Performance Investigation of ANN based Matrix Converter fed Induction Motor Drive

A DISSERTATION

Submitted in partial fulfillment of the

requirements for the award of degree

of

INTEGRATED DUAL DEGREE WITH B.TECH IN ELECTRICAL ENGINEERING

AND M.TECH IN POWER ELECTRONICS

BY

UNNAM ABHINAV CHOWDARY

11212015

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

ROORKEE – 247667

May 2016

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this report entitled "**Performance Investigation of ANN based Matrix Converter fed Induction Motor Drive**" in partial fulfilment of requirement for the award of degree of Integrated Dual Degree in Electrical Engineering with socialization in Electric Drives & Power Electronics, and submitted in the Department of electrical Engineering of Indian Institute of Technology Roorkee, India, is an authentic record of my own work carried out during the period from June 2015 to May 2016, under the supervision of Dr. Anubrata Dey, Assistant Professor, Department of Electrical Engineering, of Indian Institute of Technology Roorkee, India.

The matter embodied in this report has not been submitted by me for the award of any other degree of this or any other Institute/ University.

Date: 22^{nd} May, 2016 UNNAM ABHINAV CHOWDARY

Place: Roorkee

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of knowledge and belief.

Dr. ANUBRATA DEY

Assistant Professor

Department Of Electrical Engineering

Indian Institute of Technology, Roorkee

ROORKEE – 247667

ACKNOWLEDGEMENT

I wish to affirm my earnest acknowledgement to my respected guide Dr. Anubrata Dey, Department of Electrical Engineering, Indian Institute of Technology Roorkee, for their intuitive and meticulous guidance and perpetual inspiration in completion of report. I want to express my profound gratitude for his co-operation in scrutinizing the manuscript and his valuable suggestions throughout the work.

I would like to mention my parents for their endless support and encouragement and always believing and helping me to believe, that I can succeed at anything.

Acknowledgement would be incomplete without a word of gratitude to all my student friends for their timely help, encouragement and contribution in making it possible.

Dated: 22nd May, 2016 UNNAM ABHINAV CHOWDARY

Place: Roorkee Enrolment No. 11212015

ABSTRACT

Benefits of the AC motor over the DC motor and steady increase in industrial standards and automation where speed of rotation is of primary concern and its exercising control over its speed is the necessity has led to the development of various control strategies to efficiently control the speed of an AC motor. To implement the various control strategies, an efficient power modulator would be needed which could provide the necessary power to AC motor as desired by the control strategy.

The power modulators are expected to ensure they provide the power to the motor as desired as well as improve the quality of the power it takes from the grid. An overview of AC-AC regulators is done with emphasis on Matrix converters. Matrix converters and their various topologies are studied and simulated with.

For the induction motor controlling to achieve variable speed, Vector Control (VC) or Field Oriented Control (FOC) have become the industrial standard. The vector control technique decouples the two component of stator current: one is responsible for controlling torque and other is controlling flux independently as in the case of separately excited fully compensated DC motor.

Artificial Neural Networks or ANNs have been of special interest in the recent times, with their ability to learn instead of being programmed coupled with inherent ability to carry out parallel processing of data. This means they can fit highly non-linear functions and do calculations fast making them ideal tool in the control of induction motor drive. A novel neural controller has been studied which able to do slip calculations and matrix rotations in the original field oriented control of motor. A through study of the steps involved in the process is done along with simulations.

The analysis has been carried out on the basis of result obtained by numerical simulations. The simulation and evaluation of matrix converter and the neural network controller is performed using a fed three phase squirrel cage induction motor for a 2HP rating.

LIST OF ACRONYMS

OPERATOR	NAME
BJT	Bipolar Junction Transistor
JFET	Junction Field Effect Transistor
IGBT	Insular Bipolar Gate Transistor
MOSFET	Metal Oxide Field Effect Transistor
PWM	Pulse Width Modulation
MC	Matrix Converter
VSI	Voltage Source Inverter
CSC	Current Source Converter
IDF	Input Displacement Factor
DMC	Direct Matrix Converter
$\mathbf{i_{as}}, \mathbf{i_{bs}}, \mathbf{i_{cs}}$	Stator currents
$\mathbf{i_{ar}}, \mathbf{i_{br}}, \mathbf{i_{cr}}$	Rotor currents
$\mathbf{\bar{I}_s}, \mathbf{\bar{I}_r}$	Stator and rotor current vectors
$\mathbf{\bar{V}_s}, \mathbf{\bar{V}_r}$	Stator and rotor voltage vectors
$\mathbf{v_{as}}, \mathbf{v_{bs}}\ \mathbf{v_{cs}}$	Stator voltages
$\mathbf{i_{qs}}, \mathbf{i_{ds}}$	Quadrature and direct axis stator current component
$\mathbf{i_{qr}}, \mathbf{i_{dr}}$	Quadrature and direct axis rotor current component

$\omega_{\mathbf{r}}$	Rotor speed (rad/sec)
$\omega_{\mathbf{e}}$	Synchronous speed (rad/sec)
$\omega_{\mathbf{sl}}$	Slip speed (rad/sec)
ω_{base}	Base speed of motor
$\omega_{\mathbf{er}}$	Speed error between reference and actual
$\mathbf{i}_{\mathbf{mr}}$	Excitation current
$\Psi_{\mathbf{s}}$	Stator flux linkage
$\Psi_{\mathbf{ds}}, \Psi_{\mathbf{qs}}$	Direct and quadrature component of stator flux linkage
$\Psi_{\mathbf{r}}$	Rotor flux linkage
$\Psi_{\mathbf{dr}}, \Psi_{\mathbf{qr}}$	Direct and quadrature component of rotor flux linkage
$\boldsymbol{\gamma}_{\mathbf{sr}}$	Angle between stator and rotor flux in space
$\Theta_{\mathbf{fs}}, \Theta_{\mathbf{fr}}$	Angle between stator flux and d-axis and angle between rotor flux and d-axis
$\mathbf{L}_{\mathbf{ls}}, \mathbf{L}_{\mathbf{lr}}$	Leakage inductance of stator and rotor
$\mathbf{L}_{\mathbf{m}}$	Mutual inductance
$\mathbf{L}_{\mathbf{s}}, \mathbf{L}_{\mathbf{r}}$	Stator and rotor self-inductance
$\mathbf{R}_{\mathbf{s}}, \mathbf{R}_{\mathbf{s}}$	Stator and rotor resistance
$\boldsymbol{\tau}_{\mathbf{s}}, \boldsymbol{\tau}_{\mathbf{r}}$	Stator and rotor time constant
$\boldsymbol{\sigma} = \mathbf{1} - \frac{\mathbf{L}_{\mathbf{m}}^{2}}{\mathbf{L}_{\mathbf{s}}\mathbf{L}_{\mathbf{r}}}$	Total leakage factor
$\mathbf{T}_{\mathbf{em}}$	Electromagnetic torque
$\Theta_{\mathbf{e}}$	Position angle of synchronously rotating frame
J	Moment of inertia

List of Figures

Figure No.	Title of the Figure	Page No.
2.1	AC regulators topologies for three phase systems	6
2.2	AC Regulator design for single phase AC system	6
2.3	Diode bridge rectifier based DC link AC-AC Converter	7
2.4	Back to Back AC-AC converter	7
2.5	Current source converter (CSC) based DC link AC-AC Converter	8
2.6	Basic structure of 3×3 matrix converter scheme	9
2.7	Three phase-three phase matrix converter topology	10
2.8	Switching pulse duration for matrix converter	11
2.9	Output voltages fitting in input voltage waveform	12
2.10	Third harmonic additions to attain 0.8666 voltage gain	13
3.1	Space Phasor diagram of three phase Induction motor (synchronously rotating d-q reference frame attached with rotor flux vector).	17
3.2	Block diagram of Indirect Vector Control of Induction Motor Drive	19
4.1	Simple neuron without activation function	21
4.2	Neuron with weighted inputs	22
4.3	Three layer neural network architecture	22
5.1	Simulink structure of matrix converter with Venturini Topology	27
5.2	Simulink structure of matrix converter (3 × 3)	28
5.3	Simulink structure of bi-directional switch	29
5.4	Simulink structure of pulse generator in control block	29
5.5	Simulink model of gate pulse generator for matrix converter switches	30

5.6	Voltage and Current profiles at supply and load side at 50 Hz	31
5.7	Simulink Structure for matrix converter with induction motor in open loop fashion	32
5.8	Induction Motor response to 564V line to line voltage at 50 Hz	33
5.9	Power parameters at 564V line to line voltage and 50 Hz supply to IM	34
5.10	Induction Motor response to 564 V line to line voltage at 25 Hz	34
5.11	Power parameters at 564V line to line voltage and 25 Hz supply to IM	35
5.12	Induction Motor response to 282 V line to line voltage at 50 Hz	35
5.13	Power parameters at 282V line to line voltage and 50 Hz supply to IM	36
6.1	The System Design which is to be implemented	39
6.2	Basic simualtion model of indirect vector control of IM drive	40
6.3	Speed controller using PI control logic in MATLAB	42
6.4	(a) Estimation of i_{qs}* and i_{ds}*. (b) Estimation of slip speed. (c) Estimation of Θ_e and (d) Coordinate Transformation.	43
6.5	Sinusoidal PWM method to generate gate pulse for two level Inverter	44
6.6	Hysteresis current regulator to generate gate pule for inverter	45
6.7	Simulink diagram of neural network based field oriented control of induction motor	50
6.8	Simulation with load being 5 Nm and speed reference being 200 rad/s	51
6.9	Simulation with load being 3 Nm and speed reference being -200 rad/s	52
		55

List of Table & Code Blocks

Table No.	Table & Code Blocks Type	Page No.
5.1	Supply Parameters for matrix converter simulation	27
5.2	Machine Parameters for open loop matrix converter fed induction motor simulation	33
6.1(a)	Machine Parameters for induction motor simulation with neural network controller	45
6.1(b)	Script to randomize torque and speed references	46
6.2	Script to pre-process alpha current values	46
6.3	Script to prepare data for writing to folder	47
6.4.1	Neural network training code up to normalization	48
6.4.2	Neural network training code up-to Simulink block generation	49

CONTENTS

CONDIDATE’S DECLARATION			i
ACKNOWLEDGEMENT			ii
ABSTRACT			iii
LIST OF ACRONYMS			iv
LIST OF FIGURES			vi
LIST OF TABLE & CODE BLOCKS			viii
CHAPTER	1	Introduction	1
	1.1	General	1
	1.2	Literature Review	2
	1.2.1	Review of Matrix Converters	2
	1.2.2	Review of Vector Control	3
	1.2.3	Review of Neural Network Controller for Induction Motor Drive	4
	1.3	Objective of Thesis	4
CHAPTER	2	AC-AC power converters and their topologies	5
	2.1	AC Voltage Converters	5
	2.2	DC Link AC-AC power converters	7
	2.3	Matrix Converter and its topologies	8
	2.3.1	Direct Matrix Converters	9
	2.3.2	Modulation Strategies for Direct Matrix Converter	10
	2.3.3	Alesina Venturini Basic Method(AV-method-1980)	10
	2.3.4	Alesina Venturini Basic Method(AV-method-1989)	13
	2.3.5	Scalar Modulation Strategies	13
CHAPTER	3	Induction Motor Drive and Speed Control	15
	3.1	Concept of Vector Control for an Induction Motor	15
	3.1.1	Vector Control Modelling of an Induction Motor	16
	3.1.2	Indirect Vector Control of an Induction Motor	19
CHAPTER	4	Artificial Neural Networks	20
	4.1	Concept of Artificial Neural Networks	20

	4.1.1	Levenberg-Marquardt Algorithm	24
CHAPTER	5	Matrix Converter Simulations	27
	5.1	Simulink Model of Matrix Converter	27
	5.2	Simulation Results with Passive Load	31
	5.3	Simulation Results with Induction Motor in Open Loop	32
CHAPTER	6	Artificial Neural Networks Based Induction Motor Drive Simulations	37
	6.1	System Design & Description	37
	6.2	Implementation Methodology	39
	6.2.1	Simulation of Indirect Vector Control Model	40
	6.2.2	Data Collection Methodology	45
	6.2.3	Neural Network Training Methodology	46
	6.3	Simulation of Neural Network Controller based IFOC of an Induction Motor	50
CHAPTER	7	Conclusion	53
REFERENCES			54

CHAPTER 1

Introduction

1.1 General

From times of 1856 when Werner Siemens invented T-armature winding generator to power telegraph machines to present 2016 when the roads are seeing a 259 HP electric car by Tesla Motors, electrical machines aka motors have come a long way.

The versatility of their usage has risen exponentially, apart from generation of heat, light and sound, motion is an important component of modern world given that elevators, escalators etc. are now an inevitable part of modern lifestyle.

Electric motors create motion using electrical power through the electromagnetism principle. They can be classified as DC motors or AC motors on the basis of the kind of electrical power used to power the motion. DC motors were invented before AC but AC motors due to ease in simple design, robustness, low losses have gained an upper hand. However their control is complicated due to their nonlinear torque-speed characteristics.

To extract the maximum utility out of a motor, control over speed for different load conditions (torque) is of necessity. This can be attained by controlling the power input but since power source in India and abroad has a standard set of values such as 230V, 50 Hz, 1Φ supply, there is a need for efficient power converters or modulators.

A power modulator is a non-linear resistive network which changes the available input power to the desired output, which in turn runs the load or in this case the motor. By controlling or changing the parameters of the network we can control the output power. The power converter or a modulator consists of an intrinsic network of semi-conductor switches which are controlled using digital, analog power circuits.

Invention of power devices marked the beginning of power semi-conductors starting with thyristors in 1957, Bipolar(BJTs) in 1960, Junction Field effect transistors(JETs) in 1970s, Metal oxide semiconductor field effect transistors(MOSFETs) in 1976 and Insulated gate bipolar transistors(IGBTs) in 1982[1]. Rise in power rating has resulted in decreased costs, size with increase in switching frequency, higher reliability and robustness.

1.2 Literature Review

This section mainly deals with the review of research work already been carried out in the fields of matrix converters and applications of ANN in control of induction motor drive mainly in form indirect vector control. Emphasis is primarily on these two components of the project, not much has been studied as far as vector control of induction motor is concerned.

1.2.1 Review of Matrix converters

The matrix converters are a forced commutation based AC- AC direct converter which are an intricate array based system of controlled bidirectional switches as the primary power elements used to create a variable output voltage system with flexible frequency. The system does not need any kind of energy elements since it does not have any dc-links.

A major and important component of a matrix converter is the fully controllable four-quadrant bidirectional switch, which allows for high-frequency operation. Most of the early work was dedicated towards unrestricted frequency changers which used thyristors which were forced commutated with help of external circuits to implement the bidirectional controlled switch [5],[9]–[10],[16]. With this solution, the power circuit was bulky and the performance was poor.

With the introduction of power transistors for implementation of the bidirectional switches, matrix converter topologies have risen in popularity [11]–[15]. However, the real credit for the development of topologies for these converters goes to a paper based on the work of Venturini and Alesina published in 1980 [17], [18]. The term matrix converter was coined by them and it was presented as a matrix of bi-directional switches in its power circuit. The most fundamental contribution being in the development of a rigorous mathematical analysis describing the low-frequency behaviour of the converter, and having also introduced the "low-frequency modulation matrix" concept. The direct transfer scheme under which it was developed involves multiplication of the modulation values directly with the input voltages.

Rodriguez in 1983 [19], introduced the concept of "fictitious dc link" which led to further work in this direction. The switching arrangement in this method is designed in such a way that each output line is switched between the most negative and positive of the input lines using a pulse-width modulation (PWM) technique, typically used in the standard voltage-source inverters (VSIs). Thereby being called the "indirect transfer function" approach [20]. In 1985–1986, Ziogas et al. published [21] and [22], which further improved on the "fictitious dc link" work of Rodriguez and backed it up with a rigorous mathematical explanation. In 1983, Braun [23], and in 1985 Kastner and Rodriguez [24], introduced the use of space vectors for analysis and control of matrix converters. In 1989, Huber et al. published the first of a series of papers [25], [41]–[45] which formed the basis for use of space-vector modulation (SPVM) to matrix converter modulation problem [27].

Kastner and Rodriguez confirmed through experimental means in 1985 [24] and Neft and Schauder in 1992 [26] did back the idea that a matrix converter involving only nine switches can be an effective means in the vector control of an induction motor with high quality input and output currents.

1.2.2 Review of Vector Control

F. Blaschke [51] introduced the vector control scheme in the year1972. The method revolutionized Induction motor drive and its speed control through variable frequency which gave excellent dynamics along with steady state performance similar to the separately excited fully compensated dc motor. Before VC or Field Oriented Control (FOC), scalar control techniques of induction motor [52]-[55], and [56] were being used. The transient behaviour obtained from these techniques were not up to the mark [52]-[55] thus the requirement for a scheme which could give good transient response.

The FOC method tried to take the advantage of both the machine i.e. constructional feature of IM as well as the controlling feature of a separately excited fully compensated DC motor [This methodology provides an independent control between flux and torque, and converts a non-linear control system into a linear one [57]

The Mathematical modelling of IM was based on the foundations of several developed theories [62]. Improvements in Power Electronics area [63] has resulted in real time execution of both the technique of the FOC [64]-[65].The use of digital signal processors (DSPs) [66]-[67]. Microprocessors have made it possible to implement complex algorithm like FOC at a sensible cost. MATLAB with its toolboxes and Simulink has made the simulation and modelling of FOC simpler and iterative improvement faster.

The control aspect of the Voltage Source Inverter for both the techniques, has been handled quit differently. In FOC, popularly also known as Current Control VSI (CCVSI) are used with varied current regulating topologies [61], such as Sinusoidal PWM, synchronous dq frame PI regulator, stationary frame PR, stationary frame PI and hysteresis current regulators etc. Hysteresis Current control method of VSI offers an effective and unique transient response in contrast with other analog and digital schemes, making it ideal method for cases where high accuracy, wide bandwidth, and robustness are essential [59].

Hysteresis control technique is fundamentally an analogic technique. Despite the merits of the digital controllers, in term of integration interfacing, flexibility, and maintenance their accuracy and speed of response are often insufficient for current control in some challenging applications, such as active filters and high- precision drives [58], [59]. In all of the applications, reference current waveform with high harmonic content and fast transient must followed by good accuracy. In these belongings, the hysteresis technique can be a fine solution [58], [60], provided some improvement are introduced to overcome its main limitations, which are sensitivity to phase commutation interference and switching frequency. In this scenario, a variety of provisions both digital and analog, have been proposed [58].

1.2.3 Review of Novel Neural Network Controller for Induction Motor Drive

Induction motor owing to its rugged design, reliability and low cost has been the motor of choice for wide set of industrial and domestic applications. This has been the basis for the development of several control techniques for induction motor control which now includes the popular V/f control[36], field oriented control[37-39], direct torque control(DTC), speed sensor less control [40]. Recently there has been a spike in interest with regards to application of neural networks to various aspects of induction motor control such as adaptive control[41],sensor-less speed control[42]-[45],inverter current regulation[46]-[48],motor parameter identification purposes[49],[50].

1.3 Objective of Thesis

The objective of the thesis has been to study the matrix converter and its application in control of induction motor drive. Along with it a thorough study of Artificial Neural Networks and their applications in control of induction motor drive through indirect vector control is carried out.

1. Study of matrix converters and their various topologies namely direct topology ie. Venturini Control method and their simulation in MATLAB Simulink.
2. Study and understand the concept of indirect vector control of induction motor drive and simulate it using MATLAB Simulink.
3. Study the concept of Artificial Neural Networks (ANN) and their applications in the in control of induction motor drive by modifying the indirect vector control scheme.
4. Simulate modified indirect vector control scheme with neural network controller using MATLAB Simulink.

1.3 Thesis Organization

The thesis has been divided into 7 chapters with core content spread across 5 chapters from chapter 2 to 6.

Chapter 2 contains the theory behind AC-AC converters mainly matrix converter and their various topologies.

Chapter 3 explores the theory behind vector control of induction motor with emphasis on indirect vector control

Chapter 4 goes into the working of ANN, theory behind their working, how to train and some salient points when doing so.

Chapter 5 has simulations related to matrix converter, it has matrix converter simulation with passive load and induction motor drive in an open loop.

Chapter 6 has simulation of indirect vector control of an induction motor and modified vector control with neural controller.

