

MUHAMMAD NOUMAN

DESIGN OF NONLINEAR PID CONTROLLER FOR DRILLING RIG WITH
ROBUST AND ADAPTIVE CONTROL FOR ALL OPERATING MODES

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

MUHAMMAD NOUMAN

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2022

ATILIM UNIVERSITY 2022

DESIGN OF NONLINEAR PID CONTROLLER FOR DRILLING RIG WITH
ROBUST AND ADAPTIVE CONTROL FOR ALL OPERATING MODES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

MUHAMMAD NOUMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2022

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

Prof. Dr. Ender KESKİNKILIÇ
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Electrical and Electronics Engineering Atılım University**.

Prof. Dr. Reşat Özgür DORUK
Head of Department

This is to certify that we have read the thesis DESIGN OF NONLINEAR PID CONTROLLER FOR DRILLING RIG WITH ROBUST AND ADAPTIVE CONTROL FOR ALL OPERATING MODES submitted by MUHAMMAD NOUMAN and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Mehmet Efe Özbek
Supervisor

Examining Committee Members:

Asst. Prof. Dr. Yaser DALVEREN
Electrical and Electronics Engineering, Atılım University _____

Asst. Prof. Dr. Mehmet Efe Özbek
Electrical and Electronics Engineering, Atılım University _____

Assoc. Prof. ENVER ÇAVUŞ
Electronics and Communication Engineering,
Yildirim Beyazıt University _____

Date: 02/09/2022

All facts, knowledge, and information in this work have been collected, processed, and presented in conformity with academic principles and ethical behavior. All non-original material and findings have been adequately cited and referenced in accordance with these rules of behavior.

Name, Last name: Muhammad Nouman

Signature:

ABSTRACT

DESIGN OF NONLINEAR PID CONTROLLER FOR DRILLING RIG WITH ROBUST AND ADAPTIVE CONTROL FOR ALL OPERATING MODES

NOUMAN, MUHAMMAD

M.S., Electrical and Electronic Engineering Department

Supervisor: Asst. Prof. Dr. Mehmet Efe Özbek

September 2022, 59 pages

Drilling towers have different operating modes during a real operation like drilling, tripping etc. Each mode of operation has certain external disturbances and uncertainties. In this study the hardware and software structures of the system are presented. The drilling rig is modelled in MATLAB/SIMULINK and interfacing with TwinCAT-3 to observe the dynamics of the drill string hook, while hoisting and lowering by using real time external mode of MATLAB/SIMULINK. The optimum values of rise time and overshoot of velocity have been analyzed. Various tuning methods were applied but their results were not satisfactory due to distortions and vibrations. This comparison is based on manual tuning of linear and nonlinear PID parameters. By using the nonlinear model for modes of the operation, robust and or adaptive control systems are designed. The smoothness of the system is achieved by using nonlinear pid parameters. The study presents the design process of the controller and evaluates the performance of the proposed control system to track the reference signal and reject the uncertainties and external disturbances. The behavior of the controllers and their stabilities are studied during the kinetics of drill sting hook and tripping operation on prototype setup in the laboratory.

Keywords- Matlab/Simulink, TwinCAT 3, Nonlinear PID controller, Linear PID controller, Manual Tuning, Stability.

ÖZ

TÜM ÇALIŞMA MODLARI İÇİN SAĞLAM VE UYARLANABİLİR KONTROLLÜ SONDAJ TEÇİSİ İÇİN DOĞRUSAL OLMAYAN PID KONTROL CİHAZI TASARIMI

NOUMAN, MUHAMMAD

Master, Elektrik Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Dr.Öğr.Üyesi Mehmet Efe ÖZBEK

Eylül 2020, 59 sayfa

Delme kuleleri, delme, açma vb. gibi gerçek bir operasyon sırasında farklı çalışma modlarına sahiptir. Her çalışma modunun belirli harici rahatsızlıkları ve belirsizlikleri vardır. Bu çalışmada sistemin donanım ve yazılım yapıları sunulmuştur. Sondaj kulesi MATLAB/SIMULINK'te modellenmiştir ve MATLAB/SIMULINK'in gerçek zamanlı harici modunu kullanarak kaldırma ve indirme sırasında sondaj kancasının dinamiklerini gözlemlemek için TwinCAT-3 ile arabirim oluşturur. Yükselme süresi ve hız aşımının optimum değerleri analiz edilmiştir. Çeşitli akort yöntemleri uygulanmış ancak bozulmalar ve titreşimler nedeniyle sonuçları tatmin edici olmamıştır. Bu karşılaştırma, doğrusal ve doğrusal olmayan PID parametrelerinin manuel olarak ayarlanmasına dayanmaktadır. Çalışma modları için doğrusal olmayan model kullanılarak sağlam ve/veya uyarlanabilir kontrol sistemleri tasarlanmıştır. Sistemin düzgünlüğü, doğrusal olmayan pid parametreleri kullanılarak elde edilir. Çalışma, kontrolörün tasarım sürecini sunar ve referans sinyalinin izlemek ve belirsizlikleri ve harici bozucuları reddetmek için önerilen kontrol sisteminin performansını değerlendirir. Laboratuarda prototip kurulumunda delme kancası ve açma işleminin kinetiği sırasında kontrolörlerin davranışları ve kararlılıkları incelenmiştir.

Anahtar Kelimeler- Matlab/Simulink, TwinCAT 3, Doğrusal olmayan PID denetleyicisi, Doğrusal PID denetleyicisi, Manuel Ayarlama, istikrar

*To My Parents, Family, and Friends
you will always be my source of strength*

ACKNOWLEDGMENTS

I am grateful to my supervisor, Assist. Prof. Dr. Mehmet Efe Ozbek, for his support and guidance during the project. I also want to express my gratitude to my family for their continuous encouragement throughout this time.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ.....	iv
ACKNOWLEDGMENTS	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS.....	xii
CHAPTER 1	1
INTRODUCTION AND BACKGROUND	1
1.1 Aim of Thesis.....	3
CHAPTER 2	4
LITERATURE REVIEW	4
2.1 The Hardware Architecture:	9
CHAPTER 3	10
METHODOLOGY AND DESIGN DESCRIPTION	10
3.1 Simulation for Motor Speed Control:	13
3.2 Code Generation:	17
3.3 TE1400 TwinCAT Target for MATLAB® /Simulink®	17
3.4 Data exchange (Tc Interfaces)	21
3.5 External mode (Tc External Mode)	22
3.6 Establish connection from Simulink®.....	23
3.7 Advanced settings (Tc Advanced).....	23
3.8 TC BUILD	24
3.9 Development of the PLC Programming Algorithm.....	25
3.10 Adding TCCom Object.....	28
3.11 Integration of the Module in TwinCAT 3.....	29
3.12 Adding Tasks	30
3.13 The block diagram in the Browser Parameters tab	31
3.14 Experimental Laboratory Setup	32
3.15 Actuator	32
3.16 Encoder	33
3.17 Load-Cell	33
3.18 Embedded PC	34
3.18.1 Characteristics.....	35
3.19 Motor Driver	36

3.20 Joystick	37
3.21 Experimental Setup in Laboratory	38
3.22 MATLAB FUNCTION BLOCK.....	39
CHAPTER 4	40
TEST AND RESULTS ANALYSIS	40
4.1 LINEAR CONTROLLER	40
4.2 THE PERFORMANCE OF THE LINEAR PID PARAMETERS	42
4.3 Controlling Speed by Motor Rotational Speed Control.....	42
4.3.1 Scenarios:.....	42
4.4 THE NONLINEAR CONTROL LAW	47
4.5 NONLINEAR CONTROLLER.....	49
4.6 THE PERFORMANCE OF THE NONLINEAR PID PARAMETERS	50
4.7 Controlling Speed by Motor Rotational Speed Control.....	50
4.7.1 Scenarios	50
4.7.2 When weight moving Upward and Downward direction	51
CHAPTER 5	55
CONCLUSION.....	55
REFERENCES	56

LIST OF TABLES

Table 4.1 Optimized values for Linear PID Controller in the Simulink.....	40
Table 4.2 Optimized values for Nonlinear PID Controller in the Simulink	49



LIST OF FIGURES

Figure 2.1 Hardware Implementation	9
Figure 3.1 State-flow Model	10
Figure 3.2 Deadman State-Flow Chart	11
Figure 3.3 Slope of Deadman Delaying Signal.....	12
Figure 3.4 State-flow for Motor Speed Control.....	13
Figure 3.5 Motor Driver Control Register	14
Figure 3.6 State-flow Code for Starting the Motor.....	15
Figure 3.7 Simulation Function for Tuning PID parameters	16
Figure 3.8 Setup TwinCAT Target	17
Figure 3.9 Code Generation	18
Figure 3.10 TwinCAT.tlc.....	19
Figure 3.11 TC BUILD	20
Figure 3.12 TC Interface.....	21
Figure 3.13 Tc External Mode	22
Figure 3.14 External Mode	23
Figure 3.15 Tc Advanced.....	23
Figure 3.16 TC build.....	24
Figure 3.17 TwinCAT Project	25
Figure 3.18 License.....	26
Figure 3.19 Create Task.....	27
Figure 3.20 Add Generated File in TwinCAT	28
Figure 3.21 Generated file seen on TwinCAT.....	29
Figure 3.22 Add Task in Generated file.....	30
Figure 3.23 Block Diagram Build.....	31
Figure 3.24 3-Phase AC Motor	32
Figure 3.25 Encoder	33
Figure 3.26 Load Cell	34
Figure 3.27 Embedded PC	34
Figure 3.28 EtherCAT Coupler and I/O Terminals	35
Figure 3.29 EtherCAT Mounted on the Embedded Computer	36

Figure 3.30 Motor Driver.....	36
Figure 3.31 Joystick i.e. We Used in Our Experimental Setup	37
Figure 3.32 Laboratory Setup	38
Figure 3.33 Function of Nonlinear PID Controller.....	39
Figure 3.34 MATLAB Function Block	39
Figure 4.1 Velocity Signal during Upward.....	44
Figure 4.2 Velocity Signal during Downward.....	44
Figure 4.3 Velocity Signal during Downward.....	45
Figure 4.4 Velocity Signal during upward.....	45
Figure 4.5 Velocity Signal during upward.....	46
Figure 4.6 Velocity Signal during Downward.....	46
Figure 4.7 Slope of gain Curve by changing Alpha.....	47
Figure 4.8 Linear and Nonlinear region.....	48
Figure 4.9 Velocity Signal during upward.....	51
Figure 4.10 Velocity Signal during Downward.....	51
Figure 4.11 Velocity Signal during upward.....	52
Figure 4.12 Velocity Signal during Downward.....	52
Figure 4.13 Velocity Signal during upward.....	53
Figure 4.14 Velocity Signal during Downward.....	53

LIST OF ABBREVIATIONS

AC	-	Alternating Current
DC	-	Direct Current
PID	-	Proportional Integral Derivative
NLPID	-	Nonlinear Proportional Integral Derivative
PLC	-	Programmable Logic Control
OPC	-	Object Linking and Embedding for Process Control
PWM	-	Pulse Width Modulation
I/O	-	Input/Output
BP	-	Back Propagation
DE	-	Differential Evolution
GE	-	Genetic Algorithm
MSE	-	Mean Square Error
NCD	-	Nonlinear Control Design
V/F	-	Voltage/Frequency
SQP	-	Sequential Quadratic Programming
QP	-	Quadratic Program
RBFNN	-	Radial Basis Function Neural Network
Twin CAT	-	The Window Control and Automation Technology
IDE	-	Integrated Development Environment
CSTR	-	Continuous Stirred Tank Reactor
EN-PID	-	Enhanced-Proportional Integrated Derivative
MBD	-	Model Base Design

CHAPTER 1

INTRODUCTION AND BACKGROUND

A machine that creates boreholes in the ground is called a drilling rig. The drilling machine is equipped with a variety of tools, including an actuator, drawworks, traveling block or hook, mud pump, top-drive, drill string, and bottom-hole assembly (BHA). For system analysis and control, these equipment's dynamic modeling serves as the foundation. To be used for analysis and control, the model must be able to accurately capture the behavior of the system during actual operating modes in wells. However, because external disturbances and/or model uncertainties have a significant impact on nonlinear control systems, robust and/or adaptive control architectures can be used to address model uncertainties. The operator inputs the vertical velocity reference through a joystick and can easily set the position reference to control the drill string hook position. The drill string rotates in both the vertical and downward directions during the drilling process. Additionally, certain external disturbances and uncertainties are present in each of the operating modes. [1]

Process control engineering as well as informational and operational technology must be incorporated in the drilling rig automation industry. Interoperability which necessitates data sharing is essential for attaining this goal on various systems. Drilling rig platforms are a key source of worry since they have a challenge in meeting the demands of today's automation sector. There was a complexity to gather data of these processes in one platform to communicate with networks. Because of the ease with which automation data may be accessed, the adoption of software-based automation systems in industrial automation has increased over the years. [2]

In industrial control application PID controllers was used over the years. It was initially designed in 1890s, although it was used for a long period of time, still used in different control application of the industries. Ninety percent of the industries are used this controller according to 1989 survey. In industry this controller was widely used and recognized due to their easiness, simplicity and user friendly. It was named as PID

controller because of the output of their system which was consisting of three terms.

- i) Proportional
- ii) Integral
- iii) Derivative

Proportional term accelerates the response as it lowers the constant close-loop time, but order of the system does not affect by this because the output of the system directly proportional to the input. Steady-state error also known as offset which cannot be eliminated but it can only be reduced by proportional term. Integral term eliminates the offset by increased the order of system by one and also increase its type. It will improve response-time of the system, but at a cost of prolonged oscillations. Oscillatory response of system was primarily decreased by derivative term. It does not effect on the offset of the system. Also, there is no effect on the system's type or order. [3]

During the past few decades, most important part of industry automation was “proportional derivative integral” controller also known as PID controller, it remains the most commonly used and essential process in control engineering. Due to high control performance requirements by the industry, it was considered necessary to used advanced control techniques to overcome these requirements. While ninety percent of control techniques which are applied in industry were interconnected to this controller. A little enhancement makes an immense impact on industries which was related to this technique. Therefore, different types of enhanced PID Controllers were designed and established for industrial control applications. In 1994 HAN introduces nonlinear proportional integral derivative controller (NPID). In industrial control processes NPID controller was used as major process due to limitation of PID controller in some cases. Therefore, important algorithms were designed and fundamental modifications in NPID controller were made along with experimental findings. The major purpose of these modifications was to achieve maximum efficiency of the plant output where PID controller could not give us desire response. As a result, “proportional derivative integral” controller has been recreated by using the following non-linear function:

$$n = K_p \psi(e_p, \alpha_p, \delta_p) + K_i \psi(e_i, \alpha_i, \delta_i) + K_d \psi(e_d, \alpha_d, \delta_d)$$

Where $\psi(x, \alpha, \delta)$ is the nonlinear function:

$$\psi(x, \alpha, \delta) = \begin{cases} |x|^\alpha \cdot \text{sign}(x), & \text{when } |x| > \delta \\ \delta^{\alpha-1} \cdot x, & \text{when } |x| \leq \delta \end{cases}$$

K_p , K_i , K_d are the controller gains, and they have the same meaning as the PID gains. [4]

1.1 Aim of Thesis

The primary goals are to evaluate the performance of the Manual Tuning of nonlinear PID parameters during tripping and drilling operations, compare them practically, and enhance substantial rig performance. To achieve this goal, first we modeled a prototype setup and then simulate in MATLAB/Simulink, and then build this on TwinCAT 3. After the building is accomplished, the NLPID parameters are tuned in the Simulink environment by using real-time external mode to monitor the drill string hook's vertical and downward speed during tripping and drilling activities. During the drilling/tripping operation on the prototype setup, investigated the controller's behavior and their stability, on the system while it is rising vertically with weight and the same when it is falling. The system behavior is analyzed during the operating mode. We hope that by enhancing performance efficiency in this controlled setting, we will be able to apply what we learn in this study to real-world operations. The objective is to make the operating process more accurate and smoother by reducing risks while maintaining safe field operations.

CHAPTER 2

LITERATURE REVIEW

Nowadays the previous signal wiring in industrial applications is replaced by modern industrial fieldbuses. The paper proposed method which includes designing a MATLAB toolbox for communication with TwinCAT without the use of OPC. The methods included the segments of Interconnection with Windows applications, interconnection with MATLAB and Simulink, use of ADS Toolbox, MEX files, S-functions, and many others. The limitations or research gaps in the research manuscript are focusing on that we have found out that through the OPC communication standard we are not able to read all the TwinCAT variables. To solve this problem, the authors had to choose another type of data exchange. If we use the TcAdsDll library, we can read and write all the variables and handle some other functions of the PLC. [5]

The use of MATLAB/Simulink with model-based programming rather of the C++ programming language in this work allows researchers in the haptic domain to focus on control engineering concerns rather than programming abilities. The technique in this article included the segments of EtherCAT and TwinCAT, Architecture of Hardware Implementation of a force feedback haptic system, Software Architecture, and a variety of additional tasks. The haptic control algorithms, which included kinematics, collision detection, force generation, force to torque transformation, and virtual coupling implementation, were created in the MATLAB/Simulink environment. The positive outcomes of the testing demonstrated that the haptic system works adequately for this application. [6]

The MATLAB/SIMULINK toolbox is utilized in this work to model the complete system, which consists of a DC motor powered by a DC source voltage through a chopper transistor circuit. The goal is to create & simulate a new control system that will automatically maintain the speed of the DC motor regardless of load fluctuations using the PID controller. The methods which are used in this research is PID. It's also important to note that manufacturers' PID controller equations aren't all the same. [7]

The suggested platform is based on the use of the Beckhoff programmable logic controller, which is implemented on a personal computer/laptop utilizing the TwinCAT software environment. This paper's methodologies covered parts of BUCK Converter, Beckhoff Programmable logic controller, Voltage Control of BUCK Converter with TwinCAT software Environment, Practical Implementation, and Experimental Results. In this research, the proposed experimental platform is utilized to achieve the voltage control of a DC-DC buck converter. The proposed platform may be expanded with additional I/O terminals and power modules, enabling for rapid control prototyping of increasingly complex power electronics systems. [8]

Ziegler Nicholas technique (Z-N), Modified Z N, IMC, TL and CHR techniques are used in this paper, for process smoothness and obtain high stability. The suggested experimental platform is employed in this research to compare the five tuning strategies, resulting in the identification of the optimal controller tuning method (C-H-R METHOD). Based on investigation, the Ziegler-Nicholas method-based PID controller is determined to be the optimum option. [9]

A review paper on studies, presents the classical approaches established for PID tuning and optimization strategies used for tuning. The methods in this paper included the segments of Ziegler Nichols Technique, Cohen Coon Method and as well as various intelligent optimization techniques have been used. The proposed Techniques is used to assess the relative performance of diverse approaches on a single platform, a complete comparison analysis of all techniques, evaluated simultaneously under different situations, is still required. [10]

The author of this article concentrated on a fractional order PID controller, a Back Propagation (BP) neural network, parameter optimization, vibration suppression,

spectrum analysis, and an AC motor. In this study, maximum performance was achieved by the optimization of fractional order controller's parameter, a Back Propagation (BP) neural network is used to change the parameters of the fractional order PID controller. The modified fractional order PID controller changed the stator current component, as well as the frequency and amplitude components of the motor's vibration output. This study concludes that, after optimization of fractional order controller it performs better than regular PID controller in terms of vibration suppression. [11]

The article [12] describes how to tune a PID controller using two recent heuristic techniques: Differential Evolution (DE) and a genetic algorithm (GA). The PID Control system's performance was tuned using GA and DE approaches. For a high order system, the ideal setting of PID controllers were utilized in which non-minimum phase system and time delay system were considered. However, DE is optimized by Mean Square Error for non-minimum phase systems. MSE outperforms in settling time, maximum overshoot, & undershoot.

