Hardware Implementation of Relay Control with Enhanced Autotune Identification Using Machine Learning

A
MTP Report
Submitted in fulfillment of the requirement for
the degree of
Master of Technology

Submitted by

Manish Kumar

(224102507)

Under the Supervision of

Prof. Somanath Majhi

Department of Electronics and Electrical Engineering

Indian Institute of Technology, Guwahati

June 2024

Declaration

This is to certify that the thesis entitled "**Hardware Implementation of Relay Control with Enhanced Autotune Identification Using Machine Learning**", submitted by me to the Indian Institute of Technology Guwahati, for the award of the degree of M.Tech, is a bonafide work carried out by me under the supervision of **Prof. Somanath Majhi**. The content of this thesis, in full or in parts, have not been submitted to any other University or Institute for the award of any degree.

Manish Kumar

Department of Electronics and Electrical Engineering,

Indian Institute of Technology Guwahati, Assam.

Certificate

This is to certify that the work contained in this thesis entitled "**Hardware Implementation of Relay Control with Enhanced Autotune Identification Using Machine Learning**" is a bonafide work of **Manish Kumar** (**Roll No. 224102507**), carried out in the Department of Electronics and Electrical Engineering, Indian Institute of Technology, Guwahati under my supervision, and it has not been submitted elsewhere for a degree.

Supervisor: **Prof. Somanath Majhi**

Professor,

June, 2024 — Department of Electronics and Electrical Engineering,

Guwahati. — Indian Institute of Technology Guwahati, Assam.

Acknowledgements

I am extremely grateful to my supervisor **Prof. Somanath Majhi** for this work. The research work that I have done is a development of his idea. This development might not have been possible without his valuable feedback and constant support. The freedom, regular follow-up, and relaxed environment he gave created a conducive environment for research. Apart from research work, I am very thankful for the profound life experience during this year. It was an absolute pleasure to work under him.

I would also like to express gratitude towards my family for being the very basis of my life and for the constant love and support all through the journey of my life. I am obliged to all faculty members of the EEE Department for the knowledge they have provided.

Sincerely

Manish Kumar

Abstract

Relay-based parameter identification of dynamical systems ensures robust and effective control strategies in automatic control systems. It ensures stable, sustained oscillatory output, which is vital for the parameter identification of unknown process models. Describing function-based sets of expressions are derived to identify the parameters of the first order plus dead time (FOPDT) transfer function model for single and multiple poles. A hardware model is constructed with a transportation delay of at least one second and consists of a relay circuit, a process model, and a delay circuit. Machine learning models are integrated to enhance the accuracy of parameter identification. Real-time data from the hardware model, obtained by modifying the physical components of the process model, is utilized to train the machine learning model for improved autotune identification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In industrial setups, numerous controllers are deployed to manage processes, each requiring individual tuning to match specific process dynamics, ensuring satisfactory and robust control performance. However, manual adjustment of these controllers is not only unreliable but also time-consuming [1]. Consequently, automatic tuning techniques have garnered attention among engineers.

Relay-based automatic tuning features are commonly integrated into many industrial controllers [2]. These controllers employ a relay feedback mechanism to induce sustained oscillations within the system, enabling the identification of system parameters through explicit expressions derived from the describing function approach. These estimated parameters are subsequently employed for the automatic tuning of controllers.

The pioneering work by Astrom et al. (1984) [3] demonstrated the successful application of relay feedback techniques for autotuning proportional-integral-derivative (PID) controllers. Conducting relay feedback autotuning tests under closed-loop control allows for maintaining the process near the setpoint, achieved by choosing appropriate relay parameters.

Numerous researchers have since modified relay control-based autotuning techniques to enhance accuracy in system model parameter estimation, facilitating the design of model-based controllers. Various approaches have been explored to improve autotuned controller accuracy using relay-based parameter identification. For instance, Majhi and Padhy [4] devised a method for determining system dynamics of both stable and unstable first-order transfer functions. Sharma et al. [5] utilized neural networks to identify parameters of stable first-order plus dead time(FOPDT) transfer functions. Kumar and Padhy [6] developed analytical expressions to determine measurement sensitivity, focusing on variations in relative error between time constant (T) and time delay (D) concerning limit cycle amplitude (A_p) and frequency (f). Chang et al. [7] proposed a method for determining first-order, second-order, and higher-order transfer functions using z-transform.

However, despite these efforts, existing methods may have limitations in improving accuracy beyond a certain extent during real-time process operations. In this thesis, I propose the utilization of machine learning-based parameter identification techniques. These techniques aim to reduce dynamic errors in real-time processes and enhance the accuracy of system parameters for effective autotuning.

1.1 Problem Statement

Despite the presence of reliable sensors, measurement noise stemming from control valves, equipment, or inherent process characteristics often introduces distortions in the process output in practical settings. This noise complicates the accurate evaluation of output signal period and amplitude, thereby leading to erroneous process identification. Addressing these limitations is crucial for effective system analysis.

Our primary focus is to mitigate the impact of measurement noise by refining the original output signal to determine parameters accurately using the describing function approach. While the relay-based parameter identification method offers a straightforward means of determining system dynamics for controller autotuning, its accuracy diminishes notably with an increased ratio of time constant to time delay.

To overcome these challenges, our core strategy revolves around leveraging machine learning algorithms for precise system parameter identification in the presence of disturbances. This research endeavors to identify the most suitable machine learning model capable of accurately estimating system dynamics parameters. It involves the utilization of real-time data collected from actual systems to ascertain the optimal model for enhanced accuracy in parameter estimation, paving the way for more robust and reliable control system designs.

1.1.1 Methodology

As we have seen in the problem statement, the noisy output may lead to severe inaccuracy and malfunctioning of the relay, which reflects on accurate parameter identification of the system. To demonstrate the influence of noises, I have designed a practical dynamical system with the help of electrical and electronic components to show the efficacy of the proposed machine learning model. The system parameters are estimated with the help of explicit expressions derived from the describing function approach using real-time data that I am collecting from the proposed hardware system. Then, further test various machine learning models to estimate system parameters accurately.

1.2 Contributions to this Thesis

Although relay feedback parameter identification has been widely addressed, there still exists a lot of scope to improve the system parameters using real-time physical systems. In this thesis, I have designed a hardware process model with physical delay in the system to improve the identified system parameters accuracy of real-time systems using machine learning algorithms.

1.3 Relay

By incorporating a relay into a system, engineers can deliberately induce and study non-linear behaviors, including limit cycles and sustained oscillations. These non-linear phenomena are crucial to understanding the behavior of complex systems and are fundamental for designing non-linear control strategies.

1.4 Describing Function-Based Identification

In describing function-based identification, we estimate system parameters from sustained limit cycle output [1]. Many complex identification problems that are difficult to resolve using conventional techniques can be solved using an approximate describing function approach. For a stable system, the limit cycle output the Fourier series analysis of relay output can be done to calculate the approximated relay gain, which is used to calculate system parameters. From Fourier series analysis, Approximated relay gain,

$$N = \frac{4h}{\pi A_p} \tag{1.1}$$

where h is the relay height, and A_p is the amplitude of the output signal.

1.5 Identification Method

In relay-based parameter identification, the describing function approach is widely utilized due to its computational simplicity and straightforward methodology in determining unknown parameters of a model transfer function [1]. This technique commonly involves two types of identification processes.

1.5.1 Offline Identification scheme

The first type is the Offline Identification scheme, which encompasses the offline determination of system parameters using the describing function (DF) approach through closed-loop tests. A conventional structure for offline identification of a non-minimum process model is depicted in Figure 1.1. In this structure, an ideal relay is employed in conjunction with a typical non-minimum phase stable transfer function that includes dead time, assumed as the process model $G(s)$.

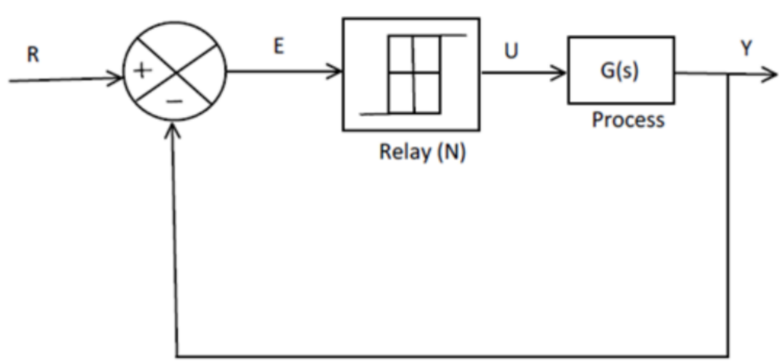


Figure 1.1: General block diagram for offline identification scheme .

In the relay feedback test, where R represents the reference input, U denotes the input applied from the relay, and Y signifies the output of the process model, a limit cycle output is generated. The frequency of the ultimate limit cycle output is determined as $\omega = \frac{2\pi}{T}$ denotes the ultimate time period. Utilizing Fourier series expansion, the amplitude (A_p) is derived from the fundamental harmonics of the limit cycle output. The equivalent gain of non-linearity (represented by the relay) is approximated as:

$$N = \frac{4h}{\pi A_p} \tag{1.2}$$

where h stands for the relay height. During the identification process, the reference input is set to zero to facilitate the generation of sustained oscillatory output or limit cycle output.

In industrial processes, the offline identification method is extensively employed to ascertain the parameters of the process model. This is accomplished by deriving an analytical expression (as outlined in Chapter 3) utilizing the amplitude and time period obtained from the sustained limit cycle output.

1.5.2 Online Identification scheme

This section describes the online identification of system parameters using the describing function (DF) approach based on closed-loop testing. Online identification represents an advanced approach compared to offline identification as it serves the dual purpose of autotuning the PID controller and concurrently estimating the process model. In this method, a PID controller $G_c(s)$ is connected in parallel with the relay within the loop configuration. Consequently, the relay operates in conjunction with both the controller and the process model G(s). The combined action of the controller and the relay assists in stabilizing the process through the inner feedback loop, thereby facilitating the simultaneous process of parameter identification for the process model and autotuning for the PID controller.

Initially, the relay is connected with the PID controller to determine the parameters of the process model using information obtained from the relay and the process loop. Subsequently, the relay is disconnected, enabling the tuning of the PID controller in alignment with the process model. Figure 1.2 illustrates the structural representation of the online identification method.

Connecting the controller in parallel with the relay offers several advantages. It aids in stabilizing sustained oscillatory output in the presence of noise by ensuring asymmetrical

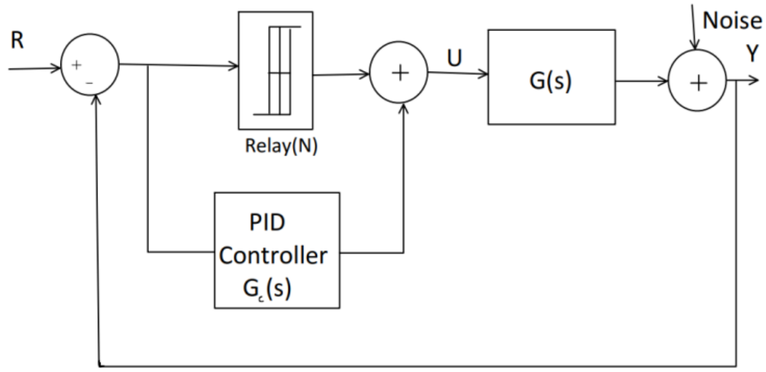


Figure 1.2: General block diagram for online identification scheme .

output is made symmetrical. An asymmetrical output can impede accurate estimation of the amplitude and time period of the limit cycle output, potentially resulting in significant errors in parameter estimation.

