DESIGN AND PERFORMANCE ANALYSIS OF NEURAL NETWORK BASED SPEED ESTIMATORS FOR SENSORLESS VECTOR CONTROLLED INDUCTION MOTOR DRIVES

A THESIS

Submitted by

K. SEDHU RAMAN

In fulfilment for the award of the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND ELECTRONICS ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
PONDICHERRY ENGINEERING COLLEGE
PUDHUCHERRY - 605 014

PONDICHERRY UNIVERSITY

August 2013

BONAFIDE CERTIFICATE

Certified that this thesis titled **"DESIGN AND PERFORMANCE ANALYSIS OF NEURAL NETWORK BASED SPEED ESTIMATORS FOR SENSORLESS VECTOR CONTROLLED INDUCTION MOTOR DRIVES"** is the bonafide work of **Mr.K.Sedhuraman** who carried out the research under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. S.Himavathi
Professor
Department of Electrical and Electronics Engineering
Pondicherry Engineering College
Puducherry – 605 014.

Dr. A.Muthuramalingam
Professor
Department of Electrical and Electronics Engineering
Pondicherry Engineering College
Puducherry – 605 014.

DECLARATION

I hereby declare that the thesis entitled **"DESIGN AND PERFORMANCE ANALYSIS OF NEURAL NETWORK BASED SPEED ESTIMATORS FOR SENSORLESS VECTOR CONTROLLED INDUCTION MOTOR DRIVES"** submitted to the Pondicherry University for the award of the degree of **Doctor of Philosophy** in **Electrical and Electronics Engineering,** is a bonafide record of original research work done by me under the supervision and guidance of **Dr.S.Himavathi** and **Dr.A.Muthuramalingam,** Professors, Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Pondicherry.

Place: Pondicherry

Date: **(K.Sedhuraman)**

Acknowledgement

My wholehearted thanks to the ALMIGHTY for enabling me to do my task successfully.

This research work was carried out in the Research Laboratory at Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Pondicherry. Numerous souls have given lot of encouragement during this work. It is my privilege to express my gratitude to all of them.

Most of all, I would like to express my sincere gratitude to my supervisors and philosophers, Dr.S.HIMAVATHI and Dr.A.MUTHURAMALINGAM, Professors, Department of Electrical and Electronics Engineering for their encouragement, guidance and support during this research work. Further more their constructive and precise comments on my research work were always invaluable.

I express my sincere thanks to the doctoral committee members Dr.K.MANIVANNAN, Professor, Department of Electrical and Electronics Engineering and Dr.S.SUNDARAMOORTHY, Professor, Department of Chemical Engineering, Pondicherry Engineering College, Pondicherry for their continuous monitoring and critical assessment.

I would like to express my sincere thanks to Dr.D.GOVINDARAJULU, Principal, Pondicherry Engineering College for extending the college facilities.

I am gratified to Dr.R.GNANADASS, Professor and Head, Department of Electrical and Electronics Engineering, Pondicherry Engineering College, Pondicherry for his support during this research work.

The hardware validation has been carried out using the research grant from All India Council for Technical Education (AICTE). I express my wholehearted thanks to the AICTE, a statutory body of Government of India**,** for granting the research project titled “AI Techniques for Electrical Drives”. File Number: No 8023/BOR/RID/RPS-79/2007-08 and 8020/RID/TAPTEC-32/2001-02.

I owe my special thanks to my friend and colleague, Mr.A.VENKADESAN, for his many useful comments and help during each stage of this research work. I also thank all teaching and non teaching staff of the Department of Electrical and Electronics Engineering for their kind support and co-operation. I profusely thank Mrs.S.GEETHA, Assistant Professor, Department of Humanities and Social Sciences, Pondicherry Engineering College, Pondicherry for her valuable suggestions in language editing.

Finally, I would like to express my deepest gratitude to my dear family, for their endless support, without whom I would not have succeeded. My thanks are due to all kind-hearted souls who directly or indirectly contributed to the progress of this research.

K.SEDHURAMAN

Contents

Chapter No. **Title** **Page No.**

Chapter No.	Title	Page No.

Chapter No.	Title	Page No.

Chapter No.	Title	Page No.

List of Figures

Figure No.	Title	Page No
6.8	Estimated Rotor Speed for 8-bit precision	109
6.9	Estimated Rotor Speed for 12-bit precision	110
6.10	Estimated Rotor Speed for 16-bit precision	110
6.11	Estimated Rotor Speed for 32-bit precision	111
6.12	Estimated Rotor Speed for 64-bit precision	111
6.13	FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator with Layer Multiplexing	113
6.14	Schematic for Implementation of SNC-NN based Speed Estimation with Layer Multiplexing	115
6.15	Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 1	118
6.16	Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 2	119
6.17	Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 3	120
6.18	Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 4	121
C.1	Neural Learning Adaptive Model (NLAM)	143

List of Tables

Table No.	Title	Page No.

List of Symbols and Abbreviations

V_{ds}	d-axis stator voltage in Volts
V_{qs}	q-axis stator voltage in Volts
I_{ds}	d-axis stator current in Ampere
I_{qs}	q-axis stator current in Ampere
λ_{ds}	d-axis stator flux in Weber
λ_{qs}	q-axis stator flux in Weber
λ_{dr}	d-axis rotor flux in Weber
λ_{qr}	q-axis rotor flux in Weber
R_s	Stator Resistance in Ohms
R_r	Stator Resistance in Ohms
L_s	Stator Inductance in Henry
L_r	Rotor Inductance in Henry
L_m	Magnetization Inductance in Henry
T_r	Rotor Electrical Time Constant in second
σ	Total leakage factor
J	Moment of Inertia in kgm^2
B	Frictional coefficient in kgm^2/s
ω_e	stator frequency in radians/second
ω_{sl}	slip frequency in radians/second
ω_r	electrical rotor speed in radians/second
Q	reactive power
ξ	Error Signal
F	Energy Function
w, W	Weight
b	Bias
ref	reference
est	estimated

IM	Induction Motor
IFOC	Indirect Field Oriented Control
LPF	Low Pass Filter
MRAS	Model Reference Adaptive System
Q-MRNLAS	Reactive Power based Model Reference Neural Learning Adaptive System
PI	Proportional Integral Controller
PC	Personal Computer
NN	Neural Network
FF	Feedforward Network
SLFF	Single Layer Feed-Forward
MLFF	Multilayer Feed-Forward
SNC-NN	Single Neuron Cascaded Neural Network
BP	Backpropagation algorithm
BPM	Backpropagation algorithm with momentum
VLR	Variable learning rate algorithm
LM	Levenberg-Marquardt algorithm
MSE	Mean Square Error
DSP	Digital Signal Processor
ASIC	Application Specific Integrated Circuits
FPGA	Field Programmable Gate Array
VHDL	Very Hard Descriptive Language
NSE	Neural Speed Estimator

Abstract

Induction motors are applied today, to a wide range of applications requiring variable speed. Accurate speed measurement is necessary to realize high performance and high-precision speed control of an induction motor. The speed can be measured with optical encoders, electromagnetic sensors or brushless DC tacho-generators. However, the use of these electromechanical devices present some limitation in their applications, like increased cost of the drive, reduces mechanical robustness, low noise immunity etc. They affect the machine inertia and require special attention in hostile environments. To overcome the drawbacks, it is necessary to go in for speed estimators. Fast and accurate speed estimators are currently an active area of research.

Since the late 1980s, speed-sensorless control methods of induction motors using the estimated speed instead of the measured speed have been reported. They have estimated speed from the instantaneous values of stator voltages and currents using induction motor model. Other approaches to estimate speed use rotor slot harmonic, extended Kalman filter (EKF), extended Luenbergern observer (ELO), saliency techniques and model reference adaptive system (MRAS). MRAS schemes offer simpler implementation and require less computational effort compared to other methods and are therefore the most popular among the strategies used for sensorless control IM drives. The MRAS structure with PI controller as the adaptive mechanism uses the rotor flux, back emf, stator currents or reactive power as state variable. The selection of reactive power as a function for MRAS based speed estimator deduces to a simpler system model, which is easier to design and implement and therefore advantageous on real time applications.

Recently, the use of Neural Networks (NNs) for identification and control of nonlinear dynamic systems in power electronics and drives have been proposed as they are capable of approximating wide range of nonlinear functions to any desired degree of accuracy. Powerful learning algorithms have been developed for neural networks. Hence, NN based speed estimation is a good alternative to conventional method and is presently an active area of research.

The objective of the research work is to identify the best possible NN based solution for on-line speed estimation for sensorless vector controlled IM Drives. The use of neural learning algorithm in the MRAS adaptation and data based model approaches from input/output data are investigated.

The powerful neural learning algorithm is applied for adaptation in MRAS based speed estimation using simplified reactive power technique. The proposed scheme combines the advantages of simplified reactive power technique and the capability of neural learning algorithm for on-line speed estimation in sensorless indirect vector controlled induction motor drives. The performance of proposed on-line speed estimator is compared with existing Q-MRAS in terms of accuracy and regenerating mode of operation in sensorless indirect vector controlled induction motor drives.

To further exploit the advantages of neural networks it is proposed to build data based model from input/output data for on-line speed estimators. The objective is to design an accurate, simple and structurally compact, computationally less complex NN model which is easy to design, simple to implement and robust to parameter variation. Three different data based NN models are proposed and developed for on-line speed estimation. Neural speed estimator (NSE) model I depends on flux. In this NN Model, the types of neural architectures and learning algorithms are considered for investigation. The proposed single neuron cascaded neural network (SNC-NN) model trained with Levenberg Marquardt algorithm is found to be more compact, less complex, easy to design and accurate. The SNC-NN model trained with Levenberg Marquardt algorithm is identified to be the most suitable model for on-line speed estimation in sensorless controlled IM drives and is named as neural speed estimator model I. The performance of proposed neural speed estimator model I is compared with existing RF-MRAS in terms of accuracy. It performs as good as equation based models but suffers from all the drawbacks of flux estimation as flux is used as input. To overcome the drawbacks, neural speed estimator model II is proposed.

The proposed neural speed estimator model II is based on simplified reactive rower equation and it uses stator frequency as one of the input to the neural model. Neural speed estimator model II combines the advantages of simplified reactive power based techniques with the novelty of robustness to R_r variation and provides stability over a wide operating range in sensor-less indirect vector controlled induction motor drives.

However, it is suitable for only sensor-less indirect vector controlled induction motor drives. Also the computation of stator frequency is done using the machine model equation. Hence, an accurate knowledge of machine parameters is mandatory. The generalized model using reactive power as input (neural speed estimator model III) is proposed for all types of sensorless controlled IM drives. The proposed neural speed estimator model III is compact, less complex, easy in design, accurate and robust to R_s and R_r variations. It has the advantage that it is suitable for all types of sensorless control, does not require knowledge of motor parameters and does not require any additional estimators for speed estimation. Hence, this thesis proposes the neural speed estimator model III for on-line speed estimation as the most suitable estimator for sensorless vector controlled IM drives and it is verified using hardware

A 3-hp three phase induction motor is considered for practical study. The experimental set up is built and practical data is obtained. The proposed SNC-NN based NSE model III for on-line speed estimator is trained and tested for real time data. The proposed speed estimator is implemented on FPGA (Field Programmable Array Logic). Hence, the implementation issues of proposed neural based on-line speed estimator in FPGA are investigated in terms of computational complexity, bit precision and effective utilization of resources. The most time consuming block in the FPGA implementation of neural architecture is the computation of tan-sigmoid function as it is an infinite series. To overcome this difficulty, a new nonlinear simple to compute activation function called Elliott function is used. The neural speed estimator model III is trained with Elliott function in the hidden layer for the real time data. The NSE model III with Elliott function performs as good as the NSE model III with tan-sigmoid function for speed estimation. To reduce the resource, a layer multiplexing technique is adopted. The lowest bit precision needed for good performance of the NSE model III with Elliott function based speed estimator is also identified.

The performance of the proposed NSE model III with identified bit precision and excitation function is implemented using layer multiplexing technique and tested on Xilinx Spartan FPGA kit (3sd1800afg676-4). These investigations would lead to the development of a cheaper, efficient and intelligent speed estimator for sensorless controlled IM drives.

Appendix A

The parameters of the induction machine used for simulation from chapter 2 to 5 are given in the table below.

Induction Motor Parameters

Parameters	***Values***	***Parameters***	***Values***
Rated Power	1.1kW	Stator Resistance (R_s)	6.03Ω
Rated voltage	415V	Rotor Resistance (R_r)	6.085Ω
Rated current	2.77A	Magnetizing Inductance (L_m)	0.4893H
Type	3 Ph	Stator Inductance (L_s)	0.5192H
Frequency	50Hz	Rotor Inductance (L_r)	0.5192H
Number of poles	4	Rated Speed	1415RPM

The parameters of the real time induction machine used for experimental setup in chapter 6 are given in the table below.

Induction Motor Parameters

Parameters	***Values***	***Parameters***	***Values***
Rated Power	2.2kW	Stator Resistance (R_s)	4.985Ω
Rated voltage	415V	Rotor Resistance (R_r)	3.432Ω
Rated current	4.7A	Magnetizing Inductance (L_m)	0.3229H
Type	3 Ph	Stator Inductance (L_s)	0.0138H
Frequency	50Hz	Rotor Inductance (L_r)	0.0138H
Number of poles	4	Rated Speed	1430RPM

Appendix B

Synthesis report summary is shown below for proposed speed estimation.

```
=========================================================
                Synthesis Options Summary
=========================================================
---- Source Parameters
Input File Name                    : "NTK.prj"
Input Format                       : mixed
Ignore Synthesis Constraint File   : NO
---- Target Parameters
Output File Name                   : "NTK"
Output Format                      : NGC
Target Device                      : xc3sd1800a-4-fg676
---- Source Options
Top Module Name                    : NTK
Automatic FSM Extraction           : YES
FSM Encoding Algorithm             : Auto
Safe Implementation                : No
FSM Style                          : lut
RAM Extraction                     : Yes
RAM Style                          : Auto
ROM Extraction                     : Yes
Mux Style                          : Auto
Decoder Extraction                 : YES
Priority Encoder Extraction        : YES
Shift Register Extraction          : YES
Logical Shifter Extraction         : YES
XOR Collapsing                     : YES
ROM Style                          : Auto
Mux Extraction                     : YES
Resource Sharing                   : YES
Asynchronous To Synchronous        : NO
Use DSP Block                      : auto
Automatic Register Balancing       : No
---- Target Options
Add IO Buffers                     : YES
Global Maximum Fanout              : 500
Add Generic Clock Buffer(BUFG)     : 24
Register Duplication               : YES
Slice Packing                      : YES
Optimize Instantiated Primitives   : NO
Use Clock Enable                   : Yes
Use Synchronous Set                : Yes
Use Synchronous Reset              : Yes
Pack IO Registers into IOBs        : auto
Equivalent register Removal        : YES
---- General Options
```

```
Optimization Goal                     : Speed
Optimization Effort                   : 1
Library Search Order                  : NTK.lso
Keep Hierarchy                        : NO
Netlist Hierarchy                     : as_optimized
RTL Output                            : Yes
Global Optimization                   : AllClockNets
Read Cores                            : YES
Write Timing Constraints              : NO
Cross Clock Analysis                  : NO
Hierarchy Separator                   : /
Bus Delimiter                         : <>
Case Specifier                        : maintain
Slice Utilization Ratio               : 100
BRAM Utilization Ratio                : 100
DSP48 Utilization Ratio               : 100
Verilog 2001                          : YES
Auto BRAM Packing                     : NO
Slice Utilization Ratio Delta         : 5
---- Other Options
Cores Search Directories              : {"ipcore_dir"  }
```

```
=========================================================
HDL Synthesis Report
Macro Statistics
# ROMs                                                : 3
 46x32-bit ROM                                       : 2
 46x34-bit ROM                                       : 1
# Multipliers                                         : 4
 32x11-bit multiplier                                : 1
 32x32-bit multiplier                                : 2
 32x8-bit multiplier                                 : 1
# Adders/Subtractors                                  : 62
 32-bit adder                                        : 61
 7-bit adder                                         : 1
# Counters                                            : 2
 32-bit up counter                                   : 2
# Registers                                           : 146
 30-bit register                                     : 2
 32-bit register                                     : 143
 7-bit register                                      : 1
# Comparators                                         : 4
 32-bit comparator equal                             : 1
 32-bit comparator greatequal                        : 2
 32-bit comparator greater                           : 1
# Multiplexers                                        : 67
 1-bit 46-to-1 multiplexer                           : 64
 32-bit 46-to-1 multiplexer                          : 3
=========================================================
```

Appendix C

The neural network (NN) is obtained as shown in Figure C.1. The back-propagation learning rule with momentum is used to minimize the energy function. (F) All neurons have linear activation function. The equations defining the neural learning adaptive model (NLAM) based on reactive power is shown in Figure C.1 (Vas. P., 1998)

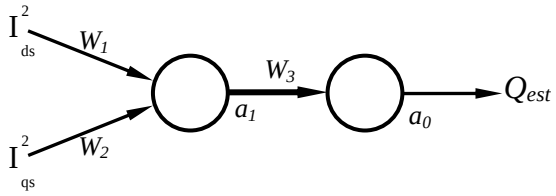


Figure C.1 Neural Learning Adaptive Model (NLAM)

The reference reactive power $Q_{ref} = V_{qs} I_{ds} - V_{ds} I_{qs}$ (C.1)

and estimated reactive power $Q_{est} = W_3 \left[W_1 I_{ds}^2 + W_2 I_{qs}^2 \right]$ (C.2)

where, $W_1 = \sigma L_s + \left(L_m^2 / L_r \right)$; $W_2 = \sigma L_s$; $W_3 = \omega_e$; $a_1 = W_1 I_{ds}^2 + W_2 I_{qs}^2$ and $a_0 = Q_{est}$.

In NN shown in Figure C.1, the adaptive weight W_3 is shown with thick solid line, and is proportional to the speed. The adaptive weight is adjusted so that energy function (F) should be a minimum, and the error is ($\xi(k)$). Thus the weight adjustments to give minimum squared error have to be proportional to the negative of the gradient of the error with respect to the weight.

$$F = \frac{1}{2}\left(Q_{ref} - Q_{est}\right)^2 \tag{C.3}$$

$$\xi(k) = Q_{ref} - Q_{est} \tag{C.4}$$

$$\Delta W_3(k) = -\alpha \frac{\partial F}{\partial W_3} \tag{C.5}$$

The equation (C.5) can be rewritten as,

$$\Delta W_3(k) = -\alpha \frac{\partial F}{\partial Q_{est}} \times \frac{\partial Q_{est}}{\partial W_3} \tag{C.6}$$

$$\frac{\partial F}{\partial Q_{est}} = -\left(Q_{ref} - Q_{est}\right) = -\xi(k) \tag{C.7}$$

$$\frac{\partial Q_{est}}{\partial W_3}=\left[W_1 I_{ds}^2 + W_2 I_{qs}^2 \right]=a_1 \tag{C.8}$$

Substituting equation (C.7) and (C.8) in (C.6)

$$\Delta W_3(k)=\alpha\xi(k)a_1 \tag{C.9}$$

To obtain W_3(k),

$$W_3(k)=W_3\left(k-1\right)+\Delta W_3(k) \tag{C.10}$$

The weight adjustments require choice of learning rate (α.). If α is large, it will lead to oscillations in the output. To overcome this difficulty a momentum (η) added to the equation (C.10), which takes the past (k-1)th weight changes on the (k)th weight. This ensures accelerated convergence of the algorithm. Thus the current weight adjustment W_3(k) is described as

$$W_3(k)=W_3(k-1)+\Delta W_3(k)+\eta\Delta W_3(k-1) \tag{C.11}$$

The inclusion of the momentum term accelerates convergence. Also, the updated W_3 gives the estimated stator frequency (ω_e). From ω_e the estimated rotor speed (ω_r) is calculated using the relation $\omega_r = \omega_e - \omega_{sl}$.

Chapter 1

Introduction

1.1 General

An important feature of worldwide industrial progress during the past several decades has been enhancing of factory automation. The manufacturing lines in an industrial plant typically involve one or more variable speed motor drives which serve to power conveyor belts, robot arms, overhead cranes, steel process lines, paper mills, plastic and fiber processing lines to name a few. Prior to 1850s, all such applications required the use of a DC motor since AC motors were not capable of true adjustable or smoothly varying speed since they inherently operated synchronously or nearly synchronously with the frequency of electrical input. But, after the invention of inverters and its various control techniques in latter in 1920s, AC drives slowly began to replace DC motor drives in the above listed applications.

Now, three phase induction machines are widely used in industries for variable speed application due to their ruggedness and low price. The induction motor can be operated directly from mains, but the variable speed and energy efficiency are achieved by means of a frequency converter between mains and the motor. A typical frequency converter consists of a rectifier, a voltage –stiff dc link or current stiff dc link and a pulse width modulated (PWM) inverter.

The variable speed control methods of induction motor are basically classified into two types: scalar control and vector control. The vector control is further classified into field oriented vector control and direct torque control. The vector control method is used for high performance drive. The overview of the variable speed control methods for induction motor is shown in Figure 1.1.

In the past, induction motors were controlled using scalar control methods. Scalar control, as the name indicates, is due to magnitude variation of the control variables

only, and disregards the coupling effect in the machine. For example, the voltage of a machine can be controlled to control the speed and frequency or slip can be controlled to control the torque. Scalar control is somewhat simpler to implement, but the inherent coupling effect (i,e., both torque and flux are functions of voltage/current and frequency) gives poor and sluggish dynamic response. This scheme can control the speed of the motor satisfactorily under steady state only.

A major revolution in the area of induction motor based drives was the invention of field oriented control (FOC) or vector control (VC) in the late 1960's (Blaschke F., 1972). In variable speed control applications, in which a small variation of motor speed with loading is tolerable, a scalar control system can produce adequate performance. However, if precision control is required, then a vector control system is the promising alternative (Blaschke F., 1972). In vector control, both the magnitude and phase alignment of vector variables are controlled and valid for steady state as well as transient conditions. Thus, the vector control method is a better option than the scalar control to obtain the desired dynamic performance. With the advent of vector control schemes, the control of an induction motor is transformed similar to the control of a separately excited DC motor by creating independent channels for flux and torque control.

Depending on the method of measurement of field angle, the vector control can be divided into two sub categories: direct vector control (DVC) and indirect vector control (IDVC). If the field angle is computed using the flux, it is termed as direct vector control. If the field angle is computed using speed and slip, it is termed as indirect vector control. Indirect vector control is the most popular technique among the industry because the speed sensing is very easy when compared to flux. Hence, in this thesis, indirect vector control setup is considered. However, vector control requires accurate knowledge of rotor speed. Speed can either be measured or estimated. The instantaneous rotor speed (ω_r) can be measured using the speed sensing instruments namely DC tachogenerator, AC tachogenerator, encoders and electromagnetic resolvers etc.

1.2 Need for Speed Estimator in Vector Controlled IM Drives

Speed sensors are usually expensive, bulky, require extra cabling and subject to failures under hostile industrial environments (Holtz J., 2002; Seong-Hwan, et al 2001). Moreover at low powers (2 to 5 kW) the cost of the sensor is about the same as the motor. Even at 50 kW, it can still be between 20 to 30% of the machine cost. Therefore, the cost and size of the drive system is increased. The mounting of the sensor to the motor is also an obstacle in many applications. Hence, researchers have concentrated on the elimination of the speed sensor at the machine shaft without deteriorating the performance of drive control system (Krishnan R., 2000; Rajashekara K., et al 1996). Speed estimation is an issue of particular interest with induction motor drives where the mechanical speed of the rotor is generally different from the speed of the revolving magnetic field. The advantages of speed sensorless induction motor drives are reduced hardware complexity, lower cost, reduced size of the drive machine, elimination of the sensor cable, better noise immunity, good reliability and less maintenance requirements. The operation in hostile environments mostly requires a motor without speed sensor. There are various speed estimation methods available in the literature which is dealt in the following sections.

1.3 Review of Speed Estimation Techniques

The on-line methods of speed estimation developed so far could be broadly classified under the following categories.

- Conventional Methods
- Neural Network based techniques

The conventional methods of speed estimation are

  - Slip calculation method
  - Direct synthesis from State Equation
  - Observer Based Techniques
  - Slot Harmonics

- Saliency Technique
- Model Referencing Adaptive Systems (MRAS)

The rotor speed can be estimated using the slip calculation method and direct synthesis from state equation (Abbondanti A., and Brennen, M. B., 1975; Bose B.K., 2005; Vas P., 1998). Both the methods directly depend on motor parameters and will tend to give poor accuracy of estimation. The next classification of rotor speed identification methods can be grouped under Observer Based Techniques (Akun B., et al. 2004; Bodson M., et al. 1995; Du T., et al. 1995; Kim Y.R., et al. 1994; Marko Hinkkanen, et al. 2010; Vicente I., et al. 2010). This class of speed estimation technique is based on either extended Kalman filter (EKF) or extended Luenberger observer (ELO) (Bose B.K., 2005; Vas P., 1998). Here, the rotor time constant is treated as additional state variable along with rotor speed so that, the above methods can be used for joint state and parameter estimation efficiently. The authors have applied extended observer techniques for state and parameter estimation for high performance AC drives. However, the problems related to extended kalman filter (EKF) and extended Luenberger observer (ELO) are high computational complexity, large computation time and large memory requirement, which makes it difficult for real time implementation even with powerful DSP (Bose B.K., 2005).

Rotor slot harmonics spectrum estimation technique (Bose B.K, 2005; Ferrah A., et al 1997; Michael W. Degner and Robert D. Lorenz, 2000) is a kind of sensorless speed detection method. The rotor slot produces harmonic components in the air gap field, which modulate the flux interlacing on the stator with a frequency proportional to the rotor speed. Thus the speed can be estimated using the slot harmonics frequency. However, this method depends on the finite number of rotor slots, small reluctance variation, and the ripple frequency. The voltage magnitude become very low at low motor speeds and speed estimation becomes difficult. The saliency techniques (Briz F., et al., 2004; Caruana C., et al., 2006; Jansen P.L., and Lorenz R.D., 1996; Jung-Ik Ha, and Seung-Ki Sul, 1999; Schroedl M., 1996) attempt to be parameter independent and estimate the speed under zero or very low frequency also. They rely on the machine response to injected test signals; they require a high precision measurement and increase the overall complexity of the system. The secondary magnetic effects do lead to complications in their implementation.

MRAS schemes offer simpler implementation and require less computational effort compared to other methods and are therefore the most popular among the strategies used for speed estimation in sensor-less control IM drives (Jevremovic V.R., et al 2010; Kojabadi H.M., 2009; 2005; Maiti S., et al 2008; Maiti S., and Chakraborty C., 2010; Ohtani T., et al 1992; Peng F.Z., and Fukao T., 1994; Peng F.Z., et al 1994; Rashed M., et al 2004; Schauder C., 1992; Teresa Orlowska-Kowalska, and Mateusz Dybkowski, 2010).

In MRAS system, the outputs of two models, one independent of the rotor speed (reference model) and the other dependent (adjustable model) are used. The error vector is driven to zero by an adaptive mechanism (PI-controller), which yields the estimated rotor speed. Depending on the choice of output quantities that form the error vector (flux (Bose B.K., 2005; Ohtani T., 1992; Schauder C., 1992; Vas P., 1998), back EMF (Peng F.Z., and Fukao T., 1994; Rashed M., and Stronach A.F., 2004), reactive power (Jevremovic V.R, et al 2010; Maiti S., et al 2008; Maiti S., and Chakraborty C., 2010; Peng F.Z., et al 1994), active power (Kojabadi H.M., 2009), stator current (Teresa Orlowska-Kowalska, and Mateusz Dybkowski, 2010)), several MRAS structures are possible. A number of drawbacks exist with MRAS based speed estimation depending on the equation used, state variable chosen and type of vector control. Attempts to overcome the drawbacks are an active area of research. The selection of reactive power as a state for MRAS based speed estimator deduces to a simpler system model equation independent of flux and R_s, which is easier to design and implement and become advantage on real time applications (Jevremovic V.R, et al 2010; Maiti S., et al 2008; Maiti S., and Chakraborty C., 2010; Peng F.Z., et al 1994) for sensorless indirect vector controlled induction motor drives. MRAS schemes are also dependent on motor parameters. However, an induction motor is highly coupled, non-linear dynamic plant, and its parameters vary with time and operating conditions. Therefore, it is very difficult to obtain good performance for the entire speed range using existing methods.

Recently, the use of Neural Network (NNs) to identify and control nonlinear dynamic systems has been proposed because they can approximate a wide range of nonlinear functions to any desired degree of accuracy (Bose B.K., 2007; Chen T.C., and Sheu T.T., 2002; Dinko Vukadinović and Mateo Bašić, 2011; Heredia, J.R., et al., 2001;

Karanayil B., et al 2007; Lazhar Ben-Brahim, et al 1999; Lech M. Grzesiak, and Marian P. Kazmierkowski, 2007; Mondal S.K., et al 2002; Narendra K.S., and Part'ms arathy K., 1990; Seong-Hwan Kim, et al., 2001; Vas P. 1999). Powerful learning algorithms have been developed for neural networks (Fredric M. Ham and Ivica Kostanic 2008; Martin T.Hagan et al 2008). Moreover, they have the advantages of extremely fast parallel computation, immunity to noise and learning from input output data. Hence NN based speed estimation is a good alternative to conventional method.

The powerful learning algorithms are used for MRAS adaptation instead of PI technique and shown to exhibit advantages such as easy tuning and wide range of operation (Chen T., and Sheu T., 2002; Karanayil B., et al 2007; Lazhar Ben-Brahim et al., 1999; Lech M. Grzesiak and Marian P. Kazmierkowski, 2007; Seong-Hwan Kim et al., 2001; Vas P., 1998). Vas P. in 1998 a four layer neural model is used for speed estimation in induction motor. Thus NN based speed estimation is still an emerging area of research as the full potential of NN is yet to be exploited. Hence, in this thesis, a NN approaches is attempted to improve the performance and reduce the complexity of speed estimators.

1.4 Objectives of the Thesis

This research work attempts to identify the best possible NN based solution for on-line speed estimator used in sensorless vector controlled IM drives. The use of neural learning algorithm and data based neural model approaches are investigated. This thesis aims to form the basis for the development and realization of a robust on-line speed estimator based on neural network, and their usefulness in complex sensorless vector controlled IM drives.

The use of neural learning algorithm for adaptation in MRAS based on-line speed estimator using reactive power (Q-MRNLAS) is proposed and designed. The model is extensively tested over the complete operating range and its performance is to be obtained. It has to be compared with the conventional Q-MRAS using PI controller and the conclusions are to be summarized.

To overcome the drawbacks of equation based methods, this thesis proposes neural network models for on-line speed estimation. The nonlinear mapping capability of the neural network is to be exploited for the design of compact robust speed estimator. To identify the best architecture and learning algorithm for speed estimation, an investigation is to be carried out using different neural architectures and learning algorithms. The most suitable architecture and learning algorithm are to be identified based on comparison in terms of complexity and accuracy.

Different input combinations for on-line speed estimation are to be modeled and analyzed. The most suitable neural based on-line speed estimator which is robust, compact, less complex, easy in design, accurate and generalized so as to suit for all types sensorless controlled IM drives has to be identified.

The identified neural based on-line speed estimator has to be designed and validated for real time data. The implementation issues of the proposed speed estimator are to be addressed. From the investigations the major conclusions are to be summarized.

1.5 Thesis Overview

The research work reported in the thesis is organized into seven chapters. Each section deals with a contribution, presents the results obtained, and highlights the advantages and possible limitations.

Chapter 2 discusses the reactive power-MRAS using neural learning adaptation (Q-MRNLAS) for speed estimation in IFOC Drives. In MRAS method, the flux or reactive power is used as state variable and both require the knowledge of flux. For sensorless control, flux can be estimated using voltage model equations only and this method uses a pure integrator leading to integrator drift problems at low speeds. This problem can be overcome in IFOC drives by using reduced reactive power equations, which is independent of flux. The rotor and stator resistance vary during motor operation and have to be estimated online. The use of reduced reactive power equations for MRAS makes the speed estimation independent of flux and in-turn independent of stator resistance. Hence, this method eliminates the need for stator

resistance estimator and reduces the complexity of the drive. Conventionally the adaptive mechanism for MRAS is done using PI controller. However it fails in certain regenerative modes.

The powerful NN learning rules can be used for adaptive mechanism in MRAS instead of PI controller. A neural learning based reactive power MRAS called Q-MRNLAS is proposed in this chapter. Different learning algorithms are investigated. The simple and most suitable learning algorithm is identified. The performance of proposed on-line speed estimator is compared with existing Q-MRAS in terms of accuracy and regenerating mode of operation in sensorless indirect vector controlled induction motor drives and presented. The proposed simplified Q-MRNLAS is independent of flux, pure integrator problems and provides stable operation in regenerative mode. It eliminates the need for stator resistance estimation and hence reduces the complexity of the drive. Hence, it is concluded that the proposed estimator is a promising alternative for existing PI based Q-MRAS for on-line speed estimation in sensor-less indirect vector controlled induction motor drives.

The proposed method again suffers from all the disadvantages of conventional equations based schemes, namely inaccuracies in motor parameters used and the need for an online rotor resistance estimator. To further exploit the advantages of neural networks it is proposed to build a data based model for on-line speed estimation. This is investigated in the next chapter.

Chapter 3 proposes a data based NN for speed estimation. The data based approach using flux based NN model is discussed in this chapter and named as neural speed estimator model I. The data based approach has to identify the input parameters, choose a neural architecture which is simple and accurate, choose a learning algorithm for offline training and suggest the best method for implementation.

The NN model for speed estimation uses flux as one of the inputs. The inputs to estimator are direct and quadrature axis stator currents measured at $(k\text{-}1)^{th}$ sample $\{I_{ds}(k-1), I_{qs}(k-1)\}$ and rotor fluxes measured at k^{th} and $(k\text{-}1)^{th}$ sample $\{\lambda_{dr}(k), \lambda_{dr}(k-1), \lambda_{qr}(k), \lambda_{qr}(k-1)\}$. The output is the estimated rotor speed $\{\omega_r\}$ at k^{th}

sample. As accuracy is important, the activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively.

The design of data based NN speed estimator to a large extent depends on the type of neural architecture and the neural learning algorithm. Hence suitable neural architecture and NN learning algorithm have to be identified for on-line speed estimation. The popular neural architectures considered for investigation are feed-forward architecture and cascade architecture. The chosen NN architectures are trained with three different neural learning algorithms namely backpropagation with momentum (BPM), variable learning rate (VLR) and Levenberg Marquardt (LM).Training data for the NN, is taken at various operating conditions over the complete operating range. No parameter variation is assumed. For performance comparison, all the three NN architectures are trained with same input/output data, same learning algorithms, and same target MSE. This is repeated for different learning algorithms and the results obtained are reported. The performance comparison is further carried out in terms of structural compactness and computational complexity. Hence, from the investigations, it is concluded that the proposed "Single Neuron Cascaded Neural Network" (SNC-NN) model trained with LM algorithm is most suitable for speed estimation in sensorless controlled IM drives and it is named as neural speed estimator model I.

The performance of the proposed neural speed estimator model I based on rotor flux is shown to be as good as the rotor flux–MRAS method for all operating conditions. The data based NN approach is validated using this model. However the model is flux dependent and suffers from the drawbacks of flux estimation in sensorless induction motor drives. The reactive power model can be simplified to eliminate flux and R_s for IFOC drives. This advantage is used to build a data based model independent of rotor flux for IFOC drives and is detailed in the next chapter.

In ***Chapter 4***, a novel data driven stator frequency based neural network model for on-line speed estimation in IFOC Drives is proposed, which is independent of flux estimator, integration problems and parameter variations. The stator frequency based NN model (Model II) is presented. The inputs to neural network model are direct and quadrature axis stator voltages and currents measured at k^{th} sample

$\{V_{ds}(k), V_{qs}(k), I_{ds}(k), I_{qs}(k)\}$ and stator frequency measured at k^{th} sample $\{\omega_e\}$. The stator frequency can be estimated using simplified reactive power equations. The output is the estimated rotor speed $\{\omega_r\}$ at k^{th} sample. The activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively. Around 95,000 data sets are obtained through simulation is used as the training data set. The model is independent of R_s. The designed speed estimator is extensively tested for a number of operating conditions and compared with simplified Q-MRNLAS.

The proposed neural speed estimator model II combines the advantages of simplified reactive power based techniques with the novelty of robustness to R_r variation and stability over a wide operating range in sensor-less indirect vector controlled induction motor drives. However, it is suitable for only sensor-less indirect vector controlled induction motor drives. Also the computation of ω_e is done using the machine model equation. Hence, an accurate knowledge of machine parameters is mandatory.

A generalized data based model which is suitable for all types of IM drives control using a black box approach, and which does not require motor parameters would provide a universal solution by addressing all the above issues. This is envisaged and investigated in the next chapter.

Chapter 5 proposes a speed estimator NN model based on reactive power (Model III) derived from the generalized equations. This is suitable for all types of sensorless controlled IM drives. The speed is defined as a function of voltages, currents and reactive power. This model requires only measured voltages and currents. Motor parameters are not required. The inputs to the SNC-NN estimator are direct and quadrature axis stator voltages and currents measured at k^{th} sample $\{V_{ds}(k), V_{qs}(k), I_{ds}(k), I_{qs}(k)\}$ and reactive power measured at k^{th} sample $\{Q\}$. The output is the estimated rotor speed $\{\omega_r\}$. Around 95,000 data sets are obtained through simulation is used as the training data set. The online parameters variations are incorporated in the simulation model while obtaining the training datasets. However, in real time the datasets would include online parameter changes.

The designed SNC-NN model III for on-line speed estimator is extensively tested for a number of operating conditions in sensorless indirect vector controlled induction motor drive and shown to perform well. The performance of the proposed on-line speed estimation is compared with simplified Q-MRNLAS for 100% change of R_r in Sensorless Indirect Vector Controlled Induction Motor Drives and results thus obtained are presented.

The neural speed estimator model III is found to be more compact, less complex, easy in design, accurate, robust to motor parameters variation namely R_s and R_r and also provides stability over a wide operating range. Hence, this thesis proposes the neural speed estimator model III for on-line speed estimation is the most suitable estimator for sensorless vector controlled IM drives. The proposed neural speed estimator model III is identified as the good alternative for conventional method of on-line speed estimation and it is also suitable for all types of sensorless controlled IM drives. The proposed novel method is verified using hardware.

Chapter 6 deals with design and implementation of speed estimator using real time data. A 3-hp three phase induction motor is considered for practical study. The experimental set up is built and practical data is obtained. The proposed SNC-NN based model III for on-line speed estimator is trained and tested for real time data. The result validates the performance of neural model based speed estimator for practical data. From the result obtained, it is found that the estimated speed closely matches the measured speed.

The designed SNC-NN based speed estimator is implemented using digital hardware. The choice can be DSP or FPGA. For fast estimation, FPGA is chosen as the target hardware. The most time consuming block in the FPGA implementation of neural architecture, is the computation of tan-sigmoid function which is an infinite series. To overcome this difficulty, a new nonlinear simple to compute activation function called Elliott function is used. The neural speed estimator model III based speed estimator is trained with Elliott function in the hidden layer for the real time data. The neural speed estimator model III with Elliott function performs as well as the neural speed estimator model III with tan-sigmoid function for speed estimation. The use of Elliott function reduces computational complexity. To reduce the resource, a layer

multiplexing technique is adopted. The lowest bit precision needed for good performance of the neural speed estimator model III with Elliott function based speed estimator is also identified.

The performance of the proposed neural speed estimator model III with Elliott function based speed estimator with identified bit precision and excitation function is implemented and tested on Xilinx Spartan FPGA kit (3sd1800afg676-4). The concept of layer multiplexing is adopted for effective resource utilization. The FPGA implementation is tested with practical data extensively.

Chapter 7 concludes the thesis by summarizing the major contributions of the research, advantages of the proposed speed estimator and its suitability for all types of sensorless controlled IM drives. The directions for future work and possible extensions are outlined.

Chapter 2

Model Reference Neural Learning Adaptive Systems

2.1 Introduction

The Model Reference Adaptive System (MRAS) based speed estimation is one of the most popular method used for sensor-less controlled induction motor drives. MRAS scheme offers simpler implementation and requires less computational effort compared to other types of speed estimation techniques (Chen T.C., and Sheu T.T., 2002; Kojabadi, H.M., 2009; 2005; Maiti S., et al 2008; Maurizio Cirrincione, and Marcello Pucci, 2005; Ohtani T., et al 1992; Peng F.Z., and Fukao T., 1994; Shauder C., 1992; Teresa Orlowska-Kowalska, and Mateusz Dybkowski, 2010; Yang G., and Chin T.-H., 1993).

The block diagram of the MRAS based speed estimation scheme is shown in Fig 2.1. The state variable x is computed in two different ways using sensed variables such as stator voltage and current. One of them is independent of rotor speed (reference model) and the other is dependent on rotor speed (adaptive model). The difference between the outputs of these two models is used to formulate the error vector signal (ξ). The error vector signal is then fed to an adaptation mechanism. The output of the adaptation mechanism is the estimated quantity ($\omega_{r,est}$) which is adjusted in the adaptive model until the errors between the two models vanish to zero.

Depending on the state variable quantity used for the formulation of the error vector signal, various kinds of MRAS are available, e.g., flux (Bose B.K., 2005; Ohtani T., 1992; Schauder C., 1992; Vas P., 1998), back EMF (Peng F.Z., and Fukao T., 1994; Rashed M., and Stronach A.F., 2004), reactive power (Jevremovic V.R., et al 2010; Maiti S., et al 2008; Maiti S., and Chakraborty C., 2010; Peng F.Z., et al 1994), active power (Kojabadi H.M., 2009), stator current (Teresa Orlowska-Kowalska, and Mateusz Dybkowski, 2010) etc.

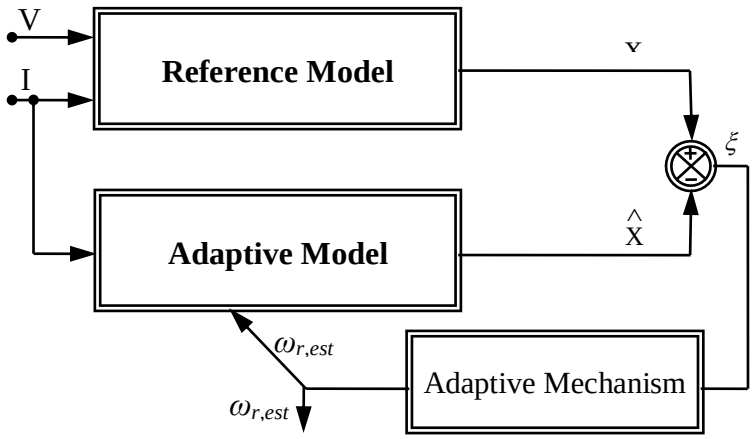


Figure 2.1 MRAS based Speed Estimation

The most popularly used MRAS schemes are rotor flux based MRAS (RF-MRAS) and reactive power based MRAS (Q-MRAS). In RF-MRAS scheme, the conventional voltage model equations are used as the reference model. Conventional voltage model suffers from the problems of pure integrator and variations of stator resistance especially at low speeds (Bose B.K., and Patel N.R., 1997; Bose, B.K., 2005; Holtz J., and Quan J., 2003; Kevin D. Hurst, et al 1998). This problem can be overcome in IFOC drives by using reduced reactive power equation, which is independent of flux (Jevremovic V.R., et al 2010; Maiti S., et al 2008; Maiti, S. and Chakraborty, C., 2010; Peng F.Z, and Fukao T., 1994).

The adaptive algorithm conventionally uses a PI controller. The PI controller has to be tuned appropriately to obtain good stable performance. Nowadays, the use of advancements in Neural Networks (NNs) for identification and control of nonlinear dynamic systems have resulted in improved neural learning algorithms. A number of algorithms based on gradient techniques are proposed to update the parameters of the network. These techniques have been applied to MRAS based systems to update the estimated variable in the adaptive model. (Bose B.K., 2005; Chen T.C., and Sheu T.T., 2002; Karanayil B., et al 2007; Lazhar Ben-Brahim, et al 1999; Mondal S.K., et al 2002; Vas P., 1998). The powerful NN learning algorithms has been used for error adaptive mechanism in MRAS instead of PI controller and shown to perform well (Karanayil B., et al 2007; Lazhar Ben-Brahim, et al 1999; Vas P., 1998).

The RF-MRAS based on neural learning algorithm as the error adaptive mechanism for speed estimation by Vas. P., (1998) and Lazhar Ben-Brahim, et al (1999) is the

earliest work on neural learning algorithm based MRAS strategy. The neural learning algorithm based MRAS has been shown to provide stable wide range operating range and does not require tuning. However this model is flux dependent and suffers from the drawbacks of flux estimation in sensorless induction motor drives.

A novel neural learning algorithm based MRAS scheme named as "Model Reference Neural Learning Adaptive System" (MRNLAS) using reactive power as state variable is proposed in this chapter. The proposed scheme combines the advantages of reactive power technique and the capability of neural learning algorithm to form a new scheme named "Reactive Power based Model Reference Neural Learning Adaptive System" (Q-MRNLAS) for speed estimator in sensorless indirect vector controlled induction motor drives.

2.2 Proposed Speed Estimator Using Simplified Reactive Power based MRNLAS Structure (Q-MRNLAS)

The simplified equation for reactive power in IFOC is very useful and widely used in MRAS as it overcomes flux dependency and the need for stator resistance estimator. Therefore it reduces the complexity of the drive. However, it fails in certain regenerative modes of operation, and hence, wide range of speed control is not possible (Maiti S., et al 2008; Maiti S, and Chakraborty C., 2010).

The neural learning algorithm is based on powerful steepest descent method (Fredric M. Ham, and Ivica Kostanic, 2008; Martin T.Hagan, et al 2008). In this method, the weights of NN are adjusted in steps to minimize performance index. The learning rate employed in the algorithm determines the step size. Larger value of learning rate means faster learning of NN. But, this can lead to oscillations in the output and miss the global minimum point. To overcome this difficulty and reduce oscillations in the output, a momentum term is added to smoothen the oscillations and accelerate the convergence.

In literature, the RF-MRAS with neural learning algorithm is proposed by Vas P. (1998) and Lazhar Ben-Brahim et al (1999) for speed estimation. It has advantages like ease in tuning and stability over wide range of operation. However, this model is

flux dependent and suffers from the drawbacks of flux estimation in sensorless induction motor drives.

To combine the advantages of reactive power equations and neural learning algorithm a novel neural learning algorithm based MRAS scheme named as “Model Reference Neural Learning Adaptive System” (MRNLAS) is proposed.

The Q-MRNLAS proposed in this thesis uses reactive power as the state variable. The reference model and neural learning adaptive model compute reference reactive power (Q_{ref}) and estimated reactive power (Q_{est}) respectively. The reference model is independent of ω_e whereas the adjustable model depends on ω_e. The error signal ($\xi = Q_{ref} - Q_{est}$) is back propagated to adjust the weight (W_3) of the neural learning adaptive model. The rotor speed (ω_r) is then computed using the relationship $\omega_r = \omega_e - \omega_{sl}$, where, ω_e is stator frequency (W_3) and ω_{sl} is slip frequency.

The block diagram of proposed reactive power based MRNLAS (Q-MRNLAS) system for rotor/motor speed estimation is illustrated in Figure 2.2(a). The equations defining the induction motor reference model and adjustable model based on reactive power are given below (Maiti S., et al 2008).

