

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/317958200>

# Design and Implementation of an Electronic Speed Controller for Brushless DC motors

Thesis · June 2017

DOI: 10.13140/RG.2.2.17641.29286

---

CITATION

1

READS

5,343

2 authors, including:



Kris Scicluna

Malta College of Arts, Science & Technology

45 PUBLICATIONS 35 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Sensorless Speed/Position Estimation in Electrical Machines [View project](#)



MCAST IET EE [View project](#)

**Design and Implementation of an Electronic Speed Controller  
for Brushless DC motors**

*Marco Mariano*

Submitted to the Malta College of Arts, Science and Technology (MCAST)  
Institute of Engineering and Transport in part fulfilment of the requirements for the  
Bachelor of Engineering (Honours) in Electronics and Control Engineering.

July 2017

# **Authorship Statement**

This dissertation is based on the results of research carried out by myself, is my own composition, and has not been previously presented for any other certified or uncertified qualification.

The research was carried out under the supervision of Mr. Kris Scicluna.

Marco Mariano 29/05/2017

# Copyright Statement

In submitting this dissertation to the MCAST Institute of Engineering and Transport, I understand that I am giving permission for it to be made available for use in accordance with the regulations of MCAST and the College Library.

Marco Mariano 29/05/2017

## Acknowledgements

I would like to express my sincere gratitude and appreciation to all the people, who with all the patience and dedication offered their time to help me achieve all the results, information, motivation and support needed during the whole process of the dissertation implementation.

I would like to thank Mr. Kris Scicluna for his continuous patience, guidance and understanding contributed to the fulfilment of the dissertation. I would also like to thank all the staff and technicians at the institute of Electrical and Electronics Engineering for their patience and help during the past years.

My deepest gratitude goes to my parents, family and to my girlfriend Francesca Grima for their unconditional love and support throughout my studies at MCAST.

# Contents

Authorship Statement .....	2
Copyright Statement .....	3
Acknowledgements .....	4
Contents .....	5
List of Abbreviations .....	13
Abstract .....	14
1 Introduction .....	16
1.1 Project Introduction .....	16
1.2 Project Aim .....	17
1.3 Project Objectives .....	17
1.4 Dissertation Layout .....	17
2 Literature Review .....	19
2.1 Introduction to the Literature Review .....	19
2.2 History and Applications of BLDC Machines .....	19
2.2.1 History of BLDC Machines .....	19
2.2.2 Advantages and Applications of BLDC Machines .....	20
2.3 Realization of BLDC Machine Drives .....	20
2.3.1 Introduction to BLDC Drives .....	20
2.3.2 Inverter Circuits .....	21
2.4 Control of BLDC Machines .....	22
2.4.1 Introduction to Control of BLDC Machines .....	22

2.4.2 Six Step Commutation .....	22
2.4.3 Direct Torque Control.....	22
2.4.4 Rotor Flux Oriented Control.....	23
2.5 Sensorless Implementation .....	25
2.5.1 Introduction to Sensorless Control of BLDC Machines .....	25
2.5.2 Conventional Speed/Position Algorithm as used in ESCs.....	25
2.5.3 Model Based Speed/Position Algorithms .....	26
3 Theory .....	27
3.1 Introduction to Theory .....	27
3.2 Mathematical Modelling of a BLDC Machine.....	27
3.3 Parameters Measurements and Calculations .....	29
3.3.1 Resistance and Inductance Measurement .....	30
3.3.2 Moment of Inertia and Viscous Coefficient of Friction Approximation .....	30
3.3.3 Determining the machine's no. of poles .....	30
3.3.4 Calculation of Rated Current, Rated Torque and $\varphi_{rd}$ .....	31
3.4 PI Controllers Design .....	32
3.4.1 Introduction.....	32
3.4.2 Currents PI Controller Design .....	34
3.4.3 Speed PI Controller Design .....	37
3.5 Observer Design .....	39
3.5.1 Introduction.....	39
3.5.2 Rotor Position Estimation .....	39
3.5.3 PI Controller Design for Back-emf Estimation .....	42
3.6 Conclusion .....	43
4 Simulation Results.....	44
4.1 Introduction to Simulation Results .....	44
4.2 Sensored Operation.....	44
4.2.1 BLDC Machine Model .....	44

4.2.2 Current and Speed PI Controllers Performance .....	46
4.3 Sensored Operation with Observer .....	49
4.3.1 Rotor Electrical Angle and Rotor Speed Estimation .....	49
4.4 Sensorless Operation .....	55
4.4.1 Transition Dynamics .....	55
4.5 Speed Control at Sensorless Operation .....	59
4.6 Conclusion .....	61
5 Experimental Setup .....	62
5.1 Introduction to Experimental Setup.....	62
5.2 The BLDC Machine .....	62
5.2.1 Construction of the BLDC Machine .....	62
5.2.2 Electrical and Mechanical Parameters of the BLDC Machine .....	63
5.3 Hardware Design .....	64
5.3.1 Introduction.....	64
5.3.2 Power Supply Circuit.....	64
5.3.3 Gate Driver Circuit .....	66
5.3.4 Three Phase Inverter Circuit .....	68
5.3.5 Current Sensing Circuit.....	70
5.3.6 Microcontroller Circuit .....	72
5.4 Software Design .....	74
5.4.1 Introduction.....	74
5.4.2 Space Vector Modulation Software Algorithm .....	76
5.5 Conclusion .....	78
6 Experimental Results.....	79
6.1 Introduction to Experimental Results .....	79
6.2 Experimental Results – Open Loop Operation.....	79
6.2.1 Introduction.....	79
6.2.2 Open Loop Operation Results.....	80

6.3 Experimental Results – Transition .....	84
6.3.1 Introduction.....	85
6.3.2 Transition Results .....	85
6.4 Experimental Results – After Transition.....	88
6.4.1 Introduction.....	89
6.4.2 After Transition Results – Set 1 .....	89
6.4.3 After Transition Results – Set 2.....	91
6.5 Conclusion .....	94
7 Conclusion.....	95
7.1 Introduction .....	95
7.2 Speed Control .....	95
7.2.1 Current PI controller Design .....	96
7.2.2 Speed PI controller Design .....	99
7.3 Observer.....	99
7.4 Further Improvements .....	101
7.5 Conclusion .....	102
References .....	104
Appendix A .....	109
Appendix B .....	114
RFOC Programming Code .....	115
SVM Programming Code .....	122
Appendix C .....	124

# List of Figures

Figure 2.1: High Level Diagram of a BLDC Machine Drive .....	21
Figure 3.1: RFOC with rotor position estimation control diagram .....	33
Figure 3.2: PI zero set on system's pole in order to obtain pole zero cancelation .....	35
Figure 3.3: Bode plot for the current closed loop transfer function .....	36
Figure 3.4: Setting of speed PI controller gains .....	38
Figure 3.5: Root locus design for the speed PI controller .....	38
Figure 3.6: BEMF $\alpha$ estimation diagram .....	40
Figure 3.7: BEMF $\beta$ estimation diagram .....	40
Figure 3.8: Rotor's speed $\omega$ estimation using a PLL.....	41
Figure 4.1: Left - BLDC machine simulation diagram. Right - BLDC machine simulation block ...	45
Figure 4.2: Left - PI Controllers simulation diagram. Right - PI Controllers simulation block .....	46
Figure 4.3: FOC simulation diagram.....	46
Figure 4.4: Three phase current at $\omega = 1300$ rad/s .....	47
Figure 4.5: Stationary reference frame currents at $\omega = 1300$ rad/s.....	47
Figure 4.6: Rotor reference frame current $i_d$ at $\omega = 1300$ rad/s .....	48
Figure 4.7: Rotor reference frame current $i_q$ at $\omega = 1300$ rad/s .....	48
Figure 4.8: BLDC machine's rotor speed with a speed reference input of $\omega = 1300$ rad/s .....	49
Figure 4.9: Left - Observer simulation diagram. Right - Observer simulation block .....	50
Figure 4.10: $i_d$ converging to $i_d^*$ at $\omega = 1300$ rad/s .....	51
Figure 4.11: $i_q$ converging to $i_q^*$ at $\omega = 1300$ rad/s .....	51
Figure 4.12: $v_d$ converging to $v_d^*$ at $\omega = 1300$ rad/s .....	52
Figure 4.13: $v_q$ converging to $v_q^*$ at $\omega = 1300$ rad/s .....	52
Figure 4.14: Rotor position angle formation from BEMF $\alpha$ and BEMF $\beta$ .....	53
Figure 4.15: Rotor position angle.....	53
Figure 4.16: PLL rotor angle estimate tracking real rotor angle .....	54
Figure 4.17: PLL rotor speed estimate tracking real rotor speed .....	54

Figure 4.18: Three phase currents during transition at $\omega = 1300$ rad/s .....	55
Figure 4.19: Rotor reference frame current $i_d$ during transition at $\omega = 1300$ rad/s .....	56
Figure 4.20: Rotor reference frame current $i_q$ during transition at $\omega = 1300$ rad/s .....	56
Figure 4.21: Rotor speed during transition with a speed reference input of $\omega = 1300$ rad/s .....	57
Figure 4.22: $v_\alpha$ converging to $v_\alpha$ during transition at $\omega = 1300$ rad/s.....	57
Figure 4.23: $v_\beta$ converging to $v_\beta$ during transition at $\omega = 1300$ rad/s.....	58
Figure 4.24: Rotor position angle during transition .....	58
Figure 4.25: Rotor reference frame current $i_d$ during sensorless operation for $\omega$ from 650 rad/s to 1300 rad/s and $\omega$ from 1300 rad/s to 650 rad/s .....	59
Figure 4.26: Rotor reference frame current $i_q$ during sensorless operation for $\omega$ from 650 rad/s to 1300 rad/s and $\omega$ from 1300 rad/s to 650 rad/s .....	60
Figure 4.27: Rotor speed during sensorless operation for $\omega$ from 650 rad/s to 1300 rad/s and $\omega$ from 1300 rad/s to 650 rad/s .....	60
Figure 5.1: BLDC machine used for the experimental setup .....	63
Figure 5.2: Power supply circuit diagram .....	65
Figure 5.3: Gate driver circuit diagram .....	67
Figure 5.4: Three phase inverter circuit diagram .....	69
Figure 5.5: Current sensing circuit diagram.....	71
Figure 5.6: STM32F407G microcontroller used for the experimental setup .....	73
Figure 5.7: Experimental setup .....	74
Figure 5.8: Flow chart diagram of the RFOC scheme software algorithm .....	75
Figure 5.9: Inverter eight useful switching states .....	76
Figure 5.10: Vector diagram .....	77
Figure 5.11: Switching times between two adjacent vector and the null vector.....	77
Figure 5.12: Alternate-reverse switching sequence (Zambada & Deb, 2010) .....	78
Figure 6.1: Three phase currents during open loop operation.....	80
Figure 6.2: Stationary Reference frame currents during open loop operation .....	80
Figure 6.3: Rotor reference frame current $i_d$ during open loop operation .....	81
Figure 6.4: Rotor reference frame current $i_q$ during open loop operation .....	81
Figure 6.5: Rotor speed estimation during open loop operation .....	82
Figure 6.6: $i_\alpha$ converging to $i_\alpha$ during open loop operation .....	82
Figure 6.7: $i_\beta$ converging to $i_\beta$ during open loop operation .....	83
Figure 6.8: Estimated Back-emf voltages during open loop operation.....	83
Figure 6.9: Rotor position angle during open loop operation .....	84

Figure 6.10: Three phase currents during transition .....	85
Figure 6.11: Rotor reference frame current id during transition.....	85
Figure 6.12: Rotor reference frame current iq during transition.....	86
Figure 6.13: Rotor speed during transition.....	86
Figure 6.14: Stationary reference frame current convergence during transition.....	87
Figure 6.15: Back-emf voltages during transition.....	87
Figure 6.16: Rotor position angle during transition .....	88
Figure 6.17: Rotor reference frame current id after transition from 40 rad/s to 100 rad/s .....	89
Figure 6.18: Rotor reference frame current iq after transition from 40 rad/s to 100 rad/s.....	89
Figure 6.19: Rotor speed after transition from 40 rad/s to 100 rad/s .....	90
Figure 6.20: Stationary reference frame current convergence after transition from 40 rad/s to 100 rad/s .....	90
Figure 6.21: Rotor position angle after transition from 40 rad/s to 100 rad/s.....	91
Figure 6.22: Rotor reference frame current id after transition from 100 rad/s to 300 rad/s .....	91
Figure 6.23: Rotor reference frame current iq after transition from 100 rad/s to 300 rad/s.....	92
Figure 6.24: Rotor speed after transition from 100 rad/s to 300 rad/s .....	92
Figure 6.25: Stationary reference frame current convergence after transition from 100 rad/s to 300 rad/s .....	93
Figure 6.26: Rotor position angle after transition from 100 rad/s to 300 rad/s.....	93
Figure 7.1: Effect of having pole-zero cancelation when applying a step input.....	97
Figure 7.2: Effect of not having pole-zero cancelation when applying a step input.....	97
Figure 7.3: Misalignment between the machine's pole and the PI controller's zero .....	98
Appendix A.1: Top - Power supply circuit PCB design. Bottom - Power supply circuit hardware implementation.....	110
Appendix A.2: Left - Gate driver circuit PCB design. Right - Gate driver circuit hardware implementation.....	111
Appendix A.3: Left - Three phase inverter circuit PCB design. Right - Three phase inverter circuit hardware implementation .....	112
Appendix A.4: Left - Current sensing circuit PCB design. Right - Current sensing circuit hardware implementation.....	113

## List of Tables

Table 5.1: BLDC machine parameters .....	63
Table 5.2: Power supply circuit inputs and outputs .....	64
Table 5.3: Gate driver circuit inputs and outputs .....	66
Table 5.4: Three phase inverter circuit inputs and outputs .....	68
Table 5.5: Current sensing circuit inputs and outputs .....	70
Table 5.6: Microcontroller circuit inputs and outputs.....	72
Table 7.1: Comparison between simulated and experimental results on speed control.....	95
Table 7.2: Comparison between simulated and experimental results on observer performance .....	99

## List of Abbreviations

- Brushless DC – BLDC
- Heating Ventilation and Air Conditioning – HVAC
- Unmanned Aerial Vehicles – UAV
- Electronic Speed Controller – ESC
- Rotor Flux Oriented Control – RFOC
- Pulse Width Modulation – PWM
- Six Switch Three Phase – SSTP
- Space Vector Modulation – SVM
- Proportional Integral – PI
- Electro Motive Force – emf
- Direct Current – DC
- Four Switch Three Phase – FSTP
- Boost Five Switch Three Phase – BFSTP
- Direct Torque Control – DTC
- Permanent Magnet Synchronous Machine – PMSM
- Sliding Mode Controller – SMC
- Phased Locked Loop – PLL
- Inductance Capacitance Resistance – LCR
- Single Input Single Output – SISO
- Printed Circuit Board – PCB
- Integrated Circuit – IC
- Field Programmable Gate Array – FPGA
- Digital Signal Processor – DSP
- National Aeronautics and Space Administration – NASA

# Abstract

The Brushless DC (BLDC) machine is becoming a very popular choice in drive technology over the last few years because of its numerous advantages when compared with other electrical machines. Such advantages include low maintenance, high power density, high efficiency and robustness. Because of its numerous advantages the BLDC machine found its self-used in different applications, such as automotive, aerospace, home appliances and Heating Ventilation, and Air Conditioning (HVAC) systems. Over the last decade, low power, low cost and high speed BLDC machines have also gained popularity in unmanned aerial vehicles (UAV), more commonly known as drones. Closed loop speed/position control of low power, low cost, high speed BLDC machines as used in low cost UAV's require the use of an inverter drive or an electronics speed controller (ESC), an encoder or tachogenerator to record the machine's speed/position information, and a control algorithm to process the speed/position information. Commonly used control algorithms offer poor efficiency and poor dynamic response. Furthermore, the use of dedicated speed/position sensors increases the cost, weight and reduces reliability.

The aim of this dissertation is to review the performance and validity of a model based back-electro motive force (emf) observer on a low power, low cost, high-speed BLDC machine using the rotor flux oriented control (RFOC) algorithm with a pulse width modulation (PWM) inverter through a simulated environment and an experimental setup. A detailed review of the relevant literature on the implementation of an ESC for a BLDC machine together with further literature about BLDC machines control schemes and sensorless control of BLDC machines is discussed. Moreover, the mathematical models of the BLDC machine, current and speed proportional integral (PI) controllers and the back-emf observer are designed in order to simulate the sensorless speed control of the BLDC

machine using the RFOC scheme together with a back-emf observer on MATLAB/Simulink.

Furthermore, an experimental setup consisting of a six switch three-phase (SSTP) inverter together with the software algorithms for the RFOC scheme and the space vector modulation (SVM) are designed and implemented in order to further test the performance and validity of the sensorless speed control of the BLDC machine using the RFOC scheme together with a back-emf observer. Simulation and experimental results are then presented and compared in order to determine the performance and validity of the model based back-emf observer on a low power, low cost, high speed BLDC machine.

# 1. Introduction

## 1.1 Project Introduction

The BLDC machine is becoming a very popular choice in drive technology over the last few years due to some of its advantageous features. These features include high torque, high efficiency, high power density, high speed, low maintenance and robustness (Saleh, et al., 2001), (Ejlali & Soleimani, 2012). Because of its advantageous features, the BLDC machine has been widely used in different applications. Such applications include, automotive, aerospace, home appliances and HVAC systems (Chun, et al., 2014), (Huang, et al., 2012). However, over the last decade, low power, high speed and low cost BLDC machines have also gained popularity in UAV or more commonly known as drones (Benito, et al., 2014), (Sanchez, et al., 2011).

Close loop speed/position control of BLDC machines used in low cost, high speed and low power UAVs requires the use of an inverter drive or an ESC, information about the shaft's speed/position that can be easily obtained with a dedicated sensor such as an encoder or tachogenerator, and a control algorithm to process the speed/position information. The currently most commonly used control scheme for BLDC machines is the six-step commutation, however, study suggests that such a control scheme has poor dynamic response, poor efficiency and high pulsating torque ripples (Rau, et al., 2015), (Abdulmelik & Hassan, 2016). Furthermore, the use of dedicated sensors would increase costs, weight and reduce reliability (Ejlali & Soleimani, 2012); hence, BLDC machines are usually combined with sensorless speed/position algorithms and improved control schemes.

## 1.2 Project Aim

The aim of this dissertation is to review the performance and validity of a model based back-emf observer on a low power, high-speed, low cost BLDC machine using the RFOC scheme with a PWM inverter through a simulated environment and an experimental setup.

## 1.3 Project Objectives

- Review relevant literature available on the implementation of an ESC for a BLDC machine
- Review the mathematical model of a BLDC machine and of a back-emf observer
- Design and tune current and speed PI controllers in cascaded form
- Design and implement an ESC as an experimental setup
- Collection and analysis of simulated and experimental results

## 1.4 Dissertation Layout

This section gives a brief overview of all the chapters presented in this dissertation. The dissertation is divided into seven chapter as follows:

**Chapter 1: Introduction** – Introduces the topic of BLDC machines, their applications and their control. Furthermore, the project aim and objectives are also discussed.

**Chapter 2: Literature Review** – A detailed review of the relevant literature on the implementation of an ESC for a BLDC machine together with further literature about BLDC machines control schemes and sensorless control of BLDC machines.

**Chapter 3: Theory** – A detailed review of the mathematical model of a BLDC machine and of a back-emf observer are discussed. Moreover, the design and tuning of current and speed PI controllers in cascade for the sensorless speed control of the BLDC machine using the RFOC scheme is discussed.

**Chapter 4: Simulation Results** – The review of the BLDC machine and back-emf observer mathematical model together with the designed and tuned current and speed PI controllers are implemented on MATLAB/Simulink where simulation results for the back-emf observer dynamics, sensored and sensorless speed control using the RFOC scheme are collected and analysed.