Chapter 2

Literature Review

In this segment, a comparative analysis of different methodologies employed for system model parameter identification is presented. The original ATV method, proposed by Luyben [8], involves a singular autotune test to ascertain the ultimate frequency (ω) and amplitude (A_p) on the limit cycle, enabling the derivation of transfer function model parameters. However, the effectiveness of this approach is confined to specific parameter ranges. Several researchers have introduced methods to enhance accuracy and leverage relay-based identification for autotuning controllers. Bajrangbali et al. [2] proposed a technique to identify process model parameters utilizing a hysteresis relay, even in the presence of measurement noise. Kumar and Padhy [6] outlined analytical expressions to determine measurement sensitivity, assessing the variation in relative error between the time constant (T_1) and time delay (D) concerning the amplitude (A_p) and frequency (ω) of the limit cycle. Luyben [8] extended this method to deduce the transfer function of non-linear and complex processes. Sharma et al. [5] utilized neural networks for identifying parameters in stable First Order Plus Dead Time (FOPDT) transfer functions. Majhi and Padhy [4] presented a method to ascertain the system dynamics of both stable and unstable first-order transfer functions. Chang et al. [7] introduced a method to establish first-order, second-order, and higher-order transfer functions using z-transforms. Lavanya et al. [9] proposed a method to estimate parameters of Second Order Plus Dead Time (SOPDT) process models, but its accuracy is limited. Lee et al. [10] applied the describing function approach using a hysteresis relay to estimate parameters of stable FOPDT process models. Majhi and Atherton [11] utilized the state-space technique to determine various process model parameters. Krzystof S. Kula [12] did an online identification of an analog SOPDT with a simulation study. Anees Peringal et al. [13] identify the parameters of aerodynamic and delay sensor dynamics using relay-based identification by selecting a discrete set of values. A. Ayyad et al. [14] used a modified relay feedback test and deep neural network, but this method requires higher mathematical computation. However, these methods did not improve the accuracy after a certain extent for various dynamic errors during real-time process operations. While these algorithms offer accuracy, they necessitate the computation of complex nonlinear equations.

Chapter 3

Parameter Identification

This section outlines the analytical expressions for the system's parameter identification.

3.1 Introduction

The above chapters show that relay feedback-based identification is a less computational and time-efficient method to determine process model parameters. However, the major limitation of this method is its inaccuracy in determining the time constant (T_1) and time delay (D) for the higher time constant to time delay ratio.

Although the relay-based identification technique is routinely used to derive transfer function parameters, the ATV methodology can offer a far more exact estimate of system characteristics. This is because the predicted ultimate gain N and anticipated ultimate frequency of the describing function are essentially approximations of information at critical frequency. A large variation exists between real and calculated parameters, especially for the greater time constant to time delay ratio.

We utilize a machine-learning technique to train varied data and identify the value of parameters using the same model to determine actual parameters and reduce error values. Therefore, with the help of machine learning techniques, we can estimate the parameters accurately after training them with actual parameters.

3.2 Analytical Expressions for Identification of FOPDT Process Model parameter

Typical non-minimum phase stable transfer function with dead time is assumed as a process model [1]. The autotune identification scheme is shown in Figure 1.1 and Figure 1.2, where, $G(s)$ is the process transfer function model.

Transfer function:

$$G(s) = \frac{Ke^{-Ds}}{T_1 s + 1} \tag{3.1}$$

Let relay height be denoted by h When $h \neq 0$, $\omega = 2\pi/T$

$$NG(j\omega) = -1 \tag{3.2}$$

When N is the equivalent gain of the relay.

$$N = \frac{4h}{\pi A_p} \tag{3.3}$$

Condition for sustained oscillatory output, from equations (3.1),(3.2), and (3.3)

$$\frac{4h}{\pi A_p} \cdot \frac{Ke^{-j\omega}}{1 + j\omega T_1} = -1 \tag{3.4}$$

Now, applying magnitude and angle conditions, one can get time constant T_1 and delay D as

$$T_1 = \frac{\sqrt{(\frac{4hK}{\pi A_p})^2 - 1}}{\omega} \tag{3.5}$$

$$D = \frac{\pi - arctan(\omega T_1)}{\omega} \tag{3.6}$$

FOPDT system consists of time constant (T_1), time delay (D), and gain (K) as unknown parameters. ATV method consists of the following steps:

i) The steady state gain (K) is estimated from a steady-state analysis of the system.

ii) Ultimate gain (N) and ultimate frequency (ω) is calculated from relay feedback.

iii) The first-order transfer function is fitted to data to determine unknown time constants and time delays, as given in (3.5) and (3.6).

3.3 Analytical Expressions for Identification of SOPDT Process Model parameter

Transfer function models in process control are often assumed to be stable time-delay process models with no zeros and single or multiple poles or stable process models with a right half-plane zero and multiple poles.

Following the transfer function model, having some generality is considered for this identification problem [1].

$$G(s) = \frac{K(-T_0 s + 1)e^{-Ds}}{(T_1 s + 1)^p} = \frac{K(-\lambda)^p(-T_0 s + 1)e^{-Ds}}{(s - \lambda)^p} \tag{3.7}$$

where$p = 1, 2, 3, ...$and $\lambda = -1/T_1$. The typical non-minimum phase process models can be obtained by setting ($T_0 = 0, D \neq 0$) and ($T_0 \neq 0, D = 0$). When ($T_0 = 0, D = 0$), the process model represents a minimum phase system. From a stable limit cycle output signal, the peak amplitude (A_p) and the full time period of the oscillation (T) are measured for the symmetrical relay amplitude (h).

Under limit cycle conditions, the operating point approximately satisfies the following expressions:

$$|G(j\omega_c)| = \frac{\pi A_p}{4h} \tag{3.8}$$

and

$$\angle G(j\omega_c) = -\pi \tag{3.9}$$

Equations for identification of minimum phase processes of order 2 with $T_0 = 0$ and $D > 0$ are given as:

$$A_p = \frac{4Kh}{\pi} cos^2(\pi/2 - \pi D/(2T)) \tag{3.10}$$

$$\pi - \pi D/T + 2arctan(\pi/(\lambda T)) = 0 \tag{3.11}$$

After solving (3.10) and (3.11), we get the final analytical expressions for the SOPDT process model parameters.

$$D = \frac{Tarcsin(1 - \pi A_p/2Kh)}{\pi} \tag{3.12}$$

$$T_1 = -\frac{Ttan((\pi D/T - \pi)/2)}{\pi} \tag{3.13}$$

With the relay setting of h, the symmetrical relay test results are used to measure A_p and T from the experiment. Now, we can estimate process model parameters for minimum phase process T_1 and D by solving (3.12) and (3.13) for stable processes. Now, we can write the estimated model of dynamics as follows:

$$G_m(s) = \frac{e^-Ds}{(T_1 s + 1)^2} \tag{3.14}$$

Chapter 4

Hardware Design

Various electrical and electronic components are used to construct hardware. We need to design a specific hardware circuit that demonstrates the behavior of a dynamical system for parameter identification. Various components of the hardware setup are discussed in this section.

The hardware development part starts with various design calculations like the design of a relay circuit for square wave input to the system model, the design of an RC circuit to show the real-time system model, and the last design of a delay circuit to provide the physical delay in hardware for analysis of low frequency sustained oscillating signal [15].

4.1 Implementation of Relay

A relay, an electrically operated switch, exhibits various types based on input-output characteristics. In this context, I've designed a symmetrical relay without hysteresis. This relay generates a square wave output with a consistent time period by employing a constant ON and OFF time strategy. To replicate this functionality in hardware, I've utilized an operational amplifier configured as a comparator circuit. This setup is engineered to produce an output similar to the relay, achieving similar square wave characteristics.

4.1.1 Comparator Circuit

The fundamental principle of a comparator circuit revolves around comparing a signal voltage at one input of the operational amplifier (op-amp) with a reference voltage present at the other input. Typically, the comparator operates in an open-loop configuration. I'm utilizing the comparator circuit specifically as a zero-crossing detector in my hardware setup. Within the entirety of my system, the input to the comparator circuit is derived from the system's feedback, representing the sustained oscillatory output. Consequently, the output of this comparator circuit produces a square wave characterized by voltage levels of +Vcc and -Vee.

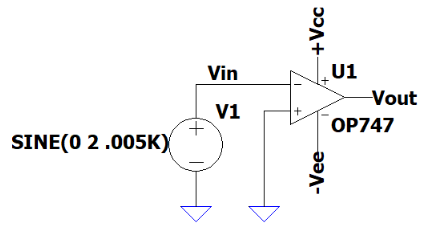


Figure 4.1: Comparator circuit diagram

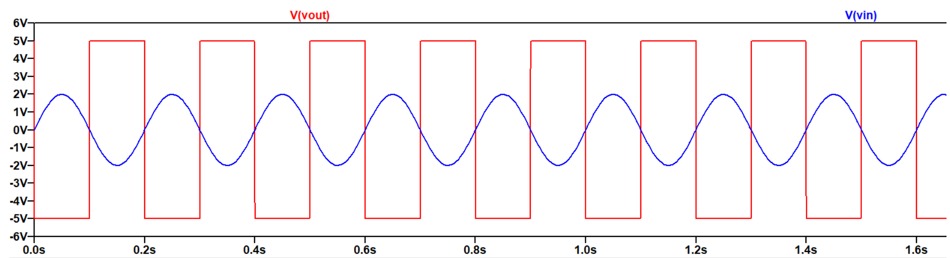


Figure 4.2: Comparator input-output waveform

4.2 Process model circuit

The process model I've developed embodies the system dynamics and is constructed using a resistor and capacitor configuration, forming a low-pass filter circuit. This model, depicted schematically, illustrates the interaction between the components in a manner that emphasizes the system's behavior. The resistor and capacitor are strategically interconnected to allow specific frequencies to pass through while attenuating higher frequencies. This design choice enables the model to exhibit characteristics akin to a low-pass filter, highlighting its ability to modulate signals and illustrate the system's dynamic behavior within certain frequency ranges.

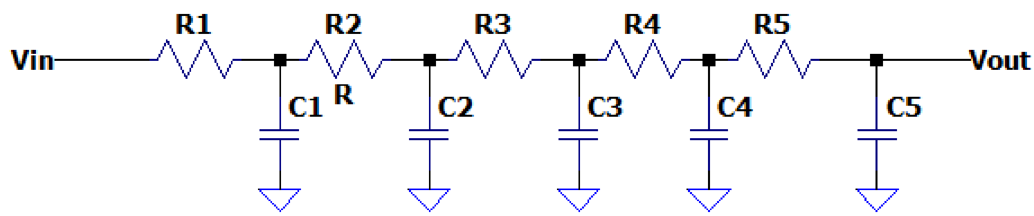


Figure 4.3: Process model circuit diagram

4.3 Time delay circuit

Implementing a time delay within the process model proves essential for capturing lower-frequency datasets, enabling a deeper understanding of process dynamics. In MATLAB simulation, incorporating a transportation delay block facilitates easy manipulation of delay parameters. However, translating this delay mechanism into hardware poses significant challenges. I've engineered a hardware delay circuit using a voltage buffer amplifier configuration to overcome this obstacle. This setup effectively introduces the necessary delay within the hardware system, mirroring the functionality of the transportation delay block in the MATLAB simulation. The voltage buffer amplifier circuit ensures accurate timing delays, facilitating the acquisition of pertinent information regarding process dynamics in the hardware implementation.