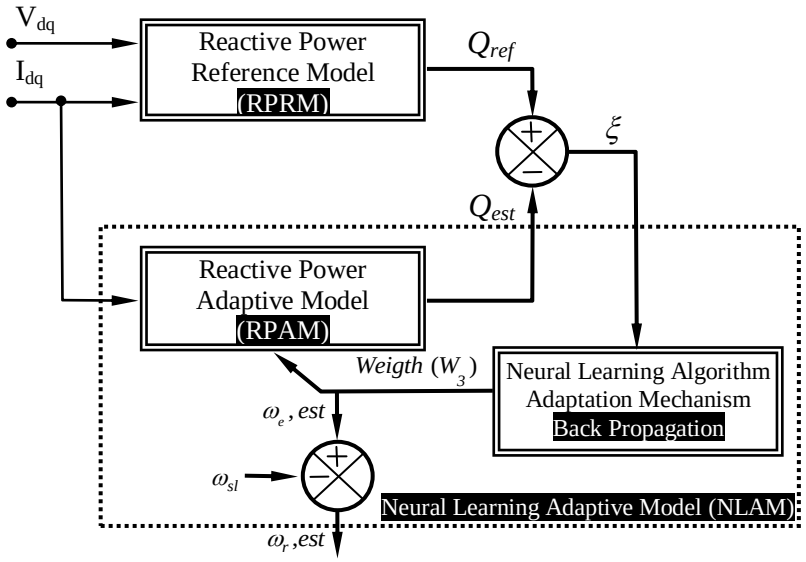


Figure 2.2(a) Speed Estimator using Reactive Power based MRNLAS (Q-MRNLAS)

The d and q axis stator voltages of an induction motor can be expressed on synchronously rotating reference frame as given in (2.1) and (2.2).

$$V_{ds} = R_s I_{ds} + \sigma L_s \frac{d}{dt} I_{ds} + \frac{L_m}{L_r} \frac{d}{dt} \lambda_{dr} - \sigma L_s \omega_e I_{qs} - \omega_e \frac{L_m}{L_r} \lambda_{qr} \quad (2.1)$$

$$V_{qs} = R_s I_{qs} + \sigma L_s \frac{d}{dt} I_{qs} + \frac{L_m}{L_r} \frac{d}{dt} \lambda_{qr} + \sigma L_s \omega_e I_{ds} + \omega_e \frac{L_m}{L_r} \lambda_{dr} \quad (2.2)$$

The actual instantaneous reactive power (Q_{ref}) absorbed by the induction motor can be expressed as in (2.3). Using the flux and parameter of the induction motor, the estimated reactive power (Q_{est}) can be expressed as (2.4) (Maiti S., et al 2008).

$$Q_{ref} = V_{qs} I_{ds} - V_{ds} I_{qs} \quad (2.3)$$

$$Q_{est} = \omega_e \sigma L_s \left(I_{ds}^2 + I_{qs}^2 \right) + \omega_e \frac{L_m}{L_r} \left(\lambda_{qr} I_{qs} + \lambda_{dr} I_{ds} \right) \quad (2.4)$$

Substituting the condition $\lambda_{dr}=L_m I_{ds}$ and $\lambda_{qr}=0$ for the indirect field oriented control (IFOC) IM drive in (2.4), the more simplified expression of Q is (2.5). The equation (2.6) is rewritten in the form of neural network and is presented in (2.7). From equation (2.7), the neural network is obtained as shown in Figure 2.2(b).

$$Q_{est} = \sigma L_s \omega_e \left(I_{ds}^2 + I_{qs}^2 \right) + \omega_e \frac{L_m^2}{L_r} I_{ds}^2 \quad (2.5)$$

$$Q_{est} = \omega_e \left(\left(\sigma L_s + \frac{L_m^2}{L_r} \right) I_{ds}^2 + \sigma L_s I_{qs}^2 \right) \quad (2.6)$$

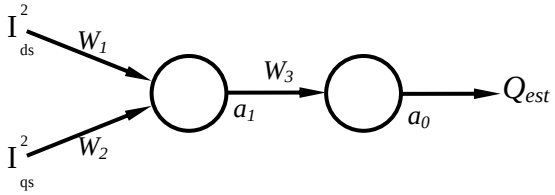


Figure 2.2(b) Neural Learning Adaptive Model (NLAM)

$$Q_{est} = W_3 \left[W_1 I_{ds}^2 + W_2 I_{qs}^2 \right] \quad (2.7)$$

All neurons have linear activation function. For the neural network shown in Figure 2.2(b) using equations (2.6) and (2.7). where, $W_1 = \sigma L_s + \left(L_m^2 / L_r \right)$; $W_2 = \sigma L_s$; $W_3 = \omega_e$; $a_1 = W_1 I_{ds}^2 + W_2 I_{qs}^2$ and $a_0 = Q_{est}$. Assume, L_m, L_r and L_s are to be constant. The weight, W_1 and W_2 are fixed and W_3 is adjusted in the neural learning adaptive model to obtain the estimated stator frequency (ω_e). The energy function (F)

minimizes the difference between actual and estimated reactive power and is given in equation (2.8). The back-propagation learning rule with momentum is used to minimize the energy function. The error gradient is backpropagted to adjusted W_3 as given equation (2.9) and (2.10). The stability of the neural learning adaptive mechanism depends on learning rate (α) and momentum (η). Appropriate choice of learning rate (α) and momentum (η) will yield the best results. The learning rate (α) and momentum (η) should lie between 0 to 1 for the system to be stable. The update equations for W_3 are given in (2.10). The updated W_3 gives the estimated stator frequency (ω_e). From ω_e the estimated rotor speed (ω_r) is calculated using the relation $\omega_r = \omega_e - \omega_{sl}$. The detailed derivation is carried out in appendix C.

$$F = \frac{1}{2}\left(Q_{ref} - Q_{est}\right)^2 \tag{2.8}$$

$$\Delta W_3(k) = \alpha\left(Q_{ref} - Q_{est}\right)a_1 \quad \text{where, } a_1 = W_1 \mathrm{I}_{ds}^2 + W_2 \mathrm{I}_{qs}^2 \tag{2.9}$$

$$W_3(k) = W_3(k-1) + \Delta W_3(k) + \eta \Delta W_3(k-1) \tag{2.10}$$

2.3 Sensorless Indirect Field Oriented Controlled IM Drives with Proposed Speed Estimator

The indirect field oriented control presented here is rotor flux oriented control without flux weakening operation. Figure 2.3 shows the complete schematic of indirect field oriented control for sensorless induction motor drives with proposed Q-MRNLAS based speed estimation. The system consists of a solid state IM drive system, rotor flux oriented control, and Q-MRNLAS based speed estimator. Q-MRNLAS based speed estimator as explained in previous section. Rotor flux oriented control consists of a PI speed controller, a current controller, and PWM generator.

The torque command is generated as a function of the speed error signal. It is generally processed through a PI controller. The torque and flux command are processed in the calculation block. The three phase reference current generated from the functional block is compared with the actual current in the hysteretic band current controller and the controller takes the necessary action to produce PWM pulses. The PWM pulses are used to trigger the voltage source inverter to drive the induction

motor. The results and discussion of sensorless indirect vector controlled induction motor drives with proposed Q-MRNLAS based speed estimation is carried out in the next section.

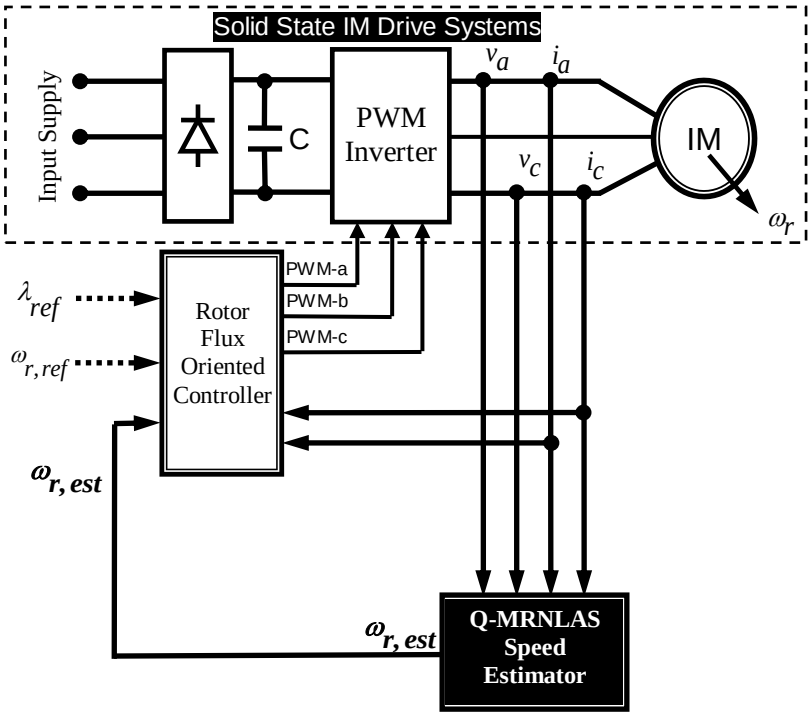


Figure 2.3 Sensorless Indirect Vector Controlled IM Drive with Q-MRNLAS based Speed Estimator

2.4 Simulation Results and Discussion of Proposed Q-MRNLAS

The performance of proposed Q-MRNLAS based rotor speed estimation utilizing the reactive power technique for sensorless indirect vector controlled induction motor drives is analyzed extensively under various operating conditions through Matlab/Simulink. The sample results of proposed Q-MRNLAS model are shown for the following operating conditions as listed below:

2.4.1 Test 1- Stair Case Speed Transients from 50 to 0 to −50rad/sec at No Load

In this test condition, the IM drive is subjected to a stair case speed commands from

50rad/sec to zero speed in a series of five steps (25rad/sec) continuing to −50rad/sec, at no load. The performance of proposed speed estimation scheme is shown in Figure 2.4(a). The mismatch error curve between actual and estimated speed is presented in Figure 2.4(b). The speed estimated from the proposed speed estimation scheme is found to closely match with the actual speed in steady state. Also the results depict stable operation for the proposed speed estimation scheme, particularly around zero speed.

2.4.2 Test 2- Load Torque Impact of 100% at 100rad/sec

The test condition 2 examines the load torque disturbance capability of the proposed speed estimation scheme. The drive is operated with reference speed of 100rad/sec. 100% step change in load torque is applied at 2.5sec and rejected at 4sec. The proposed speed estimation scheme shows better steady state and dynamic performance with negligible steady state error between the actual and estimated speed, as shown in Figure 2.5(a) and Figure 2.5(b).

2.4.3 Test 3- Low Speed Operation with Effect of Loading

The load torque disturbance capability of the proposed speed estimation scheme at very low speed of 10rad/sec with load is examined. The proposed speed estimation scheme estimates speed with good accuracy even in the case of very low speed under 50% load condition as presented in Figure 2.6(a). The speed estimation error between actual and estimated speed is observed in Figure 2.6(b). It is noticed that the estimation error is negligible at steady state.

2.4.4 Test 4- ±100rad/sec Speed at No Load

The high speed reversal capability of proposed speed estimation scheme is presented in test condition 4. Initially, the drive is operated with the speed command of 100rad/sec and the speed command is gradually reduced to -100rad/sec. The performance of proposed speed estimation scheme is shown in Figure 2.7(a). It is noticed that the sensorless drive is operated with full stability in the speed reversal

mode. The steady state error between the actual and estimated speed is very small and it is presented in Figure 2.7(b). The estimated speed follows the actual speed with good accuracy under speed reversal mode also. The proposed speed estimation scheme shows better performance and is found to estimate speed with negligible error.

2.4.5 Test 5- Very Low Speed (±1rad/sec) at No Load

This condition deals with the performance of proposed speed estimation scheme for very low speed reversal under no load. Initially, the drive is operated with the speed command of 1rad/sec up to 3sec and the speed command is gradually reduced to -1rad/sec. The performance of proposed speed estimation scheme is shown in Figure 2.8(a). The proposed speed estimation scheme performance better under steady state with negligible error and the estimated speed closely matches the actual speed. The error between actual and estimated speed is shown in Figure 2.8(b).

2.4.6 Test 6- Zero speed operation:

The performance of the proposed estimator at zero speed is tested through simulation and the results are presented in Figure 2.9(a) and error curve is shown in Figure 2.9(b). The drive is operated at zero speed from 3 to 5sec. It is observed that the estimated speed follows the actual speed with good accuracy.

2.5 Comparison of Proposed Q-MRNLAS and Existing Q-MRAS based Speed Estimators in Terms of Accuracy and Regenerating Mode of Operation

The comparisons of proposed Q-MRNLAS and commonly used Q-MRAS for speed estimation in sensorless indirect vector controlled induction motor drives under steady state are carried out at 0% and 100% loaded conditions. The results obtained are consolidated and presented in Table 2.1 and Table 2.2.

Table 2.1 Performance of the Proposed Q-MRNLAS and Q-MRAS based Speed Estimator for Various Speed Commands under No Load Condition

Reference Speed	Actual Speed (rad/sec)	Proposed Q-MRNLAS		Existing Q-MRAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	144.998	0.002	144.999	0.001
125	125.003	125.075	-0.057	125.012	-0.007
100	99.998	99.943	0.055	100.001	-0.003
75	75.001	75.025	-0.027	75.011	-0.013
50	50.002	49.984	0.022	49.991	0.022
25	24.999	24.981	0.099	25.012	-0.052
5	5.001	4.980	0.380	5.003	-0.039
1	0.999	1.004	-0.500	1.019	**-1.001**

Table 2.2 Performance of the Proposed Q-MRNLAS and Q-MRAS based Speed Estimator for Various Speed Commands under Full Load Condition

Reference Speed	Actual Speed (rad/sec)	Proposed Q-MRNLAS		Existing Q-MRAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	145.012	-0.007	145.031	0.021
125	124.998	124.984	0.011	125.001	-0.002
100	100.002	100.124	-0.124	100.102	-0.099
75	75.001	74.929	0.093	75.012	0.015
50	49.999	50.031	-0.064	50,019	-0.040
25	25.000	25.039	-0.156	25.001	-0.004
5	5.001	5.008	-0.099	4.991	0.191
1	1.001	1.009	-0.799	0.984	**1.698**

From the Tables, it is observed that both the speed estimator works very well for wide range of operating conditions from 1rad/sec to 145rad/sec. The error between the actual and estimated speed from the proposed Q-MRNLAS and existing Q-MRAS for various operating conditions is computed at 0% and 100% loaded conditions under steady state and is presented in Tables 2.1 and 2.2.

The error in the speed estimation from the existing Q-MRAS scheme under no load condition is found to be within ±0.05% for normal speed range and ±1% at low and very low speeds. Under full load condition, the error in the speed estimation from the existing Q-MRAS scheme always lies within ±0.1% for normal speed range and

±1.6% for low and very low speeds. The proposed Q-MRNLAS estimates the speed under no load condition with an accuracy of ±0.09% for normal speed range and ±0.5% at low and very low speed. The error in the speed estimated from the proposed Q-MRNLAS scheme under full load condition is found to be within ±0.1% for normal speed range and ±0.7% for low and very low speeds.

From the Tables 2.1 and 2.2, it is clear that proposed Q-MRNLAS model has same accuracy as that of existing Q-MRAS, but for implementation purpose required less complex model. The Q-MRNLAS is simpler and easy to implement in the low cost digital hardware when compared to existing Q-MRAS.

The performance comparisons of proposed Q-MRNLAS with Q-MRAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives for regenerating mode operation is also carried out in Matlab/Simulink and the results obtained are consolidated and presented.

2.5.1 Performance of Proposed Q-MRNLAS in Regenerating Mode of Operation

The performance of proposed Q-MRNLAS and Q-MRAS for on-line speed estimation is investigated for regenerating mode operation in sensorless indirect vector controlled induction motor drives. The estimators are tested for forward motoring, reverse motoring, and ramp response for very low speed under full load. Initially, the drive is operated with the speed command of 5rad/sec up to 3sec and the speed command is gradually reduced to -5rad/sec.

The performance of proposed Q-MRNLAS and Q-MRAS for on-line speed estimation for regenerating mode operation is presented in Figure 2.10(a) and Figure 2.10(b). For the comparison, both the figures are shown with same scale. From the results obtained, it is seen that the proposed Q-MRNLAS based on-line speed estimator displays stable performance and tracks the actual speed very well whereas Q-MRAS becomes unstable and fails to estimate. The proposed Q-MRNLAS based on-line speed estimator is found to be less sensitive to speed reversal. Thus, the proposed

Q-MRNLAS based on-line speed estimator exhibits stable performance whereas Q-MRAS model shows unstable performance for the regenerating mode operation.

From the investigations, the proposed Q-MRNLAS based on-line speed estimator is found to be accurate, less complex, free from integrator drift problems, robust to R_s variation and provides stable operation in regenerating mode. So, it is a promising alternative to existing Q-MRAS scheme of speed estimators for sensor-less indirect vector controlled IM drives.

2.6 Conclusion

A novel Reactive Power based MRNLAS (Q-MRNLAS) for speed estimation is proposed. The choice of reactive power as a functional candidate in Q-MRNLAS based speed estimation makes the system model equations independent of flux, simpler and easier to design. The proposed method is compared with existing Q-MRAS in terms of accuracy and regenerating mode operation. The error in the speed estimation from the existing Q-MRAS and proposed Q-MRNLAS scheme under no load and full load condition for normal operations, low and very low speeds are compared in terms of accuracy. From the results obtained, it is clear that the proposed Q-MRNLAS model has the same accuracy as that of existing Q-MRAS, but for implementation purpose required less complex model less complex model. The Q-MRNLAS is simpler and easy to implement in the low cost digital hardware when compared to the existing Q-MRAS.

The performance of proposed Q-MRNLAS and existing Q-MRAS for on-line speed estimation is also investigated for regenerating mode operation in sensorless indirect vector controlled induction motor drives. The proposed Q-MRNLAS based on-line speed estimator displays stable performance and tracks the actual speed very well whereas, popular Q-MRAS becomes unstable and fails to estimate. The proposed Q-MRNLAS based on-line speed estimator is accurate, less complex, free from integrator drift problems, robust to R_s, variation, and provides stable operation in regenerating mode. So, it is a promising alternative to popular Q-MRAS scheme of speed estimators for sensor-less indirect vector controlled IM drives.

The proposed method again suffers from the disadvantages of conventional equations based speed estimation schemes namely inaccuracies in motor parameters used and need for an online rotor resistance estimator. To further exploit the advantages of Neural Networks, it is proposed to build a data based model for on-line speed estimation. This is investigated in the next chapter.

Chapter 3

Flux Based Neural Network Model for Speed Estimation - Model I

3.1 Introduction

The equation based models for speed estimation suffer from the drawback of parameter inaccuracies. To overcome the problem, a black box approach for speed estimation using Neural Networks (NN) model is presented in this chapter. The capability of NN model trained from input/output data is well proven in the literatures (Aydogan Savran 2007; Bose B.K., 2005; Fahlman S.E., and Lebiere C., 1991; Hornik K., et al., 1989; Hornik K., et al., 1990; Lehtokangas M., 2000; Narendra K.S., and Parthasarathy, K., 1990; Shady M. Gadoue, et al 2009; Vas P., 1999).

A novel data based NN model using lux as input for on-line speed estimation is proposed. The performance of the NN model depends on the neural architecture and learning algorithm. The popular neural architectures used for function approximation are feedforward architecture and cascaded architecture (Fahlman S.E., and Lebiere C., 1991; Lehtokangas M., 2000; Narendra K.S., and Parthasarathy, K.., 1990; Shady M. Gadoue, et al 2009). The neural learning algorithms are backpropagation with momentum (BPM), variable learning rate (VLR) and Levenberg Marquardt (LM) (Martin T. Hagan, and Mohammad B. Menhaj, 1994; Martin T. Hagan, et al 2008; Narendra K.S., and Parthasarathy, K., 1991; Setiono R. and Lucas Chi Kwong Hui, 1995).

The objective of the research is to identify the most suitable NN model for on-line speed estimation in terms of accuracy, design simplicity, compactness and computational complexity.

The NN models are built from input-output data collected from the plant. Using the collected data three types of neural architectures and neural learning algorithm for on-

line speed estimation are built. The performance of the NN models is compared in terms of estimation accuracy, compactness and computational complexity. The most suitable neural architecture and learning algorithm for on-line speed estimation is identified and named as neural speed estimator model I. The performance of the proposed neural speed estimator model I is compared with RF-MRAS under various operating conditions. The results obtained through extensive simulation are presented.

3.2 Neural Architectures and Learning Algorithms

The design of data based neural network model to a large extent depends on the type of neural architectures and learning algorithms. A brief discussion about the neural architectures (single layer feed-forward (SLFF) architecture, multilayer feed-forward (MLFF) architecture and single neuron cascaded (SNC) neural architecture) considered for investigation are presented. Various learning algorithms such as backpropagation with momentum (BPM), variable learning rate (VLR) and Levenberg Marquardt (LM) are chosen for investigation.

3.3 Neural Architectures

The neural architectures considered for investigation in the thesis are the Feedforward architecture and Cascaded architecture.

3.3.1 Feedforward Architecture

Feedforward architecture consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers and output layer. Feed-forward architecture for multi-input and single-output system is shown in Figure 3.1. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The output neurons (computational nodes) constitute the output layers of the network. The remaining neurons (computational nodes) constitute hidden layers of the network. The first hidden layer is fed from the input layer made up of sensory units (source nodes). The resulting outputs of the first hidden layer are applied to the next hidden layer and so on. Thus, the neurons in each layer get input signals only

from the neurons present in the immediate previous layer. Feed-forward architecture with one hidden layer is called as single layer feed forward neural network (SLFF-NN) and when multiple layers are used it is called multilayer layer feed forward neural network (MLFF-NN).

The model of each neuron in the network includes a nonlinear activation function. The commonly used activation functions are tansigmoid, logsigmoid and purelinear functions. The choice of activation function is based on the non-linearity and output range of the system under consideration.

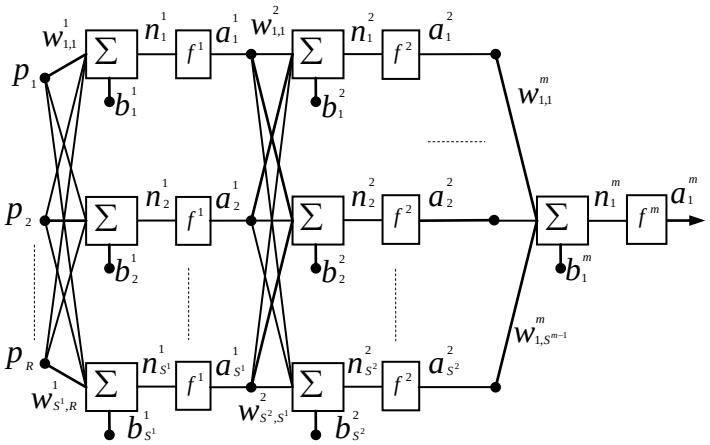


Figure 3.1 Feed-forward Network with Multiple Inputs and Single Output

The total number of parameters (P_{MLFF}) in the multilayer feed-forward architecture can be obtained from equation (3.1). As SLFF-NN is a special case of MLFF-NN with one hidden layer, the same formula suits both types of FF-NN. The first and second terms in (3.1) are deduced considering separately the weights and biases respectively.

$$P_{MLFF} = \underbrace{\sum_{m=1}^{M} S^{m-1} S^{m}}_{weights} + \underbrace{\sum_{m=1}^{M} S^{m}}_{biases} \tag{3.1}$$

where,

w_{ij}^{m} - Interconnection weight of neuron ‘i’ of layer ‘m’ for input from neuron ‘j’ of layer ‘(m-1)’.

The structure of the FF-NN architecture is denoted as S^0- S^1-…….- S^M.

3.3.2 Cascade Neural Network

The Cascade architecture consists of an input layer, hidden layers and an output layer. The first hidden layer receives only external signals as inputs. Other layers receive external inputs and outputs from all previous (m-1) 1ayers/neurons. It is called cascade because the input to a neuron consists of system inputs and outputs of all preceding layers/neurons. This is in contrast to the feed-forward architecture where inputs to a neuron are only from previous layer.

Cascade architecture results in a highly interconnected neural network, which makes learning complex if all weights are allowed to change (moving weights method). Hence most of the applications using this architecture use fixed weight method, where after training the weights are fixed for that layer (Fahlman S.E., and Lebiere C., 1991; Lehtokangas M., 2000). Fixed weight methods lead to larger networks (Nicholas K.Treadgold and Tramas D. Gedeon, 1999). In this thesis, cascade architecture with single neuron in every layer with moving weights to obtain compact network (Nicholas K.Treadgold, and Tramas D. Gedeon, 1999) is considered for investigation. Cascading a layer with single hidden neuron is resulted in "Single Neuron Cascaded (SNC) Neural Architecture". The SNC architecture is considered for investigation as it is compact, self organizing and inherits the advantages of cascaded the inputs.

The Single Neuron Cascaded (SNC) architecture with multiple inputs/single output is shown in Figure 3.2. The number of inputs to a neuron is observed to increase proportionally with the number of layers. Each neuron in the architecture includes weights, bias and a nonlinear activation function. The weights of interconnections to the previous layer are called as "input weights" and the weights of interconnections between the layers are called "link weights". The tan-sigmoid activation function is used for all hidden layers while pure-linear function is used for output layer. Initially, a hidden layer with only one neuron between the input and output is trained. To create a multilayer structure, hidden layers are added one by one and the whole network trained repeatedly using the concept of moving weights so as to obtain more compact networks. This process continues till the performance index is reached.

The weights and biases of all neurons are the parameters of the network. The generalized formula to compute the total number of parameters of a given Cascade NN is presented. The cascade NN can have any number of neurons in each layer. The total number of parameters (P_C) for a cascade neural network is presented in (3.2). The first and second terms in (3.2) are deduced considering separately the weights and biases respectively. All prefixes denote layers and suffixes denote neurons in a layer.

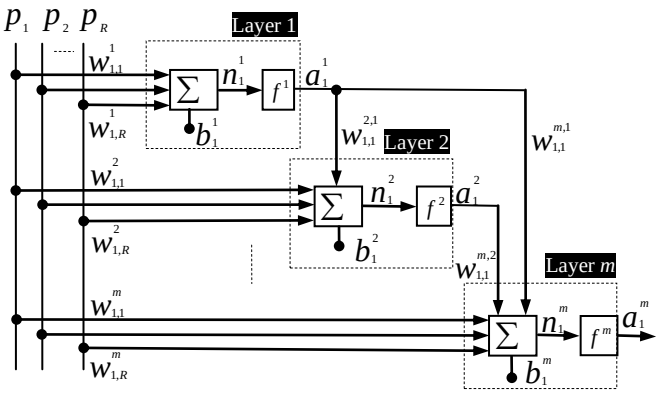


Figure 3.2 SNC-NN with Multiple Input and Single Output

$$P_C = \underbrace{\sum_{m=1}^{M}\sum_{q=0}^{m-1} S^m S^q}_{Weights} + \underbrace{\sum_{m=1}^{M} S^m}_{Biases} \qquad (3.2)$$

where,

p - Input vector, p = [1, 2,……R]

S^m - Number of neurons in the layer 'm' where m = [1,2,…M] and $S^0 = P$

$w_{i,R}^m$ - Input weight of neuron 'i' of layer 'm' for external input 'R'.

$w_{i,j}^{m,k}$ - Link weight of neuron 'i' of layer 'm' for input from neuron 'j' of layer 'k'.

b_i^m - Bias for neuron 'i' of layer 'm'.

f^m - Activation functions of all neurons in a layer 'm'.

a_i^m - Output of neuron 'i' of layer 'm'

The structure of the SNC-NN architecture is denoted as $S^0 - \sum_{i=1}^{M-1} S^i(h) - S^M$,

where, h is the number of hidden layers with single neuron in each layer.

3.4 Neural Learning Algorithms

The learning algorithms are used to obtain the optimum parameters (weights and biases) of the network by minimizing the performance index defined in terms of output mean square error function. There are different types of learning algorithms reported in the literature to train neural network (Martin T. Hagan, and Mohammad B. Menhaj, 1994; Martin T. Hagan, et al 2008; Narendra K.S., and Parthasarathy, K., 1991, Setiono R. and Lucas Chi Kwong Hui, 1995). These are mainly grouped into two types: first order approach algorithms (based on steep descent method) and second order approach algorithms. The popularly used first order approach algorithms are backpropagation algorithm with momentum (BPM) and variable learning rate backpropagation algorithm (VLR). The popularly used second order approach algorithm is Levenberg-Marquardt (LM) algorithm. These algorithms are considered for investigation and are discussed.

3.4.1 Backpropagation with Momentum (BPM)

The learning rate employed in the backpropagation (BP) algorithm determines the step size. Larger value of learning rate means faster learning of NN. This in turn would result in oscillations of the output, close to convergence leading to instability. To overcome this difficulty and reduce oscillations in the output, a momentum term is added to smoothen the oscillations. The momentum factor allows larger learning rate and accelerates convergence. The algorithm is similar to BP. The update equations are modified by adding momentum term to accelerate the convergence rate and to reduce the oscillations in error trajectory. The update equations are given in (3.3) and (3.4).

$$w^m(k+1) = w^m(k) + \eta\Delta w^m(k-1) - (1-\eta)\alpha S^m(a^{m-1})^T \quad (3.3)$$

$$b^m(k+1) = b^m(k) + \eta\Delta b^m(k-1) - (1-\eta)\alpha S^m \quad (3.4)$$

where,

η - Momentum Factor,

α - Learning rate

S^m - Sensitivity Factor of m^{th} layer

BPM is first order algorithm allowing faster convergence as compared to BP Algorithm.

3.4.2 Variable Learning Rate Backpropagation (VLR)

Variable learning rate helps to speed up the convergence. Learning rate is increased on flat surfaces and decreased when the slope increases. This algorithm accelerates the rate of convergence with adaptable learning rate. This algorithm depends on several parameters such as ξ,, ρ and η. The update equations are similar to (BPM). The learning rate is changed according to the algorithm given below.

1. If the mean square error (MSE) increases by more than some set percentage ξ (typically 1-5%) after a weight update, then the weight update is discarded, the learning rate is decreased by multiplying a factor ρ ($0<\rho<1$) and the momentum is set zero.
2. If the mean square error (MSE) decreases after a weight update, then the weight update is accepted and the learning rate is increased by multiplying the a factor $\eta>1$ and if momentum has been previously set to zero, it is reset to its original value.
3. If the mean squared error (MSE) increases by less than ξ, then weight update is accepted, but the learning rate is unchanged. If momentum has been previously set to zero, it is reset to its original value.

3.4.3 Levenberg-Marquardt algorithm (LM)

BPM and VLR algorithm are based on first order approach. The algorithms based on the second order approach are more accurate than first order approach algorithms. The higher accuracy is obtained at the cost of increased complexity of update laws. LM algorithm is a second order approach, designed especially for minimizing sums of squares functions. This is well suited to neural network training where the performance index is the mean squared error. It gives a good compromise between the faster convergence of the Gauss-Newton algorithm and the guaranteed convergence of

the steepest descent method. The update equations for LM algorithm are given in (3.5) and (3.6).

$$w(k+1) = w(k) - [J^T J+\mu I]^{-1} J^T e \quad (3.5)$$

$$b(k+1) = b(k) - [J^T J+\mu I]^{-1} J^T e \quad (3.6)$$

where,

J - Jacobian Matrix

μ - Scalar Constant

e - Mean square error

When the scalar μ is decreased to zero, the algorithm becomes Gauss Newton. When μ is increased, it approaches the steepest descent with a small learning rate. Gauss Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Gauss Newton's method as quickly as possible. Thus, μ is decreased after each successful step (if there is a reduction in MSE) and is increased only when there is an increase in MSE. In this way, MSE will be reduced at each iteration of the algorithm.

1. Initialize μ to some small value (e.g., μ=0.01) and Initialize weights and biases.
2. Present all inputs to the network and compute the corresponding network output and MSE.
3. Compute the jacobian matrix.
4. Compute the term $[J^T J+\mu I]^{-1} J^T e$ and update weights and biases of the network.
5. Recompute the MSE using updated weights and biases. If this new MSE is smaller than that computed in step 2, then divide μ by γ>1(e.g., γ=10), and go back to step 2. If the MSE is not reduced, then multiply μ by γ>1 and go back to step 4.

The algorithm is assumed to have converged when the MSE has reached the target error goal. Comparisons of three algorithms in terms of type, suitability, rate of convergence and memory requirement are presented in the Table.3.1.

Table 3.1 Comparison of NN Algorithms

Algorithm	Type	Suitability	Rate of Convergence	Memory Requirement
BPM	1st order	General error function	Faster than BP	Less
VLR	1st order	General error function	Faster than BP and BPM	Less
LM	2nd order	Sum-of –error-squares function	Faster than all the above	More

3.5 Proposed Flux based Neural Model for Speed Estimation

The chosen three architectures and learning algorithms are used to model the on-line speed estimator. The systematic design process is as follows:

- For SLFF-NN, single neuron is added in the hidden layer at a time till the target MSE is reached.
- For SNC-NN, a hidden layer with single neuron is added at a time between the inputs/outputs till the target MSE is reached.
- In MLFF-NN, the choice of number of layers and number of neurons in each layer is decided by trial and error.

The design of SLFF and SNC is systematic and the procedure can be automated. The design of MLFF-NN is more of an art than a science.

The inputs to estimator are current and flux, whose components are direct and quadrature axis stator currents measured at (k-1)th sample $\{ I_{ds}(k-1),\ I_{qs}(k-1) \}$ and rotor fluxes estimated at kth and (k-1)th sample $\{ \lambda_{dr}(k), \lambda_{dr}(k-1), \lambda_{qr}(k), \lambda_{qr}(k-1) \}$. The output is the estimated rotor speed $\{ \omega_r \}$ at kth sample. The activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively. The inputs and output of NN based rotor speed estimator is shown in Figure 3.3.

The NN architectures are trained with chosen learning algorithms. To train the NN, around 95,000 data sets were obtained for various operating conditions through simulation. Flux cannot be measured and hence, it is estimated. To obtain the training data, the voltage model is used. No parameter variation is assumed. The obtained data set is used as the training data set. For comparison, all the three NN models are trained with the same input/output data, using same learning algorithms, for the same number of epochs to reach the target MSE of 1×10^{-7}.

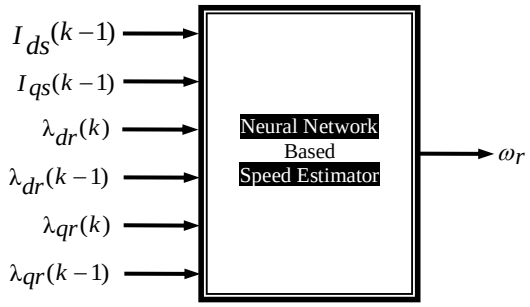


Figure 3.3 Inputs and Output of NN based Speed Estimator for Model I

The training MSE obtained for all the NN models is tabulated in Table 3.2. From the Table 3.2, it is observed that all the three NN architectures trained with LM algorithm have performed well with minimum number of neurons when compared to BPM and VLR trained NN architectures. Hence, it is concluded that LM algorithm is most suitable for offline training of speed estimation. To determine the most suitable architecture, the performance of LM-trained NN models is tested for on-line estimation of rotor speeds extensively for various operating conditions in the next section.

Table 3.2 Performance Comparison of NN Models Trained for Same Accuracy

NN Architectures	BPM	VLR	**LM**
SLFF (6-75-1)	0.082603	0.046913	**9.9909×10^{-8}**
MLFF (6-15-15-1)	0.081824	0.043246	**9.9943×10^{-8}**
SNC (6-15(h)-1)	0.067260	0.039247	**9.9992×10^{-8}**

3.6 Comparison of NN Models for Rotor Speed Estimation

3.6.1 Steady State and Dynamic Performance of LM-trained NN models for On-Line Speed Estimation

The performance of all the three LM-trained NN models is compared in terms of accuracy. The off-line trained three NN models are tested for on-line estimation of rotor speed for various operating conditions extensively. The sample results for major operating conditions are presented. The operating conditions are explained in terms of operating speed and load for convenience. The operating conditions are: (1) transient load changes, (2) transient speed changes, and (3) very low speed.

The operating condition-I examines the performance of all the three NN models for transient load disturbance. The step and ramp change load torque response results are presented at rated speed. The motor is initially operated at rated speed under 0% loaded condition and 100% step change in load torque is applied at 1sec and rejected at 2sec. The rotor speed estimated using all the three NN models for step change in load torque are shown in the Figure 3.4 respectively. The speed estimator performance for ramp change in load is presented in Figure 3.5. The motor is loaded gradually from no load (at 1sec) to full load (at 2sec). Similarly, the load is gradually decreasing from full load (3sec) to no load (4sec). The error curves between the actual and the estimated for all the models are shown. From the results obtained, it is observed that the load change capability of SNC-NN and MLFF-NN model is found to be similar and excellent whereas the load change capability of SLFF-NN model is poor as compared to SNC-NN and MLFF-NN models.

The operating condition-II test the performance of NN models for change in speed under no load condition. The performance of the NN models during the step change of rotor speed is shown in Figure 3.6. Machine operation starts at t = 0sec and a constant reference speed of 100% is considered. A step change in the reference speed occurs at sec and the new reference value is 50%. At t = 2sec, another step change in the reference speed occurs, which goes back to 100%. The error curves between the actual and the estimated for step and ramp response of all the three NN models are

shown. The tracking performance of the all three NN model is observed in Figure 3.7 for a ramp speed command. Rotor speed is gradually decreased from 100% to 50% during 1sec to 2sec. Thereafter, the speed is maintained constant at 50% up to 3sec and rotor speed is gradually increased from 50% to 100% during 3sec to 4sec. The performance of NN model for operating condition-III at very low speed is shown in Figure 3.8.

From the results obtained, it is observed that both the SNC-NN and MLFF-NN model exhibit similar and excellent dynamic performance for transient speed changes whereas SLFF-NN model shows poor dynamic performance as compared to SNC-NN and MLFF-NN models.

The test MSE for various operating conditions for all the three LM-trained NN models is evaluated and the maximum test MSE for all the three NN models are presented in the Table 3.3. From the Table 3.3, it is observed that the test MSE for SNC-NN and MLFF-NN is found to be similar and minimum as compared to SLFF-NN model. From the above analysis, it is observed that SNC-NN and MLFF-NN model have excellent mapping capability as they have multilayer structure when compared to SLFF-NN model. The SLFF-NN model has poor nonlinear mapping capability as it lacks the multilayer structure.

Table 3.3 Performance Comparison of LM-trained NN Models for Speed Estimation in Terms of Accuracy

NN Architecture	NN Model	Test MSE		
		Low Speed	Speed Change	Load Change
SLFF	6-75-1	0.1017	3.2381	0.1078
MLFF	**6-15-15-1**	**0.0042**	**0.2541**	**0.0052**
SNC	**6-15(h)-1**	**0.0023**	**0.0421**	**0.0044**

h - hidden layer with one neuron

The structural compactness and computational complexity assumes importance in real time implementation to ensure faster execution time for effective control. This motivated the comparison of LM-trained NN models in terms of structural compactness and computational complexity.

3.6.2 Structural Compactness and Computational Complexity of LM-trained NN models for On-Line Speed Estimation

The structural compactness and computational complexity assume importance in real time implementation to ensure faster execution time for effective control. This motivated the comparison of NN models in terms of structural compactness and computational complexity. The structure of neural network model depends on the number of inputs, number of outputs and the degree of nonlinearity of the system.

The number of neurons in the input/output layer is uniquely defined and is equal to that of inputs/outputs of the system to be modeled. The number of hidden layers, hidden neurons and the type of architecture are depended on the choice of the design for desired accuracy. For the desired accuracy, the number of hidden neurons is used as an index to measure the structural compactness of model. The neural network model with lesser number of hidden neurons is found to be compact and gives ease in real time implementation of the on-line flux estimator.

The number of parameters and nonlinear function extraction in the network indicates its computational complexity. Each parameter warrants some mathematical operations. The number of parameters for SNC-NN can be calculated using (3.2). As SLFF-NN is a special case of MLFF-NN with one hidden layer, the same formula (3.1) suits both type of FF-NN.

For on-line speed estimation, the complexity of the model assumes importance as the computation/estimation time has to be small enough for effective control of induction motor drives. The mathematical complexity of the model is compared by determining the number of basic operations needed by the NN model. This will depend upon the type of architecture and number of neurons. The number of hidden neurons, and parameters for the three LM-trained NN Models are tabulated in the Table 3.4.

The parameters, no. of neurons, and computations required by the SNC-NN, MLFF-NN and SLFF-NN models are tabulated in Table 3.4. From the Table 3.4, it is seen that SNC-NN model requires much lesser number of hidden neurons (15) as

compared to MLFF-NN and SLFF-NN that requires 30 and 75 hidden neurons respectively. Hence, SNC-NN model results in structurally compact model as compared to SLFF-NN and MLFF-NN model. The total number of parameters and computations required for SNC-NN is found to be lesser as compared to MLFF-NN and SLFF-NN. Hence, SNC-NN model is of lesser complexity as compared to SLFF-NN and MLFF-NN model. Hence, it can be concluded that SNC-NN architecture gives the most compact and mathematically less complex model with faster execution time for on-line speed estimation.

Table 3.4 Performance Comparison of LM-trained NN Models for Speed Estimation in Terms of Structural Compactness and Computational Complexity

NN Architectures	NN Models	No. of Hidden Neurons	No. of Parameters	Computations		
				No. of Additions	No. of Multiplications	No. of Tan-Sigmoids
SLFF	6-75-1	75	601	525	525	75
MLFF	6-15-15-1	30	361	330	330	30
SNC	**6-15(h)-1**	**15**	**232**	**216**	**216**	**15**

h - hidden layer with one neuron

Thus, it can be concluded that SNC-NN architecture trained with Levenberg Marquardt algorithm is found to be provide the required accuracy, structurally compact, computationally less complex model with faster execution time. The SNC-NN model trained with Levenberg Marquardt algorithm is identified to be the most suitable model for on-line speed estimation in sensorless controlled IM drives and is named as neural speed estimator model I. The performance of proposed neural speed estimator model I is compared with existing RF-MRAS in terms of accuracy.

3.7 Performance Comparison of Proposed NSE Model I with Existing RF-MRAS

The accuracy comparisons of proposed neural speed estimator model I and commonly used RF-MRAS for speed estimation in sensorless indirect vector controlled induction motor drives under steady state are carried out for a wide range of operations. The results obtained are consolidated and presented in Table 3.5

Table 3.5 Performance Comparison of Existing RF-MRAS and Proposed NSE Model I in Terms of Accuracy

Reference Speed (rad/sec)	Actual Speed (rad/sec)	Existing RF-MRAS		Proposed NSE Model I	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	144.997	0.00275	144.990	0.00759
125	125.003	125.002	0.00079	125.006	-0.00319
100	99.998	100.001	-0.00300	99.999	-0.00100
75	75.001	75.004	-0.00399	74.998	0.00399
50	50.002	49.996	0.01199	50.021	-0.03799
25	24.999	25.003	-0.01600	24.996	0.01200
5	5.001	4.998	0.05998	5.006	-0.09998
1	0.999	0.998	**0.10010**	1.001	**-0.20020**

From the Tables, it is observed that both the speed estimator works very well for wide range of operating conditions from 1rad/sec to 145rad/sec. The error between the actual and estimated speed from the proposed neural speed estimator model I and existing RF-MRAS for various operating conditions is computed for various operating conditions under steady state and is presented in Tables 3.5. The error in the speed estimation from the existing Q-MRAS scheme is found to be within ±0.05% for normal operating speed range and maximum value of ±0.1% at low and very low speeds. The proposed neural speed estimator model I estimates the speed with an accuracy of ±0.03% for normal operating speed range and maximum value of ±0.2% at low and very low speed. From the results, it is clear that proposed neural speed estimator model I is as good as the existing RF-MRAS.

3.8 Conclusion

A novel data based NN model using Flux as input for on-line speed estimation is proposed. To build the NN model, three types of popular neural architectures (SLFF, MLFF and SNC) and learning algorithms (BPM, VLR and LM) are considered for investigation. For comparison, all the three NN models are trained with the same input/output data, using same learning algorithms and for the same number of epochs. From the training MSE obtained, NN architectures trained with LM algorithm have performed well with minimum number of neurons when compared to BPM, VLR

trained NN architectures. Hence, it is concluded that LM algorithm is the most suitable for offline training of speed estimation. To determine the most suitable architecture, the performance of LM-trained NN models using three architectures is tested for on-line estimation of rotor speed under various operating conditions.

From the analysis, it is inferred that the steady state and dynamic performance of SNC-NN and MLFF-NN model are found to be similar and superior as compared to SLFF-NN. The SNC-NN model resulted in structurally compact, computationally less complex model as compared to MLFF-NN models. The SNC-NN can be self organized, which greatly aids design automation whereas MLFF-NN lacks the design methodology. Thus the SNC-NN model is observed to combine the advantage of multilayer mapping capability of MLFF-NN model and self-organizing feature of SLFF-NN model. Hence, it can be concluded that SNC-NN architecture trained with Levenberg Marquardt algorithm is found to provide the required accuracy, structurally compact, computationally less complex model with faster execution time.

The SNC-NN model trained with Levenberg Marquardt algorithm is identified to be most suitable model for on-line speed estimation in sensorless controlled IM drives and it is named as neural speed estimator model I. The performance of proposed neural speed estimator model I is compared with existing RF-MRAS in terms of accuracy. The proposed neural speed estimator model I based on rotor flux is as good as the existing RF–MRAS method for all operating conditions. The data based NN approach is validated using this model.

However the model is flux dependent and suffers from the drawbacks of flux estimation in sensorless induction motor drives. The reactive power model can be simplified to eliminate flux and R_s for IFOC drives. This advantage is used to build a data based model independent of rotor flux for IFOC drives and is detailed in the next chapter.

Chapter 4

Stator Frequency Based Neural Network Model for Speed Estimation – Model II

4.1 Introduction

The proposed data driven NSE model I based on rotor flux for on-line speed estimation is dependent on flux and suffers from all the drawbacks of flux estimation. To overcome the dependency on flux, simplified reactive power equations, independent of flux and R_s can be used for IFOC Drives. This approach is adopted and a novel data driven Neural Network model based on reactive power relations to compute stator frequency is proposed and named as NSE model II. The NSE model II overcomes the drawback of Model I and Q-MRNLAS.

The performance of a data based NN model to large extent depends on the choice of NN architecture and learning algorithm. Extensive study have been carried out in the pervious chapter and concluded that the single neuron cascaded neural network (SNC-NN) model trained with Levenberg Marquardt (LM) algorithm is best suited for on-line speed estimation. A novel data driven on-line speed estimator called Model II is designed using SNC-NN architecture and trained with LM algorithm for Sensor-less indirect vector controlled IM drives. The proposed SNC-NN Model II based on-line speed estimator is robust to R_r variations and stable over a wide operating range. It inherits the advantages of simplified reactive power (independent of flux and R_s) based techniques with the novelty of robustness to R_r variation and stability over a wide operating range.

The proposed SNC-NN model based on-line speed estimator is designed and tested for various operating conditions and the results are presented. Its performance is compared with Q-MRNLAS (chapter 2) scheme, which is closest to the proposed Model II as both the schemes use reduced reactive power equations.

4.2 Proposed SNC-NN Model II based On-Line Speed Estimator

In this chapter, a novel data driven neural network model based on-line speed estimator is proposed, which is independent of flux estimator, integration problems, R_s and R_r variations. Inputs to the neural network are voltage, current and stator frequency. The stator frequency can be calculated from the simplified reactive power equation (chapter 2) as given below.

$$\omega_e = Q / \left(\sigma L_s \left(I_{ds}^2 + I_{qs}^2 \right) + \frac{L_m^2}{L_r} I_{ds}^2 \right) \tag{4.1}$$

where,

$$Q = V_{qs} I_{ds} - V_{ds} I_{qs} \text{ and } \sigma = 1 - (L_m^2 / L_s L_r)$$

The ω_e obtained from (4.1) does not contain R_s, rotor flux, and derivative terms. The data driven neural network model based on-line speed estimator is designed using SNC-NN model. Around 95,000 data sets are obtained through simulation for a wide operating range. This data is used as training data set. The inputs and outputs of data driven NN Model II based rotor speed estimator is shown in Figure 4.1. The inputs to estimator are voltage, current and stator frequency, whose components are direct and quadrature axis stator voltages and currents measured at k^{th} sample $\{V_{ds}(k), V_{qs}(k), I_{ds}(k), I_{qs}(k)\}$ and stator frequency measured at k^{th} sample $\{\omega_e\}$. The output is the estimated rotor speed $\{\omega_r\}$ at k^{th} sample.

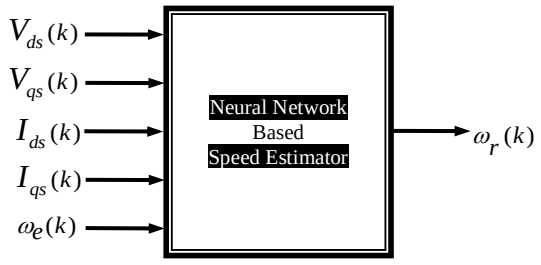


Figure 4.1 The Inputs and Outputs of Proposed NN Model II based Speed Estimator

The activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively. The proposed SNC-NN is trained with the input/output data using LM algorithm for a target mean square error (1×10^{-7}). The obtained SNC-NN model for online speed estimation has the structure 5-20(h)-1 (h-hidden layer with one neuron). Using the obtained data driven SNC-NN model based on-line speed estimator, the on-line rotor speed estimation is carried out in sensorless indirect vector controlled induction motor drive. The results and discussion are presented in next section.

4.3 Simulation Results and Discussion of Proposed SNC-NN Model II based Speed Estimation

The performance of proposed data driven SNC-NN based on-line speed estimator for sensorless indirect vector controlled induction motor drives is analyzed extensively under various operating conditions through Matlab/Simulink. The sample results for proposed NN based on-line speed estimator are shown for the following operating conditions as listed below:

4.3.1 Test 1- Stair Case Speed Transients from 50 to 0 to −50rad/sec at No Load

The sensorless indirect vector controlled IM drive is subjected to a stair case speed commands from 50rad/sec to zero speed in a series of five 25rad/sec steps continuing to −50rad/sec, at no load. The performance of proposed speed estimation scheme is shown in Figure 4.2(a). The mismatch error curve between actual and estimated speed is presented in Figure 4.2(b). The speed estimated from the proposed scheme is found to closely match with the actual speed in steady state with maximum error ±0.2rad/sec and in transient state with maximum error of ±4rad/sec. Also the results depict stable operation for the proposed data driven SNC-NN based speed estimation scheme, particularly around zero speed.

4.3.2 Test 2- Load Torque Impact of 100% at 100rad/sec

The test condition 2 examines the load torque disturbance capability of the proposed data driven speed estimation scheme. The drive is operated with reference speed of 100rad/sec. 100% step change in load torque is applied at 2.5sec and rejected at 4sec. The proposed speed estimation scheme shows better steady state and dynamic performance with negligible steady state error between the actual and estimated speed, as shown in Figure 4.3(a) and Figure 4.3(b). The error between actual and estimated speed during no load condition is ±0.2rad/sec and ±1.5rad/sec of error under dynamic state.

4.3.3 Test 3- Low Speed Operation with Effect of Loading

The load torque disturbance capability of the proposed speed estimation scheme at very low speed of 15rad/sec with load is examined. The proposed data driven speed estimation scheme estimates speed with good accuracy even in the case of very low speed under 100% load condition as presented in Figure 4.4(a). The speed estimation error between actual and estimated speed is shown in Figure 4.4(b). From the Figure 4.4(b), it is observed that the error between actual and estimated speed under no load and full load condition is with in ±0.1rad/sec. It is noticed that the estimation error is negligible under steady and dynamic state.

4.3.4 Test 4- ±100rad/sec Speed with Full Load

The high speed reversal capability of proposed speed estimation scheme is presented in test condition 4. Initially, the drive is operated with the speed command of 100rad/sec and the speed command is gradually reduced to -100rad/sec with 100% load. The performance of proposed speed estimation scheme is shown in Figure 4.5(a). It is noticed that the sensorless drive is operated with full stability in the speed reversal mode even though the peak overshoot goes 5% greater than actual speed. The steady state error between the actual and estimated speed is very small and it is presented in Figure 4.5(b). The estimated speed follows the actual speed with good

accuracy under speed reversal mode also. The proposed speed estimation scheme shows better performance and is found to estimate speed with negligible error.

