Comparison between data driven control system of DC shunt motor using System Identification Process & NARX model

by

Rifat Ahmed Khan (160021008)
Kamran Ahmmed (160021048)
Nayeer Awsaf Rahman (160021106)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

March, 2021

A Dissertation on

Quick Handover in LTE for High-Speed Metro Rails and Highways using PN Sequence Detection and Forward Handover

Approved by:

MD. Thesun Al-Amin

Assistant Professor, Supervisor,
Department of Electrical and Electronic Engineering,
Islamic University of Technology (IUT),
Board Bazar, Gazipur-1704.

Date: ……

Table of Contents

List of Figures

Acknowledgements

First and foremost, we offer our gratitude to the Almighty Allah (SWT) for giving us the capability to do this work with good health. This thesis is a result of research of eight months and this is by far the most significant scientific accomplishment in our life.

We are extremely grateful to my thesis supervisor MD. Thesun Al-Amin for the support and continuous guidance throughout the course of this work. He created a research environment for which we were able to explore many ideas without constraint. We have gained a wealth of knowledge and experience in science and engineering through his direction that is beyond value to our future endeavors.

And last but not the least we are thankful to our family, friends and well-wishers for their support and inspiration. Without them it would never have been possible for us to make it this far.

Rifat Ahmed Khan	**Kamran Ahmmed**	**Nayeer Awsaf Rahman**

Abstract

This paper presents data-driven control system of DC shunt motor by using system identification process and NARX model. In this paper we use component base modeling similar to real DC shunt motor by using simscape electronic systems for obtaining the input voltage and output speed of DC motor, the system identification toolbox and the nonlinear autoregressive with exogenous input (NARX) neural network for identification and obtaining the model of an object. The object model and training the neural network for data driven control system are developed by using MATLAB/SIMULINK platform. So, simulation results of this paper present the advantage of the suggested control method and the acceptable accuracy with respect to dynamic characteristics of the system. Also here we present advantage of one data driven control system over the other as well.

Chapter 1

Introduction

1.1 Introduction

In this chapter the background and motivation of this thesis work are overviewed, as well as the problem statement and the objective of this study.

1.2 Background and Motivation

DC shunt motors are widely used in robotic control systems, transportation, industrial control systems and other applications due to their dynamic characteristics, good speed regulation and high efficiency. On the other hand, in DC motor system may be occur such as changes in load dynamics, disturbances, variable and unpredictable inputs, unknown parameters. All that will result in high motor performance, therefore, the motor must be controlled and analysed. To analyse and design the controller of the motor, it is necessary to design its model. In this regard, to solve the problem the data driven control must be used. Data driven control is a designed controller when no plant model is available and it requires the measured input/output data about the plant, system identification process and controller design. In this case, let's consider component base modelling similar to a real DC motor by using simscape

electronic system because simscape allows to create models of physical systems rapidly within the MATLAB/SIMULINK platform for the researchers. Simscape electronic system allows building models of physical components based on physical connections that directly connect with simulink block0 and other modelling paradigms [1].

System identification is possible to solve the objective of designing mathematical models of dynamical system based on the perceived system data. The system identification toolbox is useful for representing dynamic systems and ensures some linear and nonlinear black-box model structures [1]. In this paper, nonlinear black-box model structures in the system identification toolbox have been used for representing continuous-time transfer function of the DC motor. In a broader sense, the system identification is no easy way to

derive their mathematical models for the most complex system unlike dc motors. In this case, artificial neural network (ANN) can be used for multiple input/outputs and the very complex system due to the fact that they can estimate any continuous nonlinear function to any acceptable accuracy in the system

identification [2]. So, nonlinear autoregressive with exogenous input (NARX) neural network is used to compare nonlinear black-box model structures, which generate transfer function of the plant in the system identification process. Today, we can use many kinds of control design in control system such as PID control design, model reference adaptive control design and model predictive control design. But proportional, integral, derivative controller (PID controller) still becomes very popular and is commonly used in control systems and other applications. PID controller design has the following special features: simple mathematical modelling, good reliability, high reliability, stabilization. Besides it, the system operates and corrects the error easily in the stationary state. Therefore, one should choose PID controller for the controller design.

The main goal of this paper is to implement data driven control system of the DC motor using system identification process. And another important task is to compare the performance of the nonlinear autoregressive with exogenous inputs (NARX) neural network and nonlinear black-box model structures, which generate transfer function of the plant in the system identification process. So finally, we are obtaining both estimated models using system data. Then we implement PID control mechanism to both estimated models and then compare the data-driven systems.

1.3 Thesis Outline

This whole dissertation report is structured as follows:

- **Chapter 1: Introduction**

  This chapter outlines the motivation of this study and explains the problem and the objective of the study.

- **Chapter 2: Speed control of DC shunt motor**

  Overview of the main components of DC shunt motor and its auxiliary components(DC chopper circuit, speed reference, controller circuit).

- **Chapter 3: Modeling of DC shunt motor**

  This chapter discusses the component based modeling of our plant model using simscape electronic system.

- **Chapter 4: Estimating plant model**

  In this chapter, we propose two alternative methods for obtaining an estimated model of our plant(estimating transfer function using system identification process and NARX model).

- **Chapter 5: Incorporating control on both estimated models**

  Both estimated models are controlled using PID controller. Basic overview of PID controller is given in this chapter

- **Chapter 6: Simulation results(comparison between both estimated model performances).**

  Simulation and comparison between both the data-driven systems.

- **Chapter 7: Conclusion**

  Summary of the main ideas and conclusions presented in this study, along with the considerations for the study and what could be done in the future.

Chapter 2

Speed Control of DC shunt motor

2.1 Introduction

This chapter focuses first on describing the auxiliary components of DC shunt motor(DC chopper circuit, speed reference, controller circuit).

2.2 Speed control using DC chopper citcuit

DC to DC converter is very much needed nowadays as many industrial applications are dependent upon DC voltage source. The performance of these applications will be improved if we use a variable DC supply. It will help to improve controllability of the equipments also. Examples of such applications are subway cars, trolley buses, battery operated vehicles etc. We can control and vary a constant DC voltage with the help of a chopper.

Chopper is a basically static power electronics device which converts fixed DC voltage/power to variable DC voltage or power. It is nothing but a high speed switch which connects and disconnects the load from source at a high rate to get variable or chopped voltage at the output.

Chopper can increase or decrease the DC voltage level at its opposite side. So, chopper serves the same purpose in DC circuit transfers in case of ac circuit. So it is also known as DC transformer.

A chopper is a particular kind of static device which is adept in convertingfixed dc voltage to variable dc voltage. A power semiconductor device is used as a switch in the overall chopper circuitry. This device can be a MOSFET, a GTO or an IGBT. These power electronic devices have a voltage drop of around 0.5-2.5 volts which has been neglected as such in the analysis carried out in this paper.

Chopper is basically a very high speed on/off switching device. Its basic job is to connect and disconnect the load from source at a great speed. In this way the constant dc voltage is chopped and we obtain a variable dc voltage. There are basically two time periods in chopper operation, one is the “on” time denoted as Ton and other is the “off” time denoted

as Toff. During Ton we get the constant source voltage Vs across the load and during Toff we get zero voltage across the load. The chopper plays the role of providing this pattern of providing alternate zero and Vs. In this way we obtain a chopped dc voltage in the load terminals. It is illustrated in Figure 2.1.

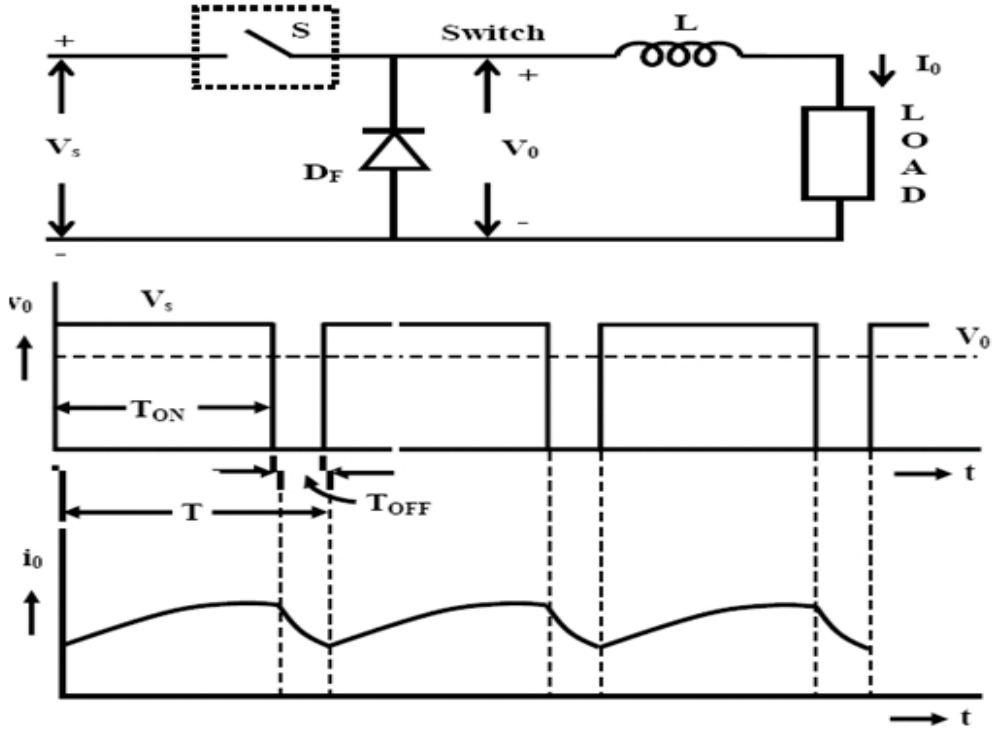


Figure 2.1 : Basic chopper circuit with the output voltage and output current curves

Vo = Average output voltage of the circuit.

Vs = Source voltage of the circuit.

$$Vo = Ton / (Ton + Toff) \times Vs \quad (2.1)$$

Ton / (Ton + Toff) = Duty cycle denoted by α.

The relation of the motor speed with the armature voltage

can be given from Equation (2)

$$\omega = (V - iaRa)/K\phi \quad (2.2)$$

Where ω is the speed in rad/sec, V is the armature voltage, ia is the armature current, Ra is the armature resistance, K is a constant and ϕ is the flux per pole.Thus we see that we can control the average output voltage and hence the speed by varying the duty cycle.A closed loop speed control mechanism is simulated in the MATLAB Simulink shown in Figure 2.2.

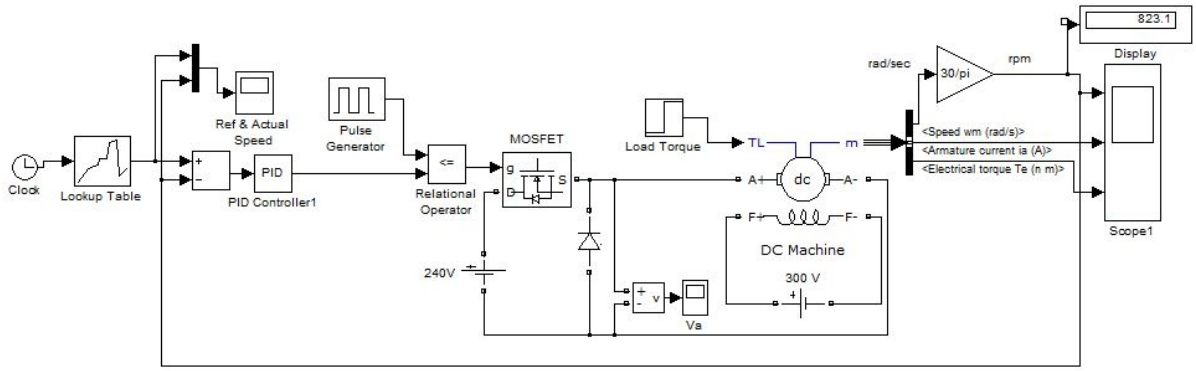


Figure 2.2 : closed loop speed control of DC shunt motor using chopper

The set of values speed at different time intervals is put as input in the lookup table. The motor speed is fed back to the relational operator through the PID controller which is a closed loop feedback mechanism controller. A PID controller continuously calculates an error value as the difference between a desired setpoint (Ref. speed) and a measured process variable (Measured Speed) The controller attempts to minimize the error over time by adjustment of a control variable, such as the position of a control valve, a damper, or the power supplied to a heating element, to a new value determined by a weighted sum.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{2.3}$$

Where Kp ,Ki , and Kd , all non-negative, denote the coefficients for the proportional, integral and derivative terms, respectively (sometimes denoted P, I, and D).

In case when the measured speed is less than the reference speed, triggering pulse is applied to the gate of the MOSFET. For the turn – ON time Ton the MOSFET behaves as a closed switch and the armature is fed with the supply and during the turn – OFF time Toff MOSFET behaves as a closed switch and the armature is cut off from the supply as shown in Figure 1. The duty cycle of the chopper circuit is preset at 0.7 (70%). As a result of this the motor speed begins to increase in order to reach up to the reference speed. If the measured speed is more than the reference speed, then no triggering pulse is sent to the gate of the MOSFET. As a result, the armature is cut off from the supply and the motor speed decreases. This close loop system always ensures that the measured speed is always near to the reference speed in order to achieve proper and optimal speed control [11].

2.3 Protection using Differential relay

Differential relay is a relay which is used to check the difference between the input and the output current of a system. The difference amongst the currents may be in phase angle or in magnitude or both. For hale and energetic operation, angle and magnitude variations must be zero. In case of fault there would be a difference between the input and the output current which in turn will give rise to a leakage current which is used to actuate the circuit breaker which in turn separates the faulty part from the rest of the system. So basically Differential relay is a sensing device which senses any fault in the system and it the circuit breaker which performs the separation [12].

