Artificial Intelligence In Electric Machines

Report submitted in fulfillment of the requirements
for the UG Project of

Third Year.

by

Aman Gope, Abhinav Barve and Vasu Bansal

Under the guidance of

Dr. Chinmaya K A

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2022

Dedicated to

My parents, teachers,.....

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date:

Aman Gope, Abhinav Barve and Vasu Bansal
B.Tech and IDD
Department of Electrical Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Artificial Intelligence In Electric Machines**" being submitted by **Aman Gope, Abhinav Barve and Vasu Bansal** (**Roll No. 19084005 , 19085003, 19085095**), carried out in the Department of Electrical Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date:

Dr. Chinmaya K A
Department of Electrical Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

We would like to express our sincere gratitude to Dr. Chinmaya K A and Dhawal Dwivedi sir for guiding us througout this project.

Place: IIT (BHU) Varanasi

Date: 10.05.2022 **Aman Gope, Abhinav Barve and Vasu Bansal**

Abstract

As we all can witness the use of artificial intelligence can be seen almost everywhere in the present world. In this paper we wanted to explore the use of Artificial Intelligence in Electrical machines and in the most part we have explored various motion control methods for machines and extended them by using ANNs (Artificial Neural Networks) in their place.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

The past decade has marked an incredibly fast-paced and innovative period in the history of AI, driven by the start of the deep learning revolution. Spurred by the development of ever-more powerful computing platforms and the increased availability of big data, deep Learning has successfully tackled many previously intractable problems, especially in computer vision and natural language processing. Deep Learning has also been applied and is in the process of transforming many real-world applications, including entertainment, healthcare, fraud detection, virtual assistants, and autonomous vehicles. [1] This is an attempt for using Artificial Inteligence (Artificial Neural Networks) for motor motion control.

1.2 Motivation of the Research Work

The conventional proportional–integral (PI) or proportional–integral–derivative (PID)-type controllers are widely used in the industry due to their simple control structure, ease of design, and inexpensive cost, but the most notable disadvantages of such controllers is the difficulty in finding the best values of their parameters using classical methods, such as trial and error and other advanced methods. [2] Therefore, various optimization algorithms can be applied in tuning these controller parameters to ensure optimal control performance at desired operating conditions. While the PI/PID controller has many advantages , it often cannot provide perfect control performance if the controlled plant is highly nonlinear and uncertain. [3] so we are attempting to use ANNs for replacing conventional PI/PID-type controllers.

Chapter 2

Artificial Neural Networks

2.1 Artificial Neurons

Among all the AI techniques, NNWs are most important, and in fact, modern AI technology is synonymous with NNW techniques and their applications. It is a generic form of AI, and therefore more powerful. The invention of NNW is often considered as significant as the invention of transistor.

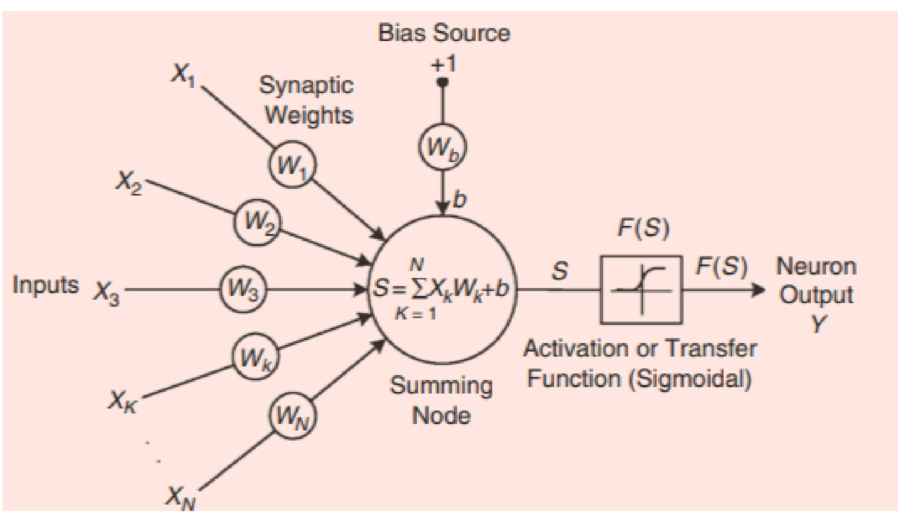


Figure 2.1 Artificial Neuron

A NNW is basically interconnection of artificial neurons. [4] it emulates the characteristics of biological neuron in our brain nervous system .

2.2 Neural Network

The input signals X1, X2, etc. which may be continuous variables or discrete pulses, flow through a gain or weight (called synaptic weight or connection strength) that can be positive or negative, integer or noninteger.

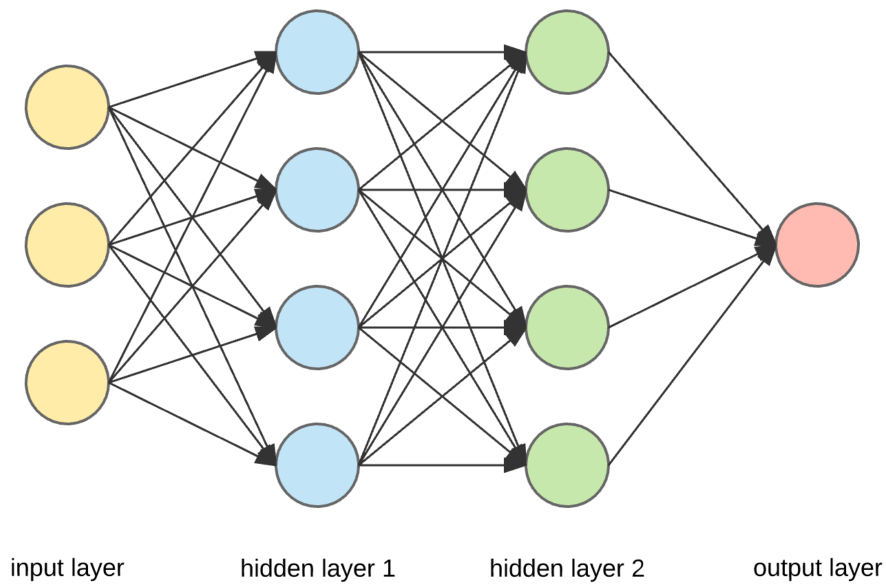


Figure 2.2 Artificial Neural Network

The summing node accumulates all the input-weighted signals, adds to the weighted bias signal b and passes to the output through the nonlinear (or linear) activation or transfer function (TF), as shown in the figure. The activation function may be linear bipolar, threshold, signum, Gaussian, sigmoidal (or log-sigmoid), or hyperbolic-tan (or tan-sigmoid). The magnitude of these functions varies between 0 and 1, or -1 to +1. The nonlinearity of TF gives nonlinear input-output mapping property of NNW. The NNW can have many feedforward and feedback (called recurrent) topologies, but the most commonly used feedforward topologies are depicted in the diagrams.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^{n} x_i w_i\right)$$

$$x \in d_{1\times n},\ w \in d_{n\times 1},\ b \in d_{1\times 1},\ z \in d_{1\times 1}$$