4.3.5 Test 5- Very Low Speed (±1rad/sec) with Full Load

In Figure 4.6, the proposed speed estimation scheme is tested for forward motoring, reverse motoring, and ramp response for very low speed under full load. Initially, the drive is operated with the speed command of 1rad/sec up to 3sec and the speed command is gradually reduced to -1rad/sec. The performance of proposed speed estimation scheme is shown in Figure 4.6(a). In this case also, the performance is good with negligible steady state error and the estimated speed is found to closely match with the actual. The error between actual and estimated speed is shown in Figure 4.6(b). From the Figure 4.6(b), it is observed that the error range between actual and estimated speed is ±0.03rad/sec. Hence, the proposed estimation scheme is very much suitable for very low speed operations also.

4.3.6 Test 6- Zero Speed Operation

The performance of the proposed estimator at zero speed is tested through simulation and the results are presented in Figure 4.7(a) and error curve is shown in Figure 4.7(b). The drive is operated at zero speed from 3 to 5sec. It is observed that the estimated speed follows the actual speed with good accuracy. Hence, the proposed estimation scheme is very much suitable for zero speed operations also.

4.4 Performance Comparison of Proposed Data Driven NSE Model II with Q-MRNLAS

The comparisons of proposed NSE Model II and Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives under steady state are carried out at 0% and 100% loaded conditions. The results obtained are presented in Table 4.1 and Table 4.2.

From the Tables, it is observed that data driven neural model based speed estimator works very well for a wide range of operating conditions from 1rad/sec to 145rad/sec. The error between the actual and estimated speed under steady state for the proposed data driven SNC-NN model II and Q-MRNLAS for various operating conditions is presented in Tables 4.1 and 4.2 for no load and full load.

The error in the speed estimation from the Q-MRNLAS under no load condition is found to be within ±0.3% for normal operating speed range and ±0.5% at low and very low speeds. The proposed neural speed estimator model II estimates the speed under no load condition with an accuracy of ±0.7% for normal operating speed range and maximum value of ±0.2% at low and very low speed.

Under full load condition, the error Q-MRNLAS lies within ±0.1% for normal operating speed range and ±0.7% for low and very low speeds. For the Model II the error in the speed estimated under full load condition is found to be within ±0.2% for normal operating speed range and ±0.8% for low and very low speeds. From the Tables 4.1 and 4.2, it is clear that proposed data driven SNC-NN Model II is nearly as good as Q-MRNLAS scheme for normal, low and very low speeds.

The performance comparison of proposed data driven SNC-NN Model II with Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives for rotor resistance (R_r) variation under steady state is carried out and the results obtained are presented in the next section.

4.5 Performance of NSE Model II for Rotor Resistance Variation

In the literature, it is reported that the change in R_r may go upto 50% and 100% (B.Karanayil et al 2007; S.Maiti et al., 2008) during motor operation. The performance of proposed data driven SNC-NN model and Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives under steady state is tested for step change in rotor resistance variation. Of course, in a real drive, the rotor resistance never undergoes abrupt variations in response to

temperature change due to the large thermal time constant. The step variation represents an extreme case and is used to show the robustness of the proposed SNC-NN based on-line speed estimation scheme. The effect of R_r variation is investigated at very low speed of 1rad/sec with full load condition. Two different cases for rotor resistance detuning are considered.

4.5.1 Slight R_r detuning

The actual R_r of the induction motor are slightly detuned with respect to the nominal ones, as follows:

$$\frac{\Delta R_r}{R_r} = -10\% \tag{4.2}$$

In this case, the performance of proposed speed estimation scheme and Q-MRNLAS for speed estimation scheme is tested for 10% increase in rotor resistance effected at 2sec. The speed estimated from the proposed data driven SNC-NN model and Q-MRNLAS is presented in Figure 4.8(a) and (b) respectively. For the comparison, both the figures are shown with same scale. From the results obtained, it is obvious that the speed estimated from the data driven SNC-NN model closely tracks the actual speed even when there is a change in the parameter and the error in the speed estimation is almost negligible whereas the Q-MRNLAS based speed estimation fails. Thus, data driven SNC-NN model based on-line speed estimator is shown to be robust to slight R_r detuning.

4.5.2 Large R_r detuning

In many real applications, the R_r may vary on ranges which are larger than those considered in the previous section. In order to check the robustness of the proposed speed estimator in the presence of larger detuning, the actual R_r of the induction motor is largely detuned with respect to the nominal ones as follows:

$$\frac{\Delta R_r}{R_r} = -50\% \tag{4.3}$$

A step change of 50% in R_r is effected at 2sec. The robust speed estimation is observed from the proposed data driven SNC-NN based on-line speed estimator even in the case of large parameter tuning which is presented in Figure 4.9(a). Whereas the speed estimated from Q-MRNLAS is observed to 4.971rad/sec as shown in Figure 4.9(b). For the purpose of comparison, both the figures are shown with the same scale. The error between actual and estimated speed from Q-MRNLAS is 3.97rad/sec. The speed keeps on increasing with the increase in R_r. Thus the proposed data driven SNC-NN based on-line speed estimator exhibits robust speed estimation even in the presence of slight and large parameter variation of R_r.

The performance of proposed data driven SNC-NN model II and Q-MRNLAS are studied for 0-100% changes in R_r at a critical operating condition namely very low speed (1rad/sec) with full load. The results shown in Table 4.3 indicate that the proposed estimator is able to track the speed closely whereas the Q-MRNLAS fails. The performance of the proposed estimator at full load is studied for a large change in R_r (50%) at various speeds and compared with Q-MRNLAS. From the results shown in Table 4.4, it can be seen that the proposed estimator outperforms the Q-MRNLAS based estimator for R_r variations.

Table 4.3 Different % Change in R_r at 1rad/sec with Full Load

Actual Speed = 1.001rad/sec

% Change in R_r	NSE Model II based Estimated Speed (rad/sec)	Q-MRNLAS based Estimated Speed (rad/sec)
100	1.045	8.901
80	1.011	7.128
60	1.090	6.354
40	1.031	3.939
20	1.025	2.897

Table 4.4 50% Change in R_r at Different Speed with Full Load

Reference Speed	Actual Speed (rad/sec)	NSE Model II based Estimated Speed (rad/sec)	Q-MRNLAS based Estimated Speed (rad/sec)
145	145.011	145.421	154.135
100	100.002	100.055	107.185
75	75.051	74.990	81.284
50	49.979	50.009	56,345
25	25.030	24.988	29.629
1	1.001	1.011	4.969

4.6 Conclusion

This chapter proposes a novel robust data driven SNC-NN model II based on-line speed estimator for sensor-less indirect vector controlled IM drives. The proposed estimator is designed using single neuron cascading architecture and LM learning algorithm using 95000 data sets for a target MSE of 9.99458×10^{-8}. The obtained SNC-NN model for online speed estimation has the structure 5-20(h)-1 (h-hidden layer with one neuron). The designed speed estimator is extensively tested for a number of operating conditions and is shown to perform well.

A performance comparison is carried out in terms of steady state accuracy with Q-MRNLAS method under no load and full load. Both the estimators are operated at various speeds. It is demonstrated that the proposed neural speed estimator model II performs as well as the Q-MRNLAS, for all operating conditions. The robustness of the proposed neural speed estimator model II is illustrated for R_r variation. This is done for slight and large detuning. The Q-MRNLAS fails to track the speed, whereas the proposed estimator tracks the speed accurately with variations in R_r. The superiority of the proposed scheme over Q-MRNLAS is well demonstrated.

The proposed on-line speed estimator combines the advantages of neural network and the simplified reactive power based techniques. It exhibits good performance over a wide operating range with good accuracy. It is less complex, robust to integrator drift

problems, parameter variations and provides stable operation in regenerating mode. It is therefore concluded that the proposed estimator is a more promising alternative to the Q-MRNLAS model based on-line speed estimators for sensor-less indirect vector controlled IM drives.

However, the proposed neural speed estimator model II has a few disadvantages. Neural speed estimator model II can be used only for sensor-less indirect vector controlled induction motor drives as the reduced reactive power equations are applicable only for IFOC. Also the proposed estimator uses mathematical equation for computation of ω_e. Hence, an accurate knowledge of machine parameters is mandatory.

A generalized data based model which is suitable for all types of vector control using a black box approach without the need for motor parameters would provide a novel solution by addressing all the above issues. This is envisaged and investigated in the next chapter.

Chapter 5

Reactive Power Based Neural Network Model for Speed Estimation – Model III

5.1 Introduction

The proposed NSE model II is suitable only for IFOC drives. Also it uses the machine model equation for computation of ω_e and therefore requires accurate knowledge of machine parameters. To make the speed estimation a generalized data based model suitable for all types of sensorless control, a black box approach without the need for motor parameters would provide a novel solution. This is envisaged and investigated in this chapter. A novel data driven neural network model based on reactive power is proposed for on-line speed estimator and is named as NSE model III.

The proposed speed estimator is independent of machine parameters, flux and is suitable for all applications of sensorless controlled induction motor drives. The proposed NSE model III is based on the chosen SNC-NN architecture trained off-line using LM algorithm. It combines the advantages of reactive power based techniques with the novelty of being independent of machine parameters and stable over a wide operating range. The proposed NSE model III based on-line speed estimator is compared in terms of accuracy and rotor resistance variations with proposed Q-MRNLAS scheme (chapter 2) and validated through Matlab/Simulink.

5.2 Proposed Model III based On-Line Speed Estimator

In this chapter, a novel data driven neural network based on-line speed estimator is proposed which is independent of flux estimator, integration problems, R_s and R_r variations. The inputs to network are voltage, current and reactive power. The reactive power can be expressed (chapter 2) as given below.

$$Q = \mathrm{V_{qs}I_{ds}} - \mathrm{V_{ds}I_{qs}} \tag{5.1}$$

The Q obtained from (5.1) is independent of motor parameters and derivative terms. Around 95,000 data sets are obtained through simulation for a wide operating range. These data sets are used as training data set. The inputs and outputs of data driven NN based rotor speed estimator is shown in Figure 5.1. The inputs to estimator are voltage, current and stator frequency, whose components are direct and quadrature axis stator voltages and currents measured at k^{th} sample $\{V_{ds}(k), V_{qs}(k), I_{ds}(k), I_{qs}(k)\}$ and reactive power frequency measured at k^{th} sample $\{Q_{est}\}$**.** The output is the estimated rotor speed $\{\omega_r\}$ at k^{th} sample.

The activation function for hidden and output layers is chosen as tan-sigmoid and pure linear function respectively. The proposed SNC-NN is trained with the input/output data using LM algorithm of the target mean square error (1×10^{-7}). The obtained SNC-NN model for online speed estimation has the structure 5-25(h)-1 (h-hidden layer with one neuron). Using the obtained data driven SNC-NN Model III based on-line speed estimator, the on-line rotor speed estimation is carried out in sensorless indirect vector controlled induction motor drive. The results and discussion are presented in next section.

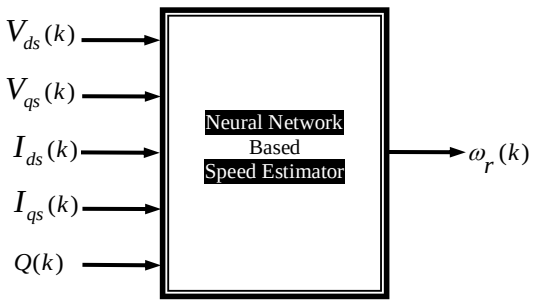


Figure 5.1 The Inputs and Outputs of Proposed NN Model III based Speed Estimator

5.3 Simulation Results and Discussion of Proposed Data Driven SNC-NN Model III based Speed Estimation

The performance of proposed data driven SNC-NN based on-line speed estimator for sensorless indirect vector controlled induction motor drives is analyzed extensively

under various operating conditions through Matlab/Simulink. The sample results for proposed NN based on-line speed estimator are shown for the following operating conditions as listed below:

5.3.1 Test 1- Stair Case Speed Transients from 50 to 0 to −50rad/sec at No Load

The sensorless indirect vector controlled IM drive is subjected to a stair case speed command from 50rad/sec to zero speed in a series of five 25rad/sec steps continuing to −50rad/sec, at no load. The performance of proposed speed estimation scheme is shown in Figure 5.2(a). The mismatch error curve between actual and estimated speed is presented in Figure 5.2(b). The speed estimated from the proposed scheme is found to closely match with the actual speed in steady state with maximum error ±0.2rad/sec and in transient state with maximum error of ±8rad/sec. Also the results depict a stable operation for the proposed neural speed estimator model III based speed estimation scheme, particularly around zero speed.

5.3.2 Test 2- Load Torque Impact of 100% at 100rad/sec

The test condition 2 examines the load torque disturbance capability of the proposed data driven speed estimation scheme. The drive is operated with reference speed of 100rad/sec. 100% step change in load torque is applied at 2.5sec and rejected at 4sec. The proposed speed estimation scheme shows better steady state and dynamic performance with negligible steady state error between the actual and estimated speed, as shown in Figure 5.3(a) and Figure 5.3(b). The error between actual and estimated speed during no load condition is ±0.1rad/sec and ±1.8rad/sec of error under dynamic state.

5.3.3 Test 3- Low Speed Operation with Effect of Loading

The load torque disturbance capability of the proposed speed estimation scheme at very low speed of 15rad/sec with load is examined. The proposed data driven speed estimation scheme estimates speed with good accuracy even in the case of very low

speed under 100% load condition as presented in Figure 5.4(a). The speed estimation error between actual and estimated speed is shown in Figure. 5.4(b). From the Figure 5.4(b), it is observed that the error between actual and estimated speed under no load and full load condition is with in ±0.08rad/sec. It is noticed that the estimation error is negligible under steady and dynamic state.

5.3.4 Test 4- ±100rad/sec Speed with Full Load

The high speed reversal capability of proposed speed estimation scheme is presented in test condition 4. Initially, the drive is operated with the speed command of 100rad/sec and the speed command is gradually reduced to -100rad/sec with 100% load. The performance of proposed speed estimation scheme is shown in Figure 5.5(a). It is noticed that the sensorless drive is operated with full stability in the speed reversal mode even though the peak overshoot goes 5% greater than actual speed. The steady state error between the actual and estimated speed is very small and is presented in Figure 5.5(b). The estimated speed follows the actual speed with good accuracy under speed reversal mode also. The proposed speed estimation scheme shows better performance and found to estimate speed with negligible error.

5.3.5 Test 5- Very Low Speed (±5rad/sec) with Full Load

In Figure.5.6, the proposed speed estimation scheme is tested for forward motoring, reverse motoring, and ramp response for very low speed under full load. Initially, the drive is operated with the speed command of 5rad/sec up to 3sec and the speed command is gradually reduced to -5rad/sec. The performance of proposed speed estimation scheme is shown in Figure 5.6(a). In this case also, better performance with negligible steady state error is obtained from proposed speed estimation scheme and found the speed estimated is found to closely match with the actual. The error between actual and estimated speed is shown in Figure 5.6(b). From the Figure 5.6(b) presented, it is observed that the error between actual and estimated speed is with in ±0.03rad/sec under steady state and transient state. Hence, the proposed estimation scheme is very much suitable for very low speed operations also.

5.3.6 Test 6- Zero Speed Operation

The performance of the proposed estimator at zero speed is tested through simulation and the results are presented in Figure 5.7(a) and error curve is shown in Figure 5.7(b). The drive is operated at zero speed from 3 to 5sec. It is observed that the estimated speed follows the actual speed with good accuracy. Hence, the proposed estimation scheme is very much suitable for zero speed operations also.

5.4 Performance Comparison of Proposed Data Driven SNC-NN Model III with Q-MRNLAS

The comparisons of proposed data driven SNC-NN model III and Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives under steady state are carried out at 0% and 100% loaded conditions. The results obtained are consolidated and presented in Table 1 and Table 2.

From the Tables, it is observed that data driven neural model based speed estimator works very well for a wide range of operating conditions from 1rad/sec to 145rad/sec. The error between the actual and estimated speed from the proposed data driven SNC-NN model III and Q-MRNLAS for various operating conditions is computed at 0% and 100% loaded conditions under steady state and is presented in Tables 5.1 and 5.2.

The error in the speed estimation from the Q-MRNLAS under no load condition is found to be within ±0.3% for normal operating speed range and ±0.5% at low and very low speeds. The proposed data driven SNC-NN model III estimates the speed under no load condition with an accuracy of ±0.7% for normal operating speed range and maximum value of ±0.8% at low and very low speed.

Under full load condition, the error lies within ±0.1% for normal operating speed range and ±0.7% for low and very low speeds. The error in the speed estimated under full load condition is found to be within ±0.6% for normal operating speed range and ±1% for low and very low speeds. From the Tables 5.1 and 5.2, it is clear that proposed data driven SNC-NN model III has nearly the same accuracy as

Q-MRNLAS scheme for normal operating speed range but distinctly better under low and very low speeds.

The performance comparisons of proposed data driven SNC-NN model III with Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives for rotor resistance (R_r) variation under steady state is carried out in Matlab/Simulink and the results obtained are consolidated and presented in the next section.

5.5 Performance of NSE Model III for Rotor Resistance Variation

The performance of proposed data driven SNC-NN model III and Q-MRNLAS for on-line speed estimation in sensorless indirect vector controlled induction motor drives under steady state is tested for step change in rotor resistance variation. In real drive, the rotor resistance never undergoes abrupt variations in response to temperature change due to the large thermal time constant. The step variation represents an extreme case and is used to show the robustness of the proposed SNC-NN based on-line speed estimation scheme. The effect of R_r variation is investigated at very low speed of 1rad/sec with full load condition. The actual R_r of the induction motor are detuned with respect to the nominal ones, as follows:

$$\frac{\Delta R_r}{R_r} = -100\% \tag{5.2}$$

In this case, the performance of proposed speed estimation scheme and popular Q-MRNLAS for speed estimation scheme is tested for 100% increase in rotor resistance. 100% step change in R_r is effected at 2sec. The speed estimated from the proposed robust data driven SNC-NN model III and existing Q-MRNLAS is presented in Figure 5.8(a) and 5.8(b) respectively. From the results obtained, it is obvious that the speed estimated from the robust data driven SNC-NN model III tracks closely the actual speed even when there is a change in the parameter and the error in the speed estimation is almost negligible whereas the speed estimated from

existing Q-MRNLAS is 8.94rad/sec. The error between actual and estimated speed from Q-MRNLAS is very large and fail to estimate the speed. The data driven SNC-NN model III based on-line speed estimator is shown to be robust to R_r variation.

Thus the proposed data driven SNC-NN model III based on-line speed estimator is accurate, less complex, free from integrator drift problems, robust to R_s, R_r variation, and provides stable operation in regenerating mode. The proposed speed estimator is independent of machine parameters and it can applicable to all types of sensorless controlled induction motor drives.

The performance of proposed data driven SNC-NN based model III and Q-MRNLAS are studied for 0-100% changes in R_r at a critical operating condition namely very low speed (1rad/sec) with full load in Table 5.3. The results shown in Table 5.3 indicate that the proposed estimator is able to track the speed closely whereas the Q-MRNLAS fails. The performance of the proposed estimator at full load is studied for a 50% change in R_r at various speeds and compared with Q-MRNLAS. The results obtained are consolidated and presented in Table 5.4. From the Table 5.4, it is very clear that the proposed speed estimator tracks the actual speed very well with minimum error. From the results shown in Table 5.3 and 5.4 it is concluded that the proposed estimator outperforms the Q-MRNLAS based estimator for R_r variations.

5.6 Comparison of Structural Complexity for the Proposed Three Data Based NN Models

In this thesis, three different types of data based NN models are proposed. The proposed data based NN models are:

1. Model I (rotor flux based NN model),
2. Model II (stator frequency based NN model) and
3. Model III (reactive power based NN model)

The proposed data based NN models are compared for the structural compactness and computational complexity in terms of hidden neurons, total number of parameters (Weight and Bias) and computations involved and shown in Table 5.5.

The proposed NN models are trained for same numbers of data and same accuracy (MSE). Model I is most compact but flux dependent and suffers from all the drawbacks of flux estimation in sensorless induction motor drives. Model II is more complex than Model I but overcomes the drawbacks of Model I. However the proposed Model II can be used only for sensor-less indirect vector controlled induction motor drives as the reduced reactive power equations are applicable only for IFOC. Also the proposed estimator uses mathematical equation for computation of ω_e. Hence, an accurate knowledge of machine parameters is mandatory. Model III has nearly 2 times the complexity of Model I and 1.4 times the complexity of Model II. However its advantages overweigh the disadvantages.

Model III is a generalized data based model which is suitable for all types of sensorless controlled IM drives using a black box approach without the need for motor parameters. The increase in compactness and computational complexity can be overcome using high speed digital hardware as the time constraints are not high for speed estimation. Hence Model III would provide a novel solution by addressing all the above issues.

5.7 Conclusion

This chapter proposes a novel robust data driven SNC-NN based on-line speed estimator (Model III) for sensor-less controlled IM drives. The proposed estimator is designed using SNC-NN and LM learning algorithm using 95000 data sets for a target MSE of 9.99×10^{-8}. The obtained SNC-NN model for online speed estimation has the structure 5-25(h)-1 (h-hidden layer with one neuron). The designed speed estimator was extensively tested for a number of operating conditions and shown to perform well.

A performance comparison is carried out in terms of steady state accuracy with Q-MRNLAS method under no load and full load. Both the estimators were operated at various speeds. It is demonstrated that the proposed neural speed estimator model III performs as well as the Q-MRNLAS, for all operating conditions. The robustness of the proposed neural speed estimator model III is illustrated for R_r variation. The Q-

MRNLAS fails to track the speed, whereas the proposed estimator tracks the speed accurately with variations in R_r. The superiority of the proposed scheme over Q-MRNLAS is well demonstrated.

The SNC-NN based neural speed estimator model III has several advantages

- It is independent of flux and hence, overcomes all flux based estimation problems.
- It is independent of machine parameters.
- Uses a Black box approach.
- It is a generalized model and hence, suitable for all type of sensorless control of IM drives.
- Does not require additional estimator and hence, reduces the complexity of drive system.
- The SNC architecture trained using LM algorithm is found to be compact, less complex, easy in design, accurate and most suitable for speed estimation.

The proposed neural speed estimator model III is little more complex than neural speed estimators model I and II. However this drawback can be overcome with high speed digital implementation for speed estimation. From the detailed investigations it is concluded that neural speed estimator model III is the most suitable model for on-line speed estimation as for all types of sensor less controlled IM drives. The proposed novel method is implemented in digital hardware and validated using practical motor setup in the next chapter.

Chapter 6

Experimental Validation and Digital Implementation of Proposed Neural Speed Estimator

6.1 Introduction

The proposed NSE model III is generalized data based model which is suitable for all types of sensorless controlled IM drives as it uses a black box approach, does not require knowledge of motor parameters and provides a novel solution by addressing all the issues of conventional methods. Hence, the proposed NSE model III, identified most suitable for on-line speed estimation in the pervious chapter is validated using real time data. Also the digital implementation of the proposed model is carried out using Field Programmable Gate Arrays (FPGA).

The laboratory motor setup is constructed to obtain the practical data. The proposed NSE model III based on-line speed estimator is designed, modelled and validated for real time data. The result proves that the proposed generalized neural model based speed estimator performs very well for practical data also. The proposed speed estimator should be implementable in digital hardware for real time applications. FPGA is the most suitable digital hardware for neural network implementation. Hence, the designed speed estimator is implemented using FPGA.

The major issues in FPGA implementation is the tradeoff between computation time and cost (resource). More complex computation will require larger resource which in turn would increase the cost. In this chapter, the most complex block of the NN based speed estimator is identified as computation of nonlinear activation function. To overcome this problem, the tan-sigmoid activation function is replaced by a much simpler Elliott function. To reduce the resource, a layer multiplexing technique is adopted. The lowest bit precision needed for good performance of the speed estimator is also identified. The proposed speed estimator is implemented with new activation function and identified bit precision using layer multiplexing technique and tested on

Xilinx Spartan FPGA kit (3sd1800afg676-4). The results obtained are reported. These investigations would lead to the development of a cheaper, efficient and intelligent speed estimator for sensorless controlled IM drives.

6.2 Experimental Setup

A 3-hp three phase squirrel cage induction motor is considered for practical study. The laboratory motor setup constructed to obtain the practical data is shown in Figure 6.1. The induction motor is driven with intelligent power module. The pulses are given to the IPM module through DSPIC controller. The induction motor is operated with the switching frequency of 10 kHz. The three phase Hall effect voltage and current sensors with optical isolator are used to measure the motor phase voltages and currents. The measured motor speed is measured by a speed sensor (encoder) mounted in induction motor shaft. The LPF (low pass filter) is used to filter out the noise in the voltage, current and speed signals. The real time data is acquired and stored in PC using 12-bit USB based data acquisition system with Agilent software.

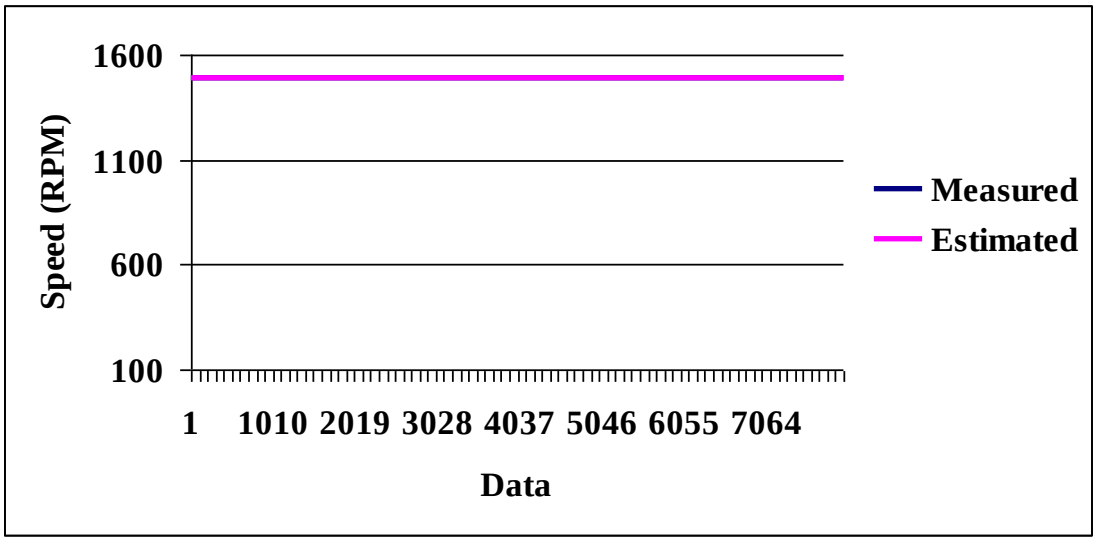
6.3 Design and Validation of Proposed Speed Estimator for Real Time Data

Around 3,300 data sets were collected from experiential setup as discussed in the pervious section. The proposed NSE model III based on-line speed estimator is designed and modelled for real time data obtained from experiential setup. Extensive study have been carried out in the previous chapter and concluded that the NSE model III is best suitable for on-line speed estimation. Hence, the same neural architecture is trained with real time data. The performance of proposed NSE model III based on-line speed estimator for real time data is analyzed extensively under various operating conditions. The sample results for proposed NSE model III based on-line speed estimator are shown below:

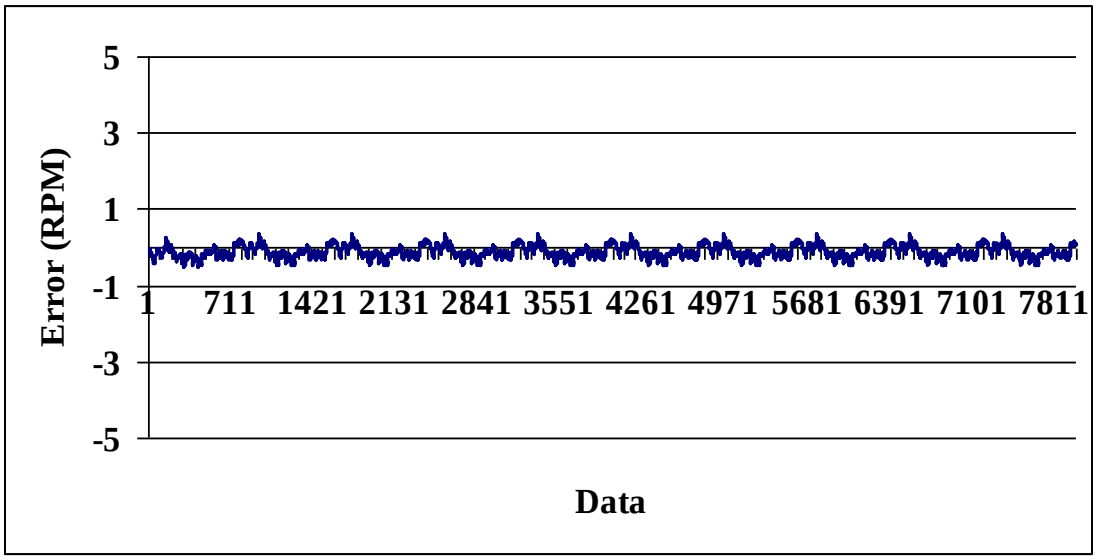
1. Operating Condition 1: The motor is operated at rated speed
2. Operating Condition 2: The motor is operated at 50% rated speed

For operating condition 1, the measured and estimated rotor speed is presented in Figure 6.2 (a) and its error is shown in Figure 6.2 (b). The speed estimated from the

proposed scheme is found to closely match with the measured speed. For operating condition 2, the measured and estimated rotor speed is presented in Figure 6.3 (a) and its error is shown in Figure 6.3 (b). From the result obtained, it is concluded that the proposed NSE model III based speed estimator performs very well for practical data. Hence, the NSE model III is implemented using FPGA.

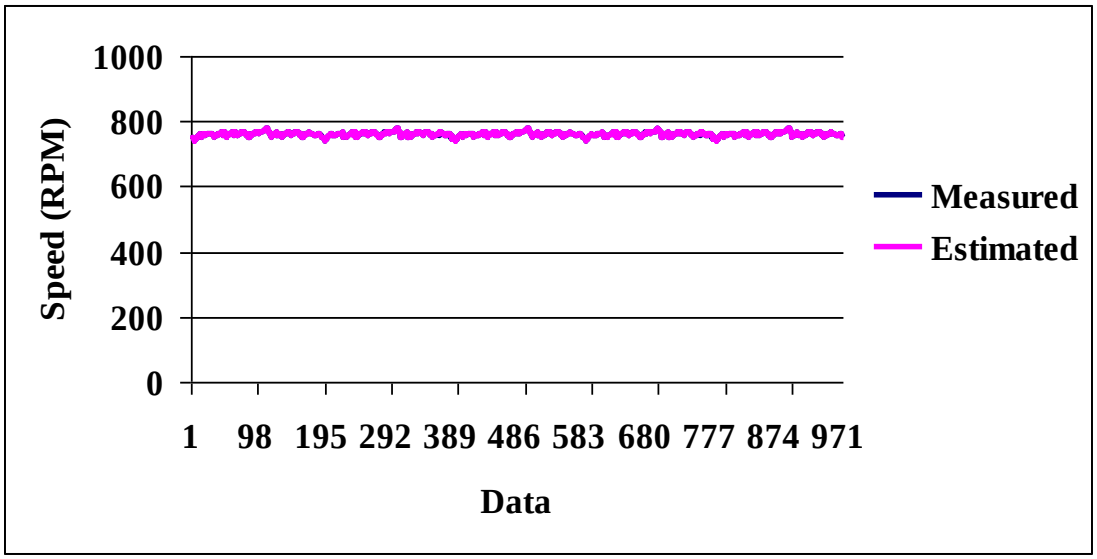


(a)

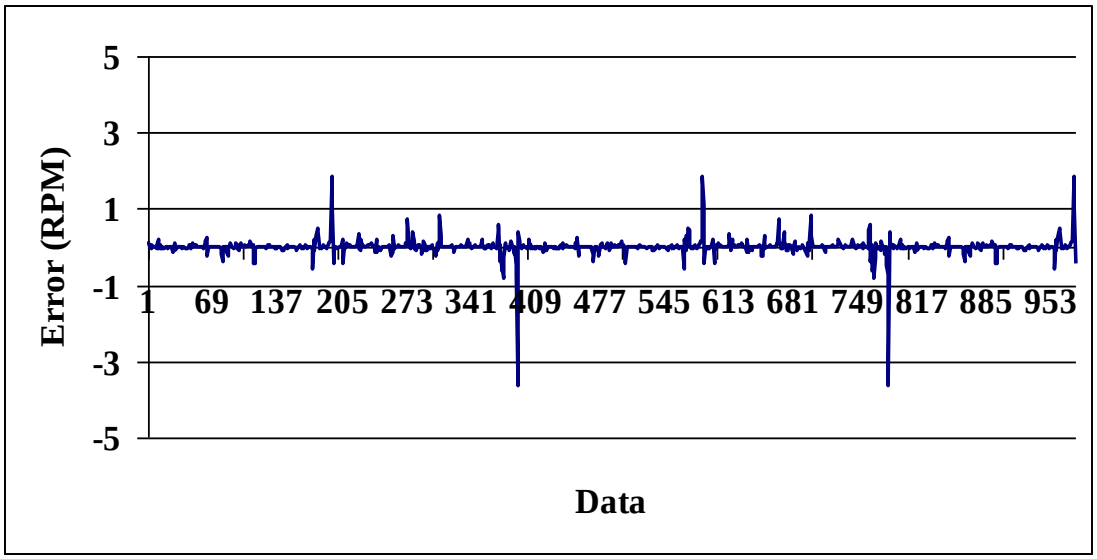


(b)

Figure 6.2 Rated Speed: (a) Measured and Estimated Speed, (b) Error between Measured and Estimated Speed



(a)



(b)

Figure 6.3 50% of Rated Speed: (a) Measured and Estimated Speed, (b) Error between Measured and Estimated Speed

6.4 Issues in FPGA Implementation of Proposed NN Based Speed Estimator

The neural network hardware implementation is broadly classified into two group namely (i) digital neural network hardware and (ii) analog neural network hardware. The digital neural network hardware implementation is preferred because of higher

accuracy, high repeatability, low noise sensitivity, better testability, higher flexibility, and compatibility with other types of preprocessors. On the other hand, analog systems are more difficult to be designed and can only be feasible for large-scale productions, or for very specific applications.

The digital neural network hardware implementations are further classified as (i) FPGA-based implementations (ii) DSP-based implementations (iii) ASIC-based implementations. The FPGA programmable logic combines the best of both DSP and ASIC technology. FPGAs are programmable and changeable (like DSPs); the designers can make changes quickly, without additional cost and time of ASIC design. On the other hand the FPGA implemented algorithms can process information faster than a general purpose DSP. Also the reconfigurability of Field Programmable Gate Array (FPGA) has improved the flexibility in the digital system design. Greater density and high speed FPGAs have enhanced the ability to realize special purpose processors for high-end applications.

Hence in this thesis, FPGA is chosen as the target hardware for the implementation of proposed speed estimator. There are still challenges in the FPGA implementation of NN, which affects chip area, performance and cost. Some of the major issues investigated in this thesis are;

- Computational Complexity,
- Bit Precision and
- Effective Utilization of Resources.

These major issues in FPGA implementation of proposed NN based speed estimator are investigated in the next section and their possible solutions are discussed.

6.4.1 Computational Complexity

The computational complexity of a neuron with n inputs is shown in Figure 6.4. The computations performed by a neuron are described by the following equation (6.1) and (6.2). The addition, multiplication and non-linear activation function are carried out on signed floating point numerals.

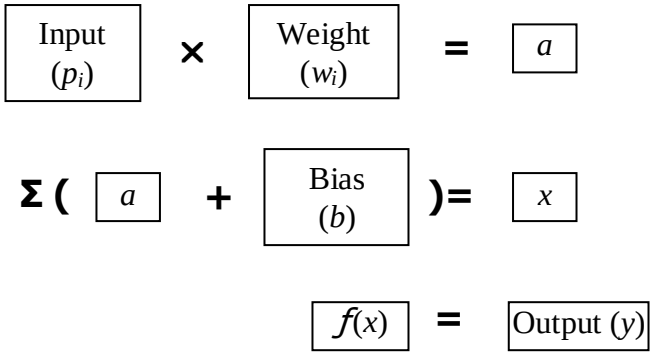


Figure 6.4 Computational Complexity of a Neuron

$$x = \sum_{i=1}^{n} p_i w_i + b \tag{6.1}$$

$$y = f(x) \tag{6.2}$$

The most time consuming block in the FPGA implementation of a neuron, is the computation of activation function. Hence, the computational complexity of neuron directly depends on activation function. The most popularly used activation function is tan-sigmoid function. The activation function used in the hidden layer of proposed NSE model III is also tan-sigmoid function. The tan-sigmoid function is defined as in equation (6.3).

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{6.3}$$

Though this function provides good accuracy the hardware implementation of this function is highly complex. The difficulties in computing exponential function (Himavathi S., et al 2001) are listed below:

1. It is an infinite series and hence has to be truncated to decrease the computational burden. This results in large truncation errors.
2. The use of Look Up Table (LUT)is not efficient due to non-uniform spacing between e^{-1} and e^{-2} and e^{-3} etc. This makes the look up table mapping inefficient. Interpolation between any two lookup table values is also complicated, as any combination is again a power of e.

To overcome this difficulty, a continuous, saturating nonlinear function which is simple to compute called Elliott function (Elliott D.L., 1993) is used. The function is defined in equation (6.4). This function is similar to the tan-sigmoid activation functions.

The advantages of Elliott function are:

- It is simple, continuous, differentiable, saturating.
- It is computationally less complex.
- It gives ease in digital implementation.

Plot of this function along with the tan-sigmoid function is shown in Figure 6.5 and the plot for gradient of both the function is shown in the Figure 6.6.

$$f(x) = \frac{x}{1+|x|} \tag{6.4}$$

Hence, the proposed NSE model III based speed estimator is trained with Elliott function in the hidden layers for real time data. The obtained NSE model III with Elliott function for online speed estimation has the structure 5-29(h)-1 (h-hidden layer with one neuron). Using the obtained data driven NSE model III with Elliott function in the hidden layers, the on-line rotor speed estimation is carried out. The performance of proposed NSE model III with Elliott function is analyzed extensively under various operating conditions. The sample results for proposed NSE model III with Elliott function based on-line speed estimator are shown below:

The measured and estimated rotor speed is presented in Figure 6.7(a) and its error is shown in Figure 6.7(b). From the result, it is clear that the proposed NSE model III with Elliott function estimates the speed with very good accuracy for practical data also. Hence, the performance of proposed NSE model III with Elliott function and NSE model III with Tansigmoid function are compared for speed estimation in various operating conditions for real data in terms of accuracy. Thus, result obtained are consolidate and Table 6.1.

Table 6.1 Performance Comparisons of the Proposed NSE Model III based Speed Estimator with Tan-sigmoid and Elliott Function in the Hidden Layers in Terms of Accuracy

Frequency (Hz)	Measured Speed (RPM)	Tan-sigmoid Function (5-25-1)		Elliott Function (5-29-1)	
		Estimated Speed (RPM)	% Error (RPM)	Estimated Speed (RPM)	% Error (RPM)
50	1485	1486.45	-0.0976	1484.21	0.0531
45	1355	1354.12	0.0649	1355.98	-0.0723
40	1216	1218.03	-0.1669	1217.45	-0.1192
35	1060	1061.09	-0.1028	1058.68	0.1245
30	910	912.28	-0.2505	911.10	-0.1208
25	760	758.31	0.2223	762.03	-0.2671

From the Table 6.4, it is clear that proposed NSE model III with Elliott function estimates the speed with very good accuracy for practical data as similar as NSE model III with tan-sigmoid function in the hidden layer neurons. But the NSE model III with Elliott function required four more additional hidden neurons when compared to NSE model III with tan-sigmoid function.

On FPGA, the execution time for the computation of a tan-sigmoid and Elliott function for a given input is computed. The tan-sigmoid function is implemented using series expansion method. Upto five terms are considered. The maximum frequency of operation on the FPGA is 20MHz. The execution time for tan-sigmoid function and Elliott function is found to be 35.05ns and 4.803ns respectively. The execution time of tan-sigmoid function is directly depended on number terms in the series expansion. If the number terms in the series increases, the time for execution also increases. From the results obtained, it is clear that the Elliott function is 7.2 times faster than the tan-sigmoid function. This would result in faster estimation of speed and hence reduces the computational complexity.

Thus it is concluded that the Elliott function performs well, superior in terms of computational complexity and found to be suitable for FPGA implementation.

However, the advantages overweigh the disadvantages. Hence, the NSE model III with Elliott function based speed estimator is studied for optimum bit precision.

6.4.2 Bit Precision

Another challenging issue in the FPGA implementation of proposed NSE model III with Elliott function based speed estimator is the identification of suitable bit precision. Bit precision or word length decides the output resolution. The inputs, outputs, weights and biases are represented using appropriate bits. Higher bit precision improves the performance of Speed estimator but increases the cost and memory size of the FPGA. Lower bit precision decreases cost and memory size of the FPGA but increases error in the speed estimation. Therefore, the optimal bit precision for satisfactory performance has to be identified. The N-bit precision for variable x can be found using the formula (6.5)-(6.7),

N-bit precision for variable x

$$Y = x \times 2^{n} \tag{6.5}$$

$$Y = \text{whole part } (Y) \tag{6.6}$$

$$Y = x \times 2^{-n} \tag{6.7}$$

Y is the n-bit precision value of x

Example: X= -2.2994797852830557

16-bit precision

$Y= -2.2994797852830557 \times 2^{16}$

Y= whole (-150698)

$Y= -150698 \times 2^{-16}$

Y= -2.2994689941406250

From the Table 6.2, it is clear that as bit precision increases, the accuracy also increases. Using the above formulae, the weights and biases of the proposed NSE model III with Elliott function based speed estimator are represented as various bit precision. The performance of proposed NSE model III with Elliott function based speed estimator is studied for various bit precision for real time data. The results obtained are consolidated and presented.

Table 6.2 Bit Precision

Actual value	**-2.2994797852830557**
8-bit	**-2.29**68750000000000
12-bit	**-2.299**3164062500000
16-bit	**-2.2994**689941406250
32-bit	**-2.299479785**1424664
64-bit	**-2.29947978528305**58

The proposed NSE model III with Elliott function based speed estimation for 8-bit precision is shown in Figure 6.8(a) and its error is shown in Figure 6.8(b). The proposed NSE model III with Elliott function based speed estimation for 12-bit precision is shown in Figure 6.9(a) and its error is shown in Figure 6.9(b). The proposed NSE model III with Elliott function based speed estimation for 16-bit precision is shown in Figure 6.10(a) and its error is shown in Figure 6.10(b). The proposed NSE model III with Elliott function based speed estimation for 32-bit precision is shown in Figure 6.11(a) and its error is shown in Figure 6.11(b). The proposed NSE model III with Elliott function based speed estimation for 24-bit precision is shown in Figure 6.12(a) and its error is shown in Figure 6.12(b).