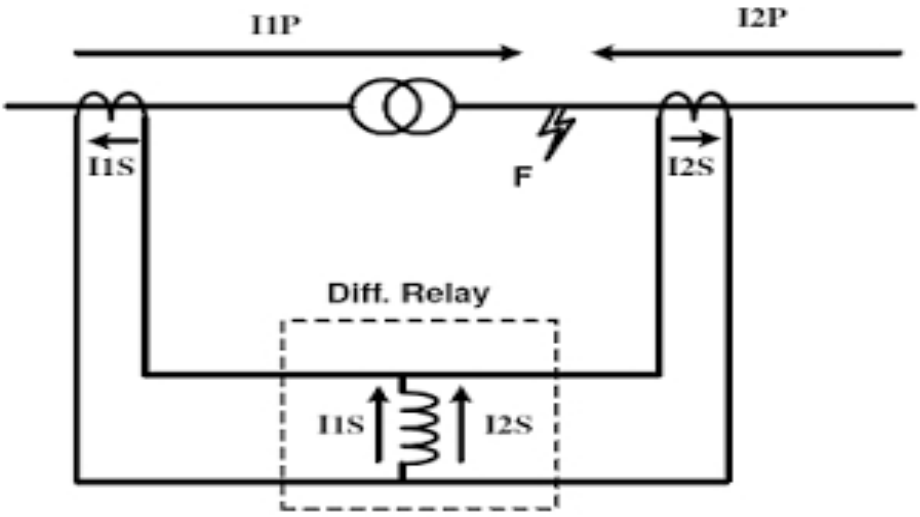


Figure 2.3: Differential relay in internal fault condition

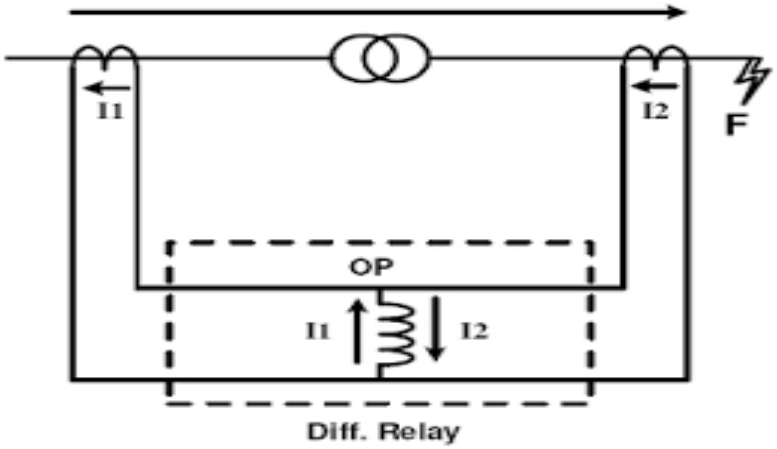


Figure 2.4: Differential relay in external fault condition

2.4 Conventional Controller

The design of methodologies, concepts, algorithms, and innovations for the design of process control systems that can adapt, self-develop, self-organize, self-evaluate, and self-improve is a major problem in the current control industries.

Over the years, specialists have used conventional proportional integral derivative controller to control procedures and mechanisms in the industry. Because these are simple, robust and low cost maintenance too.Conventional control systems cause transient and steady-state problems including such overshoot, settling time, and rise time plague traditional control systems.

To overcome these challenges, a range of techniques and adjustments have already been used, including auto tuning of proportional integral derivative (PID) controls, adaptive techniques, and compensation techniques

2.4.1 Proportional Integral Controller

The P-I controller contains a proportional in addition as AN integral term within the forward path, The integral controller has the property of creating the steady-state error zero for a step amendment, though a P-I controller makes the steady-state error zero, Since most of the method cannot work with AN offset, they have to be controlled at their set points and to realize this, additional intelligence should be more to a proportional controller ANd this can be achieved by providing an integral action to the initial proportional controller. that the controller becomes proportional –Integral controller .PI Controller as long as error is gift the controller keeps dynamical its output and once the error is zero or it disappears the controller doesn't change its output. Integration is that the mode that removes the offset or the error however typically it's going to create transient response worse. In PI Controller the output of the controller is modified proportional to the integral of the error .

Since most of the method cannot work with associate degree offset, they must be controlled at their set points and so as to achieve this, further intelligence should be extra to proportional controller and this can be achieved by providing an integral action to the first proportional controller. that the controller becomes proportional –Integral Controller .Under PI Controller as long as error is gift the controller keeps dynamical its output and once the error is zero or it disappears the controller doesn't amendment its output [13].

Integration is that the mode that removes the offset or the error however typically it's going to create transient response worse. In PI Controller the output of the controller is changed proportional to the integral of the error.

The mathematical expression of the PI Controller is:

$$u(t) = K_p\, e(t) + K_i \int_0^t e(t)dt \tag{2.4}$$

Where, Ki = Integral gain of the PI controller.

PI Controller has the subsequent disadvantages:

- The response is sluggish at the high worth of the integral time TN.
- The management loop could oscillate at the little worth of integral time TN.

2.4.2 PID Controller

The combination of proportional, integral and by-product control action is named inflammatory disease management action. inflammatory disease controllers are usually accustomed regulate the time-domain behaviour of the many differing kinds of dynamic plants. These controllers square measure very standard as a result of they will usually give sensible closed-loop response characteristics. Where it will be assumed that the plant may be a DC motor whose speed should be accurately regulated [11].The PID controller is placed within the forward path, so its output becomes the voltage applied to the motor's coil the feedback signal may be a speed, measured by a tach .the output speed signal C (t) is summed with a reference or command signal R (t) to create the error signal e (t).The corresponding 3 adjustable PID parameters are most typically Selected to be 1. Controller gain- (increased value provides a lot of proportional action and quicker control) 2. Integral time- (decreased value provides a lot of integral action and quicker control) 3. spinoff time- (increased value provides a lot of derivative action and quicker control) Although the PID controller has solely 3 parameters, it is not straightforward, while not a scientific procedure, to seek out sensible values for them. In fact, a visit to a method plant can usually show that an oversized variety of the PID controllers are poorly tuned. PID Controller includes all the 3 management actions i.e. proportional, integral and spinoff. Also,

1. A PID controller calculates and outputs a corrective action, that corrects the error between the method output and also the desired set point that adjusts the method consequently and rapidly.

2. The output of the controller or the manipulated variable is obtained by adding P, I and D components and their associated constant [14].

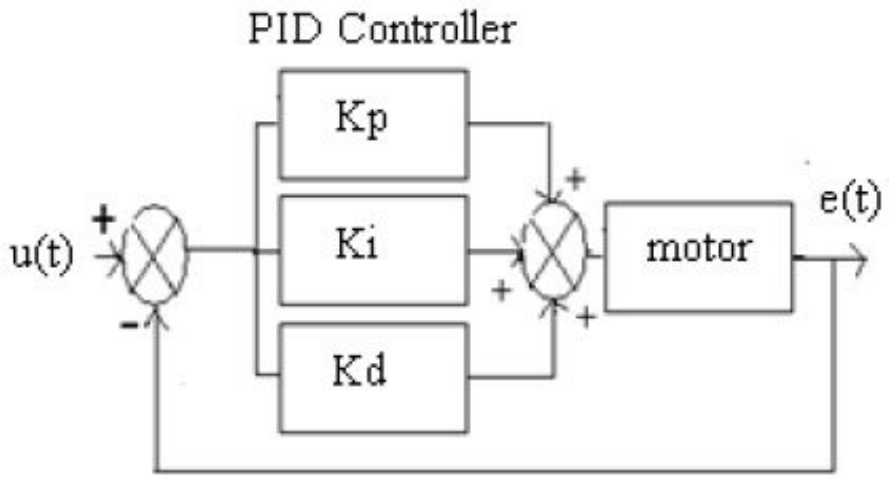


Figure 2.5: PID controller block diagram

2.4.3 Neural Network Controller

A neural network includes easy components operative in parallel. In fact, it's a massively parallel distributed processor that stores experiential information and makes it on the market to be used once required. In victimisation neural networks for system identification, coaching knowledge are often obtained by observant the input-output behavior of a plant. This method is named as “one step ahead prediction” and also the structure is named Time Delayed Neural network. Equation (3) are often simply enforced. The planned methodology are often used as adaptive or non-adaptive controller. If learning method continues, the controller are going to be associate degree adaptive controller. In non-adaptive case, learning method is dead as offline or for a precise amount of your time. In planning associate degreed coaching an ANN to emulate a perform, the sole mounted parameters square measure the quantity of inputs and outputs of the ANN, that square measure supported the input/output variables of the perform. the selection of the quantity of hidden neurons relies on expertise. it's conjointly wide accepted that most of 2 hidden layers square measure decent to find out any capricious non-linearity. The back-propagation coaching technique

adjusts the weights and bias altogether connecting links within the nodes so the distinction between the particular output and target output square measure decreased for all given coaching patterns the time delayed neural network could be a multilayer neural network and has four inputs. The MNN has one hidden layers, that comprises two neurons with tan sigmoid activation perform. The output layer of the NN has only 1 somatic cell with pure linear activation perform. Neural networks square measure terrific tools, which enable the event of quantitative expressions while not compromising the celebrated complexness of the matter. Neural networks agree the human brain within the following 2 ways: A neural network acquires information through learning. A neural network's information is keep among inter-neuron affiliation strengths referred to as conjunction weights An Artificial Neural Network (ANN) is associate degree IP paradigm that's impressed by the means biological nervous systems, like the brain, method info [1]. The key component of this paradigm is that the novel structure of the knowledge process system. it's composed of an oversized range of extremely interconnected process components (neurons) operating in unison to resolve specific issues. ANNs, like individuals, learn by example. associate degree ANN is organized for a particular application, like pattern recognition or knowledge classification, through a learning method. Learning in biological systems involves changes to the conjunction connections that exist between the neurons. this is often true of ANNs yet. actuality power and advantage of neural networks lies in their ability to represent each linear and non-linear relationships and in their ability to find out these relationships directly from the information being sculptured. ancient linear models square measure merely inadequate once it involves modelling knowledge that contains non-linear characteristics. The neural network consists of junctions that square measure connected with LINKS, conjointly known as process units. for every junction variety is ordered, this range is named weight. The weights square measure the tools for the long distance info storing within the neural network, the training method occurring with the suitable modification of weights [7]. These weights square measure changed so the network input/output behaviour is in consonance with the surroundings, which give the input file.

2.4.4 Fuzzy Logic Controller

Fuzzy logic management is predicated on logical relationship. Fuzzy data, that area unit accustomed show linguistic variables. The similarities between mathematical logic and fuzzy pure mathematics that's similar that of relation between formal logic and pure

mathematics. Fig. a pair of shows a general FLC structure. FLC is processed for linguistic definitions, whereas alternative controllers work on the accuracy and parameters of system model. within the coming up with FLC, ne'er want data of system model, as a controller. Weather, very little data of management method might result unsatisfactory . A mathematical logic model may be a logical-mathematical procedure supported associate “IF-THEN” rule system that mimics the human approach if thinking in procedure type. Generally, a fuzzy rule system categorised in four modules .1.Fuzzification a pair of.Fuzzy reasoning three.Rule base four.Defuzzification Fuzzification: the method of changing a numerical variable (real variety or crisp variables) into a linguistic variable (fuzzy number) is termed Fuzzification. By relatively tiny variety of membership functions to variable, In another manner it means that the assignment of linguistic price. Fuzzy reasoning: the reality price for the premise of every rule is computed beneath inference, and follow on the conclusion a part of every rule. for every rule this leads to one fuzzy set to be assigned to every output variable. principally reasoning rules area unit used as MIN or PRODUCT. The output membership perform is clipped off in MIN reasoning, at a height like the rule premise's computed degree of truth (fuzzy logic).The output membership perform is scaled by the rule premise's computed degree of truth in PRODUCT reasoning Rule base: For the rule bases a classic interpretation of Mamdani was used. beneath rule base, outputs rules area unit made. the foundations organized in “If Then” format and formally the If facet is termed the conditions and also the Then facet is termed the conclusion. A rule base controller is just comprehendible and simple to keep up for a non- specialist user and the same controller may well be enforced mistreatment standard techniques. Defuzzification: Defuzzification may be a method within which crisp output is obtained by the fuzzy output. In alternative words, method to convert fuzzy output into crisp variety. Here in Defuzzification strategies within which 2 of the additional common Techniques area unit named as center of mass and most strategies. in keeping with the center of mass technique, In crisp price of the output Variables area unit computed by finding the variable price of the centre of gravity of the membership perform for the fuzzy price. within the most technique, one amongst the variable prices at that the fuzzy set has its most truth price is chosen as crisp value for the output variable.

Chapter 3

Modeling of DC shunt motor

3.1 Introduction

Here the component based modeling of dc shunt motor is given in details. The data driven control system studied here consists of diode – based AC to DC three phase rectifier, power MOSFET chopper and DC shunt motor.

The simulation is performed in Simulink MATLAB V. 7.7.0.471 (2008), License No. 16051.

The system on which the simulation was performed had the following configuration:

- Processor: Intel® Core ™ i7 – 6600 CPU@
- 2.98GHz
- Memory: 8GB
- Operating System: Microsoft Windows 10
- Professional 64-Bit

3.2 Types of DC Motor

There are 3 main types of DC motor that are available:- Series, Shunt and Compound. These terms relate to the type of connection of the field windings with respect to the armature circuit [15].

3.2.1 DC Series Motor

A DC series motor will have its field windings connected in series with the armature. The series winding will have relatively few turns of larger wire or copper strip which are capable of carrying the full load current of the motor. On starting, because the windings are low resistance, a large current can be drawn producing a high starting torque.

This is an advantage for high starting loads such as traction, crane and other heavy applications. The speed of a series motor is dependent on the load, so when the full load current flowing through the circuit has reduced, the speed will have increased.