MSC, NCD, RBFNN, PID Controller, and SQP are all addressed in this study. Nonlinear Control Design (NCD) is used in this study to determine the best solution and derive the transfer function of an unknown system using sequential quadratic programming. To demonstrate the possibility of an intelligent PID controller using a nonlinear system, this work provides a thorough design sequence. This work offers a case study of V/F speed control for an induction motor with testing noise as an external variable loading condition to prove that the self-tuning PID is more resilient than the standard PID. The case study findings show that a PID controller with well-tuned gains may efficiently overcome the disruption caused by a rapid external load. The proposed technology is extremely useful for accurate motion control and production. According to the findings of this study, they recommended intelligent PID controller design over conventional controller. [13]

The article [14] presents a Fuzzy Logic Controller, Induction motor, V/F Speed Control, PID Controller and MATLAB Simulink. The system also includes a controller based on fuzzy logic to maintain constant motor speed when the load fluctuates. The response of the controller will be investigated with the

MATLAB/Simulink, the Fuzzy logic, and SimPower Systems toolbox. This paper shows that the result of fuzzy logic controller PID shows better output as compared to conventional PID. Even when the system model is known and linear, modeling, and experimental findings suggest that the fuzzy logic technique is possible and might be an effective alternative to conventional control.

The proportional-integral-derivative (PID) controller is essential in many engineering systems. There is still no theory that explains why the linear PID can efficiently cope with nonlinear uncertain dynamical systems. We will demonstrate how to explicitly create a parameter manifold such that when the PID parameters are chosen from this manifold, the multi-agent systems are globally stable, and the tracking error of each agent approaches zero exponentially rapidly. The major goal of this letter is to extend the 1-D results to higher dimensions nonlinear uncertain systems and to greatly enhance the results using a modified technique. [15]

The authors characterized the nonlinear system in this study as a series of local linear state space models, constructed local PID controllers based on these models, and used the weighted total of the output from these controllers to manage the process. On a CSTR process with dynamic nonlinearity, the efficacy of the suggested control strategies was proven. [16]

A paper [17] proposes Enhanced-PID controller by rating a sector-bounded nonlinear gain in flow with a conventional PID control design. In the features of measurement noise, to pick up a high-quality differential signal, 2 nonlinear track differentiators are utilized. The simulation's outcomes were obtained on a robot manipulator which shows that the designed controller outperforms the standard fixed-gain PID controller. The suggested EN-PID controller's key benefits are its excellent noise resistance and ease of installation.

An optimum PID controller is built to meet certain time-domain performance requirements. The optimum tuning PID control, which uses an approximated process model, while process dynamics are time variable, still it gives us a best possible PID parameters. Given method was implementing in a dSPACE, MATLAB, SIMULINK, and Real-Time working environment. In a control test rig experiment, a PID controller

was examined carefully and evaluated its performance at hydraulic position. [18]

The quality of classical regulatory legislation can be enhanced by applying a non-linear control loop for the difference between the set and real values. The most important criteria of transient phenomena are defined, which includes robustness of the system, overshooting and transient time response. The regulators were configured using a simulation modeling system and numerical minimization of the modified integral criteria. The operation of regulators was investigated for a wide range of items. The primary quality criteria of transient phenomena (time transient, overshoot, robustness) are specified for appropriate modifications of the linear and non-linear regulating laws. [19]

Ziegler-Nichols Develops fixed parameters of PID controller and other researchers result in deprived performance for transient response, as well as low performance index values. However, the modification and designing of the nonlinear controller was extremely sophisticated & challenging. PID controller with dynamics capabilities was suggested in this study to adjust different parameters over time based on the error response. The suggested PID controller's synthesis and analysis are simple, and the system performs well in simulations. [20]

Because of their fixed parameters, traditional PID controllers have poor performance. This work investigates the ideally changed connection among the object's control error & controlled parameters were observed carefully i.e. functions of a non-linear were introduce to produce a NPID controller. For tuning of the parameters Simulink was use. One primary steam temperature system is controlled by a nonlinear PID controller. In terms of performance and resilience, the simulated outcome indicates that the NPID controller outperforms the classic linear PID controller. [21]

2.1 The Hardware Architecture:

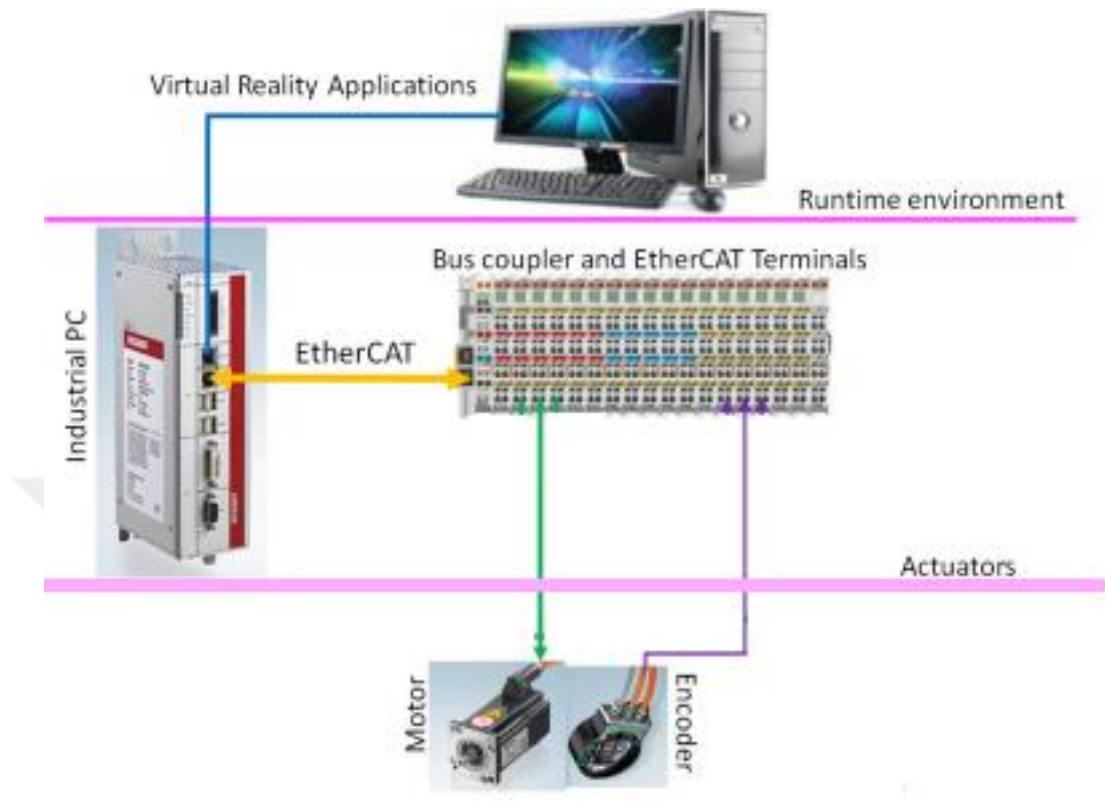


Figure 2.1 Hardware Implementation

CHAPTER 3

METHODOLOGY AND DESIGN DESCRIPTION

Simulink's State-flow is a powerful tool for designing and simulating models based on flow charts and state machines. State-flow has an animation function that shows how the state machine works. It also runs static and run-time tests to ensure that the model can be executed. We modeled and developed our goal system using the State-flow tool in this project. Figure 3.1 depicts the State-flow model.

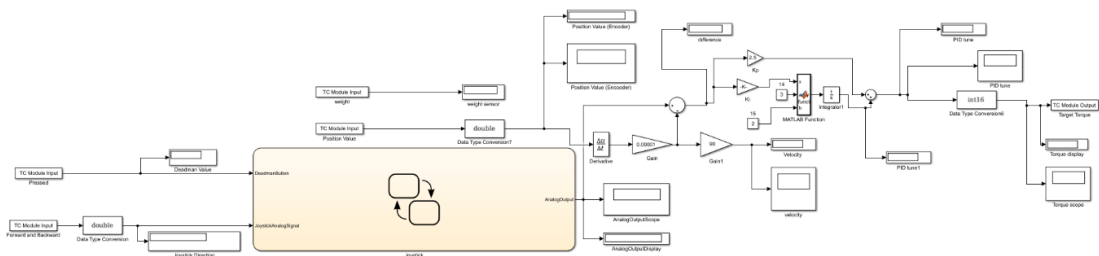


Figure 3.1 State-flow Model

On the joystick, there is a control handle and some dead-man buttons. The control handle regulates the ratio of the direction of lifting or lowering. Switching between these states is made possible by the state-flow environment. The figure 3.2 shows the State-flow chart of Deadman working and as well as it shows error state. Without pressed Deadman button from joystick the motor will not be moving. When Deadman pressed, initially it will check the error state i.e, is there any error occurs or not if there is error then it will go to error state and turn off the program automatically for the safety purpose.

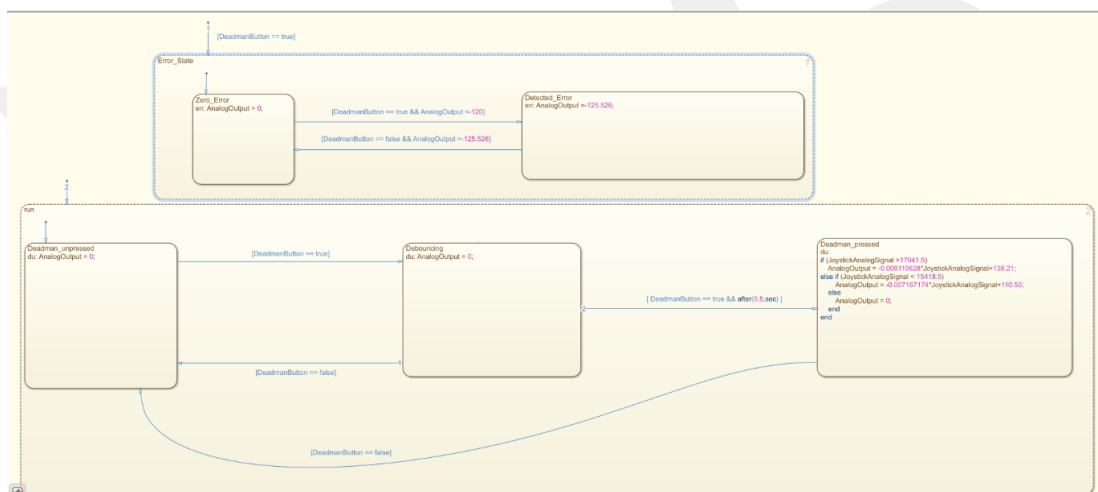


Figure 3.2 Deadman State-Flow Chart

The error state I have set with Analog output and the value of analog output is -125.55. If the analog output is less than -125.55 then it means, there is no error and program logic work accordingly. If the Analog output is greater than -125.55 it means, there is error, and the program will not run.

While leaving the joystick on mean position sometimes it vibrates for a short time, and it may cause damage physical or electrically on machine and as well as on human body. To overcome the debouncing I have added $\pm 5\%$ of delay so joystick active after crossing $\pm 5\%$ on forward and Backward Direction. As in figure 3.3 shows Delaying start of Joystick.

Logical Statements for Delay signal is as written below:

if (JoystickAnalogSignal >17041.5)

 AnalogOutput = -0.008110628*JoystickAnalogSignal+138.21;

else if (JoystickAnalogSignal < 15418.5)

 AnalogOutput = -0.007167174*JoystickAnalogSignal+110.50;

else

 AnalogOutput = 0;

end

end

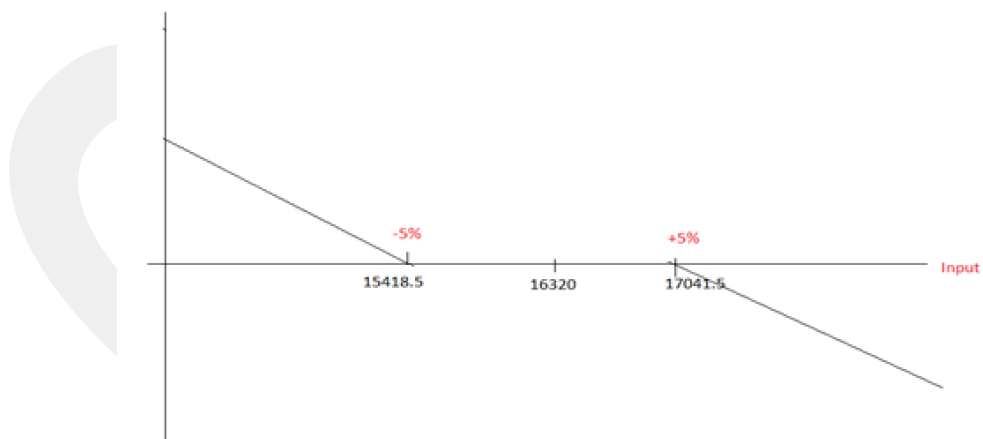


Figure 3.3 Slope of Deadman Delaying Signal

3.1 Simulation for Motor Speed Control:

The simulation on MATLAB shows the State-flow for Motor Speed Control as shown in figure 3.4.

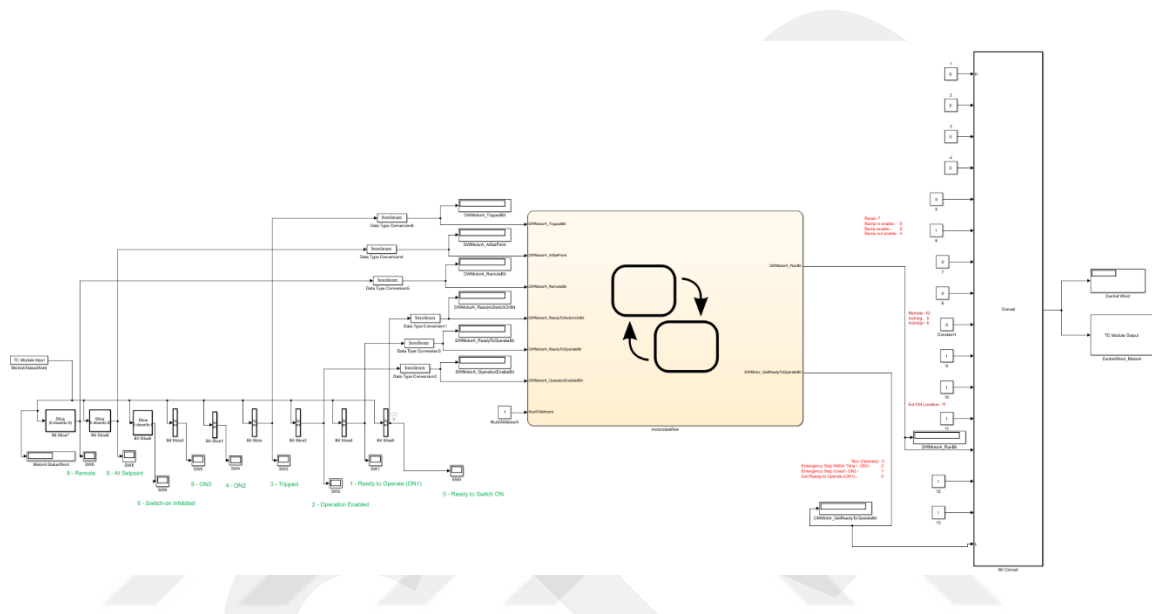


Figure 3.4 State-flow for Motor Speed Control

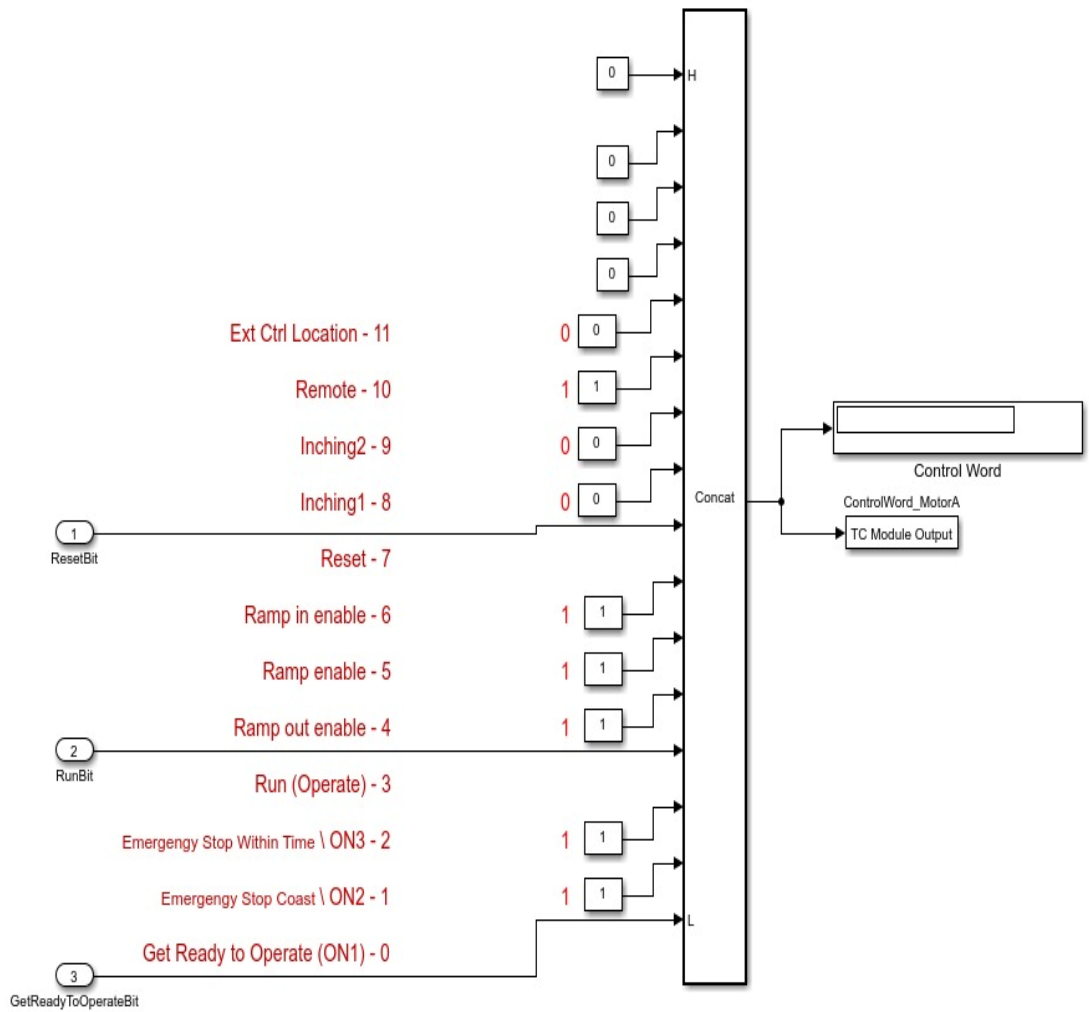


Figure 3.5 Motor Driver Control Register

The State-flow program of starting the motor from Matlab simulation and to control the motor working as well as shown in figure 3.6.