**Chapter 5: Experimental Setup** – The design and implementation of a SSTP inverter is discussed, furthermore, the software algorithms for the RFOC scheme and the SVM are designed and implemented.

**Chapter 6: Experimental Results** – The back-emf observer dynamics together with the sensorless speed control using the RFOC scheme is tested on the newly implemented experimental setup where experimental results are collected and analysed.

**Chapter 7: Conclusion** – Concludes the dissertation with an analyses and comparison of the simulated and experimental results obtained in the previous chapters, together with further improvements.

## 2. Literature Review

### 2.1 Introduction to the Literature Review

Electric drives are the single largest electrical end use and it is estimated that such systems account for 43-46% of all global electricity consumption (Waide & Brunner, 2011). Various types of electric machines are used in different practical electric drive applications including synchronous machines, induction machines, Direct Current (DC) machines, BLDC machines and switched reluctance machines (Xia, 2012). In this dissertation, a study on the design, operation and control of the BLDC machine will be discussed.

### 2.2 History and Applications of BLDC Machines

#### 2.2.1 History of BLDC Machines

Modern machine theory was first established when Faraday discovered the electromagnetism induction phenomenon in 1831, which led to the construction of the first DC machine nine years later (Xia, 2012). Harrison and Rye developed the first electronic commutator circuit powered by thyristors (Harrison & Rye, 1955). They were the first to successfully replace mechanical commutation used in a DC machine with an electronic one. As high-performance, power semiconductors and permanent magnet materials were developed the BLDC machine became more popular due to decrease in cost and increased power density. The first commercially available BLDC machine was developed in 1962 and by 1965 the national aeronautics and space administration (NASA) used the BLDC machine to run the life support system for its Apollo Lunar Module and other aircraft. In 1968, the BLDC machine was also introduced in battery-powered tape records and in 1970, the U.S. Army built and tested BLDC machines to be used in heavy-duty vehicles.

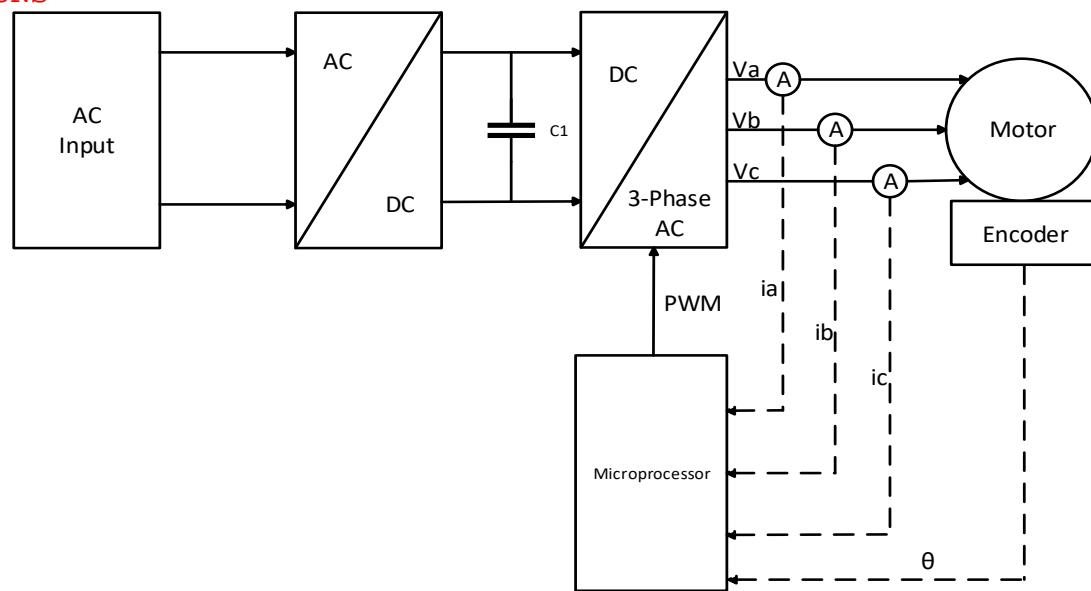
## 2.2.2 Advantages and Applications of BLDC Machines

The BLDC machine, when compared to the brushed DC machine, has a number of advantages due to the removal of commutators and brushes. Such advantages include low maintenance, high speed and easy hardware configuration (Bostanci, et al., 2015), (Liu, et al., 2016). These are critical requirements for unmanned vehicles (Benito, et al., 2014), (Sanchez, et al., 2011), (Gonzalez, et al., 2011), automotive (LoVetri & Makaran, 2001), (Chun, et al., 2014), (Jianwen, et al., 2003) and aerospace applications (Huang, et al., 2006), (Huang, et al., 2012), (Huang, et al., 2012). BLDC machines have also superseded induction machines in recent years with the use of high-energy permanent magnets instead of the common coil rotors, which significantly reduces the machine's overall weight and volume for a given power, thus leading to higher power density. Moreover, the absence of the rotor copper windings offers higher overall efficiency and better dynamic response (Chau, 2015). This further helped BLDC based drives in penetrating the market of low power home appliances and HVAC systems. BLDCs offer a number of improvements in such applications compared to the previous generation of electrical machines such as high efficiency, silent operation, compact form and high reliability (Imoru & Tsado, 2012), (Kim, et al., 2016).

## 2.3 Realization of BLDC Machine Drives

### 2.3.1 Introduction to BLDC Drives

A conventional BLDC machine drive is implemented via an ESC or a SSTP inverter, a drive circuit to switch ON and OFF inverter switches, and a microcontroller circuit to generate the necessary signals for the driving circuit (Joice, et al., 2013). The control signals are calculated depending on feedback obtained from rotor position sensors such as encoders or tachogenerators and phase current sensors. The control signals are then processed by a controller system such as a microcontroller or microprocessor. From a pre-programmed control scheme, the controller then generates the required output signals to control the SSTP inverter. Figure 2.1 shows a high-level diagram of a BLDC machine drive.



**Figure 2.1: High Level Diagram of a BLDC Machine Drive**

### 2.3.2 Inverter Circuits

Although currently the most common BLDC machine inverter used is the SSTP inverter as used in (Kumar & Singh, 2016), (Singh, et al., 2015), (Horvat, et al., 2014) four switch three phase (FSTP) inverters are becoming more popular due to the overall drive cost minimization (Xia, et al., 2015), (Lin, et al., 2006), (Shanmugapriya & Michael, 2012). In (Xia, et al., 2014) a boost five-switch three phase (BFSTP) topology for BLDC machine drives is also presented. The implementation of such a drive includes the combination of a FSTP inverter and a boost circuit. The utilization of the dc link voltage with FSTP inverter is only half of that with the SSTP inverter and therefore load and speed ranges of the drive is greatly restricted without an independent boost circuit (Xia, et al., 2014). This topology has limited practical application especially in solar and battery powered systems. Furthermore, since the phase C winding of the FSTP inverter is directly connected to the midpoints of capacitors on the side of the power supply, three phase currents are likely to be distorted by the phase C back-emf if a common bipolar switching modulation is employed. Inverters equipped with independent boost are available; however this significantly increases the size, cost and complexity of the converter. From the proposed circuit the negative effect of phase C back-emf is restricted and the voltage across capacitors on the side of the dc link is boosted. This allows for constant switching frequency, reduced current ripple and improved load and speed ranges under low power supply voltage (Xia, et al., 2014).

## 2.4 Control of BLDC Machines

### 2.4.1 Introduction to Control of BLDC Machines

The most common control technique for BLDCs is the six-step commutation sequence. This control technique is commonly used with low-cost ESC with low to medium power BLDC machines, as used in (Pola & Vittal, 2006), (Firmansyah, et al., 2014). Other control methods include the Direct Torque Control (DTC) as discussed in (Girija & Prince, 2014), (Nair, et al., 2016) and the RFOC which is the method proposed in this dissertation.

### 2.4.2 Six Step Commutation

The six-step commutation sequence is the simplest BLDC machine control technique. The name six-step comes from the six switching activation combinations possible from a SOTP inverter excluding the zero-volt voltage combinations (Firmansyah, et al., 2014). According to the rotor position angle, usually obtained from three hall-effect sensors positioned at  $120^0$  from each other, six switching patterns are achieved. Executing the six switching patterns on a SOTP inverter would move the BLDC machine one mechanical revolution. Such a control technique has limited application use because of its poor dynamic response and efficiency (Rau, et al., 2015) (Abdulmelik & Hassan, 2016).

### 2.4.3 Direct Torque Control

The first reference in literature about DTC was made by Takahashi (Takahashi, 1986) and was widely accepted as an alternative to RFOC. In DTC scheme, the stator flux linkage and electromagnetic torque can be directly controlled by the selection of optimum inverter switching states. The flux and torque errors are kept within acceptable limits by hysteresis controllers.

In (Joseph & Sreethumol, 2015) the control of a BLDC machine is investigated using the DTC scheme. Torque ripple in BLDC machines is a major issue considering their demand for high performance applications (Jahns & Soong, 1996), (Murai, et al., 1989), thus a DTC scheme on a BLDC drive is being proposed to reduce such torque ripple. In DTC scheme the selection of optimum inverting switching states allows for direct control of the stator flux linkage and electromagnetic torque. Furthermore, two level hysteresis controllers are used to keep the torque and flux deviations

within a predefined value. From the simulation results proposed by (Joseph & Sreethumol, 2015) a good speed response and low torque ripple is achieved by the BLDC machine with the use of DTC scheme.

A DTC technique with non-sinusoidal back-emf using a FSTP inverter is proposed in (Ozturk, et al., 2009) where torque hysteresis controllers are implemented to reduce the distorted currents and smooth electromagnetic torque is produced by controlling the torques on phases A and B independently. Furthermore, a look-up table for two-phase voltage selection was designed to provide faster torque response. For effective torque control a new switching pattern incorporated with the designed look-up table was developed and implemented to produce the desired torque characteristics (Ozturk, et al., 2009). Results obtained from the simulation and experiment show that it is possible to achieve two-phase conduction DTC of a BLDC machine drive using the FSTP inverter (Ozturk, et al., 2009).

#### **2.4.4 Rotor Flux Oriented Control**

The first reference in literature to RFOC was made by (Blaschke, 1972). Since then such a control scheme changed drastically. In RFOC the three-phase currents of the machine are transformed into the d-q current components through the use of Clarke and Park transforms and controlled by PI controllers. The stator current in the q-axis  $i_q$  is proportional to the torque generated by the BLDC while the stator current in the d-axis  $i_d$  controls the flux generation.

In (Rau, et al., 2015) the presentation of a BLDC machine controller using the RFOC is proposed. They discuss that the most common BLDC control method is the six-step commutation strategy however; such a control strategy has poor dynamic response and poor efficiency which limits its use to applications where control is of a non-critical nature or where the control of the overall system is dependent on other actuators. The RFOC method comprises the use of the Clarke transformation in order to adjust the motor into a two-phase system and the Park transformation to change the stator reference frame into a rotor reference frame. From such transformations, important BLDC parameter quantities were transformed from a stationary frame of reference to a rotating one. This allows for the design of speed and current controllers similar to that of a DC machine. Current

and speed loops were implemented in cascaded form and the closed-loop response tuned with PI controllers. Cascade control allows for an angle equal to ninety degrees between the rotor flux vector and the stator flux vector thus allowing maximum torque generation (Rau, et al., 2015). From the analysis presented in (Rau, et al., 2015) it is shown that an RFOC method on a BLDC machine improves dynamical behavior when compared with the six-step commutation strategy.

In (Abdulmelik & Hassan, 2016) the simulation of a five level inverter using the RFOC method is discussed. They argue that the conventional control of a BLDC machine using the six-step or trapezoidal control method causes high pulsating torque ripple and poor speed performance in both steady and transient states. Furthermore, (Abdulmelik & Hassan, 2016) also discuss that the implementation of two-level inverters when compared with the implementation of multi-level inverters, causes greater harmonic distortion in the output waveform as further discussed in more detail in (Mashhadany, 2015), (Rao, et al., 2013) and (Rao, et al., 2013). The proposed method consists of a five-level cascaded H-bridge inverter were gate signals are provided by a level-shifted phase disposition Sine Pulse Width Modulation (SPWM). (Abdulmelik & Hassan, 2016) concluded that the proposed control method reduces the torque ripple and improves speed loop performance in both steady and transient states compared to the traditional trapezoidal control method.

Both methods are based on the same concept that transforms the machine's three-phase variables into two-phase variable orthogonal components. These components are typically proportional to the flux and torque of the machine; hence the control of the three phase machine is simplified into a topology which is similar to that commonly found in DC machines. However, differences in the two control topologies exists which results in different torque/speed ripple and power converter lifetime as discussed in (Korkmaz, et al., 2013). From the numerical analysis performed in (Korkmaz, et al., 2013), it can be shown that the RFOC scheme has a faster speed dynamic response and less torque ripples suggesting that a better overall performance is achieved in RFOC when compared to the DTC method. However, the DTC scheme has switching harmonics 80% lower than the RFOC scheme, which suggests longer inverter lifetime when using DTC instead of RFOC.

## 2.5 Sensorless Implementation

### 2.5.1 Introduction to Sensorless Control of BLDC Machines

As mentioned in Section 2.3.1, in order to control the BLDC machine, information about the rotor angle must be known at all times. This can be achieved through dedicated sensors such as encoders, tachogenerators and hall-effect sensors. Such an approach is however not feasible because of the limited application area and additional costs incurred. Thus, BLDC machines are commonly combined with sensorless speed/position algorithms. Sensorless algorithms are subdivided into two main categories: model based and non-model based. As the name itself suggests, model based sensorless algorithms estimate the speed/position of the machine through an electrical and/or mechanical model of the machine and are better suited for high-speed operation (Kim, et al., 2011), (Parasiliti, et al., 2001), (Kye-Lyong, et al., 2004). Non-model based algorithms are used at close to zero speeds and rely on the injection of additional carrier signals to the fundamental ones required for control purposes (Staines, et al., 2014), (Scicluna, et al., 2016). The method proposed in this dissertation is a model-based algorithm with back-emf estimation.

### 2.5.2 Conventional Speed/Position Algorithm as used in ESCs

In (Lizuka, et al., 1985) the control of a BLDC machine without a rotor position sensor was studied. Back-emf voltage sensing at steady state operation was used in this research. Lizuka et al. argued that position sensors in BLDC machines reduce ruggedness and complicates overall machine configuration. Furthermore, the use of BLDC machines in immersed or sealed applications limits the use of sensors and feedback cables in the construction of the BLDC machine and its drive. In the proposed control algorithm at every commutation step, one phase is connected to the positive supply voltage, one phase is connected to the negative supply voltage and one phase is floating. The back-emf in the floating phase will result in a zero crossing when it crosses the average of the positive and negative supply voltage. The detection of the zero crossing points helps in determining the coil energizing sequence. The back-emf voltage from the stator windings is detected by applying the appropriate signal conditioning circuits and feeding the output of the circuit to a microcontroller. From the results presented, it was shown that such a sensorless control technique works well at

medium to high-speed ranges. However, since the back-emf is directly proportional to the motor speed, it is extremely difficult to detect the zero crossings at low speed, as the signal to noise ratio would be very small. Therefore, such sensorless control technique is unsuitable for low speed applications.

### 2.5.3 Model Based Speed/Position Algorithms

In (Zambada & Deb, 2010) the implementation of a senseless RFOC of a permanent magnet synchronous machine (PMSM) was studied. In RFOC it is very important to know the electrical angle in order to calculate the necessary transformations. In sensorless RFOC the electrical angle is calculated by estimating the back-emf. The back-emf is estimated through a current observer (Zambada & Deb, 2010). The current output from the BLDC machine and the current output from the mathematical model of the BLDC machine are subtracted to generate an error signal. A sliding mode controller (SMC) is then used to correct the error signal. When the actual current measured is equal to the estimated current it can be assumed that the estimated back-emf is the same as the actual back-emf on the motor (Zambada & Deb, 2010). The electrical angle is then calculated from the inverse tangent of the ratio between the two estimated back-emfs typically in the alpha-beta frame.

In (Rau, et al., 2015) a model based speed/position algorithm with back-emf estimation using the RFOC control technique was proposed. The proposed method of back-emf estimation operates in the stator reference frame therefore the rotor position angle can be obtained from the stator reference frame currents  $i_\alpha$  and  $i_\beta$ . A simplified model of the machine is designed from which an estimated current value for both  $i_\alpha$  and  $i_\beta$  is obtained. The estimated stator reference frame currents are compared with the actual stator reference frame currents from which an error is calculated. The error is then fed to a PI controller which outputs estimates of the back-emfs  $BEMF\alpha$  and  $BEMF\beta$ . Speed estimation was obtained from a phased locked loop (PLL).

## 3. Theory

### 3.1 Introduction to Theory

The proposed control algorithm presented in this chapter to control the BLDC machine is the RFOC technique. Such a control technique was also used in (Lazor & Stulrajter, 2014), (Itoh, et al., 2002) and (Rau, et al., 2015). Throughout this chapter the mathematical model of a BLDC machine was derived. Furthermore, the design of the inner (current) and outer (speed) PI controllers and the design of a back-emf observer are discussed.

### 3.2 Mathematical Modelling of a BLDC Machine

In order to obtain the mathematical model of a BLDC machine, a set of mathematical operations are required to convert between the abc,  $\alpha\beta$  and dq frames. Once these equations are applied correctly modeling and design of controllers is similar to that of a DC machine.