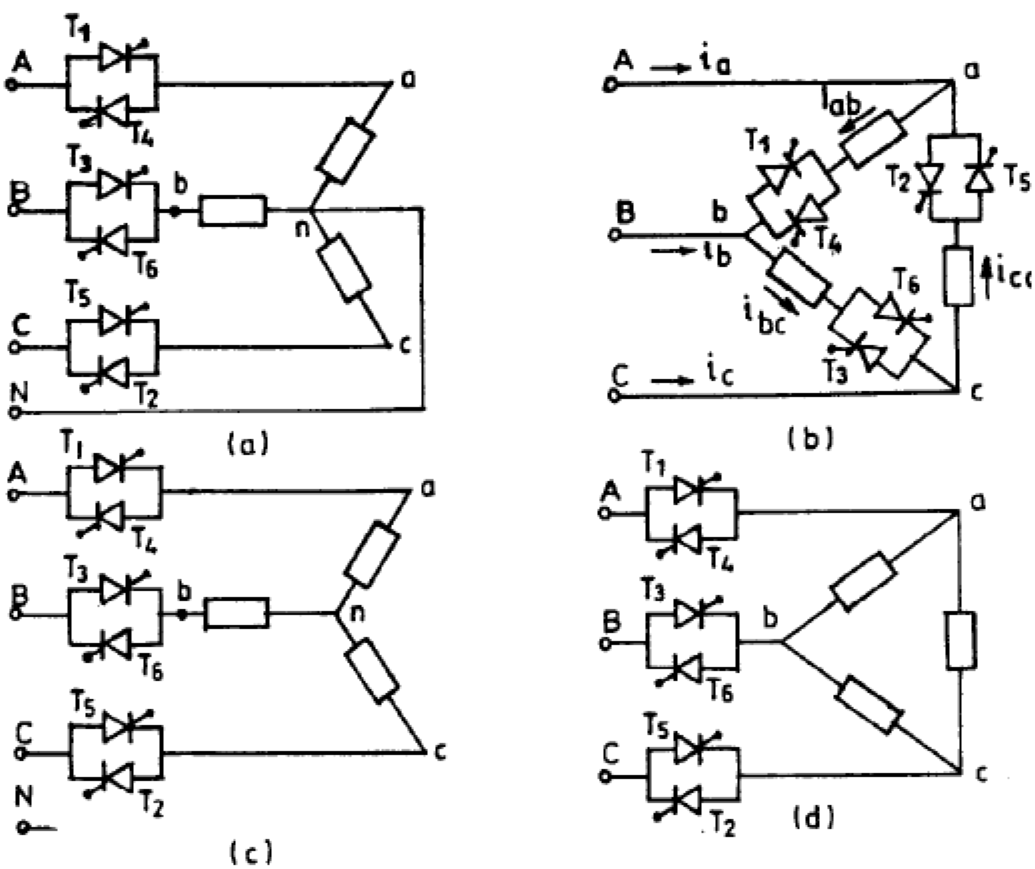
CHAPTER 2

AC-AC power converters and their topologies

AC-AC power converters alter the output power parameters at a particular AC power input. Output voltage, frequency or both can be altered at a particular voltage and frequency. These power converters can be labelled into two categories depending on the whether the frequency can be modified or not. AC converters which allow modification of both frequency and amplitude are AC-AC converters while those which allow only amplitude change are called AC-regulators.

2.1 AC voltage controllers

AC voltage regulators provide a control over the output voltage where the frequency of the output voltage remains the same. The various applications are lighting control, speed control, started to induction motor, heat regulation etc. The topologies for single phase, three phase voltages and regulators for both delta and star connection is shown in figure 2.1



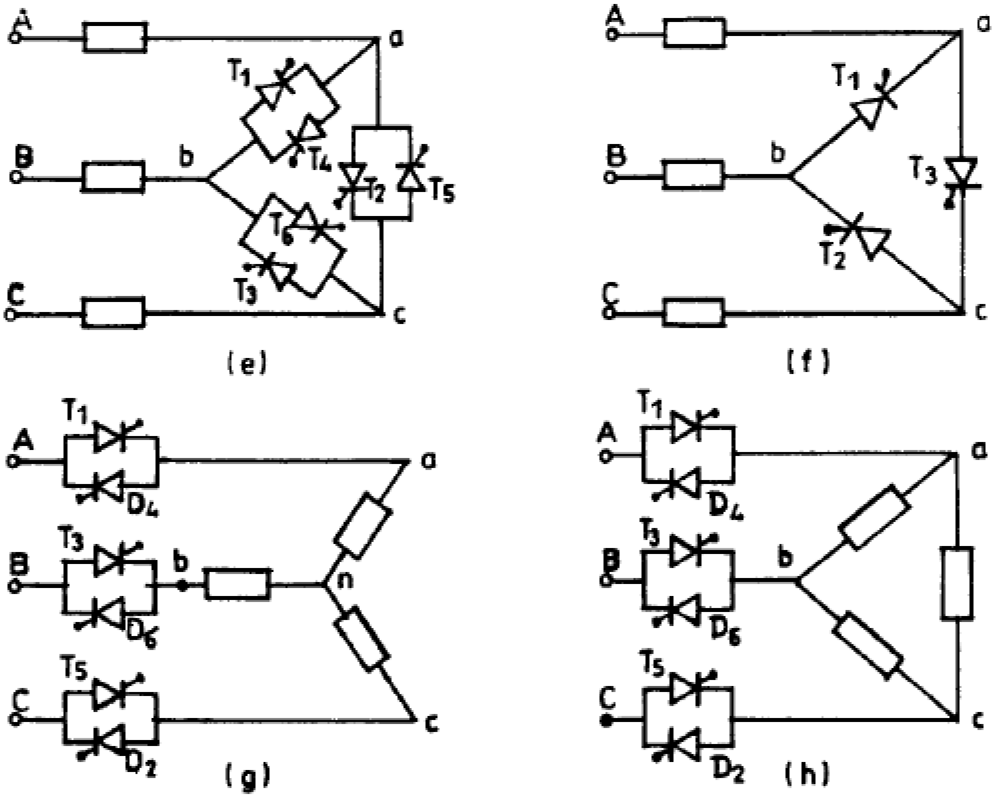


Figure 2.1 AC regulators topologies for three phase systems [1]

The advantage is simpler design, less bulky, cheaper and less complex in terms of strategy, though voltage gain is limited to maximum 1 and no modification can be made to output frequency. Voltage harmonics injection is another issue.

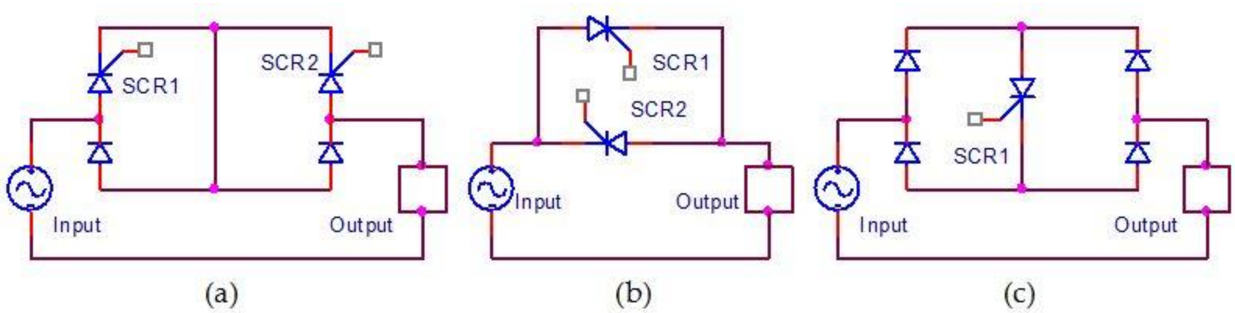


Figure 2.2 AC Regulator design for single phase AC system [2]

2.2 DC link AC-AC power converters

DC Link AC-AC Converters firstly convert power in AC to form DC using rectifier and back to AC using inverter. This involves an energy storage element in form of a capacitor for voltage source inverters (VSI) and inductors in case of current source inverters (CSI).

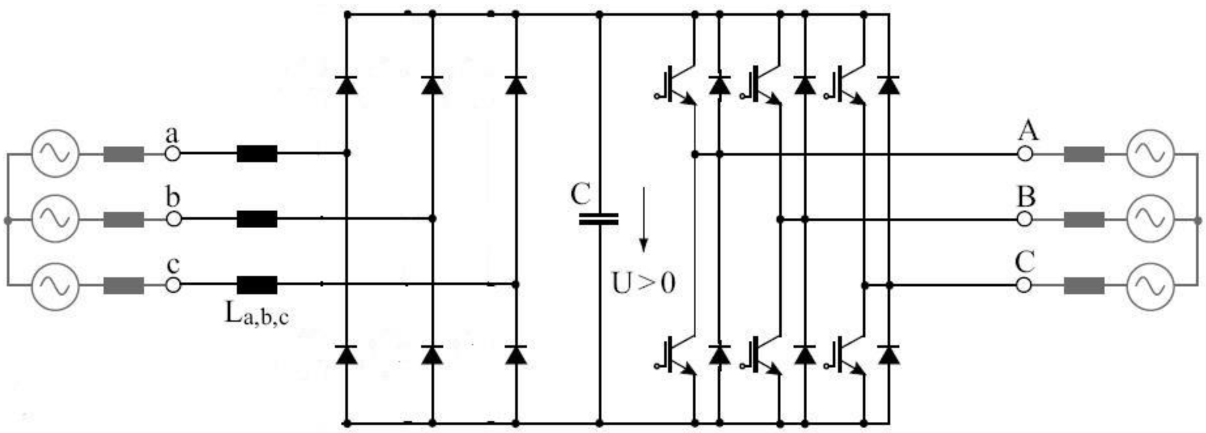


Figure 2.3 Diode bridge rectifier based DC link AC-AC Converter [3]

A rectifier converts the AC voltage into DC, then a PWM inverter converts DC power to AC power of desired voltage and frequency at the output side. The design is simpler in the sense that the rectifier side involves no control. But this introduces new harmonics from the input side and also into the electrical supply. The other disadvantage is the lack of bi-directional power flow.

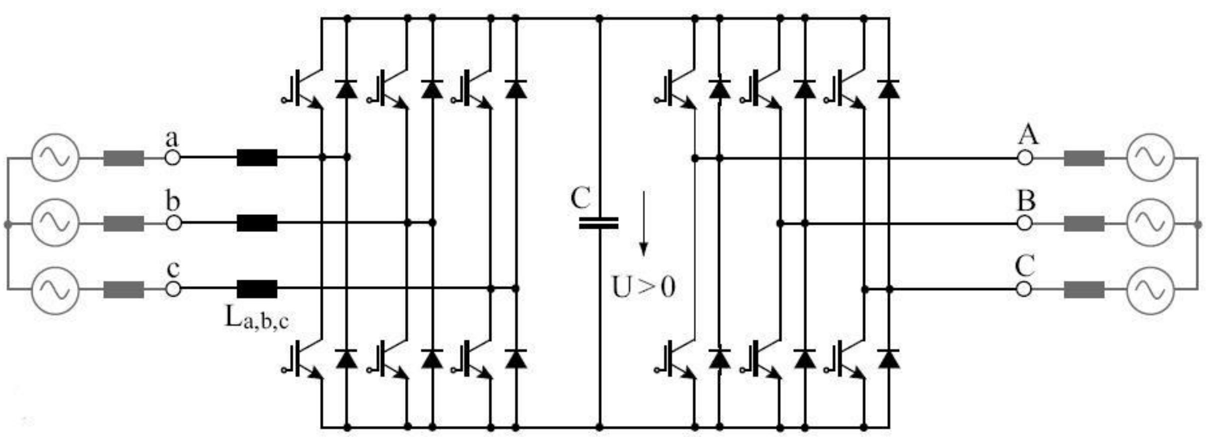


Figure 2.4 Back to Back AC-AC converter [3]

To deal with this issue, a new structure of back to back converters was proposed which involved PWM based rectifier as well as inverter. This allowed for regenerative operation but came at the cost of complexity, bulkiness and cost due to active devices.

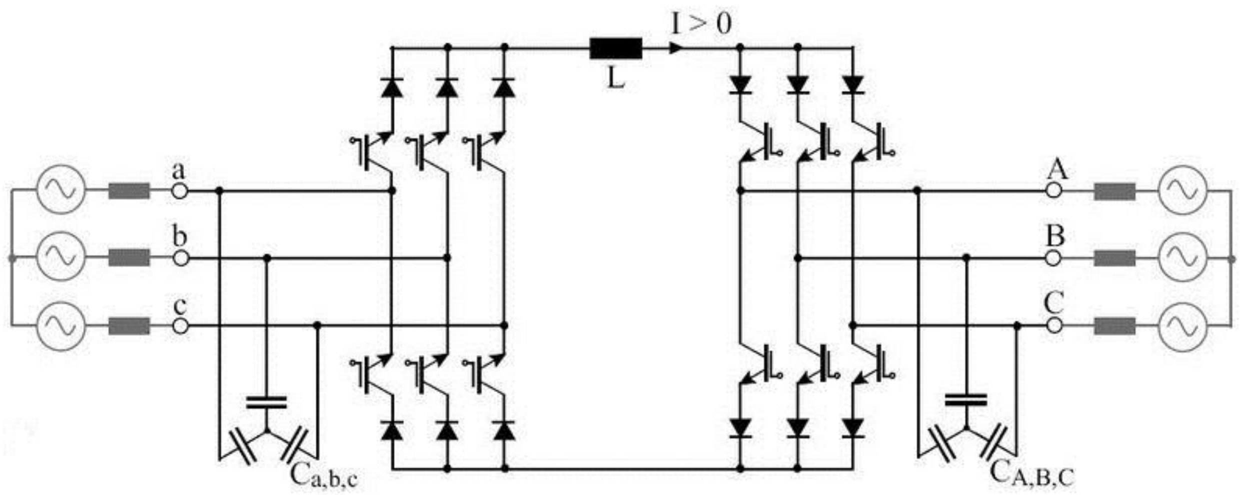


Figure 2.5 Current source converter (CSC) based DC link AC-AC Converter

The CSC based DC link AC-AC converters are used in high power applications. But as inductor is bulky and suffers from operational losses, CSC are less popular in comparison to VSI.

In VSI, voltage is controlled and current depends on it and the load impedance while in case of current source converter, load is controlled and voltage is function of current and impedance.

The major disadvantage with indirect AC-AC converter topology is the need for large energy storage element in the DC-link. Typical VSIs have an electrolytic capacitors in their DC-link with a shorter life compared to AC capacitor decreasing the overall life of the converter and involving large maintenance cost. The large size results in a bulky circuit which are unreliable at higher temperatures, thus making these topologies in-appropriate in sensitive applications.

Thus the need for direct AC-AC converters without involving any DC links.

2.3 Matrix Converter and its topologies

Gyugi introduced matrix converters in the year 1970[5] as a complete silicon solution to the AC-AC conversion. It gives ability to vary both frequency and magnitude of the output voltage. This gives it an ability to operate in AC-DC, DC-DC and DC-AC three phase converter forms. All this is achieved with minimal harmonic injection and bi-directional power flow. Certain control topologies enable control over the input displacement factor and since it has storage elements, it is not bulky.

However, the output voltage is restricted to a maximum voltage gain. The direct conversion involves more switches than other structures. Also it is susceptible to disturbances in the input side.

2.3.1 Direct Matrix Converter

A direct matrix converter between n input phases and m output phases is a $m\times n$ bi-directional switches such that each of the n input phases are connected to the m output phases through a bi-directional switch.

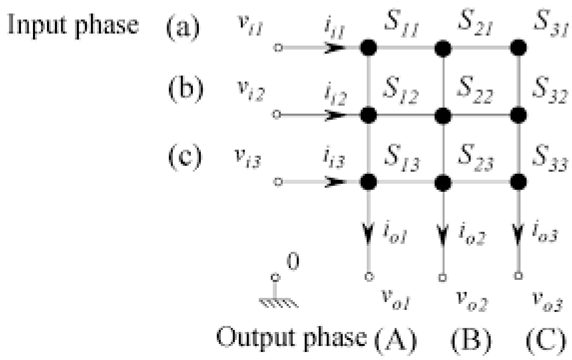


Figure 2.6 Basic structure of 3×3 matrix converter scheme

A DMC generates m output voltages by extracting voltage intercepts from all of n input voltages in a certain sampling period. This leads to pulsed output voltage with no limitation on output frequency.

For a 3×3 matrix, we have 9 bi-directional switches. This gives us 2^9 switching states. During the operation of a matrix converter, the following needs to be taken care of: no two input phases should be connected as this will lead to a high short circuit current, which might destroy the semi-conductor devices. Also any of the output phases should not be left open as due to the presence of inductive load, a high change in current can lead to extreme $\frac{di}{dt}$ values. The two limitations result in only 27 feasible switches states.

In-order to generate better input current and output voltage profiles, the switches are operated at a very high frequency. The switching period is the same one as the one where output voltage is average value obtained from available input voltages. The duty cycle for each switch is given as $mkj = tkj/tsw$, where tsw is the switching period out of which switch is ON for tkj period.

These limitations of the open and short circuits lead to condition where in at any point of time one of the output should be connected to an input and each output can only be connected with one input.

$$0 \le mkj = \frac{t_{kj}}{t_{sw}} \le 1; \text{ k = A, B, C, j= a, b,c;}$$

$$\text{M (t)} = \begin{matrix} m_{Aa}(t) & m_{Ba}(t) & m_{Ca}(t) \\ m_{Ab}(t) & m_{Bb}(t) & m_{Cb}(t) \\ m_{Ac}(t) & m_{Bc}(t) & m_{Cc}(t) \end{matrix}$$

So the input currents and output voltages in every sample period become

$$Vo = M\ (t) * Vi;$$
$$Ii = M\ (t)\ T * Io;$$

V_o, I_i are the output voltages and input current. V_i, I_o are the input voltages and output current. M (t) is the modulating matrix and M (t) ^{T} is the transpose of modulating matrix.

2.3.2 Modulation Strategies for Direct Matrix Converter

The matrix converter and the topology being discussed is given by the following figure.

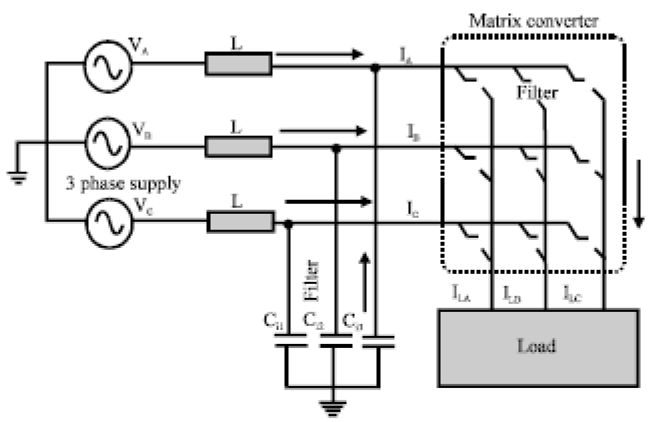


Figure 2.7 Three phase-three phase matrix converter topology [6]

2.3.3 Alesina Venturini Basic Method (AV-method-1980)

In 1980, detailed mathematical low frequency analysis of matrix converter resulted in Alesina and Venturini coming up with the first modulation strategy for matrix converter. This strategy determines the duty cycles for switches based on mathematical solutions. This direct transfer mechanism enables the converter to generate output voltages by sequentially applying input voltages to respective output terminals for certain period during the switching time. The voltage transfer ratio obtained is 0.5. This direct displacement factor doesn't depend on the output displacement factor.