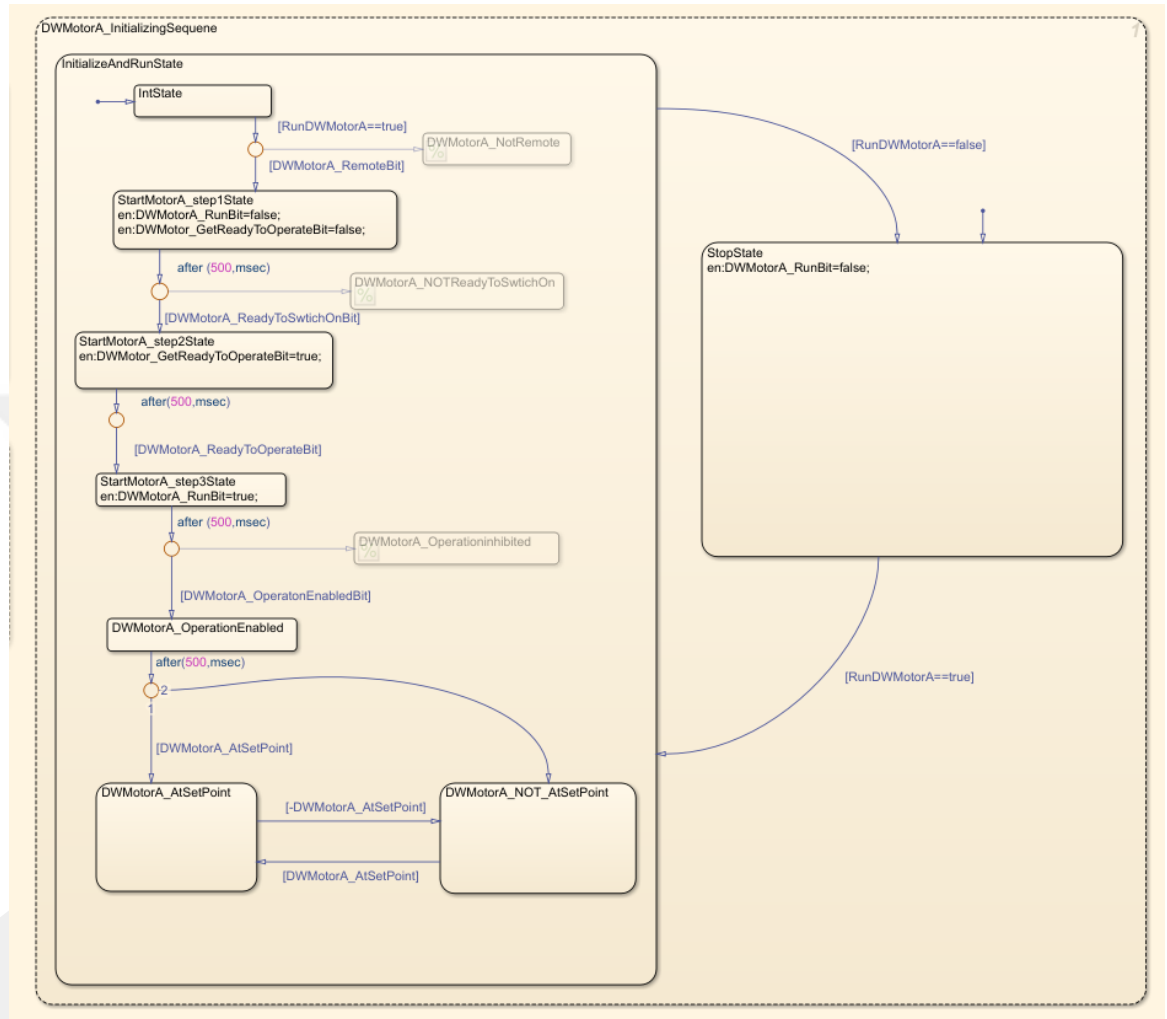


Figure 3.6 State-flow Code for Starting the Motor

Here is the State-flow chart we are tuning for Linear and Nonlinear PID parameters as shown in figure 3.7

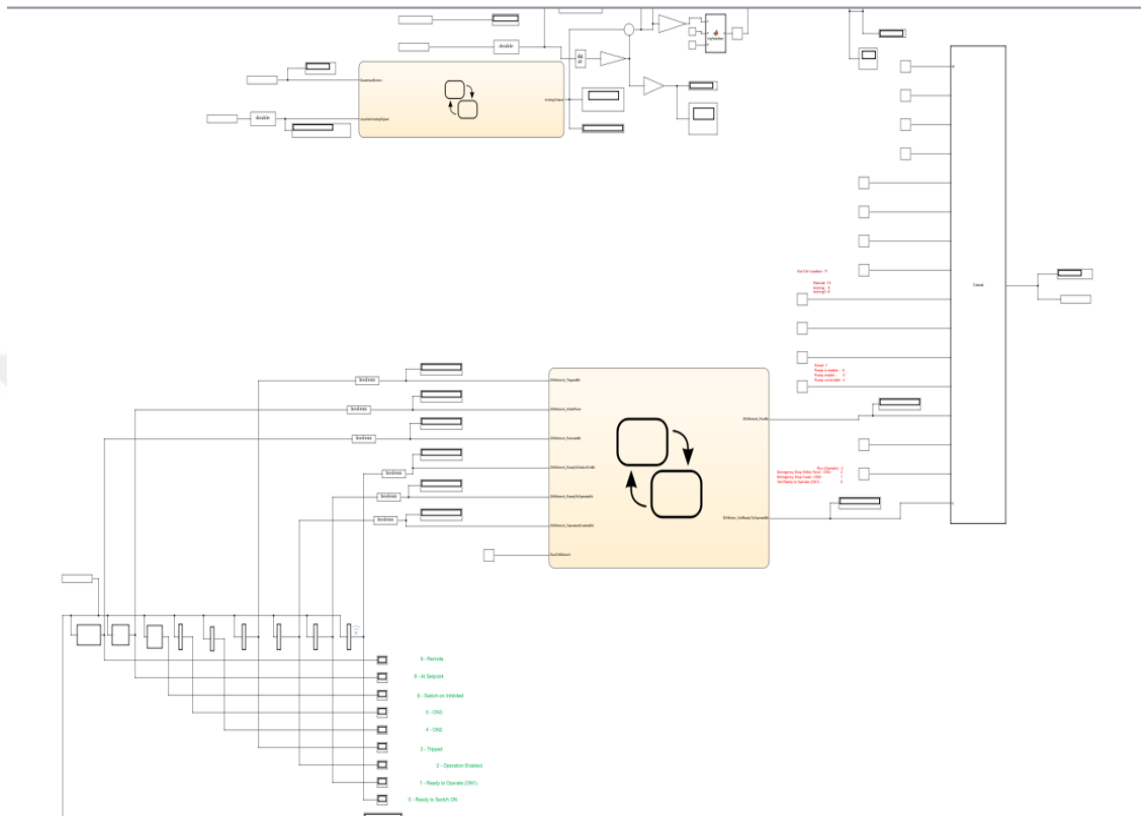


Figure 3.7 Simulation Function for Tuning PID parameters

3.2 Code Generation:

The ability to create code automatically from Simulink models is an important element of the Model-Based Design technique. Hand coding errors and development time can be reduced by adopting automatic code creation. Simulink PLC Coder was used to generate code for this project. Before the code may be created, there are various processes that must be completed. [22]

3.3 TE1400 TwinCAT Target for MATLAB® /Simulink®

A code generator is included in the Simulink Coder for the MATLAB /Simulink environment, which can produce equivalent C/C++ code from the Simulink model. This code generator is used by the TwinCAT Target for MATLAB/Simulink. This has to be set properly with TwinCAT Target, after that model of the object (TcCOM) with I/O behavior of the Simulink model was construct. In the TwinCAT 3 development environment, this module class may be instantiated (TC3 XAE). If necessary, the module instance can be modified in the TC3 XAE. The module is executed in real-time once the TC3 runtime is started and may therefore be incorporated into an actual machine control system. [23]

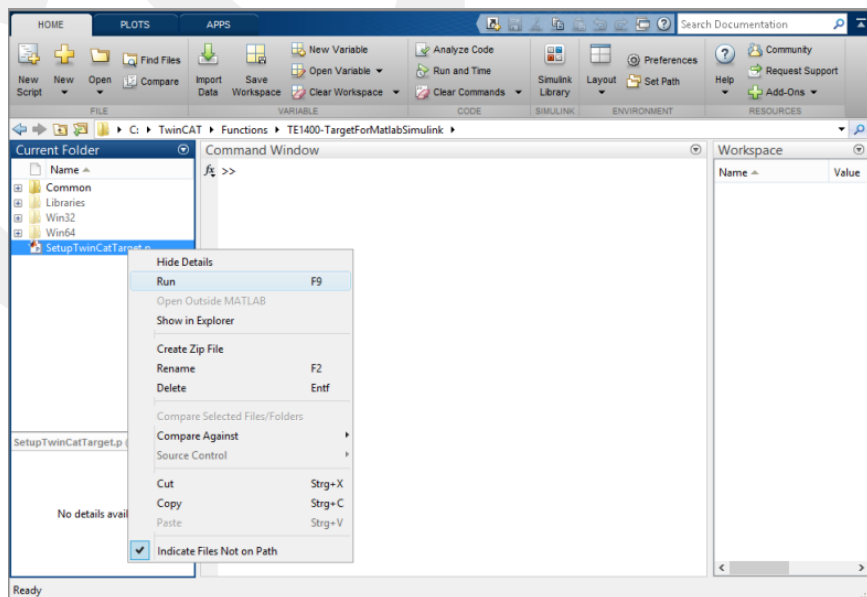


Figure 3.8 Setup TwinCAT Target

The Model Explorer in the Simulink environment's view menu and click on Code Generation Options in the Tools menu, and the Configuration Parameters dialog are all ways to access the coder settings. Select Configuration -> Code Generation from the tree view as shown in figure 3.9

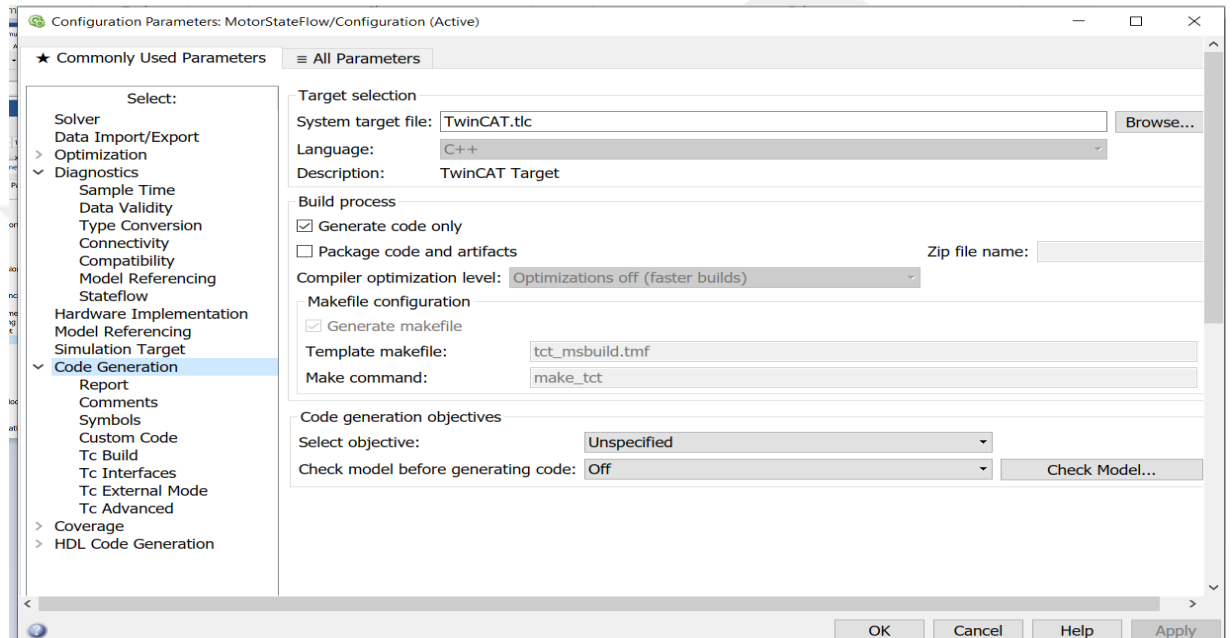


Figure 3.9 Code Generation

Then pick TwinCAT.tlc as the "System target file" in the General tab. Alternatively, open a selection window with the Browse button and pick TwinCAT Target as the target system. A fixed-step solution must also be enabled in the solver parameters to assure the Simulink model's real-time functionality.

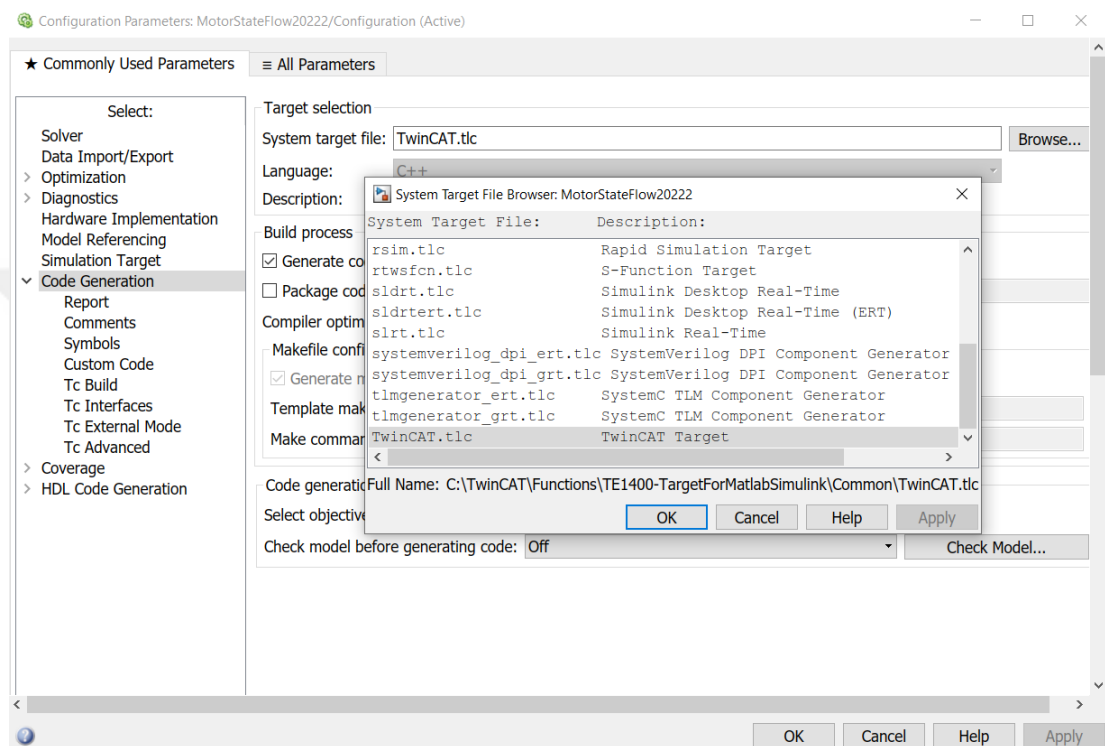


Figure 3.10 TwinCAT.tlc

A broad range of configuration setting choices for the TcCOM module to be created are accessible within MATLAB Simulink. Tc Build, Tc Interfaces, Tc External Mode, and Tc Advanced are added to the tree structure under Code Generation to do this. Many settings may be changed at the module instance level in TwinCAT 3.[23]

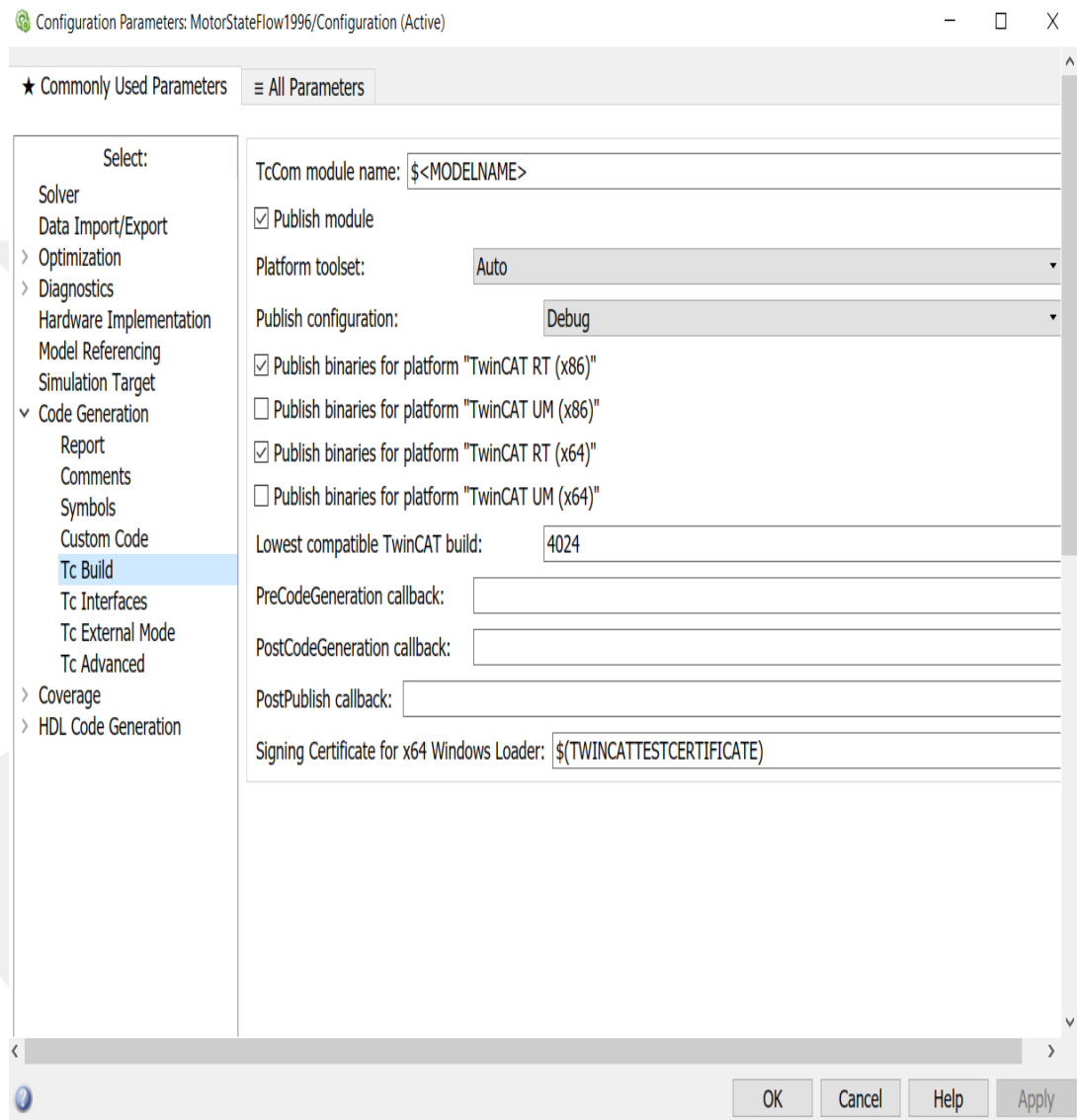


Figure 3.11 TC BUILD

3.4 Data exchange (Tc Interfaces)

In addition to the input and output variables, there are many categories of internal variables depending on the Simulink model. The process image type and ADS access can be modified as needed. These options determine how variables are related to other TwinCAT development environment process pictures and how data is exchanged.[23]