Figure 2.3 Forward propagation

So far we have described the forward pass, meaning given an input and weights how the output is computed. After the training is complete, we only run the forward pass to make the predictions. But we first need to train our model to actually learn the weights, and the training procedure works as follows:

- Randomly initialize the weights for all the nodes.
- For every training example, perform a forward pass using the current weights, and calculate the output of each node going from left to right. The final output is the value of the last node.
- Compare the final output with the actual target in the training data, and measure the error using a loss function.
- Perform a backwards pass from right to left and propagate the error to every individual node using backpropagation. Calculate each weight's contribution to the error, and adjust the weights accordingly using gradient descent. Propagate the error gradients back starting from the last layer

2.3 Implementation in Matlab/Simulink

Here we are implementing a basic example for predicting a Polynomial function in neural network $2A + 3B + 5C$.

Generating Data Set

We are generating 1000 random integers for A,B and C using matlab rand function.

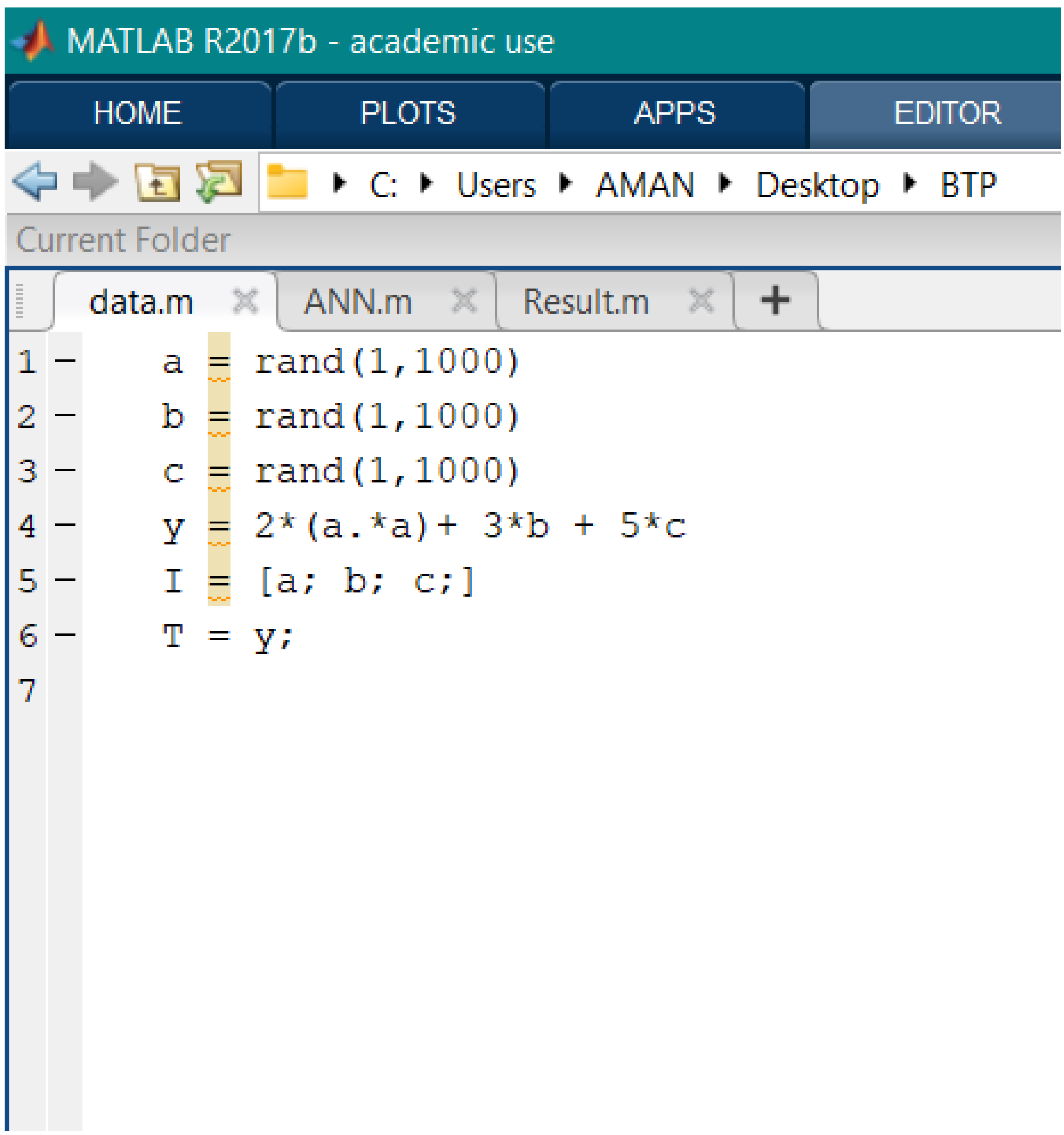


Figure 2.4 Data Generation

Training Data Set

Now we will be creating our model and train our model using the data set we already created. using the matlab command **newff()** , which creates a new network with a dialog box. and returns an N layer feed-forward backprop network.Then we first initialize and finally train our network.

Here we are using a network of initailly 3 inputs which are for A B and C values and then we have our hidden layer as 3 5 and 1 and finally we have our output from the trained network.

