

# MASTER THESIS

**Title:** Open phase fault detection in PMSM drives based on the current signature analysis

**Submitted by:** Mr. VINAYAK JOSHI

**1st Academic Supervisor:** Prof. Dr. -Ing. Thomas Schumann

**2nd Academic Supervisor:** Prof. Dr. -Ing. Herbert Krauß

**Industrial Supervisor:** Dr. Giulio Corradi

Mr. Michael Zapke

**Completion Date:** 10<sup>th</sup> Jan 2015

**Student:**

**VINAYAK**

First (Given) Name

**JOSHI**

Last (Family) Name

Date of Birth:**10/09/1988**

Matr.-No.:**732281**

1st Academic Supervisor: **Prof. Dr. –Ing. Thomas Schumann**

2nd Academic Supervisor: **Prof. Dr. –Ing. Herbert Krauß**

Title: **Open phase fault detection in PMSM drives based on the current signature analysis**

**Abstract: (max 10 Lines)**

The Permanent Magnet Synchronous Motors (PMSM) drives usage in the Industrial applications has seen a tremendous growth. Requiring more intelligent and reliable motor control solutions which address the issues of diagnostic and performance is essential.

Traditional MCUs, ASSPs and DSP do not offer enough computing power to afford new requirements for more intelligent solutions and faster control loops. FPGA based SoC (System on Chip) has proven to be most prominent.

Hence using state of the art Xilinx All programmable SoC, the Master Thesis presents mainly the following,

1. An impulse response of the system using cross – correlation method in particular reference to PMSM drives for system identification.
2. Open phase fault algorithm for diagnosis and detection using motor current signature analysis.

In partial fulfilment of the requirements of the **University of Applied Sciences Hochschule Darmstadt (h\_da)** for the degree **Master of Science in Electrical Engineering** carried out in collaboration with **Industrial Enterprise**

Company: **Xilinx GmbH**

Address: **Willy-Brandt-Allee 4, 81829 Munich, Germany**

This Master Thesis is subject to a non-disclosure agreement between the University of Applied Sciences Hochschule Darmstadt (h\_da) and the industrial partner.

(Signature)

1st Academic Supervisor:

Student:

**VINAYAK**

First (Given) Name

**JOSHI**

Last (Family) Name

1st Academic Supervisor: **Prof. Dr. –Ing. Thomas Schumann**

2nd Academic Supervisor: **Prof. Dr. –Ing. Herbert Krauß**

### **Declaration**

I hereby declare that this thesis is a presentation of my original research work and that no other sources were used other than what is cited.

I furthermore declare that wherever contributions of others are involved, this contribution is indicated, clearly acknowledged and due reference is given to the author and source.

I also certify that all content without reference or citation contained in this thesis is original work.

I acknowledge that any misappropriation of the previous declarations can be considered a case of academic fraud.

Darmstadt, **10<sup>th</sup> Jan 2015**

(Date)

(Signature)

## **CONFIDENTIALITY**

In this Master thesis, the proprietary and confidential materials are included. This text has to be accessed with the express consent of Xilinx GmbH, Munich, Germany

## **ACKNOWLEDGEMENTS**

*This Master Thesis is successful with the support of many people. Firstly, I would like to express my whole hearted gratitude to **Hochschule Darmstadt, University of Applied Sciences**, and all my Professors for the quality technical education and valuable guidance. I would like to be thankful to my academic supervisors **Prof. Dr. Thomas Schumann** and **Prof. Dr. Herbert Krauß** for their encouragement and support during this Thesis.*

*It is indeed a unique pleasure and privilege for me to thank sincerely **Xilinx GmbH, Germany** for providing an opportunity to write my Master Thesis at Munich office. I am thankful to **Mr. Christoph Fritsch**, ISM Director for initiating and monitoring this Thesis.*

*I am expressing my sincere gratitude to my Industrial supervisor's for their constant support and guidance. In spite of their busy schedules and various commitment, they made themselves available as and when required. **Dr. Giulio Corradi**, Senior System Technical Architect, has been a great source for technical advancement during the Thesis. With his step wise and simple to complex work approach, it has been a great learning period during thesis. **Mr. Michael Zapke**, ISM Technical Marketing has been a backbone for all the activities undertaken pertaining to Thesis.*

*Finally, I thank my parents, family members and friends for their love, affection, support and standing by me to achieve my dreams of pursuing higher studies in Germany.*

## **Table of Contents**

CONFIDENTIALITY .....	I
ACKNOWLEDGEMENTS .....	II
ABSTRACT .....	VII
MOTIVATION .....	VIII
ABBREVIATIONS .....	XI
<b>Chapter 1 Introduction .....</b>	1
1.1 Xilinx Inc. ....	1
1.2 Master Thesis Introductions.....	2
1.3 Thesis Structure .....	3
<b>Chapter 2 Thesis Management .....</b>	4
<b>Chapter 3 Xilinx Zynq Intelligent Drives .....</b>	6
3.0 Zynq – 7000 AP SoC Intelligent Drives Kit .....	6
3.1 Xilinx All Programmable SoC .....	7
3.2 The Processing System (PS) .....	9
3.3 The Programmable Logic (PL) .....	10
3.3.1 Block RAM .....	10
3.3.2 Programmable Data Width .....	11
3.3.3 XADC (Analog-to-Digital converters) .....	11
3.4 PS – PL Interface .....	12
3.5 Software Tools Used.....	13
3.5.1 Vivado Design Suite .....	13
3.5.2 Xilinx Software Development Kit (SDK).....	14
3.6 SCILAB .....	14
3.7 FPGA Mezzanine Card (FMC).....	15
3.7.1 Motor Control FMC.....	15
3.7.2 Brushless DC Motor (BLDC) .....	16
3.8 Motor Control Drive .....	18
3.8.1 Zynq AP SoC based Motor Control.....	18
3.8.1.1 Single Chip Embedded control .....	18
3.8.2 QDESYS Drive .....	19

3.9 Architecture.....	20
3.10 Qdesys IP .....	23
3.10.1 Environment of the drive .....	23
<b>Chapter 4 System Identification .....</b>	<b>27</b>
4.1 System Identification Fundamentals.....	27
4.2 Cross – Correlation .....	27
4.2.1 Cross – Correlation for system identification .....	28
4.3 PRBS Fundamentals .....	30
4.3.1 PRBS Generation .....	30
<b>Chapter 5 Embedded Software Implementation .....</b>	<b>32</b>
5.1 Introduction.....	32
5.1.1 Architecture of Cross Correlation .....	32
5.1.2 Challenges Faced .....	33
5.2 Improved Methodology to inject PRBS.....	34
5.2.1 Improved architecture of the Cross – Correlation.....	34
5.3 Current Acquisition & Synchronization .....	35
5.3.1 Current Acquisition.....	35
5.3.2 Synchronization .....	35
5.4 PRBS Generation .....	36
5.4.1 Algorithm.....	36
5.5 Flow Chart .....	37
5.5.1 Synchronization Timing.....	38
5.6 SCILAB Implementation .....	38
5.6.1 Steps for Cross correlation.....	38
5.7 Results.....	39
5.7.1 Cross – Correlation .....	39
5.7.2 Impulse Response. ....	41
5.7.3 Magnitude and Phase plot.....	42
<b>Chapter 6 Open Phase Fault Detection.....</b>	<b>43</b>
6.1 Introduction.....	43
6.2 Clarke and Park Transforms in FOC .....	43

---

6.2.1 Overview.....	43
6.2.2 Clarke Transform ( $\alpha$ - $\beta$ reference frame).....	44
6.2.3 Park transformation (d-q reference frame) .....	46
6.3 Fault diagnosis and detection methods .....	47
6.3.1 Current Signature Analysis (Motor CSA).....	47
6.3.2 Vector Space Angle .....	47
6.3.3 Hough Transformation.....	47
<b>Chapter 7 Current Signature Analysis .....</b>	<b>48</b>
7.1 Introduction.....	48
7.2 Current Fault Signatures .....	48
7.2.1 Results and Simulation .....	48
7.3 PMSM drive model.....	53
7.3.1 Background .....	54
7.3.2 Inductance Measurement Procedure .....	55
7.4 Guide to measure inductance in d- and q-axis .....	56
7.4.1 d- axis inductance measurement .....	57
7.4.2 q- axis inductance measurement .....	57
7.5 Calculations and Results .....	58
7.5.1 SCILAB Simulation.....	59
<b>Chapter 8 Vector Space Angle.....</b>	<b>62</b>
8.1 Introduction.....	62
8.2 Algorithm Flow chart.....	62
8.3 Simulation and Results. ....	64
<b>Chapter 9 Hough Transformation .....</b>	<b>65</b>
9.1 Introduction.....	65
9.2 Hough Transformation.....	65
9.2.1 Introduction.....	65
9.2.2 Fundamentals .....	66
9.2.3 Algorithm.....	68
9.3 Perspective for open phase fault detection.....	69
9.4 HiL Implementation.....	71

---

9.4.1 Algorithm.....	71
9.4.2 Flow Chart .....	72
9.5 Results.....	73
9.5.1 Simulations .....	73
<b>Chapter 10 Analysis of the Results.....</b>	<b>86</b>
10.1 System Identification .....	86
10.2 Open Phase fault detection.....	87
<b>Chapter 11 Conclusion .....</b>	<b>88</b>
<b>Chapter 12 Future Work.....</b>	<b>89</b>
List of Figures: .....	90
List of Tables .....	90
References.....	91
Appendix.....	94

## **ABSTRACT**

The Permanent Magnet Synchronous Motors (PMSM) drives usage in the Industrial applications has seen a tremendous growth. Requiring more intelligent and reliable motor control solutions which address the issues of diagnostic and performance is essential.

Traditional MCUs, ASSPs and DSP do not offer enough computing power to afford new requirements for more intelligent solutions and faster control loops. FPGA based SoC (System on Chip) has proven to be most prominent.

Hence using state of the art Xilinx All programmable SoC, the Master Thesis presents mainly the following,

1. An impulse response of the system using cross – correlation method in particular reference to PMSM drives for system identification.
2. Open phase fault algorithm for diagnosis and detection using motor current signature analysis.

For system identification in the real embedded environment, a PRBS perturbation is injected into the system. The system response is captured using Hardware in the Loop (HiL) on the final product. This HiL is not available today using MCUs and, Xilinx SoC allows for the real time HiL.

Based on the current trajectories, a diagnosis and detection methodology is evaluated for open phase fault. The inductance and flux parameters necessary for the diagnosis are measured directly from the motor.

This Master Thesis proposes Hough Transformation as the analytically efficient methodology to detect the current trajectories under the Healthy/Fault condition of the motor, implementing the algorithm in SCILAB under HiL.

Hence, an accurate and effective control to output response of the system is derived and motor stator current trajectory is captured real time discriminating between the Healthy/Fault conditions of the motor.

## **MOTIVATION**

PMSM drives are extensively used in the time critical applications and Industrial drives. Early diagnosis and detection of abnormalities lead to avoid expensive failures.

The following are the motivational thoughts for the motor control development and hence, the Master Thesis to meet the Industrial demands.

### **1. Market Reason**

For years, industrial motor control applications used general purpose electronic devices such as microcontrollers (MCUs) and DSPs. These devices are designed with fixed hardware, leaving software as the only method for designers to update designs and limiting the development of application-specific functions.

In comparison, FPGA based SoC (System on Chip) can integrate processor, Industrial Ethernet/fieldbus standards, custom motor interfaces, and DSP functions in a device. FPGA based SoC give designers the freedom to create custom functions completely adapted to specific application requirements by enabling both hardware and software customization. They have the capability to implement functions in hardware, accelerating performance and giving the ability to develop application specific software in the processor.

Xilinx FPGA and All Programmable SoC – based Motor control solutions and platforms meet the critical timing and performance requirements posed by today's complex control algorithms such as Field Oriented Control (FOC). Unlike MCUs, ASSPs and DSPs that handle simple motor control and low clock speeds processed in software, Xilinx 28nm FPGAs and Zynq - 7000 All Programmable SoCs provide superior parallel-processing power, real time performance, fast computational speeds, and connectivity versatility[8].

### **2. Thesis Prospects**

Utilizing the beneficiaries from Xilinx All programmable SoC, a Motor Control Drive is developed by a company QDESYS capable of tackling compute-intensive functions in compliance with the Industry 4.0 project providing Gigabit Ethernet industrial networking connectivity.

Fig I show the existing Xilinx AP SoC with the Intelligent Motor control solution. FOC (Field Oriented Control) and an application running HiL simulation is employed in the solution.

## Why and What Master Thesis Add

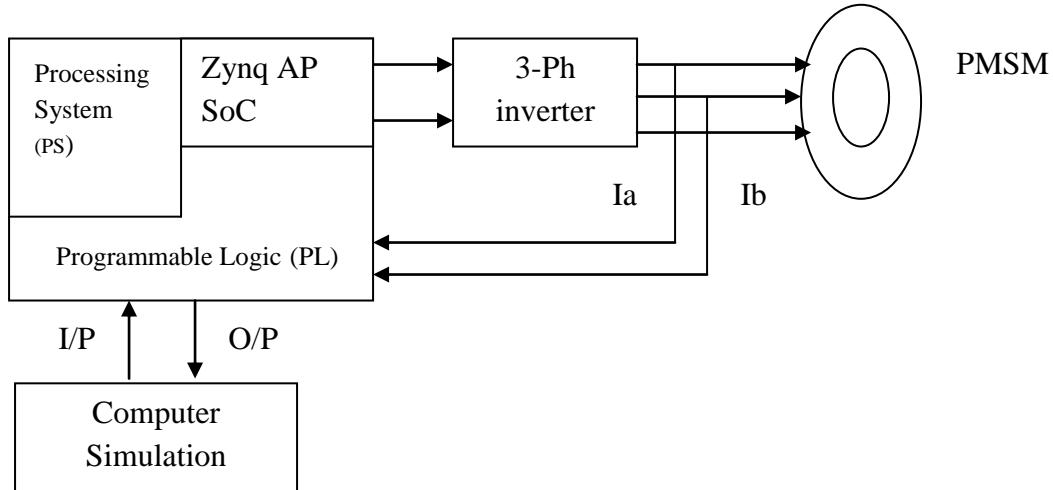


Fig I Xilinx AP SoC Motor control solution

In order to design and develop high performance control algorithms a good comprehension is necessary of the system behavior for input signal. The data sheet, containing the motor parameters that are available from the motor manufacturer would not contain the information about the system behavior when the motor is installed in application, technically magnitude and phase shift determination for the particular application is important. *Hence system identification using cross – correlation technique is dealt in this Thesis.*

With the possibility of quick response time and real time monitoring of the motor using Xilinx AP SoC, an efficient methodology to detect the open phase fault would give new avenues to Xilinx TRD (Target Reference Design). *Hence, open phase fault diagnosis and detection methodology are evaluated and the efficient algorithm is proposed and developed in this Thesis.*

In the recent years, number of papers and work has been published for system identification using cross correlation method [15], [16], from this Master Thesis, work done on time to collect the data and effective system identification compared to the earlier work can be clearly distinguished.

In [23], [24], the authors have studied stator faults and single phase for open circuit faults. Authors in [18] have given motor model along with the flux and inductances trajectories for fault detection. Analytical study of the stator trajectory is necessary to take decision on the condition of the motor.

The Master Thesis distinguishes the various methods to identify the open phase fault and proposes an image processing algorithm, Hough transformation, is an analytically efficient algorithm to detect the open phase fault.

Fig II shows Master Thesis work which would add to the existing Xilinx Motor Control Solution.

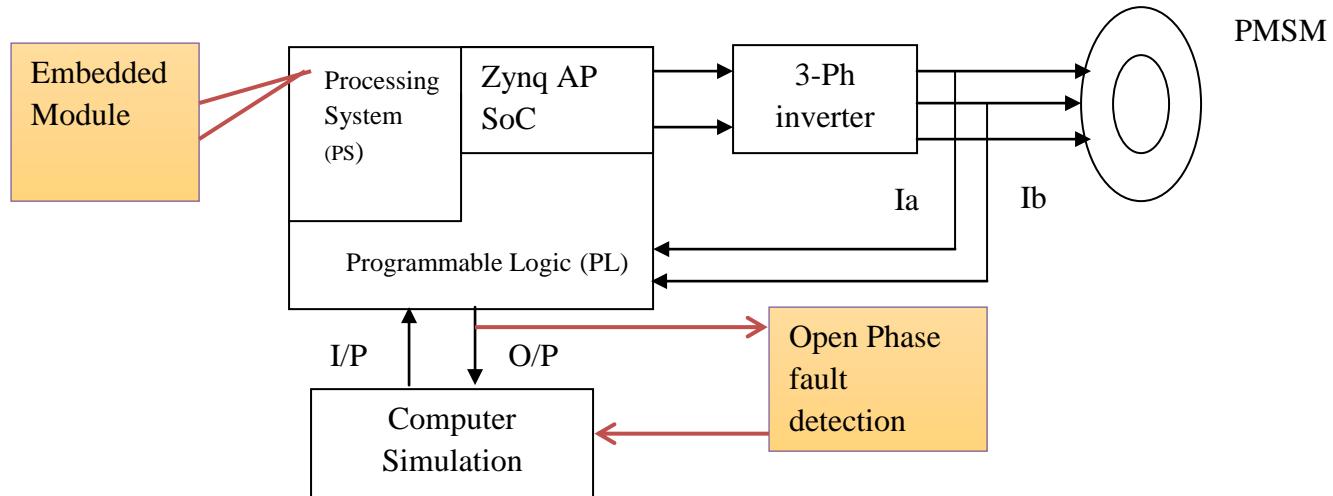


Fig II Xilinx AP SoC Motor Control + Thesis

## ABBREVIATIONS

Term	Definition
<b>Xilinx AP</b>	Xilinx All Programmable
<b>SoC</b>	System On Chip
<b>PMSM</b>	Permanent Magnet Synchronous Motors
<b>MCUs</b>	Microcontrollers
<b>ASSP</b>	Application Specific Standard Product
<b>DSP</b>	Digital Signal Processor
<b>PRBS</b>	Pseudo Random Binary Sequence
<b>HiL</b>	Hardware in the Loop
<b>FFT</b>	Fast Fourier transform

<b>FPGA</b>	Field Programmable Gate Array
<b>FOC</b>	Field Oriented Control
<b>Xilinx TRD</b>	Xilinx Target Reference Design
<b>API</b>	Application Programming Interface
<b>MPCore</b>	Multi Core Processor
<b>PS</b>	Processing System
<b>PL</b>	Programmable Logic
<b>ACP</b>	Accelerator Coherency Port
<b>BSPs</b>	Board Support Package
<b>POSIX</b>	Portable Operating System Interface
<b>NI LabView</b>	National Instruments LABVIEW

# **Chapter 1**

## **Introduction**

*In this Chapter, information about the company Xilinx Inc. and its major contributions to the semiconductor Industry is provided. Introduction to the Master Thesis work and structure of the Thesis writing is described.*

### **1.1 Xilinx Inc.**

Xilinx [1] is an American technology company, primarily a supplier of programmable logic devices. It is known for inventing the Field Programmable Gate Array (FPGA) and as the first semiconductor company with a fabless manufacturing model. Founded in Silicon Valley in 1984, the company is headquartered in San Jose, California, with additional offices in Longmont, Colorado; Dublin, Ireland; Singapore; Germany; Hyderabad, India; Beijing, China; Shanghai, China and Tokyo, Japan.

Xilinx is the world's leading provider of All Programmable FPGAs, SoCs and 3D ICs. These industry-leading devices are coupled with a next-generation design environment and IP to serve a broad range of customer needs, from programmable logic to programmable systems integration.

Recent innovations have transformed Xilinx from its programmable logic heritage to an 'All Programmable' company, creating and integrating 'All' forms of hardware, software, digital, and analog programmable technologies into its All Programmable FPGAs, SoCs and 3D ICs. These devices combine the value of programmable systems integration with embedded intelligence and flexibility, enabling the rapid development of highly programmable and smarter systems.

Xilinx customers demand ever more programmable solutions to address the industry's 'programmable imperative', caused by exponentially increasing silicon design costs and risks associated with both ASICs and ASSPs. They also demand a significantly faster time to differentiation and integration relative to these ASIC and ASSPs, and relative to their own competitors. This is enabled by the industry's first ASIC and SoC-strength design suite; Vivado.

Xilinx takes pride in supporting its community both externally and internally. Xilinx has a history of developing programs for its employees and surrounding communities that provide a social impact through outreach, volunteerism, teambuilding and philanthropy. Areas of focus include education, health, arts and social services. Xilinx culture is represented by a pride in leadership, a passion for excellence, and personal growth.

*Together, over 3,500 Xilinx employees worldwide join forces to evolve and actively transform the company and its technologies to shape the future.*

## 1.2 Master Thesis Introductions

With the fourth Industrial revolution, Industry 4.0 promoting the computerization of the manufacturing industry, the goal is the smart factory. On technological basis this is characterized by the cyber-physical systems and Internet of Things [2]. Addition of intelligence to monitoring and control of the industrial motor would leverage higher level of system performance.

Motors are the ubiquitous in Industrial applications and account for more than 66% of the electrical power consumed in the industrial systems. As the cost of the power continues to rise and the automation of the factories increases, motor efficiency becomes increasingly important.

To achieve optimum efficiency, the motor control electronics read the motor's current and voltage and performs a series of mathematical operations using computed errors and corrections. The calculations result into optimized waveforms to drive the motor. All of these need to be completed in timely manner, before the motor's feedback readings become obsolete. The shorter the loop time, the faster the motor's response to changes, leading to less ripple and less energy dissipated by the motor. This results in a motor drive system with greater precision and efficiency [3].

Effective diagnostic and detection of the abnormalities is necessary to avoid expensive failures in the Industry. Open phase fault detection is one such abnormality, which lead to damage of the motor and components in an Industrial environment. Before the development of complex algorithms, a good comprehension of the system parameters and system identification is necessary

In general, system identification is divided into parametric and Non-parametric methods. In parametric methods, the identification amounts to estimation of model parameters. In nonparametric methods, identification is used to directly compute the system response. Non-parametric methods include: correlation analysis, transient-response analysis and frequency response, Fourier or spectrum analysis [4].

This Master Thesis is focused on the non-parametric identification, with the objective of deriving impulse response and hence, frequency response of the system. Cross correlation analysis method is mainly used for the identification purpose. This is achieved by injecting N-bit Pseudo Random Binary Sequence (PRBS) into the system and measuring the system response. In order to record the dynamic response from the motor, synchronization between the PRBS injection and acquisition of the current data is achieved using ARM Cortex A-9 processor in Xilinx All programmable (AP) SoC, Zynq.

The second part of the Thesis deals with the real time diagnosis and algorithm development for open phase fault detection using the stator current trajectory by performing Hardware in the Loop (HiL) using SCILAB on the final product, Xilinx AP SoC. The current trajectory is studied to discriminate between the healthy and fault condition of the motor. Hough transformation is implemented to analytically detect the open phase fault.

Hence, this Master Thesis effectively derives impulse response of the system along with the identification of system parameters, namely Magnitude and Phase. Secondly, a computationally accepting algorithm to detect the open phase fault in PMSM drives is implemented and simulated.

### **1.3 Thesis Structure**

The Master Thesis report begins with Introduction followed with the *Chap. 2 Milestones and Thesis Management*, the different milestones laid and thesis planning.

*Chap.3 Hardware and Software Description*, evaluation board and daughter card used along with the development environment is discussed.

*Chap. 4 System Identification*, a theory behind using cross correlation method is described

*Chap.5 Embedded Software implementation*, PRBS generation and injection in embedded environment, coding and usability are discussed.

*Chap.6 Open Phase Fault Detection*, concepts and fault detection methods.

*Chap. 7 Current Signature Analysis*, study and algorithm to discriminate condition of the motor

*Chap. 8 Vector Space Angle*, open phase fault determination using current vector angle.

*Chap.9 Hough Transformation*, Theory and implementation of the transformation to retrieve the information of current vector trajectory.

*Chap 10, 11 & 12*. Results, conclusion and Future work is reported and discussed.

# Chapter 2

## Thesis Management

*In this chapter, the milestones identified for the entire Master Thesis work and the plan followed is discussed.*

Milestones for Master Thesis					
Month/Week	1	2	3	4	5
July			1. Basic study on transfer function, impulse response while manipulating the coefficients of the transfer function. 2. Study on System Identification using correlation techniques.	3. System identification(Gain and Period) excercise for a given Transfer function. 4. SCILAB Simulation analysis of the simple correlation techniques 5. Understanding and simulation of the PRBS injection.	6. Correlation of the PRBS and given Transfer Function. 7. Study on FFT for the calculation of the magnitude and Phase. 8. Studying the alternative and efficient method of correlation(FFT'ng individual data)
Aug	1. Simulation of the FFT method for an simple application. 2. Magnitude and Phase calculation by FFT'ng the correlated data. 3. Getting familiar with the QDESYS Drive by running the sample demo applications	1. Studying the QDESYS drive library(Motor_Example) for the real time data acquisition and display	1. Start with simple data acquisition from the motor by sending a simple input data.	1. Applying a PRBS signal to the the Motor drive and checking the drive response. 2. Analysing the PRBS time shift response from the drive.	
Sept	1. Derving the Impulse response similar to done in offline using PRBS. 2. Determination of tuning parameters.	1. Study on reference frames(Clarke/Park transformation). [Contd 2]	2. Simple algorithm development to understand the alpha - beta frame.	1. Fault detection algorithm development.[Reading]	
Oct	[Contd]	1. Algorithm development	[Contd]	1. Status check of the work	[Contd] with alogrithm
Nov	[Contd] with algorithhm development Mid - Term presentation [Based on availability of Company supervisors and Prof.]	Working condition check of the algorithm. 1. Start with the preparation of the Thesis report.	1. Study on the different kind of faults	2. SCILAB programming to compare the faults	[Contd 2]
Dec	[Contd 2]	Cross verification from	Mr. Michael and Dr. Giulio	Finalising/Modification of the work.	Weihnachten feiern
Jan	Buffer time	Buffer time			Weihnachten feiern

*Fig 2.1 Milestones for Master Thesis*



Milestones Identified



Milestone prolonged and achieved



Thesis work verification

Fig 2.1 shows the milestones and planning for the Master thesis designed at the start of the Thesis.

The design shows the details based on the key deliverables from the Xilinx Industrial supervisors. The Key deliverables are as follows:

1. *Describing the Xilinx-QDESYS Electric Drive major components to make the reader familiar with the Drive*
2. *Using the SCILAB environment to drive the Drive in hardware in the loop and describing its major components*
3. *Using the SCILAB environment to measure the drive response using a PRBS sequence to determine the tuning parameters as described.*
4. *Using the SCILAB environment to detect faults in the motor connections based on the PBRS response as well as the pattern generated by the currents in the  $\alpha$ - $\beta$  reference frame.*
5. *Using the SCILAB environment to compare the failures from the motor mode based on the d-q frame.*
6. *Describing the results and the future work*

The approach throughout the Thesis was developing a simple model and executing, before endeavoring complex systems.

As we can see in the plan, the first few weeks were dedicated to the theoretical study and simple exercises. In the subsequent weeks the Milestones status are as shown in the Table 2.1.

Milestone	Month-2014	Status	Prolongation
M - 1	August	Achieved	No
M - 2	September	Prolonged	Yes - Time spent on Impulse response derivation
M - 3	November	Prolonged	Yes - Shift in MidTerm Presentation
M - 4	December	Achieved	No

*Fig 2.2 Milestones status*

Along with the above milestones, 2 major verification milestones to check the status of the Thesis were identified. They were achieved on the respective planned time

The subsequent chapters are also designed and documented as per the deliverables with the experimental results.

# Chapter 3

## Xilinx Zynq Intelligent Drives

In this Chapter, the Motor control hardware kit and software tools used along with the technical details and its importance to the Master Thesis are described. Xilinx AP SoC is a most profound and complex architecture, although everything inside the Zynq is not covered, instead the modules related to thesis are explained.

### 3.0 Zynq – 7000 AP SoC Intelligent Drives Kit

The Zynq – 7000 AP SoC Intelligent Drives Platform provides all the necessary hardware, design tools, and reference designs for industrial embedded control systems. The platform is ideal for designs requiring high performance motion control and/or industrial networking capability.

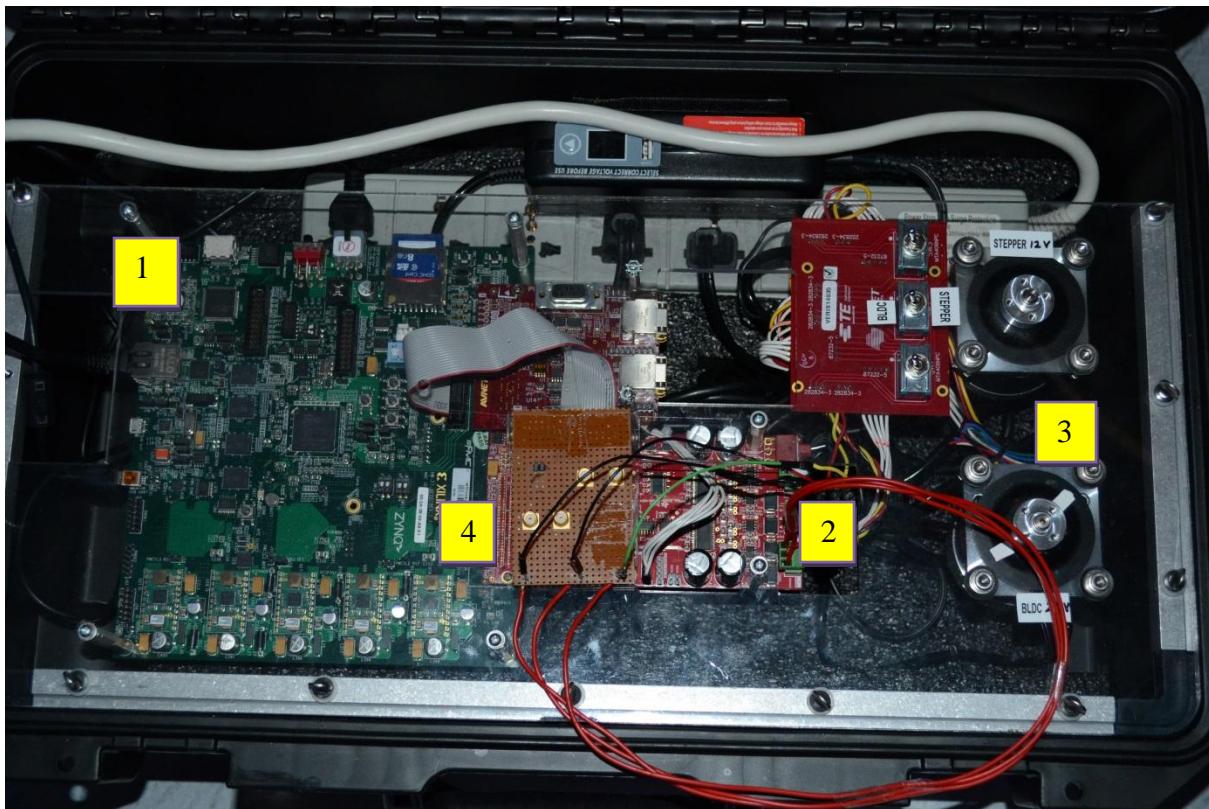


Fig 3.0 Xilinx Intelligent Motor Control Kit

The Thesis makes use of the following hardware kits to drive the motor. Referring to Fig 3.0

1. Zynq – 7000 (EK – Z7 – ZC702 – CES – G), which is described in detail in the section 3.2.
2. Avnet Motor Control FMC module
3. BLDC Motor
4. Phase handling tiny hardware

### **3.1 Xilinx All Programmable SoC**

Xilinx All Programmable SoCs are processor-centric platforms that offer software, hardware and I/O programmability in a single chip. The multicore processor and industry-leading programmable logic enable better processing systems with fewer devices and faster.

Like traditional SoCs, the processor-centric approach allows the processor to boot first, while a software-centric approach enables control and partial reconfiguration of the programmable logic to optimize system performance and power management to meet varying operating environments.

Xilinx All Programmable SoC portfolio provides multiple benefits in terms of programmable systems integration, increased system performance, BOM cost reduction and ultimately total power reduction.

Based on the Xilinx All Programmable SoC architecture, the Zynq - 7000 All Programmable SoCs, enable extensive system level differentiation, integration, and flexibility through hardware, software, and I/O programmability

#### **3.1.1 Overview of Zynq – 7000 All programmable SoC**

The Zynq-7000 family is based on the Xilinx All Programmable SoC (AP SoC) architecture. These devices are the results of integration of a dual-core ARM Cortex –A9 MPCore as Processing System (PS) and the Xilinx Programmable Logic (PL) in a single device, thus delivering high performance and low power in 28nm device technology. Besides the dual ARM Cortex A9 MPCore CPUs, the PS includes also on-chip memory, I/O interfaces and memory interfaces [5].

The Zynq – 7000 all programmable devices family is able to serve a wide range of applications based on the scalability and flexibility of an FPGA. The PS, PL and I/O resources are varying between the devices within the same family. The processors in the PS always boot first, and then PL will configure next with bit stream.

Figure 3.1 illustrates the functional block diagram of the Zynq – 7000 All Programmable SoCs.

The Zynq-7000 family offers the flexibility and scalability of an FPGA, while providing performance, power, and ease of use typically associated with ASIC and ASSPs. The range of devices in the Zynq-7000 All Programmable SoC family allows designers to target cost-sensitive as well as high-performance applications from a single platform using industry-standard tools. While each device in the Zynq-7000 family contains the same PS, the PL and I/O resources vary between the devices.

The Zynq-7000 architecture enables implementation of custom logic in the PL and custom software in the PS. It allows for the realization of unique and differentiated system functions. The integration of the PS with the PL allows levels of performance that two-chip solutions (e.g., an ASSP with an FPGA) cannot match due to their limited I/O bandwidth, latency, and power budgets.

Xilinx offers a large number of soft IP for the Zynq-7000 family. Stand-alone and Linux device drivers are available for the peripherals in the PS and the PL. The Vivado® Design Suite development environment enables a rapid product development for software, hardware, and systems engineers. Adoption of the ARM-based PS also brings a broad range of third-party tools and IP providers in combination with Xilinx's existing PL ecosystem. The inclusion of an application processor enables high-level operating system support, e.g., Linux. Other standard operating systems used with the Cortex-A9 processor are also available for the Zynq-7000 family. The PS and the PL are on separate power domains, enabling the user of these devices to power down the PL for power management if required. The processors in the PS always boot first, allowing a software centric approach for PL configuration. PL configuration is managed by software running on the CPU, so it boots similar to an ASSP.

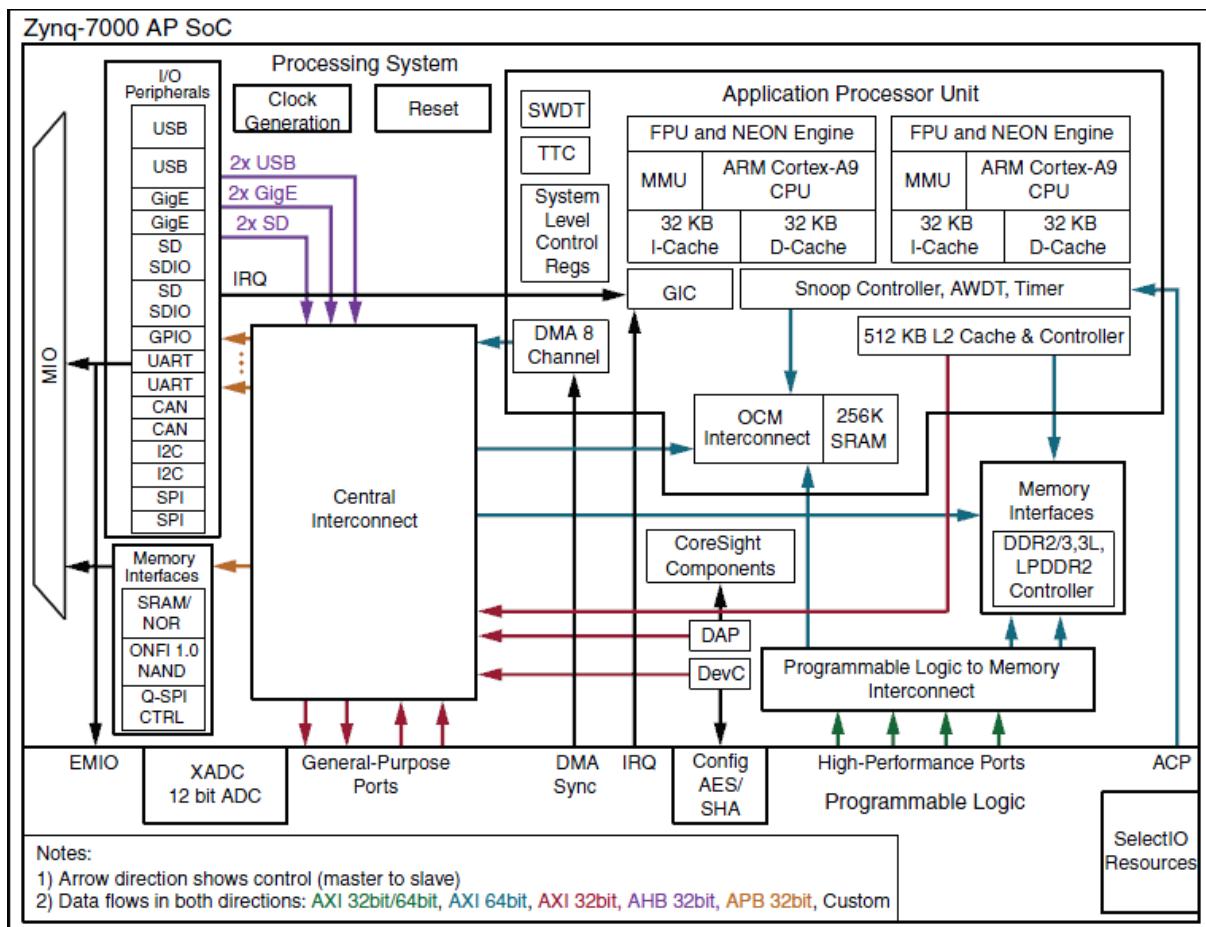


Fig 3.0.1 Zynq – 7000 block diagram

## 3.2 The Processing System (PS)

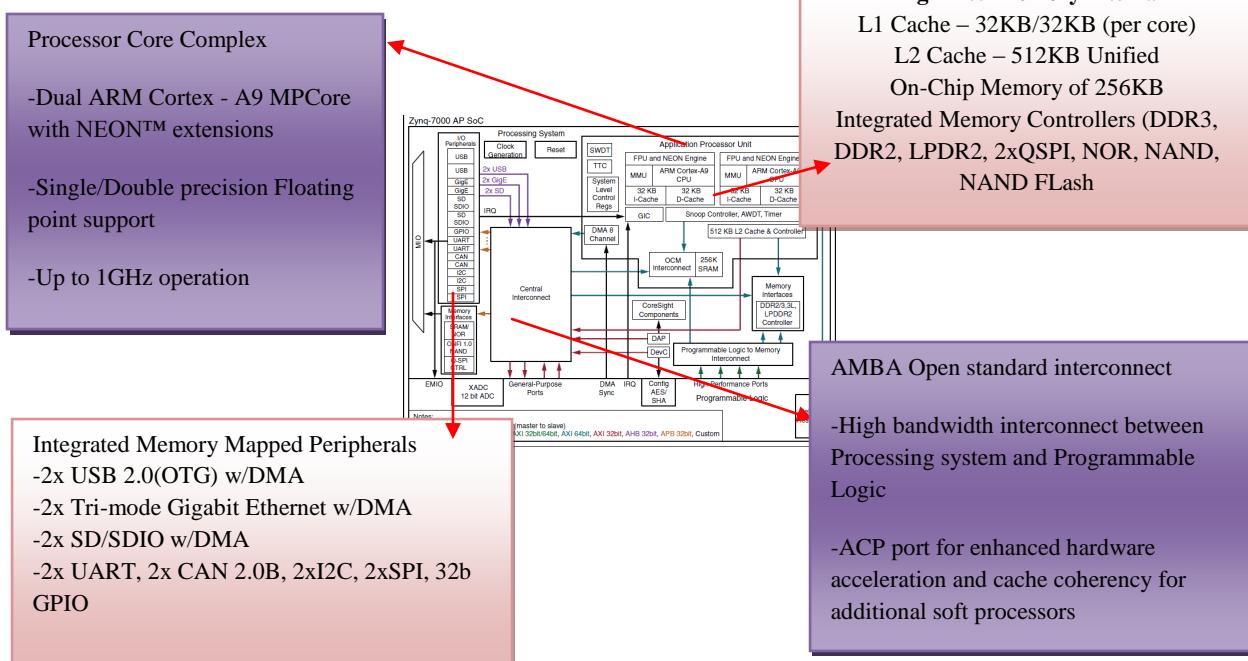


Fig 3.2 Processing system in Zynq AP SoC

Fig 3.2 illustrates the sub modules of the Processing System (PS). The ARM cortex – A9 MPCore is equipped with the 2.5 DMIPS/MHz per CPU, with the CPU frequency up to 1 GHz. The sub module called Snoop Control Unit (SCU) will maintain the L1 and L2 coherency. The PS has 256KB dual ported on-chip memory (OCM) which is accessible by the CPUs, PL and central interconnect. Accelerator Coherency Port (ACP) interface enables accesses from PL to CPU memory space.