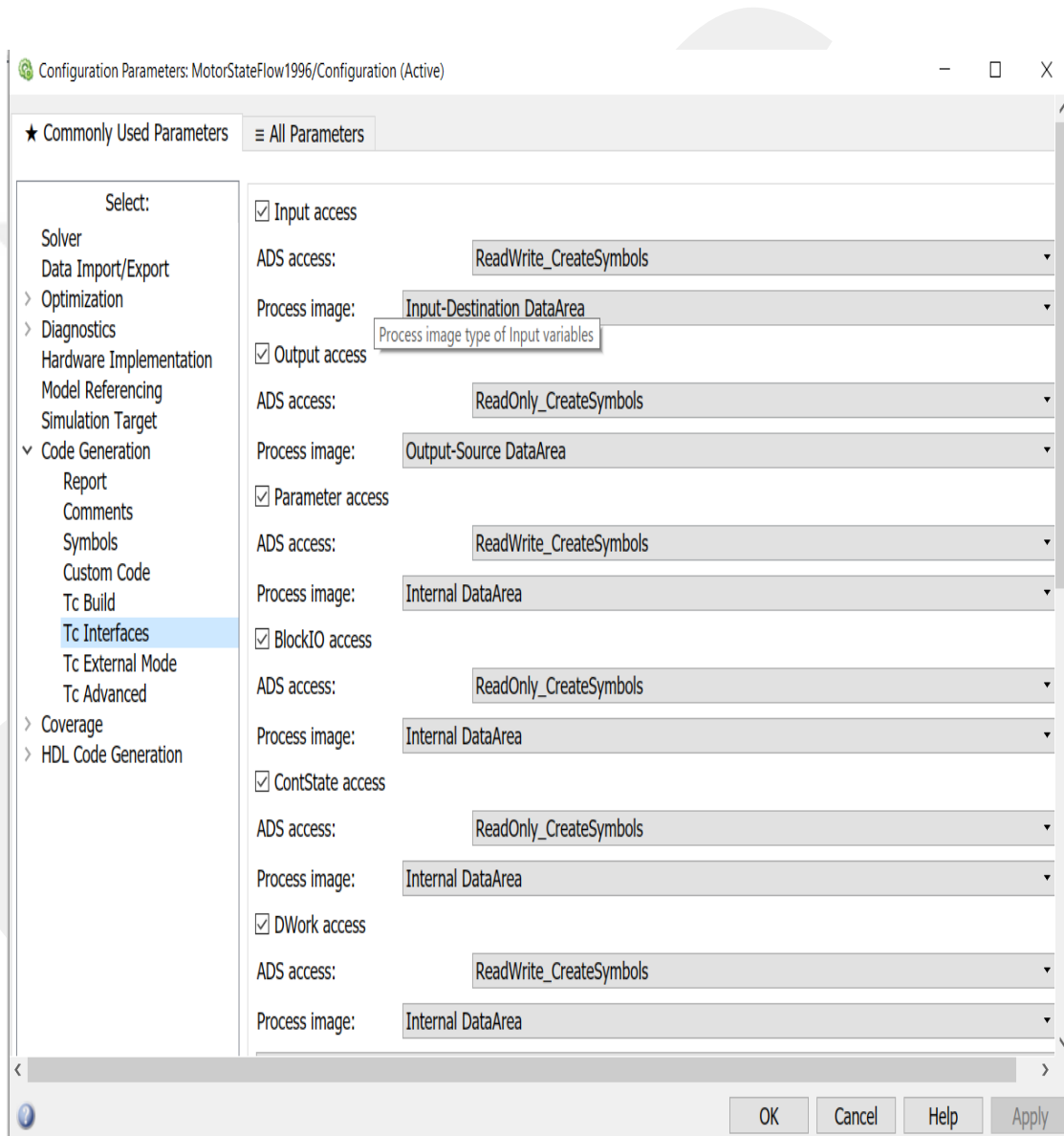


Figure 3.12 TC Interface

3.5 External mode (Tc External Mode)

Simulink has several execution modes. An "External mode" is provided with the addition of "Normal mode," that calculates the Simulink model directly from the environment of Simulink. It just works as a graphical interface in this mode, and no computations are performed in the background. After a model with necessary parameters was transformed into a TcCOM module, it might be linked with instance TcCOM object that was now executing in the real-time environment of TwinCAT. Internal module signals were transmitted to Simulink using ADS, these signals could have been displayed or recorded by using the appropriate Simulink blocks. Simulink parameters can be updated in real time and saved to the TcCOM object.[23]

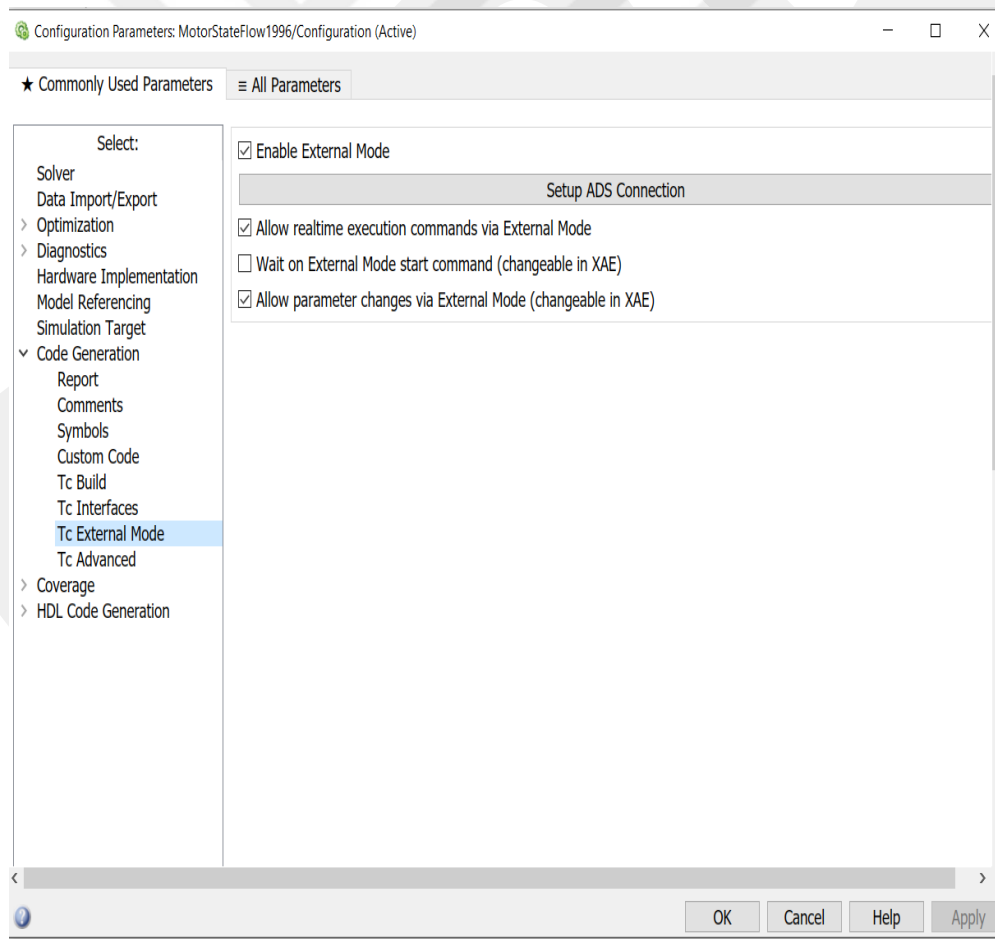


Figure 3.13 Tc External Mode

3.6 Establish connection from Simulink®

When External mode is selected, the Connect to Target icon appears in the Simulink toolbar and it is used to start the "External Mode" connection:

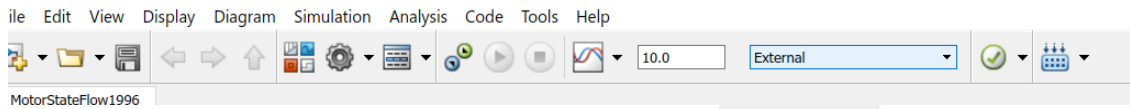


Figure 3.14 External Mode

3.7 Advanced settings (Tc Advanced)

The advanced options allow you to customize the module's execution and call behavior, as well as the appearance and features of the exported block diagram:

The block diagram export, parameter and signal format, and advanced functionality may all be customized using the following coder settings.

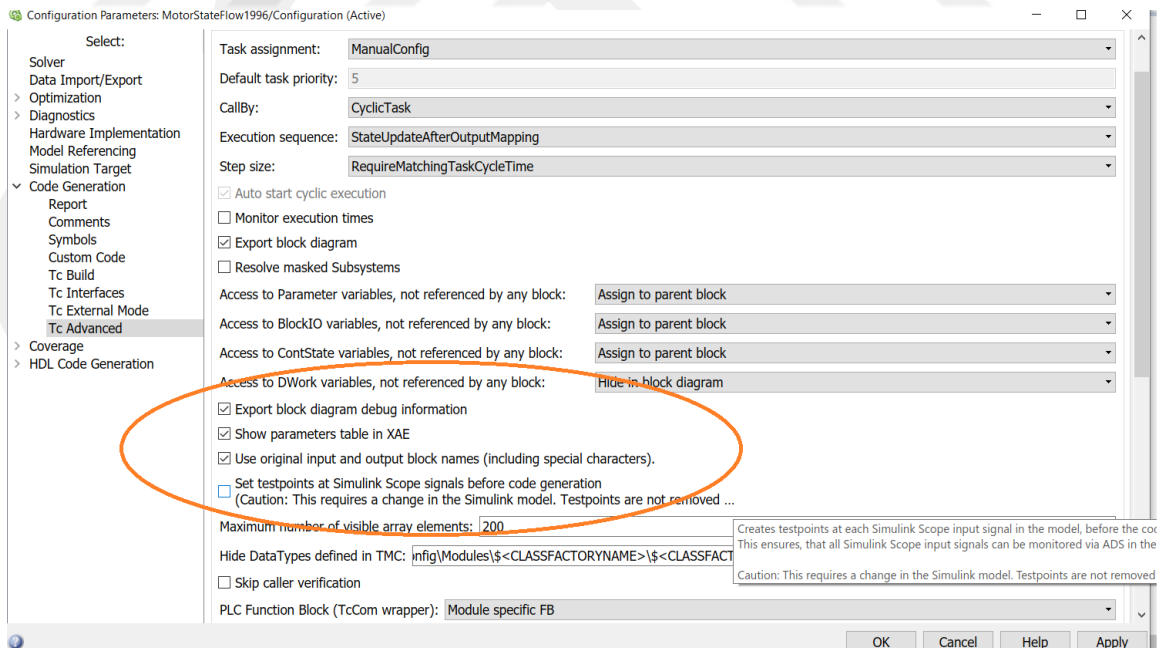


Figure 3.15 Tc Advanced

3.8 TC BUILD

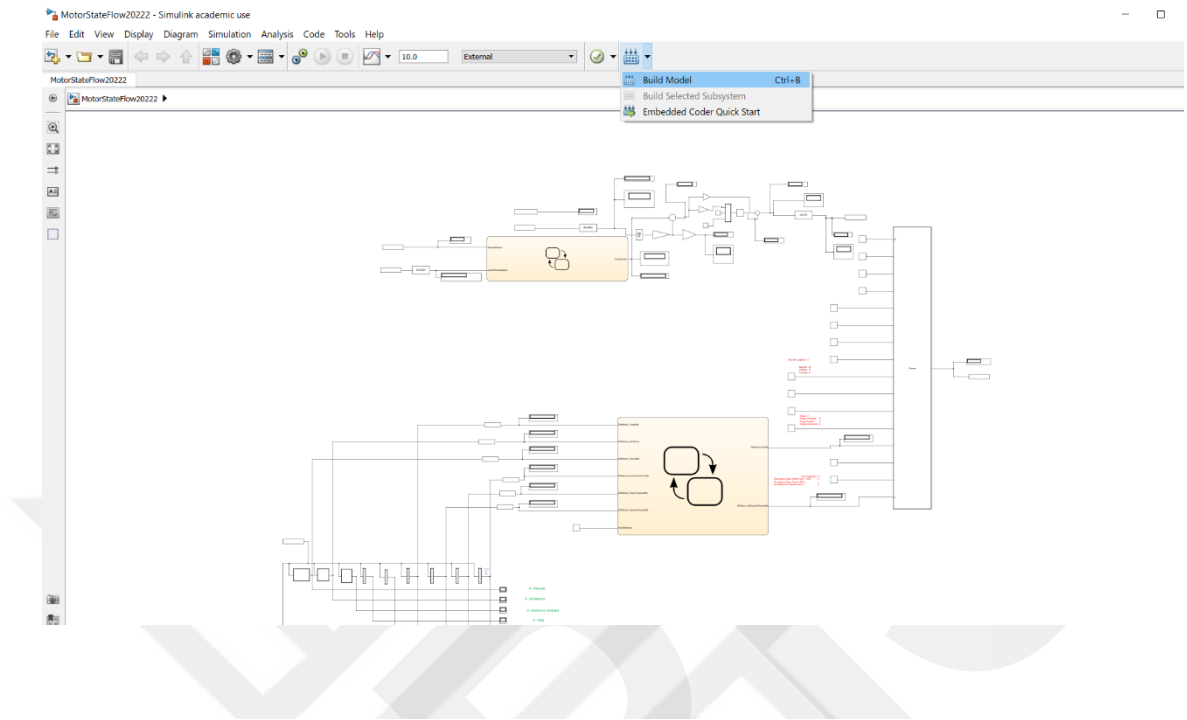


Figure 3.16 TC build

After model building completed, the report of the build process generated and build succeeded. Hence the model has been published on TwinCAT.

3.9 Development of the PLC Programming Algorithm

As shown in figure 3.17, the Twin CAT Project (XML format) must be produced in IDE. All the previous data will be utilized to create a Structured Text algorithm in the TwinCAT 3 IDE, which is integrated into MS Visual Studio.[23]

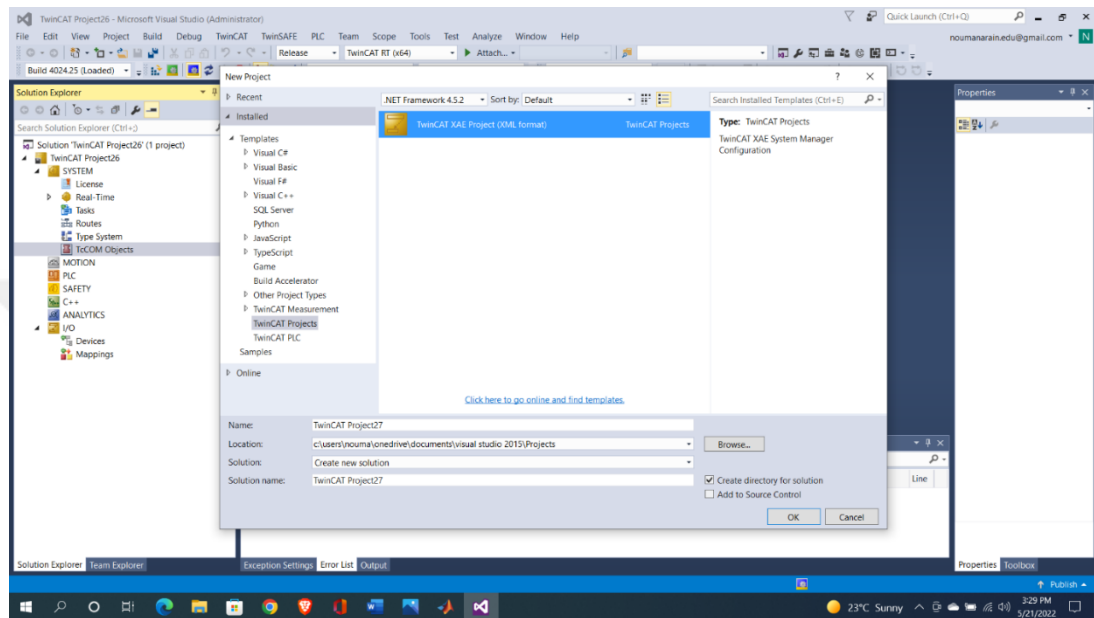


Figure 3.17 TwinCAT Project

Because Twin CAT 3 is utilized in this fashion, the creation of licenses is critical after adding a Twin CAT Project (XML format). These choices must be compatible with the connection functionalities in use.

If you wish to use the full capability of the TE1400 Target for MATLAB / Simulink, then you will need two licenses. The licenses can be activated using the instructions in "Ordering and activating TwinCAT 3 standard licenses." They can only be used for non-commercial purposes. To start a TwinCAT setup with a Simulink module, you will need the TC1320, TC1000 license (or TC1220 plus PLC license). The module and therefore the Twin CAT system, cannot be launched without an activated license. In this situation, you will see error messages about a licensing infringement. You may generate a 7-day trial license that allows you to try the software without purchasing it.[23]

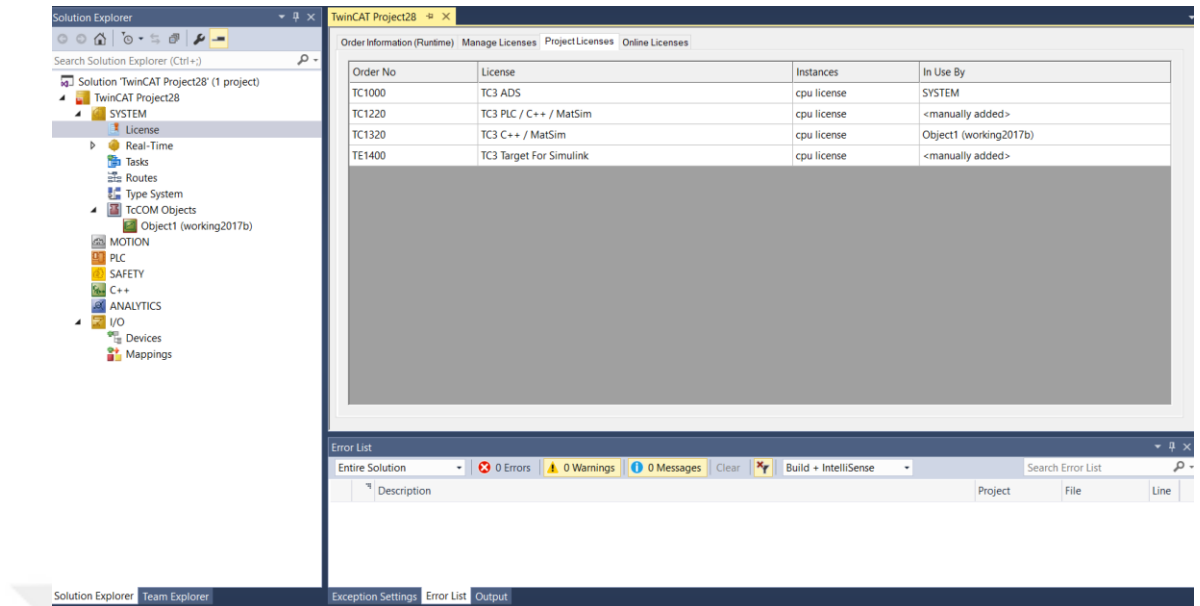


Figure 3.18 License

Create a Task and select the cycle time as the step size of the model. Set it to port 351 and priority 5. As shown in figure 3.19

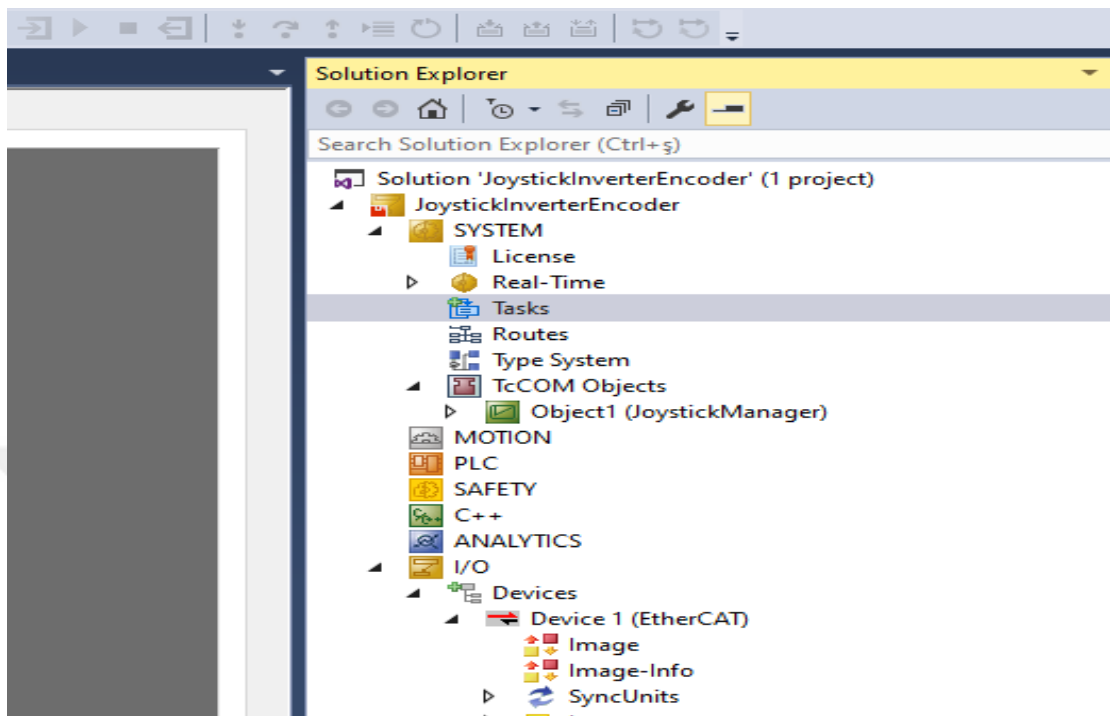


Figure 3.19 Create Task

3.10 Adding TCCOM Object

Any number of instances of the created module can be used in a TwinCAT3 project. The Add New Item context menu is typically used to attach TcCOM items to the node TcCOM Objects. This option will provide you with a list of the modules that are available on the system. Simulink-generated modules may be in the TE1400 Module Vendor > Generated Modules folder.[23]