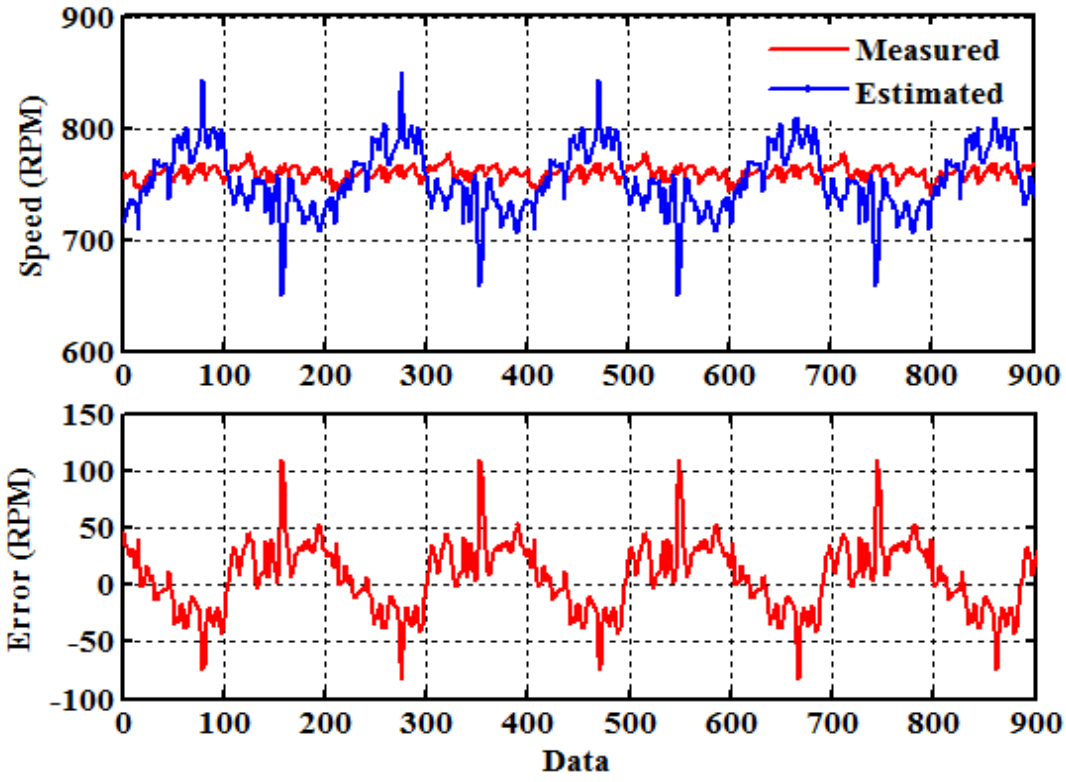


Figure 6.8 Estimated Rotor Speed for 8-bit precision

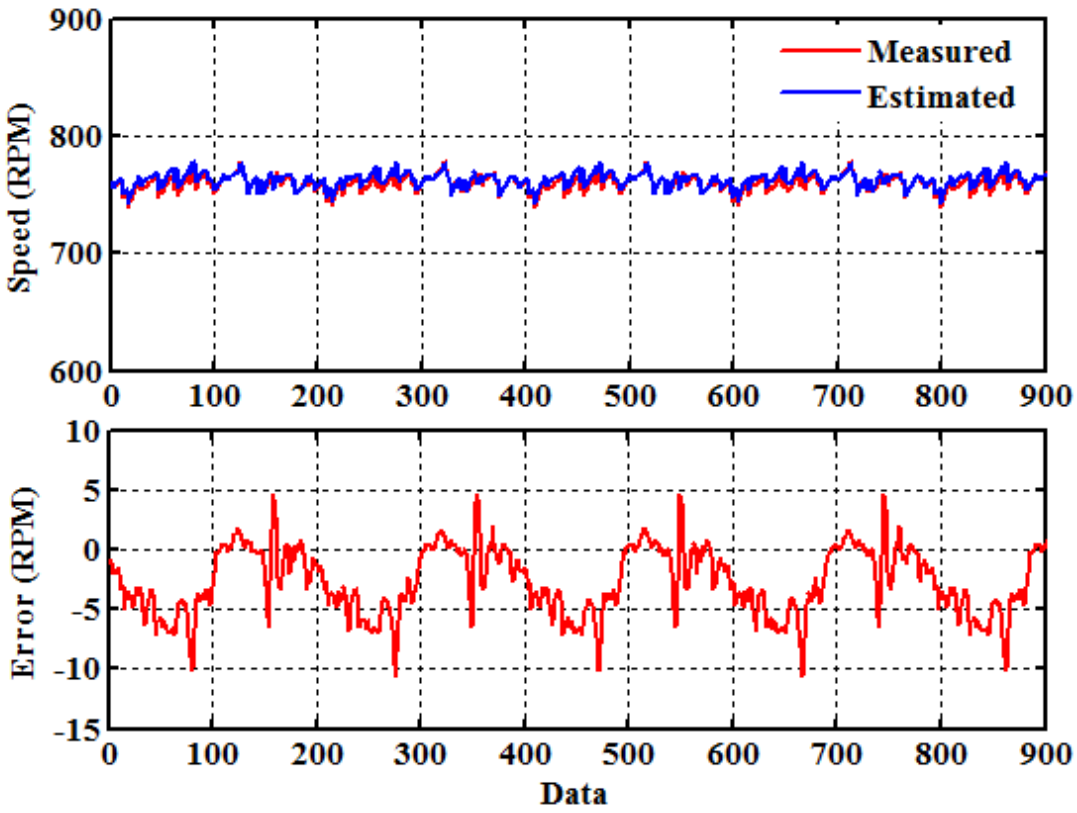


Figure 6.9 Estimated Rotor Speed for 12-bit precision

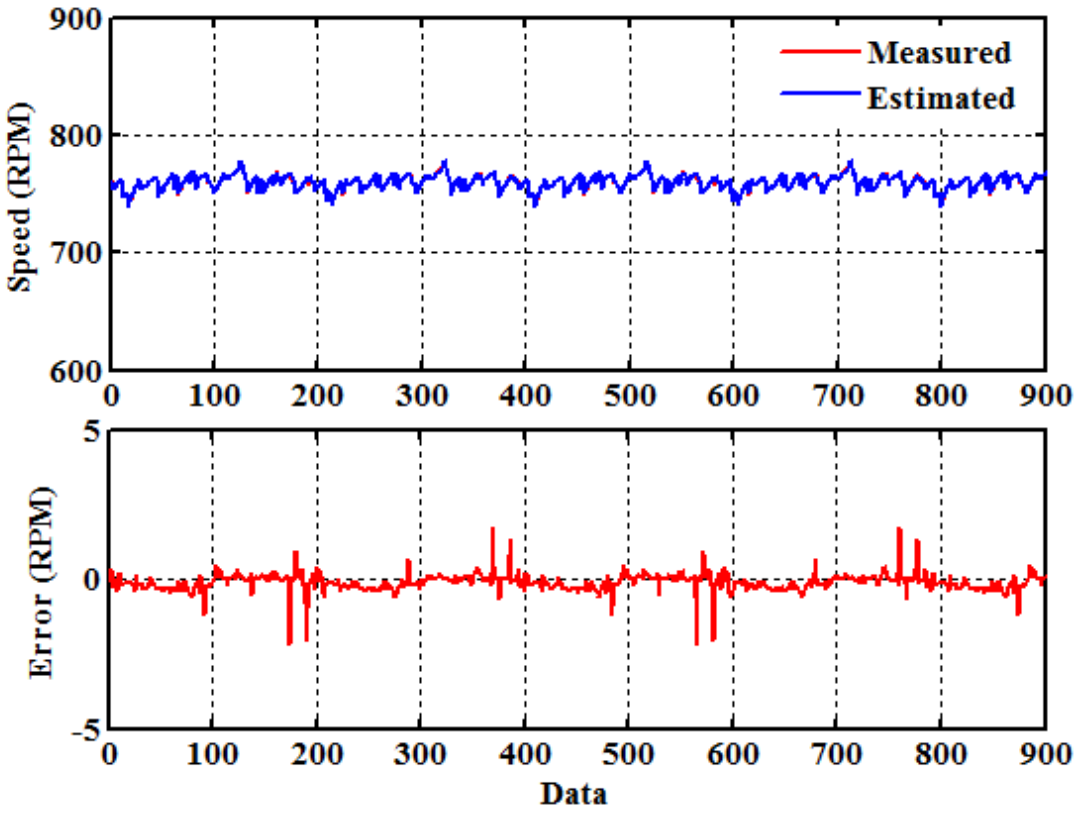


Figure 6.10 Estimated Rotor Speed for 16-bit precision

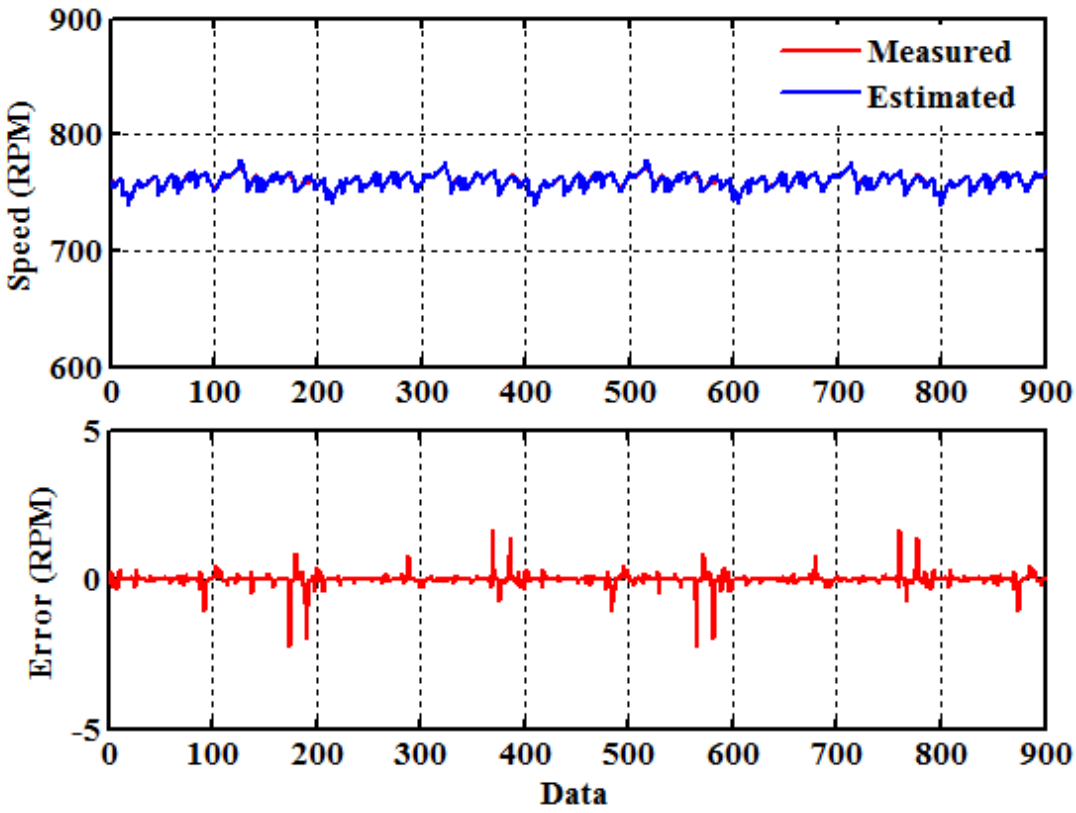


Figure 6.11 Estimated Rotor Speed for 32-bit precision

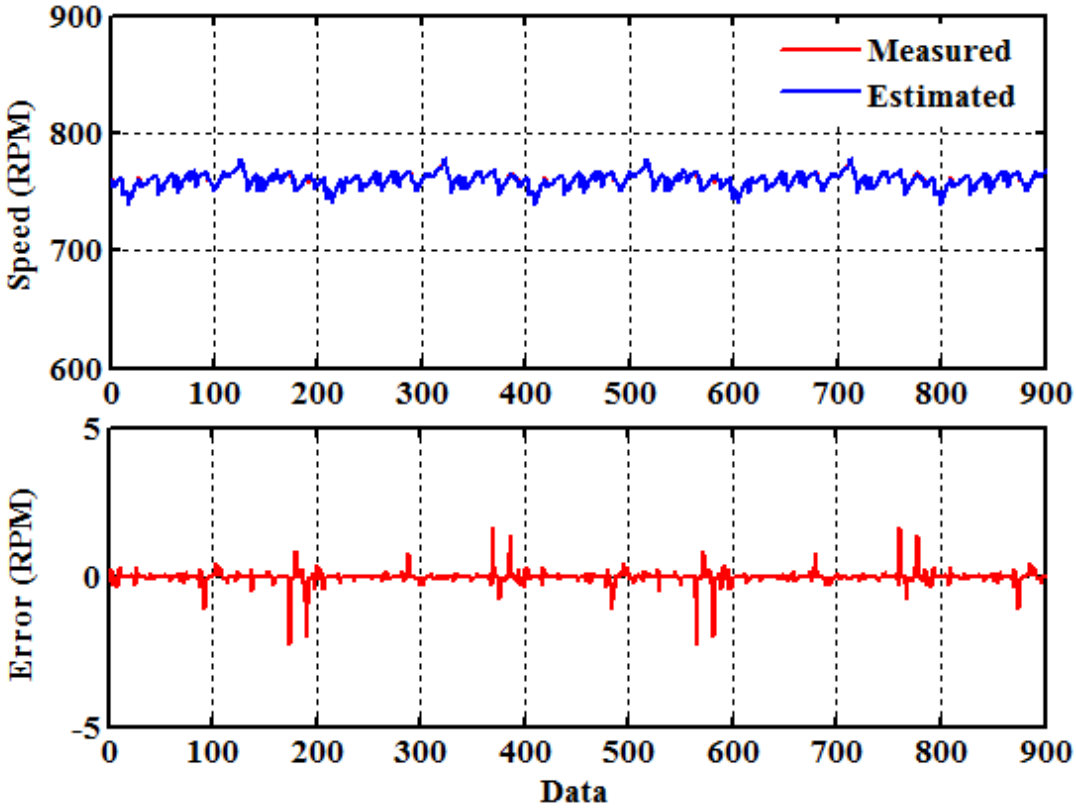


Figure 6.12 Estimated Rotor Speed for 64-bit precision

From the results obtained, it is concluded that 8-bit precision based estimator fails to estimate the speed. The 12-bit precision based speed estimator shows poor performance. But 16-bit, 32-bit and 64-bit precision based speed estimator shows good accuracy. The required accuracy is achieved using 16-bit and hence chosen for FPGA implementation of NSE model III with Elliott function based speed estimator.

6.4.3 Effective Utilization of Resources

FPGA implementation of proposed NSE model III with Elliott function can be carried out using two methods;

1. Implementation of whole architectures and
2. Layer Multiplexing

Implementation of whole architectures, high execution speed can be achieved. But resource requirement is large which in turn increases the system cost. In the literature (Himavathi S., et al 2007), layer multiplexing concept is proposed for NN implementation in FPGA to avoid the above problems. The layer multiplexing concept is aimed at reducing resource requirement, without much compromise on the execution time, so that a larger NN can be realized on a single chip at a lower cost. The sequential processes of SNC-NN have been exploited by using the layer multiplexing concept. Instead of realizing a complete network, only the neuron with a maximum number input is implemented in FPGA. The same neuron behaves as different neuron with the help of a control block. The layer multiplexing requires an additional control block to coordinate the computation of every layer of NN. The block diagram of proposed SNC-NN based Model III (Elliott function) with layer multiplexing concept is shown in the Figure 6.13.

The neuron unit acts as a single neuron with multiple inputs. Initially the weights and bias of the SNC-NN architecture for speed estimation is stored in control unit. Initially the clock, reset and start are given as the input. When the input data is given to the control unit, the start signal goes high and the respective weights and bias of the Layer1 which are collected from the control unit is loaded to the neuron unit and the operation of neuron is carried out and its output is given to the activation function and

the final output of that Layer1 is given as the input to the control unit. Then count signal shows that the neuron unit is ready for the next operation. The output of the first layer is also the input to the next unit and hence when the Layer2 weights and bias are loaded, the layer1 output from the control unit is also given as the input to the neuron unit. This process is repeated till the final output of the network is obtained. Realization of complete network with a single neuron implementation leads to a considerable resource reduction of FPGA. Hence, the proposed NSE model III with Elliott function based speed estimator is implemented in FPGA using layer multiplexing concept.

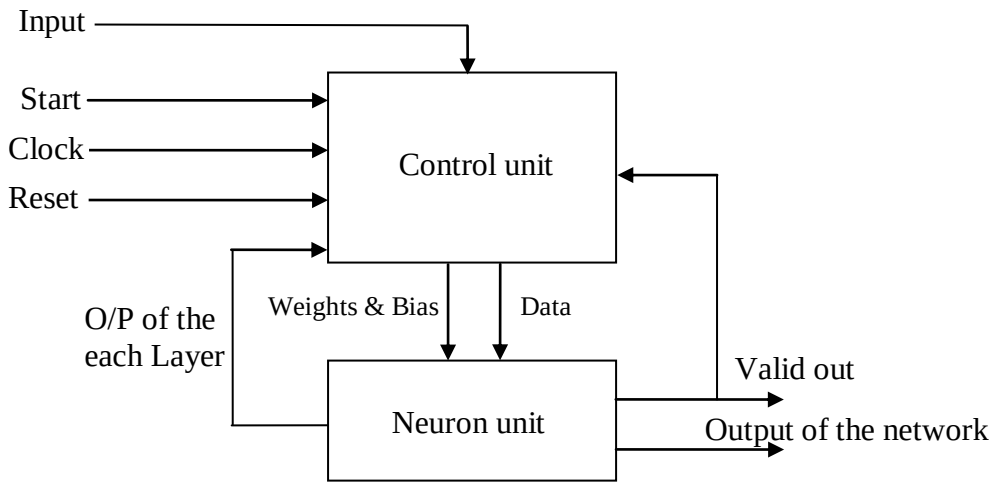


Figure 6.13 FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator with Layer Multiplexing

6.5 FPGA Implementation of Proposed NSE Model III with Elliott Function Based Speed Estimator

The proposed NSE model III based speed estimator is implemented in Spartan-3A DSP FPGA kit using layer multiplexing concept with Elliott activation functions in the hidden layer neuron for 16-bit precision. To implement the proposed speed estimator, a single neuron with the maximum of 45 inputs is implemented with layer control block assuming that generally in real time applications of NN the number of inputs to a neuron rarely exceeds 45. Using this, any SNC-NN based speed estimator with $p + (m-1) \leq 45$ can be implemented. where, p - total number of input to the model and m – total number of layers.

The speed estimator with architecture 5-29(h)-1 requires implementation of single layer/neuron with the maximum of 34 inputs. The implementation of proposed SNC-NN based speed estimator using layer multiplexing concept with Elliott function is detailed as a schematic diagram and presented in Figure 6.14. The implementation includes the SNC-NN with maximum input and a layer control block. The LAYER CONTROL BLOCK co-ordinates the computations of the different layers by placing the appropriate inputs, weights, biases & select the excitation function for each layer

The sequential operation of the SNC-NN NEURON block with LAYER CONTROL block is detailed below:

(i) The start signal is used to initiate the speed estimation process.

(ii) The LAYER CONTROL block has an internal LAYER COUNTER.

(iii) Inputs to the first layer neuron and its corresponding weights and bias are placed on the internal bus and read (RD) signal is initiated by the LAYER CONTROL block.

(iv) The ELLIOTT FUNCTION NEURON block reads the neuron inputs, weights and bias from the bus and performs the computations of a signal neuron. This involves Multiplication (MUL), Add (ADD) and computation of Elliott excitation function (E_F).

(v) At the end of computation the output of the neuron/layer is passed back to the LAYER CONTROL block and output enable signal is asserted.

(vi) On receipt of the layer output. The LAYER CONTROL block initiates the next layer computation. This continues for m-1 layers.

(vii) For the output layer computation, the LINEAR FUNCTION NEURON block is used.

(viii) The last layer uses LINEAR FUNCTION NEURON block and its enabled by LAYER COUNTER as shown in the Figure 6.14.

(ix) The output is placed on the output line and the valid output signal indicates the end of estimation.

(x) The valid output signal is used by the LAYER CONTROL block as the end of speed estimation process. The start signal is again asserted and the estimation is repeated for the next set of inputs.

The various steps involved in FPGA design flow are

- Design entry
- Synthesis
- Simulation
- Implementation
- Device programming

6.5.1 Design Entry

It is the process in which a design is created using HDL coding. The VHSIC (very high-speed integrated circuit) Hardware Description Language (VHDL) is an industry standard language specifically developed to describe digital electronic hardware and its attributes. It is a flexible language that can be applied to many different design situations used to describe hardware from the abstract to the concrete level. VHDL is rapidly being embraced as the universal communication medium of design. This language has several key advantages, including technology independence and a standard language for communication. Although this language looks similar to conventional programming languages, there are some important differences. A hardware description language is inherently parallel, i.e. commands which correspond to logic gates, are executed (computed) in parallel as soon as a new input arrives.

6.5.2 Synthesis

The goal of VHDL synthesis step is to create a design that implements the required functionality and matches the designer's constraint in speed, area or power. It translates the design into gates and optimizes it for the target architecture i.e., based on the data entered during the design entry stage a circuit is created. For an existing logical network, it must be possible to determine the function preformed by the network. This task is referred to as the analysis process. The reverse task of designing a new network that implements the desired function behavior is referred as the synthesis process. To convert the RTL (Register Transfer Level) description into gates, three steps typically occur. First, the RTL description is converted to unoptimized boolean description usually consisting of gates such as AND and OR

gates, flip-flops, and latches. This is functionally correct but completely unoptimized description. Next, boolean optimization algorithms are executed on this boolean equivalent description to produce an optimized boolean equivalent description. Finally, this optimized boolean equivalent description is mapped to Measured logic gates by making use of a technology library of the target process.

6.5.3 Simulation

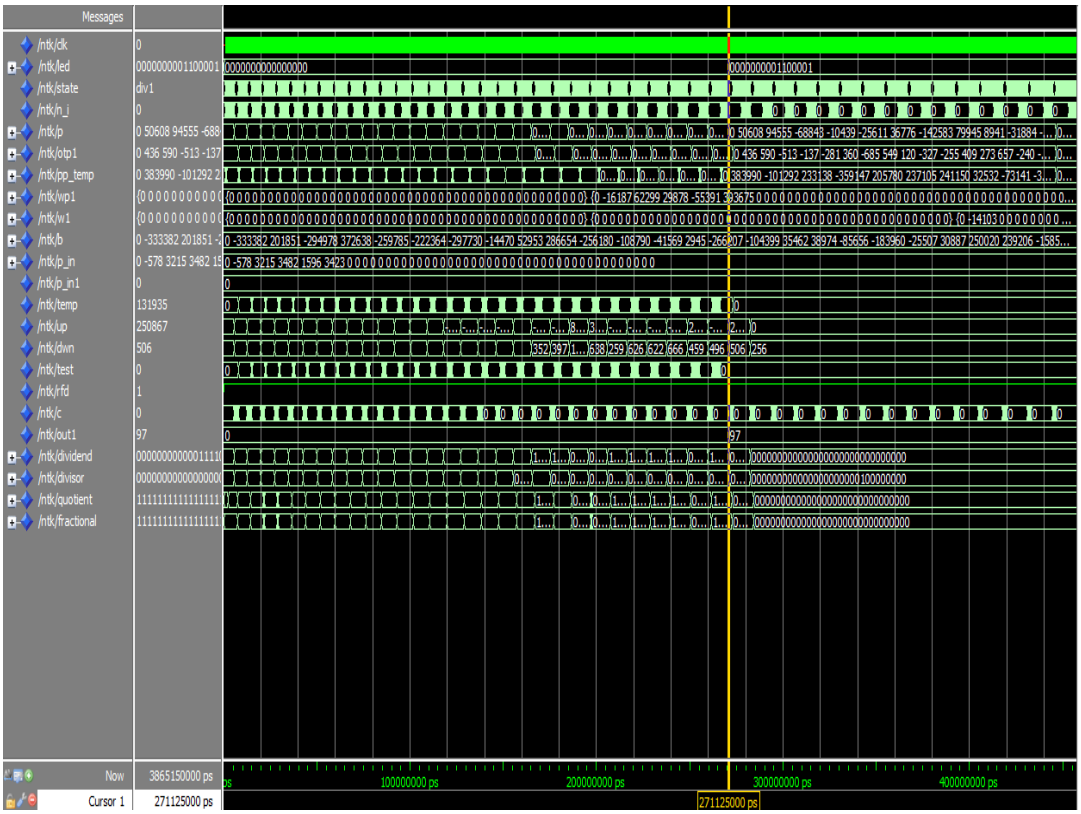
Verify the functionality of the circuit based on the circuit, based on the input provided by the designer. The simulation of the complete circuit identifies the errors if any. Some errors may be caused by incorrect connections between the blocks or some blocks may not be designed correctly. Depending on the errors encountered, it may be necessary to go back to the design. Successful completion of functional simulation suggests that the designed circuit will correctly perform all its functions.

6.5.4 Implementation

Convert the logical design file format created during design entry into a physical file format for a specific Xilinx architecture. The implementation stage consists of taking the synthesized netlist through translation, mapping and place and route. Place and route tool are used to take the design netlist and implement the design in the target technology device. The place and route tools place each primitive from the netlist into an appropriate location on the target device and then route signals between the primitives to connect the devices according to the netlist.

6.5.5 Device Programming

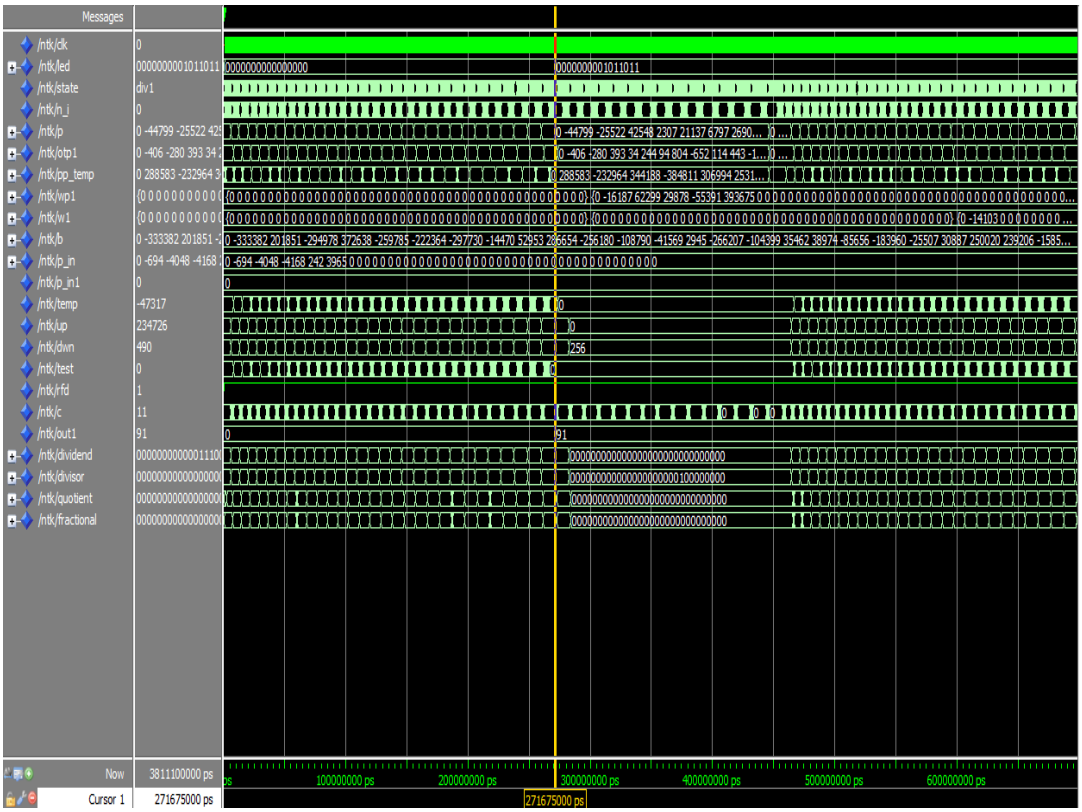
Create a programming file that can be downloaded to the target device. Once the programming file is created, it can de downloaded to the target device and the results can be verified. The performance of the proposed estimator is tested on Xilinx Spartan FPGA kit (3sd1800afg676-4) with practical data extensively. The sample results obtained are presented from Figure 6.15 to 6.18.



Input 1	Input 2	Input 3	Input 4	Input 5
-0.1411	0.7848	0.8500	0.3896	0.8353

Actual Output	FPGA Output
+0.9713	+0.97

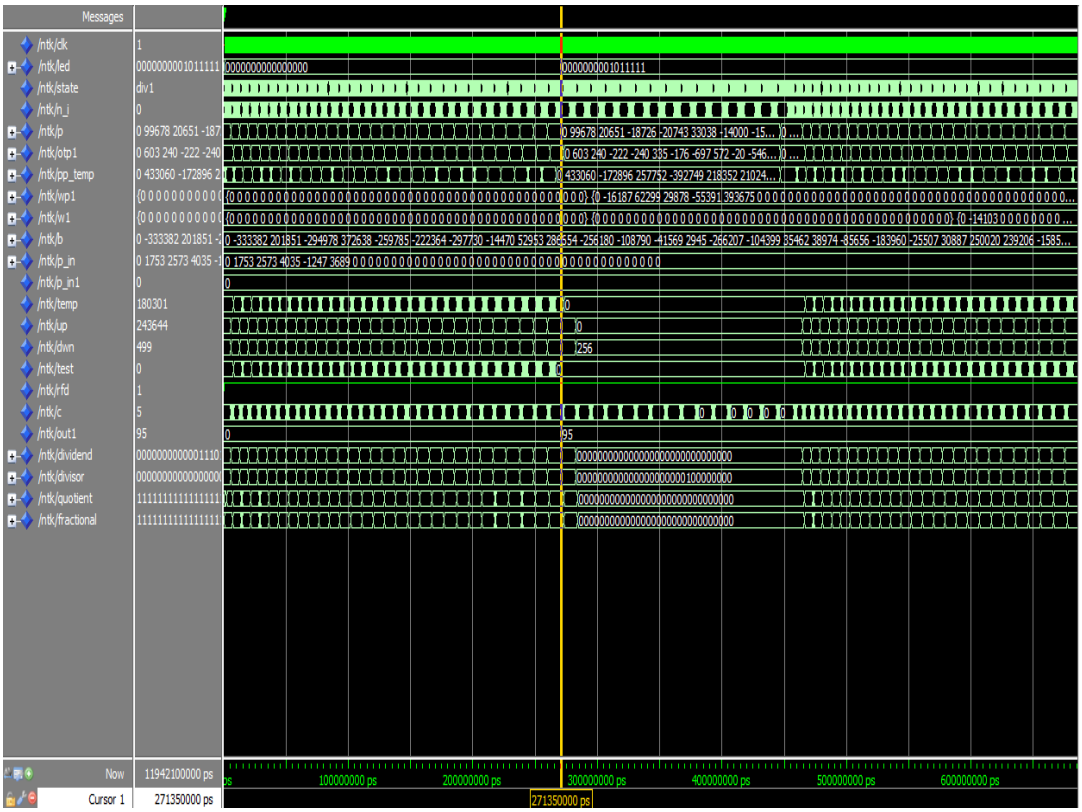
Figure 6.15 Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 1



Input 1	Input 2	Input 3	Input 4	Input 5
-0.1692	-0.9880	-1.0173	0.0588	0.9680

Actual Output	FPGA Output
+0.9125	+0.91

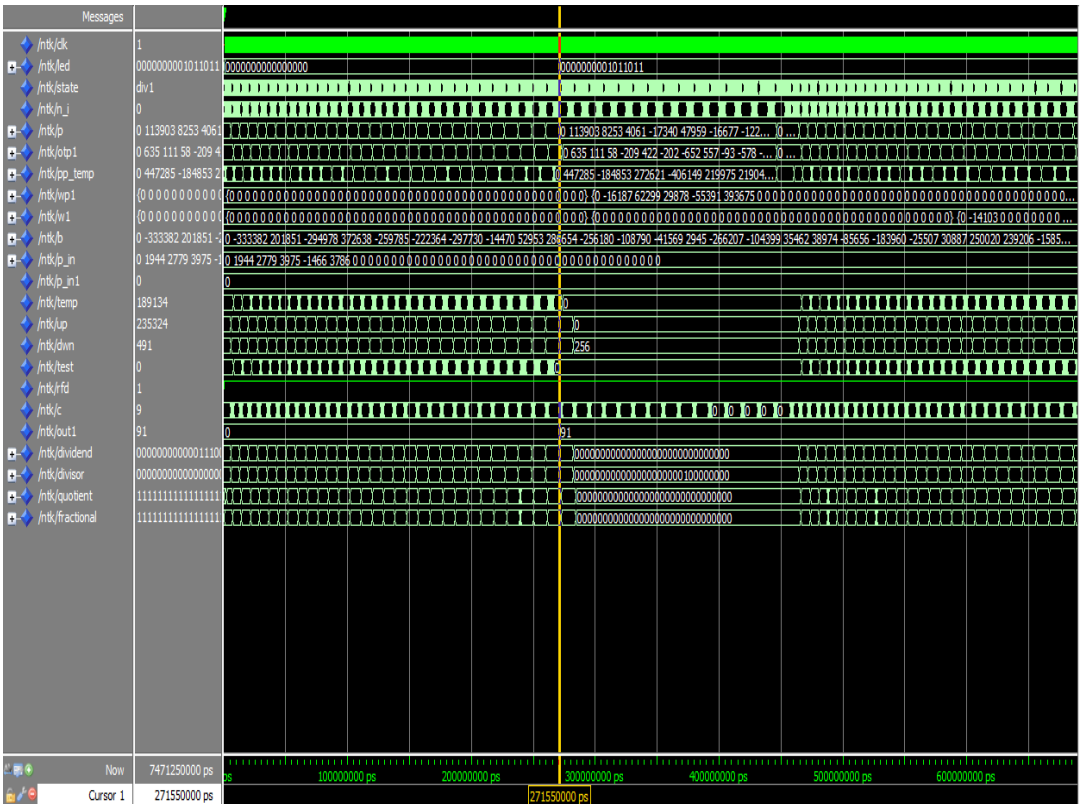
Figure 6.16 Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 2



Input 1	Input 2	Input 3	Input 4	Input 5
0.4279	0.6281	0.9849	-0.3043	0.9006

Actual Output	Practical Output
+0.9601	+0.95

Figure 6.17 Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 3



Input 1	Input 2	Input 3	Input 4	Input 5
0.4745	0.6785	0.9705	-0.3579	0.9244

Actual Output	Practical Output
+0.9056	+0.91

Figure 6.18 Results for FPGA Implementation of Proposed Model III with Elliott Function Based Speed Estimator for Test Data 4

Device Utilization Summary

Target Device : xc3sd1800a-4-fg676

Logic Utilization	Used	Available	Utilization
Number of Slices	7921	16640	47%
Number of Slice Flip Flops	7530	33280	22%
Number of 4 input LUTs	12976	33280	38%
Number of bonded IOBs	17	519	3%
Number of BRAMs	1	84	1%
Number of GCLKs	1	24	4%
Number of DSP48s	10	84	11%

On implementation, it is found that the maximum frequency of operation is 20 MHz and clock cycles required for the speed estimation is 5400 respectively. The timing requirement satisfies the requirement for speed estimation. This work would form the basis for the development and realization of an efficient and intelligent Speed Estimator based on neural network, and the usefulness in complex Sensorless Controlled IM Drives.

6.6 Conclusion

In this chapter, the performance of proposed NSE model III based speed estimator is validated for real time data. The experimental set up is built to collect the real time data. The NSE model III based speed estimator is designed and tested using the real time data extensively. The proposed NSE model III based speed estimator is found to perform well for the real time data.

The issues in FPGA implementation of proposed NSE model III based speed estimator are investigated and their possible solutions are discussed. The major issues investigated in this thesis are; computational complexity, bit precision and effective utilization of resources. A new nonlinear simple to compute activation function named as “Elliott function” is identified and proposed for speed estimation. The NSE model III based speed estimator is trained with Elliott function in the hidden layer neurons for the real time data. The performance of NSE model III with Elliott

function for real time data is as good as the NSE model III with tan-sigmoid function for on-line speed estimation. But, the NSE model III with Elliott function required four more additional hidden neurons when compared to NSE model III with tan-sigmoid function. The Elliott function is found to be 7.2 times faster than the tan-sigmoid function. Thus the Elliott function performs well and reduces the computational complexity. This results in faster speed estimation. However, the advantages overweigh the disadvantages. Thus it is concluded that the Elliott function performs well, superior in terms of computational complexity and found to be suitable for FPGA implementation. The NSE model III with Elliott function is tested for various bit precision using real time data and lowest bit precision needed for good performance of the on-line speed estimator is identified as 16-bits. The concept of layer multiplexing is adopted for effective resource utilization.

The proposed NSE model III based speed estimator with identified bit precision and Elliott function is implemented using layer multiplexing technique and tested on Spartan FPGA kit (xc3sd1800afg676-4). From the synthesis report it is found that the slices required for the proposed NSE model III with Elliott function based speed estimator is 7921, the maximum frequency of operation is 20 MHz and clock cycles required for the speed estimation is 5400 respectively. The FPGA implementation validates the suitability of the proposed NSE model III with Elliott function based speed estimator for real time applications.

Thus the proposed NSE model III with Elliott function based on-line speed estimator performs well for practical data. It is less complex, implementable in hardware and hence is an efficient low cost alternative for all applications of speed estimator in sensor-less controlled IM Drives.

Chapter 7

Conclusion

In this chapter, the importance of research work is discussed, the major contributions are summarized and the possible directions for future work are indicated.

7.1 Importance of Research Work

The availability of high speed computing devices has made high performance vector controlled drives reliable and popular. The knowledge of machine speed is mandatory for this control. The speed can be measured with optical encoders, electromagnetic sensors or brushless DC tacho-generators. However, the use of these electromechanical devices present some limitation in their applications, like increased cost of the drive, reduce mechanical robustness, low noise immunity etc. They affect the machine inertia and require special attention in hostile environments. Hence, it becomes necessary to go in for speed estimators. Fast and accurate speed estimators are currently under study.

It is possible to estimate the speed from machine terminal voltages and currents. The conventional methods are direct synthesis from state equations, rotor slot harmonic, extended Kalman filter (EKF), extended Luenbergern observer (ELO), saliency techniques and model reference adaptive system (MRAS). All these techniques use complex mathematical model of the motor which includes many assumptions. The estimation also is not robust to parameter variations. The time taken for computation is long.

Neural Network (NN) based speed estimator is robust to parameter variations, immune to noise and avoids the use of mathematical models. Such a system is not restricted by the many assumptions used in the conventional methods and is capable of mapping to any degree of non linearity. It can also be designed to yield the results

more quickly. This thesis investigates the potential of Neural Network based on-line speed estimation for sensorless vector controlled IM drives and their possible solutions are discussed.

The objective of the research work is to identify the best possible NN based solution for on-line speed estimation for sensorless vector controlled IM Drives. The use of neural learning algorithm for adaptation in MRAS based systems and data based neural network models for all types of vector control are investigated in this thesis.

7.2 Summary of the Thesis

The thesis investigates the use of neural learning algorithm for adaptation in MRAS based speed estimation schemes. A number of drawbacks exist with MRAS based speed estimation depending on the equation used, state variable chosen and type of vector control. The advantages of reactive power based MRAS is more prominent when used for sensorless indirect rotor field oriented controlled (IRFOC) drives. The equation becomes independent of rotor flux and stator resistance. Hence, the simplified equation for reactive power in IRFOC is very useful and widely used with PI adaptation. However, it fails in certain regenerative modes. A novel simplified reactive power based MRNLAS (Q-MRNLAS) for speed estimation is proposed in this thesis to overcome the above problem.

The proposed scheme combines the advantages of simplified reactive power technique and the capability of neural learning algorithm for on-line speed estimation in sensorless indirect vector controlled induction motor drives. The performance of proposed on-line speed estimator is compared with existing Q-MRAS in terms of accuracy and regenerating mode of operation. The proposed simplified Q-MRNLAS is independent of flux and pure integrator problems. It eliminates the need for stator resistance estimation and hence reduces the complexity of the drive. The neural adaptation is accurate, simple and provides stable operation in regenerating modes. Hence, it is concluded that the proposed estimator is a promising alternative for existing PI based Q-MRAS for on-line speed estimation in sensor-less indirect vector controlled induction motor drives.

The proposed method again suffers from the disadvantages of conventional equations based speed estimation schemes namely inaccuracies in motor parameters used and the need for an online rotor resistance estimator. To further exploit the advantages of neural networks, it is proposed to build a data based model for on-line speed estimation. In this thesis, three types neural network model based on-line speed estimator trained from input/output data are developed and proposed for sensorless controlled induction motor drives.

To build the NN model, three types of popular neural architectures (SLFF, MLFF and SNC) and learning algorithms (BPM, VLR and LM) are considered for investigation. For comparison, all the three NN models are trained with the same input/output data, using same learning algorithms, for the same number of epochs. From the training MSE obtained, NN architectures trained with LM algorithm have performed well with minimum number of neurons when compared to BPM and VLR trained NN architectures. Hence, it is concluded that LM algorithm is most suitable for offline training of speed estimation. To determine the most suitable architecture, the performance of LM-trained NN models using three architectures is designed for on-line estimation of rotor speed under various operating conditions.

From the analysis, it is inferred that the steady state and dynamic performance of SNC-NN and MLFF-NN model are found to be similar and superior as compared to SLFF-NN. The SNC-NN model resulted in structurally compact and computationally less complex model as compared to MLFF-NN models. The SNC-NN can be self organized which greatly aids design automation whereas MLFF-NN lacks the design methodology. Thus, SNC-NN model is observed to combine the advantage of multilayer mapping capability of MLFF-NN model and self-organizing feature of SLFF-NN model. Hence, it can be concluded that SNC-NN architecture trained with Levenberg Marquardt algorithm is found to be most suitable.

The SNC-NN model trained with Levenberg Marquardt algorithm using flux based relationship is named as neural speed estimator model I. The performance of proposed neural speed estimator model I is compared with existing RF-MRAS in terms of accuracy. The proposed neural speed estimator model I based on rotor flux is as good as the existing RF–MRAS method for all operating conditions. The data based NN

approach is validated using this model. However, neural speed estimator model I is flux dependent and suffers from all the drawbacks of flux estimation in sensorless controlled induction motor drives.

Neural speed estimator model II based on stator frequency is proposed to overcome the drawbacks of neural speed estimator model I. The proposed neural speed estimator model II based speed estimator is independent of motor flux and combines the advantages of neural network and the simplified reactive power based techniques. It exhibits good performance over a wide operating range with good accuracy. A performance comparison is carried out with Q-MRNLAS method under no load and full load. It is demonstrated that the proposed neural speed estimator model II performs as well as the Q-MRNLAS, for all operating conditions. The robustness of the proposed neural speed estimator model II is illustrated for R_r variation. This is done for slight and large detuning for R_r. The Q-MRNLAS fails to track the speed, whereas the proposed estimator tracks the speed accurately with variations in R_r. The superiority of the proposed neural speed estimator model II over Q-MRNLAS is well demonstrated.

However, the proposed neural speed estimator model II has a few disadvantages. It can be used only for sensor-less indirect vector controlled induction motor drives as the simplified reactive power equations are applicable only for IFOC. Also the proposed estimator uses mathematical equation for computation of ω_e. Hence, an accurate knowledge of machine parameters is mandatory.

A generalized data driven neural network model based on reactive power is proposed for on-line speed estimator and it is named has neural speed estimator model III. The proposed NSE model III based speed estimator is suitable for all types of sensorless control; a black box approach without the need for motor parameters would provide a novel solution. The NSE model III based speed estimator was extensively tested for a number of operating conditions and shown to perform well. Its performances are compared with Q-MRNLAS and superiority of the proposed scheme over Q-MRNLAS is well demonstrated.

The NSE model III based speed estimator has a number of advantages

- It is independent of flux and hence, overcomes all flux based estimation problems.
- It is independent of machine parameters.
- Uses a Black box approach.
- It is a generalized model and hence, suitable for all type of sensorless control of IM drives.
- Does not require additional estimator and hence, reduces the complexity of drive system.
- The SNC architecture trained using LM algorithm is found to be compact, less complex, easy in design, accurate and most suitable for speed estimation.

Based on the detailed investigations, it is concluded that NSE model III is the most suitable model for on-line speed estimation as for all types of sensor less controlled IM drives. The proposed novel method is implemented in digital hardware and validated using practical motor setup.

The experimental set up is built to collect the real time data. The NSE model III based speed estimator is designed and tested using the real time data extensively. The proposed NSE model III based speed estimator is shown to perform very well for the real time data. The issues in FPGA implementation of proposed NSE model III based speed estimator are investigated and their possible solutions are discussed. The major issues investigated in this thesis are; computational complexity, bit precision and effective utilization of resources.

To overcome the computational complexity of tan sigmoid excitation functions a new nonlinear simple and compute activation function named as “Elliott function” is identified and proposed for speed estimation. The NSE model III based speed estimator is trained with Elliott function in the hidden layer neurons for the real time data. The performance of NSE model III with Elliott function in the hidden layer neurons for real time data is as good as the NSE model III with tan-sigmoid function for on-line speed estimation. But, the NSE model III with Elliott function required four more additional hidden neurons when compared to NSE model III with

tan-sigmoid function. The Elliott function neuron is found to be 7.3 times faster than that of tan-sigmoid function and the advantages overweigh the disadvantages. Thus, it is concluded that the Elliott function performs well, superior in terms of computational complexity and found to be suitable for FPGA implementation. The NSE model III with Elliott function is tested for various bit precision using real time data and lowest bit precision needed for good performance of the on-line speed estimator is identified as 16-bits. For effective resource utilization, the concept of layer multiplexing is adopted for NSE model III with Elliott function based speed estimator in FPGA implementation.

The proposed NSE model III with Elliott function based speed estimator with identified bit precision and activation function is implemented using layer multiplexing technique and tested on Spartan FPGA kit (xc3sd1800afg676-4). From the synthesis report, it is found that the slices required for the proposed NSE model III with Elliott function based speed estimator is 7921, the maximum frequency of operation is 20 MHz and clock cycles required for the speed estimation is 5400 respectively. The FPGA implementation validates the suitability of the proposed NSE model III with Elliott function based speed estimator for real time applications.

Thus the proposed NSE model III with Elliott function based on-line speed estimator performs well for practical data. It is less complex and implementable in hardware. Hence, it is an efficient low cost alternative for all applications of speed estimator in sensor-less controlled IM Drives.

7.3 Major Contributions of the Thesis

The major contributions of the research work are listed below:

- ✓ Neural learning algorithm is proposed for simplified Q-MRAS based speed estimator in sensor-less indirect vector controlled IM drives.
- ✓ The proposed Q-MRNLAS provides stable operation for a wide operating range. It overcomes the problem of instability present in PI adaptation in regenerating mode.

- ✓ Three different type of data based NN model for speed estimation is proposed and most suitable model is identified.
- ✓ SNC-NN architecture trained with LM algorithm is identified to have high degree of nonlinear mapping capability and suitable for on-line speed estimation.
- ✓ Proposed NSE model III based speed estimator is identified as accurate, independent flux, less complex, robust to integrator drift problems, independent motor parameters and most suitable model for all types of sensorless IM Drives.
- ✓ Proposed NSE model III based speed estimator is trained and validated for real time data.
- ✓ Proposed NSE model III based speed estimator is investigated for FPGA implementation issues are: computational complexity, bit precision and effective utilization of resources.
- ✓ The most time consuming block in the FPGA implementation of a neuron, is the computation of tan-sigmoid function and it is replaced by new function called "Elliott function".
- ✓ Proposed NSE model III based speed estimator is trained and tested using Elliott function in hidden layers for real time data.
- ✓ Reduction in computational complexity for online speed estimation using Elliott function for FPGA implementation.
- ✓ The lowest bit precision needed for good performance of the on-line speed estimation of proposed speed estimator is identified.
- ✓ The concept of layer multiplexing is adopted for proposed speed estimator in FPGA implementation for effective resource utilization.

This work would form the basis for the realization of NN based estimator trained from the input/output data, and their usefulness in complex sensorless vector controlled IM drives.

7.4 Scope for Future Work

In this thesis, the proposed speed estimators have been thoroughly investigated for sensorless indirect vector controlled induction motor drives. The same estimator can

be tested for other types of speed-sensorless controlled induction motor drives in future. The proposed NN models for speed estimation are trained with direct search techniques (Gradient based optimization) could still be trained with random search techniques (Soft computing based optimization) to improve NN model accuracy and compactness in future. The FPGA implementation can be replaced by neural chip using SNC architecture. The speed estimator coding can be optimized and developed for embedded application in future. In the future, speed estimator may be embedded in the motor setup to facilitate sensorless controlled IM drives.

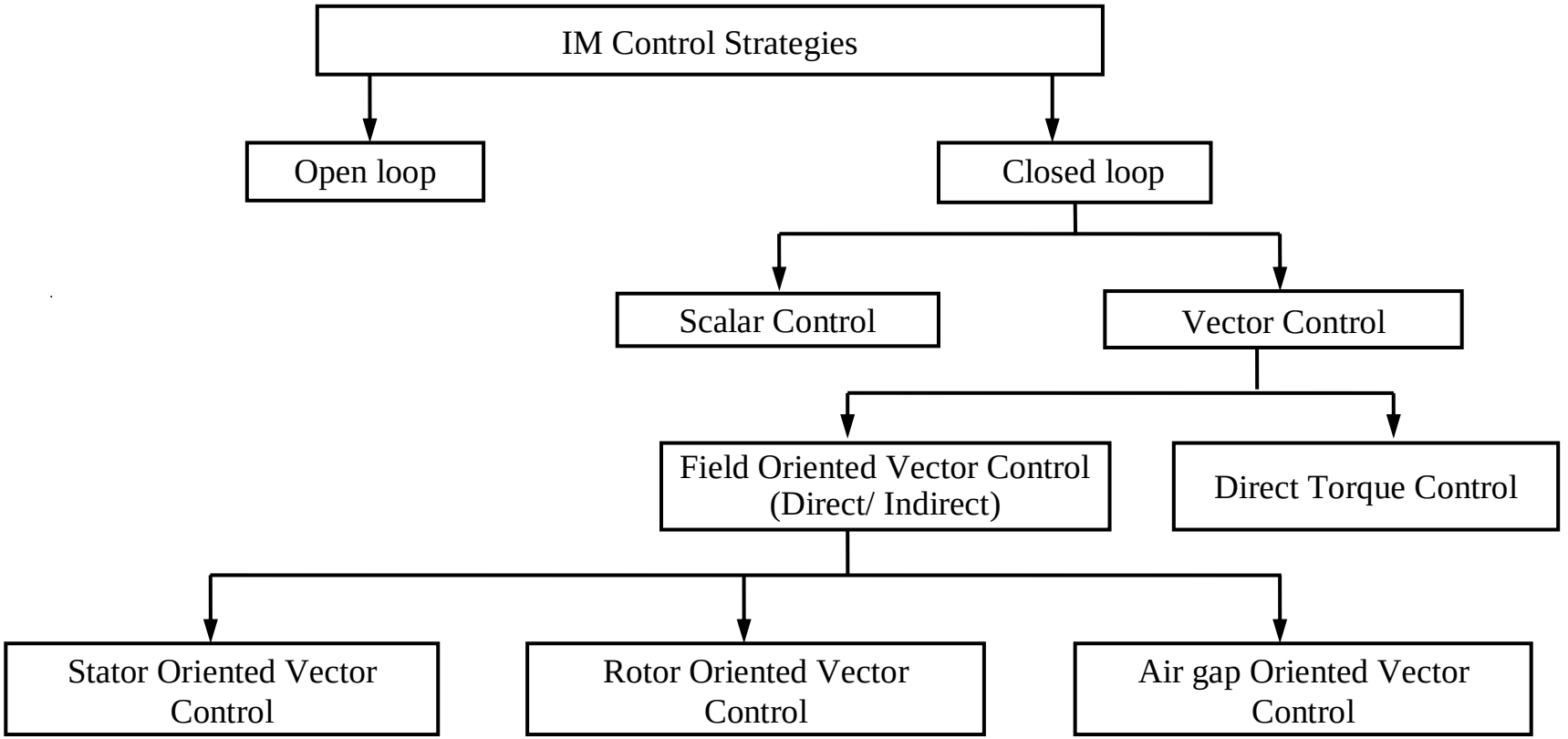
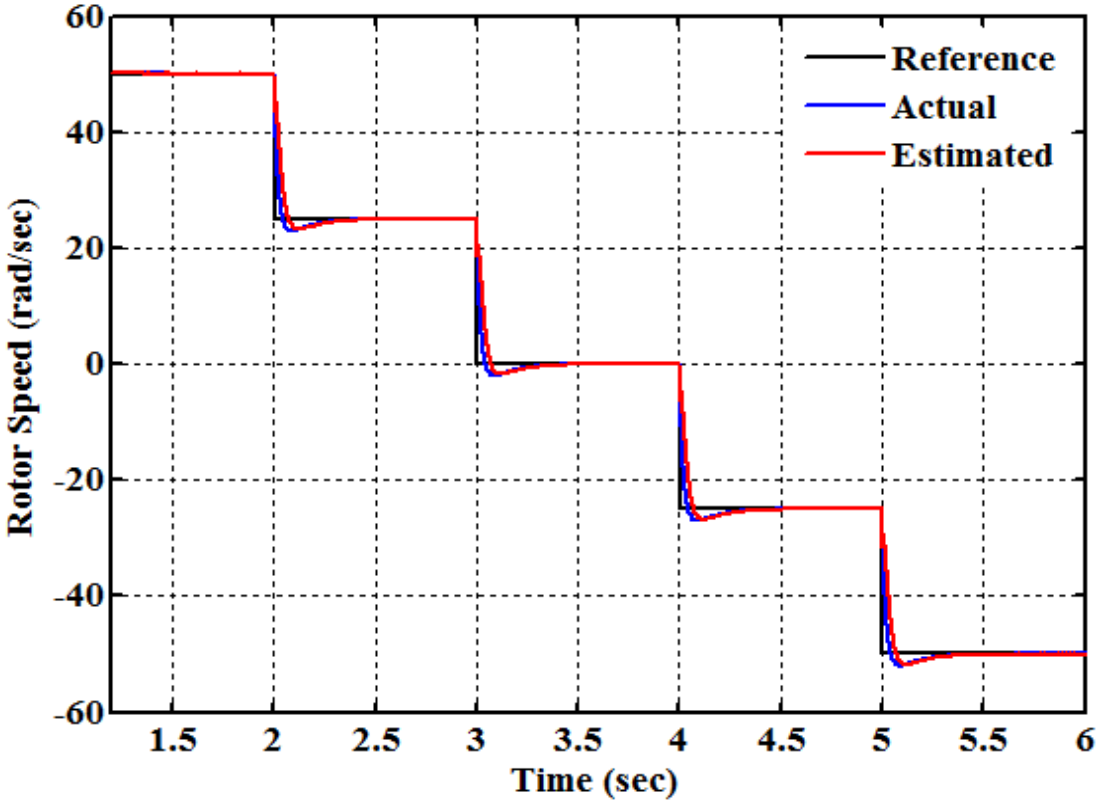
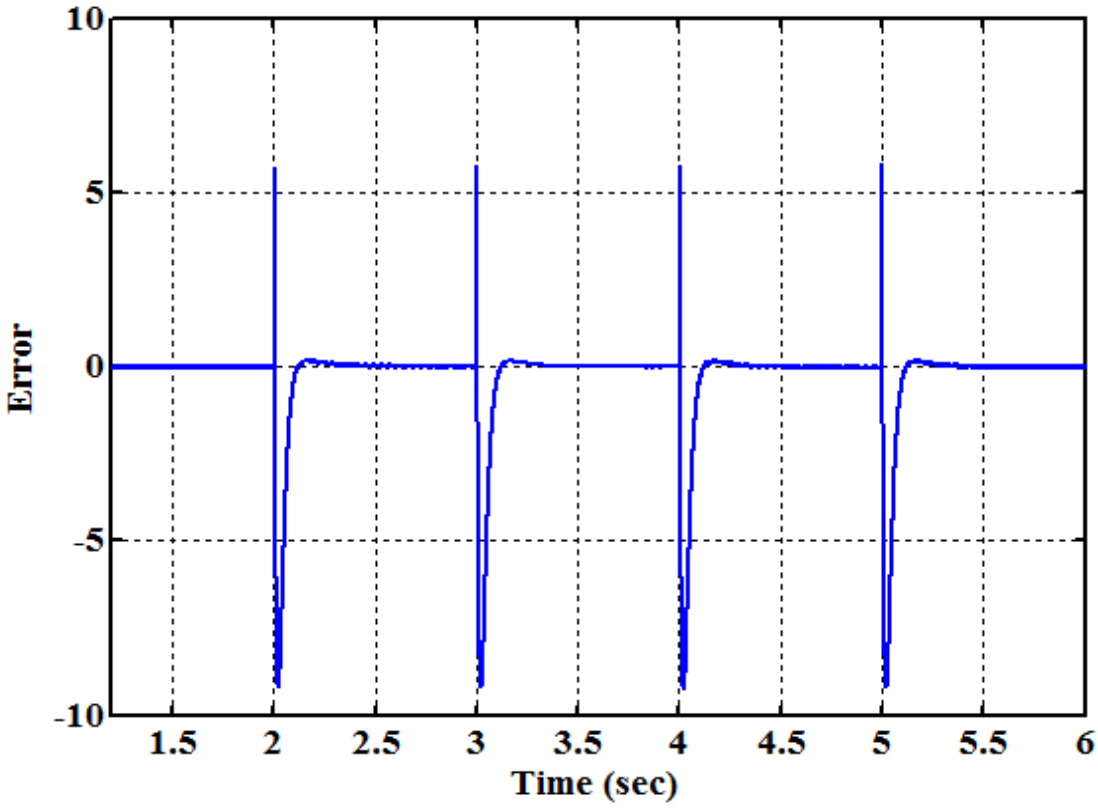


