

**SPEED CONTROL OF BRUSHLESS DC MOTOR USING ARTIFICIAL  
NEURAL NETWORK TUNED PID CONTROLLER**

*Thesis submitted towards the partial fulfillment of the requirements of the degree*

*of*

**Master of Engineering**

*in*

**Power Systems & Electric Drives**

*Submitted by*

**Amandeep Gill  
(Regd. No. 801041001)**

*Under the supervision of*

**Ms. Manbir Kaur  
Associate Professor, EIED**

**&**

**Mr. Nirbhowjap Singh  
Assistant Professor, EIED**



**JULY, 2012**

**ELECTRICAL & INSTRUMENTATION ENGINEERING DEPARTMENT  
THAPAR UNIVERSITY, PATIALA**

*(Established under section 3 of UGC act 1956)*

**PATIALA – 147004**

## CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled "**Speed control of Brushless DC motor using Artificial Neural Network tuned PID Controller**", in partial fulfillment of the requirements for the award of degree of Master of Engineering in Power Systems & Electric Drives submitted in Electrical & Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Ms. Manbir Kaur, Associate Professor, EIED and Mr. Nirbhowjap Singh, Assistant Professor, EIED.

The matter presented in the thesis has not been submitted for the award of any other degree of this or any other university.

*Amandeep Gill*  
12.07.2012

Amandeep Gill  
(Roll No: 801041001)

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

*Manbir Kaur*  
12-7-2012

Ms. Manbir Kaur  
Associate Professor, EIED  
Thapar University, Patiala

*Nirbhowjap Singh*

Mr. Nirbhowjap Singh  
Assistant Professor, EIED  
Thapar University, Patiala

**Countersigned by:**

*S. Ghosh*

Dr. Smarajit Ghosh  
Professor & Head, EIED  
Thapar University, Patiala

*S. K. Mohapatra*

Dr. S. K. Mohapatra  
Dean of Academic Affairs  
Thapar University, Patiala

## ACKNOWLEDGEMENT

I express my sincere gratitude to **Ms. Manbir Kaur**, *Associate Professor*, EIED, Thapar university, Patiala and **Mr. Nirbhawjap Singh**, *Assistant Professor*, EIED, Thapar University, Patiala for their valuable guidance, constructive suggestions and constant encouragement in nurturing the work.

I am grateful to **Dr. Smarajit Ghosh**, *Professor & Head*, EIED, Thapar university, Patiala for providing the facilities for the completion of the work.

I also express my sincere thanks to **Dr. Abhijeet Mukherji**, *Director*, Thapar University, Patiala & the faculty and staff of **EIED, Thapar University, Patiala**, for their help, inspiration and moral support.

Above all, I express my indebtedness to the “*ALMIGHTY*” for His blessings and kindness.

*Aman Gill*  
12.07.2012

**AMANDEEP GILL**  
**Regd. No. 801041001**

## ABSTRACT

In automotive industry, high performance drives are gaining popularity due to their high efficiency, good dynamic response and low maintenance. The precise rotor movement over a period of time in certain applications such as robotics, guided manipulation and dynamic actuation must be achieved even when the system loads, inertia and other controlling parameters are varying. To do this, the speed control strategy must be adaptive, robust, accurate, and simple to implement.

The conventional feedback controllers those are based on linear control theory and are much easier to understand and implement but suffer the disadvantages when the operating points of the process or the plant parameters are changed due to disturbances. Fixed-gain feedback controllers need to be returned to obtain the new optimal settings. For the processes with variable time delays, varying plant parameters, large non-linearities and considerable process noise, the PID controller does not give optimal performance.

In view of this an adaptive controller that can modify its behaviour in response to the dynamical changes in the process and the disturbances is developed. Artificial Neural Network (ANN) based intelligent controller can mimic adaptive nature of controller used in non-linear system through its highly parallel and distributed structure. Neural network can generate a nonlinear mapping between the inputs and outputs of a system without the need for a predetermined model.

The aim of the thesis is to design a simulation model of brushless dc motor and to control its speed at different values of load torques. The accurate speed control is proposed to achieve through PID controller. The parameters of PID controller are tuned by on line training of the artificial neural network. The performance of the PID type controller with fixed gain, conventional integral controller (PI) and ANN based PID controller have been compared through MATLAB simulation results with focus on feasibility, reliability and accuracy for BLDC permanent magnet synchronous motor drive system. The qualitative and quantitative comparison have shown the superiority of the performance of PID tuned through artificial neural network over integral and PID controller in terms of the reliability and feasibility. The reported percentage overshoot error is well within the permissible limits and rise time is also very low.

## TABLE OF CONTENTS

---

<b>CERTIFICATE</b>	i
<b>ACKNOWLEDGEMENT</b>	ii
<b>ABSTRACT</b>	iii
<b>TABLE OF CONTENTS</b>	iv
<b>LIST OF FIGURES</b>	ivi
<b>LIST OF TABLES</b>	viii
<b>LIST OF SYMBOLS AND ABBREVIATIONS</b>	ix
<b>1 CHAPTER-1: INTRODUCTION</b>	<b>1-4</b>
1.1 Overview	1
1.2 Literature Survey	2
1.3 Objective of Thesis work	4
1.4 Organization of Thesis	4
<b>2 CHAPTER-2: BRUSHLESS DC MOTOR (BLDCM)</b>	<b>5-17</b>
2.1 Introduction	5
2.2 Types of Brushless DC motor (BLDCM)	6
2.3 Components of BLDC Permanent magnet motor	7
2.3.1 Rotor	7
2.3.2 Stator	8
2.3.3 Hall Sensors	8
2.4 Motor operation	9
2.5 Speed-Torque characteristics	12
2.6 Modelling of BLDC Permanent magnet motor	13
2.7 Characteristic features of Brushless dc motor	17
<b>3 CHAPTER-3: ADAPTIVE CONTROLLERS</b>	<b>18-29</b>
3.1 Introduction	18
3.2 Adaptive control	18
3.2.1 Feed Forward control system	19
3.2.2 Feedback control system	19
3.2.3 Open-loop control system	20
3.3 Transient response specifications of second order control system	20
3.4 Control actions	22
3.4.1 Proportional control	22
3.4.2 Integral control	23
3.4.3 Derivative control	24
3.4.4 Characteristics of P, I & D controllers	24
3.4.5 Proportional & Integral (PI) controller	25
3.4.6 Proportional & Derivative (PD) controller	25
3.4.7 Proportional, Integral & Derivative (PID) controller	26

3.5	Tuning rules for controllers	27
3.3.1	Ziegler-Nichols rule for tuning of PID & PI controllers	27
<b>4</b>	<b>CHAPTER-4: ARTIFICIAL NEURAL NETWORK (ANN)</b>	<b>30-43</b>
4.1	Introduction	30
4.2	Salient features of Artificial Neural Network (ANN)	30
4.3	Artificial Neural Network (ANN) Overview	31
4.3.1	ANN Architecture	33
4.3.2	Weights & Network training	34
4.3.3	Transfer functions	35
4.4	Multilayer Feedforward network with Backpropagation	37
4.4.1	Convergence & Training efficiency	37
4.4.2	Backpropagation method	38
4.4.3	Backpropagation learning Algorithm	41
4.5	Levenberg-Marquardt method	42
4.5	ANN controller as an adaptive controller mechanism	42
<b>5</b>	<b>CHAPTER-5: SIMULATION AND RESULTS</b>	<b>44-58</b>
5.1	Introduction	44
5.2	Simulink model of BLDCM	44
5.2.1	Circuit description of Simulink model of BLDCM	44
5.3	Artificial Neural Network model	50
5.3.1	Parameters of ANN model	51
5.4	Transient response of Speed for BLDCM with ANN tuned PID controller	53
5.5	Comparison of Transient response of speed BLDCM with ANN tuned PID, PID & PI controllers	54
5.6	Comparison between the rise time and maximum percent overshoot	58
<b>6</b>	<b>CHAPTER-6: CONCLUSIONS &amp; FUTURE SCOPE</b>	<b>59</b>
6.1	Conclusion	59
6.2	Future Scope	59
	<b>REFERENCES</b>	<b>60-62</b>
	<b>APPENDIX-A</b>	<b>63</b>

## LIST OF FIGURES

---

<b>Serial No.</b>	<b>Name of the Figure</b>	<b>Page No.</b>
Figure 2.1	Conventional dc motor & BLDCM	6
Figure 2.2	In-runner & Out-runner BLDCM	6
Figure 2.3	Cross sectional view of a BLDCM	7
Figure 2.4	Four pole and eight pole BLDCM rotor	7
Figure 2.5	BLDCM stator	8
Figure 2.6	Stator of BLDCM with Hall Effect Sensors	8
Figure 2.7	Self-control Mode of BLDCM	10
Figure 2.8	Stator phase EMF and current waves showing the inverter conducting switches	11
Figure 2.9	Torque/Speed characteristics of BLDCM	12
Figure 2.10	Permanent magnet (PM) BLDCM waveforms	13
Figure 3.1	Feedback control system	19
Figure 3.2	Transient response specifications of second order system	21
Figure 3.3	Proportional control action	23
Figure 3.4	PID control of a plant	26
Figure 3.5	Unit-step response of a Plant	28
Figure 3.6	S-shaped response curve	28
Figure 4.1	Schematic of Biological Neurons	32
Figure 4.2	Multiple Input Neuron	32
Figure 4.3	Neural Network Architecture	33
Figure 4.4	Linear Transfer Function	35
Figure 4.5	Log-Sigmoid Transfer Function	36
Figure 4.6	Hyperbolic Tangent Sigmoid Transfer Function	36
Figure 4.7	Three layer back-propagation network	39
Figure 4.8	ANN based PID controller control model	43
Figure 5.1	Waveform of the Stator current <sub>a</sub> (A) of BLDCM	45
Figure 5.2	Waveform of Stator back EMF (V)	46

Figure 5.3	Waveform of the Electromagnetic torque $T_e$ (Nm) of BLDCM	46
Figure 5.4	Simulation model of BLDCM with ANN based PID controller	47
Figure 5.5	Simulation model of BLDCM with PID controller	48
Figure 5.6	Simulation model of BLDCM with PI controller	49
Figure 5.7	ANN model	50
Figure 5.8	Two layer structure of Neural Network	50
Figure 5.9	Training Performance	52
Figure 5.10	Training State	52
Figure 5.11	Transient response of speed for BLDCM with ANN tuned PID controller	53

## LIST OF TABLES

---

<b>Serial No.</b>	<b>List of Tables</b>	<b>Page No.</b>
Table.3.1	Effects of each of controllers $K_p$ , $K_i$ , & $K_d$ on a closed-loop system	24
Table.3.2	Values of $k_p$ , $T_i$ , & $T_d$ obtained by Ziegler & Nichols Tuning	29
Table.5.1	ANN model parameters	51
Table.5.2	Comparison Between the rise time and Percentage overshoot of ANNPID, PID & PI controllers	58

## LIST OF SYMBOLS AND ABBREVIATIONS

---

AC	Alternating Current
ANN	Artificial Neural Network
BLDCM	Brushless dc motor
BP	Back-propagation
DC	Direct Current
ECM	Electronically Commutator motor
emf	Electromotive force
eqn	Equation
IGBT	Insulated gate bipolar junction transistor
LM	Levenberg-Marquardt
MOSFET	Metaloxide semiconductor field effect transistor
NN	Neural network
PID	Proportional-integral-derivative
PMAC	Permanent magnet AC motor
PMSM	Permanent magnet synchronous motor
RFI	Radio frequency interference
RLC	Resistance inductance capacitance
rpm	Revolutions per minute
VLSI	Very large scale integrated
Z&N	Ziegler-Nichols
$\omega_m$	Angular velocity
$\phi_g$	Air gap flux per pole
$\lambda_p$	Flux linkages
$\theta_r$	Rotor position
$\delta_{oi}$	Error signal
B	Flux density
$b_k$	Externally applied bias
B(s)	Feedback signal

$C(s)$	Output signal
$d^{(k)}$	Output pattern
$e$	Processing elements of hidden layers
$e_{as}, e_{bs} \text{ \& } e_{cs}$	Induced emfs
$E_p$	Peak value of emf
$E(s)$	Error signal
$f_{as}, f_{bs} \text{ \& } f_{cs}$	Functions
$f(a)$	Linear transfer function
$f$	Activation function
$G(s)$	Forward gain
$H(s)$	Feedback gain
$i_{as}, i_{bs} \text{ \& } i_{cs}$	Stator phase currents
$J$	Inertia
$k$	Number of neurons
$K_i$	Integral gain
$K_d$	Derivative gain
$K_p$	Proportional gain
$L$	Delay time
$L, M$	Inductance & mutual inductance of BLDCM
$l$	Length of conductor
$m$	Processing elements in input layer of NN
$n$	Processing elements in output layer of NN
$N$	Number of conductors in series per phase
$P$	Proportional
$p$	$d/dt$
$r$	Radius of rotor bore
$R_s$	Stator resistance per phase
$R(s)$	Reference signal
$T$	Time constant
$T_d$	Derivative time constant

$T_e$	Electromagnetic torque
$T_i$	Integral time constant
$T_l$	Load torque
$u_k$	Linear combiner output
$v$	Velocity
$w_{km}$	Synaptic weights
$x^{(k)}$	Input pattern of NN
$X_m$	Input signals
$y^{(k)}$	Actual output of NN
$Y_k$	Output signal of the neuron

# CHAPTER 1

## INTRODUCTION

---

### 1.1 OVERVIEW

Brushless dc motors (BLDCM) provide high efficiency, reliability, ruggedness and high precision of control when compared to conventional motors. It has the best torque vs. weight or efficiency characteristics. They are used in military, grinding, aircraft, automotive applications, communications equipment etc. Brushless dc motors (BLDCM) have been desired for small horsepower control motors such as: heating, ventilation, and air conditioning systems to achieve great energy saving effects for partial loads by lowering motor speeds. In addition, BLDCM have been used as variable speed drives in wide array of applications due to their high efficiency, silent operation, compact form, reliability, and low maintenance [1]. BLDCM drives were widely employed in industry due to their intrinsic robustness and high torque-to-weight ratio. The availability of cheap embedded processing power in recent years paved the way for the widespread use of sensor less control techniques; the removal of speed and position sensors leads to substantial increase of robustness and cost savings [2]. Due to the high torque to volume ratio of BLDCM, it dominates for High Performance Drives (HPD) applications, such as robotics, guided manipulation and dynamic actuation, the precise rotor movement over a period of time must be achieved. A multi-robot system performing a complementing function must have the end effectors move about the space of operation according to a pre-selected time tagged trajectory. Also, the Brushless dc motor, as the name implies, has no brushes. This is an essential requirement for several industrial applications such as airplane actuation, food and chemical industries. This must be achieved even when the system loads, inertia and parameters are varying. To do this, the speed control strategy must be adaptive, robust, accurate, and simple to implement [3].

Conventional feedback controllers, such as the PID or the linear quadratic, need accurate mathematical models describing the dynamics of the system under control. This can be a major limiting factor for systems with unknown varying dynamics. Even if a model can be obtained for the system under control, unknown conditions such as saturation, disturbances, parameter drifts, and noise may be impossible to model with acceptable accuracy. For most of the basic electric drives applications, these unknown conditions in addition to the system nonlinearities can be ignored, but it may lead to unacceptable tracking performance. High accuracy is not usually imperative [4]. Some adaptive control techniques, such as the variable

structure and the self-tuning, do not need a model for system dynamics. The dynamic model is, rather, developed based on the on-line input/output response of the system under control [5]. These models are usually linear but updated every several sampling intervals for these reasons, properly intelligent methods like Artificial Neural Network (ANN) can be adopted to solve the problems of electric drives as BLDCM control for high performance applications. From the viewpoint of industrial applications, ANN applications can be divided into four main categories (i) Modelling and Identification (ii) Optimization and Classification (iii) Process Control (iv) Pattern Recognition. A multi-layer Neural Network architecture is proposed for the speed control of BLDCM that performs two functions. The first is to identify the nonlinear system dynamics at all times. Hence, detailed and elaborate models for the BLDCM are not needed. Furthermore, unknown nonlinear dynamics that are difficult to model such as load disturbances, system noise and parameter variations can be recognized by the trained neural network. The second function of the ANN is to control the motor voltage through tuning of gain parameters of PID controller so that the speed and position are made to follow pre-selected tracks (trajectories) at all times. The control action emulated by the ANN is based on the indirect model reference adaptive control [6]. Artificial Neural Network (ANN) using parallel and distributed processing units can achieve the functions of system mode and control. The ANN has several key features like its robustness, fault tolerant, noise free and capability of generating a nonlinear mapping between the inputs and outputs of an electric drive system without the need for a predetermined model [7] makes it suitable for speed control of BLDC motors under varying load torque.

## **1.2 LITERATURE SURVEY**

Brushless DC permanent magnet synchronous motor being a highly non-linear unit, it is quite difficult to obtain a good performance when applied to control the speed of the unit. Many researches in the past decade have been carried out to implement the offline and on line control of BLDCM a few of these are summarised as:

Pillay. P *et al.* [8] has done comparison between the permanent magnet synchronous motor (PMSM) and the brushless dc motor (BDCM) for their similarities and different operating characteristics and control requirements. These two motor drives had been differentiated on the basis of known engineering criteria like power density, torque per unit, current, speed range, feedback devices, inverter rating, cogging torque, ripple torque, and parameter sensitivity. Cheng. M *et al.* [9] presented the design and analysis of a new doubly salient permanent magnet (DSPM) motor. The initial calculation of motor dimensions and

parameters, namely, the core diameter, stack length, permanent magnet size, and winding turns, are also discussed. Ganesh. C *et al.* [10] have incorporated the estimation of mechanical parameters such as moment of inertia & friction coefficient of BLDCM and load at various load settings to achieve optimum position control. Chen. Z *et al.* [11] have proposed a new adaptive sliding observer for position and velocity-sensorless controls of a cylindrical brushless dc motor. Sliding mode is applied for the current estimation. Hemanand.T *et al.* [12] have presented the speed control of brushless dc motor drive employing hard chopping PWM technique developed using digital signal processor. Park K.H [13] had presented a novel robust speed control method based on the load torque observer of high-performance brushless DC (BLDC) motor. The feedforward compensator with disturbance torque observer was proposed to improve the servo stiffness without increasing the speed controller gain. Zhu. G *et al.* [14] have discussed the speed tracking control problem for a permanent-magnet synchronous motor (PMSM) in the presence of an unknown load torque disturbance and have introduced a speed tracking control law using the exact linearization methodology. The stability of the closed-loop system composed of a nonlinear speed tracking controller and an observer has been studied by the way of Lyapunov theory. The decoupling of the state observer and the load torque observer was discussed. Daryabegi.E *et al.* [15] have presented intelligent emotional controller for brushless DC motor (BLDCM) drive, to control the motor speed accurately. Ansari.U [16] have proposed Genetic algorithm as a global optimizer to find the optimized PID gains for position control of BLDCM. Yu.G.R *et al.* [4] have proposed a novel optimal PID control design. The PID tuning algorithm was applied to the speed control of motor. Rahman, [17] has proposed on-line self-tuning artificial neural-network (ANN)-based speed control scheme of a permanent magnet (PM) dc motor was introduced, rather than using fixed weights and biases of the ANN and experimentally verified in real time environment using a digital signal processor (DSP) controller board. ANN-based scheme are compared with a proportional-integral (PI)-controller-based PM dc motor drive system under different operating conditions. The comparative results show that the ANN-based speed control scheme is robust, accurate, and insensitive to parameter variations and load disturbances.. Zhang.Y *et al.* [18] have presented an adaptive neural PID controller for torque control of hybrid electric vehicle. Zheng.L *et al.* [19] have proposed a fault diagnostics and fault tolerant control neural system for controller of brushless direct current motor and showed the effectiveness of the proposed method with scaling location of the fault and the time of occurrence, and eliminating the noise and offering

high robustness. Mirtalaei. S.M.M. *et al.* [18] have proposed a novel sensor less control method for brushless DC motors over wide speed range using two similar fuzzy logic based neural network observers, the fuzzy logic observers has been replaced by two neural network ones with the same performance which has been reported easier to implement.

### **1.3 OBJECTIVE OF THESIS WORK**

The objective of thesis work is summarised as follows:

1. Simulation of Brushless dc PM synchronous motor in Matlab/Simulink software.
2. Speed control of BLDC synchronous motor by applying conventional PI & PID controller in BLDCM simulink model.
3. Tuning of PI & PID controllers using Zeigler Nicholas (Z&N) rule
4. Designing of Artificial Neural Network controller for adaptive tuning of PID controller.
5. Speed control of BLDCM by ANN based PID controller at different speeds and torques by applying ANN based PID tuned controller in BLDCM simulink model.
6. Comparison between results obtained for speed control of BLDC synchronous motor by PI, PID & ANN based PID tuned controller.

### **1.4 ORGANIZATION OF THESIS**

The thesis is organised into six chapters. The organisation of chapters is as follows:

Chapter 1 summarizes the overview of problem, brief literature review, scope of work and organisation of thesis.

Chapter 2 highlights the introduction of Brushless dc permanent magnet synchronous motor (BLDCM) and its modelling.

In chapter 3 PI & PID controllers and there tuning method are explored.

Chapter 4 introduces the Artificial Neural network (ANN), its multilayer feed forward structure and learning algorithm.

Chapter 5 explores the simulation model of BLDCM with PI, PID, ANN based PID controller and also describes the ANN model used for automatic tuning of PID controller. Comparison between results obtained for speed control of BLDC synchronous motor by PI, PID & ANN based PID tuned controller is done.

Chapter 6 presents the conclusions drawn and also outlines the possible avenues for future work.

## CHAPTER 2

### BRUSHLESS DC MOTOR (BLDCM)

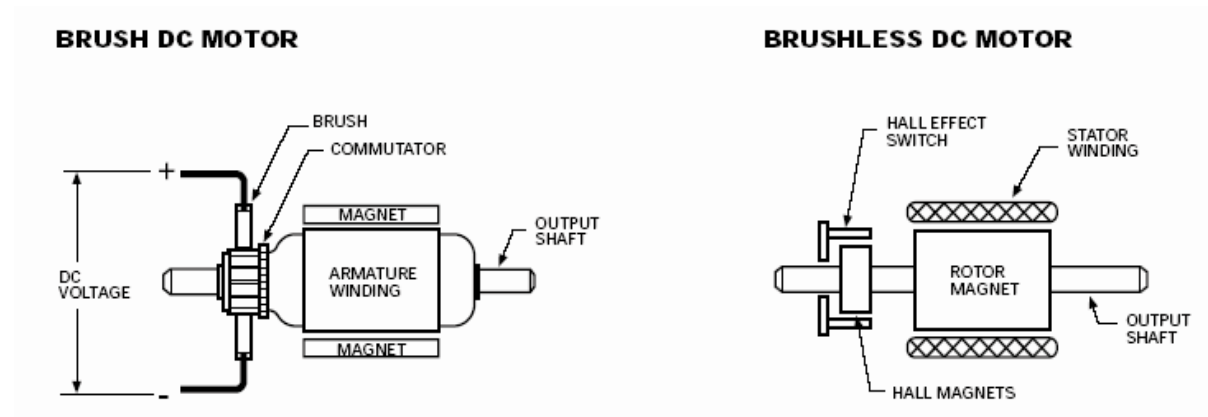
---

#### 2.1 INTRODUCTION

Permanent magnet synchronous (PMSM) motors are now commonly known as permanent magnet ac motors (PMAC). They are classified based on the nature of voltage induced in the stator as sinusoidally excited (induced voltage has a sinusoidal waveform) known as sinusoidal PMSM and trapezoidally excited (induced voltage has a trapezoidal waveform) known as trapezoidal PMSM. A sinusoidal PMSM motor has distributed winding in the stator. It employs rotor geometries such as inset or interior. Rotor poles are so shaped that the voltage induced in the stator phase has a sinusoidal waveform. The stator of trapezoidal PMSM motor has concentrated windings and rotor with a wide pole arc. The voltage induced in the stator phase has a trapezoidal waveform. It employs rotor geometries such as surface magnets. The speed of PMSM motors is controlled by feeding them from variable frequency voltage/currents. They are operated in self-controlled mode. Rotor position sensors are employed for operation in self-control mode. Alternatively induced voltage can be used to achieve self-control. Different inverter/converter circuits for PMSM motors are used. The MOSFET is used for low voltage and low power applications and IGBT for others. The self-controlled variable frequency drives employing a trapezoidal PMSM motors are called brushless dc motors (BLDCM) [21].