Switching functions, S_{kj} are used to determine the input currents and output voltages at any instant of time. They are either 1 or 0 indicating that the switch is OFF or ON respectively. The switching variables S_{kj} determines the status of connection between input phase 'k' and output phase 'j' [49]

The output voltages and input currents are given by matrix:

$$\begin{matrix} V_a(t) \\ V_b(t) \\ V_c(t) \end{matrix} = \begin{matrix} S_{Aa}(t) & S_{Ba}(t) & S_{Ca}(t) \\ S_{Ab}(t) & S_{Bb}(t) & S_{Cb}(t) \\ S_{Ac}(t) & S_{Bc}(t) & S_{Cc}(t) \end{matrix} \times \begin{matrix} V_A(t) \\ V_B(t) \\ V_C(t) \end{matrix}$$

$$\begin{matrix} I_A(t) \\ I_B(t) \\ I_C(t) \end{matrix} = \begin{matrix} S_{Aa}(t) & S_{Ba}(t) & S_{Ca}(t) \\ S_{Ab}(t) & S_{Bb}(t) & S_{Cb}(t) \\ S_{Ac}(t) & S_{Bc}(t) & S_{Cc}(t) \end{matrix} \times \begin{matrix} I_a(t) \\ I_b(t) \\ I_c(t) \end{matrix}$$

The voltage and current stiff sides should not be shorted. The conditions imposed are:

$$S_{Aj} + S_{Bj} + S_{Cj} = 1,\ j = a,b,c$$

The switching pulse generation pattern is as shown in figure

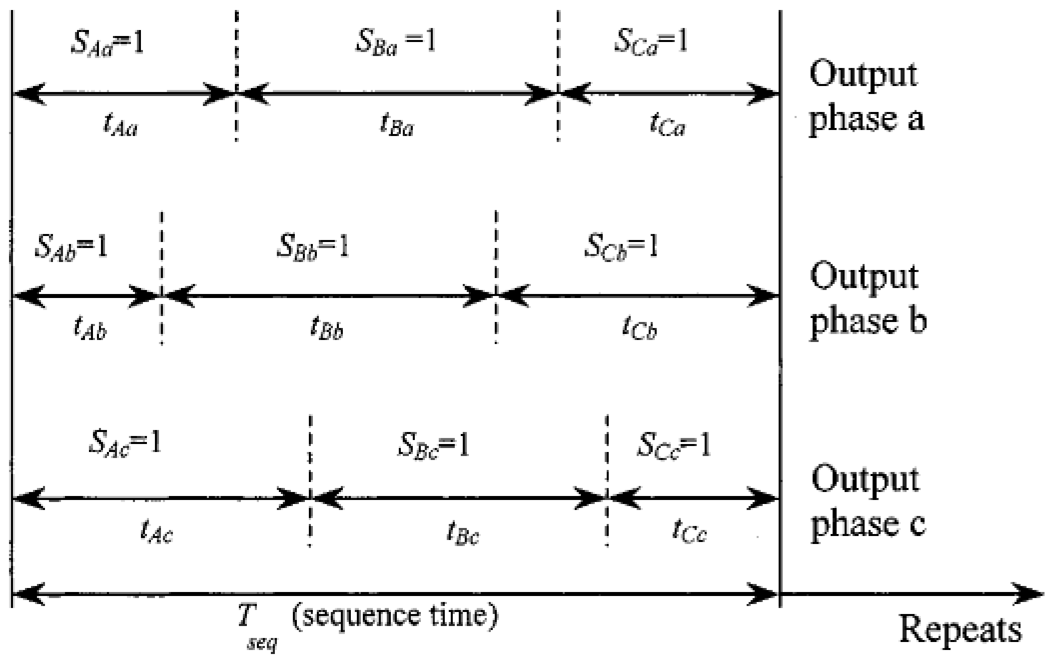


Figure 2.8 Switching pulse duration for matrix converter [7]

The scheme involves using the input voltage and output currents which are obtained to generate output voltages and input currents.

$$V_i(t) = \begin{matrix} V_i\cos(w_i\ t) \\ V_i\cos(w_i t - \frac{2\pi}{3}) \\ V_i\cos(w_i t + \frac{2\pi}{3}) \end{matrix}$$

$$i_o(t) = \begin{matrix} I_o\cos(w_o\ t + \Phi_o) \\ I_o\cos(w_o\ t - \frac{2\pi}{3} + \Phi_o) \\ I_o\cos(w_o\ t + \frac{2\pi}{3} + \Phi_o) \end{matrix}$$

These are used to synthesize the output voltages and input currents

$$V_o(t) = \begin{matrix} qV_i\cos(w_i\ t) \\ qV_i\cos(w_i t - \frac{2\pi}{3}) \\ qV_i\cos(w_i t + \frac{2\pi}{3}) \end{matrix}$$

$$i_i(t) = \begin{matrix} I_i\cos(w_i\ t + \Phi_i) \\ I_i\cos(w_i\ t - \frac{2\pi}{3} + \Phi_i) \\ I_i\cos(w_i\ t + \frac{2\pi}{3} + \Phi_i) \end{matrix}$$

On doing active power balancing on both sides we get,

P = 3*$ViIicos(\Phi i)/2$ = 3*$VoIocos(\Phi o)/2$

Ii = $q * Io * cos(\Phi o)/cos(\Phi i)$

And solving the equations

$$\begin{aligned} Vo &= M(t) * Vi\,; \\ Ii &= M(t)T * Io; \end{aligned}$$

For unity power factor, the modulation index for each switch can be computed using the equation below

$$m_{kj} = \frac{tkj}{Tseq} = \frac{1}{3}[1 + \frac{2*V_k*V_i}{V_m^2}]$$ for K = A, B,C and j = a,b,c

Based on the above analysis, the average output is equal to the target output voltage in any of the switching periods. At any instant the output voltage is equal to input side peak value and hence must fit within the upper and lower envelopes formed by three phase voltages. Thus a limitation of 50 % of input voltage is obtained.

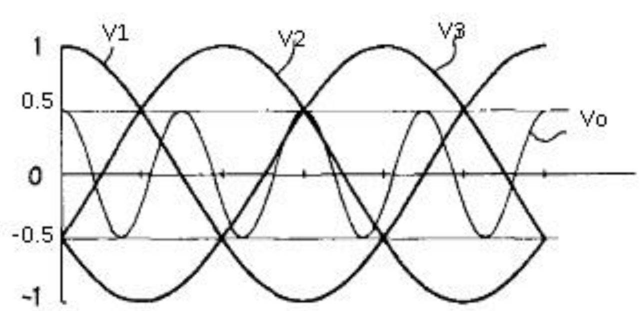


Figure 2.9 Output voltages fitting in input voltage waveform [8]

2.3.4 Alesina Venturini Basic Method (AV-method-1989)

In 1989, the original method was revised to increase the voltage transfer limit to 86.66% by optimizing the modulation strategy. This was done by adding third harmonics of the supply frequency and the output frequency to the reference output voltage [51]. This way the third harmonic components would be cancelled in a 3 phase system. This added transfer limit comes at the cost of increased circuitry and control.

The reference output voltage now becomes

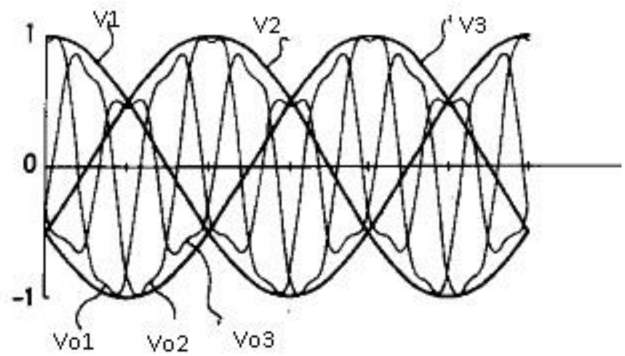


Figure 2.10 Third harmonic additions to attain 0.8666 voltage gain [8]

The reference output voltage now becomes

$$V_o(t) = qV_m \times \begin{matrix} \cos(w_o\ t) - \frac{1}{6}\cos(3w_o\ t) + \frac{1}{2\sqrt{3}}\cos(3w_i t) \\ \cos(w_o t - \frac{2\pi}{3}) - \frac{1}{6}\cos(3w_o\ t) + \frac{1}{2\sqrt{3}}\cos(3w_i t) \\ \cos(w_o t + \frac{2\pi}{3}) - \frac{1}{6}\cos(3w_o\ t) + \frac{1}{2\sqrt{3}}\cos(3w_i t) \end{matrix}$$

And thereby modulation index changes to

$m_{Kj} = \frac{1}{3}[1 + \frac{2V_{K}V_j}{V_m^2} + \frac{4q}{3\sqrt{3}}\sin(w_i t + \beta_K)\sin(3w_i t)]$ for K = A, B, C and j = a, b,c and

β_K = 0, 2π/3,4π/3 for K = A,B,C respectively

2.3.5 Scalar Modulation Strategy

In 1975, scalar modulation strategy was proposed, it produces results similar to the Venturini algorithm. This strategy can produce any displacement factor in the input side independent of the output power factor [60]. It can be increased upto 0.866 using reference output voltage as in AV method, 1989.

$$Vo = (tkVk + tLVL + tMVM) / Ts$$

$$tk + tL + tM \ = \ Ts$$

The modulation index is calculated as

$m_{Lj} = \frac{(Vj - V_M)V_L}{1.5V_m^2}$, $m_{Kj} = \frac{(V_j - V_M)V_K}{1.5V_m^2}$, m_{Mj} = 1-($m_{Lj} + m_{Kj}$) for j = a,

Vm is the input voltage which has a different polarity from the other two and $V_{KZ,}$ V_L has a value of smaller and larger magnitude among the two input voltages.

CHAPTER 3

Induction Motor Drive and Speed Control

3.1 Concept of Vector Control for an Induction Motor

Vector control is a means to control an AC Drive but involves added complexity over scalar control. The modelling of the motor in a d-q form is the fundamental basis of the vector control model. The vector control is a powerful and often used control technique that permits the use of induction and synchronous motor for high performance scenarios. This high performance control allows for smooth rotation over the entire speed of the motor, torque control at zero speed, fast acceleration and de-acceleration. Vector control in an AC model takes the rotor flux as reference while the stator current is decomposed into two orthogonal components. The d-component or the magnetic excitation component and a q-component which is orthogonal to the d-component. Vector control is fundamentally the independent control of magnitude and phase of the stator current vectors so as to meet the torque and speed requirements of the motor. Thus enabling control over parameters over which a direct control is not possible with other schemes. The currents id and iq of stator current in a synchronous rotating frame are analogous to the field current I_f and armature current Ia of the DC machine, the torque can be expressed as

$$T = Kf * If * Ia = Kf * Id * Iq$$

The dynamic modelling of the IM is needed to get a better understanding of the vector control model. The first step is the conversion of the 3-phase quantities into 2-axes system called d-axis and q-axis. This d-q frame can be both rotating as well as stationary. The most popular frame being the synchronous frame where the d-axis is aligned with rotor flux from analysis point of view. All this works only when the three phase system is balanced.

Dynamic performance of motor undergoes severe performance degradation when the flux magnitude and phase angle undergo deviation from their set values. The stator schemes only account for the magnitude changes ignoring the phase angle, thus bringing in oscillations in torque and speed. Large scale flux changes also involve large stator currents whose burden falls upon the converter switches which is not economical.

Field and armature windings in a DC machine make it easier to control the two components independently. The vector control strategy for AC machine are inspired from this phenomenon. It involves decoupling of stator currents into two components.

The modelling of vector control involves converting the 3-phase system into the two phase systems of either d-q or α-β quantities. This conversion involves knowing an accurate value of the speed and angle measurements, making this a dynamic process. Vector control algorithms require position and magnitude of rotor flux linkages phasor 'λ_r'. θ_f is the field angle made by the rotating λ_r with respect to a stationary/stator reference frame.

Since the two phase frame of reference, d axis represents the rotating flux while the q-axis represents the torque, by independently controlling the two components, we can control torque and speed.

3.1.1 Vector Control Modelling of an Induction Motor

The vector term in the vector control methodology originates from independent control of both magnitude as well as phase angle (stator mmf). Vector control, which is also known as Field Oriented Control (FOC) involves aligning of the stator flux or rotor flux or mutual (air gap) flux with synchronously rotating reference frame (d-q frame).

A space phasor of induction motor has been shown in fig.3.1. Here, reference frame (d-q) is chosen in such a way that, it is aligned along the flux vector and is rotating with the synchronous speed (rotor flux also rotates with synchronous speed). The alignment can be done along stator flux vector or air gap flux vector but they intensify the complexity of algorithm as this alignment doesn't decouple flux and torque []. A decoupling block is needed to decouple both the flux and torque or their components.

Let us first take rotor flux attached to the d_s^e-axis,

$\Psi_{dr}^* = \Psi_r = \text{Constant}$ (3.1)

Or, $\Psi_{qr}^* = 0$ (3.2)

For further analysis, some mathematical equations of induction motor drive with respect to synchronously rotating d-q axis ($d_s^* - q_s^*$ axis),

d^* Rotor axis

$V_{dr}^* = 0 = r_r\, i_{dr}^* + p\, \Psi_{dr}^* - \omega_{sl}\, \Psi_{qr}^*$ (3.3)

From equation (3.1) – (3.3)

$0 = r_r\, i_{dr}^*$

Or, $i_{dr}^* = 0$ (3.4)

q^* Rotor axis

$V_{qr}^* = 0 = r_r\, i_{qr}^* + p\, \Psi_{qr}^* + \omega_{sl}\, \Psi_{dr}^*$

Or, $0 = r_r\, i_{qr}^* + \omega_{sl}\, \Psi_r$

Or, $\omega_{sl} = -\frac{r_r\, i_{qr}^*}{\Psi_{dr}^e}$ (3.5)

Though rotor parameter are not easily available, thus above equation has to be broken into different stator parameters

$\Psi_{qr}^* = 0 = L_r\, i_{qr}^* + L_m\, i_{qs}^*$

Or, $i_{qr}^* = -\frac{L_m}{L_r}\, i_{qs}^*$

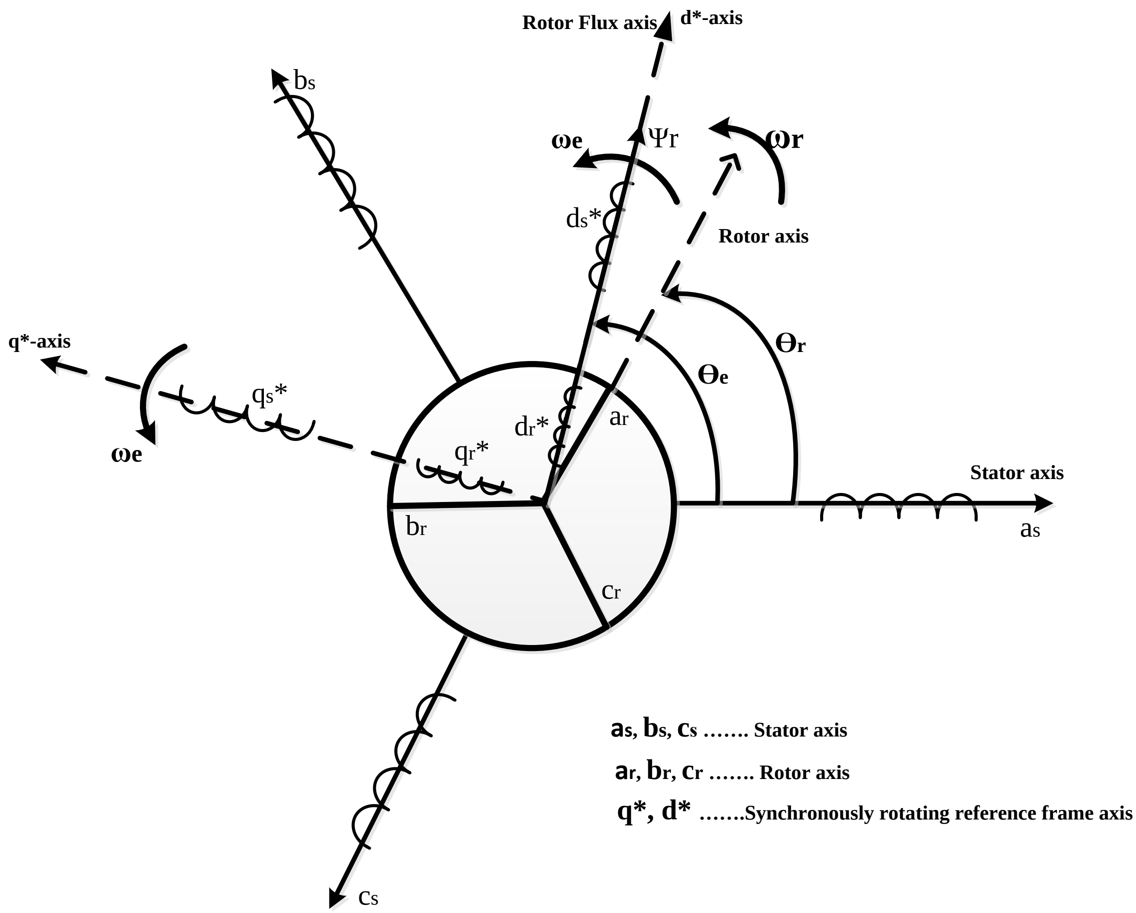


Fig.3.1 Space Phasor diagram of three phase Induction motor (synchronously rotating d-q reference frame attached with rotor flux vector).

Put the above relation in the equation (3.5) and we'll get,

$$\omega_{sl} = \frac{L_m}{L_r/r_r}\frac{i_{qs}^*}{\Psi_{dr}^*}$$

Or,

$$\omega_{sl} = \frac{L_m}{\tau_r}\frac{i_{qs}^*}{\Psi_{dr}^*} \qquad (3.6)$$

The above equation of slip speed is in the term of rotor flux and quadrature component of stator current.

As, $\Psi_{dr}^* = L_r\, i_{dr}^* + L_m\, i_{ds}^*$

From the equation (3.4)

$$\Psi_{dr}^* = L_m\, i_{ds}^* \qquad (3.7)$$

Put this relation in equation (3.6), we'll get the slip equation in term of d and q axis (synchronously rotating) component of stator current.

$$\omega_{sl} = \frac{1}{\tau_r}\frac{i_{qs}^*}{i_{ds}^*} \qquad (3.8)$$

Electromagnetic torque

The electromagnetic torque of induction motor in terms of $d^* - q^*$ reference frame components is shown in equation (3.9),

$$T_{em} = \frac{3}{2}\frac{P}{2} L_m (i_{qs}^* i_{dr}^* - i_{ds}^* i_{qr}^*) \qquad 3.9)$$

From the above equation containing the rotor current components, all quantities have to be written in terms of stator components.

$$i_{dr}^* = \frac{\Psi_{dr}^* - L_m i_{ds}^*}{L_r} \qquad (3.10)$$

$$i_{qr}^* = \frac{\Psi_{qr}^* - L_m i_{qs}^*}{L_r} \qquad (3.11)$$

By using equation (3.2), (3.9), (3.10) and (3.11), now the electromagnetic torque will be,

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i_{qs}^* \Psi_{dr}^* \qquad (3.12)$$

$$\Psi_{dr}^* = L_m i_{ds}^* \qquad (3.13)$$

From the equation (3.12) and (3.13), we can say that the torque can be controlled by quadrature component of stator current (i_{qs}^e) while keeping the rotor flux (Ψ_{dr}^e) constant or keeping the direct axis component of the stator current (i_{ds}^e) constant. Here, we can say that direct axis component and the quadrature axis component of the stator current are flux and torque controlling components respectively.

$$T_{em} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i_{qs}^* i_{ds}^* \qquad (3.14)$$

In dc motor the electromagnetic torque equation can be written as,

$$T_{em} = K_m i_a i_f$$

From the above equation and equation (3.14) we can conclude that the vector controlled induction motor are an analogue to DC motor. However, these reference axis components (i_{ds}^* and i_{ds}^*) are not actual but only some mathematical entities. So, need to convert those components into real a, b and c phase currents (i_{as} , i_{bs} and i_{cs}). For this conversion we have transform synchronously rotating d-q frame quantity into the stationary abc quantity, this transformation need the position angle of d^*or q^* axis in the space (sayΘ_e).

Transformation of $i_{qs}^e$and i_{ds}^e into i_{as} , i_{bs} and i_{cs}

$$\begin{bmatrix} i_{as}^* \\ i_{bs}^* \\ i_{cs}^* \end{bmatrix} = \begin{bmatrix} \cos\theta e & -\sin\theta e \\ \cos(\theta e - 120) & -\sin(\theta e - 120) \\ \cos(\theta e + 120) & -\sin(\theta e + 120) \end{bmatrix} [\ ] \begin{bmatrix} i_{ds}^* \\ i_{qs}^* \end{bmatrix} \qquad (3.15)$$

The evaluation of Θ_e will lead us to go for direct evaluation and indirect evaluation knows Direct Vector Control and Indirect Vector Control respectively.in this report we will concentrate only Indirect Vector control because of its simplicity in algorithm of finding the position and less sensitive to the parameter variation.

3.1.2 Indirect Vector Control of an Induction Motor

The following method can be used to get the position of rotor flux or d axis of synchronously rotating frame with respect to stator a-phase (from Fig.3.1) by integrating the synchronous speed which can be obtained by adding rotor and slip speed (ω_{sl}) .

$$\Theta_e = \int \omega_e \, dt = \int (\omega_r + \omega_{sl}) \, dt \qquad (3.16)$$

The block diagram of indirect vector control can be made by using the equations (3.8), (3.14), (3.15), (3.16) and it shown below in the Fig.3.2.