Figure 1.1 Overview of Induction Motor Control Strategies

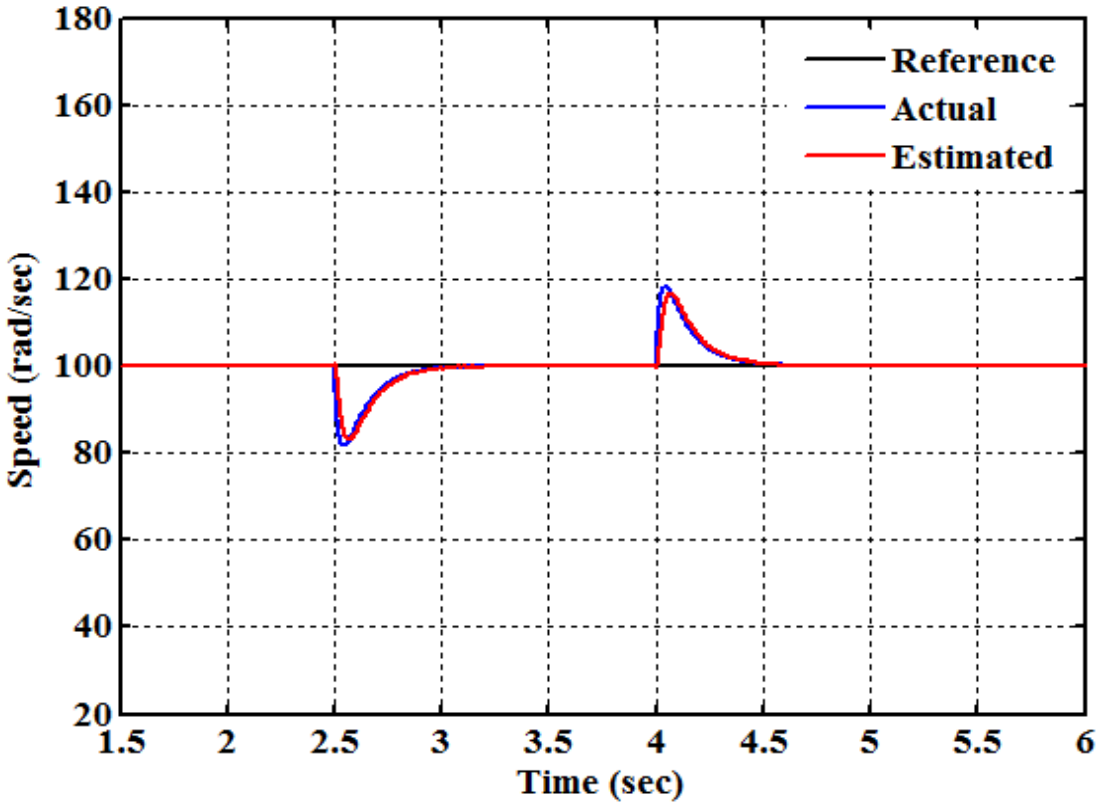


(a)

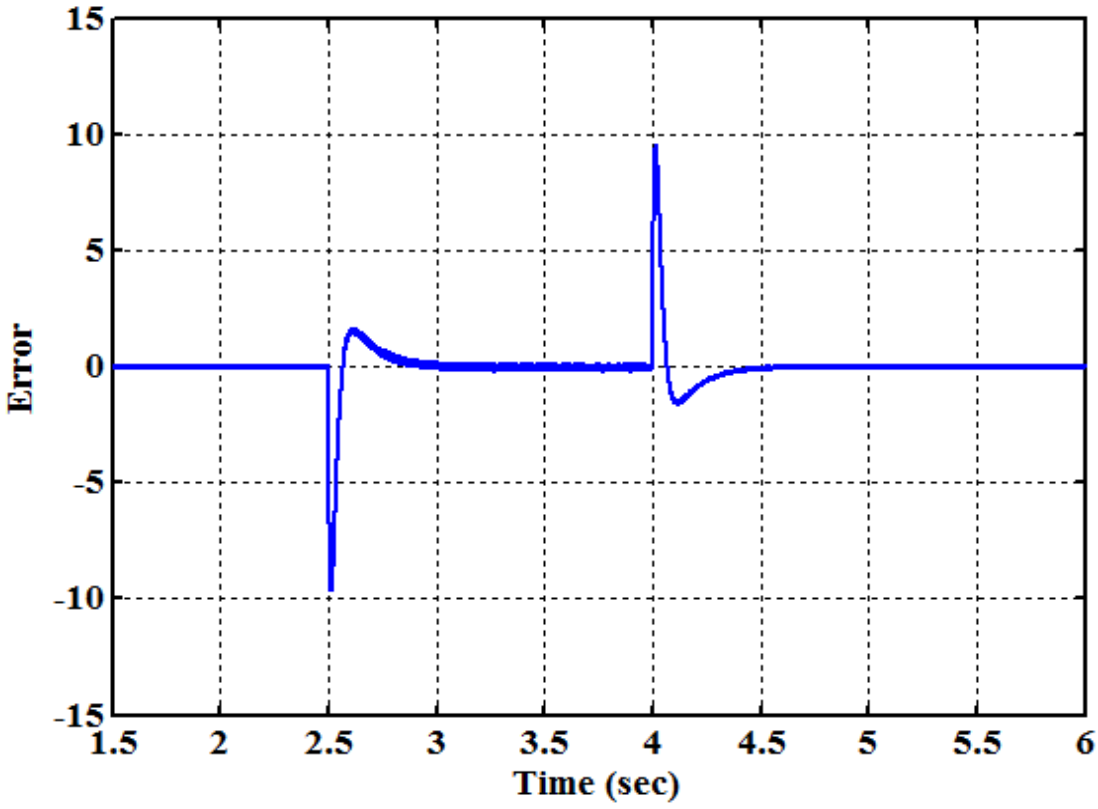


(b)

Figure 2.4 Performance Curves for Test Condition-1: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated

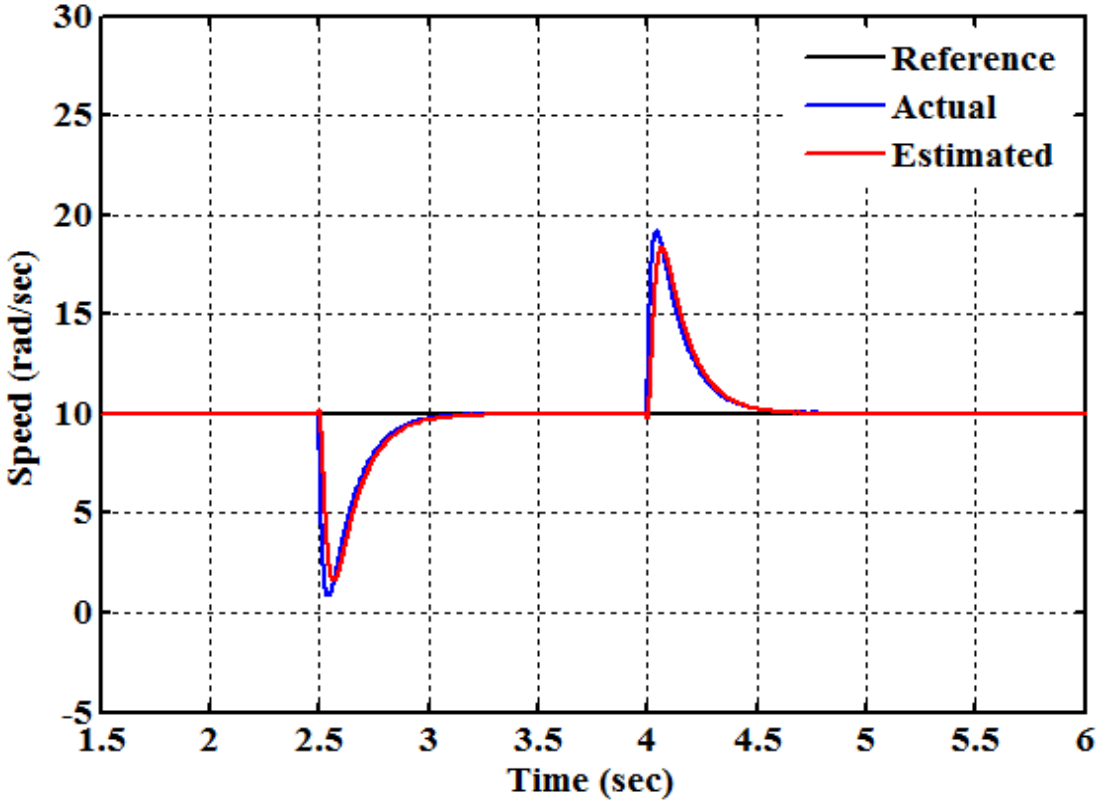


(a)

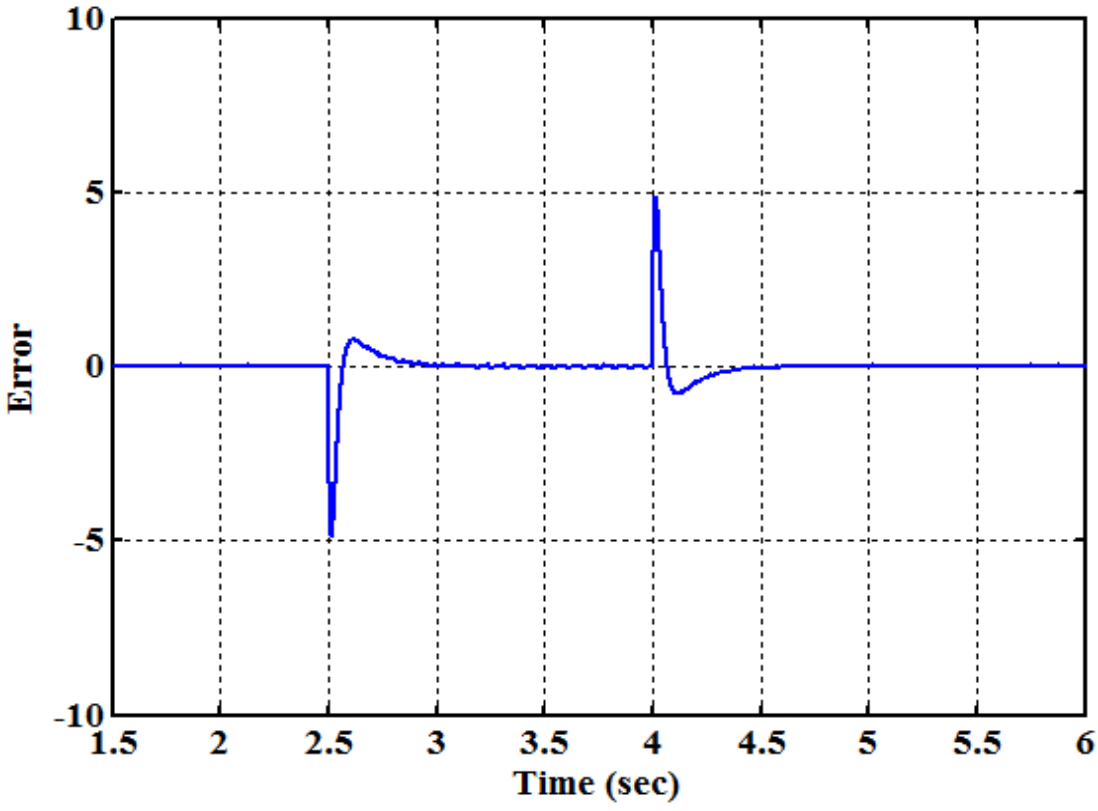


(b)

Figure 2.5 Performance Curves for Test Condition-2: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated

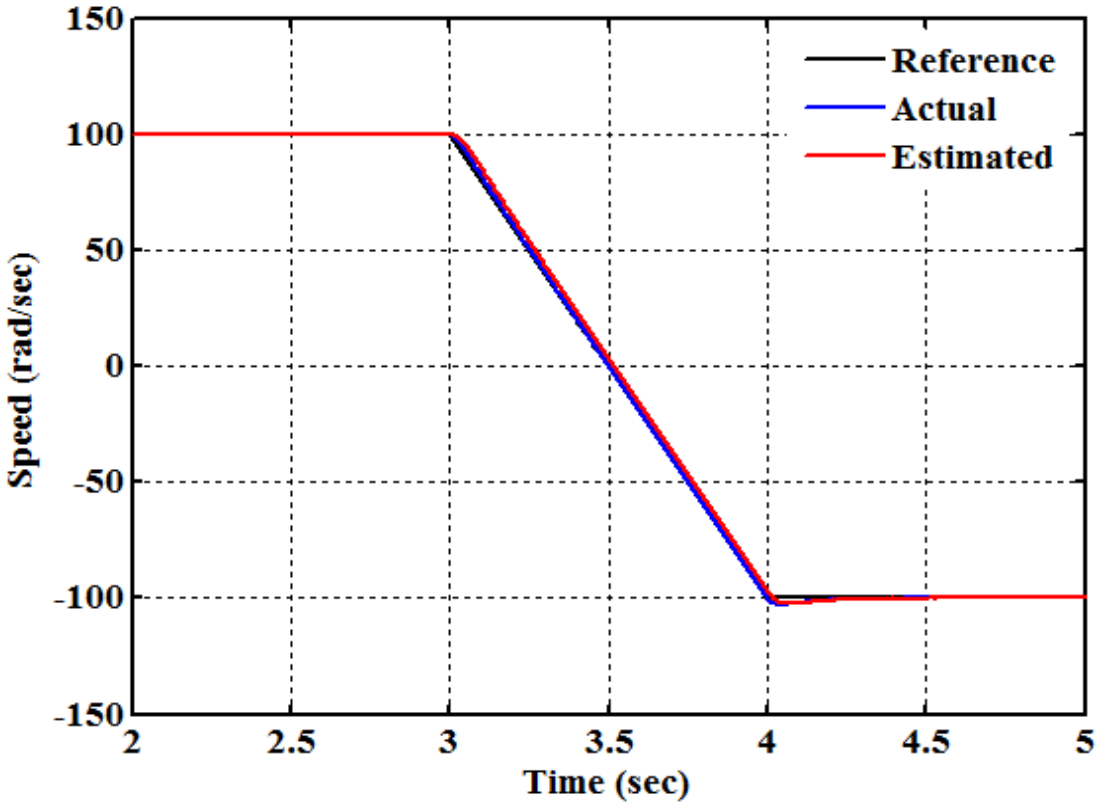


(a)

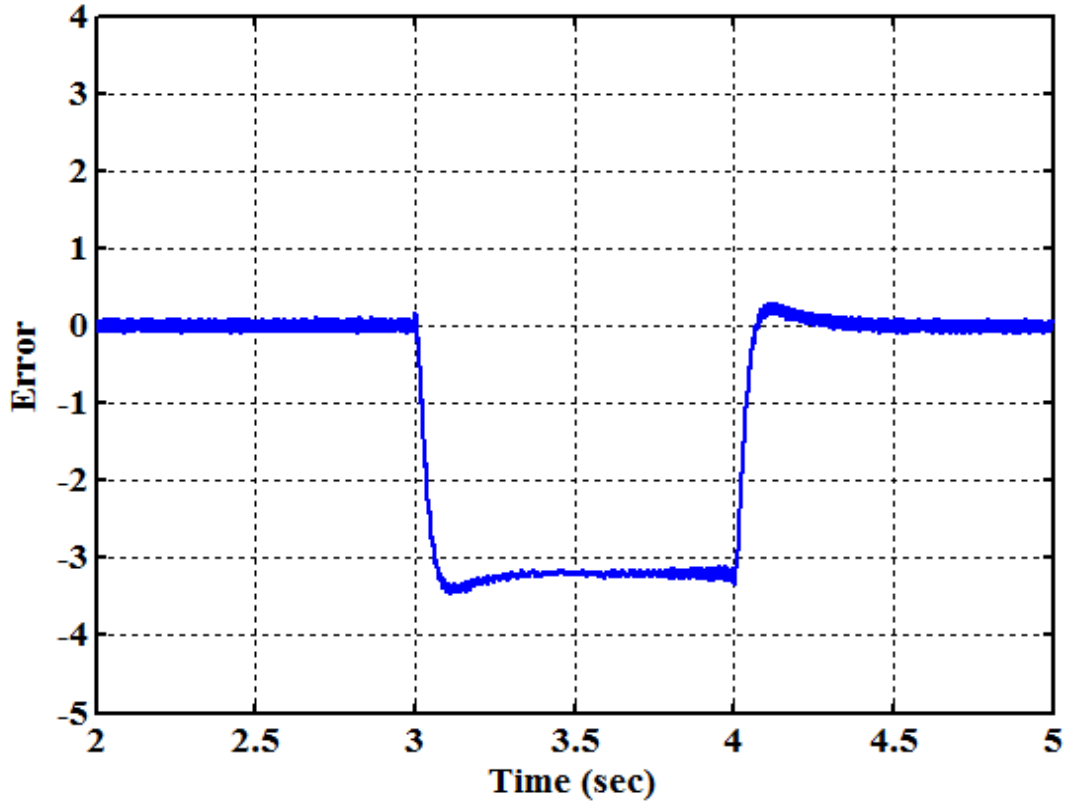


(b)

Figure 2.6 Performance Curves for Test Condition-3: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated

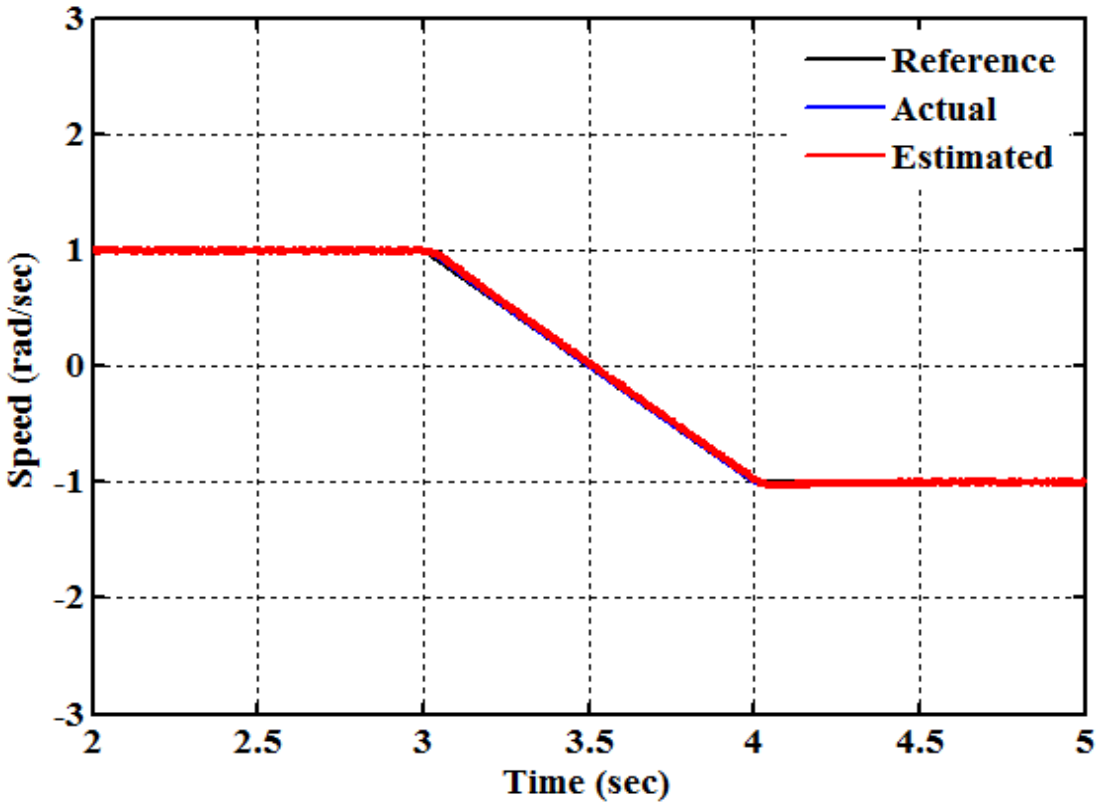


(a)

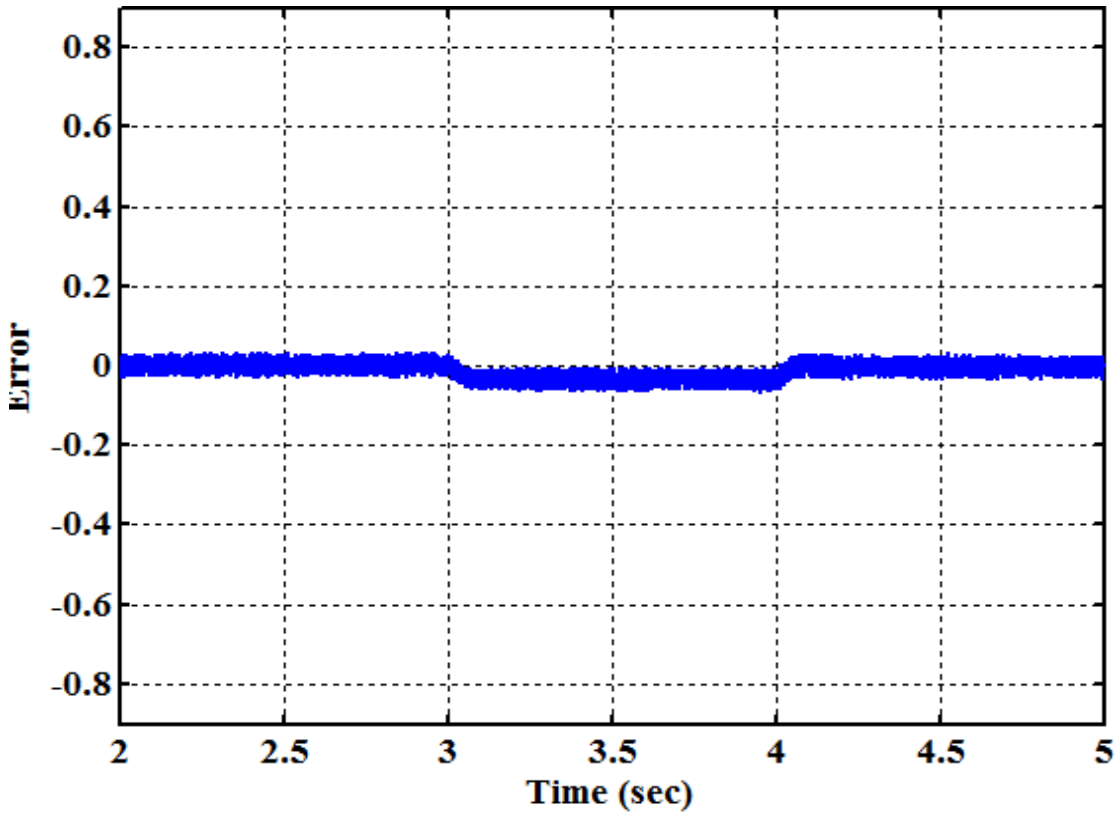


(b)

Figure 2.7 Performance Curves for Test Condition-4: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated

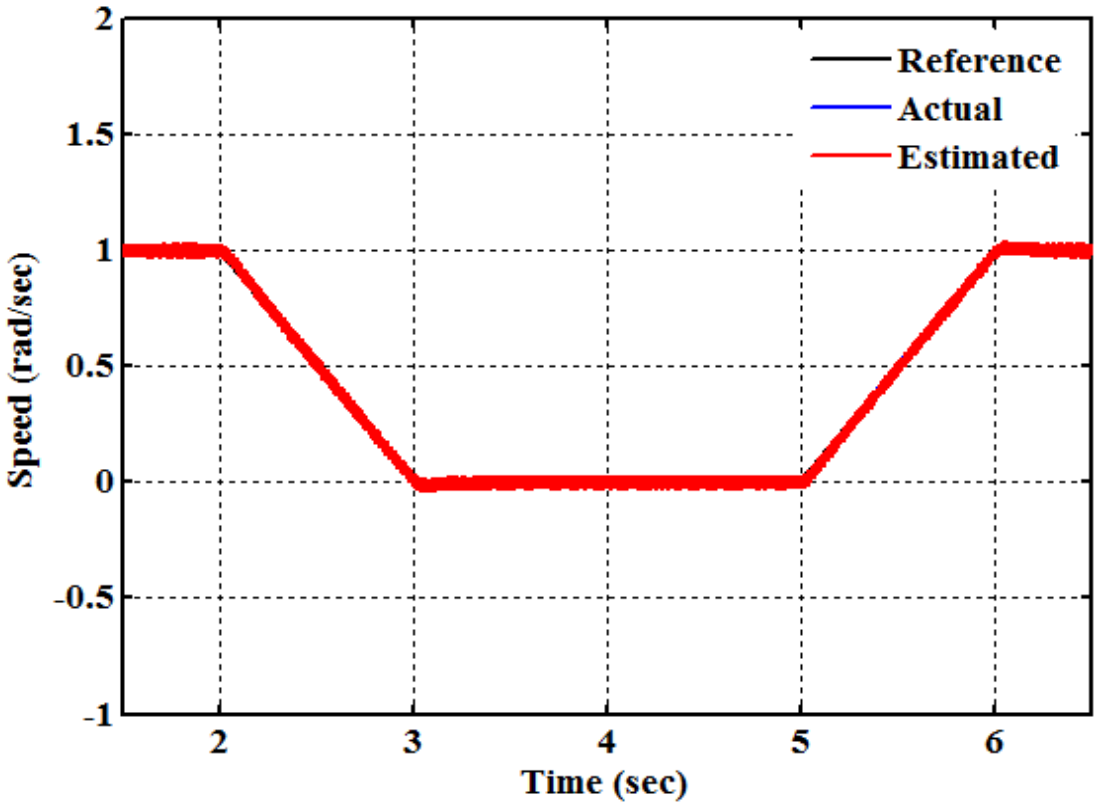


(a)

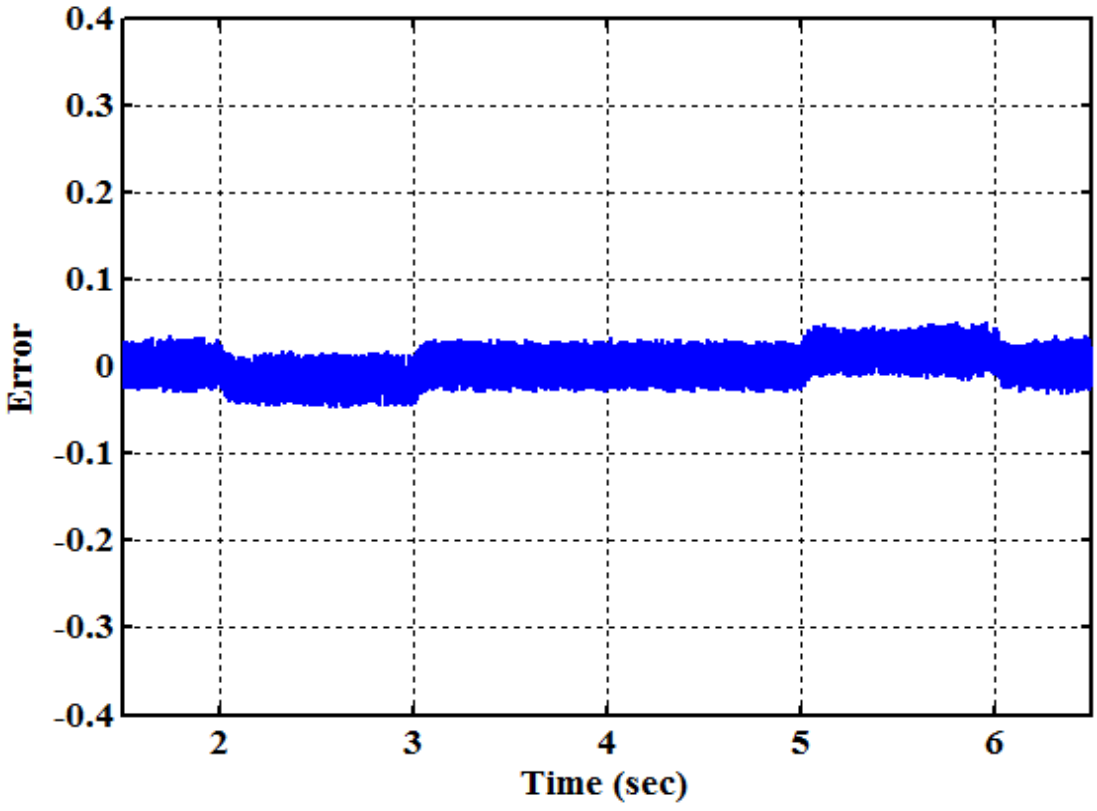


(b)

Figure 2.8 Performance Curves for Test Condition-5: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated

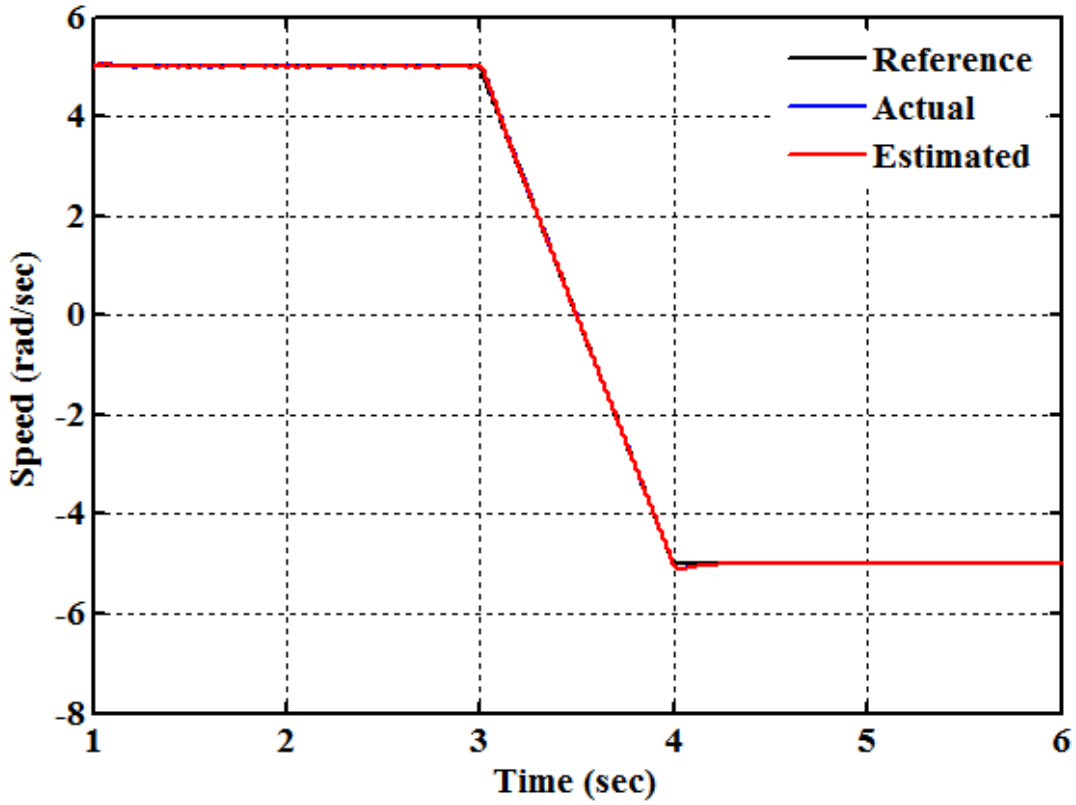


(a)

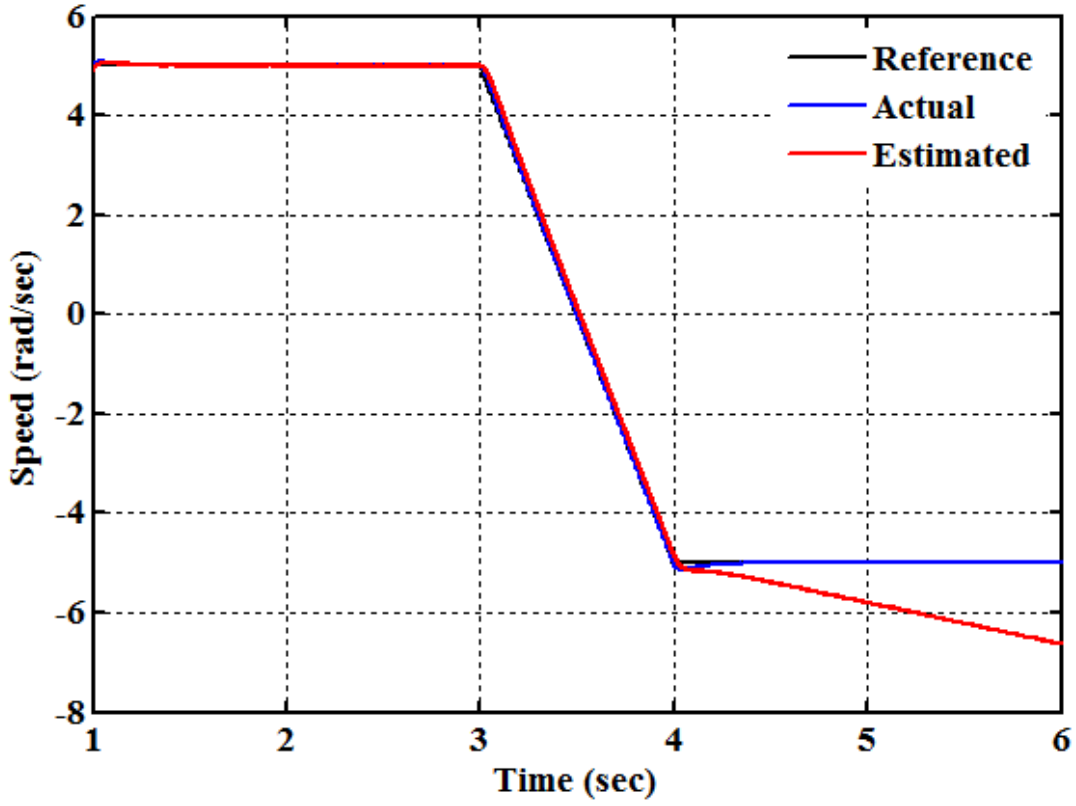


(b)

Figure 2.9 Performance Curves for Test Condition-6: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated



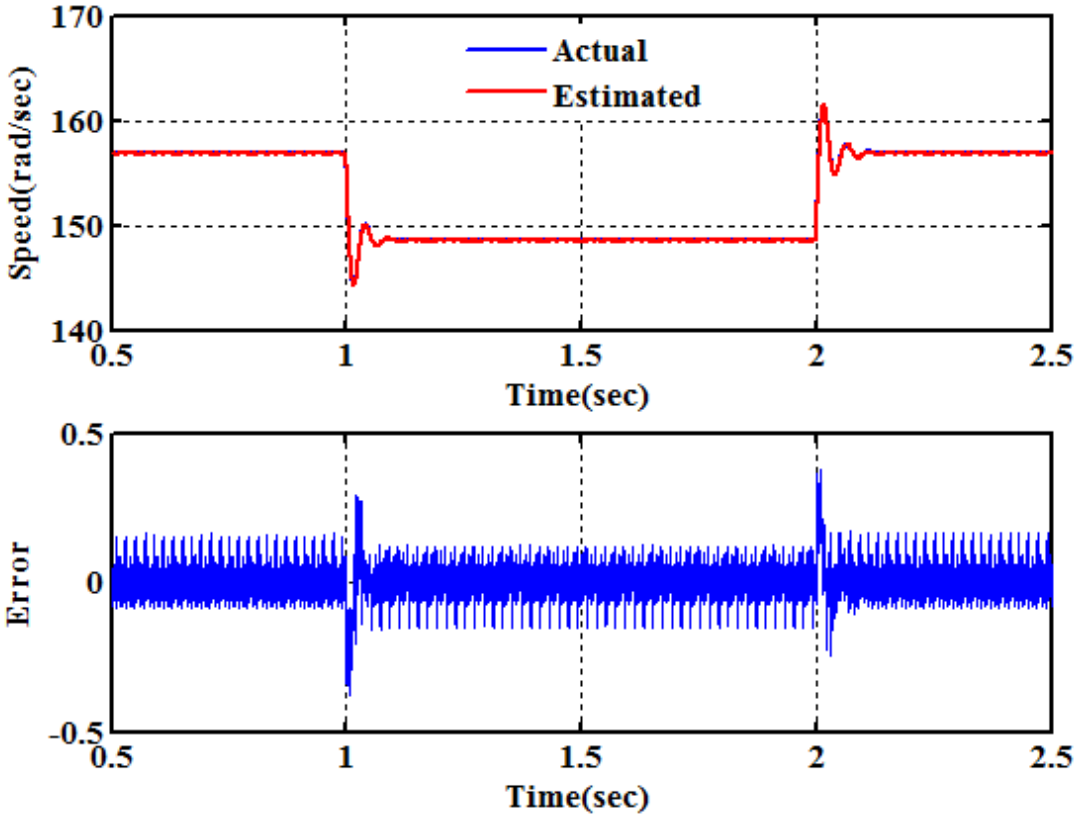
(a)



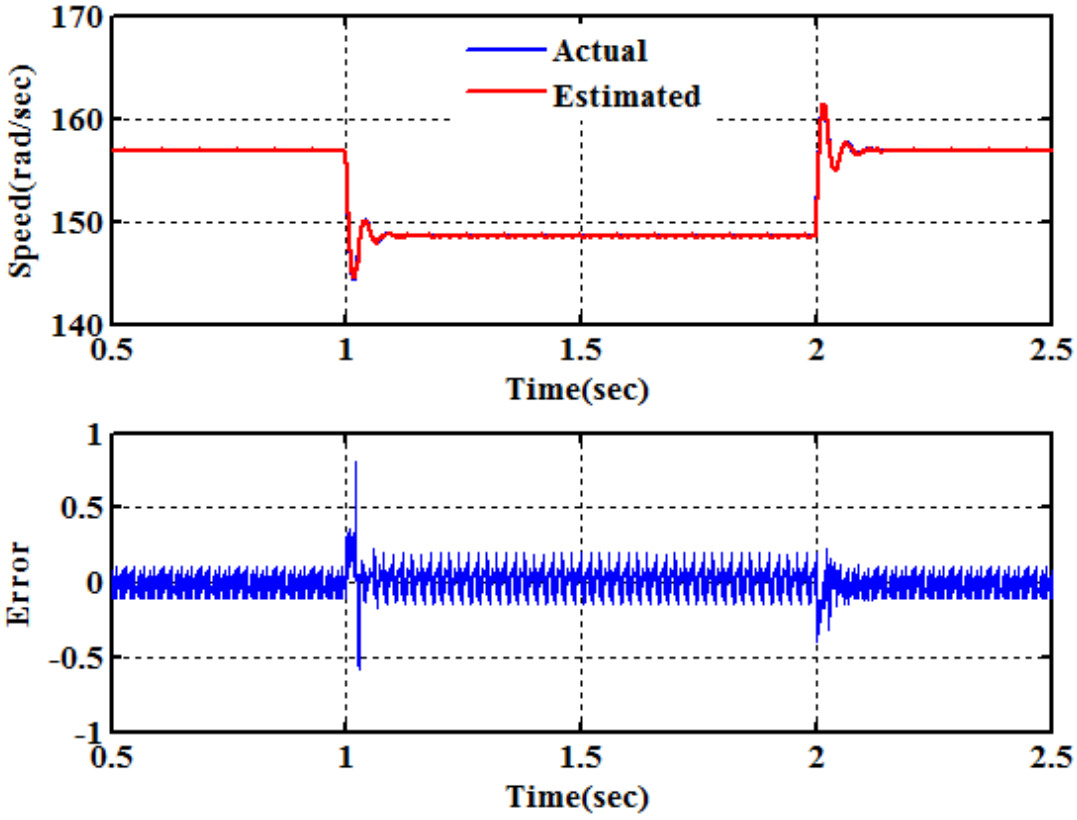
(b)

Figure 2.10 Response of Speed Estimator for Regenerating Mode:

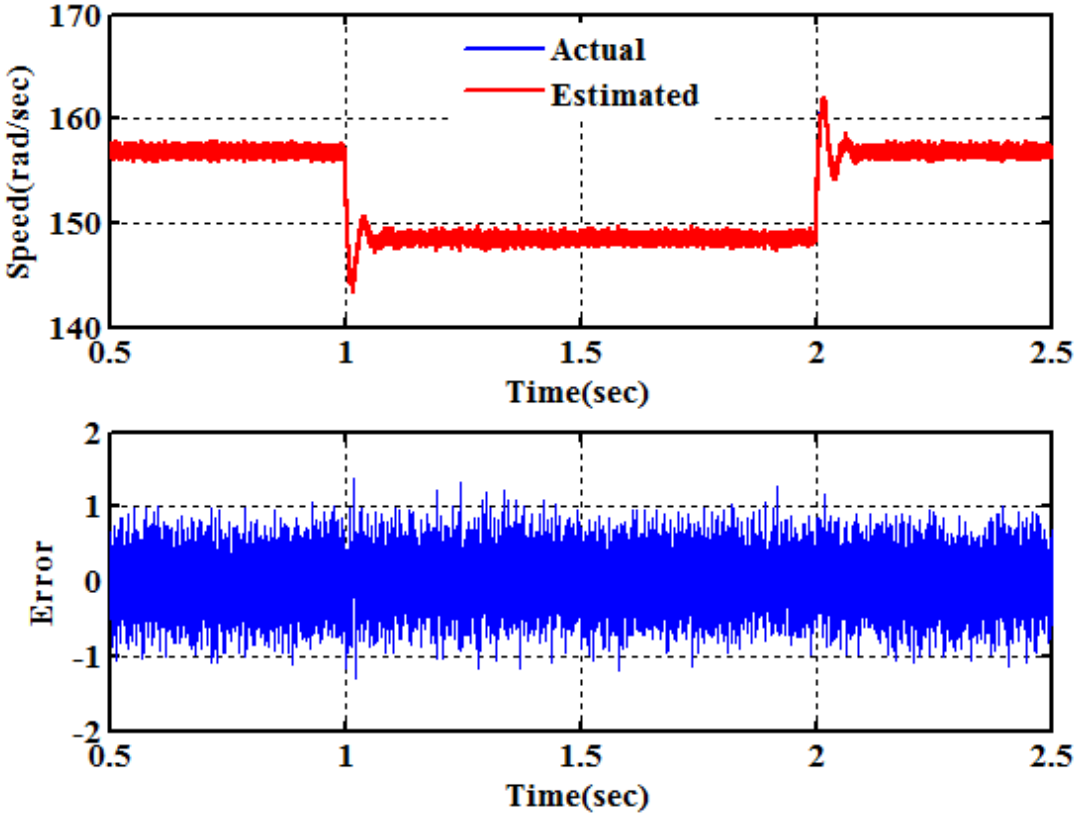
(a) Q-MRNLAS (b) Q-MRAS


Actual
Estimated
Speed(rad/sec)
170
160
150
140
0.5
1
1.5
2
2.5
Time(sec)
Error
0.5
0
-0.5
0.5
1
1.5
2
2.5
Time(sec)

(a)


Actual
Estimated
Speed(rad/sec)
170
160
150
140
0.5
1
1.5
2
2.5
Time(sec)
Error
1
0.5
0
-0.5
-1
0.5
1
1.5
2
2.5
Time(sec)

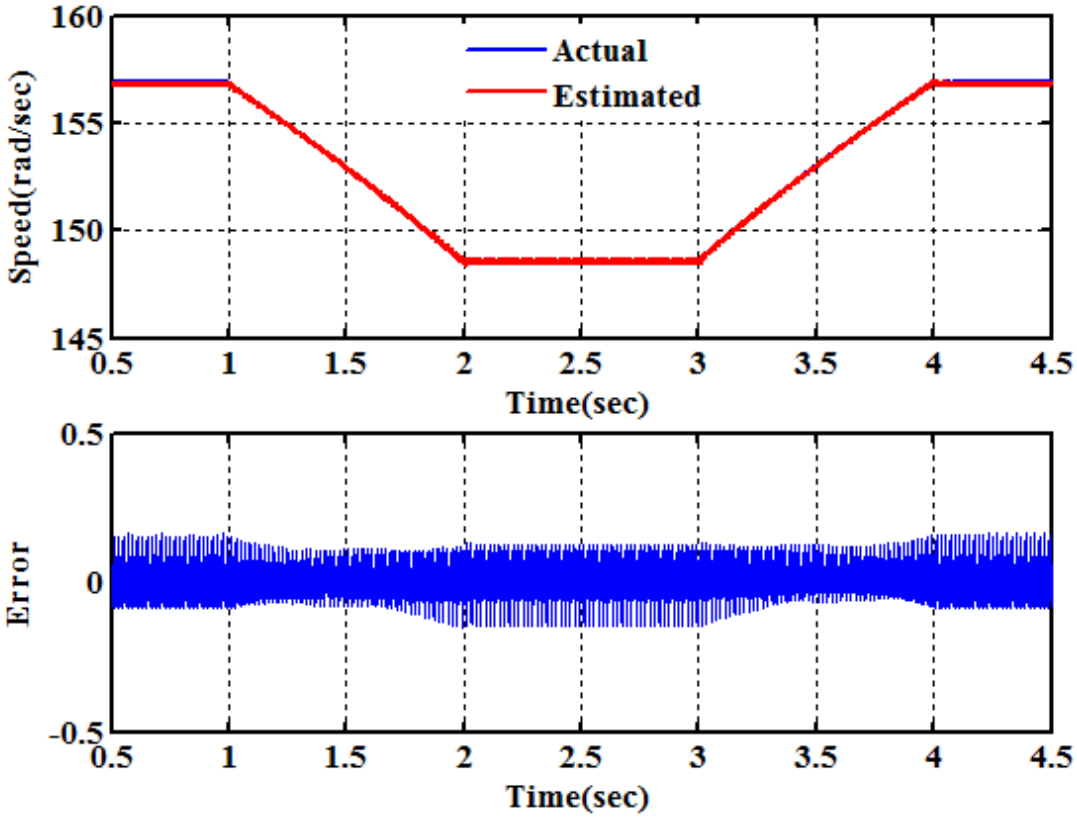
(b)



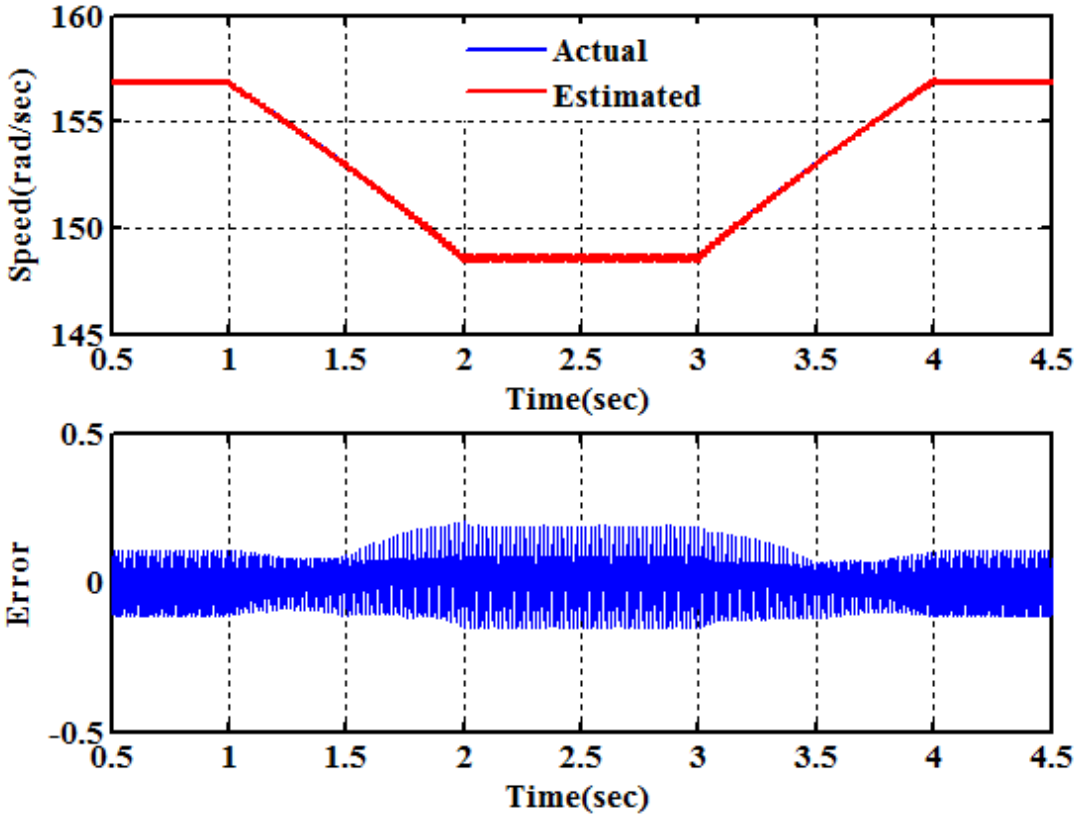
(c)

Figure 3.4 Operating Condition-I for Step Change in Load:

