Indirect Vector Control of Induction Motor Using Pi Speed Controller and Neural Networks

C. Mohan Krishna M. Tech[1], G. Meerimatha M.Tech[2], U. Kamal Kumar M. Tech[3]

[1]Assistant Professor, EEE Department SVIT Engineering College ANANTAPUR
[2]Associate Professor, EEE Department SRIT Engineering College ANANTAPUR.
[3]Assistant Professor SRIT Engineering College ANANTAPUR.

Abstract: *In this paper, an implementation of intelligent controller for speed control of an induction motor (IM) using indirect vector control method has been developed and analyzed in detail. The indirect vector controlled induction motor drive involve decoupling of the stator current in to torque and flux producing components. The comparative performance of Proportional integral (PI) and Artificial Neural Networks (NN). control techniques have been presented and analyzed in this work.*

Keywords: *indirect vector controller (IVC), Proportional integral (PI), Artificial Neural Networks (NN).*

I. INTRODUCTION

A three-phase induction motor is a singly excited a.c machine in the sense that is supplied from a single source. Its stator winding is directly connected to a.c source, whereas stator winding receives its energy from stator by means of induction. Balanced three phase currents in three phase windings produce at constant amplitude rotating m.m.f wave. The stator produced m.m.f wave and rotor produced m.m.f wave, both rotate in the air gap in the same direction at synchronous speed. These two m.m.f wave are thus stationary [1] with respect to each other, consequently the development of steady electromagnetic torque is possible at all speeds but not at synchronous speed. The vector control or field oriented control (FOC) theory is the base of a special control method for induction motor drives [2]. The most commonly used controller for the speed control of Induction motor is Proportional plus Integral (PI) controller [1]. However, the PI controller has some demerits such as: the high starting overshoot, sensitivity to controller gains and sluggish response due to sudden disturbance. To overcome these problems, replacement of PI controller by an intelligent controller based on neural networks is proposed and compared with the PI controller using simulation results

II. PI CONTROLLER BACKGROUND

A PI controller responds to an error signal in a closed control loop and attempts to adjust the controlled quantity to achieve the desired system response. The controlled parameter can be any measurable system quantity such as speed, torque, or flux. The benefit of the PI controller is that it can be adjusted empirically by adjusting one or more gain values and observing the change in system response[4]. It is assumed that the controller is executed frequently enough so that the system can be properly controlled. The error signal is formed by subtracting the desired setting of the parameter to be controlled from the actual measured value of that parameter. The sign of the error indicates the direction of change required by the control input Result is a small remaining steady state error. The Integral (I) term of the controller is used to eliminate small steady state errors. The I term calculates a continuous running total of the error signal. Therefore, a small steady state error accumulates into a large error value over time. This accumulated error signal is multiplied by an I gain factor and becomes the I output term of the PI controller**.**

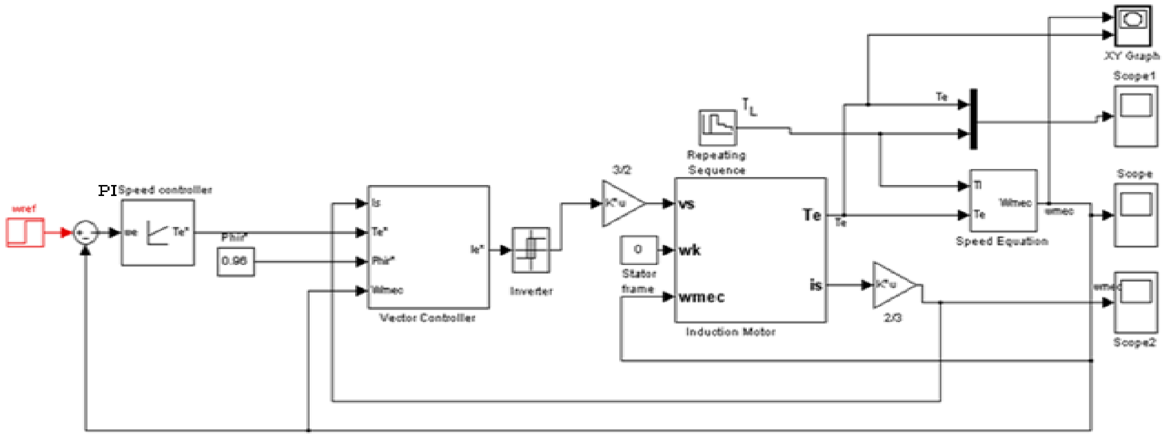


Fig-2.1 PI BASED CONTROLLER

2.1.1 Tuning of pi controllers

Proportional-integral (PI) controllers have been introduced in process control industries. Hence various techniques using PI controllers to achieve certain performance index for system response are presented[5] . The technique to be adapted for determining the proportional integral constants of the controller, called *Tuning*, depends upon the dynamic response of the plant.