In some instances, the motors speed could potentially increase to a level above the recommended maximum. For this reason, a series motor should not be connected to its load with a belt.

3.2.2 DC Shunt Motor

In a DC Shunt motor the field winding is connected in parallel (shunt) with the armature. The shunt winding is wound from many turns of small copper wire and since it is connected across the DC field supply, its field current will be constant.

The motor will run up to rated speed and this will not be greatly affected by a change in load. The starting torque will be less than a similar sized series motor but if this is not required then a constant speed shunt motor may be preferable for the application.

DC shunt motors can be used for many applications such as plastics or wire extrusion. We carry a stock of small DC shunt wound motors in IP23 IC06 format (drip proof force ventilated). Other DC motors can be made on request

3.2.3 DC Compount Motor

With a DC compound motor, the majority of the field is wound for a shunt field but with a few turns of series winding on top. The shunt is connected across the field supply and the series turns are connected in series with the armature. This provides a motor with a combination of the shunt and series characteristics.

The starting torque will be higher than a shunt motor but not as high as a series motor. The speed will change with load and the amount will depend on the % of field space allocated to the series winding. The series field can be arranged to either increase or decrease the speed with load. Applications for these motors vary but are often for larger applications such as unwind brake generators, conveyors, mixers etc.

A form of DC compound motor can also be used where the supply is from batteries with a wide range of volts. In this instance both the field and armature have the same voltage applied and by using the compound winding this helps to keep the speed within an acceptable range.

3.3 Speed Regulation of DC Shunt Motor

When voltage is applied to a DC shunt motor, the armature draws current sufficient to produce a strong magnetic field, which interacts with the magnetic field produced by the shunt windings, causing the armature to rotate [16]. The rotating armature (aka rotor) produces a back EMF, which opposes the armature voltage and reduces the armature current. If the load on the motor is increased, the armature rotation slows and back EMF is reduced, since back EMF is proportional to speed.

$$E_b = \frac{\phi \cdot P \cdot Z \cdot N}{60} \tag{3.1}$$

Where:

E_b = back EMF

Φ = flux

P = number of poles

Z = number of coils

N = rotational speed

With less back EMF voltage and a constant supply voltage (E), the net voltage increases.

$$E_{net} = E - E_b \tag{3.2}$$

The increase in net voltage results in an increase in armature current. Since torque is proportional to armature current, torque also increases.

$$T = \frac{\phi \cdot P \cdot I_a \cdot Z}{2 \cdot \pi \cdot A} \tag{3.3}$$

Where:

T = torque

A = area

Finally, this increased torque allows the motor to increase its speed and compensate for the slowdown due to loading. Hence, a DC shunt motor is able to self-regulate its speed, and can be referred to as a constant speed motor.

3.4 Applications of DC Shunt Motor

Because of their self-regulating speed capabilities, DC shunt motors are ideal for applications where precise speed control is required. Keep in mind, however, that they cannot produce high starting torque, so the load at startup must be small. Applications that meet

these criteria and are suitable for DC shunt motors include machines tool such as lathes and grinders, and industrial equipment such as fans and compressors [17].

3.5 Motor Model

The motor model is designed in simulink initially then used real life data sets to solve a real life problem on how a dc shunt motor system can gain its speed using various controllers

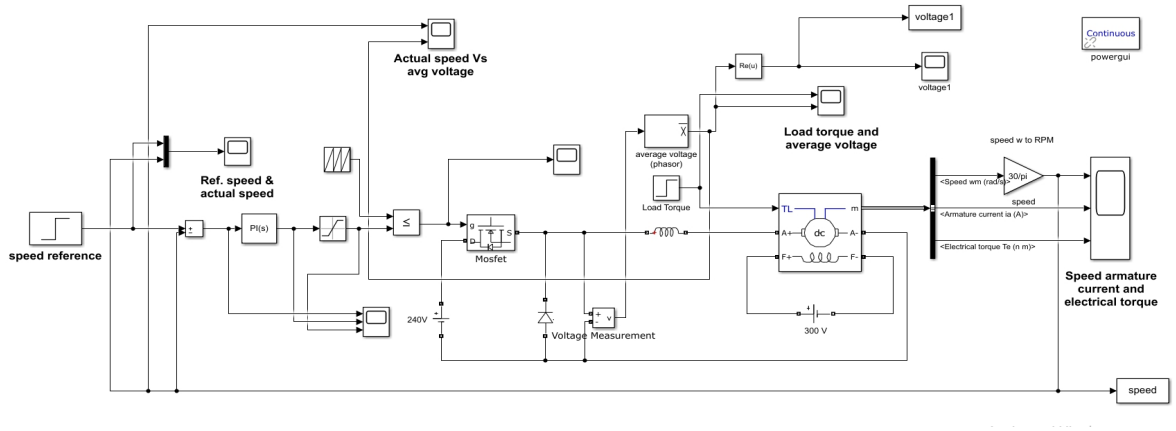


Figure 3.1: component based modeling of a DC shunt motor

3.5.1 Speed Reference Block

The Step block provides a step between two definable levels at a specified time. If the simulation time is less than the Step time parameter value, the block's output is the Initial value parameter value. For simulation time greater than or equal to the Step time, the output is the Final value parameter value.

The block's numeric parameters must be of the same dimensions after scalar expansion. If the Interpret vector parameters as 1-D option is off, the block outputs a signal of the same dimensions and dimensionality as the parameters. If the Interpret vector parameters as 1-D option is on and the numeric parameters are row or column vectors (i.e., single row or column 2-D arrays), the block outputs a vector (1-D array) signal; otherwise, the block outputs a signal of the same dimensionality and dimensions as the parameters [18].

This block determines a initial speed of 1000 rpm .

3.5.2 DC Machine

Here our dc machine consists of

240 voltage source voltage

300 voltage field voltage

Load 100 N

3.5.3 Chopper Circuit

Chopper is basically a very high speed on/off switching device. Its basic job is to connect and disconnect the load from source at a great speed. In this way the constant dc voltage is chopped and we obtain a variable dc voltage. There are basically two time periods in chopper operation, one is the "on" time denoted as Ton and other is the "off" time denoted as Toff. During Ton we get the constant source voltage Vs across the load and during Toff we get zero voltage across the load. The chopper plays the role of providing this pattern of providing alternate zero and Vs. In this way we obtain a chopped dc voltage in the load Terminals

3.5.4 Mosfet

Here the Mosfet works as a switch . When the measured speed is less than the reference speed, triggering pulse is applied to the gate of the MOSFET. For the turn – ON time Ton the MOSFET behaves as a closed switch and the armature is fed with the supply and during the turn – OFF time Toff MOSFET behaves as a closed switch and the armature is cut off from the supply as shown in Figure 1. The duty cycle of the chopper circuit is preset at 0.7 (70%). As a result of this the motor speed begins to increase in order to reach up to the reference speed. If the measured speed is more than the reference speed, then no triggering pulse is sent to the gate of the MOSFET. As a result, the armature is cut off from the supply and the motor speed decreases. This close loop system always ensures that the measured speed is always near to the reference speed

3.5.5 Motor Dynamics Outputs

Generally, three characteristic curves are considered important for DC motors which are, (i) Torque vs. armature current, (ii) Speed vs. armature current and (iii) Speed vs. torque.

These are explained below for each type of DC motor. These characteristics are determined by keeping the following two relations in mind.Ta ∝ ϕ.Ia and N ∝ Eb/ϕ

These above equations can be studied at - emf and torque equation of dc machine. For a DC motor, magnitude of the back emf is given by the same emf equation of a dc generator i.e. Eb = PϕNZ / 60A. For a machine, P, Z and A are constant, therefore, N ∝ Eb/ϕ

3.5.5.1 Torque Vs. Armature Current (Ta-Ia)

In case of DC shunt motors, we can assume the field flux ϕ to be constant. Though at heavy loads, ϕ decreases in a small amount due to increased armature reaction. As we are neglecting the change in the flux ϕ, we can say that torque is proportional to armature current. Hence, the Ta-Ia characteristic for a dc shunt motor will be a straight line through the origin. Since heavy starting load needs heavy starting current, shunt motor should never be started on a heavy load [19].

3.5.5.2 Speed Vs. Armature Current (N-Ia)

As flux ϕ is assumed to be constant, we can say N ∝ Eb. But, as back emf is also almost constant, the speed should remain constant. But practically, ϕ as well as Eb decreases with increase in load. Back emf Eb decreases slightly more than ϕ, therefore, the speed decreases slightly. Generally, the speed decreases only by 5 to 15% of full load speed. Therefore, a shunt motor can be assumed as a constant speed motor. In speed vs. armature current characteristic in the following figure, the straight horizontal line represents the ideal characteristic and the actual characteristic is shown by the dotted line [20].

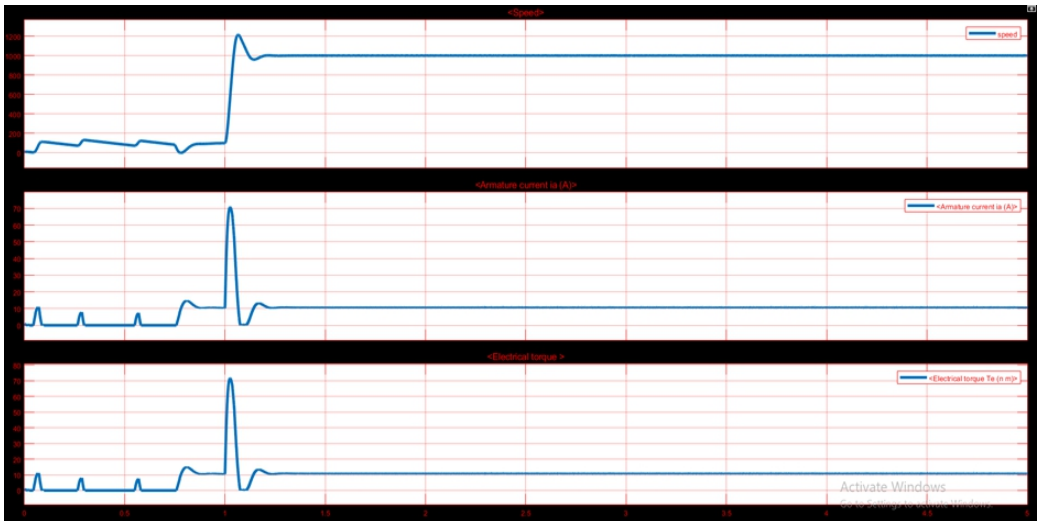


Figure 3.2: Speed current and electrical torque graph of dc shunt motor

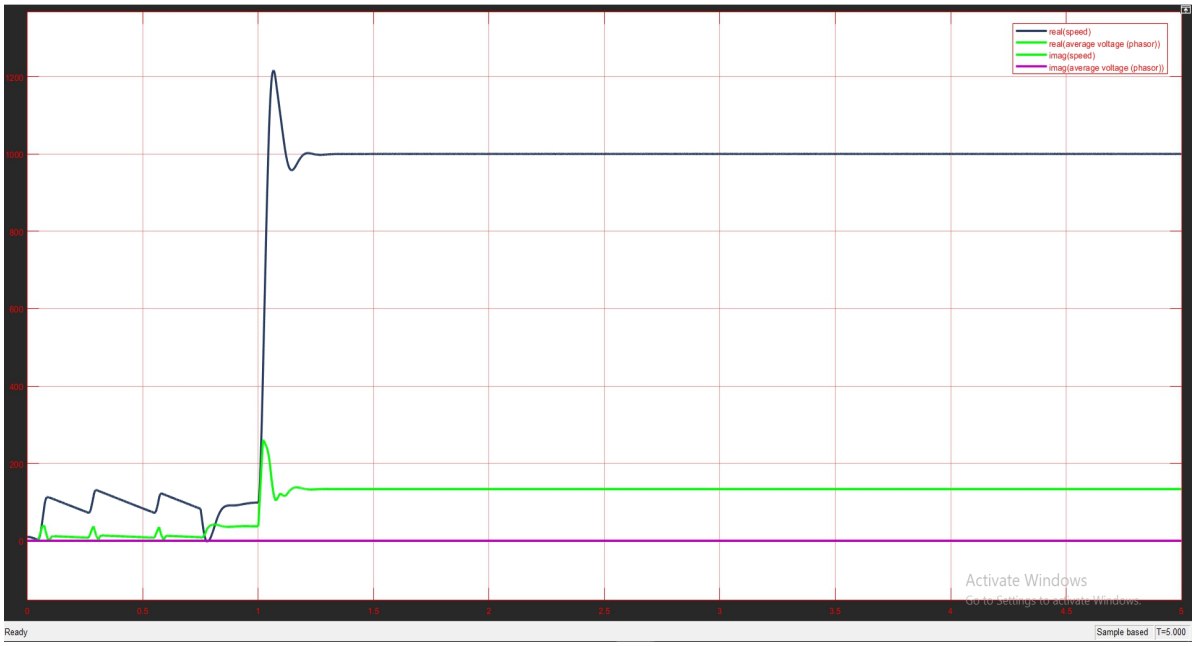


Figure 3.3: Speed vs Voltage curve

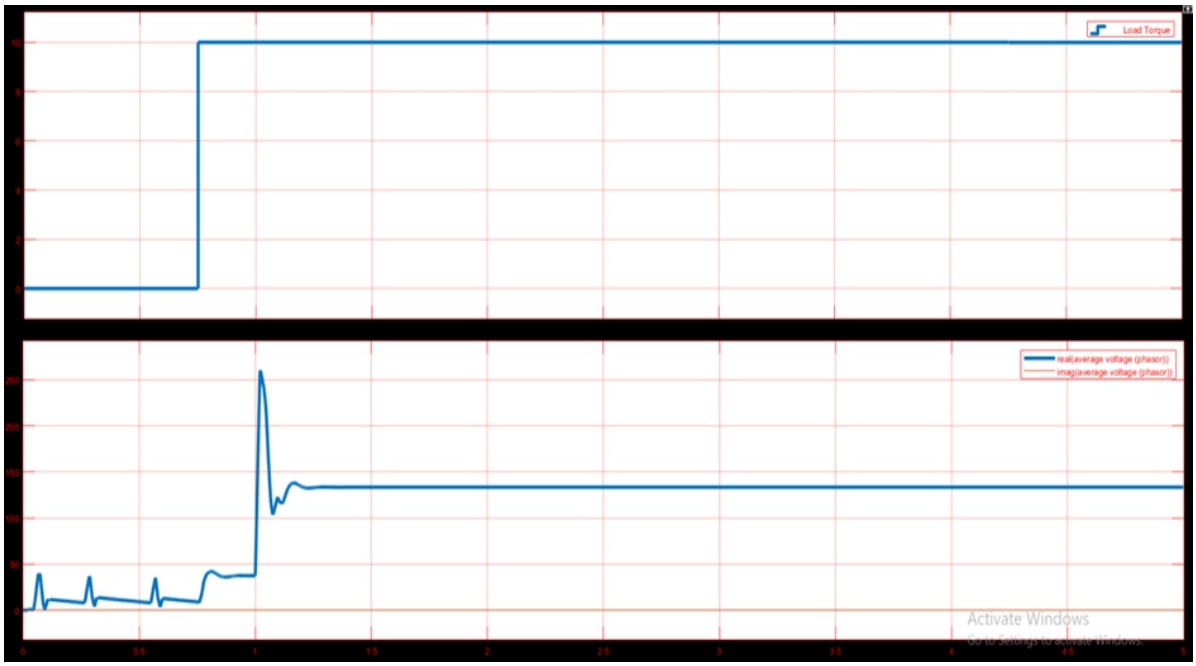


Figure 3.4: Load torque and voltage

Chapter 4

Estimating Plant Model

4.1 Introduction

In engineering, a transfer function (also known as system function[1] or network function) of an electronic or control system component is a mathematical function which theoretically models the device's output for each possible input.

