl. Mech. Mater.*, vol. 660, pp. 679–683, 2014, doi: 10.4028/WWW.SCIENTIFIC.NET/AMM.660.679.
- [36] C. Bharti and J. Barve, "Smart Data Acquisition and Control of Heat Exchanger," *Master Proj. Rep.*, 2014.
- [37] N. A. Daud and S. M. Salleh, "Modeling of Heat Exchanger by Using Bio-Inspired Algorithm," *Appl. Mech. Mater.*, vol. 660, pp. 831–835, 2014, doi: 10.4028/WWW.SCIENTIFIC.NET/AMM.660.831.
- [38] A. Sadikin and N. Asmuin, "Two-Phase Flow Pressure Drop Model for a Shell Side of a Shell of Heat Exchanger," *Appl. Mech. Mater.*, vol. 465–466, pp. 613–616, 2014, doi: 10.4028/WWW.SCIENTIFIC.NET/AMM.465-466.613.
- [39] M. Al-Naser, M. Elshafei, and A. Al-Sarkhi, "Two-Phase Flow Regimes Identification Using Artificial Neural Network With Nonlinear Normalization," in *Proceedings of the 2nd International Conference on Fluid Flow, Heat and Mass Transfer*, 2015.

- [40] N. H. Othman, “System identification of heat exchanger using generalized poisson moment functional (GPMF),” PhD Thesis, Universiti Tun Hussein Onn Malaysia, 2014.
- [41] M. Al-Naser, M. Elshafei, and A. Al-Sarkhi, “Artificial neural network application for multiphase flow patterns detection: A new approach,” *J. Pet. Sci. Eng.*, vol. 145, pp. 548–564, Sep. 2016, doi: 10.1016/J.PETROL.2016.06.029.
- [42] C. Shanthi and N. Pappa, “An artificial intelligence based improved classification of two-phase flow patterns with feature extracted from acquired images,” *ISA Trans.*, vol. 68, pp. 425–432, May 2017, doi: 10.1016/J.ISATRA.2016.10.021.
- [43] J. Xiao, X. Luo, Z. Feng, and J. Zhang, “Using artificial intelligence to improve identification of nanofluid gas–liquid two-phase flow pattern in mini-channel,” *AIP Adv.*, vol. 8, no. 1, p. 015123, Jan. 2018, doi: 10.1063/1.5008907.
- [44] J. Xavier, F. Ribeiro, R. Liao, A. M. Aliyu, and Z. Liu, “Prediction of Pressure Gradient in Two and Three-Phase Flows in Vertical Pipes Using an Artificial Neural Network Model,” *Int. J. Eng. Technol. Innov.*, vol. 9, no. 3, pp. 155–170, 2019.
- [45] T. Söderström, “Identification of stochastic linear systems in presence of input noise,” *Automatica*, vol. 17, no. 5, pp. 713–725, 1981.
- [46] T. Söderström, “Errors-in-variables methods in system identification,” *Automatica*, vol. 43, no. 6, pp. 939–958, 2007.
- [47] J. Sarangapani, *Neural network control of nonlinear discrete-time systems*. CRC press, 2018.

- [48] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino, *Stable adaptive control and estimation for nonlinear systems: neural and fuzzy approximator techniques*. John Wiley & Sons, 2004.
- [49] M. Nørgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural networks for modelling and control of dynamic systems*. Springer-Verlag London Limited. London. England, 2001.
- [50] O. Omidvar and D. L. Elliott, *Neural systems for control*. Elsevier, 1997.
- [51] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.
- [52] R. Ilin, R. Kozma, and P. J. Werbos, “Beyond feedforward models trained by backpropagation: A practical training tool for a more efficient universal approximator,” *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 929–937, 2008.
- [53] R. Chiong, “Intelligent systems for automated learning and adaptation: emerging trends and applications: emerging trends and applications,” 2009.
- [54] A. Ohata, K. Furuta, and H. Nita, “Identification of nonlinear ARX model with input and output dependent coefficients,” in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, IEEE, 2006, pp. 2577–2582.
- [55] R. Haber and H. Unbehauen, “Structure identification of nonlinear dynamic systems—a survey on input/output approaches,” *Automatica*, vol. 26, no. 4, pp. 651–677, 1990.