Other than on-chip memory, the PS contains also the following Memory interfaces.

- DDR controller memory interface
  - Supports DDR3, DDR 2, LPDDR2.
  - 16-bit or 32-bit wide and 73 dedicated DDR controller pins from PS.
- Quad-SPI Controller
- NAND flash controller.
  - 8 - bit or 16 - bit I/O width with chip select signal.
  - 16-word reads and writes data FIFOs.
  - 8-word command FIFO.
- SRAM and NOR Controllers
  - 8-bit data width up to 25 addresses signals.
  - Two chip select signals.
  - 16-word reads and writes data FIFOs.
  - 8-word command FIFO.
  - 8 - Bit data width.

### 3.3 The Programmable Logic (PL)

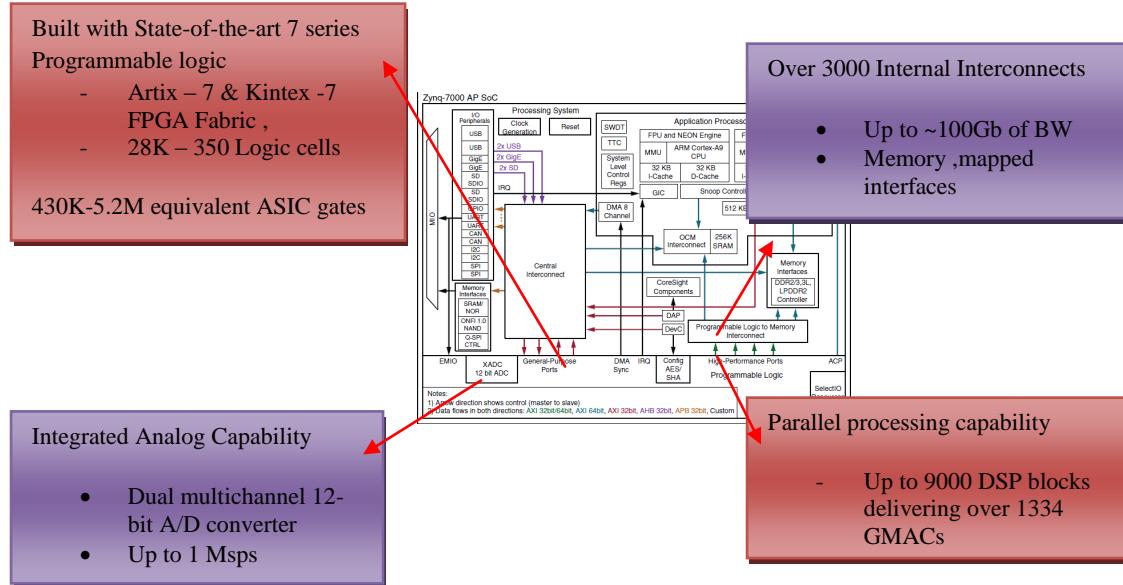


Fig 3.3 Programmable Logic in Zynq AP SoC

Fig 3.3 illustrates the features of the PL.

Key Features in PL include:

- Configurable Logic Blocks (CLB)
  - Eight LUTs per CLB for random logic implementation or distributed memory.
  - Memory LUTs are configurable as 64x1 or 32x32 bit RAM or shift register (SRL)
  - 16 flip-flops per CLB
  - 2x4-bit cascade able adders for arithmetic functions
- 36 Kb block RAM
  - True dual – port
  - Up to 36 bits wide
  - Configurable as dual 18 Kb block RAMs
- Two 12 – bit analog to digital converters (XADC)
  - On-chip voltage and temperature
  - Up to 17 external differential input channels
  -

#### 3.3.1 Block RAM

Some of the key features of the block RAM include:

- Dual-port 36 Kb block RAM with port widths of up to 72
- Programmable FIFO logic
- Built-in optional error correction circuitry

Every Zynq-7000 All Programmable SoC has between 60 and 755 dual –port RAM, each storing 36Kb. Each block RAM has two completely independent ports that share nothing but the stored data.

### **3.3.2 Programmable Data Width**

Each port can be configured as  $32K \times 1$ ,  $16K \times 2$ ,  $8K \times 4$ ,  $4K \times 9$  (or 8),  $2K \times 18$  (or 16),  $1K \times 36$  (or 32), or  $512 \times 72$  (or 64). The two ports can have different aspect ratios without any constraints. Each block RAM can be divided into two completely independent 18 Kb block RAMs that can each be configured to any aspect ratio from  $16K \times 1$  to  $512 \times 36$ . Everything described previously for the full 36 Kb Block RAM also applies to each of the smaller 18 Kb block RAMs.

Only in simple dual-port (SDP) mode can data widths of greater than 18 bits (18 Kb RAM) or 36 bits (36 Kb RAM) be accessed. In this mode, one port is dedicated to read operation, the other to write operation. In SDP mode, one side (read or write) can be variable, while the other is fixed to 32/36 or 64/72. Both sides of the dual-port 36 Kb RAM can be of variable width.

Two adjacent 36 Kb block RAMs can be configured as one cascaded  $64K \times 1$  dual-port RAM without any additional logic.

### **3.3.3 XADC (Analog-to-Digital converters)**

Highlights of the XADC architecture include:

- Dual 12-bit 1 MSPS analog-to-digital converters (ADCs)
- Up to 17 flexible and user-configurable analog inputs
- On-chip or external reference option
- On-chip temperature and power supply sensors
- Continuous JTAG access to ADC measurements.

All Zynq-7000 All Programmable SoCs integrate a new flexible analog interface called XADC. When combined with the programmable logic capability of the Zynq-7000 All Programmable SoCs, the XADC can address a broad range of data acquisition and monitoring requirements. This unique combination of analog and programmable logic is called Analog Mixed Signal

The XADC contains two 12-bit 1 MSPS ADCs with separate track and hold amplifiers, an on-chip analog multiplexer (up to 17 external analog input channels supported), and on-chip thermal and supply sensors. The two ADCs can be configured to simultaneously sample two external-input analog channels. The track and hold amplifiers support a range of analog input signal types, including unipolar, bipolar, and differential. The analog inputs can support signal bandwidths of at least 500 KHz at sample rates of 1MSPS. It is possible to support higher analog bandwidths using external analog multiplexer mode with the dedicated analog input

The XADC optionally uses an on-chip reference circuit ( $\pm 1\%$ ), thereby eliminating the need for any external active components for basic on-chip monitoring of temperature and power supply rails. To achieve the full 12-bit performance of the ADCs, an external 1.25V reference IC is recommended. If the XADC is not instantiated in a design, then by default it digitizes the output of all on-chip sensors.

The most recent measurement results (together with maximum and minimum readings) are stored in dedicated registers for access at any time via the JTAG interface. User-defined alarm thresholds can automatically indicate over-temperature events and unacceptable power supply variation. A user-specified limit (for example, 100°C) can be used to initiate an automatic power-down.

### 3.4 PS – PL Interface

Fig 3.4 illustrates the PS – PL interface system

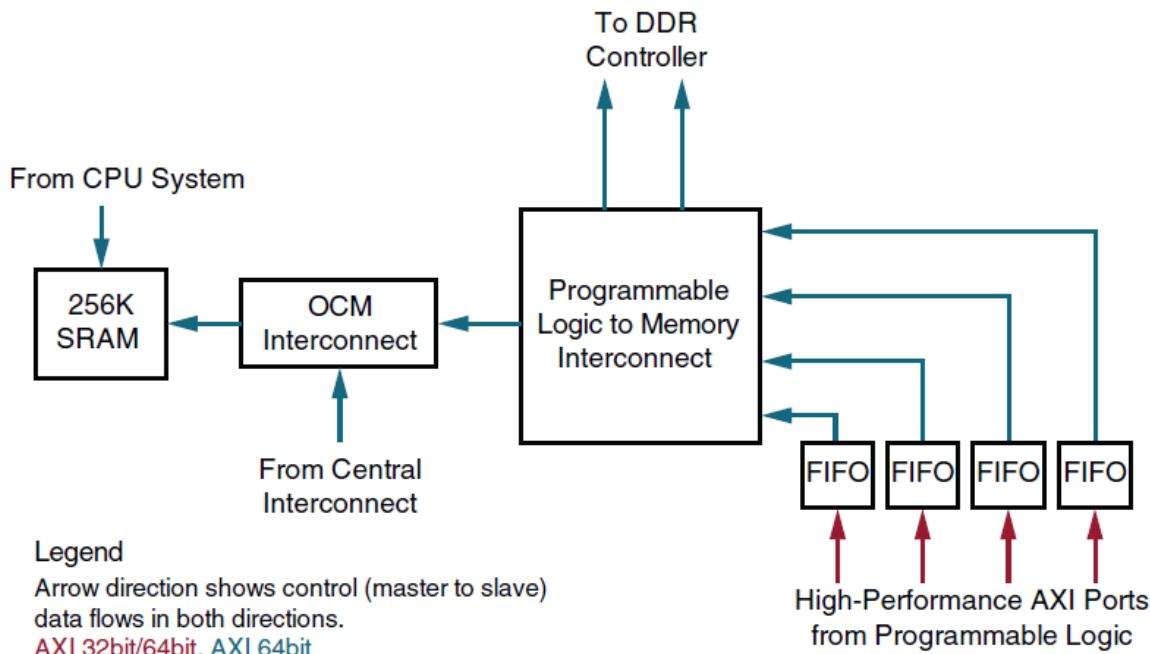


Fig 3.4 PL Interface to PS Memory subsystem

The PS-PI interface includes:

- AMBA AXI interfaces for primary data communication
  - Two 32 –bit AXI master interfaces
  - Two 32 –bit AXI slave interfaces
  - Four 64-bit/32-bit configurable, buffered AXI slave interfaces with direct access to DDR to DDR memory and OCM, referred to as high-performance AXI ports.
  - One 64-bit AXI slave interface (ACP port) for coherent access to CPU memory.
- DMA, interrupts, events signals
  - Processor event bus for signaling event information to the CPU
  - PL peripheral IP interrupts to the PS GIC
  - Four DMA channel signals for the PL
  - Asynchronous triggering signals
- Extendable multiplexed I/O (EMIO) allows unmapped PS peripherals to access PL I/O
- Clocks and resets
  - Four PS clock outputs to the PL with start/stop control
  - Four PS reset outputs to the PL

- Configuration and miscellaneous
  - Processor configuration access port (PCAP) to support full and partial PL configuration, and secured PS boot
  - image decryption and authentication
  - eFUSE and battery-backed RAM signals from the PL to the PS
  - XADC interface
  - JTAG interface

The two highest performance interfaces between the PS and the PL for data transfer are the high-performance AXI ports and ACP interfaces. The high performance AXI ports are used for high throughput data transfer between the PS and the PL. Coherency, if required, is managed under software control. When hardware coherent access to the CPU memory is required, the ACP port is to be used.

### **3.5 Software Tools Used**

#### **3.5.1 Vivado Design Suite**

The hardware design and integration of the IP is done with Vivado. The Vivado Design Suite delivers a SoC-strength, IP- centric and system – centric, next generation development environment that has been built from the ground up to address the productivity bottlenecks in the system-level integration and implementation. The Vivado Design suite is a Generation ahead in overall productivity, ease – of –use, and system level integration capabilities [6].

- Accelerating Implementation
  - 4x faster implementation
  - 20% Better Design Density
  - Up to 3-Speedgrade Performance advantage and 35% less Power
- Accelerating Integration
  - C-based IP Generation.
  - Model based DSP Design Integration
  - Block based IP Integration with Vivado IP integrator.
- Accelerating Verification
  - Integrated Design Environment for Design and simulation
  - Comprehensive hardware debug
  - Accelerate verification by >100X with C.C++ or SystemC.

### **3.5.2 Xilinx Software Development Kit (SDK)**

The Software Development Kit (SDK) is the Xilinx Integrated Development Environment for creating embedded applications on any of the Xilinx's microprocessors from Zynq – 7000 SoCs, to the industry-leading microblaze. SDK is the first application IDE to deliver true homogenous and heterogeneous multi-processor design and debug [7].

The SDK is built on industry open source IDE, eclipse. The IDE directly interfaces to both the Vivado and ISE embedded hardware design environment.



*Fig 3.5 Xilinx SDK.*

#### **Custom Design Aware**

Xilinx SDK understands the custom Xilinx-based embedded hardware design that has been defined in either the Vivado Design Suite or the Xilinx Platform Studio (XPS) with ISE. Based on this design, several key parameters are auto-configured, including memory maps, peripheral register settings, tools and library paths, compiler options, JTAG and flash memory settings, debugger connections, and Linux and bare-metal Board Support Packages (BSPs). This custom design-aware pre-configuration, combined with the auto-generation of critical system software, ensures that software development can begin with a minimal learning curve.

#### **Drivers and Libraries**

SDK includes user customizable drivers for all supported Xilinx hardware IPs, POSIX compliant kernel library and networking and file handling libraries. These libraries and drivers can scale for the custom-design based on feature needs, memory requirements and hardware capabilities.

### **3.6 SCILAB**

SCILAB is free and open source software for numerical computation providing a powerful computing environment for engineering and scientific applications.

In the Master Thesis, SCILAB is used to run the Hardware in the Loop (HiL) simulation. The SCILAB DLL is used for the communication.

### 3.7 FPGA Mezzanine Card (FMC)

Developed by a consortium of companies ranging from FPGA vendors to end users, the FPGA Mezzanine Card is an ANSI standard that provides a standard mezzanine card form factor, connectors, and modular interface to an FPGA located on a base board. Decoupling the I/O interfaces from the FPGA simplifies I/O interface module design while maximizing carrier card reuse.

Fig 3.6 shows the FMC Connecting pins in Zynq ZC-702.

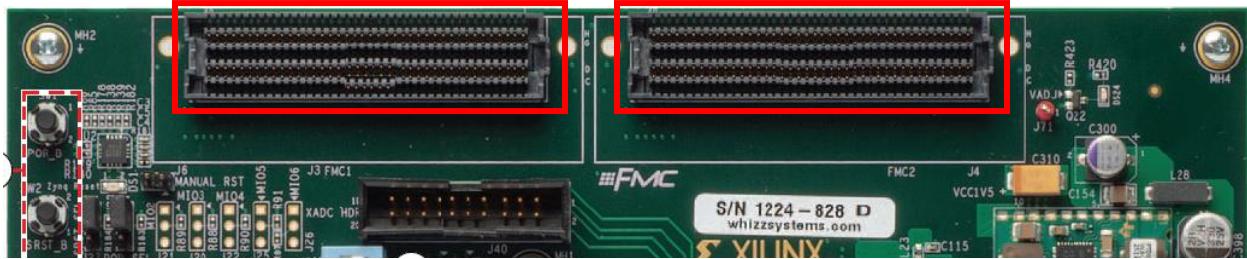


Fig 3.6 FMC Connectors.

#### Key Benefits of FMC

- **Data throughput:** Individual signaling speeds up to 10 Gb/s are supported, with a potential overall bandwidth of 40 Gb/s between mezzanine and carrier card
- **Latency:** Elimination of protocol overhead removes latency and ensures deterministic data delivery.
- **Design simplicity:** Expertise in protocol standards such as PCI, PCI Express®, or Serial RapidIO is not required.
- **System overhead:** Simplifying the system design reduces power consumption, IP core costs, engineering time, and material costs.
- **Design reuse:** Whether using a custom in-house board design or a commercial-off-the-shelf (COTS) mezzanine or carrier card, the FMC standard promotes the ability to retarget existing FPGA/carrier card designs to a new I/O. All that is required is swapping out the FMC module and slightly adjusting the FPGA design.

In the Motor Control kit, Fig 3.0, FMC -1 connector in Zynq is used to connect Avnet Motor Control FMC.

#### 3.7.1 Motor Control FMC

The Avnet Motor control is daughter board as a FMC connector to the Zynq – 7000 AP SoC and other compatible Xilinx FPGAs. This FMC is directly connected to the FMC pins.

Fig 3.7 shows the Avnet Motor control FMC



Fig 3.7 Avnet Motor Control FMC with Stepper and BLDC motor

Features:

- Low Pin Count (LPC) form factor.
- Enabling attachment to any Avnet or Xilinx FMC-enabled platform.
- Capable to drive PMSM , DC (BLDC) , Brushed DC (BDC) and Stepper motors
- Can spin two motors (12-24V) simultaneously under FPGA control.
- Integrated motor drivers from Texas instruments.
- Delta – sigma ADCs for high sensing is integrated using Texas instruments devices
- Xilinx XADC header enables low cost 7-series FPGA integration
- Hall Sensor / Encoder and GPIO ports
- Availability of user prototyping area
- Flexibility to power from FPGA/SoC base board or external source.

### Setup for Thesis

For the thesis, the Avnet FMC is connected to the FMC connector pin of the Zynq – 7000 AP SoC and it is powered externally. With the availability of two connectors to two different/same motors, the motors are connected with the switches to turn the control to stepper or BLDC motor. Fig 3.6 shows the connection.

For the Master Thesis work, only BLDC motor is used. There is an alternative to use stepper motor using toggle switches.

### 3.7.2 Brushless DC Motor (BLDC)

A BLDC motor is used to implement the thesis work and study the motor behavior. The specifications and motor values are described below and used for further calculations. Fig 3.8 shows the BLDC motor.

**Name:** BLWS232D – 36V – 4000

**Company:** Anaheim Automation



Fig 3.8 BLWS23 BLDC Motor

The BLDC motor used is cost effective solution for many velocity control applications.

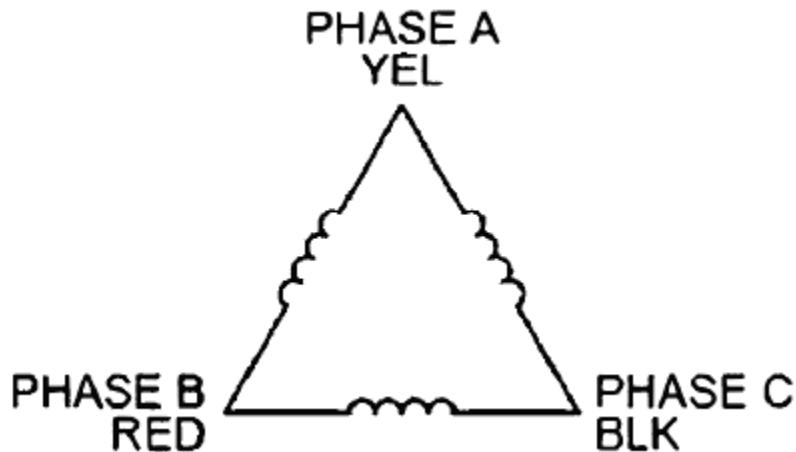


Fig 3.9 BLWS232D Winding

**Winding Type:** Delta, 4 Poles

**Hall Effect angle:** 120 Degree Elec. Angle

Wire Color	Description
Red	Hall Supply
Blue	Hall A
Green	Hall B
White	Hall C
Black	Hall Ground
Yellow	Phase A
Red	Phase B
Black	Phase C

Table 3.0.1 Wiring[9]

The Motor specification and values used for further calculations are as follows.

<b>Model</b>	<b>BLWS232D – 36V – 4000</b>
<b>Rated Voltage(V)</b>	36
<b>Rated Speed(RPM)</b>	4000
<b>Rated Power(W)</b>	4
<b>Rated Torque(oz-in)</b>	14.16
<b>No load current(A)</b>	0.40
<b>Torque Constant (oz-in/A)</b>	8.50
<b>Back EMF(V/kRPM)</b>	4.50
<b>Line to Line Resistance(ohms)</b>	2.40
<b>Line to Line Inductance (mH)</b>	4.39
<b>Rotor Inertia (oz-in-sec<sup>2</sup>)</b>	0.00106
<b>Weight(lbs)</b>	1.10
<b>Length (in)</b>	2.11

*Table 3.0.2 Motor Specification [9]*

## 3.8 Motor Control Drive

### 3.8.1 Zynq AP SoC based Motor Control

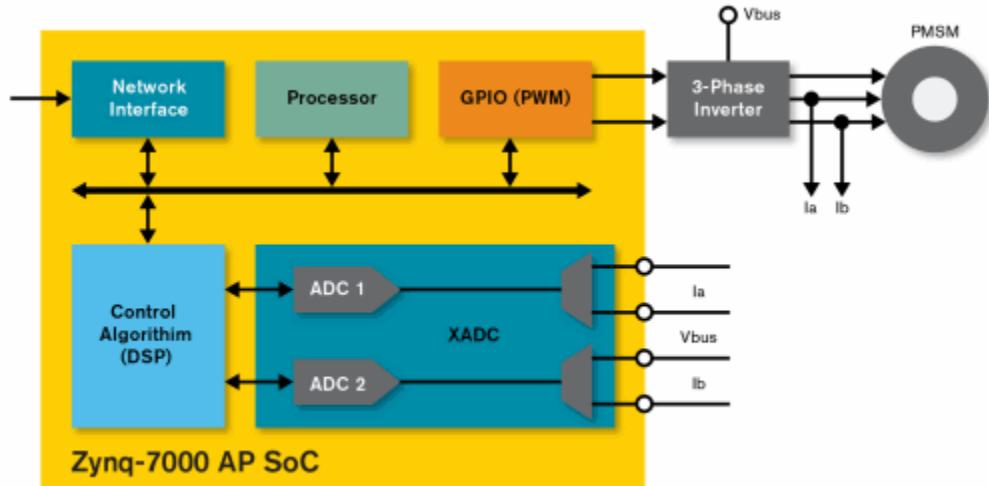
With the greater precision, performance and efficiency of Zynq AP SoC, Xilinx has a motor control drive which are capable of up to 30x faster control loop response, On-the-fly , uninterrupted algorithm upgrades and modulation switching and easy to use , optimized IP library for full vector control.

#### 3.8.1.1 Single Chip Embedded control

Featuring dual core ARM Cortex-A9 MPCore application –class processors, Zynq provides a single – chip motor control platform – integrating key functions and components such

as complex motion algorithms, modulation schemes , motor tuning, Industrial Ethernet, multi-motor control , legacy busses, ADC for sensor input, and system management[10].

Fig 3.10 shows the Motor control drive block diagram. As described, FMC daughter card is connected between the Zynq and Motor.



*Fig 3.10 Motor Control Illustration*

As seen from the Fig 3.10 two ADCs operating at 1 MSPS are integrated inside the Zynq AP SoC which makes a difference with the MCUs, ASSPs and DSP. The use of FPGAs in Motor Control enables the designer to construct a processor offloading engine that frees up the main processor to perform system tasks without interruption. This can prove critical for applications that require real-time control, and can also be more cost effective than scaling up to a higher performance system process.

The 3-phase current inputs are fed to the motor with the help of 3-Phase inverter and PMSM is driven. The stator currents  $I_a$  and  $I_b$  are recorded for the analysis.

### 3.8.2 QDESYS Drive

QDESYS is the Xilinx Alliance partner , for the motor control drive they offer 15 high – performance, easy to use , optimized motor control IP blocks in Spartan-6 and 7 series FPGAs and Zynq -7000 AP SoCs that allow the implementation of full vector control and easy industrial networking protocols to realize a fully-fledged Variable Frequency Drive (VFD).

PMSM, Brushless DC (BLDCs), and Stepper motors are fully supported. Using a simple application programming interface (API), a central control processor can issue high – level commands to configure and control the autonomous subsystem. The subsystem controls motor functions independent of the central microprocessor, and reports back status or issues interrupts as appropriate.

A major advantage of using the Xilinx solution is that multiple motors can be controlled at the same time. Also, the user can switch from one modulation scheme to another while the motor is running. For example, the motor can switch back and forth between Pulse Width Modulation (PWM) and Regenerative Pulse Frequency Modulation (RPFM) according to the requirement, without any interruption. Separating the motor controller operation through an autonomous solution encapsulates the solution, making the overall system easier to design, test, and maintain. This can also lead to lower cost and overall higher system-level performance.

The drive architecture and the APIs used for the Thesis are described in the sub-sequent chapters.

### 3.9 Architecture

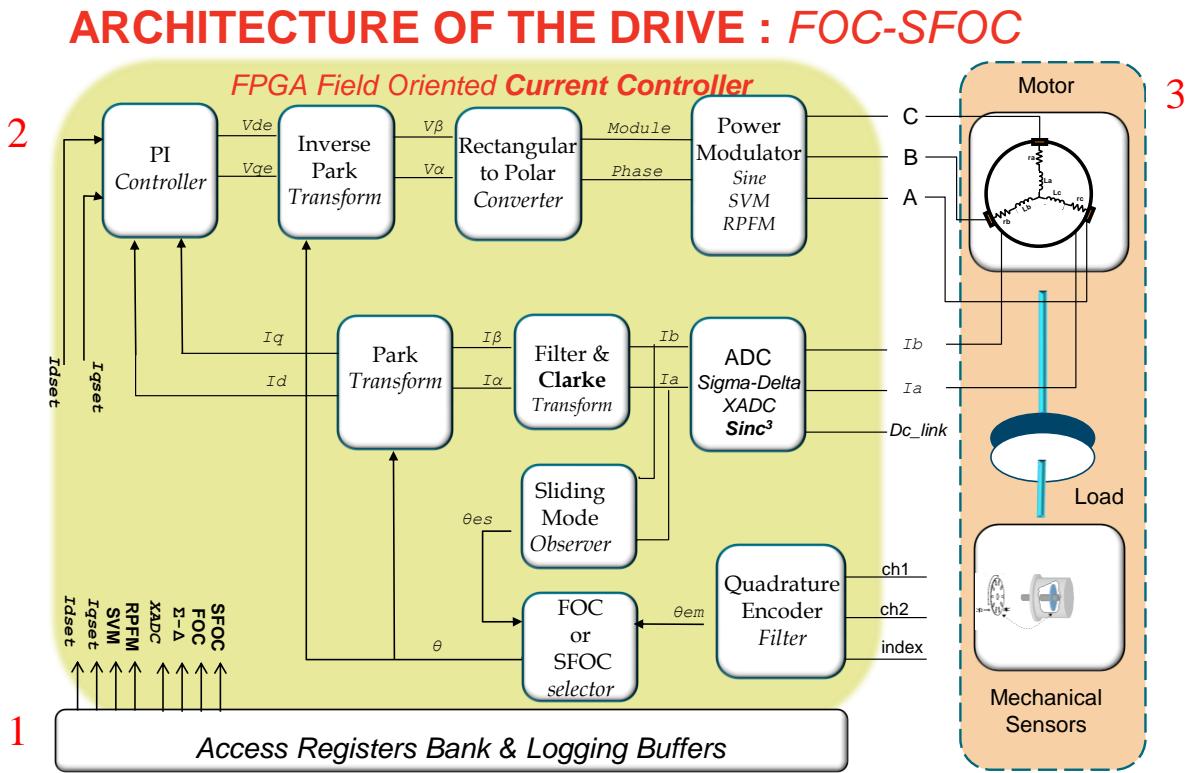


Fig 3.10 Architecture of the drive

The architecture of the drive mainly consists of the following:

- 1. Register bank and logging buffers:** The Register bank are the set of register values which are directly interfaced with the hardware. The values required for the FOC to drive and control the motor are programmed into the register and vice versa. The current set-up values Idset and Iqset are used to program the current set point values to the FOC values. The real time values from the  $\Sigma-\Delta$  ADCs and XADC are fetched from the respective register bank.

**2. FOC loop:** The FOC is integrated as an IP core which implements the main field oriented control module. Basically it is the container for the other specialized IP cores. The module uses the following functions:

1. Encoder signal interface
2. Clarke transformation
3. Park and inverse Park transformation
4. Proportional integral(PI) control for the current loop
5. Cartesian to polar transformation
6. BEMF(Back EMF) feed forward compensation
7. Pulse Width Modulation(PWM)
8. Regenerative Pulse Frequency Modulation (RPFM)
9. Very fast speed loop and position loop regulators.

Fig 3.12 shows the detailed block of the FOC IP core.

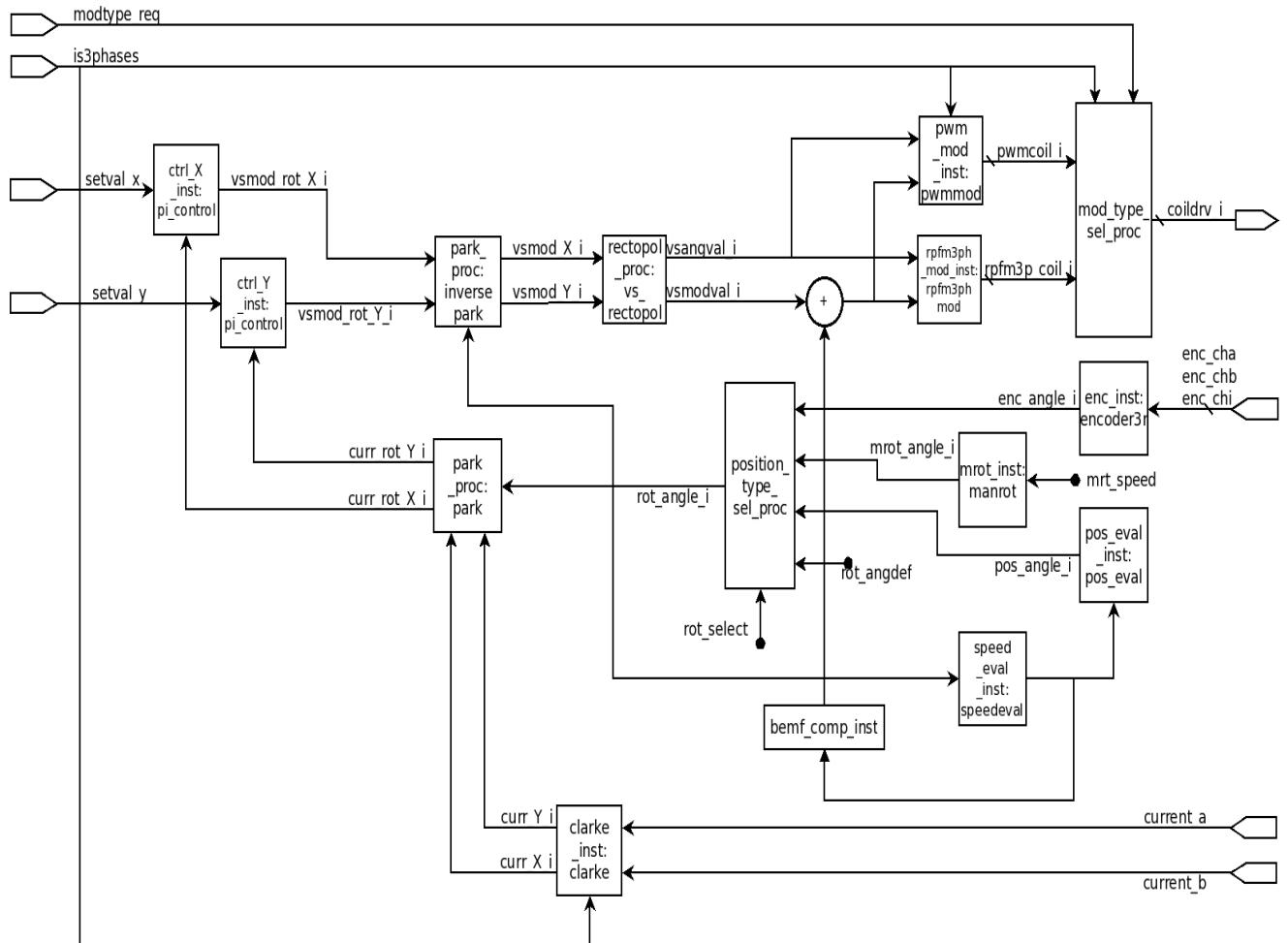


Fig 3.12 FOC IP Core

This FOC implements a **current control** accepting as reference inputs setval\_x (direct current component) and setval\_y (quadrature current component). Two stator currents I<sub>a</sub> and I<sub>b</sub> are supplied to this FOC and converted from a rotating 3-phase system into a rotating two-phase coordinate system described by the variables curr\_x and curr\_y via the Clark transform. The two currents curr\_x and curr\_y are then forwarded to a Park's transform that using the rotor's angle rot\_angle maps them into a fixed frame curr\_rot\_x and curr\_rot\_y. More details on Clarke and Park transform is discussed in *Chapter 6*.

In steady state conditions curr\_rot\_x and curr\_rot\_y are constant. The setval\_x reference controls rotor magnetizing flux, the setval\_y reference controls the output torque of the motor. The difference between curr\_rot\_y and setval\_y defines the torque error. The difference between curr\_rot\_x and setval\_x defines the rotor magnetizing flux error. The errors are fed into a PI (proportional integral) controller that transforms the current error into a voltage error vsmod\_rot\_y and vsmod\_rot\_y.

An inverse Park transform is applied to vsmod\_rot\_x and vsmod\_rot\_y mapping them from a fixed frame into a rotating frame vsmod\_x and vsmod\_y. Rectangular to polar conversion is then applied to vsmod\_x and vsmod\_y to obtain vsangval and vamodval, representing the module and angle of the stator voltage. At this level the BEMF feed forward compensation is applied by adding a value to vs. modulo. This function is driven by vs. speed evaluation module.

The module and angle are fed into the power modulation unit, PWM or RPFM that provide sinusoidal or space vector modulation using pulse width modulation (PWM) or pulse frequency / pulse density modulation (RPFM) to the motor. The FOC uses an incremental rotary encoder to capture the rotors angle position

**3. Mechanical Sensor:** The motor is equipped with the hall sensor. The input to the FOC from the motor are encoder data from the channel a, channel b and encoder index. **enc\_inst** is the sub module in the FOC. This module is an instance of encoder3r executes two basic functions:

- It decodes the signals from an incremental encoder (with “index” signal).
- It calculated the electrical angle of the rotor. To calculate electrical angle the module uses the counting of the encoder phases, a parameter that is the delta of angle for each phase and the timing signal **curr\_sync** to interpolate the electrical angle between two phase changes.

The supply voltage to the motor is given from the Power modulator module inside the FOC.

### 3.10 Qdesys IP

Qdesys has 15 fully documented IPs for the motor control. Fig 3.13 shows the IP blocks.

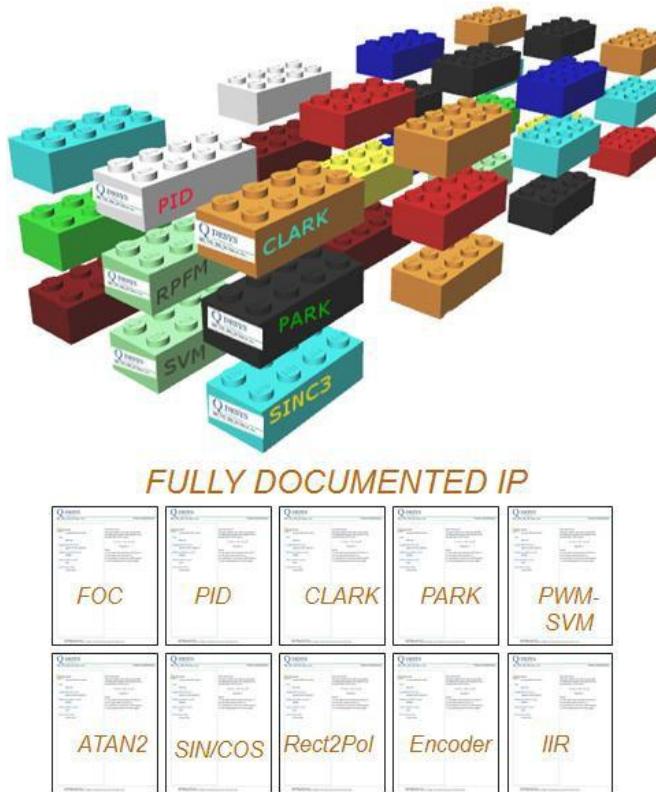


Fig 3.13 Qdesys Fully documented IP

The important features of the IP are the following.

- Basic Building Blocks
  - Fifteen control functions
  - Every function standalone
  - DSP48 centric
  - Dynamic operation with minimum footprint
- Much faster than microcontroller for example with the torque control loop.

#### 3.10.1 Environment of the drive

Fig 3.14 illustrates the complete drive put all together.

#### High level to Low level software:

- The carrier board hosts the Motor Control Agent software or MCA.
- The PC hosts the Motor Control Manager Software or MCM.

The MCM communicates with MCA through Ethernet interface and UDP/IP telegram using custom protocol for high speed and low overhead.

The drive system developed from Qdesys allows interfacing varying High level languages and platforms like Visual basic, NI LabVIEW, Hardware in the Loop (HiL) using MATLAB, SCILAB.

The DLL (Dynamic Link Library) for any high level interfaces is handled by the **Qdesys Manager** which is equipped with XML database, shared memory, and serial link to the carrier board.

The motor control agent driver in the embedded environment is responsible for various functions like logging, supervision, commands, and functions and diagnostic.

The supervision and diagnostic are provided for both the software and HDL part in the Zynq. The IP core developed in HDL is used to provide the controlled signal to the motor and interface with the Xilinx Block RAM (BRAM) to log the current setup upon new acquisition.