There are many similarities between BLDC permanent magnet synchronous motor and a dc motor. Like a dc motor, voltage induced is proportional to speed, torque is proportional to armature current and stator and rotor fields remain stationary with respect to each other. However, it does not have brushes and associated disadvantages that is why they are called brushless dc motors as shown in fig. 2.1. BLDCM are also called electronically commutated dc motors, because the inverter here performs the same function as the brushes and commutator in a dc motor, i.e. to shift currents between armature conductors to keep the stator and rotor fields stationary (and in quadrature) with respect to each other. BLDCM provide high efficiency, reliability, ruggedness and high precision of control when compared to conventional motors. It has the best torque vs. weight or efficiency characteristics. BLDCM finds applications in turn table drives in record players, tape drive for video recorders, spindle drives in hard disk drives for computers, and low cost and low power drives in computer

peripherals, instruments and control systems.



**Fig.2.1: Conventional dc motor & BLDCM.**

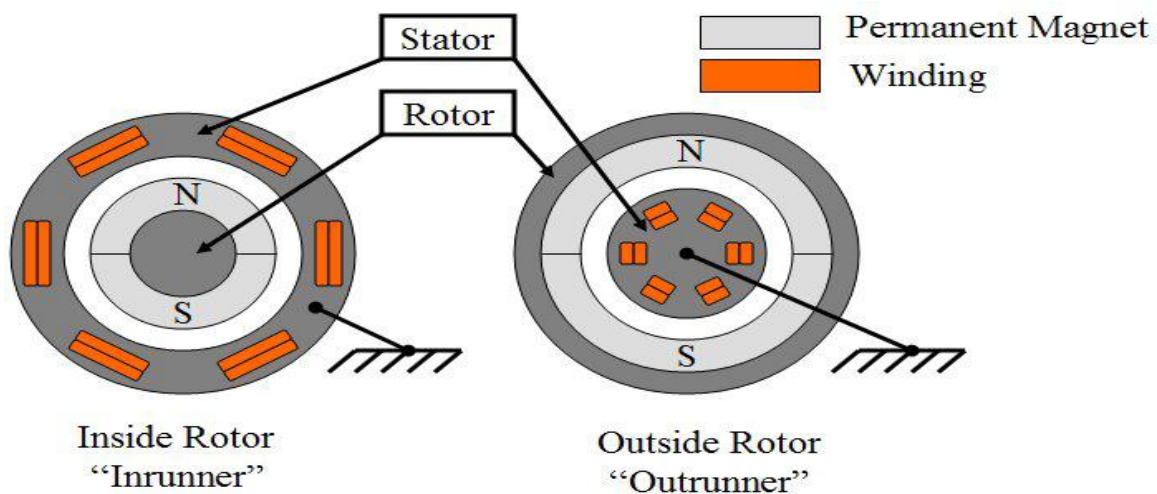
They also have applications in the fields of aerospace, e.g. gyroscope motors, and biomedical like cryogenic coolers and artificial heart pumps. They are also used for driving cooling fans for electronic circuits and heat sinks. They are used in military, grinding, aircraft, automotive applications, communications equipment etc.

## 2.2 TYPES OF BRUSHLESS DC PERMANENT MAGNET MOTORS (BLDCM)

There are two types of brushless PM motors called the in-runner and out-runner. The in-runner motor has permanent magnets located on the inside of the stationary electromagnets. An out-runner motor has the permanent magnets located on the outside as shown in Fig.2.2.

### 1) In-runner motors:

In-runner motors are good when high speeds are needed. They are more efficient than out runner motors for high speeds. However, due to the gearbox, it makes the motor more susceptible to failure of parts. The characteristics of in-runner motors are high rpm, low torque, requires gearbox & noisy running operation.



**Fig.2.2: In-runner & Out-runner BLDCM**

## 2) Out-runner motors:

Out-runner motors have slower speed but have higher output torque. They are easier to use since a gearbox is not required and run very quiet. The characteristics of out-runner motors are low RPM, high torque, & quiet running operation [1].

## 2.3 COMPONENTS OF BLDC PERMANENT MAGNET MOTOR

Major components of brushless dc PM motor are as shown in Fig.2.3

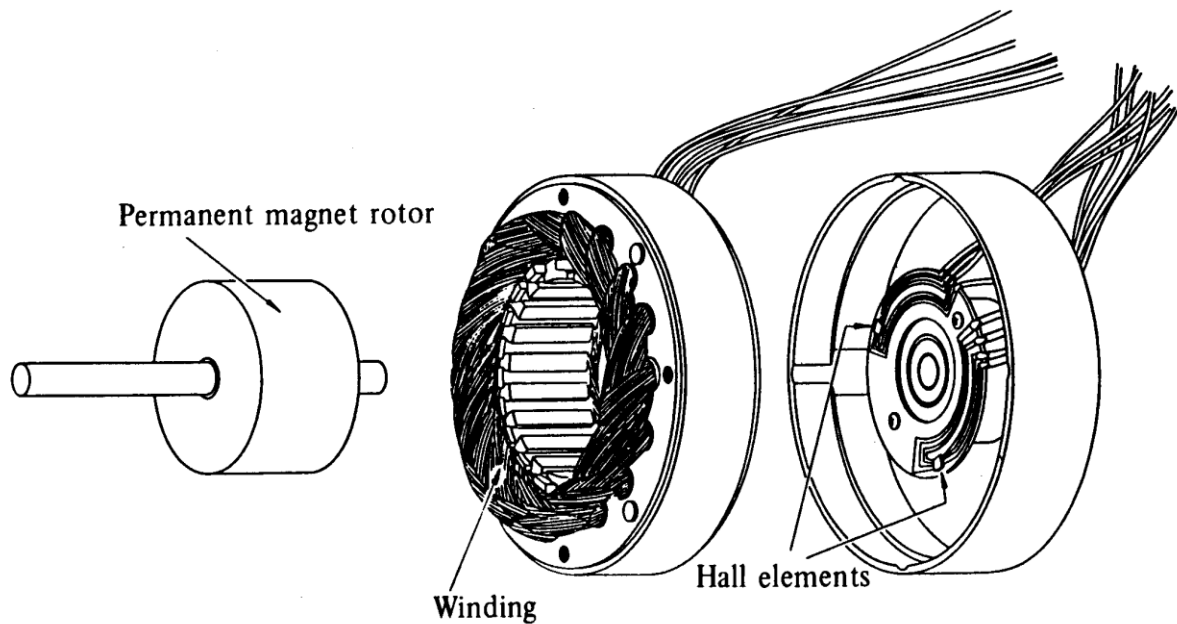


Fig.2.3: Cross sectional view of a BLDCM

### 2.3.1 ROTOR

The rotor of a typical brushless dc motor consists of the shaft, the hub (on which the magnets are glued on) the magnets and the bearings. Fig.2.4 of a four pole and eight pole BLDCM rotors: the higher the number of poles, the higher the torque but the lower the speed [2].



Fig.2.4: Four pole and eight pole BLDCM rotor

The rotor is made up of permanent magnets and consists of alternate north and south poles. Ferrite magnets are traditionally used to make permanent magnets. Rare earth alloy magnets are gaining popularity due to their high magnetic density per volume. An alloy of neodymium, ferrite and boron has been used to make permanent magnets [3].

### 2.3.2 STATOR

The BLDCM stator starts out with steel laminations (lam) which are stamped out of low loss electrical steel with a high quality stamping die as shown in Fig.2.5. The lam design and manufacturing of it is crucial to the performance of the BLDCM. The stamping die is a major investment and the type used (low volume/low cost/high maintenance high speed steel die versus high volume, higher cost/lower maintenance carbide die) used depends on the volume of the job.

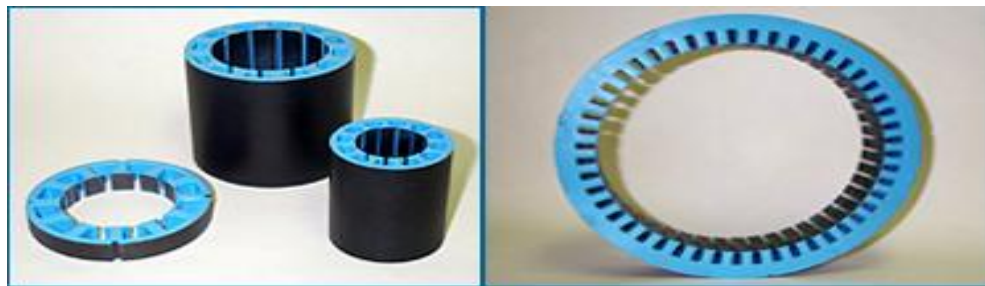


Fig.2.5: BLDCM stator

### 2.3.3 HALL SENSORS

The commutation of a BLDCM is controlled electronically. To rotate the BLDCM, the stator windings should be energized in a sequence. It is important to know the rotor position in order to understand which winding will be energized following the energizing sequence.

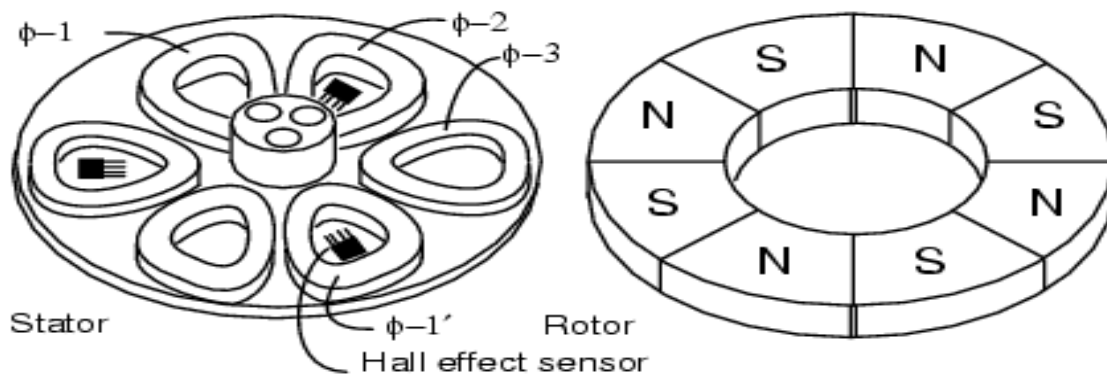


Fig.2.6: Stator of BLDCM with Hall Effect Sensors.

Rotor position is sensed using Hall Effect sensors embedded into the stator. Most BLDCM have three Hall sensors embedded into the stator on the non-driving end of the motor as shown

in Fig.2.6. Whenever the rotor magnetic poles pass near the Hall sensors, they give a high or low signal, indicating the N or S pole is passing near the sensors. Based on the combination of these three Hall sensor signals, the exact sequence of commutation can be determined.

Fig.2.6. shows a transverse section of a BLDCM with a rotor that has alternate N and S permanent magnets. Hall sensors are embedded into the stationary part of the motor. Embedding the Hall sensors into the stator is a complex process because any misalignment in these Hall sensors, with respect to the rotor magnets, will generate an error in determination of the rotor position. To simplify the process of mounting the Hall sensors onto the stator, some motors may have the Hall sensor magnets on the rotor, in addition to the main rotor magnets. These are a scaled down replica version of the rotor. Therefore, whenever the rotor rotates, the Hall sensor magnets give the same effect as the main magnets. The Hall sensors are normally mounted on a PC board and fixed to the enclosure cap on the non-driving end. This enables users to adjust the complete assembly of Hall sensors, to align with the rotor magnets, in order to achieve the best performance.

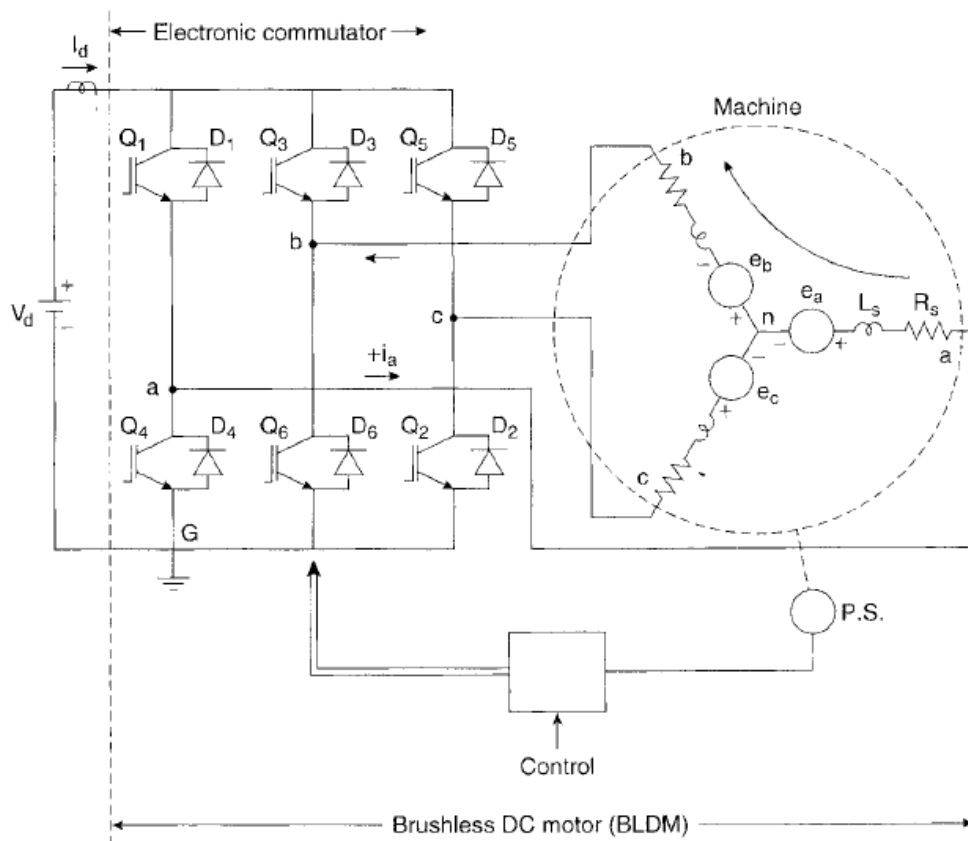
Based on the physical position of the Hall sensors, there are two versions of output. The Hall sensors may be at  $60^\circ$  or  $120^\circ$  phase shift to each other. Based on this, the motor manufacturer defines the commutation sequence, which should be followed when controlling the motor.

## **2.4 MOTOR OPERATION**

BLDCM is basically a non-salient-pole surface PM machine with a full pitch concentrated winding that generates three-phase trapezoidal counter emf waves at the machine terminal. The major reason for the popularity of these machines over their counterparts is control simplicity. To initiate the onset and commutation of current in the phase of a machine, the beginning and end of the constant portion of the induced emf have to be tracked. That amount to only six discrete positions for a three phase machine in each electrical cycle. These signals could easily be generated with three Hall sensors displaced from each other by 120 electrical degrees. The Hall sensors are mounted facing a small magnet wheel fixed to the rotor and having the same number of poles as the rotor of the BLDCM. Such an arrangement tracks the absolute position of the rotor magnets and hence the shape and position of the induced emfs in all the machine phases. As compare to PMSM, which requires continuous and instantaneous absolute rotor position, the BLDCM position-feedback requirement is much simpler: it requires only six discrete absolute positions for a three phase machine, resulting in a major cost saving in the feedback sensor. Further, the control involves significant vector operations in PMSM drive, whereas such operations are

not required for operation of the BLDCM drive [22].

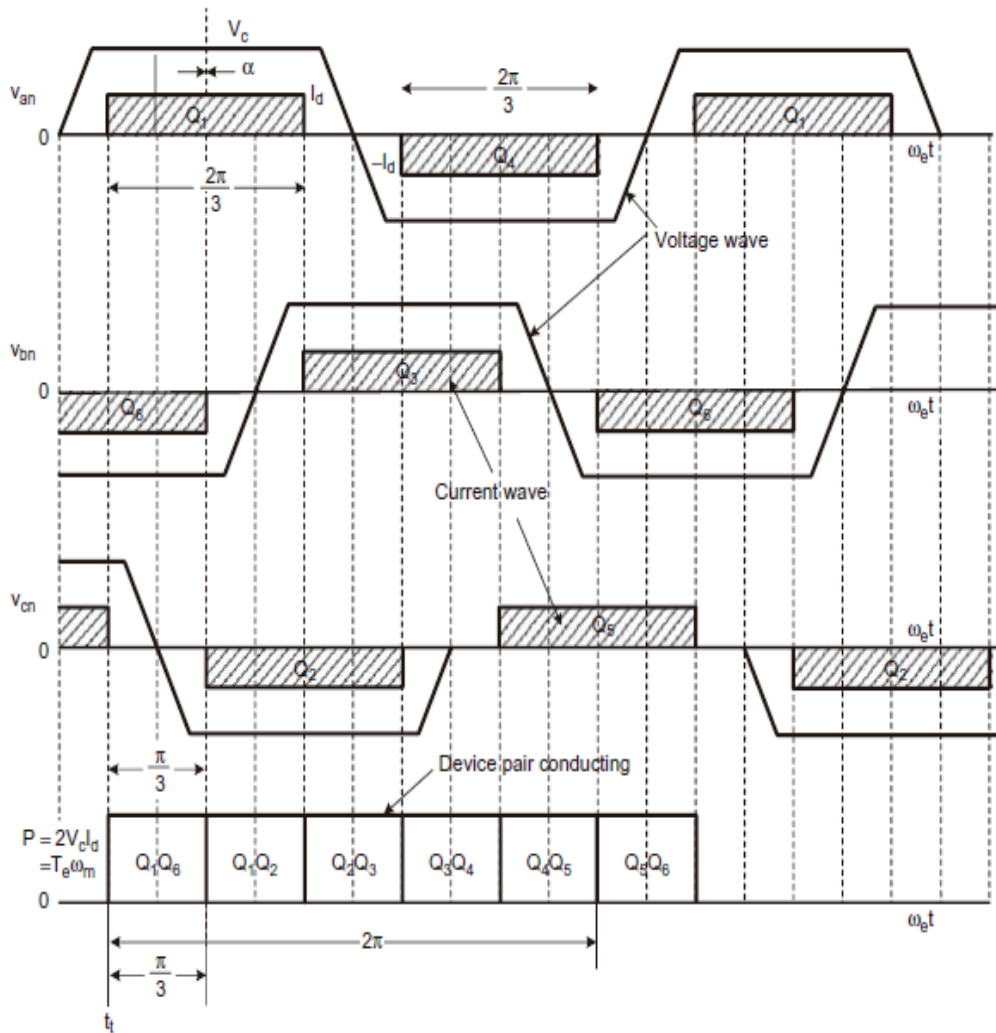
In the brushless dc motor, the polarity reversal is performed by power MOSFETS, which must be switched in synchronism with the rotor position. The stator is normally 3-phase star connected. Each commutation sequence has one of the windings energized to positive power (current entering into the winding) and the second winding energized to negative power (current exits the winding) and third winding non-energized. Torque is produced by the interaction of the magnetic field produced by the stator windings and the permanent magnets.



**Fig.2.7: Self-control Mode of BLDCM**

Fig.2.7 shows a trapezoidal PM synchronous machine in self-control mode with absolute position sensor and voltage-fed inverter. The whole configuration can be described as a brushless dc motor (BLDM), where the inverter acts as an electronic commutator, which replaces the mechanical commutator of a traditional dc machine. The drive is also called an electronically commutated motor (ECM). In fact, any type of synchronous machine with self-control can be defined as a BLDCM. However, a trapezoidal PMSM machine with self-control, as shown in this figure, most closely represents a BLDCM. The inverter in the

BLDCM is controlled by the simple 120° switch-on mode of the devices as explained in this Fig.2.8 [30].



**Fig.2.8: Stator phase EMF and current waves showing the inverter conducting switches**

The inverter switches are controlled in such a way that a 120° pulse of input current  $I_d$  is placed symmetrically at the centre of each phase voltage wave as shown. Because the phase position of the counter emf waves is related to the rotor position, generation of these pulses should be straightforward. Note that at any instant, two switches are on: one in the upper group and another in the lower group. The switching pattern changes every 60° as shown in the lower part of Fig.2.8. The input  $V_d$  or  $I_d$  is applied simultaneously to two phases in series, and the power supplied to the machine ( $P = 2V_d I_d$ ) is ideally constant as shown in the fourth waveform of Fig.2.8. With constant power, the developed torque is constant. The supply can also be variable voltage or current source. Although it is a PMAC drive with an inverter, the unit looks like a dc motor at the input through the electronic commutator [23].

## 25 SPEED-TORQUE CHARACTERISTICS

Fig.2.9 shows typical torque/speed characteristics. There are two torque parameters used to define a BLDC motor, peak torque ( $T_p$ ) and rated torque ( $T_r$ ). During continuous operations, the motor can be loaded up to the rated torque. As discussed earlier, in a BLDC motor, the torque remains constant for a speed range up to the rated speed. The motor can be run up to the maximum speed, which can be up to 150% of the rated speed, but the torque starts dropping. Applications that have frequent starts and stops and frequent reversals of rotation with load on the motor, demand more torque than the rated torque. This requirement comes for a brief period, especially when the motor starts from a standstill and during acceleration. During this period, extra torque is required to overcome the inertia of the load and the rotor itself. The motor can deliver a higher torque, maximum up to peak torque, as long as it follows the speed torque curve [3].

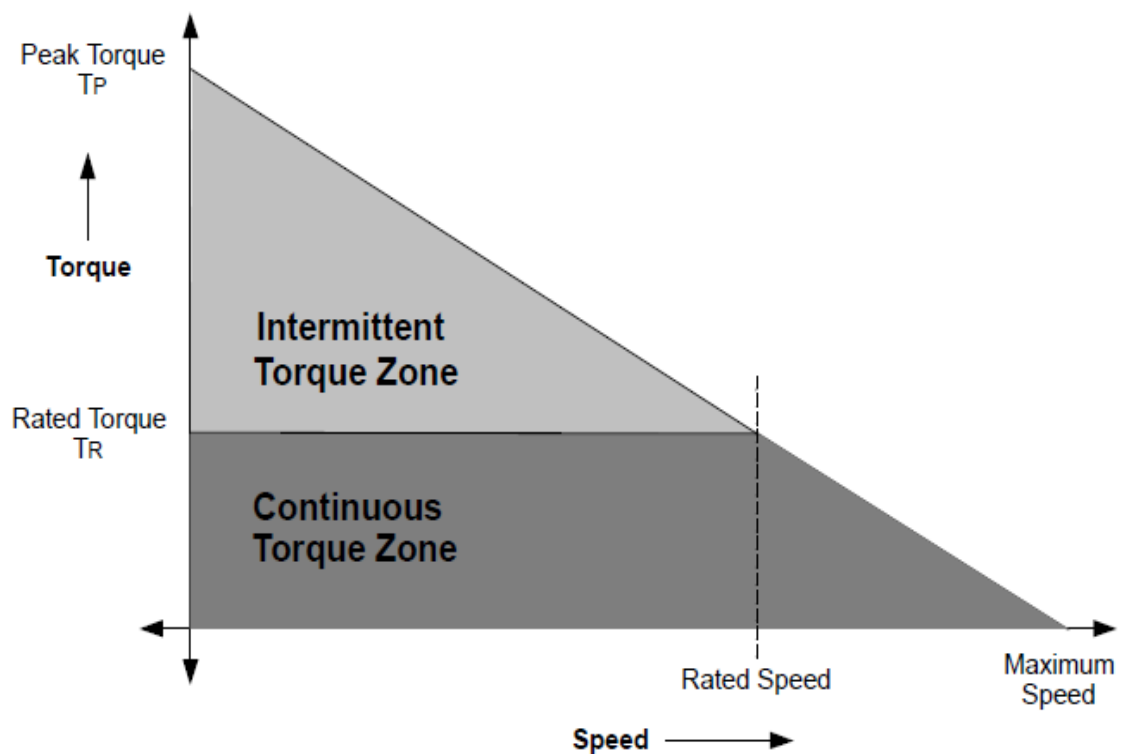


Fig.2.9: Torque/Speed characteristics of BLDCM

## 2.6 MODELLING OF BLDC PERMANENT MAGNET MOTOR

The flux distribution in a BLDCM is trapezoidal. The derivation BLDCM model is based on the assumptions that the induced currents in the rotor due to stator harmonic fields are neglected and that iron and stray losses are also neglected. Damper windings are not usually a part of the BLDCM. Damping is provided by the inverter control. The motor is considered to have three phases, even though the derivation procedure is valid for any number of phases [23].