In presenting the various tuning techniques we shall assume the basic control configuration wherein the controller input is the error between the desired output (command set point input) and the actual output. This error is manipulated by the controller (PI) to produce a command signal for the plant according to the relationship.

$U(s) = K_p\,(1+1/\,\tau_i s)$ ————————(1)
Or in time domain
$U(t) = K_p\,[e(t) + (1/\tau_i)\int edt]$ ————(2)
Where K_p = proportional gain
τ_i = integral time constant

If this response is *S*-shaped as in, Ziegler-Nichols tuning method is applicable**.**

2.1.2 Zeigler- Nichols Rules for tuning PI controllers:
First Rule:

The *S*-shaped response is characterized by two constants, the dead time *L* and the time constant *T* as shown. These constants can be determined by drawing a tangent to the *S*-shaped curve at the inflection point and state value of the output. From the response of this nature the plant can be mathematically modeled as first order system with a time constant *T* and delay time *L* as shown in block diagram[6] . The gain *K* corresponds to the steady state value of the output C_{ss}. The value of K_p,T_i and T_d of the controllers can then be calculated as below:

Kp=1.2(T/L) ---------- (3)
τ_i = 2L ----------(4)

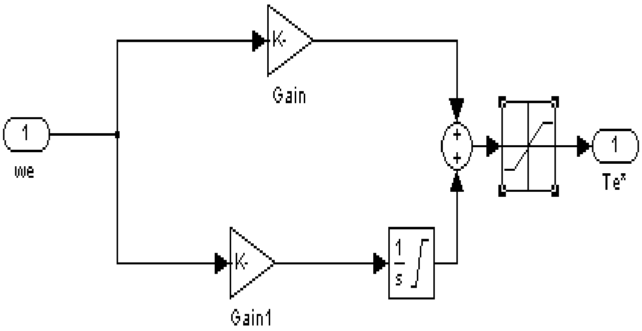


Fig 2.3 PI Controller 1st order system block diagram

III. NEURAL NETWORKS BASED CONTROLLER:

Neural networks are simply a class of mathematical algorithms, since a network can be regarded as a graphic notation for a large class of algorithms. An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. An ANN is configured for a specific application, speed control or energy saver, through a learning process. Neural network key part is a feed forward NN with two inputs and one output. NN is divided into three layers, named the input layer with 2 neurons, the hidden layer with 10 neurons, and the output layer with 1 neuron [6]. The activation function of the input neurons is linear while that of the output layer and hidden layer is sigmoid, Neural networks can perform massively parallel operations. The exhibit fault tolerance since the information is distributed in the connections throughout the network [7]. By using neural controller the peak overshoot is reduced and the system reaches the steady state quickly when compared to a conventional PI controller [8].

3.1 Program for creating the neural network:

```
Load n
k1=max (i');
k2=max (o1');
P=i'/k1;
T=o1'/K2;
n=157128;
Net = newff (minmax (P), [5 1], {'tansig' 'purelin'});
 net.trainParam.epochs = 200;
Net = train (net, P, T);
Y = Sim (net);
Plot (P, T, P, Y,’ ô') gensim (net,-1)
```