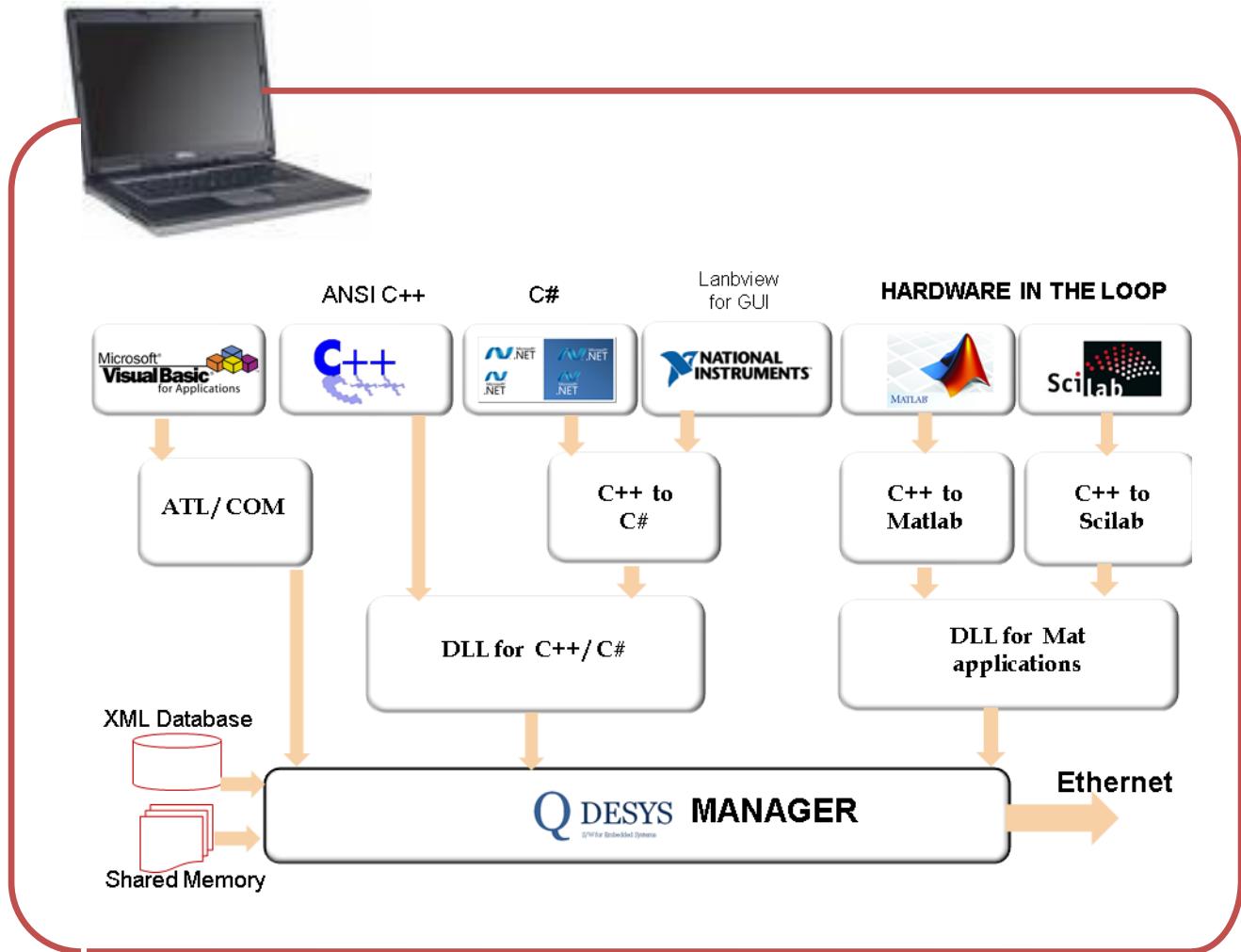
A set of three DLL libraries create API interface for different user application program

- mcm2lib.dll : “DLL for C/C++, Pascal”
- mc2com.dll: “ATL/COM”
- mcm2arlib.dll: “DLL for Math”

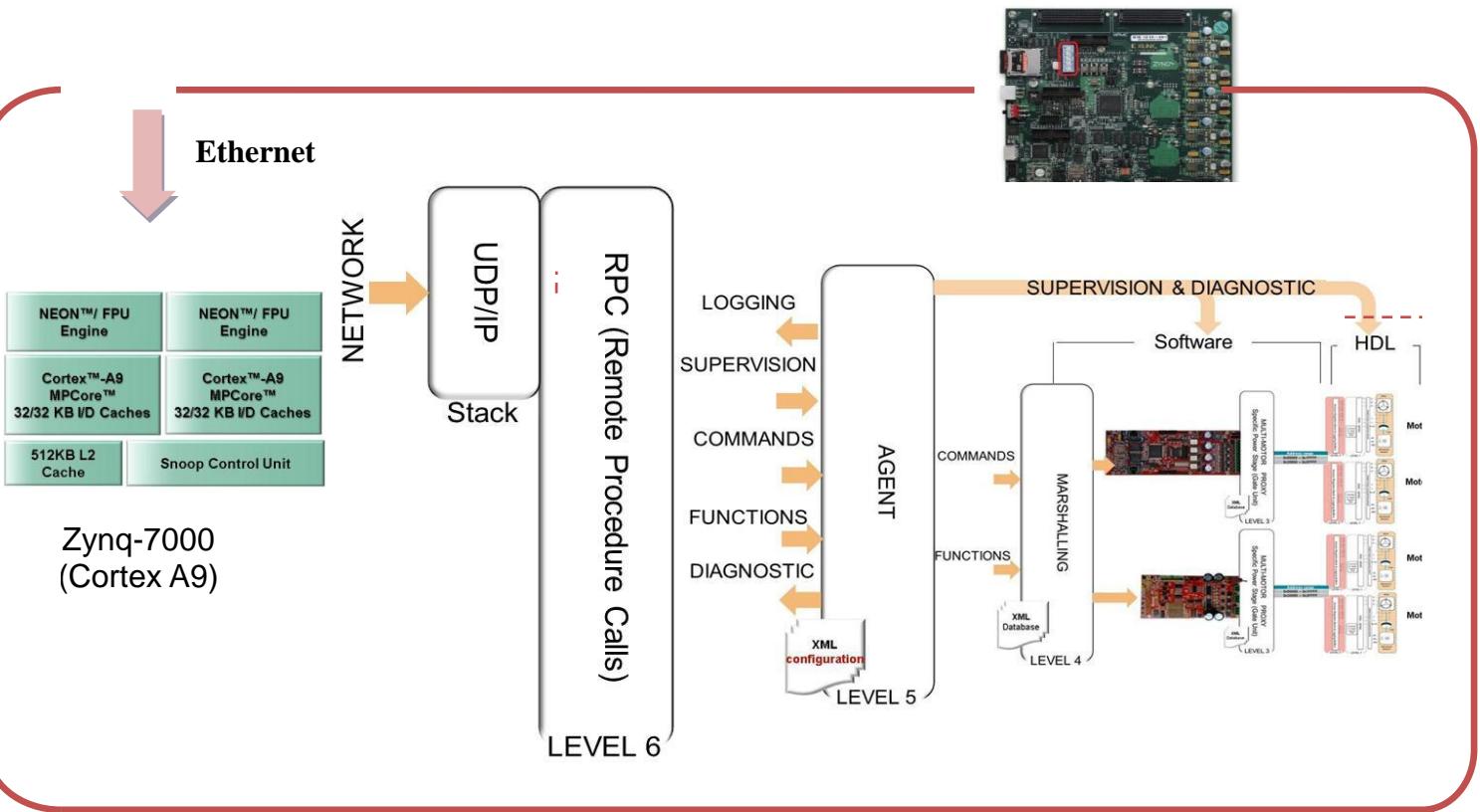
A set of three wrapper for different languages:

- mcm2cslib.dll: “C++ to C# wrapper”
- mcm2arlib.m: “C++ to MATLAB”
- mcm2arlib.sce: “C++ to Scilab”

**Note: A detailed description of the API can be issued by contacting Xilinx office.**



a.



b.

Fig 3.14 a&b All Together Motor Control Environment

# Chapter 4

## System Identification

*In this chapter, the fundamentals of system identification are explained. The Chapter follows with the cross-correlation technique used in the system identification and fundamentals of PRBS. The SCILAB simulation and results obtained are discussed.*

### 4.1 System Identification Fundamentals

A Linear Time Invariant (LTI) system can be best described by its transfer function or by its impulse responses [11]. In system identification [12], a model of a system is created by means of data from the system input and output signals. The identification methods that yield linear models can be divided (1) Parametric estimation methods, and (2) Non-parametric methods.

- (1) **Parametric Method:** In parametric, a system model is assumed that includes some parameters with the unknown values which are often called as model structure. The parameters in a model structure can be estimated off-line or on-line. If the parameters are estimated off-line, a batch of measured data is processed to yield a model. If the parameters are going to be estimated on-line, a recursive parameter estimation method is used.
- (2) **Non – Parametric:** The determination of the parameters is done by direct techniques without selecting confined set of possible models. Non – parametric methods include:
  - a. Correlation analysis
  - b. Transient response analysis
  - c. Frequency analysis, Fourier or spectrum analysis

This Master thesis deals with the Non-parametric methods of system identification using cross – correlation technique.

### 4.2 Cross – Correlation

Mathematically, correlation is the close cousin of the convolution [13].Correlation uses two signals and produces the third signal. This third signal is called the cross correlation of the two input signals. The correlation is invested by comparing them both directly superposed, and with one of them shifted left or right.

The correlation between two continuous functions  $g(t)$  and  $h(t)$ , which is denoted by  $\text{Corr}(g, h)$  is the function of *lag t*.

$$\text{Corr}(g, h) \equiv \int_{-\infty}^{+\infty} g(\Gamma + t) h(\Gamma) d\Gamma$$

The correlation will be large at some value of  $t$  if the first function ( $g$ ) is a close copy of the second ( $h$ ) but lags it in time by  $t$ , i.e., if the first function is shifted to the right of the second. Likewise, the correlation will be large for some negative value of  $t$  if the first function leads the second.

**Example [Dr. Giulio Corradi]:**

If we have two vectors  $u$  and  $h$ , where  $u$  can consider as input to the system and  $y$  as output and  $h$  as impulse function.

$$u = (x[0], x[1], \dots, x[N-1]), \text{ and } h = (h[0], h[1], \dots, h[N-1])$$

The (linear or aperiodic) convolution of two vectors is a vector,

$$Z = u * h = (z[0], z[1], \dots, z[2N-1])$$

Whereas, their correlation is a vector

$$Y = u \otimes h = (y[-(N-1), y(N-2)], \dots, y[0], y[1], \dots, y[N-2], y[N-1]).$$

Multiplying the Fourier Transform of the input and Fourier Transform of the complex conjugate of the output gives the Fourier Transform of their correlation.

$$\text{Corr}(u, h) \iff U(f) H^*(f) \quad \text{“Correlation Theorem”} \quad \text{Eq (4.2.1)}$$

### Auto – Correlation

The correlation of the function with itself is called auto correlation.

**Example:** In the case of 4.2.1, the equation becomes

$$\text{Corr}(u, h) \iff |U(f)|^2 \quad \text{Eq (4.2.2)}$$

### **4.2.1 Cross – Correlation for system identification**

Here we review and study the application of correlation method to digitally sampled system. In steady state, for small signal disturbances, our system of concentration can be regarded as a Linear Time In-variant (LTI) discrete time system [14], where the sampled system can be described as,

$$y(n) = \sum_{k=1}^{\infty} h(k)u(n-k) + v(n) \quad \text{Eq (4.2.3)}$$

Where,

$y(n)$  is the sampled output signal

$u(k)$  is the input digital signal

$h(k)$  is the discrete – time system impulse response , and

$v(n)$  represents disturbances , including quantization noise, measurement error, etc.

The Cross – correlation ( $C_{uy}$ ) of the input control signal  $u(k)$  and output signal  $y(n)$  is [4]:

$$\begin{aligned} C_{uy}(m) &= \sum_{n=1}^{\infty} u(n)y(n+m) \\ &= \sum_{n=1}^{\infty} h(n)C_{uu}(m-n) + C_{uv}(m) \end{aligned} \quad \text{Eq (4.2.4)}$$

Where,  $C_{uu}(m)$  is the auto correlation of the input signal.

Now, if the input control signal  $u(k)$  is selected to be white noise, then we benefit from the following characteristics:

$$\left\{ \begin{array}{l} C_{uu}(m) = \delta(m) \\ C_{uv}(m) = 0 \end{array} \right. \quad \text{Eq (4.2.5)}$$

In other words, the auto correlation of the input  **$C_{uu}$  is an ideal delta function** and the cross – correlation of the white noise input with the disturbance  $v(k)$  is ideally zero.

Under the conditions of 4.2.5, the cross correlation of 4.2.4 can be reduced to

$$C_{uy}(m) = h(m) \quad \text{Eq (4.2.6)}$$

Thus, the correlation of the input and output sampled signals give the discrete time system impulse response.

The control to output transfer function of the target Xilinx AP SoC based motor control solution in frequency domain can then be derived by applying Discrete Fourier Transform (DFT) to  $C_{uy}(m)$ :

$$C_{uy}(m) \xrightarrow{\text{DFT}} H(j\omega). \quad \text{Eq (4.2.7)}$$

This theoretical result requires the ability to generate white noise as an input perturbation to the system.

- A simple compromise in a digitally controlled system is to approximate white noise through use of Pseudo Random Binary Sequence (PRBS) perturbation.

## 4.3 PRBS Fundamentals

**PRBS** or Pseudo Random Binary Sequence is essentially a random sequence of binary numbers. It is periodic, deterministic signal with the white noise like properties [11].

It is deterministic because the sequence repeats itself after  $2^n - 1$  samples for n-bit of PRBS.

### 4.3.1 PRBS Generation

A PRBS bit stream can be generated by using a [4] Linear Feedback Shift Register (LFSR). Fig 4.1 shows the simple 4-bit LFSR.

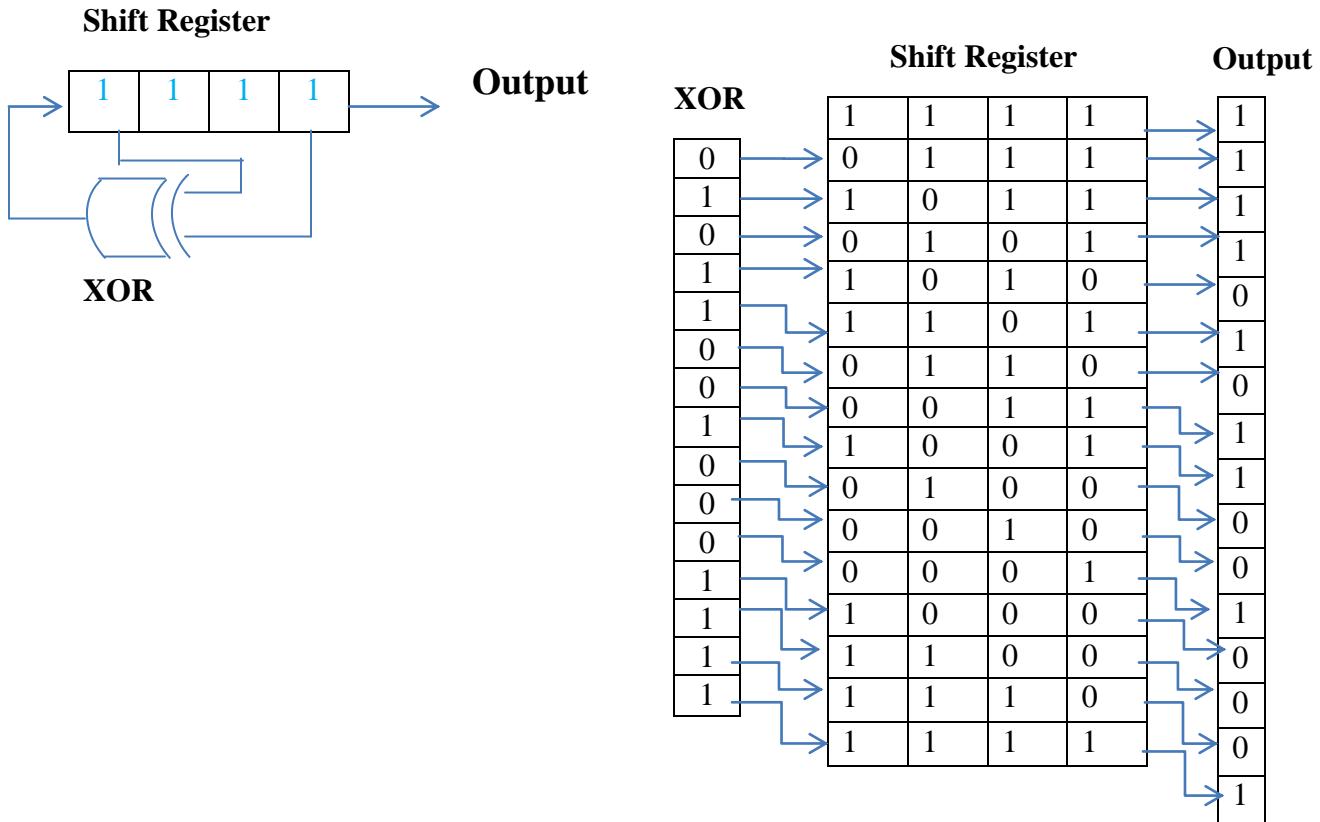
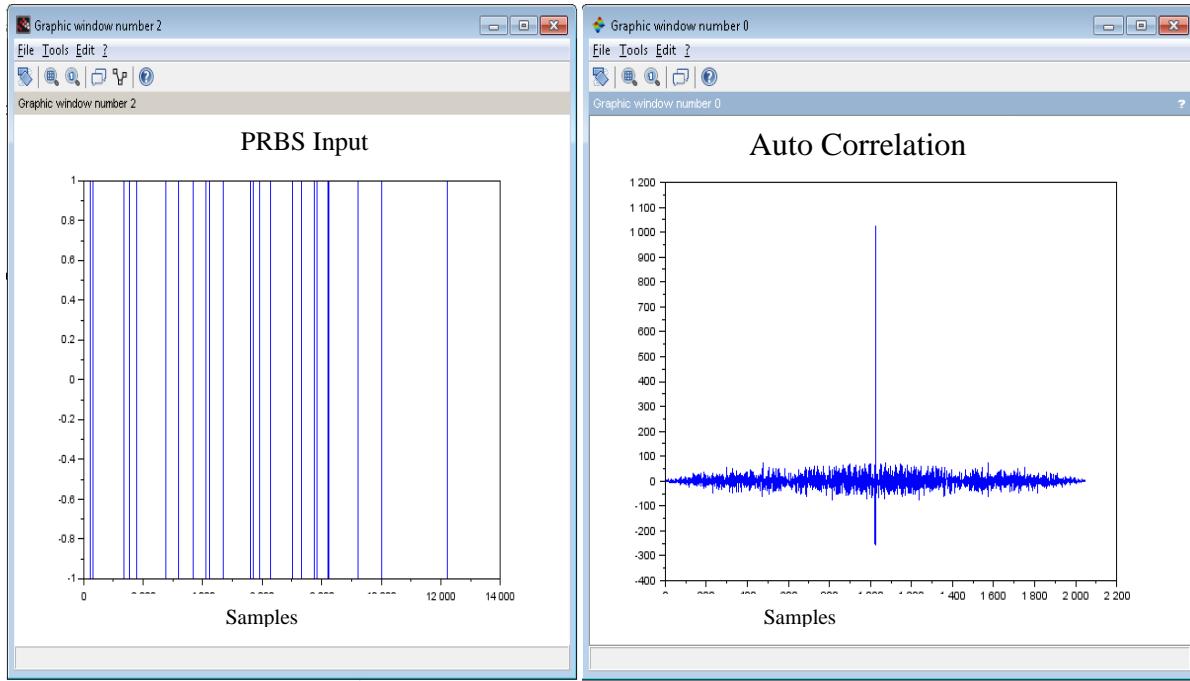


Fig 4.1 4-bit LFSR

The data length for one period of N-bit maximum length PRBS is given by  $M = 2^n - 1$  and the signal itself has two possible values:  $\pm e$ , where  $e$  is the magnitude of the PRBS [12].

Fig 4.2 shows (a) the n – bit sample PRBS (b) the auto correlation of the PRBS which shows the auto correlation of the PRBS is delta function as described by the **equation (4.2.5)**.



*a. N-bit PRBS input b. Auto correlation of PRBS*

*Fig 4.12*

We can see that the auto correlation of the PRBS which is compromised with the white noise is delta function, but with the noise components around it.

Thus (4.2.6) gives the system impulse response.

# Chapter 5

## Embedded Software Implementation

In this Chapter, how the synchronization between the current input and current acquisition is described along with the PRBS generation and injection methodologies.

### 5.1 Introduction

As explained in the *Chapter 4*, the injection of white noise into the system and cross correlating with the output would give the impulse response of the system.

The Qdesys motor control drive has provided with set of APIs<sup>1</sup> to fetch the current data from the motor. Particularly, the Ia, Ib and angle from the encoder is used for the analysis. In this chapter, the challenges faced in dealing with the cross correlation is addressed and a modified architecture for the system identification is implemented.

#### 5.1.1 Architecture of Cross Correlation

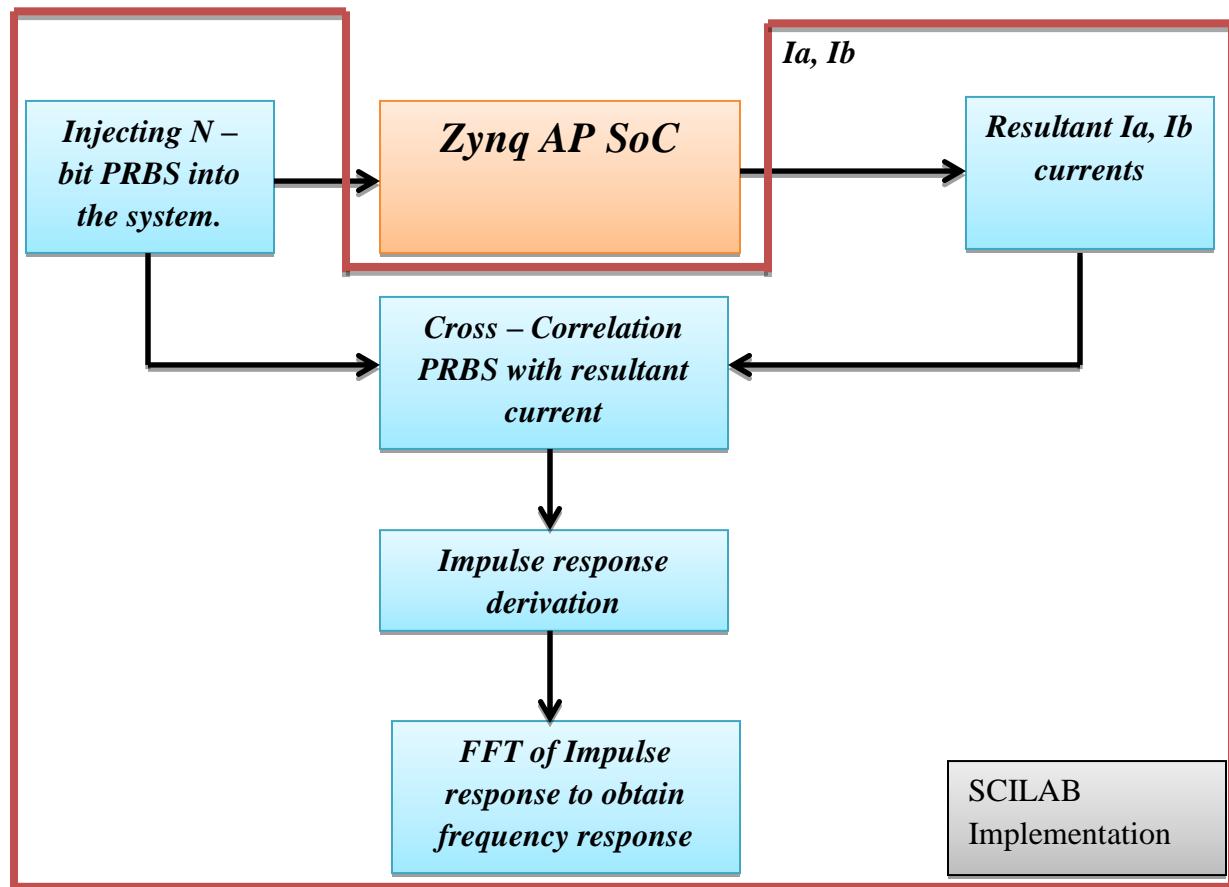


Fig 5.1 Cross Correlation architecture

<sup>1</sup> [angle,curra,currb] = qmhm\_acqi\_data\_s(motor), motor number

From Fig 5.1, it is clear that the PRBS is generated in the SCILAB environment taking Zynq AP SoC Hardware in the Loop (HiL). A single period maximum length of **13 – bit**, data length of **8192** samples is injected as perturbation into the motor control system.

### 5.1.2 Challenges Faced

The SCILAB API,  $\text{xcorr}^2$  is used to measure the cross correlation. Upon executing the function and correlated graph is plotted. Fig 5.2 shows the cross correlation graph.

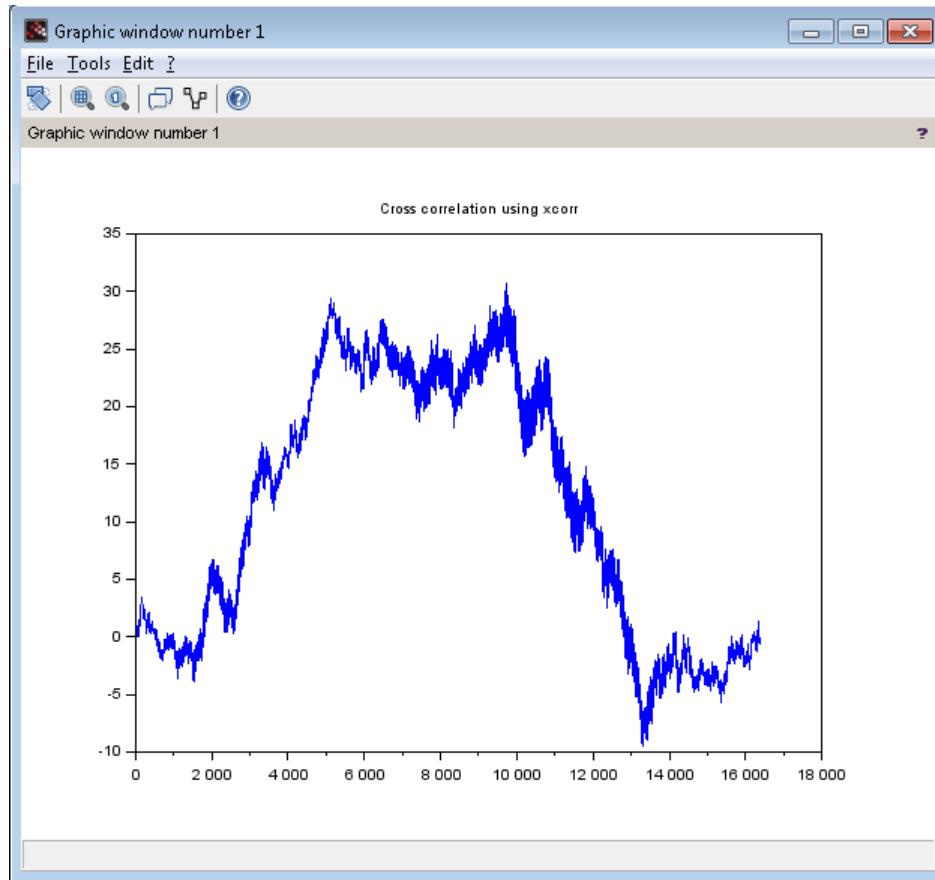


Fig 5.2 Cross – Correlation graph

Since the correlation gives the similarity between the two functions with one being the time lag of other [Eq (4.2.3)]. The correlation between the two signals obtained was not synchronizing. In a sense, when the PRBS bit was injected continuously it was not correlating to the required new current acquisition set up.

This was true due to the fact that, the SCILAB is interfaced with the Ethernet connection. The PRBS sequence injection would consume time to synchronize with the current acquisition. In other words, the current acquisition was much faster than the input to the Zynq SoC. Hence there was need to fetch the output from the motor (current acquisition) for the particular sequence of PRBS.

---

<sup>2</sup>  $\text{xcorr}(x, y)$  cross correlation between  $x$  and  $y$ .

## 5.2 Improved Methodology to inject PRBS

In order to overcome the challenges described in *section 5.1.2*, embedded coding using the Zynq AP SoC Processing System (PS), described in detail in *Chapter 3, 3.1.2* would give the possible best result.

### 5.2.1 Improved architecture of the Cross – Correlation

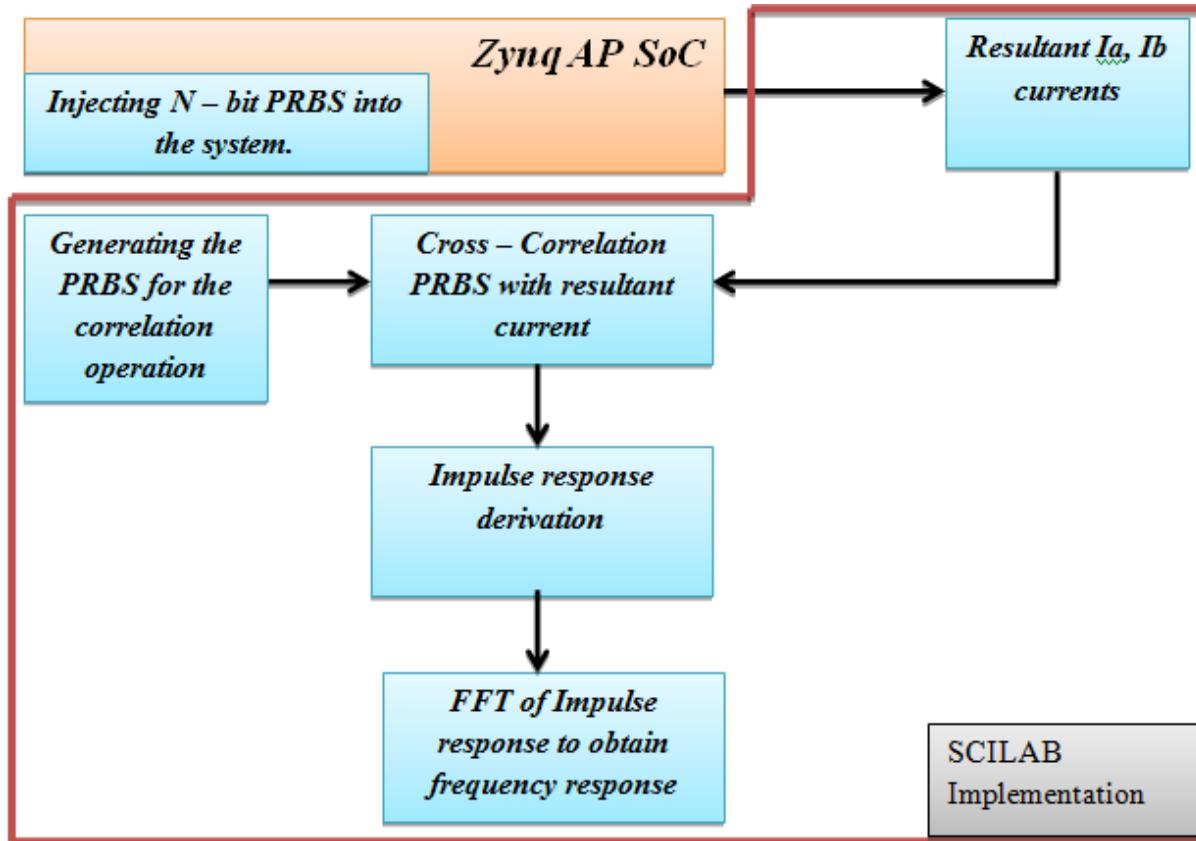


Fig 5.3 Cross – Correlation modified architecture

Fig 5.3 shows the modified cross correlation architecture where in the PRBS is generated and injected in the embedded environment. Since the correlation operation is not feasible in embedded environment a same N number of PRBS sequence is generated in the SCILAB<sup>1</sup>. The current outputs from the Zynq AP SoC are the result of the PRBS sequence generated in embedded environment.

<sup>1</sup>*Note:* Since the correlation is methodology to eliminate noise content and Chapter 4, section 4.2 explains the auto correlation is delta function. The PRBS generated would not affect the overall correlation process.

## 5.3 Current Acquisition & Synchronization

### 5.3.1 Current Acquisition

As described in *Chapter 3*, the FOC IP includes various sub modules. The current acquisition described by *Acqv\_data* is responsible for acquiring data.

Referring to Chapter 3, section 3.1.3 about the BRAM (Block RAM) used in Xilinx Zynq AP SoC a set of **0...16 Xilinx BRAM 1024 x 18 primitive are used to implement** a DPR for current acquisition system for diagnostic purpose.

A single sample consists of:

- a. 12 bits stator angle
- b. 12 bits current A value
- c. 12 bits current B value

The size of whole record is 36 bits so a 2x18 bits memory location is required for a single sample.

The 36 bits record is stored into two 18 bits memory location in following way:

- word[0] = angle[11..6] \*  $2^{12}$  + current\_a
- word[1] = angle[5..0] \*  $2^{12}$  + current\_b

A single **1024 x 18 bit** BRAM can record up to **512** acquisition samples.

The maximum implementation using a 16 set of 1024x18 BRAM is,

$$16 \times 1024 \times 18 = 8192 \text{ samples.}$$

**IP Activation Time:** With the IP activation time =  $3.2 \mu\text{s}$  a single record sample is available at 312.5 KHz so the maximum acquisition time is  $3.2 \mu\text{s} * 8192 = 26.2144 \text{ mS}$  equivalent to about 38.15 Hz (electric speed).

### 5.3.2 Synchronization

With the knowledge of IP activation time which is actually the FOC IP time which it takes to acquire the current samples from the motor. The new sequence of PRBS injection would be such a way that it waits for the FOC IP loop to finish and then apply the new sequence of PRBS. This approach allows, the current samples which are acquired are the result of the input sequence of PRBS.

## 5.4 PRBS Generation

In the *Chapter 4, section 4.3.1* the PRBS generation using 4-bit LFSR is illustrated. Since the maximum acquisition samples from the motor is 8192 samples. A 13- bit PRBS sample would give  $2^{13} = 8192$  bits,

Hence,  $N = 13$  and  $M = 8192$  samples (where  $N$  is number of PRBS bit in sequence and  $M$  total number of bits).

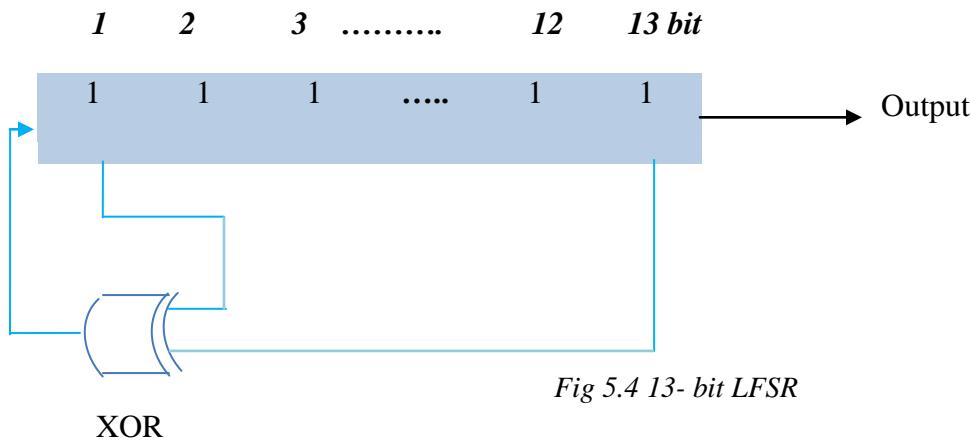


Fig 5.4 13- bit LFSR

XOR

Fig 5.4 illustrates 13 – bit LFSR (Linear Feedback Shift Register). With this kind of arrangement the 13 –bit PRBS sequence is injected into the system.

### 5.4.1 Algorithm

The algorithm used to generate the 13 – bit PRBS (the bit value could be modified depending upon the application) is as following,

#### PRBS Generation:

- Referring to Fig 5.4, the first bit and 13<sup>th</sup> bit is XOR'd.
- The resultant value is shifted into the LFSR
- The last bit is taken out as the PRBS bit
- Following this operation a 13-bit PRBS is generated.

#### PRBS Injection:

- The current high value is read from the register.
- With the indexing factor of 0.5, the 50% of the input current is taken
- The amplitude of the PRBS is set 0.01
- Ex: Supposing 3A is the input current, and then PRBS will vary between 1.2 and 1.8.

## 5.5 Flow Chart

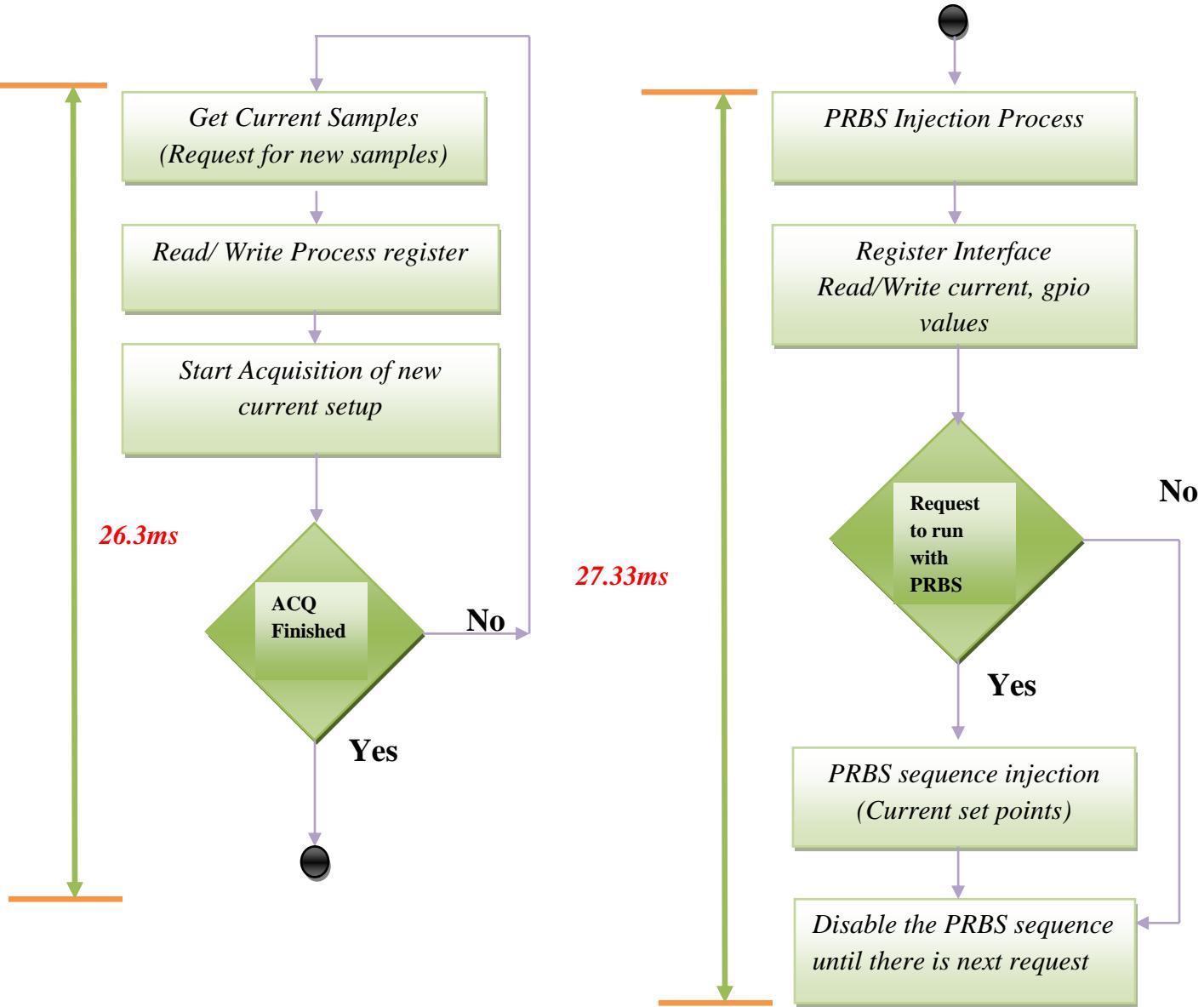


Fig 5.5 PRBS Injection and Synchronization Mechanism

Fig 5.5 shows the mechanism used to have a PRBS perturbation and acquisition of current samples which are result of that particular sequence of PRBS.