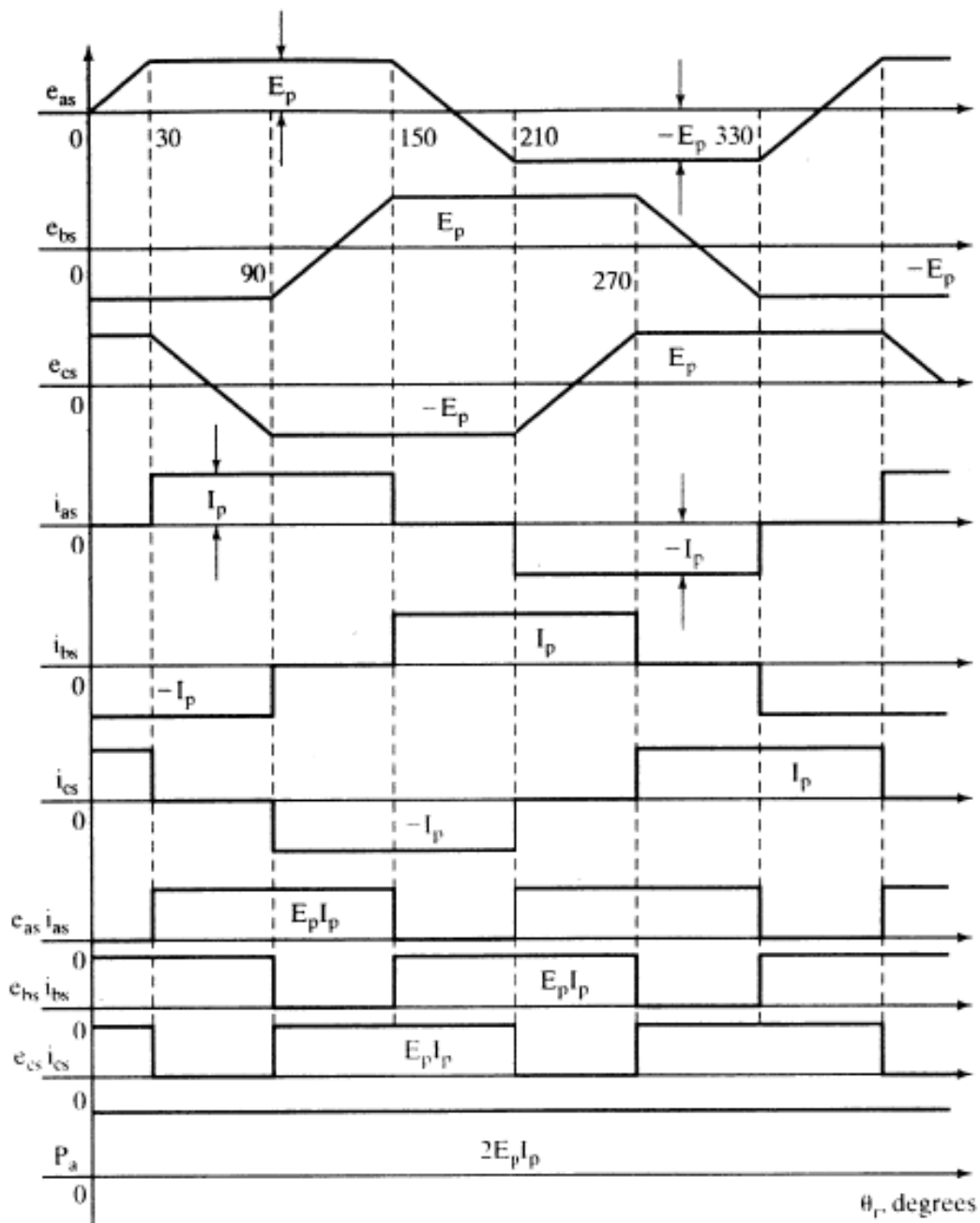


Fig.2.10: Permanent magnet (PM) BLDCM waveforms

The coupled circuit equations of the stator windings in terms of motor electrical constants are

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + p \begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_{as} \\ e_{bs} \\ e_{cs} \end{bmatrix} \quad (2.1)$$

Where  $R_s$  is the stator resistance per phase, assumed to be equal for all three phases.

The induced emfs  $e_{as}$ ,  $e_{bs}$ , and  $e_{cs}$  are all assumed to be trapezoidal, as shown in Fig.2.10, where  $E_p$  is the peak value, derived as

$$E_p = (Blv)N = N (Blr\omega_m) = N\phi_a\omega_m = \lambda_p\omega_m \quad (2.2)$$

Where N is the number of conductors in series per phase, v is the velocity, l is the length of the conductor, r is the radius of the rotor bore,  $\omega_m$  is the angular velocity, & B is the flux density of the field in which the conductors are placed. This flux density is solely to the rotor magnets. The product  $(Blr)$ , denoted as  $\phi_a$ , has the dimensions of flux and is directly proportional to the air gap flux,  $\phi_g$ :

$$\phi_a = Blr = \frac{1}{\pi} B\pi lr = \frac{1}{\pi} \phi_g \quad (2.3)$$

Note that the product of flux and number of conductors in series has the dimension of flux linkages and is denoted by  $\lambda_p$ . Since this is proportional to phase a flux linkages by a factor of  $1/\pi$ , it is hereafter referred to as modified flux linkages.

If there is no change in the rotor reluctance with angle because of non-salient rotor, and assuming three symmetric phases, hence following are obtained:

$$L_{aa} = L_{bb} = L_{cc} = L; \& L_{ab} = L_{ba} = L_{ac} = L_{ca} = L_{bc} = L_{cb} = M(H) \quad (2.4)$$

Substituting Eqns. (2.3) & (2.4) in Eqn. (2.1) gives the BLDCM model as

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = R_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \begin{bmatrix} L & M & M \\ M & L & M \\ M & M & L \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_{as} \\ e_{bs} \\ e_{cs} \end{bmatrix} \quad (2.5)$$

The stator phase currents are constrained to be balance, i.e.  $i_{as} + i_{bs} + i_{cs} = 0$ , which leads to the simplification of the inductance matrix in the model as

$$\begin{bmatrix} V_{as} \\ V_{bs} \\ V_{cs} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \begin{bmatrix} (L-M) & 0 & 0 \\ 0 & (L-M) & 0 \\ 0 & 0 & (L-M) \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_{as} \\ e_{bs} \\ e_{cs} \end{bmatrix} \quad (2.6)$$

The electromagnetic torque is given by

$$T_e = [e_{as} i_{as} + e_{bs} i_{bs} + e_{cs} i_{cs}] \frac{1}{\omega_m} \quad (N.m) \quad (2.7)$$

The instantaneous induced emfs can be written from Fig.2.10 and eqn (2.2) as

$$e_{as} = f_{as}(\Theta_r) \lambda_p \omega_m \quad (2.8)$$

$$e_{bs} = f_{bs}(\Theta_r) \lambda_p \omega_m \quad (2.9)$$

$$e_{cs} = f_{cs}(\Theta_r) \lambda_p \omega_m \quad (2.10)$$

Where the functions  $f_{as}(\Theta_r)$ ,  $f_{bs}(\Theta_r)$ , &  $f_{cs}(\Theta_r)$ , have the same shape as  $e_{as}$ ,  $e_{bs}$ , &  $e_{cs}$ , with a maximum magnitude of  $\pm 1$  (per unit basis). The induced emfs do not have sharp corners, as is shown in trapezoidal functions, but rounded edges. The emfs are the result of the flux-linkages derivatives, and the flux linkages are continuous functions. Fringing also makes the flux density functions smooth with no abrupt edges [23]. The electromagnetic torque then is

$$T_e = \lambda_p [f_{as}(\Theta_r) i_{as} + f_{bs}(\Theta_r) i_{bs} + f_{cs}(\Theta_r) i_{cs}] \quad (N.m) \quad (2.11)$$

It is significant to observe that the phase-voltage equation is identical to the armature-voltage equation of a dc machine. That is one of the reasons for naming this machine PM brushless dc machine. The equation of motion for a simple system with inertia J, friction coefficient B, and load torque  $T_l$  is

$$J \frac{d\omega_m}{dt} + B \omega_m = (T_e - T_l) \quad (2.12)$$

And electrical rotor speed and position are related by

$$\frac{d\Theta_r}{dt} = \frac{P}{2} \omega_m \quad (2.13)$$

Combining all the relevant equations, the system in state space form is

$$\dot{x} = Ax + Bu \quad (2.14)$$

Where

$$x = [i_{as} \ i_{bs} \ i_{cs} \ \omega_m \ \Theta_r]^t \quad (2.15)$$

$$A = \begin{bmatrix} -\frac{R_s}{L_1} & 0 & 0 & -\frac{\lambda_p}{L_1} f_{as}(\Theta_r) & 0 \\ 0 & -\frac{R_s}{L_1} & 0 & -\frac{\lambda_p}{L_1} f_{bs}(\Theta_r) & 0 \\ 0 & 0 & -\frac{R_s}{L_1} & -\frac{\lambda_p}{L_1} f_{cs}(\Theta_r) & 0 \\ -\frac{\lambda_p}{L_1} f_{as}(\Theta_r) & -\frac{\lambda_p}{L_1} f_{bs}(\Theta_r) & -\frac{\lambda_p}{L_1} f_{cs}(\Theta_r) & -\frac{B}{J} & 0 \\ 0 & 0 & 0 & \frac{P}{2} & 0 \end{bmatrix} \quad (2.16)$$

$$B = \begin{bmatrix} \frac{1}{L_1} & 0 & 0 & 0 \\ 0 & \frac{1}{L_1} & 0 & 0 \\ 0 & 0 & \frac{1}{L_1} & 0 \\ 0 & 0 & 0 & -\frac{1}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.17)$$

$$L_1 = L - M \quad (2.18)$$

$$u = [V_{as} \ V_{bs} \ V_{cs} \ T_l]^t \quad (2.19)$$

The state variable  $\Theta_r$ , rotor position/speed as output,

$$\frac{d\Theta_r}{dt} = \frac{P}{2} \omega_m$$

It is required so as to have the functions  $f_{as}(\Theta_r)$ ,  $f_{bs}(\Theta_r)$ , &  $f_{cs}(\Theta_r)$  in three phase voltages as input.

$$e_{as} = f_{as}(\Theta_r) \lambda_p \omega_m$$

$$e_{bs} = f_{bs}(\Theta_r) \lambda_p \omega_m$$

$$e_{cs} = f_{cs}(\Theta_r) \lambda_p \omega_m$$

Hence this represents the modelling of BLDCM which is implemented.

## 2.7 CHARACTERISTIC FEATURES OF BRUSH LESS DC MOTOR

Some of the salient features of BLDC motors in comparison with other AC machines are listed as:

S. No.	BLDC Advantage	Underlying Reasons
1	Smaller motor	Modern permanent magnets and no losses in the rotor enable BLDCM to be smaller compared to both brush dc motors and induction ac motors.
2	More efficient motor	Permanent magnet in the rotor. Unlike ac induction motors, there are no core losses in the rotor.
3	Higher speeds	No brushes to limit speed, lower speed losses by design. BLDCM have been designed for speeds as high as 100,000 RPM. No longer problem of retention of magnets, in a rotor spinning at high speeds.
4	No maintenance	No brushes to replace, inspect or maintain
5	Faster response	Lower rotor inertia compared to a brushless motor or an induction motor
6	Lower RFI ( radio frequency interference)	No brushes
7	Linear speed-torque Characteristics	Internal shaft position feedback. Permanent magnet design with internal shaft position feedback gives BLDCM linear speed-torque characteristics when compared to “open loop” ac induction motors.
8	High starting torque	Internal shaft position feedback gives BLDCM higher starting and low speed torque when compared to “open loop” ac induction motors.
9	Adjustable speed	The commutation electronics can be used for speed control without added cost.
10	Better heat removal	The heat generating element is the stator which is on the outside of the motor as opposed to being in the inside as in the case of a brush dc motor
11	Much better controllability	Linear speed torque characteristics due to internal Shaft position feedback and permanent magnet design.

## CHAPTER 3

### ADAPTIVE CONTROLLERS

---

#### 3.1 INTRODUCTION

Some basic terminologies of control system are defined [23]:

**Controlled variable:** The controlled variable is the quantity or condition that is measured and controlled. Normally, the controlled variable is the output of the system. Control means measuring the value of the controlled variable of the system.

- 1) **Manipulated variable:** The manipulated variable is the quantity or condition that is varied by the controller so as to affect the value of the controlled variable. The manipulated variable is applied to the system to correct or limit deviation of the measured value from a desired value.
- 2) **Plants:** A plant is a piece of equipment, perhaps just a set of machine parts functioning together, the purpose of which is to perform a particular operation.
- 3) **Processes:** A process could be a natural, progressively continuing operation or development marked by series of gradual changes that succeed one another in a relatively fixed way or lead toward a particular result or end.
- 4) **Systems:** A system is a combination of components that act together and perform a certain objective. A system is not limited to physical ones. The concept of the system can be applied to abstract, dynamic phenomena such as those encountered in economics.
- 5) **Disturbances:** A disturbance is a signal that tends to adversely affect the value of the output of a system. If a disturbance is generated within the system, it is called internal, while an external disturbance is generated outside the system and is an input.
- 6) **Feedback control:** Feedback control refers to an operation that, in the presence of disturbances, tends to reduce the difference between the output of a system and some reference input and does so on the basis of this difference.

#### 3.2 ADAPTIVE CONTROL

Adaptive controllers are characterized by their ability to gather information about the parameters of a process during actual control and by their ability to make changes to their control laws accordingly based on the information collected. The adaptive controllers are categorised into two main classes: feed forward adaptive controllers and feedback adaptive controllers.

### 3.2.1 FEED FORWARD CONTROL SYSTEM

Feed forward controllers operate based on the fact that the changing properties of the plant which are collected by measurement of signals generated in the process. The feed forward adaptation system can be realized as an open loop system. There is no feedback of ‘inner’ closed-loop signals to adapt the controller parameters. The disturbance input is measured and the adaptive mechanism changes the parameters of the controller in such a way as to maintain good control performance. One advantage of feed forward adaptive control is that fast reaction to process changes can be achieved because the process behaviour could be anticipated and need not be identified with measurable process input and output signals. There are some disadvantages in this system. They are neglect of effects based on unmeasured signals or disturbances, unpredictable changes of the process behaviour and the amount of parameter storage that may be necessary to accommodate many operating conditions and the limitations to slow processes and parameter changes.

### 3.2.2 FEEDBACK CONTROL SYSTEM

A system that maintains a prescribed relationship between the output and the reference input by comparing them and using the difference as a means of control is called a feedback control system as shown in Fig.3.1.

**Closed-loop control system:** Feedback control systems are often referred to as closed-loop control systems. In practise, the term feedback control and closed-loop are using interchangeably. In a closed loop control system the actuating error signal, which is the difference between the input signal and the feedback signal, is fed to the controller so as to reduce the error and bring the output of the system to a desired value. The term closed-loop control always implies the use of feedback control action in order to reduce system error [23].

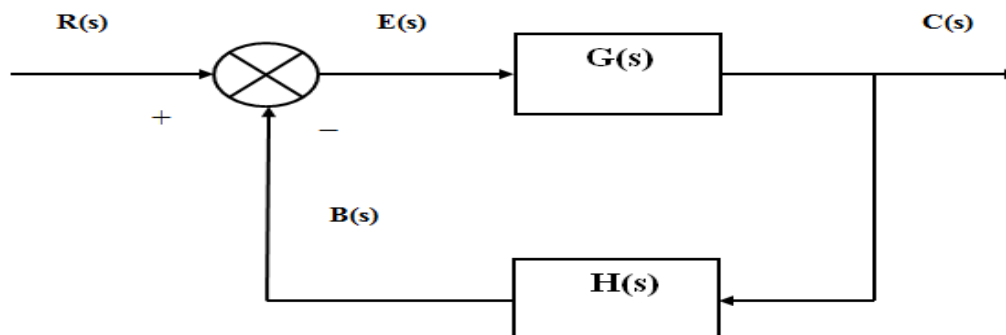


Fig.3.1: Feedback control system

Where,

$R(s)$  = Reference input signal,  $C(s)$  = Output signal,  $E(s)$  = Error Signal,  $G(s)$  = Forward gain,  $H(s)$  = Backward gain &  $B(s)$  = Feedback signal.

Error can be calculated by;

$$E(s) = R(s) - B(s) \quad (3.1)$$

$$\frac{C(s)}{G(s)} = R(s) - C(s).H(s) \quad (3.2)$$

$$C(s) = G(s).R(s) - G(s).H(s).C(s) \quad (3.3)$$

$$C(s)[1 + G(s).H(s)] = G(s).R(s) \quad (3.4)$$

$$\frac{C(s)}{R(s)} = \left[ \frac{G(s)}{1 + G(s).H(s)} \right] \quad (3.5)$$

### 3.2.3 OPEN-LOOP CONTROL SYSTEM

Those systems in which the output has no effect on the control action are called open-loop control systems. In other words, in an open-loop control system the output is neither measured nor feedback for comparison with the input. In an open-loop control system the output is not compared with the reference input. Thus, to each reference input there corresponds a fixed operating condition; as a result, the accuracy of system depends on calibration. In the presence of disturbances, an open-loop control system will not perform the desired task. Open-loop control system can be used, in practice, only if the relationship between the input and output is known and if there are neither internal nor external disturbances. Clearly, such systems are not feedback control systems.

### 3.3 TRANSIENT-RESPONSE SPECIFICATIONS OF SECOND ORDER SYSTEM

The systems that stores energy cannot respond instantaneously, they exhibit a transient response when they are subjected to inputs or disturbances. Consequently, the transient response characteristics constitute one of the most important factors in system design. In many practical cases, the desired performance characteristics of control systems can be given in terms of transient-response specifications. Frequently, such performance characteristics are specified in terms of the transient response to unit-step input, since such an input is easy to generate and is sufficiently drastic. (If the response of a linear system to a step input is known, it is mathematically possible to compute the system's response to any input). The transient response of a system to a unit step-input depends on initial conditions. For convenience in comparing the transient responses of various systems, it is common practice

to use standard initial conditions: The system is at rest initially, with its output and all time derivatives there of zero. Then the response characteristics can be easily compared.

The transient response of a practical control system often exhibits damped oscillations before reaching a steady state. In specifying the transient-response characteristics of a control system to a unit step input are shown in Fig.3.2.

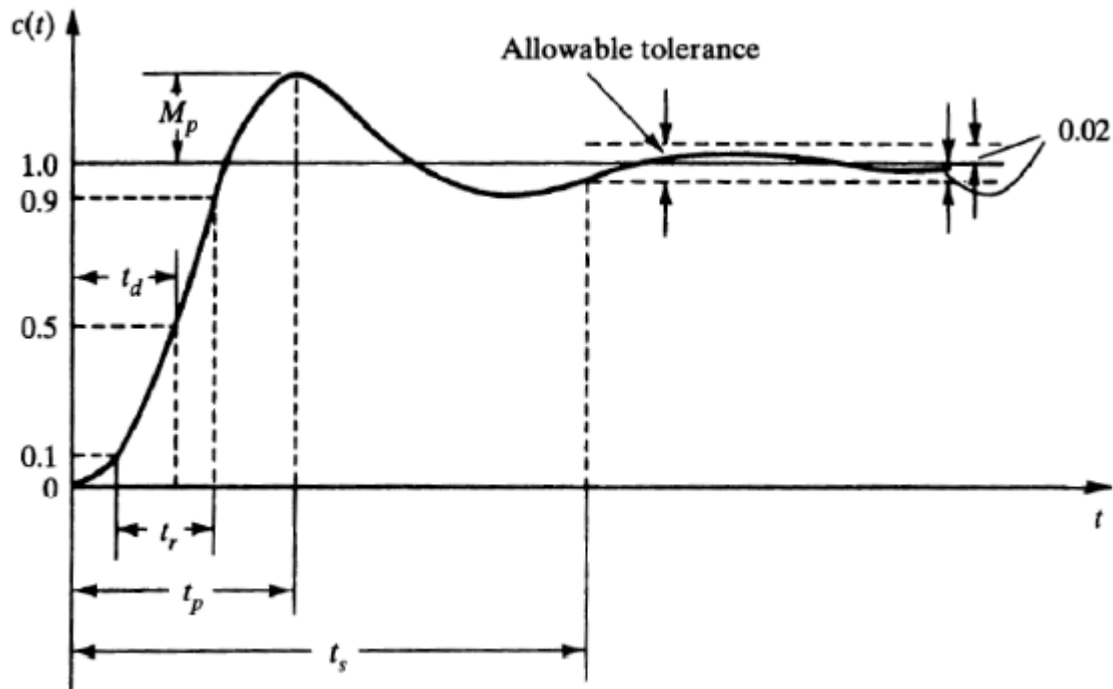


Fig.3.2: Transient response specifications of second order system.

Where,  $T_d$  = Delay time,  $T_r$  = Rise time,  $T_p$  = Peak time,  $M_p$  = Maximum Percent Overshoot,  $T_s$  = Settling time

- 1) **Delay time ( $T_d$ ):** The delay time is the time needed for the response to reach half of its final value the very first time.
- 2) **Rise time ( $T_r$ ):** The rise time is the time required for the response to rise from 10 % to 90 % of its final value. For underdamped second order systems, the 0 % to 100 % rise time is normally used. For overdamped systems, the 10 % to 90 % rise time is common.
- 3) **Peak time ( $T_p$ ):** The peak time is the time required for the response to reach the first peak of the overshoot.

**4) Maximum Percent Overshoot ( $M_p$ ):** The Maximum percent overshoot is the maximum peak value of the response curves actual value, measured from desired value.

$$\% M_p = \frac{\text{Actual value} - \text{Desired value}}{\text{Desired value}} \times 100$$

**5) Settling time ( $T_s$ ):** The settling time is the time required for the response curve to reach and stay within the 2 % of the final value. The settling time is the largest time constant of the system.

### **3.4 CONTROL ACTIONS**

An automatic control system is used to maintain its output within desirable limits by means of control action. Any deviation of the output from the reference input is detected by an error detector. The error thus detected is used as an actuating signal for control action through the controller. The proportional control action and types of control actions used for improving transient and steady state response of a control system.

#### **3.4.1 PROPORTIONAL CONTROL**

In proportional control the actuating signal for the control action in a control system is proportional to the error signal. The error signal being the difference between the reference input signal and the feedback signal obtained from the output. For the system as shown in Fig.3.3, the actuating signal is proportional to the error signal; therefore, the system is called proportional control system. It is desirable that the control system be underdamped for the point of view of quick response. An underdamped control system exhibits exponentially decaying oscillations in the output time response during the transient period. The sluggish overdamped response of a control system can be made faster by increasing forward path gain of the system. The increase in the forward path gain reduces the steady state error, but at the same time maximum overshoot is increased. For the satisfactory performance of the control system a convenient adjustment has to be made between the maximum overshoot and steady state error. Without sacrificing the steady state accuracy, the maximum overshoot can be reduced to some extent by modifying the actuating signal [23].

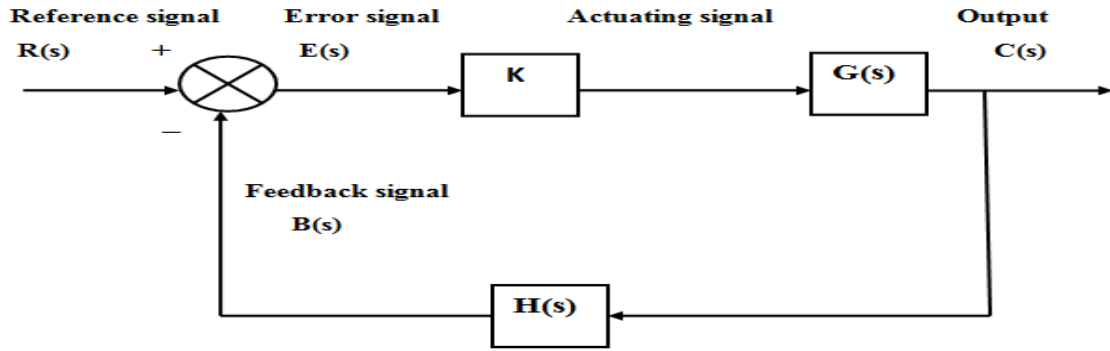


Fig.3.3: Proportional control action

$$P \propto e \quad (3.6)$$

$$P = K_p e \quad (3.7)$$

Where,  $K_p$  = Proportional gain

By taking Laplace transform, we get

$$P(s) = K_p E(s) \quad (3.8)$$

### 3.4.2 INTEGRAL CONTROL

Integral control is also known as proportional mode because the rate of change of the controller output can be reset by changing the value of integral gain ( $K_i$ ) at any given time  $t$ . Integral controller introduces oscillations in the response. It is capable of eliminating any type of error in system. The disadvantage of this mode is its response to errors is very slow or sluggish [23].

$$P \propto \int e dt \quad (3.9)$$

$$P = K_i \int e dt \quad (3.10)$$

Where,  $K_i$  = Integral gain, so reset time  $T_i = 1/K_i$ .

$$P = \frac{1}{T_i} \int e dt \quad (3.11)$$

By taking Laplace transform, we get

$$P(s) = \frac{1}{T_i s} E(s) \quad (3.12)$$

### 3.4.3 DERIVATIVE CONTROL

Derivative control is also known as anticipatory controller mode as it sends a control signal in anticipation of error. The disadvantage of this mode is that it cannot respond to certain types of errors.

$$P = \frac{de}{dt} \quad (3.13)$$

$$P = K_d \frac{de}{dt} \quad (3.14)$$

Where,  $K_d$  is derivative gain, rate time  $T_d = K_d$ .

$$P = T_d \frac{de}{dt} \quad (3.15)$$

By taking Laplace transform, we get

$$P(s) = T_d s E(s) \quad (3.16)$$

### 3.4.4 CHARACTERISTICS OF P, I, & D CONTROLLERS

A proportional controller ( $K_p$ ) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error. An integral control ( $K_i$ ) will have the effect of eliminating the steady-state error, but it may make the transient response worse. A derivative control ( $K_d$ ) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response [23]. Effects of each of controllers  $K_p$ ,  $K_i$ , &  $K_d$  on a closed-loop system are summarized in the table shown

**Table.3.1: Effects of each of controllers  $K_p$ ,  $K_i$ , &  $K_d$  on a closed-loop system**

Closed-Loop Response	Rise time	Overshoot	Settling time	Steady-State error
$K_p$	Decrease	Increase	Small change	Decrease
$K_i$	Decrease	Increase	Increase	Eliminate
$K_d$	Small change	Decrease	Decrease	Small change

Note that these correlations may not be exactly accurate, because  $K_p$ ,  $K_i$ , &  $K_d$  are dependent of each other. In fact, changing one of these variables can change the effect of the other two. For this reason, the table should only be used as a reference when you are determining the values for  $K_p$ ,  $K_i$ , &  $K_d$ .

