JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

ANN Based Speed Observer and Field Oriented Sensorless Controller for Permanent Magnet Synchronous Motor Drive

By

Addis Gashaw Zegeye

This thesis is submitted to School of Graduate Studies of Jimma University in partial fulfillment of the requirements for the degree of Master of Science in Control and Instrumentation Engineering

May, 2020

Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

ANN Based Speed Observer and Field Oriented Sensorless Controller for Permanent Magnet Synchronous Motor Drive

By

Addis Gashaw Zegeye

This thesis is submitted to School of Graduate Studies of Jimma University in partial fulfillment of the requirements for the degree of Master of Science in Control and Instrumentation Engineering

APPROVED BY THE BOARD OF EXAMINERS :

This thesis is approved by faculty of Electrical and Computer Engineering thesis examination members

1.	_____	_____	_____
	Chairman	Signature	Date
2.	_____	_____	_____
	Examiner (internal)	Signature	Date
3.	G/michael Te-ame	[signature]	24/05/20
	Examiner (External)	Signature	Date

Declaration

This thesis titled "ANN Based Speed Observer and Field Oriented Sensorless Controller for Permanent Magnet Synchronous Motor Drive" is based on the results of thesis carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

THESIS SUBMITTED BY:

Addis Gashaw _____ _____

SIGNATURE DATE

APPROVED BY ADVISORES:

ADVISOR: Dr. Kinde Anlay _____ 27/05/2020

SIGNATURE DATE

CO-ADVISOR: _____ _____-

SIGNATURE DATE

Acknowledgements

First and above all, I express gratitude toward God, the almighty for all his blessing throughout my life and for granting me the capacity to continue effectively in my investigations. I am heartily appreciative to everyone of the individuals who supported me in any regard towards the completion of my Master thesis. I owe my deepest appreciation to my advisors Dr. Kinde Anlay and Mrs. Addise Belay who supported me with there well-explained illustrations empowered me to build up a solid understanding of the subject in both practical works as well as the documentation. It is an honor for me to recognize the Jimma University which offered me a scholarship during the entire time of my master studies. Last, however, by no means least, I would like to express my deep gratitude to my lovely parents and friends for their personal support. Thanks to them for always being my main source of motivations for success.

Abstract

Sensorless speed control of Permanent Magnet Synchronous Motor (PMSM) is still in development stage to overcome the existing drawbacks such as speed ripples, torque ripples and poor performance at low speeds with no-load or lightly loaded conditions. In this thesis, sensorless Field Oriented Control (FOC) system which is based on artificial neural network (ANN) implementation is proposed. The control system uses ANN aided Model Reference Adaptive System (MRAS) for speed estimation. Both the speed estimator and the controller of the proposed system use ANN based implementations. The performance of the proposed system is analyzed for speed control of PMSM, for various speed and load torque values. The simulation results of the proposed ANN based speed estimator and controller model demonstrate that ANN aided MRAS based controller of PMSM has the capability to enhance the performance of the system. Specifically, it reduces the overall complexity and initial speed overshoot by 1% as compared to the conventional PI based MRAS controllers employed for PMSM speed control. It has acceptable settling and rise time in its dynamic response.

Keyword: ***PMSM, Sensorless speed control, FOC, MRAS, ANN.***

Contents

List of Figures

List of Tables

Abbreviations

AC	**A**lternating **C**urrent
ANFIS	**A**daptive **N**euro-**F**uzzy **I**nference **S**ystem
ANN	**A**rtificial **N**eural **N**etwork
back-EMF	back **E**lectromotive **F**orce
BLDC	**B**rushless **DC**
DC	**D**irect **C**urrent
EKF	**E**xtended **K** kalman **F**ilter
FIS	**F**uzzy **I**nference **S**ystem
FOC	**F**ield-**O**riented **C**ontrol
IPMSM	**I**nterior **P**ermanent **M**agnet **S**ynchronous **M**otor
MRAC	**M**odel **R**eference **A**daptive **C**ontroller
MRAS	**M**odel **R**eference **A**daptive **S**ystem
PI	**P**roportional **I**ntegral
PMSM	**P**ermanent **M**agnet **S**ynchronous **M**otor
Q-MRAS	**R**eactive **P**ower based **M**odel **R**eference **A**daptive **S**ystem
SMO	**S**liding **M**ode **O**bserver
SPMSM	**S**urface **P**ermanent **M**agnet **S**ynchronous **M**otor
SVM	**S**pace **V**ector **M**odulation
SVPWM	**S**pace **V**ector **P**ulse **W**idth **M**odulation
VC	**V**ector **C**ontrol

Chapter 1

Introduction

The AC electric motors are commonly used in the industry. An AC motor is a machine that converts alternating current into torque, in comparison to the DC motor that produces torque from direct current. Transportation (such as trains or automobiles), washing machines and industrial cranes are some applications for AC motors [1].

There are two main classifications of AC motors: induction and synchronous [1]. The difference is that for an induction motor, currents are induced in the rotor windings whenever the speed of the rotor differs from the speed of the rotating magnetic field generated from the currents in the stator windings. While for synchronous motors, the rotor always rotates at the same speed as the rotating magnetic field. The synchronous speed can be achieved by either injecting current into the rotor or using permanent magnets in the rotor.

Induction motors are the most popular in the industry, mainly because they are rugged, robust and cheap motors available off the shelves but, difficult to control because of its complexity and nonlinear behavior. However, their dominance is challenged by permanent magnet synchronous motors (PMSM), because of their high power density and high deficiency due to reduced rotor losses [2].

A motor that is very similar to the PMSM, and perhaps more used, is the brushless DC (BLDC) motor. They have the same basic construction, consisting of permanent magnets on the rotor and windings on the stator. However, one principal

difference is that the coils in the stator are evenly wound in a BLDC motor, and in the case of PMSM they are wound symmetrically. As a result, the Back-EMF generated in a BLDC motor is trapezoidal and Sinusoidal in PMSM motors. Consequently, the PMSM motors are more efficient and produce less noise and torque ripple however, it requires a more complex control algorithm [3, 4].

The control of the PMSM relative to the DC motor is not a trivial matter. Only after the implementation of the idea of Vector Control (VC) in the early 1970s was a specific control method used for both steady-state and transients [5].

Field Oriented Control (FOC) was the first and most common form of VC. A new VC method called Direct Torque Control (DTC) appeared to be an alternative to FOC later in the mid-eighties. Both FOC and DTC methods achieve decoupled torque and flux control and were first implemented in Induction Motor (IM) drive control. Due to high efficiency, smooth operation at low and high speeds resulting in a wide range of speeds, fast dynamic response, better transient and steady-state performance FOC is chosen in the drive system [3, 5].

Traditionally, some type of sensor is used for speed-dependent applications to measure the speed of the motor and return the value to the controller. The position of the rotor is also needed when using FOC. Extra sensors, however, require additional physical space in the application, and it also adds another cause of system failure. Economic, robust and compact drives are therefore needed; it is of great interest to remove the position or speed sensors. The sensor can be replaced by an estimator that measures the rotor's speed or position mathematically. This is called sensorless control. The advantages of this control include reduced hardware complexity, lower cost, reduced drive size, elimination of the sensor cable, improved noise tolerance, increased reliability and lower maintenance requirements [6, 7].

1.1 Statement of the Problem

To achieve efficient control of a PMSM it is necessary to know the position of the permanent magnet flux. Such expertise has traditionally been gained by measuring the physical position of the motor's shaft. A lot of effort has been put into estimating the flux based on electrical measurements alone to save cost and increase the reliability of the motor drive.

The field of motor speed and position estimation is still in development and exists a lot of literature of this technique with several methods used in applications to overcome the existing drawbacks such as complexity, simplicity, speed ripples, torque ripples and very poor performance at low speeds with no-load or lightly loaded conditions. This thesis will, therefore, focuses on speed estimators to overcome some of the above problems.

1.2 Scope of the Thesis

This thesis will focus on sensorless speed control of PMSM. A mathematical model that represents PMSM is simulated on MATLAB/Simulink 2017a software. After designing the model, FOC is applied for controlling the speed and torque of the motor and simulate to check the ability of the controller applied with PI speed controller then, the sensorless controller is modeled using MRAS as an estimator and the PI controller in FOC as well as in MRAS are replaced by ANN, this also simulated under MATLAB/Simulink 2017a software. After all, the effectiveness of the proposed system will be studied in terms of percentage overshoot, rise time, settling time, and steady-state error for a given reference speed with different load torque.

1.3 Limitation of the Thesis

The proposed system must be implemented on real motors under all the test cases to access its performance in real control scenario but, due to lack of the hardware accessories, the system is modeled and simulated only using MATLAB/Simulink software.

1.4 Objectives of the Thesis

1.4.1 General Objective

The main objective of this thesis is to model ANN aided MRAS-based speed and position estimator to increase the performance of PMSM.

1.4.2 Specific Objectives

The specific objectives of this thesis are:

- To review relevant literature available on the implementation of sensorless speed controller of PMSM.
- To develop the mathematical model of PMSM
- To design and tuning PI controllers in cascaded form
- To design MRAS based estimator based on Hyper-stability theory
- To design ANN based controller
- To test and validate the results

1.5 Methodology

In this thesis, we propose Sensorless Field Oriented Control Based on ANN aided MRAS Speed Observer for PMSM drive. The first task is to review distinctive literature about sensorless speed control systems to build up a solid understanding

of the thesis work have been done. The subsequent task is finding the mathematical model of the PMSM with its respective parameter value, then control the flux and torque by using a vector control method, FOC through the best tuning method for PI controllers. This is followed by transferring the control strategies from sensored to sensorless by designing an observer called model reference adaptive control system. The stability analysis for the observer is based on the Popov hyper-stability criterion. After these the mathematical model and the controllers are simulated on MATLAB/Simulink 2017a, then all PI controllers presented in FOC as well as in the adaptation mechanism of MRAS are replaced by trained artificial neural network and re-simulate again. After all, the comparison of the conventional method and the proposed system is presented.

1.6 Thesis Outline

The rest of the report is organized as follows. Chapter 2 presents different sensorless control methods found in the literature. Different applications for these methods will be discussed and the comparison among these methods is evaluated. Chapter 3 presents system designs and mathematical models that this thesis will build upon. This includes PMSM dynamical equations, Clarke and Park vector transformations, FOC and MRAC. The simulation procedure, simulation model, test cases and simulation results will be presented in Chapter 4. The work will then be concluded and future work or recommendation will be presented in Chapter 5.

Chapter 2

Literature Review

In this chapter, several existing sensorless control methods for PMSMs will be analyzed and discussed. In the analysis, the different methods will be compared through some specific aspects. Such as, how much computational power is needed. Thus, the lower the computational power needed, the better. The ability to estimate the position and speed of the rotor at all speed ranges. The estimation error should be as low as possible and minimum sensitivity to motor parameter variations. In recent years, several pieces of literature have proposed many speed estimation techniques to perform vector-controlled PMSM drive speed-sensorless operation. The approach based on the back electromotive force (EMF) is the most common method for sensorless PMSM control [8]. The calculation is simple and easy thus showing great performance in [9] high-speed control applications. Nevertheless, with this type of method, a well-known problem is that the back-EMF depends on the speed of the rotor. This means that the back-EMF is very small at low speeds and therefore difficult to accurately estimate [9].

Extended Kalman Filter (EKF) is a state observer that estimates the state of a dynamic linear system based on least-square optimization, it finds the optimal state variables values. This usually gives a very good [10] estimate. Besides, EKF serves as a low-pass filter, eliminating input noise effectively, but demanding a high processor. Because multiplications of matrixes occur, the EKF is very costly in computational terms and inefficient in time compared to other approaches.

Sliding Mode Observer(SMO)estimation method is considered simple and performs well in comparison to its non-complex implementation [11]. The main reason the SMO is widely used is due to its robustness against variations in parameters, disturbances, and noise [12]. The SMO is considered a method of back-EMF estimation. This also means that the SMO shares certain features with the back-EMF system, particularly the unsatisfactory low-speed performance[12].

Model reference adaptive system-based technique is implemented in [13], which provides more benefits than other approaches because of its simple hardware requirements and good stability [14]. Depending on the state variable to be estimated, choice of a reference model, selection of adjustable model and choice of adaptive mechanism, the MRAS can be implemented in various ways. A motor variable, e.g. stator current or power, should be chosen as the state variable [13] when using MRAS to estimate the rotor speed and position of a PMSM. Two different models are then used: the reference model calculates the state variable without using the speed or position of the rotor and the adjustable model calculates the same state variable using the speed or position value of the adaptive mechanism given by the rotor.

The choice of adaptation mechanism shows a little more diversity. The simplest and most common method in applications is to tune the estimated speed value with a PI adaptive mechanism [13]. This is essentially a PI controller performing the actual and estimated state variables on a particular expression. By using Popov's hyper-stability theorem [15], this rule can be selected to ensure the estimator's stability. This is one reason why the process of PI adaptation is so common.

Some other solutions to the adaptation process can be used as an attempt to improve the estimation efficiency. In [16] model reference adaptive observer based on fuzzy PI is proposed. The parameters of adaptive PI regulator can be adjusted by fuzzy controller.

Sugeno Fuzzy Inference System (FIS) also used in [17], which uses the error and derivative error between the reference model and the adjustable model to estimate the rotor speed. In [9], a combination of Sugeno FIS and a hybrid learning

algorithm called the adaptive neuro-fuzzy inference system (ANFIS) is used as the adaptation mechanism. To estimate the rotor speed, the mechanism uses the estimated currents from the adjustable model. The error between measured and estimated current is then used to adjust the estimation parameters.

The adaptive mechanisms in [9, 16, 17] improve the estimator accuracy of rotor position and speed for PMSM by model reference adaptive observer. But, data processing in PI and fuzzy controllers is a complex task that requires heavy computation time. The neural network is a non-linear algorithm that can be worked out easily because of its mathematical nature [18].

In [19] Reactive power (Q)-MRAS-based speed, sensorless PMSM drive with adaptive neural network for Performance enhancement at Low Speeds is proposed, this thesis use ANN in the Q-MRAS based speed estimator. In the proposed estimator, the adjustable part is modeled by ANN. This thesis improves existing Q-MRAS-based speed estimator especially at low speeds and low torques but it only focuses on the adjustable model.

The most common criticism against MRAS is its sensitivity to variations in motor parameters [10, 20]. Since the model used includes mostly motor mathematical equations, the accuracy of the parameter is important. However, according to [17], using the motor itself as a reference model can somewhat reduce sensitivity to variations in motor parameters. It makes easier to implement and ensures stability. In studies and applications, this strategy is also well adopted.

In this thesis, ANN-based controller is introduced for both speed controller as well as an adaptive controller to replace the conventional PI controllers to enhance the performance by reducing the complexity and overshoot of sensorless FOC based MRAS speed estimator for low speed at no load or slightly loaded PMSM, it also introduces a simple tuning strategy by reducing the number of tuned controllers from four to two.

Chapter 3

System Design and Mathematical Modeling

A block diagram of sensorless FOC based speed estimator for PMSM is shown in Figure 3.1. The controller works by decoupling currents. Speed observer block differs from sensor-based approaches by removing the use of sensors for rotor information assessment.

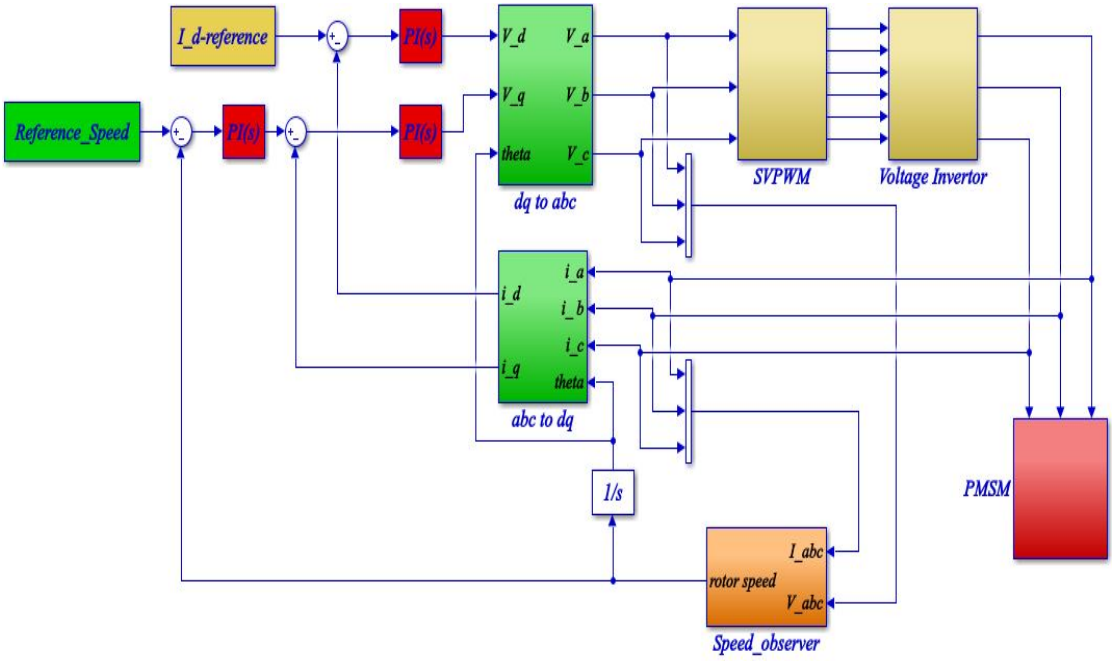


FIGURE 3.1: Conventional Sensorless FOC based speed controller for PMSM.

The estimated speed and angle values are obtained from PMSM's input phase current and voltage. The estimator then provides a rotor position for the transformations as well as rotor speed for the speed control. The model of the proposed system, sensorless Field Oriented Control (FOC) based on ANN aided Model Reference Adaptive System (MRAS) is shown in Figure 3.2. This system uses MRAS as a speed observer with FOC as a vector controller and the controllers are implemented using ANN.

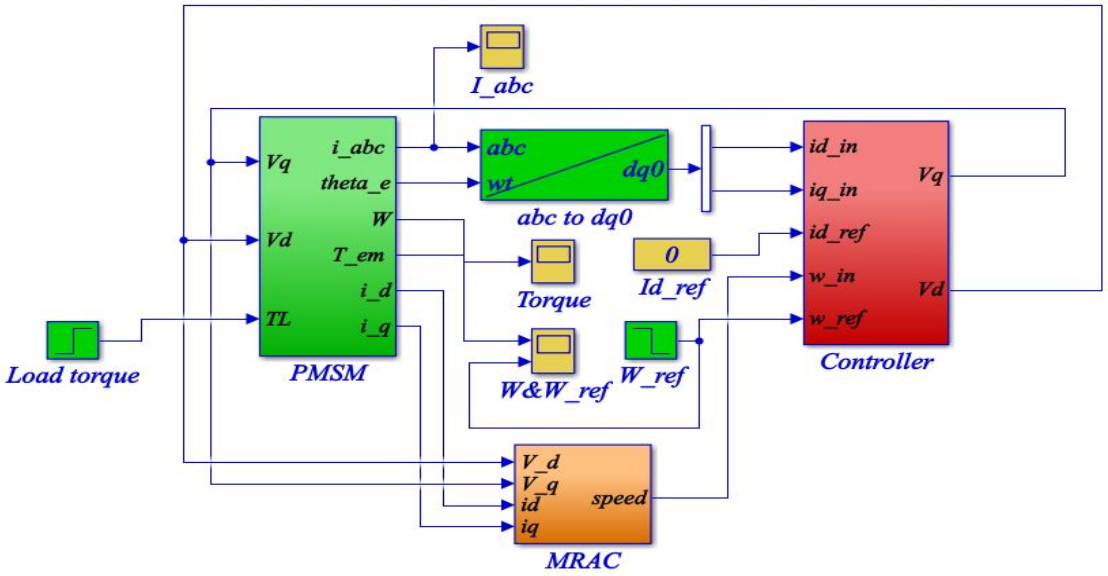
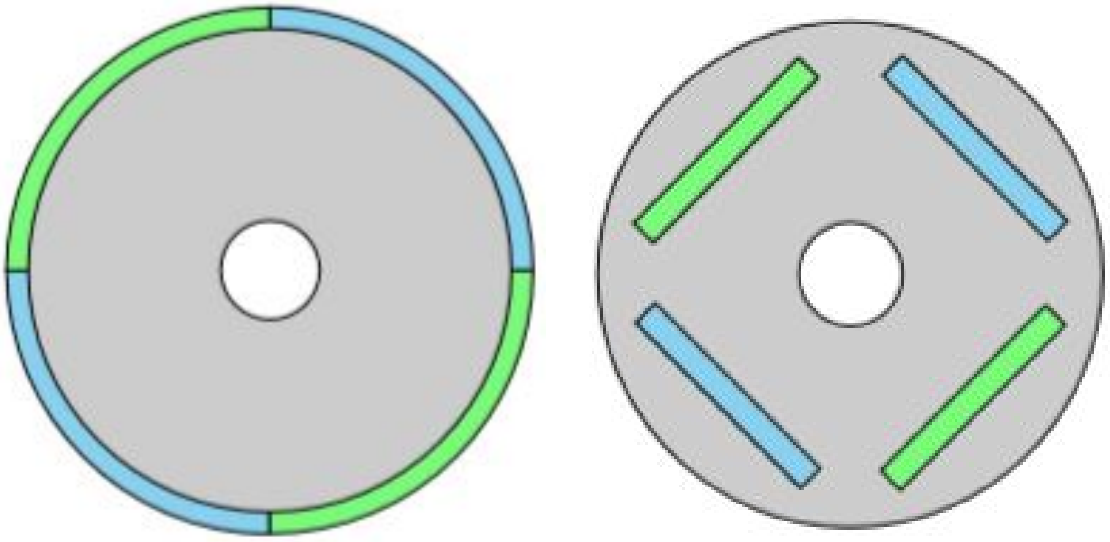


FIGURE 3.2: Sensorless FOC based MRAS speed estimator for PMSM.

3.1 The Model of PMSM

There are two different kinds of PMSMs that differ in distribution of the permanent magnet in the rotor. These are surface permanent magnet synchronous motor (SPMSM) and interior permanent magnet synchronous motor (IPMSM) which are shown in Figure 3.3.