A three-phase system is converted into a two-axis system by using the Clarke transform:

$$\begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \end{bmatrix} \quad (3.1)$$

The two-axis system is rotated to align with the rotor flux by using the Park transform:

$$\begin{bmatrix} X_d \\ X_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_\alpha \\ X_\beta \end{bmatrix} \quad (3.2)$$

In design by RFOC it is assumed that the stator flux linkage due to the rotor flux is aligned to the d-axis such that:

$$\varphi_r = \varphi_{rd} \text{ and } \varphi_{rq} = 0 \text{ Wb}$$

For the BLDC machine the stator flux linkage in the dq frame is:

$$\begin{aligned}\varphi_{sd} + j\varphi_{sq} &= (L_d i_{sd} + \varphi_{rd}) + j(L_q i_{sq} + \varphi_{rq}) \\ \varphi_{sd} + j\varphi_{sq} &= (L_d i_{sd} + \varphi_{rd}) + jL_q i_{sq}\end{aligned}\quad (3.3)$$

Considering the stator voltage in the  $\alpha\beta$  frame:

$$v_{\alpha\beta} = R i_{s\alpha\beta} + L \frac{di_{s\alpha\beta}}{dt} + bemf_{\alpha\beta} \quad (3.4)$$

(3.4) can be simplified to:

$$v_{\alpha\beta} = R i_{s\alpha\beta} + \frac{d\varphi_{s\alpha\beta}}{dt} \quad (3.5)$$

If  $X_{dq} = X_{\alpha\beta} e^{j\theta}$ , (3.5) becomes:

$$\begin{aligned}v_{\alpha\beta} e^{j\theta} &= R i_{s\alpha\beta} e^{j\theta} + \frac{d}{dt} [\varphi_{s\alpha\beta} e^{j\theta}] \\ v_{\alpha\beta} e^{j\theta} &= R i_{s\alpha\beta} e^{j\theta} + \frac{d}{dt} [\varphi_{s\alpha\beta} e^{j\theta}] + j\omega e^{j\omega t} \varphi_{s\alpha\beta} \\ v_{sdq} &= R i_{sdq} + \frac{d}{dt} \varphi_{sdq} + j\omega \varphi_{sdq}\end{aligned}\quad (3.6)$$

Resolving (3.6) in d and q components gives:

$$v_{sd} = R i_{sd} + \frac{d}{dt} \varphi_{sd} - \omega \varphi_{sq} \quad (3.7)$$

$$v_{sq} = R i_{sq} + \frac{d}{dt} \varphi_{sq} + \omega \varphi_{sd} \quad (3.8)$$

Resolving (3.3) in d and q components gives:

$$\varphi_{sd} = L_d i_{sd} + \varphi_{rd} \quad (3.9)$$

$$\varphi_{sq} = L_q i_{sq} \quad (3.10)$$

Substituting (3.9) and (3.10) in (3.7) gives:

$$\begin{aligned}v_{sd} &= R i_{sd} + \frac{d}{dt} [L_d i_{sd} + \varphi_{rd}] - \omega L_q i_{sq} \\ v_{sd} &= R i_{sd} + L_d \frac{di_{sd}}{dt} + \frac{d\varphi_{rd}}{dt} - \omega L_q i_{sq} \\ v_{sd} &= R i_{sd} + L_d \frac{di_{sd}}{dt} - \omega L_q i_{sq}\end{aligned}$$

$$i_{sd} = \frac{1}{L_d} \int v_{sd} - R_{sd} + \omega L_q i_{sq} dt \quad (3.11)$$

Substituting (3.9) and (3.10) in (3.8) gives:

$$\begin{aligned} v_{sq} &= R_{sq} + L_q \frac{di_{sq}}{dt} + \omega(L_d i_{sd} + \varphi_{rd}) \\ v_{sq} &= R_{sq} + L_q \frac{di_{sq}}{dt} + \omega L_d i_{sd} + \omega \varphi_{rd} \\ i_{sq} &= \frac{1}{L_q} \int v_{sq} - R_{sq} + \omega L_d i_{sd} - \omega \varphi_{rd} dt \end{aligned} \quad (3.12)$$

The total torque output from a BLDC machine can be expressed as:

$$T_e = 1.5 p i_{sq} \varphi_{rd} \pm T_l \quad (3.13)$$

The speed  $\omega$  can be determined from:

$$T_e = J\dot{\omega} + B\omega \quad (3.14)$$

Applying the Laplace transform on (3.14) gives:

$$\begin{aligned} T_e(s) &= Js\omega(s) + B\omega(s) \\ T_e(s) &= \omega(s)[sJ + B] \\ \omega(s) &= \frac{1}{(sJ + B)} T_e(s) \end{aligned} \quad (3.15)$$

The angle  $\theta$  can be determined by:

$$\int \omega dt = \theta \quad (3.16)$$

### 3.3 Parameters Measurements and Calculations

In order to simulate the model of the BLDC machine, parameters had to be either measured or estimated through appropriate calculations. This section describes how the machine's phase and synchronous resistances and inductances, moment of inertia, viscous coefficient of friction, no. of poles, rated current, rated torque and the rotor flux aligned on the d-axis needed for the simulation were measured or calculated.

### 3.3.1 Resistance and Inductance Measurement

The phase resistance and inductance were measured by using an Inductance Capacitance Resistance (LCR) meter at a frequency of 10 kHz. The meter was connected between two of the three phases of the machine and the values for resistance and inductance were noted. The measured value for the resistance was found to be  $0.163 \Omega$  and the measured value for the inductance was found to be  $13 \mu\text{H}$ . Since the measured value is across two phases both values were divided by two. Thus, the phase resistance and inductance of the BLDC machine are  $0.0815 \Omega$  and  $6.5 \mu\text{H}$  respectively. For simplicity all the phase resistances and inductances were assumed to be equal to each other and also equal to the two-axis dq frame resistance and inductance, therefore,  $R_d = R_q = 0.0815 \Omega$  and  $L_d = L_q = 6.5 \mu\text{H}$ . This assumes that the BLDC motor to be used in practice is a non-salient machine.

### 3.3.2 Moment of Inertia and Viscous Coefficient of Friction Approximation

The approximation for the machine moment of inertia was determined by assuming the BLDC machine to be a solid cylinder or radius “r” and mass “m”. The equation to determine the moment of inertia for a solid cylinder is:

$$J = \frac{mr^2}{2} \quad (1.17)$$

Parameters “r” and “m” were measured to be 0.015 m, and 0.065 kg, therefore:

$$J = \frac{(0.065)(0.015^2)}{2}$$

$$J = 7.312e^{-6} \text{ kgm}^2$$

The viscous coefficient of friction is determined from the moment of inertia by the rule of thumb:

$$B = \frac{J}{10} \quad (3.18)$$

$$B = \frac{0.000007312}{10}$$

$$B = 7.312e^{-7}$$

### 3.3.3 Determining the machine's no. of poles

The number of poles of the BLDC machine was determined by applying the following test on the BLDC machine (Bobek, 2013):

- From a bench power supply a voltage of +11 V and GND was connected to two of the three phases of the BLDC machine. For safety reasons the output current from the bench power supply was limited to 2 A.
- The +11 V and GND from the power supply energized the stator of the machine causing the permanent magnet rotor to move to the closest pole pair. The rotor was then slightly pushed outside the current pole pair it was stationed at in order to move itself to the next pole pair.
- The above procedure is repeated for one full turn of the machine's rotor. If "n" number of pushes were needed to turn one full turn of the machine's rotor, then that means that there are "n" number of pole pairs. Therefore, the number of poles should be "2n".

From the above test it was determined that the BLDC machine has 14 poles because 7 pushes were needed to turn one full rotation of the machine's rotor.

### 3.3.4 Calculation of Rated Current, Rated Torque and $\varphi_{rd}$

Rated power, voltage and speed values for the BLDC machine were obtained from the BLDC machine datasheet. They were found to be 187 W, 11 V and 12430 rpm respectively. The rated current can then be calculated from such parameters as:

$$I_{\text{rated}} = \frac{P_{\text{rated}}}{3V_{\text{rated}}} \quad (3.19)$$

$$I_{\text{rated}} = \frac{187}{(3)(11)}$$

$$I_{\text{rated}} = 5.67 \text{ A}$$

The rated torque can be calculated by using the below equation:

$$T_{\text{rated}} = \frac{P_{\text{rated}}}{\omega_{\text{rated}}} \quad (3.20)$$

$$T_{\text{rated}} = \frac{P_{\text{rated}}}{2\pi \frac{\omega_{\text{rpm}}}{60}} \quad (3.21)$$

$$T_{\text{rated}} = \frac{187}{(2)(\pi) \left( \frac{12430}{60} \right)}$$

$$T_{\text{rated}} = 0.1437 \text{ Nm}$$

The value of  $K_t$  can be determined from the rated torque and rated current as:

$$K_t = \frac{T_{\text{rated}}}{I_{\text{rated}}} \quad (3.22)$$

$$K_t = \frac{0.1437}{5.67}$$

$$K_t = 0.02534 \text{ Nm/A}$$

Since the reference input for current  $i_{sd} = 0$ , then  $I_{\text{rated}} = I_{sq}$  which gives  $\varphi_{rd}$  as:

$$\varphi_{rd} = \frac{T_{\text{rated}}}{3pI_q} \quad (3.23)$$

$$\varphi_{rd} = \frac{0.1437}{(3)(14)(5.67)}$$

$$\varphi_{rd} = 0.0012 \text{ Wb}$$

The inputs  $v_d$  and  $v_q$  for the BLDC machine are the outputs of the current PI controllers  $i_{sd}$  and  $i_{sq}$  respectively. The implementation of such controllers is discussed in Section 3.4.

## 3.4 PI Controllers Design

### 3.4.1 Introduction

Three PI controllers were designed to control currents  $i_{sq}$  and  $i_{sd}$  and speed  $\omega$ . Current  $i_{sq}$  controls the torque output of the machine, current  $i_{sd}$  controls the rotor magnetizing flux and speed  $\omega$  controls the speed of the rotor shaft. The reference input for current in the d-axes  $i_{sd} *$  was set to constant 0 A, the reference input for q-axis current  $i_{sq} *$  is determined from the speed PI controller the reference input for speed  $\omega *$  is chosen by the user. The designed PI controllers are in cascaded form where PI controllers for  $i_{sq}$  and  $i_{sd}$  are set as the inner loop PI controllers while the PI controller for speed  $\omega$  is set as the outer loop PI controller. The PI controller for current  $i_{sq}$  controls the torque generated for the machine while the PI controller for current  $i_{sd}$  controllers the magnetising flux of the machine. Figure 3.1 shows the control system for the RFOC algorithm together with the rotor position and speed estimation block.

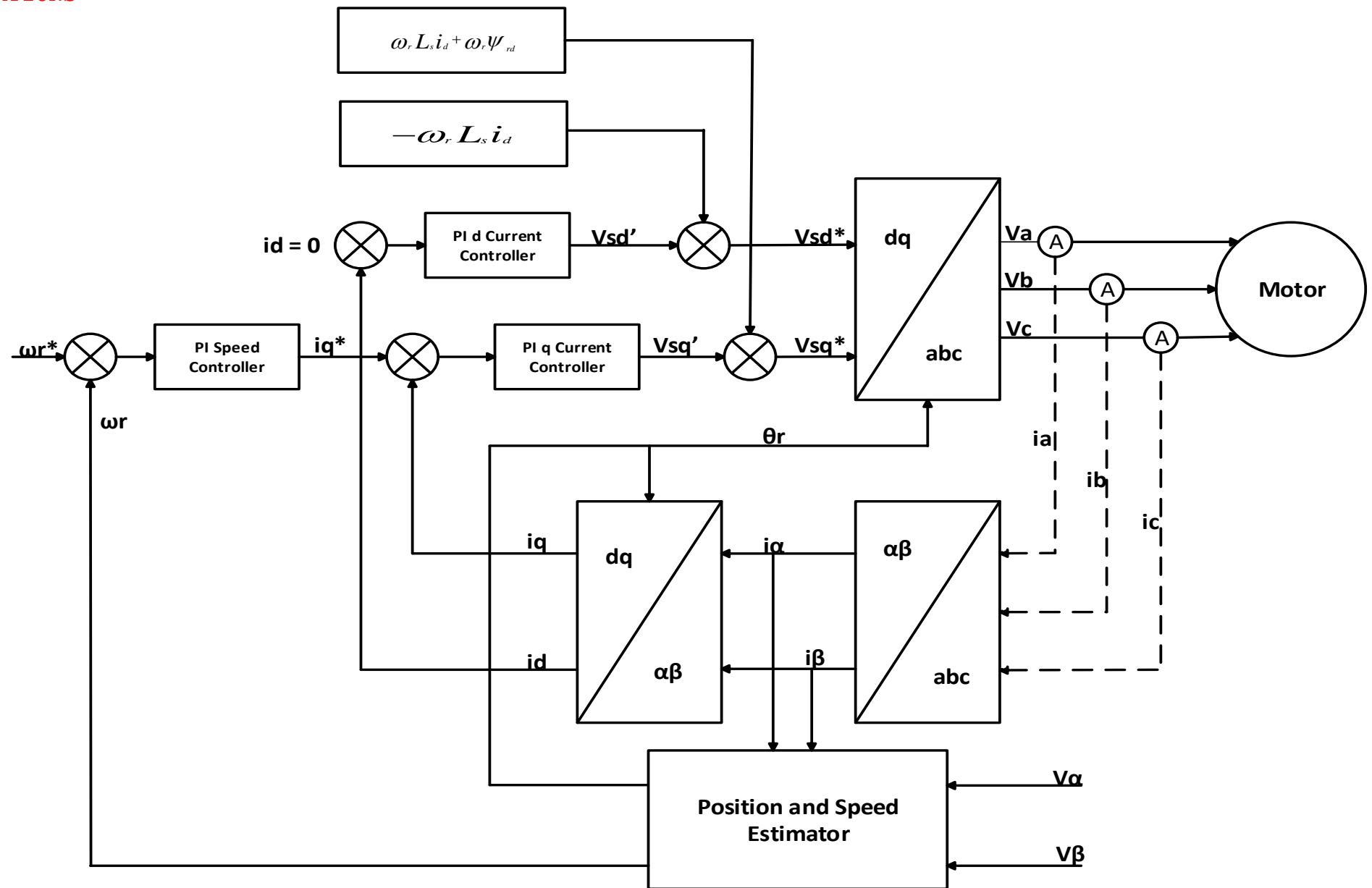


Figure 3.1: RFOC with rotor position estimation control diagram

### 3.4.2 Currents PI Controller Design

The electrical dynamics (3.11 – 3.12) of the BLDC in the synchronous dq-frame where derived in Section 3.2 and are repeated below. The equations were used to model the BLDC machine in Simulink/MATLAB.

$$v_{sd} = R_{sd} + L_d \frac{di_{sd}}{dt} - WL_q i_{sq}$$

$$v_{sq} = R_{sq} + L_q \frac{di_{sq}}{dt} + WL_d i_{sd} + W\varphi_{rd}$$

In order to obtain a Single Input Single Output (SISO) transfer function on which Laplace transforms can be applied, the cross-coupling terms of the equations above are eliminated and added into the control algorithm as feedforward terms. Therefore, the above equations are simplified to:

$$v_{sd} = R_{sd} + L_d \frac{di_{sd}}{dt} \quad (3.24)$$

$$v_{sq} = R_{sq} + L_q \frac{di_{sq}}{dt} \quad (3.25)$$

Since  $R_d = R_q$  and  $L_d = L_q$  the frequency response in both the d and q axes will be the identical.

Applying the Laplace transform on (3.24) gives:

$$\begin{aligned} V_{sd}(s) &= RI_{sd}(s) + sL_d I_{sd}(s) \\ \frac{I_d(s)}{V_d(s)} &= \frac{1}{L_d s + R} \end{aligned} \quad (3.26)$$

From the parameters obtained in Section 3.3, (3.26) becomes:

$$\frac{I_d(s)}{V_d(s)} = \frac{1}{6.5e^{-6}s + 0.0815}$$

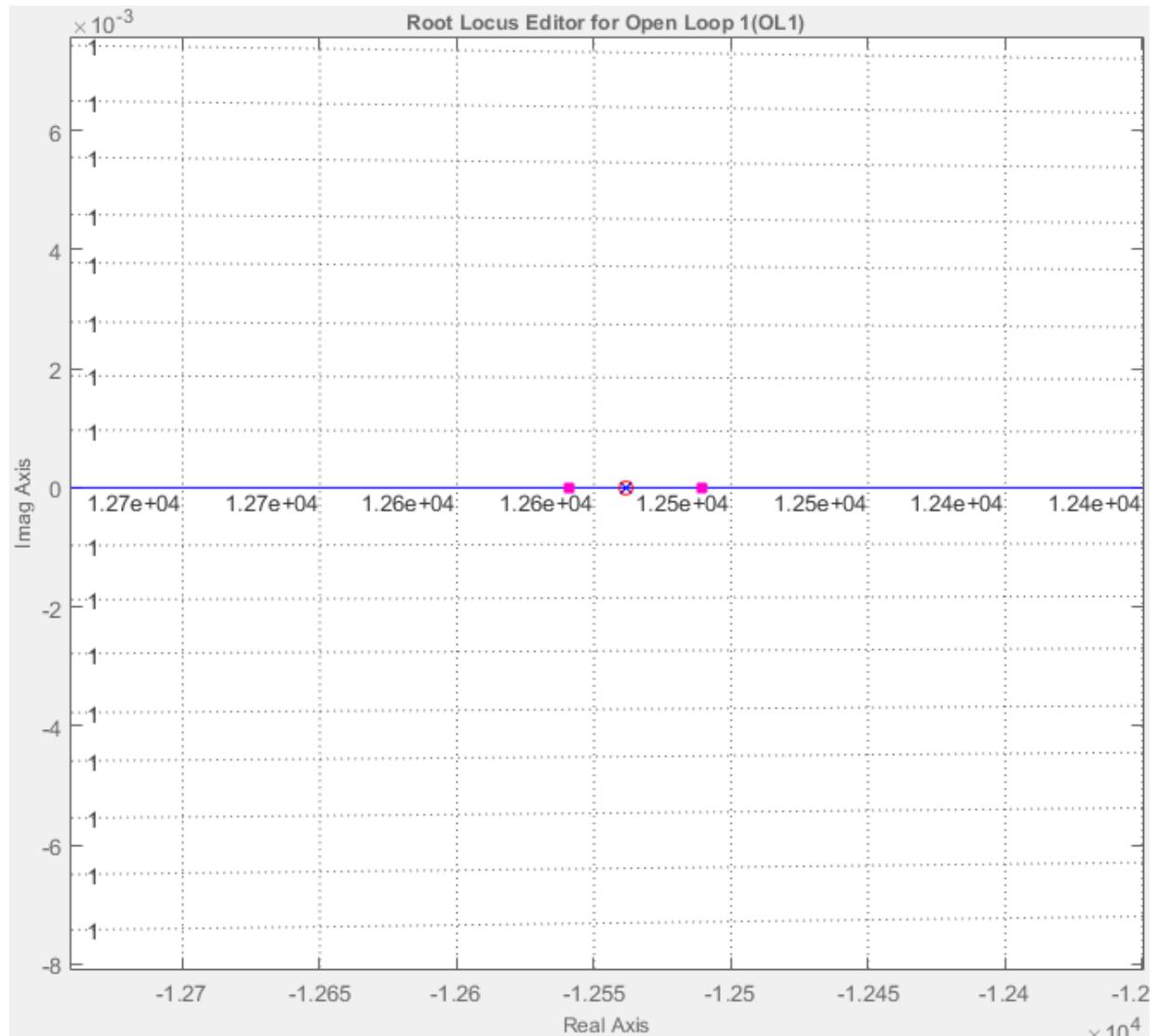
The general equation for a PI controller is defined as follows:

$$\begin{aligned} PI &= K_p + \frac{K_i}{s} \\ PI &= K_p \frac{s + \frac{K_i}{K_p}}{s} \end{aligned} \quad (3.27)$$

Gain values for  $K_p$  and  $\frac{K_i}{K_p}$  are to be calculated in order to design the PI controller. The value for gain

$\frac{K_i}{K_p}$  introduces a zero in the overall closed loop transfer function, while gain  $K_p$  sets the closed loop

poles position. If the value for gain  $\frac{K_i}{K_p}$  is designed in such a way that the zero introduced by the gain cancels out the pole of  $\frac{I_d(s)}{V_d(s)}$ , the overall response of the system will be first order and therefore no peaking or oscillations will result on the output response of the system. If the zero is not designed in such a way as to cancel the pole of  $\frac{I_d(s)}{V_d(s)}$  the overall system response would be second order and therefore peaking and oscillations would occur during any current transition period which could possibly deteriorate the dynamic performance. Hence, the gain values were set for  $\frac{K_i}{K_p}$  equal to  $\frac{R_q}{L_q} = \frac{0.0815}{6.5e^{-6}} = 12538.46154$  in order to achieve pole zero cancellation. Figure 3.2 shows the added real zero introduced by the PI controller placed on top of the  $\frac{I_d(s)}{V_d(s)}$  open loop pole.