### 3.4.5 PROPORTIONAL & INTEGRAL (PI) CONTROLLER

The PI controller is characterized by the transfer function

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (3.17)$$

Where  $K_p$  = Proportional gain,  $T_i$  = Integral time constant. The PI controller is a lag compensator. It possesses a zero at  $s = -1/T_i$  and a pole at  $s = 0$ . Thus, the characteristic of the PI controller is infinite gain at zero frequency. This improves the steady-state characteristics. However, addition of PI controller action in the system increases the type number of the compensated system by 1, and this causes the compensated system to be less stable or even makes the system unstable. Therefore, the values of  $K_p$  &  $T_i$  must be chosen carefully to ensure a proper transient response. By properly designing the PI controllers, it is possible to make the transient response to a step input exhibit relatively small or no overshoot [24]. The speed of response, however, becomes much slower. This is because the PI controller, being a low-pass filter, attenuates the high-frequency components of the signal.

### 3.4.6 PROPORTIONAL & DERIVATIVE (PD) CONTROLLER

The PD controller is simplified version of lead compensator. The PD controller is characterized by the transfer function

$$G_c(s) = K_p (1 + T_d s) \quad (3.18)$$

Where  $T_d$  = Derivative time constant. The value of  $K_p$  is usually determined to satisfy the steady-state requirement. The corner frequency  $1/T_d$  is chosen such that the phase lead occurs in the neighbourhood of the gain crossover frequency. Although the phase margin can be increased, the magnitude of the compensator continues to increase for the frequency region  $1/T_d < \omega$ . Thus PD controller is a high pass filter. Such a continued increase of the magnitude is undesirable, since it amplifies high-frequency noises that may be present in the system. Lead compensation can provide a sufficient phase lead, while the increase of the magnitude for the high-frequency region is very much smaller than that for PD control. Therefore lead compensation is preferred over PD control because the transfer function of the PD controller involves one zero, but no pole, it is not possible to electrically realize it by passive RLC elements only. Realization of the PD controllers using op amps, resistors, and capacitors is possible, but because PD controller is a high-pass filter, the differentiation process involved may cause serious noise problems in some cases. The PD control, as in the case of the lead compensator, improves the transient-response characteristics, improves system stability, and increases the system band width, which implies fast rise time.

### 3.4.7 PROPORTIONAL, INTEGRAL & DERIVATIVE (PID) CONTROLLER

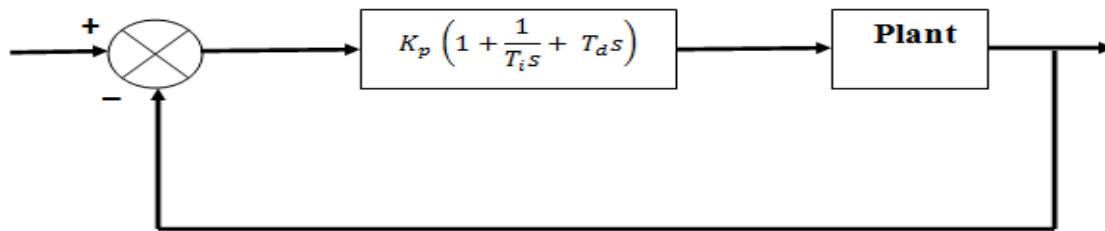
The PID controller is a combination of the PI & PD controllers. It is a lag-lead compensator. PI control action occurs in low-frequency region and PD control action occurs at the high

frequency region, therefore the PID control may be used when the system requires improvements in both transient and steady-state performances as shown in Fig3.4.

The PID controller is characterized by the transfer function

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (3.19)$$

More than half of the industrial controllers in use today utilize PID or modified PID control schemes. Most PID controllers are adjusted on-site, many different types of tuning rules have been proposed in the literature. Using these tuning rules, delicate and fine tuning of PID controllers can be made on-site. The use of PID controllers lies in their general applicability to most control systems. In particular, when the mathematical model of the plant is not known and therefore analytical design methods cannot be used, PID controls prove to be most useful. In the field of process control systems, it is well known that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although in many given situations they may not provide optimal control [23].



**Fig.3.4: PID control of a plant.**

### 3.5 TUNING RULES FOR CONTROLLERS

For the PID control of a plant as shown in Fig. 3.4, if a mathematical model of the plant can be derived, then it is possible to apply various design techniques for determining of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complicated that its mathematical model cannot be easily obtained, then an analytical approach to the design of a PID controller is not possible. Then we must resort to experimental approaches to the tuning of PID controllers. The process of selecting the controller parameters to meet given performance specifications is known as controller tuning. Ziegler and Nichols suggested rules for tuning PID controllers (meaning to set values  $K_p$ ,  $T_i$ , &  $T_d$ ) based on experimental step responses or based on the value of  $K_p$  that

result in marginal stability when only proportional control action is used. Ziegler-Nichols rules, which are briefly presented in the following, are useful when mathematical models of plants are not known. Such rules suggest a set of values of  $K_p$ ,  $T_i$ , and  $T_d$  that will give a stable operation of the system. However, the resulting system may exhibit a large maximum overshoot in the step response, which is unacceptable. In such a case we need series of fine tunings until an acceptable result is obtained. The Ziegler-Nichols tuning rules [25] gives an improved guess for the parameter values and provide a starting point for fine tuning, rather than giving the final settings for  $K_p$ ,  $T_i$ , &  $T_d$  in a single shot.

### 3.5.1 ZIEGLER-NICHOLS RULE FOR TUNING OF CONTROLLERS

Ziegler and Nichols proposed rules for determining values of the proportional gain  $K_p$ , integral time  $T_i$ , and derivative time  $T_d$  based on the transient response characteristics of a given plant. Such tuning of the PID controllers can be made by engineers on-site by experiments on the plant. There are two methods called Ziegler-Nichols tuning rules: the first method and the second method but here we are discussing First method only as we have applied this method only for tuning.

**First method:** In the first method, we obtain experimentally the response of the plant to a unit-step input, as shown in Fig.3.5.

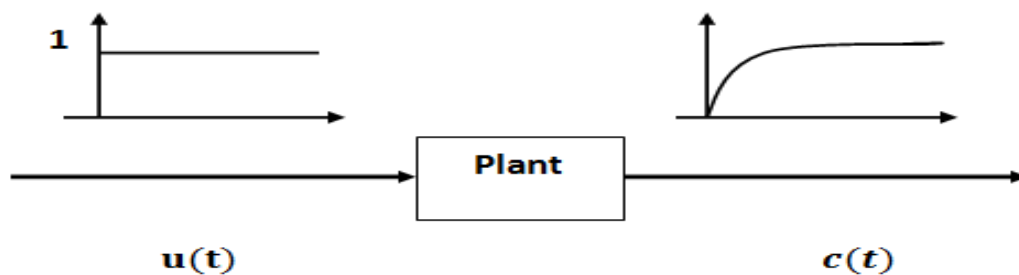
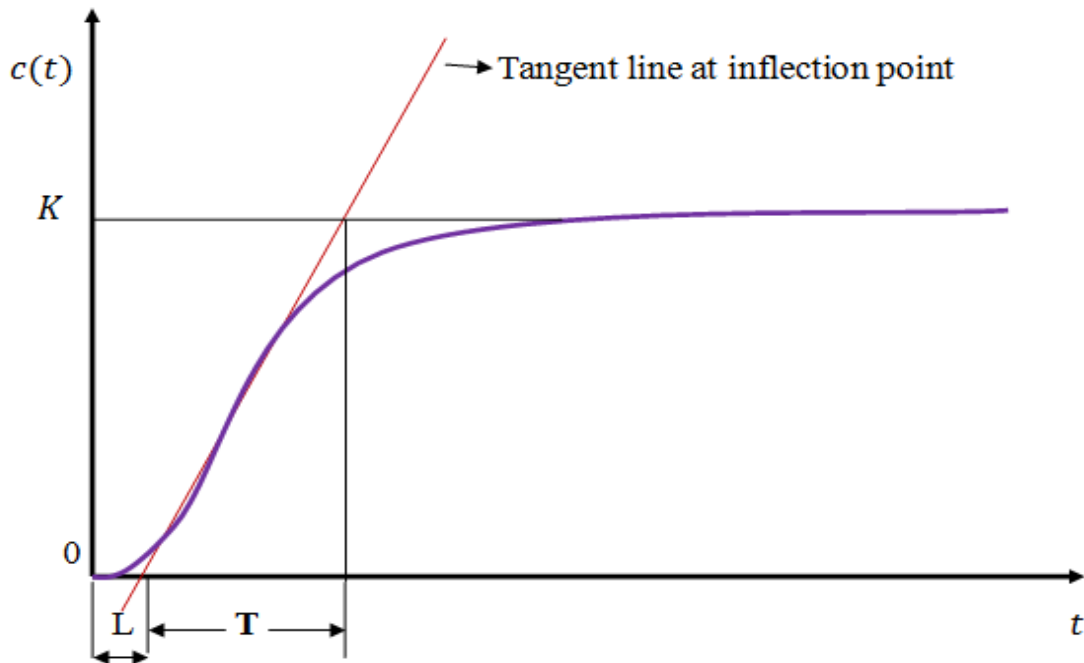


Fig.3.5: Unit-step response of a Plant

If the plant involves neither integrator nor dominant complex-conjugate poles, then such a unit step response curve may look S-shaped, as shown in Fig.3.6. This method applies if the response to a step input exhibits an S-shaped curve. Such step-response curves may be generated experimentally or from a dynamic simulation of the plant. The S-shaped curve may be characterized by two constants, delay time  $L$  and time constant  $T$ . The  $L$  &  $T$  are determined by drawing a tangent line at the inflection point of the S-shaped curve and

determining the intersections of the tangent line with the time axis and line  $c(t)$ , as shown in Fig.3.6.



**Fig.3.6: S-shaped response curve**

The transfer function  $C(s)/U(s)$  may then be approximated by a first-order system with a transport lag as follows:

$$\frac{C(s)}{U(s)} = \frac{K e^{-Ls}}{Ts + 1} \quad (3.20)$$

Ziegler & Nichols suggested setting the values of  $K_p$ ,  $T_i$ , &  $T_d$  according to the formula shown in the table 3.2 below:

**Table.3.2: Values of  $k_p$ ,  $T_i$ , &  $T_d$  obtained by Ziegler & Nichols Tuning**

Type of Controller	$K_p$	$T_i$	$T_d$
<b>P</b>	$T/L$	$\infty$	0
<b>PI</b>	$0.9 T/L$	$L/0.3$	0
<b>PID</b>	$1.2 T/L$	$2L$	$0.5L$

The PID controller tuned by the first method of Ziegler-Nichols rules gives:

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (3.21)$$

$$G_c(s) = 1.2 \frac{T}{L} \left( 1 + \frac{1}{2L_s} + 0.5L_s \right) \quad (3.22)$$

$$G_c(s) = 0.6 T \frac{\left( s + \frac{1}{L} \right)^2}{s} \quad (3.23)$$

Thus, the PID controller has a pole at the origin and double zeros at  $s = -1/L$ .

## CHAPTER 4

### ARTIFICIAL NEURAL NETWORK (ANN)

---

#### 4.1 INTRODUCTION

The conventional feedback controllers find wide applications in the process industry. One of the earliest controllers that were used for control was the Proportional-Integral-Derivative (PID) controller. PID controller has proved to be remarkably effective in controlling a wide range of processes. The use of PID controller does not require an exact process model and hence, it is effective in controlling industrial processes whose models are difficult to derive. Moreover, PID controllers are based on linear control theory and are much easier to understand and implement. However, in spite of these advantages of the PID controllers, they possess several disadvantages as well. If the operating points of the process or the plant parameters are changed due to disturbances etc., such fixed-gain feedback controllers need to be returned to obtain the new optimal settings. For the processes with variable time delays, varying plant parameters, large non-linearities and considerable process noise, the PID controller does not give optimal performance. For the control of such highly complex and non-linear systems the PID controllers fail miserably because of their limitations.

In this light, new control schemes should be devised for a better solution of a non linear system. Recently, work has been started toward the development of Artificial Neural Network (ANN) based intelligent controllers. A lot of research is going on applications of ANNs in process control and have been successfully implemented. The most widely applied neuro-control scheme is the direct inverse model neuro-control approach. In this approach neural network is trained to learn the inverse of the plant either offline or online. Once trained, it can then be configured to control the plant. The salient features of ANNs make them ideally suitable for application in intelligent control systems [27].

#### 4.2 SALIENT FEATURES OF ARTIFICIAL NEURAL NETWORK (ANN)

The following characteristics and properties of neural networks are significant [28]:

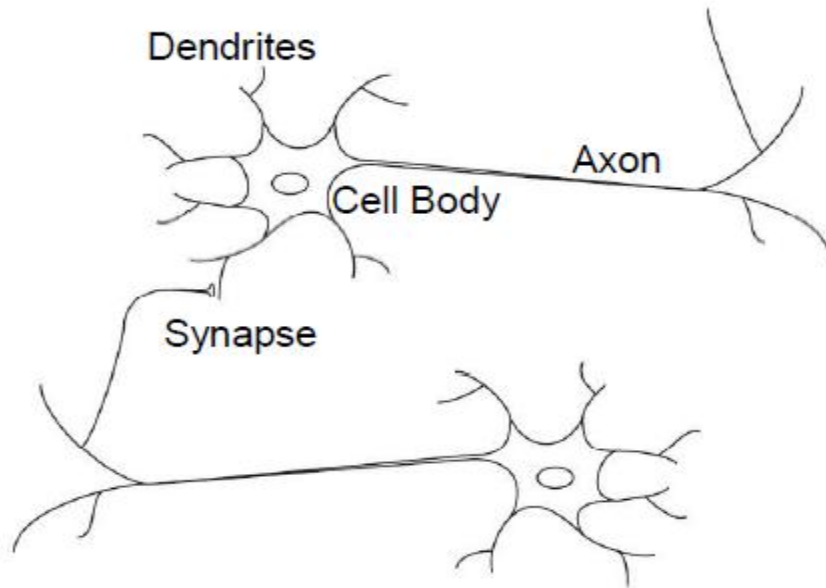
- i. **Non-Linearity:** Neural networks have great promise in diverse fields because of their theoretical ability to approximate arbitrary non-linear mappings. In a neural network, the non-linearity is distributed throughout the network.
- ii. **Provide Model Free Environment:** ANN controllers do not require derivation of the mathematical model of the process. Thus, the cumbersome mathematics and the added

assumptions which introduce approximations in the model are not needed. ANN model or controller does not require a priori knowledge of the plant dynamics.

- iii. **Parallel Distributed Processing:** Neural networks have a highly parallel structure, which lends itself immediately to parallel implementation. The basic processing unit in a neural network has a very simple structure. This, along with parallel implementation results in very fast overall processing.
- iv. **Input-Output Mapping:** A popular paradigm of learning called supervised learning involves modification of the synaptic weights of a neural network by applying a set of input-output training samples. The network learns from the examples by constructing an input-output mapping for the problem at hand.
- v. **Learning with Experience:** Neural network have a built-in capability to learn and adapt their synaptic weights to changes in the surrounding environment.
- vi. **Fault Tolerance:** A neural network, implemented in hardware, has the potential to be inherently fault tolerant, or capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions.
- vii. **Data Fusion:** Neural networks can operate simultaneously on both quantitative and qualitative data. The neural network can deal with information that is fuzzy, probabilistic, noisy and inconsistent.
- viii. **Hardware Implementation:** Not only can the networks be implemented in parallel, but also dedicated VLSI implementations are being introduced recently. This brings additional speed and increases the scale of neural networks, which can be implemented.
- ix. **Multivariable Systems:** Neural network naturally process many inputs and have many outputs. Thus, these are readily applicable to multivariable systems.

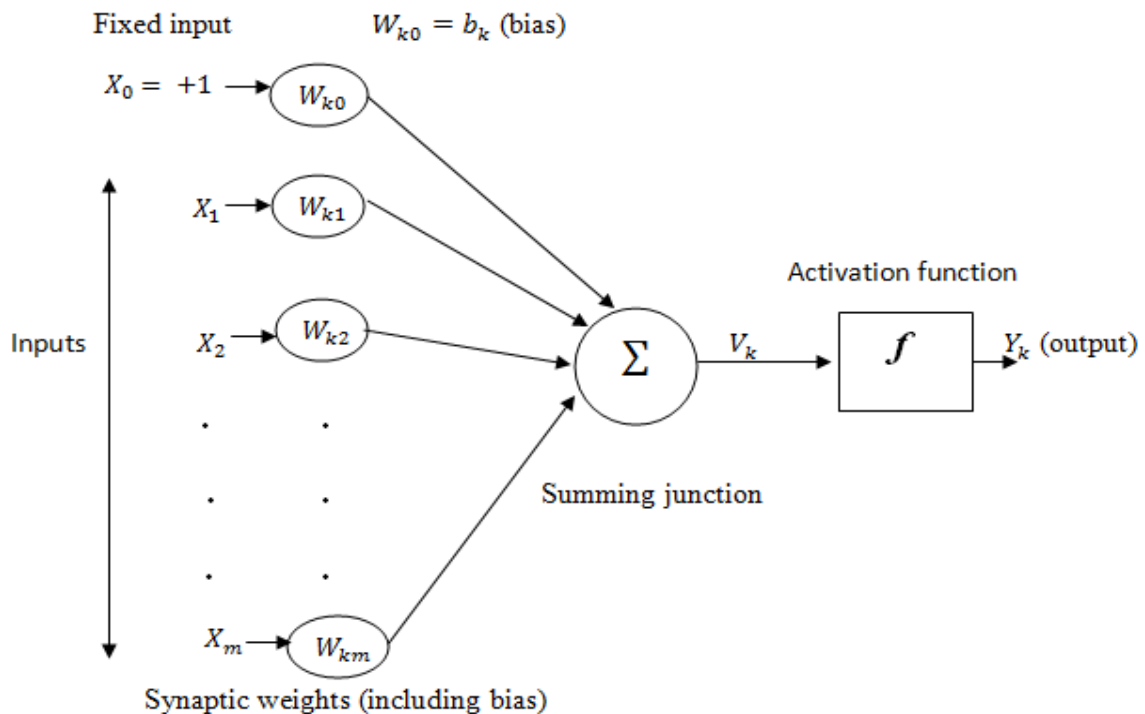
### 4.3 ARTIFICIAL NEURAL NETWORKS (ANN) OVERVIEW

A biological neuron consists of three principle components: dendrites, cell body and axon as shown in Fig 4.1. The tree like structures is called dendrites. They are receptive networks of nerve fibres that carry electrical signals into the cell body. The cell body effectively sums and thresholds these incoming signals. The signal from the cell body is carried out to other neurons by the single long fibre called axon. The point where an axon of one cell and a dendrite of another cell are in contact is called synapses [24].



**Fig.4.1: Schematic of Biological Neurons**

Artificial neural networks are neither as powerful as biological neurons in the brain nor are they as complex. They have two basic similarities between them, 1. Both are simple computational devices that are highly interconnected, 2. The connections between neurons determine the function of the network. The neuron modal in Fig.4.2 includes an externally applied bias, denoted by  $b_k$ .



**Fig.4.2: Multiple Input Neuron**

The bias  $b_k$  has the effect of increasing or decreasing the net input of the activation function, depending on whether it is positive or negative, respectively.

In mathematical terms, we can describe a neuron  $k$  by writing the following pairs of eqns:

$$u_k = \sum_{j=1}^m W_{kj} X_j \quad \& \quad Y_k = f(u_k + b_k)$$

Where  $X_1, X_2, \dots, X_m$  are the input signals;  $W_{k1}, W_{k2}, \dots, W_{km}$  are the synaptic weights of neuron  $k$ ;  $u_k$  is the linear combiner output due to the input signals;  $b_k$  is the bias;  $f$  is the activation function; and  $Y_k$  is the output signal of the neuron. The use of  $b_k$  has the effect of applying an affine transformation to the output  $u_k$  of the linear combiner in the model of Fig.4.2 [28].

$$V_k = u_k + b_k$$

### 4.3.1 ARTIFICIAL NEURAL NETWORK ARCHITECTURE

The arrangement of neurons into layers and the connections between them defines the network architecture. All the neurons in any particular layer perform similarly. Their behaviour is conditioned by the transfer function and weights.

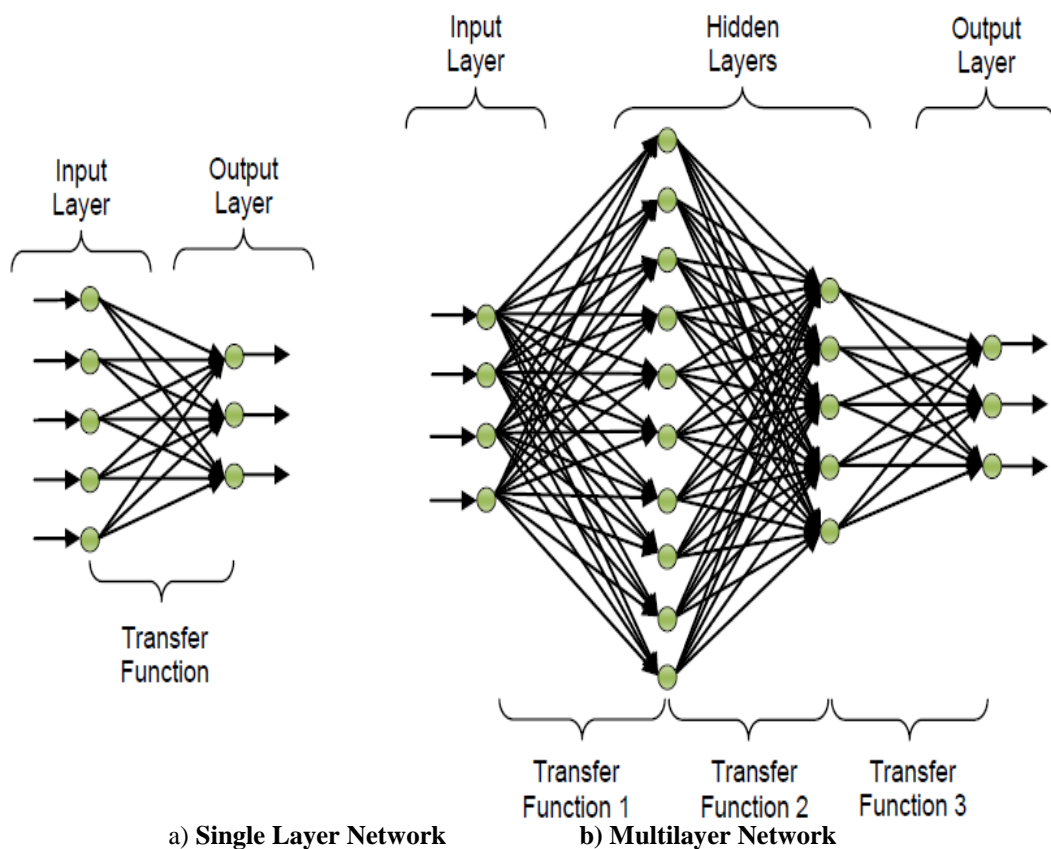


Fig.4.3 Neural Network Architecture

Networks are classified into two based on the number of layers they have, (1).Single-layer, (2).Multi-layer networks. Fig.4.3 shows typical architectures of a single layer and a multilayer network. Single layer networks have an input layer and an output layer. While counting the number of layers in a network, input layer is not counted since its only task is to provide input data. Input layer does not perform any calculations. Multi-layer network has several layers which include an input layer, one or more hidden layers and an output layer. The hidden layers don't interact directly with the external surroundings of the network. The multilayer network shown in Fig.4.3 consists of three layers; two hidden layers and one output layer connected by three layers of weights [28].