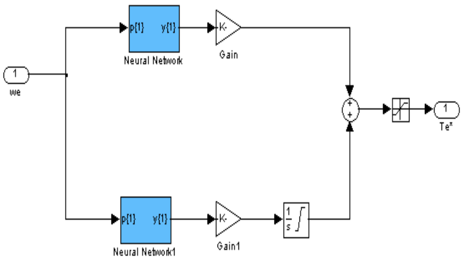


Fig 3.1 NN Controller 1st order system block diagram

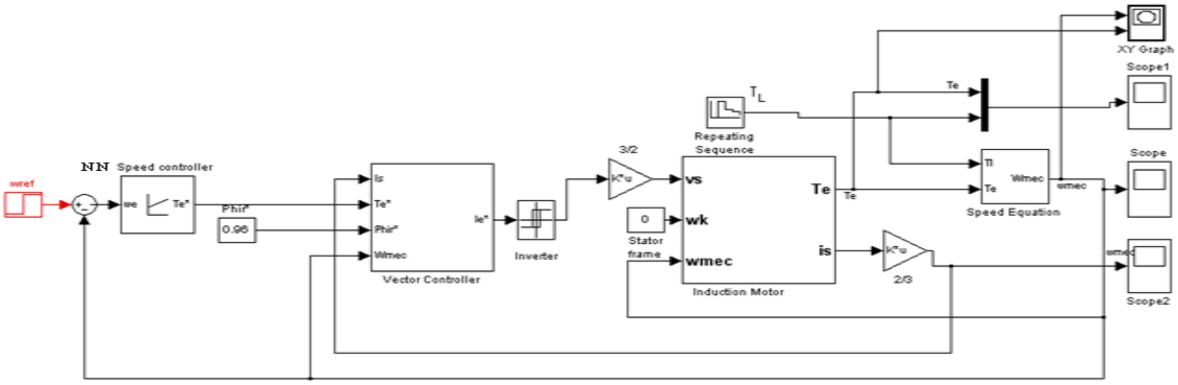


Fig-3.2 NN BASED CONTROLLER

IV. Simulation Results and discussion:

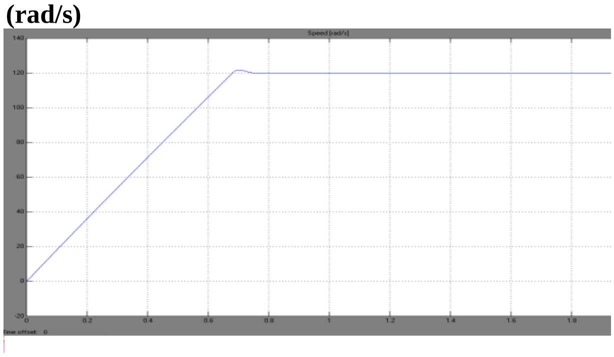
The simulation was done using the Matlab/Simulink package.

4.1 Case 1: No-Load :(for Speed □ =120rad/s, torque Te= 0, 0,0,0,0 N-m)

4.1.1 PI Controller:

Speed (rad/s) Vs Time(s):

Speed (rad/s)

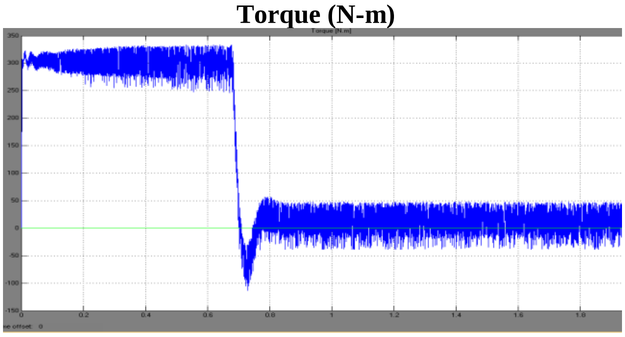


Time(s)

Fig: 4.1

Torque (N-m) Vs Time(s)

Torque (N-m)

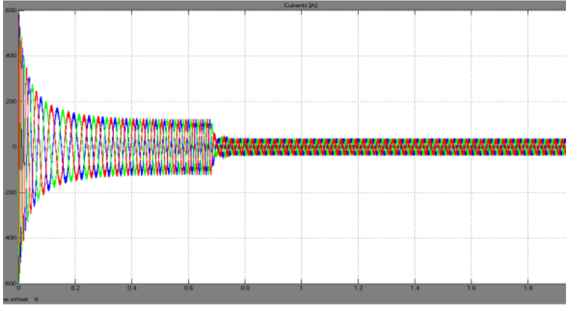


Time(s)

Fig: 4.2

Current (A) Vs Time(s)

Current (A)