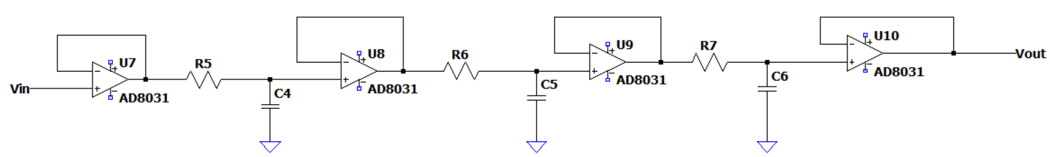


Figure 4.4: Physical delay circuit

4.3.1 Voltage buffer Amplifier

A voltage buffer amplifier, often called a voltage follower circuit, maintains the input voltage amplitude due to its negative unity feedback configuration. To transform the buf-

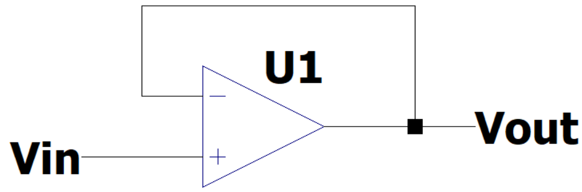


Figure 4.5: Ideal buffer circuit diagram

fer circuit into a delay circuit, I've utilized an AD8031 rail-to-rail operational amplifier known for its high gain bandwidth (80MHz), elevated input impedance, and low output impedance. While ideal buffer circuits typically introduce minimal delay, additional delay in the input signal can be achieved by strategically incorporating resistors and capacitors at the output of the buffer.

For validation, I subjected the input to a sinusoidal signal of 0.01Hz frequency while setting all resistors at 10KΩ and capacitors at 10μF. The output response revealed a delay of approximately 5 seconds, as depicted in Figure 4.4. This delay measurement underscores the effectiveness of the designed buffer circuit as a delay element, allowing for the observation and analysis of delayed signals within the hardware setup.

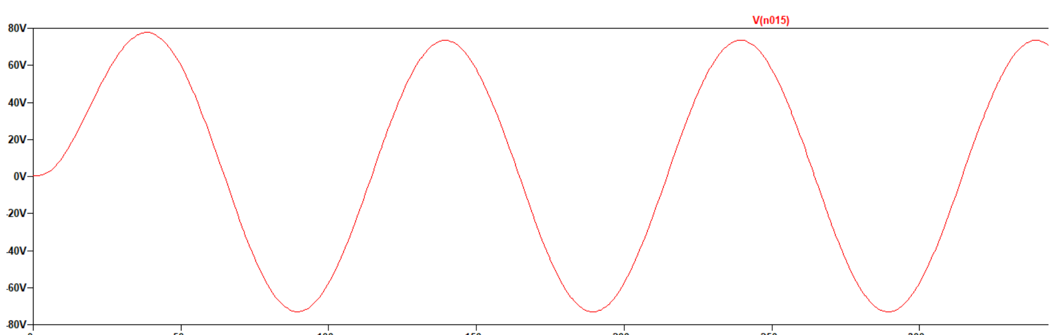


Figure 4.6: Delayed sinusoidal waveform

4.3.2 Final hardware circuit

Successfully cascading all individual circuits to create a comprehensive dynamic hardware circuit. This circuit includes a negative feedback loop from the final output to the input of the comparator circuit. The system has exhibited satisfactory performance through rigorous testing using LTSpice simulation and physical hardware implementation. The output at V_{out1} reliably triggers the relay, while V_{out2} demonstrates sustained oscillations as expected.

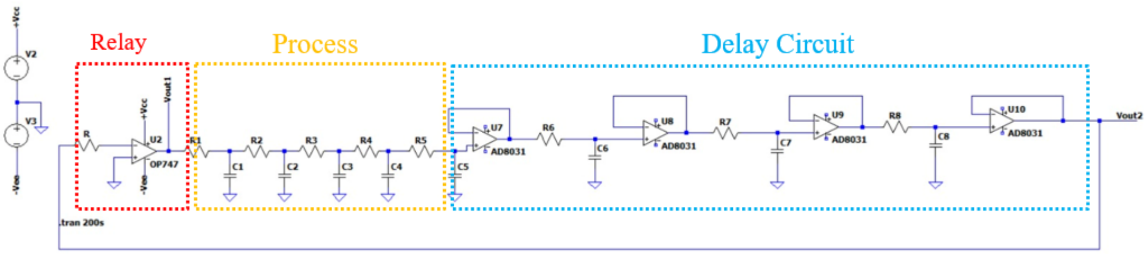


Figure 4.7: Simulated circuit

The open-loop transfer function of the process model is given as:

$$G_p(s) = \frac{1}{(R_1C_1s+1)(R_2C_2s+1)(R_3C_3s+1)(R_4C_4s+1)(R_5C_5s+1)} \tag{4.1}$$

$$G_d(s) = \frac{1}{(R_6C_6s+1)(R_7C_7s+1)(R_8C_8s+1)} \tag{4.2}$$

$$G(s) = G_p(s) \cdot G_d(s) \tag{4.3}$$

now, the close loop transfer function is given as:

$$T(s) = \frac{G(s)}{G(s)+1} \tag{4.4}$$

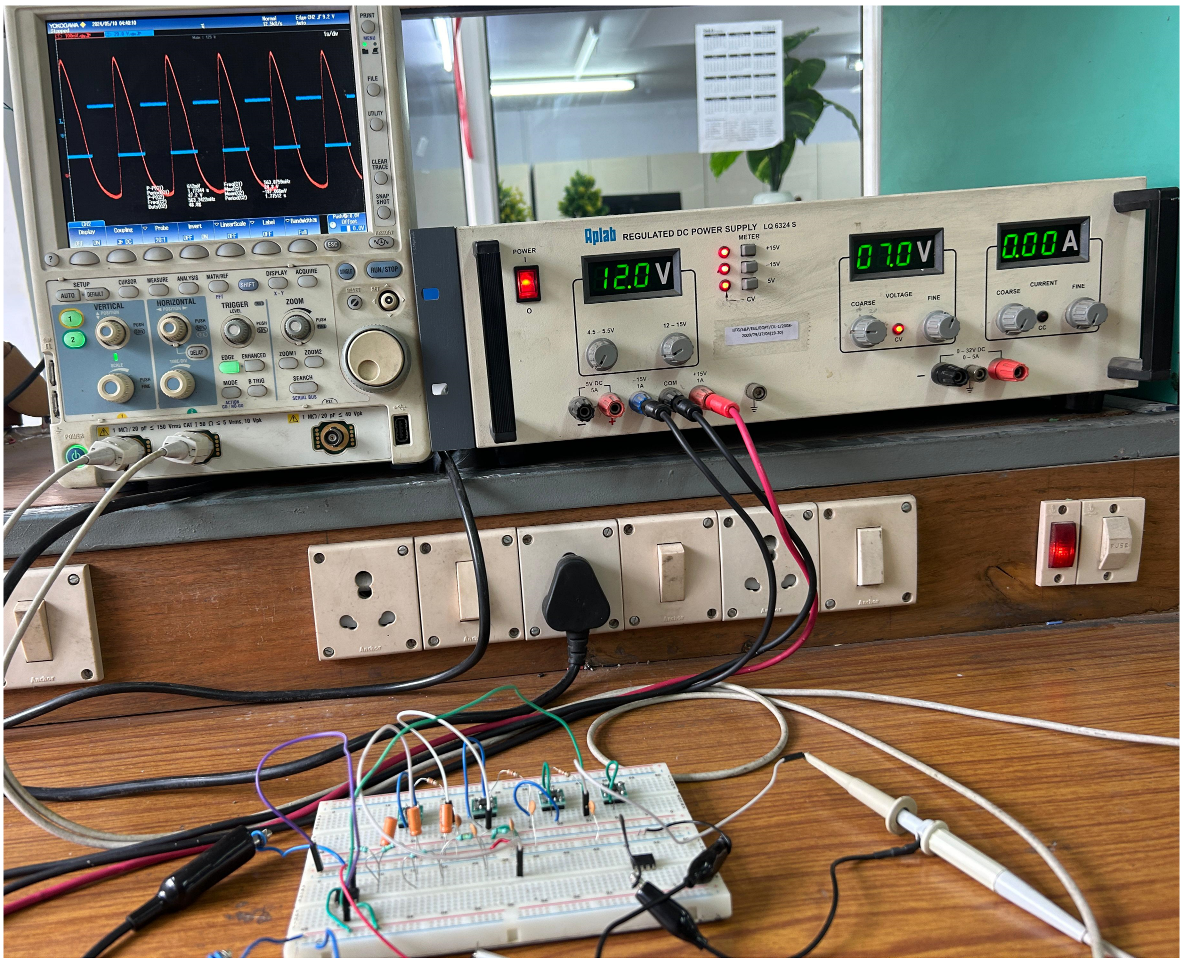


Figure 4.8: Final hardware circuit implementation setup

Chapter 5

Improved Autotune Using Machine Learning Algorithm

In this research, describing function-based analytical expressions are derived by approximation of relay gain, so there may be some error in estimated parameters. Here, optimization-based machine learning algorithms are used to predict accurate parameters of the unknown process model input parameters provided. Accurate identification of parameters simplifies the work of process engineers, especially when autotuning the controllers.

5.1 Input/Output Data Generation and Analysis

5.1.1 Data Generation

Data sets play a crucial role in machine learning. Typically, hardware is constructed to collect real-time data. To generate the real-time data, $\pm V_(sat)$ dc voltage supply is provided to the relay circuit op-amp, and negative feedback is given as input of the relay circuit op-amp. The output from this setup is a square wave signal, as depicted in Figure 5.1. A Square wave signal is given to the process model as input. Due to the delay in the circuit, a low-frequency sustained oscillatory output signal is generated, shown in Figure 5.2. The amplitude (A_p) and time period (T) of sustained oscillatory output are recorded to identify the process parameters. Considering the parameters relay height (h), amplitude (A_p), and time period (T) as input parameters, the parameters of the FOPDT transfer function model Delay (D) and time constant (T_1) are identified by using the analytical expressions.

To create a robust data set for machine learning training, it's essential to introduce variation in the process model. The variation is achieved by varying the process model components resistors and capacitors. Input-output data sets for FOPDT and SOPDT transfer function models are shown in Tables 5.1 and 5.2.

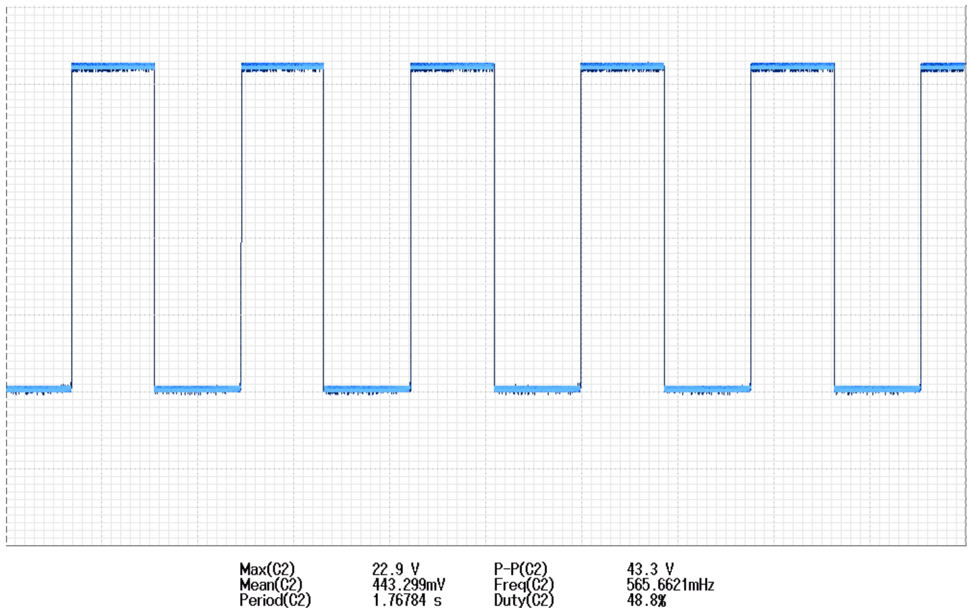


Figure 5.1: Hardware relay output signal

Table 5.1: Dataset for FOPDT transfer function model

$S.N.$	H	A_p	T	T_1	D
0	6.0	0.0655	2.5988	48.241712	0.653227
1	6.0	0.0830	2.7200	39.844859	0.684683
2	6.0	0.0960	2.8004	35.466772	0.705680
3	6.0	0.0840	2.8240	40.875800	0.710921
4	6.0	0.0625	2.8216	54.891868	0.709053
...	...	...	...	...	...
191	12.0	0.3530	18.4440	127.028622	4.678690
192	12.0	0.2600	20.3600	190.405133	5.144994
193	12.0	0.2770	21.5952	189.558470	5.460955
194	12.0	0.3115	22.1608	172.971370	5.611949
195	12.0	0.2635	23.1928	214.015430	5.861692