Here we have used system identification toolbox in matlab to estimate transfer function of the plant .

In this chapter we use the system data obtained from the previous chapter to estimate two equivalent plant models. First we obtain a model using System Identification Process and then we estimate another plant model by training a NARX network using the same system data.

4.2 System Identification Process

System identification is a methodology for building mathematical models of dynamic systems using measurements of the input and output signals of the system.

The process of system identification requires that you:

- Measure the input and output signals from your system in time or frequency domain.
- Select a model structure.
- Apply an estimation method to estimate values for the adjustable parameters in the candidate model structure.
- Evaluate the estimated model to see if the model is adequate for your application needs [21].

4.2.1 Estimating Transfer Function Using System Identification Toolbox

System Identification Toolbox™ provides MATLAB® functions, Simulink® blocks, and an app for constructing mathematical models of dynamic systems from measured input-output knowledge. It enables you to produce and use models of dynamic systems not simply modeled from initial principles or specifications. you'll use time-domain and frequency-

domain input-output knowledge to spot continuous-time and discrete-time transfer functions, method models, and state-space models. The chest conjointly provides algorithms for embedded on-line parameter estimation.

The chest provides identification techniques like most probability, prediction-error step-down (PEM), and mathematical space system identification. To represent system dynamics, you'll estimate Hammerstein-Wiener models and nonlinear ARX models with ripple network, tree-partition, and sigmoid network nonlinearities. The chest performs grey-box system identification for estimating parameters of a user-defined model. you'll use the known model for system response prediction and plant modeling in Simulink. The chest conjointly supports time-series knowledge modeling and time-series prognostication [21].

Figure 4.1: System Identification Toolbox interface (with our imported dataset)

Here we have taken voltage data as input and speed data as output and imported both of them. There's no preprocessing done here , while estimating transfer function models we used IV method and for iteration method we used Lavenberg Marquart system for best accuracy . Same process has been repeated for variable poles and zeros and we got different fit to estimation value.

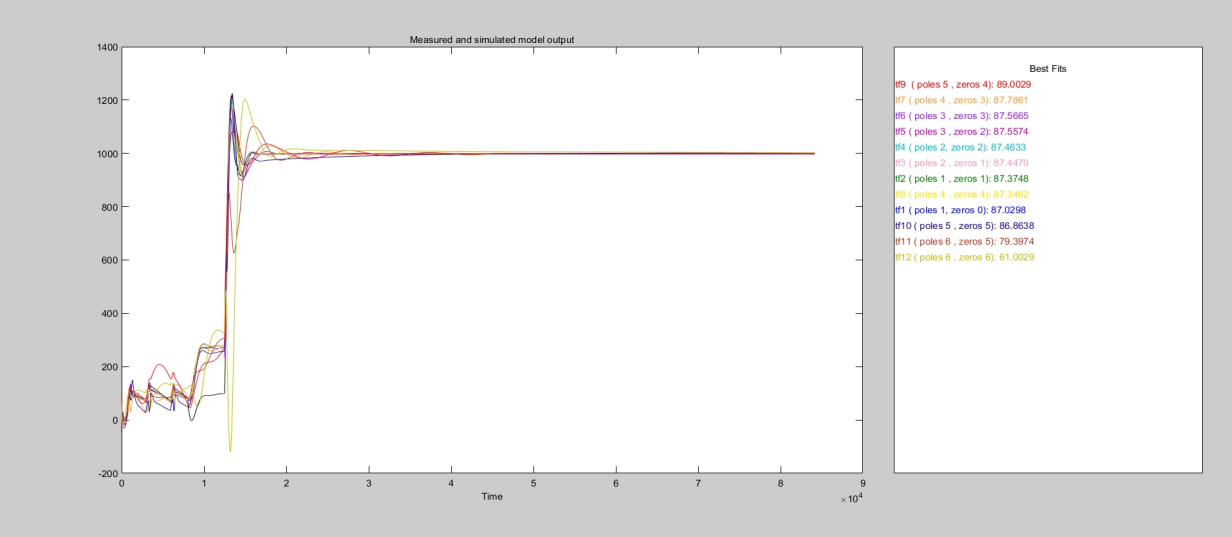


Figure 4.2: All examined outputs of estimated transfer functions by varying number of poles and zeros(with fit to estimations)

Here, the transfer function with the highest fit to estimation is the best one for our case. TF9(number of poles: 5, number of zeros: 4) has the highest fit to estimation: 89.009%. Isolating tf9 the response is shown in fig 4.3

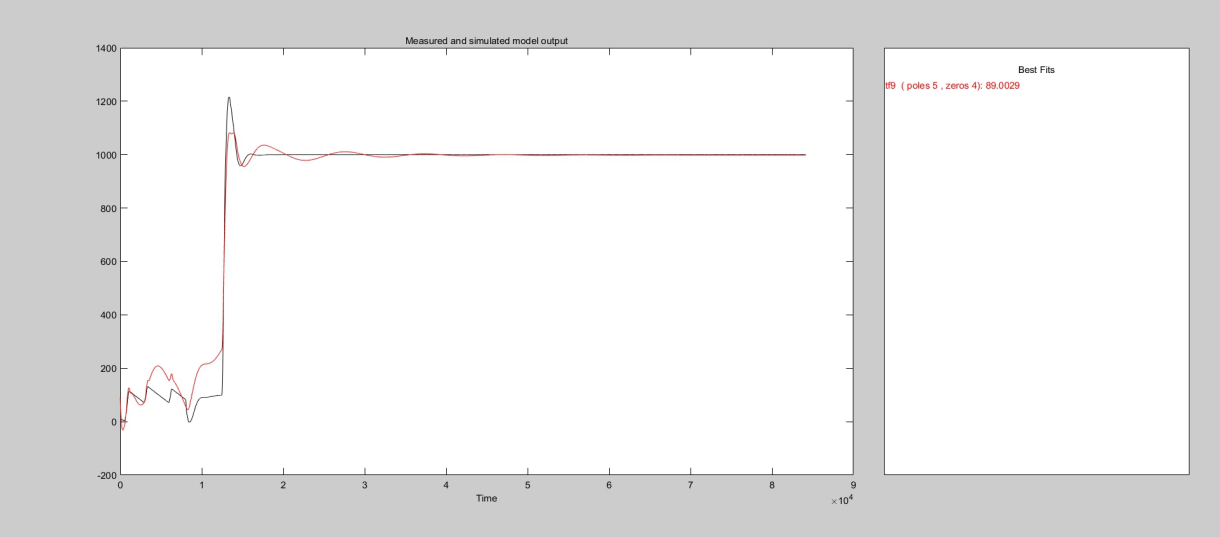


Figure 4.3: output of tf9 having 5 poles and 4 zeros

Estimation of the best transfer function is also graphically represented in fig 4.4, where a plot of fit to estimations vs number of poles and zeros is given.

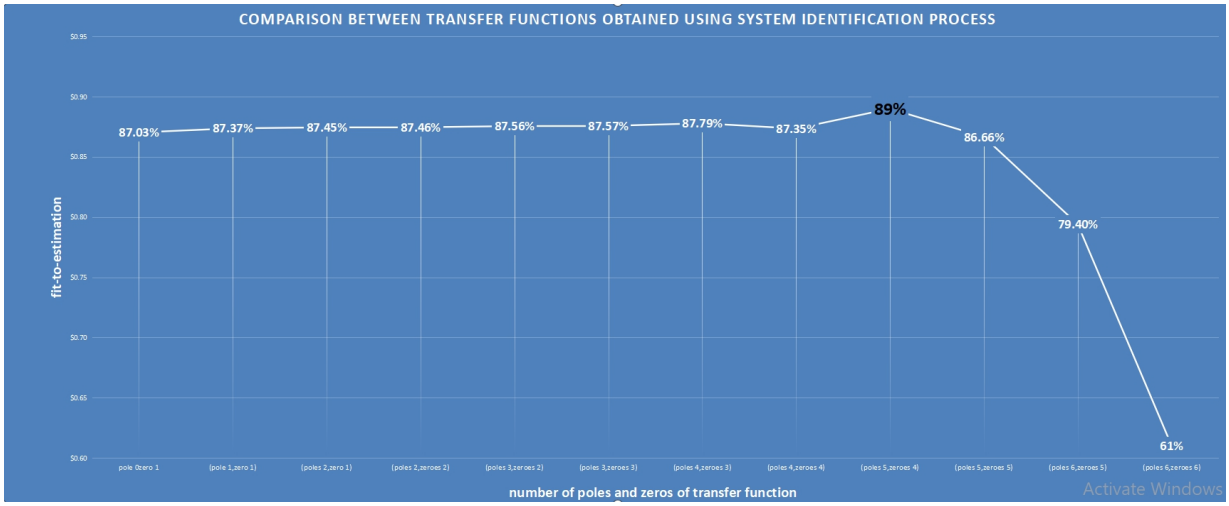


Figure 4.4: Fit to estimation vs number of poles and zeros

From here it is also evident that tf9 is the best transfer function for our case. The data/model information of tf9(5 poles and 4 zeros) is shown in figure 4.5

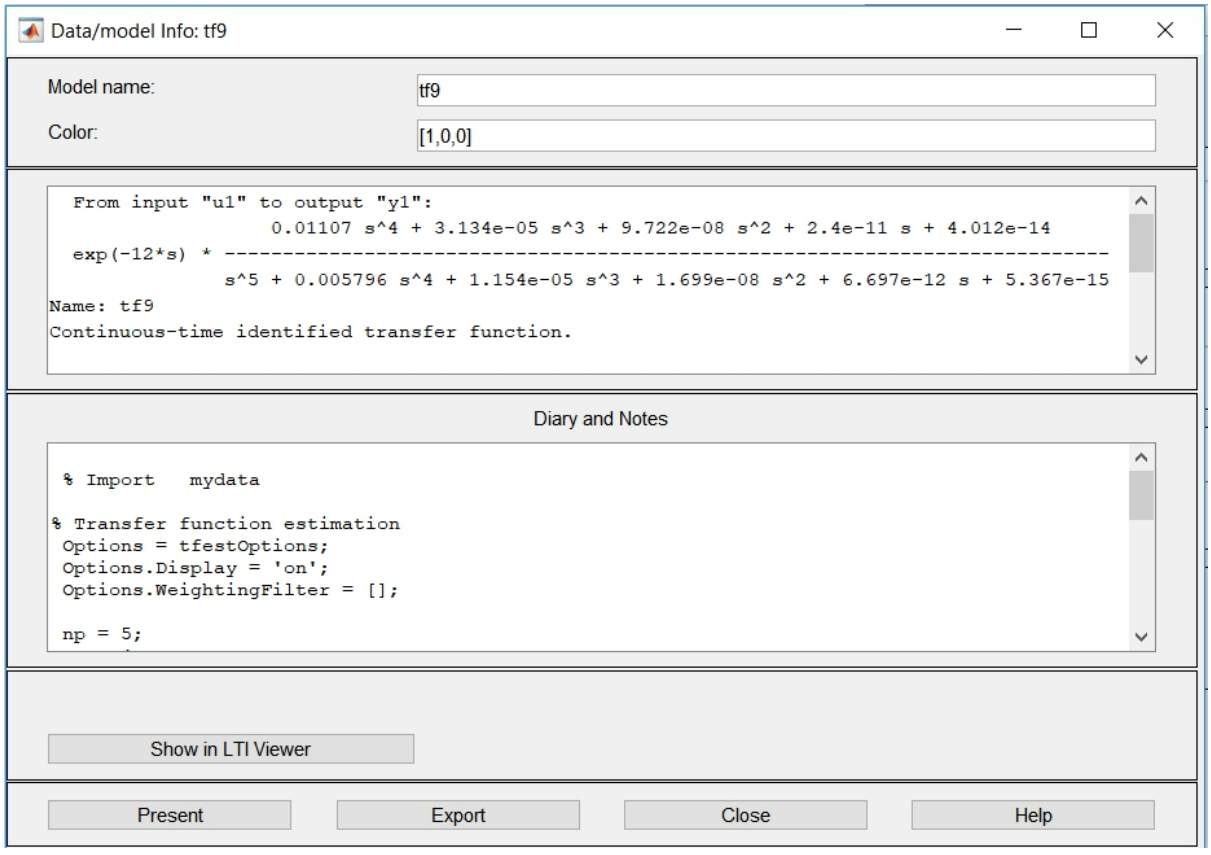


Figure 4.5: Data/Model info of the estimated transfer function

This is our estimated transfer function.