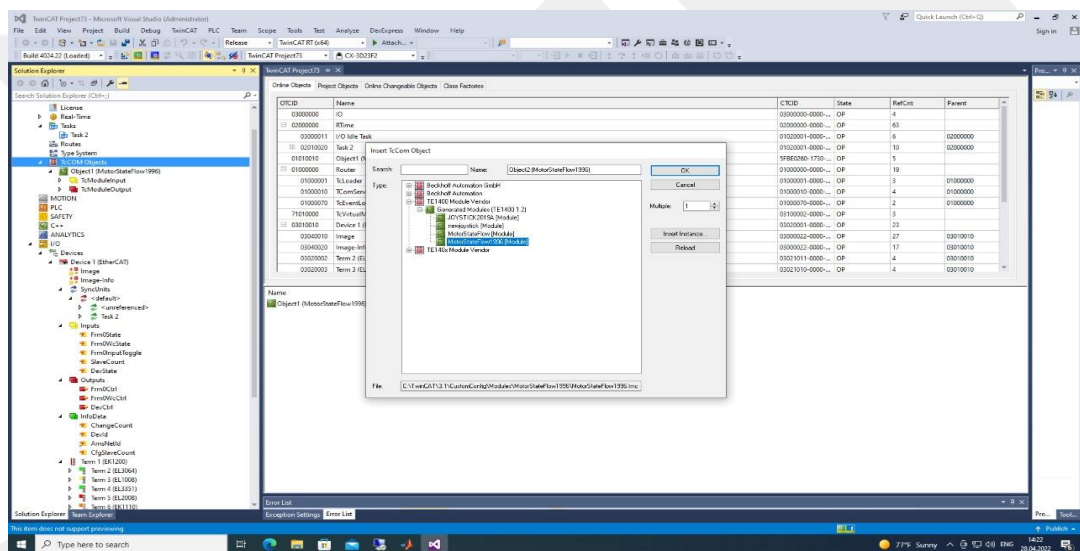


Figure 3.20 Add Generated File in TwinCAT

3.11 Integration of the Module in TwinCAT 3

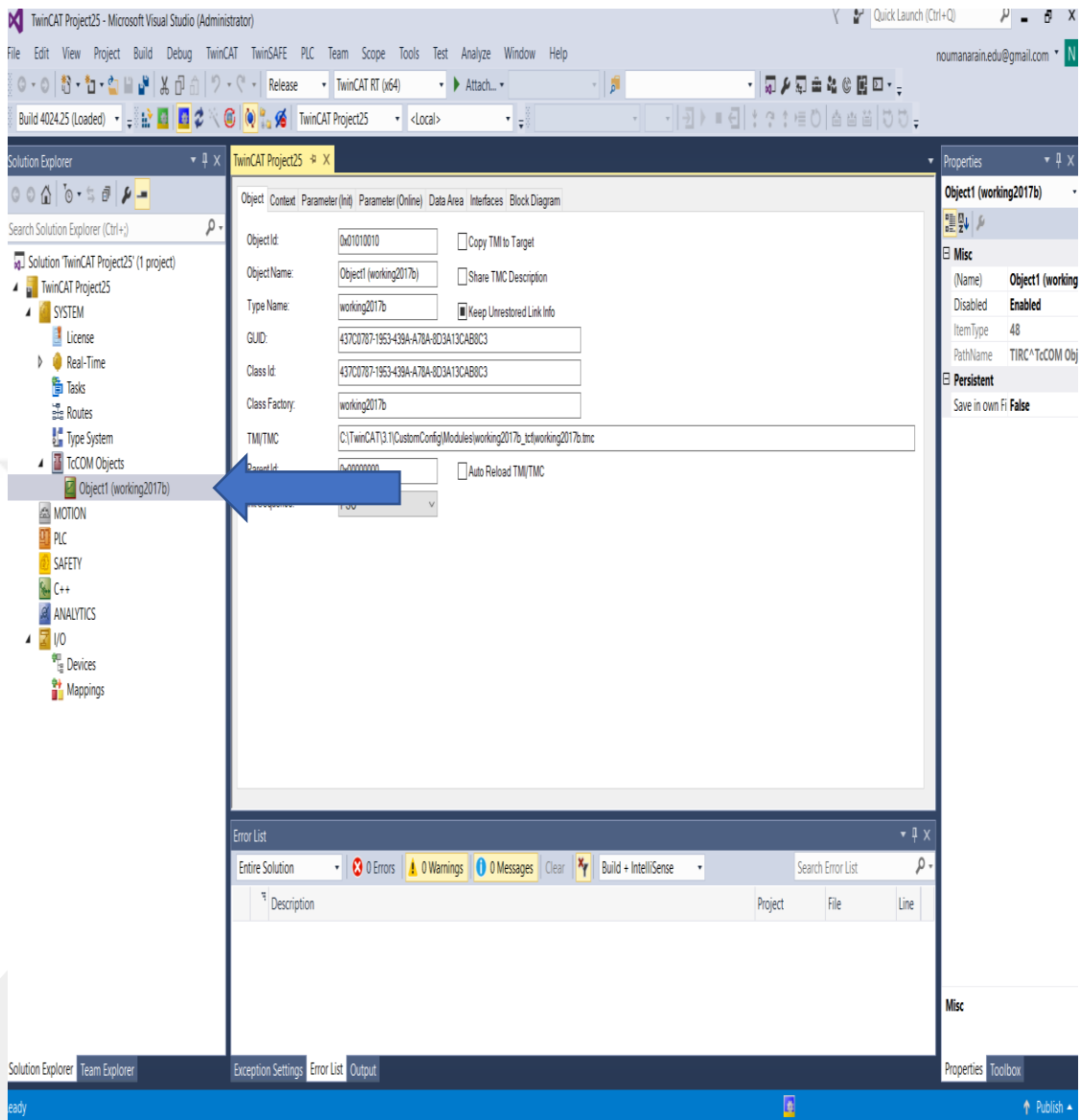


Figure 3.21 Generated file seen on TwinCAT

3.12 Adding Tasks

All the module's contexts that must be given to a real-time job may be found under the Context tab of the module instance the task Properties are automatically allocated to tasks with cycle times and priorities that match the reported values.[23]

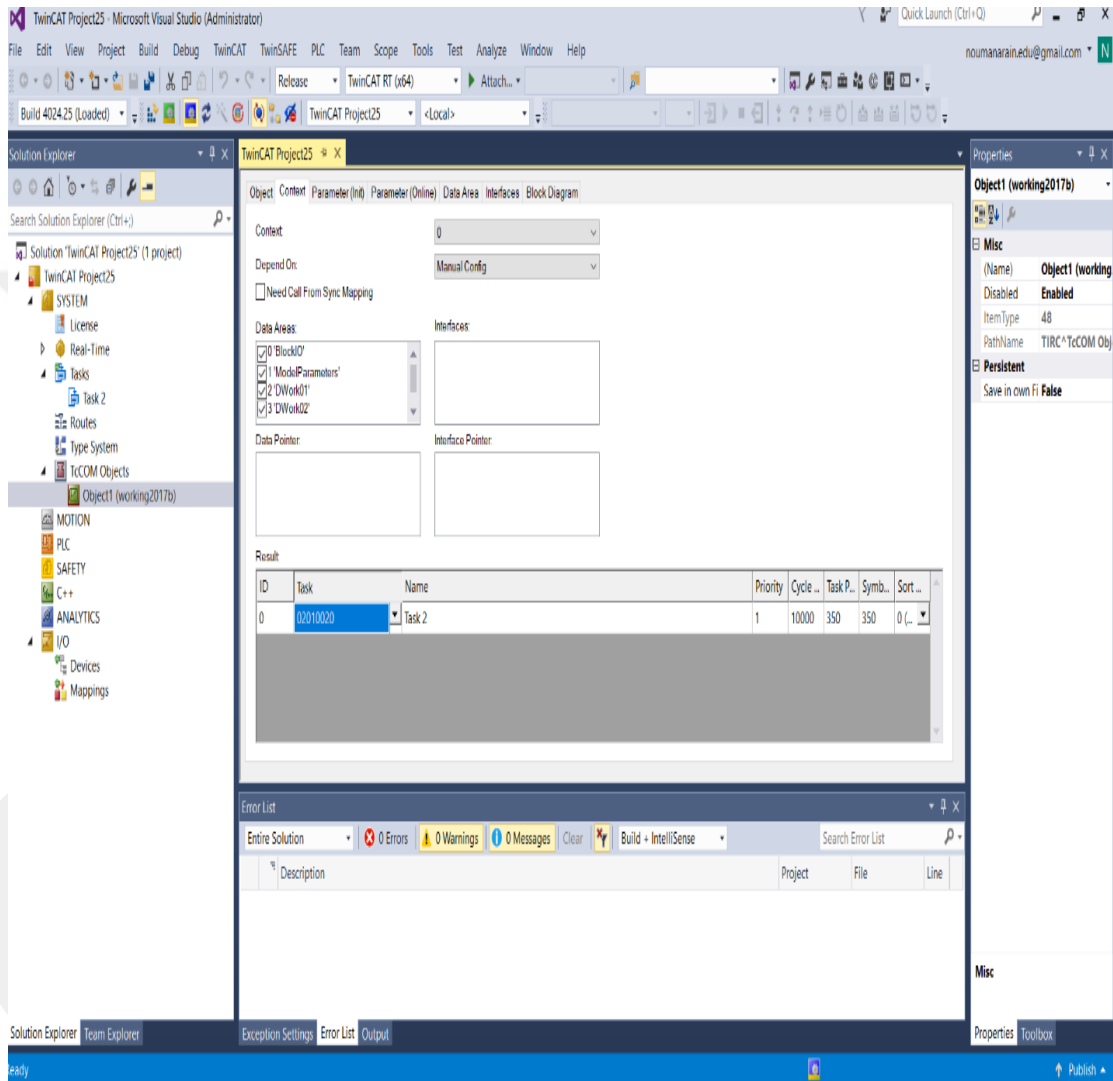


Figure 3.22 Add Task in Generated file

3.13 The block diagram in the Browser Parameters tab

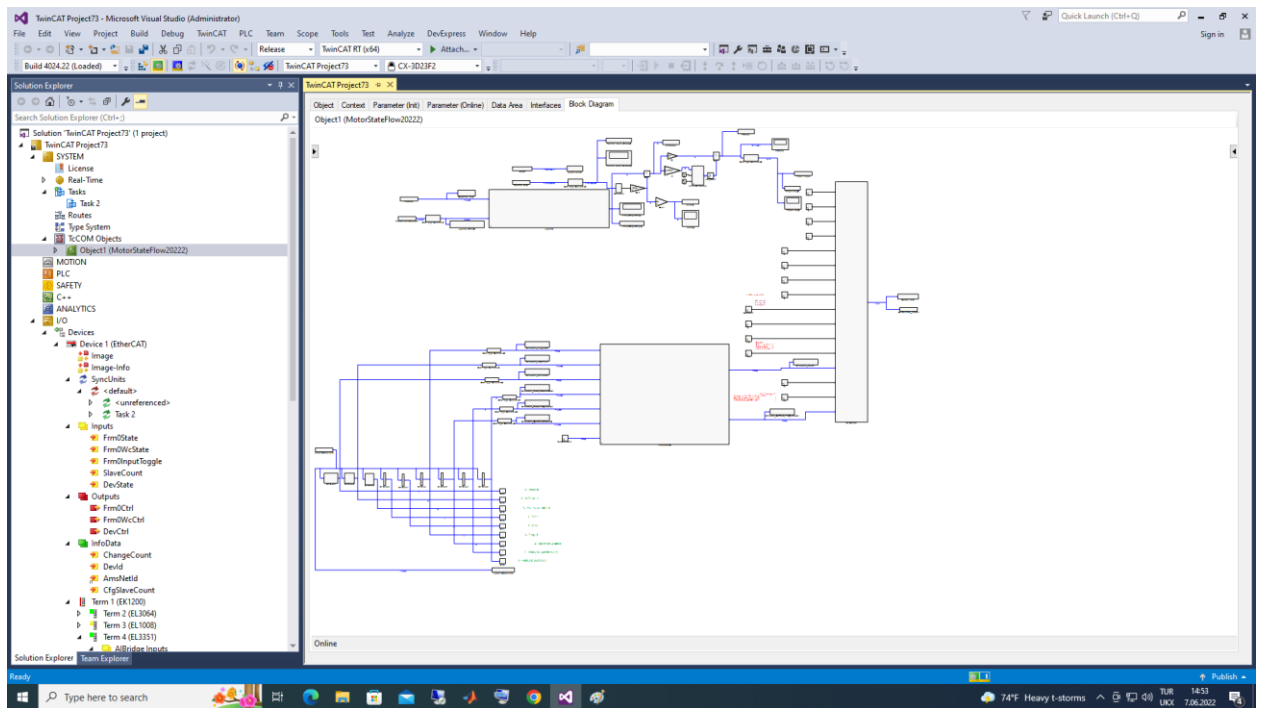


Figure 3.23 Block Diagram Build

3.14 Experimental Laboratory Setup

Experimental setup for the drilling rig was designed and builds to achieve the goals. For performing experimental analysis of the electro-mechanical elements of the setup we establish electronics communications among the elements & computer. After this we evaluate and compare the controller's performance during operations of drilling and improve physically rig efficiency.

To create an experimental drilling rig, the electrical & mechanical components used are given individually, after that the installing & mounting technique was explained.

3.15 Actuator

The actuator is a VOLT Electric type V2E-A-80-M-4-A AC 3-phase asynchronous electric motor with a voltage of 400V, a frequency of 50 Hz, a current of 2 A, and a power of 0.75 KW. The maximum rotational velocity is 1415 revolutions per minute, and the nominal torque is 5.1 N.m. [24]



Figure 3.24 3-Phase AC Motor

3.16 Encoder

A numerical value that is clearly coded for each shaft position was provided by absolute rotary encoder. The encoder we're using is a Pepperl + Fuchs type EN58IL absolute rotary encoder. The following are the important technical specifications: [25]

- Operating voltage: 10. 30 V DC
- Output code: binary code
- Linearity error: $\leq \pm 0.1^\circ$
- Overall resolution: up to 30 B
- Transfer rate: 100 MB/s
- Resolution: Single turn; up to 16 B, Multi-turn; up to 14 B
- Connector; Ethernet: two sockets M12 * 1, 4-pin, D-coded
- Supply: one plug M12 * 1, 4-pin, A-coded



Figure 3.25 Encoder

3.17 Load-Cell

A load cell made by ESIT, model TB1000, is used to calculate the weight of hook, also calculate the tension force by the side of a deadline. It has a capacity of 1000 kg, with a minimum division of 200 g and a maximum overflow capacity of 1500 kg. [26]



Figure 3.26 Load Cell

3.18 Embedded PC

The CX series small DIN rail PCs combine with numerous input-output modules to put up a space-saving industrial controller in the control cabinet. Beckhoff Company manufactured type “CX2020” embedded PC. It has a 1.4 GHz Intel Celeron 827E CPU with two cores, a Microsoft Windows Embedded Compact operating system, control software, and TwinCAT 2 and TwinCAT 3 runtime. It features two RJ45 ports, 4 USB 2.0 ports, 1 optional port and 10/100/1000 Mb/s (Digital Video Interface-Integrated) DVI-I interface, [27]

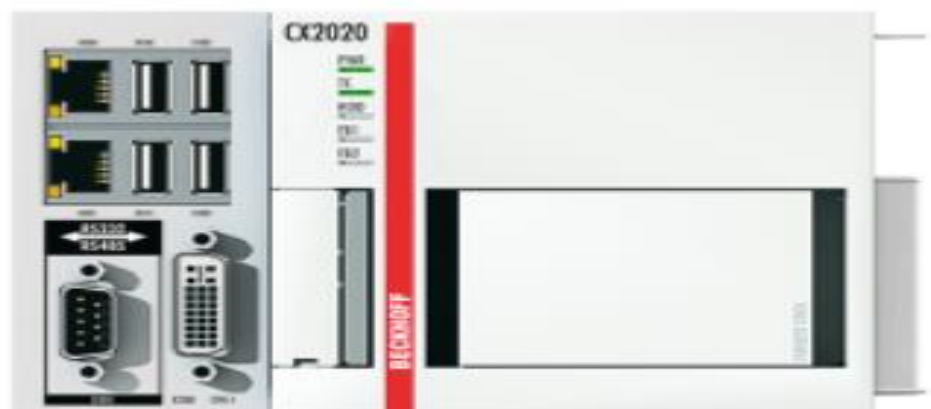


Figure 3.27 Embedded PC

The Input-output terminals listed below are placed on embedded PC based on our application and needs.

1. 4-channel Analog Input Terminal (EL3064: Type).
2. 8-channel Digital Input Terminal (EL1008: Type).
3. 1-channel Resistor Bridge Terminal (EL3351: Type).
4. EtherCAT extension (EK1110: Type).

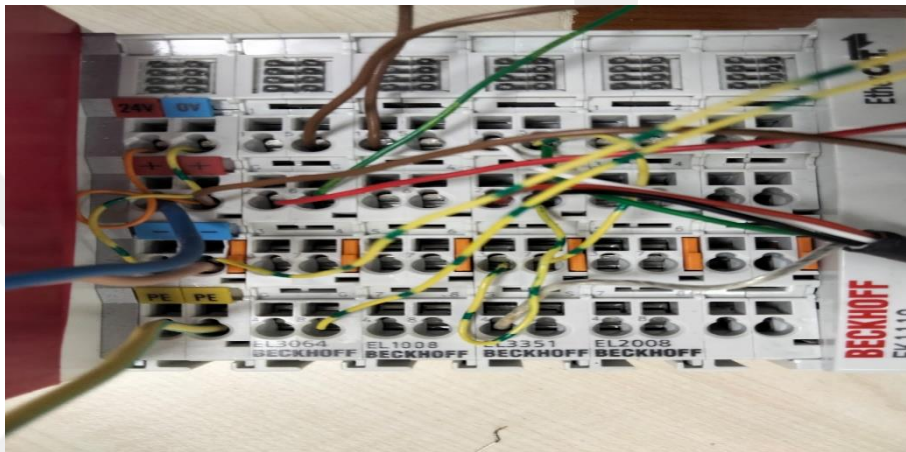


Figure 3.28 EtherCAT Coupler and I/O Terminals

The EtherCAT extension EK1110 is installed at the end of an EtherCAT Terminal segment or an Embedded PC from the CX series and allows the EtherCAT network to be extended in a line topology. [28]

3.18.1 Characteristics

- Technology for making connections: 1 × RJ45 connector
- Extension of an EtherCAT network in line topology with connection lengths up to 100 m.