The new current samples (8192) values from the motor are taken from the motor after 26.33 ms (section 5.3). And, the PRBS injection is 27.33ms, i.e. every time waits for the 8192 values to be acquired, this should show that when a 13 – bit PRBS is injected into the system as a result of which the motor real time data is acquired from the motor.

### 5.5.1 Synchronization Timing

In this section the details about how the time is calculated is described. Step followed to calculate the injection timing is:

1. Initiation(init) time of the current acquisition is recorded
2. The time at which the PRBS injection is recorded and subtracted with init time.
3. The resulting time is recorded for say 10 runs and the mean of the time is taken which is **27.33ms**.

### 5.6 SCILAB Implementation

Referring to Fig 5.3, the modified architecture of the cross correlation the currents from the Xilinx AP SoC is used for the analysis and calculation.

In the SCILAB environment a set of APIs which interface to the QDESYS motor control agent (*Chapter 3, section 3.4*) is used as a library. The library is loaded which will allow accessing the register values like current set points, angle, etc.

Acquisition function in the SCILAB will call such APIs and fetch the rotating angle, current a , current b.

#### 5.6.1 Steps for Cross correlation

- **Step I:** Considering the 13-bit PRBS, 8192 values is generated using the function `prbs_a1` with the frequency of 1/4<sup>th</sup> of the total PRBS value, i.e. for the 8192 samples 2048 times the sign will change from +1 to -1 and vice versa.
- **Step II:** Using the `gpio` (General Purpose I/O) set method, the PRBS injection is enabled.
- **Step III:** Using a SCILAB function to acquire the current data, rotating angle, current a and current b is recorded.
- **Step IV:** Applying the Clarke's transformation the 3 – Ph current is converted to (I $\alpha$  and I $\beta$ ) 2 – Ph (Chapter 6).
- **Step V:** The resultant of I $\alpha$  and I $\beta$  is calculated.
- **Step VI:** Cross – correlation of PRBS input and resultant (Is) of I $\alpha$  and I $\beta$  is performed
- **Step VII:** The maximum of the cross-correlation point is determined and standard deviation from the various runs is calculated.
- **Step VIII:** Using FFT function Magnitude and Phase is determined.

## 5.7 Results

### 5.7.1 Cross – Correlation

The result of cross correlation is verified by running the correlation process several times and calculating the peak mean standard deviation. Fig 5.6, (a) (b) (c) shows the cross correlation.

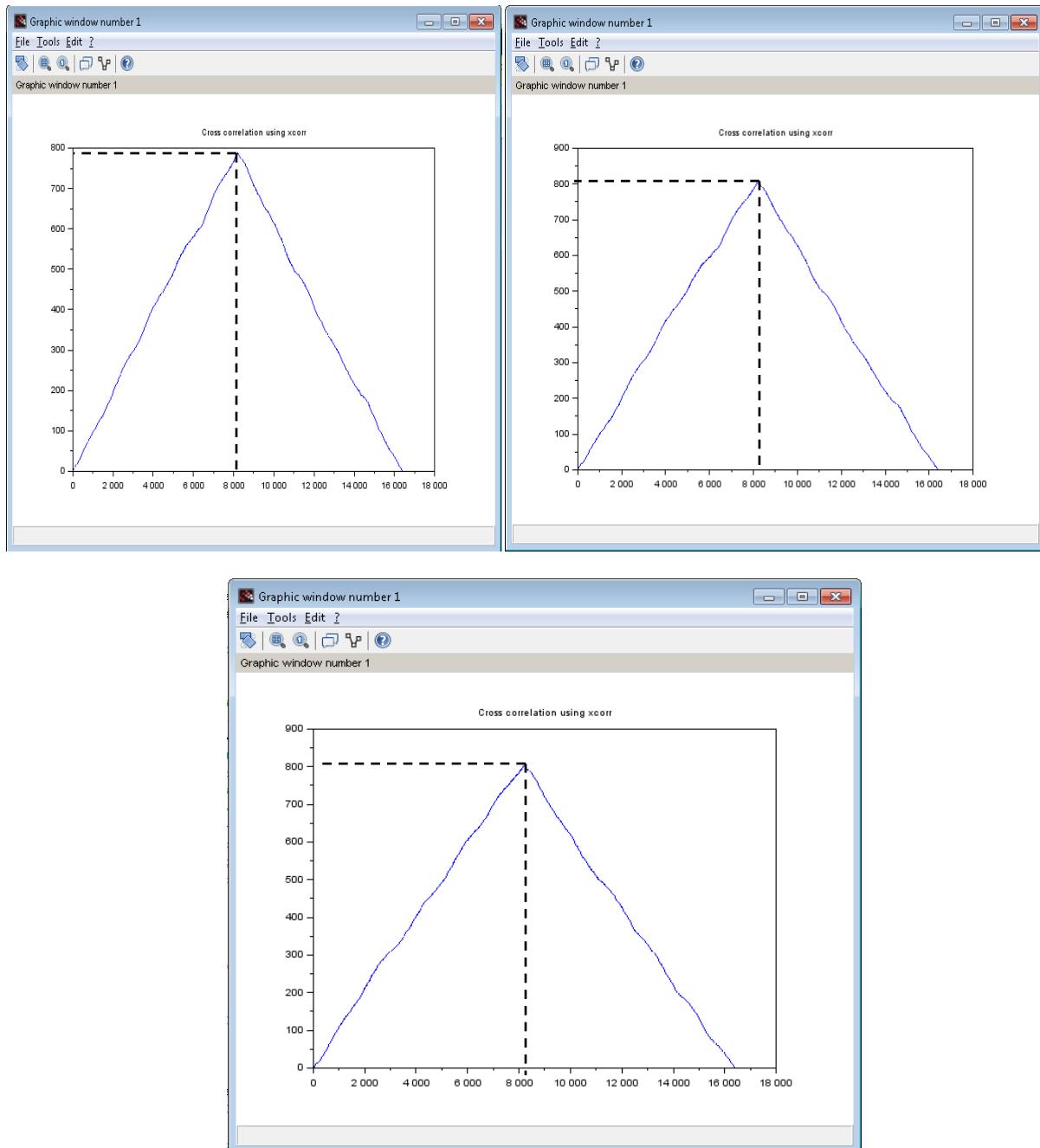


Fig 5.6 (a) 1<sup>st</sup> run peak at ~780 (b) 2<sup>nd</sup>, peak at 800 (c) 3<sup>rd</sup> at 810

Fig 5.6 shows the cross correlation. Comparing Fig 5.2 with the Fig 5.6, it is clear that the modified cross correlation used to inject PRBS gave the prominent results. The Steps followed to calculate the standard deviation is as follows:

1. Using  $\max^*$  function in SCILAB the maximum of the correlation data is taken.
2. The mean value of the maximum per acquisition display is taken using  $\text{mean}^*$  function.
3. Finally the standard deviation ( $\text{stdev}^*$ ) of the mean value for say N runs is taken.

Fig 5.7 is the screen shot from the SCILAB simulation. As we can see that the Standard deviation obtained is very less. This should show that the PRBS injection and acquisition of the current set up is synchronized.

```
The Average of Max of Correlation peaks of 3 runs is : 8.034190
The std deviation is : 0.155554
110 display in 30 seconds: rate is 3.666667
That's all folks!
```

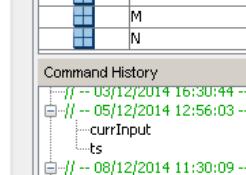


Fig 5.7 SCILAB standard deviation calculation

Alternative approach to know the PRBS injection and current acquisition is to study the space vector trajectory. Fig 5.8 shows the vector trajectory plotted in  $\alpha\beta$  reference plane (*details in Chapter 5*). It can be seen that the current makes the trajectory without the deviation.

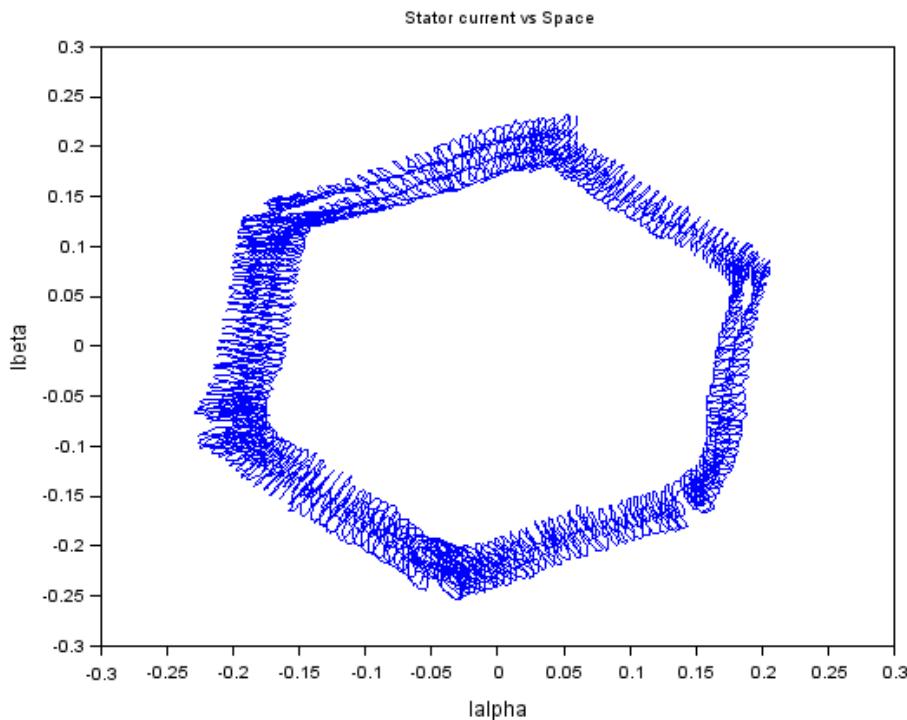


Fig 5.8 Space vector trajectory

---

\*max , m = max(A1,A2,...,An) finds maximum in A1...An

\*mean, m = mean(A) , mean value in matrix or vector A

\*stdev, m = stdev(mean), used to calculate standard deviation in mean values

### 5.7.2 Impulse Response.

From (Eq (4.2.6)), it is proved that the cross correlation between the input and output gives the Impulse response. Using the SCILAB simulation, Fig 5.9 shows the Impulse response of the system.

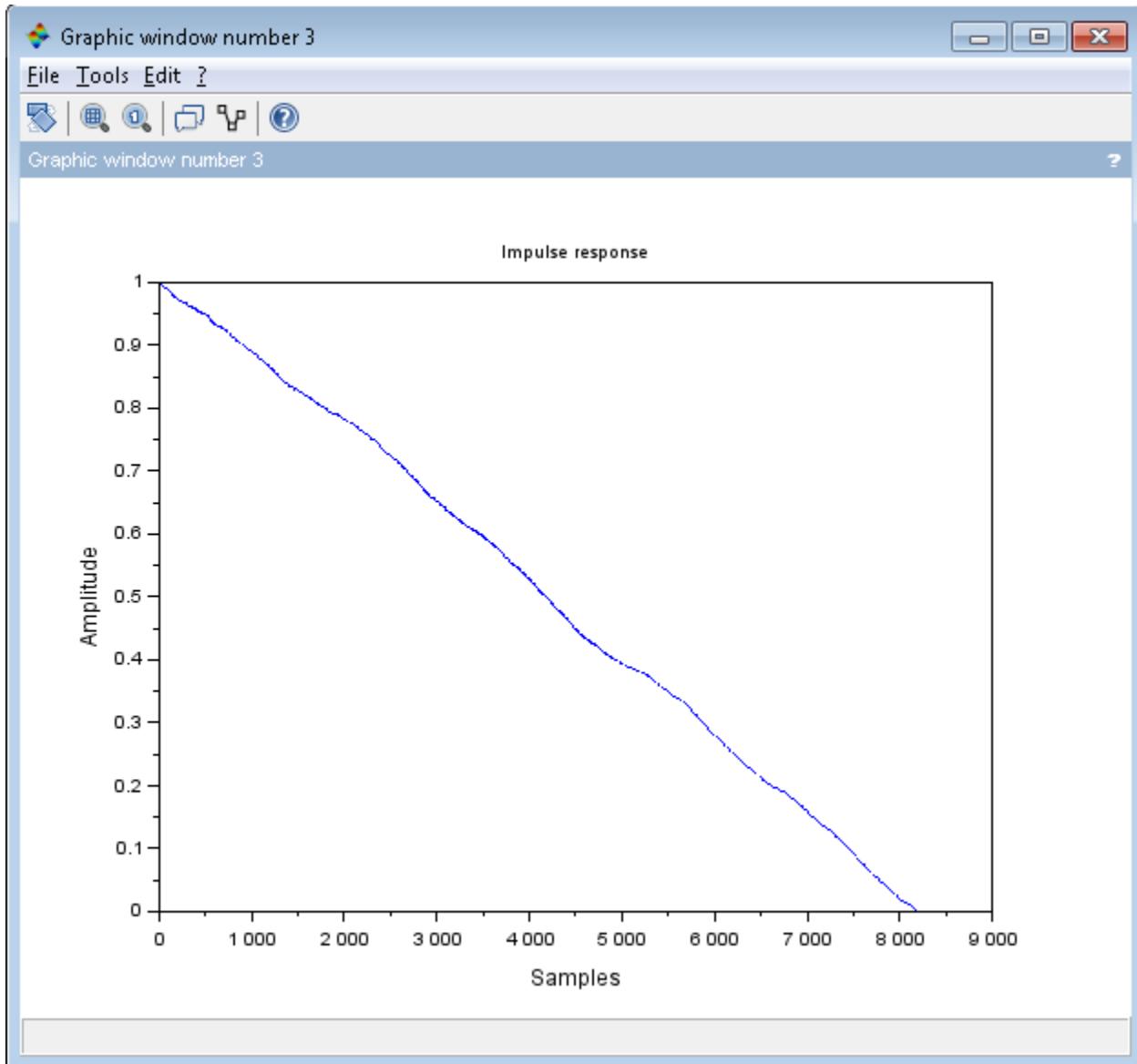


Fig 5.9 Impulse response of the system

### 5.7.3 Magnitude and Phase plot

The frequency response can be plotted using the impulse response [15]. Applying Fast Fourier Transform (FFT) should give the magnitude and phase plot.

In SCILAB, fft\* function is used. Fig 5.10 shows the frequency response obtained. It also shows the currents, phase A and phase B with the amplitudes and in phase.

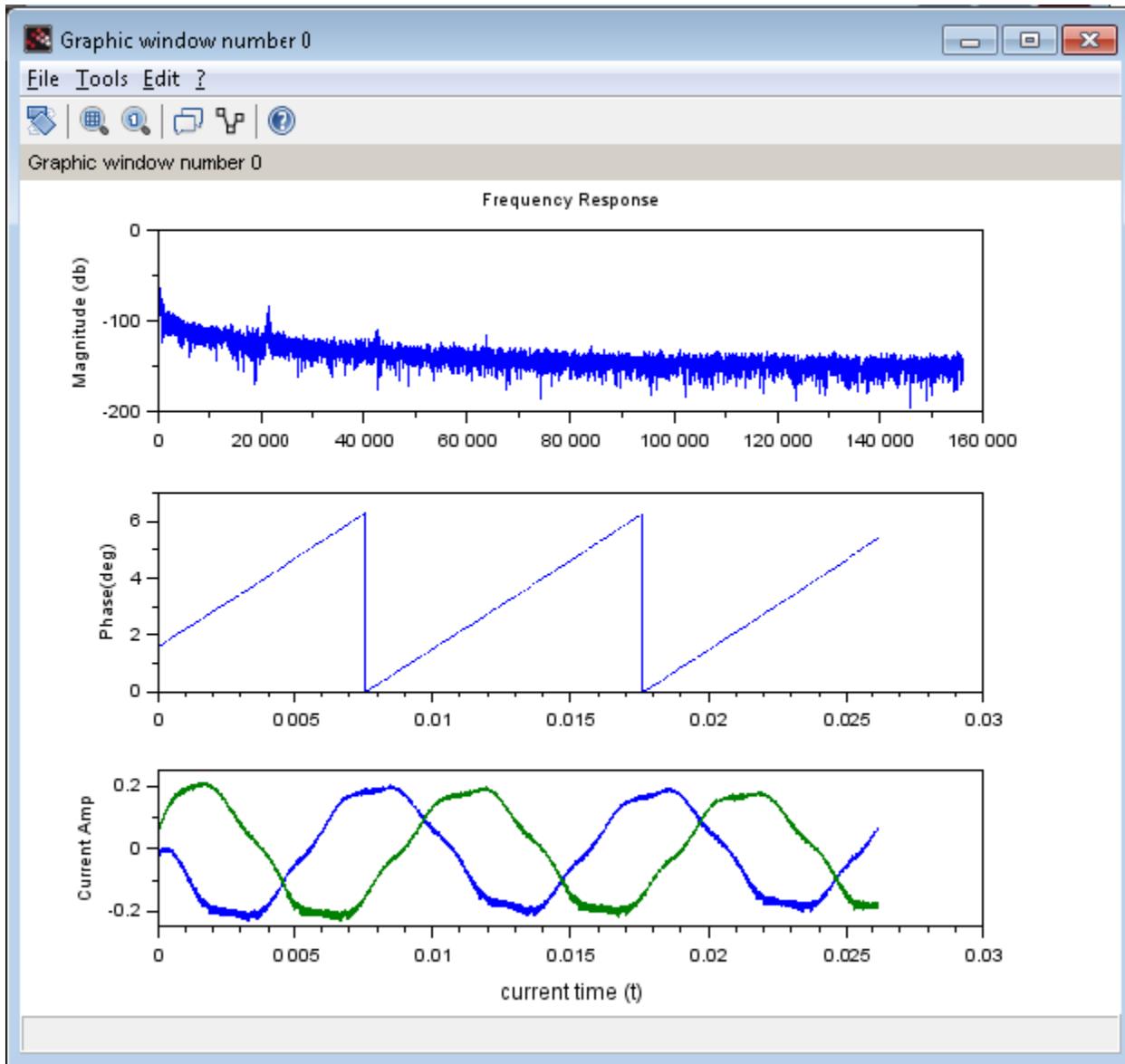


Fig 5.10 Frequency response and current.

\*fft, fft(A), A is real/complex array

# Chapter 6

## Open Phase Fault Detection

*In this Chapter, the introduction to open phase fault detection is dealt and the different methodologies to detect the fault is discussed.*

### 6.1 Introduction

Permanent Magnet Synchronous Motors (PMSM) have captured a wide range of Industrial market due to their high power density, precise controllability and large torque to inertia ratio. The reliability of PMSM drives is one of the critical factors in various industries. This is especially true in time critical applications, in these areas sudden breakdown of the drives result severe damages and economical losses.

Modern industry has widely used the fast acting and high reliable techniques for the maintenance and diagnosing the faults in the system, such that the techniques can reduce the unexpected failures and down time. Nevertheless, the faults are accidental and unavoidable, using the latest technology and fast response time electronics, the early detection of the faults is possible. These all is possible with the less expense and multi-tasking and/or all in one chip technology.

The open phase fault can be detected using the current signatures from the motor. A good insight of the important concepts related to motor is essential.

The Clarke ( $\alpha$ - $\beta$ ) and Park (d-q) reference frames is described below, which helps to understand motor model and parameters, described in *Chapter 7*.

### 6.2 Clarke and Park Transforms in FOC

#### 6.2.1 Overview

Clarke and Park transforms are used in high performance drive architectures (vector control) related to permanent magnet synchronous and asynchronous machines.

Through the use of the Clarke transform, the real ( $I_{ds}$ ) and imaginary ( $I_{qs}$ ) currents can be identified. The Park transform can be used to realize the transformation of the  $I_{ds}$  and the  $I_{qs}$  currents from the stationary to the moving reference frame and control the spatial relationship between the stator vector current and rotor flux vector.

The FOC consists of controlling the components of the motor stator currents, represented by a vector, in a rotating reference frame  $d, q$  aligned with the rotor flux. The vector control system requires the dynamic model equations of the induction motor and returns the instantaneous currents and voltages in order to calculate and control the variables.

The electric torque of an AC induction motor can be described by the interaction between the rotor currents and the flux wave resulting from the stator currents induction. Since the rotor currents cannot be measured with cage motors, this current is replaced by an equivalent quantity described in a rotating system coordinates called  $d, q$  following the rotor flux.

The Clarke transform uses three-phase currents  $I_a$ ,  $I_b$  and  $I_c$  to calculate currents in the two-phase orthogonal stator axis:  $I_\alpha$  and  $I_\beta$ . These two currents in the fixed coordinate stator phase are transformed to the  $I_{sd}$  and  $I_{sq}$  currents components in the  $d,q$  frame with the Park transform.

These currents  $I_{sd}$ ,  $I_{sq}$  and the instantaneous flux angle  $\theta$ , calculated by the motor flux model, are used to calculate the electric torque of a BLDC motor.

After such a transformation, the stator variables (currents and angle) are translated into a flux model. This flux model is compared with the reference values and updated by PI controllers. After a back transformation from field to stator coordinates, the output voltage will be impressed to the machine with Pulse Width Modulation (PWM).

### 6.2.2 Clarke Transform ( $\alpha$ - $\beta$ reference frame)

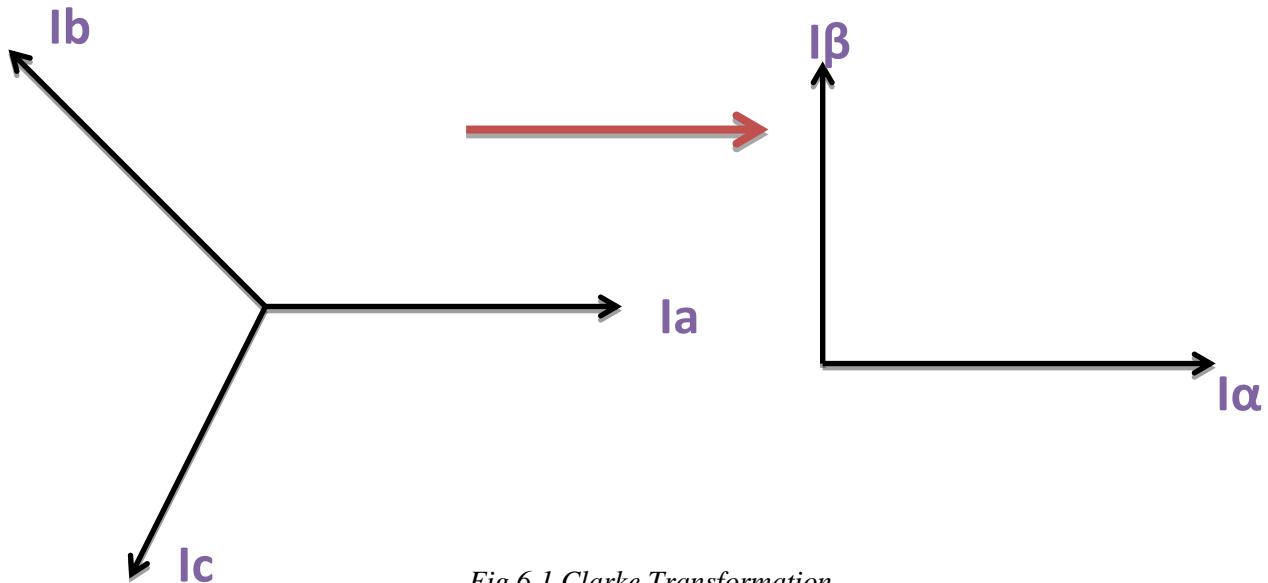


Fig 6.1 Clarke Transformation

Fig 6.1 shows the transformation of 3 – Ph system to 2 – Ph orthogonal systems.

## Mathematical Considerations

The mathematical transformation of Clarke transformation is transformation of 3 – Ph to 2-Ph orthogonal system:

$$\dot{i}_\alpha = \frac{2}{3} \cdot \dot{i}_a - \frac{1}{3} \cdot (\dot{i}_b - \dot{i}_c) \quad \text{Eq (6.1)}$$

$$\dot{i}_\beta = \frac{2}{\sqrt{3}} \cdot (\dot{i}_b - \dot{i}_c) \quad \text{Eq (6.2)}$$

$$\dot{i}_o = \frac{2}{3} \cdot (\dot{i}_a + \dot{i}_b + \dot{i}_c) \quad \text{Eq (6.3)}$$

Where,  $\dot{i}_\alpha$  and  $\dot{i}_\beta$  components of an orthogonal reference frame and

$\dot{i}_o$  the homopolar component of the system.

In many applications, the homopolar component is absent or is less important. In this way, in absence of homopolar component the space vector  $u = u_a + j.u_b$  represents the original three-phase input signal.

Consider now a particular case with  $\dot{i}_a$  superposed with  $\dot{i}_b$  and  $\dot{i}_a + \dot{i}_b + \dot{i}_c$  is zero, in this condition  $\dot{i}_a$ ,  $\dot{i}_b$  and  $\dot{i}_c$  can be transformed to  $\dot{i}_a$  and  $\dot{i}_b$  with following mathematical transformation.

$$\dot{i}_\alpha = \dot{i}_a \quad \text{Eq (6.4)}$$

$$\dot{i}_\beta = \frac{1}{\sqrt{3}} \cdot \dot{i}_a + \frac{2}{\sqrt{3}} \cdot \dot{i}_b \quad \text{Eq (6.5)}$$

$$\dot{i}_a + \dot{i}_b + \dot{i}_c = 0 \quad \text{Eq(6.6)}$$

### 6.2.3 Park transformation (d-q reference frame)

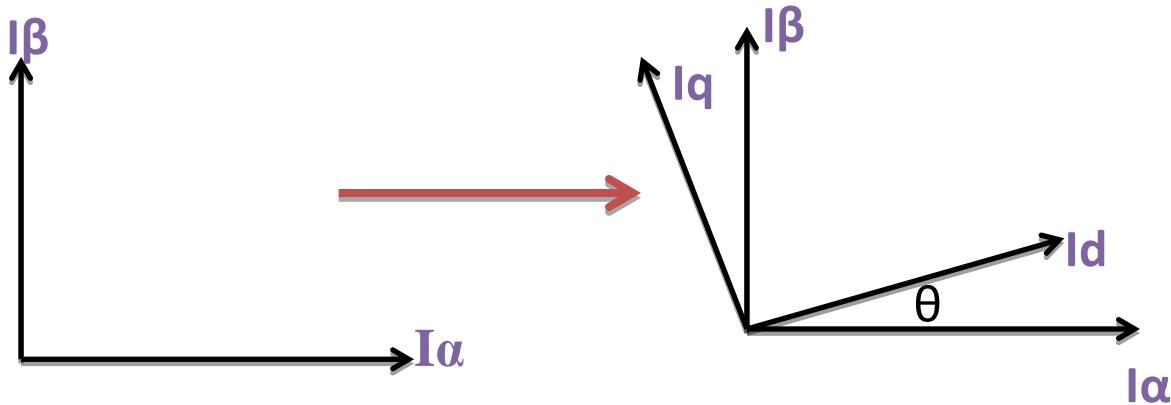


Fig 6.2 Park Transformation

Fig 6.2 shows the rotating coordinate system. The two phases a, b frame representation calculated with the Clarke transform is then fed to a vector rotation block where it is rotated over an angle  $\theta$  to follow the frame d,q attached to the rotor flux.

The rotation over an angle  $\theta$  is done according to the formula,

$$i_{sd} = i_\alpha \cdot \cos(\theta) + i_\beta \cdot \sin(\theta) \quad \text{Eq (6.7)}$$

$$i_{sq} = -i_\alpha \cdot \sin(\theta) + i_\beta \cdot \cos(\theta) \quad \text{Eq (6.8)}$$

Fig 6.3 illustrates the vector convention used for further stator current analysis.

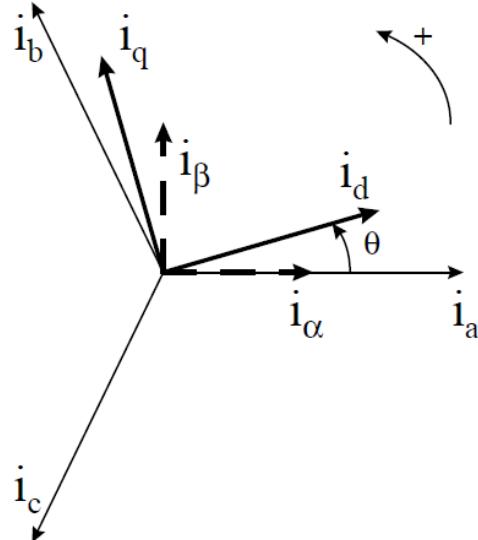


Fig 6.3 Vector Convention [17]

### **6.3 Fault diagnosis and detection methods**

The various methods to identify the open phase fault is identified and with these methods the algorithm is developed. At the end of such effort an efficient solution to detect the open phase fault is delivered.

The details of the each method identified are described in *Chapter 7*.

#### **6.3.1 Current Signature Analysis (Motor CSA)**

It is observed that the technique called Motor Current Signature Analysis is based on the current monitoring of the induction motor. Therefore it is not very expensive. The MCSA uses the current spectrum of the machine for locating fault frequencies.

In the presence of the fault, the frequency spectrum of the current becomes different from the healthy motor. Analyzing the trajectory traced by the vector would make possible to detect the fault.

Apart from this, an Inductance and flux trajectory is analyzed under open phase and results are documented with this Thesis.

#### **Why MCSA?**

- Electrical signals are easier, simpler, and cheaper to be measured and stored.
- Online fault monitoring can be achieved without the need to shut down the motor.
- Faults can be identified at an early stage before becoming severe. Hence, sudden motor shutdown is avoided, and maintenance cost is reduced.

#### **6.3.2 Vector Space Angle**

In this Master Thesis, apart from the Motor CSA method to identify the open phase fault, an effort is made to keep a track of vector rotating angle [*section 6.2*] and determination open phase fault in the motor is made.

#### **Why this method?**

- In the vector space upon an open phase fault, the current trajectory tends to change and it is different for each of the phase open.
- Simple calculations of the rotating angle and monitoring.

#### **6.3.3 Hough Transformation**

Hough transformation is the image processing algorithm. In the Thesis, this has been identified as computationally efficient algorithm to detect open phase fault.

# Chapter 7

## Current Signature Analysis

In this chapter, the detailed study on current signature analysis is depicted. How the condition of the motor could be analyzed using the motor signature pattern is shown along with the simulation and experimental results.

### 7.1 Introduction

In order to deal with the Motor Current Signature Analysis (MCSA), the contributions to the work has to be from various accepts. First, the current fault signature is analyzed. Secondly, the Motor model has been identified and the parameters of Inductance and flux have been determined. Using the equations, Hardware in the Loop (HiL) simulation is carried out and the stator trajectory is distinguished.

### 7.2 Current Fault Signatures

Open phase fault can be detected by analyzing the vector trajectory in  $\alpha$ - $\beta$  reference frame (details in *chapter 6*). The phase plane plot for each open phase is unique [18].

#### 7.2.1 Results and Simulation

Referring to Fig 3.6, the open phase simulation is conducted using the Xilinx Motor control kit. In the Thesis, the open phase is established by opening the each of the phase wires, which are directly connected to the motor.

The wire notations are described in Fig 3.9. The simulation is conducted while motor is in motion.

*Note: Caution has to be taken while connecting and disconnecting wires for short circuit and grounding*

- **Healthy Stator Phase:** As shown in the Fig 7.1, the vector trajectory in the healthy condition of the motor traces hexagonal path.

For the healthy condition of the motor the following equations from *Eq (6.4, 6.5, 6.6)* holds good.

$$\begin{aligned}\triangleright \dot{i}_\alpha &= \dot{i}_a \\ \triangleright \dot{i}_\beta &= \frac{1}{\sqrt{3}} \cdot \dot{i}_a + \frac{2}{\sqrt{3}} \cdot \dot{i}_b \\ \triangleright i_a + i_b + i_c &= 0\end{aligned}$$

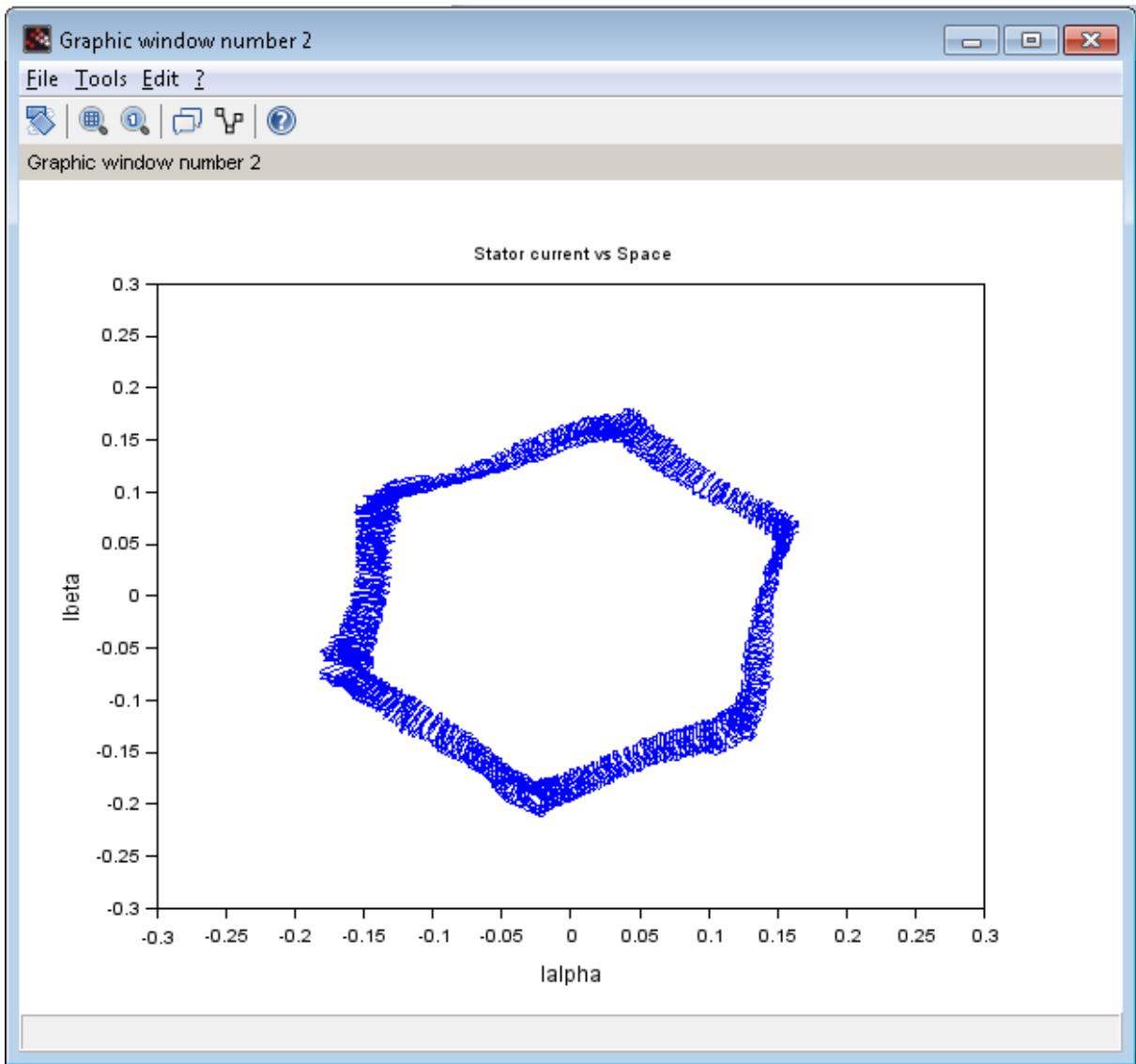


Fig 7.1 Healthy Motor Condition: Locus of  $\alpha$ - $\beta$  currents.

- **Open Phase A:** For an open phase A, plugging out the Yellow (Phase A) wire.

The Eq (6.4, 6.5) for open A phase is,

$$\begin{aligned} > \dot{\mathbf{i}}_a = 0 \\ &\left. \begin{aligned} \dot{\mathbf{i}}_a &= 0 \\ \mathbf{i}_\beta &= 2/\sqrt{3} \end{aligned} \right\} \end{aligned}$$

Fig 7.2 shows the trajectory traced by the vector under phase A open.

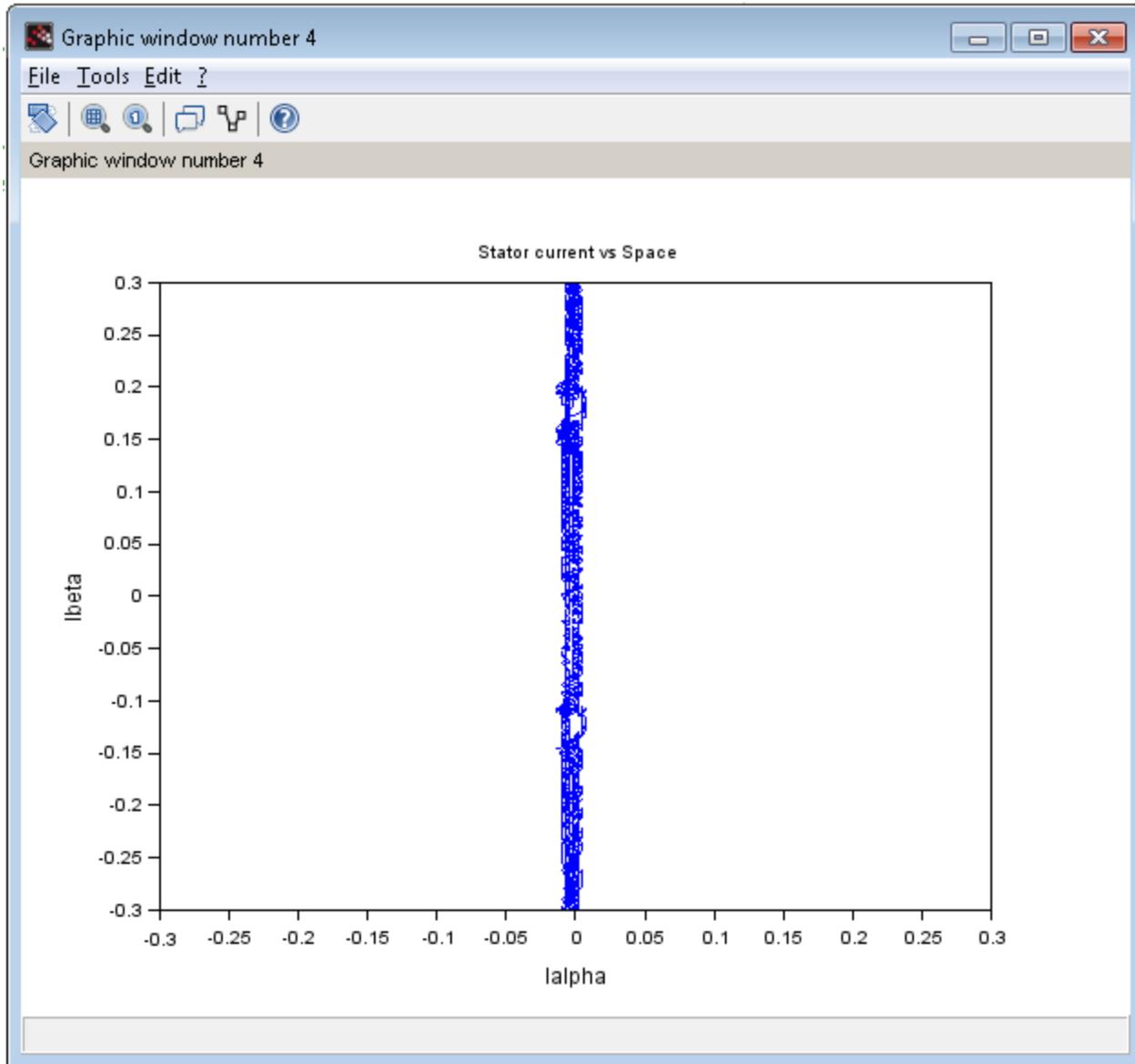


Fig 7.2 Open A Phase Condition: Locus of  $\alpha\beta$  currents

- **Open B Phase:** For a B open phase, the red (Phase B) is plugged out.