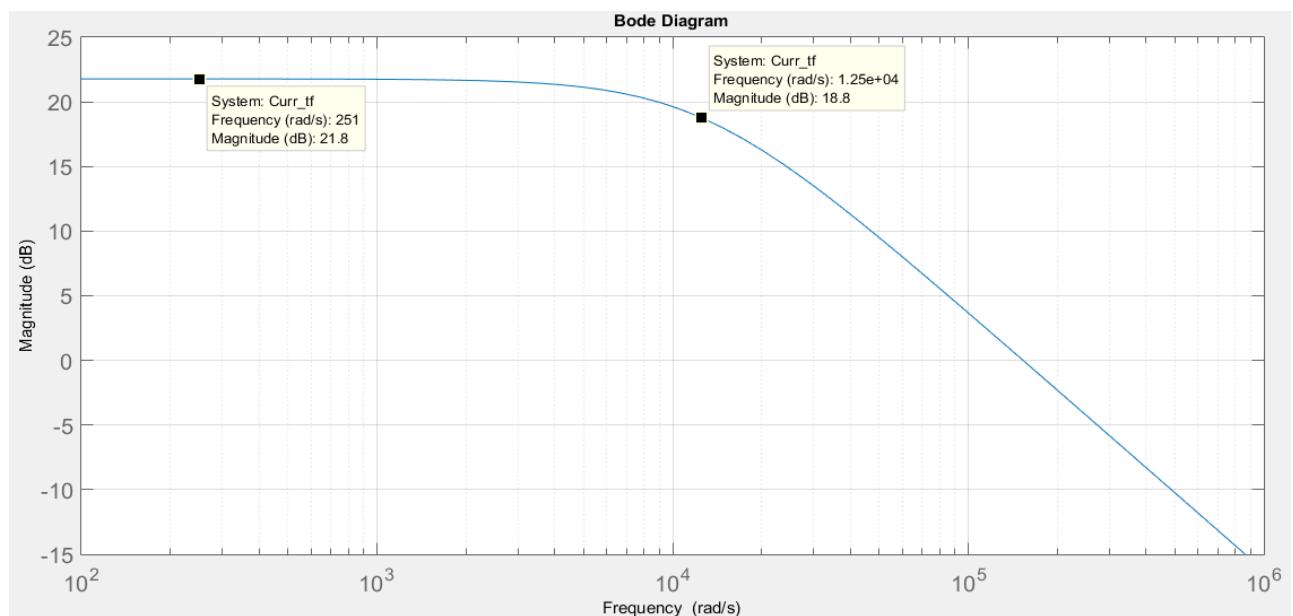
**Figure 3.2: PI zero set on system's pole in order to obtain pole zero cancelation**

The value for gain  $K_p$  is set to be large enough in order to decrease any error between the desired output and the current output from  $\frac{1}{1+GH}$ . However, it would be impossible to have high gain for all frequencies since at some point such high gain would introduce a phase delay in the system because every system in nature would have another pole in its transfer function at such high frequency which could distort or even put the system unstable. A good design practice for choosing gain  $K_p$  for the current controller design is to set gain  $K_p$  equal to the inductance  $L_q$  multiplied by the closed loop bandwidth. Figure 3.3 shows the bode plot of the closed loop transfer function  $\frac{I_d(s)}{V_d(s)}$  where the system's bandwidth is found at -3dB from the highest dB point.

$$21.8 \text{ dB} - 3 \text{ dB} = 18.8 \text{ dB}$$

$$\text{At } 18.8 \text{ dB, } W_n = 12500 \text{ rad/s}$$

$$\therefore K_p = 6.5e^{-6} * 12500 = 0.08125$$



**Figure 3.3: Bode plot for the current closed loop transfer function**

Therefore, the current PI controller is:

$$PI_{Current}(s) = 0.08125 \frac{s + 12538.46154}{s}$$

Where in parallel PI controller form:

$$K_p = 0.08125$$

$$K_i = 1018.75$$

### 3.4.3 Speed PI Controller Design

The only important design requirement for the outer PI controller (speed PI controller) is to have a bandwidth requirement which is at least 10-20 times minimum slower than the inner loop (current PI controller). Recalling that the closed loop bandwidth of the inner PI controller was 12500 rad/s, if the bandwidth for outer speed controller is set at 625 rad/s i.e. 20 times slower, it should be satisfactory since  $12500 \text{ rad/s} \gg 625 \text{ rad/s}$ . Therefore, if the desired closed loop bandwidth is 625 rad/s and  $\zeta$  is chosen to be 0.707 then a speed PI controller with such transient requirements can be designed in sisotool on MATLAB.

The speed controller design is determined from (3.15) from Section 3.2 which was:

$$\omega(s) = \frac{1}{(sJ + B)} T_e(s)$$

From the parameters obtained in Section 3.3, (3.15) becomes:

$$\frac{\omega(s)}{T_e(s)} = \frac{1}{(7.312e^{-6}s + 7.312e^{-7})}$$

Using the sisotool in MATLAB the root locus for (3.15) was plotted. Inputting the desired speed transient requirements i.e.  $\zeta = 0.707$  and  $W_n = 625 \text{ rad/s}$  an intersection point on the s-plane is obtained on which the closed loop poles should be placed. The values for the PI controller gains can be designed from the MATLAB toolbox as shown in Figure 3.4. Modifying the integral gain changes the path of the closed loop gains, while modifying the proportional gain changes the position of the closed loop poles on the path. The designed PI controller places the closed loop poles on the intersection point as seen in Figure 3.5.

Therefore, the speed PI controller is:

$$PI_{speed}(s) = 0.25223 \frac{s + 443.35}{s}$$

Where in parallel PI controller form:

$$K_p = 0.25223$$

$$K_i = 111.826$$

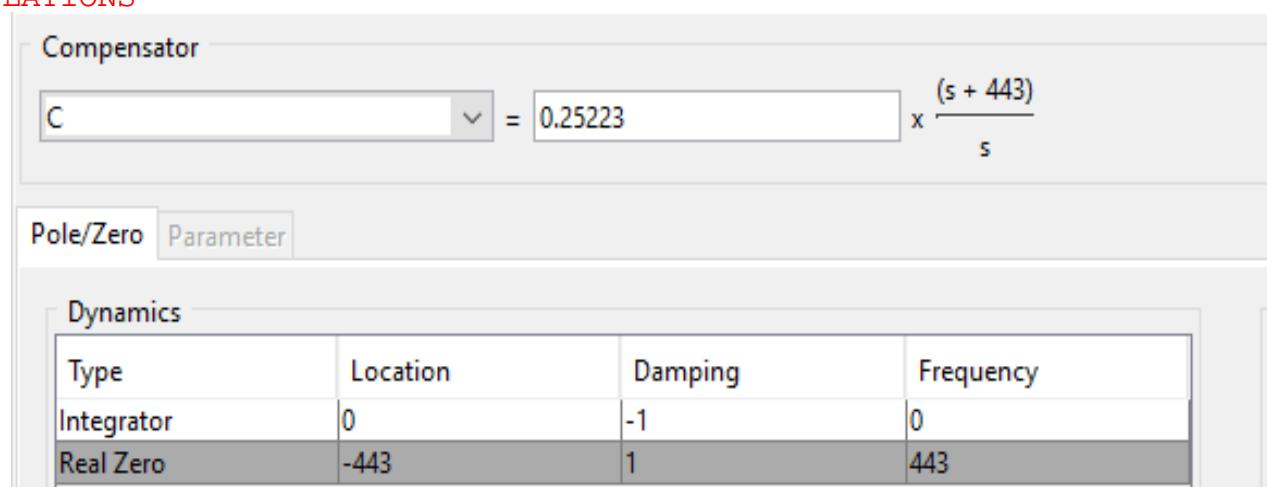


Figure 3.4: Setting of speed PI controller gains

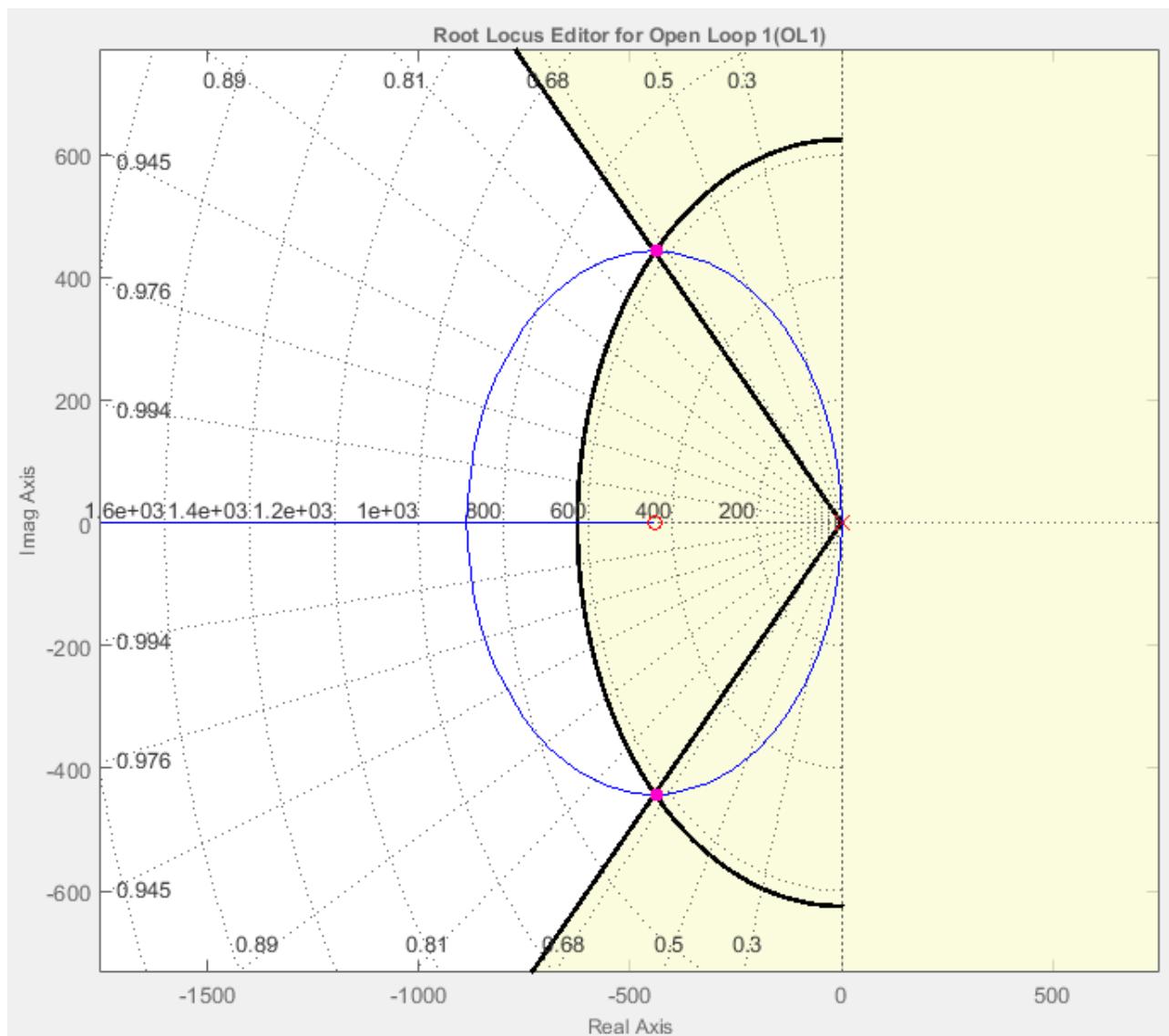


Figure 3.5: Root locus design for the speed PI controller

## 3.5 Observer Design

### 3.5.1 Introduction

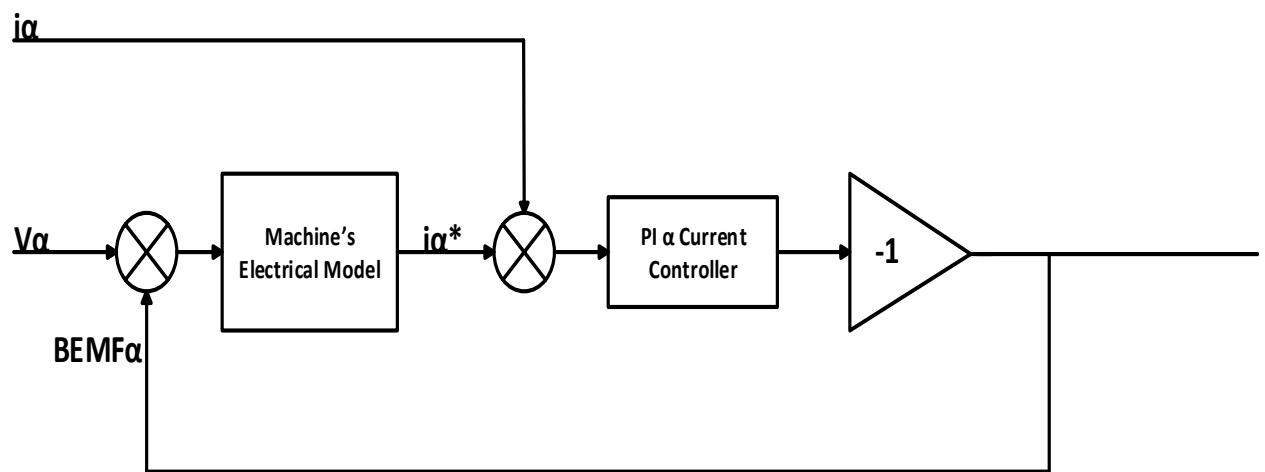
When a three-phase supply is supplied to the machine's stator, a rotating stator magnetic field is produced. The opposing poles on the permanently magnetized rotor and on the rotating stator attract each other; hence, the shaft of the BLDC rotates. The stator and rotor fields must not be aligned as otherwise the rotor remains stationary. For machine rotation, an angle between the two fields exists; this angle contributes to the output torque generation of the machine as a sinusoidal function. This means that if the angle between the stator magnetic field and the rotor magnetic field is  $0^\circ$ , then  $\sin(0^\circ)$  would be equal to 0 i.e. no torque generation. However, if the angle between the stator magnetic field and the rotor magnetic field is  $90^\circ$ , then  $\sin(90^\circ)$  would be equal to 1 i.e. maximum torque generation.

The aim of RFOC is to continually achieve a  $90^\circ$  angle between the stator magnetic field and the rotor magnetic field in order to achieve the maximum torque generation per ampere current flowing through the stator of the machine. Hence, if the rotor magnetic field angle is detected, the controller can generate a stator-rotating magnetic field that meets the  $90^\circ$  offset requirement. Therefore, sensored RFOC requires knowledge of the rotor position at all times. In sensorless RFOC, there is no dedicated sensor that gives rotor positon information and therefore the rotor position needs to be estimated by an observer.

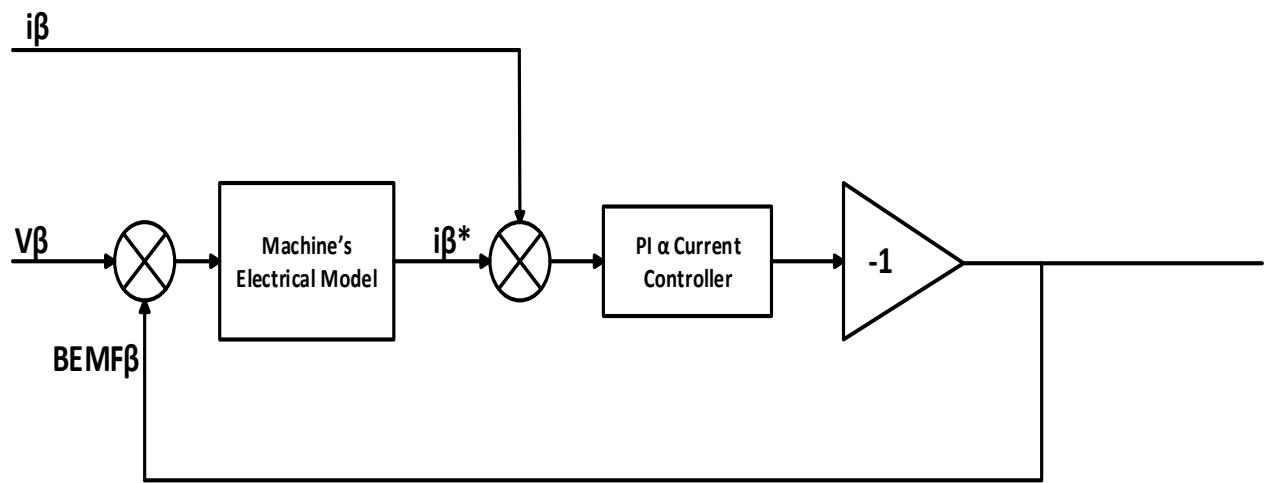
### 3.5.2 Rotor Position Estimation

The estimation of the rotor position  $\theta$  is derived from the estimation of the back-emf quantities in the stationary alpha-beta frame,  $BEMF_\alpha$  and  $BEMF_\beta$ . Each back-emf estimation can be derived from the simulation diagram shown in Figures 3.6 and 3.7 respectively.  $v_\alpha$  is used as an input in the back-emf simulation diagram in order to generate an estimated  $\hat{i}_\alpha$  through a digital representation of the machine's L and R electrical parameters. The estimated  $\hat{i}_\alpha$  is then subtracted from the real  $i_\alpha$  producing an error term which can be corrected by a PI controller whose output is fed back to the input of the observer to be deducted from  $v_\alpha$ . This feedback signal is the representation of the  $BEMF_\alpha$ . Therefore, if the error signal is 0, that means that  $\hat{i}_\alpha = i_\alpha$  which means that the feedback signal i.e.

the back-emf is the current real value of the machine because from  $\frac{v_\alpha - \text{BEMF}\alpha}{L+R}$  produced an  $\hat{i}_\alpha$  which is the same as the real  $i_\alpha$ .



**Figure 3.6: BEMF $\alpha$  estimation diagram**



**Figure 3.7: BEMF $\beta$  estimation diagram**

The rotor position  $\theta$  can be derived as follows:

BEMF $\alpha$  is the horizontal component of the vectorial sum of BEMF $\alpha$  and BEMF $\beta$  i.e.  $E_s$ . Therefore:

$$\sin\theta = \frac{\text{BEMF}\alpha}{E_s} \quad (3.29)$$

$$\text{BEMF}\alpha = E_s \sin\theta$$

BEMF $\beta$  is the vertical component of the vectorial sum of BEMF $\alpha$  and BEMF $\beta$  i.e.  $E_s$ . Therefore:

$$\cos\theta = \frac{\text{BEMF}\beta}{E_s} \quad (3.30)$$

$$\text{BEMF}\beta = E_s \cos\theta$$

Solving for  $\theta$  gives:

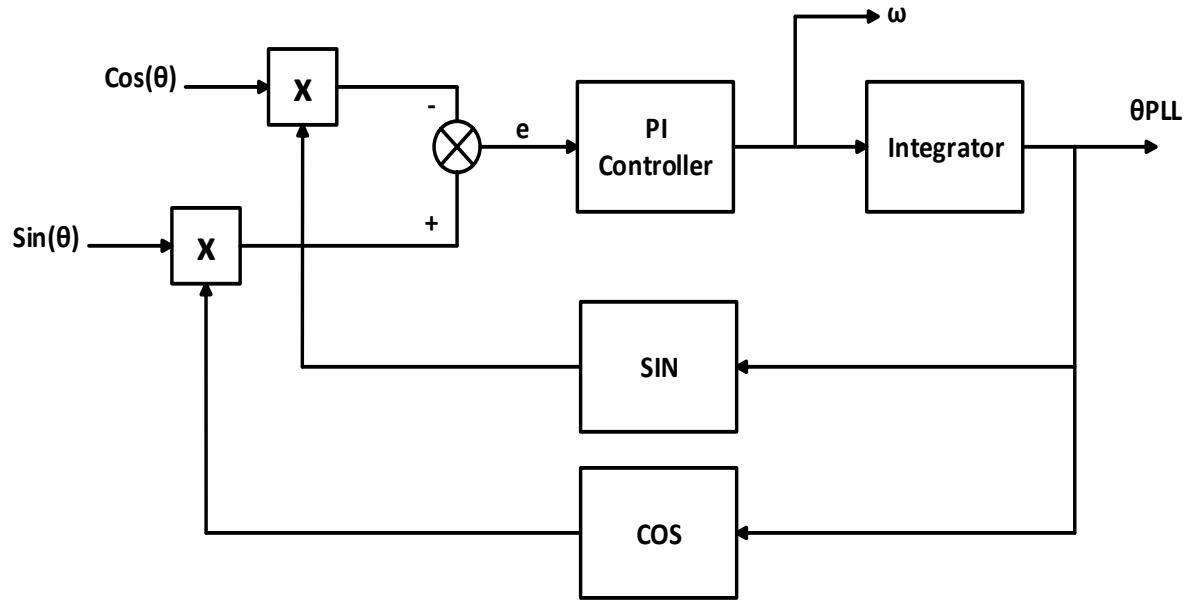
$$\frac{\text{BEMF}\alpha}{\text{BEMF}\beta} = \frac{E_s \sin\theta}{E_s \cos\theta}$$

$$\frac{\text{BEMF}\alpha}{\text{BEMF}\beta} = \frac{\sin\theta}{\cos\theta}$$

$$\frac{\text{BEMF}\alpha}{\text{BEMF}\beta} = \tan\theta$$

$$\theta = \tan^{-1} \left( \frac{\text{BEMF}\alpha}{\text{BEMF}\beta} \right) \quad (3.31)$$

If the rotor's position angle  $\theta$  is replicated via a PLL, information about the rotor's speed can be found within the PLL. Figure 3.8 shows the simulation diagram of the PLL proposed for extracting the rotor's speed information.