Figure 3.29 EtherCAT Mounted on the Embedded Computer

3.19 Motor Driver

A self-contained drive which is used to control a lot of applications was manufactured by ABB named as “ACS880” type single drive. The following are the important technical specifications: [29]

- Intuitive control panel and PC tool
- Direct torque control (DTC) for precise open and closed loop control
- Built-in safety features for simplified configuration
- Communication with all major automation networks
- Energy optimizer and energy efficiency information for monitoring and saving energy



Figure 3.30 Motor Driver

3.20 Joystick

Avioni Company manufactured joystick is utilized for our purposes and meets the following requirements.[22]

- Operational Voltage: DC-5, 12V, 24V
- Electric Angle: 36.67°
- Accuracy: $\leq \pm 2\%$
- Protection Class: IP65



Figure 3.31 Joystick i.e. We Used in Our Experimental Setup

3.21 Experimental Setup in Laboratory

All the mechanical, electrical equipment, final construction and physical setup that are being utilized in laboratory is shown in fig 3.32.

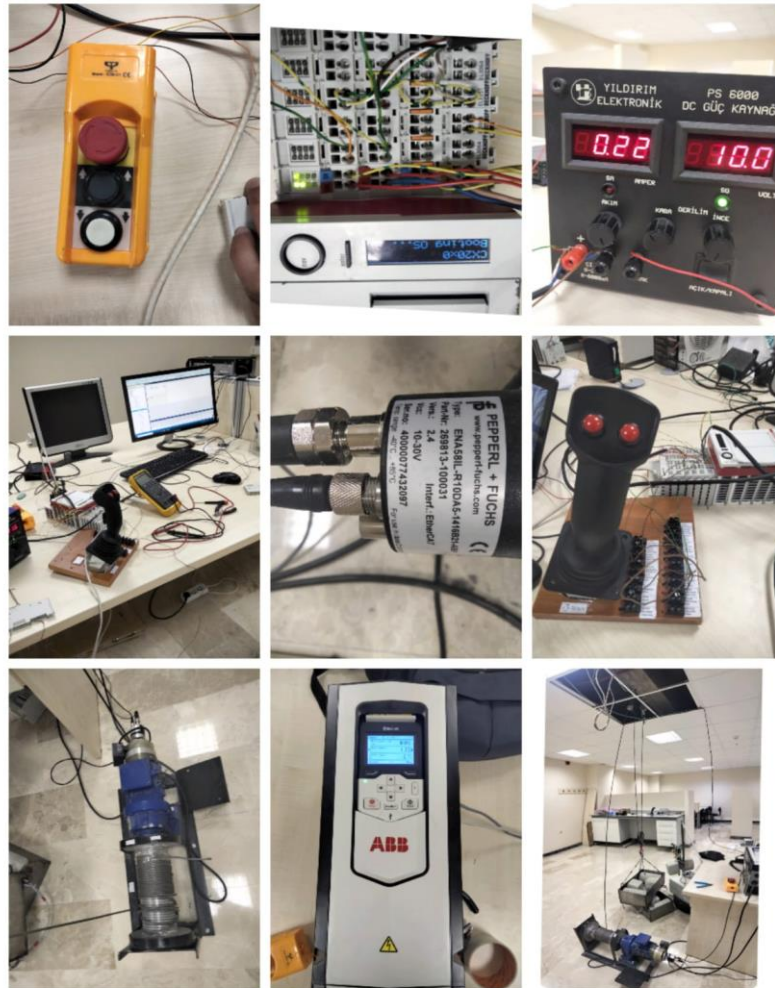


Figure 3.32 Laboratory Setup

3.22 MATLAB FUNCTION BLOCK

In prototype setup the PID controller tuning does not work accurately. The reason behind is that when we are trying to tune parameters the driver becomes unstable. The problem is occurring with Integral values. To overcome such types of instability or vibration and make a suitable controller we are using Nonlinear PID parameters and inserts function in MATLAB Function block which relates to Kp and Ki in the loop.

To deploy code and embedded code in processors, use the MATLAB Function block in Simulink models. You may create legible, efficient, and concise C/C++ code for desktop and embedded applications by using the MATLAB Function block.

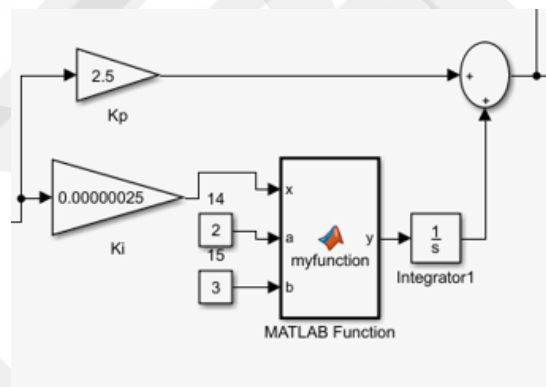


Figure 3.33 MATLAB Function Block

$$y = \begin{cases} |x|^{\alpha} \operatorname{sgn}(x) & : |x| \geq \delta \\ x \delta^{\alpha-1} & : |x| < \delta \end{cases}$$

Figure 3.34 Function of Nonlinear PID Controller

CHAPTER 4

TEST AND RESULTS ANALYSIS

Overshoot, rise time, and fall time values were calculated for optimal characteristics. The robustness of the settings was tested by increasing and decreasing the object's gain. Simulation on the MATLAB system was used to find the best settings for linear and nonlinear PID controllers.

4.1 LINEAR CONTROLLER

Table 4.1 Optimized values for Linear PID Controller in the Simulink

Sr No:	Kp	Ki	Rise Time ms	Overshoot %	Fall Time ms	Overshoot %
1.	2	0.000040	374.875	1.286	712.214	1.873
2.	2	0.000080	389.349	1.315	1146.00	2.381
3.	2	0.000060	373.975	1.453	763.854	1.783
4.	2	0.000090	378.014	1.531	798.617	1.875
5.	2	0.000025	520.271	2.577	604.031	2.019
6.	2	0.000020	393.162	3.873	764.235	1.694
7.	2	0.000030	398.940	4.241	751.985	2.039
8.	10	0.000025	66.074	4.737	74.384	13.617
9.	10	0.0025	73.172	4.737	75.184	12.519
10.	2.5	0.000025	343.126	4.737	350.950	4.064
11.	2	0.000050	348.706	5.851	871.346	1.688
12.	2	0.000070	382.842	5.851	632.647	1.650
13.	4	0.0000005	167.875	9.284	169.870	9.507
14.	7.5	0.0000005	95.823	11.798	83.282	10.490
15.	8	0.00003	77.025	13.068	75.766	13.734
16.	5	0.0000005	123.675	13.068	126.639	11.171
17.	6.5	0.0005	93.146	14.368	100.434	11.214
18.	6	0.000033	100.611	14.368	102.360	11.853

19.	10	0.00013	61.486	24.375	69.069	21.719
20.	8.5	0.00000025	67.965	24.375	80.908	18.084

GCPR

4.2 THE PERFORMANCE OF THE LINEAR PID PARAMETERS

To analyze the stability of the system by tuning manually linear PID parameters we reduce the overshoot from maximum peak to minimum and reduced the rise time as well.

4.3 Controlling Speed by Motor Rotational Speed Control

4.3.1 Scenarios:

In the two cases below, we will observe the system behavior and how controller impacts on the system with respect to velocity is studied in graphs by using linear PID controller and by tuning manually. The first scenario is when weight moves upward direction, and the second scenario is when weight moves in downward direction. The measured vertical and downward velocities are derived. When we start tuning initially from setting the $K_p = 8.5$, $K_i = 0.00000025$ are considered. From fig 4.1 and 4.2, we obtained the Rise time = 67.965, Overshoot = 24.375% and Fall-time = 80.908, Overshoot = 18.084%, respectively. The overshoot increased too much as well as the oscillations, and the results are not satisfied. Then tuning with different values by setting the $K_p = 2.5$, $K_i = 0.00000025$. The fig 4.3 & 4.4 demonstrate the Rise time = 343.126, Overshoot = 4.737% and Fall time = 350.950, Overshoot = 4.064%, respectively. As it seen form figures the overshoot does not decrease as we have required for our system to make it smooth and uniformly moves in upward and downward without any distortion. By decreasing the $K_p = 2$ keeping the $K_i = 0.000040$, during the drilling operation, a smooth vertical and downward movement of the hook is observed. With the Rise time = 374.875, Overshoot = 1.286% and Fall time=712.214, Overshoot = 1.873% as shown in fig: 4.5 & 4.6 respectively, which demonstrates the measured vertical and downward velocity, respectively.

As a result, we obtained the minimum overshoot and rise time but the strategy yields that it is insufficient to smoothly regulate process dynamics. To mitigate the shortcomings of the Linear PID, the nonlinear PID approach was considered. As a result, our approach has less overlapping and a shorter settling time than Linear PID.

Force the joystick in upward direction till it reaches the maximum height accordingly. As it can be seen from the figure 4.1 illustrates the signals vertical velocity. Force the joystick in downward direction till weight on pulley touches the ground. As it can be seen from the figure 4.2 illustrates the velocity graph during downward. During upward/downward operation, the entire controller follows the scenario's method and implements it entirely.

By Setting up the values of $K_p = 8.5$, $K_i = 0.00000025$, we obtained the rise time = 67.965 Overshoot = 24.375% and Fall time = 80.908 Overshoot = 18.084% as shown in fig 4.1 & 4.2 respectively.

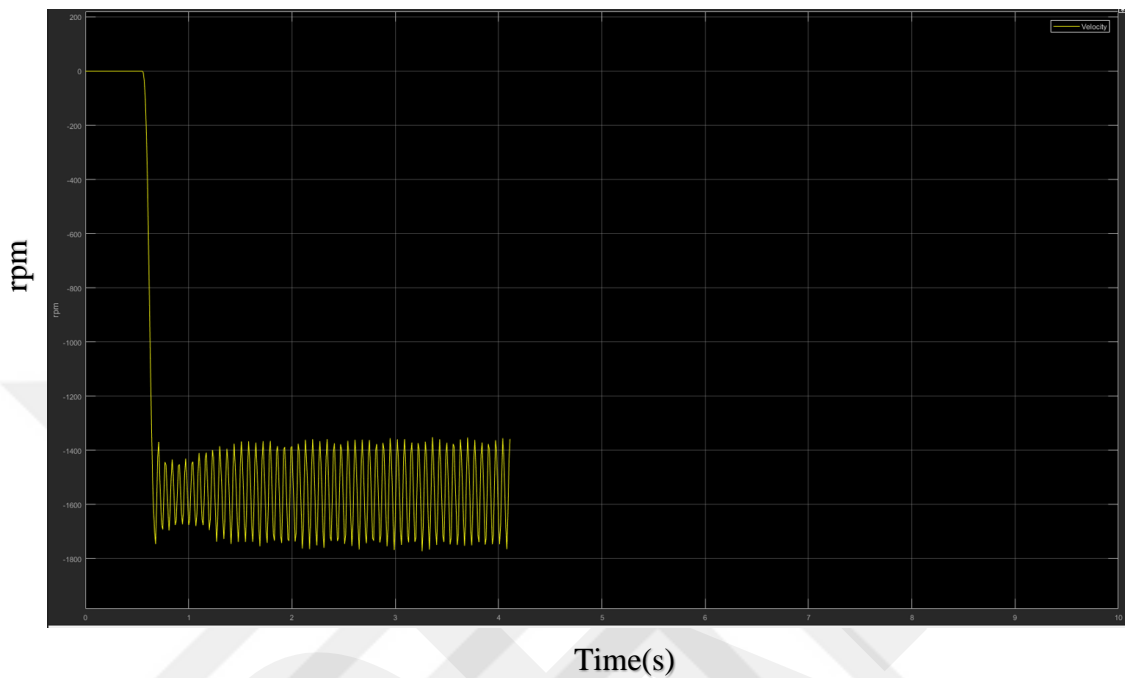


Figure 4.1 Velocity Signal during Upward

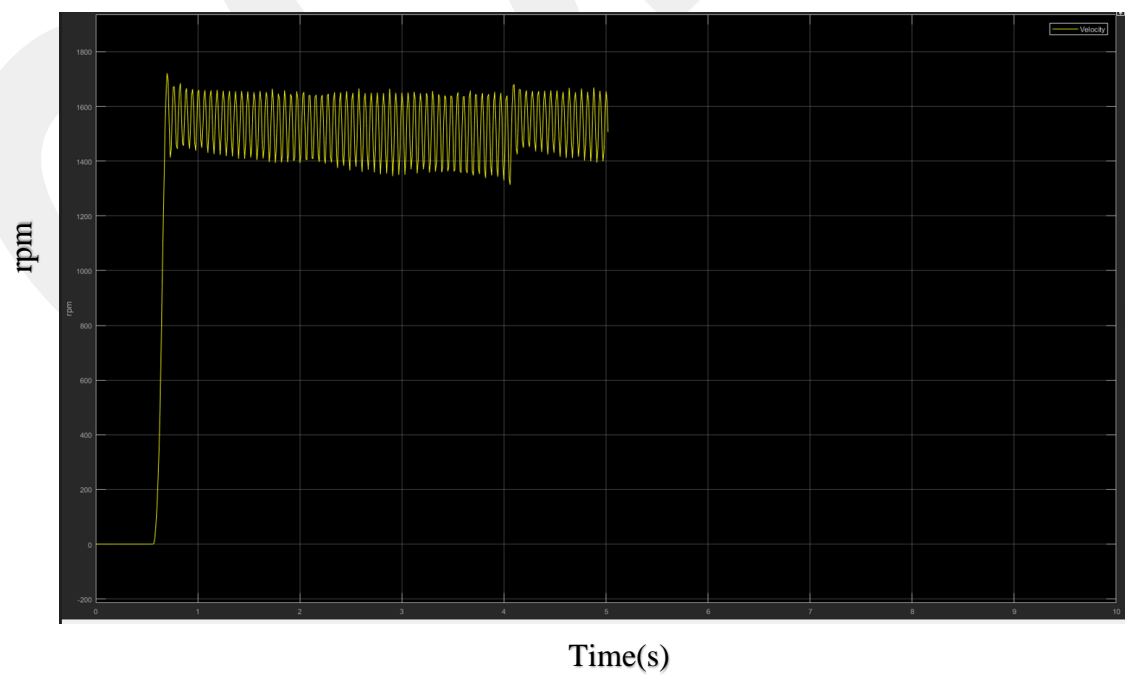


Figure 4.2 Velocity Signal during Downward

By Setting up the values of $K_p = 2.5$ and $K_i = 0.0000025$. We obtained the Rise time = 343.126 Overshoot = 4.737% Fall time = 350.950 Overshoot = 4.064% as shown in fig: 4.3 & 4.4 respectively.

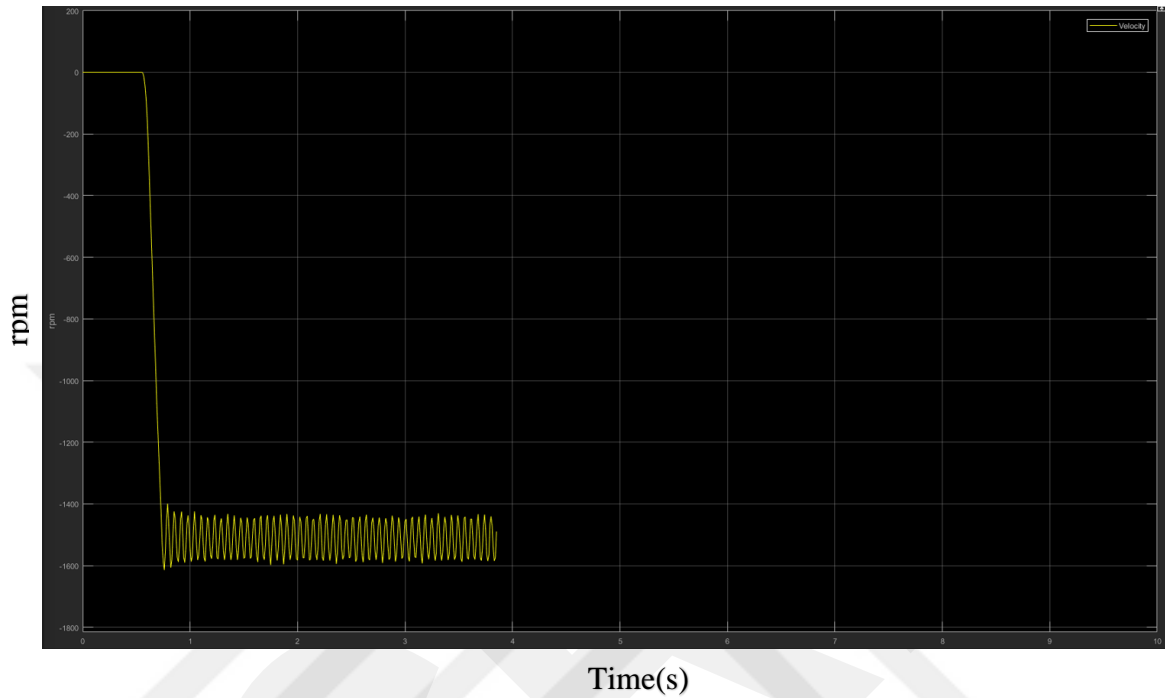


Figure 4.4 Velocity Signal during upward

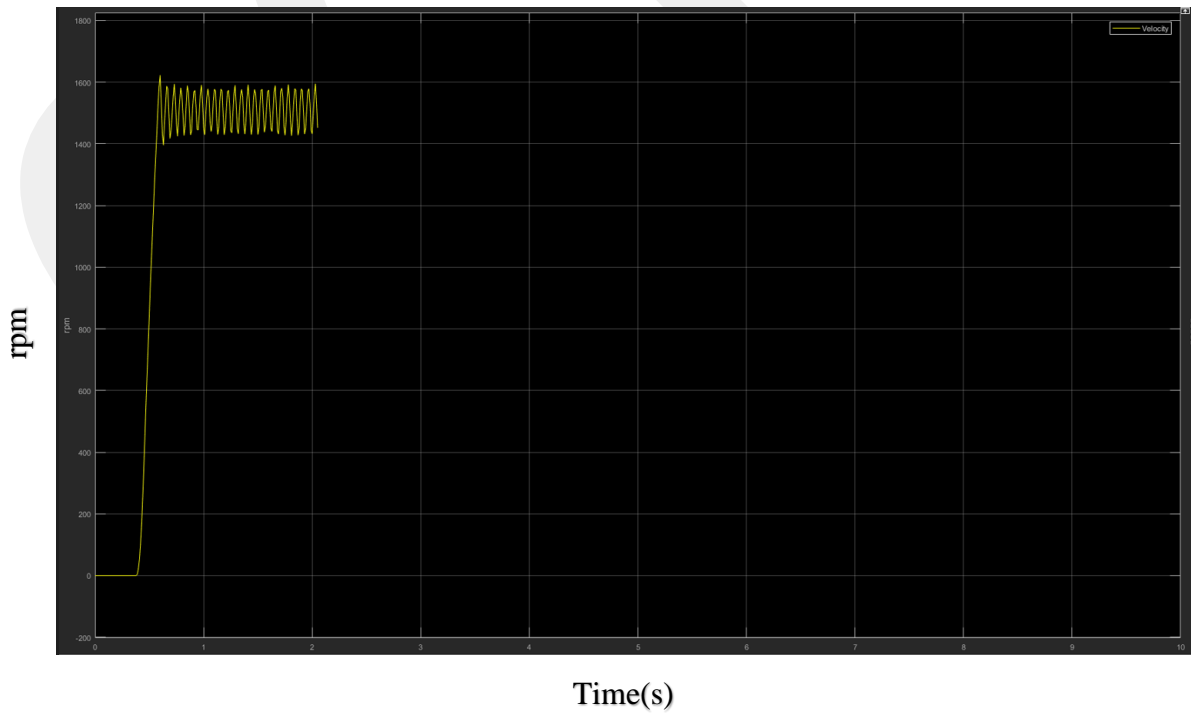


Figure 4.3 Velocity Signal during Downward

By setting up the values of $K_p = 2$ and $K_i = 0.000040$. We obtained the Rise time = 374.875 Overshoot = 1.286% Fall time = 712.214 Overshoot = 1.873% as shown in fig: 4.5 & 4.6 respectively.