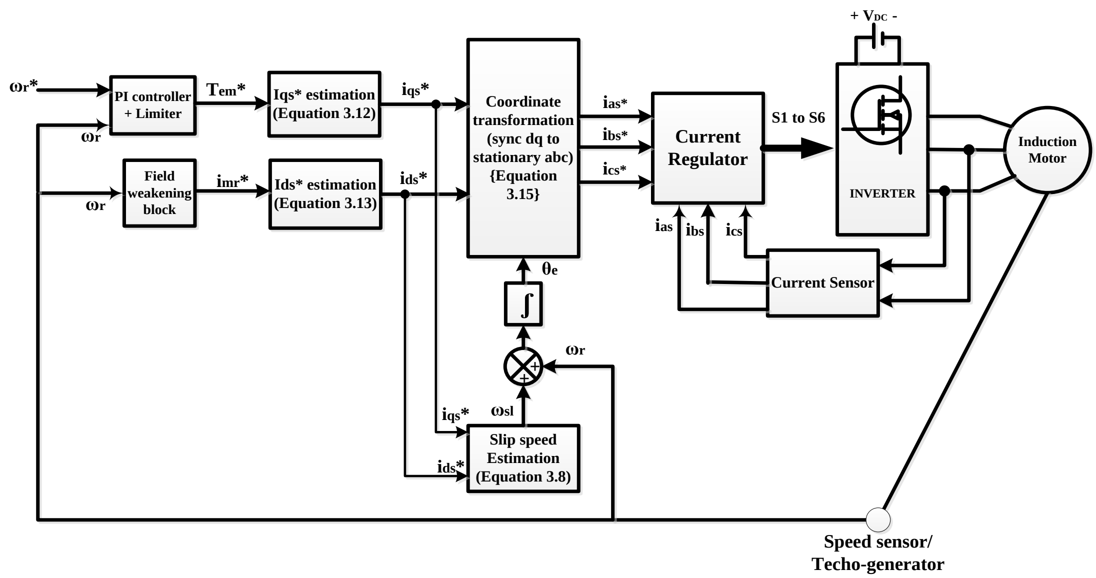


Fig.3.2 Block diagram of Indirect Vector Control of Induction Motor Drive

The description of each block above block diagram will be explain in simulation part of the indirect vector control later chapter.

CHAPTER 4

Artificial Neural Networks

Artificial Neural Networks (ANN) are a means/system to process information, similar to the biological nervous systems, specifically as the brain. A key component of these systems is the novel nature of the information processing system. They are made of a number of highly interconnected processing elements (neurons) working in unison to solve certain problems. Like people, ANNs learn by example. It is configured for a specific application, such as data classification or pattern recognition through learning. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. This is true of ANNs as well.

4.1 Concept of Artificial Neural Networks

Neural networks, have the remarkable ability to derive meaning from imprecise or imperfect data and can be used to detect trends and extract patterns that are too complicated to be noticed by either humans or other computer techniques. A trained neural network is an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions.

Other advantages include:

1. Adaptive learning: During the training process, it comes with an ability to learn on how to do tasks based on the initial experience.
2. Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.
3. Real Time Operation: ANN architecture enables them to carry out computations in parallel, and special hardware devices are being designed and manufactured which can exploit this advantage.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network can lead to the corresponding erosion of performance. Though, most of the network capability might still be available making them robust.

Neural networks fundamentally approach a problem differently than the conventional computers. Conventional computers use an algorithmic approach where in the computer carries out a set of instructions in order to solve a particular problem. The computer cannot solve the problem without knowing the specific steps involved in it, this restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But they would be so much more useful if they were able to operate on problems that they exactly didn’t know how to do.

Neural networks are similar to the human brain in the way they process information. The network has a large number of highly interconnected processing elements (neurons) working in parallel to solve a particular problem. Neural networks learn by example. They cannot be programmed to perform a specific task. This makes the training process an extremely important step in the whole process, the examples must be selected carefully otherwise valuable time is wasted or even worse the network might not function correctly. The disadvantage though is that since the network finds out how to solve a problem by itself, its operation may not be predictable.

On the other hand, cognitive approach to problem solving is used by conventional computers; the approach towards the problem must be known to solve it and must be stated in small unambiguous instructions. These instructions are then converted to a high level language program and then finally into a machine code that the computer can comprehend. These machines are totally predictable; in case any fault occurs due to either software or hardware issues.

Neural networks and conventional algorithmic computers are not in competition but rather can complement each other. Some tasks are better suited to an algorithmic approach based problem solving like arithmetic operations and some are more suited to neural networks. Even then, a large number of tasks, require systems that involve a combination of the two approaches (typically a conventional computer supervises the neural network) in order to perform at maximum efficiency.

An artificial neuron can be thought of as a device with many inputs and single output. The neuron has two stages of operation; the training stage and then usage stage. In the training stage, the neuron is trained to fire (or not), for a particular set of input patterns. In the usage stage, when a taught input pattern is detected at the input, its corresponding output becomes the current output. If the input pattern does not occur in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

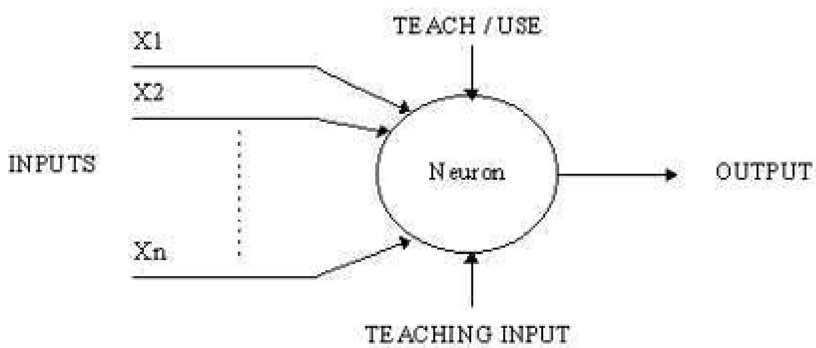


Figure 4.1 Simple neuron without activation function

The previous neuron simply replicates a conventional computers and doesn't do anything new. A more sophisticated neuron (figure 2) is the McCulloch and Pitts model (MCP). The difference from the previous type of model is that the inputs are 'weighted', the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

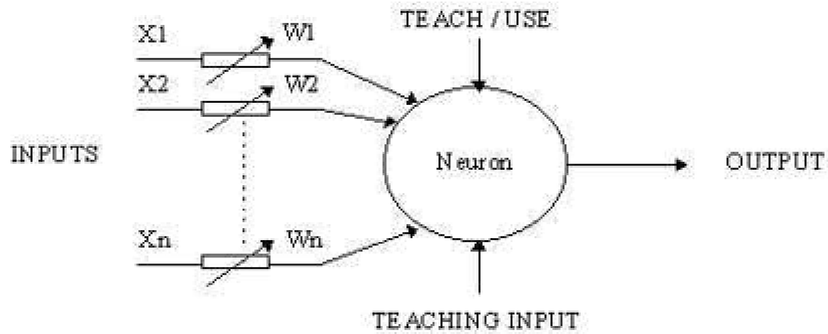


Figure 4.2 Neuron with weighted inputs

In mathematical terms, the neuron fires if and only if;

$$X1W1 + X2W2 + X3W3 + \ldots > T$$

The presence of input weights and corresponding threshold make the neuron flexible and powerful. The MCP neuron can adjust to a particular situation by changing its weights and/or threshold. Several algorithms allow the neuron to 'adapt'; the most used ones being the Delta and the back error propagation ones. The former is used as a feed-forward neural network and the latter is used as an error feedback network.

Neural networks are typically organized in form of layers. A Layer is made up of a large number of interconnected 'nodes' which contain the 'activation function'. Patterns are presented to the network via the 'input layer', which are then communicated to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link the output to an 'output layer' where the final answer is output as shown in the graphic

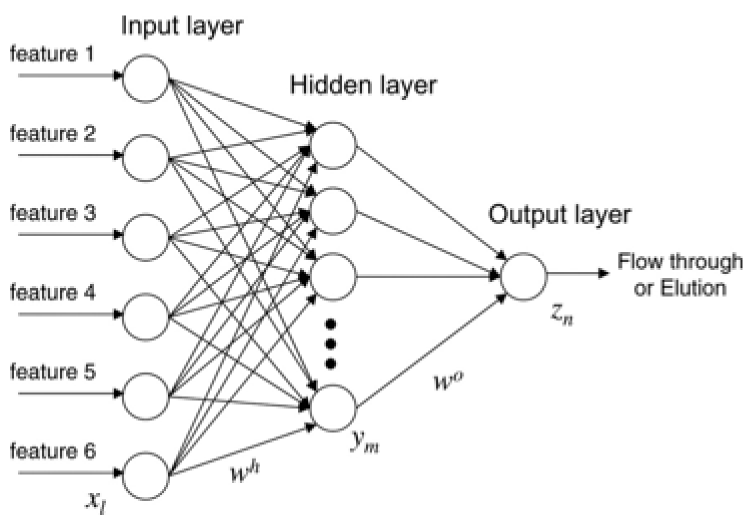


Figure 4.3 Three layer neural network architecture

Almost all ANNs have some form of 'learning rule' which modify the weights of the connections according to the input patterns that it is fed with. In a sense, ANNs learns by example similar to their biological counterparts; a child learns to recognize dogs by learning from examples of dogs.

There are many different kinds of learning rules used by neural networks, with the delta rule being discussed here. The delta rule is often characterized by the most common class of ANNs called 'backpropagation neural networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that happens in steps with each cycle or 'epoch' (i.e. each time the network is fed with a new input pattern) through a forward activated flow of outputs, and the backwards propagation of error and corresponding weight adjustments. Simply, when the neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. Based on how far the answer is from the actual one and it then makes an appropriate adjustment to its connection weights. More graphically, the process looks something like this:

$$I = f(\sum W_i \ * Input)$$

$$W_{new} = W_{old} + \beta EX/| X |$$

Backpropagation carries out the gradient descent within the solution's vector space towards a 'global minimum' by moving along the steepest vector of the error surface. The global minimum is the theoretical solution with the lowest possible error. The error surface itself is a hyper paraboloid but is rarely 'smooth' as is depicted in the graphic below. Indeed, in most problems, the solution space is quite irregular with numerous 'pits' and 'hills' which may hamper the process.

Since the nature of the error space cannot be known before hand, neural network analysis often requires a large number of individual iterations to reach the best solution. Most learning rules have built-in mathematical means to assist in this process which control and adjust the 'speed' (Beta-coefficient) and the 'momentum' of the learning. The speed of learning is actually the rate with which the current solution converges to the global minimum. Momentum helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global minimum.

Once a neural network is 'trained' to a satisfactory level of accuracy it may be used as an analytical tool to process newer set of data. The user no longer needs to specify any training runs and instead simply allows the network to work in a forward propagation mode only. New inputs are presented to the network where they filter into it and are then processed by the middle layers as though training were taking place, however, at this point the output is retained and no backpropagation occurs. The output of a forward propagation run is the predicted model for the data which can then be used for further analysis and interpretation.

It is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one set of input; which is similar to rote memorization. If this happens then learning is no longer possible and the network is referred to as having been "grand-mothered" in neural network jargon. In real-world applications this defeats the purpose of a network since one would need a separate grand-mothered network for each set of input.

Before jumping further into artificial neural computing it is of paramount significance to know how a conventional 'serial' computer and its software processing of information takes place. A serial computer contains a central processor that addresses an array of memory locations where data and instructions are stored. Computations are made by the processor after reading an instruction as well as any data the instruction might require from memory addresses, the instruction is then executed and the results are saved in the specified memory location as required. In a serial system (and a standard parallel one as well) the computational steps of processing are deterministic, sequential and logical, and the state of a given variable can be tracked from one operation to another.

On the other hand, ANNs are neither sequential nor necessarily deterministic. They do not have any complex central processors, rather they have several simple ones which generally do nothing more than taking the weighted sum of their inputs from other processors. ANNs do not perform programed instructions; rather they respond in parallel (either simulated or actual) to the pattern of inputs presented to it. No separate memory addresses for storing data is involved either. Instead, information is contained in form of the overall activation 'state' of the network. 'Knowledge' is thereby represented by the network itself, which is more than the sum of its individual components.

Based on the specifics of the application and the quality of the internal data patterns you can typically expect a network to train quite well. This is especially true for applications involving dynamic and non-linear type of systems. ANNs provide an analytical means to have an alternative solution to the conventional techniques, often limited by stricter assumptions of normality, linearity, variable independence etc. Because of their ability to model non-linearity efficiently and because it makes no inherent assumption about data or function, they can fit and explain or rather learn patterns that others might not be able to comprehend.

4.1.1 Levenberg-Marquardt Algorithm

The simple gradient descent works for simplest of models, but is a bit too simplistic a method for more complicated and complex models with free parameters. Convergence in this scenario can take an extremely long time: the key reason into why this might be the case is that the problem is stiff in the sense that the some zones where small step sizes are needed denigrate the whole problem solution. For example, when descending the walls of a very sharp/step local minimum bowl we need to take very small step sizes to avoid jumping out of the bowl space. While when moving along a gently sloped part of the error surface we would want to take large steps so that we can proceed to other locations and zones faster otherwise it will take forever to get anywhere. This problem is further complicated by the manner in which gradient descent has been implanted which is a bit out of place (we generally move by taking a step that is some constant times the negative gradient rather than a step of constant length in the direction of the negative gradient. This basically translates into scenarios where in steep regions (we have to be careful here not to make our steps sizes too large) we move fast and in shallow/flat zones (where we need to move in big steps) we move slowly

Let's say there is a narrow and deep valley in the error surface, the component of the gradient pointing in the direction along the base of the valley is very small while the component which points perpendicularly to the valley walls is quite big even though we have to move a much larger distance along the base and a smaller distance perpendicular to the walls of valley. This brings the need to use a slightly more sophisticated gradient descent algorithms than simple steepest descent, which is just

$$w_{i+1} = w_i - \mu \nabla E(w)$$

w is the value that is being minimized, this at the global minima of solution space, would be the least, μ is the constant step size times the negative gradient of w. But as mentioned above, the simple gradient descent suffers from several issues, to better upon it, the following

Simplest descent can also be written as where d represents average of $\nabla E(w)$ for past values.

$$w_{i+1} = w_i - \mu d$$

Update rule based on quadratic approximation involves

$$w_{i+1} = w_i - H^{-1} d$$

Here, H represents the approximate Hessian matrix

The Levenberg algorithm is a mix of two which has the following update rule,

$$w_{i+1} = w_i - (H + \lambda I)^{-1} d$$

Here I is the identity matrix as λ gets small, the rule approaches the quadratic approximation update rule above. If λ is large, the rule approaches the below update form.

$$w_{i+1} = w_i - \frac{1}{\lambda} d$$

This equation is same as the simple descent algorithm. The algorithm λ adjusts as ∇E is either increasing or decreasing.

1. Update the value as given above.

2. Evaluate error for the new weight vector.

3. If the error increases as a result of the update, then retract the step (i.e. weights are reset back to their previous values) and then increase λ by a factor of 10 or some such significant factor. Then go to (1) and try an update again.

4. If the error decreases as a result of the update, then accept the step (i.e. weights are given the new values) and decrease λ by a factor of 10 or so.

The intuition behind all this is that if error increases, our quadratic approximation will not working well and we are likely not to be near a minimum, so λ needs to be increased in order to blend more towards simpler gradient descent approach. Conversely, if the error decreases, our approximation has been working well, and we can expect to get closer to the minimum so λ is decreased to bank more on the Hessian. Marquardt bettered this method with a clever addition of estimated local curvature information, resulting in the Levenberg-Marquardt method. The insight of Marquardt was that when λ is high and we are doing essentially gradient descent, we can still get some more benefit from the Hessian matrix that we had estimated. In essence, he had suggested that we move further in the directions in which the gradient is smaller in order to circumvent the classic error valley problem. So he replaced the identity matrix in Levenberg's original equations with the diagonal of the Hessian:

$$w_{i+1} = w_i - (H + \lambda diagH)^{-1} d$$

As you can see, all this method needs is to operate are the same things steepest descent needs: y, f (x; w) and ∇f (x; w). In other words, we can compute **d** and **H** based only on the value of the function and its gradient which we know how to evaluate. Don't confuse this with being an alternative to backpropagation: it is an alternative to simple gradient descent. Backpropagation is nothing more than a clever and efficient algorithm for evaluating ∇f (x; w) for networks. We can then use this gradient in any way we want, either to do steepest descent or to do something tricky like Levenberg-Marquardt.

CHAPTER 5

Matrix Converter Simulations

5.1 Simulink Model of Matrix Converter

The matrix converter is modelled using the power system block set library of SIMULINK, a feature within MATLAB made available by matrix labs. The Alesina-Venturini 1981 (AV method) is adopted to carry out the simulation. The basic structure is given below

The entire simulation has been run on a discrete frame with the sampling time well below the solver time period. The sampling time for the simulation is 1e-5s. The supply parameters are given below

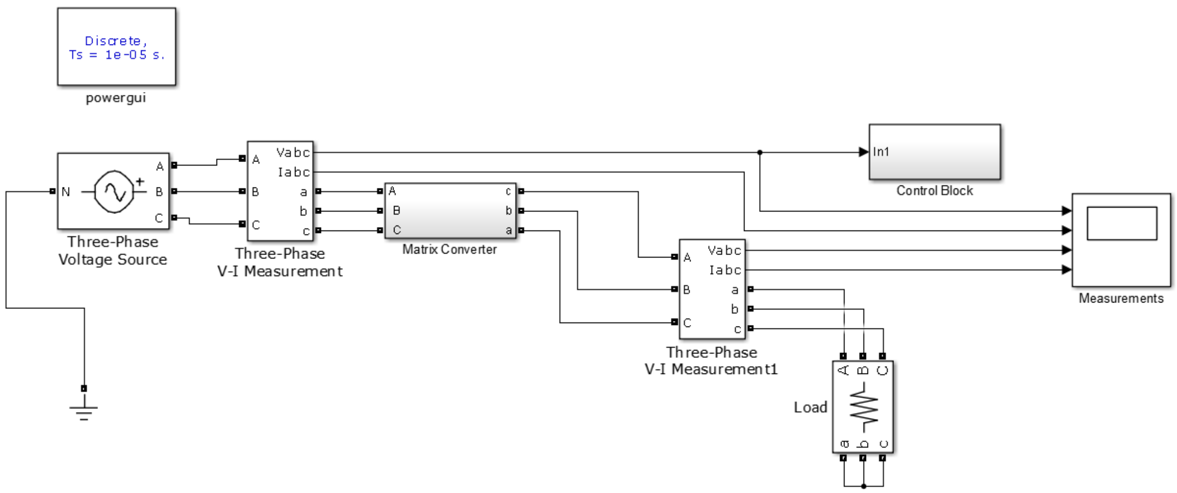


Figure 5.1 Simulink structure of matrix converter with Venturini Topology

Number of phases	Three phase 3 Φ
Line – Line Voltage	230 V
Frequency	50 Hz

Table 5.1 Supply Parameters for matrix converter

To measure the various parameters phase/line voltage, phase/line currents etc., three phase measurement block is provided by the SIMULINK is used. The control block consists of the entire control logic including the reference voltages etc. The matrix converter subsystem contains the 9 bi-directional switches. Each bi-directional switch has been implemented using two IGBTs placed in an anti-parallel fashion.

The internal structure of the block is given below, each pair of IGBTs here represent one bi-directional switch.