#### 4.3.2 WEIGHTS & NETWORK TRAINING

Weight corresponds to the strength of a synapse between two neurons of layers. It is also referred to as synaptic weight. Initially, weights are set to either zero or a very small number. Those weights are changed using a learning rule during the iterative training process; A positive weight represents an excitatory stimulus while a negative weight corresponds to an inhibitory stimulus. A zero weight value indicates no connection. The weight connections between layers of neurons are denoted as weight matrices  $W$ . Typically, the matrix element  $w_{ij}$ , is used to denote the weight connecting the output of  $i^{\text{th}}$  neuron to the input of  $j^{\text{th}}$  neuron. Network training is performed by the use of learning algorithms. Network training process can be classified into three major types:

- i. **Supervised learning:** In supervised learning, pattern or input vector is provided with an associated target or output vector. The supervised learning can be thought of as learning with a teacher, in the form of a function that provides continuous feedback on the quality of solutions obtained thus far. Weights are updated continuously based on the feedback received about the quality of solutions obtained thus far. Supervised learning is used in pattern recognition and regression.
- ii. **Unsupervised learning:** Unsupervised learning uses only an input vector. The weights are modified so that similar input patterns are assigned to the same target. Unsupervised learning can be used for general estimation, estimation of statistical distribution and filtering. In reinforced learning, input data are not given, but generated by an agent's interactions with the environment. At each point in time  $t$ , the agent performs an action and the environment generates an observation and an instantaneous cost, according to some unknown dynamics.

- iii. **Reinforced learning:** Reinforced learning is used in control problems, games and other sequential decision making tasks. In the present study supervised learning will be utilized with pattern or input and target or output generated using a commercial reservoir simulator [29].

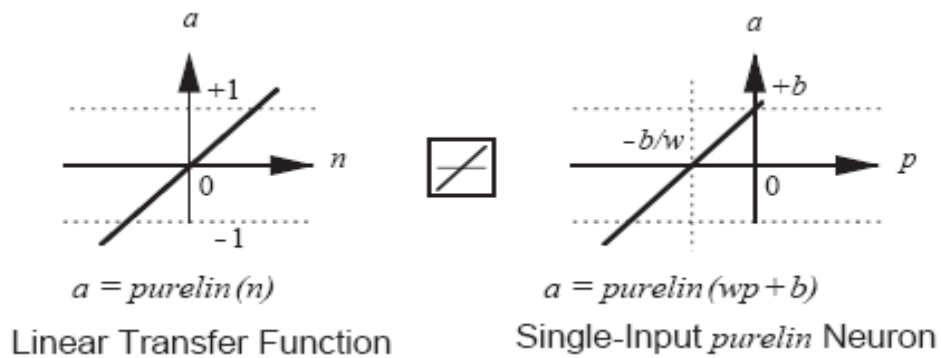
### 4.3.3 TRANSFER FUNCTIONS

Transfer function scales the response of an artificial neuron to an external stimulus and generates the neuron activation. Transfer function can be either a linear or non-linear function. Any multilayer perceptron using a linear transfer function has an equivalent single-layer network; a non-linear function is therefore necessary to gain the advantages of a multi-layer network..

The output ( $a$ ) of a linear transfer function is equal to its input ( $n$ ):

$$f(a)=n$$

The *purelin* transfer function is shown in the left of Fig.3.4, while the output ( $a$ ) versus input ( $n$ ) characteristic of a single-input linear neuron with a bias is shown on the right of Fig.4.4. This function commonly applied to the output layer since it allows the network to produce its output within the desired limits without renormalizing them.

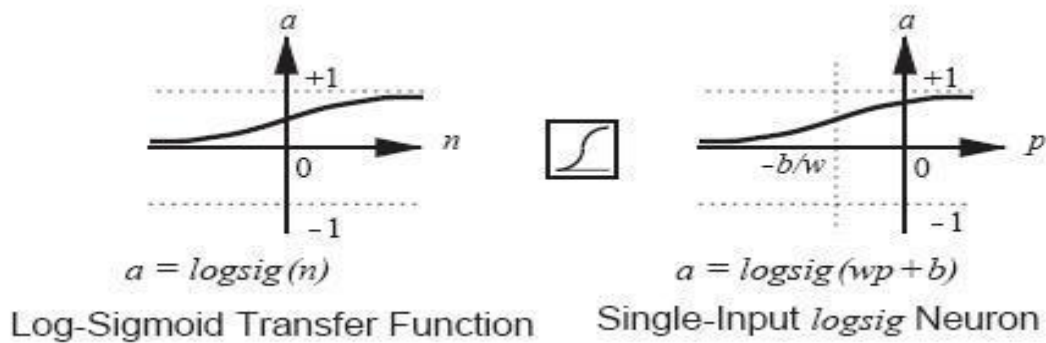


**Fig.4.4: Linear Transfer Function**

Log-sigmoid and hyperbolic tangent sigmoid are the transfer functions commonly used in multilayer networks using the back-propagation algorithm. Log-sigmoid transfer function takes the input (which may have any value between plus and minus infinity) and scales its output to range in between 0 and 1. The output ( $a$ ) of a log-sigmoid transfer function is calculated according to the expression:

$$a = \frac{1}{1 + e^{-n}}$$

Log-sigmoid transfer function is shown in the left of Fig.4.5, while the output ( $a$ ) versus input ( $p$ ) characteristic of a single-input linear neuron with a bias is shown on the right of Fig.4.5.

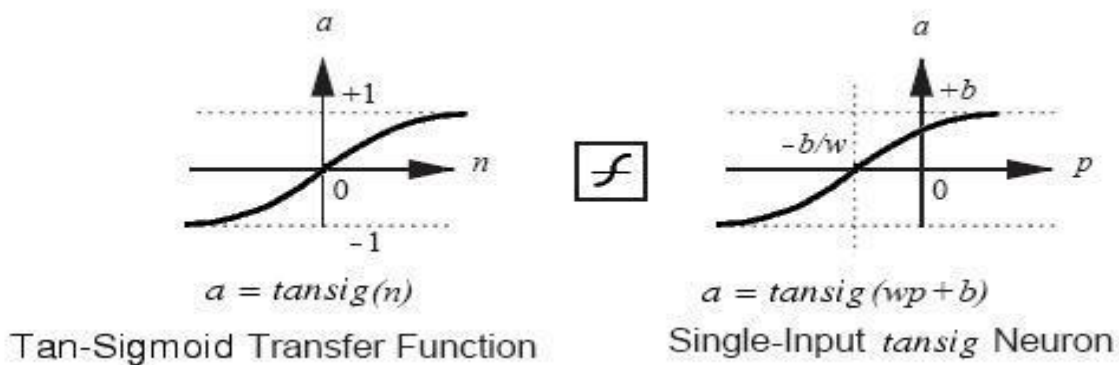


**Fig.4.5: Log-Sigmoid Transfer Function**

Hyperbolic tangent sigmoid transfer function has an advantage over log-sigmoid function of being able to deal directly with negative numbers. The output ( $a$ ) of a hyperbolic tangent sigmoid transfer function is calculated according to the expression:

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$$

Hyperbolic tangent sigmoid transfer function is shown in the left of Fig.4.6, while the output ( $a$ ) versus input ( $p$ ) characteristic of a single-input linear neuron with a bias is shown on the right of Fig.4.6.



**Fig.4.6: Hyperbolic Tangent Sigmoid Transfer Function**

#### 4.4 MULTILAYER FEEDFORWARD NETWORK WITH BACKPROPAGATION

Multilayer feedforward network with backpropagation is the most widely used network architecture. Feedforward networks are the simplest artificial neural networks. It has no feedback. Backpropagation is a supervised learning method. It is an implementation of the delta rule. The term backpropagation is an abbreviation for backwards propagation of errors. Feedforward network are most benefited when working with backpropagation since it adds the

component of feedback to the network. Feedforward networks with backpropagation are easy to implement, trains faster than other types of networks and solves many types of problems correctly. They operate in two steps. First is the feedforward step. During this step, input pattern is presented to the input layer and the information is transferred through hidden layers to the output layer.

Transfer functions process the information as they move from layer to layer. Second is the backpropagation step when, backpropagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. During this step, networks response is compared to the desired output and the errors are propagated from the output layer to the inner layers. This error signal is used to adjust the network weights. Each intermediate layer receives a portion of the total error signal based roughly on the relative contribution of the unit made to the original output. Thus after several iterations of this process, the error signal generated becomes small. At this stage the network is considered trained for the intended purpose and able to make predictions from a novel set of inputs [28].

#### **4.4.1 CONVERGENCE & TRAINING EFFICIENCY**

Convergence problem refers to a situation where the total error of the current iteration is lower than the one from previous iteration. There can be several causes for convergence problems. The most common cause is the presence of several local minima on the error surface. This problem can be prevented by using a momentum parameter. Using a high momentum parameter can also help to increase speed of convergence. But care should be taken not to use a very high momentum since this may lead to overshooting the actual minimum thus making the network unstable. The optimization method used may not be guaranteed to converge when the system is far away from the local minimum. Learning efficiency can be improved by taking the following measures, 1) Using high momentum parameter, 2) Using functional links, 3) Using faster learning functions. As discussed earlier use of high momentum parameter can improve the speed of learning. Use of functional links in the input and output layers usually help the network interpret the data better and thus improve the learning efficiency. Training and learning functions are mathematical procedures used to automatically adjust the network's weights and biases. The training function dictates a global algorithm that affects all the weights and biases of a given network. Gradient decent is the most commonly used learning method. Other learning algorithms like conjugate gradient methods, the Levenberg-Marquardt algorithm (LM) can be used to make the learning faster. Levenberg-Marquardt algorithm is one of the fastest backpropagation algorithms, and is

highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms.

#### 4.4.2 BACKPROPAGATION WITH FEEDFORWARD NETWORK

Back-propagation has reawakened the scientific and engineering community to the modelling and processing of many quantitative phenomena using neural-networks. The learning algorithm is applied to the multilayer feed forward networks consisting of processing elements with continuous differentiable activation functions. Such networks associated with the back-propagation learning algorithm are also called back-propagation networks. Given a training set of input-output pairs  $\{(x^{(k)}, d^{(k)})\}, k = 1, 2, \dots, p$ , the algorithm provides a procedure for changing the weights in the back-propagation network to classify the given input patterns correctly. The basis for this weight update algorithm is simply the gradient-descent method as used for simple perceptrons with differentiable units.

For a given input-output pair  $(x^{(k)}, d^{(k)})$ , the back-propagation algorithm performs two phases of data flow. First, the input pattern  $x^{(k)}$  is propagated from the input layer to the output layer and, as a result of this forward flow of data, it produces an actual output  $y^{(k)}$ . Then the error signals resulting from the difference between  $d^{(k)}$  and  $y^{(k)}$  are back-propagated from the output layer to the previous layers for them to update their weights. Consider a three-layer network as shown in Fig.4.7 to illustrate the details of the back-propagation learning algorithm. Here we have  $m$  processing elements in the input layer,  $l$  processing elements in the hidden layer, and  $n$  processing elements in the output layer. First, let us consider an input-output training pair  $(x, d)$ , where the superscript  $k$  is omitted for notation simplification.

Give an input pattern  $x$ , a processing element  $q$  in the hidden layer receives a net input of

$$net_q = \sum_{j=1}^m v_{qj}x_j \quad (4.1)$$

And produces an output of

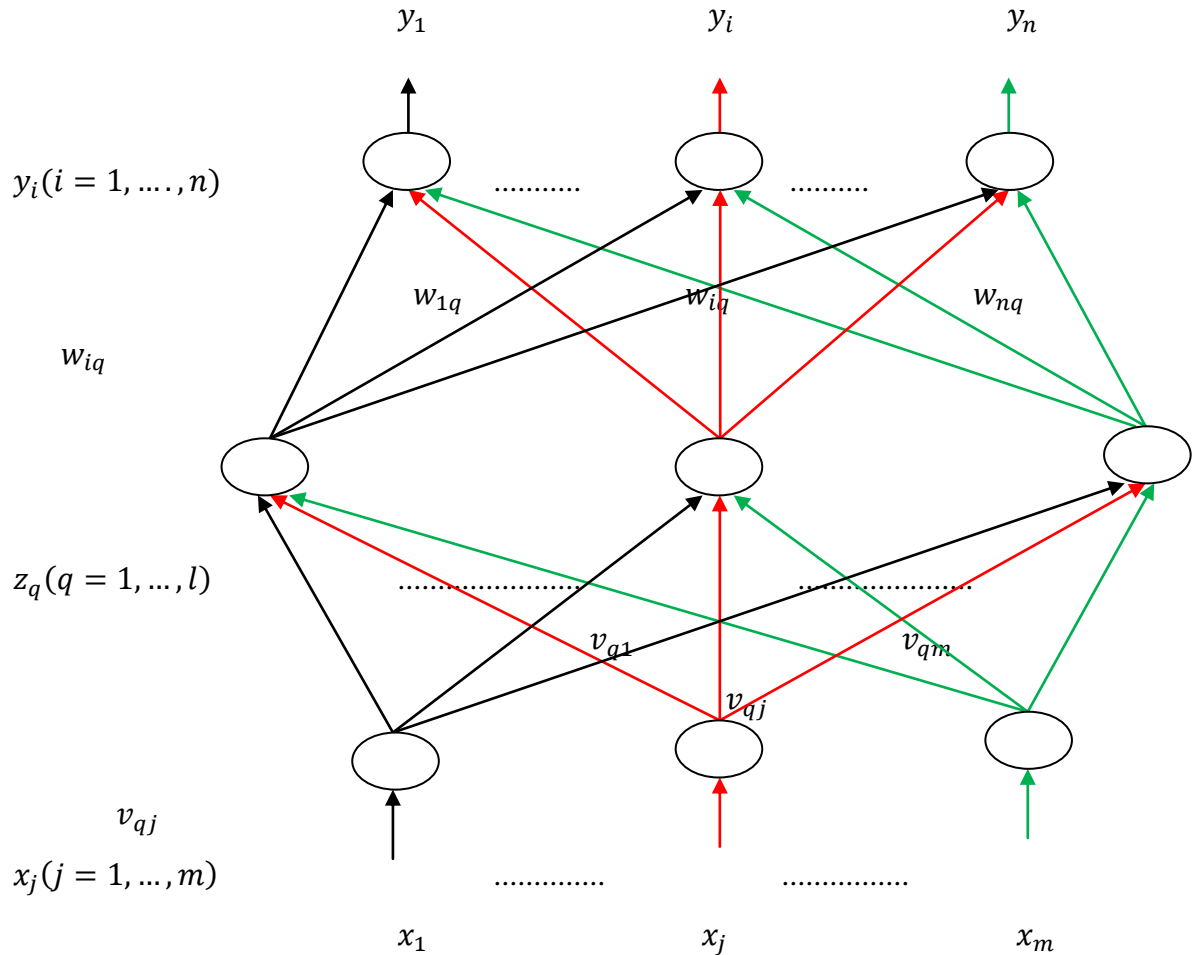
$$z_q = a(net_q) = a\left(\sum_{j=1}^m v_{qj}x_j\right) \quad (4.2)$$

The net input for a processing element  $i$  in the output layer is then

$$net_i = \sum_{q=1}^l w_{iq}z_q = \sum_{q=1}^l w_{iq}a\left(\sum_{j=1}^m v_{qj}x_j\right) \quad (4.3)$$

And it produces an output of

$$y_i = a(\text{net}_i) = a\left(\sum_{q=1}^l w_{iq}z_q\right) = a\left(\sum_{q=1}^l w_{iq}a\left(\sum_{j=1}^m v_{qj}x_j\right)\right) \quad (4.4)$$



**Fig.4.7 Three layer back-propagation network**

The above equations indicate the forward propagation of input signals through the layers of neurons till output layer.

The cost function is defined as:

$$E(w) = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^n [d_i - a(\text{net}_i)]^2 \quad (4.5)$$

$$= \frac{1}{2} \sum_{i=1}^n \left[ d_i - a\left(\sum_{q=1}^l w_{iq}z_q\right) \right]^2 \quad (4.6)$$

Error signal is propagated back up to input layer and according to the gradient-descent method, the weights in the hidden-to-output connections are updated by

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}} \quad (4.7)$$

Using Eqs. (4.3)-(4.6) and the chain rule for  $\partial E/\partial w_{iq}$ , we have

$$\Delta w_{iq} = -\eta \left[ \frac{\partial E}{\partial y_i} \right] \left[ \frac{\partial y_i}{\partial net_i} \right] \left[ \frac{\partial net_i}{\partial w_{iq}} \right] = \eta [d_i - y_i][a'(net_i)][z_q] \triangleq \eta \delta_{oi} z_q \quad (4.8)$$

Where  $\delta_{oi}$  is the error signal and its double subscript indicates the  $i$ th node in the output layer. The error signal is defined

$$\delta_{oi} \triangleq -\frac{\partial E}{\partial net_i} = -\left[ \frac{\partial E}{\partial y_i} \right] \left[ \frac{\partial y_i}{\partial net_i} \right] = [d_i - y_i][a'(net_i)] \quad (4.9)$$

Where  $net_i$  is the net input to processing element  $i$  of the output layer and  $a'(net_i) = \partial a(net_i)/\partial net_i$ .

For the weight update on the input-to-hidden connections, the weight update on the link weight connecting processing element  $j$  in the input layer to processing element  $q$  in the hidden layer

$$\Delta v_{qj} = -\eta \left[ \frac{\partial E}{\partial v_{qj}} \right] = -\eta \left[ \frac{\partial E}{\partial net_q} \right] \left[ \frac{\partial net_q}{\partial v_{qj}} \right] = -\eta \left[ \frac{\partial E}{\partial z_q} \right] \left[ \frac{\partial z_q}{\partial net_q} \right] \left[ \frac{\partial net_q}{\partial v_{qj}} \right] \quad (4.10)$$

From Eq. (4.5) & (4.6), it is clear that each error term  $[d_i - y_i], i = 1, 2, \dots, n$ , is a function of  $z_q$ . Evaluating the chain rule, we have

$$\Delta v_{qj} = \eta \sum_{i=1}^n [(d_i - y_i)a'(net_i)w_{iq}] a'(net_q)x_j \quad (4.11)$$

Using Eq. (4.9), we can rewrite Eq. (4.11) as

$$\Delta v_{qj} = \eta \sum_{i=1}^n [d_{oi}w_{iq}] a'(net_q)x_j = \eta \delta_{hq} x_j \quad (4.12)$$

Where  $\delta_{hq}$  is the error signal of processing element  $q$  in the hidden layer and is defined as

$$\delta_{hq} \triangleq -\frac{\partial E}{\partial net_q} = -\left[ \frac{\partial E}{\partial z_q} \right] \left[ \frac{\partial z_q}{\partial net_q} \right] = a'(net_q) \sum_{i=1}^n \delta_{oi} w_{iq} \quad (4.13)$$

Where  $net_q$  is the net input to the hidden processing element  $q$  [Eq. (4.1)]. The error signal of a processing element in a hidden layer is different from the error signal of a processing

element in the output layer, as in Eqs. (4.9) & (4.13). Because of this difference, the above weight update procedure is called the generalized delta learning rule.

#### 4.4.3 BACKPROPAGATION LEARNING ALGORITHM STEPS:

Consider a network with  $Q$  feed forward layers,  $q= 1,2,\dots,Q$ , and let  $q_{net_i}$  and  $q_{y_i}$  denote the net input and output of the  $i$ th in the  $q$ th layer, respectively. The network has  $m$  input nodes and  $n$  output nodes. Let  $q_{w_{ij}}$  denote the connection weight from  $q - 1_{y_j}$  to  $q_{y_i}$  [27].

**Input:** A set of training pairs  $\{(x^{(k)}, d^{(k)}) | k = 1, 2, \dots, p\}$ , where the input vectors are augmented with the last elements as -1, that is,  $x_{m+1}^{(k)} = -1$ .

**Step 0:** Choose  $\eta > 0$  and  $E_{max}$  (maximum tolerable error). Initialize the weights to small random values. Set  $E=0$  and  $k=1$ .

**Step 1** (Training loop): Apply the  $k^{\text{th}}$  input pattern to the input layer ( $q=1$ ):

$$q_{y_i} = 1_{y_i} = x_i^{(k)} \quad \text{for all } i. \quad (4.17)$$

**Step 2:** Propagate the signal forward through the network using

$$q_{y_i} = a(q_{net_i}) = a\left(\sum_j q_{w_{ij}} q_{y_j}^{-1}\right) \quad (4.18)$$

**Step 3:** Compute the error value and error signals  $Q_{\delta_i}$  for the output layer:

$$E = \frac{1}{2} \sum_{i=1}^n (d_i^{(k)} - Q_{y_i})^2 + E \quad (4.19)$$

$$Q_{\delta_i} = (d_i^{(k)} - Q_{y_i}) a'(Q_{net_i}) \quad (4.20)$$

**Step 4:** Propagate the errors backward to update the weights and compute the error signals  $q^{-1}\delta_i$  for the preceding layers:

$$\Delta q_{w_{ij}} = \eta q_{\delta_i}^{q-1} y_j \quad \text{and} \quad q_{w_{ij}}^{new} = q_{w_{ij}}^{old} + \Delta q_{w_{ij}} \quad (4.21)$$

$$q^{-1}\delta_i = a'(q^{-1}net_i) \sum_j q_{w_{ij}} q_{\delta_j} \quad \text{for } q = Q, Q - 1, \dots, 2 \quad (4.22)$$

**Step 5 :** Check whether the whole set of training data has been cycled once. If  $k < p$ , then  $k=k+1$  and go to step 1; otherwise, go to step 6.

**Step 6:** Check whether the current total error is acceptable: If  $E < E_{max}$ , then terminate the training process and output the final weights; otherwise,  $E=0$ ,  $k=1$ , and initiate the new training epoch by going to step 1.

**End BP.**

#### 4.5 LEVENBERG-MARQUARDT METHOD

The convergence rate of back propagation algorithm is significantly increased by improving its learning rate. The gradient descent method is a simplest technique that considers only the first order derivative of error function. The Marquardt method attempts to take the advantage of steepest descent method and Newton method that uses the second derivative in addition to the gradient to determine the step size for updating the weights of links. This method modifies the diagonal of Hessian matrix.

$$H_{ij} = [H_{ij}] + \alpha[I] \quad (4.23)$$

Where  $[I]$  is an identity matrix and  $\alpha$  is a positive constant which is sufficiently large of the order of  $10^4$ .

The energy function is evaluated at  $w_0$  as

$$H(w) = H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} \quad (4.24)$$

*Such that weights are updated as:*

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - H_{ij}^{-1}(w_{ij}^{(k)}) \nabla E(w_{ij}^{(k)}) \quad (4.25)$$

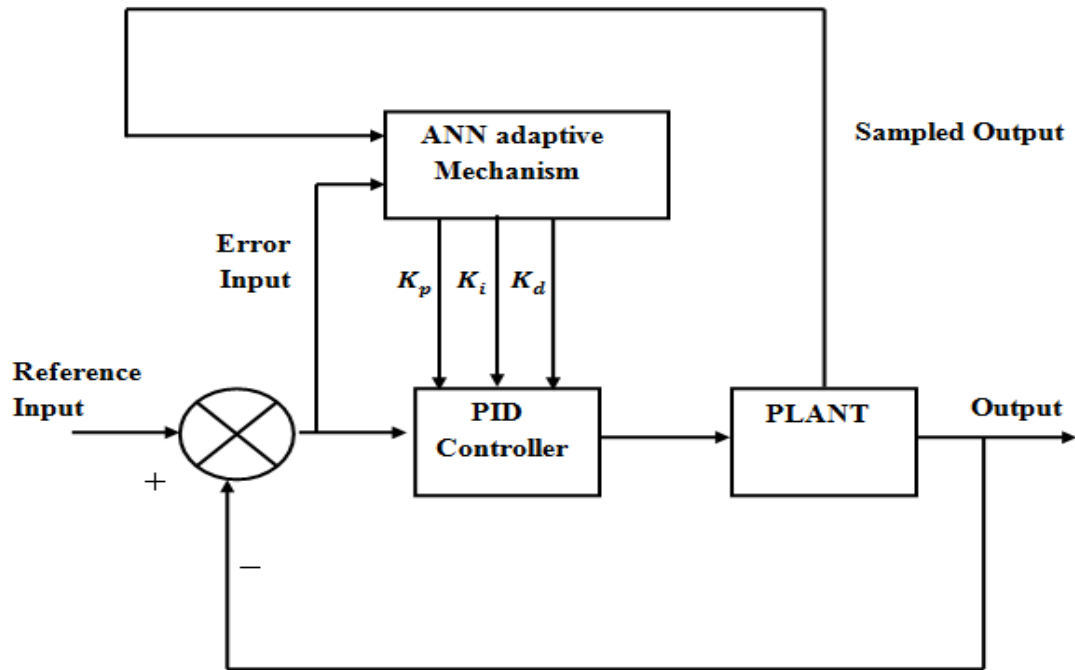
Weights are updated till the error at the output layer is within the acceptable limits or maximum number of iterations are achieved.