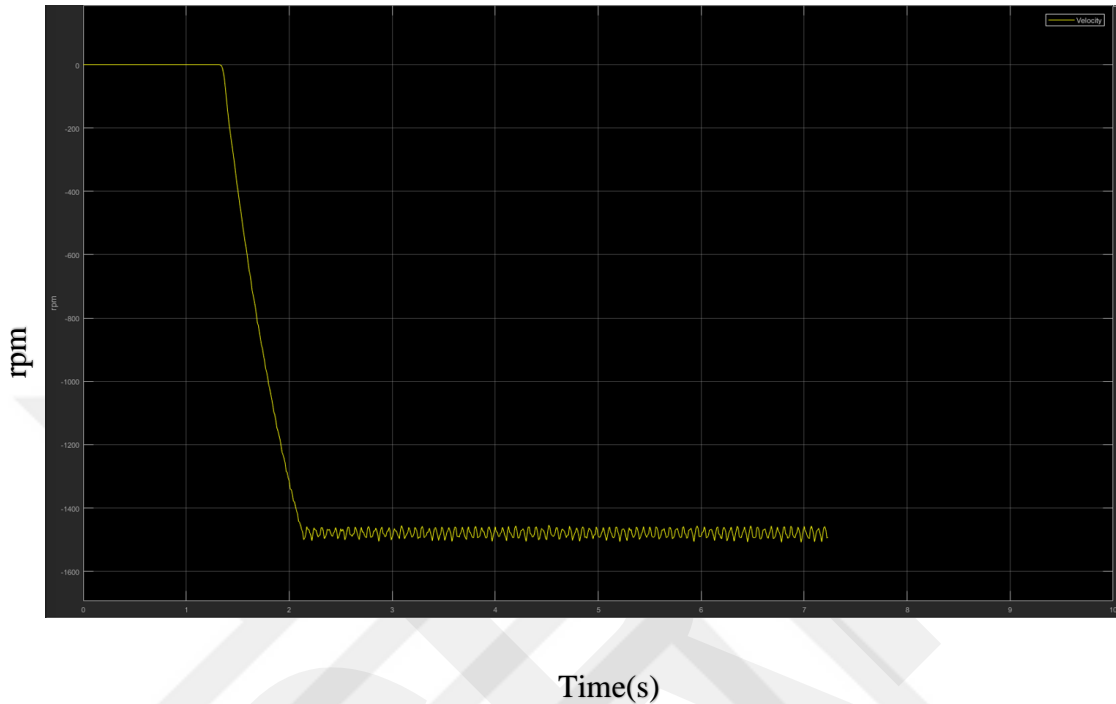
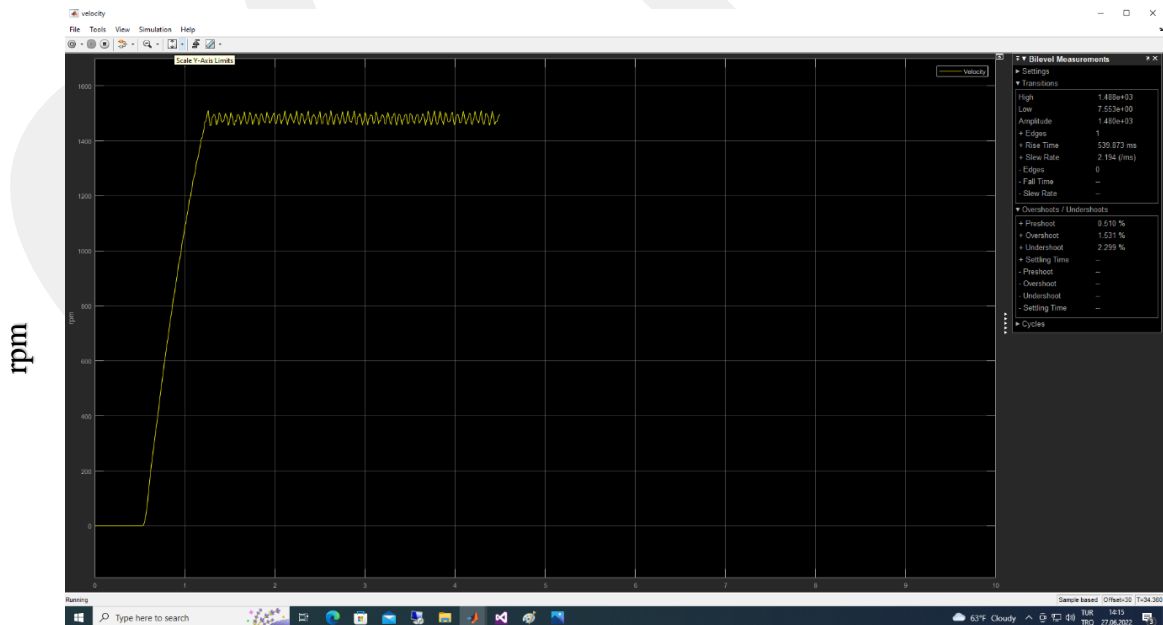


Figure 4.5 Velocity Signal during upward



Time(s)

Figure 4.6 Velocity Signal during Downward

4.4 THE NONLINEAR CONTROL LAW

The basic idea behind the NLPID is to shape the input to each of the controller parts according to a nonlinear law. Our nonlinear law is based on a power function where the exponent is a tunable parameter. To ensure the input-output curve of the shaped law exists in the first and third quadrants.

The exponent alpha is a tunable parameter, which determines the degree of bending present in the input-output curve. With alpha set to 1, the curve is linear with a gain of 1. Setting alpha to a value different from 1 induces bending, as shown in fig: 4.7, the family of input-output curves on the right. Here the x-axis is the input and the y-axis the output.

The gain through the path is the slope of this curve at each point. So, for each value of alpha, the gain changes with the input.

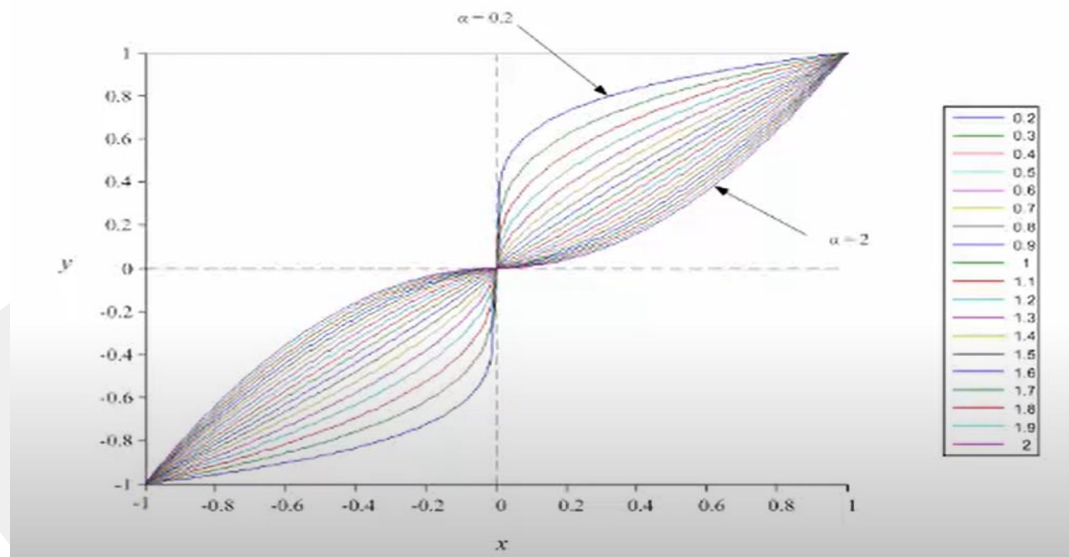
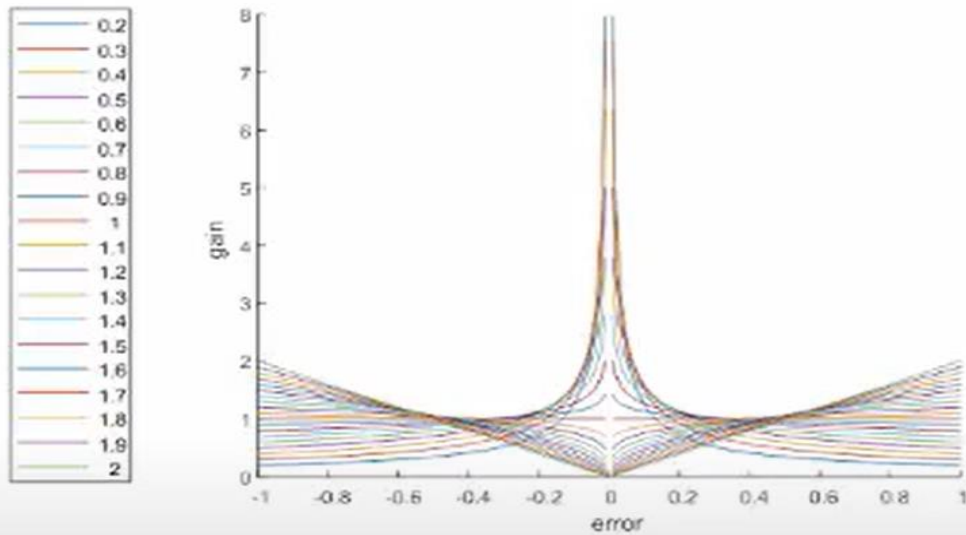


Figure 4.7 Slope of gain Curve by changing Alpha

When alpha is set to a value greater than 1, the gain at the origin becomes 0, implying poor sensitivity there. When alpha is set to a value less than 1, the gain at the origin becomes infinite, which can induce limit cycle oscillation into the loop. Therefore, to ensure superior performance, we need to manage the gain around the origin carefully. The NLPID does this by fixing the region symmetrical with the origin and applying a linear gain there.



The semi-width of the linear region is denoted by δ , and the linear gain in this region is chosen such that the linear and nonlinear curves intersect when the input equals δ . This ensures we have a smooth, glitch-free transition between linear and nonlinear regions. The linear region gain is named γ .

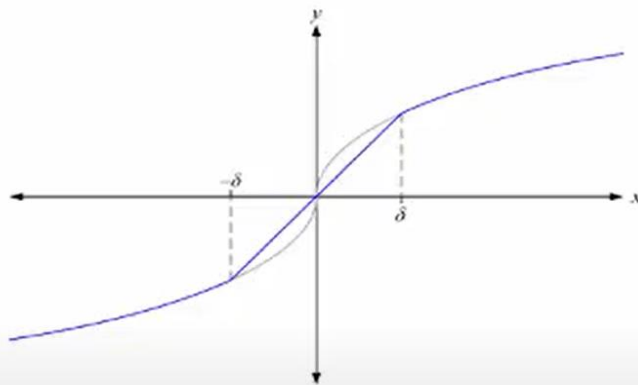


Figure 4.8 Linear and Nonlinear region

4.5 NONLINEAR CONTROLLER

Table 4.2 Optimized values for Nonlinear PID Controller in the Simulink

Sr No:	Kp	Ki	Alpha	Sigma	Rise Time ms	Overshoot %	Fall Time ms	Overshoot %
1.	2.5	0.00000025	0.01	0.5	355.541	0.688	367.397	0.640
2.	2.5	0.00000025	0.1	0.5	286.202	0.857	342.187	4.450
3.	2.5	0.00000025	0.001	0.5	328.688	1.00	671.947	2.697
4.	2.5	0.00000025	0.0000 1	0.005	333.166	1.026	631.423	2.058
5.	5	0.0000005	0.2	0.5	183.990	4.737	115.950	12.062
6.	2.5	0.00000025	0.2	0.5	728.178	1.396	269.049	6.933
7.	6	0.000033	0.2	0.5	143.699	1.531	110.444	12.345
8.	4	0.0000005	0.2	0.5	227.961	1.209	170.590	9.896
9.	2	0.00000025	0.2	0.5	909.930	1.365	791.847	1.650
10.	2.5	0.00000025	0.0001	0.5	634.486	2.071	470.569	1.603
11.	2.5	0.00000025	0.0001	0.005	442.614	3.646	306.106	5.964
12.	2.5	0.00000025	0.3	0.5	287.874	5.832	311.667	5.237
13.	2.5	0.00000025	0.3	0.4	279.975	5.851	405.809	4.419
14.	2.5	0.00000025	0.0001	0.05	300.828	5.924	492.964	3.277
15.	2.5	0.00000025	0.3	0.6	277.975	6.596	288.150	5.481
16.	6.5	0.0005	0.2	0.5	128.768	13.06 8	94.794	11.280
17.	8	0.00003	0.2	0.5	97.728	14.36 8	84.899	16.938
18.	7.5	0.0000005	0.2	0.5	99.128	14.36 8	87.259	18.756
19.	8.5	0.00000025	0.2	0.5	81.273	17.05 9	80.679	18.756
20.	10	0.00013	0.2	0.5	80.282	23.13 4	70.159	25.394

4.6 THE PERFORMANCE OF THE NONLINEAR PID PARAMETERS

To analyze the stability of the system by tuning manually nonlinear PID parameters we reduce the overshoot from maximum peak to minimum and reduced the rise time as well.

4.7 Controlling Speed by Motor Rotational Speed Control

4.7.1 Scenarios

In the two cases below, we will observe the system behavior and how controller impacts on the system with respect to velocity is studied in graphs by using nonlinear PID controller and by tuning manually. The first scenario is when weight moves upward direction, and the second scenario is when weight moves in downward direction. The measured vertical and downward velocities are derived. When initially we starts tuning from setting the $K_p = 10$, $K_i = 0.00013$, $\text{Alpha} = 0.2$ are considered. As it can be seen from fig: 4.9 & 4.10, we obtained the Rise time = 80.282, Overshoot = 23.134% and Fall time = 70.159 Overshoot = 25.394% respectively. The overshoot increased too much, and the results are not satisfied. Then tuning with different values by setting the $K_p = 2.5$, $K_i = 0.00000025$, $\text{Alpha} = 0.3$. The fig: 4.11 & 4.12 demonstrate the Rise time = 287.874, Overshoot = 5.832% and Fall time = 311.667, Overshoot = 5.237% respectively. As it seen form figures the overshoot does not decrease as we have required for our system to make it smooth and uniformly moves in upward and downward without any distortion. Finally, by decreasing the $K_p = 2.5$, $\text{alpha} = 0.01$ and keeping $K_i = 0.00000025$ during the drilling operation, the hook moves smoothly vertically and downwardly. With the Rise time = 355.541, Overshoot = 0.688%, and Fall time = 367.397, Overshoot = 0.640% we achieved the required results as shown in fig: 4.13 & 4.14 respectively, which demonstrates the measured vertical and downward velocity respectively.

4.7.2 When weight moving Upward and Downward direction

Force the joystick in upward direction till it reaches the maximum height accordingly. As it can be seen from the figure 4.9 illustrates the signals vertical velocity. Force the joystick in downward direction till weight on pulley touches the ground. As it can be seen from the figure 4.10 illustrates the velocity graph in downward. During upward/downward operation, the entire controller follows the scenario's method and implements it entirely.

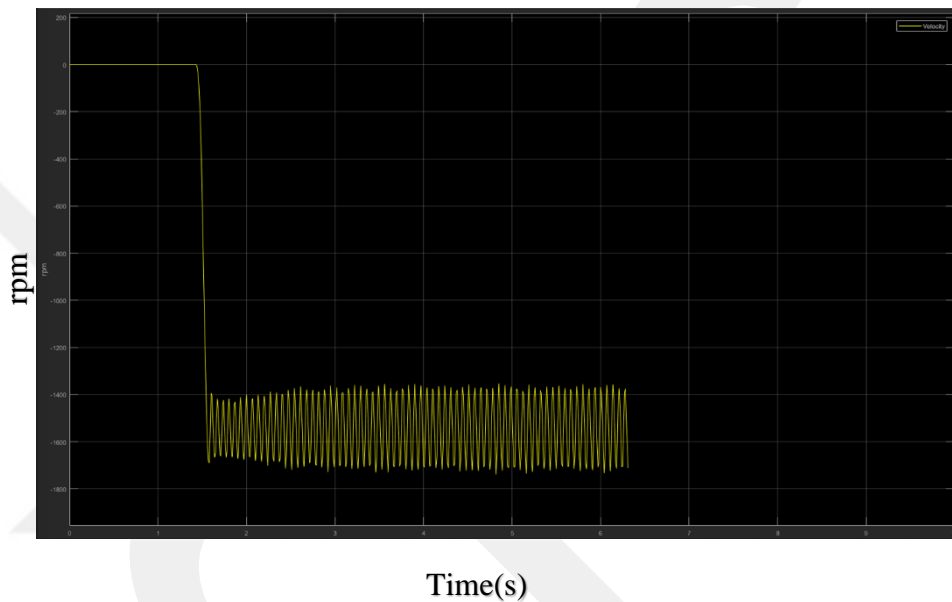


Figure 4.9 Velocity Signal during upward

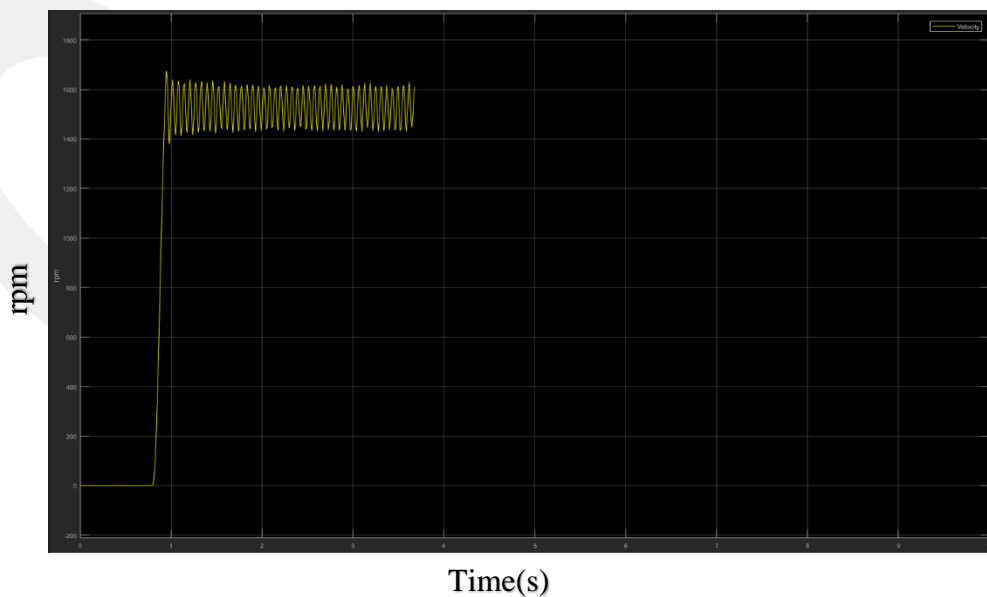


Figure 4.10 Velocity Signal during Downward

By setting up the values of $K_p = 2.5$, $K_i = 0.00000025$, $\text{Alpha} = 0.3$. We obtained the Rise time = 287.874, Overshoot = 5.832%, Fall time = 311.667, Overshoot = 5.237% as shown in fig: 4.11 & 4.12 respectively.