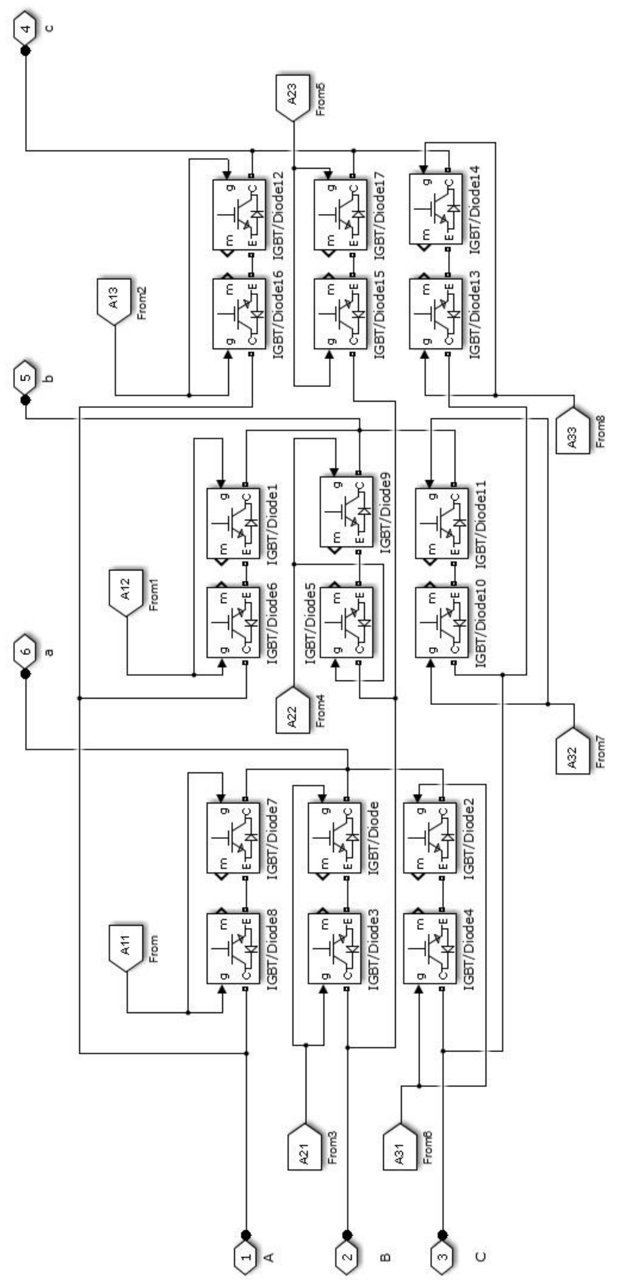


Figure 5.2 Simulink structure of matrix converter (3 × 3)

The structure of a single bi-directional switch.

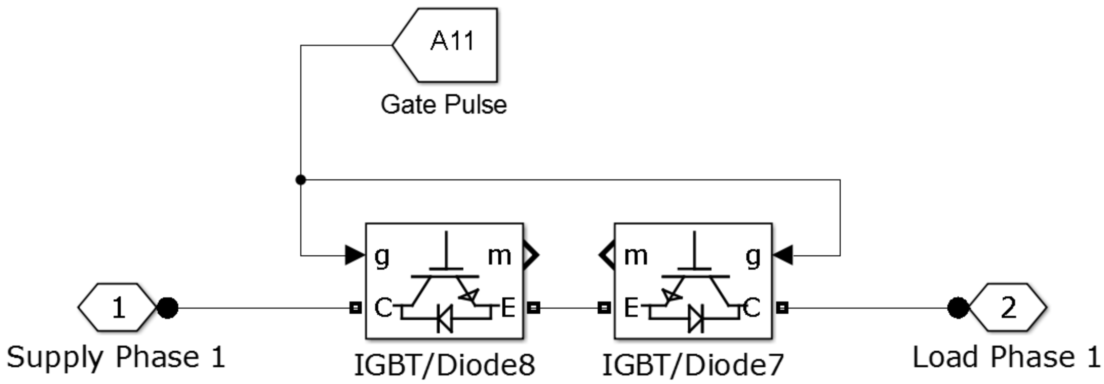


Figure 5.3 Simulink structure of bi-directional switch

The switching pulse for the control logic has the following structure. A saw tooth generator with a switching frequency of 1K Hz is used.

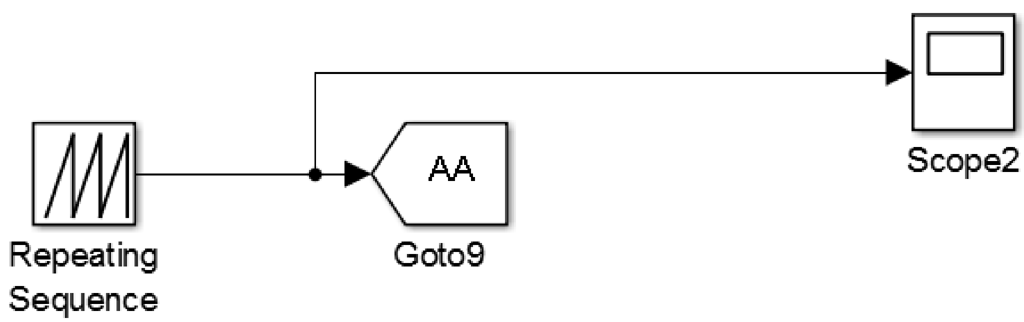


Figure 5.4 Simulink structure of pulse generator in control block

The structure has the logic for switches connecting the supply from 3 phases to one of the load phases. The modulation value m is calculated within the function block. V supply is the three phase supply, V reference has the reference value for one of the three phase, V_{rms} is the rms value of the supply line to ground voltage.

The function implemented within the function block is

$$\frac{1}{3}(1+\frac{u(1)*u(2)}{u(3)^2})$$

The internal control logic flows as in the diagram below

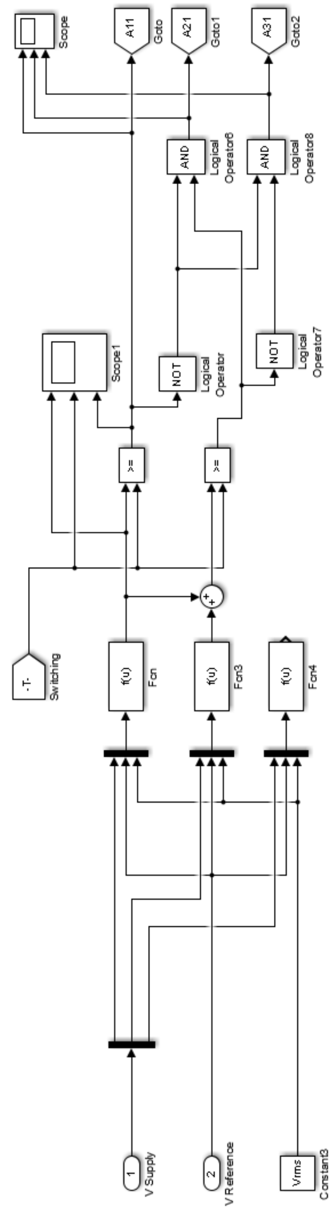


Figure 5.5 Simulink model of gate pulse generator for matrix converter switches

The output from the three phase's then pass through a combinatorial logic to give the final gate pulses for the three switches. The logic ensures that at any point of time no two input switch to same output and at every point of time one of the input is connected to an output.

5.2 Simulation Results with Passive Load (R/RL)

The simulations are carried out for 50 Hz frequency. The load is 10 Ω resistance, with amplitude gain kept at 0.3 below the plausible 0.5 for simple Venturini Model.

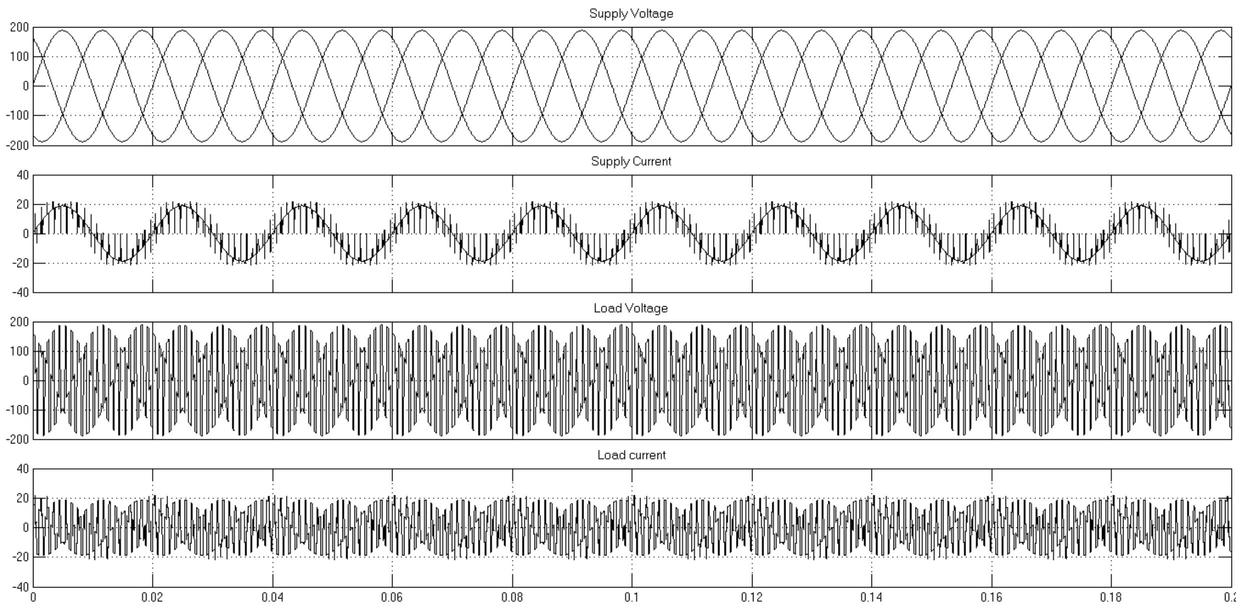


Figure 5.6 Voltage and Current profiles at supply and load side at 50 Hz

Plot 1: Supply Voltage (V), Plot 2: Supply Current (I), Plot 3: Load Voltage (V) Plot 4: Load Current (I)

The matrix converter is operable and is able to generate output voltage and input current waveforms.

5.3 Simulation Results with Induction Motor in Open Loop

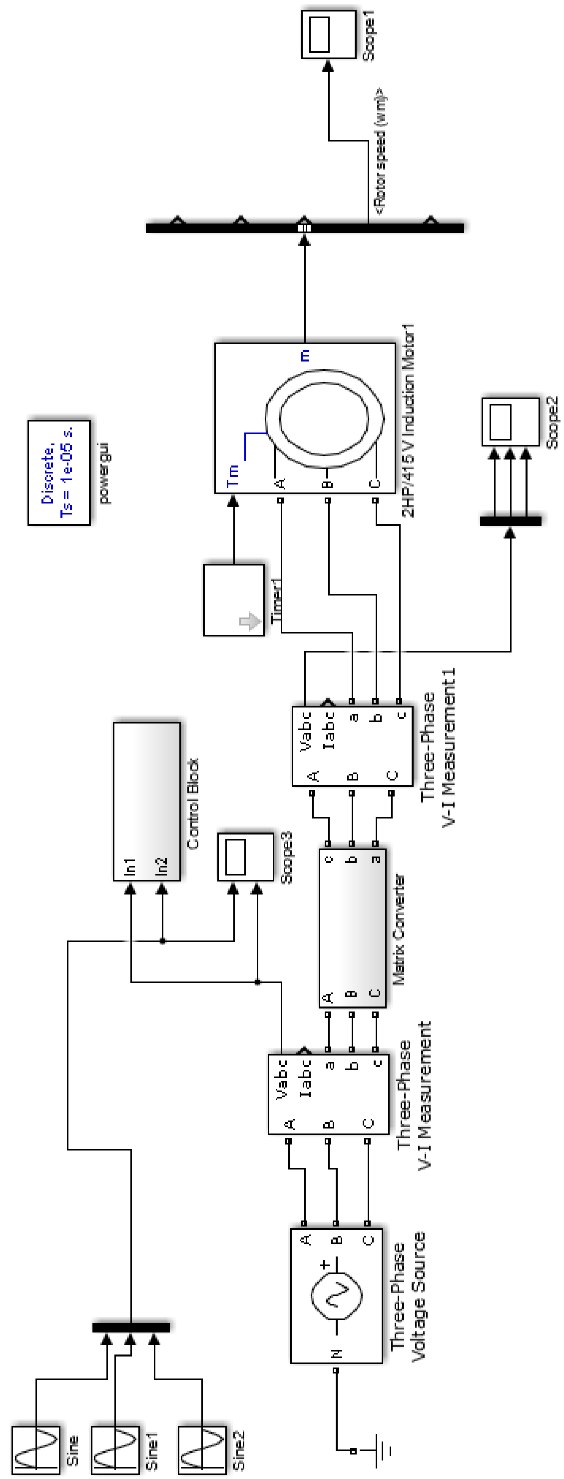


Figure 5.7 Simulink Structure for matrix converter with induction motor in open loop fashion

The above figure has the Simulink structure used for the open loop simulation. It has been carried out for two different input voltages and keeping frequency fixed at rated value and at two different frequencies with voltage kept at rated value.

Machine Ratings

Nominal Power	2 HP
Voltage(Line-Line)	415 V
Frequency	50 Hz
Stator Resistance	5.4 Ω
Stator Inductance	0.02840
Rotor resistance referred to stator side	3.1093
Rotor inductance referred to stator side	0.02840
Mutual Inductance	0.58372
Inertia	0.00436
Friction Factor	0
Pole Pairs	2

Table 5.2 Machine Parameters for open loop matrix converter fed induction motor simulation

The simulation of matrix converter fed induction motor drive is carried out at two different stator voltages and frequency while keeping one of them at rated value.

Open loop simulation at input voltage 415 line to line and frequency 50 Hz

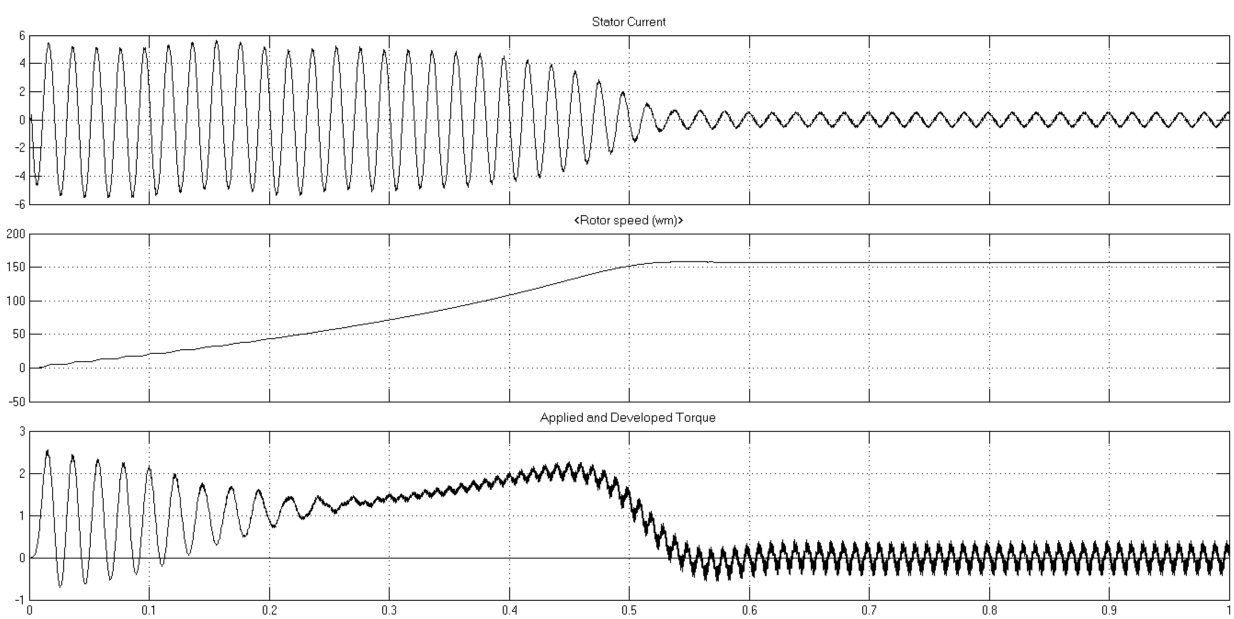


Figure 5.8 Induction Motor response to 415V line to line voltage at 50 Hz

Plot 1: Stator current (I), Plot 2: Rotor Speed (rad/s), Plot 3 Applied and Developed Torque (Nm)

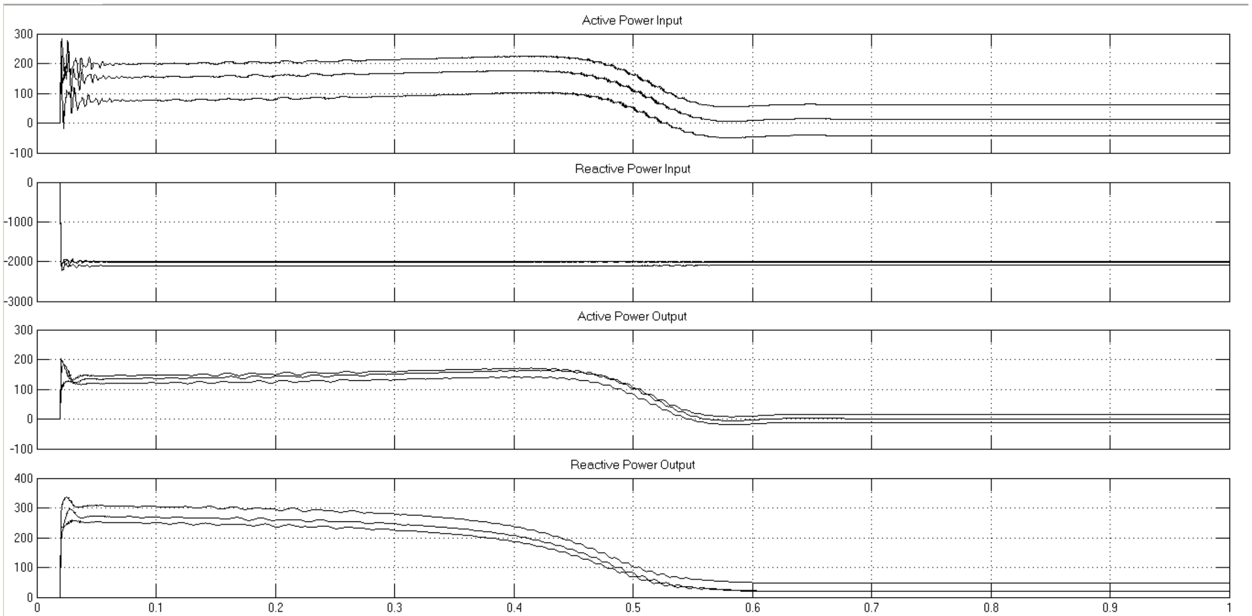


Figure 5.9 Power parameters at 415V line to line voltage and 50 Hz supply to IM

Plot 1: IM input Instantaneous Active Power (W), Plot 2: IM input Instantaneous Reactive Power (VAr), Plot 3: IM output Instantaneous Active Power (W), Plot 4: IM output Instantaneous Reactive Power

Open loop simulation at input voltage 415V and frequency 25 Hz

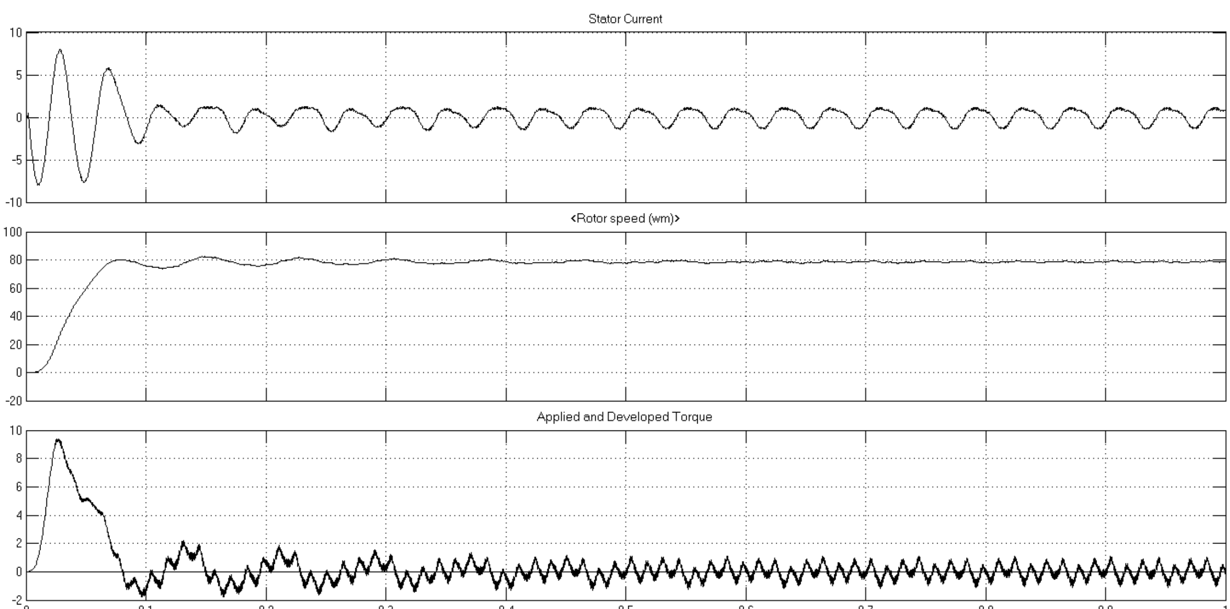


Figure 5.10 Induction Motor response to 415 V line to line voltage at 25 Hz

Plot 1: Stator current (I), Plot 2: Rotor Speed (rad/s), Plot 3 Applied and Developed Torque (Nm)