5.1.2 Data Analysis

Machine learning facilitates the learning process for machines by utilizing training data. The learning process begins with thorough data analysis and establishing the mathematical relationship between the input and output variables, which can take linear or nonlinear

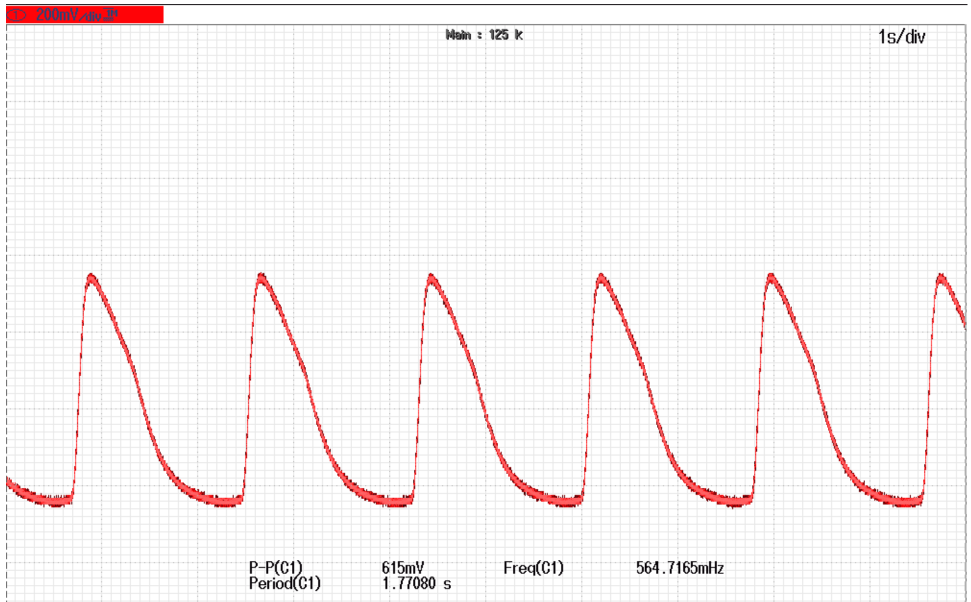


Figure 5.2: hardware process output signal

Table 5.2: Dataset for SOPDT transfer function model

$S.N.$	H	A_p	T	D	T_1
0	6.0	0.0655	2.5988	1.146022	0.996787
1	6.0	0.0830	2.7200	1.179217	1.068457
2	6.0	0.0960	2.8004	1.199967	1.118009
3	6.0	0.0840	2.8240	1.223173	1.110738
4	6.0	0.0625	2.8216	1.248142	1.077511
...	...	...	...	...	...
191	12.0	0.3530	18.4440	7.430553	8.004221
192	12.0	0.2600	20.3600	8.484617	8.443974
193	12.0	0.2770	21.5952	8.941145	9.035209
194	12.0	0.3115	22.1608	9.059374	9.431709
195	12.0	0.2635	23.1928	9.652096	9.636430

forms. Data statistics in machine learning involve analyzing and summarizing data to uncover patterns and insights essential for model training, as shown in Tables 5.3-5.4.

Table 5.3: Data statistics for FOPDT transfer function model

$S.N.$	H	A_p	T	T_1	D
$count$	196	196	196	196	196
$mean$	8.936224	0.268494	6.056677	61.837044	1.531448
std	2.221056	0.161311	6.097992	70.916846	1.538799
min	6.000000	0.047000	0.458960	2.228819	0.117130
25%	6.750000	0.121500	1.969300	10.336777	0.502197
50%	8.500000	0.229500	3.551400	35.762119	0.897597
75%	10.125000	0.392500	7.792500	73.763197	1.977233
max	12.000000	0.616000	23.656800	298.013355	5.979381

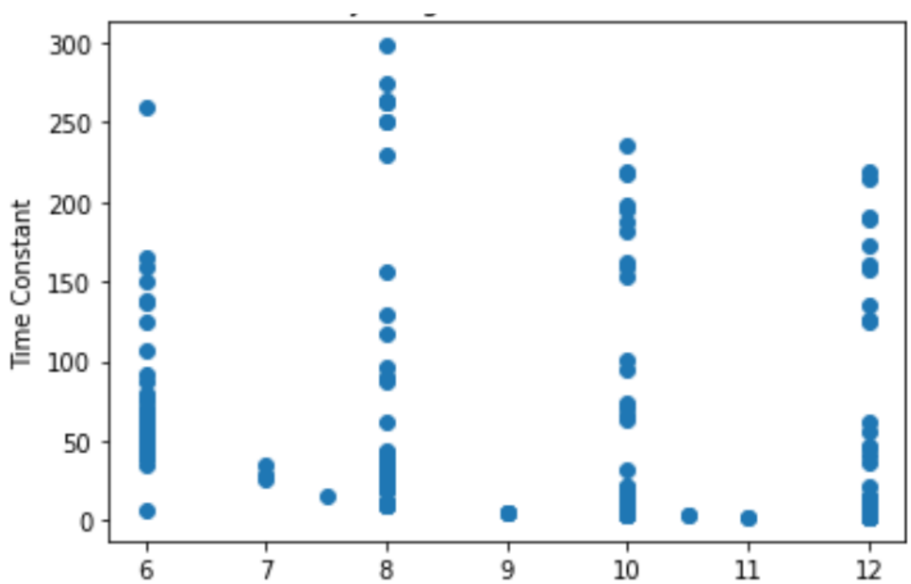


Figure 5.3: Relay height vs Time constant plot

Analyzing the graphical relationship between input and output features within the training dataset, as depicted in Figure 5.3-5.8, it becomes evident that a nonlinear relationship is observed in certain cases. Specifically, the relationship between time constant and amplitude in Figure 5.4 and between delay and amplitude in Figure 5.7 exhibits significant non-linearity. Time Period vs. Delay exhibits a linear relation, which means that as the delay in the system increases, the time period of sustained oscillatory output of the system also increases. Given these findings, it is reasonable to conclude that a linear regression model within machine learning performs poorly when applied to this dataset.

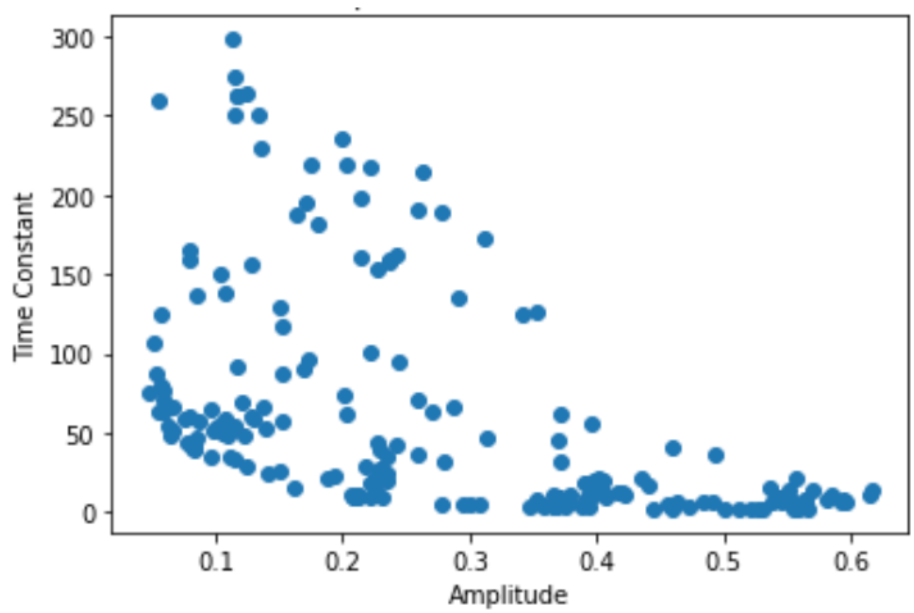


Figure 5.4: Amplitude vs Time constant plot

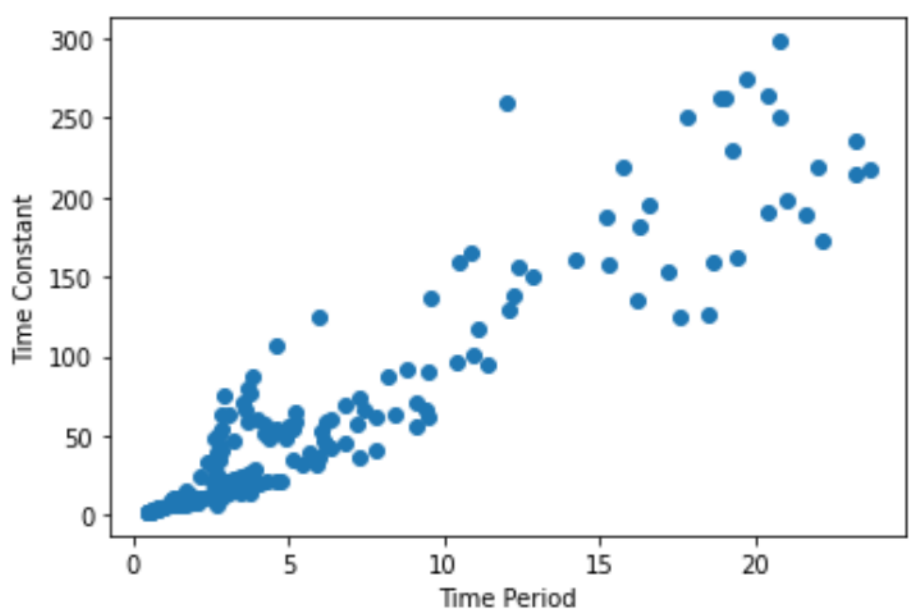


Figure 5.5: Time period vs Time constant plot

Table 5.4: Data statistics for SOPDT transfer function model

$S.N.$	H	A_p	T	D	T_1
$count$	196	196	196	196	196
$mean$	8.936224	0.268494	6.056677	2.519363	2.522897
std	2.221056	0.161311	6.097992	2.577443	2.486277
min	6.000000	0.047000	0.458960	0.176311	0.211973
25%	6.750000	0.121500	1.969300	0.760828	0.892959
50%	8.500000	0.229500	3.551400	1.401115	1.469920
75%	10.125000	0.392500	7.792500	3.104687	3.356139
max	12.000000	0.616000	23.656800	9.838772	9.837909