4.2.2 Implementing the estimated motor model in SIMULINK

The entire plant model can be estimated in Simulink by a transport delay and a transfer function block. This simple model should have all the characteristics of our original plant model.

4.2.2.1 Transport delay block:

The Transport Delay block delays the input by a specified amount of time. You can use this block to simulate a time delay. The input to this block should be a continuous signal.

At the start of simulation, the block outputs the Initial output parameter until the simulation time exceeds the Time delay parameter. Then, the block begins generating the delayed input. During simulation, the block stores input points and simulation times in a buffer. You specify this size with the Initial buffer size parameter.

When you want output at a time that does not correspond to times of the stored input values, the block interpolates linearly between points. When the delay is smaller than the step size, the block extrapolates from the last output point, which can produce inaccurate results. Because the block does not have direct feedthrough, it cannot use the current input to calculate an output value. For example, consider a fixed-step simulation with a step size of 1 and the current time at t = 5. If the delay is 0.5, the block must generate a point at t = 4.5. Because the most recent stored time value is at t = 4, the block performs forward extrapolation.

The Transport Delay block does not interpolate discrete signals. Instead, the block returns the discrete value at the required time.

This block differs from the Unit Delay block, which delays and holds the output on sample hits only [22].

Here the time delay is 12, the block parameters are shown in fig 4.6

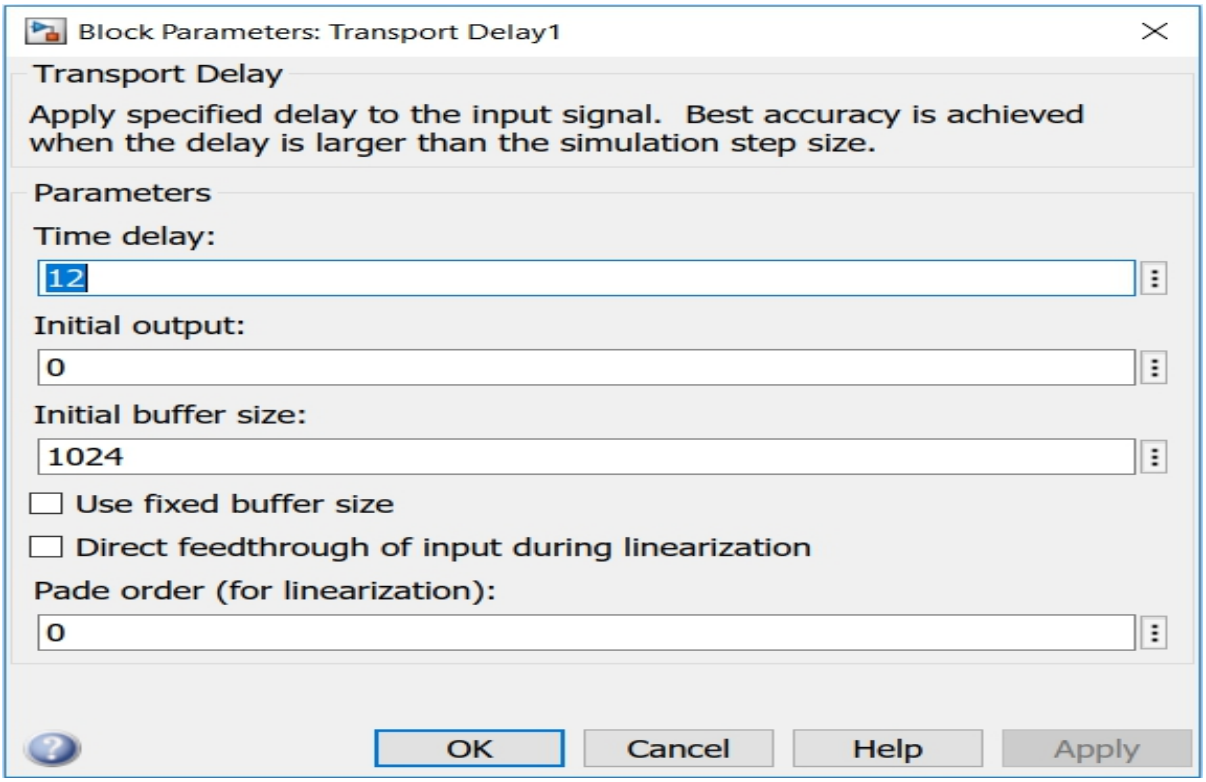


Figure 4.6: block parameters of transport delay

4.2.2.2 Transfer function block:

A transfer function is a convenient way to represent a linear, time-invariant system in terms of its input-output relationship. It is obtained by applying a Laplace transform to the differential equations describing system dynamics, assuming zero initial conditions. In the absence of these equations, a transfer function can also be estimated from measured input-output data.

Transfer functions are frequently used in block diagram representations of systems and are popular for performing time-domain and frequency-domain analyses and controller design. The key advantage of transfer functions is that they allow engineers to use simple algebraic equations instead of complex differential equations for analyzing and designing systems [23].

The Transfer Fcn block models a linear system by a transfer function of the Laplace-domain variable s. The block can model single-input single-output (SISO) and single-input multiple-output (SIMO) systems.

The Transfer Fcn block assumes the following conditions:

- The transfer function has the form

$$H(s) = \frac{y(s)}{u(s)} = \frac{num(s)}{den(s)} = \frac{num(1)s^{nn-1} + num(2)s^{nn-2} + \ldots + num(nn)}{den(1)s^{nd-1} + den(2)s^{nd-2} + \ldots + den(nd)}, \quad (4.1)$$

where u and y are the system input and outputs, respectively, nn and nd are the number of numerator and denominator coefficients, respectively. num(s) and den(s) contain the coefficients of the numerator and denominator in descending powers of s.

- The order of the denominator must be greater than or equal to the order of the numerator.
- For a multiple-output system, all transfer functions have the same denominator and all numerators have the same order.

The transfer function block parameters are given in figure 4.7

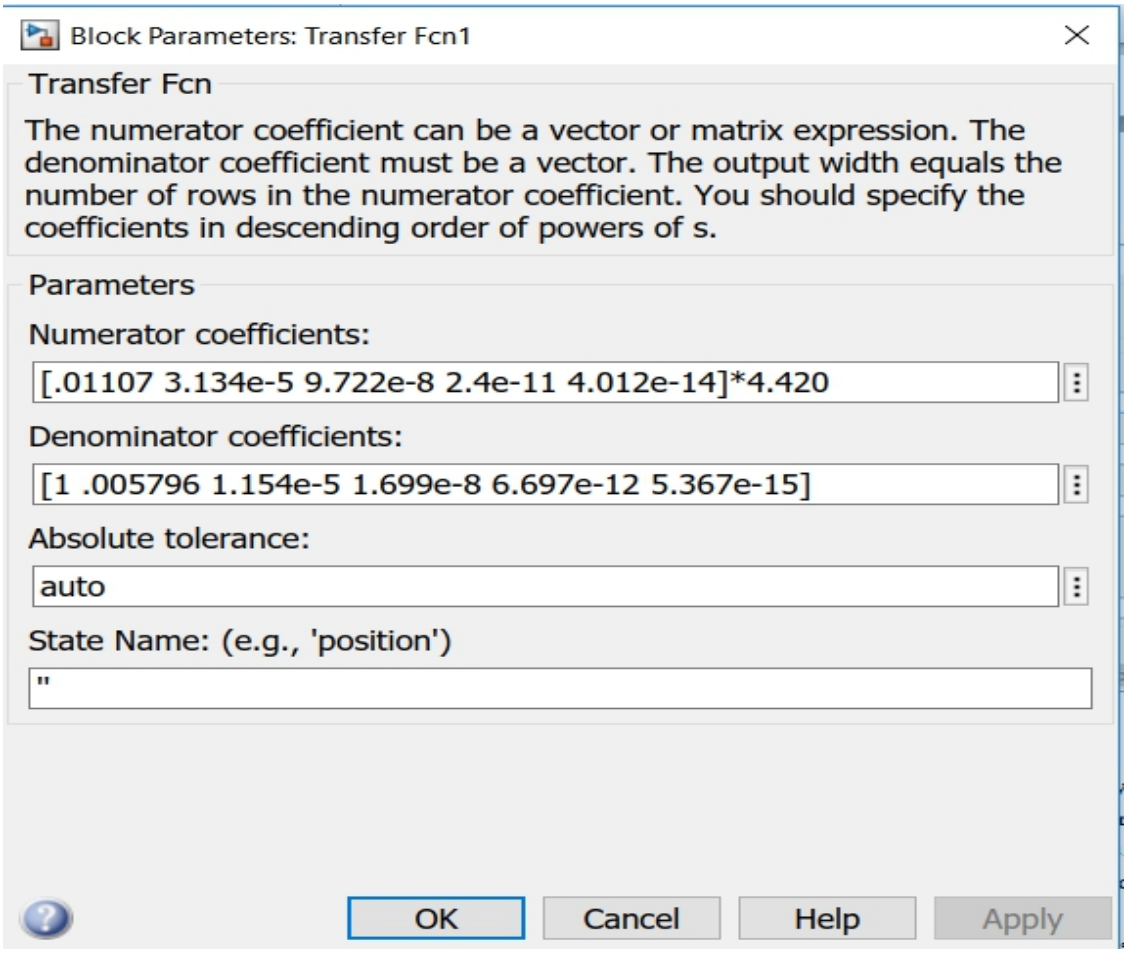


Figure 4.7: Block parameters of transfer function block

4.3 Time Series NARX Feedback Neural Network

All the specific dynamic networks discussed so far have either been focused networks, with the dynamics only at the input layer, or feedforward networks. The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network. The NARX model is based on the linear ARX model, which is commonly used in time-series modeling.

The defining equation for the NARX model is

$$y(t)=f(y(t-1),y(t-2),\ldots,y(t-n_y),u(t-1),u(t-2),\ldots,u(t-n_u)) \qquad (4.2)$$

where the next value of the dependent output signal y(t) is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal. You can implement the NARX model by using a feedforward neural network to approximate the function f. A diagram of the resulting network is shown below, where a two-layer feedforward network is used for the approximation. This implementation also allows for a vector ARX model, where the input and output can be multidimensional.

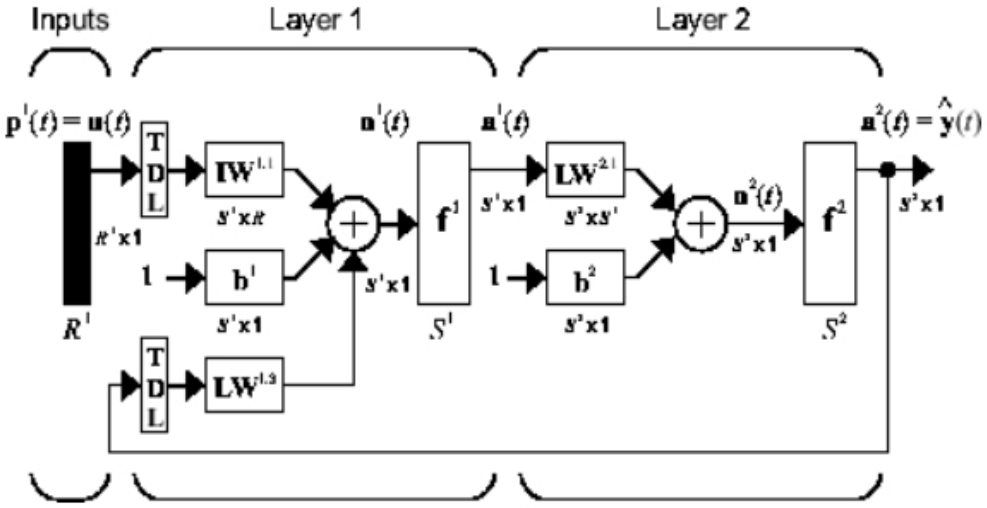


Figure 4.8: Block diagram of a NARX model

There are many applications for the NARX network. It can be used as a predictor, to predict the next value of the input signal. It can also be used for nonlinear filtering, in which the target output is a noise-free version of the input signal. The use of the NARX network is shown in another important application, the modeling of nonlinear dynamic systems [24].

4.3.1 Training a NARX network using the system dataset

NARX (Nonlinear autoregressive with external input) networks can learn to predict one time series given past values of the same time series, the feedback input, and another time series, called the external or exogenous time series.

narxnet(inputDelays,feedbackDelays,hiddenSizes,feedbackMode,trainFcn) takes these arguments,

inputDelays	Row vector of increasing 0 or positive delays (default = 1:2)
feedbackDelays	Row vector of increasing 0 or positive delays (default = 1:2)
hiddenSizes	Row vector of one or more hidden layer sizes (default = 10)
feedbackMode	One of 'open', 'closed', or 'none' (default is 'open')
trainFcn	Training function (default is 'trainlm')

and returns a NARX neural network [25].