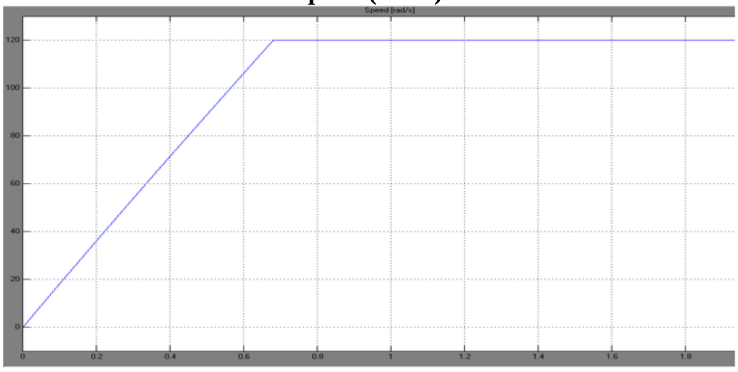
Time(s)

Fig: 4.3

4.1.2 NN Controller:

Speed (rad/s) Vs Time(s):

Speed (rad/s)

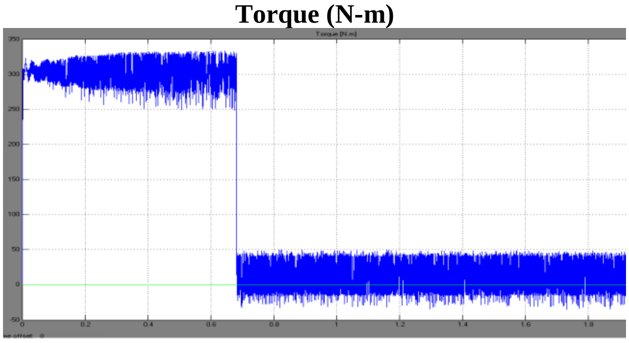


Time(s)

Fig: 4.4

Torque (N-m) Vs Time(s)

Torque (N-m)

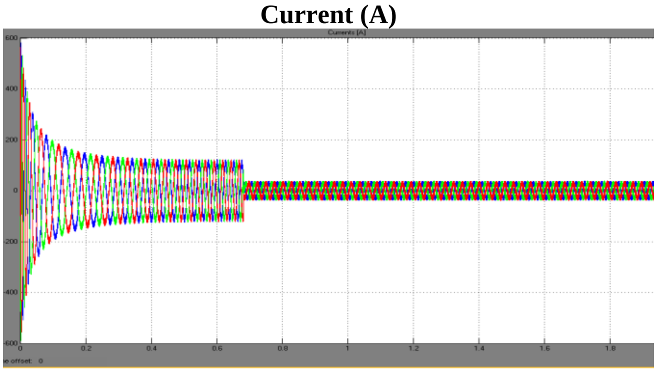


Time(s)

Fig: 4.5

Current (A) Vs Time(s)

Current (A)



Time(s)

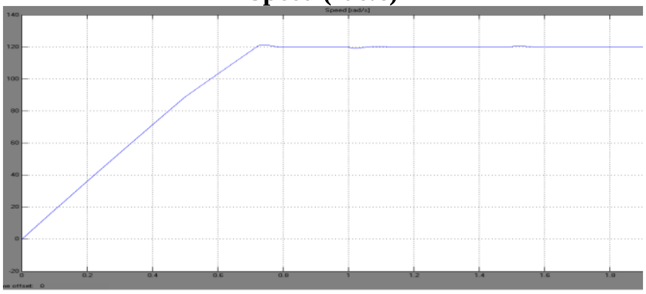
Fig: 4.6

4.2 Case 2: Step Change in –Load :(for Speed □ =120rad/s Torque Te=0, 50, 200, 100, 50 N-m)

4.2.1 PI Controller:

Speed (red/s) Vs Time(s):

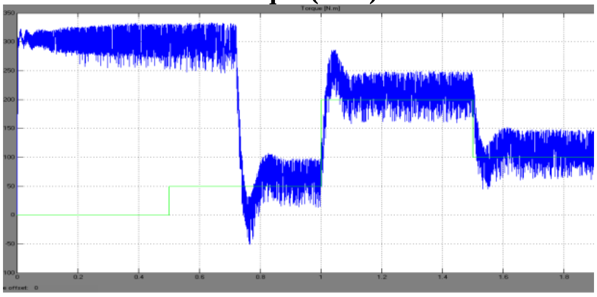
Speed (rad/s)



Time(s)
Fig: 4.7

Torque (N-m) Vs Time(s)