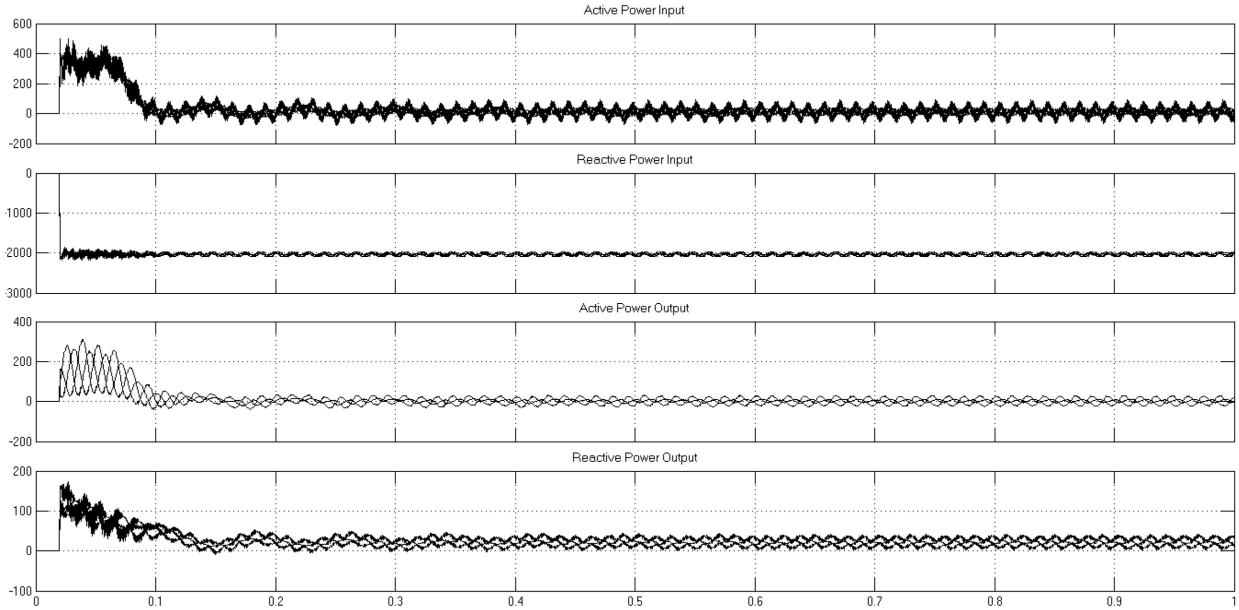


Figure 5.11 Power parameters at 415V line to line voltage and 25 Hz supply to IM

Plot 1: IM input Instantaneous Active Power (W), Plot 2: IM input Instantaneous Reactive Power (VAr), Plot 3: IM output Instantaneous Active Power (W), Plot 4: IM output Instantaneous Reactive Power

Open loop simulation at input voltage 207.5V and frequency 50 Hz

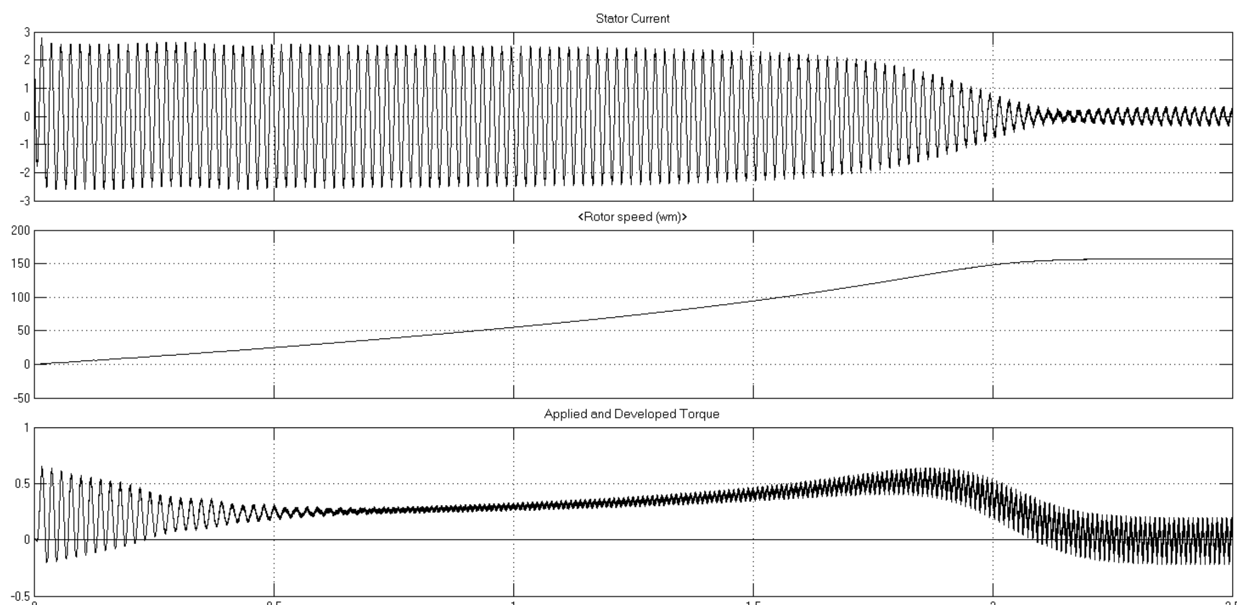


Figure 5.12 Induction Motor response to 207.5 V line to line voltage at 50 Hz

Plot 1: Stator current (I), Plot 2: Rotor Speed (rad/s), Plot 3 Applied and Developed Torque (Nm)

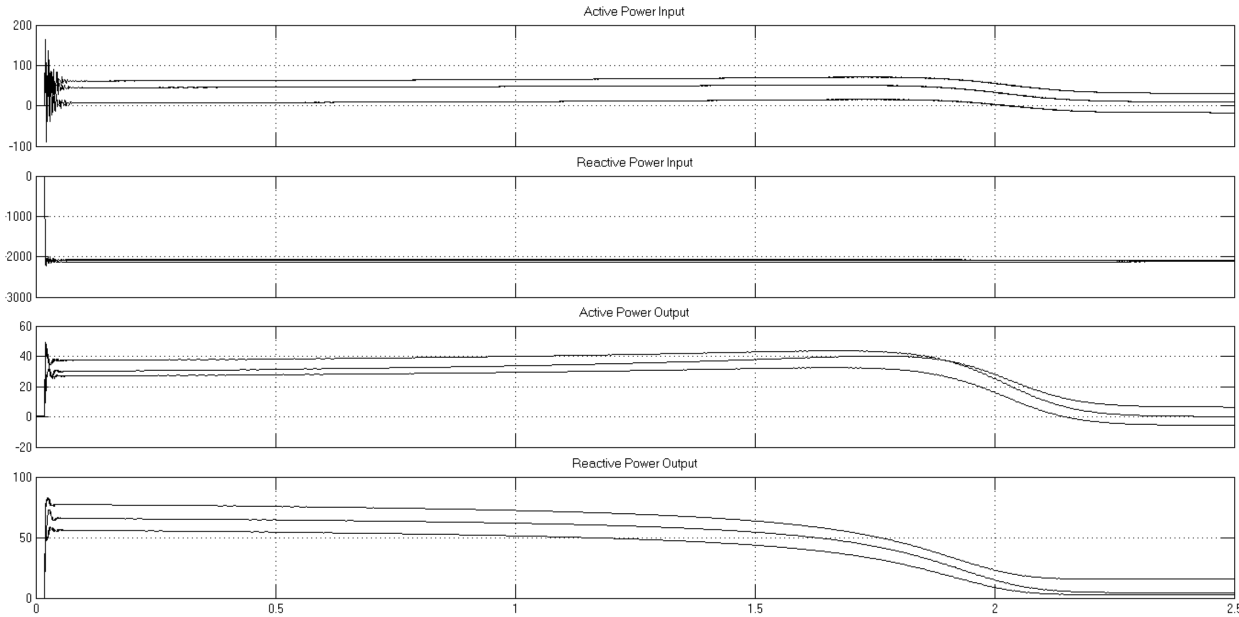


Figure 5.13 Power parameters at 207.5V line to line voltage and 50 Hz supply to IM

Plot 1: IM input Instantaneous Active Power (W), Plot 2: IM input Instantaneous Reactive Power (VAr), Plot 3: IM output Instantaneous Active Power (W), Plot 4: IM output Instantaneous Reactive Power

Condition A: 415V and 50 Hz; Condition B: 415V and 25 Hz; Condition C: 207.5 V and 50 Hz

An analysis of graphs and performance reveals the following

The rotor speed gets halved in condition B in comparison to condition A because the synchronous speed depends linearly on the frequency and induction motor runs at synchronous speed. As the reactive power is proportional to square of stator voltage and inversely proportional to frequency, the reactive power is supposed to double on halving the frequency thus reactive power intake doubles in condition B in comparison to A.

Condition A and C demonstrate the speed control of induction motor by stator voltage control. Speed reduced but by a smaller amount in this case. Speed thereby is less sensitive in this case. Active power consumption remains the same though reactive power goes down by a factor of square of 0.5.

The most important aspect of matrix converter is that input side reactive power is very less in comparison to output side reactive power thus making matrix converter very popular with induction motor as it is able to maintain power factor along with providing variable frequency and magnitude without any DC link.

CHAPTER 6

Artificial Neural Networks Based Induction Motor Drive Simulations

The proposed controller is an ANN which emulates the in-direct field oriented control as a function. The training is a one-step process and is trained using experimental data or an existing FOC system. An induction motor is a non-linear time varying system which is difficult to control due to fast changing state variables. Further the changes in rotor resistance and induction motor saturation during operation make the control difficult. A speed control system which can process large number of state variable calculations in short period besides handling the non-linear nature well is capable of showing high performance.

An artificial neural network has been trained and used where in it emulates indirect field control for an induction motor drive by generating direct and quadrature current commands in stationary reference frame. The neural network performs the functions of slip calculation and matrix rotation internally.

There are five input signals to the neural network which include the motor speed, quadrature current signals; present and delayed in synchronous frame, besides the two delayed output signals as input. The output signals are the currents in stationary frame (α-β frame).

The proposed network has a three layer structure with 15 neurons in the hidden layer. ANN used is a back-propagation neural network trained using Levenberg-Marquardt method.

Thus neural networks due to their inherent parallel nature and ability to model non-linearity well are suited for this particular task. The objective here is to emulate an in-direct FOC performance at lower computational cost. Cost savings from this non-recurrent engineering effort makes this valuable in high volume industries such as home appliance industries.

6.1 System Description & Design

Primary equations describing the behaviour of IFOC are

$$T = K_t \; i_{qs}^{e} \, \lambda_{dr}^{\prime e} \tag{1}$$

$$w_e = \frac{M}{\Gamma r} \frac{i_{qs}^{e}}{\lambda_{dr}^{\prime e}} + w_r \tag{2}$$

Here, the primed notation stands for stator referred. Equations transforming synchronous frame quantities (subscript e) to stationary reference frame(s subscript) are

$$i_{qs}^{s} = i_{qs}^{e}\cos(\theta_e) + i_{ds}^{e}\sin(\theta_e) \qquad (3)$$

$$i_{ds}^{s} = -i_{qs}^{e}\sin(\theta_e) + i_{ds}^{e}\cos(\theta_e) \qquad (4)$$

The four equations follow the following notations

T Electromagnetic torque

K_t Motor torque proportionality constant

i_{qs}^{e} Synchronous frame q-axis stator current

i_{ds}^{e} Synchronous frame d-axis stator current

w_e Rotor flux angular velocity

θ_e Rotor flux angular position

λ'^{e}_{dr} Stator referred direct axis flux linkage

i_{qs}^{s} Stationary frame q-axis stator current

i_{ds}^{s} Stationary frame d-axis stator current

Γ_r Rotor time constant

M 3/2 × (stator referred stator/motor mutual inductance)

Now the equation (2) can further be rewritten as

$$\frac{d\theta_e}{dt} = \frac{1}{\Gamma r}\frac{i_{qs}^{e}}{i_{ds}^{e}} + w_r$$

Now assuming a constant rotor flux magnitude, using discrete-time approximation, we can transform the equation into following difference equation:

$$\left(\frac{1-z^{-1}}{T_s}\right)\theta_e(k) = \frac{1}{\Gamma r}\frac{i_{qs}^{e}(k)}{i_{ds}^{e}(k)} + w_r(k)$$

T_s represents the sampling time, Z^{-1} is the delay operator, k refers to the sampling interval. Rotor flux in each sampling period is given as

$$\theta_e(k) = T_s\left(\frac{1}{\Gamma_r}\frac{i_{qs}^{e}(k)}{i_{ds}^{e}(k)} + w_r(k)\right) + \theta_e(k-1)$$

Now to obtain θ_e(k) at t = kT$_S$, the above variables need to be known. All except θ_e(k-1) are known. This can be obtained using the following equation

$$\theta_e(\text{k-1}) = \tan^{-1}\left(\frac{i_{ds}^{e}(k-1)i_{qs}^{s}(k-1)-i_{qs}^{e}(k-1)i_{ds}^{s}(k-1)}{i_{qs}^{e}(k-1)i_{qs}^{s}(k-1)-i_{ds}^{e}(k-1)i_{ds}^{s}(k-1)}\right)$$

Here, the terms inside the inverse tan function can be calculated using the (3) and (4) set of equations.

The neural network controller is based on the above set of equations, with inputs and outputs defined on the basis of above equations. The flux command to the controller is assumed to be constant. The neural network has three layers: one input, one hidden and one output layer. The output layer has two neurons corresponding to the outputs i_{ds}^{s} and i_{qs}^{s} command values. It has five input neurons corresponding to w_r(k), i_{qs}^{e}(k), i_{qs}^{e}(k-1), i_{ds}^{s}(k-1) and i_{qs}^{s}(k-1).

The network is fully connected with a feedforward layer through weights, a bias signal through weight is also present. All neurons in the layer have a hyper tangent sigmoid function. To ensure normalization, for proper training of the network, normalization units are also present. Since we used a 15 layer hidden layer obtained using trial and error, the complete network has 22 neuron structure.

The new structure of the structure of the vector control with the neural network controller is

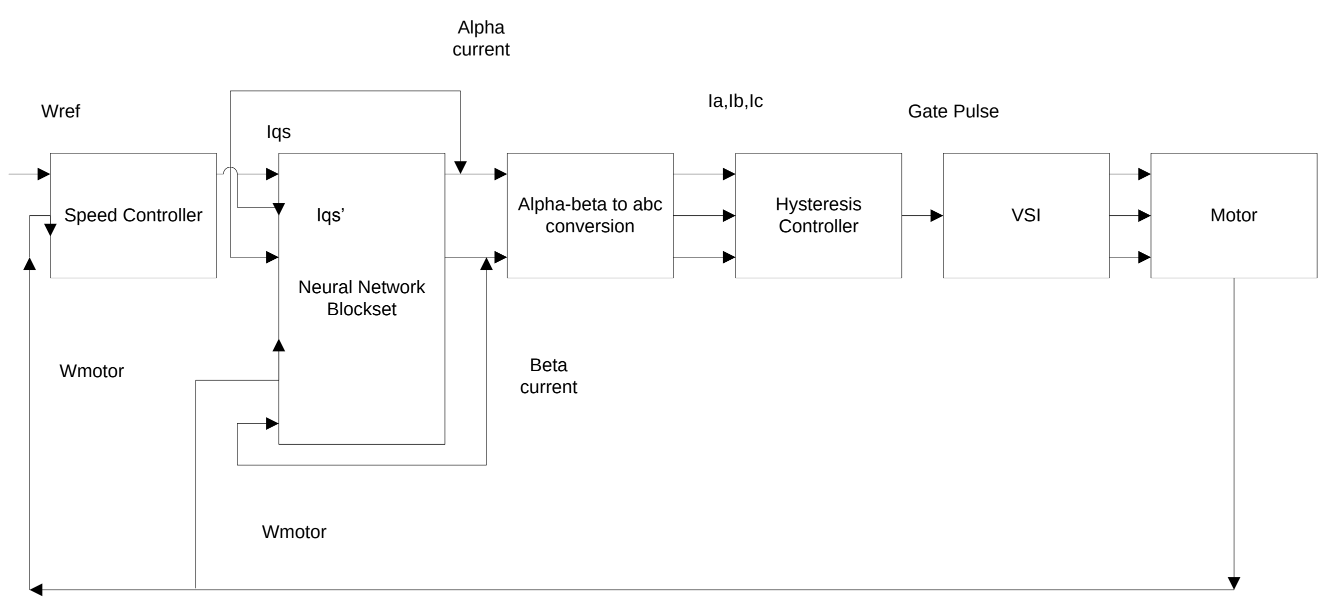


Figure 6.1 The System Design which is to be implemented

6.2 Implementation Methodology

The implementation of the entire system has the following steps,

1. First step is the collection of data for training the neural network
2. Second step is to train the neural network
3. Third step is to run the neural network with the structure as shown above

6.2.1. Simulation of Indirect Vector Control Model

A 2 HP machines was simulated under indirect vector control strategy of MATLAB platform using SPS toolbox in the Discrete Time Frame (DTF). The Simulink model is shown in Fig. 6.2.

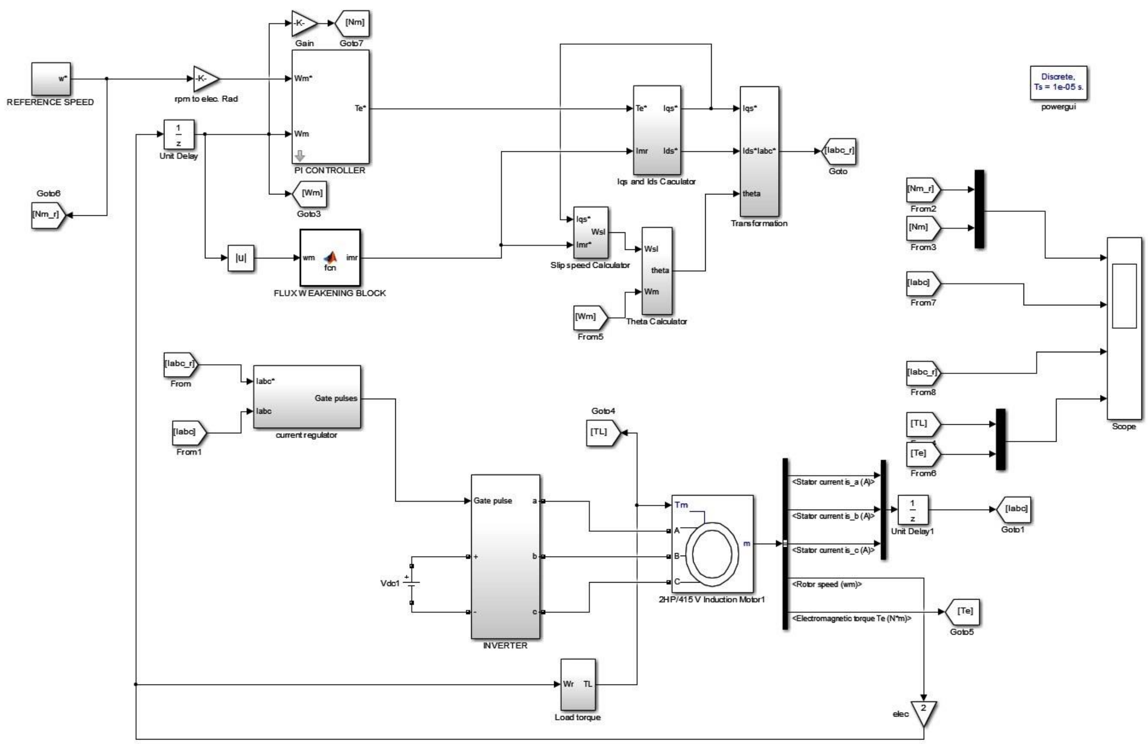


Fig. 6.2 Basic simualtion model of indirect vector control of IM drive

Speed Controller

The work of speed controller is to generate suitable reference torque signal from the speed error (difference between actual speed and the reference speed). There are many speed controller techniques are available like PI controller, Fuzzy Logic base controller, Intelligent Speed controller (Fuzzy + PI) etc. from all of the above, I have used PI controller to compelete this simulation.