For the SPMSM, the magnets are mounted on the surface of the rotor. The magnets are evenly distributed on the surface so the stator inductances do not depend on the rotor position therefore, it has the same inductance's along any axis through the center of the machine. This type of motor is the easier one to produce and control but the magnets are more exposed to damage [21].



(A) SPMSM (B) IPMSM

FIGURE 3.3: Types of PMSM rotor design [21].

In the IPMSM, the magnets are integrated with the rotor which contributes to more mechanical durability and robustness. This motor is a saliency pole type and the inductance does depend on the rotor position therefore it has unequal inductances along with the d and the q axis. However, it is more expensive to manufacture and more complex to control [21]. In this thesis, an SPMSM motor type is used and it will be the only type studied. Therefore, for the rest of this report, when referring to a PMSM it is the SPMSM.

3.1.1 Clark and Park Transformations

The three-phase motor dynamics can be described from a different point of view by using different reference frames. The most standard frame is the three-phase stationary reference frame, also called abc-frame. This is the physical reference frame of the motor where the three axes a, b and c represent the three electrical phases in the stator. Current and voltage measurements are also done in this reference frame. The abc-frame is thus fixed to the stator. However it is difficult to find the angle and magnitude of the three-phase frame so, it must be converted into a two-phase system.

To overcome the above problem Clarke and Park Transformations and their inverse are introduced. By applying these transformations, we can transform the three phase currents of the stator into the rotating frame of the rotor. Using Clarke Transformation, three-phase quantities are converted into the two-axis orthogonal stationary reference frame. The quantities are still in a fixed reference frame, however, while the reference frame of the rotor is revolving continuously. Transformation of Park transforms these quantities into an orthogonal frame of reference consisting of the axis of direct and quadrature [2]. The three frames of reference are shown in Figure 3.4.

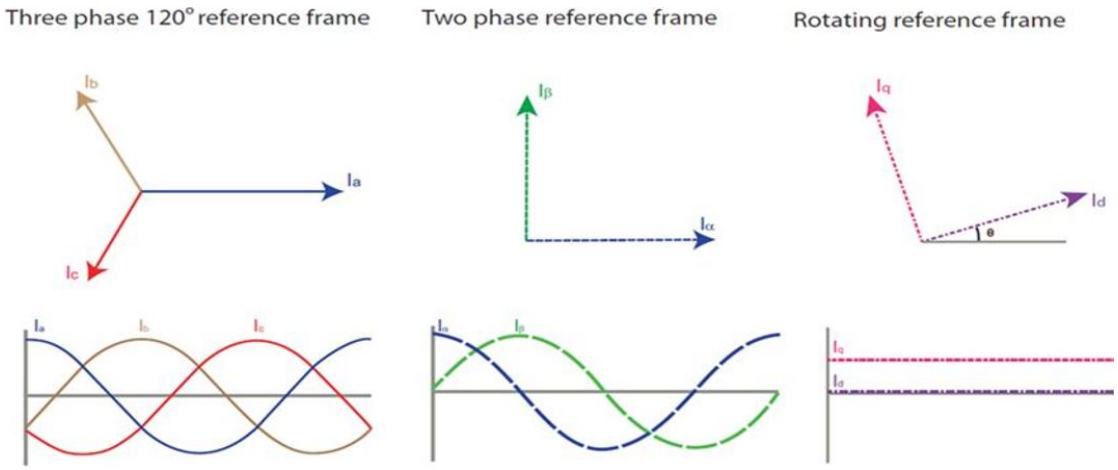


FIGURE 3.4: The three reference frames [2]

The combined transformations of Clarke and Park can be written as:

$$\begin{bmatrix} I_d \\ I_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \sin(\theta) & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{3.1}$$

where I_d and I_q are the direct axis and quadrature axis respectively, θ is the angular position of the rotor. Likewise, the inverse transformation is given by:

$$\begin{bmatrix} I_a \\ I_b \\ I_b \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \cos(\theta - \frac{2\pi}{3}) & -\sin(\theta - \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} I_d \\ I_q \end{bmatrix} \tag{3.2}$$

PMSM can be modeled by using one of the three reference frames. Therefore, it can be modeled in dq, $\alpha\beta$ or abc reference frames.

3.1.2 PMSM model in abc-frame

The three-phase stationary frame representation of a PMSM is given by [7].

$$V_{abc} = R_s i_{abc} + \frac{d}{dt}\phi_{abc} \tag{3.3}$$

where $V_{abc} = \begin{bmatrix} V_a & V_b & V_c \end{bmatrix}^T$ is the voltages of each stator phase; R_s is the stator resistance; $i_{abc} = \begin{bmatrix} i_a & i_b & i_c \end{bmatrix}^T$ is the current in the stator windings in respective phase and $\phi_{abc} = \begin{bmatrix} \phi_a & \phi_b & \phi_c \end{bmatrix}^T$ is the stator fluxes, which are given as

$$\phi_{abc} = L_{ss} i_{abc} + \phi_{rabc}. \tag{3.4}$$

The first term is the flux produced by the currents in the stator winding's; L_{ss} is the stator inductance matrix which depends on the type of machine. Since this thesis only deals with the SPMSM, the inductance of the d-axis is equal to the inductance in the q-axis. This means that constant value can be used for the stator inductances in the transformed representations of the machine [7]. The second component, ϕ_{rabc}, is the flux induced in each stator phase by the permanent magnets in the rotor. It has a sinusoidal waveform and depends on the rotor position. Equation (3.3) can be rewritten as

$$V_{abc} = R_s i_{abc} + \omega_e \frac{d}{d\theta_e}\phi_{abc} \tag{3.5}$$

Both electrical speed and position has the same relation to their mechanical counterparts, see Equations (3.6) and (3.7).

$$\omega_e = P\omega_r \tag{3.6}$$

$$\theta_e = P\theta_r \tag{3.7}$$

where ω_r is the mechanical speed of the rotor, θ_r is the mechanical position of the rotor and P is the number of pole pairs of the magnets in the rotor.

In this thesis we use the FOC method, the principle with field-oriented control is that it controls the motor current in dq-frame, instead of directly in abc-frame. Therefore for simplicity, we design PMSM in dq frame.

3.1.3 PMSM Model in dq-frame

By using (3.1), (3.2) and (3.5), the electrical and mechanical equations of the PMSM in the rotor reference frame (d-q) are as follows and the equations are mapped in MATLAB Simulink as shown in Figure 3.5. Assuming that there is no core loss, constant rotor flux, and no damping winding [5]. The current dynamics of the PMSM in dq-coordinates can be expressed as

$$\begin{cases} \frac{d}{dt}I_d = \frac{-R_s}{L_d}I_d + \omega_r\frac{L_q}{L_d}I_q + V_d \\ \\ \frac{d}{dt}I_q = \frac{-R_s}{L_q}I_q - \omega_r\frac{L_d}{L_q}I_d - \frac{1}{L_q}\omega_r\lambda_{af} + V_q \end{cases} \tag{3.8}$$

where:I_d is direct axis stator current in ampere (A), I_q is quadrature axis stator current in ampere(A), R_s is stator resistance in ohms (Ω), L_d is direct axis inductance in Henry (H), L_q is quadrature axis inductance in Henry (H), J is moment of inertia in Kgm^2, B is friction vicious gain in Nm/rad/sec, P = number of poles, λ_{af} is rotor flux constant in V/rad/sec , ω_e is rotor's electrical speed in rad/sec, T_L is load torque in Nm and T_e is electromagnetic torque in Nm.

Equation (3.8) can also be represented in state space form with state vector X^T = $[I_d\ I_q]$ is

$$\begin{aligned} \dot{X} &= AX + Bu \\ \\ Y &= CX \end{aligned} \tag{3.9}$$

With coefficients:-

$$A = \begin{bmatrix} \frac{-R_s}{L_s} & \omega_e \\ -\omega_e & \frac{-R_s}{L_s} \end{bmatrix}, Bu = \begin{bmatrix} \frac{V_d}{L_s} \\ \frac{V_q - \lambda_{af}\omega_e}{L_s} \end{bmatrix} \text{ and } C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since the motor is a surface permanent magnet synchronous motor (SPMSM), all phase resistance and inductance were assumed to be equal to each other and equal to the two-axis dq frame resistance and inductance. Therefore, $R_d = R_q = R_s$ and $L_d = L_q = L_s$ The mechanical equations can be written as

$$T_e = \frac{3}{2}P(\lambda_{af} I_q + (L_d - L_q) I_d I_q) \tag{3.10}$$

$$\omega_r = \frac{P}{2J} \int (T_e - T_L \frac{2B\omega_e}{P}) dt \tag{3.11}$$

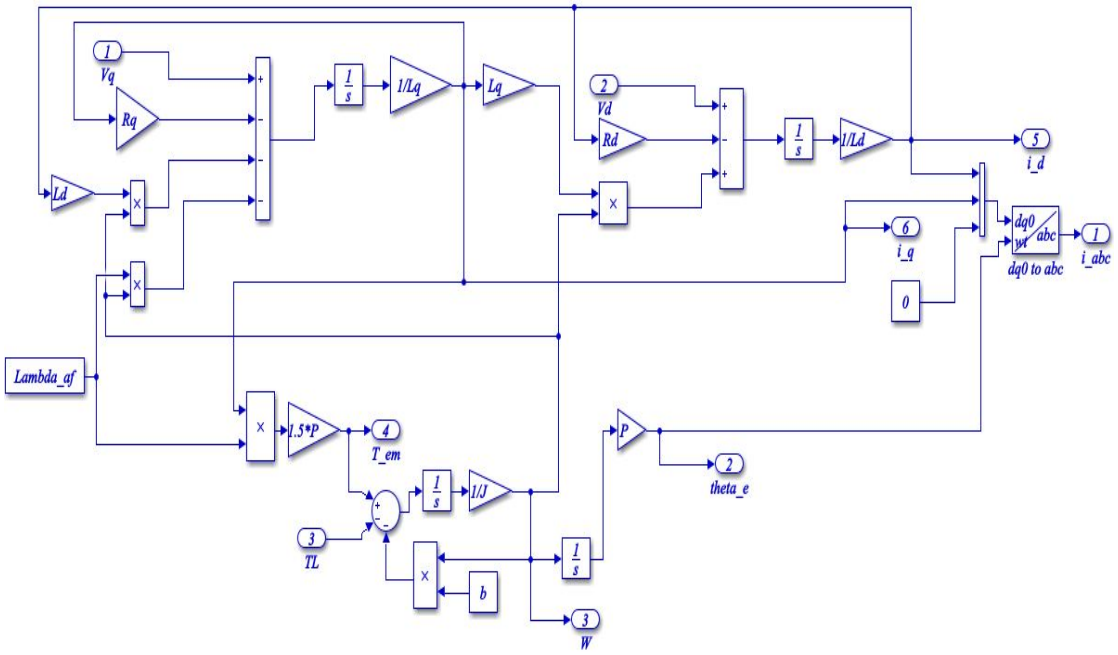


FIGURE 3.5: PMSM model in dq-frame

3.2 Controller Modeling

3.2.1 Vector Control Method

There is an optimal direction for the net stator field for any rotor position that maximizes the torque produced [1]. If the net stator field is in the same direction as the rotor field, then no torque will be generated. This is because of the forces produced by the field interaction act in line with the rotor's axis of rotation and

only result in motor bearings being compressed rather than rotated. If the stator field is perpendicular to the field of the rotor, the maximum torque will be produced.

Any stator field can be represented as a result of two vector components, the orthogonal (quadrature) component that produces torque, and the parallel (direct) component that produces unwanted heat and compression forces. An optimal drive should, therefore, be aimed at minimizing the parallel portion while maximizing the quadrature component. This method is known as vector control [1].

3.2.2 Field-Oriented Control

Like induction motor, it is possible to achieve a decoupled control of the torque and flux magnitudes in the PMSM. This is achieved using the d-q transformation separating the d and q components of the stator current responsible for the production of flux and torque respectively. There is no need to produce flux through the I_d current due to the presence of the permanent magnet's constant flux, and this current can be held to a zero value, which in effect decreases the stator current and improves the drive's performance [22].

As shown in Figure 3.1, the control system is divided into two loops: the d loop controlling the flux; and the q-loop controlling the velocity and torque. The d-loop executes I_d control with an existing PI regulator. The reference value can be set to zero for this loop. In a cascade, the q-loops are linked. By controlling I_q with a current PI regulator, the internal loop regulates the torque.

3.2.2.1 Field-Oriented Control Design

As shown in the Figure 3.6, current controllers and the speed controller in the FOC control system have to be tuned correctly to get a good system response. The tuning process used is well adopted by the industry and is specified by [23]. The basic FOC has three PI controllers and therefore six separate controller gains that need to be tuned.

This tuning process is aimed at reducing the number of parameters to be tuned to two: δ and τ. First, using the series PI topology for the current controllers, the proportional and integral gains are given by the constant values in (3.12) and (3.13), respectively.

$$IK_P = L_S.BWc \tag{3.12}$$

$$IK_i = \frac{R_S}{L_S} \tag{3.13}$$

BWc is the current controller's bandwidth. All current controllers use the same values. The speed controller depends on the δ and τ tuning parameters. δ measures how fast the system is and τ shows the damping of the system. Equations (3.14) and (3.15), respectively, give the speed controller's proportional and integral gains.

$$SK_P = \frac{1}{\delta C \tau} \tag{3.14}$$

$$SK_i = \frac{1}{\delta^2 \tau} \tag{3.15}$$

In (3.14), C is a combination of the motor and load parameters and is given by

$$C = \frac{3P\lambda_{af}}{4J} \tag{3.16}$$

To tune the FOC the parameters δ and τ are chosen such that the system response is satisfactory. This is done by try and error method. In this thesis the conventional PI controller is replaced by ANN based controller as shown in Figure. 3.6. The error between the reference speed (w_ref) and the estimated speed (w_in) from MRAS is given to ANN controller, then the output of ANN is compared to PMSM quadrature current(iq_in) after that it passes through PI current controller to produce quadrature voltage(Vq) for PMSM. In the second loop, the difference between reference direct current (id_ref) and PMSM direct current (iq_in) is passed through PI current controller to produce direct voltage (Vd) for PMSM.

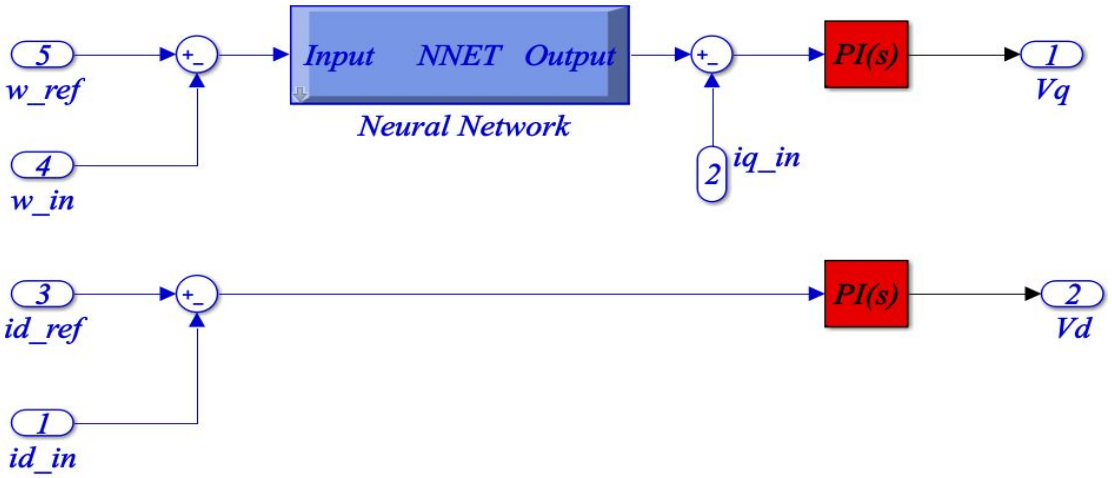


FIGURE 3.6: Controller scheme for FOC based PMSM

3.2.3 Model Reference Adaptive System

The basics of MRAS are to use two independent models: One reference model which is independent on the variable to be estimated and another one is the adjustable model which is dependent on the variable to be estimated. The two models use different sets of inputs to calculate the same state variables which are in turn fed to a certain adaption mechanism. The adaption mechanism uses the difference between the two signals to tune the estimated variable and feed it back to the adjustable model. The estimated value will this way be driven to its true

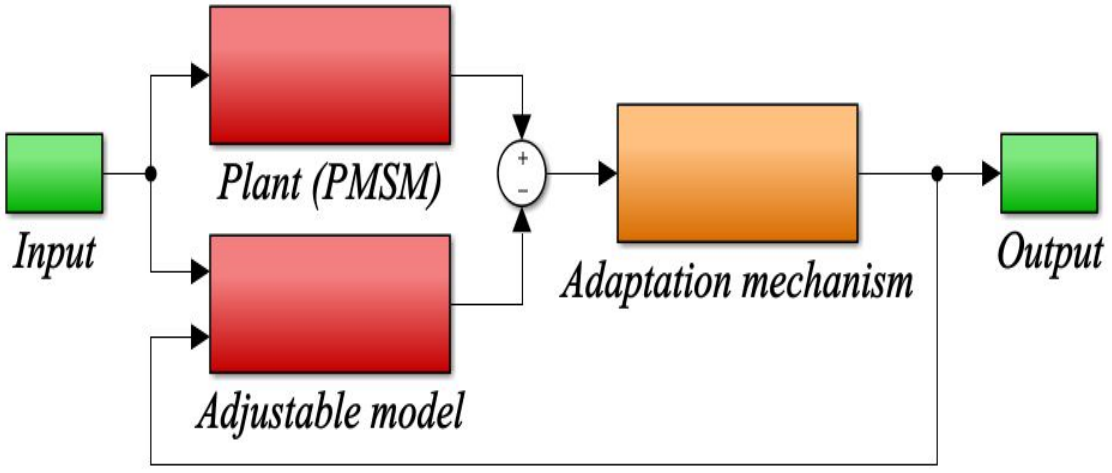


FIGURE 3.7: Model refrence addaptive system

3.2.3.1 MRAS Estimation Model

In the case of estimating the PMSM's rotor speed and position the usual approach is to allow the motor to serve as a reference model to improve robustness against parameter uncertainties.

The measured motor currents are then compared to the currents of the adjustable model; the adjustable model consists of the electrical motor equation provided with the same voltage values as the real motor. The rotor speed is also included in the electrical motor equations. The adaptation mechanism modifies the speed value of the adjustable model to minimize the current estimation error, given the different values for the current.

The adjustable model will be the PMSM electrical equations in the dq-frame from (3.9) using estimated values for the currents and the rotor electrical speed. The adjustable model is given in (3.17).

$$\begin{aligned} \dot{\widehat{X}} &= \widehat{A}\widehat{X} + \widehat{B}u \\ Y &= C\widehat{X} \end{aligned} \tag{3.17}$$

With coefficients:-

$$\widehat{A} = \begin{bmatrix} \frac{-R_s}{L_s} & \widehat{\omega_e} \\ -\widehat{\omega_e} & \frac{-R_s}{L_s} \end{bmatrix}, \widehat{B}u = \begin{bmatrix} \frac{V_d}{L_s} \\ \frac{V_q - \lambda_{af}\widehat{\omega_e}}{L_s} \end{bmatrix} \; and \; C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The estimated state vector $\widehat{X} = \begin{bmatrix} \widehat{i_d} & \widehat{i_q} \end{bmatrix}^T$ consists of the estimated currents and $\widehat{\omega_e}$ is the estimated speed. The adaptive mechanism will be the commonly used PI-controller. This is because it is simple to implement while still performing well. Also because the stability of the estimator can be assured [24]. Given the measured state vector $X = \begin{bmatrix} i_d & i_q \end{bmatrix}^T$, the generalized error $\epsilon = \widehat{X} - X$ can be introduced. By analyzing the generalized error together with the Popov hyper-stability theorem [24], which is described in detail in Appendix A. The transfer

function of the adaptive PI-mechanism can be derived as [25].

$$\widehat{\omega_e} = \left(MK_p + \frac{MK_i}{S}\right)\left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_{af}}{L_s}) - (\widehat{i_d} - i_d)i_q\right) + \widehat{\omega_e}(0) \tag{3.18}$$

By integrating the speed, the position of the rotor is obtained.

$$\widehat{\theta_e} = \int (\widehat{\omega_e})dt \tag{3.19}$$

A PI controller is tuned using many methods. If a plant mathematical model can be derived, then different design techniques can be applied to determine controller parameters that will satisfy the closed-loop system's transient and steady-state requirements. However, if the plant is so complex that it can not easily obtain its mathematical model, then an analytical approach to PI controller design is not feasible. Then, we have to turn to experimental approaches to PI controller tuning [26].

The best tuning method for the PID controller was given by Ziegler and Nichols, which was now accepted as standard technique in control systems practice [26]. In this thesis, the controller gains MK_p and MK_i are determined by using the Ziegler-Nicholas method. The general formula for the controller of Ziegler-Nicholas tuning rule for u(t) as input, y(t) as output and e(t) as the error is shown in equation (3.20).

$$u(t) = K_p e(t) + \overbrace{\frac{K_p}{T_i}}^{K_i}\int_0^t e(t)d\tau + \overbrace{K_p T_d}^{K_d}\frac{de(t)}{dt} \tag{3.20}$$

Where:-K_p, K_i, K_d are proportional, integral and derivative constants respectively. There are some steps to determine the controller parameters under Ziegler-Nicholas method [26].

1. Start with $K_p = small$ and set other parameters to zero. Hence $K_i = K_d = 0$.

2. Increase K_p from 0 to some critical value $K_p = K_{cr}$ at which oscillations occur. If it does not occur then other method has to be applied.

3. Record the ultimate or critical period T_{cr} of the oscillation.

4. Finally find the value of K_p, K_i, K_d from the Ziegler Nicholas tuning formula which is shown in Table 3.1.

TABLE 3.1: Ziegler-Nichols Tuning Formula [27]

Controller Type	K_p	T_i	T_d
P	$0.5K_{cr}$		
PI	$0.4K_{cr}$	$0.8T_{cr}$	
PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.12T_{cr}$

In this thesis we have done all the procedures above and we get the following values from the simulated Figure 3.8.