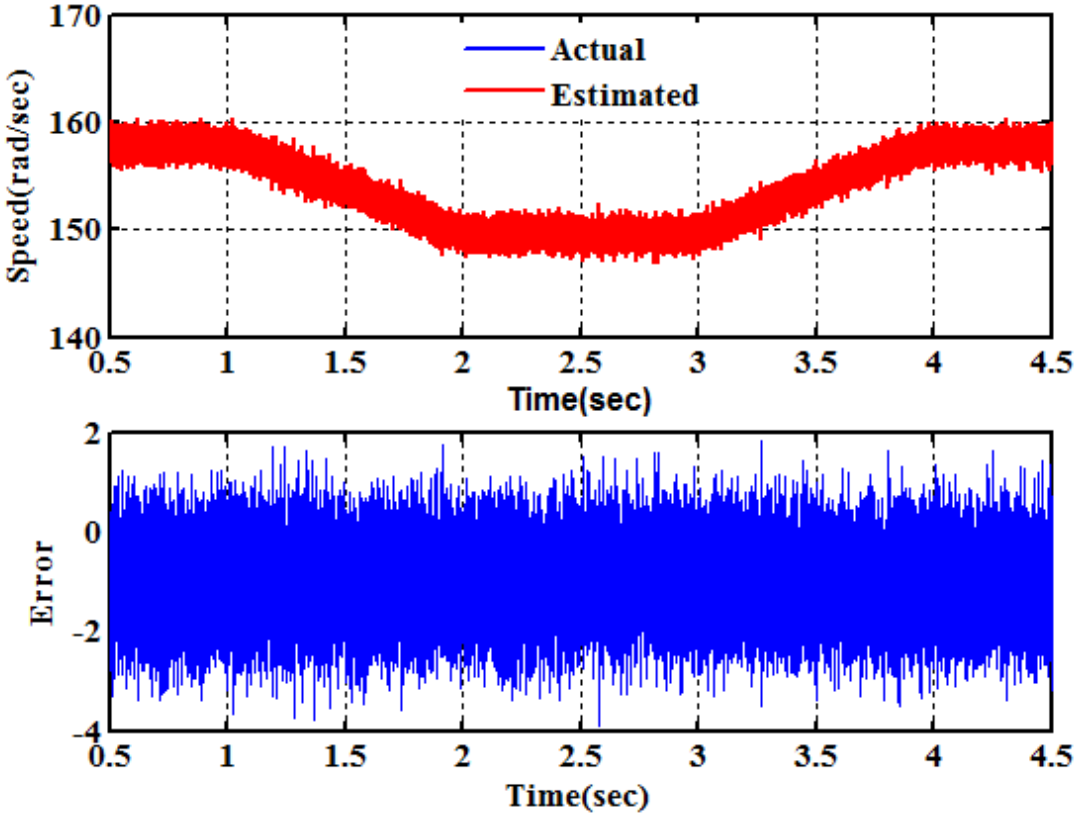
(a) SNC-NN (b) MLFF-NN (c) SLFF-NN


Actual
Estimated
Speed(rad/sec)
160
155
150
145
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time(sec)
Error
0.5
0
-0.5
Time(sec)

(a)


Actual
Estimated
Speed(rad/sec)
160
155
150
145
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time(sec)
Error
0.5
0
-0.5
Time(sec)

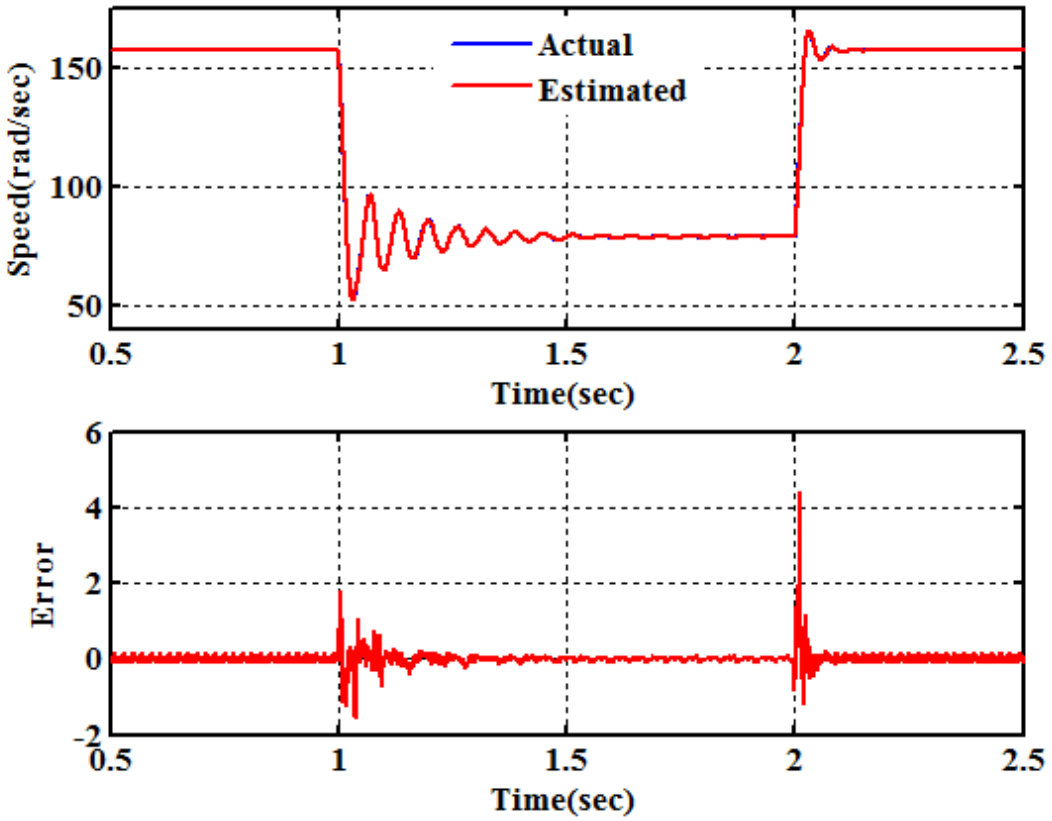
(b)



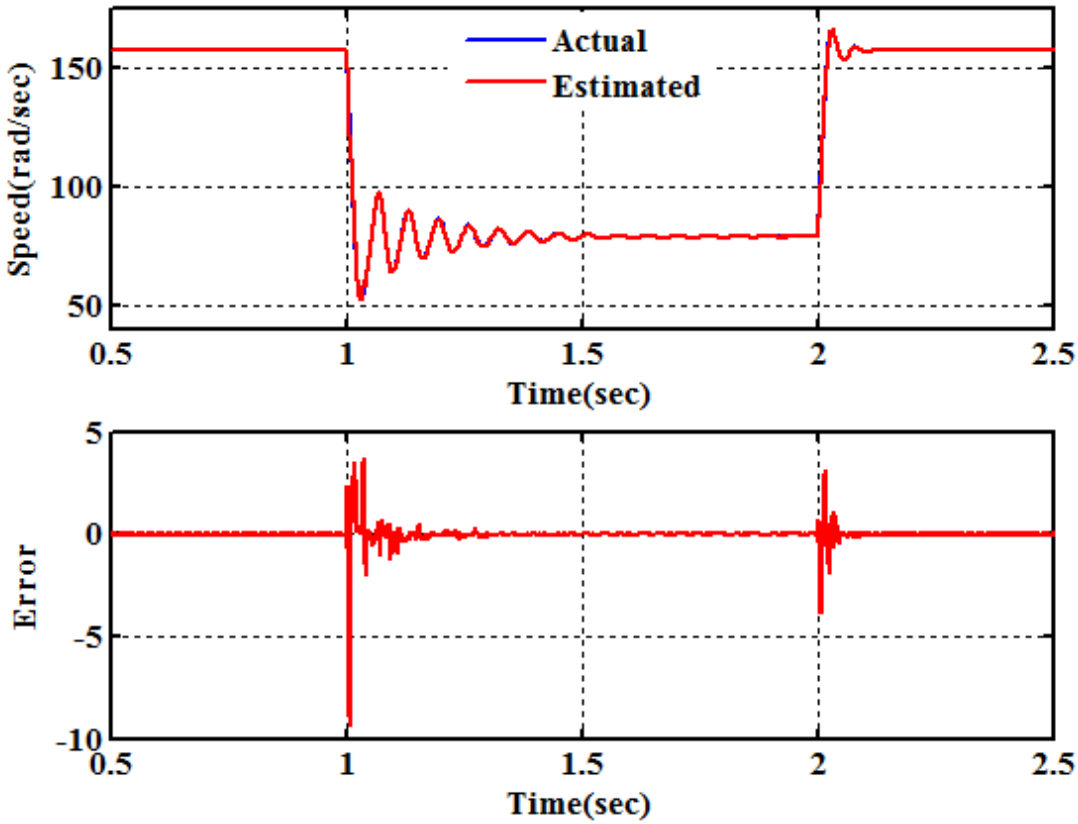
(c)

Figure 3.5 Operating Condition-I for Ramp Change in Load:

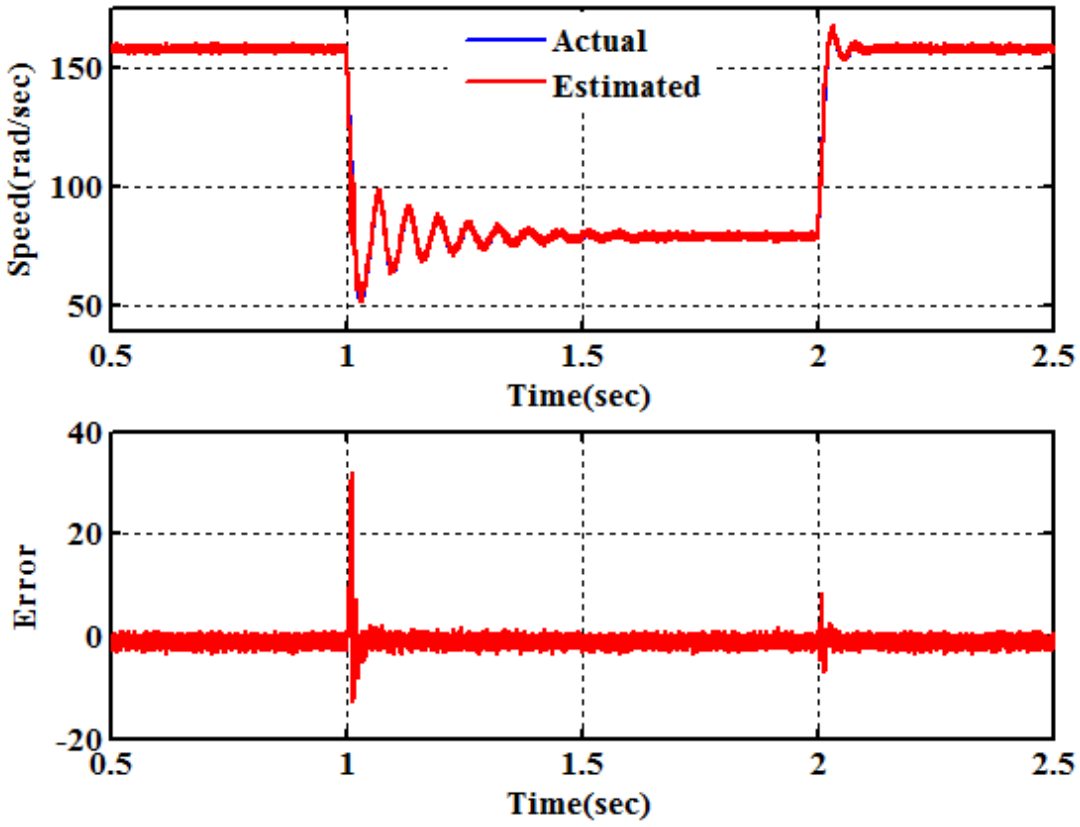
(a) SNC-NN (b) MLFF-NN (c) SLFF-NN


Actual
Estimated
Speed(rad/sec)
150
100
50
0.5
1
1.5
2
2.5
Time(sec)
Error
6
4
2
0
-2
0.5
1
1.5
2
2.5
Time(sec)

(a)


Actual
Estimated
Speed(rad/sec)
150
100
50
0.5
1
1.5
2
2.5
Time(sec)
Error
5
0
-5
-10
0.5
1
1.5
2
2.5
Time(sec)

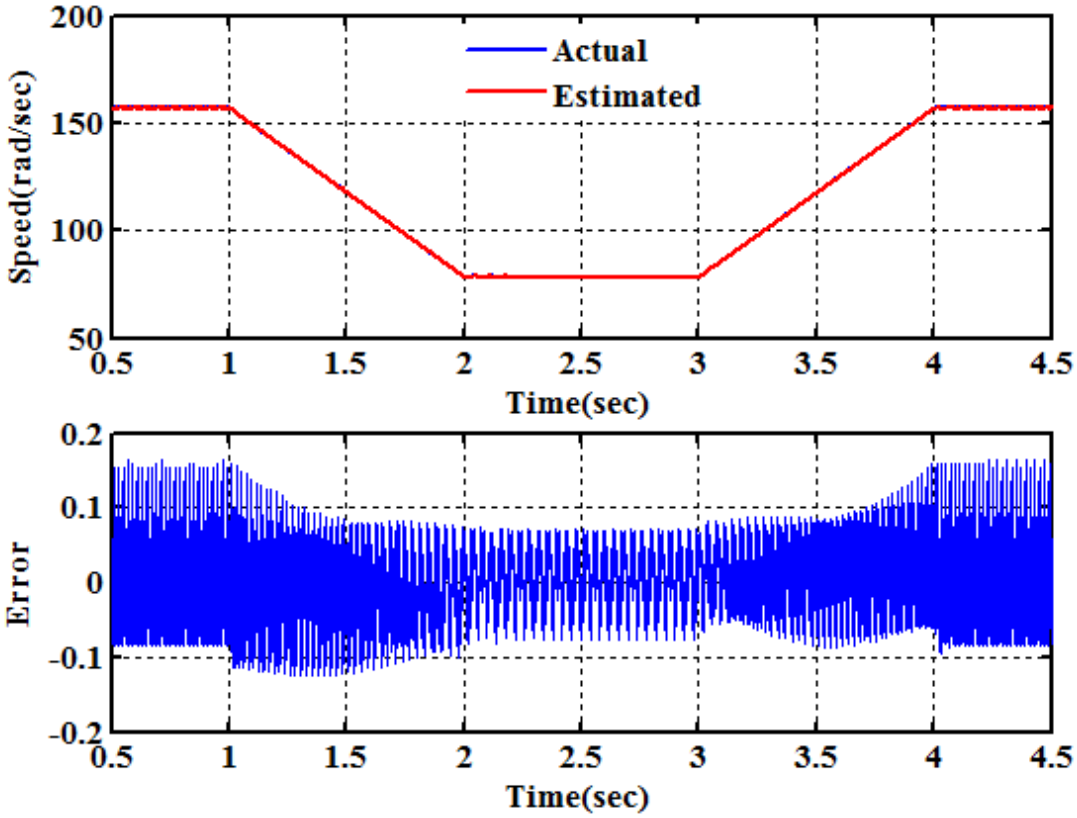
(b)



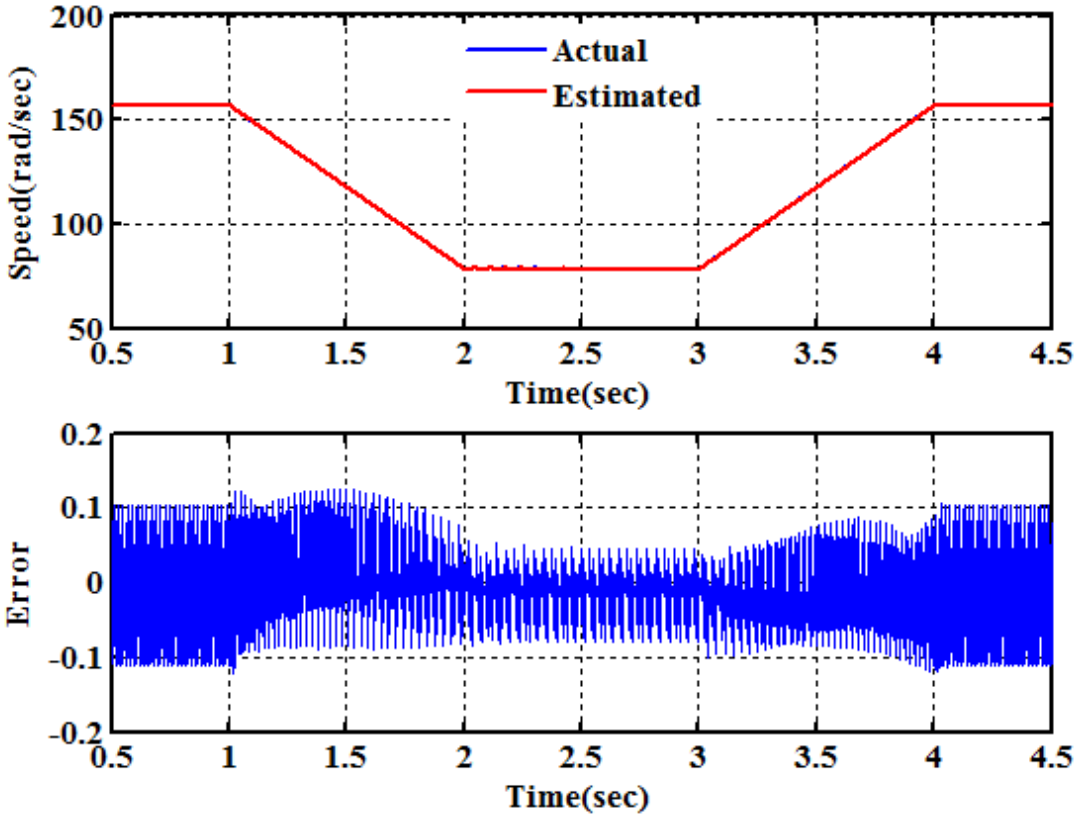
(c)

Figure 3.6 Operating Condition-II for Step Change in Speed:

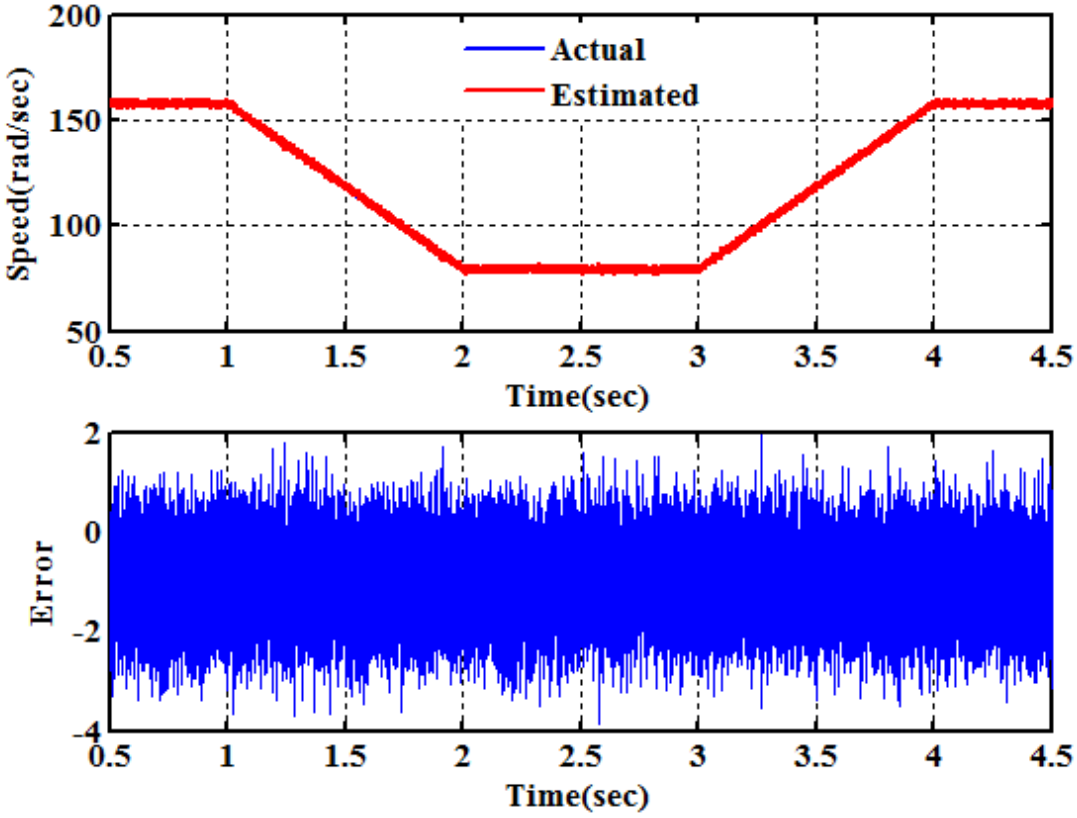
(a) SNC-NN (b) MLFF-NN (c) SLFF-NN


Actual
Estimated
Speed(rad/sec)
200
150
100
50
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time(sec)
Error
0.2
0.1
0
-0.1
-0.2
Time(sec)

(a)


Actual
Estimated
Speed(rad/sec)
200
150
100
50
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time(sec)
Error
0.2
0.1
0
-0.1
-0.2
Time(sec)

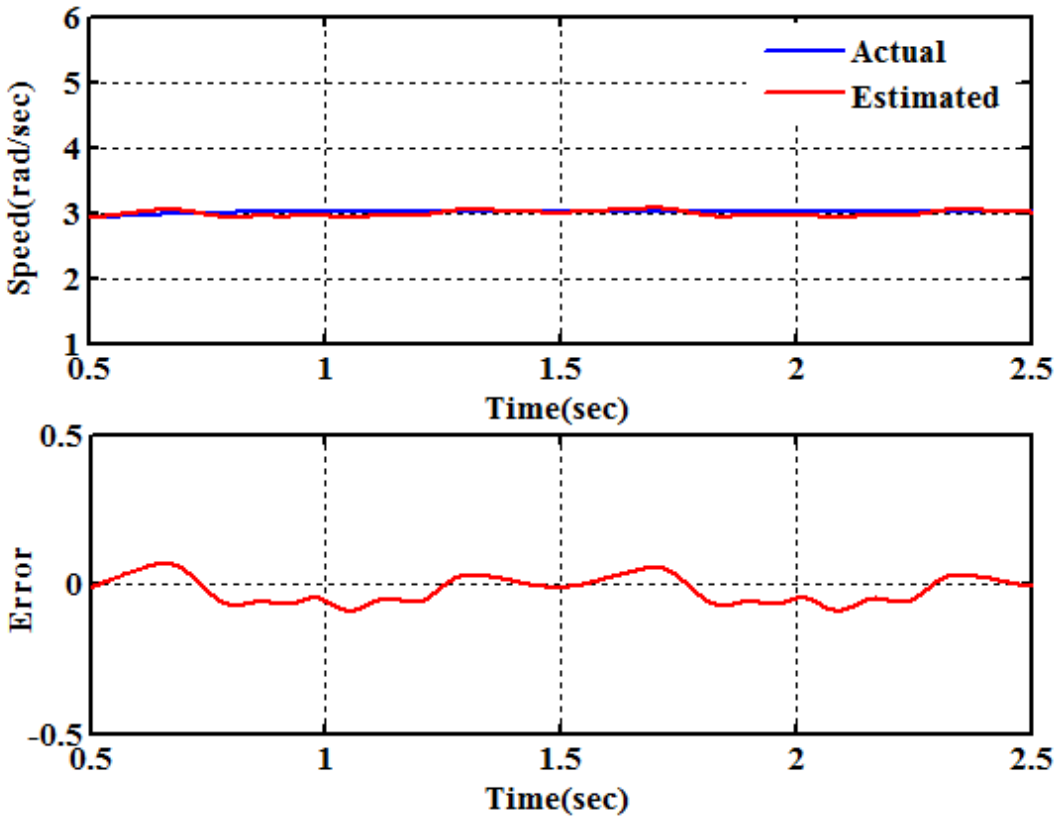
(b)



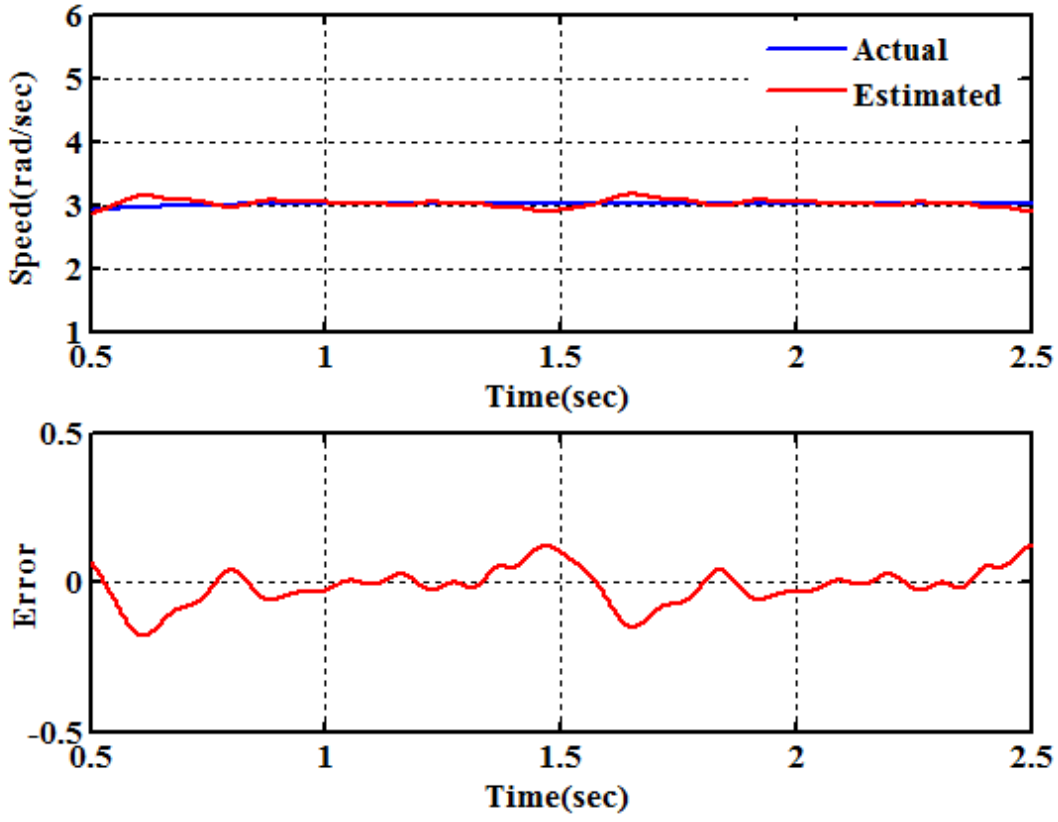
(c)

Figure 3.7 Operating Condition-II for Ramp Change in Speed:

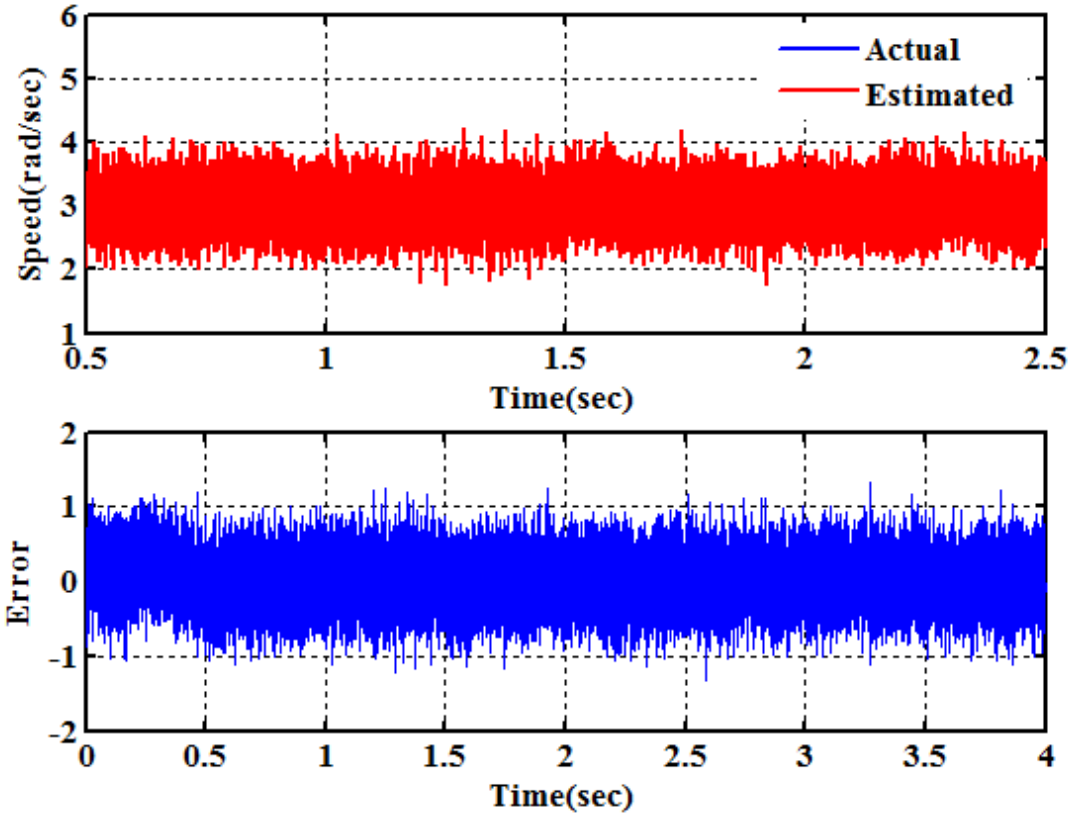
(a) SNC-NN (b) MLFF-NN (c) SLFF-NN


Actual
Estimated
Speed(rad/sec)
Time(sec)
Error
Time(sec)

(a)


Actual
Estimated
Speed(rad/sec)
Time(sec)
Error
Time(sec)

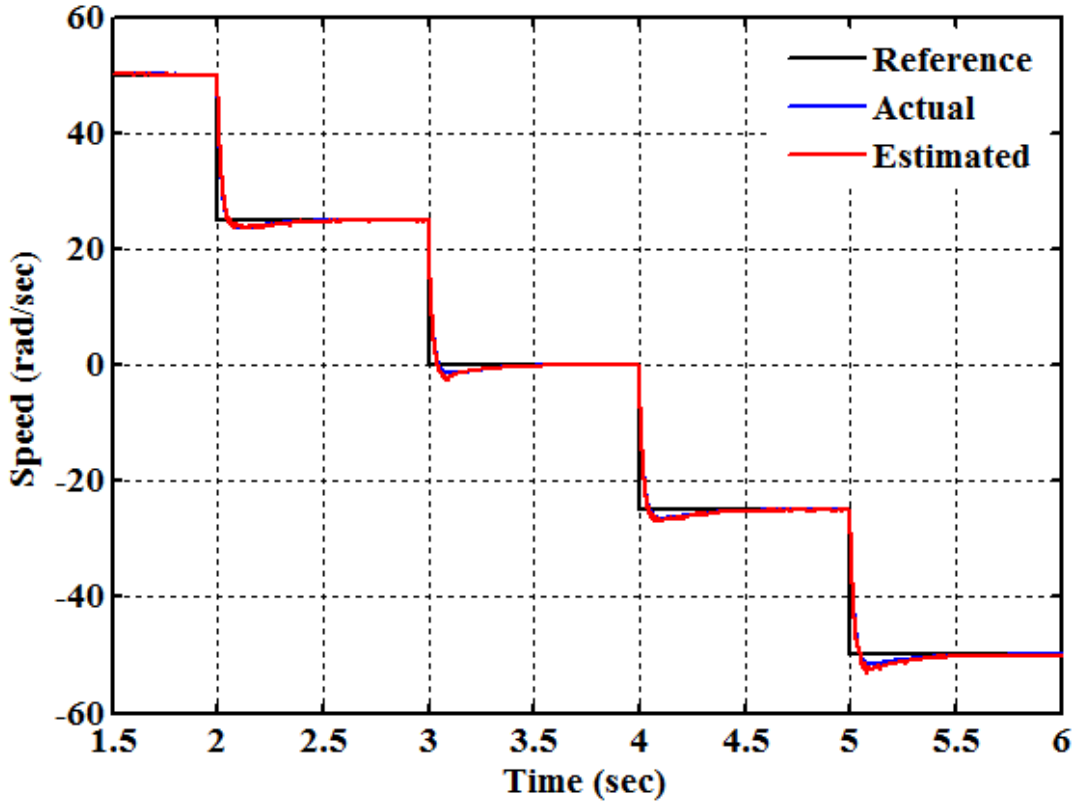
(b)



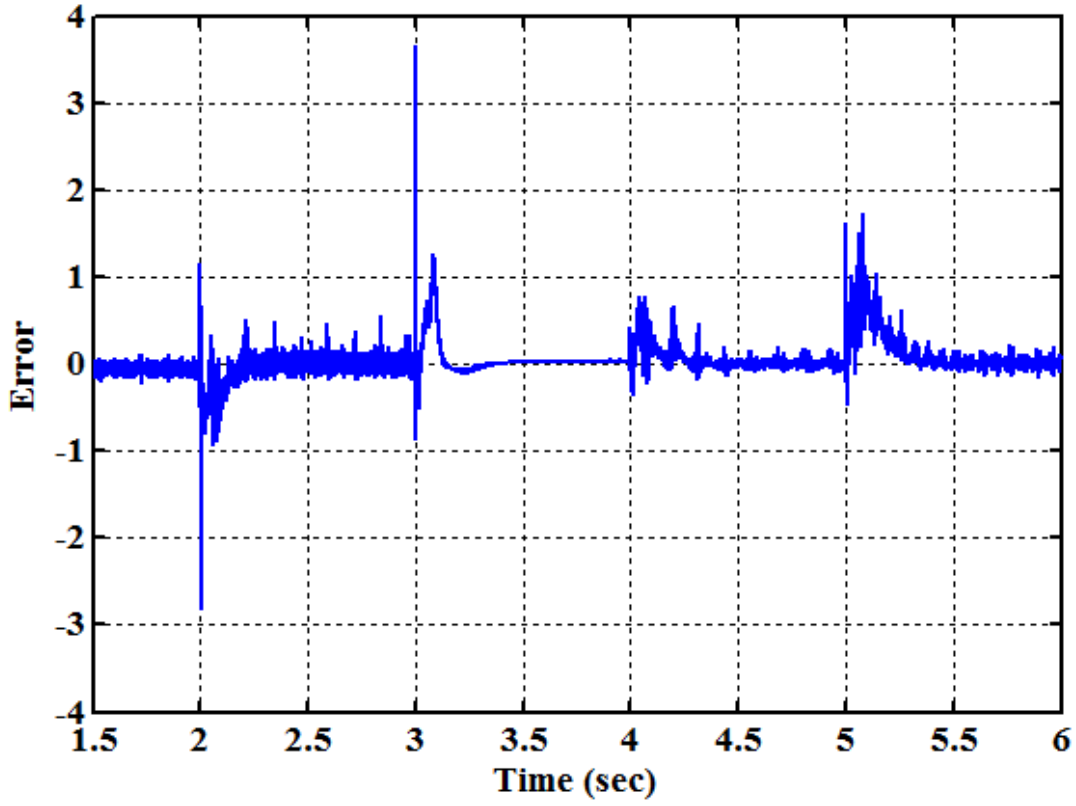
(c)

Figure 3.8 Operating Condition-III for Low Speed:

(a) SNC-NN (b) MLFF-NN (c) SLFF-NN

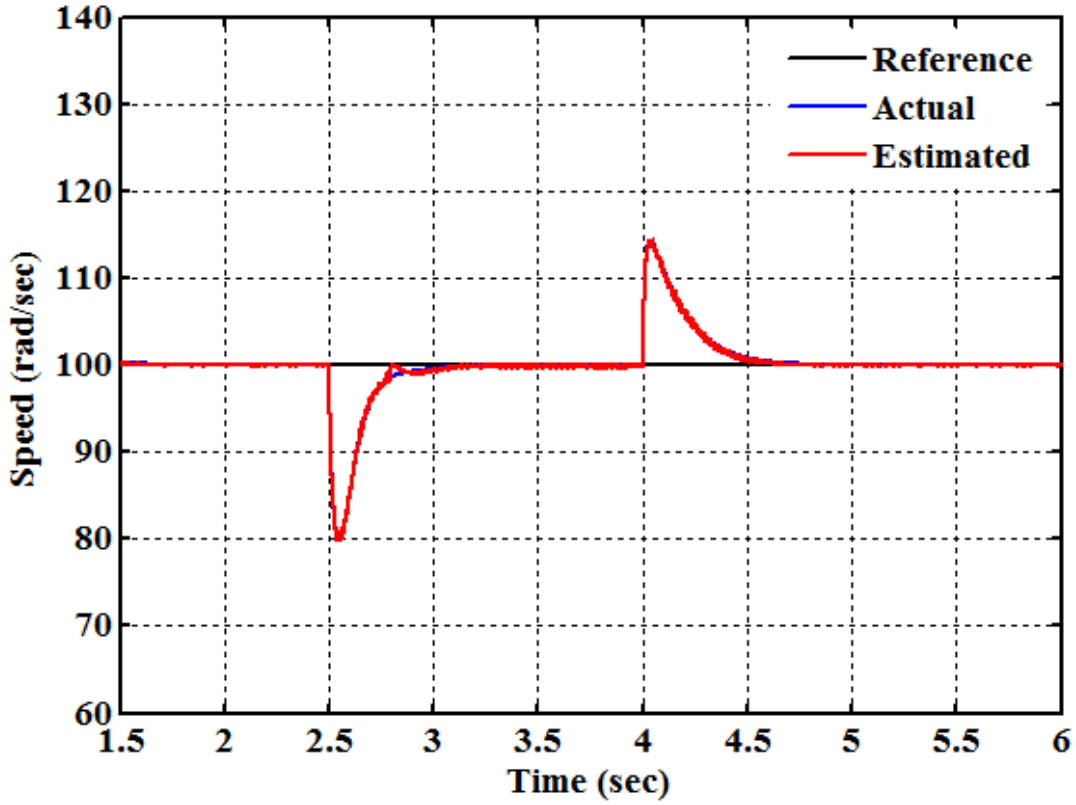


(a)

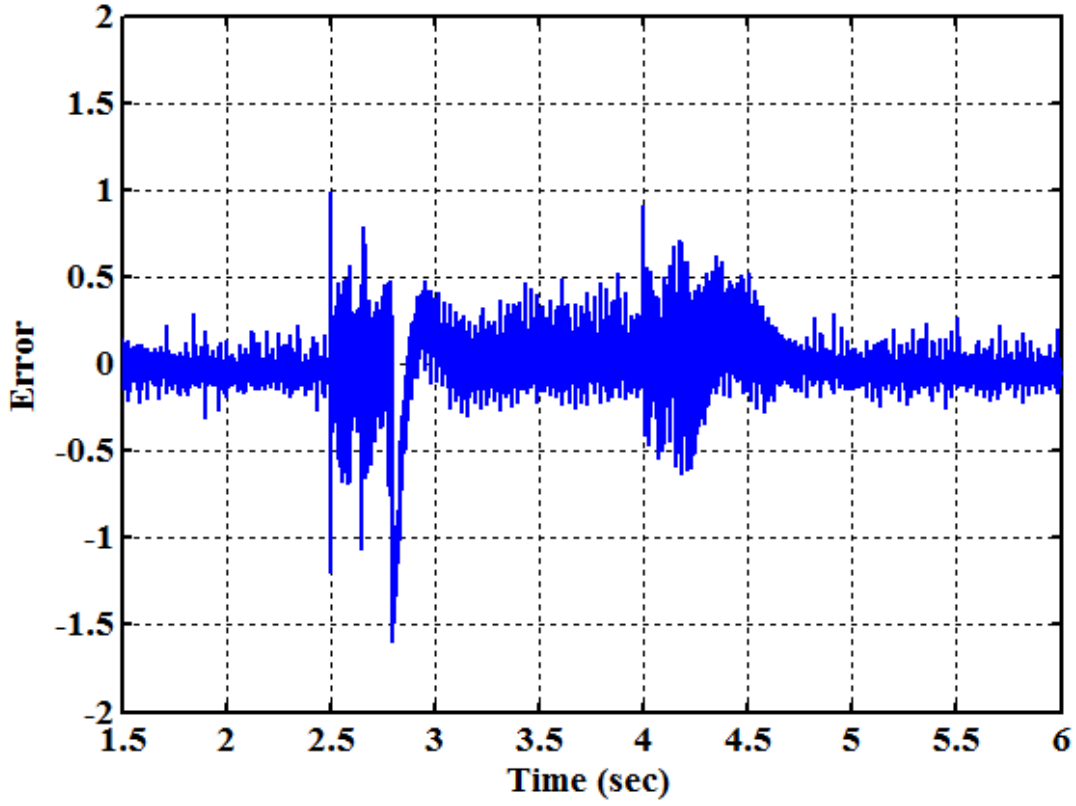


(b)

Figure 4.2 Stair Case Speed Change: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

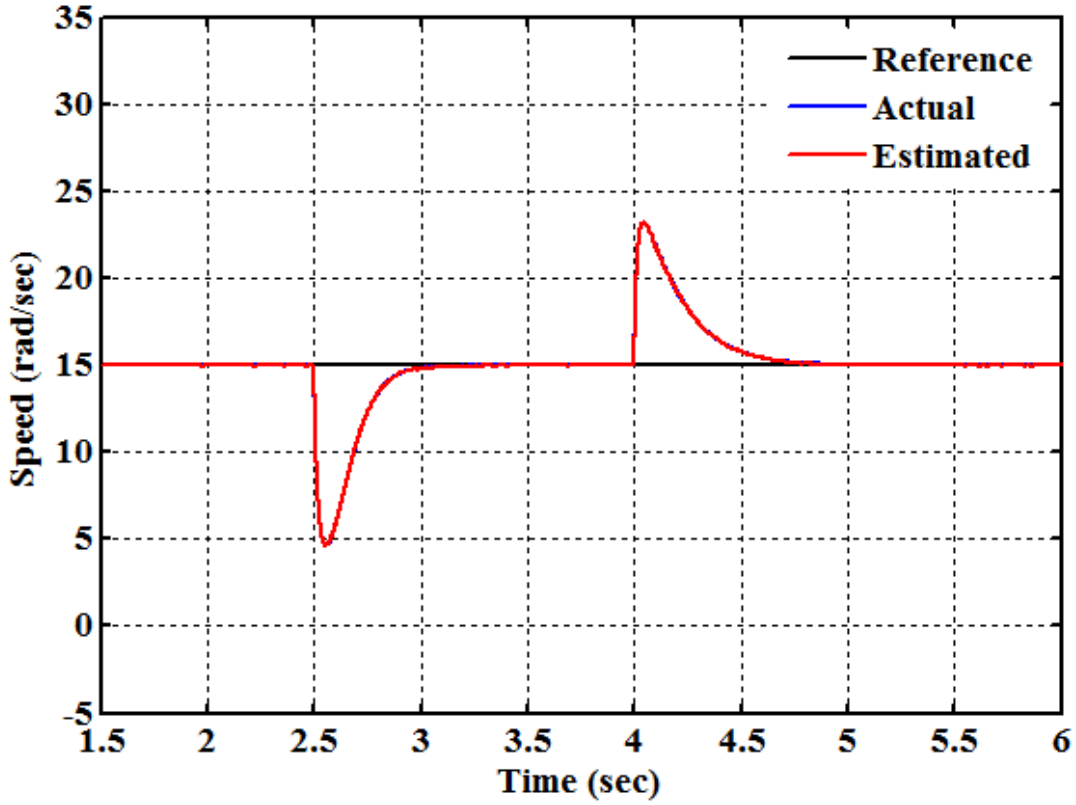


(a)

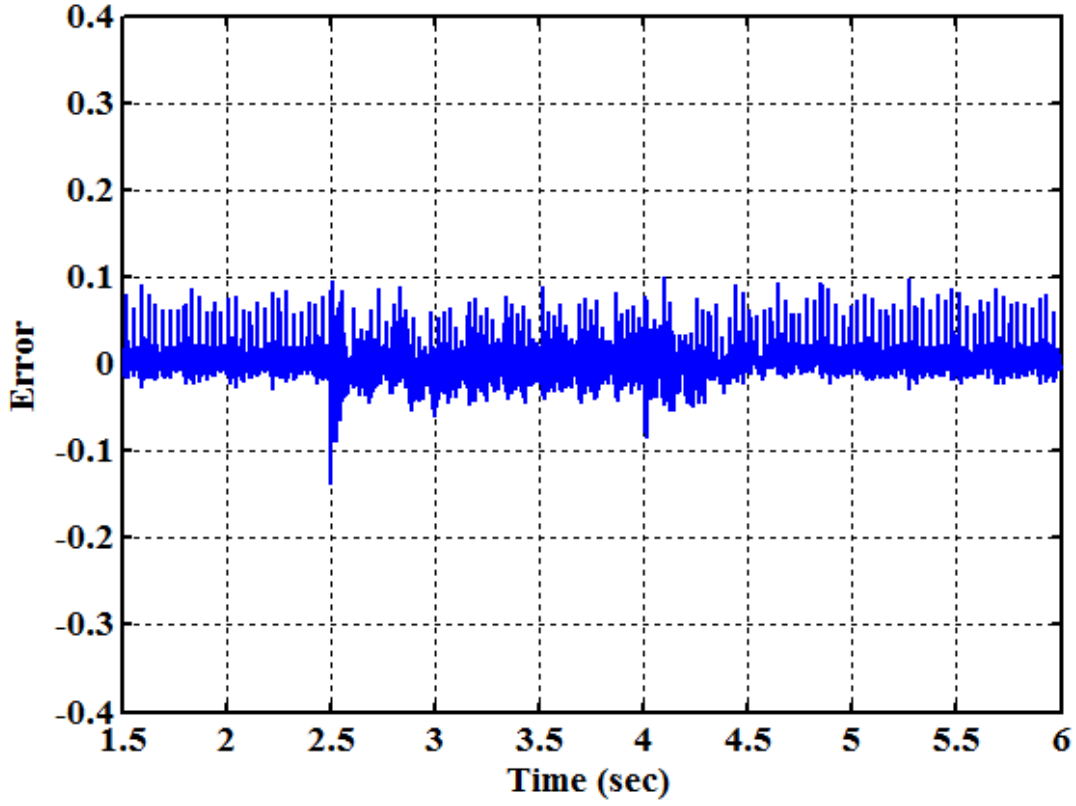


(b)

Figure 4.3 Effect of Loading at 100rad/sec: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

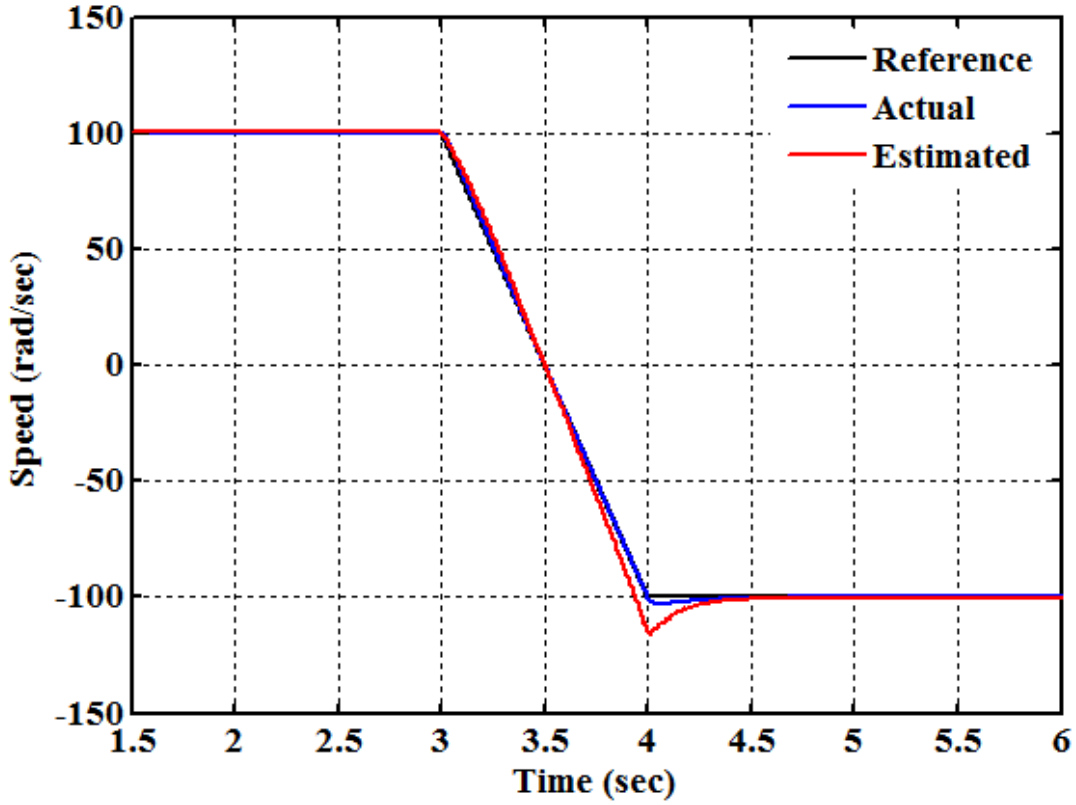


(a)