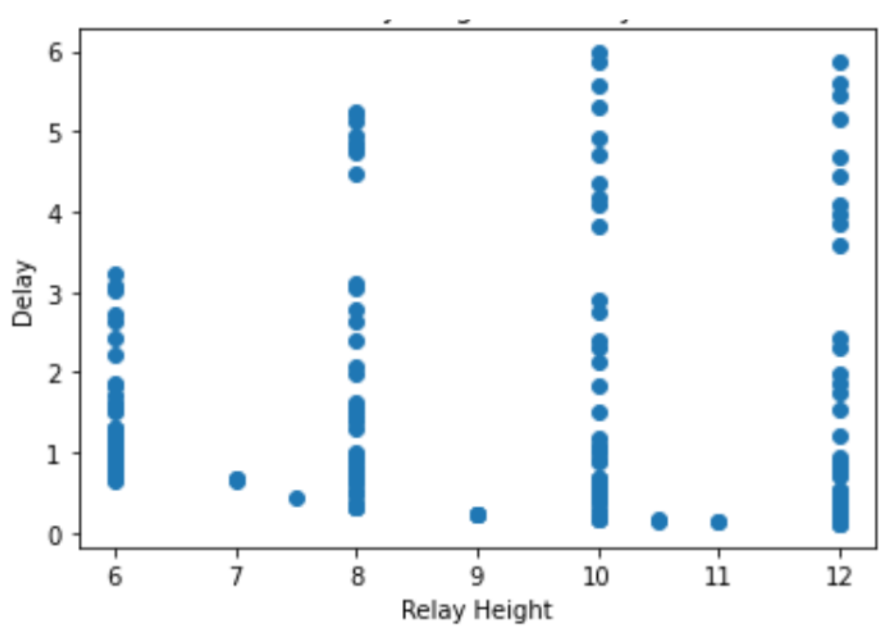


Figure 5.6: Relay height vs Delay plot

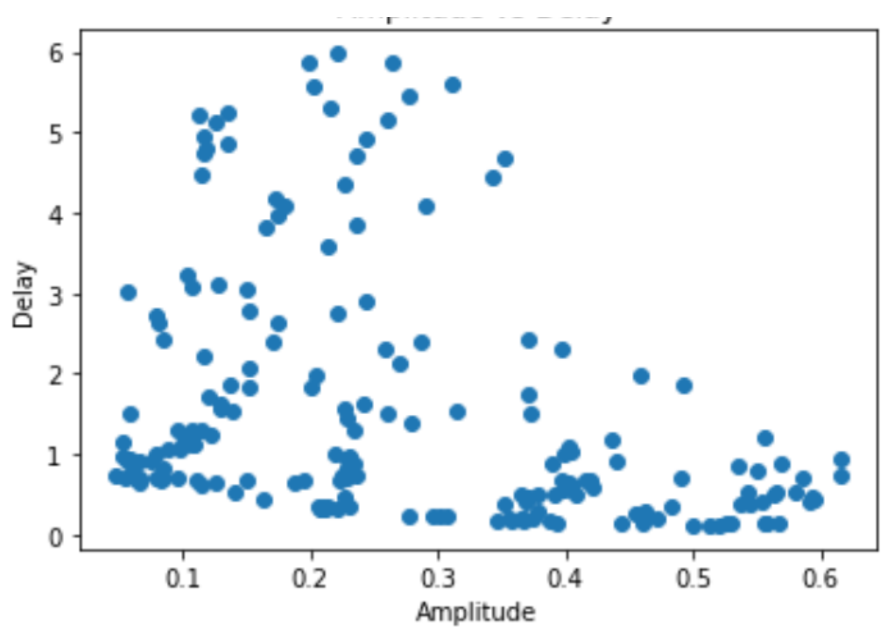


Figure 5.7: Amplitude vs Delay plot

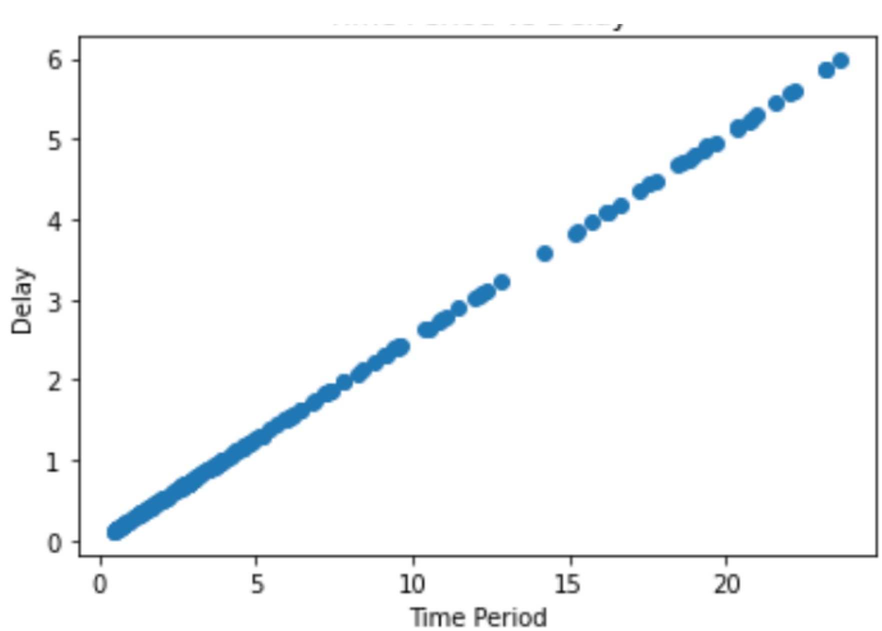


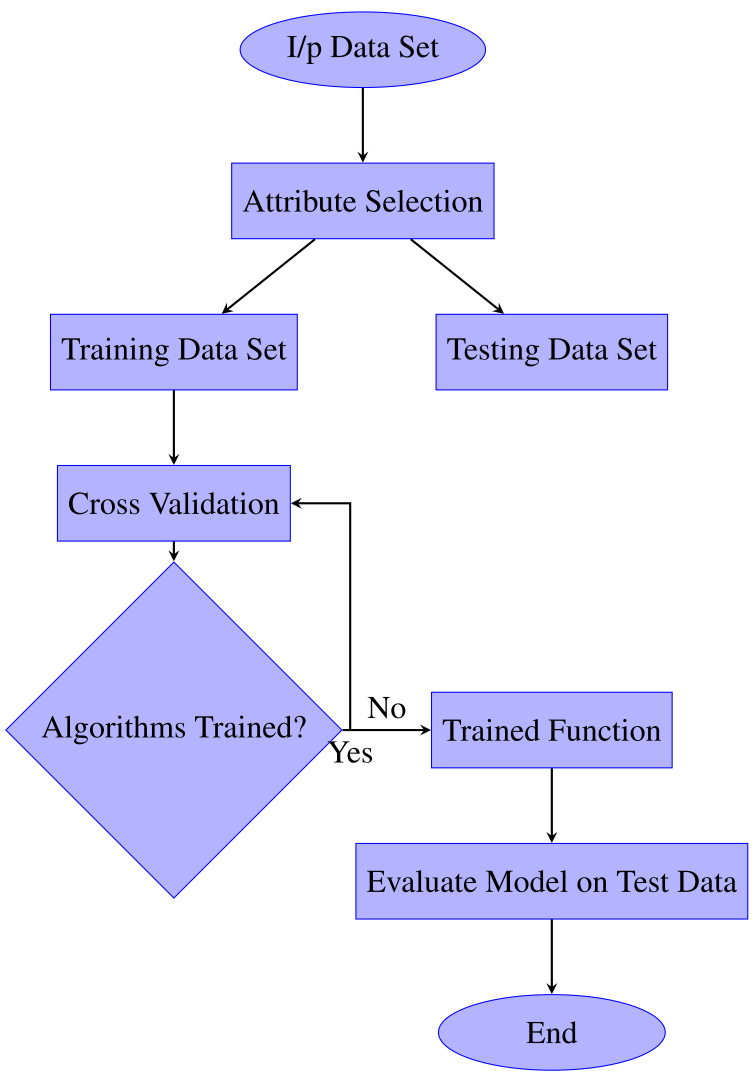
Figure 5.8: Time period vs Delay plot

Consequently, non-linear machine learning models are being utilized to predict the output parameters accurately.

5.2 Identification Procedure and Algorithm

Once the primary data set is prepared for training the machine learning model, various machine learning algorithms are trained, and their performance is evaluated. In this analysis, we found that different machine learning models showed varying levels of accuracy and how well they fit the data.

Identification Procedure



5.2.1 Machine Learning

Machine Learning (ML) is a branch of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to perform specific tasks without being explicitly programmed [16]. Instead of following hard-coded rules, ML systems learn patterns from data and make decisions or predictions based on that learning.

5.2.2 Supervised Machine Learning

Supervised machine learning is a type of machine learning where an algorithm learns from labeled training data to make predictions or decisions [17]. The labeled data consists of input-output pairs, where the input is a set of features, and the output is the corresponding label or value. The goal of supervised learning is to learn a mapping from inputs to outputs that can be used to predict the labels or values for new, unseen data. Here's a detailed overview of supervised machine learning.

Decision Tree Regression Model

A decision tree regressor (DTR) is a predictive model that estimates an output feature based on input parameters. It operates by partitioning the dataset into progressively smaller sub-datasets, aiming to minimize the standard deviation within each subset. This process, along with the fundamental terminology of the DTR model, is depicted in Figure 5.9.

The root node, containing the entire dataset, initiates the splitting process. From this primary node, the model splits into decision nodes, further dividing into sub-nodes. The splitting continues until reaching the leaf nodes, which represent the final decisions with no further division. In a DTR model, recursive partitioning starts at the root node, and nodes are split based on information gain. This gain is the difference between the impurity of the parent node and the combined impurities of its child nodes. The objective of the DTR model is to maximize information gain, ensuring that data becomes increasingly pure at each node.

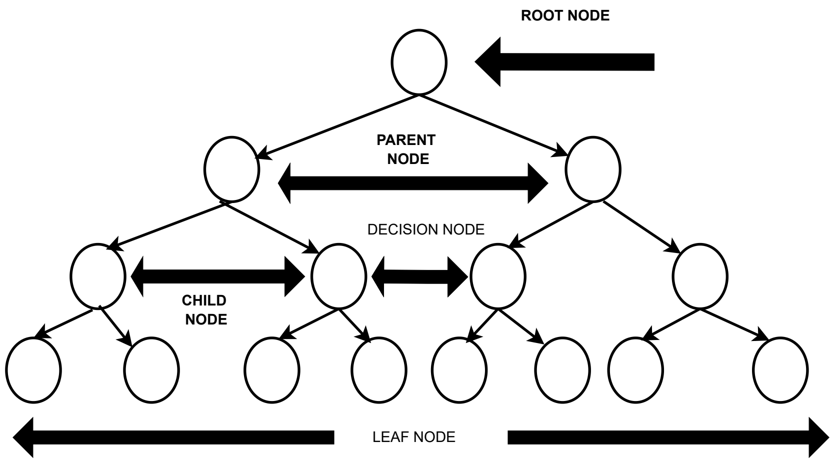


Figure 5.9: Decision Tree Regression model Terminology

Random Forest Regression Model

Random forest regression is an ensemble learning method that enhances predictive performance and mitigates overfitting by combining multiple decision trees, as shown in Figure 5.10. It operates on the principles of bagging (bootstrap aggregating), where each tree is trained on a randomly sampled subset of the data with replacement, and random feature selection, which selects a random subset of features at each split to ensure tree decorrelation. The final prediction is obtained by averaging the outputs of all individual trees, reducing overall model variance and improving accuracy. Additionally, the random forest provides out-of-bag (OOB) error estimation using data not included in the bootstrapped samples, offering an unbiased performance evaluation. It also ranks feature importance by measuring each feature's contribution to reducing impurity, aiding in feature selection and data understanding.