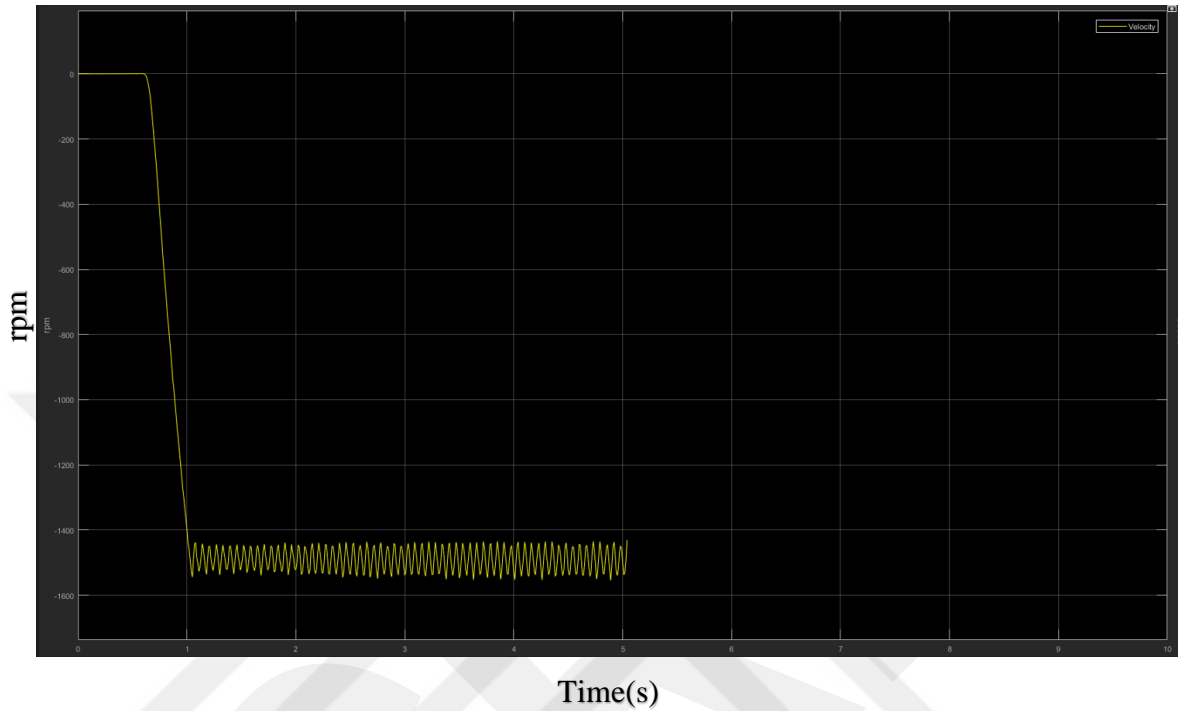


Figure 4.11 Velocity Signal during upward

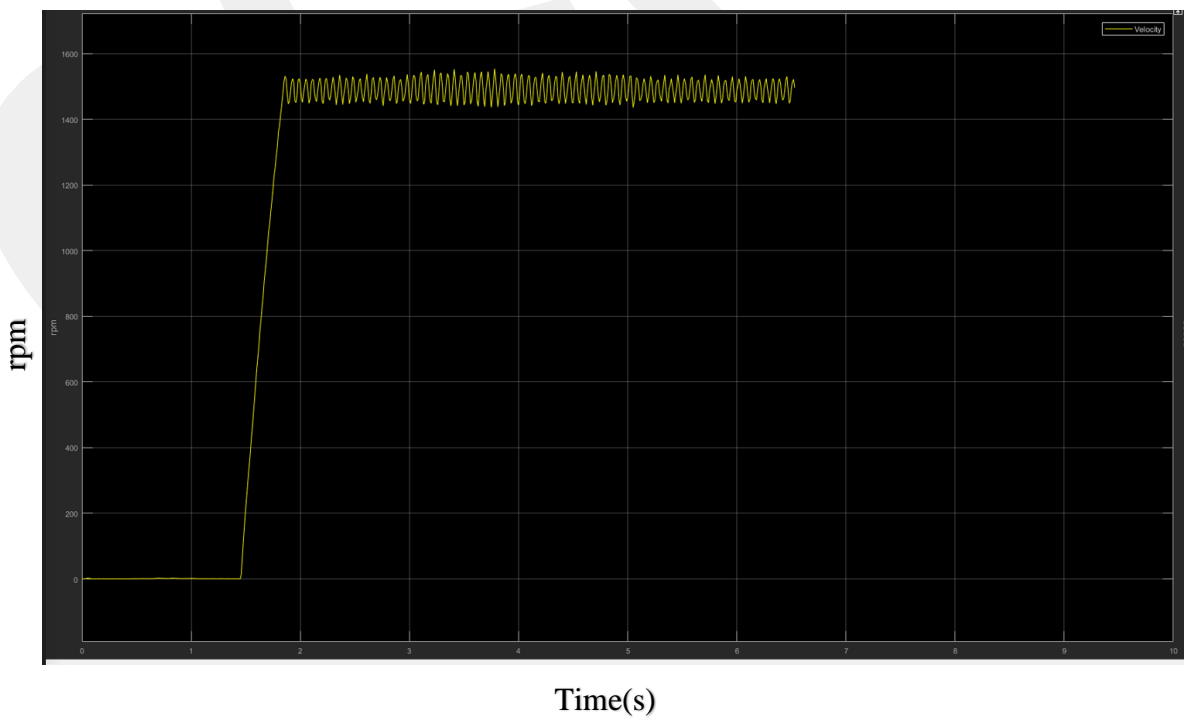


Figure 4.12 Velocity Signal during Downward

By setting up the values of $K_p = 2.5$, $K_i = 0.00000025$, $\text{Alpha} = 0.01$. We obtained the Rise time = 355.541, Overshoot = 0.688%, Fall time=367.397, Overshoot = 0.640% as shown in fig: 4.13 & 4.14 respectively.

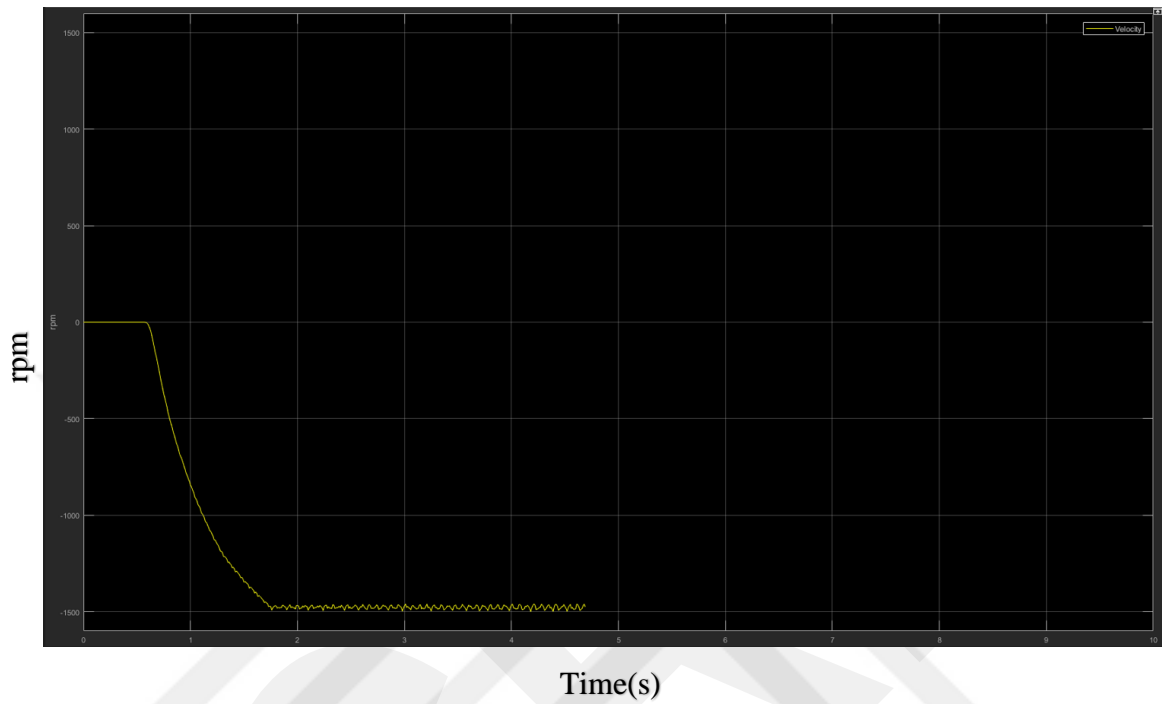


Figure 4.13 Velocity Signal during upward

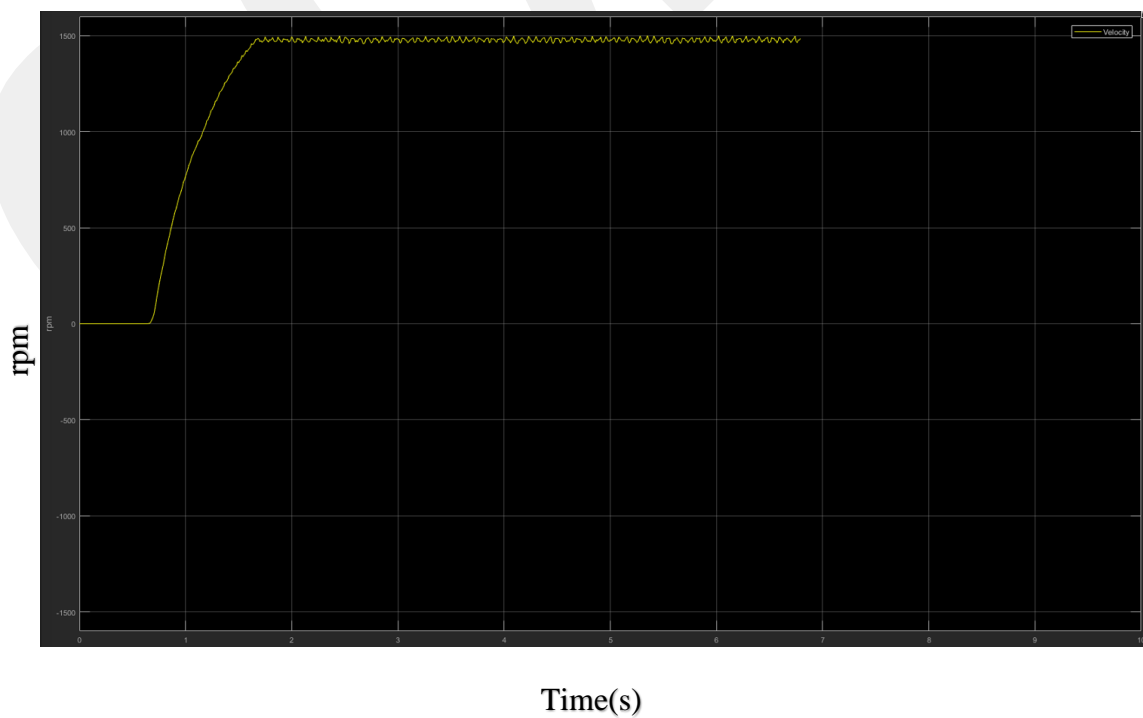


Figure 4.14 Velocity Signal during Downward

In both the controllers we observe that increasing the Proportional gain the rise time decreased and overshoot increased directly. By considering both controllers we will take that controller whose overshoot and rise time is minimum. If we consider both linear and nonlinear PID controller we can say that Nonlinear PID controller is best to reduce the rise time and overshoot as well. Also, the nonlinear PID controller has more smoothness than linear PID controller. In nonlinear PID parameters we have alpha which is tunable parameter we can tune during online while using MATLAB/Simulink external mode. While tuning online we have achieved our required results. When we increased the value of alpha the rise time increases and when we decrease the value of alpha, we can observe that the rise time decreased, and its achieved maximum peak very quickly. By comparing both the results of PID & NPID parameters results, the NLPID parameters have better quality control over linear PID parameters.

CHAPTER 5

CONCLUSION

The main aim behind this research is that we designed the drilling rig and simulated in MATLAB and interfaced with TwinCAT-3, by building from MATLAB/Simulink to TwinCAT-3 then implemented physically on drilling rig in prototype setup. We analyzed the optimum values of rise time and overshoot in a real-time external mode of MATLAB/Simulink. During a real operation, drilling towers can operate in a variety of ways, including drilling, tripping, and so on. There are certain external disturbances and uncertainties for each mode of operation. Ziegler Nicholas method is best for manual tuning of the pid parameters. When we applied this method the value of K_i increased too much as compared to K_p which causes the system unstable or starts vibrating. To remove the vibration and distortion on the drill string hook, we applied linear and nonlinear pid controllers and tuned the pid parameters manually. The system behavior is observed when we move the drill string hook in upward/downward with joystick control. Before moving the drill string hook with joystick, I have added some safety features for the system. Joystick has Deadman button without pressing this button the joystick will not forward any signal to the motor and will not be working. The second feature is I have added $\pm 5\%$ of delay so joystick active after crossing $\pm 5\%$ on forward and backward direction. Sometime when we leave the joystick in mean position it vibrates very little, and it may cause any damage for joystick or for the system. Initially when I started to tune linear pid parameters manually by setting the $K_p=8.5$ and $K_i= 0.00000025$ then we observed that the drill string hook vibrates too much due to increase in the overshoot. Slowly and gradually when I tuned the pid parameters I reached at some values then system becomes almost stable and works smoothly. Those values are $K_p= 2$, and $K_i= 0.000040$. The aim is to tune as much as the system remove the disturbances and vibration completely and/or approximately. Then we applied nonlinear pid controller and, we tuned it manually same as linear pid. When I started tuning by setting $K_p= 8.5$, $K_i= 0.00000025$ and $\text{Alpha}= 0.2$. Then we observed that there is a lot of distortion in the drill string hook due to increase in the overshoot. Then slowly and gradually I tuning the nonlinear pid parameters when I

concluded that the drill string hook moves very smoothly and there is no vibration in the hook. Those values of nonlinear pid parameters are $K_p= 2.5$, $K_i= 0.00000025$ and $\alpha= 0.01$. The effects of unknown forces on the system, such as external disturbances which were observed during the operations were eliminated successfully by using this prescribed controller.

REFERENCES

- [1] S. Nobahar, "Robust and adaptive control design of a drilling during the operating modes," *Measurement and Control*, vol. 52, pp. 702-719, 2019.
- [2] O. U. OVERVIEW, "rtaautomation," [Online]. Available: <https://www.rtaautomation.com/technologies/opcu/>. [Accessed 21 May 2022].
- [3] H. Bansal and P. Shreeraman, "PID controller tuning techniques," *Journal of control engineering and technology*, vol. 2, no. 4, pp. 168-176, 2012.
- [4] K. Zaidner, "Nonlinear PID and its application in Process Control," in *IEEE 26th Convention of Electrical and Electronics Engineers in Israel*, 2010.
- [5] F. JANÍČEK, "Interconnecting MATLAB with TwinCAT," in *International Conference on Manufacturing Engineering, Quality and Production Systems.*, 2011.
- [6] D. Allouche, "Real-time control of a force feedback haptic interface via EtherCAT fieldbus," in *IEEE International Conference on Industrial Technology (ICIT) pp.441-446*, 2015.
- [7] H. Flayyih, "Design and implementation speed control system of DC motor based on PID control and MATLAB Simulink," *International Journal of Power Electronics and Drive Systems*, vol. 1, p. 127, 2020.
- [8] P. Lale, "Implementation of Voltage control of buck converter using Beckhoff programmable logic controller," *International Journal of Electrical Engineering and Computing*, vol. 2, pp. 86-93, 2021.
- [9] A. Karpagam, "Comparison of tuning methods of PID controller," *International Journal of Managment, Information Technology and Engineering*, vol. 8, pp. 1-8, 2014.
- [10] B. Shreeraman, "PID control tuning techniques," *Journal of control engineering and technology*, vol. 2, no. 4, pp. 168-176, 2012.
- [11] Z. Deng, "An optimized fractional order PID controller for suppressing vibration of AC motor," *Journal of Vibro engineering*, vol. 4, pp. 2205-2220, 2016.
- [12] S. Jamaluddin, "Implemenation of PID controller tuning using different evolution and genetic algorithms," *International journal of Innovative Computing, Information and Control*, vol. 11, pp. 7761-7779, 2012.
- [13] H. Chung, "An intelligent design for a PID controller for nonlinear systems," *Asian Journal of Control*, vol. 2, pp. 447-445, 2016.
- [14] D. GAUTAM, "Speed control of 3-phase induction motor using self-tuning fuzzy PID

controller and conventional PID controller.,” *International Journal of Information and Computation Technology*, vol. 12, pp. 1185-1193, 2014.

- [15] Z. Guo, “Theory and design of PID controller for nonlinear uncertain systems,” *IEEE Control Systems Letters*, vol. 3, pp. 643-648, 2019.
- [16] P. Srinivasan, “Design of nonlinear PID controller and nonlinear model predictive controller for a continuous stirred tank reactor,” *ISA transactions*, vol. 3, pp. 273-282, 2019.
- [17] S. Duan, “Design of an enhanced nonlinear PID controller,” *Mechatronics*, vol. 8, pp. 1005-1024, 2005.
- [18] L. Daley, “Optimal-tuning nonlinear PID control of hydraulic systems,” *Control Engineering Practice*, vol. 9, pp. 1045-1053, 2000.
- [19] H. Suhorukova, “Nonlinear PID controller and methods of its setting,” in *Open Conference of Electrical, Electronics and Information Sciences (eStream) IEEE pp.1-4*, 2017.
- [20] A. Korkmaz, “A simple approach to design of variable parameter nonlinear pid controller,” in *International Conference on Electrical Engineering and Applications pp 81-85*, 2011.
- [21] Z. Gu, “Application of nonlinear PID controller in main steam temperature control,” in *Asia-Pacific Power and Energy Engineering Conference IEEE*, March, 2009.
- [22] “BECKHOFF,” [Online]. Available: https://infosys.beckhoff.com/english.php?content=../content/1033/te1400_tc3_target_matlab/189856267.html&id. [Accessed 22 May 2022].
- [23] “BECKHOFF,” [Online]. Available: https://infosys.beckhoff.com/english.php?content=../content/1033/te1400_tc3_target_matlab/189856267.html&id. [Accessed 23 May 2022].
- [24] T. C. Volt Electric Motors, “Voltmotor,” [Online]. Available: <http://voltmotor.com.tr/downloads/pdf/en/catalogue-en.pdf>. [Accessed 22 May 2022].
- [25] A. E. E. Pepperl+Fuchs Products. [Online]. Available: https://www.pepperlfuchs.com/great_britain/en/classid_362.htm?view=productdetails&prodid=69_676. [Accessed 24 May 2022].
- [26] S. T. L. C. ESİT Products. [Online]. Available: https://www.esit.com.tr/en/load_cells/tb_load_cell/urun/176. [Accessed 23 May 2022].
- [27] Beckhoff. [Online]. Available: <https://www.beckhoff.com/en-en/products/ipc/embedded-pcs/cx20x0-intel-celeron-core-i7/cx2020.html>. [Accessed 23 May 2022].

- [28] Beckhoff. [Online]. Available: <https://www.beckhoff.com/en-en/products/i-o/ethercat-terminals/ek1xxx-bk1xx0-ethercat-coupler/ek1110.html>. [Accessed 23 May 2022].
- [29] ABB. [Online]. Available: <https://new.abb.com/drives/low-voltage-ac/industrial-drives/acs880-single-drives>. [Accessed 23 May 2022].
- [30] S. F. K. & H. A. Hammodi, "Design and implementation speed control system of DC motor based on PID control and MATLAB Simulink," *International Journal of Power Electronics and Drive System*, p. 127, 2020.
- [31] D. Industry. [Online]. Available: <https://www.directindustry.com/prod/avioni-electronic-technologies-co-ltd/product-201665-2072815.html>. [Accessed May 2022].