We divided our dataset into 3 sections:

- 70% of the data is used for training the model(These are presented to the network during training, and the network is adjusted according to its error.)
- 15% is used for validation(These are used to measure network generalization, and to halt training when generalization stops improving.)
- 15% is used for testing(out of sample testing)

Here we have used nntraintool in matlab toolbox to train our dataset to match our target dataset. Here the number of hidden layers are 20 and Lavenberg Marquart is used again for iteration. The process interface is shown in figure 4.9

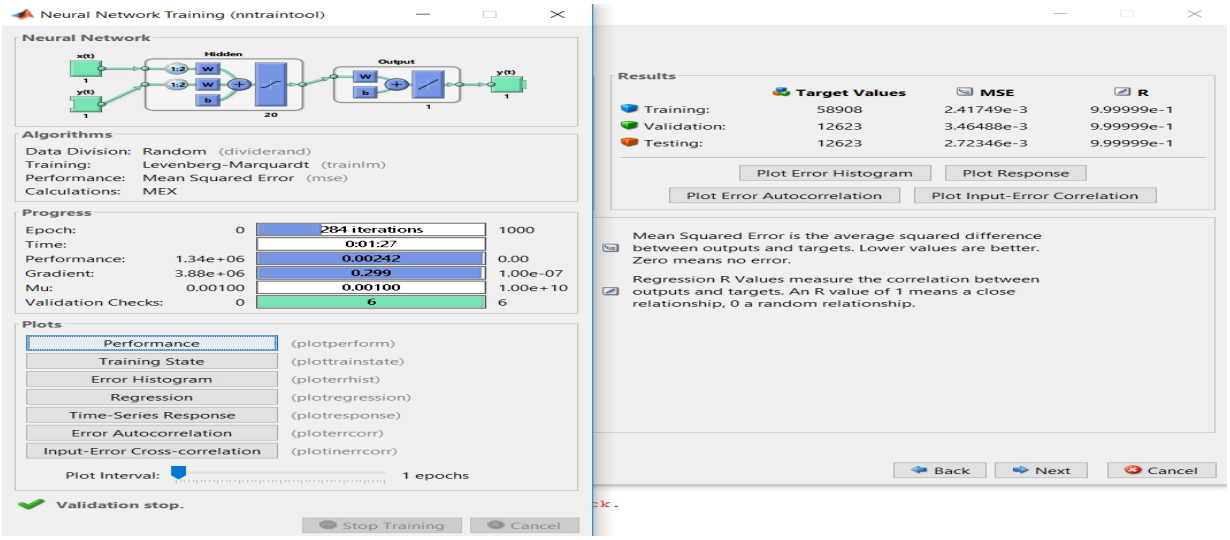


Figure 4.9: Interface of neural network time series for training using the system dataset

We can see the results in figure 4.9. The trained model gives negligible MSE(2.7e-3) after 284 iteration.

4.3.2 Implementing the estimated NARX model in simulink

Now we can export this network to Simulink and implement another equivalent of the plant model. This is shown in figure 4.10

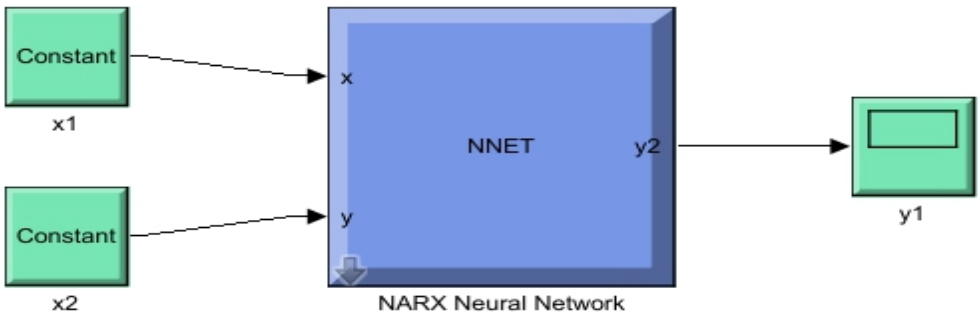


Figure 4. 10: Equivalent plant model using NARXnet

Here, the NNET block estimates our whole plant model. The internal block diagram of this NNET block is shown in figure 4.11

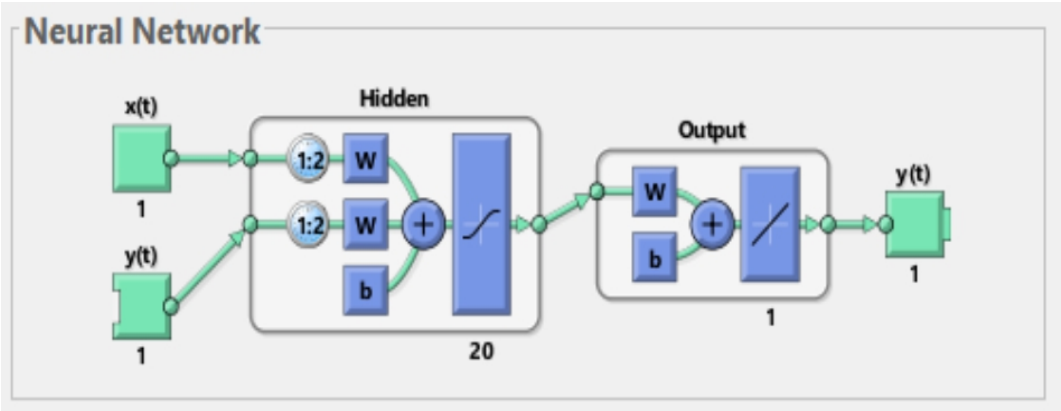


Figure 4. 11: Internal Diagram of NNET

Parameters of the neural network:

- Delay: 2
- Number of hidden layers: 20

Preset algorithms used for NNET:

- Data division: RAMDOM (dividerand)
- Training: Lavenberg-Marquardt
- Performance: Mean Squared Error
- Calculations: MEX

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

All of the training is done in open loop (also called series-parallel architecture), including the validation and testing steps. The typical workflow is to fully create the network in open loop, and only when it has been trained (which includes validation and testing steps) is it transformed to closed loop for multistep-ahead prediction. Likewise, the R values in the GUI are computed based on the open-loop training results.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Now, after obtaining two estimated models that represents our plant model, we can move on to incorporating control mechanisms to the models and find out the model which gives us the most accurate static and dynamic characteristics.

Chapter 5

Data-driven control system

5.1 Introduction

Data-driven control systems are a broad family of control systems, in which the identification of the process model and/or the design of the controller are based entirely on experimental data collected from the plant[1]

In many control applications, trying to write a mathematical model of the plant is considered a hard task, requiring efforts and time to the process and control engineers. This problem is overcome by data-driven methods, which allow to fit a system model to the experimental data collected, choosing it in a specific models class. The control engineer can then exploit this model to design a proper controller for the system. However, it is still difficult to find a simple yet reliable model for a physical system, that includes only those dynamics of the system that are of interest for the control specifications. The direct data-driven methods allow to tune a controller, belonging to a given class, without the need of an identified model of the system. In this way, one can also simply weight process dynamics of interest inside the control cost function, and exclude those dynamics that are out of interest.

In this chapter we are interested in exploring how both of our designed estimated models work with PID control and objectify which method is the better one.

5.2 Proportional–Integral–Derivative controller

A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value {\displaystyle e(t)}e(t) as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

In practical terms it automatically applies an accurate and responsive correction to a control function. An everyday example is the cruise control on a car, where ascending a hill would lower speed if only constant engine power were applied. The controller's PID algorithm restores the measured speed to the desired speed with minimal delay and overshoot by increasing the power output of the engine.

The first theoretical analysis and practical application was in the field of automatic steering systems for ships, developed from the early 1920s onwards. It was then used for automatic process control in the manufacturing industry, where it was widely implemented in pneumatic, and then electronic, controllers. Today the PID concept is used universally in applications requiring accurate and optimized automatic control. [26]

In 1939, the Taylor Instrument Companies introduced a completely redesigned version of its Fulscope pneumatic controller. In addition to proportional and reset control actions, this new instrument provided an action which the Taylor Instrument Companies called pre-act. In the same year the Foxboro Instrument Company added Hyper-reset to the proportional and reset control actions provided by their Stabilog pneumatic controller. The pre-act and Hyper-reset actions each provide a control action proportional to the derivative of the error signal. Reset provides a control action proportional to the integral of the error signal and hence both controllers offered PID control.

5.2.1 History of PID controller

The first evolution of the PID controller was developed in 1911 by Elmer Sperry. However, it wasn't until 1933 that the Taylor Instrumental Company (TIC) introduced the first pneumatic controller with a fully tunable proportional controller. A few years later, control engineers went eliminate the steady state error found in proportional controllers by resetting the point to some artificial value as long as the error wasn’t zero. This resetting “integrated” the error and became known as the proportional-Integral controller. Then, in 1940, TIC developed the first PID pneumatic controller with a derivative action, which reduced overshooting issues. However, it wasn’t until 1942, when Ziegler and Nichols tuning rules were introduced that engineers were able to find and set the appropriate parameters of PID controllers. By the mid-1950’s, automatic PID controllers were widely adopted for industrial use.

5.2.2 Design and Implementation of PID controller in MATLAB

PID control respectively stands for proportional, integral and derivative control, and is the most commonly used control technique in industry. The following video explains how PID control works and discusses the effect of the proportional, integral and derivative terms of the controller on the closed-loop system response.

PID control involves several tasks that include:

- Selecting an appropriate PID algorithm (P, PI, or PID)
- Tuning controller gains
- Simulating the controller against a plant model
- Implementing the controller on a target processor

While simple in theory, design and implementation of PID controllers can be difficult and time consuming in practice.

MATLAB and add-on products bring efficiency to these design tasks by enabls to:

- Configure your Simulink PID Controller block for PID algorithm (P,PI, or PID), controller form (parallel or standard), anti-windup protection (on or off), and controller output saturation (on or off)
- Automatically tune controller gains against a plant model and fine-tune your design interactively
- Autotune controller gains in real time against a physical plant
- Tune multiple controllers in batch mode
- Run closed-loop system simulation by connecting your PID Controller block to the plant model
- Automatically generate C code for targeting a microcontroller
- Automatically generate IEC 61131 structured text for targeting a PLC or PAC
- Automatically scale controller gains to implement your controller on a processor with fixed-point arithmetic [27]

5.2.3 PID controller tuning in Simulink

PID Tuner provides a fast and widely applicable single-loop PID tuning method for the Simulink® PID Controller blocks. With this method, you can tune PID controller parameters to achieve a robust design with the desired response time.

A typical design workflow with the PID Tuner involves the following tasks:

(1) Launch the PID Tuner. When launching, the software automatically computes a linear plant model from the Simulink model and designs an initial controller.

(2) Tune the controller in the PID Tuner by manually adjusting design criteria in two design modes. The tuner computes PID parameters that robustly stabilize the system.

(3) Export the parameters of the designed controller back to the PID Controller block and verify controller performance in Simulink.

When the PID Tuner launches, the software computes a linearized plant model seen by the controller. The software automatically identifies the plant input and output, and uses the current operating point for the linearization. The plant can have any order and can have time delays.

The PID Tuner computes an initial PI controller to achieve a reasonable tradeoff between performance and robustness. By default, step reference tracking performance displays in the plot.

The following figure shows the PID Tuner dialog with the initial design:

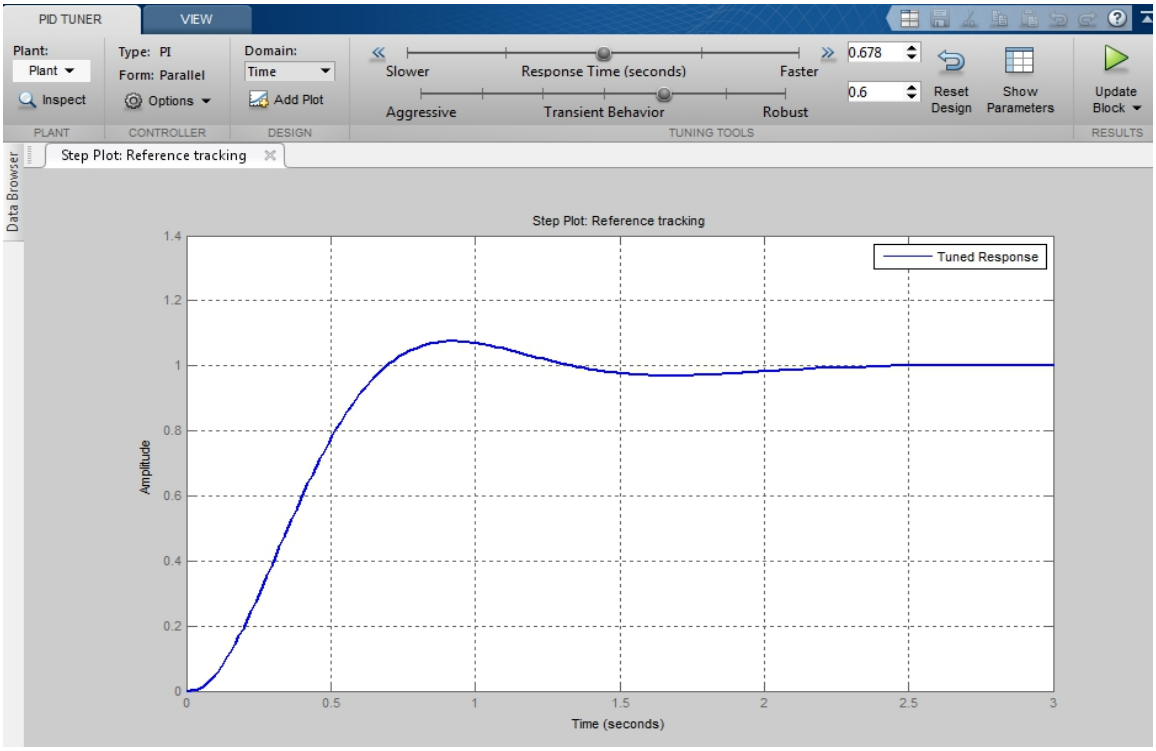


Figure 5.1: PID tuner dialogue

The overshoot of the reference tracking response is about 7.5 percent. Since we still have some room before reaching the settling time limit, you could reduce the overshoot by increasing the response time. Move the response time slider to the left to increase the closed loop response time. Notice that when you adjust response time, the response plot and the controller parameters and performance measurements update.