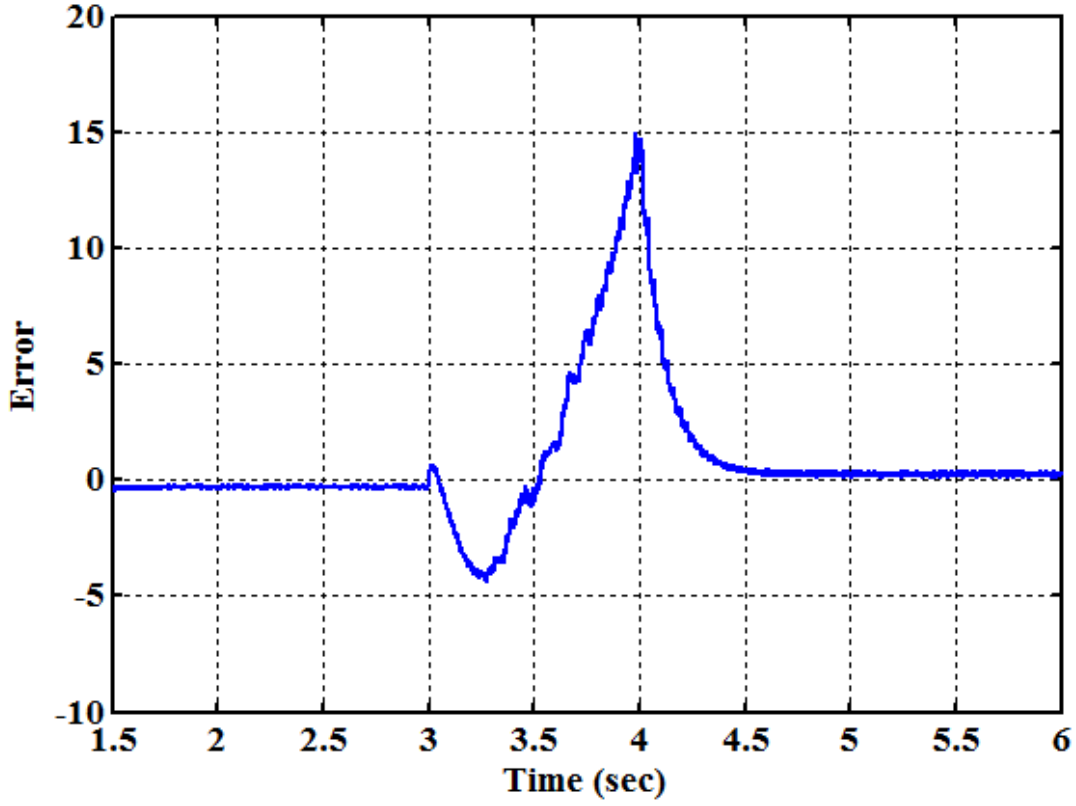


(b)

Figure 4.4 Effect of Loading at 15rad/sec: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

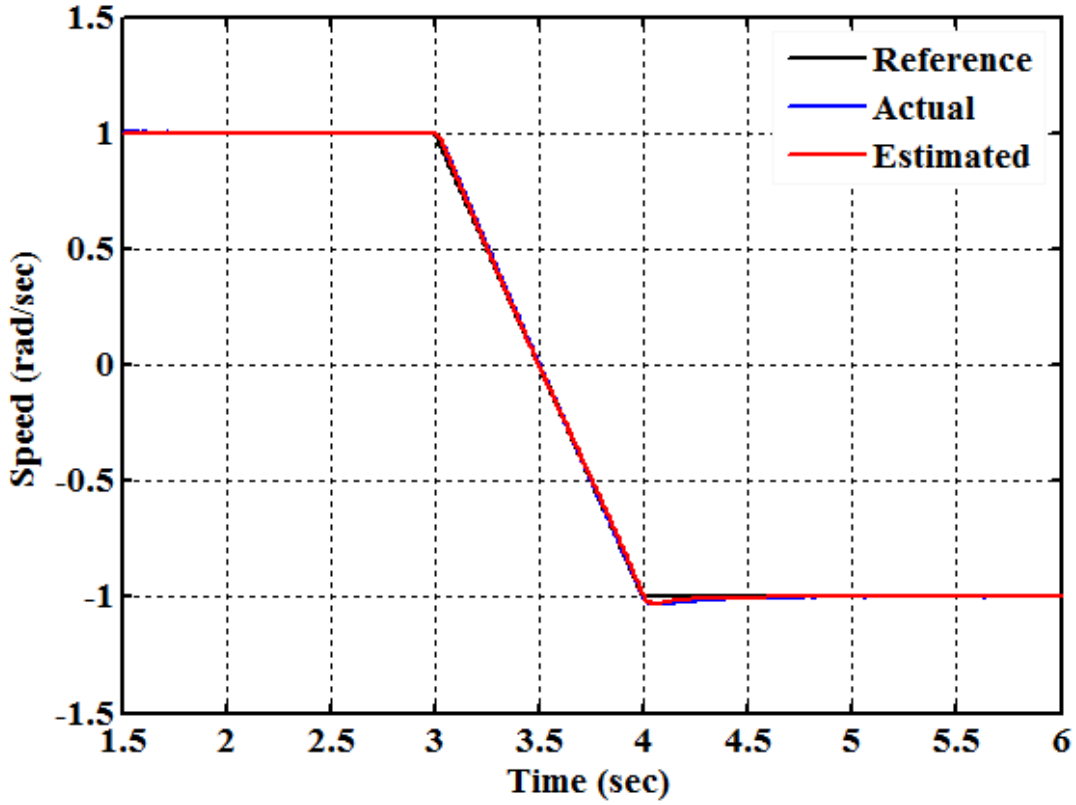


(a)

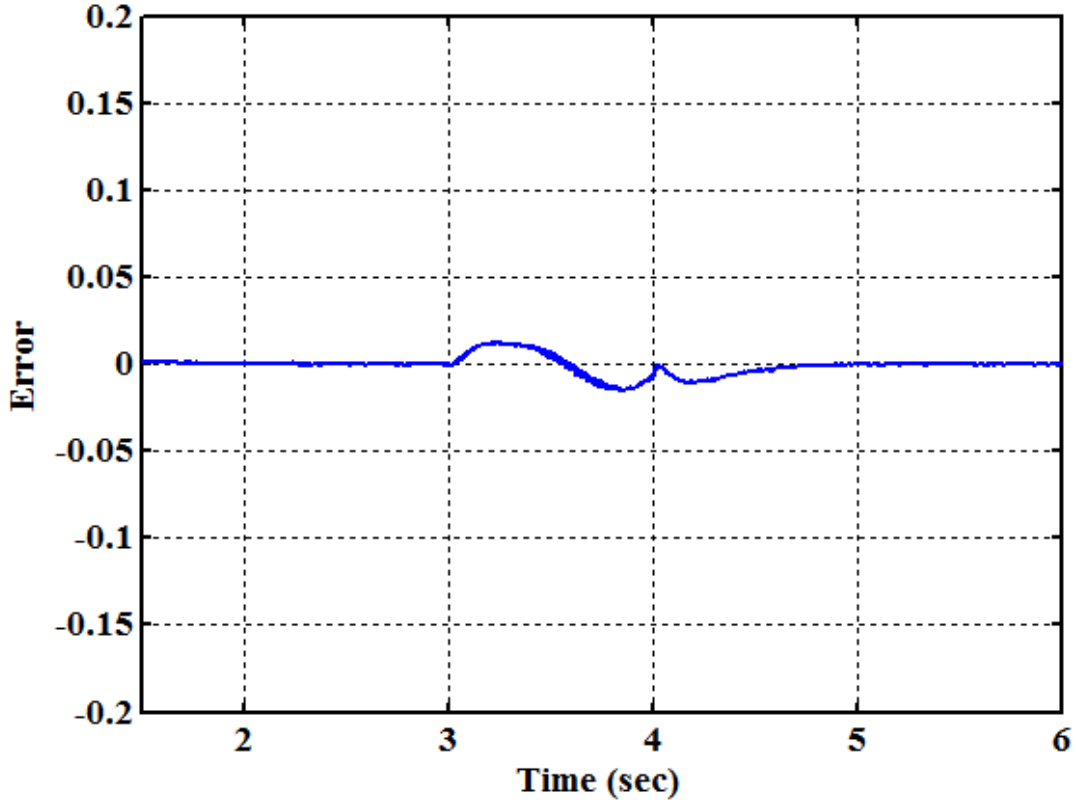


(b)

Figure 4.5 Regenerating Mode Operation (±100rad/sec): (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

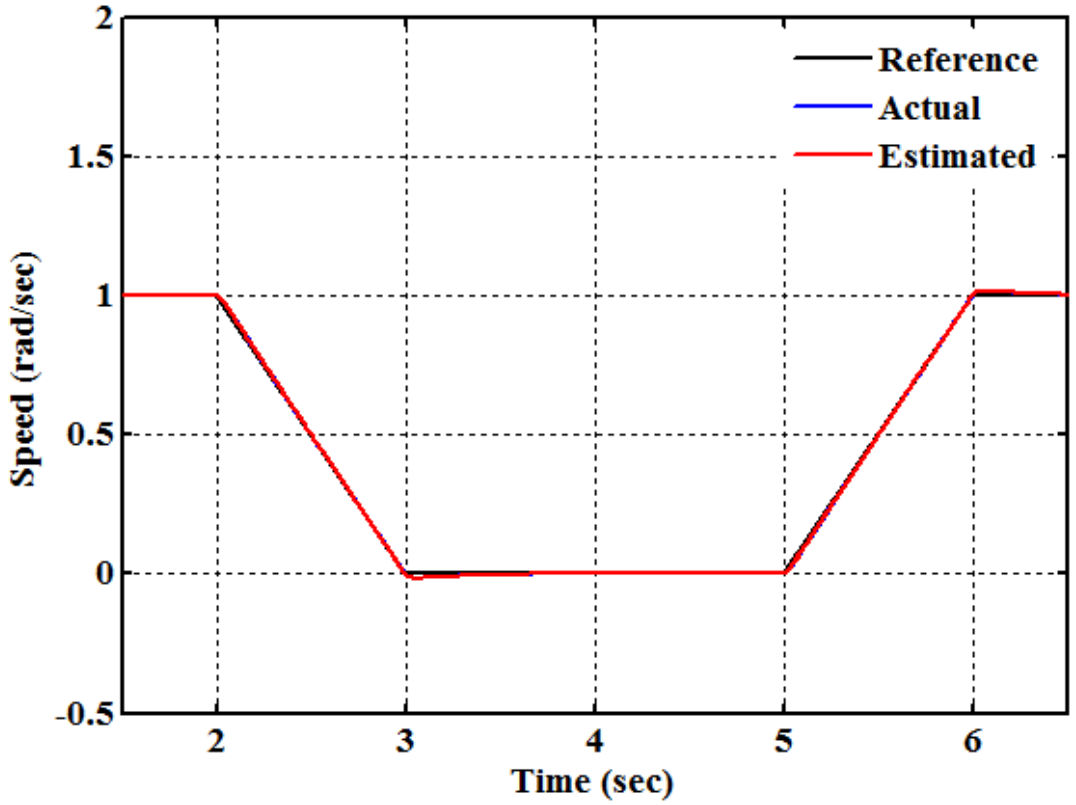


(a)

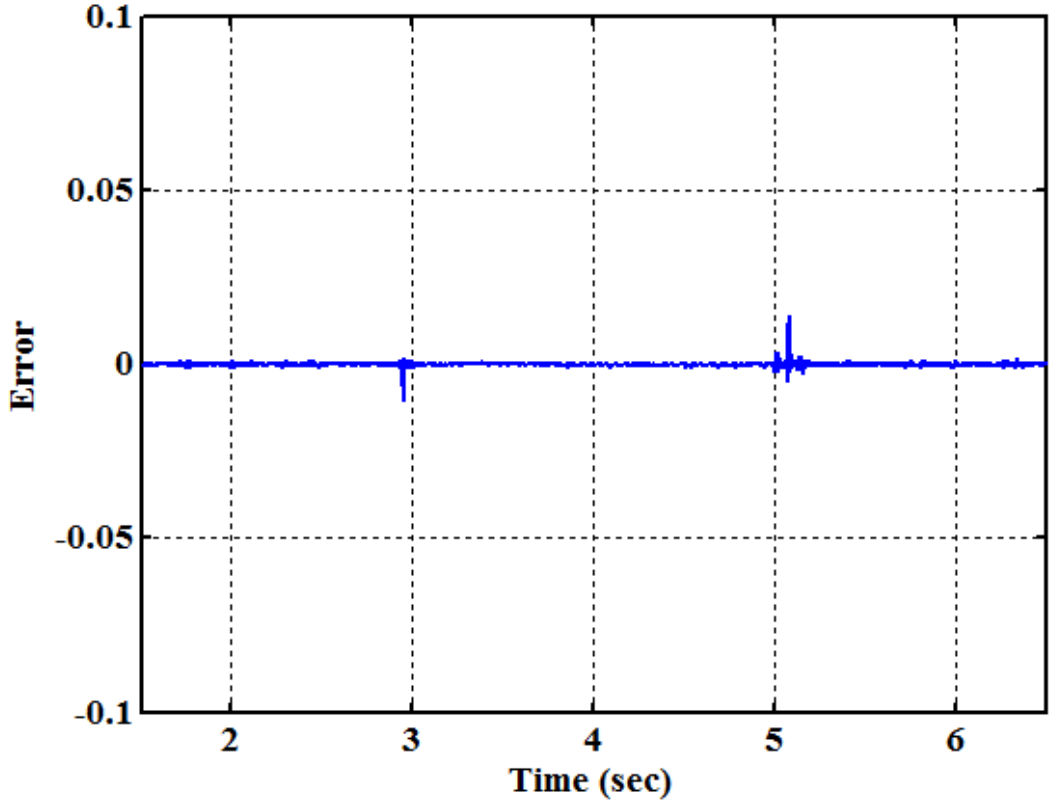


(b)

Figure 4.6 Regenerating Mode Operation (±1rad/sec): (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed



(a)



(b)

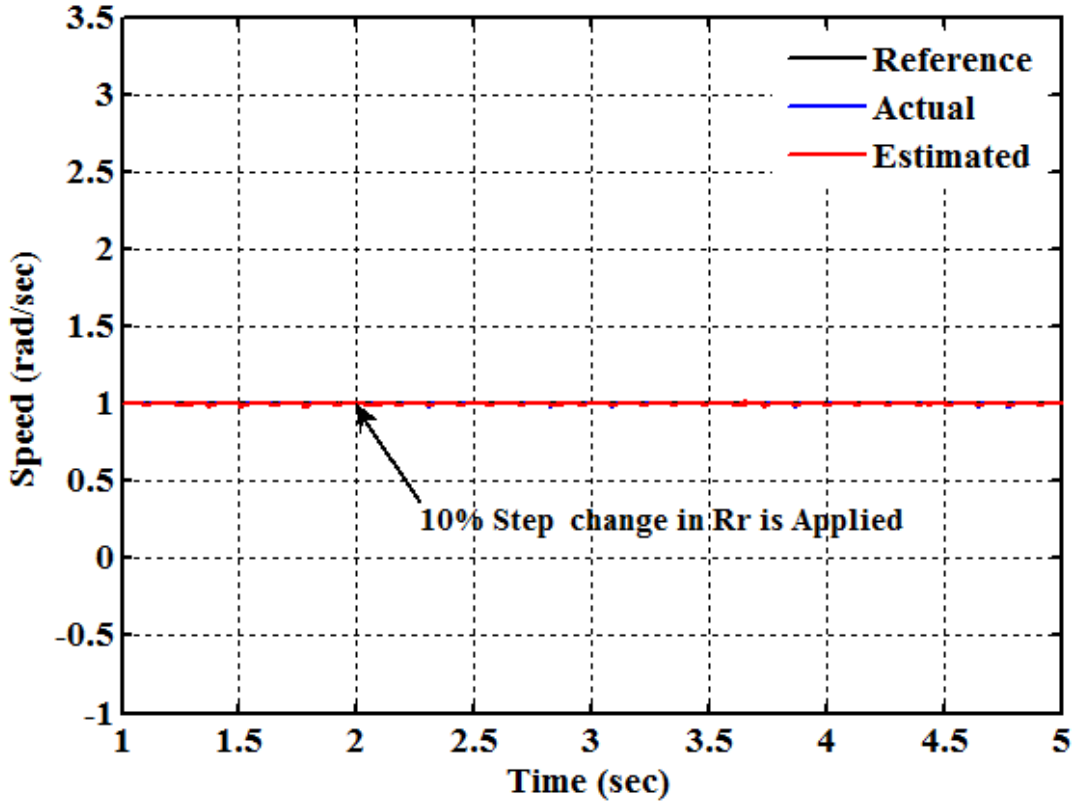
Figure 4.7 Zero Speed Operation: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

Table 4.1 Performance Comparisons of the Proposed Data Driven SNC-NN with Q-MRNLAS for Various Speed Commands under No Load Condition

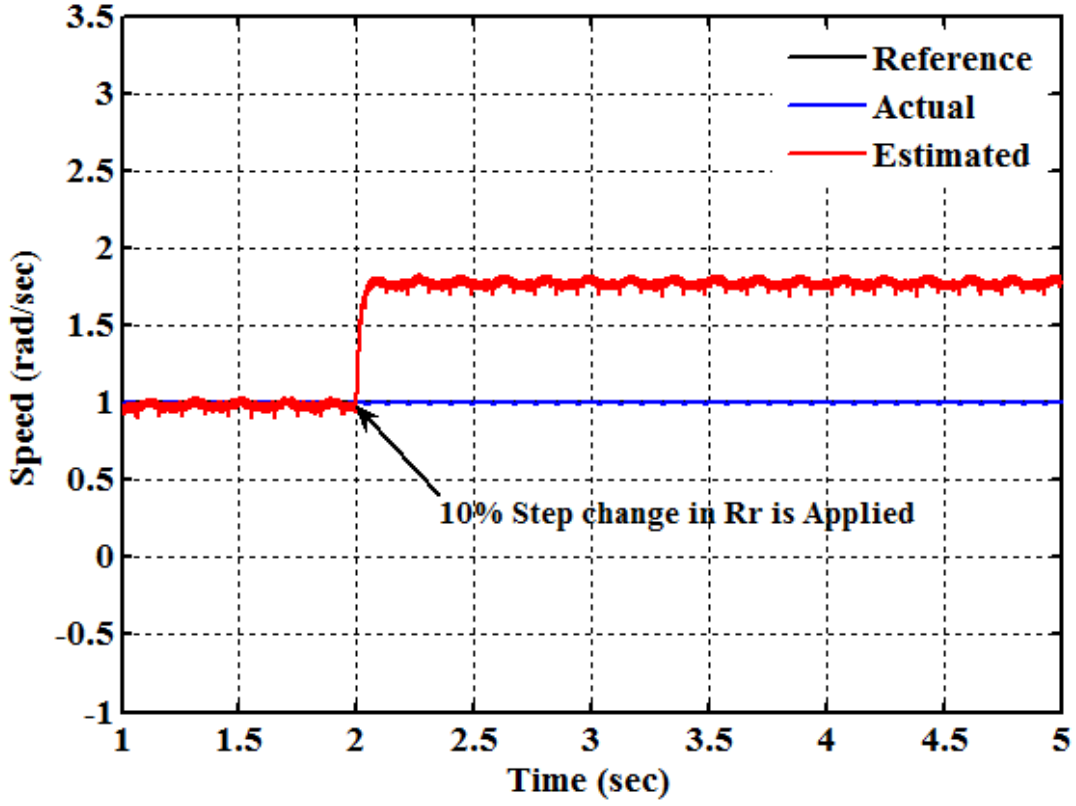
Reference Speed (rad/sec)	Actual Speed (ra d/sec)	NSE model II		Q-MRNLAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	145.021	-0.014	144.998	0.002
125	125.003	124.089	0.731	125.075	-0.057
100	99.998	100.012	-0.014	99.943	0.055
75	75.004	74.998	0.003	75.025	-0.027
50	49.995	50.013	-0.021	49.984	0.022
25	25.006	25.001	-0.008	24.981	0.099
5	4.999	4.999	0.039	4.980	0.380
1	0.999	1.001	-0.200	1.004	**-0.500**

Table 4.2 Performance Comparisons of the Proposed Data Driven SNC-NN with Q-MRNLAS for Various Speed Commands under Full Load Condition

Reference Speed (rad/sec)	Actual Speed (rad/sec)	NSE model II		Q-MRNLAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	145.401	-0.275	145.012	-0.007
125	124.998	125.021	-0.018	124.984	0.011
100	100.002	100.042	-0.039	100.124	-0.124
75	75.001	74.990	0.015	74.929	0.093
50	49.999	50.098	-0.198	50.031	-0.064
25	25.000	24.998	0.008	25.039	-0.156
5	5.001	5.010	-0.179	5.008	-0.099
1	1.001	1.010	-0.899	1.009	**-0.799**

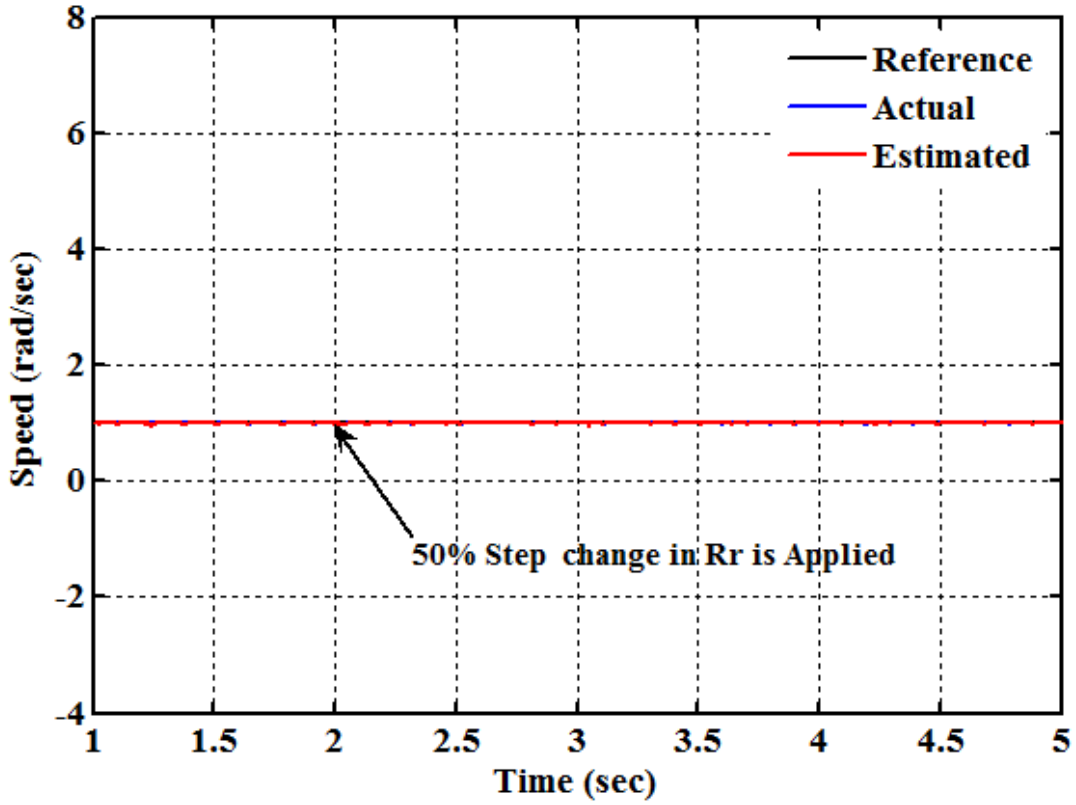


(a)

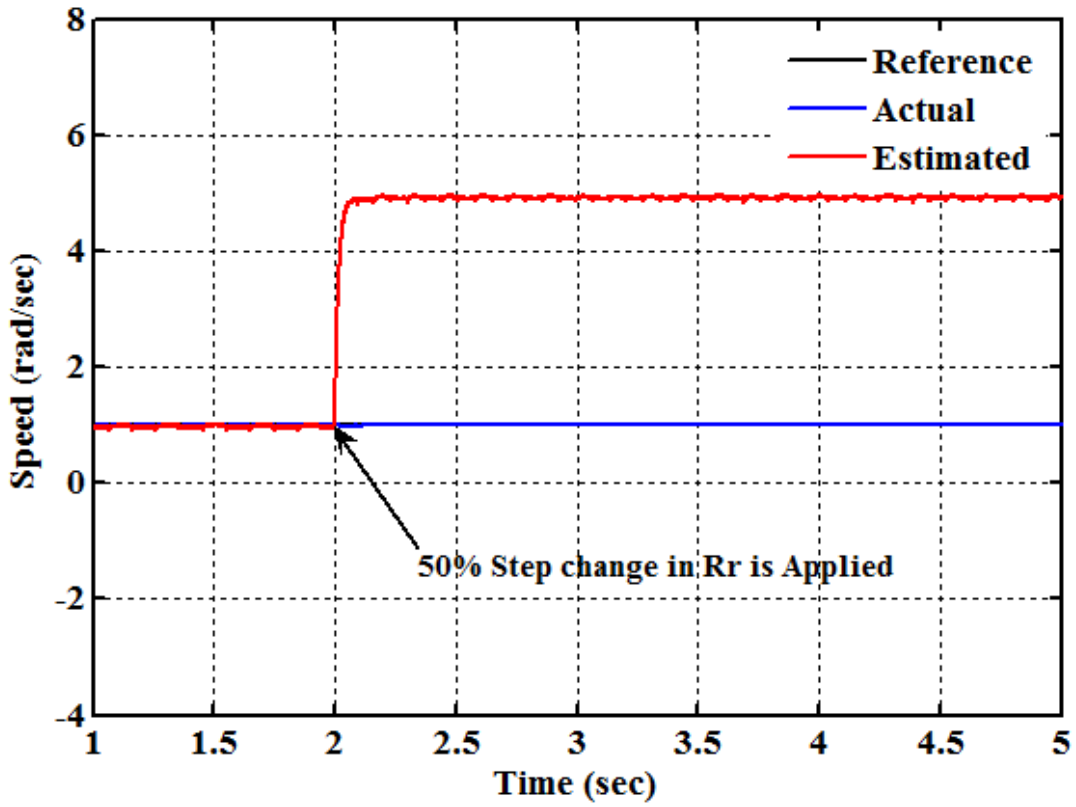


(b)

Figure 4.8 Rotor Speed with Slight R_r Detuning: (a) SNC-NN (b) Q-MRNLAS

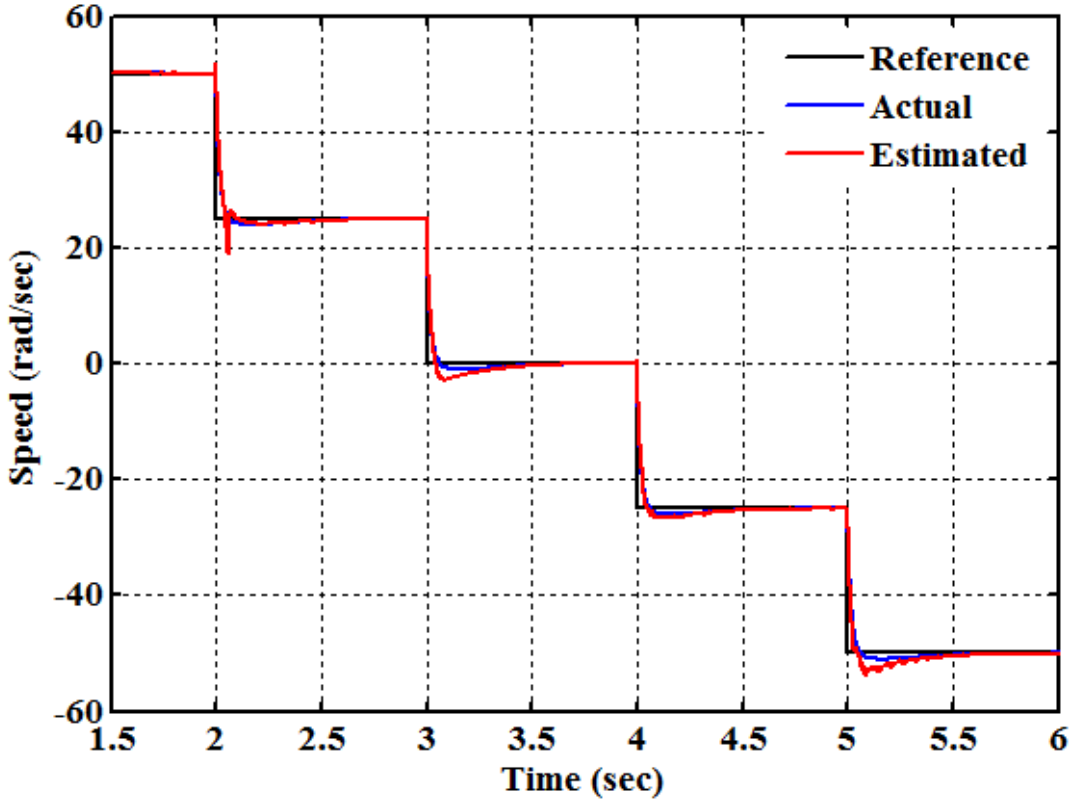


(a)

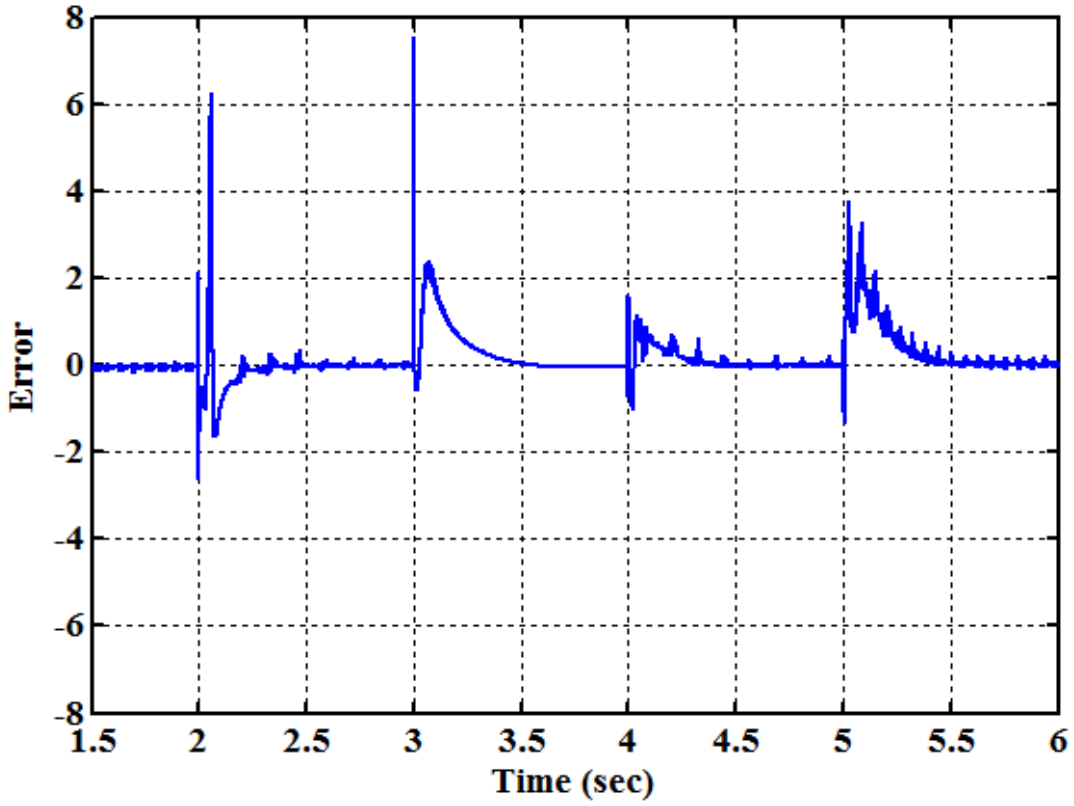


(b)

Figure 4.9 Rotor Speed with Large R_r Detuning: (a) SNC-NN (b) Q-MRNLAS

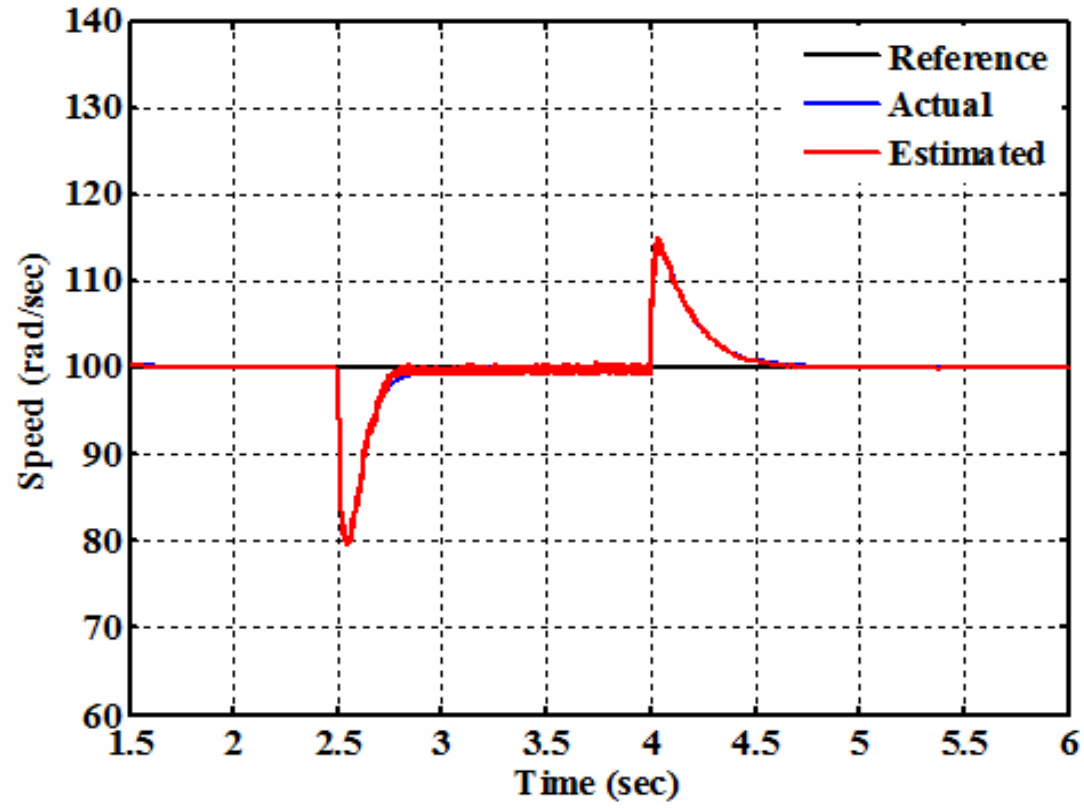


(a)

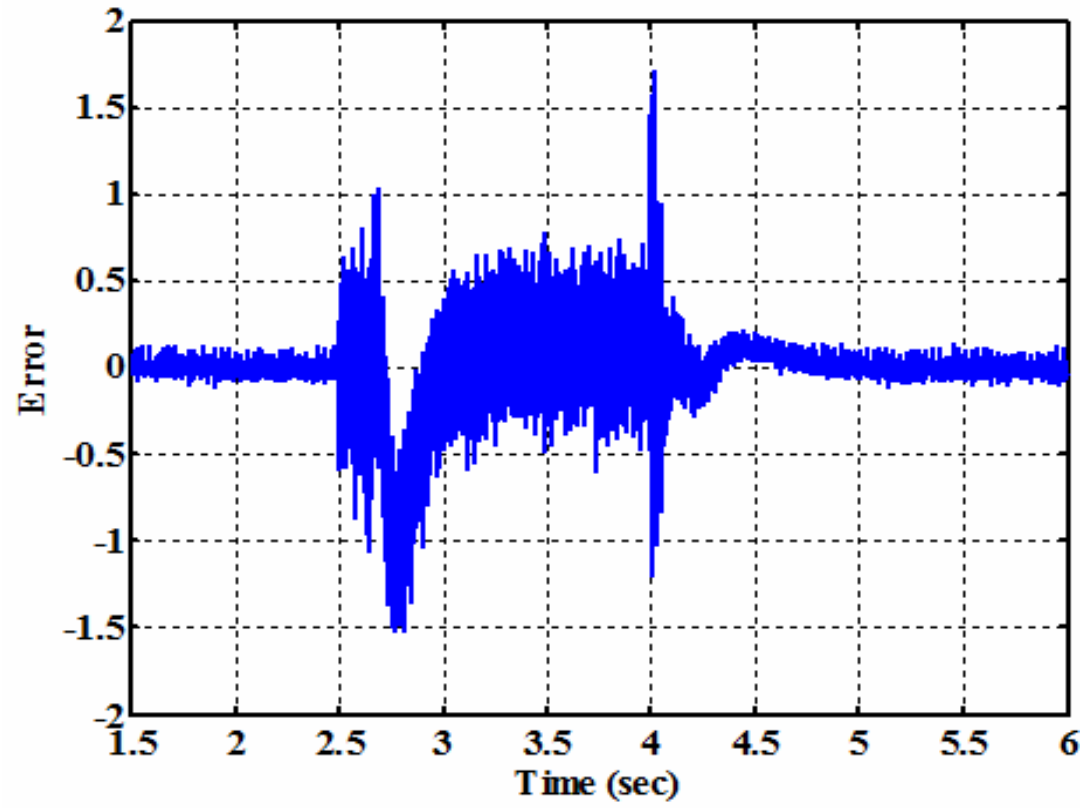


(b)

Figure 5.2 Stair Case Speed Change: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

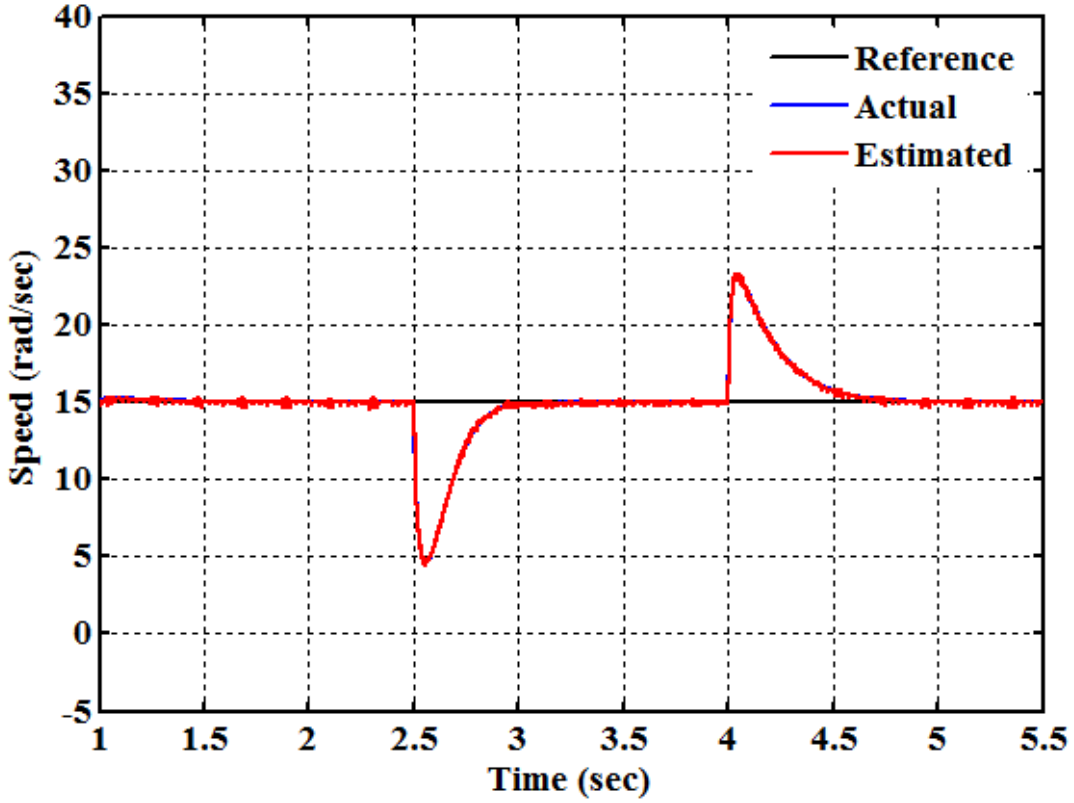


(a)

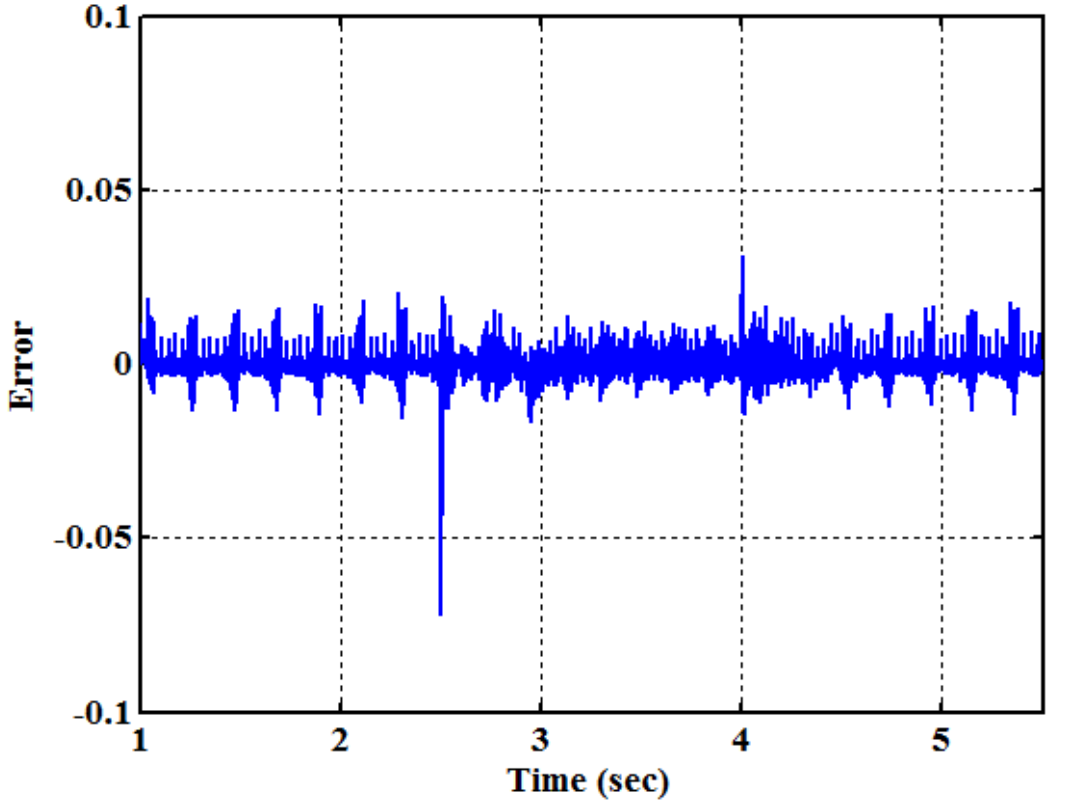


(b)

Figure 5.3 Effect of Loading at 100rad/sec: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

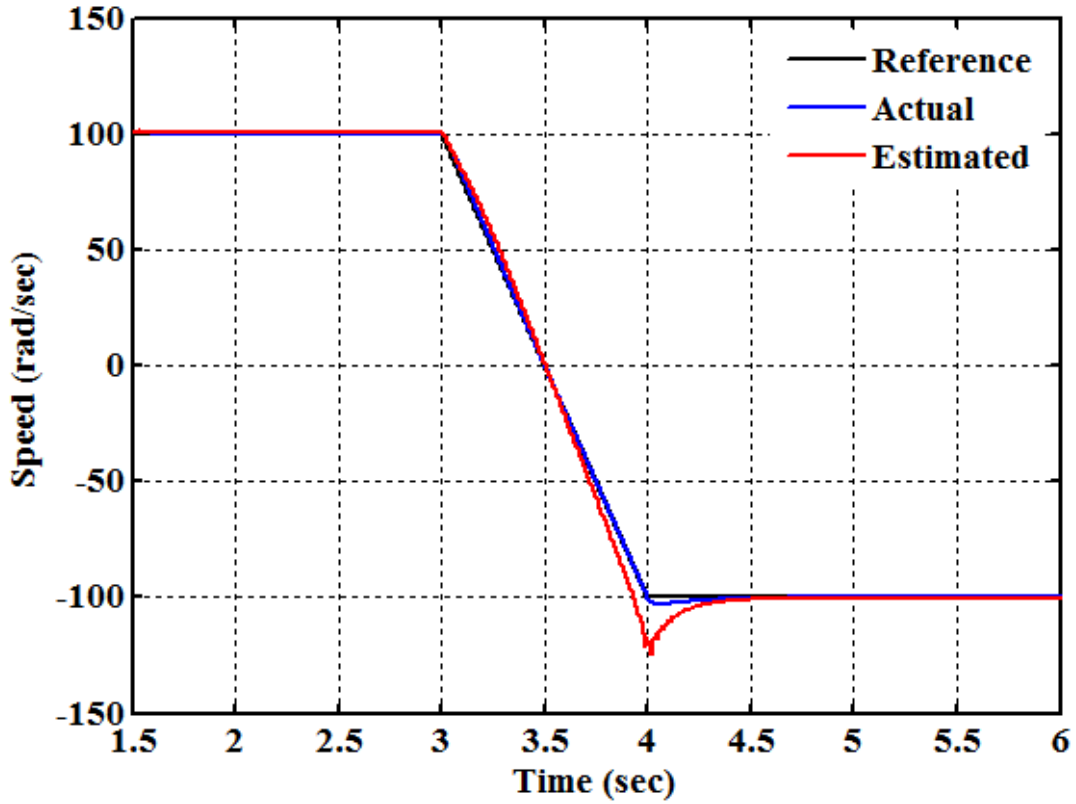


(a)

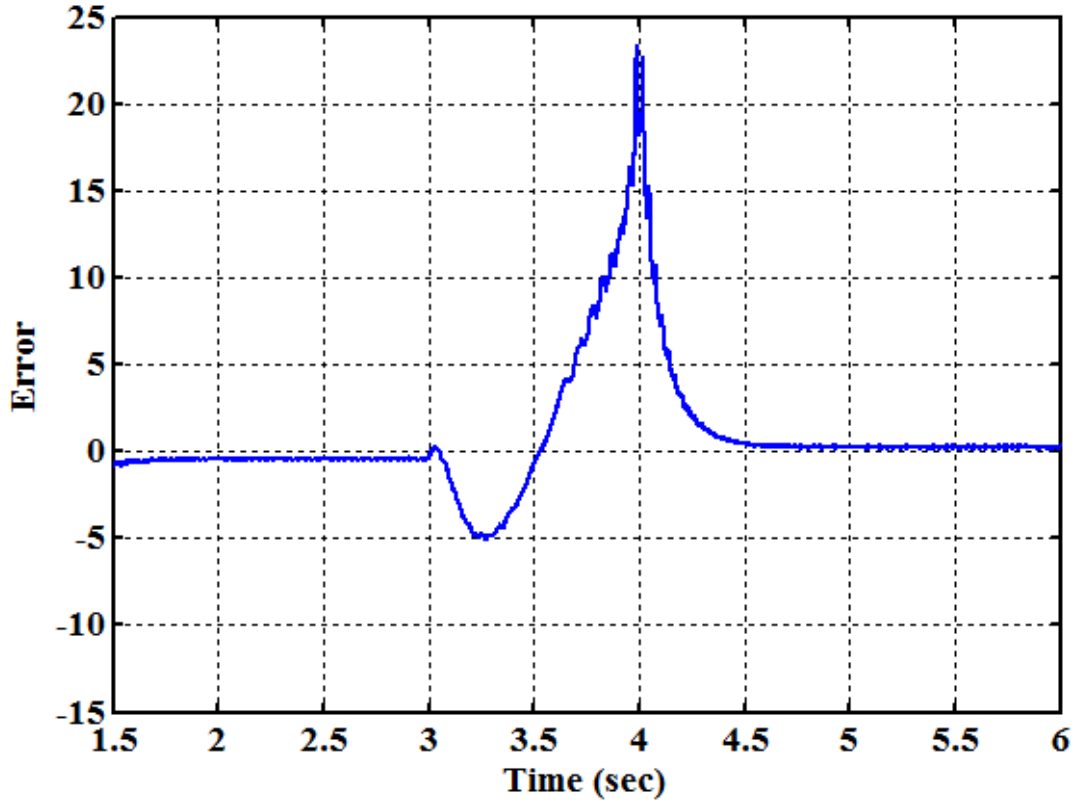


(b)

Figure 5.4 Effect of Loading at 15rad/sec: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

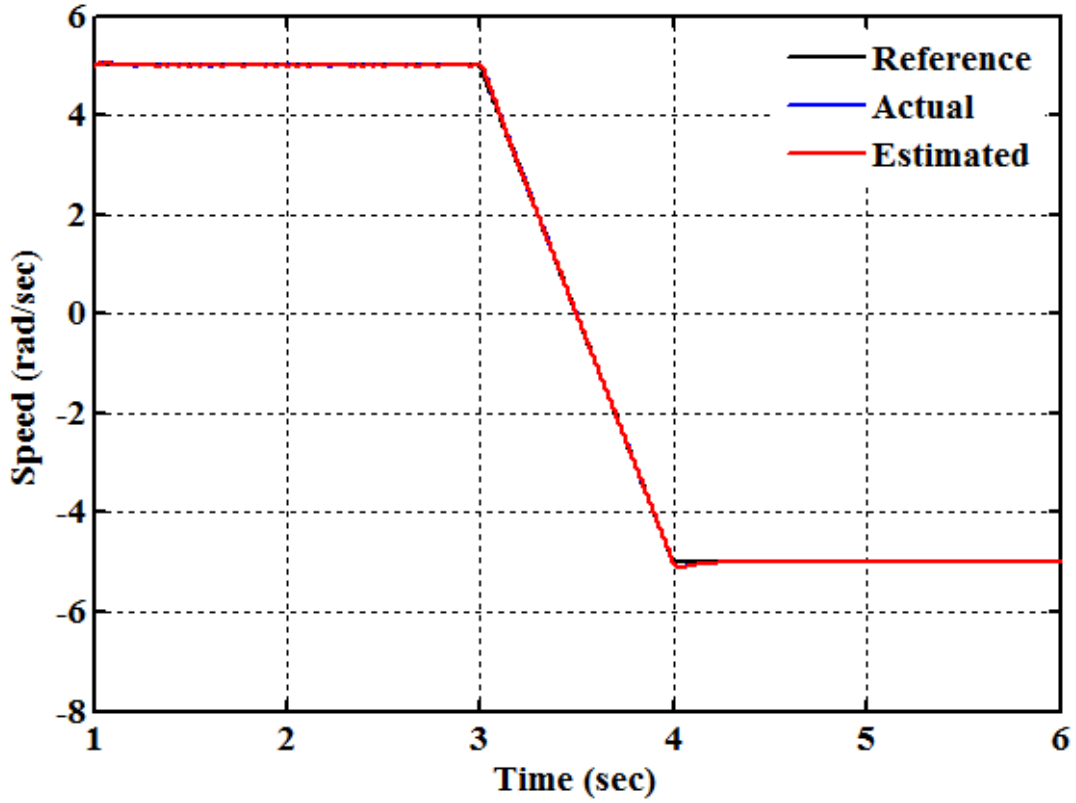


(a)