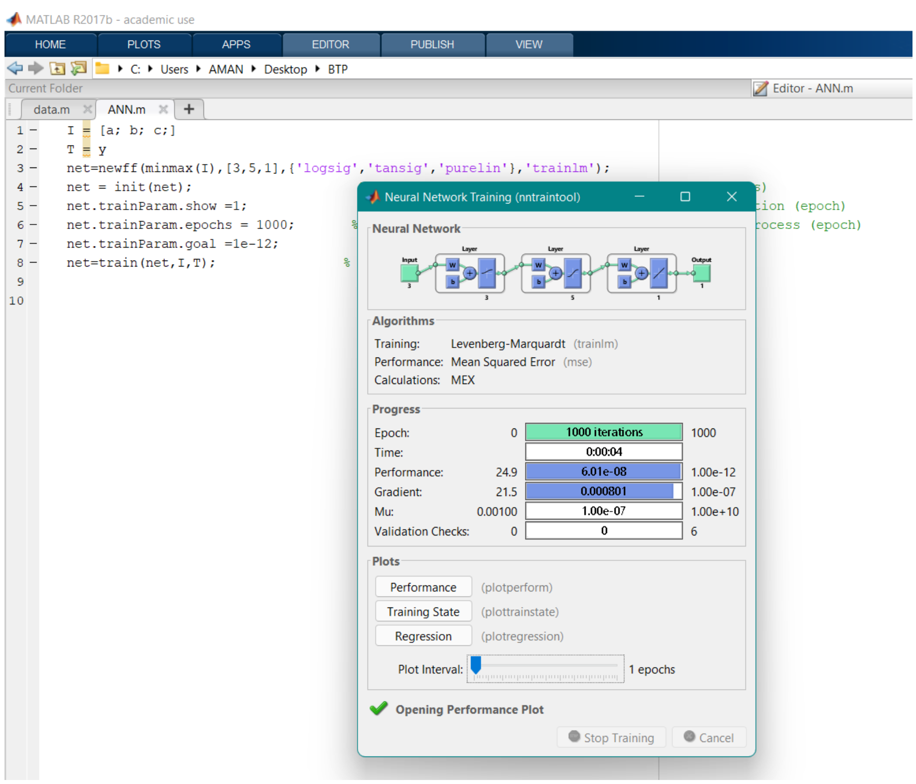


Figure 2.5 training The Dataset

Results

```
MATLAB Command Window                                        Page 1

>> Accuracy

expected =

    3.5381


ANNOutput =

    3.5379


z =

  -4.6245e-05


expected =

    5.3909


ANNOutput =

    5.3908


z =

  -4.6245e-05


expected =

    4.4421


ANNOutput =

    4.4421


z =

  -4.6245e-05


expected =
```

Figure 2.6 Results

```
MATLAB Command Window                                              Page 2

    7.8910

ANNOutput =

    7.8909

z =

  -4.6245e-05

expected =

    3.6267

ANNOutput =

    3.6266

z =

  -4.6245e-05

expected =

    2.9109

ANNOutput =

    2.9109

z =

  -4.6245e-05

expected =

    5.2071

ANNOutput =
```

Figure 2.7 Results

```
MATLAB Command Window                                    Page 3

    5.2071


z =

  -4.6245e-05


expected =

    5.4983


ANNOutput =

    5.4984


z =

  -4.6245e-05


expected =

    6.6691


ANNOutput =

    6.6690


z =

  -4.6245e-05


expected =

    2.7352


ANNOutput =

    2.7351
```

Figure 2.8 Results

```
MATLAB Command Window                                    Page 4

z =

  -4.6245e-05


Mean_Squared_Error =

   2.1386e-09

>>
```

Figure 2.9 Results

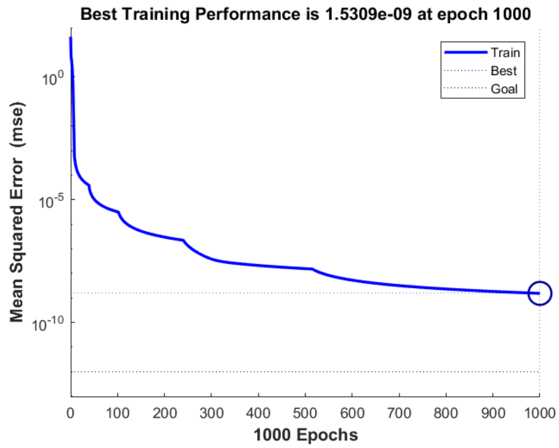


Figure 2.10 Performance

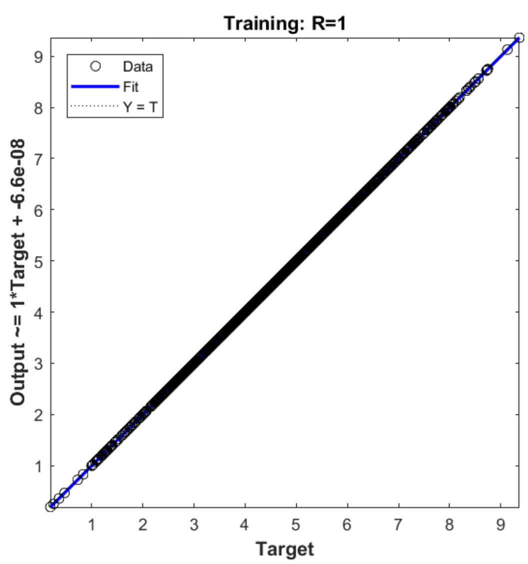


Figure 2.11 Regression

Chapter 3

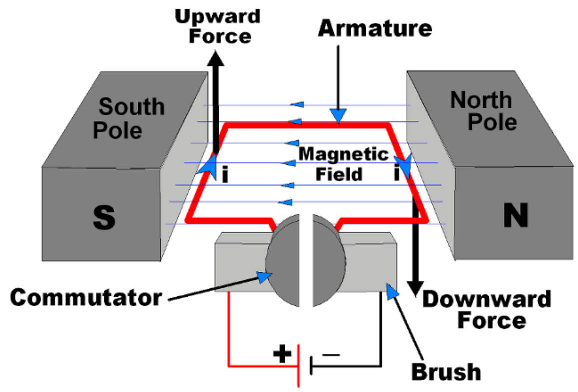
DC Motor

3.1 What is DC Series Motor?

The DC Series Motor is similar to any other motor because the main function of this motor is to convert electrical energy to mechanical energy. The operation of this motor mainly depends on the electromagnetic principle. Whenever the magnetic field is formed approximately, a current carrying conductor cooperates with an exterior magnetic field, and then a rotating motion can be generated.

3.2 Components used in DC Series Motor

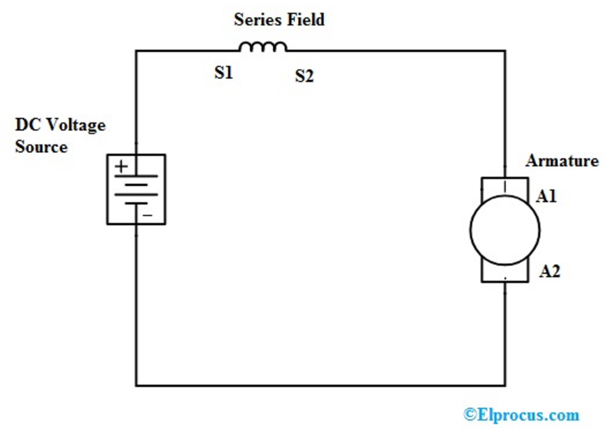
The components of this motor mainly include the rotor (the armature), commutator, stator, axle, field windings, and brushes. The fixed component of the motor is the stator, and it is built with two otherwise more electromagnet pole parts. The rotor includes the armature and the windings on the core allied to the commutator. The power source can be connected toward the armature windings throughout a brush array allied to the commutator.