The Eq (6.4, 6.5) for open B phase is,

$$\begin{aligned} \Rightarrow \mathbf{i}_b &= \mathbf{0} \\ &\left. \begin{cases} \mathbf{i}_a = \mathbf{i}_a \\ \mathbf{i}_b = 1/\sqrt{3} \end{cases} \right\} \end{aligned}$$

Fig 7.3 shows the trajectory traced by the vector under phase B open.

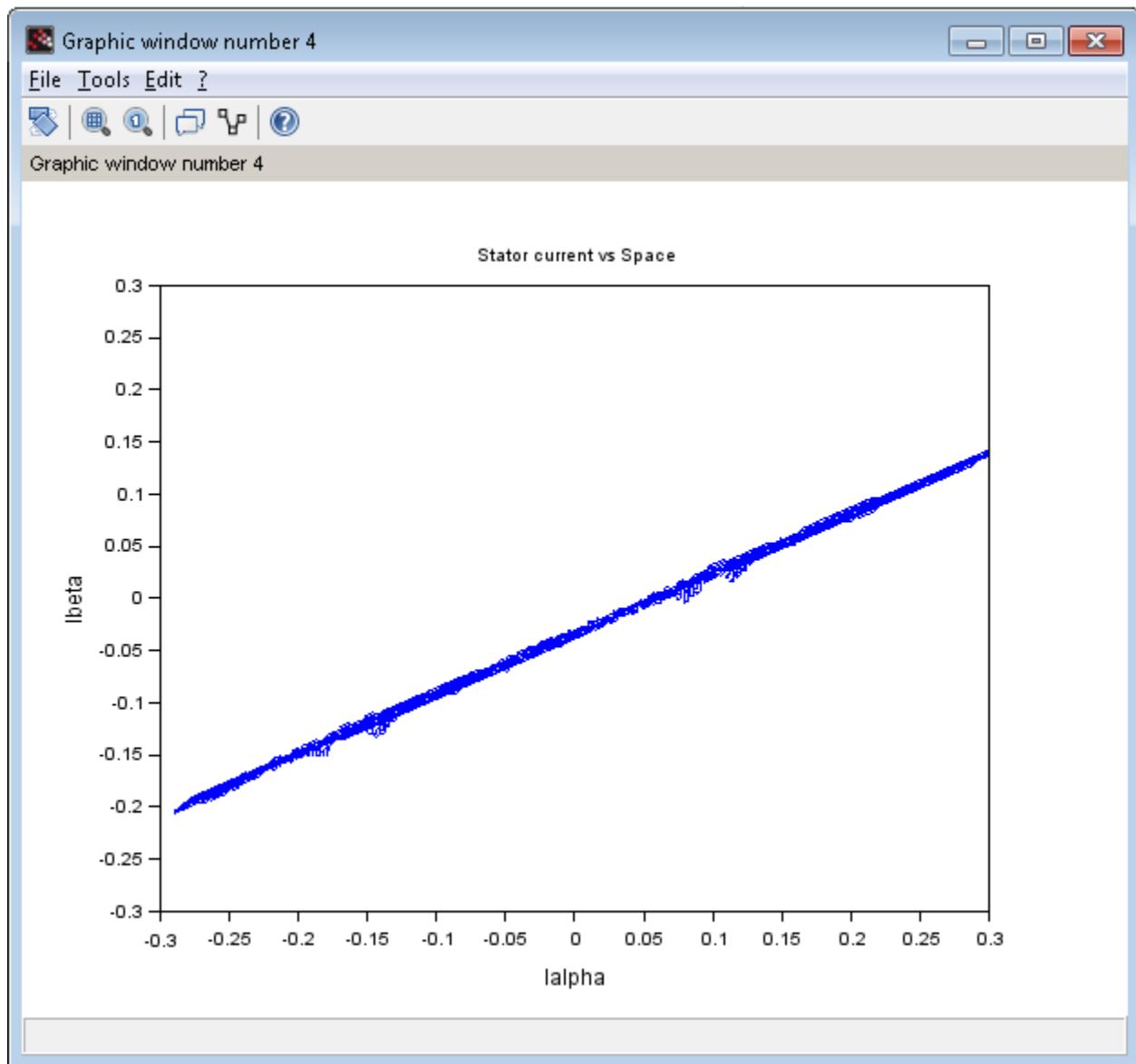


Fig 7.3 Open B Phase Condition: Locus of  $\alpha\text{-}\beta$  currents

- **Open C Phase:** For a C open phase, the black (Phase C) is plugged out.

The Eq (6.4, 6.5) for open C phase is,

$$\Rightarrow \mathbf{i}_c = 0 \quad \left. \begin{array}{l} \mathbf{i}_a = -\mathbf{i}_b \\ \mathbf{i}_b = 1/\sqrt{3} * \mathbf{i}_b \end{array} \right\}$$

Fig 7.4 shows the trajectory traced by the vector under phase C open.

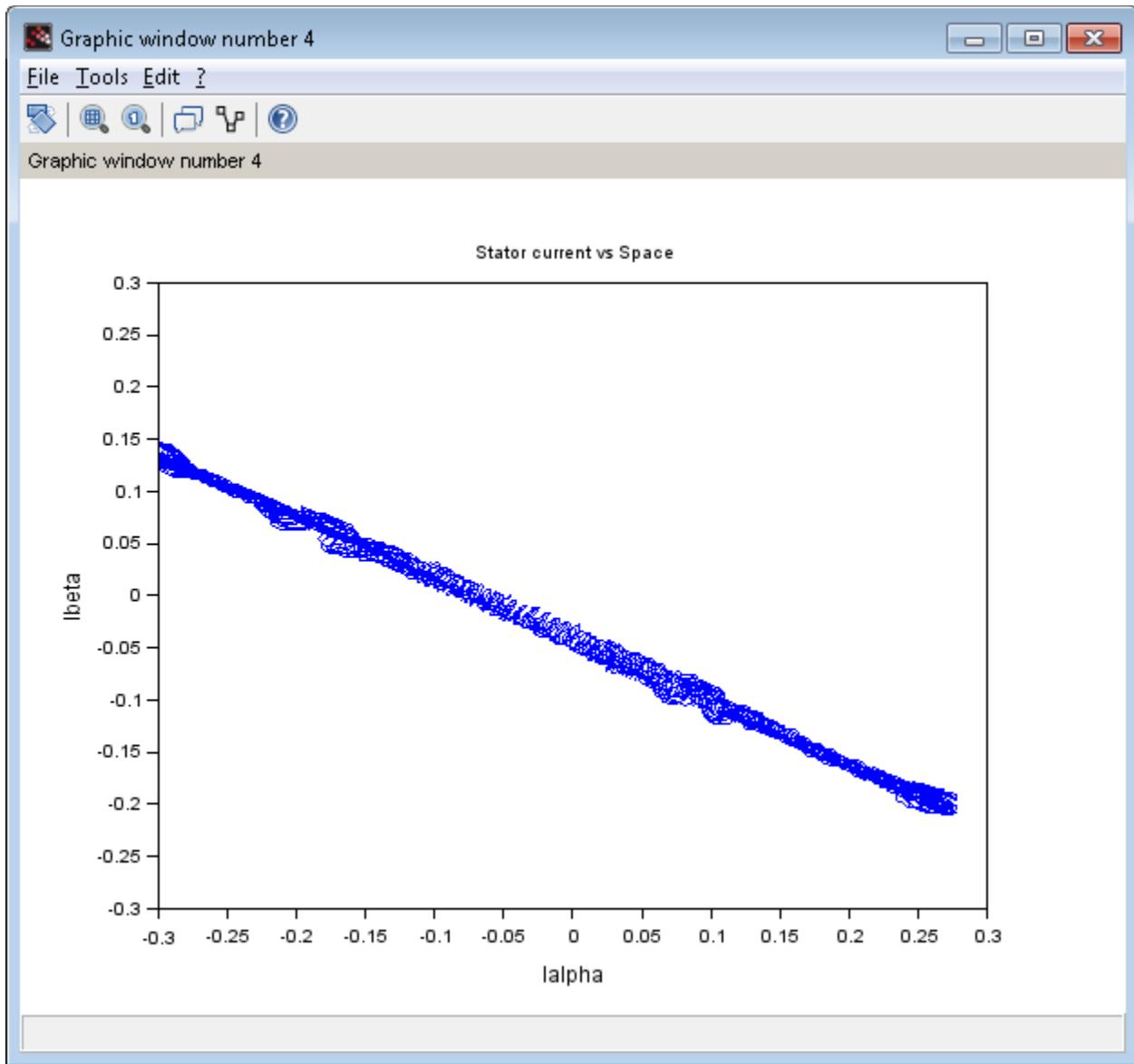


Fig 7.4 Open C Phase Condition: Locus of  $\alpha$ - $\beta$  currents

Hence, from the Fig 7.2, 7.3, 7.4 the particular phase open condition could be detected. The advantages and disadvantages of this method is discussed *Chapter 10, Results*.

### 7.3 PMSM drive model

Fig 7.5 shows the Xilinx Zynq AP SoC with the Motor control drive.

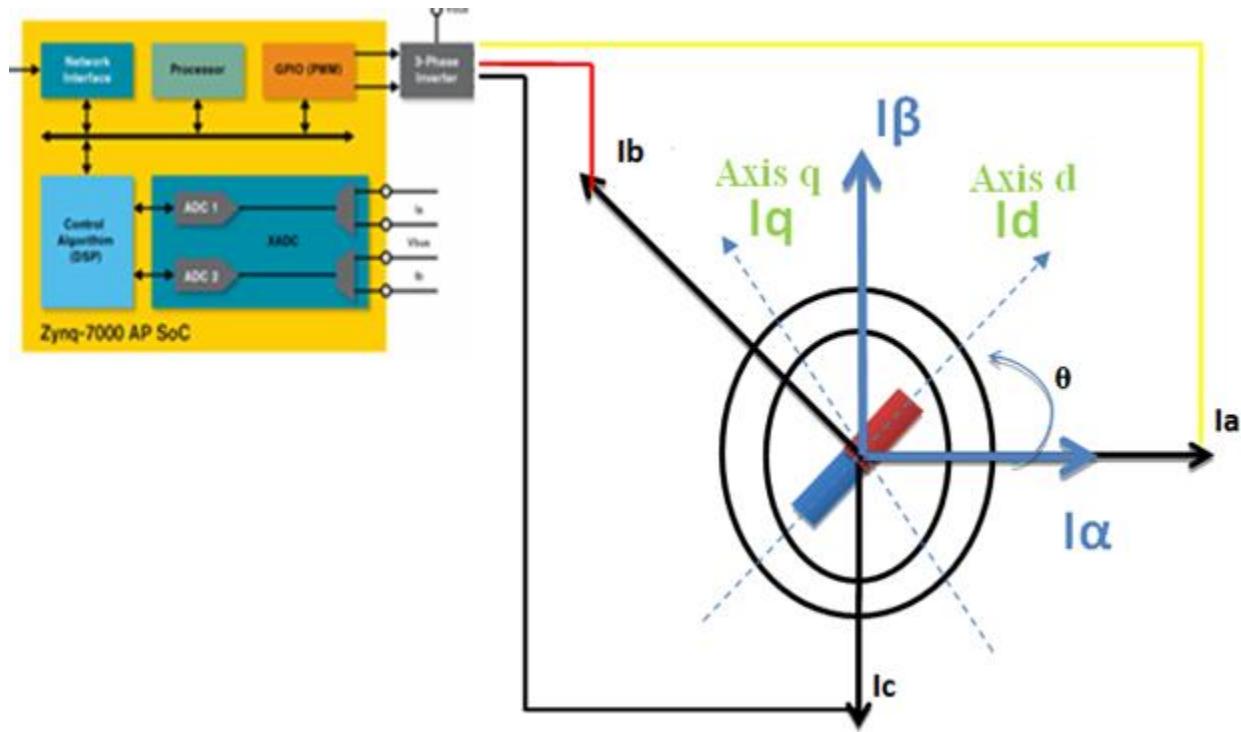


Fig 7.5 Xilinx AP SoC Motor Model

The stator d and q axis flux linkages can be expressed as:

$$\mathcal{O}_d = L_d \cdot i_d + \mathcal{O}_{md} \quad \text{Eq (7.1)}$$

$$\mathcal{O}_q = L_q \cdot i_q \quad \text{Eq (7.2)}$$

The electromagnetic torque is given by,

$$T_e = N_p (\mathcal{O}_{md} \cdot i_q + (L_d - L_q) i_d i_q) \quad \text{Eq (7.3)}$$

Where,

- $\mathcal{O}_d, \mathcal{O}_q$  Stator d and q axis flux linkages
- $\mathcal{O}_{md}$  Permanent Magnet Flux linkage
- $i_d, i_q$  Stator d and q axis currents

## Synchronous Inductances Measurement

### 7.3.1 Background

The synchronous inductances of Interior Permanent Synchronous Motor (IPMSM) winding are different ( $L_d < L_q$ ), because of lower reluctance in  $q$  – axis. The synchronous inductances of Surface Mounted Permanent Magnet Synchronous Motor (SMPM) motor are almost equal, because the permanent magnets are surface mounted and reluctance is the same in every position, that is:  $\mu_{PM} \sim \mu_{air} \rightarrow L_d \sim L_q$ ,

Where,  $\mu_{PM}$  is the relative permeability of the permanent magnet, and  $\mu_{air}$  is the relative permeability of the air.

Fig 7.6 depicts the reluctance paths of  $d$ - and  $q$  axis in IPMSM.

In practice, magnetic circuits are subjected to saturation as the current increases. Especially, when current  $I_q$  is increased, the value of  $L_q$  is decreased. Since  $I_d$  is maintained to zero or negative value (demagnetizing) in most operating conditions, saturation of  $L_d$  rarely occurs. The flux linkage  $\emptyset_{md}$  and  $L_d$  are subject to armature reaction. Fig 7.7 makes it clear.

In order to measure synchronous inductance, the users must maintain balanced three phase current condition. When the rotor is aligned with the center of Phase A winding,  $L_d$  ( $L_q$ ) can be derived from the measured equivalent inductance  $L$  of the motor circuit.

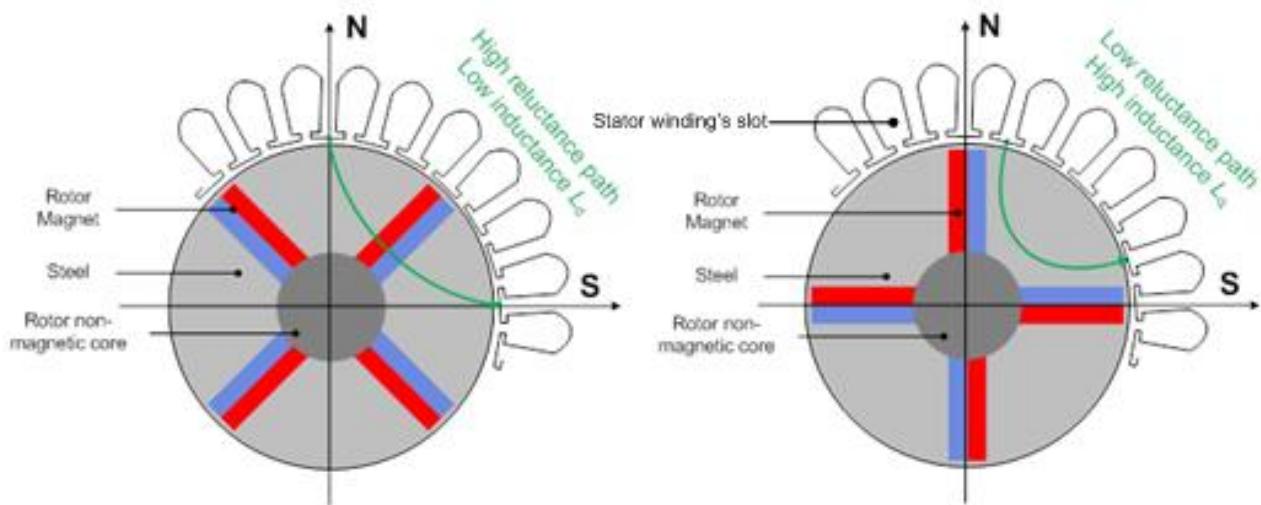


Fig 7.6 Reluctance paths in  $d$ - and  $q$ -axis. Source: [19]

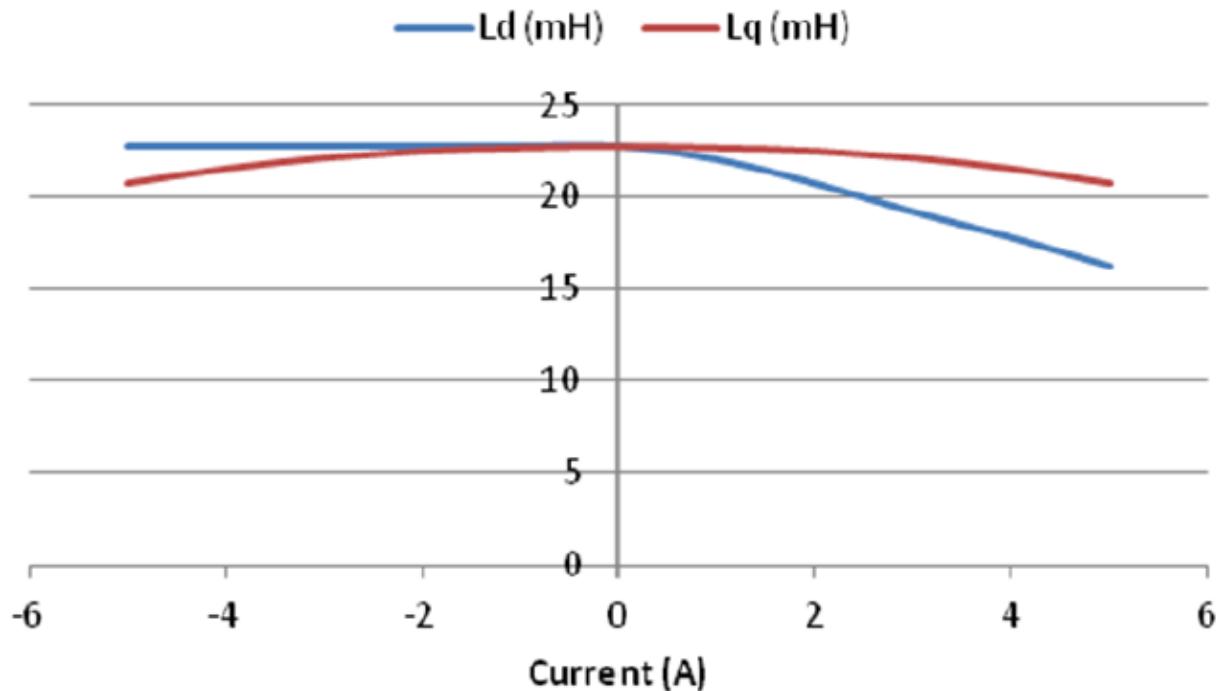


Fig 7.7 Typical Inductance characteristic in PMSM. Source: [19]

### 7.3.2 Inductance Measurement Procedure

- Depending on the rotor angle  $\theta$ , it is possible to measure inductance in d –axis and q-axis, where  $L$  is the total inductance.

$$L_d = (2/3) L \quad (\theta = 0 \text{ deg}) \quad \text{Eq (7.4)}$$

$$L_q = (2/3) L \quad (\theta = 90 \text{ deg}) \quad \text{Eq (7.5)}$$

- When the rotor is aligned in the Phase A ( $\theta = 0 \text{ deg}$ ) and locked, then the current response is first order RL circuit.

$$i_d = V/R \times (1 - e^{-t/T}) \quad \text{Eq (7.6)}$$

- Where  $T$  is the time constant of the circuit

$$T = L/R$$

$$\text{Hence, } L_d = 2/3 \times T_d R, \text{ Where } T_d \text{ is } d \text{ axis time constant.} \quad \text{Eq (7.7)}$$

Similarly, the calculation can be undertaken for q axis. Then the equation would as follows,

$$L_q = 2/3 \times T_q R \quad \text{Eq (7.8)}$$

## 7.4 Guide to measure inductance in d- and q-axis

The equipment required for the inductance in d- and q –axis measurements are as follows:

1. DC Power supply
2. Oscilloscope
3. Current and Voltage probes

Fig 7.8 shows the basic set up arranged in the laboratory.



Fig 7.8 Basic set up to measure the motor parameters.

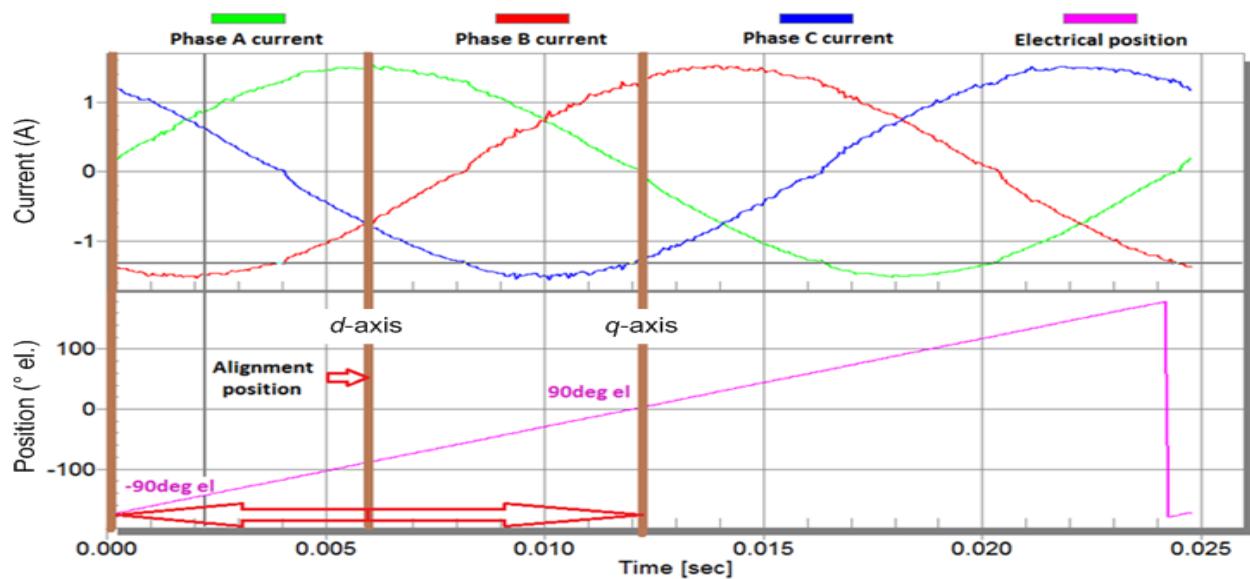


Fig 7.9 d- and q- axis alignment. Source [19]

#### 7.4.1 d- axis inductance measurement

The steps followed to measure the d-axis inductance are as follows:

1. Align the motor to phase A [Fig 7.9]. Phase A is connected to the positive potential and Phase B is grounded. (Note: Since the motor used in Thesis is delta connected phase C is left open).
2. Apply negative voltage to Phase A and positive voltage to Phase B. Usual level of the current is about 10% of the rated phase current.
3. Measure the step response of the current by a current probe. With Fig 7.10, Step response is noted.
4. Using the Eq (7.7) d-axis Inductance is measured.

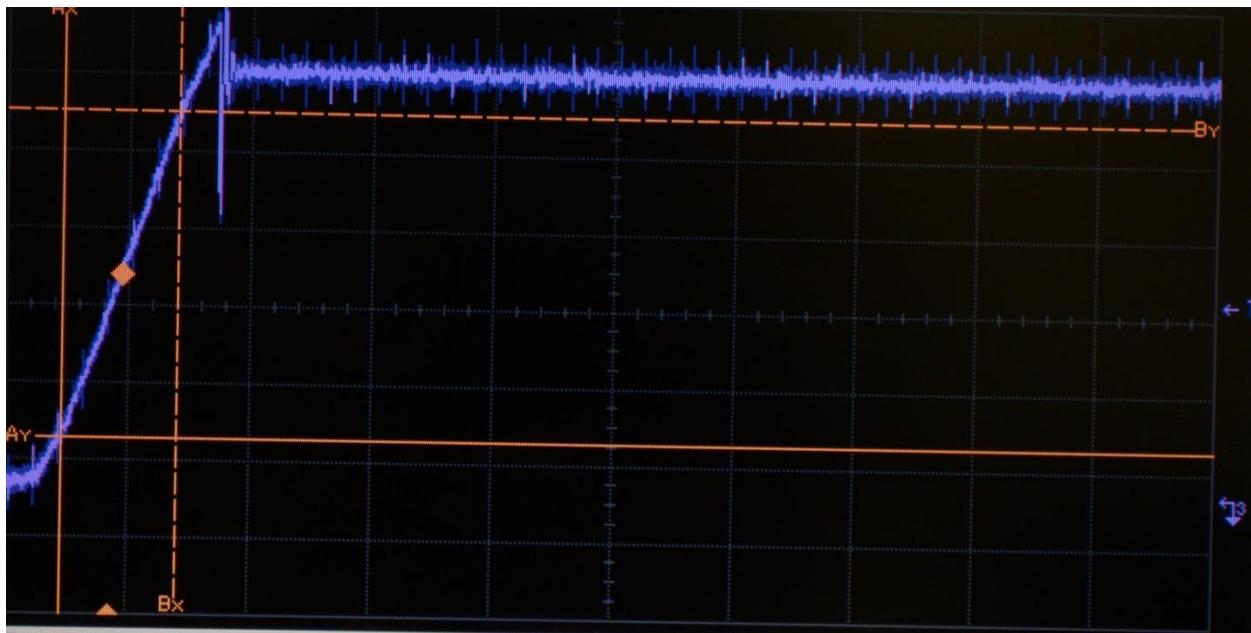


Fig 7.10 Step Input in d-axis alignment.

#### 7.4.2 q- axis inductance measurement

The steps followed to measure the q-axis inductance are as follows:

1. Align the motor to phase B [Fig.7.9]. Connect the Phase B to positive terminal and C is grounded with A as floating.
2. Lock the rotor in this position because current step response in q-axis creates torque.
3. Generate the step response in the following configuration: phase A is connected to positive potential (+) of the voltage source and phase B to negative (-) with C open. Fig 7.11, step response is noted.
4. Calculate Lq using Eq (7.8).

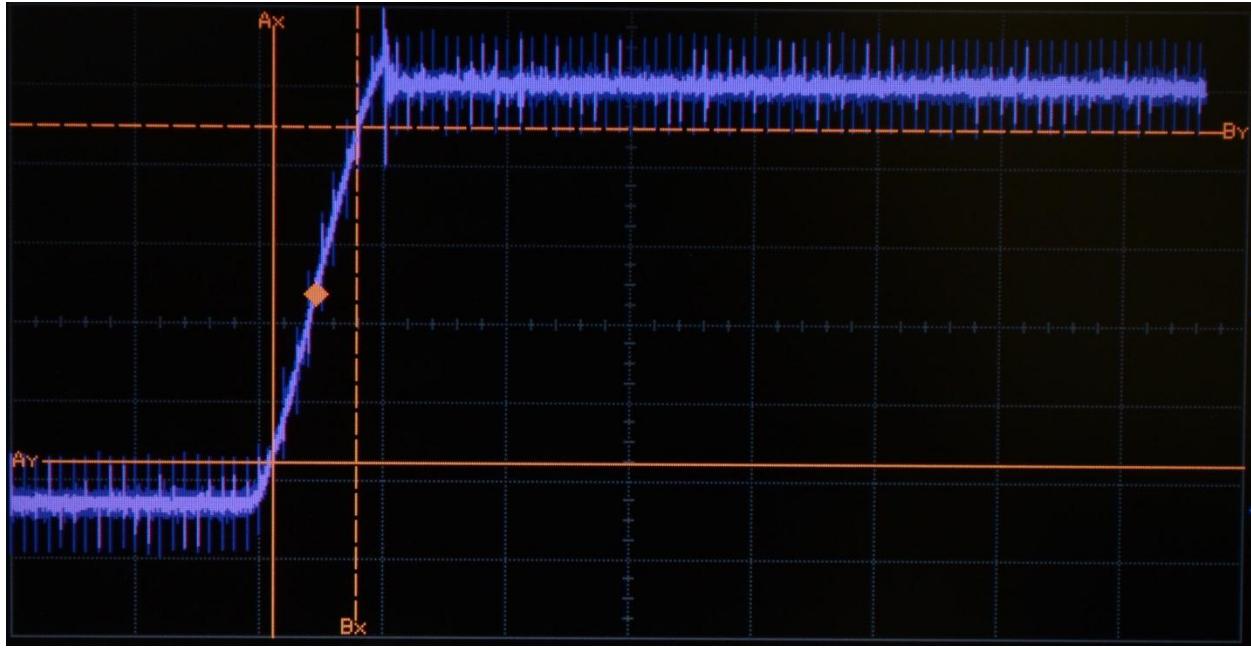


Fig 7.11 Step input in q – axis alignment

The Rise time for both d- and q-axis inductance measurements is read directly from the Agilent Infiniium 54855A DSO, digital oscilloscope.

## 7.5 Calculations and Results

Following the steps followed in 7.4.1, the rise time calculated is,

$$T_d = 38\text{ms} \quad \text{Eq (7.9)}$$

And,  $T_q = 45\text{ms}$  Eq (7.10)

From the Table 3.2, Line to Line Resistance ( $R$ ) =  $2.40 \Omega$

Using Eq (7.7),

$$L_d = \frac{2}{3} * 38\text{m} * 2.40$$

$L_d = 60.8\text{mH}$

Eq (7.11)

And, following the steps described in 7.4.2, rise time is,

$$T_q = 45\text{ms}$$

Using Eq (7.8),

$L_q = 72\text{mH}$

Eq (7.11)

From the Eq (7.11, 7.12) it is clear that  **$L_d < L_q$  for PMSM drives [18]**.

Reconsidering Eq (7.1, 7.2),

$$\dot{\Phi}_d = L_d \cdot i_d + \dot{\Phi}_{md}$$

$$\dot{\Phi}_q = L_q \cdot i_q$$

$\dot{\Phi}_{md}$  is the Permanent Magnet Flux linkage, from the torque speed curve [9], the value is

$$\dot{\Phi}_{md} = 5.29 \text{ mWb}$$

### 7.5.1 SCILAB Simulation

Considering the PMSM drive model Fig 7.5 and the equations derived, the SCILAB mathematical operations is carried out and simulated.

Using the equations, Eq (6.4, 6.5)  $i_\alpha, I_\beta$  is calculated as,

$$\begin{aligned} i_\alpha &= i_a \\ i_\beta &= \frac{1}{\sqrt{3}} \cdot i_a + \frac{2}{\sqrt{3}} \cdot i_b \end{aligned}$$

$i_a, i_b$  are obtained from the SCIALB API [Chapter 5, 5.1]

With the park transformation formulas, Eq (6.7, 6.8)

$$i_{sd} = i_\alpha \cdot \cos(\theta) + i_\beta \cdot \sin(\theta)$$

$$i_{sq} = -i_\alpha \cdot \sin(\theta) + i_\beta \cdot \cos(\theta)$$

Thus,  $\dot{\Phi}_d, \dot{\Phi}_q$  are calculated by using,

$$\dot{\Phi}_d = L_d \cdot i_d + \dot{\Phi}_{md}$$

$$\dot{\Phi}_q = L_q \cdot i_q$$

The real time data of  $i_a, i_b$  is used for each acquisition display.

Here, the simulation result is presented. Using the motor model derived the motor parameters are studied under the fault condition of the motor.

The motor is run at the constant speed and for demonstration the open A phase is performed. With the torque load of about say, 2Nm 7.12 and 7.13 show the motor behavior.

Fig 7.12 shows the d – axis and q- axis currents. Upon the open phase, the d-axis current is directly proportional to torque. Q-axis currents tend to oscillate with the current amplitude of  $\pm 0.5\text{A}$ .

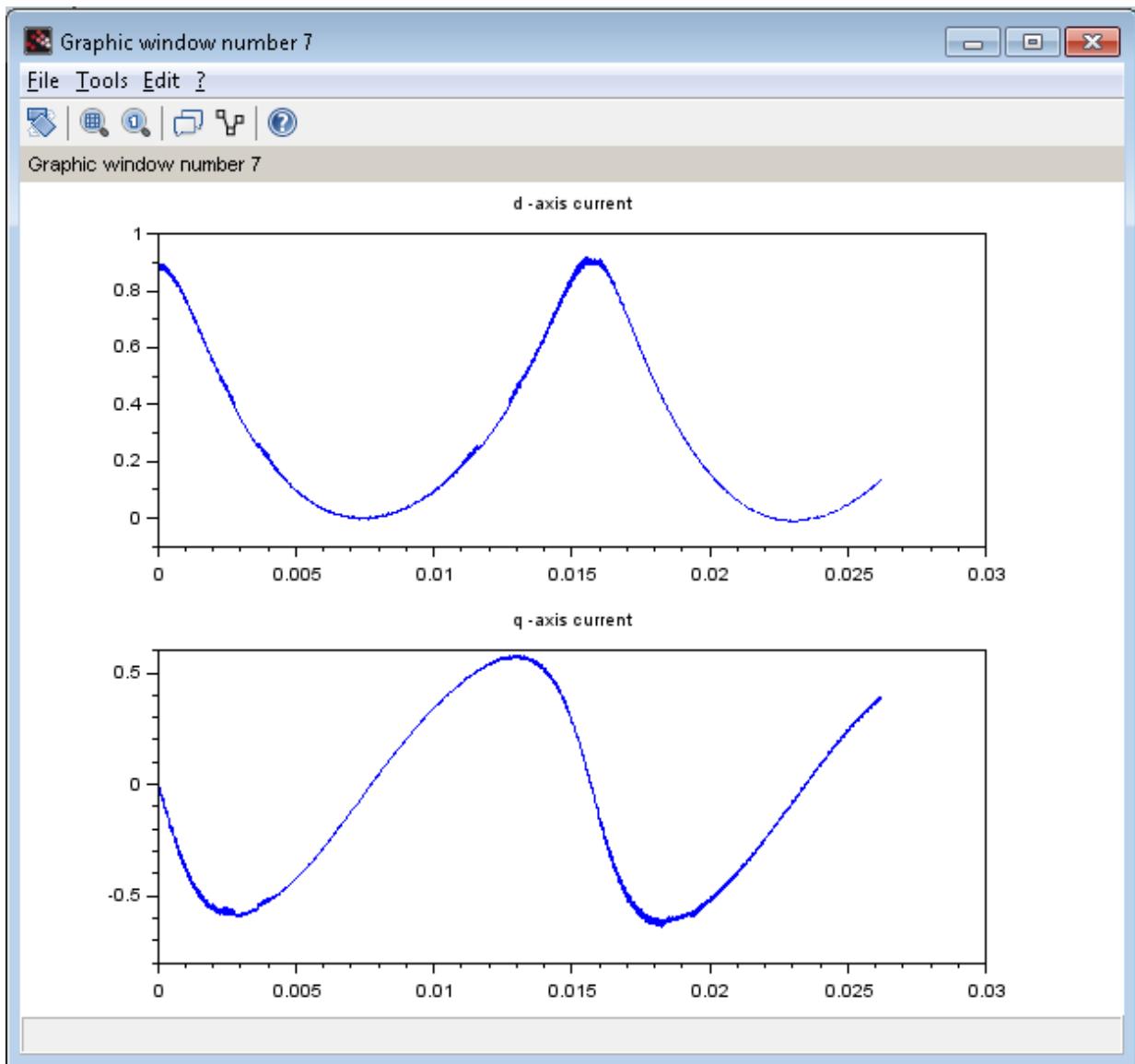


Fig 7.12  $I_d, I_q$  under open phase

Fig 7.11 shows the d- and q- axis flux behavior under the open phase condition. The open phase fault produces electromagnetic torque oscillations which cause abnormal vibrations and give abnormalities to the drive operations. The d-q axis fluxes are affected with ripples as shown in Fig 7.13.

d-axis stator flux exhibit the torque characteristics upon the open phase and q-axis flux sustain with the oscillations with an average value equal to 0.

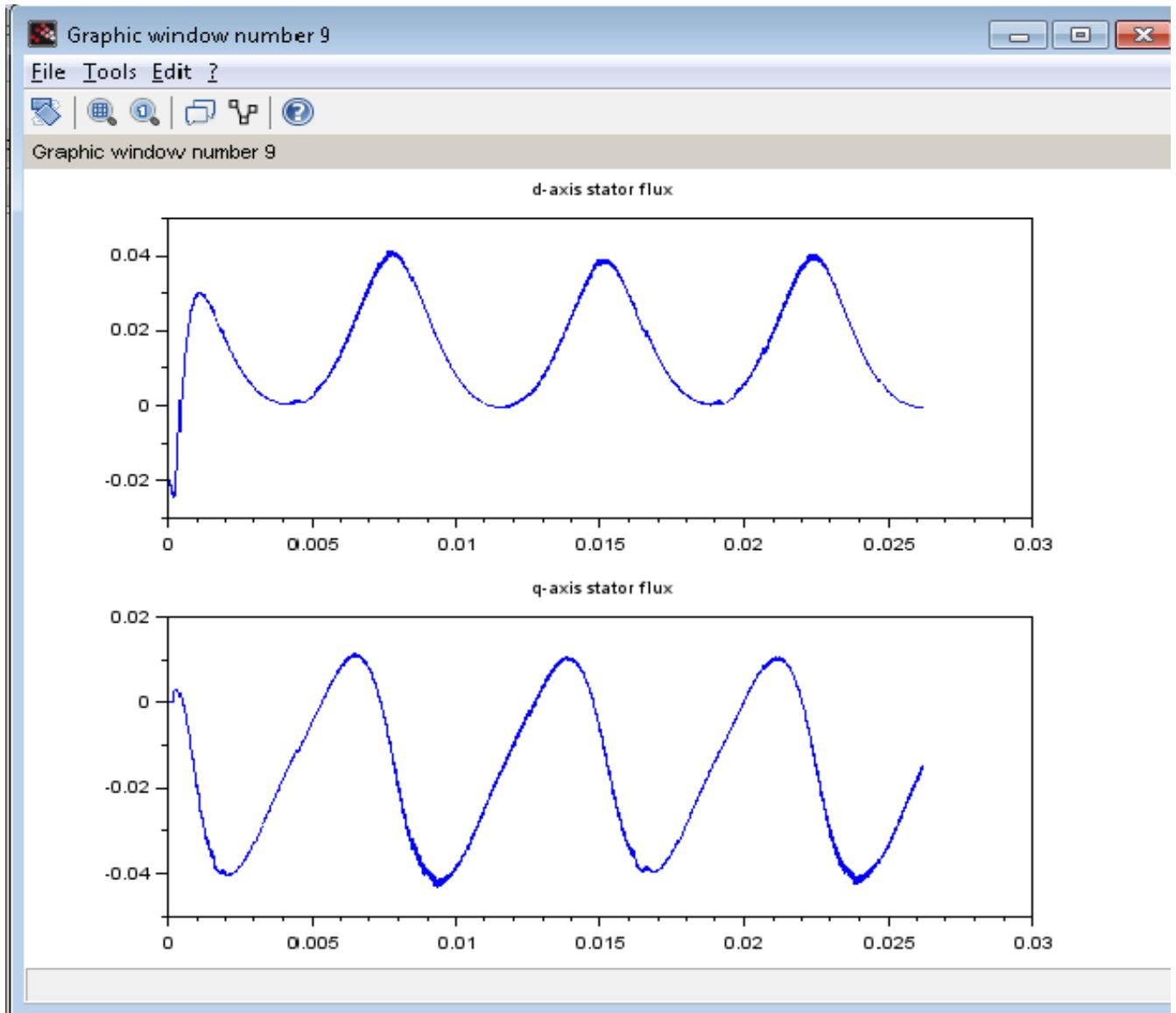


Fig 7.13  $\mathcal{O}_d, \mathcal{O}_q$  under Open phase.