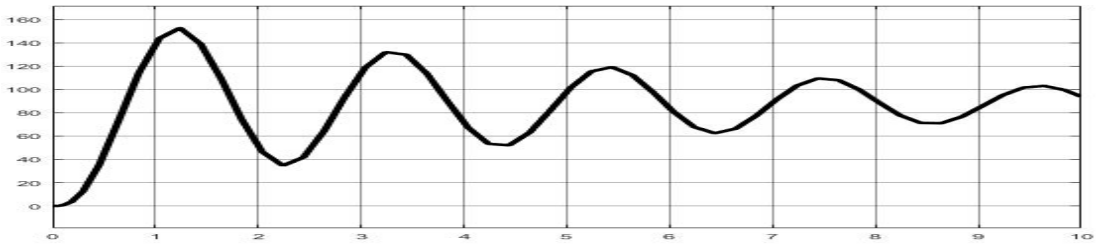


FIGURE 3.8: Oscilation responce at critical value

The critical value, K_{cr} is 1.2 and the critical time, T_{cr} is 2 seconds. Based on equation (3.20) and Table 3.1, we can calculate MK_p and MK_i. Therefore,$MK_p = 0.4K_{cr} = 0.48$ and $MK_i = 0.8T_{cr} = 1.6$. The overall model reference adaptive system is shown in Figure 3.9. In this thesis we replace conventional PI controller by neural network controller as an adaptation mechanism.

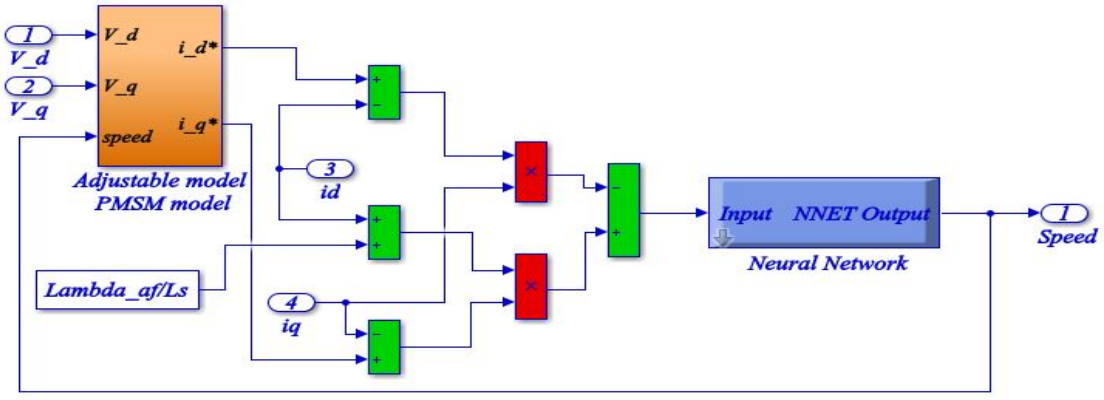


FIGURE 3.9: ANN aided MRAS controller

3.2.4 ANN Controller

The artificial neural network architecture is inspired by the features of biological neurons. A natural brain is capable of learning new things, adapting to new and changing environment. The brain has the most amazing ability to analyze imperfect and vague, blurry information and make out of it its own decision. For example, we can read the handwriting of others even though the way they write may be completely different from the way we write. Even from a blurry photo we can recognize a known person [28, 29].

Processing Units: a neural network's fundamental processing element is a neuron. The main tasks associated with processing units are to acquire input from neighbors providing incoming activation, compute output, and sending output to their neighbors. Such a system is inherently parallel since many processing units can concurrently carry out their computations. In a neural network the processing units can be classified as one of three types [30]:

1. **Input processing units**: which receive input from external sources, compute their activation level, compute their output as a function of an activation level, and transmitting this information to the rest of the network.

2. **Output processing units:** which calculates and broadcasts their output to external receivers upon receipt of input from the rest of the network, or feeds their output back to the network input layer for further processing.

3. **Hidden processing units:** which only receive input from, and broadcast their computed output to, processing units within the network

Weighting Factors: a neuron usually receives many simultaneous inputs. Each input has its relative weight which determines the intensity of the input signal as registered by the artificial neuron. They are a measure of the connection strength of an input. These strengths can be modified in response to various training sets and according to a network's specific topology or through its learning.

Summation (or Net) Function: The net function determines how the network inputs are combined inside the neuron. In Figure 3.10, a weighted linear combination is adopted. The amount of information about the input that is required to solve a problem is stored in the form of weights. Each signal is multiplied with an associated weight $w_1, w_2, w_3, ..., w_n$ before it is applied to the summing block. In addition, the artificial neuron has a bias term w_0, a threshold value "θ" that has to be reached or extended for the neuron to produce a signal, a linear or a nonlinear function "F" that acts on the produced signal "net" and an output "y" after this function. It should be noted that the input to the bias neuron in Figure 3.10 is assumed to be 1.

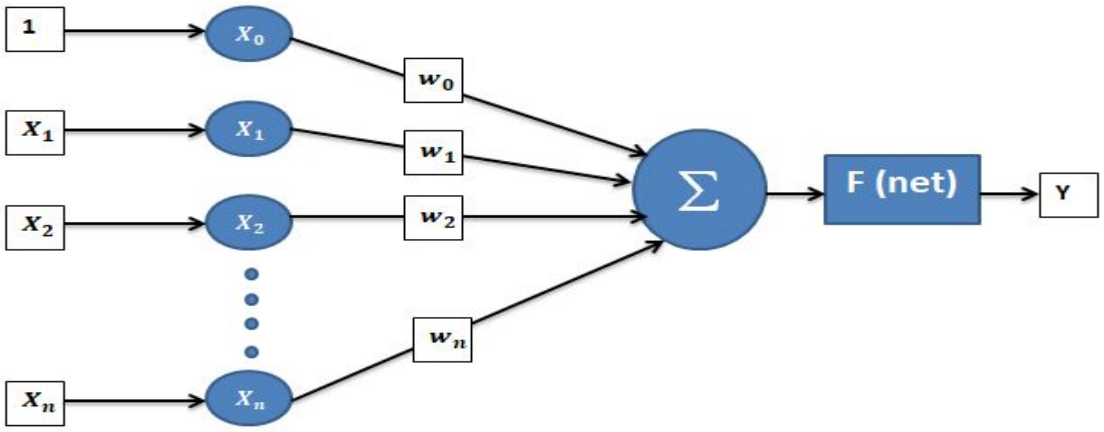


FIGURE 3.10: Basic neural model [28]

The following relation describes the transfer function of the basic neuron model:

$$net = w_0 + \sum_{k=1}^{n} w_k x_k \tag{3.21}$$

$$Y = F(net) \tag{3.22}$$

Activation Function: The purpose of the linear or nonlinear activation function is to ensure that the neuron's response is bounded. Further, to achieve the advantages of multilayer nets compared with the limited capabilities of single-layer networks, nonlinear functions are used, depending upon the paradigm and the algorithm used for training the network [31].

For each layer, you can define one specific activation function. There is no appropriate rule theoretically for defining the activation function of the various layers. Although many studies present different functions for the individual layers, some do have the same function for the input, hidden and output layers. The most commonly used activation functions are summarized in Table 3.2.

TABLE 3.2: Activation functions [31]

Activation Function	Formula	MATLAB function
Linear	$f(x) = x$	purelin
Logistic(Sigmoid)	$f(x) = \frac{1}{1+e^{-x}}$	logsig
Hyperbolic Tangent	$\frac{e^{x}-e^{-x}}{e^{-x}+e^{x}}$	tansig
Sinusoidal	Weight	sin

3.2.4.1 Artificial Neural Network Architectures

The organization of neurons into layers and the patterns of connections within and between layers are called net architecture. A typical neural network is constructed by layers. There is an input layer of source nodes and an output layer of neurons within a single-layered network. A multi-layer network has, besides one or more hidden layers of hidden neurons. More hidden neurons enhance the ability of the network to extract higher-order statistics from input data.

The signal flow, and the connection between neurons, will define the net type. Hence, ANNs have two architecture types, feed-forward networks, and Recurrent or Feed-Back Networks.

Figure 3.11, shows feed-forward networks, the signal flow is from the input to the output units, from the input nodes, through the hidden nodes (if any) and to the output nodes. The network is without cycles or loops. Three types of Feed-Forward networks are single layer perceptron, multi-layer perceptron (MLP), and radial basis function nets.

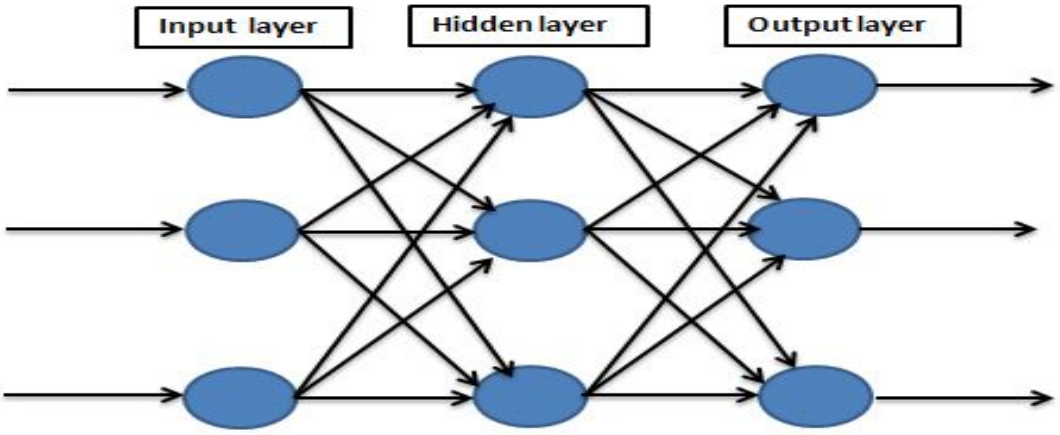


FIGURE 3.11: Feed-Forward Network

3.2.4.2 Training an Artificial Neural Network

A good set of weights is required to approximate a given target function. The problem is that altering one weight will possibly alter the function's output over the entire input space, so it's not as straightforward as using a grid-based approximation. One possible solution is to minimize an error function which measures just how bad an approximation is. The process of finding weights that minimize the error function is called training or learning by artificial intelligence researchers. Learning or training method can be either supervised training or unsupervised training.

3.2.4.3 Training Algorithms

In artificial neural networks, the Training algorithm refers to the process of altering the weights of connections between the nodes of a given network to find the error by comparing the network's output value with the target value and then reducing the difference (error) by altering the weights. There are many different training algorithms for MLP. The backpropagation algorithm is one of the most popular algorithms for training a network due to its success from both the simplicity and applicability viewpoints [32].

The algorithm is built into two stages: the stage of training and the stage of recall. The weights of the network are initialized randomly during the training

phase. Then the network output is measured and correlated with the desired value. Next, the network's training error is measured and used to adjust the output layer weights. Similarly, the network error is also propagated backward and used to update the weights of the previous layers.

Figure 3.12 shows how the error values are generated and propagated to adjust the weights of the network. In the recall phase, only the feed-forward computations are used to assign weights from the training phase. The feed-forward process is used in both phases of recall as well as preparation.

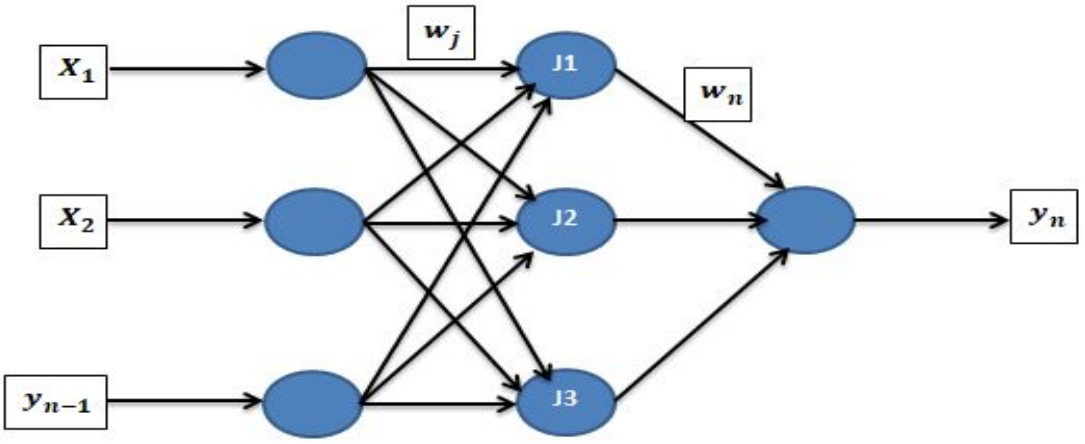


FIGURE 3.12: Back propagation algorithm [32]

The training phase will be terminated when the error value is less than the minimum set value provided by the designer. One disadvantage of the Backpropagation algorithm is that the training phase is very time-consuming [32].

The network with the final weights resulting from the training process will be used during the recall phase. Therefore, the output will be determined using both linear calculation and nonlinear activation functions for each input pattern in this step. An important advantage of this method is that in the recall step, it gives a very fast network [32].

In this thesis, we replace the PI speed controller in the field-oriented controller as well as PI adaptive controller in the model reference adaptive controller to enhance estimator performance at low speed with no-load or full load conditions.

A neural network controller, which is trained to replace the conventional PI speed controller is shown in Figure 3.13. To train neural network controller we have used speed error as input and a vector controlled PMSM drive as a plant for FOC and the current error used as the input for MRAS adaptive neural network controller with the estimated speed as target data.

The main advantage of implementing the controller with the neural network controller is to simplify the control implementation by reducing the complexity. This is because the ANN controller is a digital controller as compared to the conventional PI controller which is analog and it needs additional hardware [33].

The ANN have the advantages of that they can be implemented in parallel, which gives relatively fast computation [34]. Another advantage is that the neural network has the generalization property and the neural controller's control surface is smooth [35].

In addition to the above benefits, the tuning effort of an AI based system is less than that of a conventional PI system. Such a system leads to reduced development time [33].

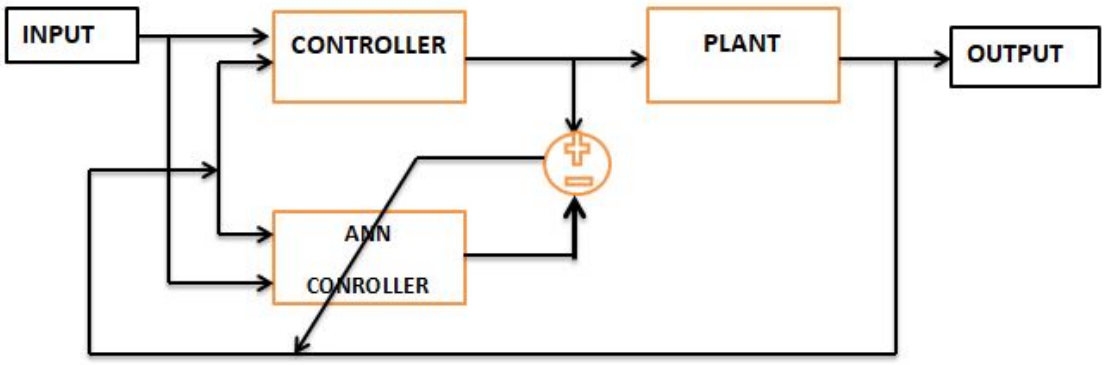


FIGURE 3.13: Training of neural network controller [33].

Chapter 4

Simulation Results and Discussion

4.1 Experimental Setup

A simulation is a good and effective way to see the overall performance of a system. The results from the simulation should mirror the results of the real hardware. The simulations are made in Simulink. For each method connected to the PMSM, i.e conventional PI and ANN controllers, the simulation will go through a few test cases that represent different operating situations of the application. The methods will then be compared based on these test cases.

4.1.1 Motor Parameters:

To simulate the PMSM model, parameters had to be either measured or estimated through appropriate calculations. The phase and synchronous resistance and inductance of the machine, the moment of inertia, the viscous friction coefficient, the number of the poles, the rated current, the rated torque and the rotor flux aligned with the d-axis required for simulation can be measured or calculated [36]. PMSM parameters used in this simulation are shown in Table 4.1.

TABLE 4.1: PMSM parameters [37]

Parameter	Symbol	Value	SI unit
Reference Speed	S	1000	rpm
Load Torque	T_L	2	Nm
Stator Resistance	R_s	2.875	Ohms (ω)
d,q-axis Inductance	L_d,L_q	8.5e-6	Henry(H)
Moment of Inertia	J	0.0010	Kgm^2
Friction Vicious Gain	B	3.8e-4	Nm/rad/sec
Rotor Flux Constant	λ_{af}	0.0012	weber(Wb)
Number of Poles	P	4	nos

4.1.2 ANN Training:

A three-layer neural feed-forward network is built with one input neuron and one output neuron. For the input- and output-layer neurons, the activation function is log sigmoids and purelin, respectively. The network with 15 hidden layer neurons is trained for the given set of inputs and desired outputs. A supervised back propagation training algorithm is used [38], and the neural network is trained until an error goal of $5e^{-8}$ is reached. Training data is obtained from simulating MRAS speed estimator for PMSM with PI controllers based on sensorless field oriented controller. First the ANN controller in the adaptation of MRAS is trained, then ANN speed controller is trained after replacement of PI adaptive controller by ANN adaptive controller.

4.2 Simulation Result

The simulation of the proposed PMSM speed controller model is performed in various load situations. Here we present two cases, one in which the motor is started on with no load that remains on no load condition throughout the operation, and the other in which the load applied is increased after some time has passed.

4.2.1 PI speed and PI adaptive MRAS:

4.2.1.1 Case 1:-Constant Reference Speed Response With No Load:

The reference speed is set to a constant of 100 rpm through out the time. The motor is started on no load which remains on no load condition throughout the operation.

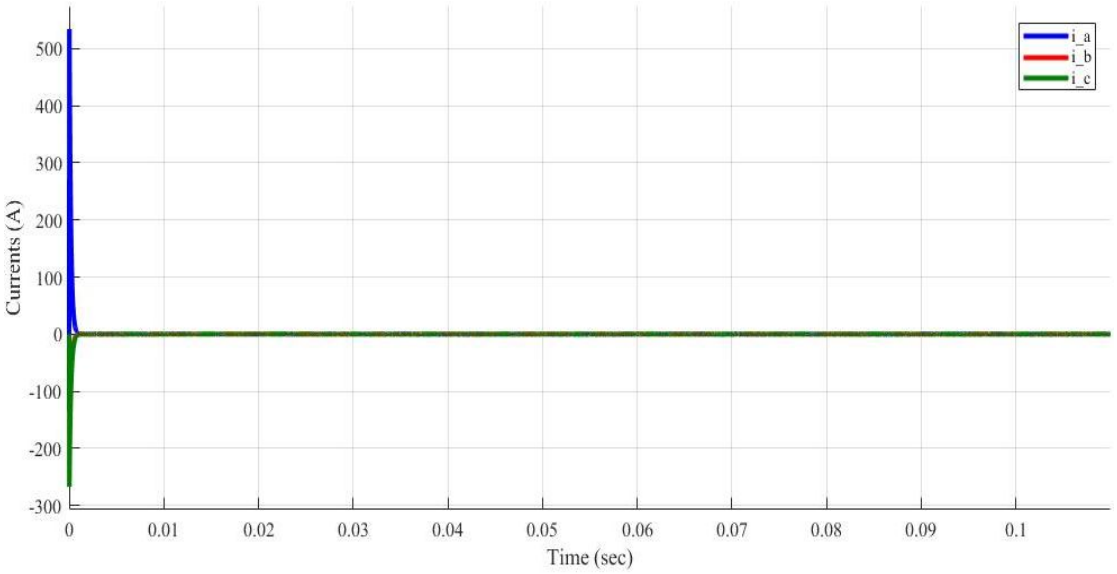


FIGURE 4.1: Phase currents response.

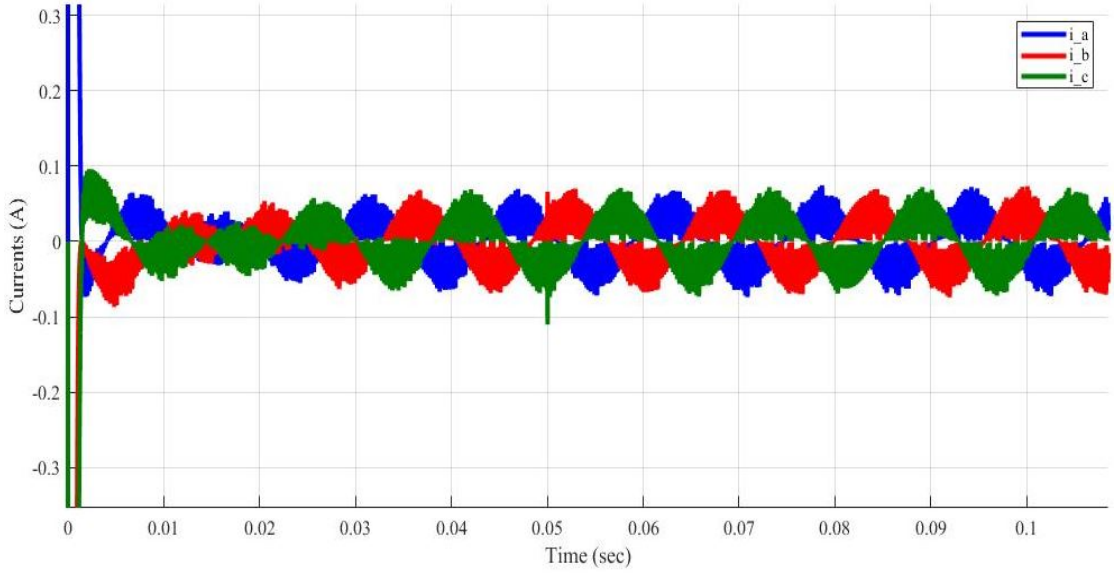


FIGURE 4.2: Zoomed view.

Due to the high starting torque, Figure. 4.1 and Figure. 4.2 show, the initial current overshoots but soon settles to sinusoidal waveforms within 0.01 seconds but gains its uniformity after 0.03 seconds but, they have higher rippled wave. We can also note that each waveform is displaced 120^o from each other.