#### 4.6 Artificial Neural Network Controller as an Adaptive Controller Mechanism

Adaptive controller measures the disturbances from the output and tunes the parameters accordingly so as to minimise the disturbance and improves the performance of the controller. Feed forward adaptive control offers the advantage of fast action without involving any inner closed loops however suffers from the disadvantage of effect of unmeasured disturbances and amount of disturbances. PID gains are tuned by using neural networks. In order to get fast convergence and attain the better solution for a local minimum problem backpropagation method is used to tune the PID gains. Neural network can be trained to perform as a controller by learning an inverse model of the plant or as an emulator by indenting the forward model. Among many neural networks learning methods the back-propagation algorithm is the most widely used in a wide variety of applications.

In control problems, there has been development on neuro-control for robots control problems. Although various intelligent control methods may be applied to this problem, PID control is a major approach since it is robust to noise and stable for parameter change. Using

the back-propagation method, various kinds of neuro-controllers could be trained in such a way that the desired plant output is attained as much as possible.



**Fig.4.8: ANN based PID controller control model**

There are three typical types of neuro-controllers (a) a series type, (b) a parallel type, and (c) a self-tuning type. The series type neuro-controller constitutes the inverse dynamics of the plant in the part of the neural network.

The parallel type neuro-controller is to adjust the control input by using a conventional controller based on the neural network. The self-tuning neuro-controller is to tune the control parameters included in a conventional controller. In our thesis, we consider the self-tuning type of neuro-controller as shown in Fig.4.8. The reason why the PID scheme is adopted here is that the PID controller has been used in the process control system which includes various types of nonlinearities and non-Gaussian noises. The control structure of the self-tuning neuro-PID controller is shown in Fig.4.8 where the outputs of the neural network are proportional gain( $K_p$ ), integral gain( $K_i$ ), and derivative gain( $K_d$ ) and the inputs are selected in a suitable way according to the specific problem [30].

## **5.1 INTRODUCTION**

This chapter presents the simulated investigations of artificial neural network based adaptive controller developed for Brushless dc motor (BLDCM). The results obtained from PI and PID controller for speed control of BLDC are analysed and compared with ANN controller. The PI and PID controllers are designed for damped response of the speed with best possible estimation of controller parameters. Both PI and PID controller have the problem of overshoot when there is change in reference speed and subjected to step change in load. The gain and time constant parameters of PID controller are further tuned using Zeigler's Nicholas rule to enhance its performance.

For this purpose, the Simulink model of BLDC motor with feedback controller has been developed and has been tested with PI, PID, and tuned PID controller to obtain the transient response of speed of motor. In ANN controller is trained with the set of data to determine the PID controller parameters: Proportional constant,  $K_P$ , Integral constant,  $K_i$ , and Derivative constant,  $K_d$ . The performance of the BLDCM model is tested with adaptive ANN controller. The results obtained with ANN controller show the robustness and reliability of controller.

## **5.2 SIMULINK MODEL OF BLDCM**

There are three Simulink models of BLDCM are shown in figure 5.4, 5.5 and 5.6 respectively

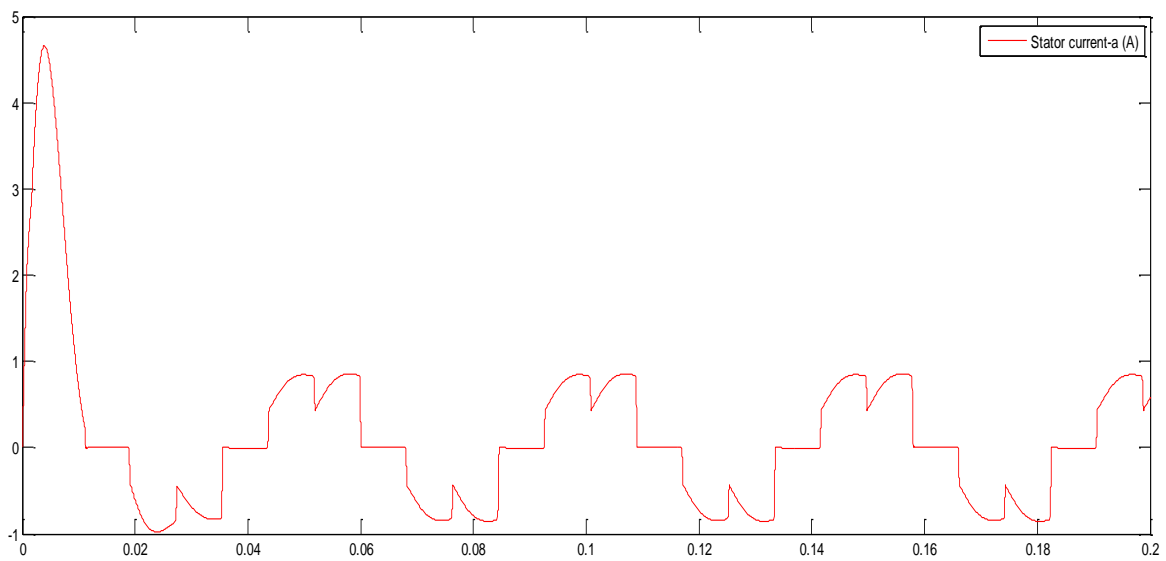
- Simulink model with PI controller
- Simulink model of BLDCM with PID controller
- Simulink model of BLDCM with ANN based PID

### **5.2.1 CIRCUIT DESCRIPTION OF SIMULINK MODEL OF BLDCM**

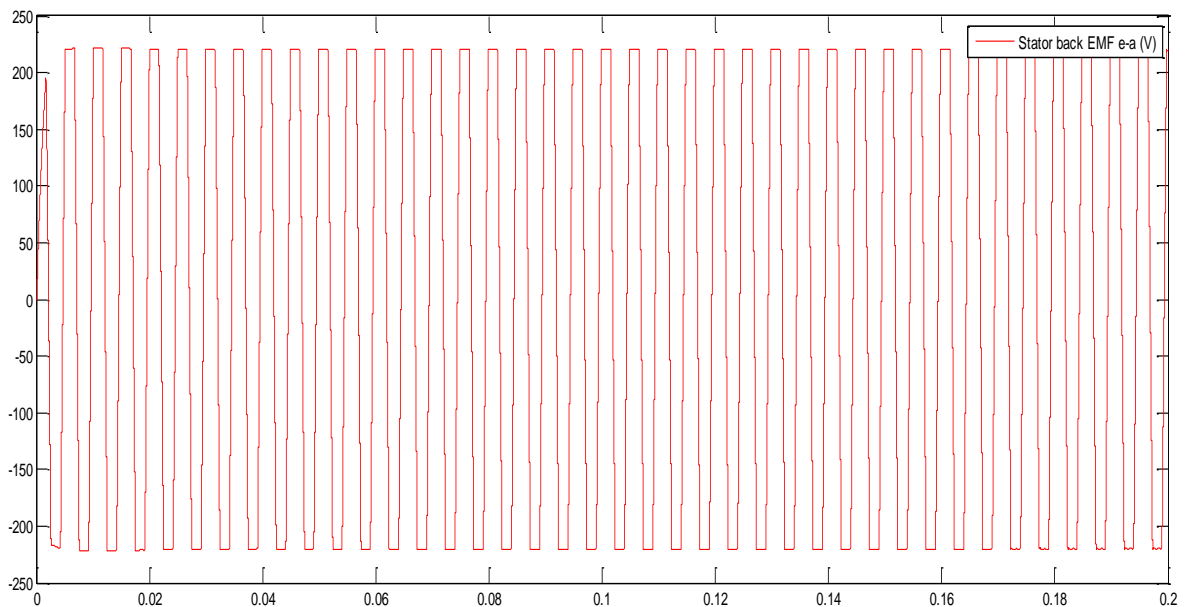
The simulated circuit includes a three phase brushless dc motor with specifications given in Appendix –A, is fed with six step voltage source inverter. The inverter is a MOSFET H-bridge of the Simulink Power systems Library of MATLAB. A speed regulator is used to control the dc bus voltage to a level of 500V. The inverter gates signals are produced by decoding the Hall position sensor signals of the motor. The three-phase output of the inverter is applied to the PMSM block's stator windings. The load torque applied to the machine's shaft is by using step input as 0, 1, 2, 3 & 4 Nm. Two control loops are used. The inner loop synchronises the inverter gates signals with the electromotive forces obtained based on

position signals through Hall sensors. The outer loop controls the motor's speed by varying the DC bus voltage at the input of driver unit using a feedback controller.

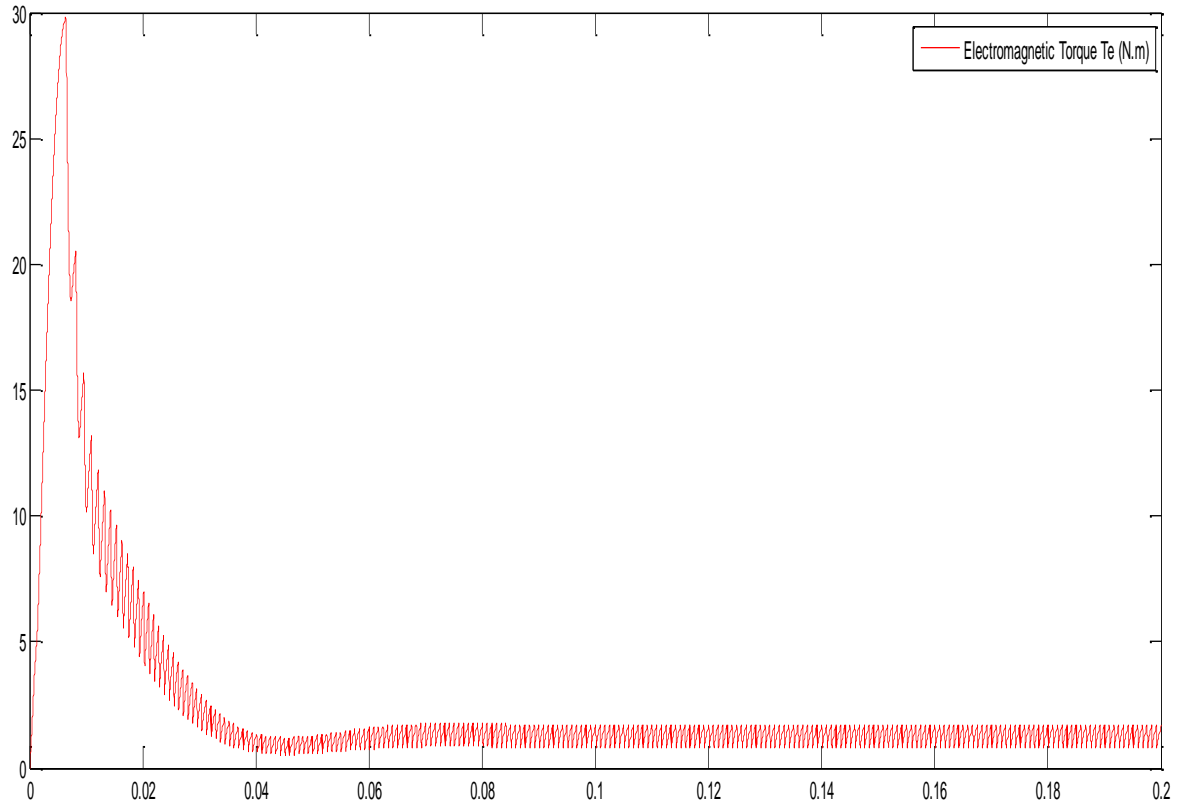
The initial current is high and decreases during the acceleration to the nominal speed. When the nominal torque is applied, the stator current increases to maintain the nominal speed.. However, the motor's inertia prevents this noise from appearing in the motor's speed waveform. Stator back emf is shown in Fig.5.2. The waveform is also observed in the electromagnetic torque signal  $T_e$  as shown in Fig.5.3



**Fig.5.1: Waveform of the Stator current (Phase A) of BLDCM**



**Fig.5.2 Waveform of Stator back EMF (V)**



**Fig.5.3: Waveform of the Electromagnetic torque  $T_e$  (Nm) of BLDCM**

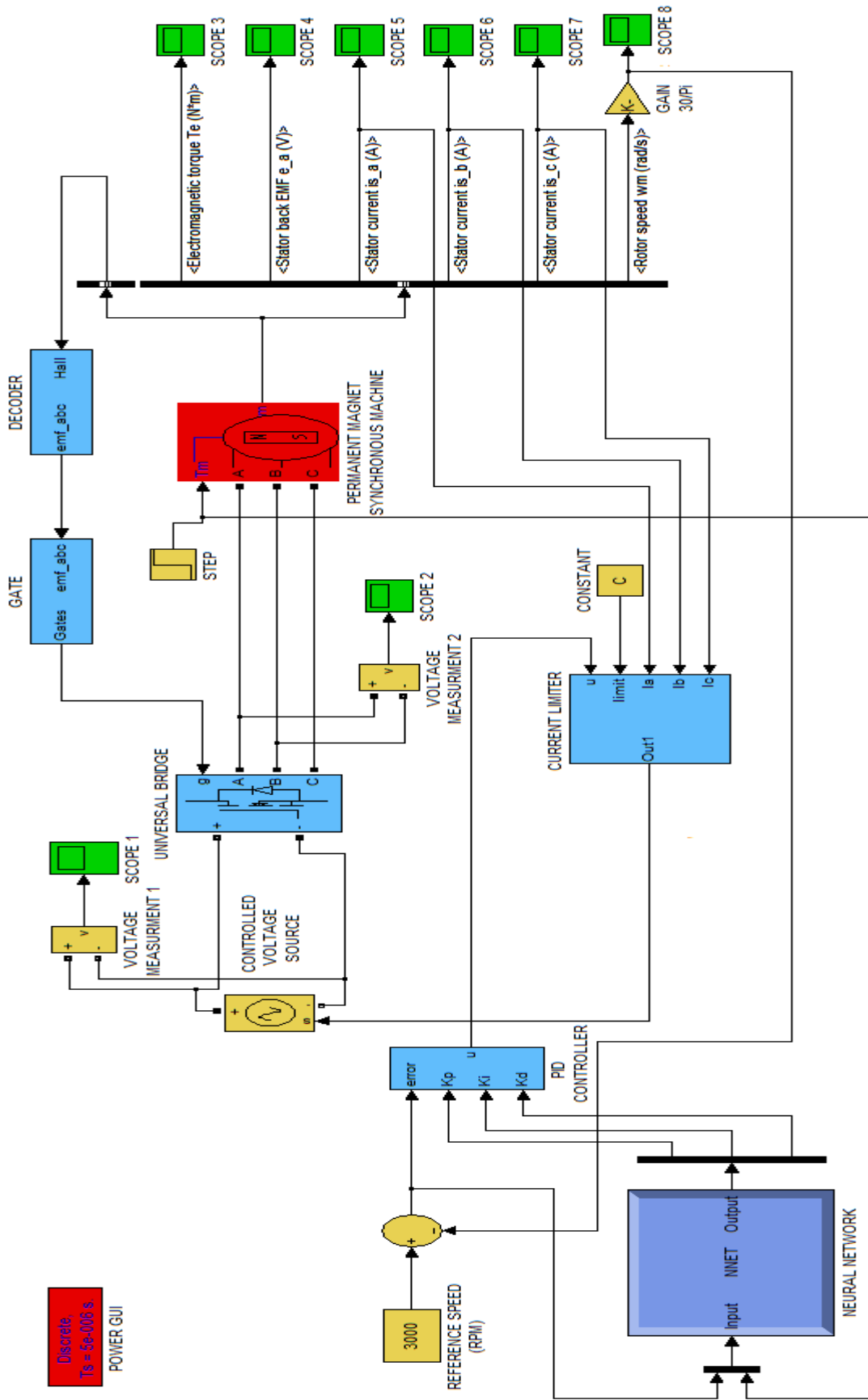


Fig.5.4: Speed control of BLDCM using ANN based PID controller

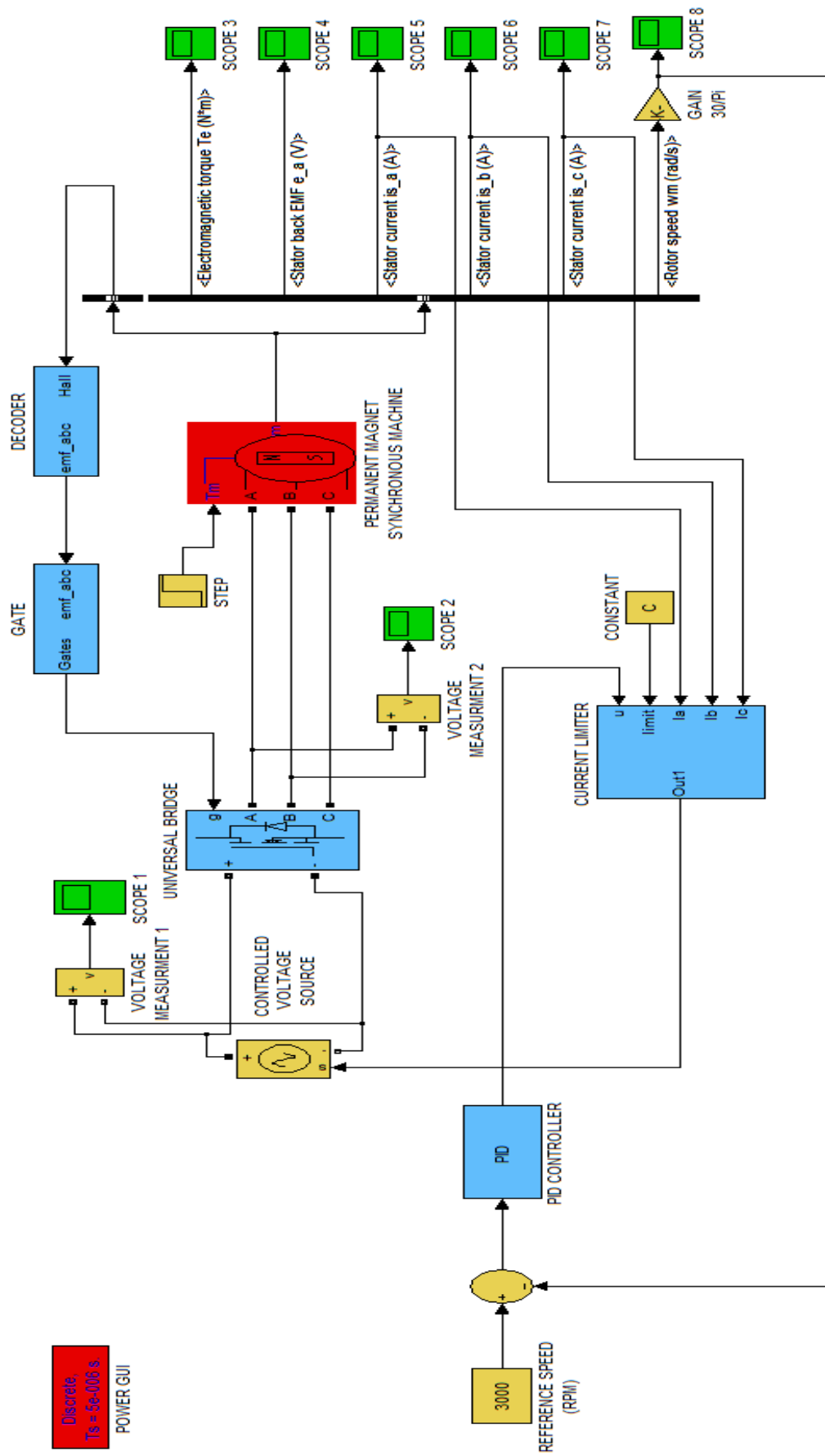


Fig.5.5: Speed control of BLDCM using PID controller

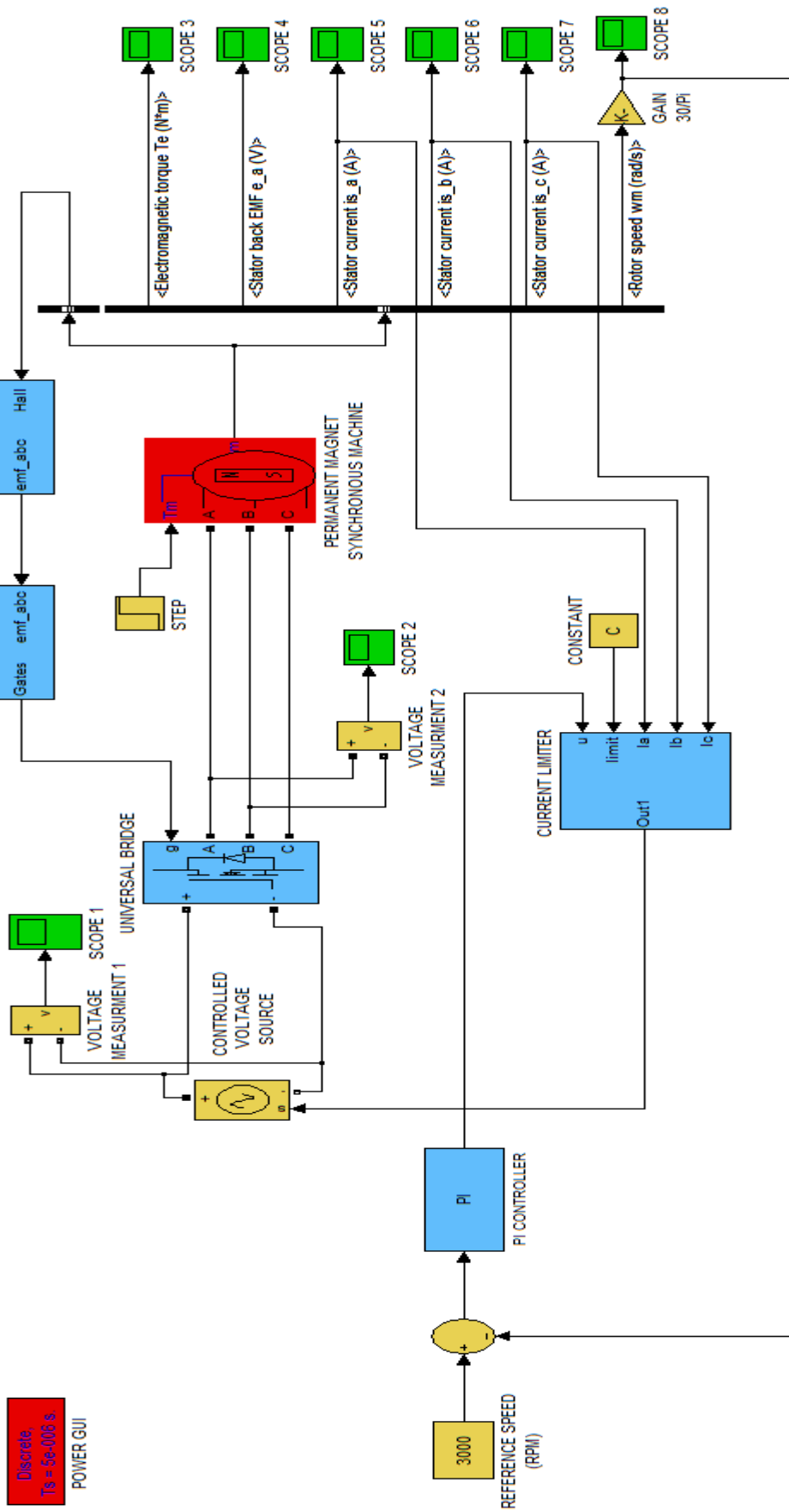


Fig.5.6: Speed control of BLDCM using PI controller

### 5.3 ARTIFICIAL NEURAL NETWORK MODEL

ANN model is developed for adaptive tuning of the parameters of PID controller. The feed forward neural network configuration as shown in fig.5.6, consists of three layers namely input layer, one hidden layer and one output layer. The input layer has two inputs as speed and torque of BLDCM and three output neurons as  $K_p$ ,  $K_i$ , &  $K_d$ , which are the input gains of PID controller for PID automatic tuning. The hidden layer is considered to have ten neurons.