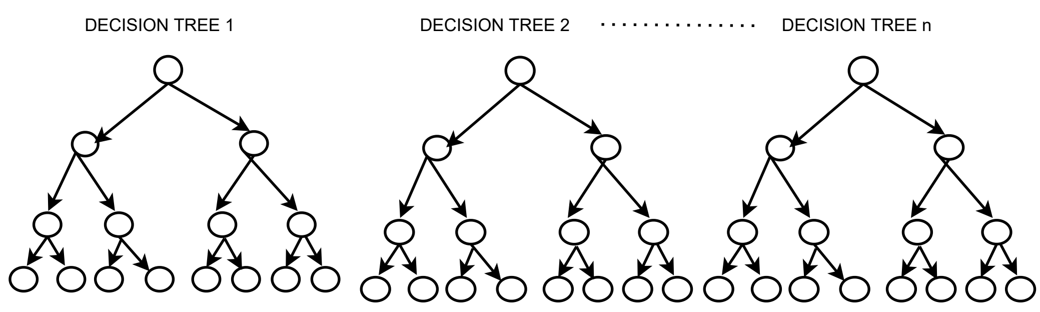


Figure 5.10: Random Forest Regression Model Terminology

Support Vector Regression (SVR)

Support Vector Regression (SVR) is an extension of Support Vector Machines (SVM) for regression tasks. It is used to predict continuous values rather than discrete labels. SVR aims to find a function that approximates the relationship between the input features and the continuous target variable while maintaining a tolerance margin (epsilon) around the predicted values. The SVR model introduces an epsilon (ϵ) parameter, defining a margin of tolerance within which errors are ignored. The objective is to ensure the predictions fall within this margin for as many data points as possible. SVR minimizes a loss function that penalizes predictions outside the epsilon-insensitive tube. The loss function includes terms for both the margin violations and the complexity of the model.

Extreme Gradient Boosting(XGBoost)

Extreme Gradient Boosting (XGBoost) is a leading ensemble learning algorithm known for its exceptional performance, efficiency, and versatility. Leveraging the power of gradient boosting, XGBoost optimizes model training through parallel computing, tree pruning, and regularization techniques like L1 and L2 regularization to prevent overfitting and enhance generalization. Its ability to handle diverse optimization objectives, support various evaluation metrics, provide insights into feature importance, and scale seamlessly to large datasets has made it a popular choice in both machine learning competitions and real-world applications, making it a reliable tool for achieving state-of-the-art results across regression, classification, and ranking tasks.

Gradient Boosting Machines (GBM)

Gradient Boosting Machines (GBM) are a powerful class of machine learning algorithms that excel in predictive modeling tasks. GBM sequentially adds weak learners (often decision trees) to improve model accuracy. It trains each new learner to correct errors made by the previous ones, gradually refining predictions. Key features of GBM include its ability to handle complex relationships, handle both regression and classification tasks, and robustness against overfitting through techniques like tree pruning and regularization. GBM is widely used in various domains due to its high predictive accuracy and flexibility in tuning hyperparameters to optimize performance.

K-nearest neighbors(KNN)

K-Nearest Neighbors (KNN) is a simple and intuitive machine learning method used for both classification and regression tasks. With KNN, the prediction for a new data point relies on the majority class (for classification) or the average value (for regression) of its nearest k neighbors in the training set. The "k" signifies the number of neighbors considered, and the algorithm computes distances (like Euclidean or Manhattan) to gauge proximity. KNN is non-parametric, meaning it doesn't make assumptions about the data's underlying probability distribution, rendering it adaptable to various datasets. Nonetheless, it's prone to outlier influence and necessitates thoughtful selection of the k value for optimal performance. A more enhanced approach for automatic parameter tuning leverages Machine Learning Algorithm 30.

Chapter 6

Result and Discussion

Most of the models are performing well and showing almost the same R2 score, but some models have comparatively higher mean square error (MSE). The Random Forest Regressor (RFR) model is showing good accuracy and minimum mean square error when predicting the time constant parameter. The R2 scores of all models are shown in Figures 6.1, 6.3, 6.5, and 6.7. The R2 score is a statistical measure that indicates how well the data fits the respective model. The closer the R2 score is to one, the better the model is fitted. Mean square error (MSE) is calculated for the output features, as shown in Figures 6.2, 6.4, 6.6, and 6.8. MSE provides a more comprehensive view of prediction accuracy by penalizing larger errors more heavily. Lower MSE values indicate better model performance. Some examples are taken below to show the efficacy of proposed machine learning models.