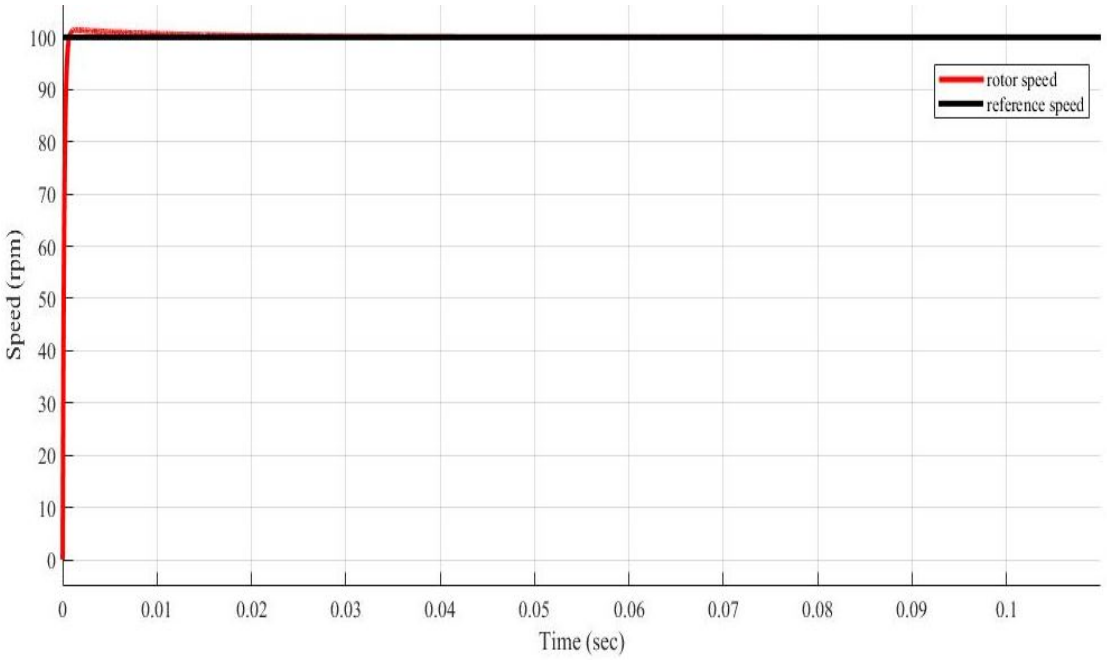


FIGURE 4.3: Speed response of the motor.

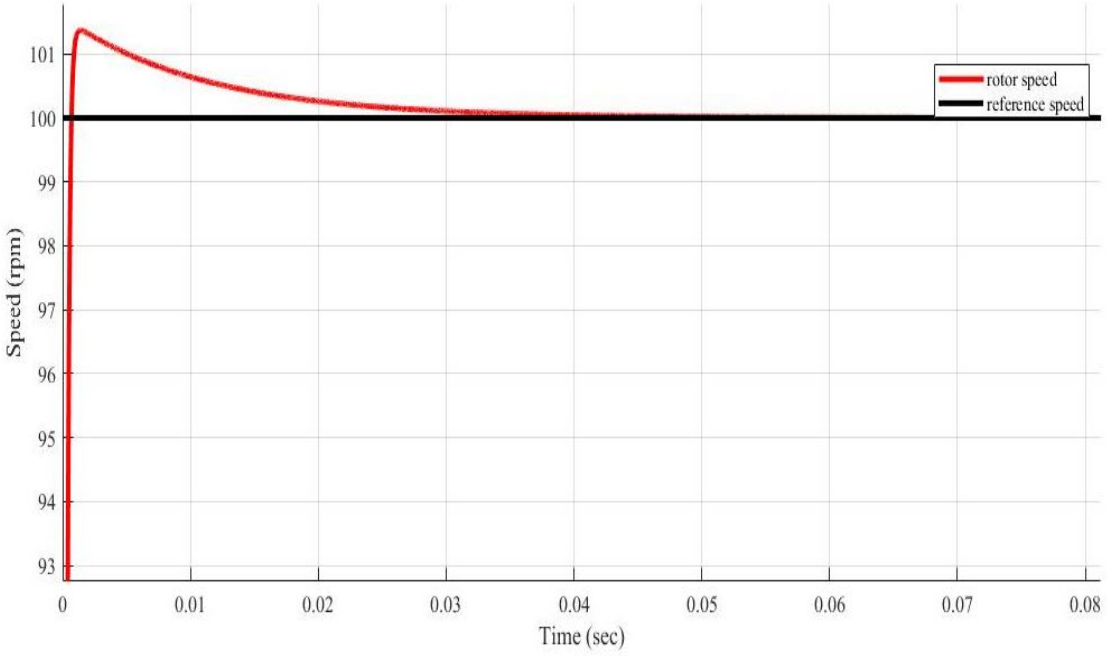


FIGURE 4.4: zoomed speed response of the motor.

Figure. 4.3 and Figure. 4.4, show the response of the speed in of PMSM. The estimated speed has high overshoot initially and tries to track the reference speed within 0.03 seconds.

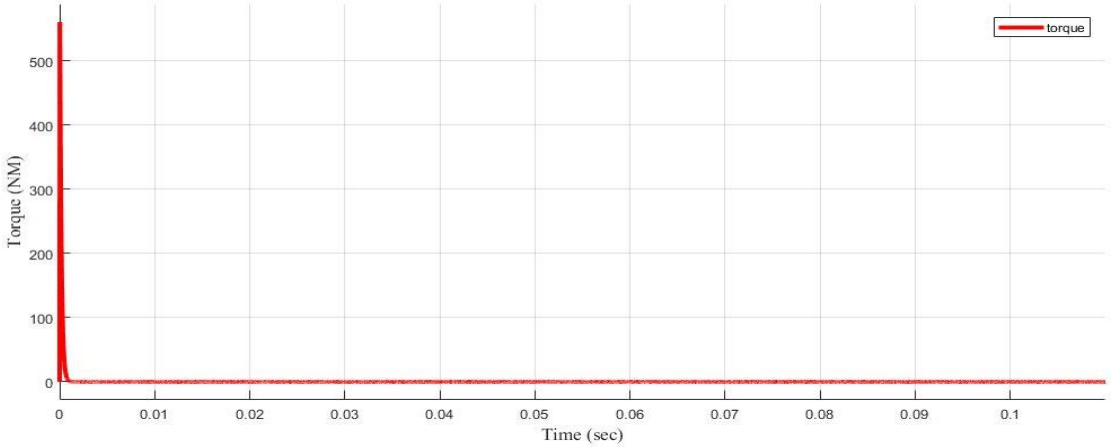


FIGURE 4.5: Torque response.

As shown in Figure. 4.5, the torque response is very high initially but, it approaches to zero immediately before 0.01 second.

4.2.1.2 Case 2:-Constant Reference Speed Response With Full Load:

The reference speed is set to a constant of 100 rpm through out the time but, the motor is started on no load and full loaded(***rated torque***$=2NM$) at time $t = 0.02sec$.The system has the following response.

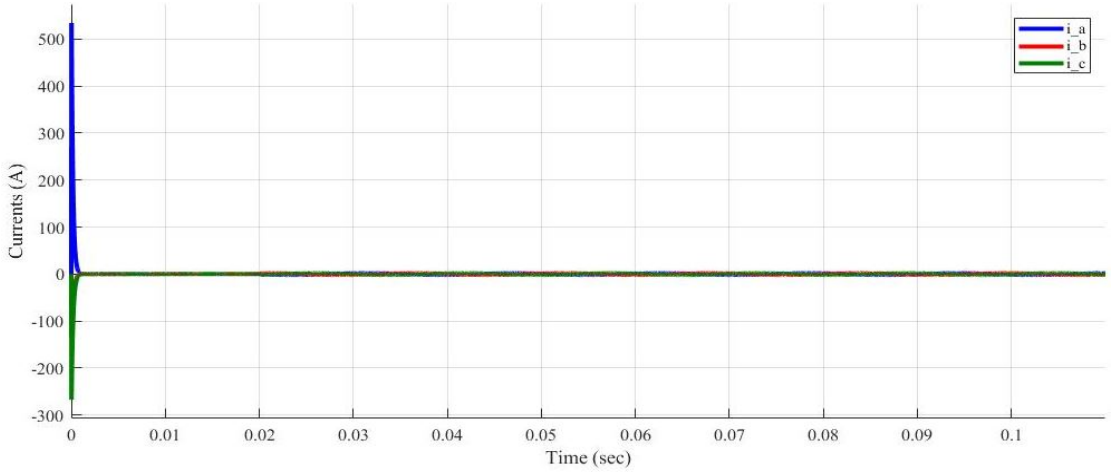


FIGURE 4.6: Phase currents response.

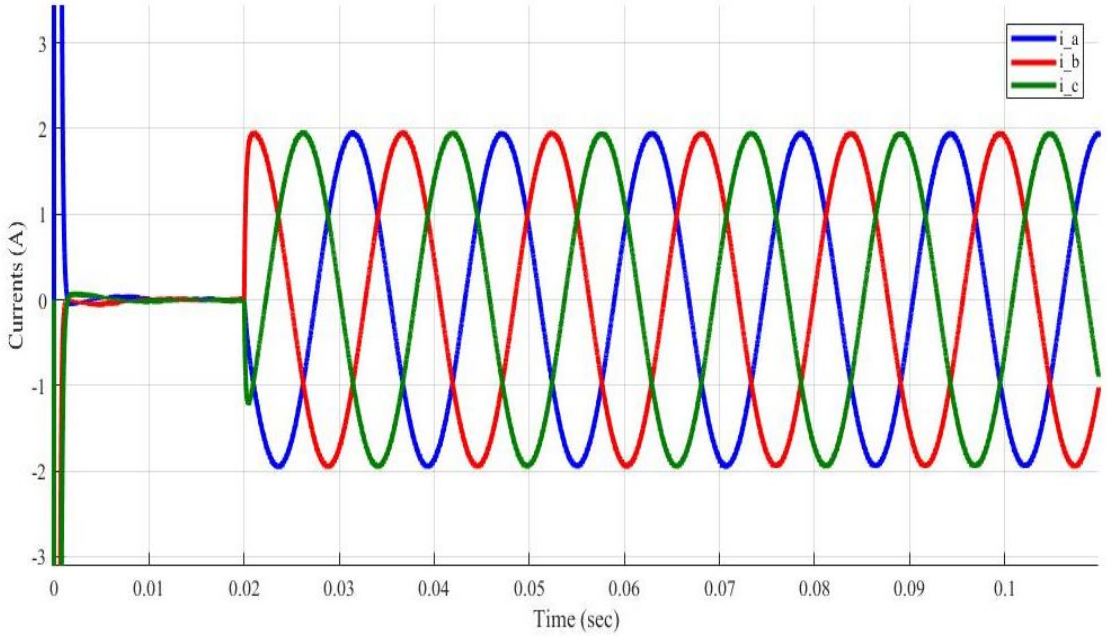


FIGURE 4.7: Zoomed Phase currents response.

Due to the high starting torque, Figure. 4.6 and Figure. 4.7 show, the initial current overshoots but soon settles to sinusoidal waveforms within 0.01 seconds. The amplitude of the currents are increased, due to the addition of external load. We can also note that each waveform is displaced 120^o from each other and have no ripple after addition of the load.

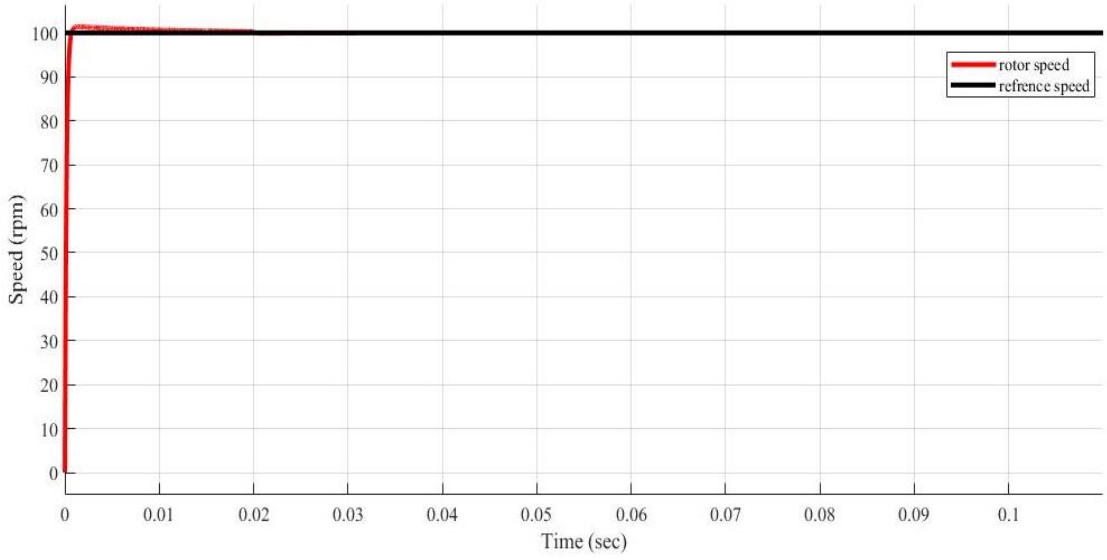


FIGURE 4.8: Speed response of the motor.

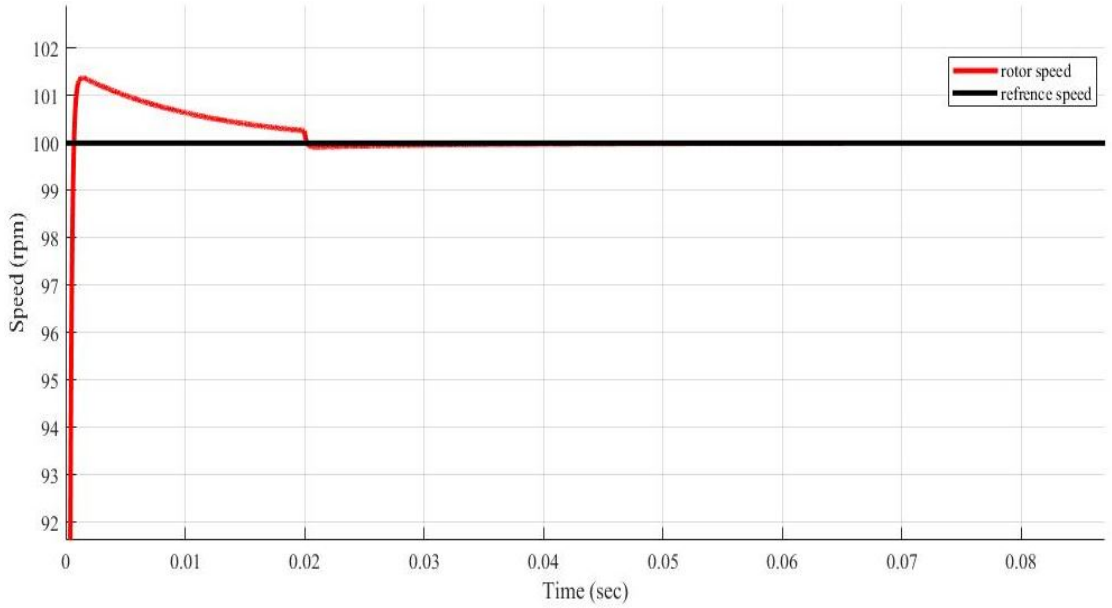


FIGURE 4.9: Zoomed speed response of the motor.

Figure. 4.8 and Figure. 4.9, show the response of the speed of PMSM. The estimated speed has high overshoot initially and tries to track the reference speed within few seconds but, it goes down at 0.02 second due to the addition of external load. The system tries to track the reference speed even after the load disturbance.

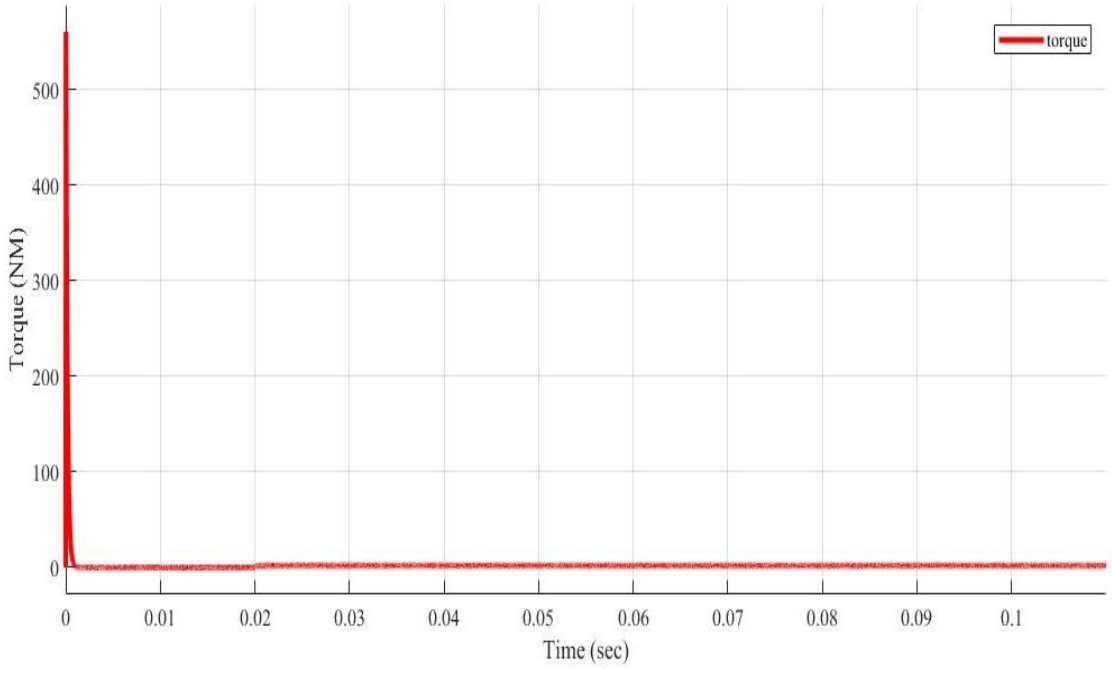


FIGURE 4.10: Torque response.

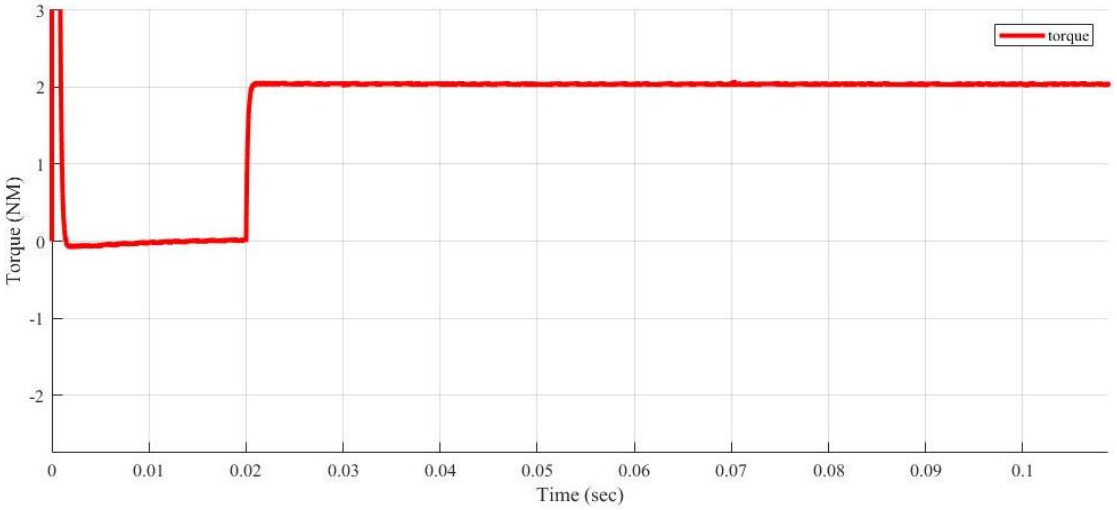


FIGURE 4.11: Zoomed view of torque response.

Figure. 4.10 and its zoomed view Figure. 4.11 show, the response of the system torque, initially it has high torque then drop to zero until additional 2NM load torque is applied, then it remains the same through out the operation time.

4.2.1.3 Case 3:-Step Reference Speed Response at no Load

The reference speed is set to 100 rpm for the first 0.03 seconds then drops to 70 rpm for the next 0.02 seconds and again drops to -100 rpm for the 0.02 seconds then goes to -70 rpm for 0.02 seconds, finally the reference speed reaches 80 rpm for the remaining seconds. The motor is started on no load and remaining no load through out the time.

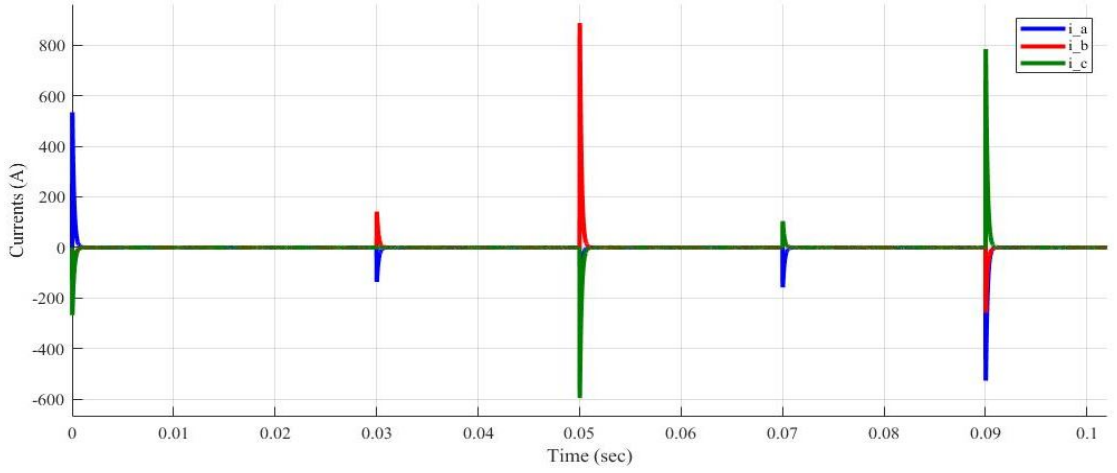


FIGURE 4.12: Phase currents response.