DC Motor Conceptual Diagram

3.3 DC Series Motor Circuit Diagram

In this motor, field, as well as stator windings, are coupled in series by each other. Accordingly the armature and field current are equivalent. Huge current supply straightly from the supply toward the field windings. The huge current can be carried by field windings because these windings have few turns as well as very thick. Generally, copper bars form stator windings. These thick copper bars dissipate heat generated by the heavy flow of current very effectively. Note that the stator field windings S1-S2 are in series with the rotating armature A1-A2.



3.4 Speed Torque Characteristics of DC Series Motor

In general, for this motor, there are 3-characteristic curves are considered significant like Torque Vs. armature current, Speed Vs. armature current, Speed Vs. torque. These three characteristics are determined by using the following two relations. The above two equations can be calculated at the equations of emf as well as torque. For this motor, the back emf's magnitude can be given with the similar DC generator e.m.f equation like Eb= P NZ / 60A. For a mechanism, A, P, and Z are stable, thus, N Eb/. The DC series motor torque equation is,

Torque= Flux* Armature current

T = If * Ia

Here If= Ia, then the equation will become

T= Ia^2

Chapter 4

Motion Control in DC Motor

4.1 How is Motion Control done?

One of the conventional methods is by using a constant gain feedback controller, which uses some kind of sensor to take the feedback. But the conventional constant gain feedback controller fails to maintain the performance of the system at acceptable levels.

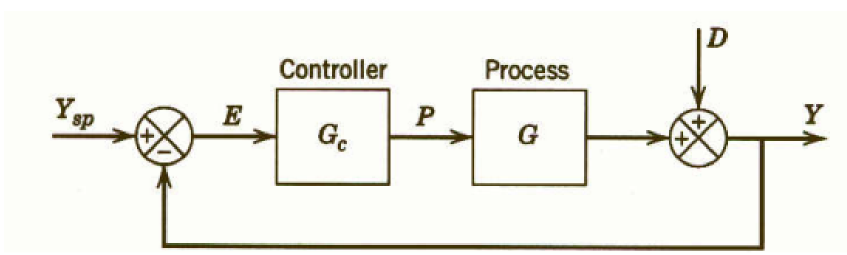


Figure 4.1 Conventional Feedback Controller

The proportional Integral (PI) controller is one of the conventional controllers and it has been widely used for speed control of dc motor drives .The major features of the PI controller are its ability to maintain a zero steady-state error to a step change in reference.

The last decade has seen an increasing interest in computational intelligence (CI) applications in control of various dynamic systems, including electric motor drives.

Most frequently used CI methods, Artificial Neural Networks (ANN) and Fuzzy logic (FL), are widely utilized in area of modeling, identification, diagnostics and control. The ANN based technique is advantageous over then conventional ones because it has a non-algorithmic parallel-distributed architecture as shown in Fig(2.2). This allows it to learn any complex input-output mapping. So, ANN are rapidly gaining popularity among power system researches. ANN are extremely useful in the area of learning control.

Advantages of using ANNs:

- Learning ability
- Massive parallelism
- Fast adaption
- Inherent approximation capability
- High degree of tolerance

4.2 Basic Derivations

Instantaneous field current:

vf= Rf * If + Lf * d(If/dt)

Where Rf and Lf are the field resistance and inductor, respectively.

And we can calculate Instantaneous armature current as:

Va= Ra * Ia+La * d(Ia/dt)+Eg

Where Ra and La are the armature resistance and inductor, respectively.

While the motor back emf, which is also known as speed voltage, is expressed as:

eb= Kv * w * If

Where Kv is the motor voltage constant (in V/A-rad/s) and is the motor speed (in

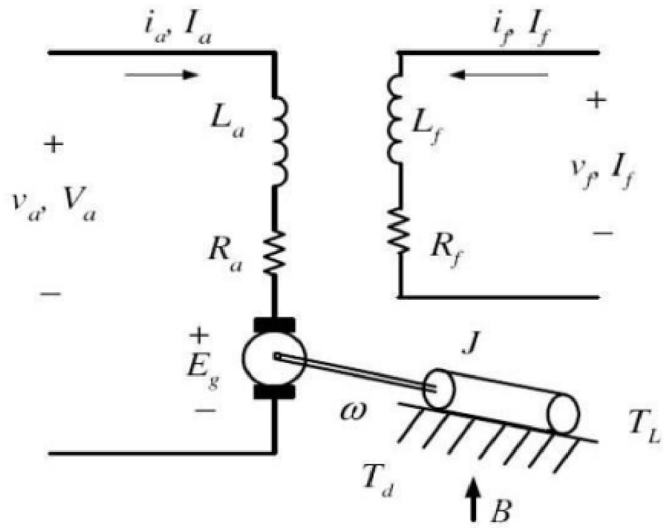


Figure 4.2 Seperately Excited DC Motor

rad/sec)

4.2.1 Basic Torque Equation

Td = J *(d/dt) + B + TL

The torque developed by the motor is:

Td=Kt * If * ia

Where (Kt=Kv) is torque constant in V/A-rad/sec.

Sometimes it is written as:

Td=Kt * flux(phi) * Ia

For normal operation, the developed torque must be equal to the load torque plus the friction and inertia, i.e.: where

B : viscous friction constant,(N.m/rad/s)

TL :load torque (N.m)

J : inertia of the motor (Kg.m2

4.2.2 Steady State Operations

Under study state operation, a time derivative is zero.

Assuming the motor is not saturated.

For field circuit,

Vf=If * Rf

The back emf is given by:

Eg=Kv * w * If

The armature circuit,

Va = Ia * Ra + Eg = Ia * Ra + Kv * If

Now the developed torque can be easily derived.

The developed torque is:

Td = Kt * If * Ia = B + TL

The required power is:

Pd = T * d

4.2.3 Torque and Speed Control

From the derivation, several important facts can be deduced for steady-state operation of DC motor.

a)For a fixed field current, or flux (If), the torque demand can be satisfied by varying the armature current (Ia).

b)The motor speed can be varied by: Controlling Va (voltage control)

Controlling Vf (field control)

c)These observations lead to the application of variable DC voltage for controlling the speed and torque of DC motor.