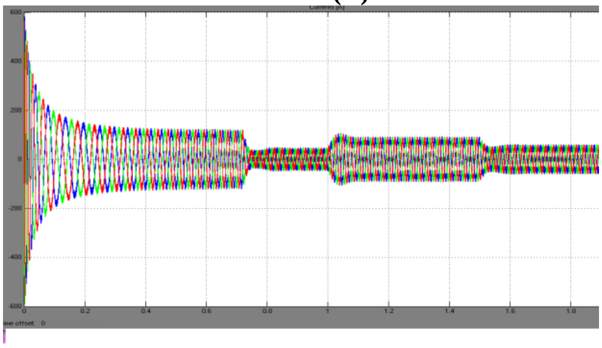
Torque (N-m)



Time(s)
Fig: 4.8

Current (A) Vs Time(s)

Current (A)

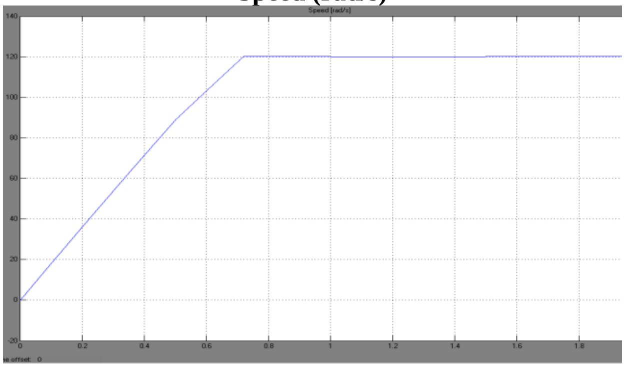


Time(s)
Fig: 4.9

4.2.2 NN Controller:

Speed (rad/s) Vs Time(s):

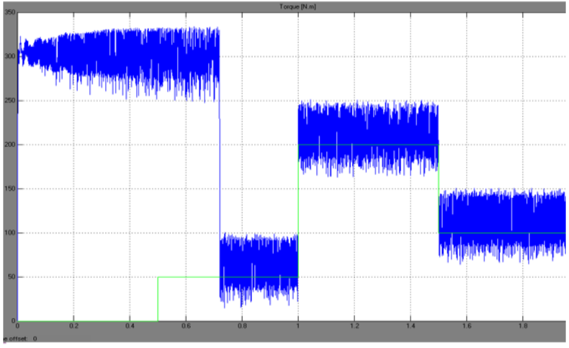
Speed (rad/s)



Time(s)
Fig: 4.10

Torque (N-m) Vs Time(s)

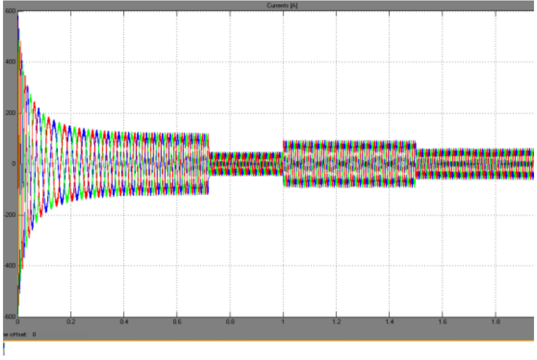
Torque (N-m)



Time(s)
Fig: 4.11

Current (A) Vs Time(s)

Current (A)



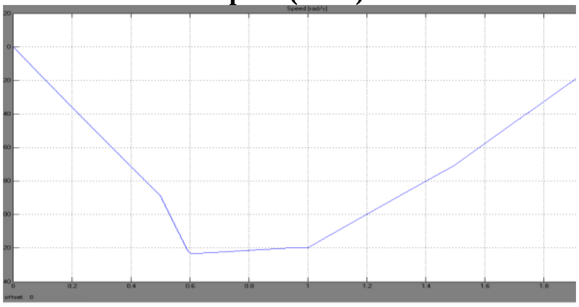
Time(s)
Fig: 4.12

4.3 Case 3: Speed Reversal: (For Speed □ =120 to 120 rad/s, Torque Te=0, 50,200,100,50N-m)

4.3.1 PI Controller:

Speed (rad/s) Vs Time(s):

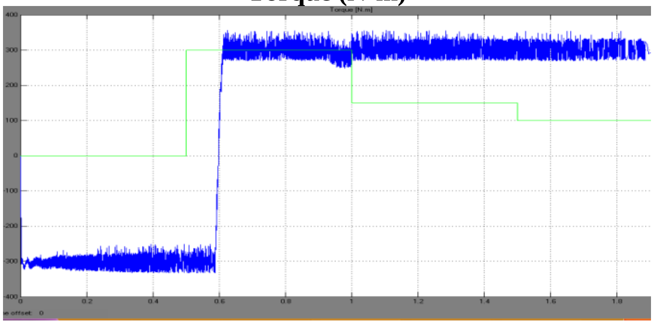
Speed (rad/s)