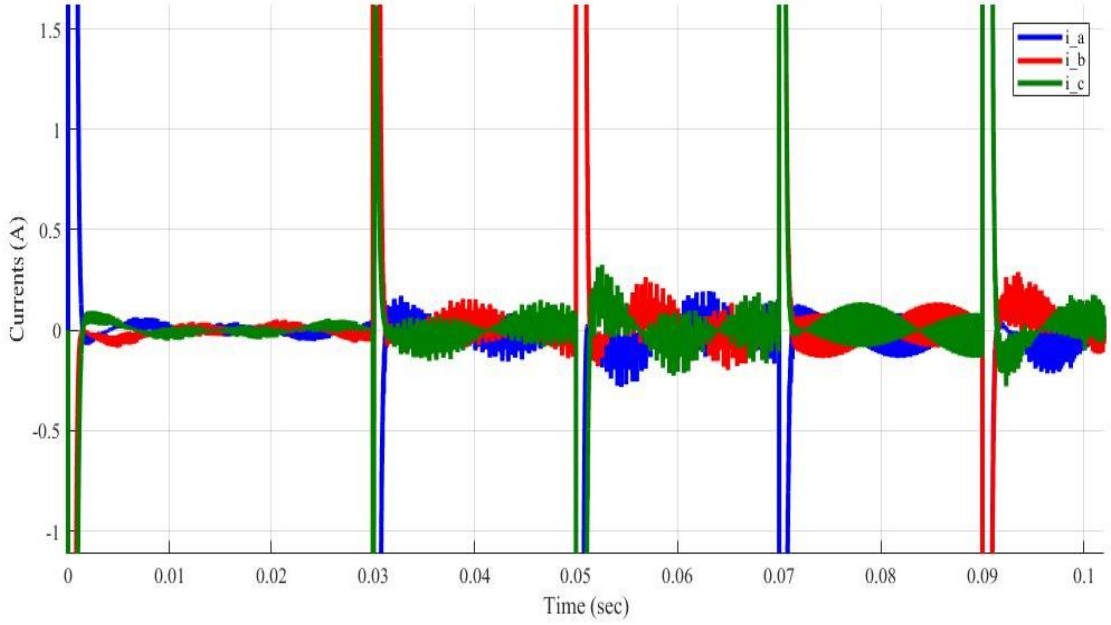


FIGURE 4.13: Zoomed view Phase currents response.

Due to the high starting torque, Figure. 4.12 and its zoomed view Figure. 4.13 show, the initial current overshoots but soon settles for uniform sinusoidal waveforms within 0.01 seconds, but at the time of transition the amplitude is getting high. We can also note that each waveform have ripple and displaced 120^o from each other and for the reverse operation two phases change their wave form but still they displaced 120^o from each other.

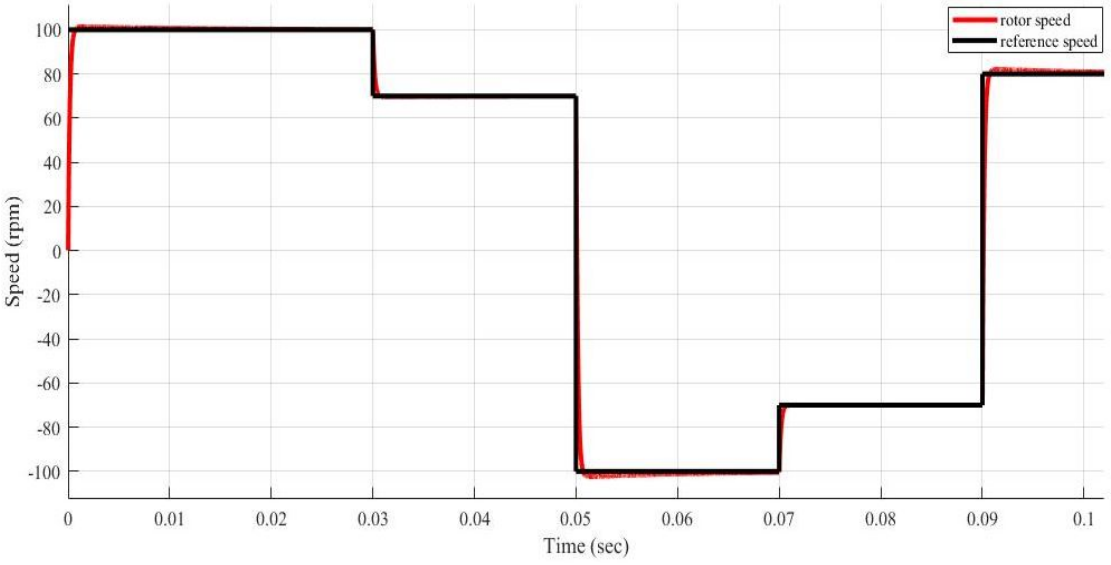
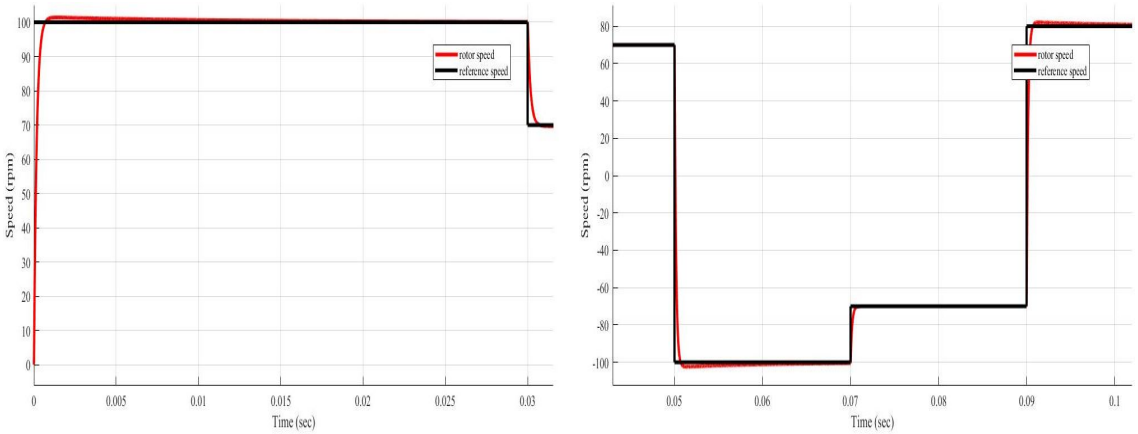


FIGURE 4.14: Speed response of the motor.



(A) speed response at motoring mode (B) speed response at reversing mode

FIGURE 4.15: Zoomed speed responses.

Figure. 4.14 and its zoomed view Figure. 4.15, show the response of the speed for different step reference input. The estimated speed tries to track the reference speed with high initial overshoot and low tracking error. Figure. 4.21 shows, the response of the system torque, initially it has high torque then drop to zero and getting high at the transition time. After time of transition, it comeback to zero and gets high when another transition is coming.

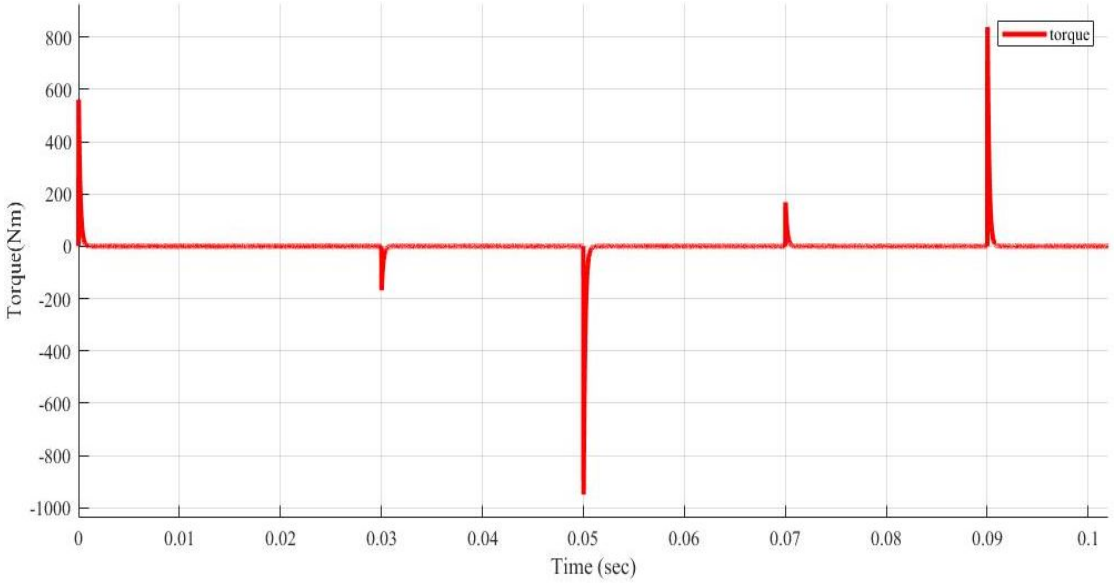


FIGURE 4.16: Torque response of the motor.

4.2.1.4 Case 4:-Step Reference Speed Response With Full Load:

The reference speed is set to 100 rpm for the first 0.03 seconds then drops to 70 rpm for the next 0.02 seconds and again drops to -100 rpm for the 0.02 seconds then goes to -70 rpm for 0.02 seconds, finally the reference speed reaches 80 rpm for the remaining seconds. The motor is started on no load and full loaded (***rated torque***$=2NM$) at time $t = 0.02sec$.

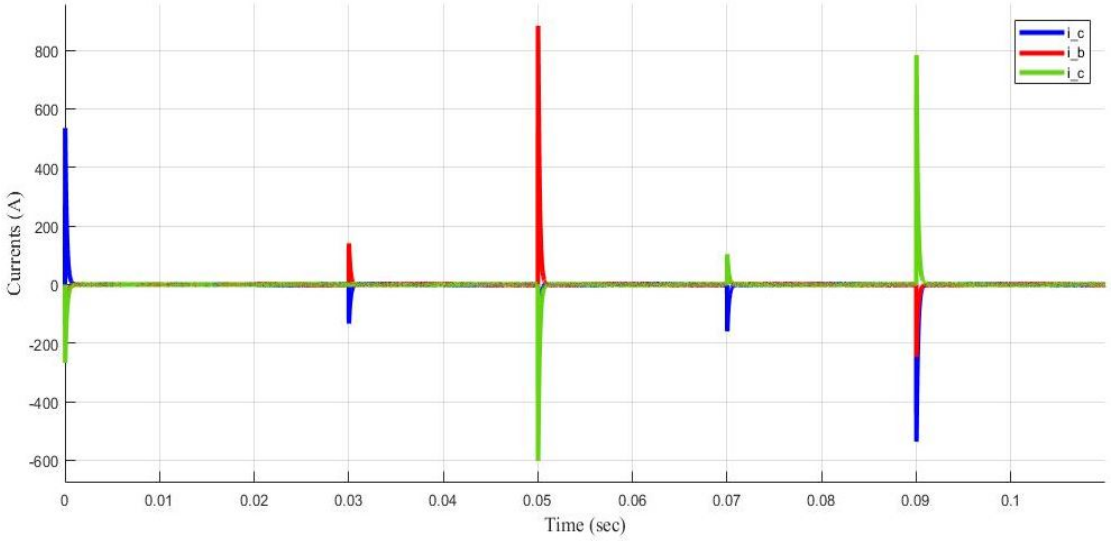


FIGURE 4.17: Phase currents response.

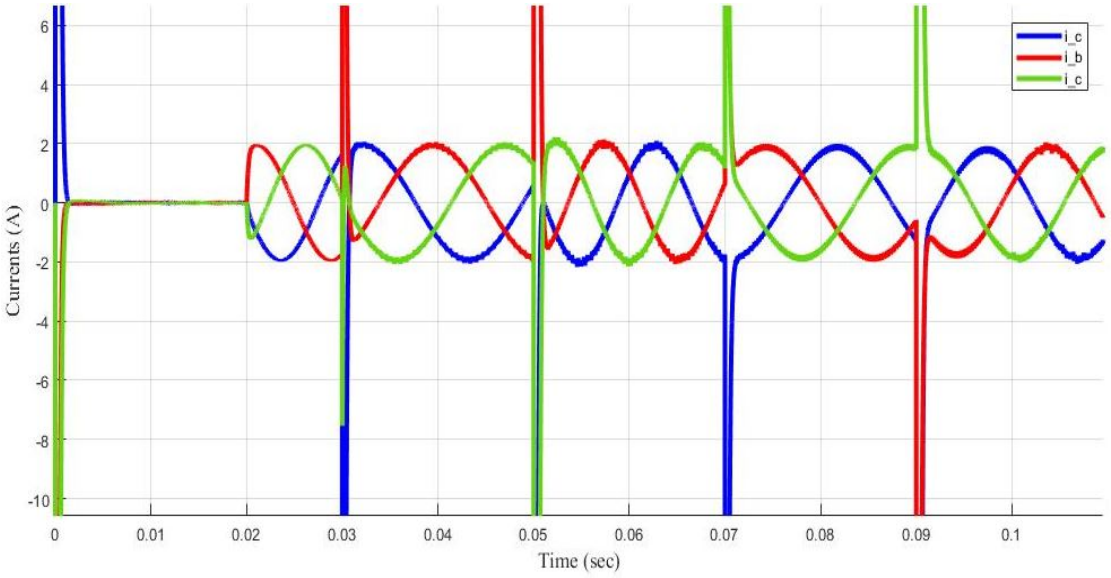


FIGURE 4.18: Zoomed view Phase currents response.

Due to the high starting torque, Figure. 4.17 and its zoomed view Figure. 4.18 show, the initial current overshoots but soon settles for uniform sinusoidal waveforms within 0.01 seconds,but, when load is applied at 0.02 second the amplitude of waves gets larger. The amplitude also high at the time of transition. We can also note that each waveform have no ripple after the load disturbance and displaced 120^o from each other and for the reverse operation two phases change their wave form but still they displaced 120^o from each other.

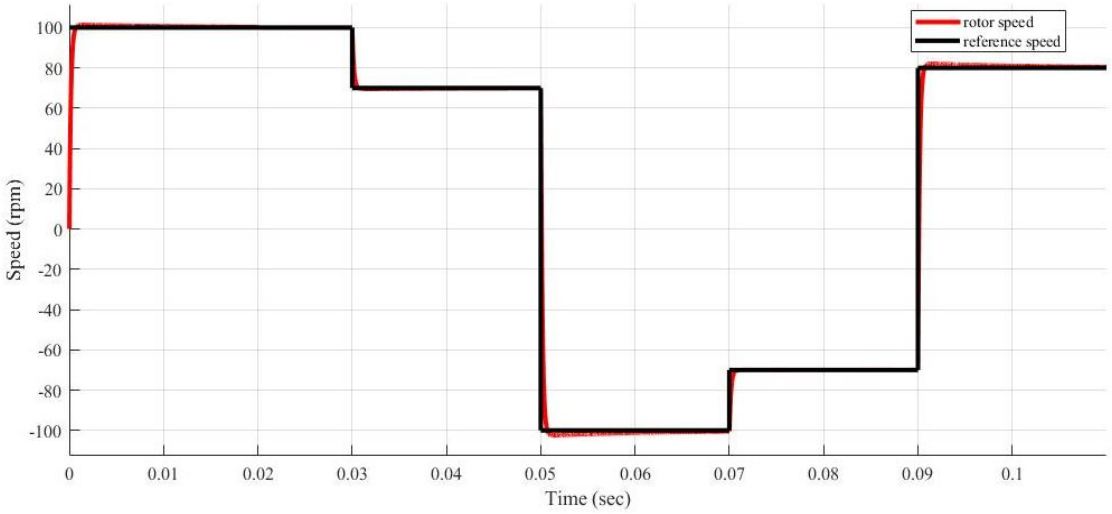
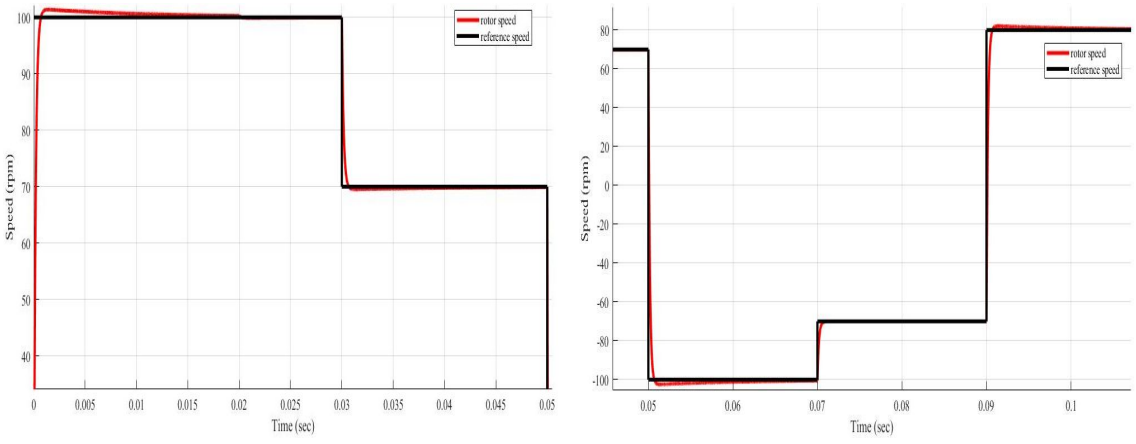


FIGURE 4.19: Speed response of the motor.



(A) Speed response at motoring mode (B) Speed response at reversing mode

FIGURE 4.20: Zoomed speed responses.

Figure. 4.19 and its zoomed view Figure. 4.20, show the response of the speed for different step reference input. The estimated speed tries to track the reference speed with high initial overshoot and low tracking error but,it goes down at 0.02 second due to the addition of external load. The system tries to track the reference speed after the load disturbance.

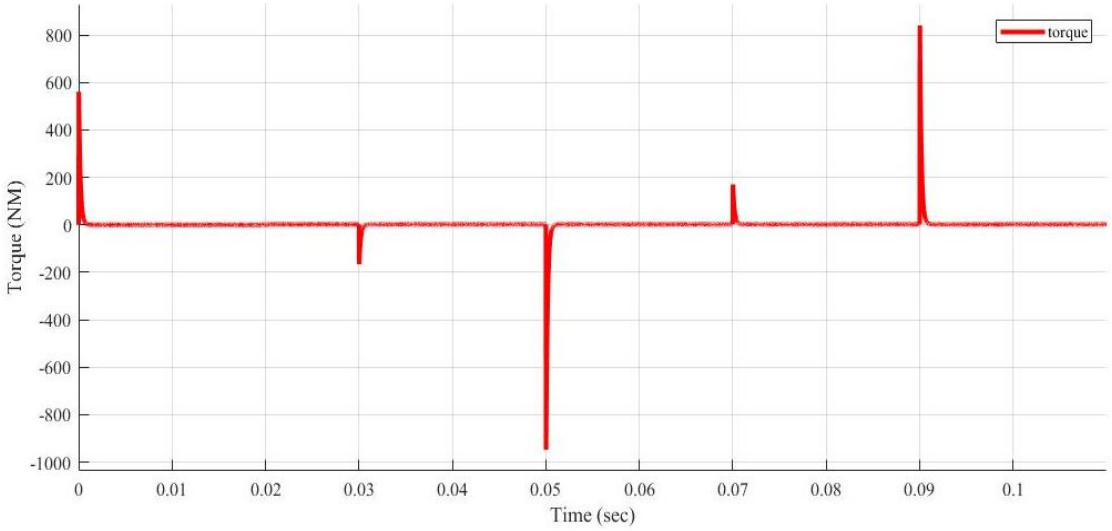


FIGURE 4.21: Torque response of the motor.

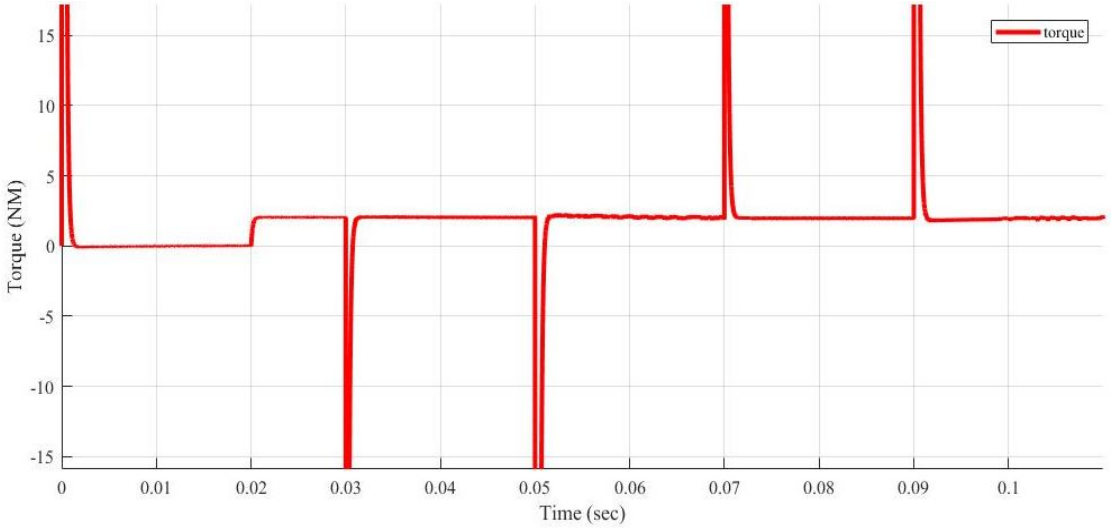


FIGURE 4.22: Zoomed view of torque response.

Figure. 4.21 and its zoomed view Figure. 4.22 show, the response of the system torque, initially it have high torque then drop to zero until additional load is applied (***rated torque***$=2NM$) at time $t = 0.02sec$. The torque also getting high at the transition system. After time of transition it comeback to the previous torque value and gets high when another transition is come.

4.2.2 ANN speed and ANN adaptive MRAS:

4.2.2.1 Case 1:-Constant Reference Speed Response With No Load

The reference speed is set to a constant of 100 rpm through out the time. The motor is started on no load which remains on no load condition throughout the operation.