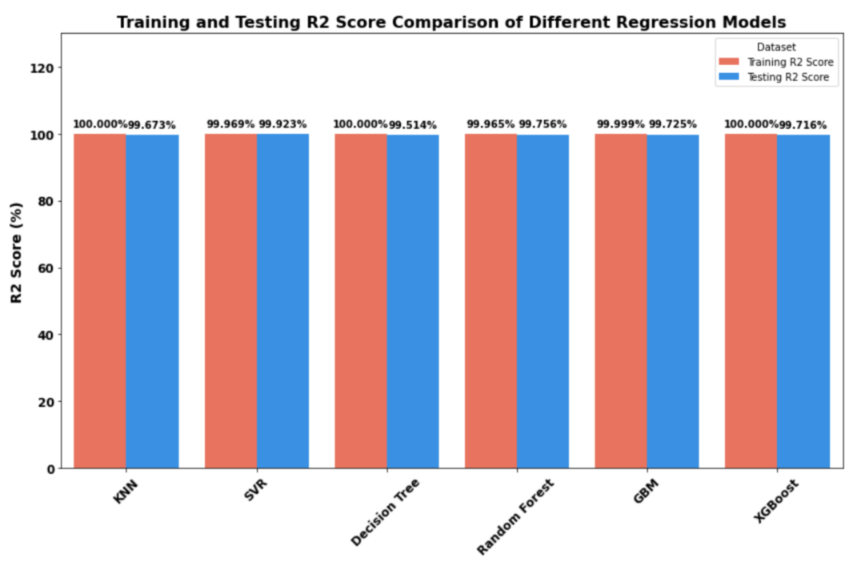


Figure 6.1: R2 score barplot for delay of FOPDT model

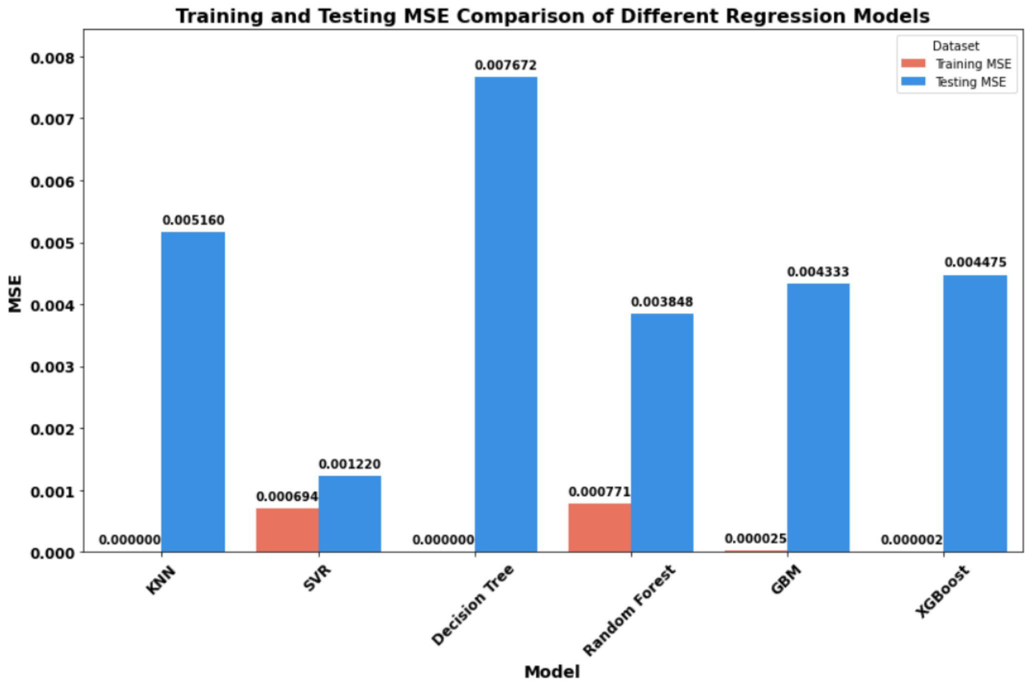


Figure 6.2: MSE barplot for Delay of FOPDT model

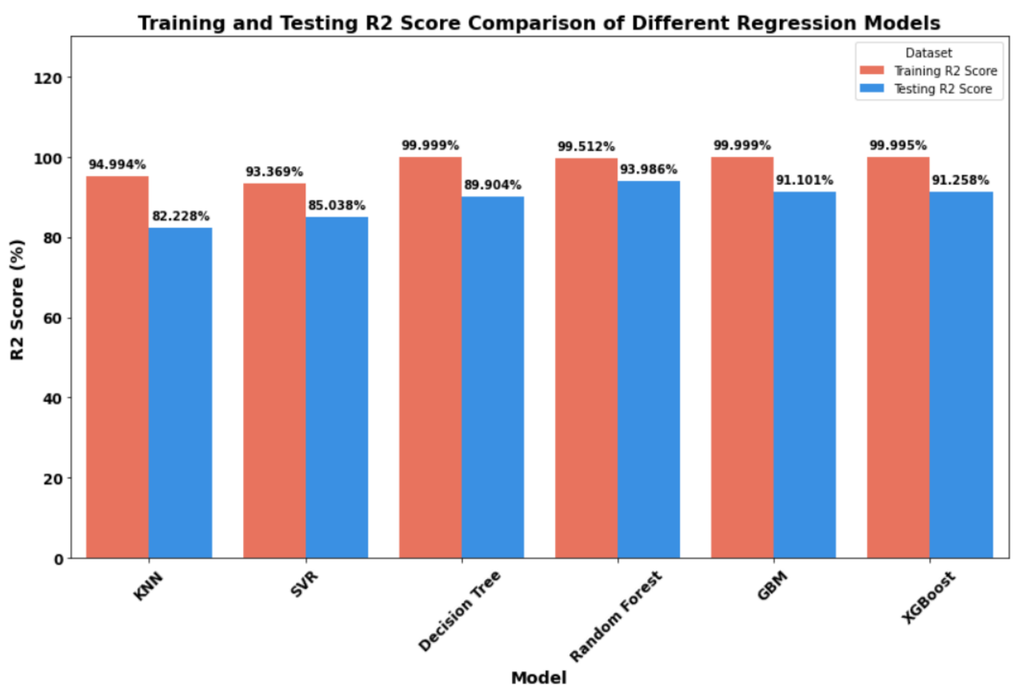


Figure 6.3: R2 score barplot for Time Constant of FOPDT model

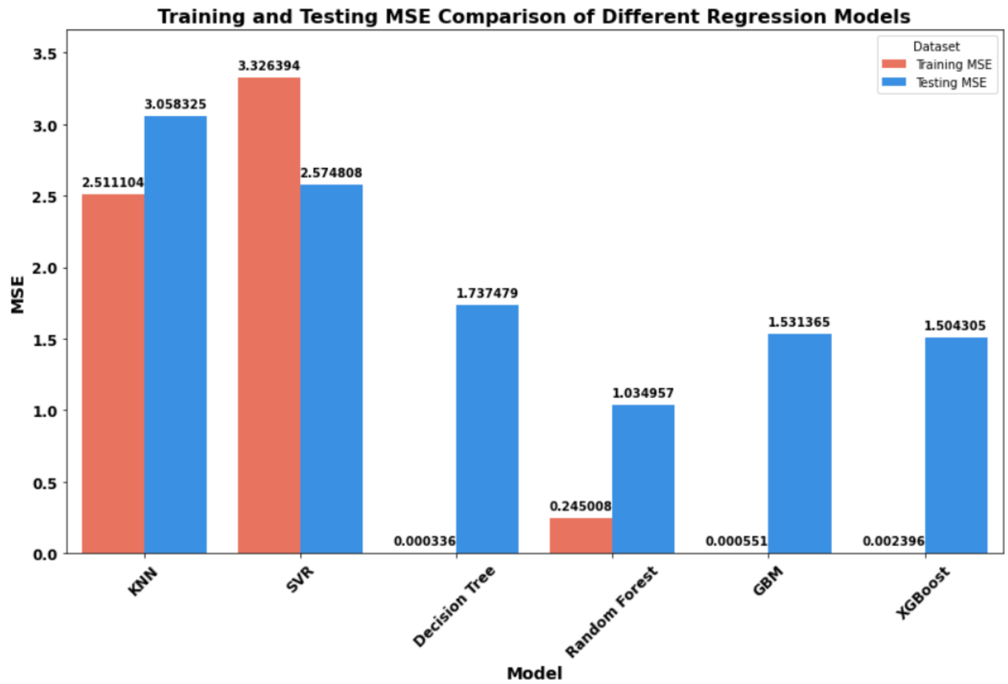


Figure 6.4: MSE barplot for Time Constant of FOPDT model

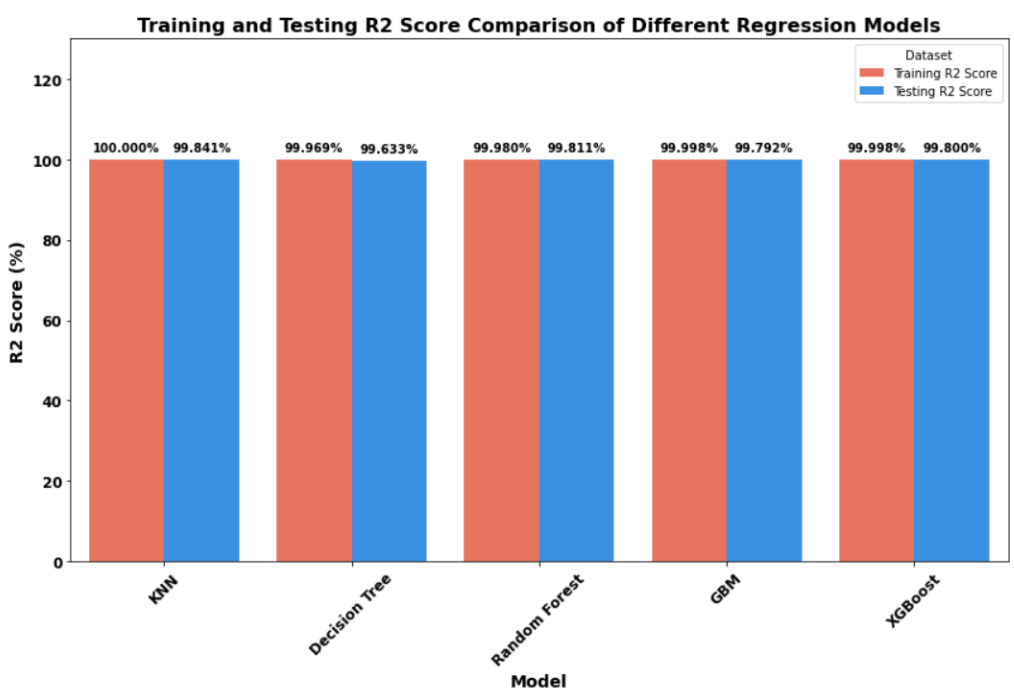


Figure 6.5: R2 score barplot for delay of SOPDT model

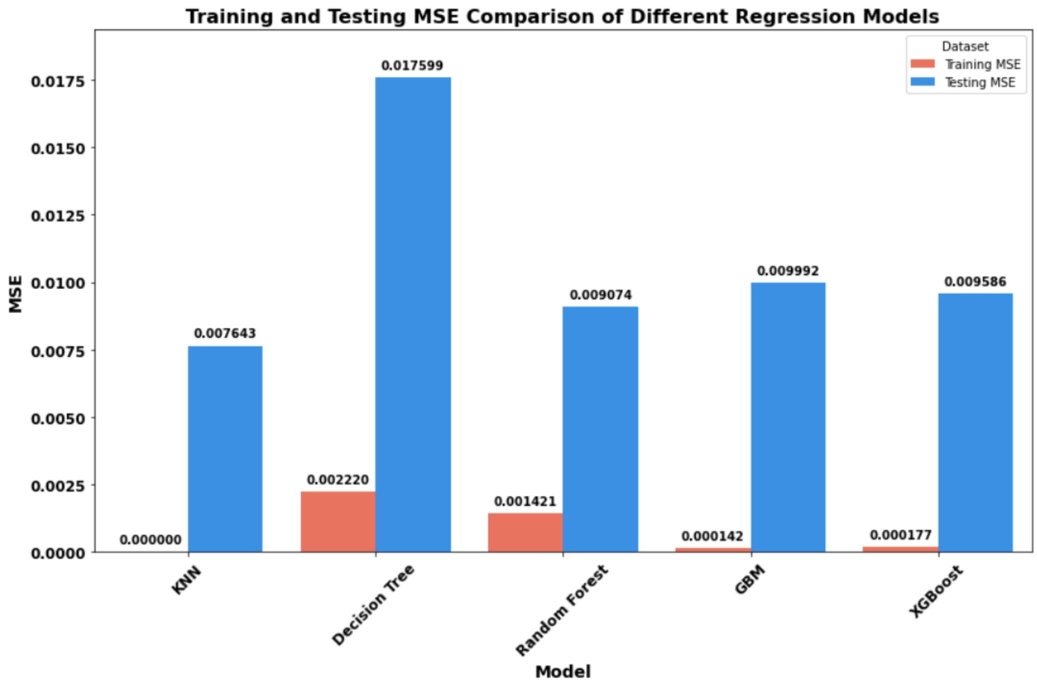


Figure 6.6: MSE barplot for delay of SOPDT model

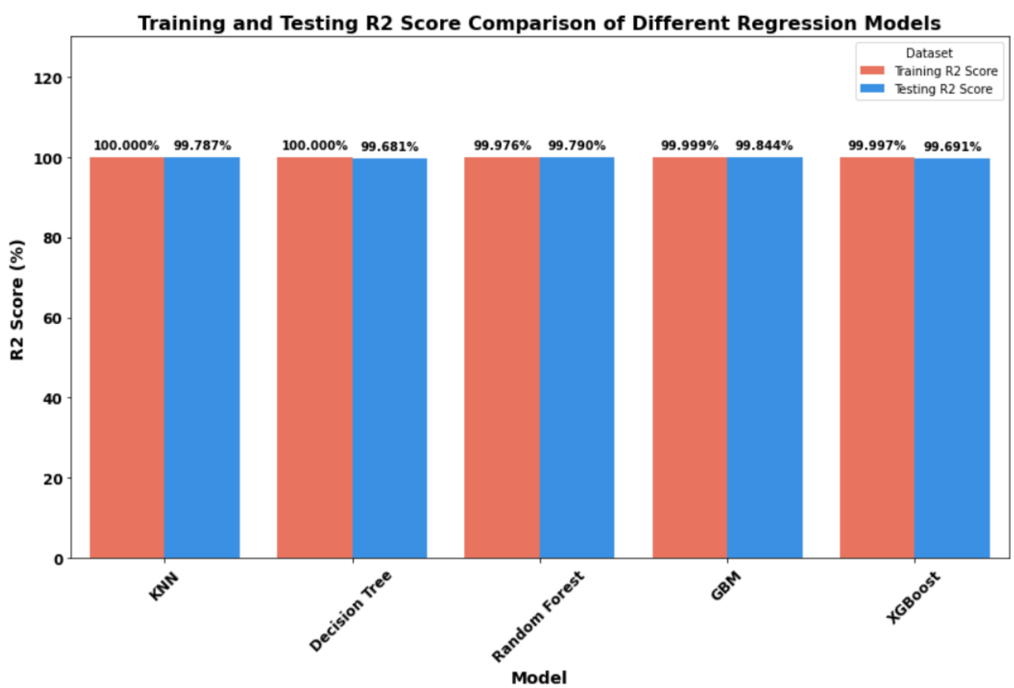


Figure 6.7: R2 score barplot for Time Constant of SOPDT model

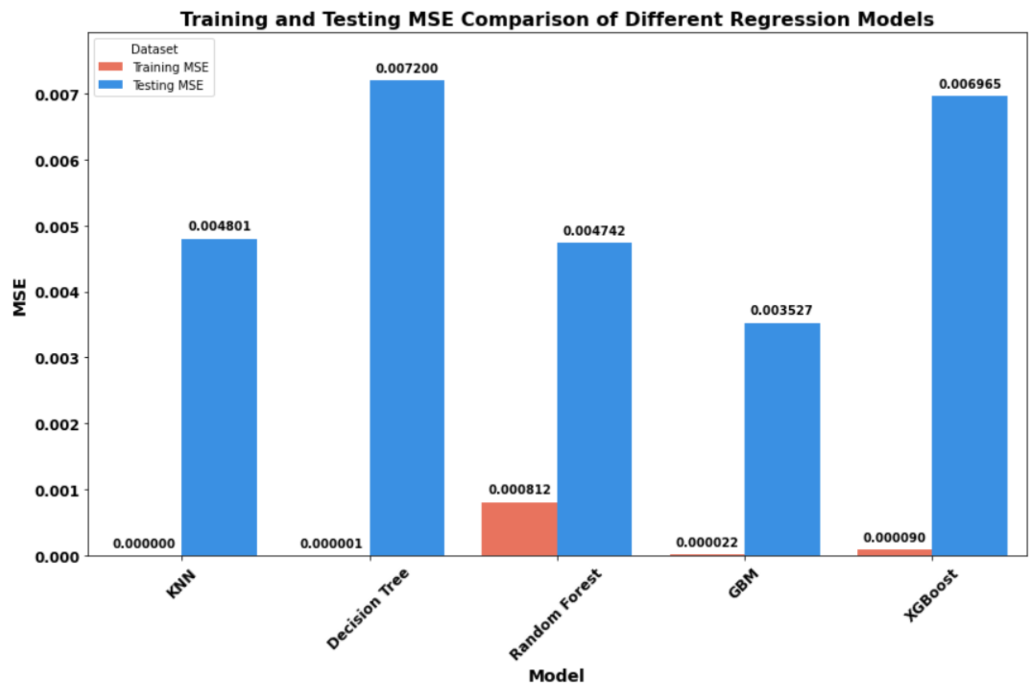


Figure 6.8: MSE barplot for Time Constant of SOPDT model

6.1 Example 1

Let us consider a real-time system which has a transfer function

$$G(s) = \frac{1}{(0.1s+1)^3(0.001s+1)^5} \tag{6.1}$$

Generates sustained oscillatory limit cycle output from an experiment with the parameters $A = 0.162$ and $T = 2.1564$ by setting relay height $h = 7$.

A stable FOPDT process model transfer function is identified using a relay feedback test as

$$G_1(s) = \frac{e^{-1.017s}}{18.879s+1} \tag{6.2}$$

Using the proposed machine learning model, the estimated FOPDT transfer function is

$$G_2(s) = \frac{e^{-1.2101s}}{18.2042s+1} \tag{6.3}$$

The Nyquist plots of the actual and identified transfer functions are shown in Figure 6.9 to illustrate the accuracy of the identification methods.