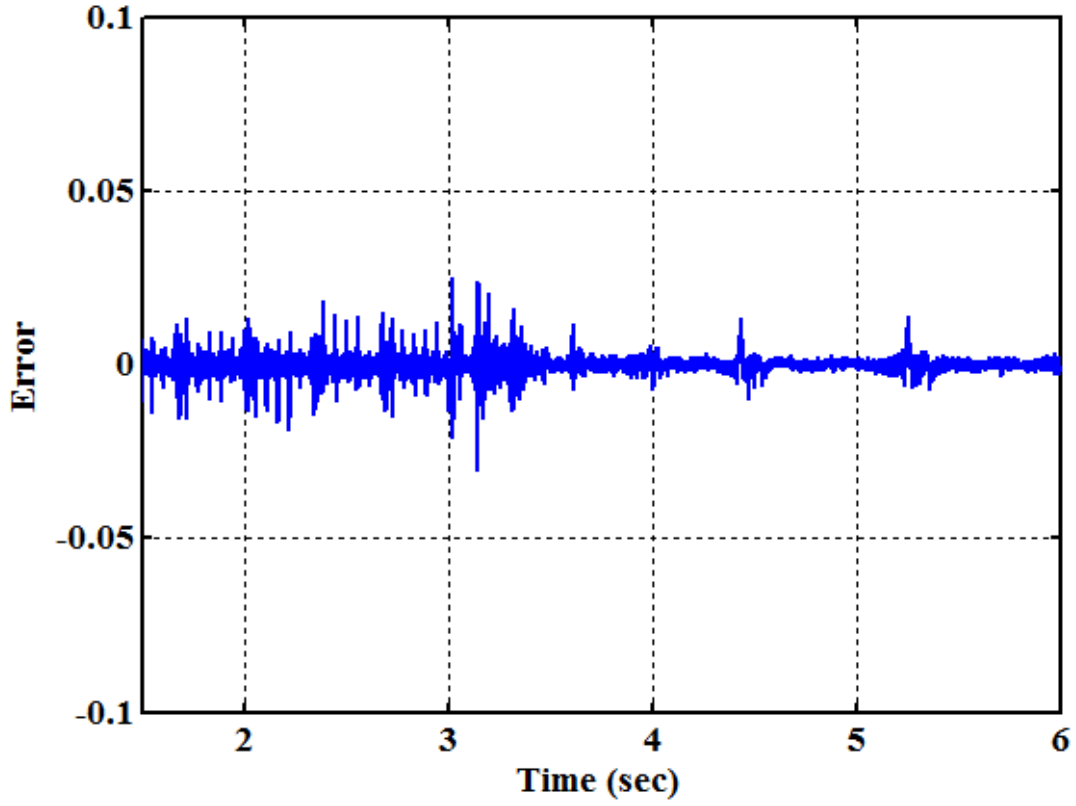


(b)

Figure 5.5 Regenerating Mode Operation (±100rad/sec): (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

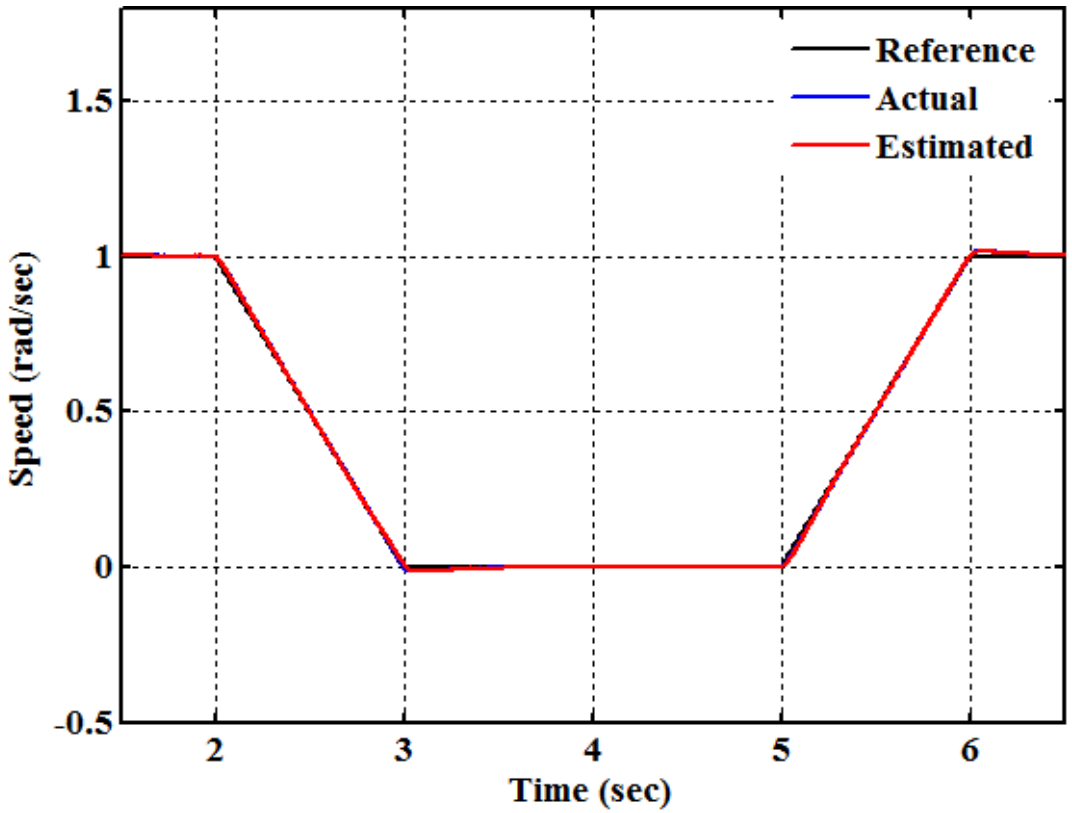


(a)

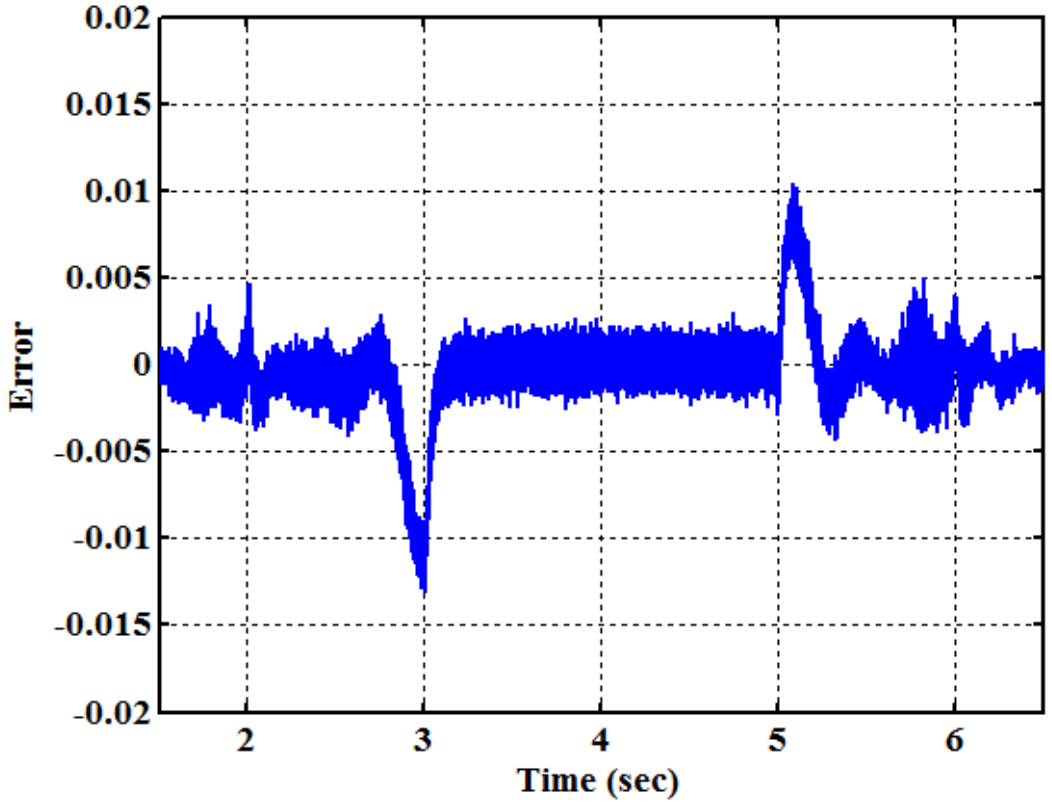


(b)

Figure 5.6 Regenerating Mode Operation (±5rad/sec): (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed



(a)



(b)

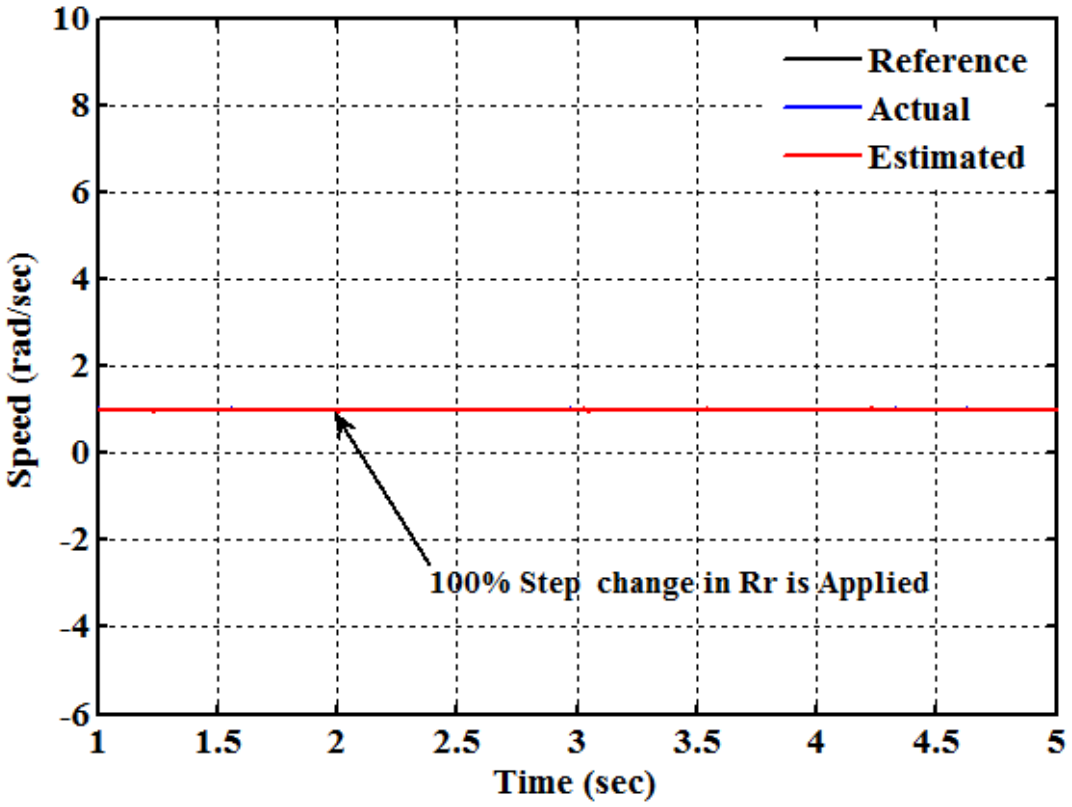
Figure 5.7 Zero Speed Operation: (a) Actual and Estimated Speed, (b) Error between Actual and Estimated Speed

Table 5.1 Performance Comparisons of the Proposed NSE Model III with Q-MRNLAS for Various Speed Commands under No Load Condition

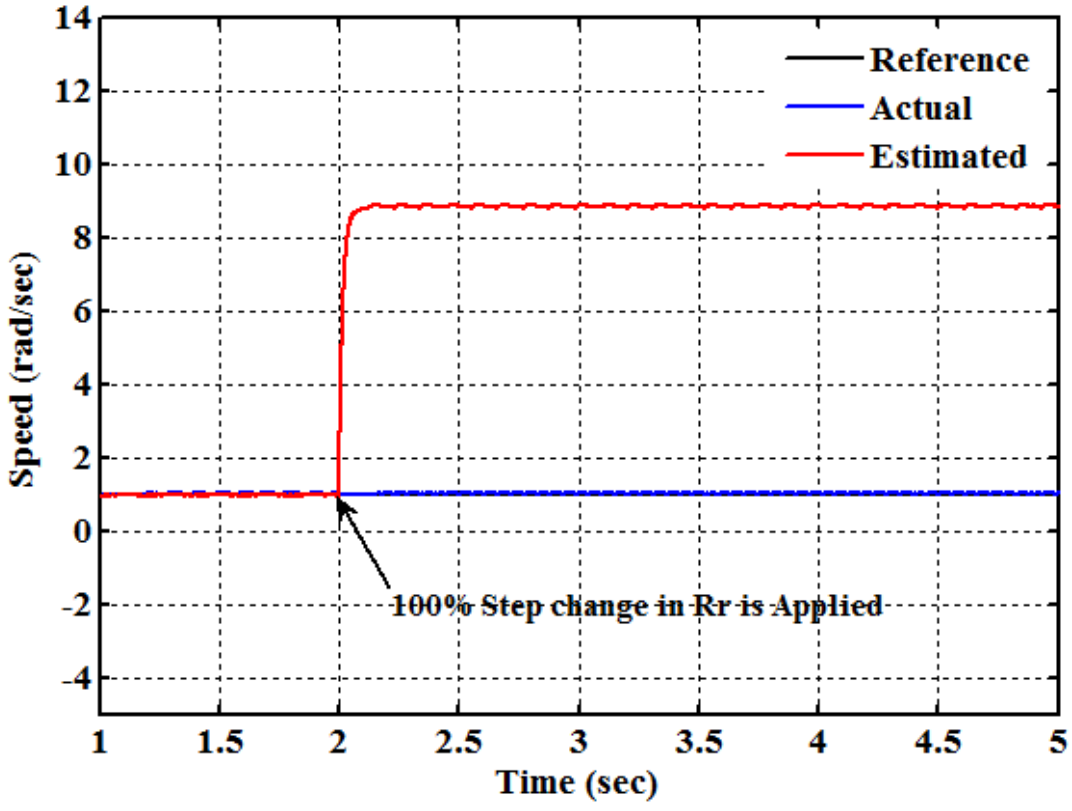
Reference Speed (rad/sec)	Actual Speed (ra d/sec)	NSE model III		Q-MRNLAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	145.201	-0.137	144.998	0.002
125	125.003	124.099	0.723	125.075	-0.057
100	99.998	99.699	0.299	99.943	0.055
75	75.004	75.012	0.015	75.025	-0.027
50	49.995	49.998	-0.021	49.984	0.022
25	25.006	25.012	-0.052	24.981	0.099
5	4.999	5.009	-0.159	4.980	0.380
1	0.999	0.991	-0.801	1.004	**-0.500**

Table 5.2 Performance Comparisons of the Proposed NSE Model III with Q-MRNLAS for Various Speed Commands under Full Load Condition

Reference Speed (rad/sec)	Actual Speed (rad/sec)	NSE model III		Q-MRNLAS	
		Estimated Speed (rad/sec)	% Error (rad/sec)	Estimated Speed (rad/sec)	% Error (rad/sec)
145	145.001	144.012	0.682	145.012	-0.007
125	124.998	125.082	-0.067	124.984	0.011
100	100.002	100.030	-0.027	100.124	-0.124
75	75.001	75.101	-0.013	74.929	0.093
50	49.999	50.009	-0.020	50.031	-0.064
25	25.000	25.025	-0.100	25.039	-0.156
5	5.001	5.035	-0.679	5.008	-0.099
1	1.001	1.011	-1.098	1.009	**-0.799**



(a)



(b)

Figure 5.8 Rotor Speed with 100% R_r Detuning: (a) SNC-NN Model III (b) Q-MRNLAS

Table 5.3 Performance Comparison for Changes in R_r at 1rad/sec with Full Load

% Change in R_r	Model III based Estimated Speed (rad/sec)	Q-MRNLAS based Estimated Speed (rad/sec)
100	1.025	8.901
80	1.041	7.128
60	1.060	6.354
40	1.072	3.939
20	1.050	2.897

Table 5.4 Performance Comparison for 50% Change in R_r at Different Speeds on Full Load

Reference Speed	Actual Speed (rad/sec)	Model III based Estimated Speed (rad/sec)	Q-MRNLAS based Estimated Speed (rad/sec)
145	145.011	145.324	154.135
100	100.002	100.070	107.185
75	75.051	74.903	81.284
50	49.979	49.908	56,345
25	25.030	25.008	29.629
1	1.001	0.998	4.969

Table 5.5 Comparison of Structural Compactness and Computational Complexity for the Proposed NN Models

NN Models	NN Architectures	No. of Hidden Neurons	No. of Parameters	Computations		
				No. of Additions	No. of Multiplications	No. of Tan-Sigmoids
Model I	6-15(h)-1	15	232	216	216	15
Model II	5-20(h)-1	20	336	315	315	20
Model III	5-25(h)-1	25	481	455	455	25

h - hidden layer with one neuron

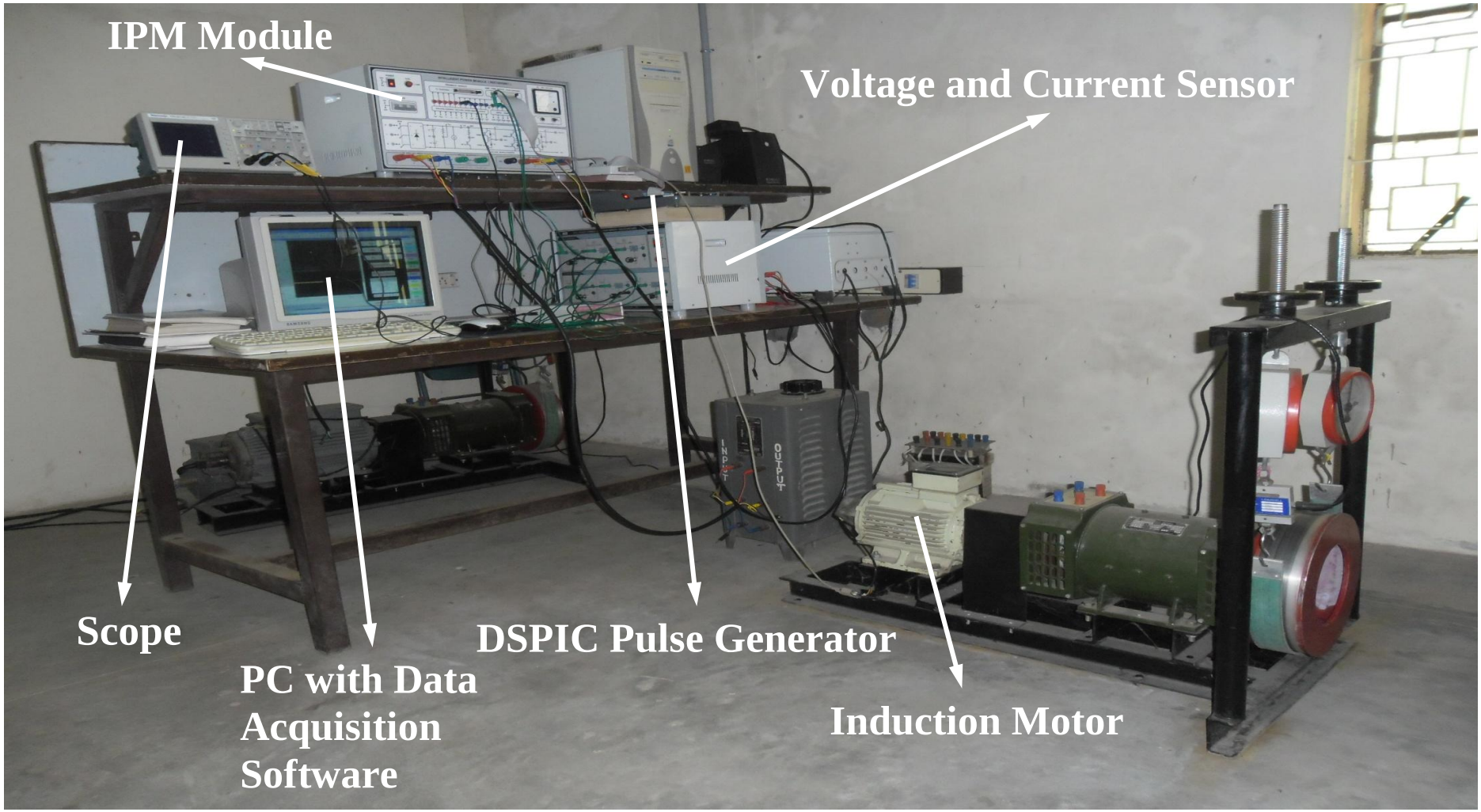


Figure 6.1 Experimental Setup to obtain the Practical Data.

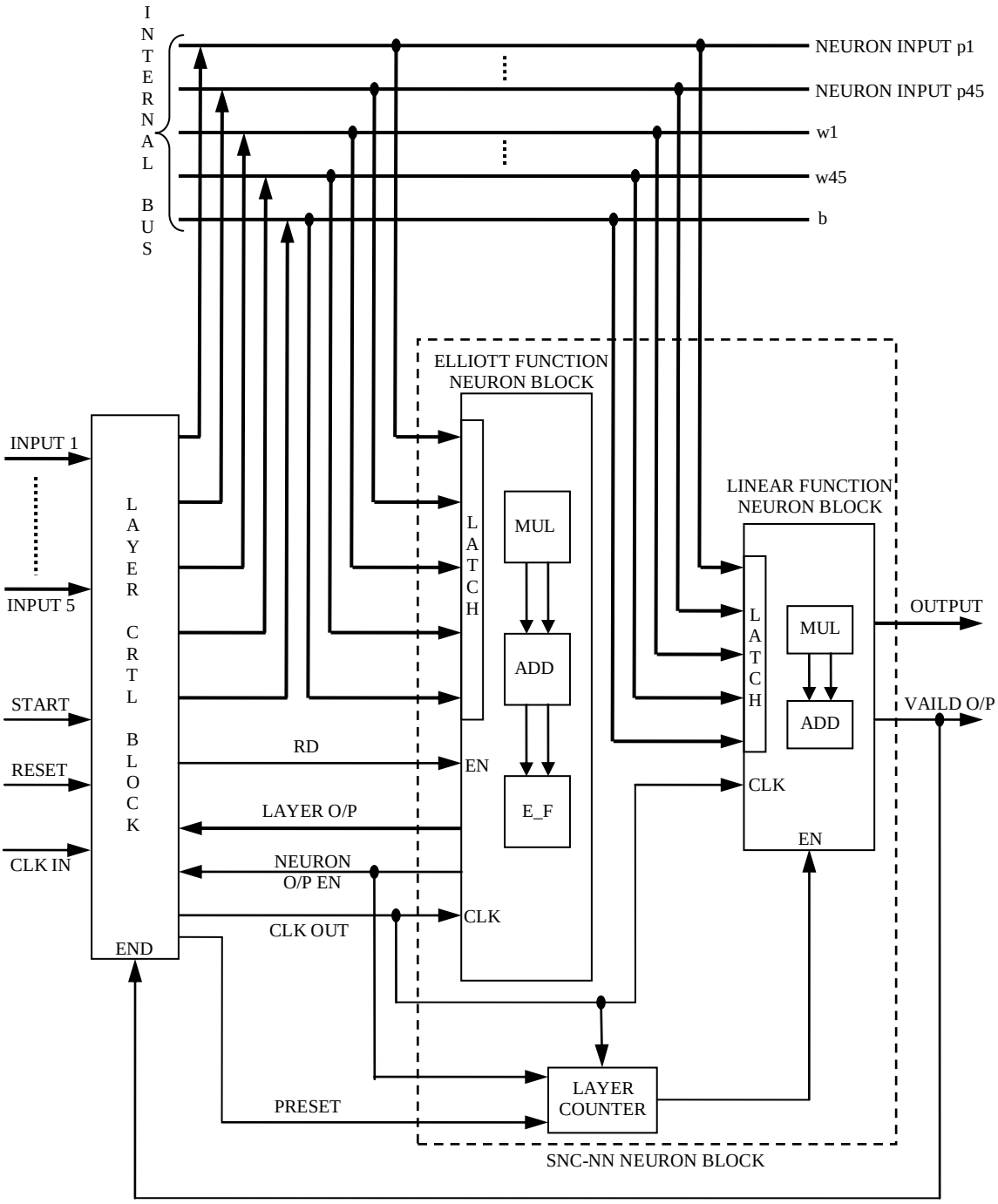


Figure 6.14 Schematic for Implementation of SNC-NN based Speed Estimation with Layer Multiplexing

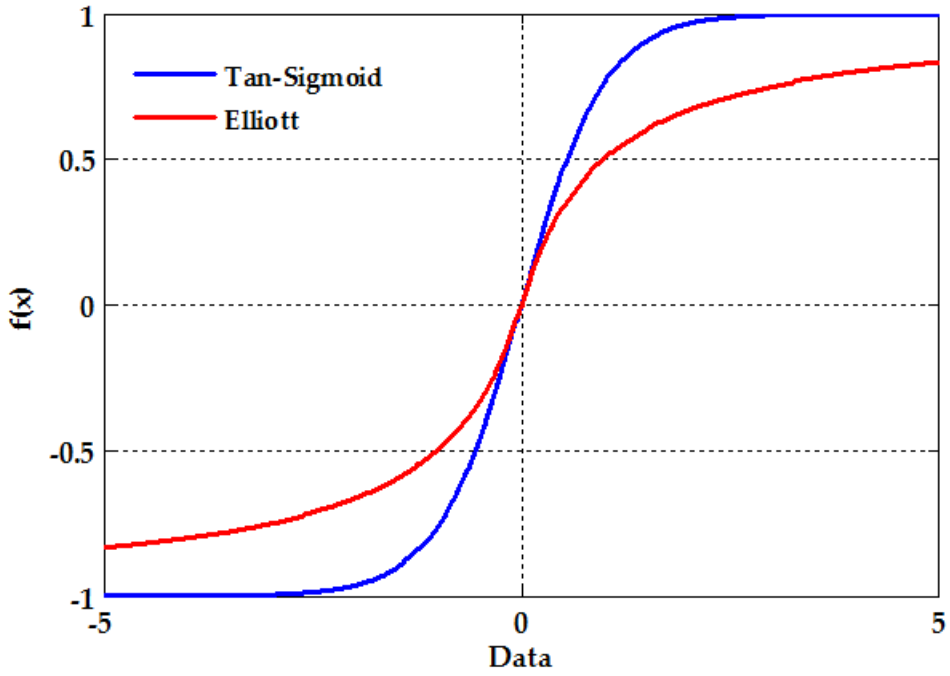


Figure 6.5 Plot of Non-linearity for Tansigmoid and Elliott Function

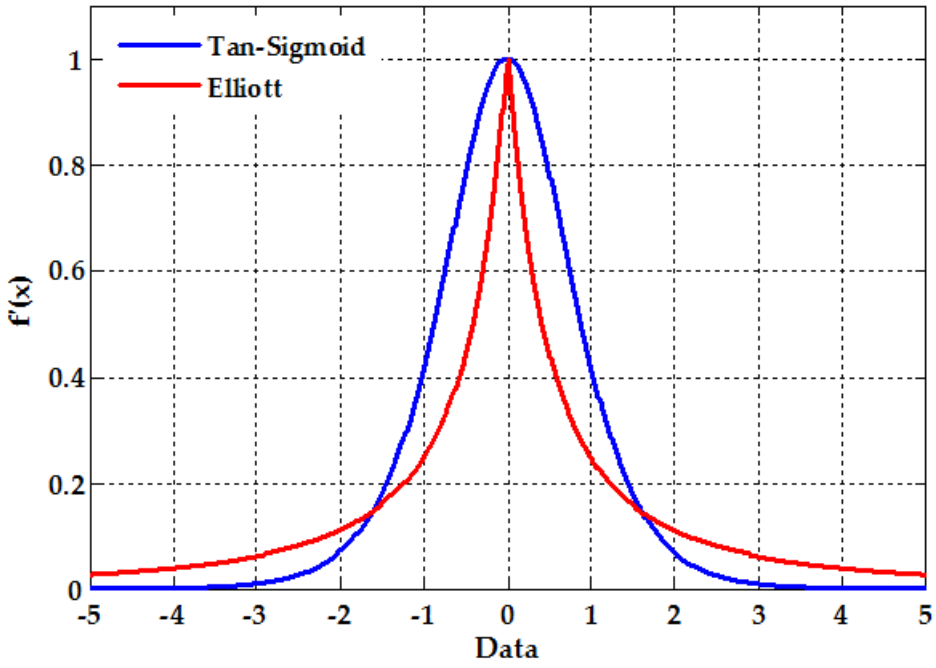
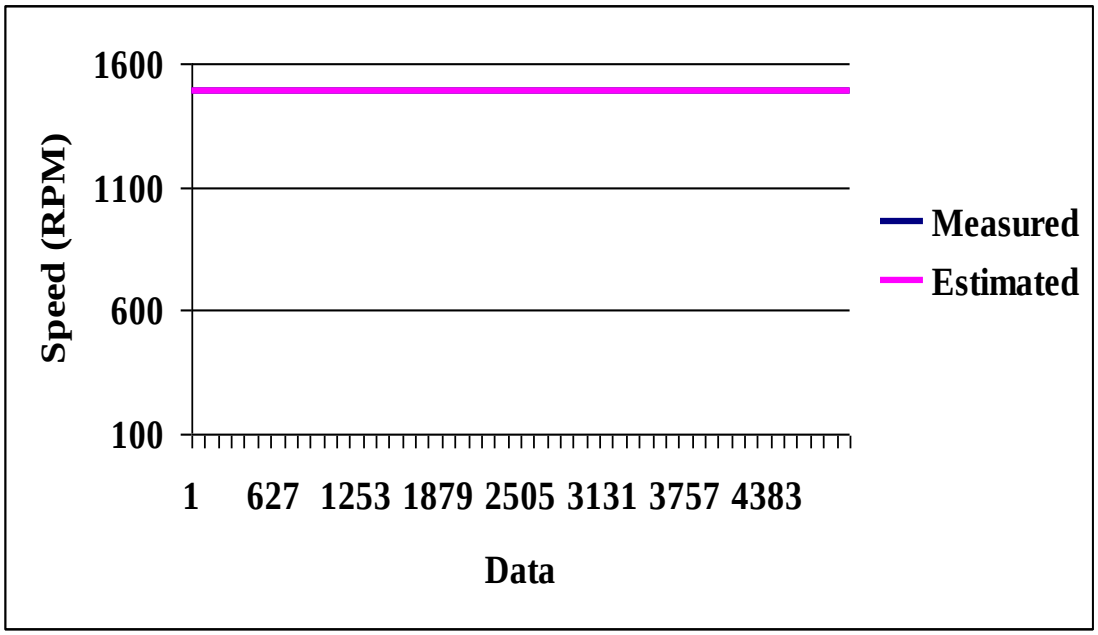
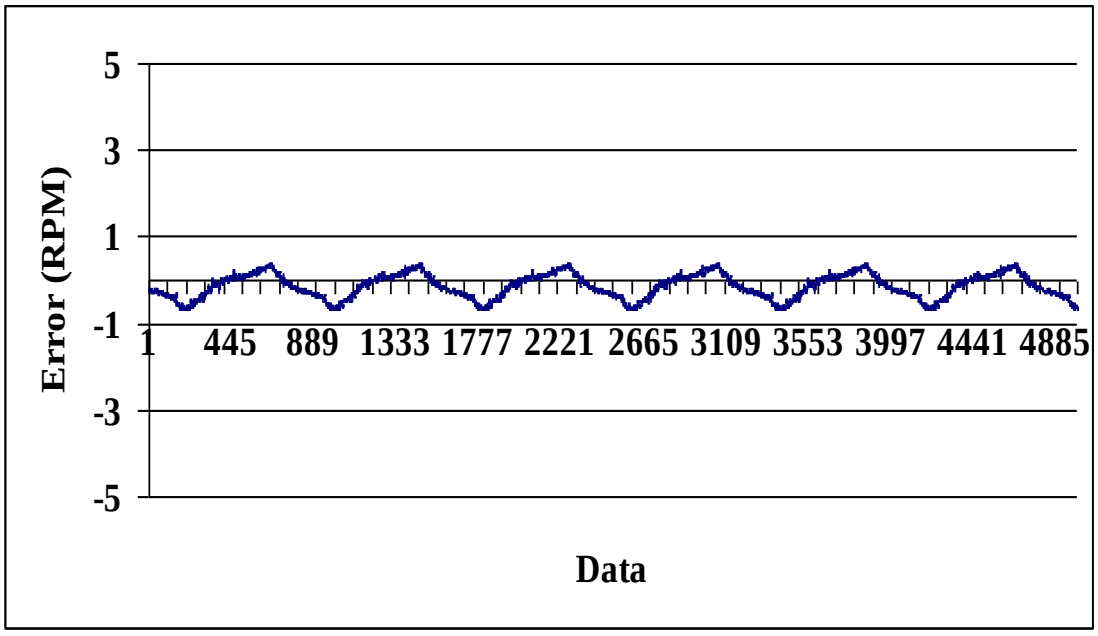


Figure 6.6 Plot of Gradient for Tan-sigmoid and Elliott Function



(a)



(b)

Figure 6.7 Experimental Result for Model III with Elliott function based Speed Estimator: (a) Measured and Estimated Speed, (b) Error between Measured and Estimated Speed

Publications

International Journals

1. **K.Sedhuraman**, S.Himavathi and A.Muthuramalingam, "Neural Learning Adaptive System Using Simplified Reactive Power Reference Model based Speed Estimation in Sensorless Indirect Vector Controlled Induction Motor Drives," *Archives of Electrical Engineering*. vol. 62, no. 1, pp. 25-41, Mar. 2013.

2. **K.Sedhuraman**, S.Himavathi and A.Muthuramalingam, "Reactive Power based Model Reference Neural Learning Adaptive System for Speed Estimation in Sensor-less Induction Motor Drives," *The Journal of Engineering Research*, Vol. 9, No. 2, pp. 17-26, Jun. 2012.

3. S. Himavathi, A. Muthuramalingam, A. Venkadesan and **K. Sedhuraman**, "Nonlinear System Modeling Using Single Neuron Cascaded Neural Network For Real-Time Application," *ICTACT Journal On Soft Computing*, Vol. 02, Iss. 03, pp. 309-318, Apr. 2012.

4. **K.Sedhuraman**, S.Himavathi and A.Muthuramalingam, "Performances Comparison of Neural Architectures for On-Line Speed Estimation in Sensorless IM Drives," *World Academy of Science, Engineering and Technology*, Iss. 60, pp.1318-1325, Dec. 2011.

5. **K.Sedhuraman**, A.Muthuramalingam and S.Himavathi, "Single Neuron Cascaded Neural Network Model based Speed Estimation for Sensorless Induction Motor Drives," *International Journal of Recent Trends in Engineering and Technology*, vol. 4, no.3, pp.115-119, Nov. 2010.

International Conferences

1. **K.Sedhuraman,** S.Himavathi and A.Muthuramalingam, "Comparison of Learning Algorithms for Neural Network based Speed Estimator in Sensorless Induction Motor Drives," Proc. *IEEE – International Conference on Advances in*

Engineering, Science and Management (ICAESM-2012), E.G.S Pillay Engineering College, Naggapattinum, Tamil Nadu, India, Mar. 2012.

2. S.Himavathi, **K.Sedhuraman,** and A.Muthuramalingam, "Neural Learning Algorithm for MRAS based Speed Estimation using Reactive Power Technique in Sensor-less IM Drives," *International Conference on System Dynamics and Control* (ICSDC-2010), Manipal Institute of Technology, Manipal, Aug. 2010.

Under Review

1. **K.Sedhuraman,** S.Himavathi and A.Muthuramalingam, "Neural Network based On-Line Speed Estimator Independent of Rotor Resistance for Sensorless Indirect Vector Controlled Induction Motor Drives," *Journal Electrical Engineering – Springer*.

References

1. Abbondanti A., and Brennen M.B., "Variable speed induction motor drives use electronic slip calculator based on motor voltages and currents," IEEE Trans. Ind. Appl., vol. IA-11, no. 5, pp. 483 – 488, Sept. 1975.

2. Akun B., Orguner U., Ersak A., and Ehsani M., "A comparative study on non-linear state estimators applied to sensorless AC drives: MRAS and Kalman filter," Inc. Proc. 30th Annual Conf. IEEE Industrial Electronic Society, vol. 3, pp. 2148–2153, Nov. 2004.

3. Aydogan Savran, "Multifeedback-Layer Neural Network," IEEE Tran. on Neural Networks, vol.18, no.2, pp.373-384, Mar. 2007.

4. Blaschke F., "The principle of field orientation as applied to the new transvector closed loop system for rotating field machines," Siemens Review, vol. 34, pp. 217-220, May 1972.

5. Bodson M., Chiasson J., and Novotnak R.T., "Nonlinear speed observer for high - performance induction motor control," IEEE Trans. Ind. Elec., vol. 42, no. 4, pp. 337-343, Aug. 1995.

6. Bose B.K., and Patel N.R., "A programmable cascaded low-pass filter-based flux synthesis for a stator flux-oriented vector-controlled induction motor drive," IEEE Tran. Ind. Electro., vol.44, no.1, pp.140-143, Feb. 1997.

7. Bose B.K., "Modern Power Electronics and AC Drives," Prentice Hall of India, 2005.

8. Bose B.K., "Neural network applications in power electronics and motor drives—Introduction and perspective," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 1–20, Feb. 2007.

9. Briz F., Degner M.W., Garcia P., and Lorenz R.D., “Comparison of saliency-based sensorless control techniques for AC machines,” IEEE Trans. Ind. Appl., vol. 40, no. 4, pp. 1107–1115, July/Aug. 2004.

10. Caruana C., Asher G.M., and Sumner M., “Performance of HF signal injection techniques for zero-low-frequency vector control induction machines under sensorless conditions,” IEEE Trans. Ind. Electron., vol. 53, no. 1, pp. 225–238, Feb. 2006.

11. Chen T.C., and Sheu T.T., “Model Reference Neural Network Controller for Induction Motor Speed Control," IEEE Trans. on Energy Conversion, vol. 17, no. 2, pp. 157- 163, Jun. 2002.

12. Dinko Vukadinović and Mateo Bašić, “Artificial Neural Network Applications in Control of Induction Machines,” Nova Science Publishers, Inc., New York, 2011.

13. Du T., Vas P., and Stronach F., “Design and application of extended observers for joint state and parameter estimation in high performance ac drives,” IET Proc. Elec. Power Appl., vol. 142, no. 2, pp. 71-78, Mar. 1995.

14. Elliott D.L., “A Better Activation Function for Artificial Neural Networks,” ISR Technical Report TR 93-8, Jan. 1993.

15. Fahlman S.E., and Lebiere C., “The Cascade-Correlation Learning Architecture,” CMU-CS-90-100, pp. 1-13, Aug. 1991.

16. Ferrah A., Hogben-Laing P.J., Bradley K.J., Asher G.M., and Woolfson, M.S., “The effect of rotor design on sensorless speed estimation using rotor slot harmonics identified by adaptive digital filtering using the maximum likelihood approach,” In: Conference Proceedings of the IEEE Thirty-Second IAS Annual Meeting, vol. 1, pp.128-135, Oct. 1997.

17. Fredric M. Ham, and Ivica Kostanic, "Principles of Neurocomputing for Science & Engineering," Tata McGraw-Hill, Pvt. Ltd., India 2008.

18. Heredia J.R., Perez Hidalgo F., and Duran Paz J.L., "Sensorless control of induction motors by artificial neural networks," IEEE Tran. Ind. Electro., vol. 48, no. 5, pp. 1038-1040, Oct. 2001.

19. Himavathi S., and Umamaheswari B., "New Membership Functions for Effective Design and Implementation of Fuzzy Systems," IEEE Tran. on Systems, Man and Cybernetics Part A: Systems and Humans, vol.31, no.6, pp. 717-723, Nov. 2001.

20. Himavathi S., Anitha D., and Muthuramalingam A., "Feed-forward Neural Network Implementation in FPGA Using Layer Multiplexing for Effective Resource Utilization," IEEE Tran. on Neural Networks, vol.18, no. 3, pp. 880-888, May 2007.

21. Holtz J., "Ssensorless control of induction motor drives," Proceedings of the IEEE, vol. 90, no. 8, pp. 1359-1394, Aug. 2002.

22. Holtz J., and Quan J., "Drift- and Parameter-Compensated Flux Estimator for Persistent Zero-Stator- Frequency Operation of Sensorless-Controlled Induction Motors," IEEE Trans. Ind. Appl., vol.39, no.4, pp. 1052-1060, Jul./Aug. 2003.

23. Hornik K., Stinchcombe M., and White H., "Multilayer feedforward networks are universal approximators," Neural Networks, vol.2, no.5 pp. 359-366, 1989.

24. Hornik K., Stinchcombe M., and White H., "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," Neural Networks, vol.3, no.5, pp. 551-560, 1990.

25. Jansen P.L., and Lorenz R.D. "Transducerless field orientation concepts employing saturation-induced saliencies in induction machines," IEEE Trans Ind. Appl., vol. 32, no. 6, pp. 1380-1393, Nov./Dec.1996.

26. Jevremovic V.R., Vasic V., Marcetic D.P., and Jeftenic B., “Speed-sensorless control of induction motor based on reactive power with rotor time constant identification,” IET Electr. Power Appl., vol. 4, no. 6, pp. 462–473, July 2010.

27. Jung-Ik Ha, and Seung-Ki Sul, “Sensorless field-orientation control of an induction machine by high-frequency signal injection,” IEEE Trans. Ind. Appl., vol. 35, no. 1, pp. 45– 51, Jan./Feb. 1999.

28. Karanayil B., Rahman M..F., and Grantham C., “Online Stator and Rotor Resistance Estimation Scheme Using Artificial Neural Networks for Vector Controlled Speed Sensorless Induction Motor Drive,” IEEE Trans. on Ind. Electro., vol. 54, no. 1, pp. 167-176, Feb. 2007.

29. Kevin D. Hurst, Habetler T.G., Griva G., and Profumo F., “Zero-speed tacholess IM torque control: simply a matter of stator voltage integration,” IEEE Tran. Ind. Appl., vol. 34, no. 4, pp. 790-795, July/August 1998.

30. Kim Y.R., Sul S.K, and Park M.H., “Speed sensorless vector control of induction motor using extended Kalman filter,” IEEE Trans. Ind. Appl., vol. 30, no. 5, pp.1225-1233, Sep./Oct.1994.

31. Kojabadi H.M., “Simulation and experimental studies of model reference adaptive system for sensorless induction motor drive,” Simulat. Model Practice Theory, vol. 13, pp. 451–464, 2005.

32. Kojabadi H.M., “Active power and MRAS based rotor resistance identification of an IM drive,” Simulat. Model Practice Theory, vol. 17, no. 2, pp. 376–389, 2009.

33. Krishnan R., “Electirc Motor Drives Modeling Analysis and Control,” Prentice Hall, 2000.

34. Lazhar Ben-Brahim, Susumu Tadakuma, and Alper Akdag, “Speed Control of Induction Motor Without Rotational Transducers,” IEEE Trans. Ind. Appl., vol. 35, no. 4, pp. 844-850, Jul./Aug. 1999.

35. Lech M. Grzesiak, and Marian P. Kazmierkowski, “Improving flux and speed estimators for sensorless AC drives,” IEEE Industrial Electronics Magazine, pp. 8-19, Full 2007.

36. Lehtokangas M., “Modified cascade-correlation learning for classification,” IEEE Tran. on Neural Networks, vol. 11, no. 3, pp. 795-798, May 2000.

37. Maiti S., Chakraborty C., Hori Y., and Ta M.C., “Model Reference Adaptive Controller-Based Rotor Resistance and Speed Estimation Techniques for Vector Controlled Induction Motor Drive Utilizing Reactive Power,” IEEE Trans Ind. Electron., vol. 55, no. 2, pp. 594–601, Feb. 2008.

38. Maiti S., and Chakraborty C., “A new instantaneous reactive power based MRAS for sensorless induction motor drive,” Simulat. Model Practice Theory, vol. 18, pp. 1314-1326, May 2010.

39. Marko Hinkkanen, Lennart Harnefors, and Jorma Luomi “Reduced-order flux observers with stator-resistance adaptation for speed-sensorless induction motor drives,” IEEE Trans. Power Elect., vol. 25, no. 5, pp. 1173-1183, May 2010.

40. Martin T. Hagan and Mohammad B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” IEEE Tran. on Neural Networks, vol. 5, no. 6, pp. 989-993, Nov. 1994.

41. Martin T.Hagan, Howard B. Demuth, and Mark Beale, “Neural Network Design,” Cengage Learning, Pvt. Ltd., India 2008.

42. Maurizio Cirrincione, and Marcello Pucci, “An MRAS-Based Sensorless High-Performance Induction Motor Drive with a Predictive Adaptive Model,” IEEE Trans. Ind. Electron., vol. 52, no. 2, pp. 532-551, Apr. 2005.

43. Michael W. Degner, and Robert D. Lorenz, “Position estimation in induction machines utilizing rotor bar slot harmonics and carrier-frequency signal injection,” IEEE Trans. Ind. Appl. vol. 36, no. 3, pp. 736-742, May/June 2000.

44. Mondal S.K., Pinto J.O.P., and Bose B.K., “A neural network based space vector PWM controller for a three voltage fed inviter induction motor drive,” IEEE Trans. Ind. App., vol. 30, no. 3, pp 660-669, May/June 2002.

45. Narendra K.S., and Parthasarathy, K., “Identification and control of dynamical system using neural network,” IEEE Trans. on Neural Networks, vol. 1, no. 1, pp.4-27, Mar. 1990.

46. Narendra, K.S., and Parthasarathy, K., “Gradient methods for the optimization of dynamical systems containing neural networks,” IEEE Tran. on Neural Networks, vol. 2, no. 2, pp 252-262, Mar. 1991.

47. Nicholas K.Treadgold and Tramas D. Gedeon, “Exploring constructive cascade networks,” IEEE Tran. on Neural Networks, vol. 10, pp.1335-1350, Nov. 1999.

48. Ohtani T., Takada N., and Tanaka K., “Vector control of induction motor without shaft encoder,” IEEE Trans. Ind. Appl., vol. 28, no. 1, pp. 157-164, Jan./Feb. 1992.

49. Peng F.Z., and Fukao T., “Robust speed identification for speed sensorless vector control of induction motor,” IEEE Trans Ind. Appl., vol. IA-30, no. 5, pp. 1234–1240, Sep./Oct.1994.

50. Peng F.Z., Fukao T., and Lai J.S., “Low-speed performance of robust speed identification using instantaneous reactive power for tacholess vector control of induction motors.” In: Proceeding of IEEE-IAS Annual Meetings vol. 1, pp. 509–514, Oct. 1994.

51. Rajashekara K., Kawamura A. and Matsue K., "Sensorless Control of AC Motor Drives," IEEE Press Book, 1996.

52. Rashed M., and Stronach A.F., “A stable back-EMF MRAS-based sensorless low speed induction motor drive insensitive to stator resistance variation,” IEE Proc. Elec. Power Appl., vol. 151, no. 6, pp. 685–693, Nov. 2004.

53. Schauder C., “Adaptive speed identification for vector control of induction motors without rotational transducers,” IEEE Trans. Ind. Appl., vol. 28, no. 5, pp. 1054-1061, Sep./Oct. 1992.

54. Schroedl M., “Sensorless control of AC machines at low speed and standstill based on the “INFORM” method,” In: Conference Proceedings of the IEEE Thirty-First IAS Annual Meeting, vol. 1, pp. 270-277, Oct. 1996.

55. Seong-Hwan Kim, Tae-Sik Park, Ji-Yoon Yoo, and Gwi-Tae Park, ”Speed-sensorless vector control of an induction motor using neural network speed estimation,” IEEE Tran. Ind. Electron., vol. 48, pp. 609-614, June 2001.

56. Setiono R. and Lucas Chi Kwong Hui, “Use of a quasi-Newton method in a feedforward neural network construction algorithm,” IEEE Tran. on Neural Networks, vol. 6, no. 1, pp. 273–277, Jan. 1995.

57. Shady M. Gadoue, Damian Giaouris, and John W.Finch, “Sensorless Control of Induction Motor Drives at Very Low and Zero Speeds Using Neural Network Flux Observers,” IEEE Tran. Ind. Electro., vol. 56, no. 8, pp. 3029-3039, Aug. 2009.

58. Teresa Orlowska-Kowalska, and Mateusz Dybkowski, “Stator-Current-Based MRAS Estimator for a Wide Range Speed-Sensorless Induction-Motor Drive,” IEEE Trans. Ind. Electron., vol. 57, no. 4, pp. 1296-1308, April 2010.

59. Vas P., "Sensorless Vector and Direct Torque Control," Oxford Science Publication, 1998.

60. Vas P., “Artificial-Intelligence-Based Electrical Machines and Drives-Application of Fuzzy. Neural, Fuzzy-Neural and Genetic Algorithm Based Techniques,” Oxford University Press, New York, 1999.

61. Vicente I., Endemaño A., Garin X., and Brown M., “Comparative study of stabilising methods for adaptive speed sensorless full-order observers with stator resistance estimation,” IET Control Theory Appl., vol. 4, no. 6, pp. 993–1004, June 2010.

62. Yang G., and Chin T.-H., "Adaptive-speed identification scheme for a vector-controlled speed sensorless inverter-induction motor drive," IEEE Trans. Ind. Appl., vol. 29, no. 4, pp. 820-825, Jul./Aug. 1993.