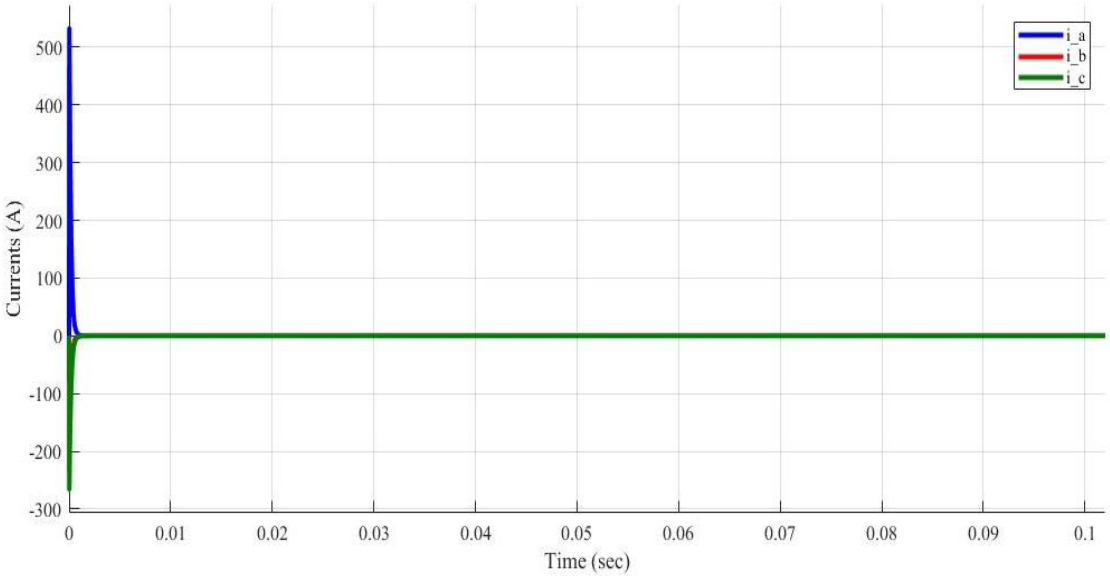


FIGURE 4.23: Phase currents response.

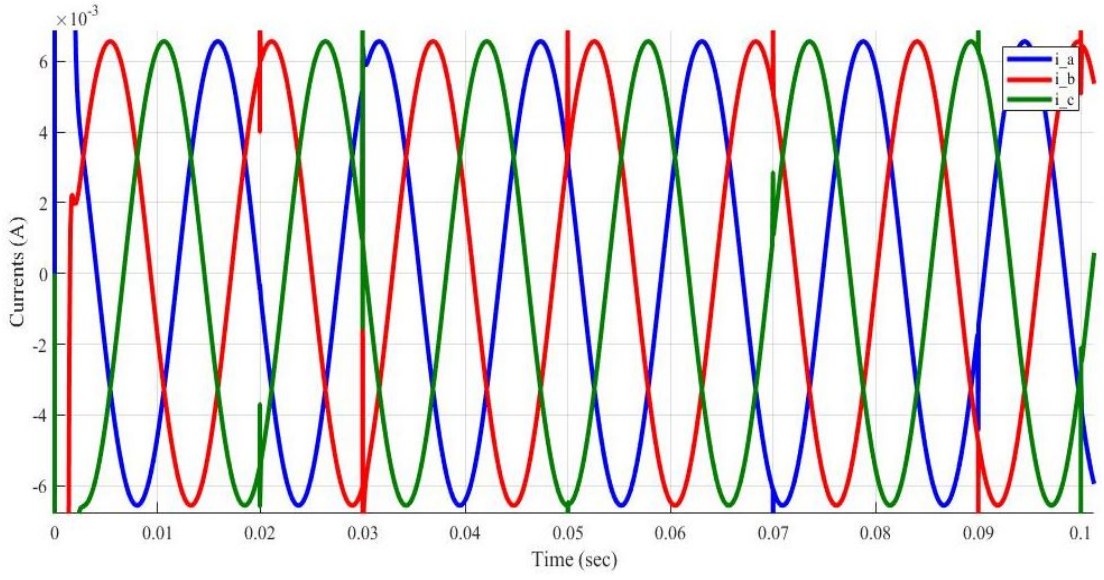


FIGURE 4.24: Zoomed Phase currents response.

Due to the high starting torque, Figure. 4.23 and Figure. 4.24 show, the initial current overshoots but soon settles to uniform sinusoidal waveforms within 0.01 seconds. We can also note that each waveform has no ripple and they displaced 120^o from each other.

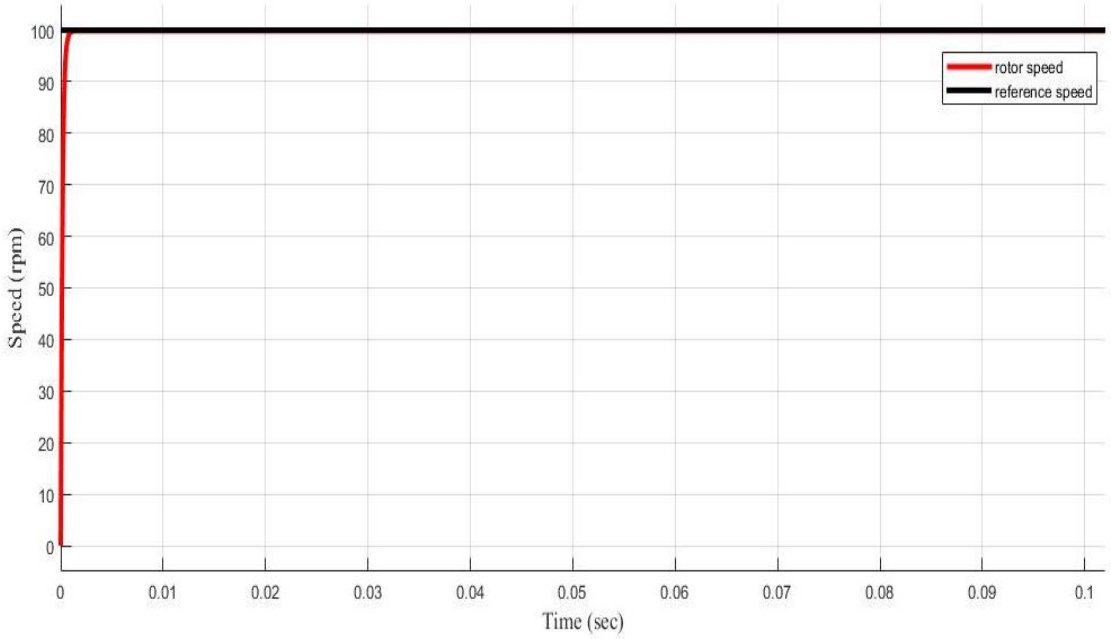


FIGURE 4.25: Speed response of the motor.

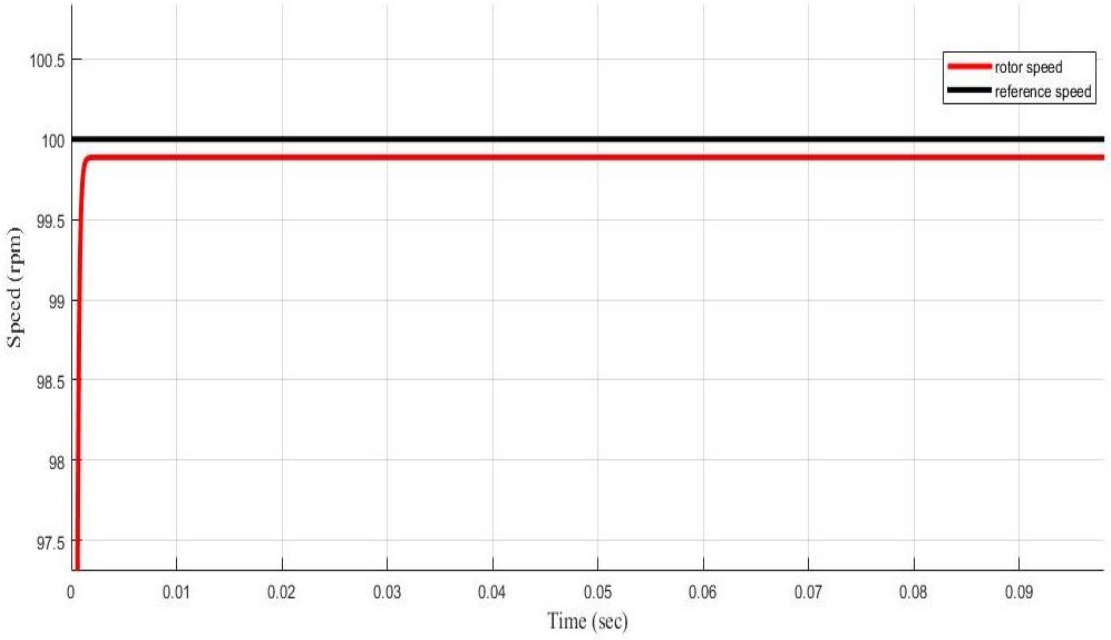


FIGURE 4.26: Zoomed speed response of the motor.

Figure. 4.25 and Figure. 4.26, show the response of the speed in of PMSM. The estimated speed has low overshoot initially and tries to track the reference speed with minimum steady state error.

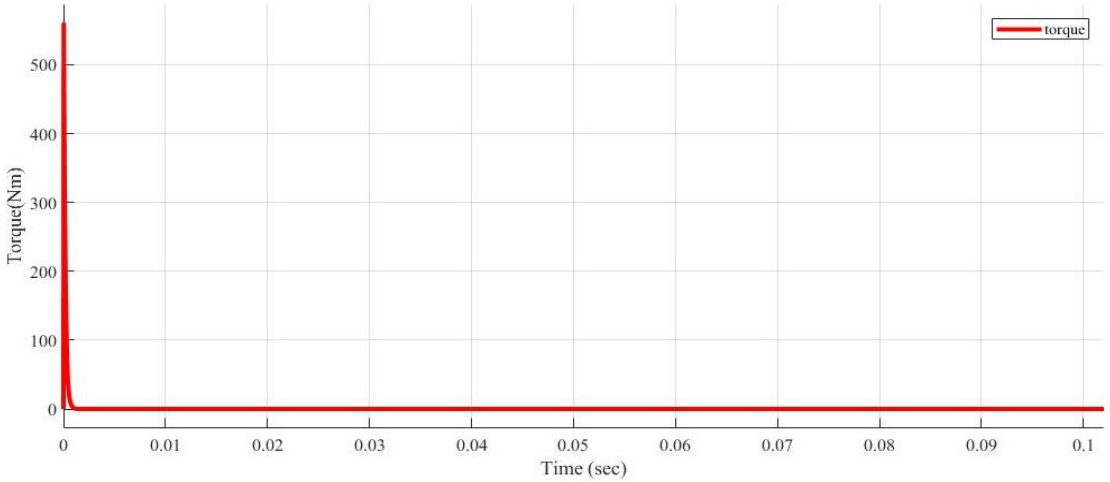


FIGURE 4.27: Torque response.

As shown in Figure. 4.27, the torque response is very high initially but, it approaches to zero immediately before 0.01 second.

4.2.2.2 Case 2:-Constant Reference Speed Response With Full Load

The reference speed is set to a constant of 100 rpm through out the time but, the motor is started on no load and full loaded(***rated torque***$=2NM$) at time $t = 0.02sec$. The system has the following response.

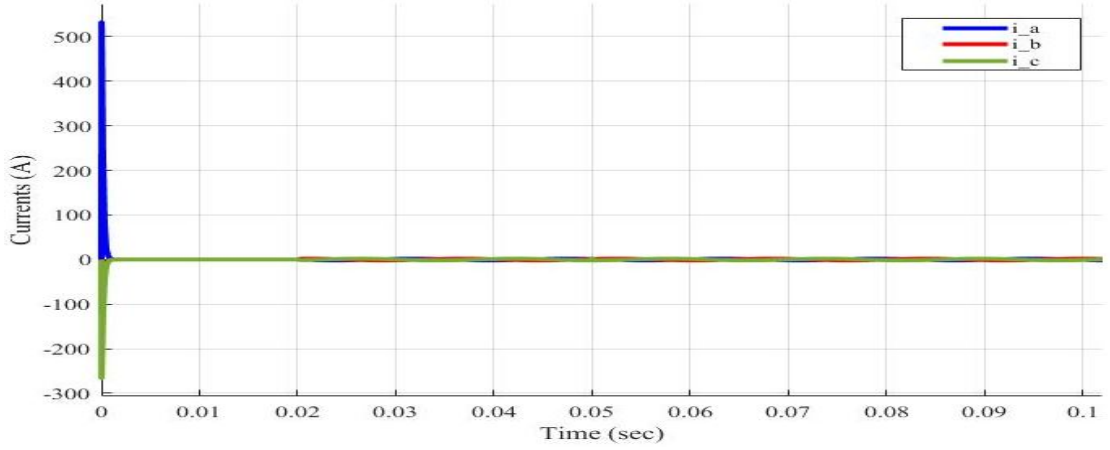
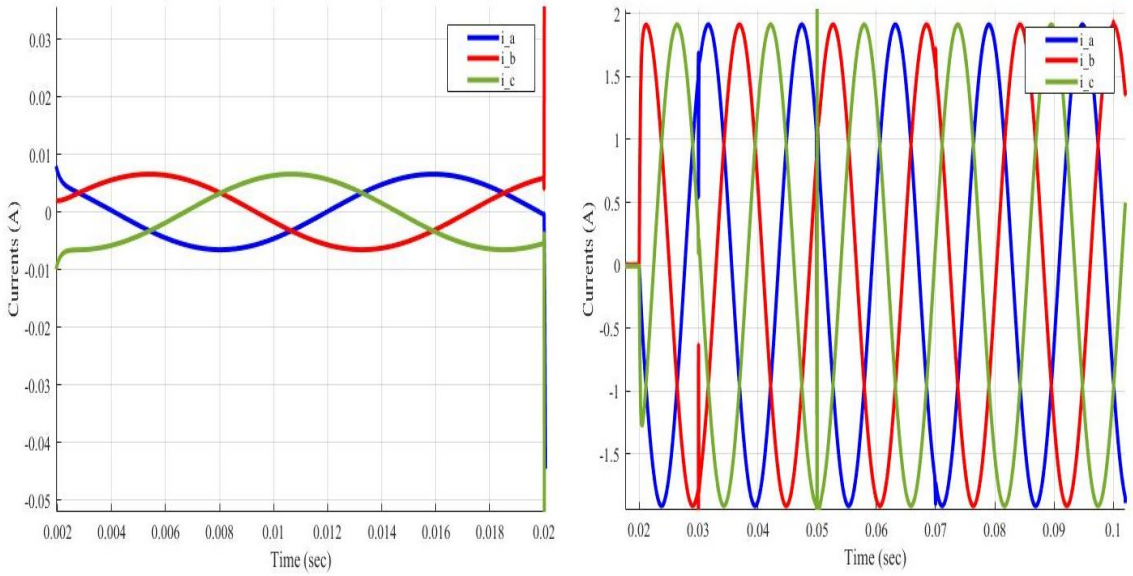


FIGURE 4.28: Phase currents response.



(A) Before addition of load torque (B) After addition of load torque

FIGURE 4.29: Zoomed Phase currents responses.

Due to the high starting torque, Figure. 4.28 and Figure. 4.29 show, the initial current overshoots but soon settles to uniform sinusoidal waveforms within 0.03 seconds. The amplitude of the currents are increased its amplitude due to the addition of external load. We can also note that each waveform is displaced 120^o from each other and have no ripple before and after the addition of the external load.

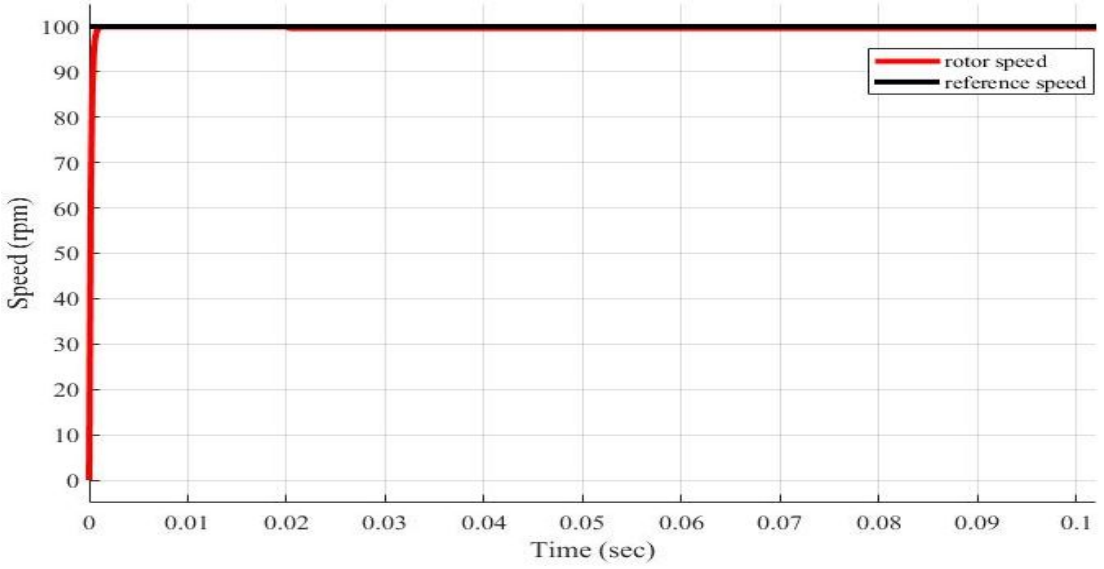


FIGURE 4.30: Speed response of the motor.

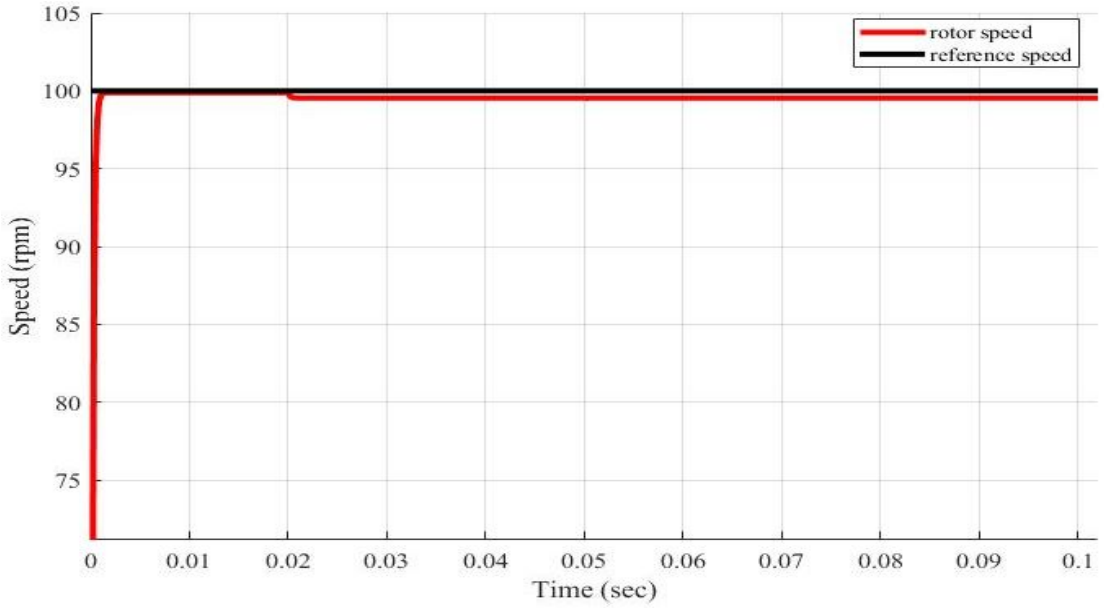


FIGURE 4.31: Zoomed speed response of the motor.

Figure. 4.30 and Figure. 4.31, show the response of the speed of PMSM. The estimated speed has very low overshoot initially and tries to track the reference speed with very low steady state error but, it goes down at 0.02 second due to the addition of external load. The system tries to track the reference speed even after the load disturbance, the steady state error increases slightly but, it is acceptable even for sensitive operations.

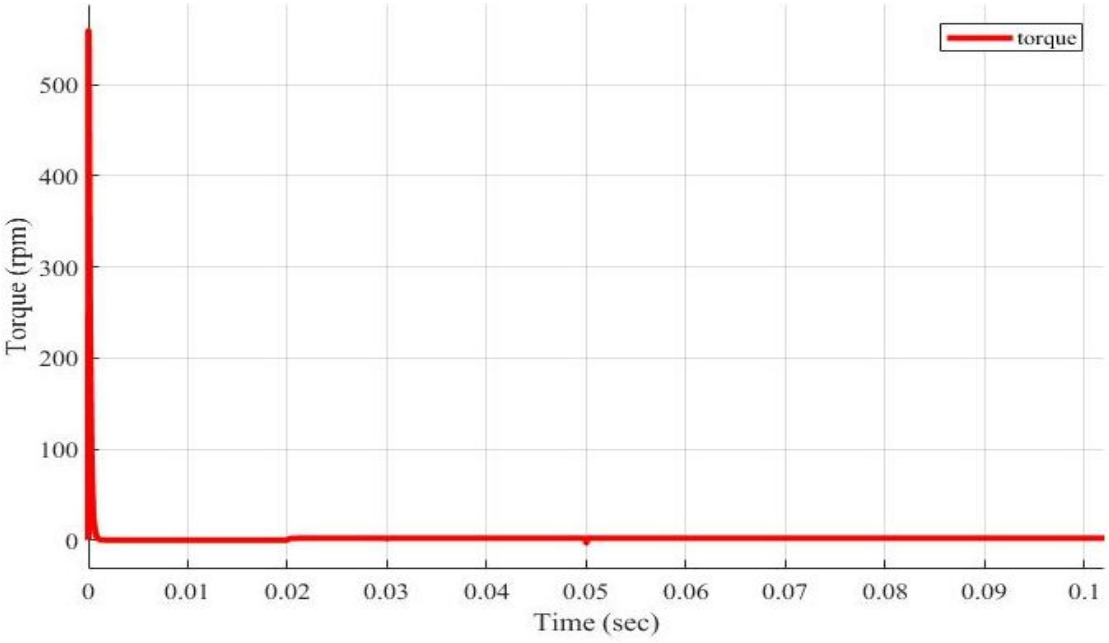


FIGURE 4.32: Torque response.