**Figure 3.8: Rotor's speed  $\omega$  estimation using a PLL**

Solving for error  $e$ :

$$e = \sin(\theta) \cos(\theta) - \cos(\theta_{\text{PLL}}) \sin(\theta_{\text{PLL}}) \quad (3.32)$$

From the product to sum trigonometric identities, (3.32) becomes:

$$e = \sin(\theta - \theta_{\text{PLL}}) \quad (3.33)$$

For small angles, (3.33) becomes:

$$e \approx \theta - \theta_{\text{PLL}}$$

The PI controller controls error  $e$  to zero with no steady state error in position tracking. The gain of the PI controller determines how fast the PLL can track changes in speed and how much noise (speed change) will be filtered out.

### 3.5.3 PI Controller Design for Back-emf Estimation

The PI controllers of Figures 3.5 and 3.6 used for the estimation of the back-emfs were designed to be nine times faster than the current loop. From Section 3.4.2, the current loop was designed to have a closed loop bandwidth i.e. natural frequency  $\omega_n$  of 12500 rad/s therefore the settling time of the current PI controllers is:

$$T_s \approx \frac{3.9}{\zeta \omega_n} \quad (3.34)$$

$$T_s(\text{Current Loop}) \approx \frac{3.9}{(1)(12500)}$$

$$T_s(\text{Current Loop}) \approx 0.000312s$$

Thus for the observer loop to be nine time faster than the current loop, the observer loop has to have a closed loop bandwidth equal to:

$$T_s(\text{Observer Loop}) \approx \frac{0.000312s}{9}$$

$$T_s(\text{Observer Loop}) \approx 0.000034666s$$

From (3.34) a settling time of 0.000034666s has a natural frequency equal to:

$$\omega_n \approx \frac{3.9}{(1)(0.000034666)}$$

$$\omega_n \approx 112500 \text{ rad/s}$$

Therefore, the observer PI controller is:

$$PI_{\text{Observer}}(s) = 0.08125 \frac{s + 112846.1538}{s}$$

Where in parallel PI controller form:

$$K_p = 0.08125$$

$$K_i = 9168.75$$

### **3.6 Conclusion**

Throughout this chapter, the mathematical model of a BLDC machine in the synchronous reference frame and the machine's most important electrical and mechanical parameters were derived so as to be used for the design of current and speed controllers for the RFOC scheme. Furthermore, the design of the current and speed PI controllers were discussed through the use of sisotool and MATLAB. Lastly, the mathematical models for the back-emf observer and PLL were derived and discussed.

## 4. Simulation Results

### 4.1 Introduction to Simulation Results

This chapter illustrates the implementation of the mathematical theory obtained in the previous chapter on a simulation model of a BLDC machine designed on Simulink/MATLAB. Simulation results showing the machine's voltages, currents, rotor speed and electrical rotor angle during sensored and sensorless operation are discussed. Furthermore, the ability of the back-emf observer to correctly estimate the electrical rotor angle during sensored and sensorless operation is also discussed.

### 4.2 Sensored Operation

The results presented in this section illustrate the performance of the cascaded PI speed and current controllers. The simulation results in Section 4.2 assume a sensored operation of the BLDC machine, that is, the machine's rotor electrical angle and speed are obtained from a dedicated speed/position sensor. While as discussed previously in Section 2.5 low to medium power BLDCs are rarely coupled with a speed/position sensor in practice; it is good practice to validate the simulation performance of the cascaded loop in a sensored mode such that the sensorless error estimate is not introduced in preliminary testing.

#### 4.2.1 BLDC Machine Model

From equations (3.11), (3.12), (3.13), (3.15) and (3.16) a simulation model representing the BLDC machine was implemented on Simulink as seen in Figure 4.1 Left. The BLDC machine simulation model of Figure 4.1 Left was included in the block “**BLDC Machine Model**” with inputs  $V_d$  and  $V_q$  and outputs  $i_{abc}$ ,  $\theta_{elec}$ ,  $\omega$ ,  $BEMF\alpha$  and  $BEMF\beta$  as shown in Figure 4.1 Right.

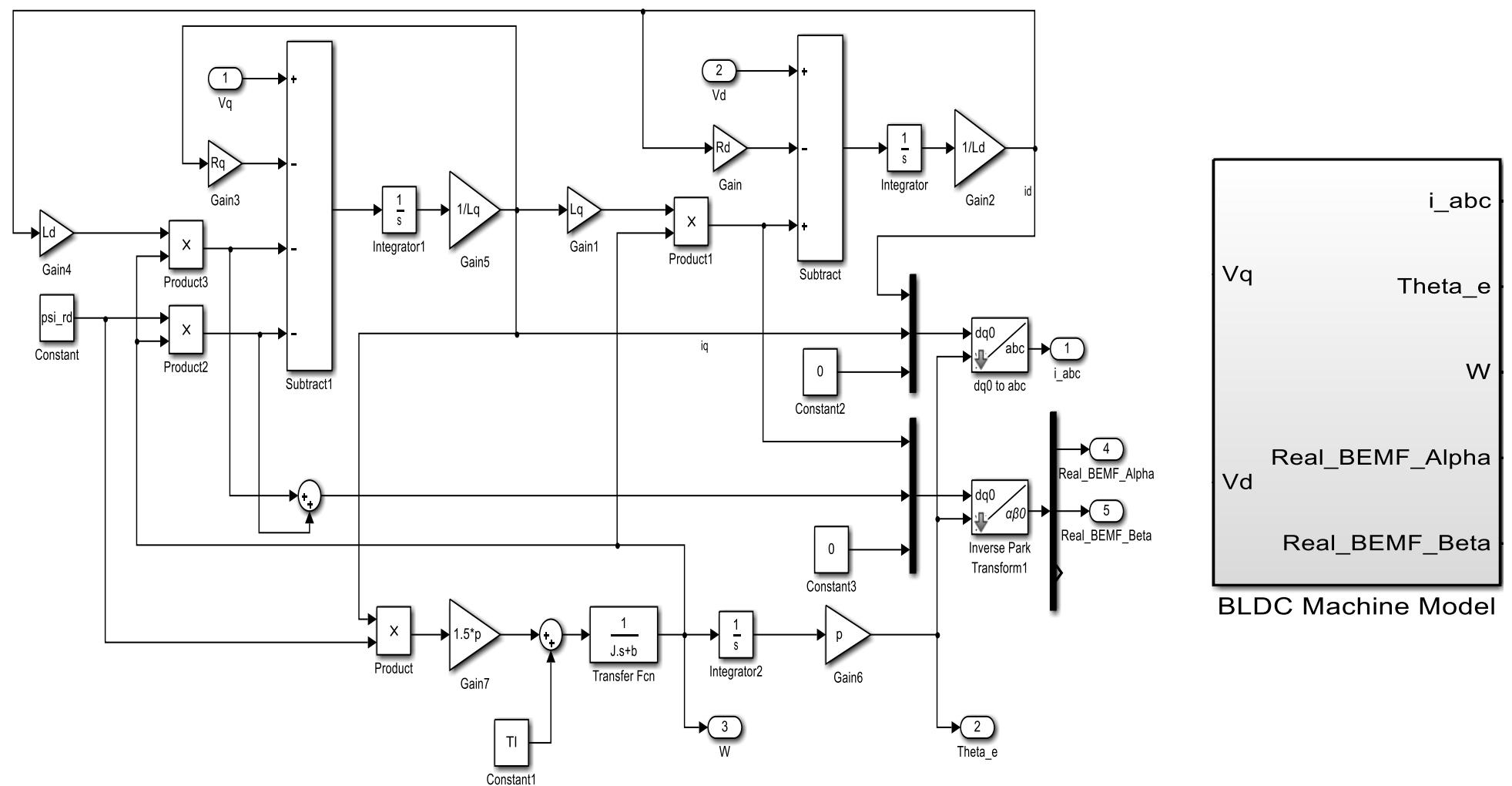
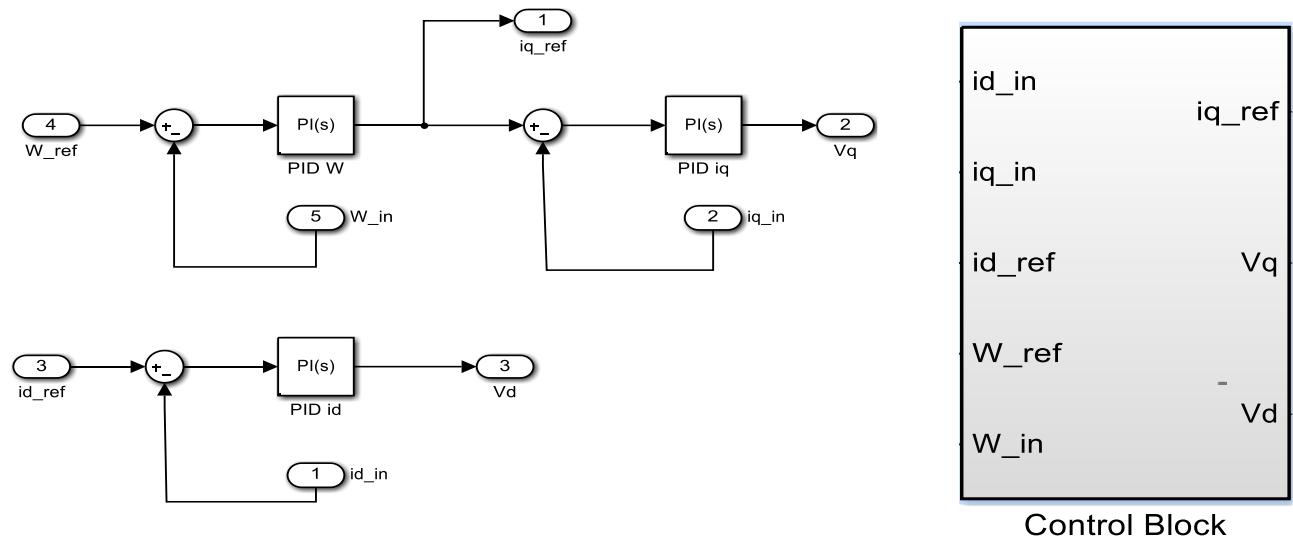


Figure 4.1: Left - BLDC machine simulation diagram. Right - BLDC machine simulation block

#### 4.2.2 Current and Speed PI Controllers Performance

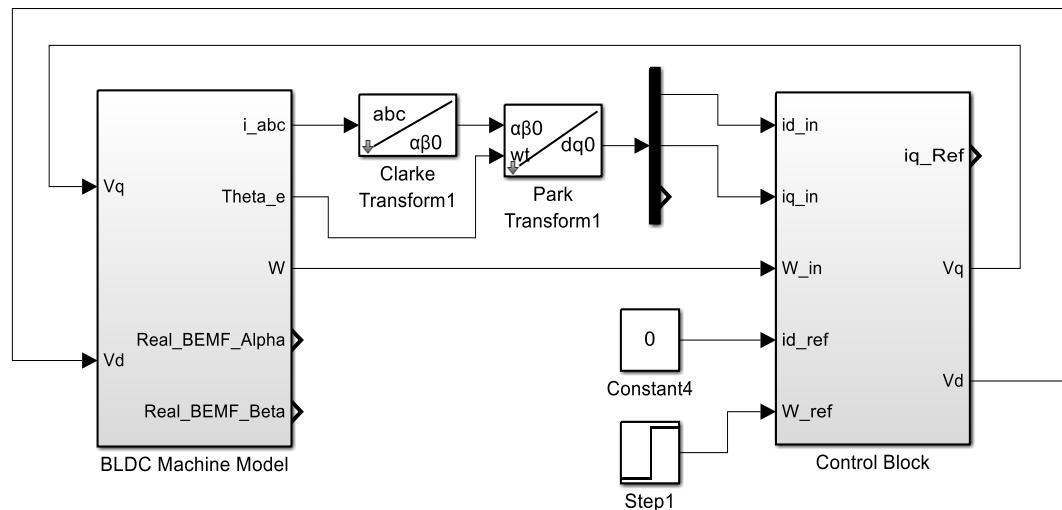
The designed PI controllers were implemented as shown in Figure 4.2 Left. All three PI controller models were simplified in the block “Control Block” with inputs  $i_{d\text{in}}$ ,  $i_{q\text{in}}$ ,  $i_{d\text{ref}}$ ,  $\omega_{\text{ref}}$  and  $\omega_{\text{in}}$  and outputs  $i_{q\text{ref}}$ ,  $V_q$  and  $V_d$  as shown in Figure 4.2 Right



**Figure 4.2: Left - PI Controllers simulation diagram. Right - PI Controllers simulation block**

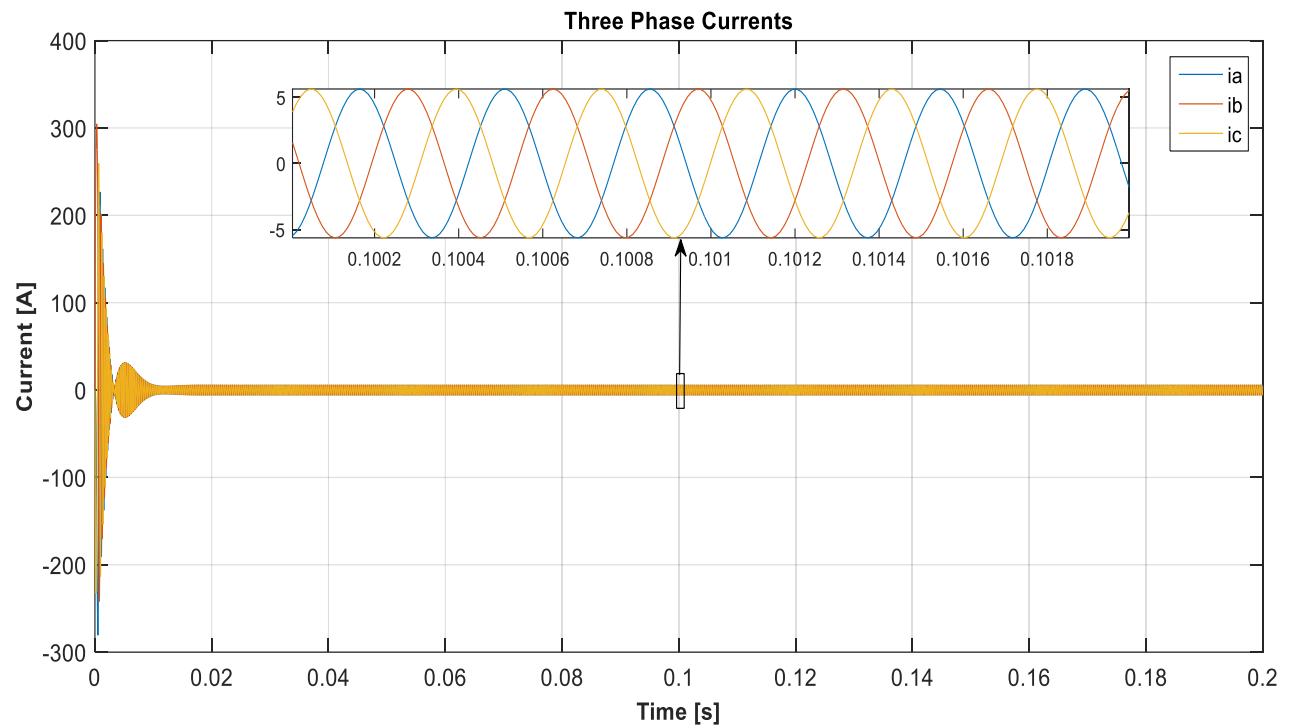
The closed loop performance of the simulated BLDC machine model was tested as shown in Figure 4.3 with:

- A speed reference of 1300 rad/s (Rated speed)
- A load torque of 0.1437 Nm i.e. (Rated torque)
- A d-axis reference current of 0 A

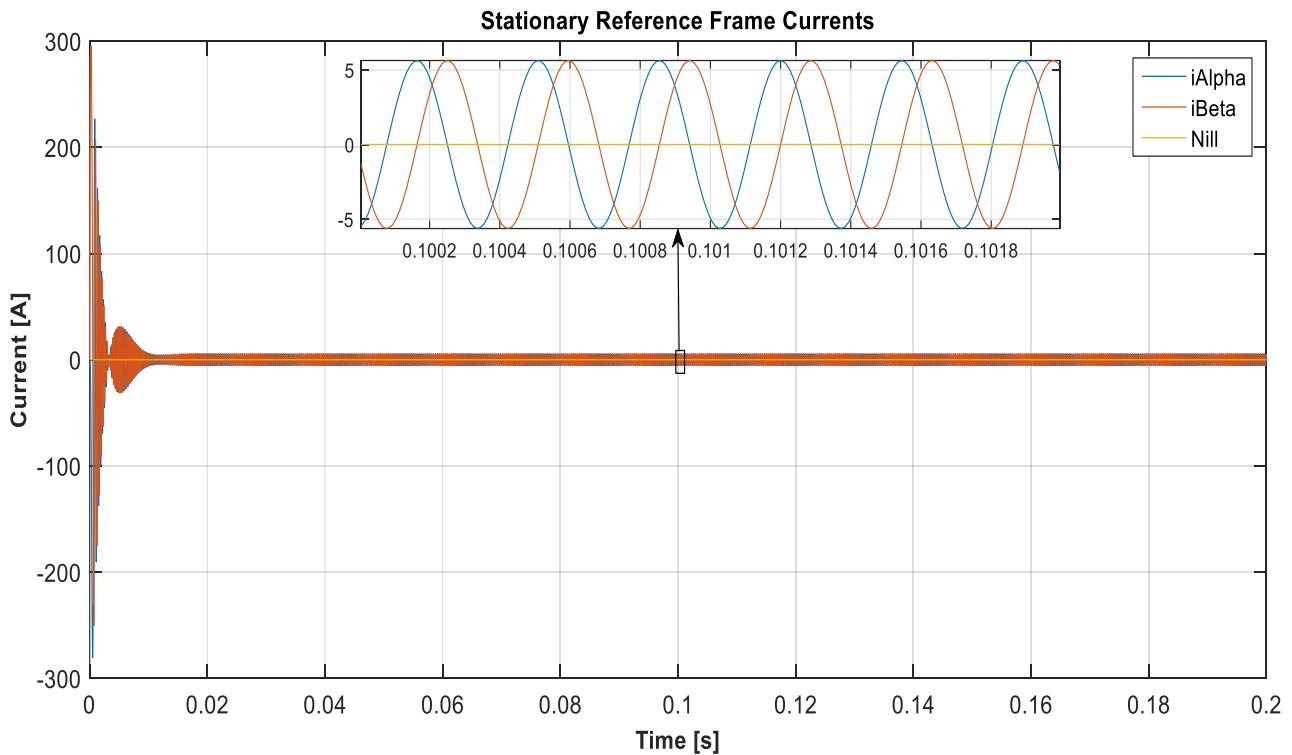


**Figure 4.3: FOC simulation diagram**

Figure 4.4 and 4.5 show the three phase currents  $i_a$ ,  $i_b$  and  $i_c$  and the stationary frame currents  $i_\alpha$  and  $i_\beta$  for the reference speed of 1300 rad/s respectively. The current is significantly high at starting due to the low back-emf but it settles to the required value in less than 0.02s. This is due to the fast speed change requirement from the speed reference step input. Furthermore currents  $i_\alpha$  and  $i_\beta$  have a peak value of 5 A.