Hence, using the motor model described with Fig 7.5, the characteristics for the d- and q axis currents/fluxes under the open phase fault is simulated. This would provide a significant data about the condition of the motor. Using Xilinx AP SoC, this is possible real time.

# Chapter 8

## Vector Space Angle

In this chapter, one of the methodologies to detect the open phase fault is dealt. Here mainly, the possibility of monitoring the vector space is studied and implemented. The results are documented along with the simulation.

### 8.1 Introduction

The Clarke and Park transform gives the possibility to convert the fixed frame to rotating frame (*Chapter 6*). With the knowledge of vector space ( $\alpha$ - $\beta$ ) angle, the angle of rotation  $\theta$  can be monitored to detect the rotor position.

From Chapter 7, it is clear that for each of the open phase, a vector trajectory exhibit the unique path. Hence, the angle associated with the vector, which is the resultant of the  $I_\alpha$  -  $I_\beta$  can be calculated and monitored to check the trajectory.

### 8.2 Algorithm Flow chart

Eq (6.4, 6.5) gives the  $I_\alpha$  -  $I_\beta$  real time running HiL using Xilinx AP SoC. The value is calculated for every new set up of current acquisition.

In  $I_\alpha$  -  $I_\beta$  reference fame, the angle can be calculated using inverse tangent mathematical function. Hence,

$$\text{IsAngle} = \tan^{-1}(I_\beta / I_\alpha)$$

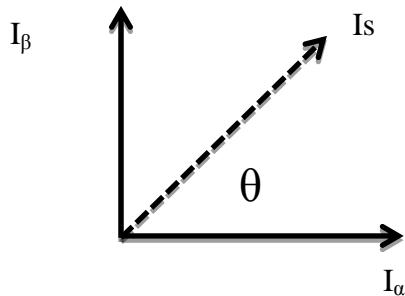


Fig 8.1 shows the activity diagram used to track the vector space angle in SCILAB environment.

In the SCILAB environment, the detection of open phase is carried out upon compilation of the demo time\*, i.e the obtained angle theta is buffered until the motor is stopped.

---

\*Demo time is specified in SCILAB to run  
the demonstration

**Flow Chart:**

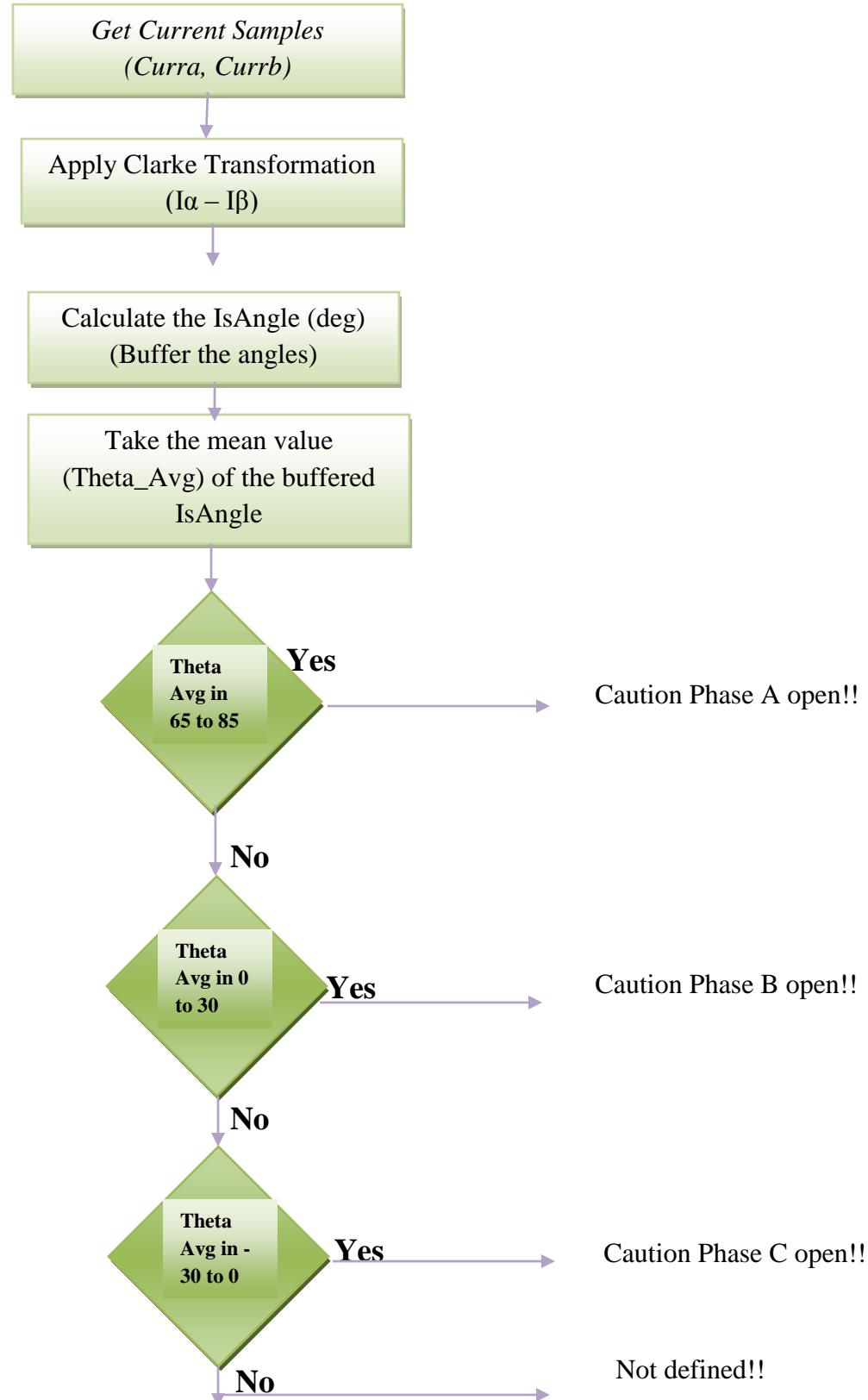


Fig 8.1 Vector Space Angle activity diagram

### 8.3 Simulation and Results.

Following the Fig 8.1, the simulation of the results is as shown below.

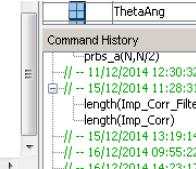
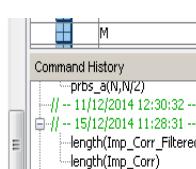
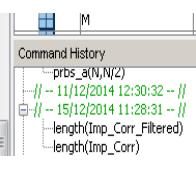
Condition	Indication	Theta Average (Deg)
Open A Phase	<pre>~~~~~Caution Phase A open!! : Avg Speed -&gt; 2661~~~~~ 131 display in 90 seconds: rate is 1.455556 That's all folks!</pre> 	69.2
Open B Phase	<pre>~~~~~Caution Phase B open!! : Avg Speed -&gt; 2659~~~~~ 138 display in 90 seconds: rate is 1.533333 That's all folks!</pre> 	28.9
Open C Phase	<pre>~~~~~Caution Phase C open!! : Avg Speed -&gt; 2667~~~~~ 138 display in 90 seconds: rate is 1.533333 That's all folks!</pre> 	-27.2
Normal Motor Run	Un-predictable indication	Average is Lost!!

Table 8.1 Simulation for Vector Space

Table 8.1 is the simulation results for the algorithm developed to detect the open phase fault. As we can see from the table 8.1, the test runs pass for all the conditions of the open phase. This is true when the open phase condition last for longer time and the time is provided to find out the theta average and make the decision.

The condition for normal run of the motor, it is observed that the vector makes a full trajectory of 360 deg. Hence it is very difficult to capture the angle of the vector. The results obtained are unsatisfactory to make decision for the motor condition while motor is running normally.

Thus, this drawback calls for the development of an efficient algorithm to detect the open phase fault. In *Chapter 9*, Hough transformation implementation is described, which is computationally efficient to detect the open phase fault.

# Chapter 9

## Hough Transformation

In this chapter, Hough transformation is introduced. The application to Open phase fault detection is illustrated. The simulation and experimental results are shown to detect the open phase fault.

### 9.1 Introduction

In the *chapters 7&8*, the methodologies to detect the open phase fault in PMSM using Xilinx Motor control solution is studied and results have been drawn. The analysis has been done using current signature to discriminate between Healthy/Fault conditions of the motor.

In Current Signature Analysis, *chapter 7* an expert advice is needed to know the condition of the motor. And, in vector space angle method, *chapter 8*, is not providing the analytically strong proof of concept.

An algorithm which could give analytical insight of the vector space trajectory under the healthy/fault condition of the motor would address the issue of efficient methodology to detect open phase fault. Hough transformation is one such possibility to detect the vector trajectory.

### 9.2 Hough Transformation

#### 9.2.1 Introduction

Hough has proposed an interesting and computationally efficient procedure for detecting lines in pictures. He proposed that use of angle-radius rather than slope-intercept parameters simplifies the computation further [19].

Rosenfeld [20] has described an ingenious method due to Hough [21] for replacing the original problem of finding colinear points by a mathematically equivalent problem of finding concurrent lines. This method involves transforming each of the figure points into a straight line in a *parameter space*. The parameter space is defined by the parametric representation used to describe lines in the picture plane. Hough chose to use the familiar slope-intercept parameters, and thus his parameter space was the two-dimensional slope-intercept plane. Unfortunately, both the slope and the intercept are unbounded, which complicates the application of the technique. Alternative parameterization that eliminates this problem is used [19].

This theoretical concept will be used to detect the lines of the vector space trajectory. More details can be found in *section 9.2.3*.

## 9.2.2 Fundamentals

The set of all straight lines in the picture plane constitutes a two-parameter family. If we fix a parameterization for the family, then an arbitrary straight line can be represented by a single point in the parameter space. For reasons that become obvious, we prefer the so-called *normal parameterization*.

As illustrated in Figure 9.1, this parameterization specifies a straight line by the angle  $\theta$  of its normal and its algebraic distance  $\rho$  from the origin O.

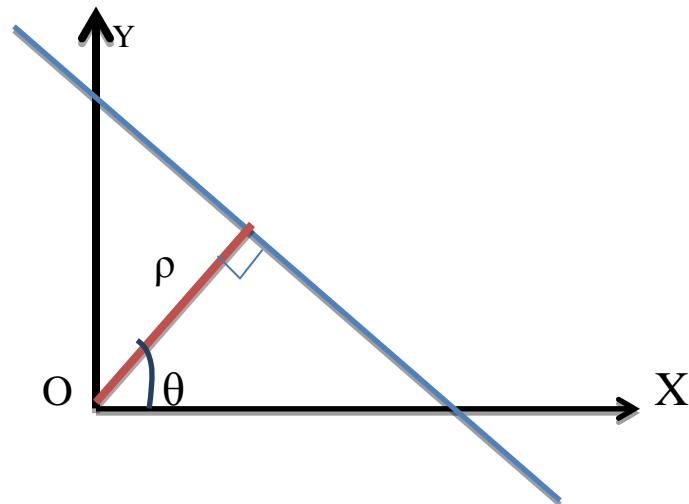


Fig 9.1 Normal parameters of the line

The equation of a line corresponding to this geometry is

$$x \cos \theta + y \sin \theta = \rho \quad \text{Eq (9.1)}$$

If we restrict  $\theta$  to the interval  $[\theta, \pi]$ , then the normal parameters for a line are unique. With this restriction, every line in the  $x$ - $y$  plane corresponds to a unique point in the  $\theta$  -  $\rho$  plane.

Suppose now, that we have some set  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  of  $n$  figure points and we want to find a set of straight lines that fit them. We transform the points  $(x_i, y_i)$  into the sinusoidal curves in the  $\theta$  -  $\rho$  plane defined by,

$$\rho = x_i \cos \theta + y_i \sin \theta \quad \text{Eq (9.2)}$$

Fig 9.2 shows the projection of colinear points in the XY plane. With this we go for transformation to  $\theta$  -  $\rho$  plane.

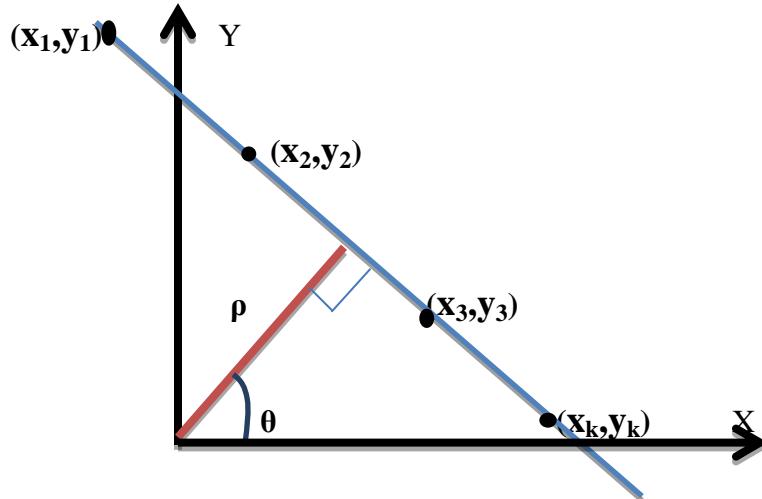


Fig 9.2 Projection of colinear points onto a line

It is easy to show that the curves corresponding to colinear figure points have a common point of intersection. This point in the  $\theta - \rho$  plane, say  $(\theta_0, \rho_0)$ , defines the line passing through the colinear points. Thus the problem of detecting colinear points can be converted to the problem of finding concurrent curves.

Fig 9.3 makes it clear how the transformation from the X-Y plane to  $\theta - \rho$  plane is achieved.

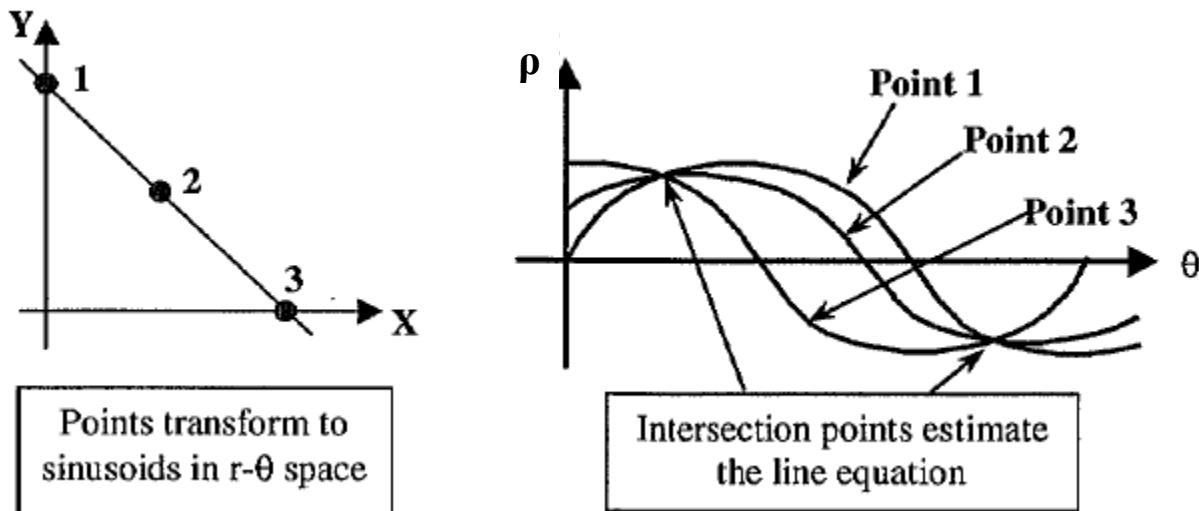


Fig 9.3 Transformation from XY to  $\rho$ -  $\theta$  plane [22]

A dual property of the point-to-curve transformation can also be established. Suppose we have a set  $\{(\theta_1, \rho_1), \dots, (\theta_k, \rho_k)\}$  of points in the  $\rho$ -  $\theta$  plane, all lying on the curve,

Then the equation in  $\rho$ -  $\theta$  can be defined by,

$$\rho = x_0 \cos \theta + y_0 \sin \theta \quad \text{Eq (9.3)}$$

Then it is easy to show that all these points correspond to lines in the  $x$ - $y$  plane passing through the point  $(x_0, y_0)$ .

**Properties:** We can summarize these interesting properties of the point-to-curve transformation as follows:

- Property 1. A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.
- Property 2. A point in the parameter plane corresponds to a straight line in the picture plane.
- Property 3. Points lying on the same straight line in the picture plane correspond to curves through a common point in the parameter plane.
- Property 4. Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

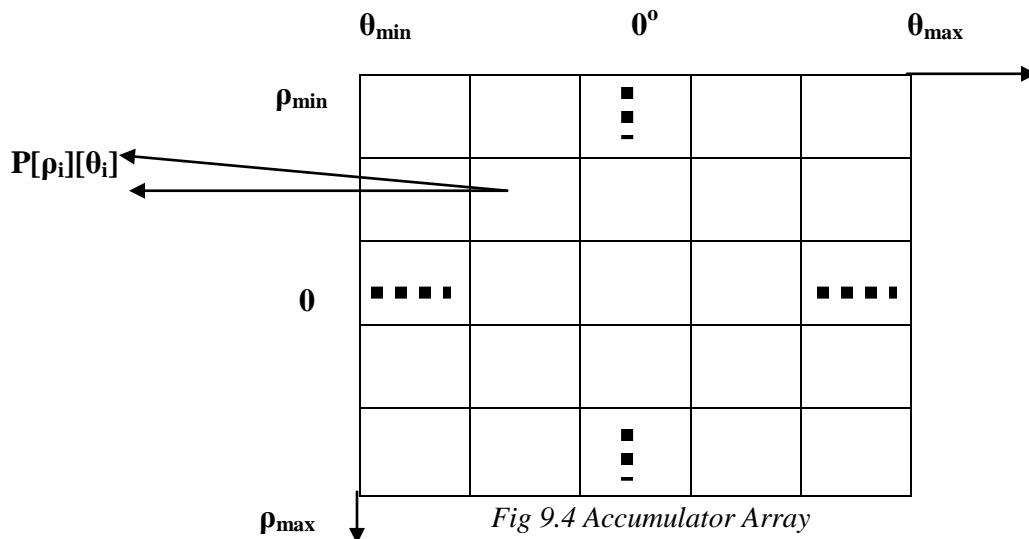
### 9.2.3 Algorithm

The Algorithm of the Hough transformation is as follows:

**Step I:** For each of the X and Y values, using the *Eq (9.1)* the  $\rho$  value is obtained. Depending upon the image, the angle  $\theta$  could be rotated.

**Step II:** Quantizing the parameter space. In order to accumulate the  $\rho$  value calculated. The accumulator array is declared with  $\rho$  and  $\theta$  indices. The size of the accumulator array depends on resolution and dimensions of the image.

Accumulator Array,  $P[\rho][\theta] = [\rho_{\min} \dots \dots \rho_{\max}][\theta_{\min} \dots \dots \theta_{\max}]$ .



**Step III:** For each edge point (x , y)

For ( $\theta = \theta_{\min}$ ;  $\theta \leq \theta_{\max}$ ;  $\theta++$ )

{

$$\rho = x \cos \theta + y \sin \theta \quad (\text{Rounding if needed})$$

**P[ρ][θ]** ++

(Incrementing the Accumulator cell)

}

Step IV: Setting up the Threshold and finding the local maxima. In the *Fig 9.4*, the entry of  $\rho$  value to the particular  $\rho$  and  $\theta$  is incremented. **Hence the accumulator cell with the maximum number of entry decides the maximum number of colinear points at particular  $\rho$  with the angle  $\theta$ .**

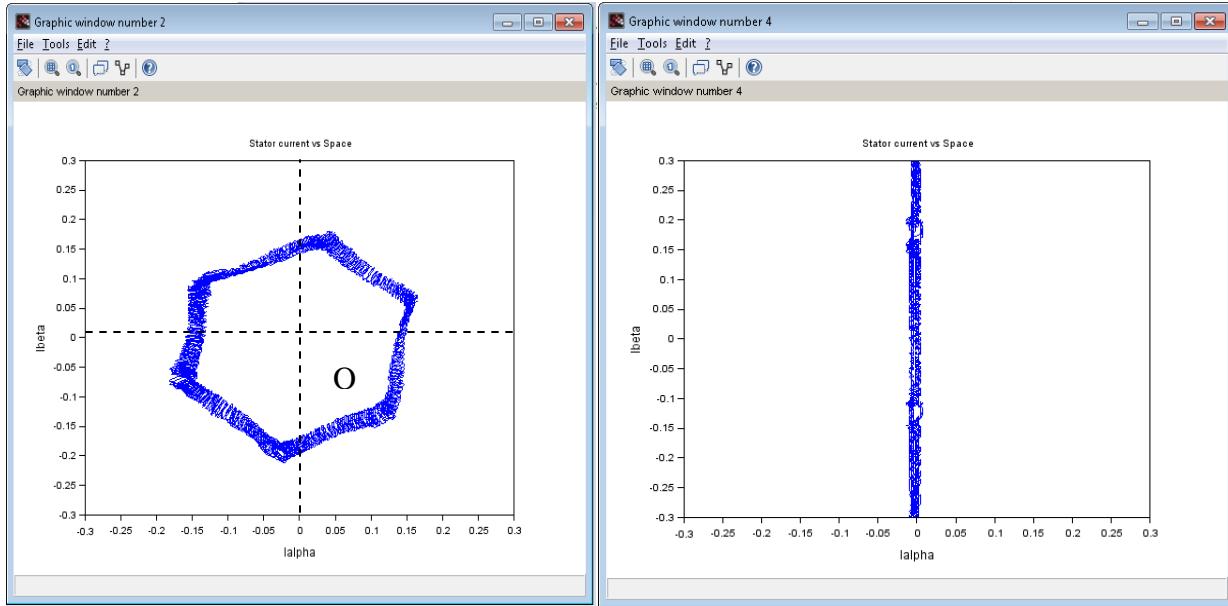
### 9.3 Perspective for open phase fault detection

With the knowledge of Hough transformation of detecting the lines and curves in an image, the transformation could be applied to detect the trajectory traced by the current vector [Chapter 7, 7.2].

Fig 9.5 shows the various trajectory traced by the current vector in  $\alpha$ - $\beta$  plane. If we observe carefully, the trajectory of the motor is varying for each of the condition. With this observation we can continue to detect the lines and angle of the vector values.

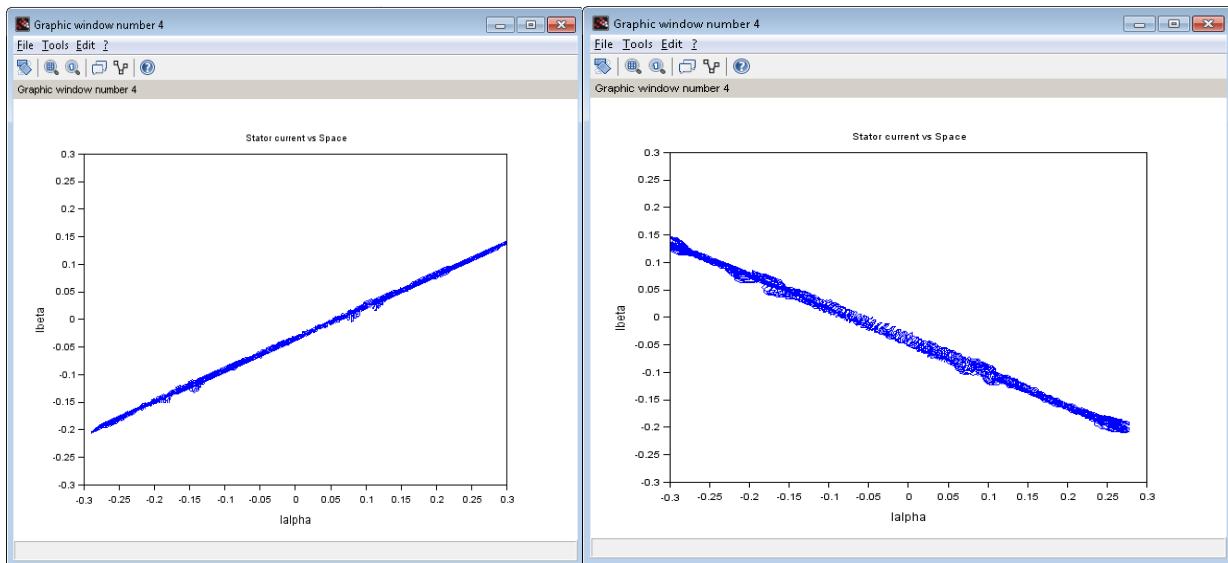
The  $\alpha\beta$  plane is considered as MxN image. With the origin as (0, 0) and using Hough transformation [19] the trajectory can be transformed into  $\rho$ -  $\theta$  plane. This should give, the vector distance  $\rho$  from the origin O, at particular angle  $\theta$ . Using the fundamentals of Hough transformation [9.2.2] the colinear points ( $I\alpha_1, I\beta_1 \dots I\alpha_i, I\beta_i$ ) can be identified. And, the maximum number of colinear points which lie on a particular  $\rho$  value and  $\theta$  is used to decide the condition of the motor. More details on the application of the algorithm is described in section 9.3.

In the Thesis, the projection of colinear points is done real time with the HiL on Xilinx AP SoC and SCILAB. The current data is collected using the SCILAB function [*chapter 5, 5.1*] and Hough transformation is implemented with this data.



a. Motor Normal condition

b. Open A Phase



c. Open B Phase

d. Open C Phase

Fig 9.5 Trajectory traced by current vector.

## 9.4 HiL Implementation

### 9.4.1 Algorithm

Following the Hough transformation algorithm described in the *section 9.2.3*, the trajectory of  $\alpha$ - $\beta$  plane for the current space vector can be traced considering the decimation points as current acquisition samples, which is 8192 samples and the  $360^\circ$  rotation.

#### Step Followed:

Depending on the resolution, the accumulator array dimension can be set, for Thesis, it is set to 32. So,  $P [32][32] = 0$  [At start all cell are 0].

**Step I:** Decimating the  $I_\alpha$  and  $I_\beta$  values of 8192 in **32 steps** which allows to take every  $8192/32 = 256$  values.

**Step II:**  $\theta$  value is also decimated in **32 steps** for  $360^\circ$ , which is  $360/32 = 11.25$ , running the loop for 360 degree for **same  $I_\alpha$  and  $I_\beta$  value**.

**Step III:** Calculating the  $\rho$  value, since the value obtained is expected to be a rational number,  $\rho$  value is normalized and rounded. This will act as  $\rho$  Index in the accumulator array.

**Step IV:** Since the  $\theta$  is rotated for  $360^\circ$ , the values expected are both positive and negative. Hence, a  $\rho$  value with the positive value and angle  $\theta$  taken in positive accumulator array, same dimension as P.

**Step V:** Similarly, the negative  $\rho$  value for particular  $\theta$  is taken in the negative accumulator array.

**Step VI:** This effort would suffice, adding the  $\rho$  value to the accumulator cell (Positive to Accumulator\_Array\_Pos and Negative to Accumulator\_Array\_Neg) with the particular  **$\rho$  and  $\theta$**  index.

**Step VII:** A threshold is set (1 and -1) to  $\rho$  enter the accumulator array.

**Step VIII:** At the end, it is expected to have a **Max  $\rho$**  value in a particular accumulator array cell corresponding  $\rho$  and  $\theta$ , where the maximum number of colinear points is placed.

*Fig 9.6* shows the flow chart of the Hough transformation implemented using SCILAB. As described in the above steps,  $I_\alpha$  and  $I_\beta$ ,  $\theta$  is decimated. While  $I_\alpha$  and  $I_\beta$  are taken in the outer loop for traversing

#### 9.4.2 Flow Chart

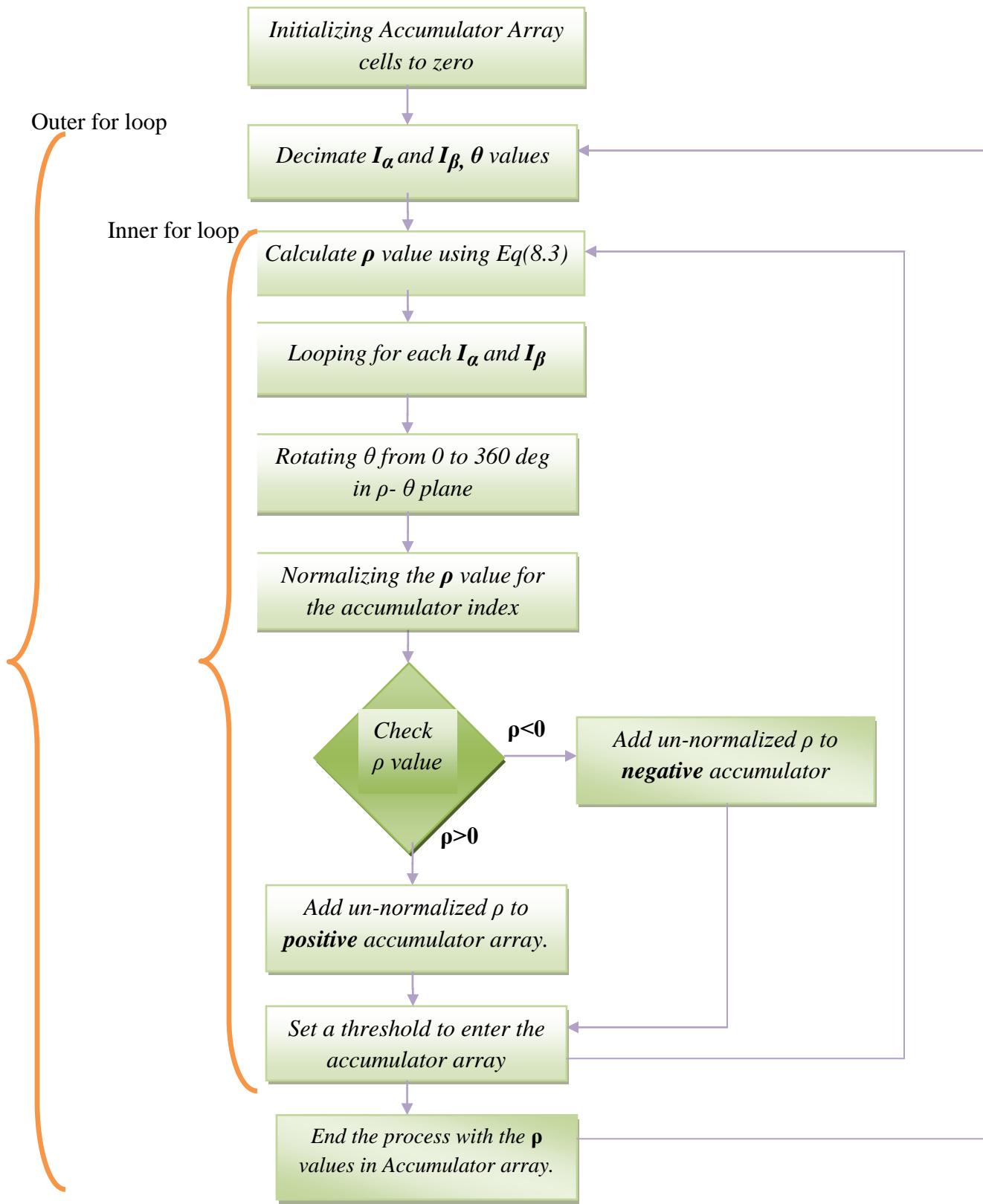


Fig 9.6 Hough Transformation flow chart

## 9.5 Results

### 9.5.1 Simulations

As explained in *section 9.4.1, step VII*, the aim is to find the maximum  $\rho$  value in an accumulator cell. The following results are obtained with Xilinx Motor control solution and SCILAB HiL.

- 1. Normal motor run:** For motor running normally, the current vector trajectory is as shown in *Fig 9.3*. It is clear that in the  $\rho$ -  $\theta$  plane, the current vector is rotating from **0 to 360 deg** with the particular  $\rho$ .

Fig 9.7 (a) & (b) shows the simulated values for both the positive accumulator array and negative accumulator array. As we can see that the  $\rho$  values are distributed from 0 to 32, as per the decimation in *section 9.4, Step II*, and  $32 * 11.25 = 360 \text{ deg}$ . This is true for both positive accumulator array (a) and negative accumulator array (b) cells which shows that the current vector is rotating with the 360 deg rotations.  $\rho$  values has magnitude until a certain range which shows that the vector is rotating with particular  $\rho$  magnitude.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.1106
3	1.496	0	1.3626	1.7597	0	0	1.5755	0	1.2078	0	0	0	0	0	0	1.1413	1.3859
4	1.7808	0	1.0431	1.0794	0	0	1.4461	0	1.3239	1.4923	0	1.3153	0	2.2219	1.9737	1.2004	0
5	2.5659	1.9282	1.9016	2.5641	0	2.0891	1.4296	1.4648	1.3377	2.1406	0	2.6908	1.6245	1.4722	2.0648	1.1979	0
6	1.1393	3.7134	3.8369	1.4076	0	1.1076	1.3311	2.8756	1.4289	2.0743	0	1.5613	3.3487	2.3023	2.8601	0	2.2619
7	1.4744	1.2692	2.8523	1.8612	3.2437	0	2.7203	3.4806	3.1383	1.7333	3.5292	0	4.5776	3.5898	3.0466	2.7097	1.271
8	2.8842	5.6274	4.5562	1.7753	2.5451	0	1.6616	3.5971	3.6745	0	2.9361	1.2343	4.588	5.4233	0	1.9967	0
9	1.4098	7.3959	2.5723	3.1414	2.2487	3.5205	1.1677	5.5284	4.3226	3.4451	2.6814	4.8664	4.9543	6.2312	2.2961	4.1359	1.9148
10	1.3236	3.8829	7.4344	4.546	1.788	3.576	3.7557	2.5592	6.6207	3.2097	1.6185	7.3537	5.4866	4.6051	5.9434	1.3776	4.586
11	11.3071	3.5641	3.6059	5.1464	3.4918	4.9285	5.7129	3.4959	4.0904	9.9501	4.6608	10.0852	5.9566	5.525	6.5962	2.2371	7.9418
12	9.4329	2.5461	1.5359	6.3386	11.2038	8.2655	9.8366	5.3968	3.7464	10.5746	19.6699	8.0256	7.975	5.2828	6.0208	12.9047	8.947
13	6.5511	6.123	5.9597	8.0666	8.1743	9.0027	5.3234	7.6847	7.8271	10.959	16.5292	9.2589	3.8798	5.9974	11.7147	13.9738	12.4344
14	5.9767	5.9269	5.6245	5.1679	7.9523	9.4822	6.1464	6.8428	9.18	8.6266	3.4216	10.0383	8.5601	9.4855	11.0236	15.4064	12.5469
15	0	3.5867	4.3228	3.7325	5.016	4.1775	7.3541	6.6248	6.7133	2.2208	0	1.4731	7.8782	7.9607	5.9771	3.6989	5.6023
16	0	0	1.7389	1.7229	0	1.5897	2.5504	3.6489	1.7371	0	0	0	0	1.4149	1.2641	0	2.3624
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.2859
3	0	1.0852	0	1.301	1.9005	1.3301	0	0	1.0952	0	1.15	0	0	1.2132	0
4	1.5037	0	1.9479	1.0901	1.8968	0	0	0	1.558	0	1.2836	0	0	1.0096	0
5	1.8644	0	2.264	1.0433	1.6835	1.0807	1.6667	2.0673	1.6288	0	1.995	1.2696	1.4446	1.8547	1.7467
6	2.7693	2.068	2.4022	2.328	1.9639	2.9899	0	1.9319	1.1147	2.9475	0	1.8411	2.5615	1.2773	2.2023
7	2.4213	3.3596	2.4805	3.4936	0	3.5947	2.4639	1.4192	0	2.7148	0	2.591	2.477	1.2705	2.1863
8	4.296	4.1481	1.8462	2.386	1.3837	1.2768	6.0855	2.6491	2.006	1.0265	2.4139	2.8649	4.7183	3.9144	0
9	3.3158	6.5526	3.8767	2.6825	4.0637	2.4546	6.0315	3.1024	2.3224	0	2.9261	2.4543	5.2888	4.0098	0
10	4.0132	7.0769	4.3566	1.3912	2.3839	6.0462	3.369	4.6225	3.5816	0	1.7976	6.4464	5.0743	4.1196	4.6141
11	6.0763	6.5791	4.7832	3.2133	3.1844	6.0451	5.0783	6.8681	2.853	2.3314	1.9943	4.2723	7.8238	5.9565	3.146
12	9.5035	4.2072	9.6254	3.3817	6.9093	4.4423	5.8386	6.8087	4.5665	3.842	3.7575	6.2614	3.6809	7.1529	13.6054
13	8.701	4.8216	6.7796	9.3633	7.5516	8.1926	5.8368	5.3065	7.0625	8.6572	9.8331	7.0602	2.3742	6.8145	14.0542
14	8.8343	9.4337	10.1046	17.5945	7.2876	5.5856	7.6693	6.1274	11.4785	13.066	14.2873	7.2898	7.3097	7.1193	3.863
15	7.1508	10.3933	11.1152	11.9584	16.5243	9.3091	5.6611	8.5678	12.4656	17.1007	12.2236	8.3945	8.5024	7.1667	0
16	6.0779	6.0986	6.8381	10.0227	13.0084	12.2262	8.1898	10.0575	10.0327	9.8131	9.2601	7.1612	5.7333	0	0
17	1.6945	3.3801	3.0335	0	2.8915	6.5607	9.8048	6.9386	5.2342	3.0286	0	2.5065	0	0	0
18	0	0	0	0	0	2.3537	2.8401	3.7325	2.5212	0	0	0	0	0	0
19	0	0	0	0	0	1.688	3.2272	3.2378	1.1143	0	0	0	0	0	0

Fig 9.7 (a) Positive Accumulator array for Normal motor runs

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	-1.1106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.2859	0
3	-1.3859	0	-1.0852	0	-1.301	-1.9005	-1.3301	0	0	-1.0952	0	-1.15	0	0	-1.2132	0	-1.496
4	0	-1.5037	0	-1.9479	-1.0901	-1.8968	0	0	0	-1.558	0	-1.2836	0	0	-1.0096	0	-1.7808
5	0	-1.8644	0	-2.264	-1.0433	-1.6835	-1.0807	-1.6667	-2.0673	-1.6288	0	-1.995	-1.2696	-1.4446	-1.8547	-1.7467	-2.5659
6	-2.2619	-2.7693	-2.068	-2.4022	-2.328	-1.9639	-2.9899	0	-1.9319	-1.1147	-2.9475	0	-1.8411	-2.5615	-1.2773	-2.2023	-1.1393
7	-1.271	-2.4213	-3.3596	-2.4805	-3.4936	0	-3.5947	-2.4639	-1.4192	0	-2.7148	0	-2.591	-2.477	-1.2705	-2.1863	-1.4744
8	0	-4.296	-4.1481	-1.8462	-2.386	-1.3837	-1.2768	-6.0855	-2.6491	-2.006	-1.0265	-2.4139	-2.8649	-4.7183	-3.9144	0	-2.8842
9	-1.9148	-3.3158	-6.5526	-3.8767	-2.6825	-4.0637	-2.4546	-6.0315	-3.1024	-2.3224	0	-2.9261	-2.4543	-5.2888	-4.0098	0	-1.4098
10	-4.586	-4.0132	-7.0769	-4.3566	-1.3912	-2.3839	-6.0462	-3.369	-4.6225	-3.5816	0	-1.7976	-6.4464	-5.0743	-4.1196	-4.6141	-1.3236
11	-7.9418	-6.0763	-6.5791	-4.7832	-3.2133	-3.1844	-6.0451	-5.0783	-6.8681	-2.853	-2.3314	-1.9943	-4.2723	-7.8238	-5.9565	-3.146	-11.3071
12	-8.947	-9.5035	-4.2072	-9.6254	-3.3817	-6.9093	-4.4423	-5.8386	-6.8087	-4.5665	-3.842	-3.7575	-6.2614	-3.6809	-7.1529	-13.6054	-9.4329
13	-12.4344	-8.701	-4.8216	-6.7796	-9.3633	-7.5516	-8.1926	-5.8368	-5.3065	-7.0625	-8.6572	-9.8331	-7.0602	-2.3742	-6.8145	-14.0542	-6.5511
14	-12.5469	-8.8343	-9.4337	-10.1046	-17.5945	-7.2876	-5.5856	-7.6693	-6.1274	-11.4785	-13.066	-14.2873	-7.2898	-7.3097	-7.1193	-3.863	-5.9767
15	-5.6823	-7.1508	-10.3933	-11.1152	-11.9584	-16.5243	-9.3091	-5.6611	-8.5678	-12.4656	-17.1007	-12.2236	-8.3945	-8.5024	-7.1667	0	0
16	-2.3624	-6.0779	-6.0986	-6.8381	-10.0227	-13.0084	-12.2262	-8.1898	-10.0575	-10.0327	-9.8131	-9.2601	-7.1612	-5.7333	0	0	0
17	0	-1.6945	-3.3801	-3.0335	0	-2.8915	-6.5607	-9.8048	-6.9386	-5.2342	-3.0286	0	-2.5065	0	0	0	0
18	0	0	0	0	0	0	-2.3537	-2.8401	-3.7325	-2.5212	0	0	0	0	0	0	
19	0	0	0	0	0	0	-1.688	-3.2272	-3.2378	-1.1143	0	0	0	0	0	0	

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	-1.496	0	-1.3626	-1.7597	0	0	-1.5755	0	-1.2078	0	0	0	0	0	0	-1.1413
4	-1.7808	0	-1.0431	-1.0794	0	0	-1.4461	0	-1.3239	-1.4923	0	-1.3153	0	-2.2219	-1.9737	-1.2004
5	-2.5659	-1.9282	-1.9016	-2.5641	0	-2.0891	-1.4296	-1.4648	-1.3377	-2.1406	0	-2.6908	-1.6245	-1.4722	-2.0648	-1.1979
6	-1.1393	-3.7134	-3.8369	-1.4076	0	-1.1076	-1.3311	-2.8756	-1.4289	-2.0743	0	-1.5613	-3.3487	-2.3023	-2.8601	0
7	-1.4744	-1.2692	-2.8523	-1.8612	-3.2437	0	-2.7203	-3.4806	-3.1383	-1.7333	-3.5292	0	-4.5776	-3.5898	-3.0466	-2.7097
8	-2.8842	-5.6274	-4.5562	-1.7753	-2.5451	0	-1.6616	-3.5971	-3.6745	0	-2.9361	-1.2343	-4.588	-5.4233	0	-1.9967
9	-1.4098	-7.3959	-2.5723	-3.1414	-2.2487	-3.5205	-1.1677	-5.5284	-4.3226	-3.4451	-2.6814	-4.8664	-4.9543	-6.2312	-2.2961	-4.1359
10	-1.3236	-3.8829	-7.4344	-4.546	-1.788	-3.576	-3.7557	-2.5592	-6.6207	-3.2097	-1.6185	-7.3537	-5.4866	-4.6051	-5.9434	-1.3776
11	-11.3071	-3.5641	-3.6059	-5.1464	-3.4918	-4.9285	-5.7129	-3.4359	-4.0904	-9.9501	-4.6608	-10.0852	-5.9566	-5.525	-6.5962	-2.2371
12	-9.4329	-2.5461	-1.5359	-6.3386	-11.2038	-8.2655	-9.8366	-5.3968	-3.7464	-10.5746	-19.6699	-8.0256	-7.975	-5.2828	-6.0208	-12.9047
13	-6.5511	-6.123	-5.9597	-8.0666	-8.1743	-9.0027	-5.3234	-7.6847	-7.8271	-10.959	-16.5292	-9.2589	-3.8798	-5.9974	-11.7147	-13.9738
14	-5.9767	-5.9269	-5.6245	-5.1679	-7.9523	-9.4822	-6.1464	-6.8428	-9.18	-8.6266	-3.4216	-10.0383	-8.5601	-9.4855	-11.0236	-15.4064
15	0	-3.5867	-4.3228	-3.7325	-5.016	-4.1775	-7.3541	-6.6248	-6.7133	-2.2208	0	-1.4731	-7.8782	-7.9607	-5.9771	-3.6989
16	0	0	-1.7389	-1.7229	0	-1.5897	-2.5504	-3.6489	-1.7371	0	0	0	-1.4149	-1.2641	0	0

Fig 9.7 (b) Negative Accumulator arrays for Normal motor runs.