PI controller

In the continuous time frame, the proportional integral controller can be represent as,

$$T_{em} = K_p(\omega_r^* - \omega_r) + K_I \int(\omega_r^* - \omega_r)\, dt \quad (6.1)$$

But we simulate the model in the discreet time frame, so the above equation can be converted in DTF as below,

For the N^{th} sample equation can be written as

$$T_{em\,(N)} = K_p\, \omega_{er(N)} + K_I \sum_{N=1}^{N}\{\omega_{er(N)}\} \quad (6.2)$$

For $(N\text{-}1)^{th}$ sample

$$T_{em\,(N-1)} = K_p\, \omega_{er(N-1)} + K_I \sum_{N=1}^{N-1}\{\omega_{er(N-1)}\} \quad (6.3)$$

If the $(N\text{-}1)^{th}$ sample pass through limiter, it would become the reference of $(N\text{-}1)^{th}$ sample,

$$T^*_{em(N-1)} = K_p\, \omega_{er(N-1)} + K_I \sum_{N=1}^{N-1}\{\omega_{er(N-1)}\} \quad (6.4)$$

By subtracting the equation (5.2) from equation (5.4) we'll get,

$$T_{em\,(N)} = T^*_{em(N-1)} + K_p\, \{\omega_{er(N)} - \omega_{e(N-1)}\} + K_I\, \omega_{er(N)} \quad (6.5)$$

The equation (5.5) shows the basic PI controller in DTF, can be modelled as shown in Fig.6.3. The electromagnetic torque of N^{th} sample will be pass through the Limiter, so that the reference value of torque will in certain liming band.

Field Weakening Block

Field weakening (FW) operation is considered when reference speed (ω_r^*) is more than the base speed (ω_{base}). After all the vector control technique make induction motor operation as dc motor and in dc motor we use Field weakening method to operate it for greater than base speed.

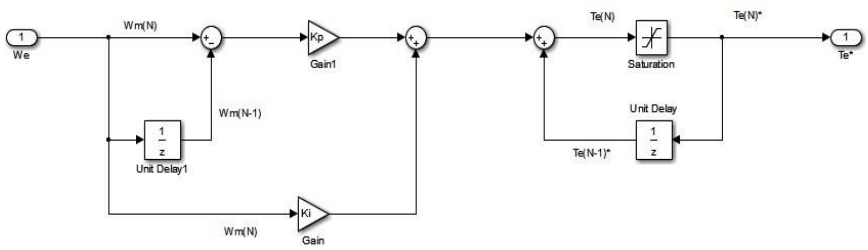


Fig.6.3 Speed controller using PI control logic in MATLAB

Here also the same method has been consider to reduce the flux in proportional to the speed when speed is greater than base speed [].

$i^*_{mr} = i_m$; When $\omega_r < \omega_{base}$

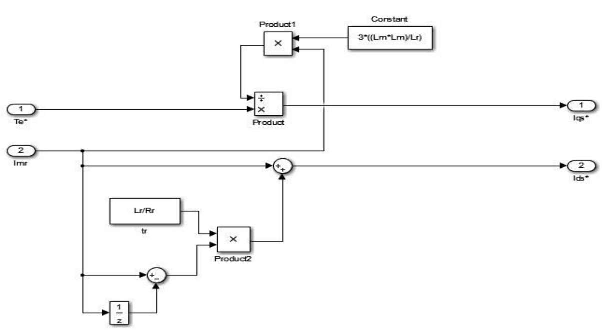
$i^*_{mr} = K_f \frac{i_m}{\omega_r}$ When $\omega_r \geq \omega_{base}$

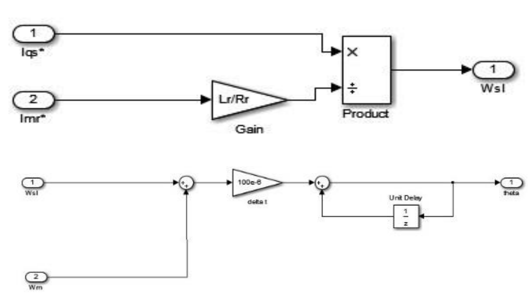
Where *imr* refers to rms value of magnetizing current, and K_f refers to flux constant.

i_{qs}* AND i_{ds}* ESTIMATION, SLIP SPEED ESTIMATION AND COORDINATE TRANSFORMATION BLOCK:

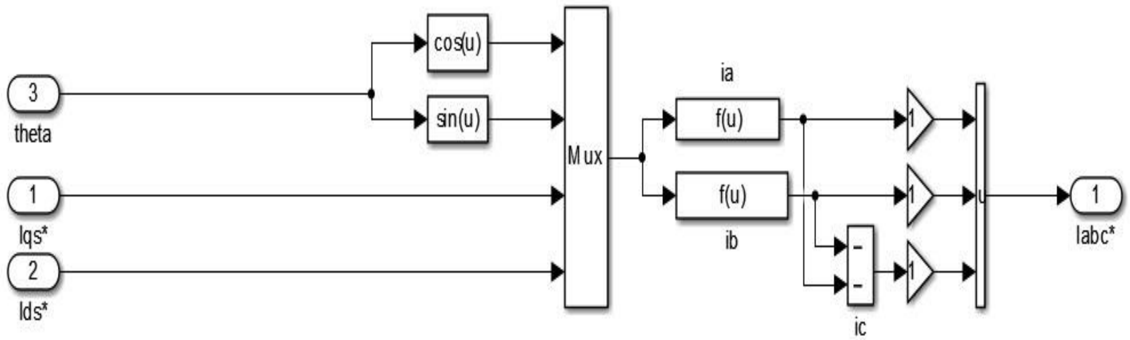
Some other blocks in the simulation part like Direct and Quadrature axis (synchronously rotating ref. frame) component of stator current Calculation, Slip speed calculation, Transformation from d^*-q^* current to stationary axis abc-frame component etc. can be easy modelled by equations (3.8), (3.13), (3.14), (3.15) and (3.16), as shown below Fig.6.4.

(b)





(a) (c)



(d)

Fig.6.4 (a) Estimation of i_{qs}* and i_{ds}*. (b) Estimation of slip speed. (c) Estimation of Θ_e and (d) Coordinate Transformation.

Speed Sensors

It sense the speed of rotating machine it can either be a speed encoder or Tacho-generator with Voltage sensor.

Current Regulators

Diverse current regulating techniques like, Sinusoidal PWM, synchronous dq frame PI regulator, stationary frame PR, stationary frame PI and hysteresis current regulator etc. In all of those regulators Hysteresis Current control method of VSI offers an matchless transient response in contrast with other analog and digital method, which makes it suitable to accept this method in all cases where high accuracy, wide bandwidth, and robustness are essential.

Sinusoidal PWM

Here, depending upon the frequency of carrier, the switching device will operated by comparing the reference current and actual current, the error signal will generate then it will pass through PI controller to give the Modulation Index (MI). MI signal will pass through Limiter then multiply with sine wave having unity peak value, after that it compare with carrier wave to generate Gate Pulse signal. Modelling of this technique is shown in Fig 6.5.

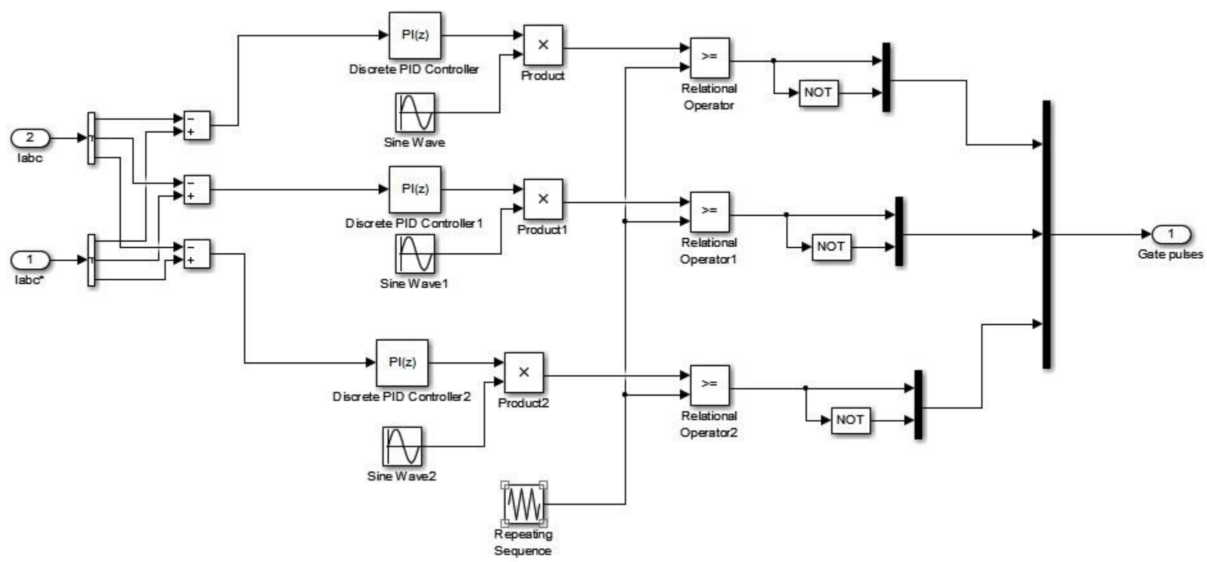


Fig.6.5 Sinusoidal PWM method to generate gate pulse for two level Inverter

Hysteresis Current Regulator

Hysteresis control technique is fundamentally an analogic technique. In spite of the merits given by digital controllers, in term of flexibility, maintenance integration interfacing, their correctness and response speed are often insufficient for current control in highly challenging applications, such as active filters and high- precision drives. In these applications, reference current waveform categorised by high harmonic content and fast transient must followed by good accuracy. In these belongings, the hysteresis technique can be a fine solution, provided some improvement are introduced to overcome its main limitations, which are sensitivity to phase commutation interference and switching frequency. The modelling of this technique is shown in Fig. 6.6.

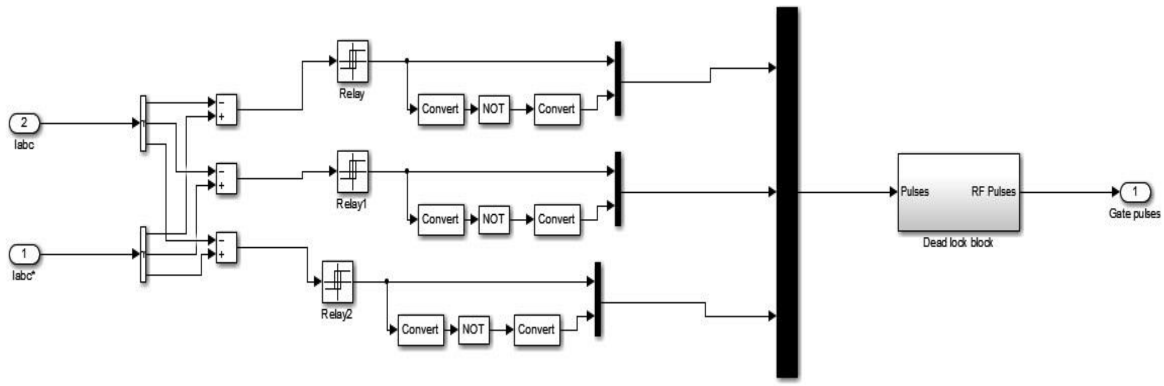


Fig. 6.6 Hysteresis current regulator to generate gate pule for inverter

6.2.2 Data Collection Methodology

The data for training the neural network was obtained from the Simulink model of vector control. The motor being simulated has the following set of parameters, the same motor was used with vector control model as well.

Machine Ratings

Nominal Power	2 HP
Voltage(Line-Line)	564 V
Frequency	50 Hz
Stator Resistance	5.4 Ω
Stator Inductance	0.02840
Rotor resistance referred to stator side	3.1093
Rotor inductance referred to stator side	0.02840
Mutual Inductance	0.58372
Inertia	0.00436
Friction Factor	0
Pole Pairs	2

Table 6.1(a) Motor/Machine Parameters of 2 HP

The model was run for 20 sec. The four values, w_r(k), i_{qs}^{e}(k), i_{ds}^{s}(k) and i_{qs}^{s}(k) were logged at a sampling rate of 100e-6. Thus a total of 2, 000, 00 data points were collected in total for these four values.

To capture as much behaviour as possible, the motor speed and torque was continuously varied over the period of 20 sec. To ensure and give the motor time to settle, torque and speed weren't varied at the same time. Also to ensure proper training of the model, it is important to generate speed and torque commands randomly so that there is no inherent pattern to these reference values. Speed commands were 20 in total with speed varying from -30/30 to -300/300 with changes in steps of 30. Torque was given 20 commands with it being varied from 0 to 7.6 in steps of 0.4. The code used to randomize the speed and torque commands is given below

```
% the 20 speed commands
speed_ref = [-30, 30,-60, 60,-90, 90,-120, 120,-150,
150,-180, 180,-210, 210,-240, 240,-270, 270,-300,300];
y = randperm (20);
% jumbling up the commands
speed_ref = speed_ref(y);
speed_ref

% the 20 torque commands
torque_ref = 0.0:0.4:7.6;
y = randperm(20);
% jumbling up the commands
torque_ref = torque_ref(y);
torque_ref
```

Code Block 6.1(b) Script to randomize torque and speed references

6.2.3 Neural Network Training Methodology

Once the data has been collected, it can be used to train the neural network. To ensure proper training, the data was initially pre-processed to remove extreme spikes, this is necessary because the neural network would use MSE (mean squared error) to fit the function. Presence of spikes would make fitting difficult and we would not be able to get a well-trained network. To limit the range of values, alpha(i_{qs}^{s}), beta (i_{ds}^{s}) and i_{qs}^{e} values can be limited by their 2^{nd} and 98^{th} percentile of values. Also each value is replaced by median of nearest 20 data points, this removes any spikes in the data. The same is carried out for beta current, i_{qs}^{e} as well.

```
alpha(alpha<prctile(alpha,2))= prctile(alpha,2);
alpha(alpha>prctile(alpha,98))= prctile(alpha,98);
alpha = medfilt1(alpha,20);
```

Code Block 6.2 Script to pre-process alpha current values.

The processed data is then saved as .csv files. The data is used to create two sets of files Input and Output. Input consists of the five inputs to the neural network. The five inputs are w_r(k), i^e_{qs}(k), i^e_{qs}(k-1), i^s_{ds}(k-1) and i^s_{qs}(k-1). Now k stands for latest value and k-1 for previous values. The output file consists of i^s_{ds}(k) and i^s_{qs}(k), which are the latest values. Code to carry out the above is given below

```
% Obtain number of rows in data set
L = Length (speed);
% Input file is created: latest values go from 2 to end,
previous values
% from 1 to L-1
Input = horzcat(speed(2:(L)),Iqs(2:(L)),Iqs(1:(L-
1)),alpha(1:(L-1)),beta(1:(L-1)));
Output = horzcat(alpha(2:(L)),beta(2:(L)));
% Data is written to the folder
dlmwrite('Input.csv',Input,'precision',5) ;
dlmwrite('Output.csv',Output,'precision',5) ;
```

Code Block 6.3 Script to prepare data for writing to folder

After generating the data, a script containing the neural network code is run. The training algorithm is Levenberg-Marquardt one. The code run till we have either 1000 iterations or 6 failed validation checks in the test set.

The data is initially normalized to so that all values lie between -1 to 1 and finally de-normalized when giving the output. Also the data set is randomly divided into three parts: Training containing 70% of data, 15 % for validation and 15 % for testing purposes. The script runs the code and finally generates a Simulink block-set.

The code to train the neural network is given in the Code Block 6.4 as given in next page.

```
%   Input - input data.
%   Output - target data.

x = Input';
t = Output';

% Training Algorithm

trainFcn = 'trainbr';  % Levenberg-Marquardt

% Create a Fitting Network
hiddenLayerSize = 15;
net = fitnet(hiddenLayerSize,trainFcn);

% Input and Output Pre/Post-Processing Functions
(Normalization)

net.input.processFcns = {'mapminmax'};
net.output.processFcns = {'mapminmax'};
```

Code Block 6.4.1 Neural network training code up to normalization

```
% Setup Division of Data for Training, Validation,
Testing
% For a list of all data division functions type: help
nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help
nnperformance
net.performFcn = 'mse'; % Mean squared error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns =
{'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};

% Train the Network
net.trainParam.epochs = 1000;
net.trainParam.max_fail = 6;
[net,tr] = train (net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform (net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t  .* tr.valMask{1};
testTargets = t  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view (net)
genism (net);
```

Code Block 6.4.2 Neural network training code up-to Simulink block generation

6.3 Simulation of Neural Network Controller based IFOC of an Induction Motor

Now the trained neural network Simulink block is simulated with the modified architecture as shown below

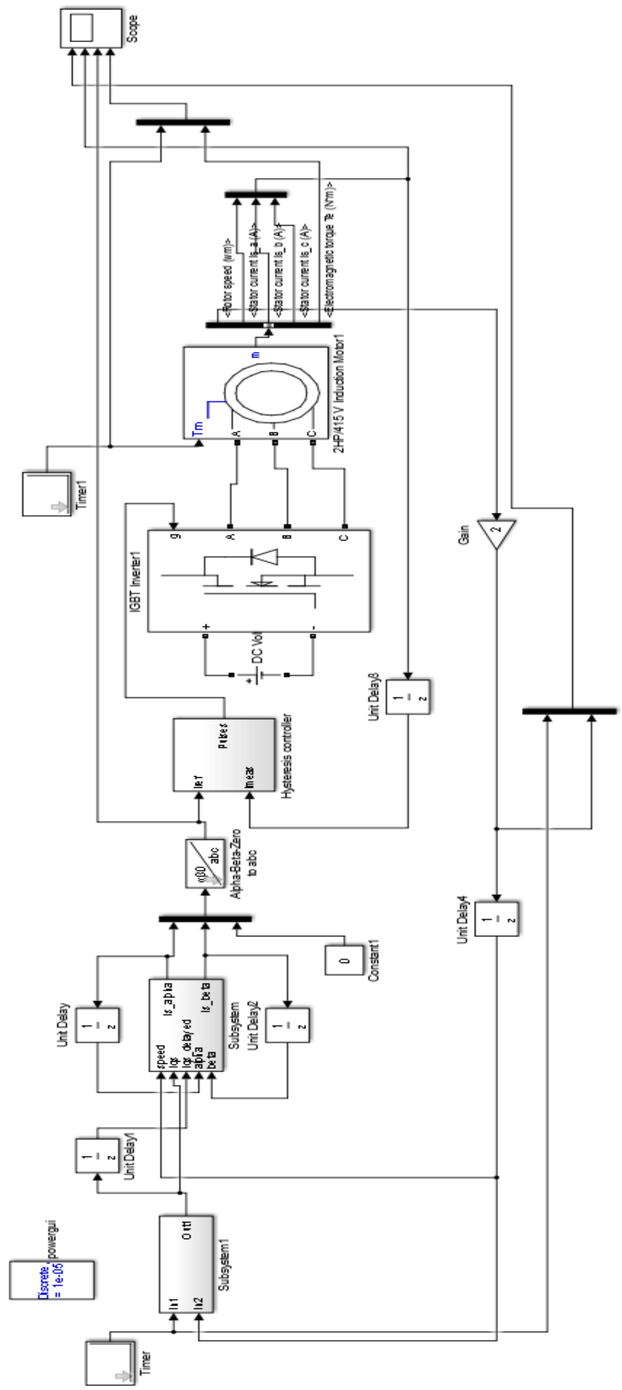


Figure 6.7 Simulink diagram of neural network based field oriented control of induction motor

The unit delay value is the same as the one used while collecting data, it has a delay value of 100e-6. The PI values need to be tuned afresh while simulating the new model. The motor parameters are the same as before.

6.2.3.1 Performance of new architecture for Reference Torque = 5 N.m, Reference Speed = 200 rad/s

Initially the reference speed is 100 at time t = 0s. Reference speed is increased to 200 at 1 sec and then torque is changed from 0 to 5 N.m at t = 2s. The simulation is run for 3 sec.