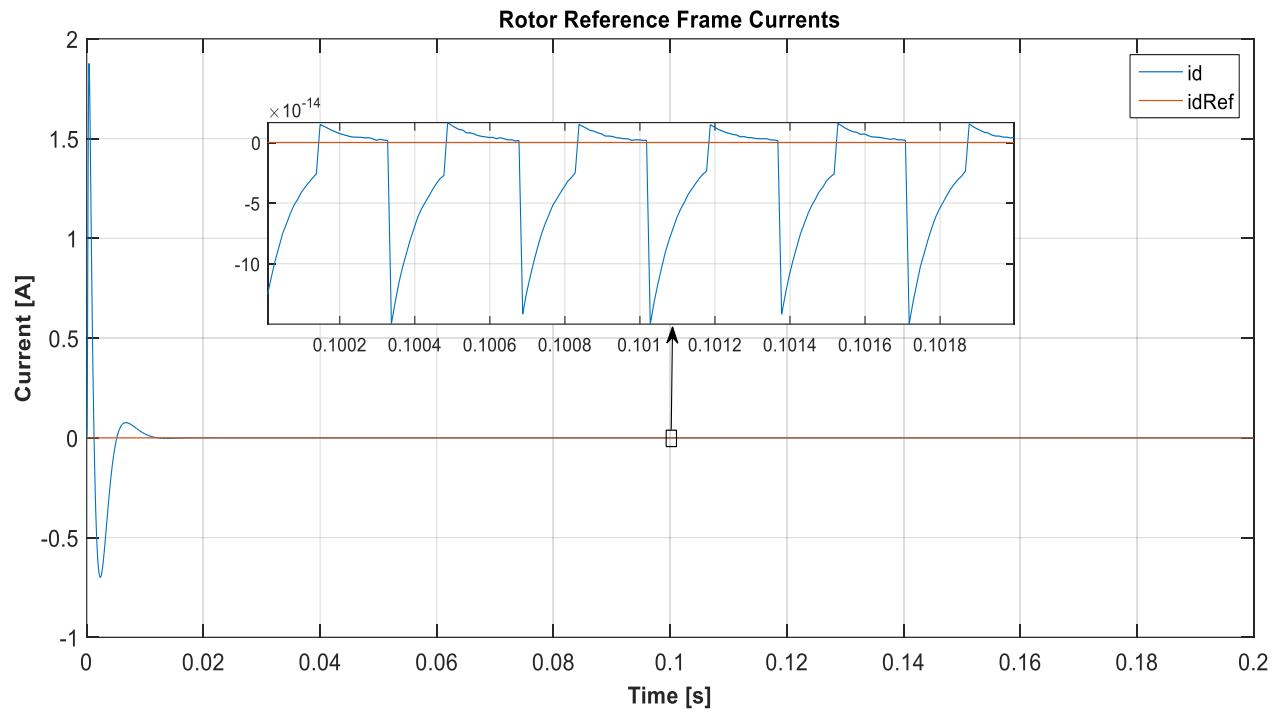


**Figure 4.4: Three phase current at  $\omega = 1300$  rad/s**

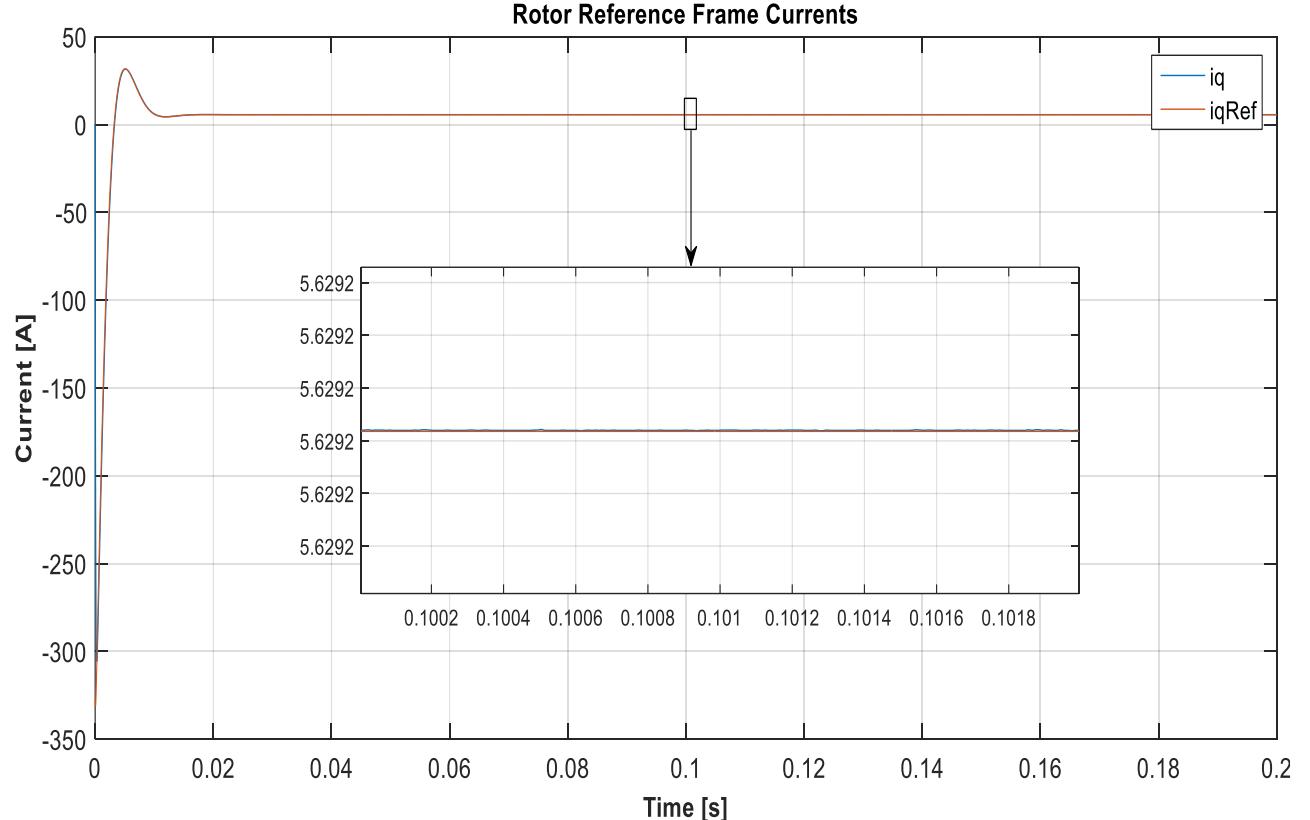


**Figure 4.5: Stationary reference frame currents at  $\omega = 1300$  rad/s**

Figure 4.6 and 4.7 shows the synchronous frame stator currents  $i_d$  and  $i_q$  at a reference speed of 1300 rad/s respectively. The current is significantly high at starting due to the low back-emf but it settles to the required value in less than 0.02s. This is due to the fast speed change requirement from the speed reference step input. Current  $i_d$  can be seen to settle at 0 A while current  $i_q$  settles at 5.6292 A.

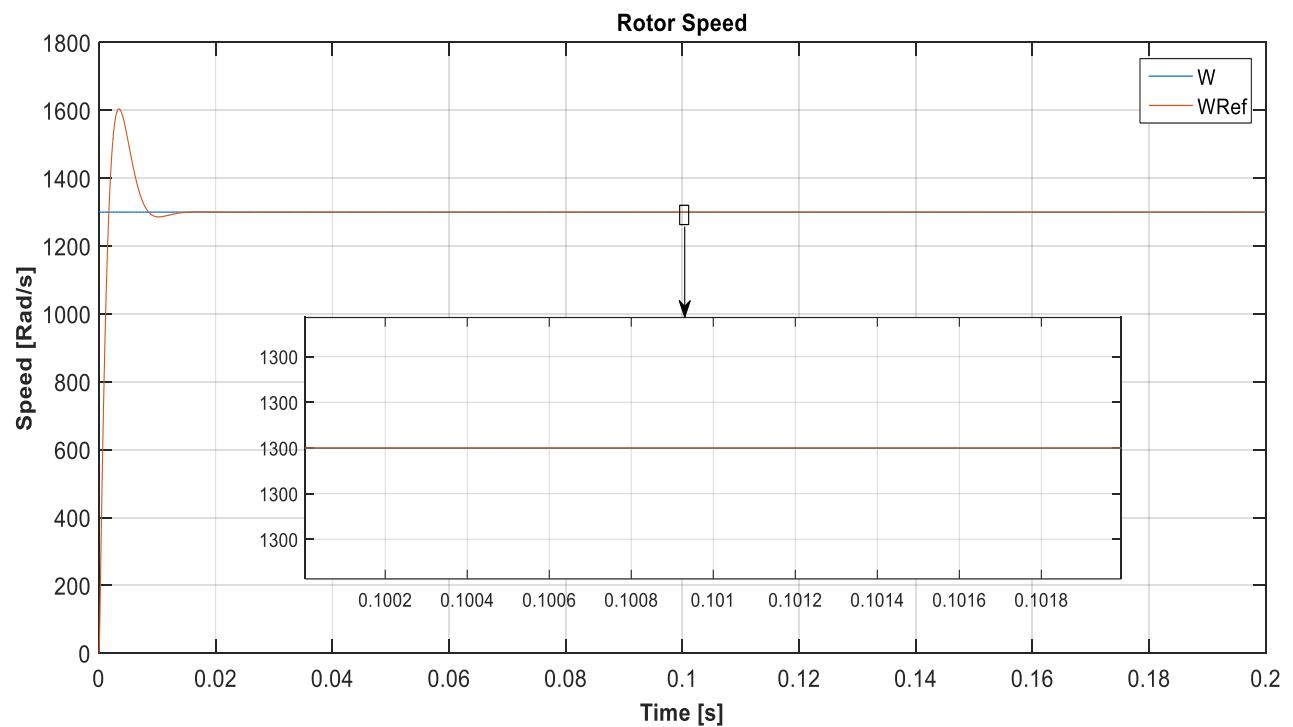


**Figure 4.6: Rotor reference frame current  $i_d$  at  $\omega = 1300$  rad/s**



**Figure 4.7: Rotor reference frame current  $i_q$  at  $\omega = 1300$  rad/s**

Figure 4.8 shows the machine's speed output after a speed reference input of 1300 rad/s with a load torque of 0.1437 Nm.



**Figure 4.8: BLDC machine's rotor speed with a speed reference input of  $\omega = 1300$  rad/s**

## 4.3 Sensored Operation with Observer

From Figures 3.7, 3.8 and 3.9 and from equations (3.31), (3.32) and (3.33) the simulation model of the observer was implement in Simulink/MATLAB as shown in Figure 4.9 Left. The observer simulation model was simplified in a function block called "**Observer Block**" with inputs  $i_\alpha$ ,  $i_\beta$ ,  $V_\alpha$ ,  $V_\beta$ ,  $\theta_{elec}$  and  $\omega$  and outputs  $\hat{\theta}_{elec}$  BEMF $\alpha$ , BEMF $\beta$  and  $\hat{\omega}$  as shown in Figure 4.9 Right.

### 4.3.1 Rotor Electrical Angle and Rotor Speed Estimation

Figures 4.10 and 4.11 show  $\hat{i}_\alpha$  converging to  $i_\alpha$  and  $\hat{i}_\beta$  converging to  $i_\beta$  respectively due to the effect of the designed PI controllers. The convergence between the estimated and real values of currents  $i_\alpha$  and  $i_\beta$  show that an estimated BEMF $\alpha$  and BEMF $\beta$  produced must represent the actual values of the BEMF $\alpha$  and BEMF $\beta$ . Figures 4.12 and 4.13 show how the estimated BEMF $\alpha$  and BEMF $\beta$  are converging to the actual BEMF $\alpha$  and BEMF $\beta$ .

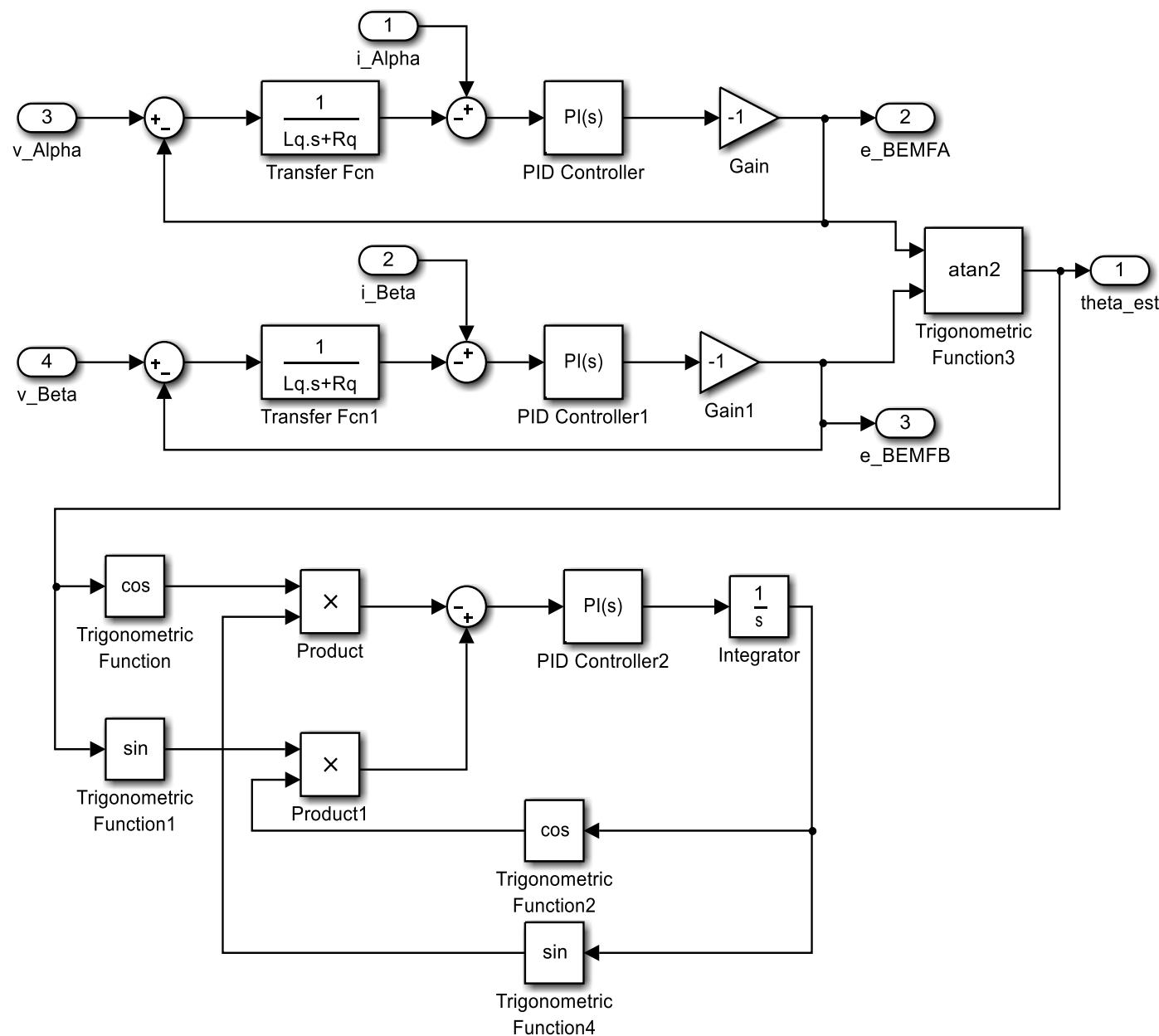
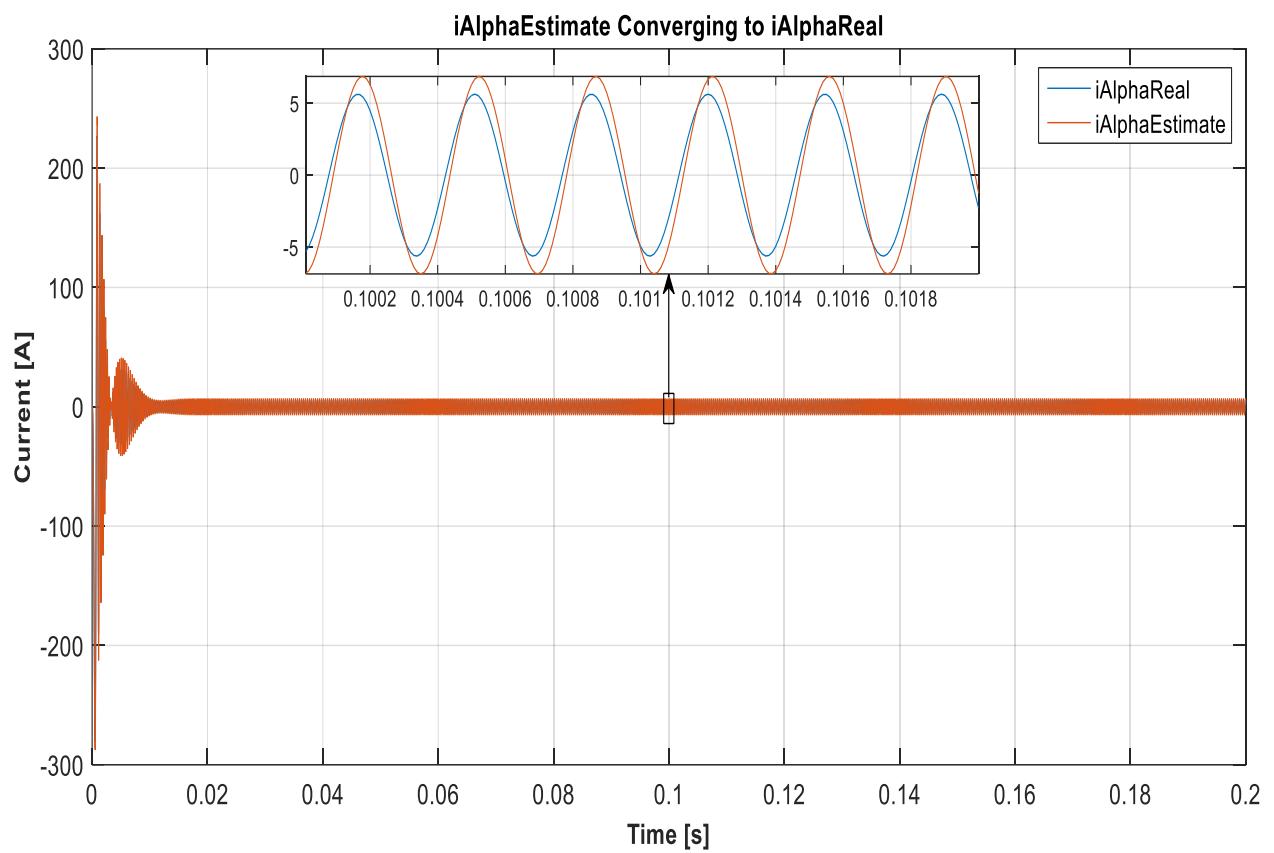
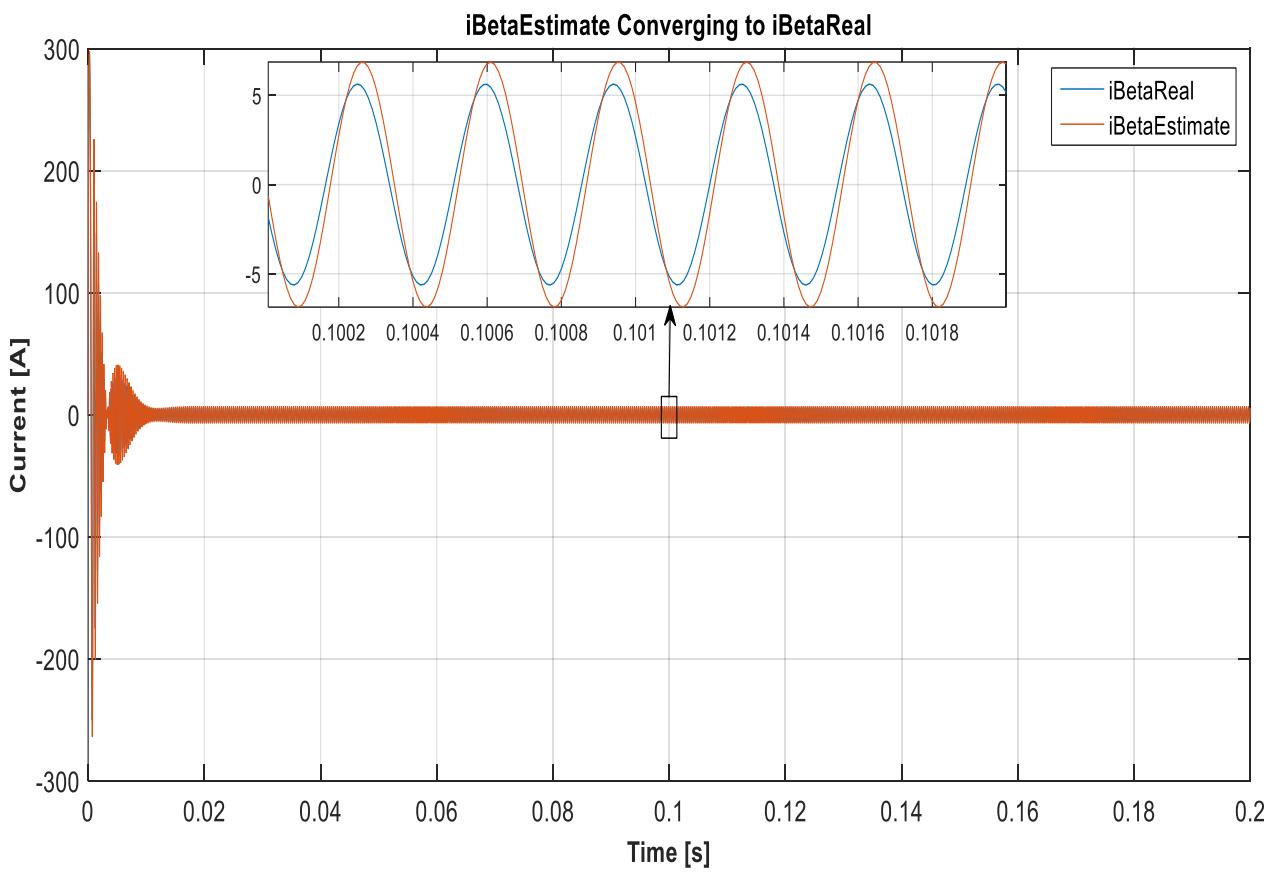