The following figure shows an adjusted PID design with an overshoot of zero and a settling time of 4 seconds. The designed controller effectively becomes an integral-only controller [2].

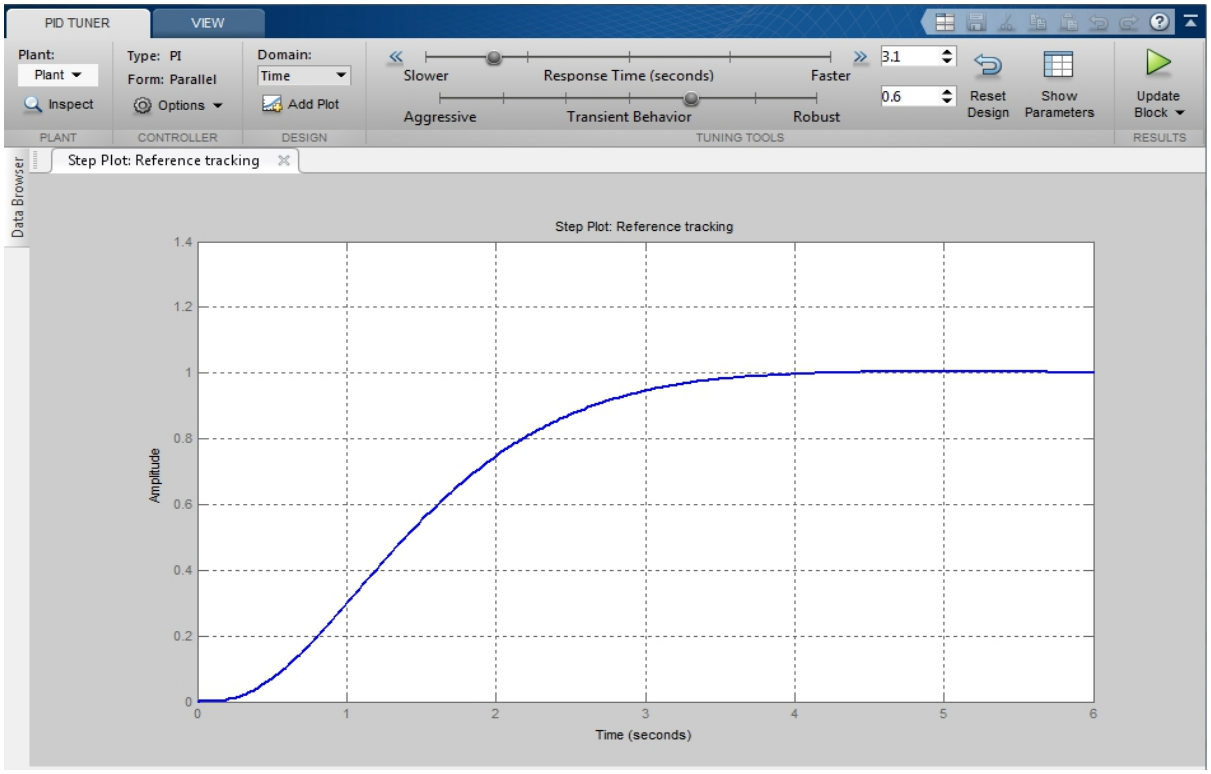
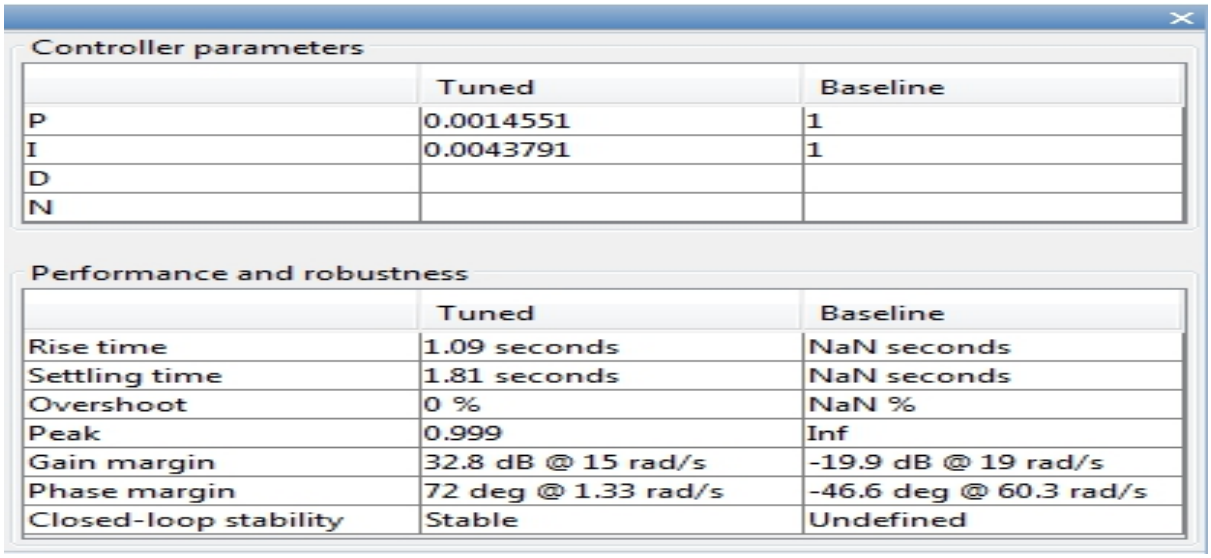


Figure 5.2: tuned PID response



Controller parameters

	Tuned	Baseline
P	0.0014551	1
I	0.0043791	1
D		
N		

Performance and robustness

	Tuned	Baseline
Rise time	1.09 seconds	NaN seconds
Settling time	1.81 seconds	NaN seconds
Overshoot	0 %	NaN %
Peak	0.999	Inf
Gain margin	32.8 dB @ 15 rad/s	-19.9 dB @ 19 rad/s
Phase margin	72 deg @ 1.33 rad/s	-46.6 deg @ 60.3 rad/s
Closed-loop stability	Stable	Undefined

Figure 5.3: PID tuner parameters window

5.3 Incorporating PID Control to SIP based estimated model

We already have the transfer function with 5 poles and 4 zeros, which most closely estimates our plant model. So, now we are in a position to incorporate control mechanism to our estimated plant model and design a stable controller for our physical plant. Because of using system identification process the circuit diagram gets very simplified. In figure 5.4 this is implemented.

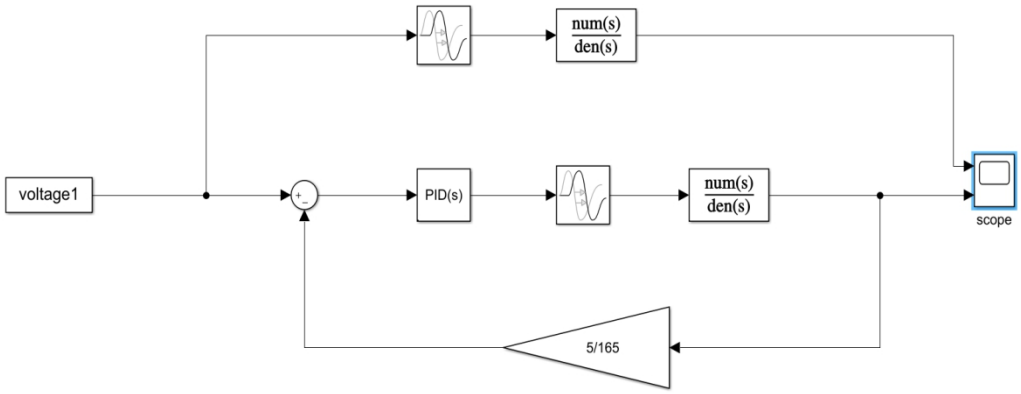


Figure 5.4: incorporating PID control to transfer function base estimated model and comparing both results

This PID tuner block implements continuous- and discrete-time PID control algorithms and includes advanced features such as anti-windup, external reset, and signal tracking. The interface of the tuner is shown in figure 5.5

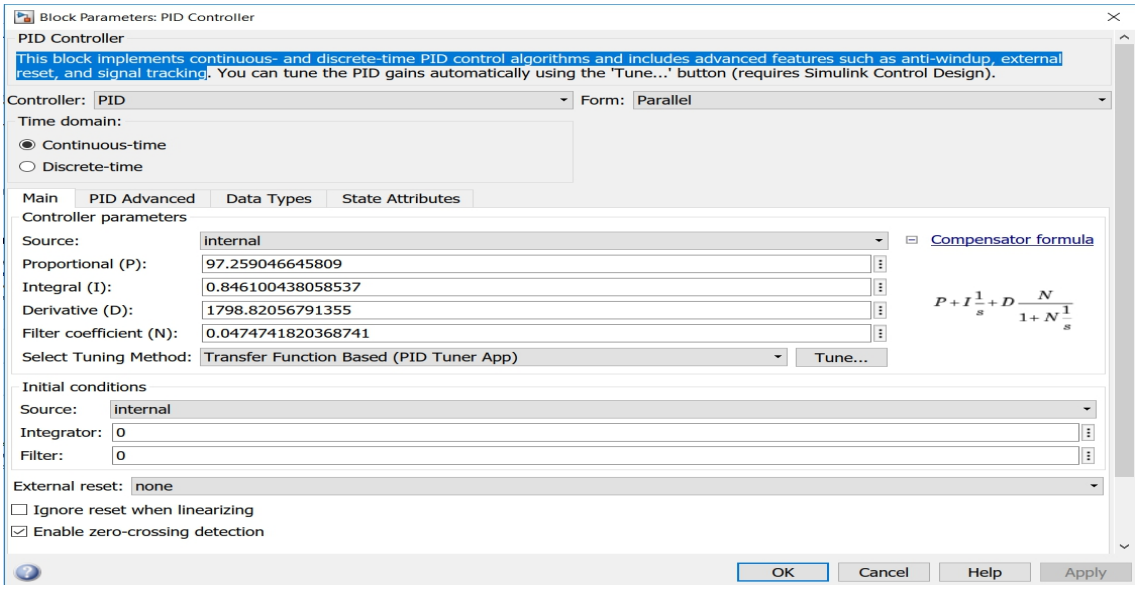


Figure 5.5: PID tuner block parameters

We can tune the PID gains automatically using the 'Tune...' button (requires Simulink Control Design). The tuner block parameters are shown in figure 5.6

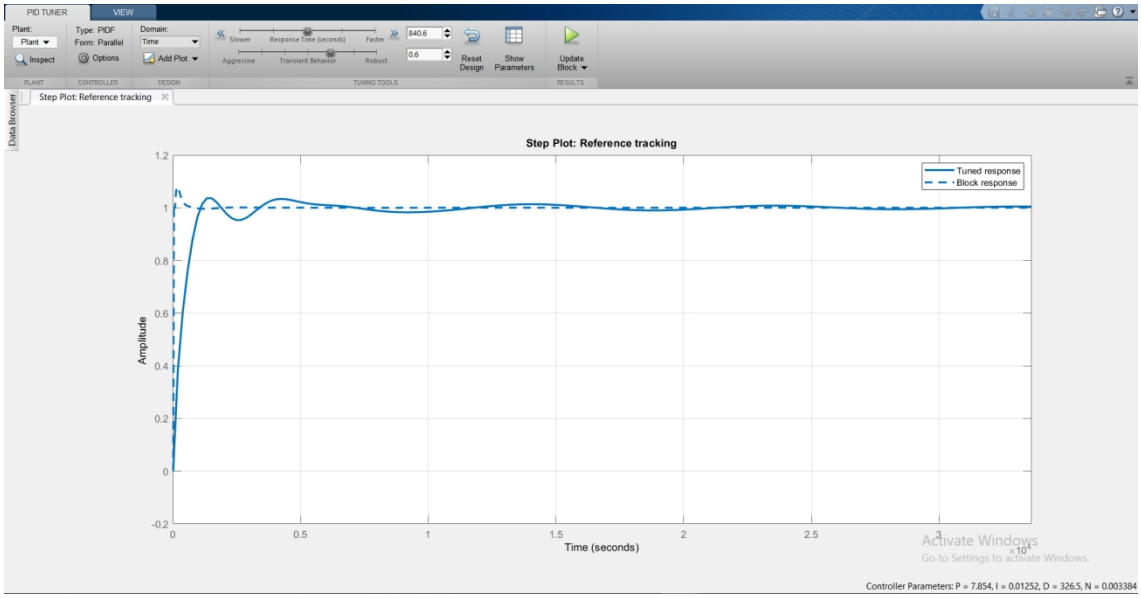


Figure 5.6: PID tuner step plot: reference tracking (incorporated with transfer function based estimated model)

From here we tune the PID block accordingly and update the block parameters.