4.3 Seperately Excited DC Motor Model

The dynamics of the SEDM. As shown in fig. (1) are described by the following electrical and mechanical differential equations:-

$$L_a \frac{di_a}{dt} = -i_a R_a - kw + v_a \ldots\ldots\ldots\ldots (1)$$

$$J \frac{dw}{dt} = k i_a R_a - B w T_L \ldots\ldots\ldots\ldots\ldots (2)$$

Figure 4.3 Seperately Excited DC Motor

Where va is the motor input voltage; ia is the armature current; w is the rotor speed; TL is the load torque; Ra is the armature resistance; La is the armature inductance; J is the motor rotation inertia; B is the damping constant and K is the torque or EMF constant

fig.(4.3) illustrate Basic mathematical model of separately excited dc motor, where: Ta -Time constant of motor armature circuit and Ta=La/Ra (s) Tm – Mechanical time constant of the motor Tm=J/B (s)

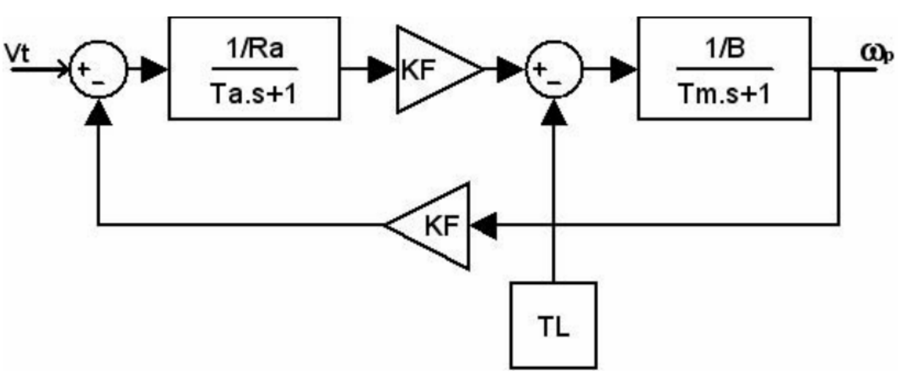


Figure 4.4 The Basic mathematical model of separately excited dc motor.

4.4 Motion Control using PI controller

Here We have developed the Model of separately excited DC Motor using the transfer function block diagram of Separately excited DC motor.

Motor Specifications

- Ta -Time constant of motor armature circuit and Ta=La/Ra (s)
- Tm – Mechanical time constant of the motor Tm=J/B (s)
- The parameters of the SEDM are : 1800 rpm, 220 volts, La= 0.0025H, Ra=0.5,TL=21.4N.m, J=0.0013kg/m2, B=0.001 N.m, the speed at full-load=1500rpm

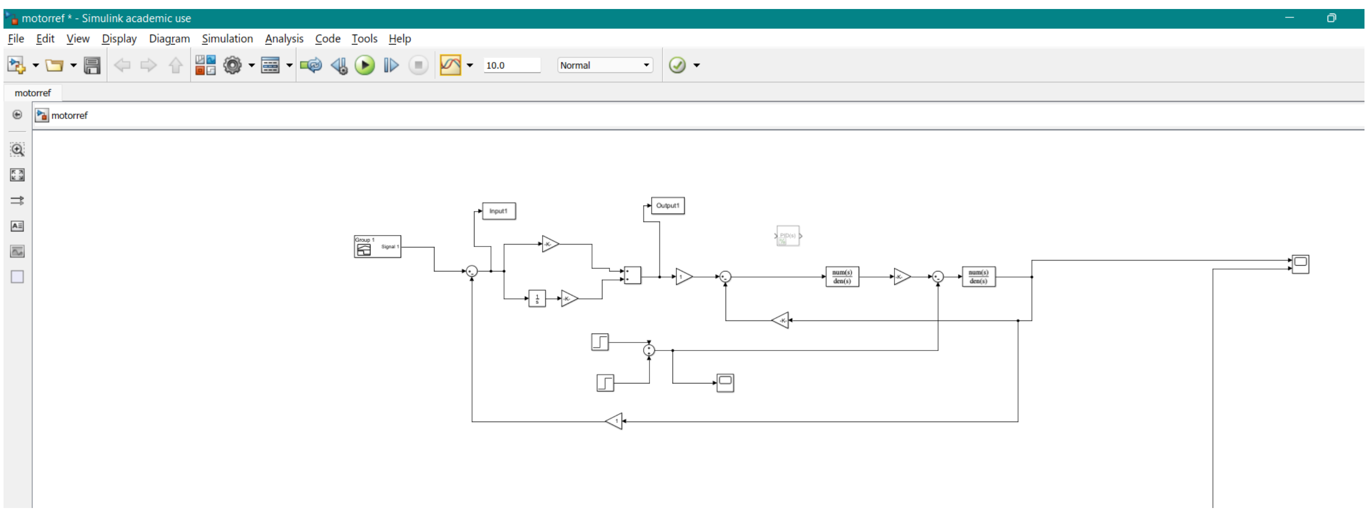


Figure 4.5 Simulink Block Diagram for PI based controller

4.5 Motion Control using NARMA-L2 Controller

NARMA-L2 stands for **Nonlinear Autoregressive Moving Average**. Basically This is a block in simulink NN Toolbox which makes it very easy to design ANN based neural network.

It's simply a rearrangement of the neural network plant model, which is trained off-line, in batch form. The only online computation is a forward pass through the neural

Figure 4.6 Comparison of reference signal v/s PI controlled output

network controller. NARMA-L2 controller, a multilayer neural network has been successfully applied in the identification and control of dynamic systems.
For this we have to build a separate reference model via which it can identify and train itself. then this can be further used as controller.

4.6 Building Custom Neurals Network as controller

For Building custom Neural Networks we should only have input and output data for any system. which could be used for ANN training. so what we did was , we used the PI controlled plant and generated Input and output data by exporting input and output data points to workspace.