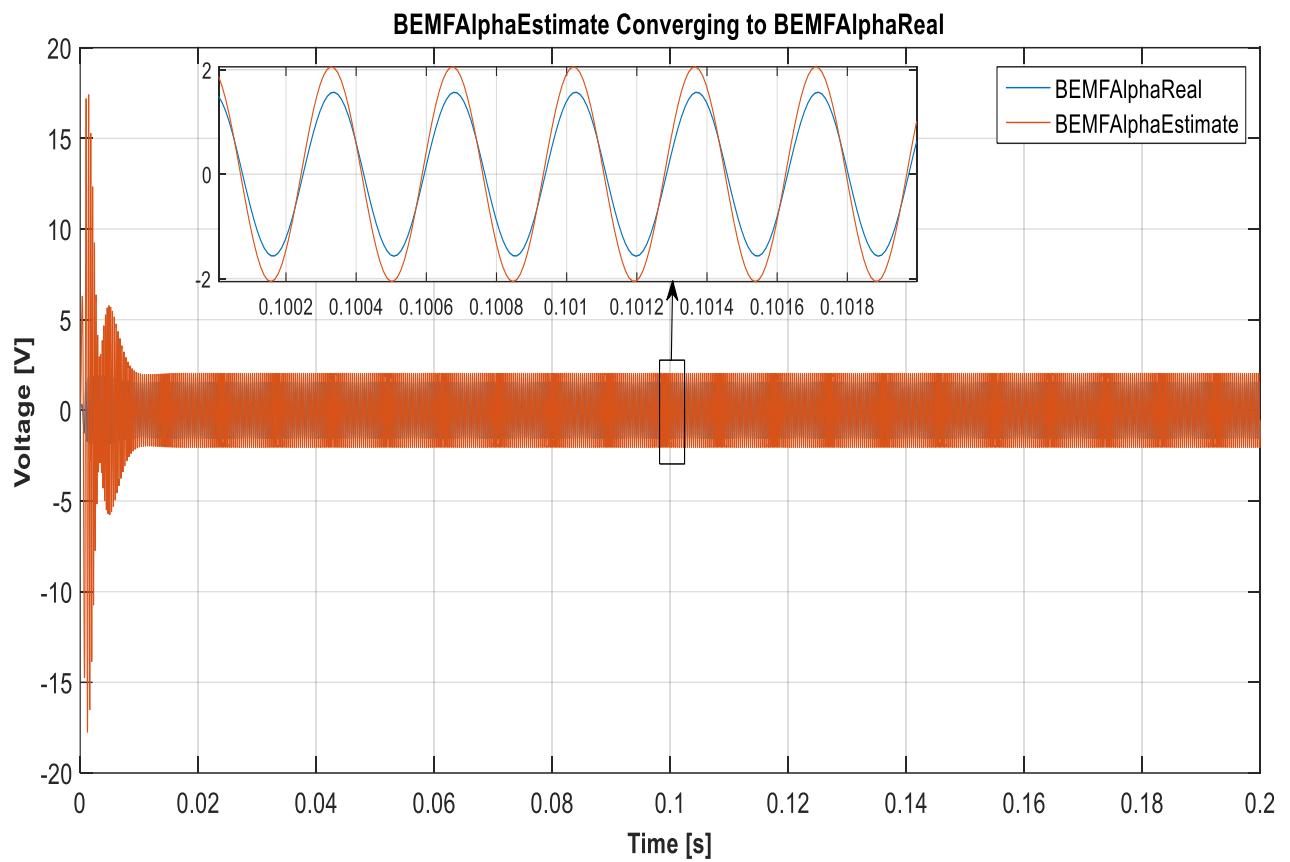
Figure 4.9: Left - Observer simulation diagram. Right - Observer simulation block



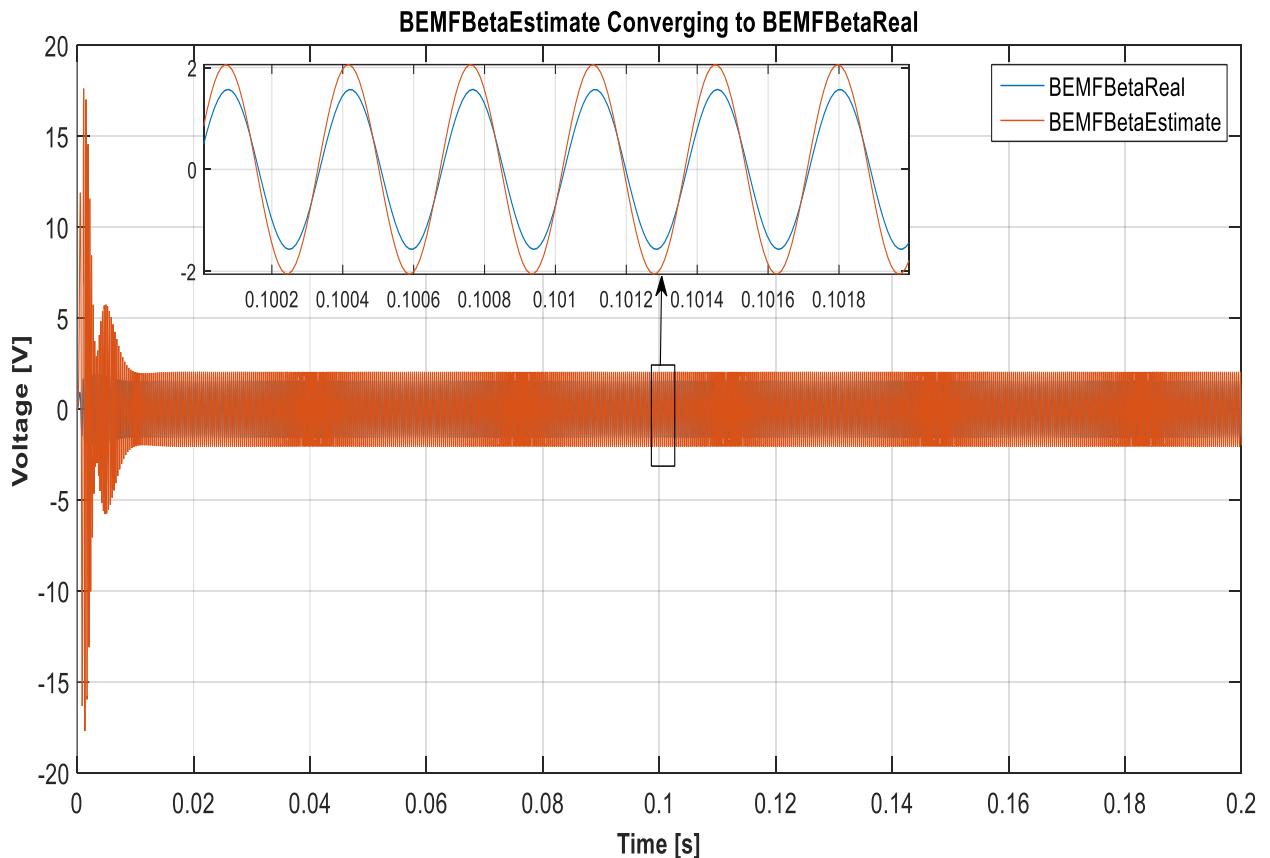
**Figure 4.10:**  $i_{\alpha}$  converging to  $\hat{i}_{\alpha}$  at  $\omega = 1300$  rad/s



**Figure 4.11:**  $i_{\beta}$  converging to  $\hat{i}_{\beta}$  at  $\omega = 1300$  rad/s

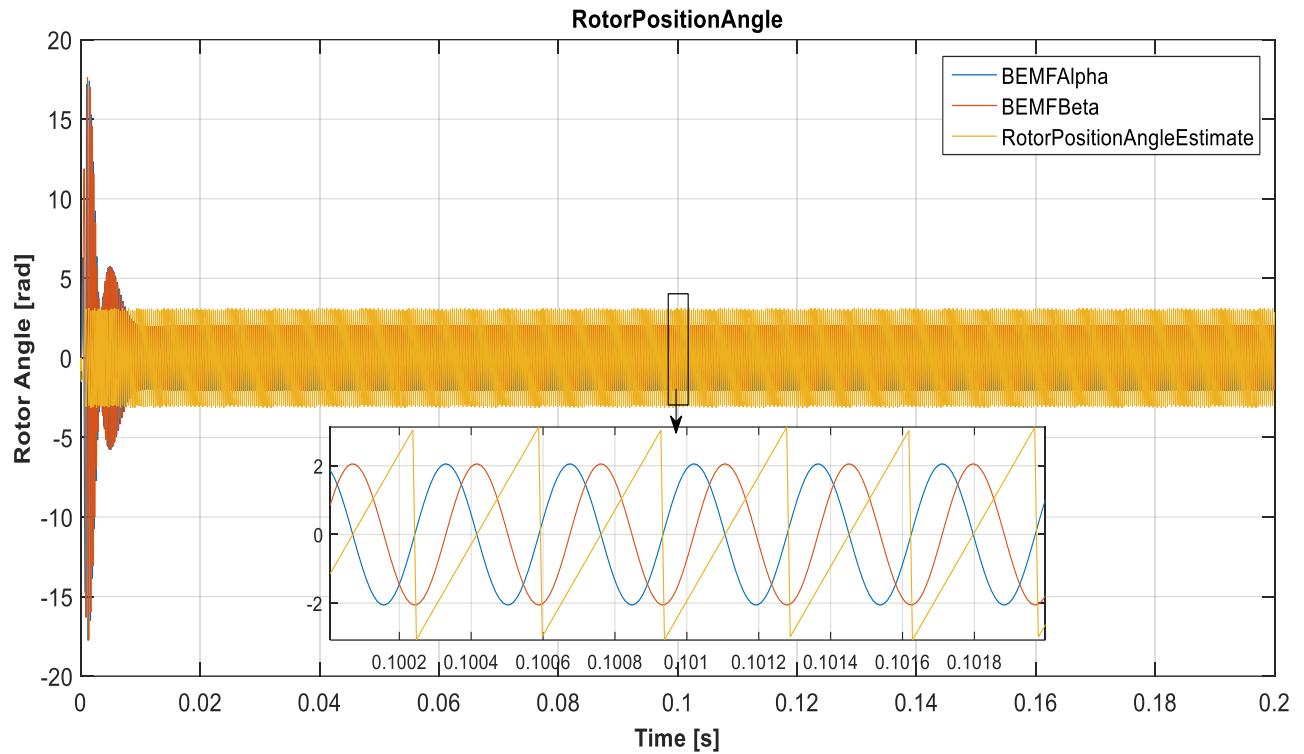


**Figure 4.12:**  $v_\alpha$  converging to  $\hat{v}_\alpha$  at  $\omega = 1300$  rad/s

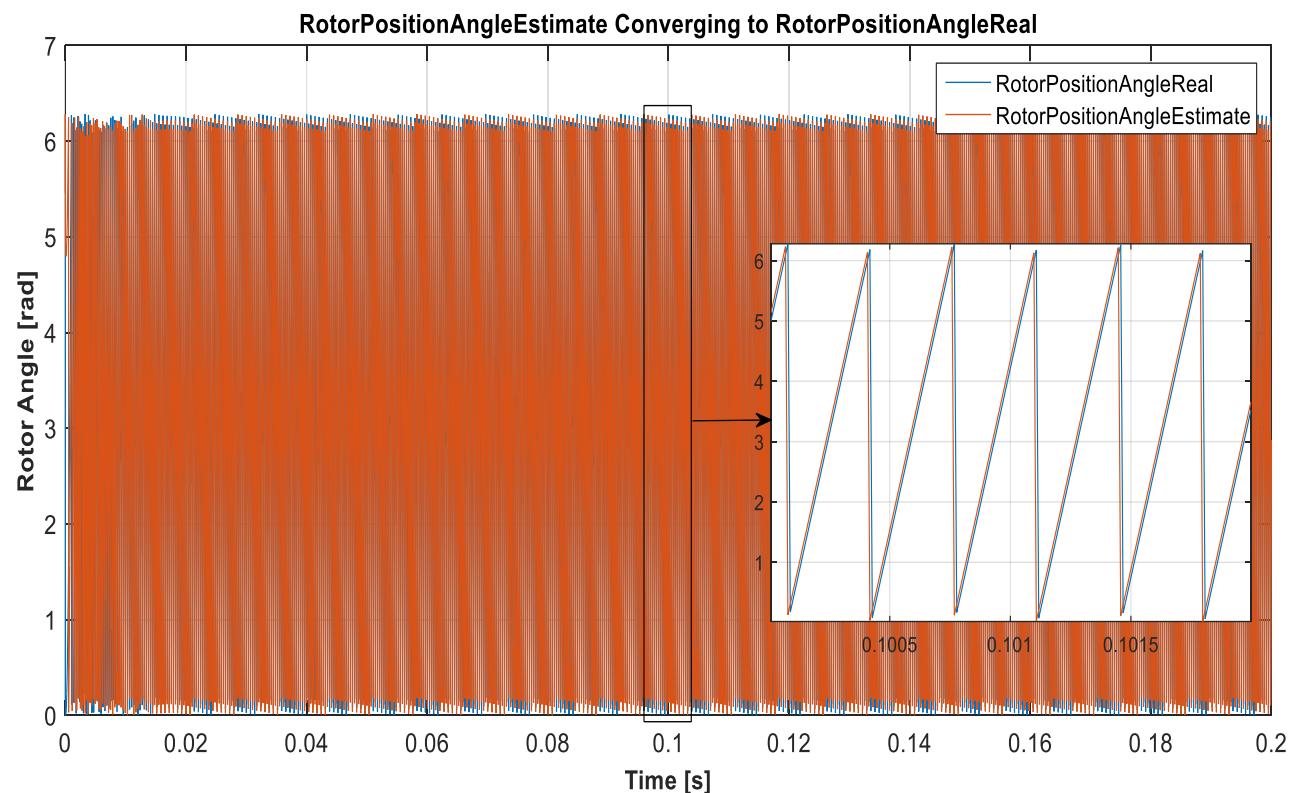


**Figure 4.13:**  $v_\beta$  converging to  $\hat{v}_\beta$  at  $\omega = 1300$  rad/s

Applying equations (3.31), (3.32) and (3.33) on the estimated  $\text{BEMF}\alpha$  and  $\text{BEMF}\beta$  of Figures 4.12 and 4.13, the electrical  $\theta$  representing the rotor position can be estimated. Figure 4.14 shows how the estimated  $\text{BEMF}\alpha$  and  $\text{BEMF}\beta$  create the electrical  $\theta$  representing the rotor position and Figure 4.15 shows how accurate such an observer is by comparing the actual and estimated electrical  $\theta$ .