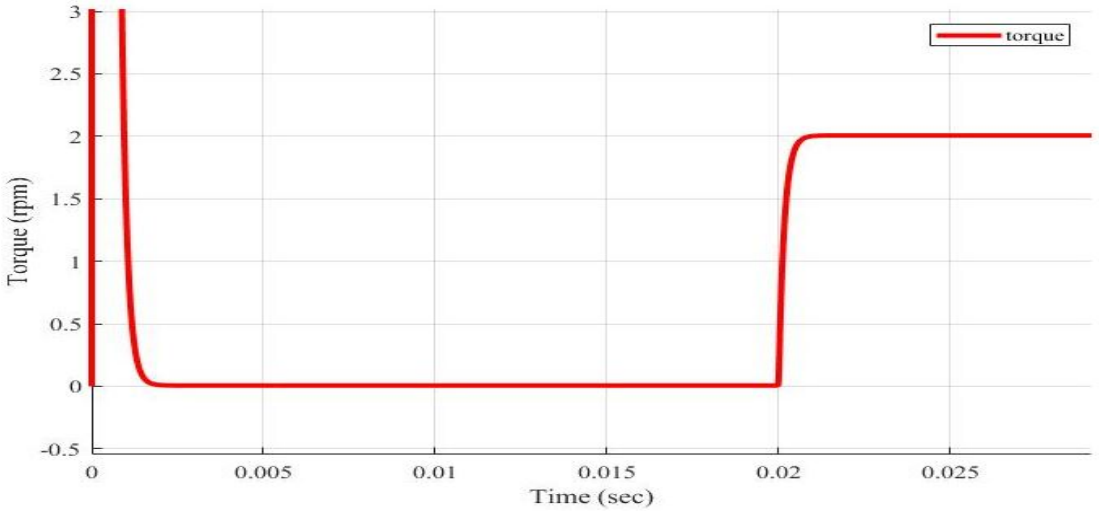


FIGURE 4.33: Zoomed view of torque response.

Figure. 4.32 and its zoomed view Figure. 4.33 show, the response of the system torque, initially it has high torque then drop to zero until additional 2NM load torque is applied, then it remains the same through out the process.

4.2.2.3 Case 3:-Step Reference Speed Response at no Load:

The reference speed is set to 100 rpm for the first 0.04 seconds then drops to -100 rpm for the next 0.02 seconds and go-up to -50 rpm for 0.02 seconds , finally the reference speed reaches 70 rpm for the remaining seconds. The motor is started on no load and remaining no load through out the time.

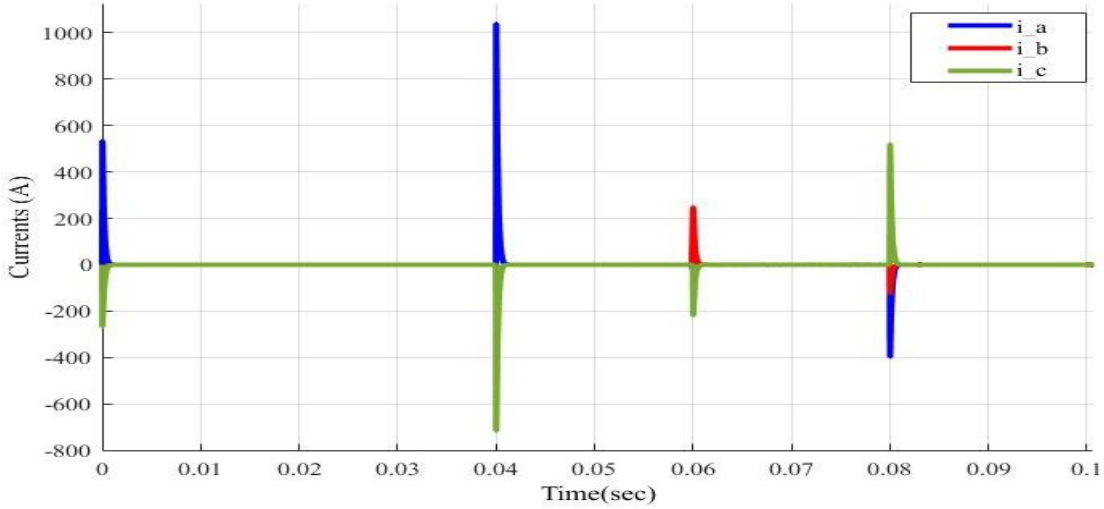


FIGURE 4.34: Phase currents response.

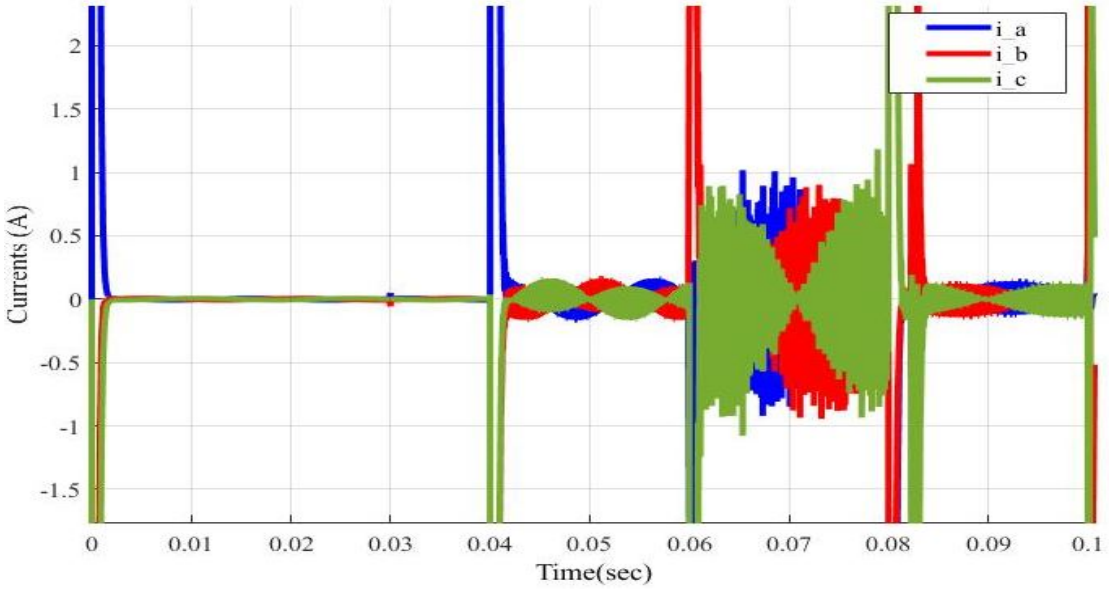


FIGURE 4.35: Zoomed view Phase currents response.

Due to the high starting torque, Figure. 4.34 and its zoomed view Figure. 4.35 show, the initial current overshoots but soon settles for uniform sinusoidal waveforms within 0.01 seconds but, at the time of transition the amplitude is getting high. We can also note that each waveform have ripple and displaced 120^o from each other and for the reverse operation two phases change their wave form but still they displaced 120^o from each other.

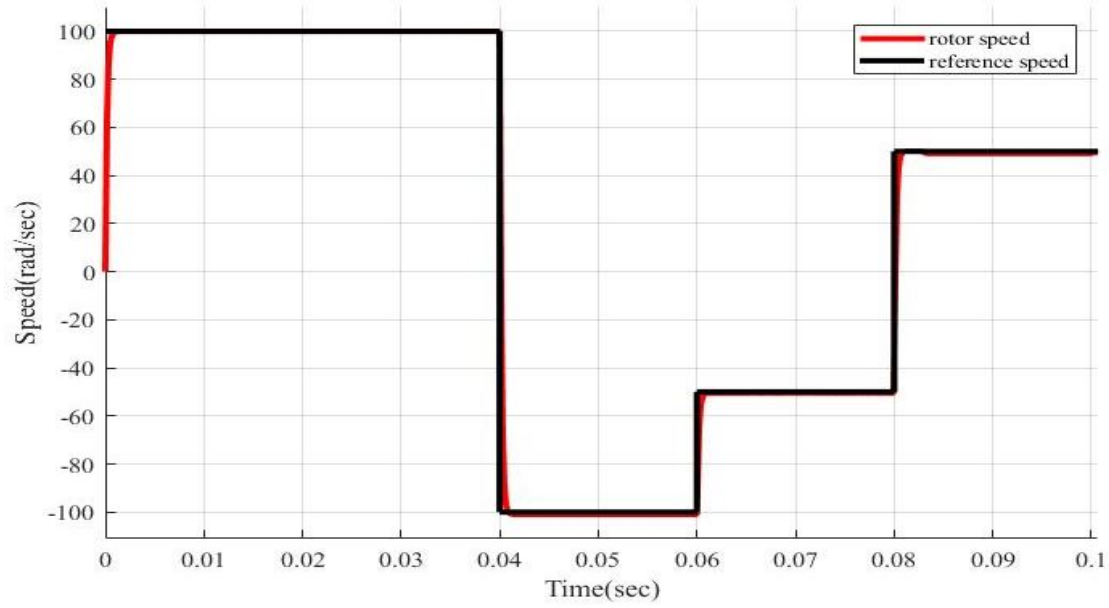
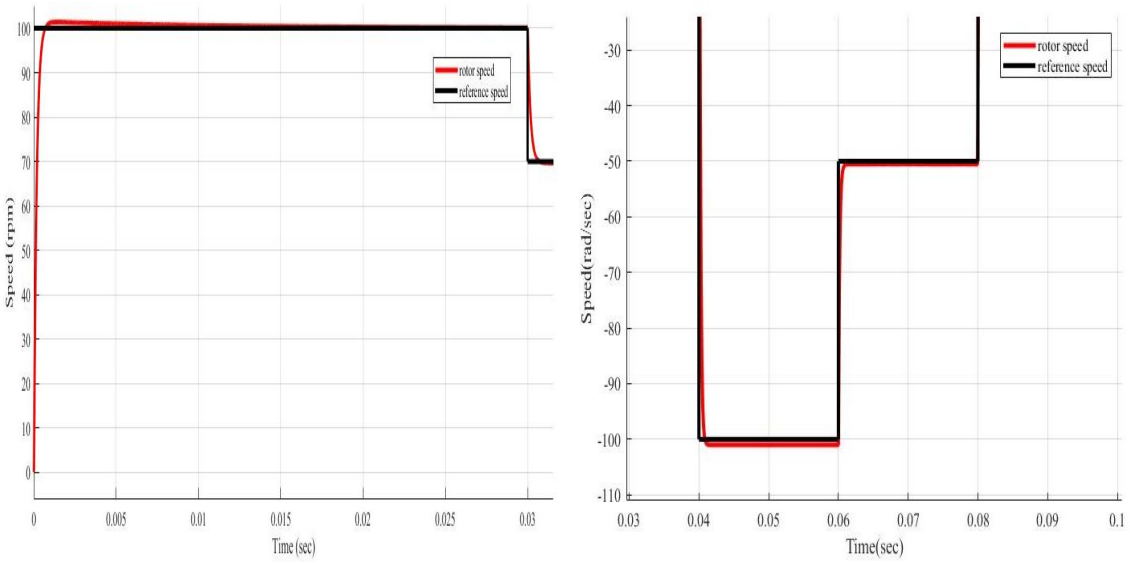


FIGURE 4.36: Speed response of the motor.



(A) Speed response at motoring mode (B) Speed response at reversing mode

FIGURE 4.37: Zoomed speed responses.

Figure. 4.36 and its zoomed view Figure. 4.37, show the response of the speed for different step reference input. The estimated speed tries to track the reference speed with high initial overshoot and low tracking error. Figure. 4.38 shows, the response of the system torque, initially it have high torque then drop to zero and getting high at the transition system. After time of transition it comeback to zero and gets high when another transition is come.

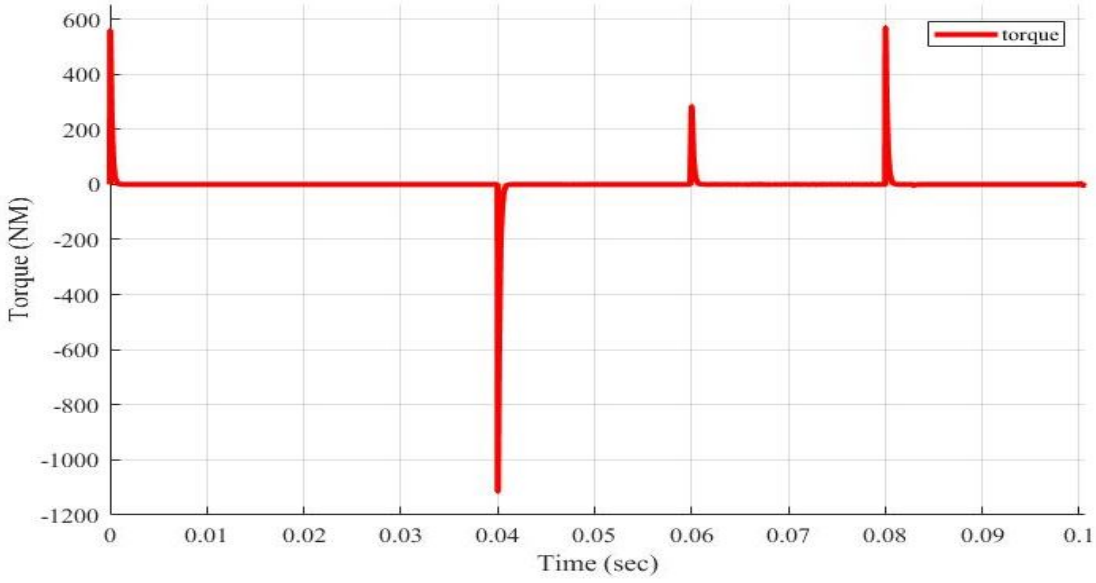


FIGURE 4.38: Torque response of the motor.

4.2.2.4 Case 4:-Step Reference Speed Response With Full Load

The reference speed is set to 100 rpm for the first 0.04 seconds then drops to -100 rpm for the next 0.02 seconds and go-up to -50 rpm for 0.02 seconds , finally the reference speed reaches 70 rpm for the remaining seconds. The motor is started on no load and full loaded (***rated torque***$=2NM$) at time $t = 0.02sec$.

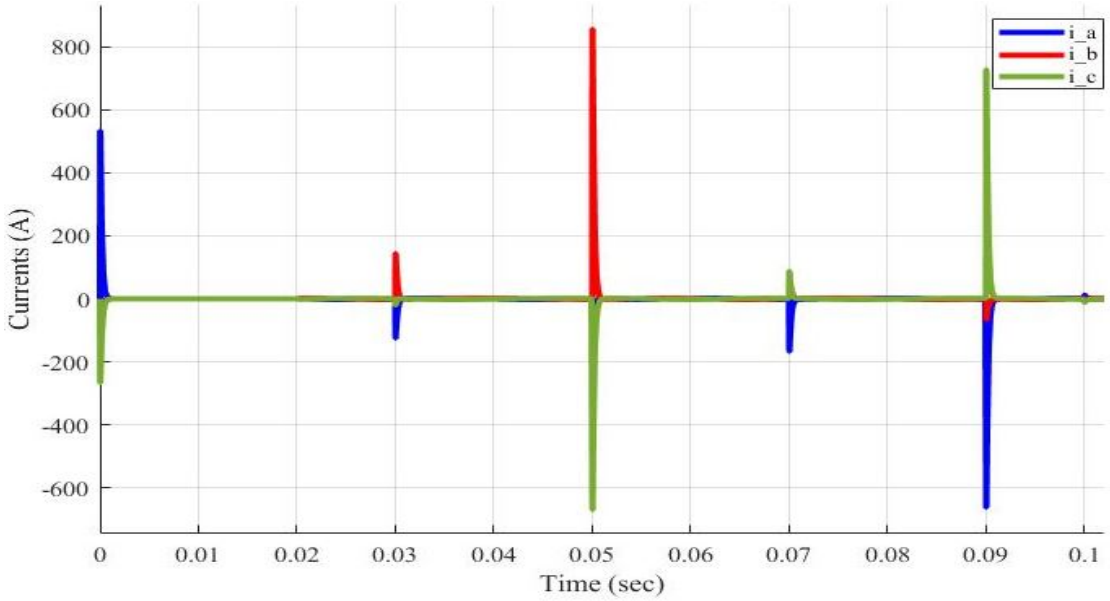


FIGURE 4.39: Phase currents response.

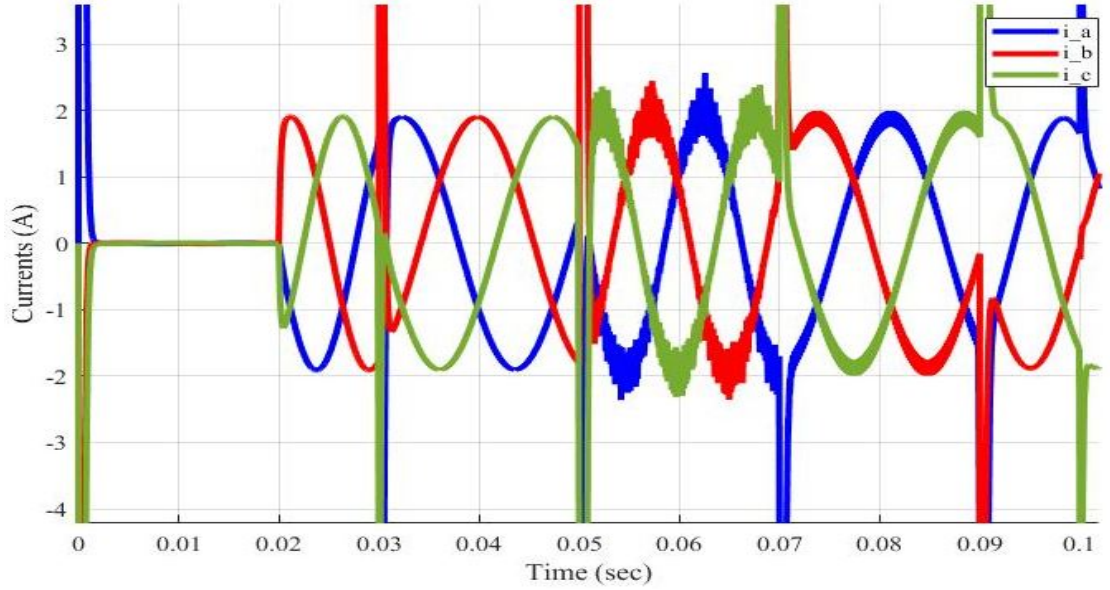


FIGURE 4.40: Zoomed Phase currents response.

Due to the high starting torque, Figure. 4.39 and its zoomed view Figure. 4.40 show, the initial current overshoots but soon settles for uniform sinusoidal waveforms within 0.01 seconds, but, when load is applied at 0.02 second the amplitude of waves gets larger. The amplitude also high at the time of transition. We can also note that each waveform have no ripple and displaced 120^o from each other and for the reverse operation two phases change their wave form but still they displaced 120^o from each other.

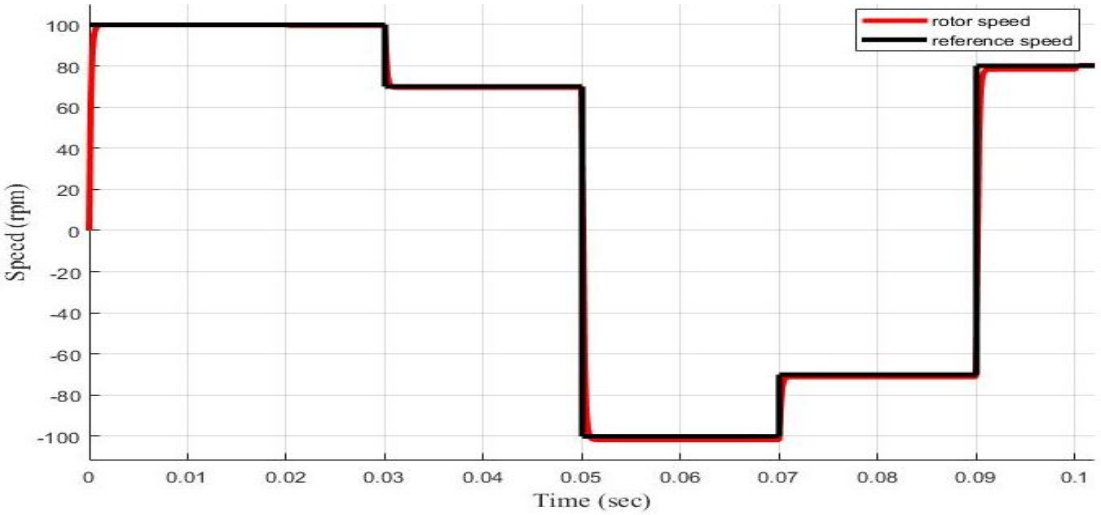
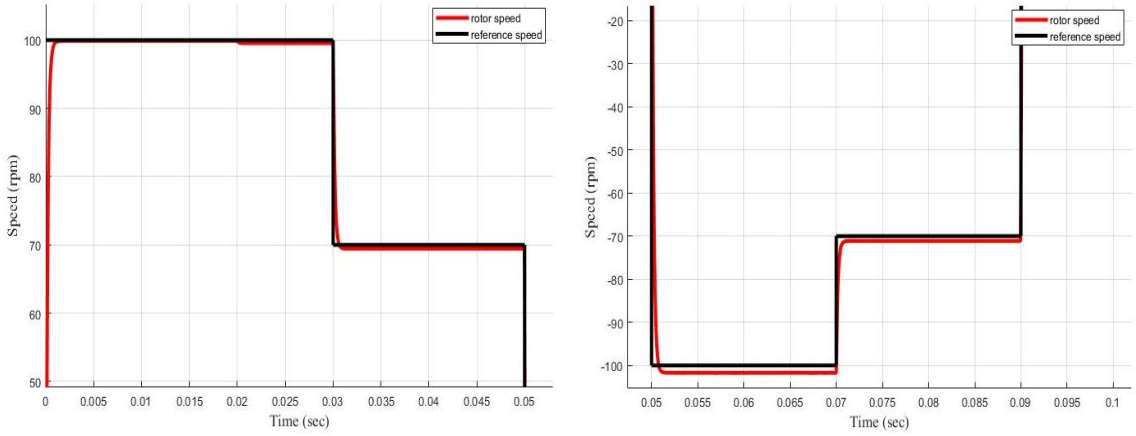


FIGURE 4.41: Speed response of the motor.