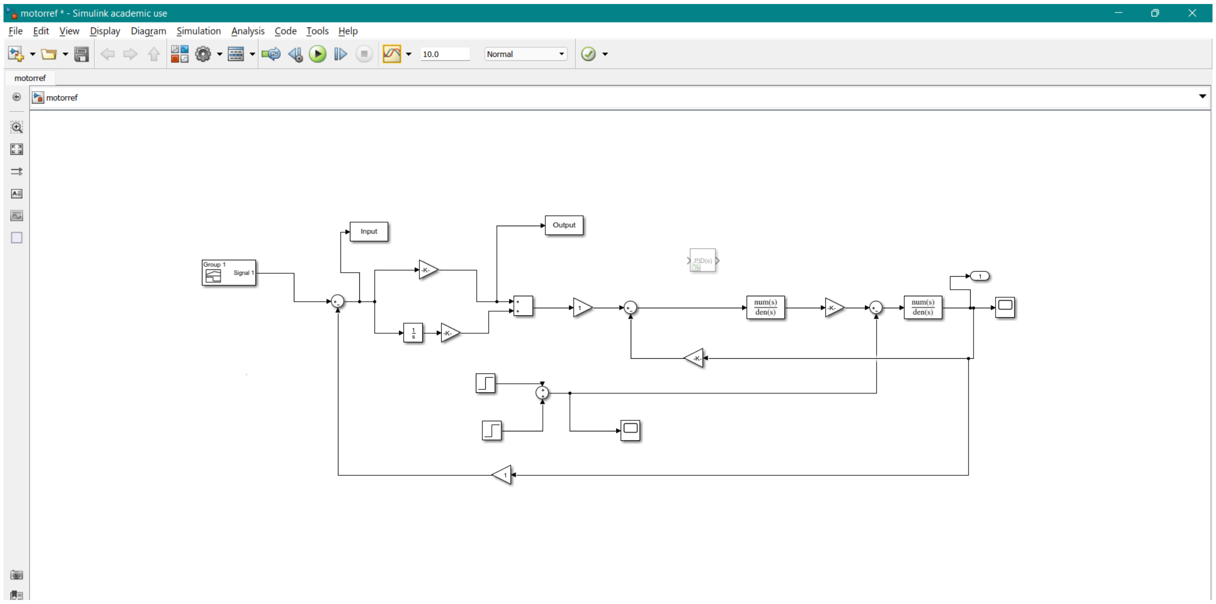


Figure 4.7 Reference Model for NARMA-L2 controller

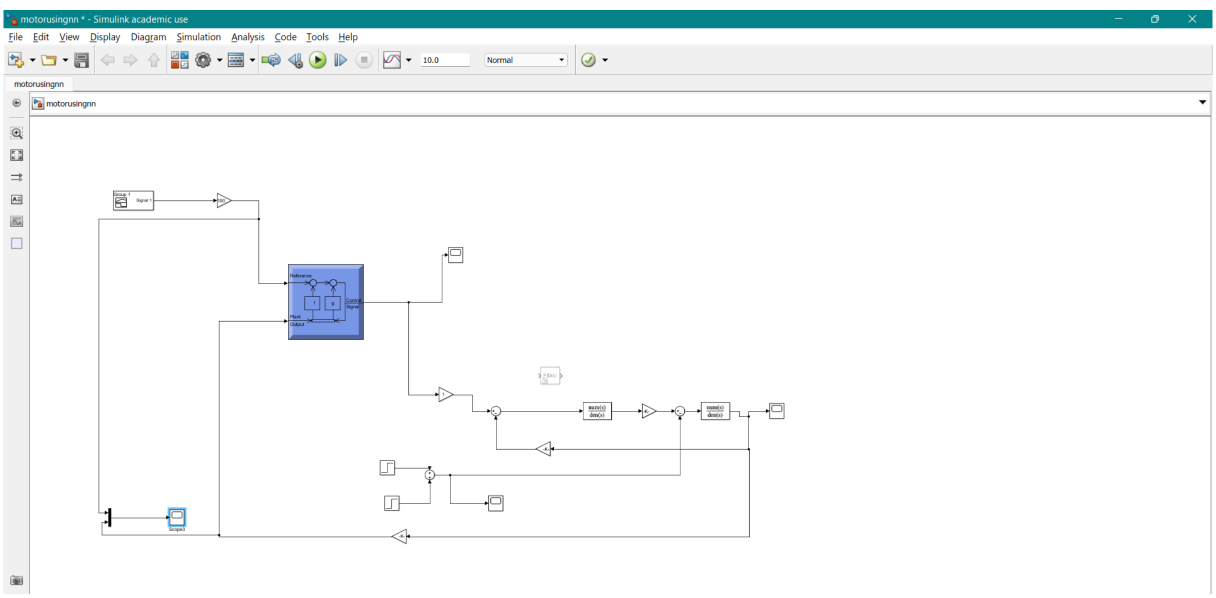


Figure 4.8 DC Motor motion control using NARMA-L2

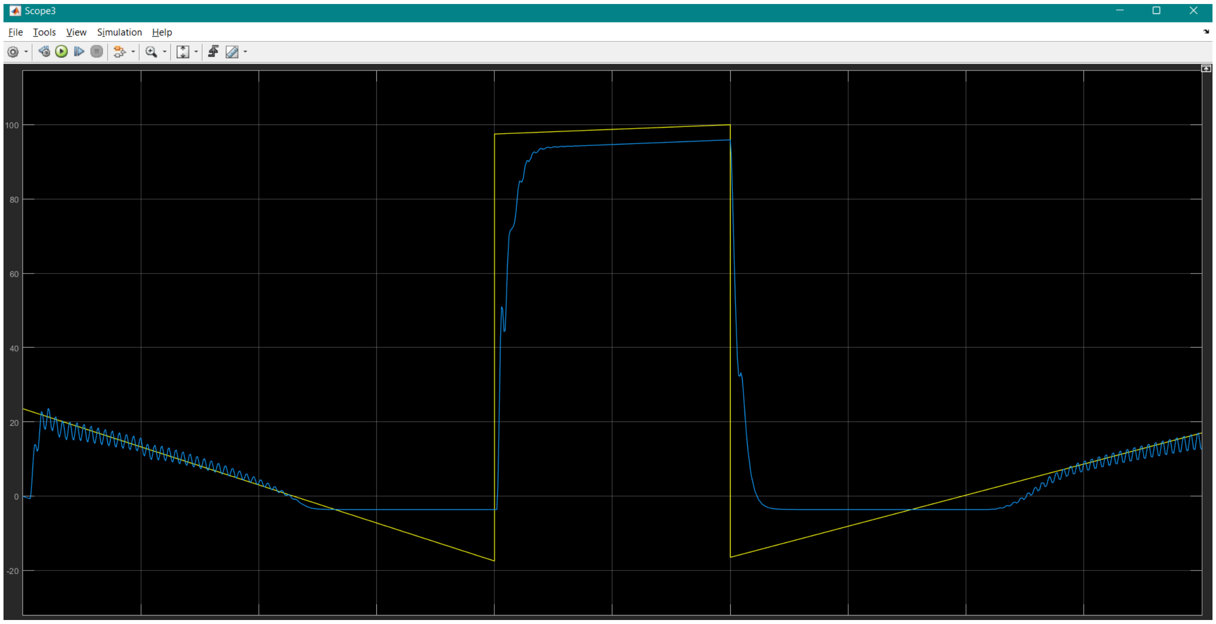


Figure 4.9 comparing speed between reference and NARMA-L2 controlled motion

Figure 4.10 comparing speed between reference and NARMA-L2 controlled motion