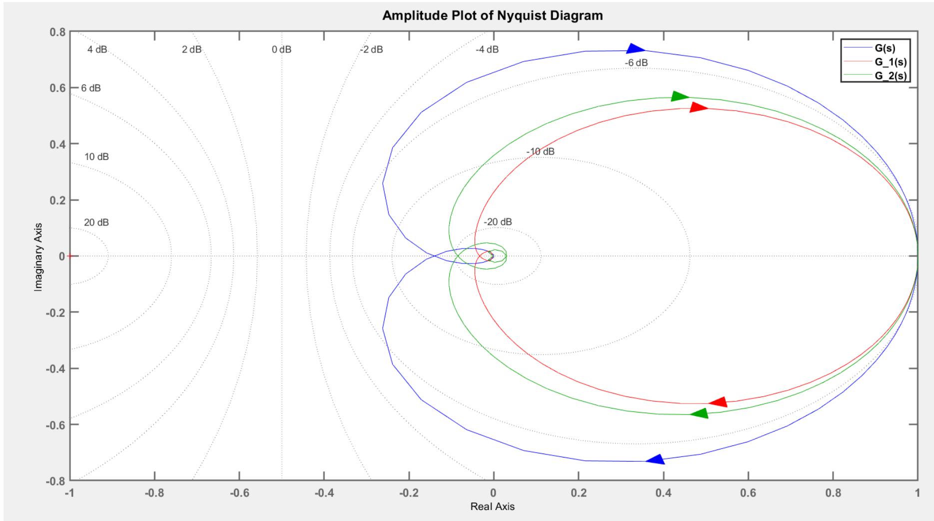


Figure 6.9: Nyquist plot of the actual and identified plants for Example 1

6.2 Example 2

Again, consider the real-time system transfer function $G(s)$ in Example 1. Now $G(s)$ is estimated as a SOPDT transfer function with the same sustained oscillatory limit cycle output from an experiment with the parameters $A = 0.162$ and $T = 2.1564$ by setting relay height $h = 7$.

A stable SOPDT process model transfer function is identified using a relay feedback test as

$$G_1(s) = \frac{e^{-0.8925s}}{(0.9025s + 1)^2} \tag{6.4}$$

Using the proposed machine learning model, the estimated SOPDT transfer function is

$$G_2(s) = \frac{e^{-0.8221s}}{(1.0069s + 1)^2} \tag{6.5}$$

The Nyquist plots of the actual and identified transfer functions are shown in Figure 6.10 to illustrate the accuracy of the identification methods.

6.3 Example 3

Let us consider another real-time system that has a transfer function

$$G(s) = \frac{1}{(0.1s + 1)^5(0.001s + 1)^3} \tag{6.6}$$

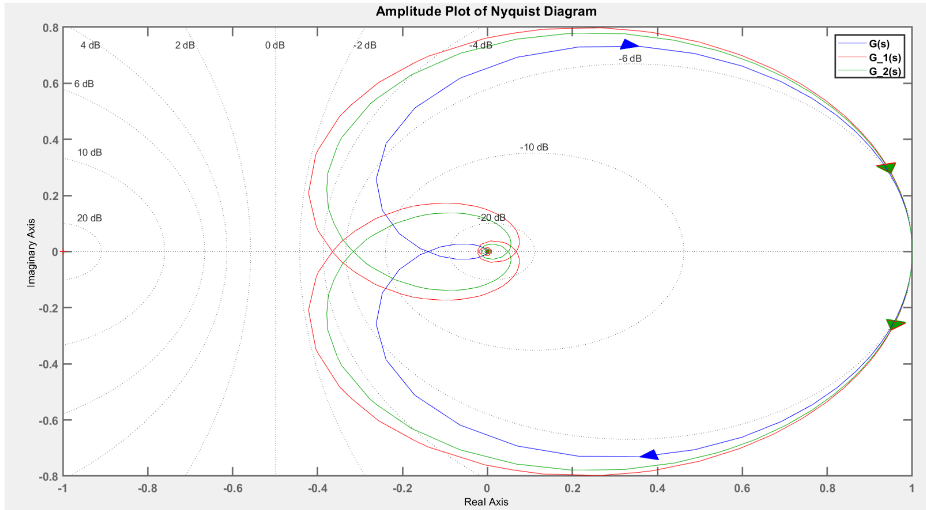


Figure 6.10: Nyquist plot of the actual and identified plants for Example 2

Generates sustained oscillatory limit cycle output from an experiment with the parameters $A = 0.5135$ and $T = 2.9002$ by setting relay height $h = 11$.

A stable FOPDT process model transfer function is identified using a relay feedback test as

$$G_1(s) = \frac{e^{-1.4134s}}{12.5818s + 1} \tag{6.7}$$

Using the proposed machine learning model, the estimated FOPDT transfer function is

$$G_2(s) = \frac{e^{-2.1965s}}{13.5152s + 1} \tag{6.8}$$

The Nyquist plots of the actual and identified transfer functions are shown in Figure 6.11 to illustrate the accuracy of the identification methods.

6.4 Example 4

Again, consider the real-time system transfer function $G(s)$ in Example 3. Now $G(s)$ is estimated as a SOPDT transfer function with the same sustained oscillatory limit cycle output from an experiment with the parameters $A = 0.5135$ and $T = 2.9002$ by setting relay height $h = 11$.

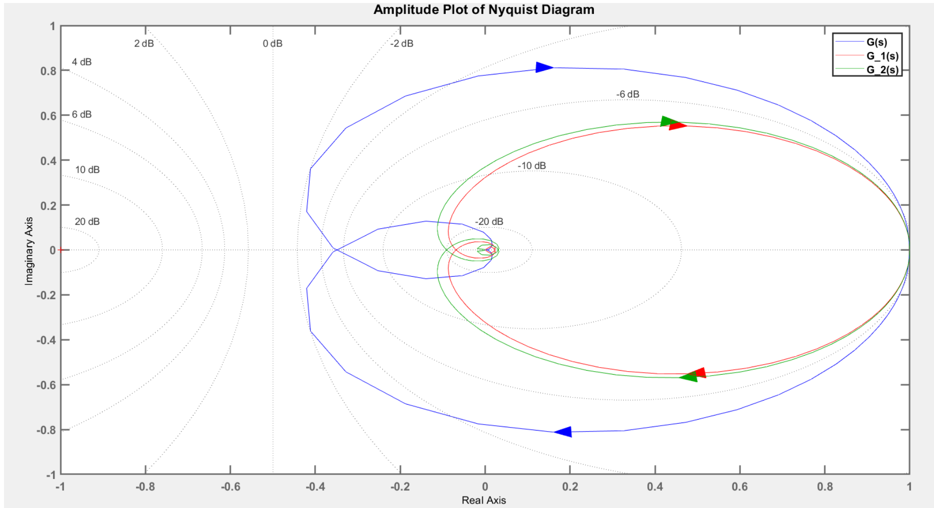


Figure 6.11: Nyquist plot of the actual and identified plants for Example 3

A stable SOPDT process model transfer function is identified using a relay feedback test as

$$G_1(s) = \frac{e^{-1.0944s}}{(1.3705s+1)^2} \tag{6.9}$$

Using the proposed machine learning model, the estimated SOPDT transfer function is

$$G_2(s) = \frac{e^{-1.1739s}}{(1.3344s+1)^2} \tag{6.10}$$

The Nyquist plots of the actual and identified transfer functions are shown in Figure 6.12 to illustrate the accuracy of the identification methods.

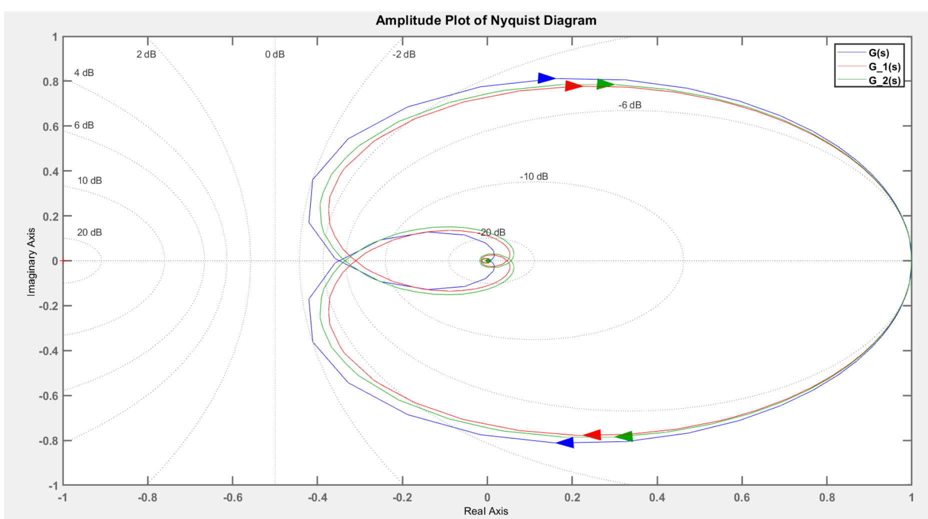


Figure 6.12: Nyquist plot of the actual and identified plants for Example 4

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

A machine learning-based identification method is introduced to identify the complex higher-order systems as FOPDT and SOPDT tansfer function models with reduced identification error and simple experiments without mathematical computation. A hardware model is constructed with physical delay, and real-time data have been collected to make the proposed method more accurate for real-time system identifications and autotune the controllers precisely. The Nyquist plots of the actual and identified transfer functions are plotted to show the accuracy of the proposed identification technique. Finally, it concludes that the proposed method provides better estimation accuracy with real-time data.

7.2 Future Scope

In this research thesis, parameter identification of First Order Plus Dead Time (FOPDT) and Second Order Plus Dead Time (SOPDT) process models using machine learning algorithms is conducted for a hardware process model that includes physical delay in the system. This research work can be extended to autotune PID controllers by designing a hardware model for real-time experience. The further approach involves identifying the process model and autotuning the controllers using machine-learning algorithms to tune PID controllers online effectively.

References

[1] S. Majhi, *Advanced control theory: a Relay feedback approach*. Cengage Learning Asia, 2009.

[2] Bajarangbali and S. Majhi, "Bajrangbali and s. majhi, "relay based identification of systems,"." *International Journal of Scientific and Engineering Research*, vol. 3, 2000.

[3] K. J. Åström and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol. 20, no. 5, 1984.

[4] P. Padhy, S. Majhi, and D. Atherton, "Pid-p controller for tito systems," *IFAC Proceedings Volumes*, vol. 38, no. 1, 2005.

[5] S. Sharma, B. Verma, R. Trivedi, and P. K. Padhy, "Identification of stable fopdt process parameters usingneural networks," in *2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*. IEEE, 2018.

[6] A. Kumar and P. K. Padhy, "Relay based identification for fopdt system & comparison on measurement accuracy & sensitivity analysis," in *2021 1st International Conference on Power Electronics and Energy (ICPEE)*. IEEE, 2021.

[7] R. C. Chang, S. H. Shen, and C. C. Yu, "Derivation of transfer function from relay feedback systems," *Industrial & engineering chemistry research*, vol. 31, no. 3, 1992.

[8] W. L. Luyben, "Derivation of transfer functions for highly nonlinear distillation columns," *Industrial & Engineering Chemistry Research*, vol. 26, no. 12, 1987.

[9] K. Lavanya, B. Umamaheswari, and R. C. Panda, "Identification of second order plus dead time systems using relay feedback test," *Indian Chemical Engineer*, vol. 48, no. 2, 2006.

[10] J. Lee, J.-S. Kim, J. Byeon, S. W. Sung, and T. F. Edgar, "Relay feedback identification for processes under drift and noisy environments," *AIChE journal*, vol. 57, no. 7, 2011.

[11] S. Majhi and D. Atherton, "Autotuning and controller design for processes with small time delays," *IEE Proceedings-Control Theory and Applications*, vol. 146, no. 5, 1999.

[12] K. S. Kula, "Online sopdt model identification method using a relay," *Applied Sciences*, vol. 13, no. 1, 2023.

[13] A. Peringal, M. Chehadeh, I. Boiko, and Y. Zweiri, "Relay-based identification of aerodynamic and delay sensor dynamics with applications for unmanned aerial vehicles," *IEEE Sensors Journal*, 2024.

[14] A. Ayyad, M. Chehadeh, P. H. Silva, M. Wahbah, O. A. Hay, I. Boiko, and Y. Zweiri, "Multirotors from takeoff to real-time full identification using the modified relay feedback test and deep neural networks," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, 2021.

[15] R. L. Boylestad and L. Nashelsky, *Electronic devices and circuit theory*. Pearson Education India, 2009.

[16] A. Ng, "Cs229 lecture notes," *CS229 Lecture notes*, vol. 1, no. 1, 2000.

[17] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.