(A) Speed response at motoring mode (B) Speed response at reversing mode

FIGURE 4.42: Zoomed speed responses.

Figure. 4.41 and its zoomed view Figure. 4.42, show the response of the speed for different step reference input. The estimated speed tries to track the reference speed with high initial overshoot and low tracking error but, it goes down at 0.02 second due to the addition of external load. The system tries to track the reference speed even after the load disturbance.

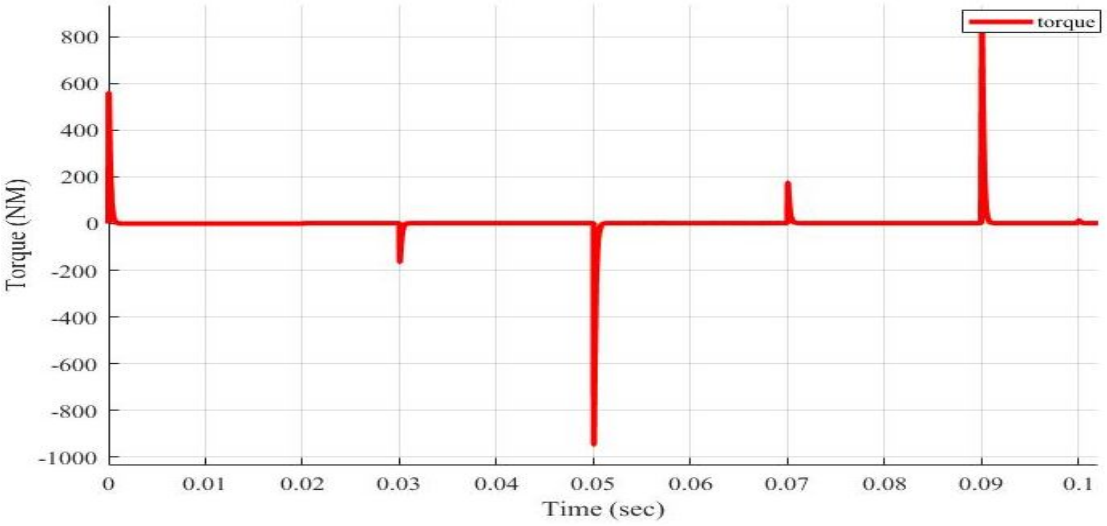


FIGURE 4.43: Torque response of the motor.

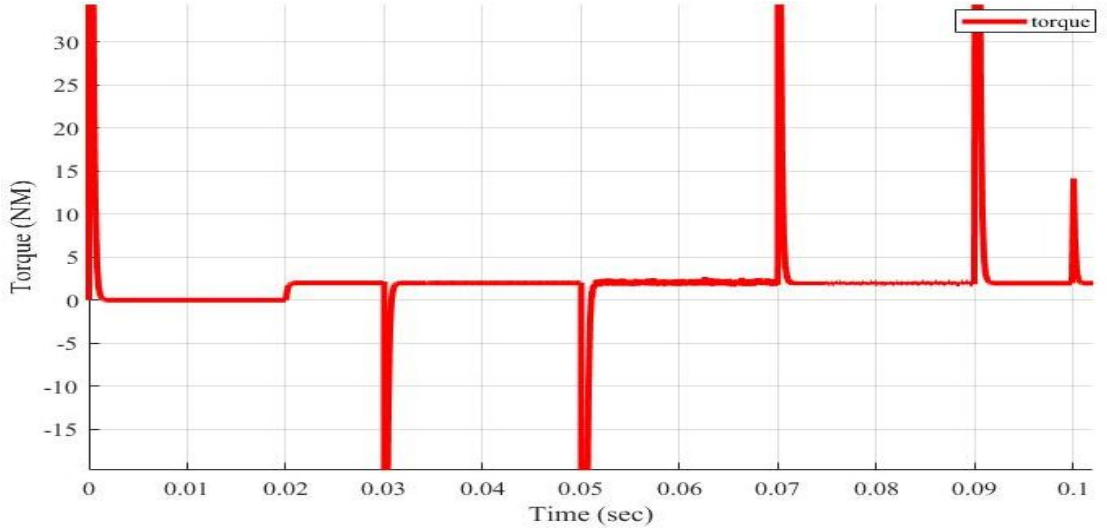


FIGURE 4.44: Zoomed view of torque response.

Figure. 4.43 and its zoomed view Figure. 4.44 show, the response of the system torque, initially it have high torque then drop to zero until additional load is applied (***rated torque***$=2NM$) at time $t = 0.02sec$. The torque also getting high at the transition system. After time of transition it comeback to the previous torque value and gets high when another transition is come.

4.3 Comparison

The comparison of both conventional method and proposed method based on key performance parameters is shown in Table 4.2.

TABLE 4.2: Performance Analysis

Method	Rise Time (ms)	Overshoot (%)	Settling Time (ms)	St.state error
Proposed	0.337	0.500	0.448	0.110
Conventional	0.336	1.534	0.432	0.109

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

This thesis proposes ANN aided Model Reference Adaptive System speed observer in sensorless FOC to enhance the PMSM performance at no load and low speed operation. The PMSM, FOC and MRAS estimator were modeled in Simulink. A predefined set of test cases were then simulated using the conventional and the proposed estimation method. The results from the test cases were then compared between each method.

From the result we conclude that, ANN based implementation of the speed estimator and FOC reduces the complexity of the controller. Hence, tuning the parameters become a trivial task.

This system also replaces those PI controllers with ANN, due to their mathematical nature ANN can work easily, resulting in a reduction of complexity. Besides simplicity, the proposed controller plays a major role in reducing the initial overshoot, which is about 1% improvement from the traditional PI controller. It also achieve a comparable settling time, steady state error and rise time to that of a conventional implementation.

5.2 Recommendation

For further improvement of overall sensorless system,we recommend an additional stator resistance estimator. In MRAS, there are instances where the stator resistance is estimated together with the speed and position. Stator resistance estimation could improve the performance further and also gives the opportunity to estimate the motor temperature.

In addition, to get the model more realistic it would also be good to model the SVM/SVPWM-function and the voltage inverter.

The next step would be to implement the proposed controller in hardware and access its performance in real control scenario in future. Then apply all the test cases with the real motor and analyze the motor behavior. A preference would be if the motor had an encoder so the speed and position values could be properly compared.

Bibliography

[1] F. Giri, Ed., AC ELECTRIC MOTORS CONTROL. John Wiley and Sons, Ltd, 2013.

[2] H. Abu-Rub, A. Iqbal, and J. Guzinski, HIGH PERFORMANCE CONTROL OF AC DRIVES WITH MATLAB/SIMULINK MODELS. John Wiley and Sons, Ltd, 2012.

[3] Kiran Y, Puttaswamy PS. Field oriented control of a permanent magnet synchronous motor using a DSP. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014;**3**(10):12364-78.

[4] P. Pillay and R. Krishnan, "Application Characteristics of Permanent Magnet Syncronous and Brushless dc Motors for Servo Drives", IEEE Transactions on Industry Applications, vol. 21, no. 5, pp. 986–996, 1991.

[5] Merzoug MS, Naceri F. Comparison of field-oriented control and direct torque control for permanent magnet synchronous motor (PMSM). World Academy of Science, Engineering and Technology. 2008 Nov;45:299-304.

[6] Benjak O, Gerling D. Review of position estimation methods for IPMSM drives without a position sensor part II: adaptive methods. In The XIX International Conference on Electrical Machines-ICEM 2010 2010 Sep 6 (pp. 6-8). IEEE.

[7] Glumineau A, de Len Morales J. Sensorless AC electric motor control. Cham: Springer International Publishing. 2015.

[8] Nahid-Mobarakeh B, Meibody-Tabar F, Sargos FM. Back EMF estimation-based sensorless control of PMSM: Robustness with respect to measurement errors and inverter irregularities. IEEE Transactions on Industry Applications. 2007 Mar 19;**43**(2):485-94.

[9] Hassan M, Mahgoub O, El Shafei A. ANFIS based MRAS speed estimator for sensorless control of PMSM. In 2013 Brazilian Power Electronics Conference 2013 Oct 27 (pp. 828-835). IEEE.

[10] Kang J, Hu B, Liu H, Xu G. Sensorless control of permanent magnet synchronous motor based on extended Kalman filter. In 2009 IITA International Conference on Services Science, Management and Engineering 2009 Jul 11 (pp. 567-570). IEEE.

[11] Lv CX, Gao Q. PMSM Sensorless Field-Oriented Control System Based on Sliding-Mode Observer. In Applied Mechanics and Materials 2014 (Vol. 667, pp. 411-416). Trans Tech Publications.

[12] Zhou F, Li B, Yang J, Yan R. A sliding model speed/position observer integrated with a PI controller for PM synchronous motors. In 2007 IEEE International Conference on Robotics and Biomimetics (ROBIO) 2007 Dec 15 (pp. 1372-1377). IEEE.

[13] Mishra A, Mahajan V, Agarwal P, Srivastava SP. MRAS based estimation of speed in sensorless PMSM drive. In 2012 IEEE Fifth Power India Conference 2012 Dec 19 (pp. 1-5). IEEE.

[14] Shah R, Gajjar R. A comparative study of various methods for parameter estimation of PMSM. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) 2017 Aug 1 (pp. 1712-1715). IEEE.

[15] Pillay P, Krishnan R. Modeling of permanent magnet motor drives. IEEE Transactions on industrial electronics. 1988 Nov;**35**(4):537-41.

[16] Jianying Xu ,Ting Wu1, Weizhi Gu,Ying Wang and Riyao Che.A Position Observer Design for Interior Permanent Magnet Synchronous Motor Based on Fuzzy PI Model Reference Adaptive Control.The 30th Chinese Control and Decision Conference(CCDC 2018),2018,4798-4803.

[17] Samat AA, Ishak D, Iqbal S, Tajudin AI. Implementation of Sugeno FIS in model reference adaptive system adaptation scheme for speed sensorless control of PMSM. In 2014 IEEE International Conference on Control System, Computing and Engineering

[18] Kumar NS, Sadasivam V, Asan Sukriya HM. A comparative study of PI, fuzzy, and ANN controllers for chopper-fed DC drive with embedded systems approach. Electric Power Components and Systems. 2008 Jun 17;**36**(7):680-95.

[19] Badini Sai Shiva, Vimlesh Verma and Yawer Abbas Khan,Q-MRAS Based Speed Sensorless Permanent Magnet Synchronous Motor Drive with Adaptive Neural Network for Performance Enhancement at Low Speed.Simulation Modeling Practice and Theory, Elsevier 2019,103-116.

[20] Fan S, Luo W, Zou J, Zheng G. A hybrid speed sensorless control strategy for PMSM Based on MRAS and Fuzzy Control. In Proceedings of The 7th International Power Electronics and Motion Control Conference 2012 Jun 2 (Vol. 4, pp. 2976-2980). IEEE.

[21] Mohan N. Advanced electric drives: analysis, control, and modeling using MATLAB/Simulink. John Wiley & sons; 2014 Jul 22.

[22] Shyu KK, Lai CK, Tsai YW, Yang DI. A newly robust controller design for the position control of permanent-magnet synchronous motor. IEEE Transactions on Industrial Electronics. 2002 Aug 7;**49**(3):558-65.

[23] T. Instruments, InstaSPIN Projects and Labs User's Guide, Version 1.0.14, Texas Instruments.

[24] Landau I. A hyperstability criterion for model reference adaptive control systems. IEEE Transactions on Automatic Control. 1969 Oct;**14**(5):552-5.

[25] Xingming Z, Xuhui W, Feng Z, Xinhua G, Peng Z. Wide-speed-range sensorless control of interior PMSM based on MRAS. In 2010 International Conference on Electrical Machines and Systems 2010 Oct 10 (pp. 804-808). IEEE.

[26] Prakash CB, Naik RS. Tuning of PID Controller by Ziegler-Nichols Algorithm for Position Control of DC Motor. International Journal of Innovative Science, Engineering & Technology. 2014 May 3;1(3):379-82.

[27] Sheel S, Gupta O. New techniques of PID controller tuning of a DC motor—development of a toolbox. MIT IJEIE. 2012 Aug;2(2):65-9.

[28] Harsh Kukreja, N. Bharath, Siddesh C. S., Kuldeep S. An introduction to artificial neural network. In International Journal Of Advance Research And Innovative Ideas In Education 2016 (Vol-1 Issue-5),pp. 27-30.

[29] Mahdi Akhoondzadeh and Kamal Azizi. Artificial neural network(ANN), research gate, Presentation 03 November 2019, *https* : *//www.researchgate.net/publication/*336999943.

[30] Layth Abdulbari Al-Jaberi, Chapter three artificial neural networks, research gate, 11 September 2018, *https* : *//www.researchgate.net/publication/*327573940.

[31] Karlik,B.,and Olgac, A.V. Performance analysis of various activation functions in generalized MLP architectures of artificial intelligence and expert systems. International Journal of Artificial Intelligence And Expert Systems (IJAE)· February 2011, 1(4), 111-122.

[32] Zegai ML, Bendjebbar M, Belhadri K, Doumbia ML, Hamane B, Koumba PM. Direct torque control of Induction Motor based on artificial neural networks speed control using MRAS and neural PID controller. In2015 IEEE Electrical Power and Energy Conference (EPEC) 2015 Oct 26 (pp. 320-325). IEEE.

[33] Gaur P, Singh B, Mittal AP. Artificial neural network based controller and speed estimation of permanent magnet synchronous motor. In 2008 Joint International Conference on Power System Technology and IEEE Power India Conference 2008 Oct 12 (pp. 1-6). IEEE.

[34] Abu-Rub H, Awwad A. Artificial neural networks and fuzzy logic based control of AC motors. In 2009 IEEE International Electric Machines and Drives Conference 2009 May 3 (pp. 1581-1586). IEEE.

[35] Kumar NS, Sadasivam V, Asan Sukriya HM. A comparative study of PI, fuzzy, and ANN controllers for chopper-fed DC drive with embedded systems approach. Electric Power Components and Systems. 2008 Jun 17;36(7):680-95.

[36] Bobek V. PMSM electrical parameters measurement. Freescale Semiconductor. Feb 2013,7(8):13.

[37] Singh, Shweta, A.N. Tiwari, and S.N. Singh. Sensorless Speed Estimation of PM Synchronous Motor Drive using MRAS. In 2017 IEEE International WIE con- ference on electrical and computer engineering (WIECON-ECE) 2017 December 18-19 (pp. 35-38). IEEE.

[38] Beale MH, Hagan MT, Demuth HB. Neural network toolbox. User's Guide, Math- Works. 2010 Sep;2:77-81.

[39] W. Lipeng, Z. Huaguang, and L. Xiuchong, "Robust sensorless of ADRC controlled PMSM based on MRAS with stator resistance identification", in Control Conference (CCC), 2011 30th Chinese, 2011, pp. 3575–3579.

Appendix A

Adaptation Mechanism

Formulation of the adaptation mechanism plays a significant role on the performance of the MRAS. The adaptation mechanism is formed according to the following steps.

Step 1: Formulation of the equivalent model In this step, the MRAS is transformed into an equivalent model composed of linear time invariant (LTI) feedforward block and a linear time varying (LTV) feedback block, as shown in Figure A.1.

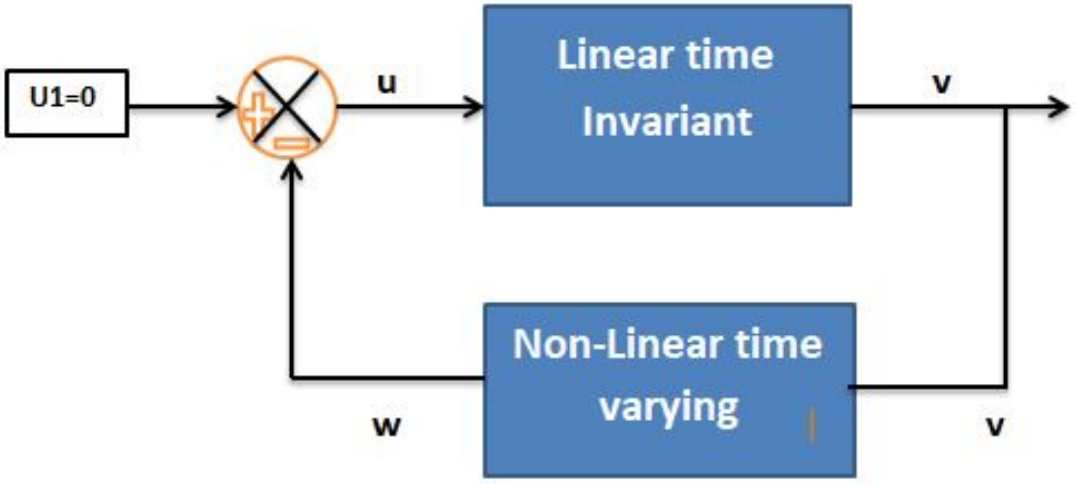


FIGURE A.1: Nonlinear time varying feedback system.

The generalized error is given by $\epsilon = \widehat{X} - X$ and using Equations 3.9 and 3.17 gives A.1

$$\begin{aligned} \dot{\epsilon} &= \dot{\widehat{X}} - \dot{X} \\ &= (\widehat{A}\widehat{X} + \widehat{B}u) - (AX + Bu) \\ &= \widehat{A}(\widehat{X} - X) + (\widehat{A} - A)X + \widehat{B}u) - Bu \\ &= \widehat{A}\epsilon + \omega_1 \end{aligned} \tag{A.1}$$

let ω =-w_1

$$\begin{aligned} \omega &= \left[(\widehat{A} - A)X + \widehat{B}u) - Bu\right] \\ &= \begin{bmatrix} -i_q \\ i_d + \frac{\lambda_r}{L_s} \end{bmatrix} (\widehat{\omega} - \omega) \\ &= K(\widehat{\omega} - \omega) \end{aligned} \tag{A.2}$$

Introducing the generalized error output as

$$v = D\epsilon \tag{A.3}$$

A.1 Popov hyper-stability theorem

Given a system of this form, three conditions must be met to guarantee that Equations A.1-A.3 is an asymptotically hyperstable system. These conditions are [24], [39]:

1. The pair [D,$\widehat{A}$] is completely observable and the pair [$\widehat{A}$,I] is completely controllable.
2. The transfer function $Z(s) = D(sI - \widehat{A})^{-1}$ must be strictly positive real.
3. The nonlinear time-varying block satisfies the Popov's integral inequality Equation A.8.

$$\int_0^t v^T w dt \geq -\gamma^2 \tag{A.4}$$

Here,γ^2 is a positive constant, independent of t.

Since rank[I $\quad \widehat{A}$I] = 2 and rank[$D^T \quad \widehat{A}^T \; D^T$] = 2, condition 1 is fulfilled.

According to [39], condition 2 is satisfied if there exist two definite symmetric matrices, P and Q such that

$$\widehat{A}^T P + P\widehat{A} = -Q \tag{A.5}$$

Choosing $Q = I$ yields

$$P = \begin{bmatrix} \frac{L_s}{2R_s} & 0 \\ 0 & \frac{L_s}{2R_s} \end{bmatrix}$$

Both P and Q are symmetrical and definite then condition 2 is proven.

As for condition 3, according to [**?**], to verify Equation A.8, another integral inequality can be used, given by

$$\int_0^t \left[\frac{df(t)}{dt}\right](kf(t))dt \geq -\frac{1}{2}kf^2(0) \tag{A.6}$$

where k is a positive constant.

Inserting v and w into Equation A.8 gives

$$\int_0^t \left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right](\widehat{\omega_e} - \omega_e)dt \geq -\gamma^2 \tag{A.7}$$

As the adjustable parameter $\widehat{\omega_e}$ which has converged to the real parameter ω_e in reference model is necessary to vector control, there should be "memory" in adaptive mechanism. Then the integrator can be adopted and $\widehat{\omega_e}$ can be assumed as follow

$$\widehat{\omega_e} = \int_0^t \phi(e)dt + \widehat{\omega_e}(0) \tag{A.8}$$

Where,$\phi(e)$ is an expression relative to generalized error e.Then the inequality A.7 will transform to inequality A.9.

$$\int_0^t \left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right](\int_0^t \phi(e)dt + \widehat{\omega_e}(0) - \omega_e)dt \geq -\gamma^2 \tag{A.9}$$

According to the inequality A.6, it is easy to make assumptions described by equation (15) and (16).

$$\frac{df(t)}{dt} = (\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q \tag{A.10}$$

$$kf(t) = \int_0^t \phi(e)dt + \widehat{\omega_e}(0) - \omega_e) \tag{A.11}$$

Where $f(t)$ is a function for the use of derivation, $\widehat{\omega_e}(0)$ is initial estimated rotor speed. Then the derivation from equation A.11 will be obtained:

$$\phi(e) = k\left((\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right) \tag{A.12}$$

Therefore the estimated speed will be obtained by combining equation A.8 and A.12

$$\widehat{\omega_e} = \int_0^t k\left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right]dt + \widehat{\omega_e}(0) \tag{A.13}$$

or the transfer function representation of equation A.13 is:

$$\widehat{\omega_e} = \frac{k}{s}\left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right]dt + \widehat{\omega_e}(0) \tag{A.14}$$

In order to improve the dynamic performance, the proportional regulation can be added into the adaptive mechanism Ḟinally the adaptive mechanism is a PI regulator shown by equation A.15:

$$\widehat{\omega_e} = \left(k_p + \frac{k_i}{s}\right)\left[(\widehat{i_q} - i_q)(i_d + \frac{\lambda_r}{L_s}) - (\widehat{i_d} - i_d)i_q\right]dt + \widehat{\omega_e}(0) \tag{A.15}$$