- [56] L. Sun, J. Hou, C. Xing, and Z. Fang, “A robust hammerstein-wiener model identification method for highly nonlinear systems,” *Processes*, vol. 10, no. 12, p. 2664, 2022.
- [57] M. Schoukens and K. Tiels, “Identification of block-oriented nonlinear systems starting from linear approximations: A survey,” *Automatica*, vol. 85, pp. 272–292, 2017.
- [58] L. Ljung, “System Identification Toolbox for use with Matlab. Version 7. The MathWorks,” *Inc Natick MA*, 2007.
- [59] J. Xavier, S. Patnaik, and R. C. Panda, “Process modeling, identification methods, and control schemes for nonlinear physical systems—A comprehensive review,” *ChemBioEng Rev.*, vol. 8, no. 4, pp. 392–412, 2021.
- [60] S. Gambhire, D. R. Kishore, P. Londhe, and S. Pawar, “Review of sliding mode based control techniques for control system applications,” *Int. J. Dyn. Control*, vol. 9, pp. 363–378, 2021.
- [61] S. Emelyanov, “Control of first order delay systems by means of an astatic controller and nonlinear correction,” *Autom Remote Control*, vol. 8, pp. 983–991, 1959.
- [62] V. Utkin, “Variable structure systems with sliding modes,” *IEEE Trans. Autom. Control*, vol. 22, no. 2, pp. 212–222, 1977.
- [63] L. Wu, J. Liu, S. Vazquez, and S. K. Mazumder, “Sliding mode control in power converters and drives: A review,” *IEEECAA J. Autom. Sin.*, vol. 9, no. 3, pp. 392–406, 2021.

- [64] H. Sira-Ramirez, “On the dynamical sliding mode control of nonlinear systems,” *Int. J. Control*, vol. 57, no. 5, pp. 1039–1061, 1993.
- [65] P. Samuelsson, H. Norlander, and B. Carlsson, “An integrating linearization method for Hammerstein models,” *Automatica*, vol. 41, no. 10, pp. 1825–1828, 2005.
- [66] E. Eskinat, S. H. Johnson, and W. L. Luyben, “Use of Hammerstein models in identification of nonlinear systems,” *AIChE J.*, vol. 37, no. 2, pp. 255–268, 1991.
- [67] K. J. Åström and B. Wittenmark, *Adaptive control*, 2. ed. Mineola, N.Y: Dover Publ, 2008.
- [68] C.-F. Lindberg and B. Carlsson, “Nonlinear and set-point control of the dissolved oxygen concentration in an activated sludge process,” *Water Sci. Technol.*, vol. 34, no. 3–4, pp. 135–142, 1996.
- [69] R. C. Smith, “Inverse compensation for hysteresis in magnetostrictive transducers,” *Math. Comput. Model.*, vol. 33, no. 1–3, pp. 285–298, 2001.
- [70] G. C. Goodwin, *Control System Design*. CONTROL SYSTEM DESIG, 2000.
- [71] A. Al-Mahturi and H. Wahid, “Optimal tuning of linear quadratic regulator controller using a particle swarm optimization for two-rotor aerodynamical system,” *Int. J. Electron. Commun. Eng.*, vol. 11, no. 2, pp. 196–202, 2017.
- [72] J. M. Kiss, P. T. Szemes, and P. Aradi, “Sliding mode control of a servo system in LabVIEW: Comparing different control methods,” *Int. Rev. Appl. Sci. Eng.*, vol. 12, no. 2, pp. 201–210, May 2021, doi: 10.1556/1848.2021.00250.

- [73] R.-E. Precup, S. Preitl, and P. Korondi, “Fuzzy controllers with maximum sensitivity for servosystems,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1298–1310, 2007.
- [74] H. Maghfiroh, A. Sujono, C. H. B. Apribowo, and others, “Basic tutorial on sliding mode control in speed control of DC-motor,” *J. Electr. Electron. Inf. Commun. Technol.*, vol. 2, no. 1, pp. 1–4, 2020.
- [75] E. H. Dursun and A. Durdu, “Speed control of a DC motor with variable load using sliding mode control,” *Int. J. Comput. Electr. Eng.*, vol. 8, no. 3, p. 219, 2016.

Vitae

Name	Aiman Farid Bawazir
Nationality	Yemeni
Date of Birth	11/7/1993
Email	aymenfareed1@gmail.com
Contact Details	+966-504388489
Address	Mukalla, Hadramout 4337, Yemen
Academic Background	B.Sc. Control and Instrumentation System Engineering, (2014-2019), KFUPM, Dhahran, Saudi Arabia. M.S Control and Systems Engineering, (2021 - 2023), KFUPM, Dhahran, Saudi Arabia. My research in interests are in system identification, optimal control and control theory.