**2. Open A Phase:** For motor running with phase A open, the current vector trajectory is as shown in *Fig 9.3(b)*. Relating the trajectory in  $\rho$ -  $\theta$  plane, we see that current vector has  $I_a$  and  $I_b$  arranged in particular angle in both 1<sup>st</sup> and 4<sup>th</sup> quadrant. From the Fig 9.3 (b) we could approximate that possible angle values would be 70-90 and 250 - 280 deg.

Fig 9.8 (a) & (b) shows the  $\rho$  values for the positive accumulator array cell and recorded  $\theta$  for the positive values respectively. For the positive accumulator cell (a) the maximum  $\rho$  value appears at  $\theta = 9$ , which is  $9 * 11.25 = 101.25$ .

In (b) the  $\theta$  values are recorded in an array, and we can see that for maximum times the values are **101.25**.

Fig 9.9 (a) & (b) shows the  $\rho$  values for the negative accumulator array cell and recorded  $\theta$  for the negative values respectively. For the negative accumulator cell (a) the maximum  $\rho$  value appears at  $\theta = 25$ , which is  $25 * 11.25 = 281.25$ .

In (b) the  $\theta$  values are recorded in an array and we can see that for maximum times the values are **281.25**.

If we imagine these two  $\theta$  values (**78.25 & 281.25**) plotted with the  $\rho$  value obtained, then we could see the same trajectory as *Fig 9.5 (b)* which is **trajectory for open phase A**.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
13	0	0	8.9537	3.0711	1.0233	3.504	1.4193	3.6314	4.143	3.6289	1.5463	3.0188	3.88	5.0759	9.1872	0	0
14	0	0	0	1.69	2.132	4.9183	2.2664	0	1.5215	0	1.7082	3.773	1.9673	3.7829	0	0	0
15	0	0	0	6.2841	9.8007	2.2313	4.0249	2.5829	1.668	1.9746	4.1885	4.6465	6.6416	4.1752	0	0	0
16	0	0	0	5.3765	4.769	0	5.2916	3.9862	4.1595	4.4578	4.3443	1.4532	5.7043	6.5685	0	0	0
17	0	0	0	9.0474	1.6805	5.3229	1.0122	4.603	3.4023	4.6233	3.0339	2.9228	3.8993	10.0343	0	0	0
18	0	0	0	18.4908	2.71	10.4312	0	1.9524	4.2729	2.684	1.2792	10.6298	3.7753	15.0482	0	0	0
19	0	0	0	7.3145	6.6489	3.4215	5.1673	0	0	0	3.2661	2.8416	4.924	9.2597	0	0	0
20	0	0	0	6.6117	2.1846	12.1729	3.4308	1.2106	2.8006	11.5771	3.9663	5.4206	0	0	0	0	0
21	0	0	0	0	4.6504	2.5296	2.518	12.6074	8.4379	12.8258	3.1394	3.1432	8.8623	0	0	0	0
22	0	0	0	0	13.4772	4.8441	2.832	4.37	9.596	4.1437	3.5114	5.2709	11.0519	0	0	0	0
23	0	0	0	0	16.0671	6.4218	2.5283	3.9147	3.0043	4.381	3.222	4.3654	15.3582	0	0	0	0
24	0	0	0	0	10.9564	5.9693	3.8433	1.9363	2.627	2.1731	3.8505	7.1986	11.7493	0	0	0	0
25	0	0	0	0	0	6.2699	6.2322	3.0136	3.007	3.259	5.4618	8.281	0	0	0	0	0
26	0	0	0	0	0	14.8551	5.9979	6.0045	3.3843	6.0013	5.2213	13.3132	0	0	0	0	0
27	0	0	0	0	0	16.9968	5.6464	4.0443	6.7257	3.5068	5.9051	15.3597	0	0	0	0	0
28	0	0	0	0	0	13.4212	7.0147	7.568	4.7773	7.5758	8.9603	13.1682	0	0	0	0	0
29	0	0	0	0	0	0	15.3765	3.7616	7.2329	4.3426	14.5165	1.9986	0	0	0	0	0
30	0	0	0	0	0	0	18.8841	11.1141	6.3195	10.1928	17.0679	0	0	0	0	0	0
31	0	0	0	0	0	0	13.9749	13.963	10.2123	15.1953	14.2911	0	0	0	0	0	0
32	0	0	0	0	0	0	1.585	18.8322	22.0939	17.5528	2.8489	0	0	0	0	0	0
33	0	0	0	0	0	0	0	13.8776	11.8858	14.2236	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	2.3559	11.5265	2.0168	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.2518
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
13	0	8.5173	1.9303	2.0747	4.0292	0	0	0	0	4.5468	0	0	8.4526	0	
14	0	6.2842	1.576	2.7756	3.5039	3.1188	0	0	0	3.2472	4.3251	3.9717	3.3581	4.5368	0
15	0	1.3141	5.838	6.2982	2.2305	3.4352	3.7668	2.8965	3.6168	4.6421	1.0442	6.5525	6.6437	0	0
16	0	0	4.4545	2.0667	2.725	3.6624	3.4917	4.3281	4.4643	3.5034	1.1282	0	3.6903	0	0
17	0	0	4.2559	1.01	3.951	1.3373	4.0586	3.5778	3.209	0	7.6603	1.2159	3.7622	0	0
18	0	0	6.3094	3.4592	5.7164	2.8638	1.0726	1.6072	0	2.7312	4.3002	3.7792	7.0529	0	0
19	0	0	9.0042	5.1213	1.7065	5.9355	3.0493	1.7301	2.4905	7.0652	0	5.7255	7.4065	0	0
20	0	0	6.8048	5.982	0	5.1785	4.4368	3.5985	5.2187	4.3918	0	4.9605	8.7587	0	0
21	0	0	3.9522	3.1592	2.3384	1.8849	5.6205	7.5452	6.2613	0	3.7678	3.5556	0	0	0
22	0	0	0	5.0714	4.8148	0	2.6054	3.2848	2.1882	0	4.8449	5.0873	0	0	0
23	0	0	0	5.9871	7.3598	1.6193	0	1.1484	0	3.2362	7.5882	5.7559	0	0	0
24	0	0	0	7.2377	3.1077	5.7727	0	0	0	4.0865	3.1367	9.3631	0	0	0
25	0	0	0	9.7278	4.2458	5.0256	3.7656	3.0153	4.2462	6.0191	3.748	9.0243	0	0	0
26	0	0	0	6.4933	5.2074	6.2301	4.6716	4.6847	5.475	5.4227	6.0235	4.9128	0	0	0
27	0	0	0	1.5999	6.4825	2.4291	7.8325	5.6921	7.0314	3.7805	4.5948	0	0	0	0
28	0	0	0	0	5.3291	3.9023	4.1933	6.4286	3.6176	3.3588	10.3952	0	0	0	0
29	0	0	0	0	11.5564	5.2072	3.4881	3.4866	4.6485	6.4156	6.6934	0	0	0	0
30	0	0	0	0	7.8023	7.1895	3.9119	3.5874	2.4087	5.1094	9.2727	0	0	0	0
31	0	0	0	0	5.258	5.593	4.6498	4.3489	7.1452	8.7227	2.7691	0	0	0	0
32	0	0	0	0	0	10.5243	7.0147	7.3752	5.1206	7.6729	0	0	0	0	0
33	0	0	0	0	0	9.2468	6.6251	4.628	8.6078	10.5609	0	0	0	0	0
34	0	0	0	0	0	6.7882	9.1291	10.5702	7.4487	5.427	0	0	0	0	0
35	0	0	0	0	0	1.7334	10.8388	6.2987	10.8361	0	0	0	0	0	0
36	0	0	0	0	0	0	6.1031	11.1465	5.7464	0	0	0	0	0	0
37	0	0	0	0	0	0	2.5663	5.5525	1.8287	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	1.2001	0	0	0	0

Fig 9.8 (a) Positive accumulator cell for Open A Phase (0 to 360 deg).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
7	0	-7.722	-3.4427	-1.6215	-3.268	0	-1.0367	-1.0481	-1.1276	-1.1838	-1.121	0	-1.2653	-2.6132	-1.4074	-4.6719	0
8	0	0	-3.7359	-1.1385	-1.4937	-2.1466	0	0	0	0	0	-1.1288	-3.8125	-1.7371	-5.0912	0	0
9	0	0	-2.0469	-3.0457	-1.336	-3.4844	-1.5387	0	0	0	0	-3.6813	0	-4.7771	-1.8364	0	0
10	0	0	-3.7652	-3.403	0	-1.3095	-4.6533	-2.8331	-1.3018	-1.9273	-4.0143	-2.1692	0	0	-4.0273	0	0
11	0	0	-4.271	-2.4006	-3.5425	0	-1.3266	-3.5792	-5.0191	-4.4712	-2.5943	0	-4.5151	-4.4828	-5.0131	0	0
12	0	0	-5.037	-5.2416	-3.2595	0	0	-1.5622	-1.5533	-1.4201	0	-1.104	-3.9362	-4.3831	-4.8375	0	0
13	0	0	-8.5173	-1.9303	-2.0747	-4.0292	0	0	0	0	-4.5468	0	0	-8.4526	0	0	0
14	0	0	-6.2842	-1.576	-2.7756	-3.5039	-3.1188	0	0	0	-3.2472	-4.3251	-3.9717	-3.3581	-4.5368	0	0
15	0	0	-1.3141	5.838	-6.2982	-2.2305	-3.4352	-3.7668	-2.8965	-3.6168	-4.6421	-1.0442	-6.5525	-6.6437	0	0	0
16	0	0	0	-4.4545	-2.0667	-2.725	-3.6624	-3.4917	-4.3281	-4.4643	-3.5034	-1.1282	0	-3.6903	0	0	0
17	0	0	0	-4.2559	-1.01	-3.951	-1.3373	-4.0586	-3.5778	-3.209	0	-7.6603	-1.2159	-3.7622	0	0	0
18	0	0	0	-6.3094	-3.4592	-5.7164	-2.8638	-1.0726	-1.6072	0	-2.7312	-4.3002	-3.7792	-7.0529	0	0	0
19	0	0	0	-9.0042	-5.1213	-1.7065	-5.9355	-3.0493	-1.7301	-2.4905	-7.0652	0	-5.7255	-7.4065	0	0	0
20	0	0	0	-6.8048	-5.982	0	-5.1785	-4.4368	-3.5985	-5.2187	-4.3918	0	-4.9605	-8.7587	0	0	0
21	0	0	0	-3.9522	-3.1592	-2.3384	-1.8849	-5.6205	-7.5452	-6.2613	0	-3.7678	-3.5556	0	0	0	0
22	0	0	0	0	-5.0714	-4.8148	0	-2.6054	-3.2848	-2.1882	0	-4.8449	-5.0873	0	0	0	0
23	0	0	0	0	-5.9871	-7.3598	-1.6193	0	-1.1484	0	-3.2362	-7.5882	-5.7559	0	0	0	0
24	0	0	0	0	-7.2377	-3.1077	-5.7727	0	0	0	-4.0865	-3.1367	-9.3631	0	0	0	0
25	0	0	0	0	-9.7278	-4.2458	-5.0256	-3.7656	-3.0153	-4.2462	-6.0191	-3.748	-9.0243	0	0	0	0
26	0	0	0	0	-6.4933	-5.2074	-6.2301	-4.6716	-4.6847	-5.475	-5.4227	-6.0235	-4.9128	0	0	0	0
27	0	0	0	0	-1.5999	-6.4825	-2.4291	-7.8325	-5.6921	-7.0314	-3.7805	-4.5948	0	0	0	0	0
28	0	0	0	0	0	-5.3291	-3.9023	-4.1933	-6.4286	-3.6176	-3.3588	-10.3952	0	0	0	0	0
29	0	0	0	0	0	-11.5564	-5.2072	-3.4881	-3.4866	-4.6485	-6.4156	-6.6934	0	0	0	0	0
30	0	0	0	0	0	-7.8023	-7.1895	-3.9119	-3.5874	-2.4087	-5.1094	-9.2727	0	0	0	0	0
31	0	0	0	0	0	-5.258	-5.593	-4.6498	-4.3489	-7.1452	-8.7277	-2.7691	0	0	0	0	0
32	0	0	0	0	0	0	-10.5243	-7.0147	-7.3752	-5.1206	-7.6729	0	0	0	0	0	
33	0	0	0	0	0	0	-8.2468	-6.6251	-4.628	-8.6078	-10.5609	0	0	0	0	0	
34	0	0	0	0	0	0	-6.7882	-9.1291	-10.5702	-7.4487	-5.427	0	0	0	0	0	
35	0	0	0	0	0	0	-1.7334	-10.8388	-6.2987	-10.8361	0	0	0	0	0	0	
36	0	0	0	0	0	0	-6.1031	-11.1465	-5.7464	0	0	0	0	0	0	0	
37	0	0	0	0	0	0	-2.5663	-5.5525	-1.8287	0	0	0	0	0	0	0	
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
7	-3.6542	-2.1946	-3.2135	0	0	-1.6042	-1.5568	-1.2133	-1.1947	-1.2586	-1.0517	-1.1136	-3.3396	-4.9881	-3.9946
8	0	-5.214	-2.7128	-1.4416	0	-1.2852	-1.1968	-1.357	-1.4309	-1.1069	-1.1056	-1.7969	-1.7397	-3.2257	0
9	0	-3.5087	-3.9846	-3.6709	-1.0909	0	0	-1.0848	-1.0865	-1.2528	0	-3.3927	-3.1641	-5.8646	0
10	0	-3.9525	-1.9612	-1.8754	-2.8255	0	0	0	0	0	-3.1918	-1.8558	-4.509	-3.7643	0
11	0	-5.4172	-2.8973	-3.2963	-3.3747	-2.8824	-1.5538	-1.3501	-1.3349	-3.306	-3.7052	-2.6756	-2.5167	-7.0794	0
12	0	-14.5991	-8.176	-4.9128	-1.4212	-3.9519	-3.6206	-2.8965	-3.8429	-3.8375	0	-3.8424	-6.2646	-12.6429	0
13	0	-8.9537	-3.0711	-1.0233	-3.504	-1.4193	-3.6314	-4.143	-3.6289	-1.5463	-3.0188	-3.88	-5.0759	-9.1872	0
14	0	-1.69	-2.132	-4.9183	-2.2664	0	-1.5215	0	-1.7082	-3.773	-1.9673	-3.7829	0	0	0
15	0	0	-6.2841	-9.8007	-2.2313	-4.0249	-2.5829	-1.668	-1.9746	-4.1885	-4.6465	-6.6416	-4.1752	0	0
16	0	0	-5.3765	-4.769	0	-5.2916	-3.9862	-4.1595	-4.4578	-4.3443	-1.4532	-5.7043	-6.5685	0	0
17	0	0	-9.0474	-1.6805	-5.3229	-1.0122	-4.603	-3.4023	-4.6233	-3.0339	-2.9228	-3.8993	-10.0343	0	0
18	0	0	-18.4908	-2.71	-10.4312	0	-1.9524	-4.2729	-2.684	-1.2792	-10.6298	-3.7753	-15.0482	0	0
19	0	0	-7.3145	-6.6489	-3.4215	-5.1673	0	0	0	-3.2661	-2.8416	-4.924	-9.2597	0	0
20	0	0	0	-6.6117	-2.1946	-12.1729	-3.4308	-1.2106	-2.8006	-11.5771	-3.9863	-5.4206	0	0	0
21	0	0	0	-4.6504	-2.5296	-2.518	-12.6074	-8.4379	-12.8258	-3.1394	-3.1432	-8.8623	0	0	0
22	0	0	0	-13.4772	-4.8441	-2.832	-4.37	-9.596	-4.1437	-3.5114	-5.2709	-11.0519	0	0	0
23	0	0	0	-16.0671	-6.4218	-2.5283	-3.9147	-3.0043	-4.381	-3.222	-4.3654	-15.3582	0	0	0
24	0	0	0	-10.9564	-5.9693	-3.8433	-1.9363	-2.627	-2.1731	-3.8505	-7.1986	-11.7493	0	0	0
25	0	0	0	0	-6.2699	-6.2322	-3.0136	-3.007	-3.259	-5.4818	-8.281	0	0	0	
26	0	0	0	0	-14.8551	-5.9979	-6.0045	-3.3043	-6.0013	-5.2213	-13.3132	0	0	0	
27	0	0	0	0	-16.9968	-5.6464	-4.0443	-6.7257	-3.5068	-5.9051	-15.3597	0	0	0	
28	0	0	0	0	-13.4212	-7.0147	-7.568	-4.7773	-7.5758	-8.9603	-13.1682	0	0	0	
29	0	0	0	0	0	-15.3765	-3.7616	-7.2329	-4.3426	-14.5165	-1.9986	0	0	0	
30	0	0	0	0	0	-18.8841	-11.1141	-6.3195	-10.1928	-17.0679	0	0	0	0	
31	0	0	0	0	0	-13.9749	-13.963	-10.2123	-15.1953	-14.2911	0	0	0	0	
32	0	0	0	0	0	-1.585	-18.8322	-22.0939	-17.5528	-2.8489	0	0	0	0	
33	0	0	0	0	0	0	-13.8776	-11.8858	-14.2236	0	0	0	0	0	
34	0	0	0	0	0	0	-2.3559	-11.5265	-2.0168	0	0	0	0	0	

Fig 9.8 (b) Negative accumulator cell for Open A Phase (0 to 360 deg).

3	90	7	258.75
4	90	8	270
5	78.75	9	303.75
6	78.75	10	281.25
7	78.75	11	281.25
8	90	12	281.25
9	123.75	13	281.25
10	101.25	14	281.25
11	101.25	15	281.25
12	101.25	16	281.25
13	101.25	17	281.25
14	101.25	18	281.25
15	101.25	19	281.25
16	101.25	20	281.25
17	101.25	21	281.25
18	101.25	22	281.25
19	101.25	23	281.25
20	101.25	24	281.25
21	101.25	25	281.25
22	101.25	26	281.25
23	101.25	27	281.25
24	101.25	28	281.25
25	101.25	29	281.25
26	101.25	30	281.25
27	101.25	31	281.25
28	101.25	32	281.25
29	101.25	33	281.25
30	101.25	34	281.25
31	101.25	35	281.25
32	101.25	36	281.25
33	101.25	37	281.25
34	101.25	38	281.25

Fig 9.9 Recorded  $\theta$  values

3. **Open B Phase:** Similar to open A phase methodology, Fig 9.10 (a) & (b) shows the maximum  $\rho$  value for the positive accumulator array with the particular  $\theta$  value.

As explained in 2, the  $\theta$  values which are obtained while the B phase is open are as follows,

In Fig 9.10 (a), we can find that the maximum  $\rho$  value obtained as result of Hough transformation is at  $\theta = 3$ , which as per the decimation is  $3 * 11.25 = 33.75$  for positive accumulator array and  $19 * 11.25 = 213.75$  for negative accumulator array.

In the Fig 9.10 (c), we can see that maximum  $\theta$  value is **33.75** and **213.75**, now if these angles are plotted with the  $\rho$  value, the trajectory of current vector is similar to Fig 9.5 (c), which is the **Open B Phase trajectory**.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	0	0	0	1.1529	0	0	1.1699	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1.5038	0	0	1.2525	0	0	0	0
3	0	0	0	0	0	0	1.4851	0	1.7813	2.1819	0	0	2.135	2.1611	0	0	1.0795
4	0	0	0	0	0	1.7729	0	1.2801	1.4662	1.9175	0	0	3.1322	3.1068	0	1.2044	0
5	0	0	0	1.0713	1.9961	1.1839	0	2.3302	2.6118	1.8766	0	0	4.7132	1.6646	2.3374	0	0
6	1.1345	0	1.3535	2.498	1.5942	0	2.21	1.7651	3.1646	3.4405	0	0	0	1.6514	3.1381	0	1.2422
7	3.0757	3.1948	3.2174	1.3633	0	1.4106	2.4796	2.1306	3.0913	5.3928	0	0	0	4.6676	3.9112	2.2554	0
8	1.9627	1.4792	0	0	1.6867	3.8394	1.3608	3.6672	2.2981	0	0	0	2.2195	2.4245	3.8596	1.1274	
9	1.5486	1.3458	1.3673	2.166	3.8991	1.5317	1.7878	2.6305	2.048	0	0	0	6.3936	1.8873	2.7684	2.5324	
10	4.4855	5.0679	4.4712	4.4876	1.9782	1.5923	4.2165	4.072	2.4239	0	0	0	2.6037	1.7911	6.2196	4.3203	
11	2.5012	2.2925	2.6228	1.8593	1.7596	2.0692	2.8132	3.605	4.4898	0	0	0	0	7.544	6.6486	1.5364	3.7196
12	2.1733	1.9345	2.0566	1.6753	2.4065	4.6926	3.7273	2.2483	8.6549	0	0	0	0	2.8201	2.6142	2.6403	5.4501
13	2.8579	2.0939	1.967	2.86	3.0426	3.5836	5.1929	2.716	3.85	0	0	0	0	0	6.8398	2.7251	3.99
14	4.5238	1.8105	1.8084	2.9912	5.8824	3.3547	3.7606	1.5542	0	0	0	0	0	4.4232	3.6747	2.2427	
15	4.9091	5.9974	5.2606	5.7058	2.2472	5.5386	2.9776	5.0981	0	0	0	0	0	2.6871	6.9973	2.8172	
16	2.3911	4.1223	4.5923	2.226	3.5308	4.3063	2.0799	4.4538	0	0	0	0	0	0	7.8286	2.2179	1.9053
17	6.4865	2.3809	2.5559	3.2225	5.4273	3.2196	1.8439	10.5318	0	0	0	0	0	0	6.7792	6.5035	4.6267
18	5.6009	5.5768	5.2488	5.7473	4.3053	2.3241	2.7434	5.5822	0	0	0	0	0	0	1.4148	5.6892	7.0155
19	3.4299	5.4929	5.5188	4.9175	3.6084	1.896	4.1981	0	0	0	0	0	0	0	2.8447	2.6592	
20	2.7909	4.3856	4.2094	3.7972	2.3911	2.1695	4.5712	0	0	0	0	0	0	0	5.4524	3.8249	
21	2.1039	3.5692	2.949	2.9421	2.1055	1.2779	10.9707	0	0	0	0	0	0	0	6.9443	9.6316	
22	2.1805	1.5459	1.9727	1.5478	2.4097	4.5958	6.1535	0	0	0	0	0	0	0	6.5905	3.5353	
23	2.5731	2.9841	2.2977	2.9825	0	5.953	4.3203	0	0	0	0	0	0	0	2.7297	2.0728	
24	4.5764	0	2.1394	0	5.5174	5.5351	0	0	0	0	0	0	0	0	0	6.2735	
25	6.4937	4.2579	1.2689	4.0261	6.2758	10.9338	0	0	0	0	0	0	0	0	0	7.5061	
26	8.0901	5.723	5.7214	4.688	4.1666	7.8009	0	0	0	0	0	0	0	0	0	7.0346	
27	9.1194	6.1893	6.2191	5.6458	10.2795	1.8667	0	0	0	0	0	0	0	0	0	3.7674	
28	10.3544	9.8338	4.7621	9.0137	8.9361	0	0	0	0	0	0	0	0	0	0	0	
29	0	9.8401	12.4878	9.8609	6.6756	0	0	0	0	0	0	0	0	0	0	0	
30	0	8.1216	7.1889	6.2968	0	0	0	0	0	0	0	0	0	0	0	0	
31	0	0	7.4325	5.2244	0	0	0	0	0	0	0	0	0	0	0	0	
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
1	0	0	0	0	0	0	0	0	1.9483	2.6816	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	2.0633	5.9341	1.2111	0	0	0	0		
3	0	0	0	0	0	0	0	0	1.5872	3.9618	8.3689	1.7976	1.3202	1.0921	0		
4	1.3666	0	0	0	0	0	0	1.0572	2.1127	5.8829	10.716	1.7102	0	0	0		
5	1.282	1.7113	1.2129	0	1.0916	1.0467	1.2496	2.0203	3.0711	8.1173	8.1946	5.4297	1.1495	0	0		
6	1.2336	1.246	1.9087	1.929	1.9342	1.9196	2.5646	2.6809	2.766	9.4057	0	3.822	4.3039	2.2746	0		
7	1.0988	1.4926	0	1.4382	1.7239	1.9154	1.9615	1.8144	6.0741	7.6466	0	6.4977	4.3948	3.1932	3.5847		
8	0	1.4269	2.071	1.6911	1.9539	2.3277	1.9001	2.685	5.9738	0	0	6.2159	1.9779	4.4319	1.4999		
9	1.2561	0	1.0711	1.5196	1.2501	1.0757	1.5388	5.7805	4.4149	0	0	3.26	6.0009	2.2498	4.7268		
10	2.2078	1.4942	0	0	1.3142	1.5014	4.008	5.2001	5.3586	0	0	10.2703	4.399	1.8905	2.1852		
11	5.312	2.3198	1.6489	1.5416	1.6799	3.4243	6.1552	5.9353	8.4778	0	0	4.7353	7.0323	4.5355	1.7416		
12	3.604	4.5733	3.0231	2.7814	3.6269	5.9075	4.327	2.8785	8.5394	0	0	0	3.8125	5.2045	2.8777		
13	3.2424	4.6666	5.8475	5.4925	6.2618	4.1411	5.5545	5.5124	6.3884	0	0	0	2.5914	5.374	6.415		
14	5.9	3.0688	4.4719	4.7721	3.6413	5.7524	4.1691	5.0301	1.7811	0	0	0	4.7934	6.3006	3.8941		
15	4.0002	5.7318	3.4472	4.3448	5.4203	5.5337	2.2323	9.4357	0	0	0	0	9.6158	4.0544	3.9206		
16	2.2445	5.4197	6.577	6.4189	5.5869	3.3613	5.3152	5.4392	0	0	0	0	9.3955	2.5592	6.5541		
17	2.6958	2.0346	4.0502	4.2364	3.7212	2.377	5.7762	5.7764	0	0	0	0	0	1.6739	4.0692		
18	1.9709	2.8731	2.162	2.3427	2.8793	2.7172	6.6784	7.3296	0	0	0	0	0	4.6734	3.4061		
19	3.4502	1.9118	2.6427	2.4536	1.714	6.6307	7.6063	7.1707	0	0	0	0	0	7.1966	2.0957		
20	7.2147	1.6078	2.0027	2.2025	4.034	6.1891	5.3921	0	0	0	0	0	0	11.1762	1.5734		
21	4.1763	5.8885	2.3257	4.027	5.6602	5.2764	4.4189	0	0	0	0	0	0	7.5488	3.795		
22	2.642	7.0183	7.0388	5.7243	5.9014	7.2676	8.3622	0	0	0	0	0	0	0	6.3885		
23	7.8426	3.4413	6.4183	6.179	4.1275	5.2599	6.6944	0	0	0	0	0	0	0	6.4263		
24	6.9176	4.5929	2.8746	2.6297	7.9072	4.3101	5.502	0	0	0	0	0	0	0	11.4423		
25	3.0058	9.0353	5.7707	7.4957	6.7078	6.022	0	0	0	0	0	0	0	0	8.7405		
26	2.589	4.1302	9.3688	8.5619	3.6387	6.7304	0	0	0	0	0	0	0	0	0		
27	7.0267	3.502	3.2291	3.7974	3.4943	8.1063	0	0	0	0	0	0	0	0	0		
28	8.4063	2.4982	4.198	3.0757	7.8121	5.0276	0	0	0	0	0	0	0	0	0		
29	7.2885	8.6995	3.203	6.405	7.8024	0	0	0	0	0	0	0	0	0	0		
30	4.5025	7.2004	8.382	7.7996	7.2088	0	0	0	0	0	0	0	0	0	0		
31	0	7.7696	7.4277	5.8977	4.3222	0	0	0	0	0	0	0	0	0	0		
32	0	5.1171	8.0253	7.3378	0	0	0	0	0	0	0	0	0	0	0		

Fig 9.10 (a) Positive Accumulator array for Open B Phase (0 to 360 deg)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	0	0	0	0	-1.9483	-2.6816	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	-2.0633	-5.9341	-1.2111	0	0	0	0	
3	-1.0795	0	0	0	0	0	0	0	-1.5872	-2.1127	-5.8829	-10.716	-1.7102	-1.3202	-1.0921	0	
4	0	-1.3666	0	0	0	0	0	0	-1.0572	-2.1127	-5.8829	-10.716	-1.7102	0	0	0	
5	0	-1.282	-1.7113	-1.2129	0	-1.0916	-1.0467	-1.2496	-2.0203	-3.0711	-8.1173	-8.1946	-5.4297	-1.1495	0	0	
6	-1.2422	-1.2336	-1.246	-1.9087	-1.929	-1.9342	-1.9196	-2.5646	-2.6809	-2.766	-9.4057	0	-3.822	-4.3039	-2.2746	0	
7	0	-1.0988	-1.4926	0	-1.4382	-1.7239	-1.9154	-1.9615	-1.8144	-6.0741	-7.6466	0	-6.4977	-4.3948	-3.1932	-3.5847	
8	-1.1274	0	-1.4269	-2.071	-1.6911	-1.9539	-2.2377	-1.9001	-2.685	-5.9738	0	0	-6.2159	-1.9779	-4.4319	-1.4999	
9	-2.5324	-1.2561	0	-1.0711	-1.5196	-1.2501	-1.0757	-1.5388	-5.7805	-4.4149	0	0	-3.26	-6.0009	-2.2498	-4.7268	
10	-4.3203	-2.2078	-1.4942	0	0	-1.3142	-1.5014	-4.008	-5.2001	-5.3586	0	0	-10.2703	-4.399	-1.8905	-2.1852	
11	-3.7196	-5.312	-2.3198	-1.6489	-1.5416	-1.6799	-3.4243	-6.1552	-5.9353	-8.478	0	0	-4.7353	-7.0323	-4.5355	-1.7416	
12	-5.4501	-3.604	-4.5733	-3.0231	-2.7814	-3.6269	-5.9075	-4.327	-2.8785	-8.5394	0	0	-3.8125	-5.2045	-2.8777	-2.1733	
13	-3.99	-3.2424	-4.6666	-5.8475	-5.4925	-6.2618	-4.1411	-5.5545	-5.5124	-6.3884	0	0	-2.5914	-5.374	-6.415	-2.8579	
14	-2.2427	-5.9	-3.0688	-4.4719	-4.7721	-3.6413	-5.7524	-4.1691	-5.0301	-1.7811	0	0	-4.7934	-6.3006	-3.8941	-4.5238	
15	-2.8172	-4.0002	-5.7318	-3.4472	-4.3448	-5.4203	-5.537	-2.2323	-9.4357	0	0	0	-9.6158	-4.0544	-3.9206	-4.9091	
16	-1.9053	-2.2445	-5.4197	-6.577	-6.4189	-5.5869	-3.3613	-5.3152	-5.4392	0	0	0	-9.3955	-2.5592	-6.5541	-2.3911	
17	-4.6267	-2.6958	-2.0346	-4.0502	-4.2364	-3.7212	-2.377	-5.7762	-5.7764	0	0	0	0	-1.6739	-4.0692	-6.4865	
18	-7.0155	-1.9709	-2.8731	-2.162	-2.3427	-2.8793	-2.7172	-6.6784	-7.3296	0	0	0	0	-4.6734	-3.4061	-5.6009	
19	-2.6592	-3.4502	-1.9118	-2.6427	-2.4536	-1.714	-6.6307	-7.6063	-7.1707	0	0	0	0	-7.1966	-2.0957	-3.4299	
20	-3.8249	-7.2147	-1.6078	-2.0027	-2.2025	-4.034	-6.1891	-5.3921	0	0	0	0	0	-11.1762	-1.5734	-2.7909	
21	-9.6316	-4.1763	-5.8885	-2.3257	-4.027	-5.6602	-5.2764	-4.4189	0	0	0	0	0	-7.5488	-3.795	-2.1039	
22	-3.5353	-2.642	-7.0183	-7.0388	-5.7243	-5.9014	-7.2676	-8.3622	0	0	0	0	0	-6.3885	-2.1805		
23	-2.0728	-7.8426	-3.4413	-6.4183	-6.179	-4.1275	-5.2599	-6.6944	0	0	0	0	0	-6.4263	-2.5731		
24	-6.2735	-6.9176	-4.5929	-2.8746	-2.6297	-7.9072	-4.3101	-5.502	0	0	0	0	0	-11.4423	-4.5764		
25	-7.5061	-3.0058	-9.0353	-5.7707	-7.4957	-6.7078	-6.022	0	0	0	0	0	0	-8.7405	-6.4937		
26	-7.0346	-2.589	-4.1302	-9.3688	-8.5619	-3.6387	-6.7304	0	0	0	0	0	0	0	-8.0901		
27	-3.7674	-7.0267	-3.502	-3.2291	-3.7974	-3.4943	-8.1063	0	0	0	0	0	0	0	-9.1194		
28	0	-8.4063	-2.4982	-4.198	-3.0757	-7.8121	-5.0276	0	0	0	0	0	0	0	0	-10.3544	
29	0	-7.2885	-8.6995	-3.203	-6.405	-7.8024	0	0	0	0	0	0	0	0	0	0	
30	0	-4.5025	-7.2004	-8.382	-7.7996	-7.2088	0	0	0	0	0	0	0	0	0	0	
31	0	0	-7.7696	-7.4277	-5.8977	-4.3222	0	0	0	0	0	0	0	0	0	0	
32	0	0	-5.1171	-8.0253	-7.3378	0	0	0	0	0	0	0	0	0	0	0	
33	0	0	0	-4.6219	-2.9456	0	0	0	0	0	0	0	0	0	0	0	
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
1	0	0	0	0	0	0	0	0	-1.1529	0	-1.1699	0	0	0	0		
2	0	0	0	0	0	0	0	0	-1.5038	0	0	-1.2525	0	0	0		
3	0	0	0	0	0	-1.4851	0	-1.7813	-2.1819	0	0	-2.135	-2.1611	0	0		
4	0	0	0	0	-1.7729	0	-1.2801	-1.4662	-1.9175	0	0	-3.1322	-3.1068	0	-1.2044		
5	0	0	-1.0713	-1.9961	-1.1839	0	-2.3302	-2.6118	-1.8766	0	0	-4.7132	-1.1646	-2.3374	0		
6	0	-1.3535	-2.498	-1.5942	0	-2.21	-1.7651	-3.1646	-3.4405	0	0	0	-1.6514	-3.1381	0		
7	-3.1948	-3.2174	-1.3633	0	-1.4106	-2.4796	-2.1306	-3.0913	-5.3928	0	0	0	-4.6676	-3.9112	-2.2554		
8	-1.4792	0	0	-1.6867	-3.8394	-1.3608	-3.6672	-2.2981	0	0	0	0	-2.2195	-2.4245	-3.8596		
9	-1.3458	-1.3673	-2.166	-3.8991	-1.5317	-1.7878	-2.6305	-2.048	0	0	0	0	-6.3936	-1.8873	-2.7684		
10	-5.0679	-4.4712	-4.4876	-1.9782	-1.5923	-4.2165	-4.072	-2.4239	0	0	0	0	-2.6037	-1.7911	-6.2196		
11	-2.2925	-2.6228	-1.8593	-1.7596	-2.0692	-2.8132	-3.605	-4.4989	0	0	0	0	-7.544	-6.6466	-1.5364		
12	-1.9345	-2.0566	-1.6753	-2.4065	-4.6926	-3.7273	-2.2483	-8.6549	0	0	0	0	-2.8201	-2.6142	-2.6403		
13	-2.0939	-1.967	-2.86	-3.0426	-3.5836	-5.1929	-2.716	-3.85	0	0	0	0	0	-6.8398	-2.7251		
14	-1.8105	-1.8084	-2.9912	-5.8824	-3.3547	-3.7606	-1.5542	0	0	0	0	0	0	-4.4232	-3.6747		
15	-5.9974	-5.2606	-5.7058	-2.2472	-5.5386	-2.9776	-5.0981	0	0	0	0	0	0	-2.6871	-6.9973		
16	-4.1223	-4.5923	-2.226	-3.5308	-4.3063	-2.0799	-4.4538	0	0	0	0	0	0	-7.8286	-2.2179		
17	-2.3809	-2.5559	-3.2225	-5.4273	-3.2196	-1.8439	-10.5318	0	0	0	0	0	0	-6.7792	-6.5035		
18	-5.5768	-5.2488	-5.7473	-4.3053	-2.3241	-2.7434	-5.5822	0	0	0	0	0	0	-1.4148	-5.6892		
19	-5.4929	-5.5188	-4.9175	-3.6084	-1.896	-4.1981	0	0	0	0	0	0	0	0	-2.8447		
20	-4.3856	-4.2094	-3.7972	-2.3911	-2.1695	-4.5712	0	0	0	0	0	0	0	0	-5.4524		
21	-3.5692	-2.949	-2.9421	-2.1055	-1.2779	-10.9707	0	0	0	0	0	0	0	0	-6.9443		
22	-1.5459	-1.9727	-1.5478	-2.4097	-4.5958	-6.1535	0	0	0	0	0	0	0	0	-6.5905		
23	-2.9841	-2.2977	-2.9825	0	-5.953	-4.3203	0	0	0	0	0	0	0	0	-2.7297		
24	0	-2.1394	0	-5.5174	-5.5351	0	0	0	0	0	0	0	0	0	0		
25	-4.2579	-1.2689	-4.0261	-6.2758	-10.9338	0	0	0	0	0	0	0	0	0	0		
26	-5.723	-5.7214	-4.688	-4.1666	-7.8009	0	0	0	0	0	0	0	0	0	0		
27	-6.1893	-6.2191	-5.6458	-10.2795	-1.8667	0	0	0	0	0	0	0	0	0	0		
28	-9.8338	-4.7621	-9.0137	-8.9361	0	0	0	0	0	0	0	0	0	0	0		
29	-9.8401	-12.4678	-9.8609	-6.6756	0	0	0	0	0	0	0	0	0	0	0		
30	-8.1216	-7.1889	-6.2968	0	0	0	0	0	0	0	0	0	0	0	0		
31	0	-7.4325	-5.2244	0	0	0	0	0	0	0	0	0	0	0	0		
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Fig 9.10 (b) Negative Accumulator array for Open B Phase (0 to 360 deg)

	1		1
13	11.25	16	213.75
14	11.25	17	180
15	11.25	18	258.75
16	33.75	19	258.75
17	360	20	258.75
18	78.75	21	258.75
19	78.75	22	258.75
20	78.75	23	180
21	78.75	24	213.75
22	78.75	25	213.75
23	360	26	213.75
24	33.75	27	213.75
25	33.75	28	213.75
26	33.75	29	213.75
27	33.75	30	213.75
28	33.75	31	213.75
29	33.75	32	213.75
30	33.75	33	213.75
31	33.75	34	213.75
32	33.75	35	213.75
33	33.75	36	213.75
34	33.75	37	213.75
35	33.75	38	213.75
36	33.75	39	213.75
37	33.75	40	213.75
38	33.75	41	213.75
39	33.75	42	213.75
40	33.75	43	
41	33.75	44	
42	33.75		

Fig 9.10 (c) Recorded  $\theta$  values

4. **Open C Phase:** Similar to open A phase methodology, Fig 9.11 (a) & (b) shows the maximum  $\rho$  value for the positive accumulator array with the particular  $\theta$  value.