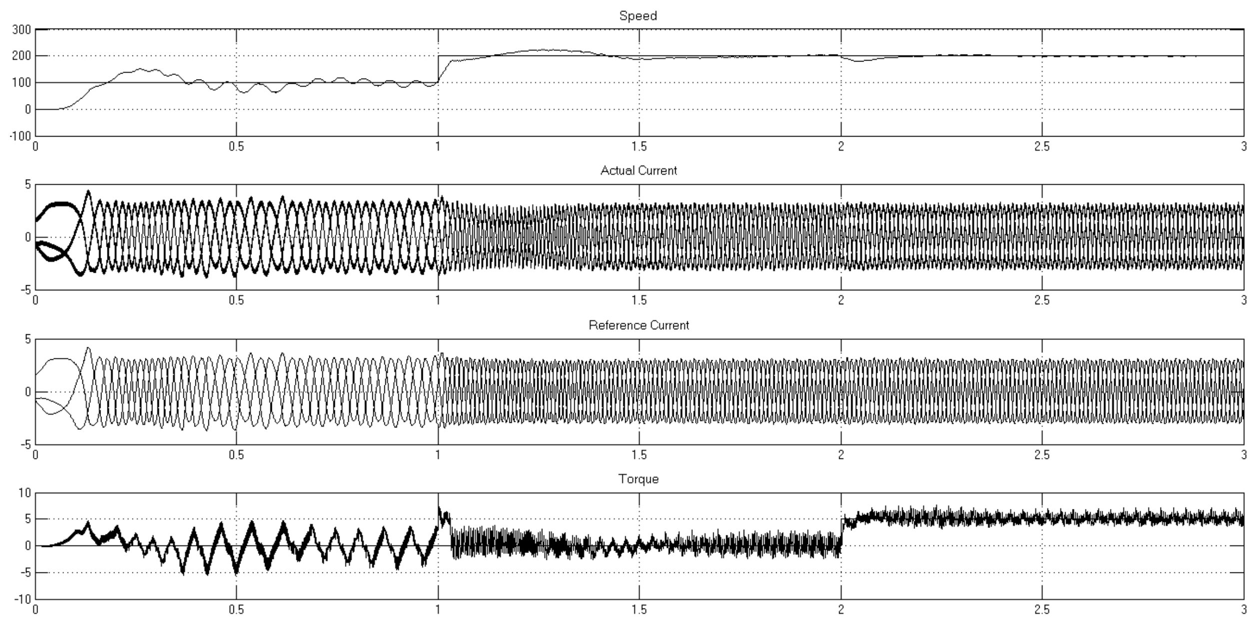


Figure 6.8 Simulation with load being 5 Nm and speed reference being 200 rad/s

Plot 1 : Reference and motor speed(rad/s), Plot 2 : Actual Current(I), Plot 3 : Reference Current(I), Plot 4 : Reference and actual motor torque(Nm)

Eventually the torque and speed reach their steady state values. Torque sees significant amount of fluctuations during initial transition time before eventually steadying.

6.2.3.1 Performance of new architecture for Reference Torque = 3 N.m, Reference Speed = - 150 rad/s

Initially the reference speed is 100 at time t = 0s. Reference speed is increased to -200 at 1 sec and then torque is changed from 0 to 3 N.m at t = 2s. The simulation is run for 3 sec.

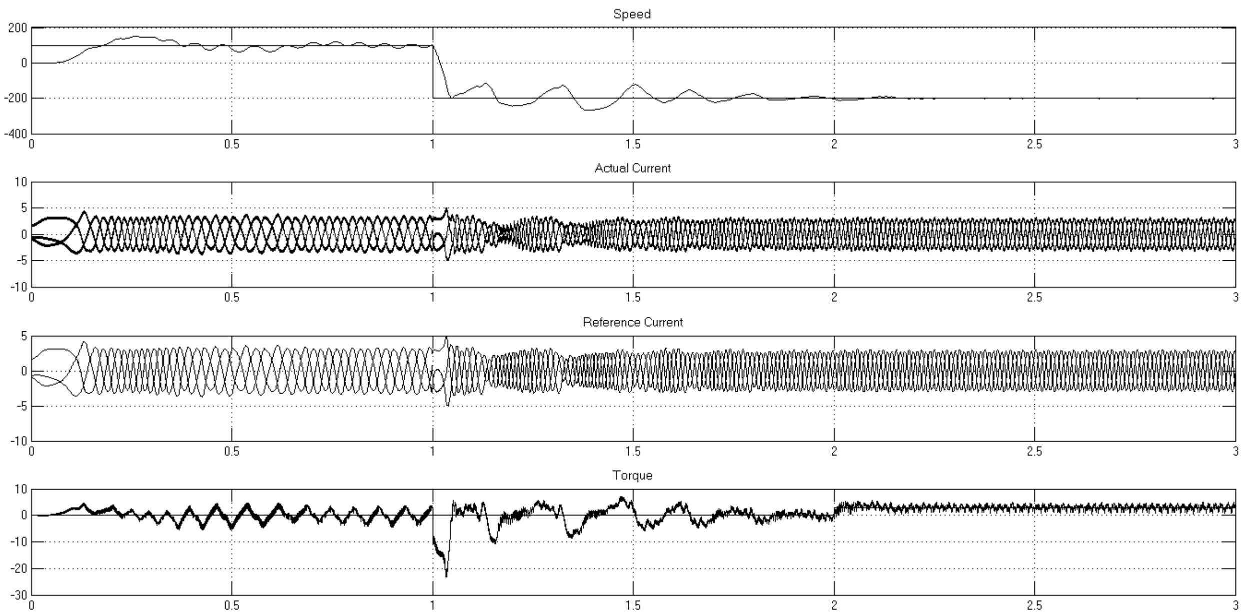


Figure 6.9 Simulation with load being 3 Nm and speed reference being -200 rad/s

Plot 1 : Reference and motor speed(rad/s), Plot 2 : Actual Current(I), Plot 3 : Reference Current(I), Plot 4 : Reference and actual motor torque(Nm)

Here the motor has been given reverse speed command, which it is able to reach towards the end of the steady state along with the torque reference of 3 N.m.

The performance of neural controller is good in the sense that the motor is eventually able to reach it steady state values but there is presence of significant torque fluctuations under transient state.

CHAPTER 7

Conclusion

An overview of AC-AC conversion systems was done with emphasis on Matrix Converters. The various direct control techniques including starting from Venturini to modified Venturini were studied and implemented in Simulink (MATLAB 2014a). Matrix converters owing to their inherent nature of no-dc link provide dynamic power transfer with ability to control both voltage and frequency and a low IDF at the input side makes them ideal for induction motor applications.

Vector control owing to the decoupling property gives a fast and independent control over both torque and speed of an induction motor. Using some mathematical transformations, the control of two dependent quantities is made de-coupled and independent. However the mathematics behind the control is highly complicated and difficult to implement as a control logic. The control thereby is computationally heavy and its engineering is difficult owing to the complexity of it.

Artificial Neural Networks as a phenomenon have been a rage in the scientific community with them being the brain behind several modern day marvels such as Google Deep mind, autonomous cars etc. Their ability to fit complex non-linear functions and also act as black box means of computations makes them a perfect complement for the vector control. Their inherent parallel nature makes computations fast it can process data fast and once a network has been trained, it can carry out the computations at a very high speed.

To reduce the recurrent computing involving in a vector control and to also simplify the vector control loop from an engineering point of view, a neural controller was trained and implemented as a part of field oriented control. However, the dynamic nature of the induction motor drive with high fluctuations in values during transients made the training of the neural network block set challenging and difficult. The realization is that the data used for training and its quality are of most importance here.

The ANN simulation was done with two-level VSI instead of matrix converter due to time constraints. Also it was believed that the first step in implementing the entire structure would involve perfecting the training and implementation of ANN before venturing out further.

The performance of ANN in this particular structure can improved many folds by experimenting and working around with type of data used to train, the training algorithm etc. With advent of hardware specifically designed for running ANNs, their application in this particular can be expected to rise.

References

[1] Muhammad H. Rashid, “Power Electronics Handbook”, Canada, Academic Press, 2001

[2] Ashour, H.A, Ibrahim R.A, “Comparison Analysis of AC Voltage Controllers Based on Experimental and Simulated Animated Studies,” The 2006 International Conference on Computer Engineering and Systems, 5-7 Nov, 2006.

[3] I. Takahashi, Y. Itoh, “Electrolytic Capacitor-Less PWM Inverter“, in Proceedings of the IPEC’90, Tokyo, Japan, , pp. 131 – 138, April 2 – 6, 1990.

[4] J.R. Rodriguez, J.W Dixon, J.R Espinoza, J. Pontt and P. Lezanna, “PWM regerative rectifier: state of art”, IEEE Transactions on Industrial Electronics, vol 52, no1, pp.5-z2, 2005

[5] Gyugi Lazlo,” Generalized Theory of Static Power Frequency Changers”, PhD thesis, University of Salford, 1970

[6] Karaca, Hulusi, and Ramazan Akkay. "Control of Venturini method based matrix converter in input voltage variations." *Proceedings of the International Multi Conference of Engineers and Computer Scientists IMECS, Hong Kon*. Vol. 2. 2009.

[7] Wheeler, P. W., Rodriguez, J., Clare, J. C., Empringham, L., & Weinstein, A. (2002). Matrix converters: a technology review. *Industrial Electronics, IEEE Transactions on*, *49*(2), 276-288.

[8] Alesina, A., & Venturini, M. G. (1989). Analysis and design of optimum-amplitude nine-switch direct AC-AC converters. *Power Electronics, IEEE Transactions on*, *4*(1), 101-112.

[9] A. Brandt, “Der Netztaktumrichter,” Bull. ASE, vol. 62, no. 15, pp. 714–727, July 1971.

[10] W. Popov, “Der Direktumrichter mit zyklischer Steuerung,” Elektrie, vol. 29, no. 7, pp. 372–376, 1975

[11] V. Jones and B. Bose, “A frequency step-up cycloconverter using power transistors in inverse-series mode,” Int. J. Electron., vol. 41, no. 6, pp. 573–587, 1976

[12] M. Steinfels and P. Ecklebe, “Mit Direktumrichter Gespeiste Drehstromantriebe für den Industriellen Einsatz in einem Weiten Leistungsbereich,” Elektrie, vol. 34, no. 5, pp. 238–240, 1980.

[13] P. Ecklebe, “Transistorisierter Direktumrichter für Drehstromantriebe,” Elektrie, vol. 34, no. 8, pp. 413–433, 1980.

[14] A. Daniels and D. Slattery, “New power converter technique employing power transistors,” Proc. Inst. Elect. Eng., vol. 125, no. 2, pp. 146–150, Feb. 1978.

[15] “Application of power transistors to polyphase regenerative power converters,” Proc. Inst. Elect. Eng., vol. 125, no. 7, pp. 643–647, July 1978.

[16] E. Stacey, “An unrestricted frequency changer employing force commutated thyristors,” in Proc. IEEE PESC’76, 1976, pp. 165–173.

[17] M. Venturini, “A new sine wave in sine wave out, conversion technique which eliminates reactive elements,” in Proc. POWERCON 7, 1980, pp. E3_1–E3_15

[18] M. Venturini and A. Alesina, “The generalized transformer: A new bidirectional sinusoidal waveform frequency converter with continuously adjustable input power factor,” in Proc. IEEE PESC’80, 1980, pp. 242–252

[19] J. Rodriguez, “A new control technique for AC–AC converters,” in Proc. IFAC Control in Power Electronics and Electrical Drives Conf., Lausanne, Switzerland, 1983, pp. 203–208.

[20] J. Oyama, T. Higuchi, E. Yamada, T. Koga, and T. Lipo, “New control strategy for matrix converter,” in Proc. IEEE PESC’89, 1989, pp. 360–367.

[21] P. D. Ziogas, S. I. Khan, and M. H. Rashid, “Analysis and design of forced commutated cycloconverter structures with improved transfer characteristics,” IEEE Trans. Ind. Electron., vol. IE-33, pp. 271–280, Aug. 1986

[22] P. Ziogas, S. Khan, and M. Rashid, “Some improved forced commutated cycloconverter structures,” IEEE Trans. Ind. Applicat., vol. 1A-21, pp. 1242–1253, Sept. /Oct. 1985

[23] M. Braun and K. Hasse, “A direct frequency changer with control of input reactive power,” in Proc. IFAC Control in Power Electronics and Electrical Drives Conf., Lausanne, Switzerland, 1983, pp. 187–194

[24] G. Kastner and J. Rodriguez, “A forced commutated cycloconverter with control of the source and load currents,” in Proc. EPE’85, 1985, pp. 1141–1146.

[25] L. Huber, D. Borojevic, and N. Burany, “Voltage space vector based PWM control of forced commutated cycloconvertors,” in Proc. IEEE IECON’89, 1989, pp. 106–111.

[26] C. L. Neft and C. D. Schauder, “Theory and design of a 30-HP matrix converter,” IEEE Trans. Ind. Applicat., vol. 28, pp. 546–551, May/June 1992

[27] E. Wiechmann, J. Espinoza, L. Salazar, and J. Rodriguez, “A direct frequency converter controlled by space vectors,” in Proc. IEEE PESC’93, 1993, pp. 314–320

[28] L. Huber and D. Borojevic, “Space vector modulator for forced commutated cycloconverters,” in Conf. Rec. IEEE-IAS Annu. Meeting, 1989, pp. 871–876.

[29] L. Huber, D. Borojevic, and N. Burany, “Analysis design and implementation of the space-vector modulator for forced-commutated cycloconvertors,” Proc. Inst. Elect. Eng., pt. B, vol. 139, no. 2, pp. 103–113, Mar. 1992.

[30] L. Huber, D. Borojevic, X. Zhuang, and F. Lee, “Design and implementation of a three-phase to three-phase matrix converter with input power factor correction,” in Proc. IEEE APEC’93, 1993, pp. 860–865.

[31] L. Huber, D. Borojevic, and N. Burany, "Digital implementation of the space vector modulator for forced commutated cycloconverters," in Proc. IEE PEVD Conf., 1990, pp. 63–65.

[32] L. Huber and D. Borojevic, "Space vector modulated three phase to three phase matrix converter with input power factor correction," IEEE Trans. Ind. Applicat., vol. 31, pp. 1234–1246, Nov. /Dec. 1995

[36] J. Zubek, A. Abbondanti, and C. J. Norby, "Pulsewidth modulated inverter motor drives with improved modulation," IEEE Trans. Ind. Applicat., vol. 11, pp. 695–703, Nov./Dec. 1975.

[37] K. Hasse, "Zur Dynamic Drehzahlgeregelter Antrieebe Mit Stromrichter Gespeisten Asyn-chron Kuzschlublaufermaschinen," Ph.D. dissertation, Technische Hochschule Darmstadt, Darmstadt, Germany, 1969.

[38] F. Blaschke, "Das Verfahren der Feldorientierung zur Regelung der Drehfeldmaschine," Ph.D. dissertation, Univ. Braunschweig, Braunschweig, Germany, 1973

[39] "The principle of field orientation as applied to the new transvector closed-loop control system for rotating-field machines," Siemens Rev., vol. 34, pp. 217–220, May 1972.

[40] P. Vas, A. F. Stronach, and M. Neuroth, "DSP-controlled intelligent high-performance ac drives present and future," in IEE Colloq. Vector Control and Direct Torque Control of Induction Motors, Oct. 1995, pp. 7/1–7/8

[41] Y. S. Kung, C. M. Liaw, and M. S. Ouyang, "Adaptive speed control for induction motor drives using neural networks," IEEE Trans. Ind. Electron., vol. 42, pp. 25–32, Feb. 1995

[42] D. L. Sobczuk and P. Z. Grabowski, "DSP implmentation of neural network speed estimator for inverter fed induction motor," in Conf. Rec. IEEE IECON'98, 1998, pp. 981–985.

[43] P. Vas, A. F. Stronach, and M. Neuroth, "DSP-based speed-sensorless vector controlled induction motor drives using AI-based speed estimator and two current sensors," in Proc. IEE 7th Int. Conf. Power Electronics and Variable Speed Drives, 1998, pp. 442–446.

[44] H.-C. Lu, T.-H. Hung, and C.-H. Tsai, "Sensorless vector control of induction motor using artificial neural network," in Proc. IEEE Int. Symp. Circuits and Systems, vol. II, 2000, pp. 489–492

[45] Q. Xie, S. Wan, Y. Yi, J. Zhao, and Y. Shen, "Speed-sensorless control using Elman neural network," J. Syst. Eng. Electron., vol. 12, no. 4, pp. 53–58, 2001

[46] M. R. Buhl and R. D. Lorenz, "Design and implementation of neural networks for digital current regulation of inverter drives," in Conf. Rec. IEEE-IAS Annu. Meeting, 1991, pp. 415–423.

[47] D. R. Seidl, D. A. Kaiser, and R. D. Lorenz, "One-step optimal space vector pwm current regulation using a neural network," in Conf. Rec. IEEE-IAS Annu. Meeting, 1994, pp. 867–874.

[48] B. Burton, F. Kamran, R. G. Harley, T. G. Habetler, M. A. Brooke, and R. Poddar, "Identifcation and control of induction motor stator currents using fast on-line random training of a neural network," IEEE Trans. Ind. Applicat., vol. 33, pp. 697–704, May/June 1997.

[49] K. Madani, G. Mercier, M. Dinarvand, and J.-C. Dpecker, “A neurovector based electrical machines driver combining a neural plant identifier and a conventional vector controller,” in Proc. SPIE 2nd Conf. Applications and Science of Computational Intelligence II, 1999, pp. 476–485.

[50] L. R. Valdenebro, J. R. Hernandez, and E. Bim, “A neuro-fuzzy based parameter identification of an indirect vector-controlled induction motor drive,” in Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics, 1999, pp. 347–352

[51] F. Blaschke, “The Principle Of Field Orientation As Applied To New Transvector Closed-Loop Control System For Rotating-Field Machines,” *Siemens Review*, vol.34, no.3, pp. 217-220, May 1972

[52] W. Leonard, “*Control of Electric Drives*”, New Delhi, Narosa Publications, 1985

[53] P.C. Krause, *Analysis of Electrical Machinery*, Prentice Hall, 1985.

[54] Lipo, Thomas A. "*Recent Progress in the Development of Solid State AC Motors Drives*." (1987): 125.

[55] J.M.D. Murphy and F.G. Turnbull, *Power Electronics Control of AC Motors*, Oxford Pergamum Press, 1988

[56] Daboussi, Z.; Mohan, N., "Digital simulation of field-oriented control of induction motor drives using EMTP," in *Energy Conversion, IEEE Transactions on* , vol.3, no.3, pp.667-673, Sep 1988.

[57] Rong-Jong Wai; Kuo-Min Lin, "Robust decoupled control of direct field-oriented induction motor drive," in *Industrial Electronics, IEEE Transactions on* , vol.52, no.3, pp.837-854, June 2005

[58] Malesani, Luigi, and Paolo Tenti. "A novel hysteresis control method for current-controlled voltage-source PWM inverters with constant modulation frequency." *IEEE Transactions on Industry Applications* 26.1 (1990): 88-92.

[59] Malesani, Luigi, Paolo Mattavelli, and Paolo Tomasin. "Improved constant-frequency hysteresis current control of VSI inverters with simple feedforward bandwidth prediction." *Industry Applications, IEEE Transactions on* 33.5 (1997): 1194-1202

[60] Pan, Ching-Tsai, and Ting-Yu Chang. "An improved hysteresis current controller for reducing switching frequency." *Power Electronics, IEEE Transactions on* 9.1 (1994): 97-104.

[61] Holmes, Donald Grahame, Brendan Peter McGrath, and Stewart Geoffrey Parker. "Current regulation strategies for vector-controlled induction motor drives." *Industrial Electronics, IEEE Transactions on* 59.10 (2012): 3680-3689.

[62] Krause, P.C.; Thomas, C.H., "Simulation of Symmetrical Induction Machinery," in *Power Apparatus and Systems, IEEE Transactions on* , vol.84, no.11, pp.1038-1053, Nov. 1965

[63] Takahashi, I.; Noguchi, T., "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor," in *Industry Applications, IEEE Transactions on* , vol.IA-22, no.5, pp.820-827, Sept. 1986

[64] S.G. Choudhuri, “analysis and development of vector control of induction motor drive”, Ph.D dissertation, Dept. of Elec. Eng., Indian Institute of Technology, Delhi, India, 2004.

[65] A. Tilak Raja, “Comparative study of vector control abd direct torque control of induction motor drive”, M.Tech dissertation, Dept. of Elec. Eng., Indian Institute of Technology, Roorkee, India, 2008

[66] Gabriel, R., and W. Leonhard. "Microprocessor control of induction motor."*IEEE/IAS Int. Sem. Power Conv. Conf. Rec*. 1982.

[67] Leonhard, W. "’Control of AC Machines with the help of Microelectronics,’."*Proc. IFAC Symp. On Control In Power Elec. and Elec. Drives*. 1983