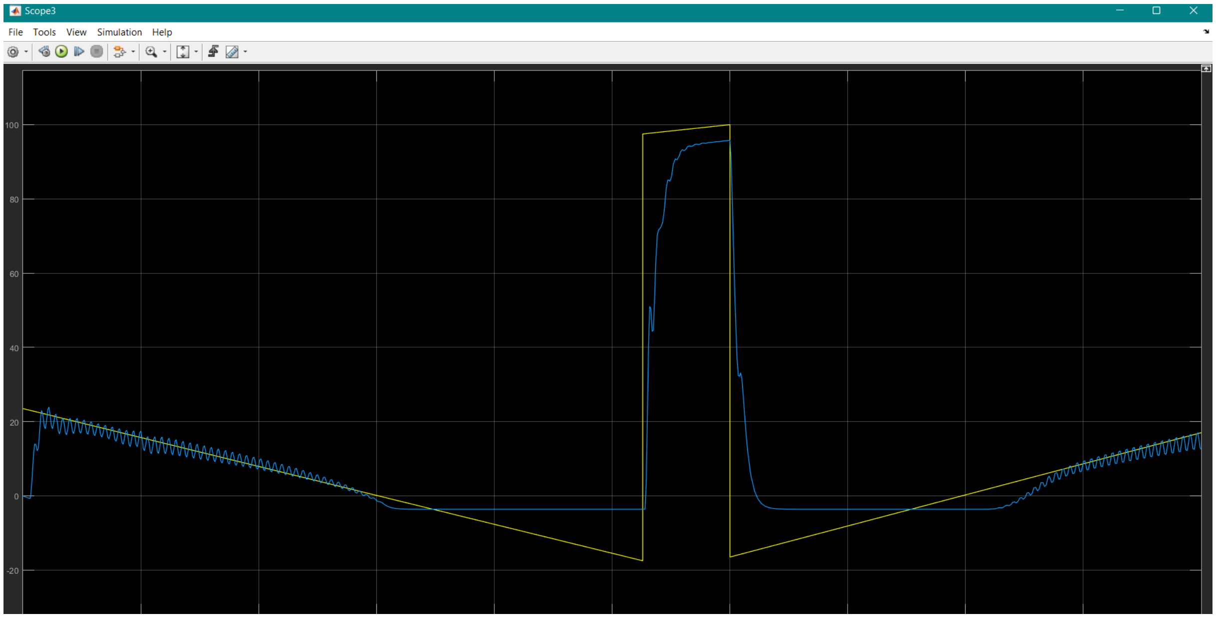


Figure 4.11 comparing speed between reference and NARMA-L2 controlled motion

Further we can use our previous matlab we used for training a simple ANN .

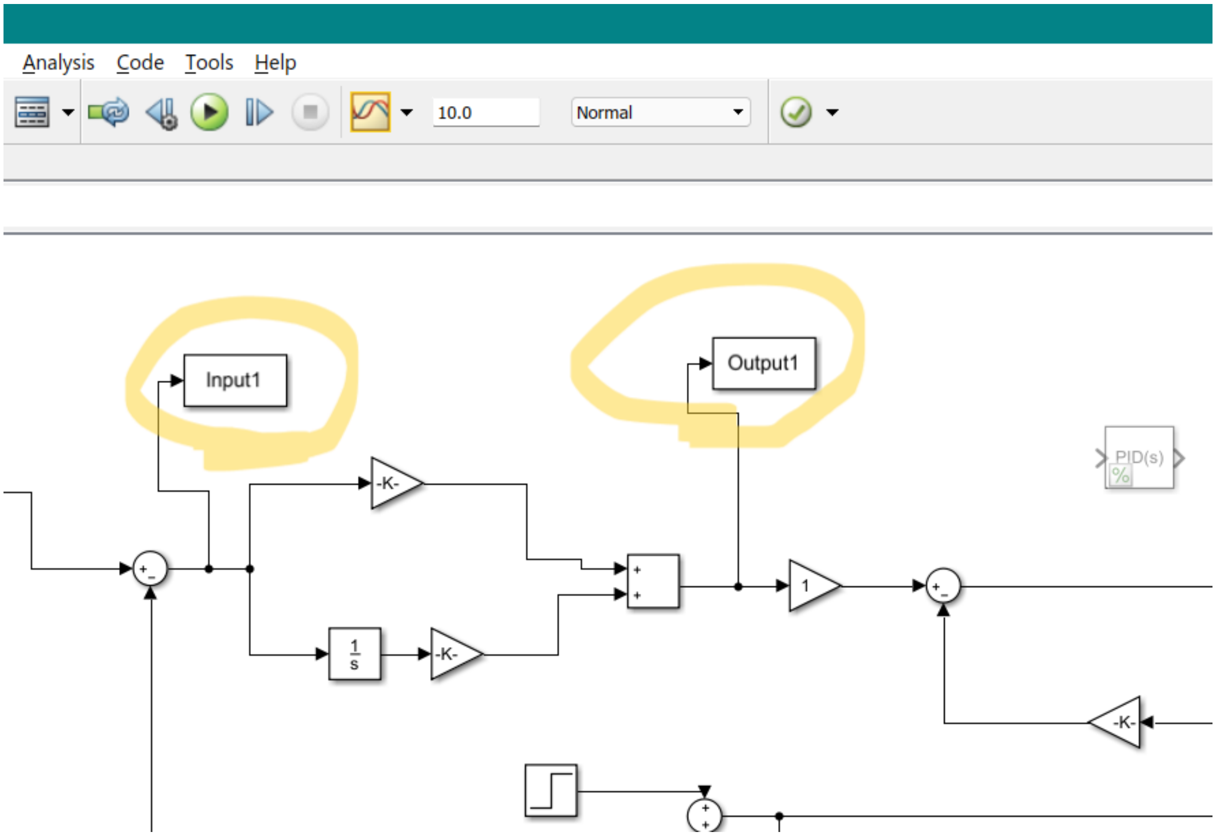


Figure 4.12 Using Input and Output variable for exporting input and output data to workspace