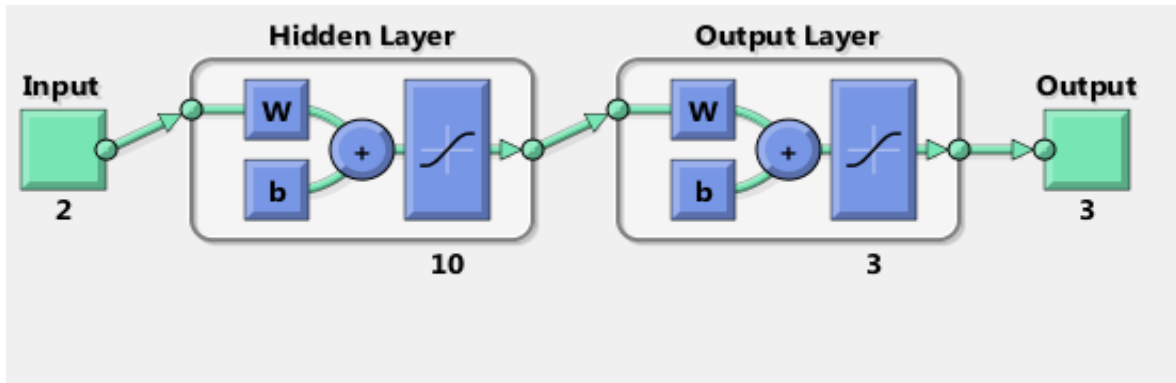


Fig.5.7: ANN feed-forward model

This Neural network is a two layer model, as shown in Fig.5.7

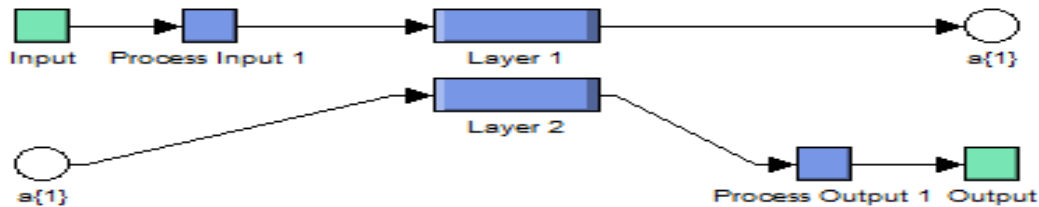


Fig.5.8: Two layer structure of Neural Network

This input – target pair is fed and new neural network is formed using “*NNTOOL*” in the MATLAB Simulink software. *NNTOOL* method provides the facility to train through one of the methods: Conjugate gradient method and Levenberg-Marquardt method. Levenberg-Marquardt method for back propagation is superior to approximate steepest descent [29]. method. Hence at first training is carried out using the ‘*NNTOOL*’ method. In the neural network TANSIG as transfer function is employed in the hidden layer and PURELIN in the output layer.

In the training stage, the network is trained with 35 input-output patterns. The input-output patterns are generated by varying the input within the error range for which the training is

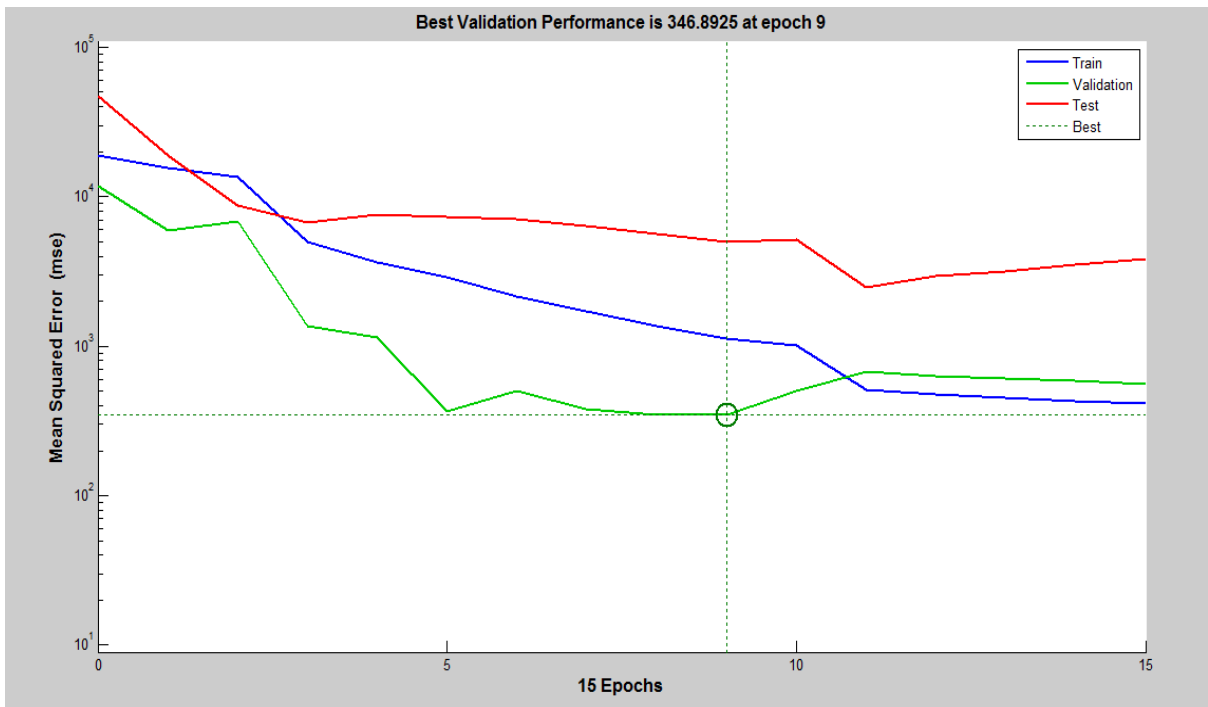
required and obtaining the corresponding output from conventional controller transient response. The trained neural network is tested for test patterns and acceptable results are obtained.

### 5.3.1 PARAMETERS OF ANN MODEL

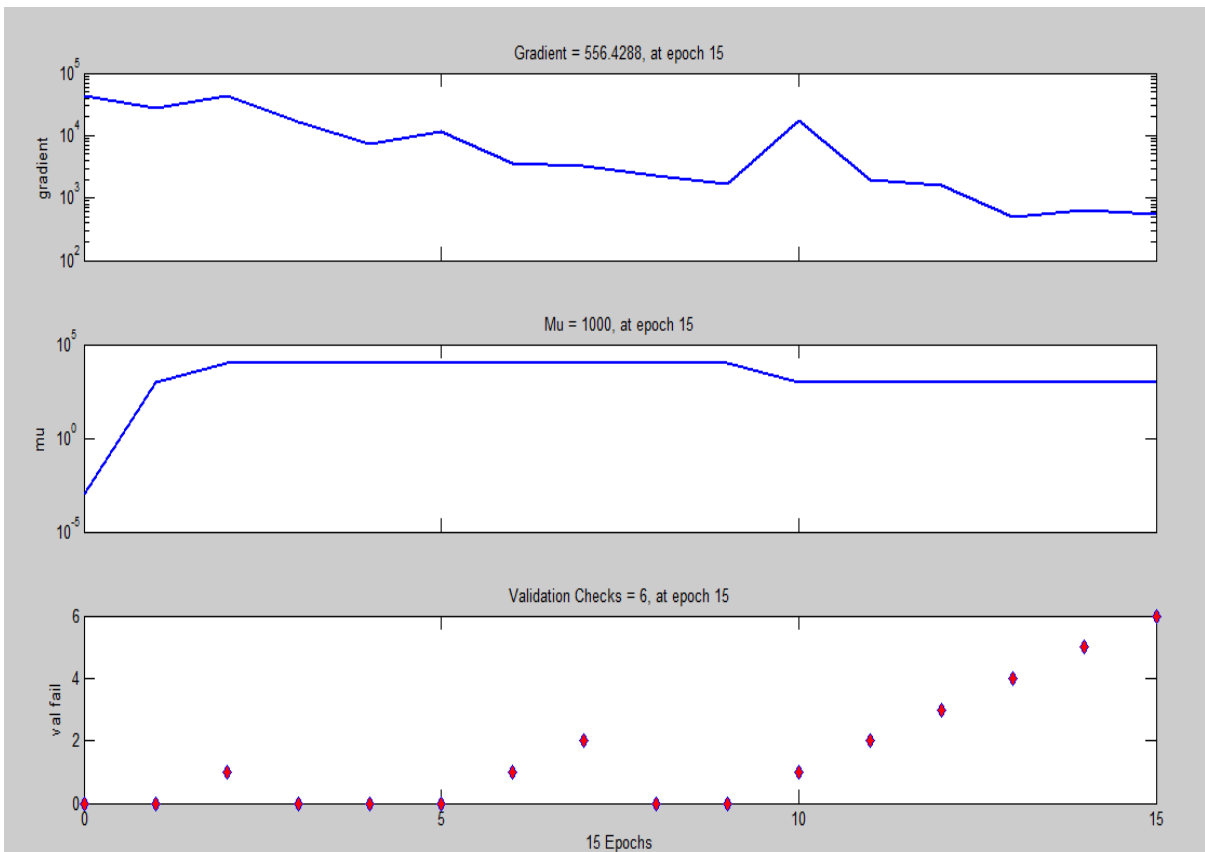
The parameters of proposed back propagation neural network configuration are mentioned in table 5.1. The model is configured using ‘NNTOOL’ box of MTALAB.

**Table.5.1: ANN model Parameters**

Inputs	2
Outputs	3
Input-output Patterns	35
Network configuration	Multilayer Feed forward
Training function	TRAINLM (Levenberg-Marquardt)
Adaption learning function	LEARNGDM
Performance function	Mean square error (MSE) = 0.001
Number of hidden layers	1
Number of Neurons in Hidden layer	10
Transfer function of Hidden layer	TANSIG
Transfer function of output layer	PURELIN
Epoch	15
Validation Test patterns	6
Gradient	534
Mu value	1000



**Fig.5.9: Training Performance**



**Fig.5.10: Training State**

## 5.4 TRANSIENT RESPONSE OF SPEED FOR BLDCM WITH ANN TUNED PID CONTROLLER

### (1) Speeds at Torque = 2 Nm (50% load)

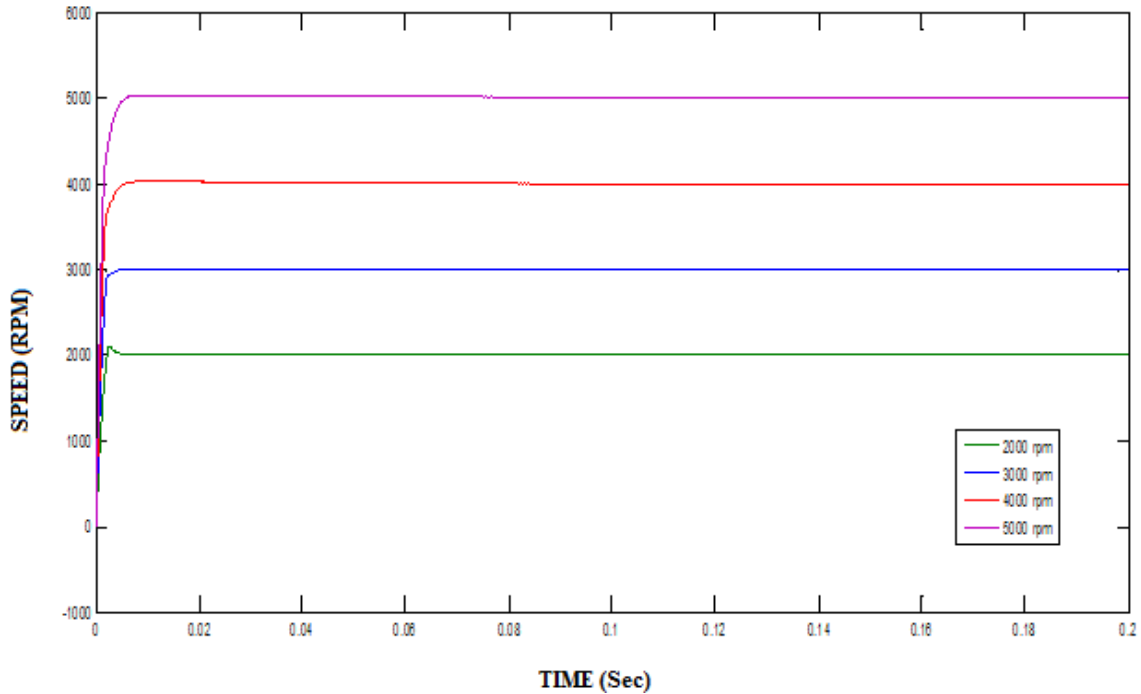


Fig.5.11: Transient response of speed for BLDCM with ANN tuned PID controller at TORQUE= 2 Nm

### (2) Speeds at Torque = 3 Nm

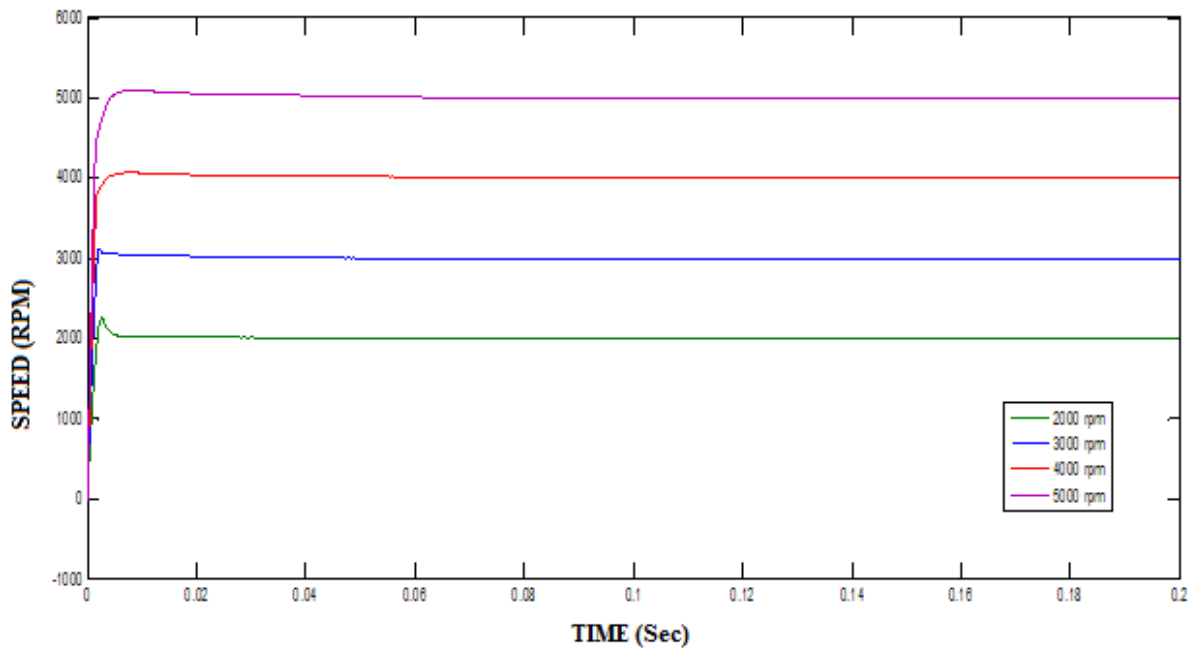
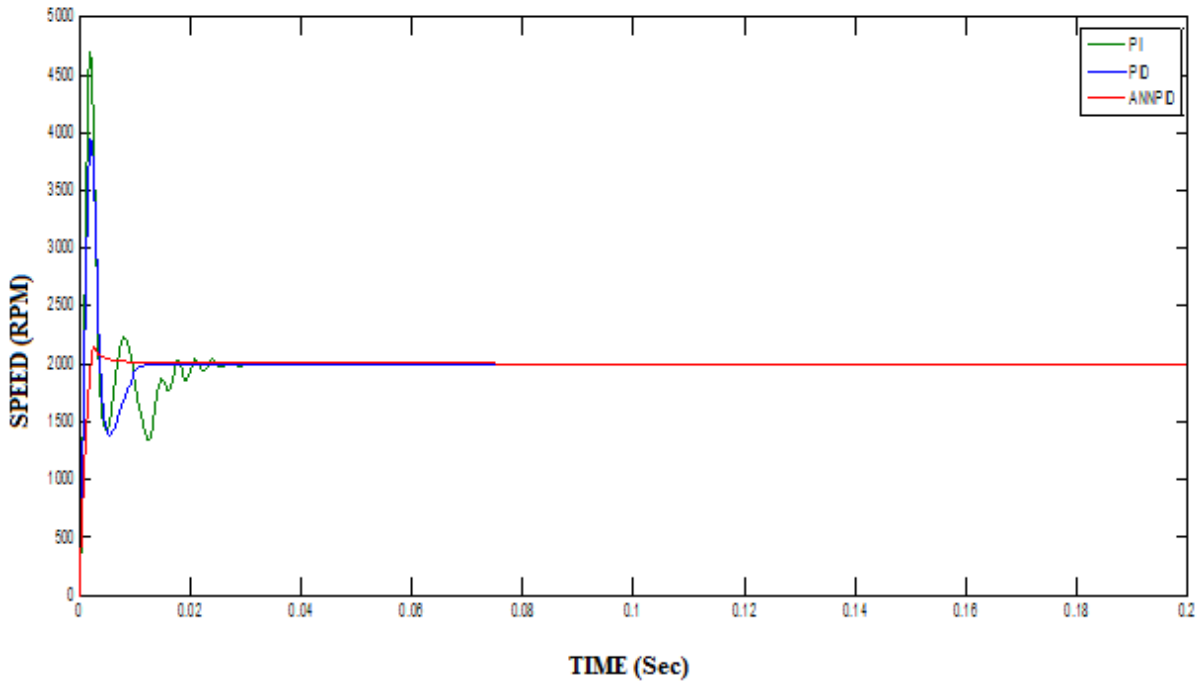


Fig.5.12: Transient response of speed for BLDCM with ANN tuned PID controller at TORQUE= 3 Nm

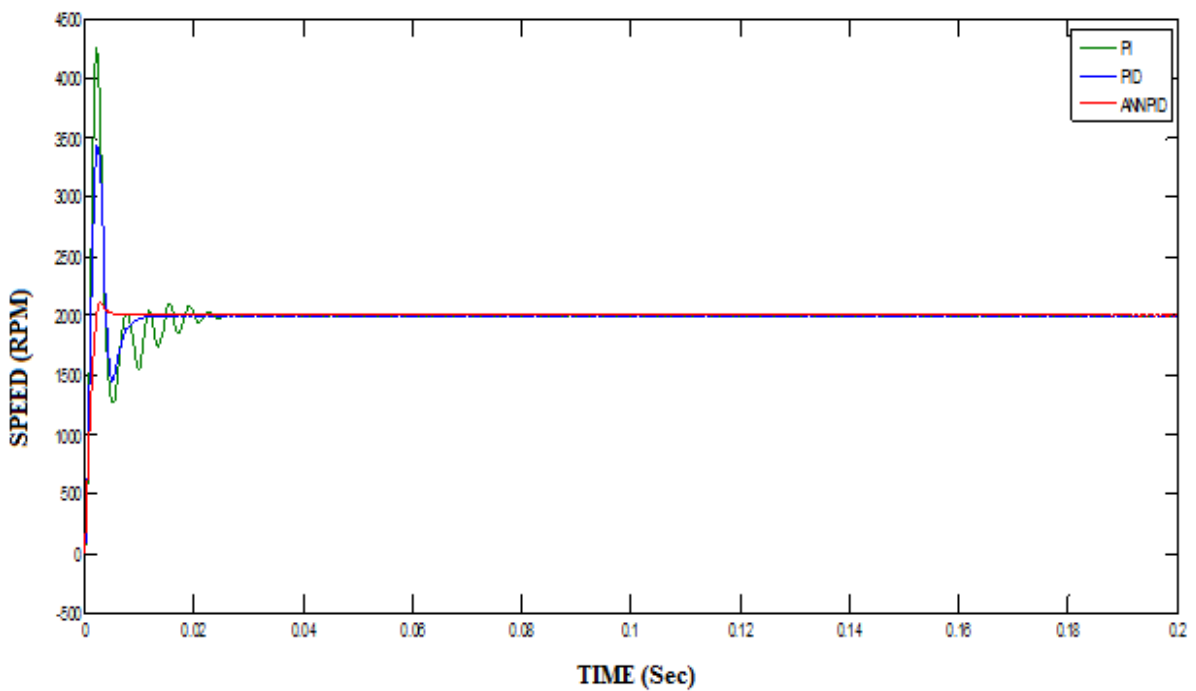
## 5.5 COMPARISON OF TRANSIENT RESPONSE OF SPEED FOR BLDCM WITH ANNPID, PID & PI CONTROLLERS

Here the comparison between the speeds of BLDCM with ANN based PID controller, PID controller & PI controller is shown at different torques.

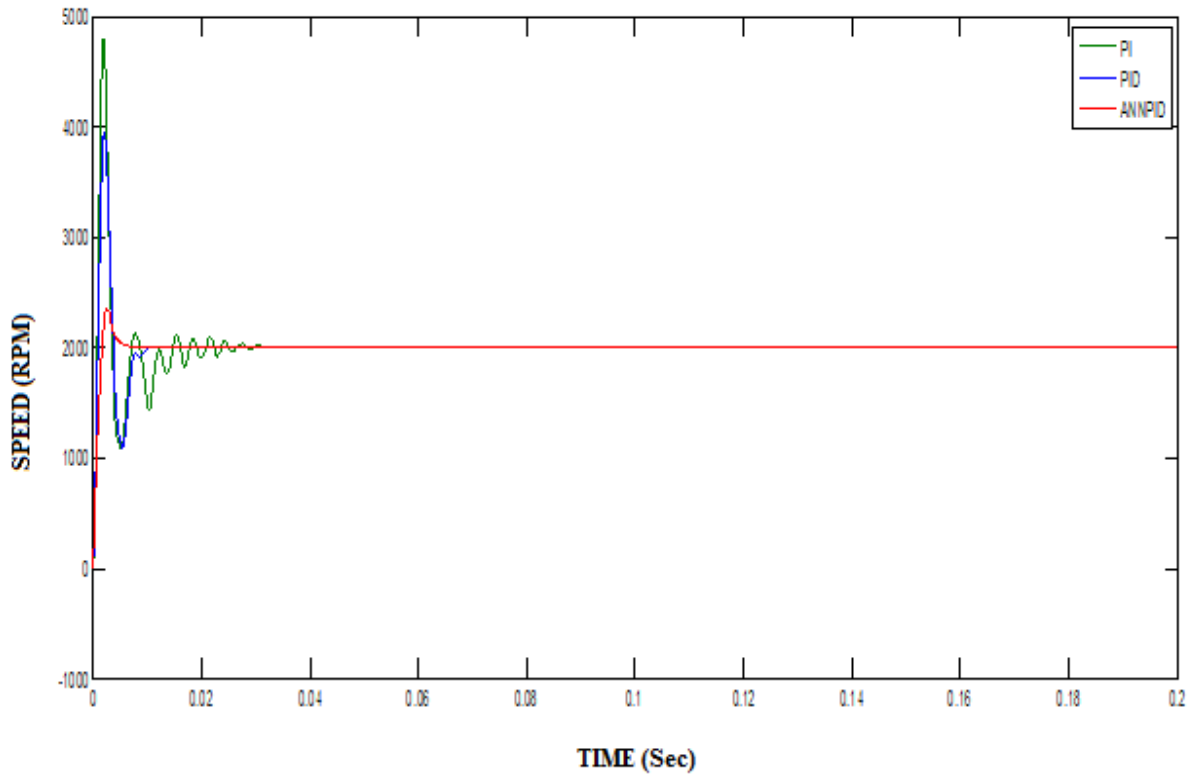
(1) Speed = 2000 rpm, Load Torque = 0 Nm (no load)



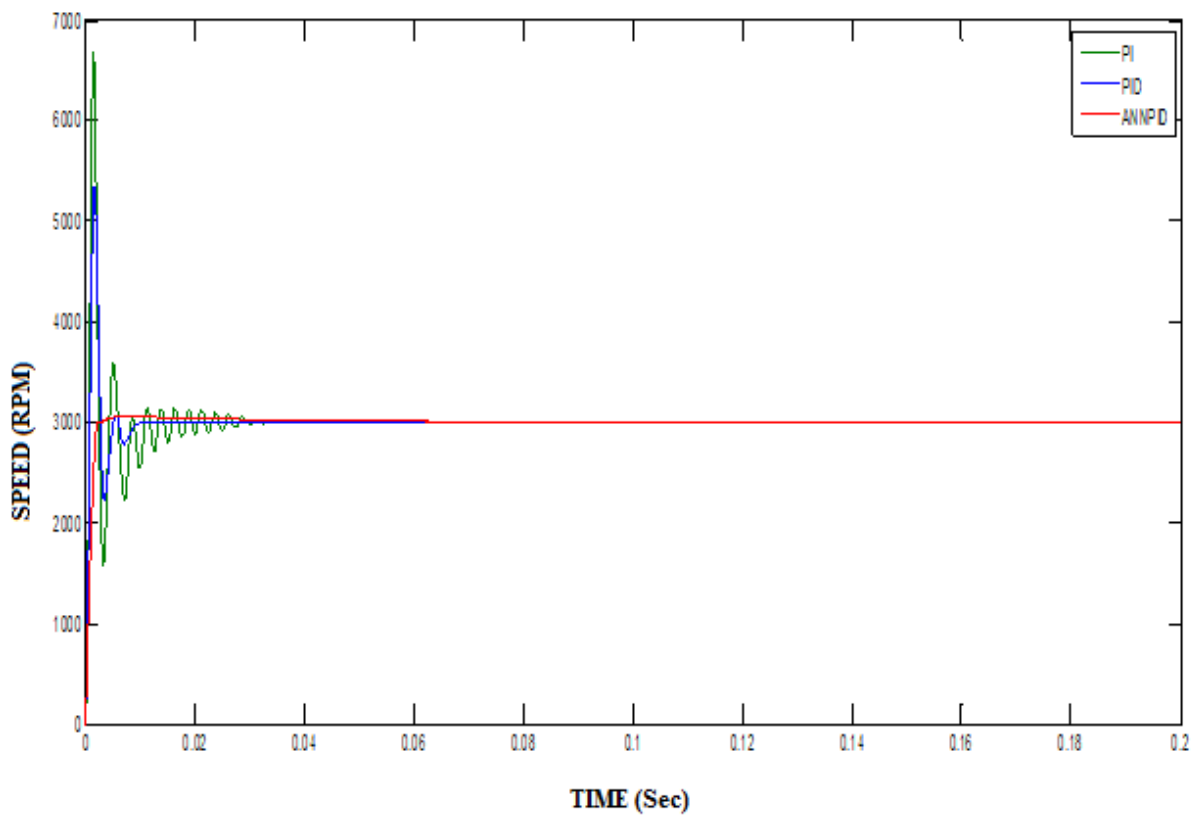
(2) Speed = 2000 rpm, Load Torque = 1 Nm.