As explained in 2, the  $\theta$  values which are obtained while the C phase is open are as follows,

In Fig 9.11 (a), we can find that the maximum  $\rho$  value obtained as result of Hough transformation is at  $\theta = 15$ , which as per the decimation is  $15 * 11.25 = 168.75$  for positive accumulator array and  $31 * 11.25 = 348.75$  for negative accumulator array.

In the Fig 9.10 (c), we can see that maximum  $\theta$  value is **168.75** and **348.75**, now if these angles are plotted with the  $\rho$  value, the trajectory of current vector is similar to *Fig 9.5 (d)*, which is the **Open C Phase trajectory**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	1.2124	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1.5064	0	0	0	1.783	1.0323	0	0	0	0	0	0	0	0
3	0	0	1.0104	1.1401	1.0088	0	0	1.8024	1.1069	1.3297	0	0	0	0	0	1.0362	1.3428
4	0	0	2.9808	0	3.6801	0	0	2.3458	1.3349	1.5333	1.3683	0	0	0	0	0	0
5	0	1.4302	2.5506	4.1328	1.4953	0	0	1.5706	2.9918	1.2878	1.8132	1.7098	0	0	0	0	0
6	0	4.0245	0	3.4282	0	0	0	6.9885	2.3506	2.147	2.2735	2.2576	1.1093	0	0	0	0
7	3.1451	3.9893	0	0	0	0	0	5.6446	2.6698	2.1288	1.6096	2.4219	2.7373	1.901	0	0	0
8	4.7751	1.1701	2.8662	0	0	0	0	1.8115	3.803	2.945	2.9088	2.1745	2.3962	3.3451	2.352	1.6318	1.709
9	4.7344	0	7.2888	4.5455	0	0	0	0	2.6671	3.0334	2.2692	1.7178	3.4287	2.4354	3.0727	3.8419	3.3916
10	0	0	4.3336	7.6148	0	0	0	0	1.6182	3.4309	2.812	3.4061	2.7869	3.1065	2.8237	2.3964	2.6688
11	0	3.625	0	3.4707	0	0	0	8.6107	3.3715	3.5158	3.4439	3.319	3.405	3.1746	3.8802	4.7153	
12	0	7.0036	0	1.1927	0	1.0835	0	0	13.2166	5.3062	4.3513	3.5932	2.7558	3.7506	3.4735	3.5014	4.3443
13	2.3552	7.9321	0	0	0	0	1.0439	0	5.2761	5.4411	4.6824	4.1578	3.6439	3.5196	4.394	3.8826	3.923
14	3.8991	1.5148	4.6722	0	1.5256	1.8118	0	0	1.9305	1.6798	5.4309	5.1595	3.6537	2.9443	4.2065	5.7127	4.6171
15	8.533	1.2108	11.6024	0	1.0492	1.0429	1.1958	1.51	1.2023	2.995	2.101	2.6893	5.4198	4.1966	3.7263	4.1969	4.2007
16	7.1662	0	6.722	0	0	0	0	0	9.3024	5.4148	3.3625	3.873	4.4652	4.7928	3.8301	4.7823	
17	3.5017	0	0	0	0	0	0	0	14.8237	3.5551	3.5329	1.7046	4.0806	3.7547	3.7359	4.8972	
18	0	1.2832	0	0	0	0	0	0	8.2195	0	3.0624	3.4275	2.1426	3.4135	3.7526	2.5324	
19	0	7.0275	0	0	0	0	0	0	3.1942	2.6661	5.3179	4.3428	2.6485	2.4747	2.6633	4.4041	
20	0	14.158	0	0	0	0	0	0	0	7.0427	3.5825	3.2155	5.209	2.6037	4.8349	4.002	
21	0	6.7144	0	0	0	0	0	0	0	14.0848	1.0286	4.6397	2.3131	4.9896	3.148	2.506	
22	2.4582	0	0	0	0	0	0	0	0	15.7809	2.2118	4.6064	3.7371	3.5102	3.5142	7.7192	
23	9.0117	0	0	0	0	0	0	0	0	7.1291	2.9947	2.0511	6.9036	4.6108	5.5293	2.7502	
24	13.9064	0	0	0	0	0	0	0	0	0	11.9765	1.2199	3.3707	5.7454	5.0205	1.9436	
25	8.7412	0	0	0	0	0	0	0	0	0	17.5034	2.2443	0	1.9834	0	2.2523	
26	1.5486	0	0	0	0	0	0	0	0	0	14.2201	7.5814	2.8674	1.3157	2.865	10.6943	
27	0	0	0	0	0	0	0	0	0	0	7.2746	13.4973	2.977	2.143	2.7087	16.1827	
28	0	0	0	0	0	0	0	0	0	0	0	22.6828	12.6174	9.0148	13.4249	24.2465	
29	0	0	0	0	0	0	0	0	0	0	0	9.2839	17.1388	13.3671	21.7766	4.9427	
30	0	0	0	0	0	0	0	0	0	0	5.0778	18.8278	24.9175	17.9357	0		
31	0	0	0	0	0	0	0	0	0	0	0	8.9632	11.1546	5.5767	0		
32	0	0	0	0	0	0	0	0	0	0	0	2.2225	4.4719	0	0		
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
1	0	0	0	0	1.7571	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	4.3896	2.8629	0	0	1.1631	0	0	0	0	0	0	0
3	1.294	1.7146	1.8888	1.815	7.5203	2.8011	2.4029	2.0438	1.3699	1.6165	1.0451	0	0	0	0	0
4	1.5207	1.5165	2.1743	3.1866	13.5251	5.5981	2.1696	1.6417	1.355	0	0	0	0	0	0	0
5	0	0	1.0133	2.1058	14.4934	2.7438	0	0	0	0	0	0	0	0	0	0
6	0	0	0	7.6364	4.1928	6.1424	3.7283	0	0	0	0	0	0	0	0	0
7	0	1.8083	4.9515	4.1087	0	11.0059	3.2371	2.3188	0	0	0	0	0	0	0	0
8	2.9482	4.3942	5.4204	6.8548	0	6.6296	3.5686	4.0877	0	0	0	0	0	0	0	1.386
9	3.4207	4.5279	3.3967	3.8131	0	0	2.1036	4.143	3.9732	1.1822	0	0	0	1.4607	4.8	
10	4.4261	4.5483	5.2126	11.0587	0	0	9.0381	3.5971	4.6626	4.6435	2.5518	1.7249	3.5431	5.2196	5.4861	
11	4.3845	4.0462	5.2511	14.4373	0	0	5.8109	0	4.1487	4.7091	6.4184	6.3083	5.9694	5.3706	3.3817	
12	4.5775	4.5685	5.6686	3.6474	0	0	2.4332	2.9034	1.9963	4.6523	4.2134	5.8563	5.259	5.9536	0	0
13	2.724	6.0065	4.1617	0	1.1838	0	13.093	7.4374	1.0514	2.5799	4.5383	5.3233	4.6381	1.2862	1.9242	
14	5.0392	3.918	8.1053	1.1373	1.1358	0	6.5087	11.8471	1.8256	1.1335	2.066	1.3718	0	1.129	1.2536	
15	5.5408	5.4161	7.5632	2.3751	1.3461	0	0	1.346	5.4318	1.8159	1.6577	0	0	1.0606	0	
16	4.4646	4.7734	21.4279	2.0726	1.4393	1.1189	0	0	11.1694	3.3758	1.4374	1.1256	0	0	3.1815	
17	4.607	6.1324	8.4491	0	1.1732	1.3423	1.3639	3.0913	10.0033	6.3372	1.5412	1.0182	1.0196	2.5589	6.9954	
18	4.8524	3.7292	0	0	0	0	15.411	0	8.8477	3.4162	2.713	2.523	4.7151	8.0674		
19	2.8384	3.6107	0	0	0	0	10.9281	0	10.1153	6.6349	5.178	5.347	8.7779	6.9897		
20	6.8127	14.8225	0	0	0	0	0	0	1.7735	9.3355	8.6465	9.2317	7.6072	0		
21	2.2884	21.1328	0	0	0	0	0	1.4938	0	8.9656	8.0422	7.7886	4.7778	0		
22	2.6401	5.9216	0	0	0	0	0	15.4547	0	0	4.7963	4.7673	0	0		
23	6.9523	0	0	0	0	0	0	0	13.9851	0	0	0	0	0	0	
24	17.77	0	0	0	0	0	0	2.3768	0	0	0	0	0	0	0	
25	21.4203	0	0	0	0	0	0	0	8.7897	0	0	0	0	2.0253		
26	4.936	0	0	0	0	0	0	0	18.9252	0	0	0	0	9.3784		
27	0	0	0	0	0	0	0	0	10.2095	5.1579	0	0	1.9125	15.3566		
28	0	0	0	0	0	0	0	0	0	18.8025	3.9608	1.9809	9.823	10.8954		
29	0	0	0	0	0	0	0	0	0	14.2125	12.8142	10.4547	17.1077	2.583		
30	0	0	0	0	0	0	0	0	0	4.1643	17.6668	18.5772	11.0952	0		
31	0	0	0	0	0	0	0	0	0	0	9.8979	11.7769	3.3771	0		
32	0	0	0	0	0	0	0	0	0	0	0	1.902	0	0		
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig 9.11 (a) Positive Accumulator array for Open C Phase (0 to 360 deg)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
1	0	0	0	0	0	-4.3896	-2.8629	0	0	-1.1631	0	0	0	0	0	0	0	
2	0	0	0	0	0	-1.3428	-1.294	-1.7146	-1.8888	-1.815	-7.5203	-2.8011	-2.4029	-2.0438	-1.3699	-1.6165	-1.0451	
3	0	-1.5207	-1.5165	-2.1743	-3.1866	-13.5251	-5.5981	-2.1696	-1.6417	-1.355	0	0	0	0	0	0	0	
4	0	0	0	-1.0133	-2.1058	-14.4934	-2.7438	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	-7.6364	-4.1928	-6.1424	-3.7283	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	-1.8083	-4.9515	-4.1087	0	-11.0059	-3.2371	-2.3188	0	0	0	0	0	0	-3.1451	
7	0	0	-1.709	-2.9482	-4.3942	-5.4204	-6.8548	0	-6.6296	-3.5668	-4.0877	0	0	0	0	-1.386	-4.7751	
8	0	-3.3916	-3.4207	-4.5279	-3.3967	-3.8131	0	0	-2.1036	-4.143	-3.9732	-1.1822	0	0	-1.4607	-4.8	-4.7344	
9	0	-2.6688	-4.4261	-4.5483	-5.2126	-11.0587	0	0	-9.0381	-3.5971	-4.6526	-4.6435	-2.5518	-1.7249	-3.5431	-5.2196	-5.4861	
10	0	-4.7153	-4.3845	-4.0462	-5.2511	-14.4373	0	0	-5.8109	0	-4.1487	-4.7091	-6.4184	-6.3083	-5.9694	-5.3706	-3.3817	
11	0	-4.3443	-4.5775	-4.5685	-5.6686	-3.6474	0	0	-2.4332	-2.9034	-1.9963	-4.6523	-4.2134	-5.8563	-5.259	-5.9536	0	
12	0	-3.923	-2.724	-6.0065	-4.1617	0	-1.1838	0	-13.093	-7.4374	-1.0514	-2.5799	-4.5383	-5.3233	-4.6381	-1.2862	-1.9242	
13	0	-4.6171	-5.0392	-3.918	-8.1053	-1.1373	-1.1358	0	-6.5087	-11.8471	-1.8256	-1.1335	-2.066	-1.3718	0	-1.129	-1.2536	
14	0	-4.2007	-5.5408	-5.4161	-7.5632	-2.3751	-1.3461	0	0	-1.346	-5.4318	-1.8159	-1.6577	0	0	-1.0606	-8.533	
15	0	-4.7823	-4.4646	-4.7734	-21.4279	-2.0726	-1.4393	-1.1189	0	0	-11.1694	-3.3758	-1.4374	-1.1256	0	0	-3.1815	-7.1662
16	0	-4.8972	-4.607	-6.1324	-8.4491	0	-1.1732	-1.3423	-1.3639	-3.0913	-10.0033	-6.3372	-1.5412	-1.0182	-1.0196	-2.5589	-6.9954	
17	0	-2.5324	-4.8524	-3.7292	0	0	0	0	-15.411	0	-8.8477	-3.4162	-2.713	-2.523	-4.7151	-8.0674	0	
18	0	-4.4041	-2.8384	-3.6107	0	0	0	0	-10.9281	0	-10.1153	-6.6349	-5.178	-5.347	-8.7779	-6.9897	0	
19	0	-4.002	-6.8127	-14.8225	0	0	0	0	0	0	-1.7735	-9.3355	-8.6465	-9.2317	-7.6072	0	0	
20	0	-2.506	-2.2884	-21.1328	0	0	0	0	0	-1.4938	0	-8.9566	-8.8422	-7.7886	-4.7778	0	0	
21	0	-7.7192	-2.6401	-5.9216	0	0	0	0	0	-15.4547	0	0	-4.7963	-4.7873	0	0	-2.4582	
22	0	-2.7502	-6.9523	0	0	0	0	0	0	-13.9851	0	0	0	0	0	-9.0117		
23	0	-1.9436	-17.77	0	0	0	0	0	0	-2.3768	0	0	0	0	0	-13.9064		
24	0	-2.2523	-21.4203	0	0	0	0	0	0	-8.7897	0	0	0	0	-2.0253	-8.7412		
25	0	-10.6943	-4.936	0	0	0	0	0	0	-18.9252	0	0	0	0	-9.3784	-1.5486		
26	0	-16.1827	0	0	0	0	0	0	0	-10.2095	-5.1579	0	0	-1.9125	-15.3566	0		
27	0	-24.2465	0	0	0	0	0	0	0	-18.8025	-3.9608	-1.9809	-9.823	-10.8954	0			
28	0	-4.9427	0	0	0	0	0	0	0	-14.2125	-12.8142	-10.4547	-17.1077	-2.583	0			
29	0	0	0	0	0	0	0	0	0	-4.1843	-17.6668	-18.5772	-11.0952	0	0			
30	0	0	0	0	0	0	0	0	0	-9.8979	-11.7769	-3.3771	0	0				
31	0	0	0	0	0	0	0	0	0	-1.902	0	0	0	0				
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
1	0	0	0	-1.2124	0	0	0	0	0	0	0	0	0	0	0			
2	0	0	-1.5064	0	0	0	-1.783	-1.0323	0	0	0	0	0	0	0			
3	0	-1.0104	-1.1401	-1.0068	0	0	-1.8024	-1.1069	-1.3297	0	0	0	0	-1.0362				
4	0	-2.9808	0	-3.6801	0	0	-2.3458	-1.3349	-1.5333	-1.3683	0	0	0	0	0			
5	0	-1.4302	-2.5506	-4.1328	-1.4953	0	0	-1.5706	-2.9918	-1.2878	-1.8132	-1.7098	0	0	0			
6	0	-4.0245	0	-3.4282	0	0	-6.9885	-2.3506	-2.147	-2.2735	-2.2576	-1.1093	0	0	0			
7	0	-3.9892	0	0	0	0	-5.6446	-2.6698	-2.1288	-1.6096	-2.4219	-2.7373	-1.901	0	0			
8	0	-1.1701	-2.8662	0	0	0	-1.8115	-3.803	-2.945	-2.9088	-2.1745	-2.3962	-3.3451	-2.352	-1.6318			
9	0	-7.2888	-4.5455	0	0	0	0	-2.6671	-3.0334	-2.2692	-1.7178	-3.4267	-2.4354	-3.0727	-3.8419			
10	0	-4.3336	-7.6148	0	0	0	0	-1.6182	-3.4309	-2.812	-3.4061	-2.7869	-3.1065	-2.8237	-2.3964			
11	0	-3.625	0	-3.4707	0	0	0	-8.6107	-3.3715	-3.5158	-3.4439	-3.319	-3.405	-3.1746	-3.8802			
12	0	-7.0036	0	-1.1927	0	-1.0835	0	0	-13.2166	-5.3062	-4.3513	-3.5932	-2.7558	-3.7506	-3.4735	-3.5014		
13	0	-7.9321	0	0	0	-1.0439	0	-5.2761	-5.4411	-4.6824	-4.1578	-3.6439	-3.5196	-4.394	-3.8826			
14	0	-1.5148	-4.6722	0	-1.5256	-1.8118	0	0	-1.9305	-1.6798	-5.4309	-5.1595	-3.6537	-2.9443	-4.2065	-5.7127		
15	0	-1.2108	-11.6024	0	-1.0492	-1.0429	-1.1958	-1.51	-1.2023	-2.995	-2.101	-2.6893	-5.4198	-4.1966	-3.7263	-4.1969		
16	0	-6.722	0	0	0	0	0	0	-9.3024	-5.4148	-3.3625	-3.873	-4.4652	-4.7928	-3.8301			
17	0	0	0	0	0	0	0	0	-14.8237	-3.5551	-3.5329	-1.7046	-4.0806	-3.7547	-3.7359			
18	0	-1.2832	0	0	0	0	0	0	-8.2195	0	-3.0624	-3.4275	-2.1426	-3.4135	-3.7526			
19	0	-7.0275	0	0	0	0	0	0	-3.1942	-2.6661	-5.3179	-4.3428	-2.6485	-2.4747	-2.6633			
20	0	-14.158	0	0	0	0	0	0	0	-7.0427	-3.5825	-3.2155	-5.209	-2.6037	-4.8349			
21	0	-6.7144	0	0	0	0	0	0	0	-14.0848	-1.0286	-4.6397	-2.3131	-4.9896	-3.148			
22	0	0	0	0	0	0	0	0	-15.7809	-2.2118	-4.6064	-3.7371	-3.5102	-3.5142				
23	0	0	0	0	0	0	0	0	-7.1291	-2.9947	-2.0511	-6.9036	-4.6108	-5.5293				
24	0	0	0	0	0	0	0	0	0	-11.9765	-1.2199	-3.3707	-5.7454	-5.0205				
25	0	0	0	0	0	0	0	0	0	-17.5034	-2.2443	0	-1.9834	0				
26	0	0	0	0	0	0	0	0	0	-14.2201	-7.5814	-2.8674	-1.3157	-2.865				
27	0	0	0	0	0	0	0	0	0	-7.2746	-13.4973	-2.977	-2.143	-2.7087				
28	0	0	0	0	0	0	0	0	0	0	-22.6828	-12.6174	-9.0148	-13.4249				
29	0	0	0	0	0	0	0	0	0	0	-9.2839	-17.1388	-13.3671	-21.7766				
30	0	0	0	0	0	0	0	0	0	0	-5.0778	-18.8278	-24.9175	-17.9357				
31	0	0	0	0	0	0	0	0	0	0	-8.9632	-11.1546	-5.5767					
32	0	0	0	0	0	0	0	0	0	0	-2.2225	-4.4719	0					
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Fig 9.11 (b) Negative Accumulator array for Open C Phase (0 to 360 deg)

9	168.75
10	168.75
11	168.75
12	168.75
13	168.75
14	168.75
15	168.75
16	168.75
17	168.75
18	168.75
19	168.75
20	168.75
21	168.75
22	168.75
23	168.75
24	168.75
25	168.75
26	168.75
27	168.75
28	168.75
29	168.75
30	168.75
31	168.75
32	168.75
33	168.75
34	168.75
35	168.75
36	168.75
6	303.75
7	303.75
8	303.75
9	348.75
10	348.75
11	348.75
12	348.75
13	348.75
14	348.75
15	348.75
16	348.75
17	348.75
18	348.75
19	348.75
20	348.75
21	348.75
22	348.75
23	348.75
24	348.75
25	348.75
26	348.75
27	348.75
28	348.75
29	348.75
30	348.75
31	348.75
32	348.75
33	348.75

Fig 9.11 (b) Recorded  $\theta$  values

Hence, from the above observations, the summary of the  $\theta$  values from the Hough transformation are:

Condition	Positive $\theta$	Negative $\theta$
<b>Normal</b>	Distributed	Distributed
<b>Open A Phase</b>	101.25	281.25
<b>Open B Phase</b>	33.75	213.75
<b>Open C Phase</b>	168.75	348.75

Table 9.1 Summary for Hough Transformation results

# Chapter 10

## Analysis of the Results

In this chapter, the results obtained throughout the thesis are consolidated to get an overview of the work. The Figures, tables and concepts are referenced with the previous chapters

### 10.1 System Identification

Using the cross – correlation fundamentals for system identification [4.2 Cross – Correlation], and architecture [5.2.1 Improved architecture of the Cross – Correlation] for Xilinx Motor control system, the impulse response is derived.

The phase shift determined from the cross correlation method and timing to synchronize the input and output are as follows.

- 8192 Acquisition values: **26.3ms**
- 13 – bit PRBS sequence injection : **27.33ms**

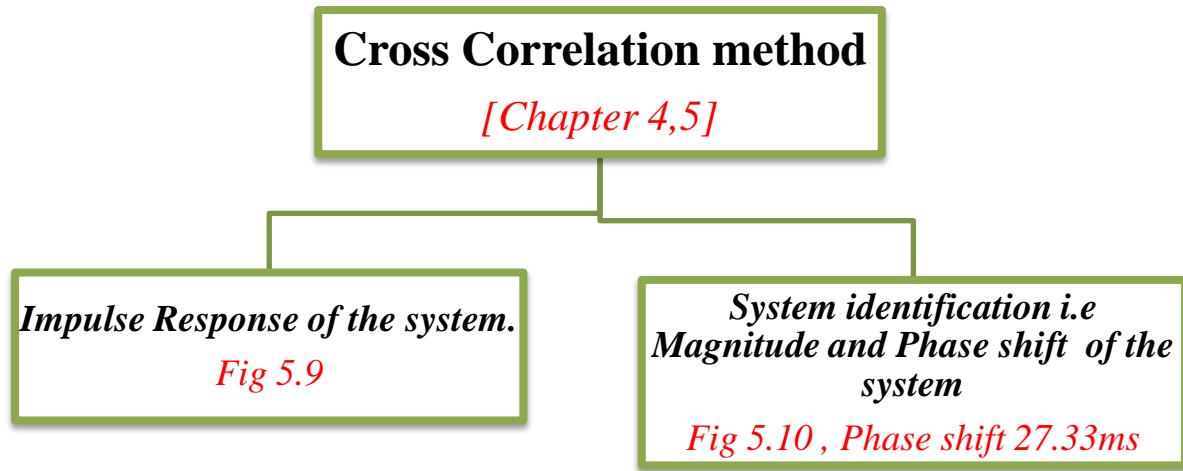
This shows that, single bit of PRBS is injected into the system in  $3.3\mu s$ . The timing obtained is much improved compared to [4].

Hence, from the time of input and acquisition in output, the phase shift obtained for the system as whole is **27.33ms**.

The trajectories from the cross – correlation method is shown in [5.7.1 Cross – Correlation] and the standard deviation in Correlation shows the **minimum deviation or input and output are in correlation**.

The impulse response derived from the Xilinx Motor control solution system is shown in [5.7.2 Impulse Response.]. Converting the impulse response in time domain to frequency domain using FFT, the Magnitude and Phase shift are shown in [5.7.3 Magnitude and Phase plot]

Fig 5.9 shows the summary and reference of the result obtained for the system identification.

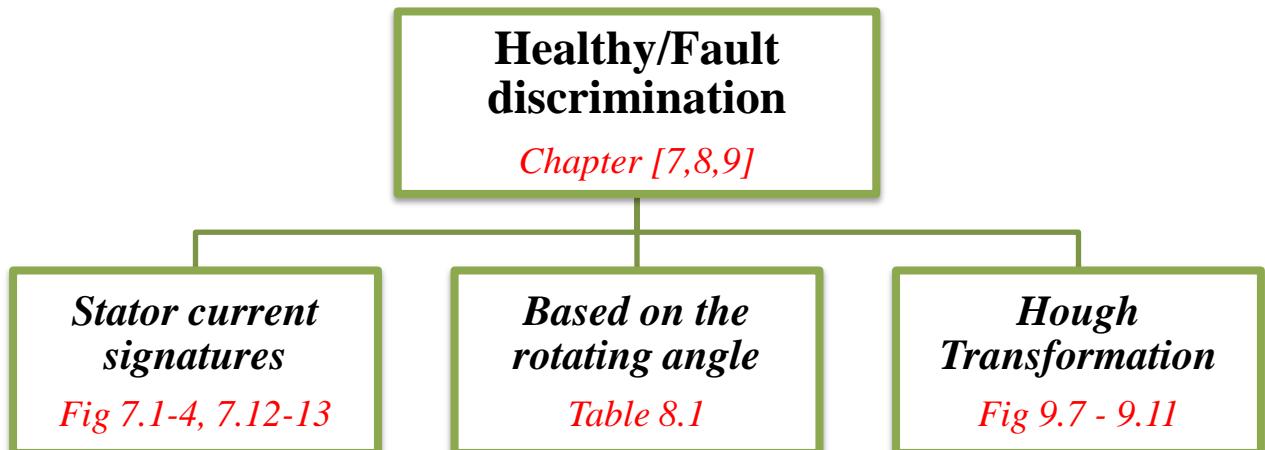


*Fig 10.1 Cross correlation results Analysis*

## 10.2 Open Phase fault detection

In the thesis, the different methods to analyze and detect the open phase are dealt. With the simulations results from different methodology a computational data is presented which would give a clear idea of open phase fault.

Fig 10.2 shows the analysis of the results for the open phase fault detection along with the reference to the figures.



*Fig 10.2 Open phase fault detection results analysis*

# Chapter 11

## Conclusion

In this Master thesis, mainly two aspects have been dealt. *Firstly*, system identification with particular reference to PMSM drives and *secondly*, open phase fault detection algorithm using Hough transformation is implemented.

System parameters are identified using the cross – correlation technique with the injection of Pseudo Random Binary Sequence (PRBS) perturbation into the PMSM drives. The need for PRBS injection in embedded environment is presented with the simulation results [5.1 Introduction].

The synchronization between the PRBS injection and output current acquisition is achieved. The results show that, the 13 – bit PRBS is injected in **27.33 ms** or single bit PRBS is injected at **3.3us**. In comparison with the FOC IP activation time, which takes **26.3ms** for acquisition of 8192 current values, the PRBS injection has sufficient time to derive the impulse response of the system.

To detect the open phase fault, the motor current signatures are used. The different methods to identify the fault are described and the simulation results are compared. The following are the proposals for the open phase fault detection:

- The current signature analysis [7.1 Introduction] needs an expert advice while studying the current vector rotating in the reference frame. This method gives an insight of the motor model parameters like inductances and fluxes under the open phase fault.
  - Vector space angle [8.1 Introduction] method gives an analytical data about the rotating angle of current vector under open phase condition. Based on the angle, the condition of the motor is analyzed.
- Limitation:** With this method, the data to trace current vector trajectory for normal operation of the motor is inappropriate.
- Hough Transformation [9.1 Introduction] implemented and experimental results provide a computationally efficient data to detect the trajectory of the current vector under the open phase and hence, this Master Thesis proposes that **Hough Transformation** is an efficient algorithm for the open phase fault detection.

Thus, the Thesis works on the state art of the art Xilinx All programmable SoC technology to identify the system parameters and derivation of impulse response. With the availability of real time Hardware in the Loop (HiL) on the final product, Hough Transformation is implemented to detect the open phase fault.

## Chapter 12

### Future Work

The Zynq AP SoC Intelligent Drives platform [25] provides all necessary hardware, design tools, IP cores, and reference designs for industrial embedded control systems. The platform is ideal for designs requiring high performance motion control and/or industrial networking capability.

Featuring dual core ARM Cortex – A9 MPCore application-class processors, Zynq – 7000 All Programmable SoCs provide a single-chip motor control platforms integrating key functions and components such as complex motion algorithms, modulation schemes, etc. I believe that using such technologies most intelligent and reliable motor control solutions and algorithms can be developed.

With the QDESYS embedded design for the motor control and HiL, complex control algorithms could be implemented. The solution has many API's to interface motor data giving developers ready to use feasibility.

With the system identification and synchronization between input and output obtained from this thesis, the complex algorithms like **online auto –tuning** based on the cross – correlation technique can be developed. The controller's auto tuning mechanism could be directly performed on the digital systems. The algorithm for the **vibration suppression** in the PMSM drives can be designed and developed using the system parameters obtained from the thesis.

Hough transformation is proposed as an efficient algorithm to detect the open phase fault. With this algorithm, a **control mechanism** can be developed which could detect the open phase fault under the Industrial environment and automatically control the motor abnormal behavior in order to avoid damage of motor and controllers.

Faults are unavoidable and could occur at any part of the system. Using current signature analysis methods, algorithms which could detect the **abnormalities** in switching converters, sensor failure, broken stator or rotor bars could be implemented.

My experience with the Xilinx Hardware and Software tools were very impressive. Xilinx AP SoC Intelligent drives kit for the motor control is capable to deliver the Intelligent Industry requirements. With the interfacing of the high level languages, software engineers could easily develop most profound complex algorithms for motor control.

## **List of Figures:**

Fig 2.1 Milestones for Master Thesis .....	4
Fig 2.2 Milestones status.....	5
Fig 3.0 Xilinx Intelligent Motor Control Kit .....	6
Fig 3.0.1 Zynq – 7000 block diagram.....	8
Fig 3.2 Processing system in Zynq AP SoC .....	9
Fig 3.4 PL Interface to PS Memory subsystem .....	12
Fig 3.5 Xilinx SDK.....	14
Fig 3.6 FMC Connectors.....	15
Fig 3.9 BLWS232D Winding .....	17
Fig 3.10 Motor Control Illustration .....	19
Fig 3.12 FOC IP Core .....	21
Fig 4.12 .....	31
Fig 5.6 (a) 1 <sup>st</sup> run peak at ~780 (b) 2 <sup>nd</sup> , peak at 800 (c) 3 <sup>rd</sup> at 810.....	39
Fig 5.8 Space vector trajectory .....	40
Fig 6.2 Park Transformation .....	46
Fig 7.1 Healthy Motor Condition: Locus of $\alpha$ - $\beta$ currents.....	49
Fig 7.2 Open A Phase Condition: Locus of $\alpha$ - $\beta$ currents.....	50
Fig 7.3 Open B Phase Condition: Locus of $\alpha$ - $\beta$ currents .....	51
Fig 7.4 Open C Phase Condition: Locus of $\alpha$ - $\beta$ currents .....	52
Fig 7.7 Typical Inductance characteristic in PMSM. Source: [19].....	55
Fig 7.9 d- and q- axis alignment. Source [19].....	56
Fig 7.10 Step Input in d-axis alignment.....	57
Fig 7.11 Step input in q – axis alignment .....	58
Fig 7.12 Id, Iq under open phase.....	60
Fig 7.13 $\emptyset_d$ , $\emptyset_q$ under Open phase .....	61
Fig 9.1 Normal parameters of the line .....	66
Fig 9.2 Projection of colinear points onto a line .....	67
Fig 9.7 (a) Positive Accumulator array for Normal motor runs .....	74
Fig 9.7 (b) Negative Accumulator arrays for Normal motor runs .....	75
Fig 9.8 (a) Positive accumulator cell for Open A Phase (0 to 360 deg). .....	77
Fig 9.8 (b) Negative accumulator cell for Open A Phase (0 to 360 deg). .....	78
Fig 9.9 Recorded $\theta$ values.....	79
Fig 9.11 (b) Negative Accumulator array for Open C Phase (0 to 360 deg) .....	84
Fig 10.1 Cross correlation results Analysis .....	87
Fig 10.2 Open phase fault detection results analysis .....	87

## **List of Tables**

Table 3.0.1 Wiring[9].....	17
Table 3.0.2 Motor Specification [9].....	18
Table 8.1 Simulation for Vector Space.....	64
Table 9.1 Summary for Hough Transformation results .....	85

## References

### Chapter 1

- [1]. www.xilinx.com
- [2]. [http://en.wikipedia.org/wiki/Industry\\_4.0](http://en.wikipedia.org/wiki/Industry_4.0)
- [3]. <http://www.xilinx.com/applications/industrial/motor-control.html>
- [4]. Botao Miao, Regen Zane , Dragan Maksimovic, “*A Modified Cross – Correlation method for system identification of Power converters with digital control*” 2004 35<sup>th</sup> Annual IEEE Power Electronics Specialties conference, Aachen Germany.

### Chapter 3

- [5]. [www.xilinx.com/support/documentation/data\\_sheets/ds190-Zynq-7000-Overview.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf)
- [6]. [www.xilinx.com/support/university/vivado.html](http://www.xilinx.com/support/university/vivado.html)
- [7]. [www.xilinx.com/tools/sdk.htm](http://www.xilinx.com/tools/sdk.htm)
- [8]. [www.scilab.org](http://www.scilab.org)
- [9]. [www.anaheimautomation.com/products/brushless/brushless-motor-item.php?sID=146&pt=i&tID=96&cID=22](http://www.anaheimautomation.com/products/brushless/brushless-motor-item.php?sID=146&pt=i&tID=96&cID=22)
- [10]. [www.xilinx.com/applications/industrial/motor-control.html](http://www.xilinx.com/applications/industrial/motor-control.html)

### Chapter 4

- [11]. Ljung L System Identification Theory for User
- [12]. [4].
- [13]. <http://www.aip.de/groups/soe/local/numres/bookfpdf/f13-2.pdf>
- [14]. Bengt Johansson, Matz Lenells , „Possibilities of Obtaining Small-Signal Models of DC-to-DC Power Converters by Means of System Identification“

### Chapter 6

- [15]. [4]
- [16]. C.C. Hang and Kok Kee Sin, “*Online Auto tuning of PID controllers based on the cross correlation technique*”, IEEE Transaction on Industrial Electronics, Vol. 38, No. 6, December 1991.
- [17]. TI Clarke and Park Transformations

### Chapter 7

- [18]. A. Khlaief, M. Boussak, *Senior Member, IEEE*, M. Gossa “Open Phase Faults Detection in PMSM Drives Based on Current Signature Analysis”.
- [19]. Freescale Semiconductors. PMSM Electrical parameters Measurement.

## **Chapter 9**

- [20]. Richard O. Duda and Peter E.Hart, Stanford Research Institute, Menlo Park, California  
“Use of the Hough Transformation to detect lines and curves in Pictures”.
- [21]. Rosenfeld, A, *Picture Processing by Computer*. Academic Press, New York, 1969.
- [22]. Hough P.V.C. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 18, 1962.

## **Chapter 10**

- [23]. Byoung, Lee, Kim, Hyun “ Low cost fault diagnosis algorithm for switch open damage in BLDC motor drives”
- [24]. Peuget , Courtine and Rognon, “Fault detection and isolation on a PWM inverter by knowledge based Model”

## **Chapter 11**

- [25]. <http://www.xilinx.com/products/boards-and-kits/zynq-7000-intelligent-drives-platform.html#overview>

## **Appendix**

A CD is provided with this Thesis. The structure of the contents included is as follows:

10\_Thesis\_Report

- |
- |\_\_\_\_ Joshi\_Vinayak\_Master\_Thesis.docx
- |\_\_\_\_ Joshi\_Vinayak\_Master\_Thesis.pdf

20\_EMBEDDED\_Development\_Work

- |
- |\_\_\_\_ Source\_Code\_PRBS\_Injection.c
- |\_\_\_\_ Readme.txt

30\_SCILAB\_Work

- |
- |\_\_\_\_ Correlation\_Fault\_detection.sce
- |\_\_\_\_ Correlation\_Sync.sce
- |\_\_\_\_ Hough\_Transformation.sce
- |\_\_\_\_ mcm2arlib.sce
- |\_\_\_\_ Readme.txt