Time(s)
Fig: 4.13

Torque (N-m) Vs Time(s)

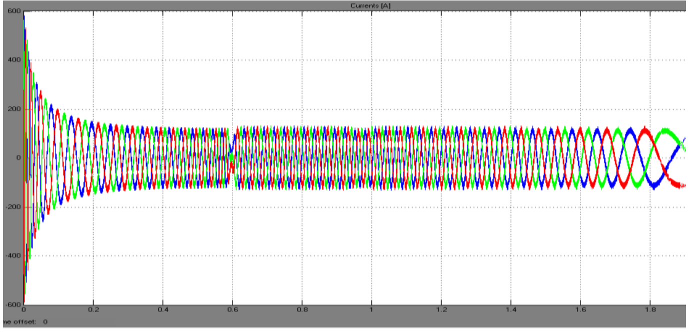
Torque (N-m)



Time(s)
Fig: 4.14

Current (A) Vs Time(s)

Current (A)

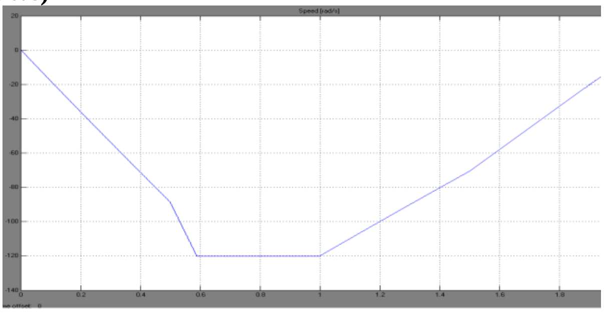


Time(s)
Fig 4.16

4.3.2 NN Controller:

Speed (rad/s) Vs Time(s):

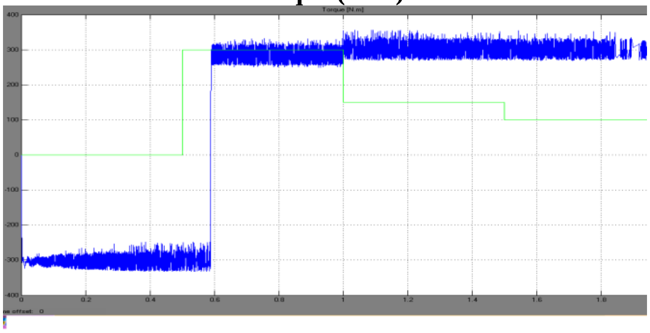
Speed (rad/s)



Time(s)
Fig: 4.18

Torque (N-m) Vs Time(s)

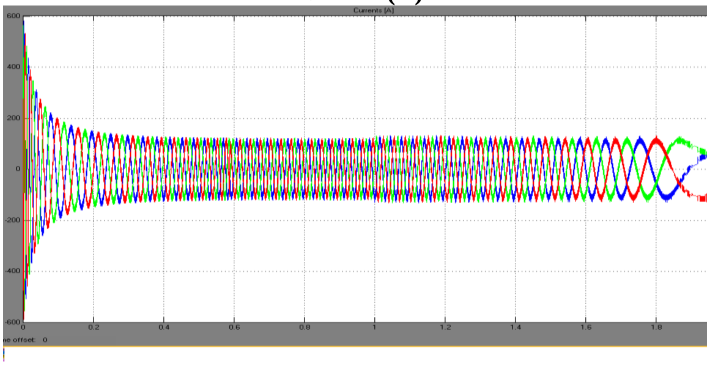
Torque (N-m)



Time(s)
Fig 4.17

Current (A) Vs Time(s)

Current (A)



Time(s)
Fig: 4.18

From the simulation results it is clear that, at no-load using PI based controller with speed 120 rad/s, torque Te=0, 0,0,0,0 N-m the characteristics obtained from PI based controller are shown in fig 4.1, 4.2, and 4.3, from the characteristics

Speed Vs Time it is observed that overshoot appeared at 0.7sec. From Torque Vs time characteristics, it is observed that under shoots appeared at 0.75 sec and from Current Vs time characteristics it is observed that disturbances occurred at 0.75 sec because of mismatching of op-amps used in speed controller. Using neural network based speed controller, over shoots at 0.7 sec in Speed Vs Time characteristics are eliminated, under shoots at 0.75 sec in Torque Vs Time characteristics are eliminated and disturbances at 0.75 sec in Current Vs Time characteristics are eliminated. These characteristics are shown in fig 4.4, 4.5, 4.6.