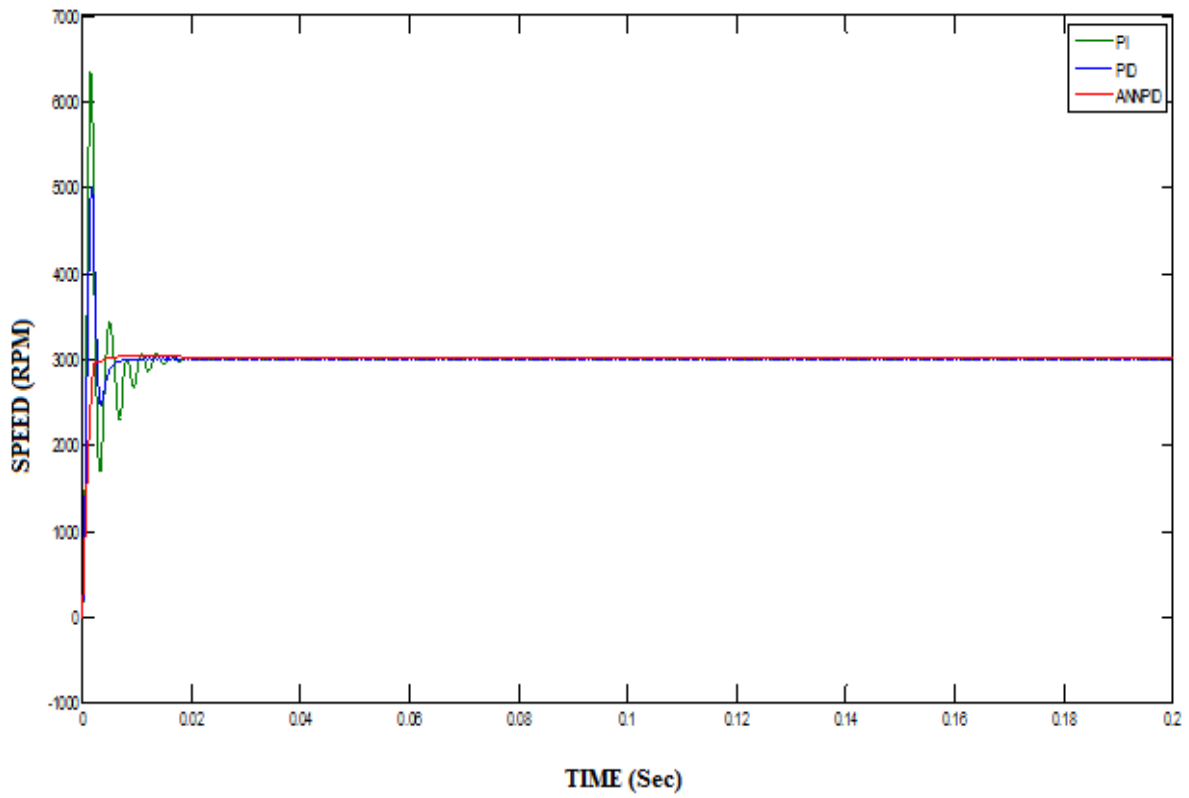
**(3) Speed = 2000 rpm, Load Torque = 4 Nm.**



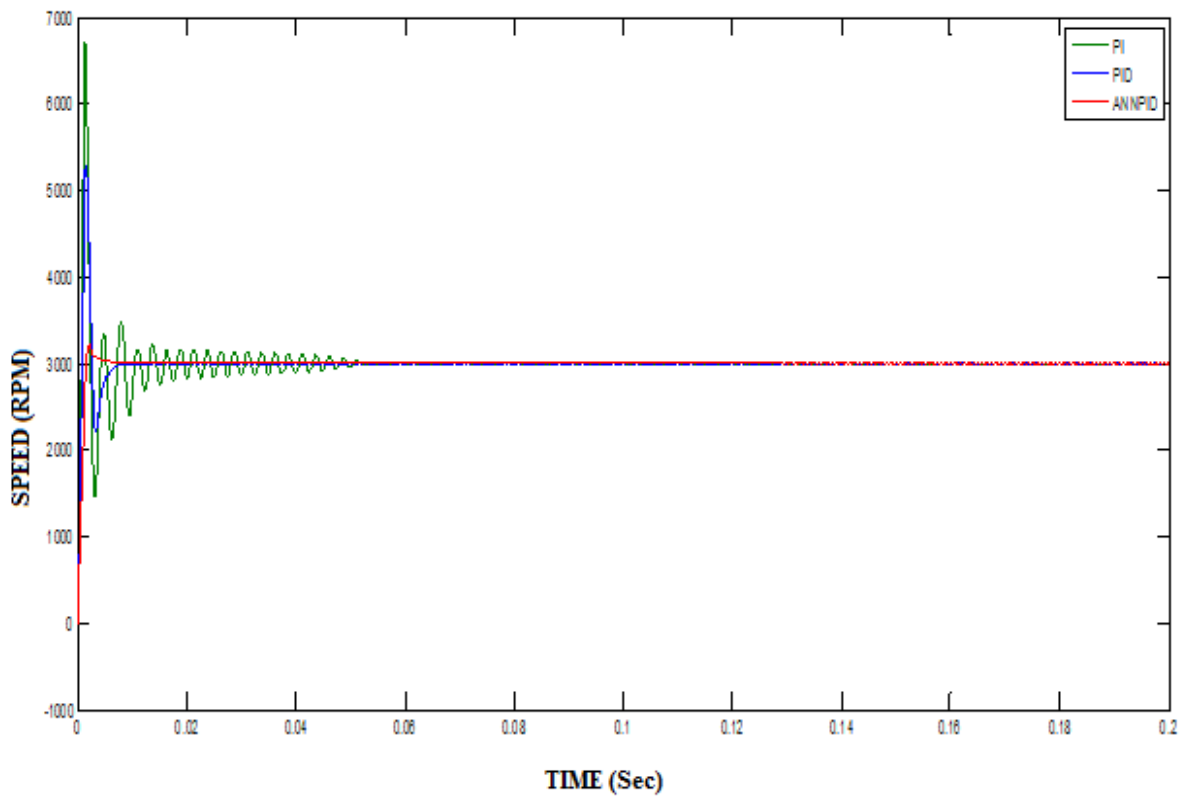
**(4) Speed = 3000 rpm, Load Torque = 0 Nm (No Load)**



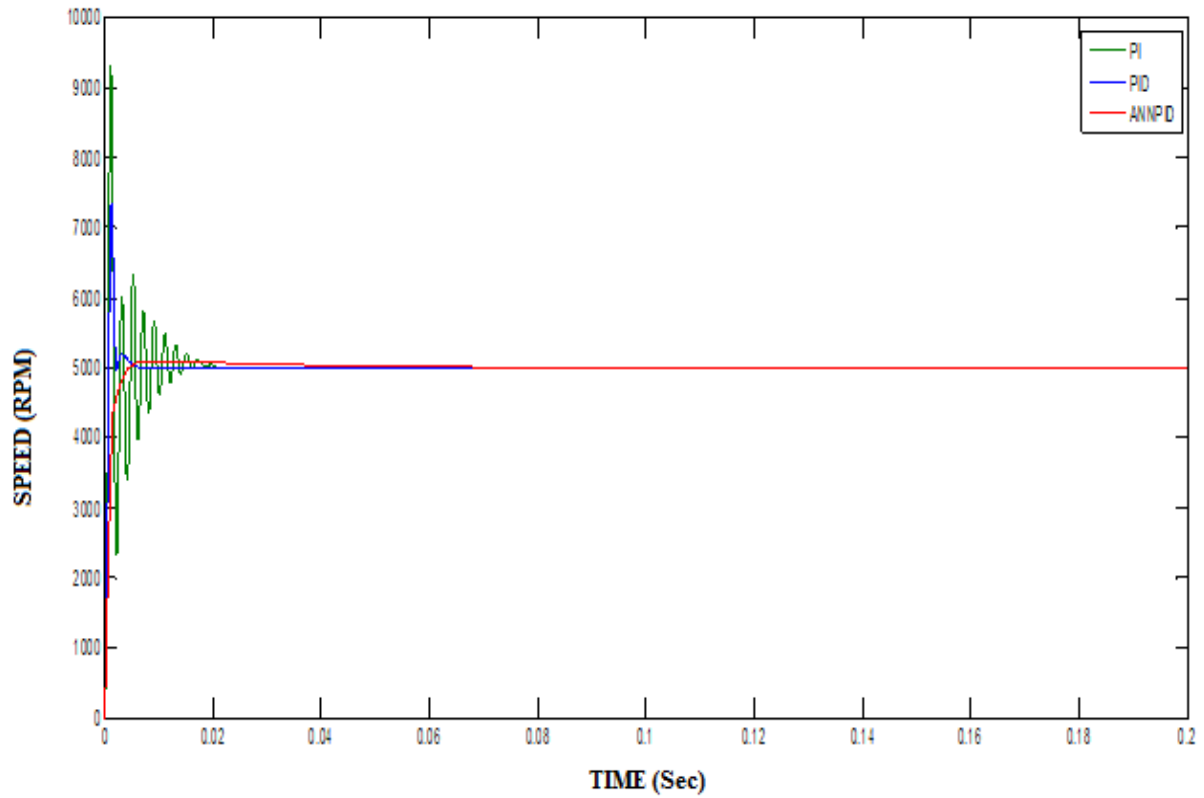
**(5) Speed = 3000 rpm, Load Torque = 1 Nm.**



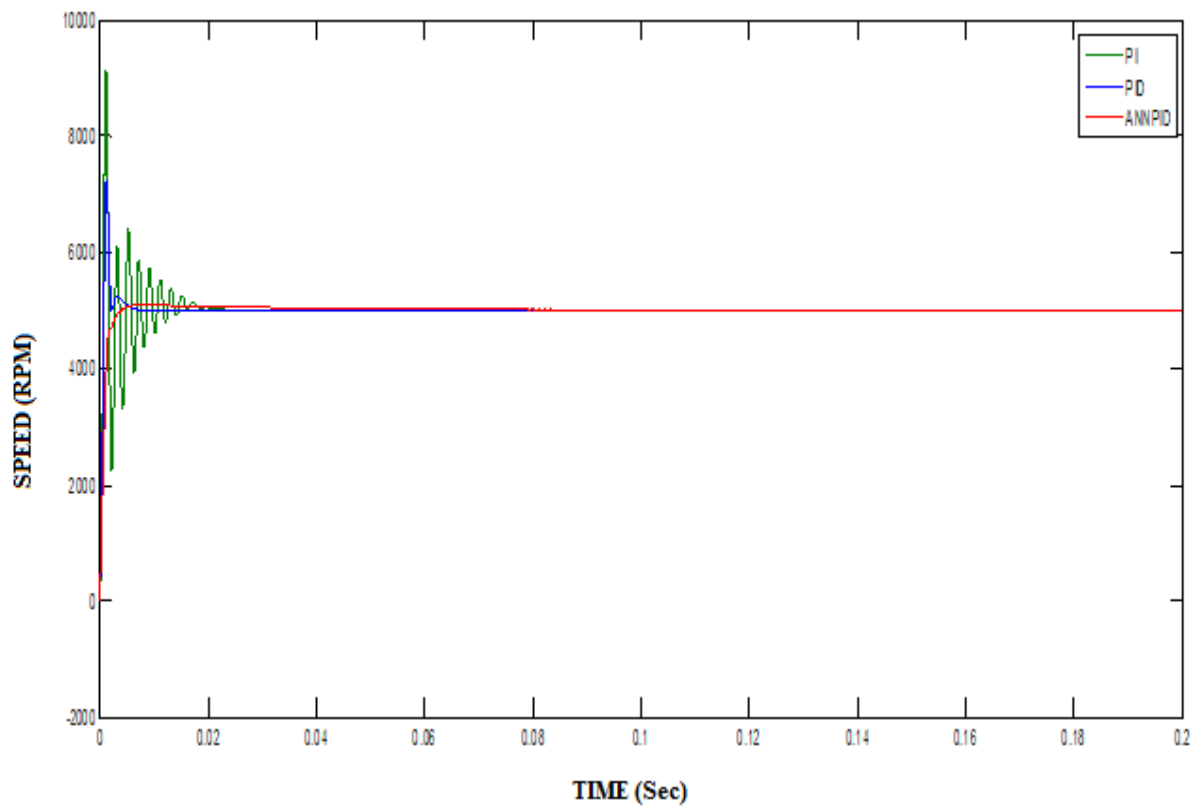
**(6) Speed = 3000 rpm, Load Torque = 4 Nm.**



**(7) Speed = 5000 rpm, Load Torque = 0 Nm (No Load).**



**(8) Speed = 5000 rpm, Load Torque = 4 Nm (Rated load)**



## 5.6 COMPARISON BETWEEN RISE TIME AND MAXIMUM % OVERSHOOT

The performance indices are computed from the results obtained for PI , PID and tuned PID controller and are compared as reported in table 5.2

**Table.5.2: Comparison Between the rise time and Percentage overshoot of ANNPID, PID & PI controllers**

<b>Controllers</b>	<b>Speed (rpm)</b>	<b>Load Torque</b>	<b>Rise Time (sec)</b>	<b>Percentage Overshoot</b>
<b>ANN Based PID Controller</b>	2000	No Load	1.57e-3	7.11
	2000	Rated Load	1.58e-3	6.38
	3000	No Load	1.495e-3	1.5
	3000	Rated Load	1.565e-3	0.672
	5000	No Load	1.785e-3	1.84
	5000	Rated Load	2.22e-3	0.941
<b>Ziegler-Nicholus based PID Controller</b>	2000	No Load	0.615e-3	97.5
	2000	Rated Load	0.755e-3	72.6
	3000	No Load	0.545e-3	77.963
	3000	Rated Load	0.61e-3	66.18
	5000	No Load	0.52e-3	47.07
	5000	Rated Load	0.525e-3	45.8
<b>PI Controller</b>	2000	No Load	0.55e-3	134.76
	2000	Rated Load	0.65e-3	112.39
	3000	No Load	0.45e-3	122.59
	3000	Rated Load	0.49e-3	111.7
	5000	No Load	0.405e-3	86.24
	5000	Rated Load	0.426e-3	82.05

## **CHAPTER 6**

### **CONCLUSIONS & FUTURE SCOPE**

---

#### **6.1 CONCLUSION**

In this thesis simulation of Brushless dc permanent magnet synchronous motor (BLDCM) is carried in MATLAB SIMULINK software, stator current and electromagnetic torque has been analyzed. A Neural Network has been developed using Neural Network Tool box in MATLAB simulation environment. Transient response of speed control of motor is studied using ANN based PID controller, PID controller and PI controller. A comparison has been done at various speeds (2000, 3000, 5000 rpm) at load torques ranging from 0-4N-m of BLDCM with ANN based PID, PID and PI controllers. The performance indices in terms of rise time and percentage overshoot of the controller are computed and compared. Also the rise time and percentage overshoot of these waveforms has been calculated and compared. After this comparison it is concluded that ANN based PID controller is far better controller for the speed control of BLDCM as compare to these conventional controllers, as percentage overshoot of ANN based PID controller is in the permissible limit of 10 percent, very less rise time, no undershoots, only one overshoot and very less settling time of speed waveform is achieved as compared to the conventional controllers.

#### **6.2 FUTURE SCOPE**

- (1) Real time hardware system for BLDCM for speed control using ANN based PID controller can be implemented.
- (2) Speed control of BLDCM could be done using other techniques like Adaptive neuro fuzzy inference system (ANFIS), Fuzzy Logic, GA, PSO, Ant Colony algorithm, etc.

## REFERENCES

---

- [1] Zhang.Y,Akujuobi.C.M, Ali.W.H, Tolliver.C.L, and Shieh.L.S,“ Load disturbance resistance speed controller design for permanent magnet synchronous motor,” *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1198–1207, Aug. 2006.
- [2] Concari.C, Torri.F,“ Sensor less control of BLDC motor at Low Speed based on differential bmf measurement , IEEE, *Conference on energy conversion congress & exposition*, pp. 1772-1777, sept.2010
- [3] Rahman.M.A," Special section on Permanent magnet motor drives; Guest editorial," IEEE, *Trans. on Industrial Electronics*, vol. 43, no. 2, pp. 245, 1996.
- [4] Yu.G.R, Hwang.R.C,“ Optimal PID Speed control of BLDC motor lqr approach”, IEEE, *Conference on system, man & cybernetics*, vol.1, pp. 473-478, Oct.2004.
- [5] Chan.C.C, Jiang.J.Z. Chen.G.H, Wang.X.Y and Chau.K.T,“ A novel poly phase multi pole square wave permanent magnet motor drive for electric vehicles”, IEEE, *Transactions on Industry Applications*, vol. 30, no. 5, pp. 1258-1266, 1994.
- [6] Meireles.M.R.G, Almeida.P.E.M and Simões.M.G,“ A Comprehensive Review for Industrial Applicability of Artificial Neural Networks”, IEEE, *Trans. on Ind. Electron.* vol. 50, no. 3, pp. 585-601, Jun. 2003.
- [7] Weerasooriya.S and Sharkawi. M. A. El," Identification and control of a dc motor using back-propagation neural Network”, IEEE, *Transaction on Energy Conversion*, pp 663-669, December, 1991.
- [8] Pillay. P and Krishnan. R, “Application Characteristics of Permanent Magnet Synchronous and Brushless dc Motors for Servo Drives,” *IEEE Trans. Ind. Appl.*, vol. 27, no.5, pp. 986-996 Sept./Oct. 1991.
- [9] Cheng. M , Chau. K.T, and Chan. C.C, “Design and analysis of a new doubly salient permanent magnet motor,” *IEEE Trans. Magn.*, vol. 37, pp. 3012–3020, July 2001.
- [10] Ganesh.C, Prabhu.M, Rajalakshmi.M, Sumathi.G, Bhola.V, Patnaik.S.K,“ ANN Based PID Controlled Brushless dc Drive System ”, *Proc.of int. conf on AEEE*, 2011
- [11] Chen. Z, Tomita. M, Doki. S, and Okuma. S, “New adaptive sliding observers for position and velocity sensorless controls of brushless DC motors, ” *IEEE Trans. Ind. Electron.*, vol. 47, no. 3, pp. 582-591, June2000.

- [12] Hemanand.T, Rajesh.T, “ Speed Control of BLDC Motor drive employing hard chopping PWM technique using DSP”, *Proceedings of India International Conference of Power Electronics*, 2006.
- [13] Park K.H, Kim T.S, Ahn S.C, Hyun S.D, “Speed control of high-performance brushless dc motor drive by load torque estimation,” *IEEE conference on power electronics*, vol. 4, pp. 1677-1681, June 2003.
- [14] Zhu. G, Dessaint. L.A, Akhrif. O., and Kaddouri. A., “Speed tracking control of a permanent-magnet synchronous motor with state and load torque observer,” *IEEE Trans*, vol. 47, no. 2, pp. 346-355, Apr.2000.
- [15] Daryabegi.E, Sadeghi.H, Lucas.C,” Speed control of BLDC motor using Emotional intelligent controller”, *IEEE, 36<sup>th</sup> annual conference on industrial electronics society*, pp. 2920-2925, Nov.2010.
- [16] Ansari.U, Alam.S, Jafri.S,” Modelling and control of three phase BLDC motor using PID with Genetic algorithm ”, *IEEE, 13<sup>th</sup> international conference on computer modelling and simulation*, pp.189-194, April.2011.
- [17] Rahman.M.A,” On line self tuning ANN based speed control of PM DC motor”, *IEEE, ASME Transactions on Mechatronics*, vol.2, no.3, Sep 1997.
- [18] Zhang.Y, Mingshuang.B, Sina.X, Xlumin.Y,” Adaptive Neural PID Controller for torque control of hybrid electric vehicle”, *IEEE, Conference on computer science & education*, pp. 901-903, Aug.2011.
- [19] Zheng.L,” Neural Network based fault diagnosis and fault tolerant control for BLDC motor”, *IEEE, Conference on power electronics and motion control*, pp.1925-1929, May.2009.
- [20] Mirtalaei.S.M.M, Moghani.J.S, Malebian.K, Abdi.B, ” A novel sensorless control strategy for BLDC motor drive using a Fuzzy logic based Neural network observer”, *IEEE, Symposium on power electronics, electrical drives, automation & motion*, pp. 1491-1496, June.2008.
- [21] Dubey.G.K,”*Fundamentals of Electrical Drives*”, Narsoa publications, ISBN 978-81-7319-428-3.
- [22] Ogata.K,”*Modern Control Engineering*”, Pearson Education, Inc. ISBN 81-7808-579-8.
- [23] Krishnan.R,”*Electric Motor Drives*”, Pearson Education, Inc. ISBN 81-297-0139-1.

- [24] A. Wang, W. Xu and C. Liu, "On-Line PI Self-Tuning Based on Inertia Identification For Permanent Magnet Synchronous Motor Servo System," *Intl. Conf. on Power Electronics and Drives, Systems, PEDS 2009*, Taiwan, pp. 1406–10, Nov.2009
- [25] Hang C-C. K. J. Astrom and W. K. Ho, "Refinements of the Ziegler-Nichols tuning formula", *IEEE Proc. Part D*, vol. 138, No. 2, pp. 111-118, March 1991
- [26] Cai.C, Zhang.H, Liu.J, Gao.Y, "Modelling and simulation of BLDC motor in electric power steering", *IEEE, Asia-pacific conference on power and energy*, pp.1-4, march.2010.
- [27] Hagan, M.T., Demuth, B.T. and Beale, M.H," *Neural Network Design*", PWS Publishing Company, ISBN 0-9717321-0-8.
- [28] Maren, A., Harston, C and Pap, R,"*Handbook of Neural Computing Applications*", Academic Press, Inc., San Diego, ISBN 0-12-546090-2, 1990
- [29] Teng Lin, Lee,"*Neural Fuzzy Systems*", Prentice-Hall, Inc., New Jersey, ISBN 0-13-261413-8.
- [30] V.Shanmuga sundaram and T.Jayabarathi, An investigation of PID tuned ANN controllers using load frequency control in single area power system , *International journal of recent trends in engineering and technology* , Vol.5. no 2. March 2011
- [31] Bose.B,"*Power Electronics & Motor Drives*", Elsevier Inc, ISBN13 978-0-12-088405-6.

## Appendix-A

---

### BLDCM PARAMETERS

	Motor Specifications: 3 phase, 1kW, 3500rpm, 500V	
1	Number of phases	3
2	Back EMF waveform	Trapezoidal
3	Mechanical input	Torque $T_m$
4	Stator phase resistance $R_s$ ( Ohms)	2.875
5	Stator phase inductance $L_s$ (Henry)	8.5e-3
6	Flux linkage established by magnets (V.s)	0.175
7	Voltage constant ( $V_{peak}/rpm$ )	146.6077
8	Torque constant ( $N.m/A_{peak}$ )	1.4
9	Back EMF flat area (degrees)	120
10	Inertia ( $J(kg.m^2)$ )	0.8e-3
11	Friction factor $F$ (N.m-s)	1e-3
12	Pole pairs	4
13	Initial conditions [ $\omega_m$ (rad/sec) $\theta_{m}$ (deg) $i_a, i_b$ (A)]	[0, 0, 0]