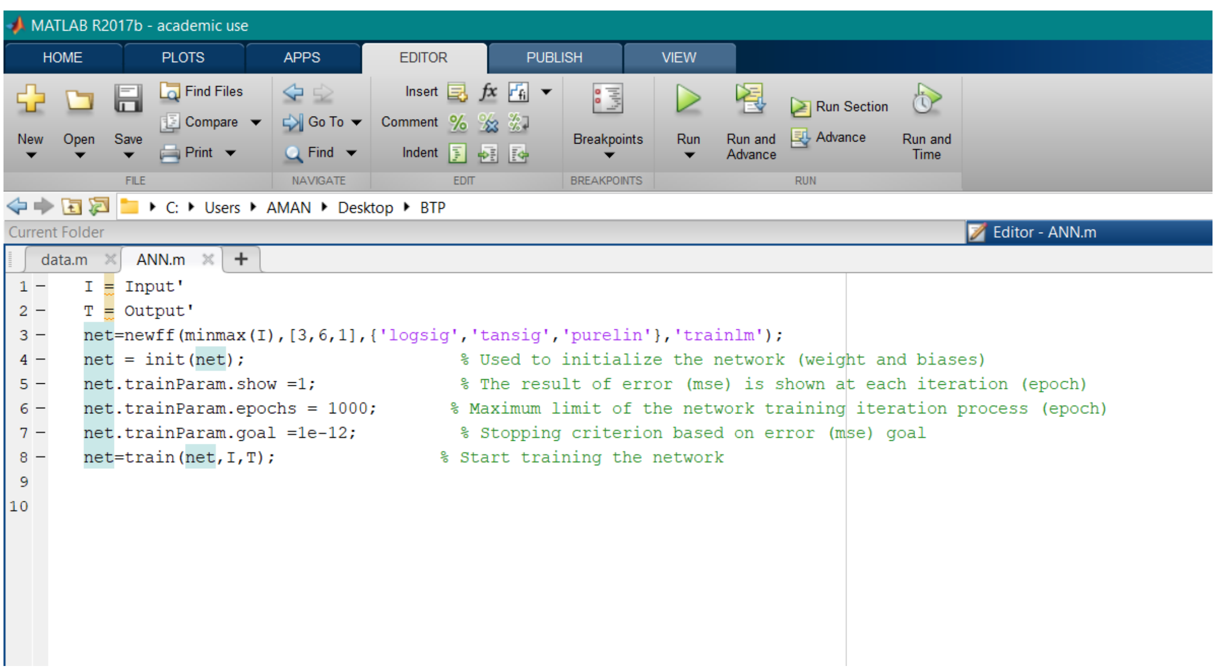


Figure 4.13 Matlab code for training simple ANN

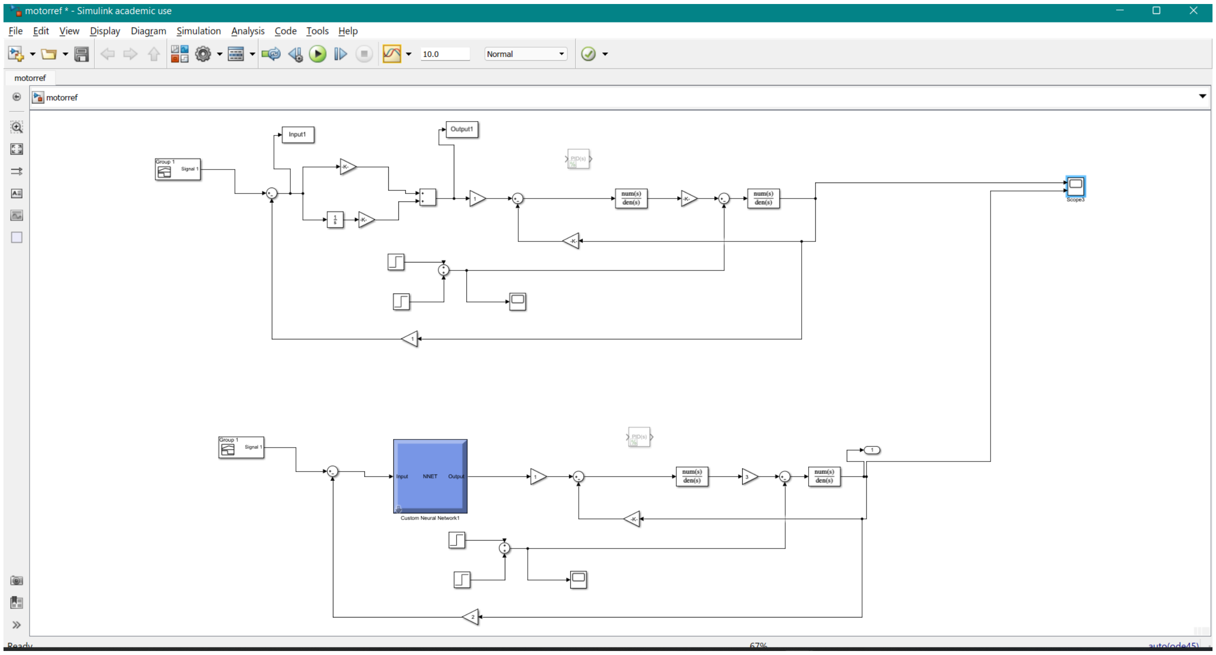


Figure 4.14 Simulink Model of PI and Custom Neural Net

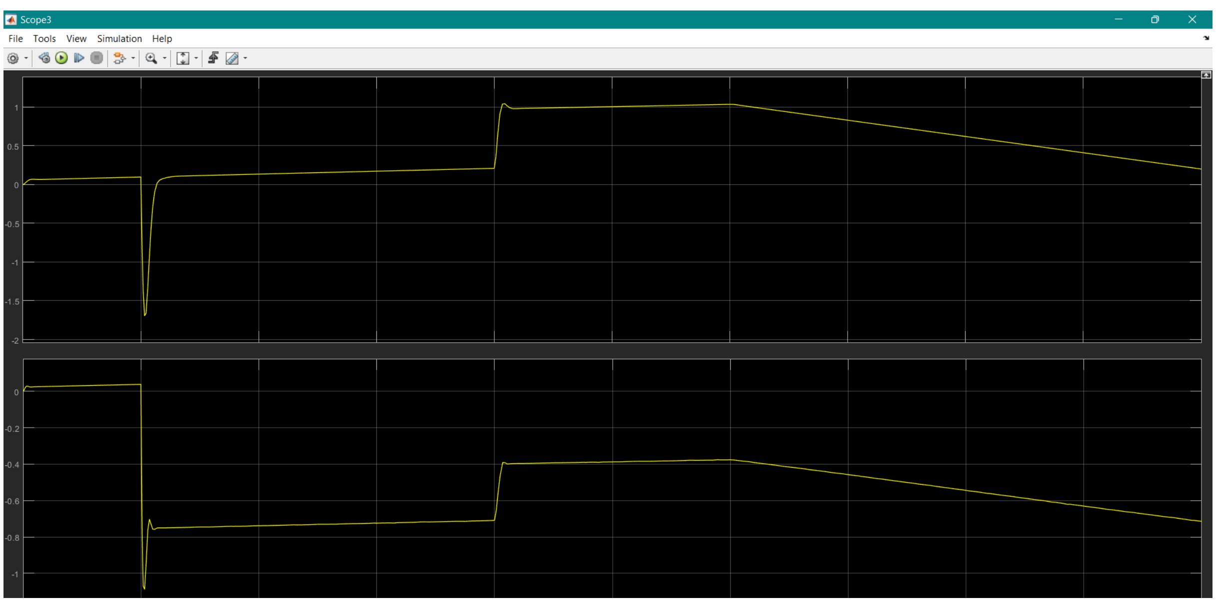


Figure 4.15 Comparing output of PI and Custom ANN 1

Figure 4.16 Comparing output of PI and Custom ANN 2

Bibliography

[1] S. Zhang, “Artificial intelligence in electric machine drives: Advances and trends.”

[2] T. H. a. K. J. Astr om, “The future of pid control.”

[3] J. L. H. H. C. R.J. Wai, R.-Y. Duan, “Wavelet neural network control for induction motor drive using sliding-mode design technique.”

[4] B. K. Bose, “Artificial intelligence techniques: How can it solve problems in power electronics?”