At step change using PI based controller with speed 120 rad/s, torque Te=0,50,200,100,50 N-m the characteristics obtained from PI based controller are shown in fig 4.7,4.8,and 4.9, from the characteristics Speed Vs Time it is observed that overshoot appeared at 0.7,1,1.5 sec . From Torque Vs time characteristics, it is observed that under shoots appeared at 0.75, 1, 1.5 sec. and from Current Vs time characteristics it is observed that disturbances occurred at 0.75, 1, 1.5 sec. Using neural network based speed controller, over shoots at 0.7, 1, 1.5 sec in Speed Vs Time characteristics are eliminated, under shoots at 0.75, 1, 1.5 sec in Torque Vs Time characteristics are eliminated and disturbances at 0.75, 1, 1.5 sec in Current Vs Time characteristics are eliminated . These characteristics are shown in fig 4.10, 4.11, and 4.12.At speed reversal using PI based controller with speed -120 to 120 rad/s, torque Te=0,50,200,100,50 N-m the characteristics obtained from PI based controller are shown in fig 4.13,4.14,and 4.15, from the characteristics Speed Vs Time it is observed that overshoot appeared at 0.5,1,1.5 sec. From Torque Vs time characteristics, it is observed that under shoots appeared at 0.95 sec and from Current Vs time characteristics it is observed that disturbances occurred at 0.6 sec. Using neural network based speed controller, over shoots at 0.5, 1, 1.5 sec in Speed Vs Time characteristics are eliminated, under shoots at 0.95 sec in Torque Vs Time characteristics are eliminated and disturbances at 0.6 sec in Current Vs Time characteristics are eliminated. These characteristics are shown in fig 4.16, 4.17, and 4.18.

V. CONCLUSION

Challenging and excelling the human brain is one of our long cherished dreams. Intelligent controllers reflect human thinking, human perception and human way of reasoning. Simulation studies show that the Neural Networks based controller provides better results for an induction motor when compared to a conventional PI controller. So, Neural Networks controller is an attractive technique when the plant model is complex.

REFERENCES

[1]. B. K. Bose. Modern Power Electronics and AC Drives. Prentice-Hill PTR Companies, Inc. Upper Saddle River, NJ07458, 2002.
[2]. C.M.Liaw.Y.S.Kung and C.M.Wu Design and implementation of a high performance field oriented induction motor drive. IEEE Tran Ind.Applicat. Vol.38.pp.275-282, 1991.
[3]. R. Krishnan. Electric motor drives modeling, analysis and Control. PHI Pvt. Ltd, New Delhi, 2003.
[4] Y miloud and A.Draou." Application of a Three Level GTO Voltage Source PWM Inverter to feed AC Motors for Variable-Speed Drive with a PI- Controller". International Conference on Electronics, ICEL' 2000. Vol. 1, November 13-15. pp 223-227. Oran, Algeria
[5] Woodley, K.M., Li, H., Foo, S.Y.: Neural Network modeling of Torque Estimation and d-q Transformation for Induction Machine. Eng. Appl. of Artif. In tell. 18(1), 57–63 (2005)
[6]. C.M.Liaw.Y.S.Kung and M.S.Ouyang Identification and control of induction machines using artificial neural networks. IEEE Trans Ind.Applicat. Vol.31.pp.612-619, 1995.
[7]. H.A.Al-Rashidi, A.Gastli, A.Al-Badi, "Optimization of Variable Speed Induction Motor Efficiency Using Artificial Neural Network".
[8] Howard Demuth, Mark Beal, "Neural Network Toolbox for Use with Matlab", User Guide, the Math Work Inc. June 1992

AUTHORS

Mohan Krishna M.Tech Assistant Professor SriVenkateshwaraInstitute of Technology Engineering College ANANTAPUR.

G.Meerimatha M.Tech Associate Professor, EEE Department SriRamanujan Institute of Technology Engineering College ANANTAPUR.

U.Kamal Kumar M.Tech Assistant Professor SriRamanujan Institute of Technology Engineering College ANANTAPUR.
.