5.4 Incorporating PID control to NARX equivalent model

Now, we are to do the same thing as the previous article, but with NNET model. This is shown in figure 5.7

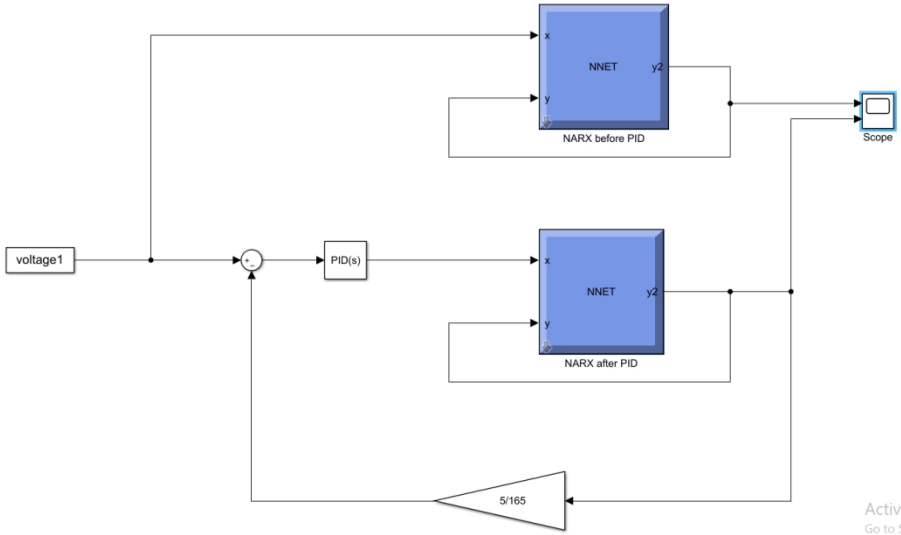


Figure 5.7: Incorporating PID control to NNET model

The PID tuner block parameters are shown in figure 5.8

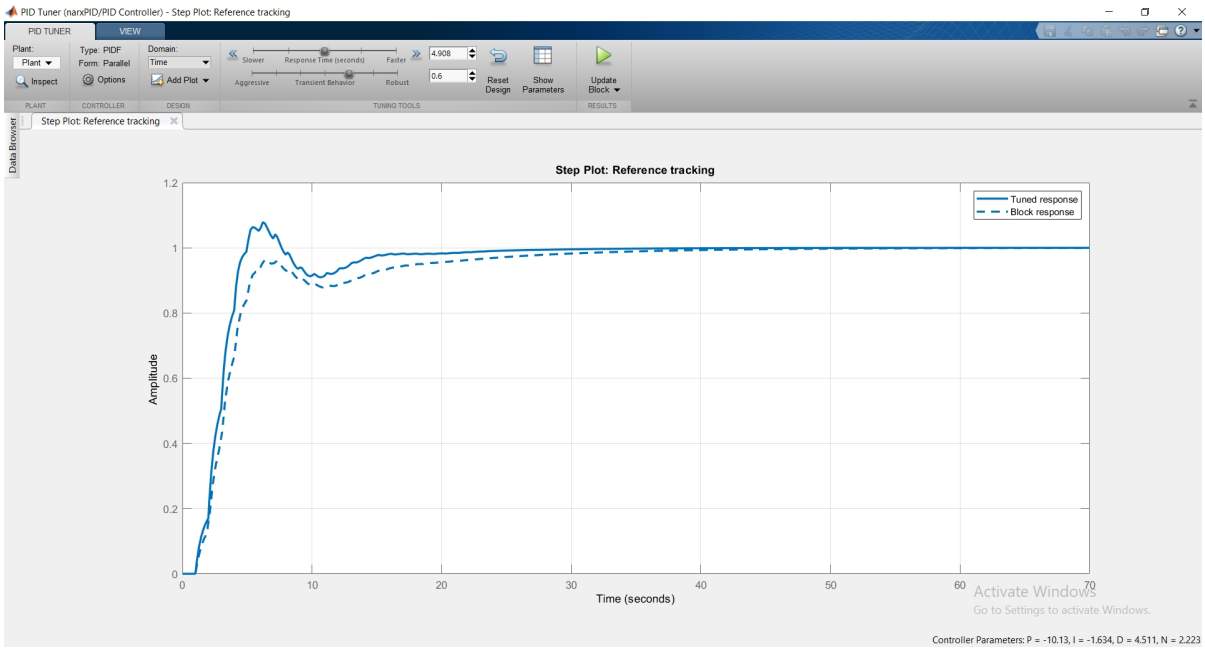


Figure 5.8: PID tuner step plot: reference tracking (incorporated with NNET)

Here the continuous line is the tuned response and the dashed line is the block response without tuning.

Chapter 6

Simulation Result

6.1 Introduction

Now, having everything at hand, we are now in a position to compare both system identification process based model and the NNET estimated model. In this chapter, we dive into this comparison of these two data driven control system models.

6.2 Response of SIP Based Estimated Model

After finding out the estimated equivalent transfer function of the plant model, we implemented it in Simulink using transfer function block and a transport delay block. After that we incorporated PID control to this model. This is shown in figure 5.4. Now, we are interested in looking into the output of this model. The output of the Simulink scope is shown in figure 6.1

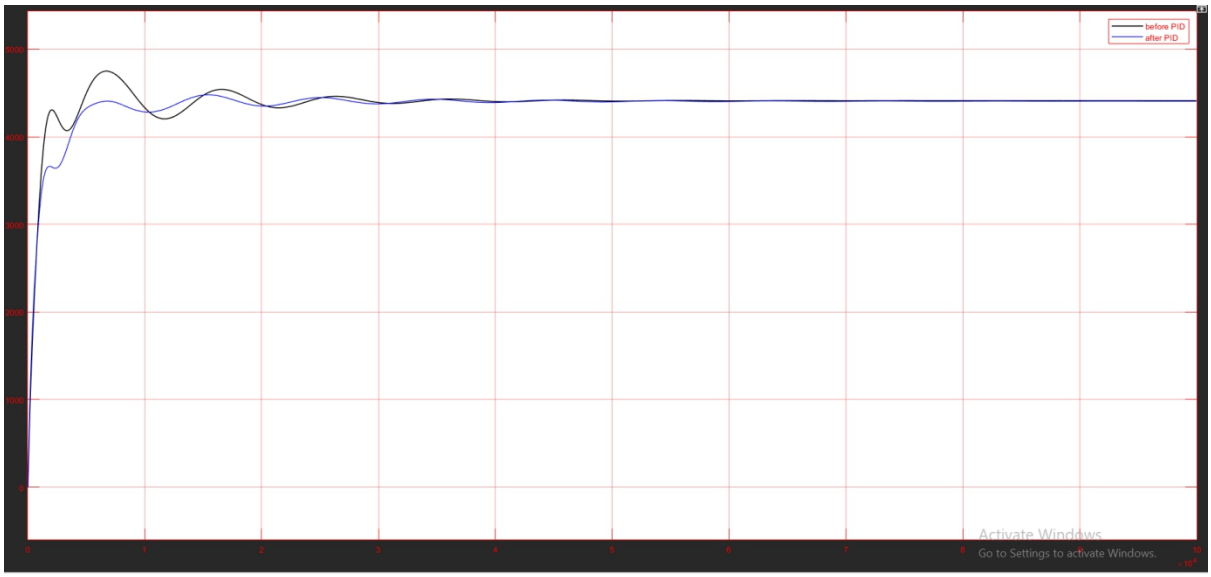


Figure 6.1: befor and after incorporating PID control to transfer function based model

Here, the black line is before implementing PID and the blue line is after implementing PID control. Clearly, the system dynamics improve after incorporating PID which is expected. Also, the output matches with that of the original physical plant.

6.3 Response of NARX equivalent model

We have already implemented NARX in Simulink, (figure 5.7) and also we have incorporated PID control with it. The output of the scope is presented in figure 6.2

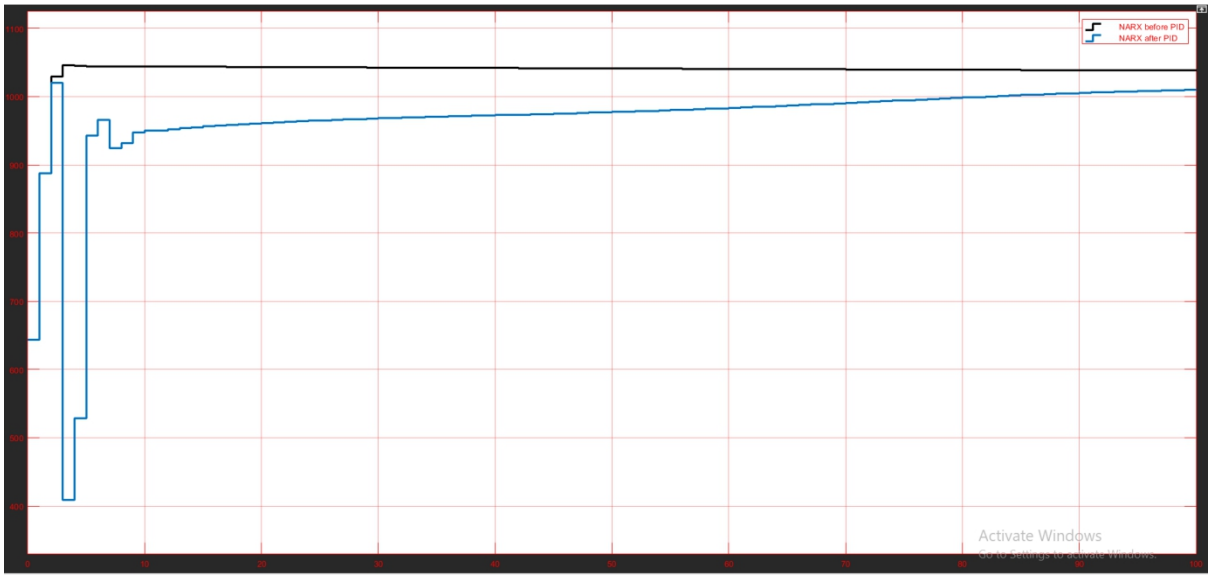


Figure 6.2: befor and after incorporating PID control to NARX based model

Here, the blue line is response after incorporating PID and the black line is the response before incorporating PID control. Here, we find some interesting results. The response is showing some unwanted spikes in the transient response which is not desirable and not in line with the transient response of a DC shunt motor. Although, the response time faster than SIP based model.

As NARX model failed to simulate the transient behavior of a DC shunt motor, we can say the system identification process based model works better. Although, SIP is a more rigorous and time consuming process, for small plant models, it can estimate the plant very accurately.

Chapter 7

Conclusion

The speed management mechanism of the DC Drive system using closed-loop system control with PID controller was used. The speed control can be additional tag by dynamic the pulse dimension period being fed to the Gate of the MOSFET that successively would modification the duty cycle and hence would modification the coil voltage. data driven control system of DC motor is implemented by using system identification process. The system identification based on system identification toolbox and (NARX) neural network for DC motor model estimation is produced. Nowadays, neural network system identification is becoming more popular and it's useful for more complex dynamics models and multiple input/output systems. The produced model can be realistic dynamic model of DC motor, that able adopt the DC motor to dynamics completely. Moreover this model can be used to analysis controller design and the control system of DC motor for students, researchers and control engineers.

Using the methods discussed in this work, in the future we can calculate any parameters of a physical plant without having the plant model! Various researches can be made having these calculated data available. Also, any complex integrated system can be modeled using these methods and we can do stability analysis of the plant in contingency situations as well.

References

[1] https://en.wikipedia.org/wiki/Data_science

[2] Santana, J., Naredo, J. ., Sandoval, F., Grout, I., & Argueta, O. . (2002). Simulation and construction of a speed control for a DC series motor. Mechatronics, 12(9-10), 1145–1156. doi:10.1016/s0957-4158(02)00019-3

[3] https://www.engineeringenotes.com/electrical-engineering/electric-motors/chopper-control-of-dc-motors-operation-and-set-up-electrical-engineering/36996

[4] https://www.mathworks.com/products/sysid.html

[5] https://www.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html

[6] Malviya, P. (2015) "Speed control of DC Motor a Review". International Journal of Engineering sciences & Research Technology

[7] Ye Naung1, Anatolii Schagin, Htin Lin Oo, Kyaw Zaw Ye, Zaw Min Khaing ©2018 IEEE Implementation of Data Driven Control System of DC Motor by Using System Identification Process

[8] https://www.mathworks.com/help/nnet/ug/design-time-series-narx_x0002_feedback-neural-networks.html

[9] Hussein, A., Hirasawa, K., Hu, J., Wada, K., 2003. Application of

[10] universal learning networks to PV-supplied DC motor drives. Research Reports on Information Science and Electrical Engineering of Kyushu University8(2),129–134.)

[11] Moleykutty George, "Speed Control of Separately Excited DC Motor" in *American Journal of Applied Sciences 5 (3): 227-233, 2008*,

[12] https://www.electrical4u.com/differential-relay/

[13] https://en.wikipedia.org/wiki/PID_controller

[14] S. Bennett, "Development of the PID controller" in IEEE Control Systems Magazine (Volume: 13, Issue: 6, Dec. 1993), DOI: 10.1109/37.248006

[15] https://www.mawdsleysber.co.uk/types-dc-motors/

[16] Rinita Rudra1*, Rumrum Banerjee2 (2017), "Modeling and Simulation of DC Motor Speed Regulation by Field Current Control Using MATLAB" in International Journal of Computer Electrical Engineering, Volume 9, Number 2, December 2017

[17] https://en.wikipedia.org/wiki/DC_motor

[18] https://www.mathworks.com/help/simulink/slref/step.html

[19] https://www.electricaleasy.com/2014/07/characteristics-of-dc-motors.html

[20] Anurag Dwivedi, (2013), “Speed Control of DC Shunt Motor with Field and Armature Rheostat Control Simultaneously” in Advance in Electronic and Electric Engineering ISSN 2231-1297, Volume 3, Number 1 (2013), pp. 77-80

[21] https://www.mathworks.com/products/sysid.html

[22] https://www.mathworks.com/help/simulink/slref/transportdelay.html

[23] https://www.mathworks.com/help/simulink/slref/transferfcn.html

[24] https://www.mathworks.com/help/deeplearning/ug/design-time-series-narx-feedback-neural-networks.html

[25] https://www.mathworks.com/help/deeplearning/modeling-and-prediction-with-narx-and-time-delay-networks.html

[26] Daniel E. Rivera, Manfred Morari, and Sigurd Skogestad, “Internal model control: PID controller design” in nd. Eng. Chem. Process Des. Dev. 1986, 25, 1, 252–265 Publication Date:January 1, 1986 https://doi.org/10.1021/i200032a041

[27] https://www.mathworks.com/help/slcontrol/ug/designing-controllers-with-the-pid-tuner.html