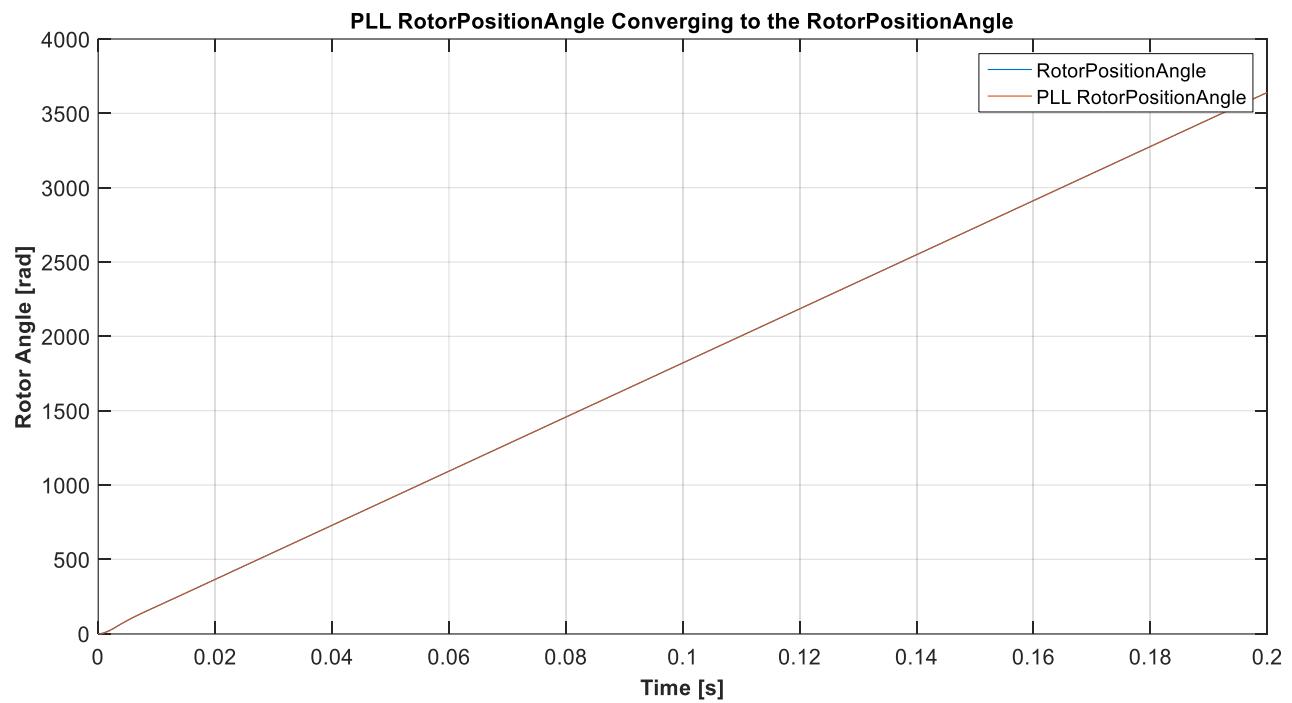


**Figure 4.14: Rotor position angle formation from  $\text{BEMF}\alpha$  and  $\text{BEMF}\beta$**

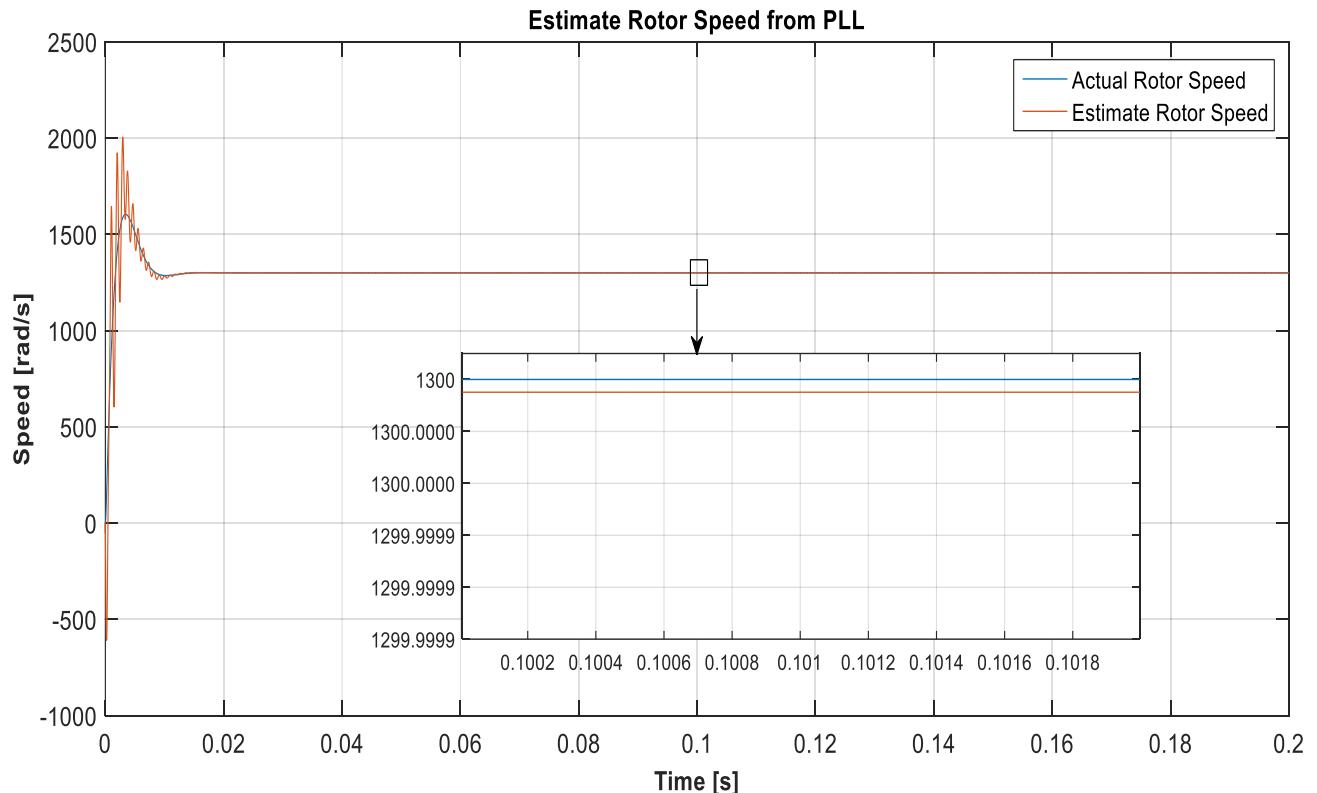


**Figure 4.15: Rotor position angle**

Figures 4.16 and 4.17 show how accurately the rotor speed is being estimated, where Figure 4.16 shows how accurately the PLL output is tracking the estimated rotor electrical angle while Fig 4.17 shows how accurately the estimated speed position is being generated from the PLL when compared with the actual machine's rotor speed.



**Figure 4.16: PLL rotor angle estimate tracking real rotor angle**



**Figure 4.17: PLL rotor speed estimate tracking real rotor speed**

## 4.4 Sensorless Operation

After obtaining an accurate estimate of the rotor electrical angle and rotor speed, the simulation analysis should be repeated with the estimated speed/position quantities used for control purposes. Since the back-emf algorithm being used in this dissertation is suitable only for high speed operation, the BLDC is started in a sensored mode and a transition over to a sensorless operation is performed once a sufficiently enough high speed is obtained.

### 4.4.1 Transition Dynamics

The transition from actual electrical angle/speed to the estimated electrical angle/speed occurs at 0.2 s. Figures 4.18, 4.19, 4.20 and 4.21 show currents  $i_a$ ,  $i_b$  and  $i_c$ , currents  $i_d$  and  $i_q$  and speed  $\omega$  respectively about the 0.2 s mark at which sensorless transition is performed. During transition, both synchronous frame currents have spikes of  $\pm 6$  A while the rotor speed spikes at +1303 rad/s. The BLDC returns to steady state operation after 0.045 s. After sensorless changeover the three phase currents increase from  $\pm 5$  A to  $\pm 7.37$  A. Figures 4.22, 4.23 and 4.24 show the effect of the transition on  $BEMF\alpha$ ,  $BEMF\beta$  and the rotor electrical angle. Both  $BEMF\alpha$  and  $BEMF\beta$  experience some spikes during transition. After transition, the error between the actual the estimate of the rotor electrical angle increases by 0.1 degrees. The increase in error is due to the bandwidth of the PI controllers and the deterioration of the back-emf estimates.

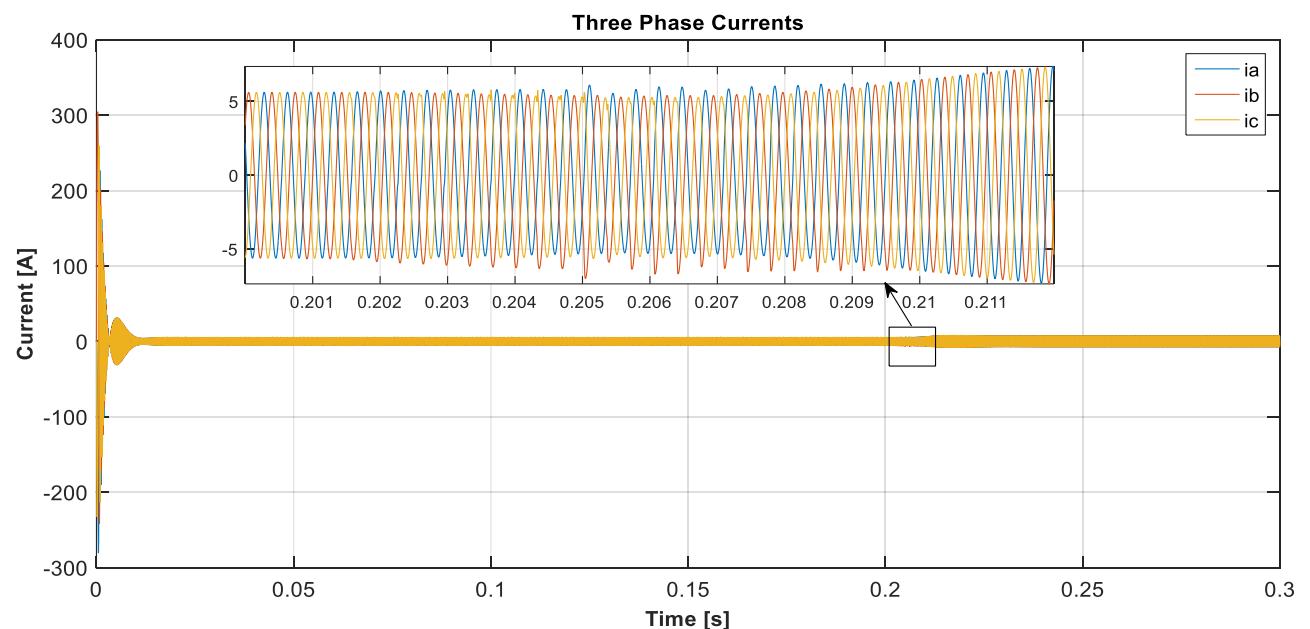
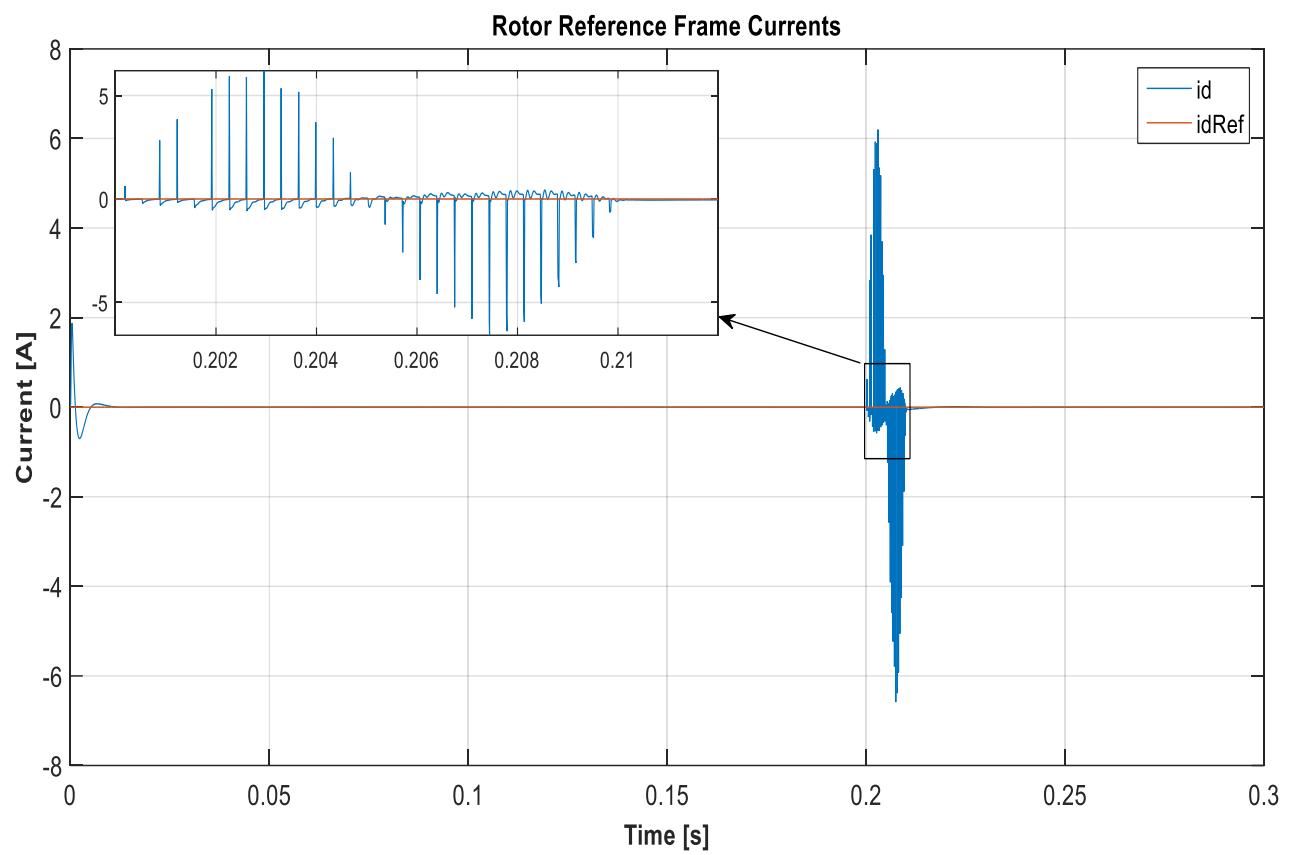
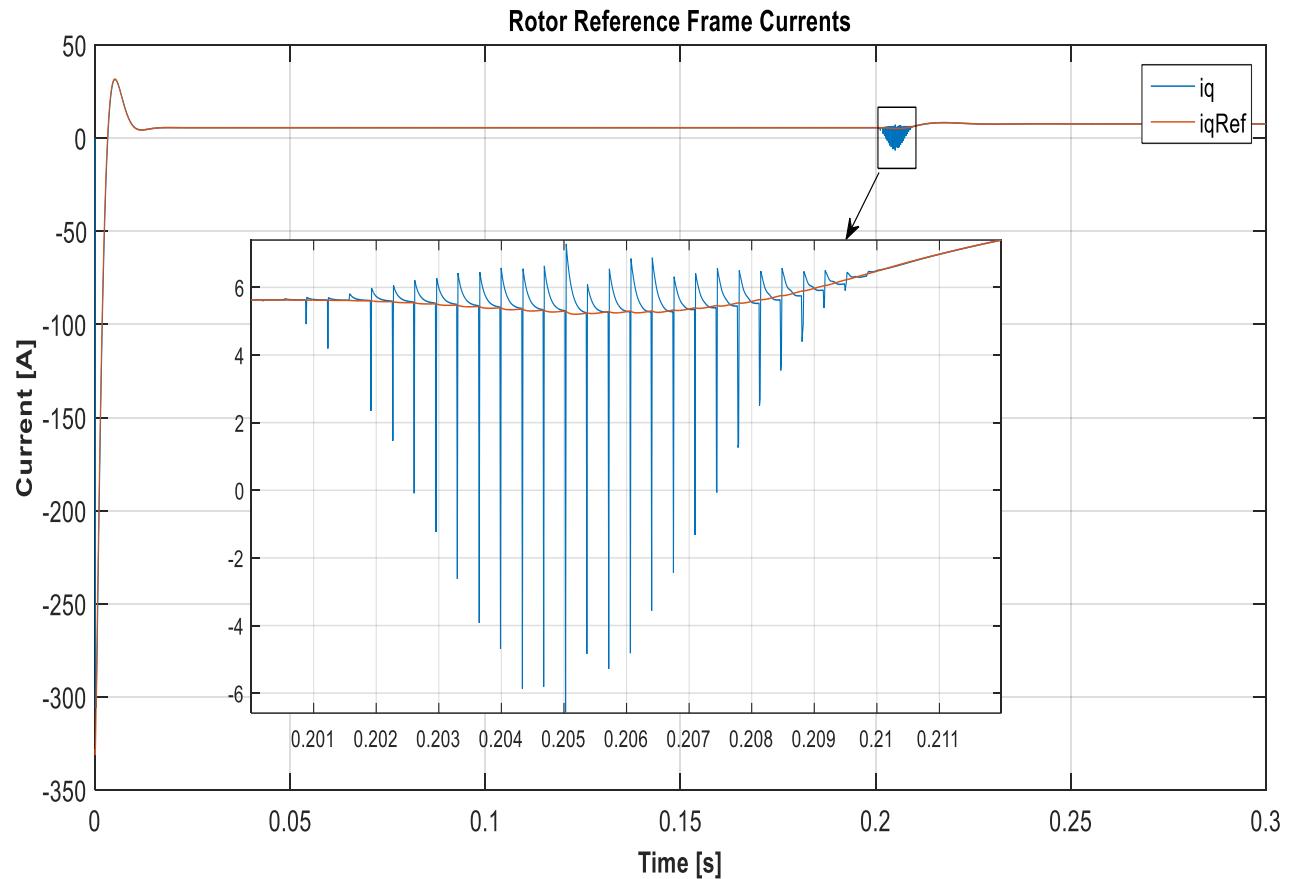


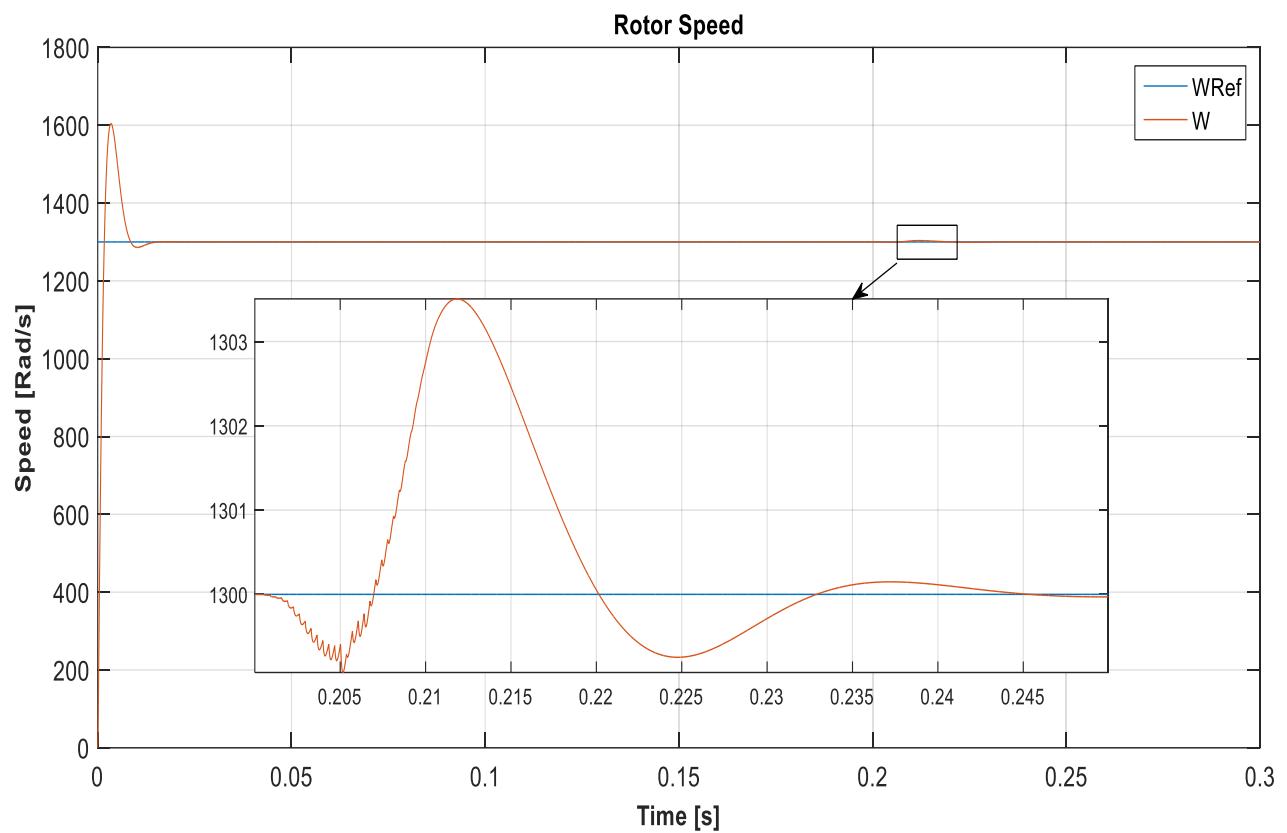
Figure 4.18: Three phase currents during transition at  $\omega = 1300$  rad/s



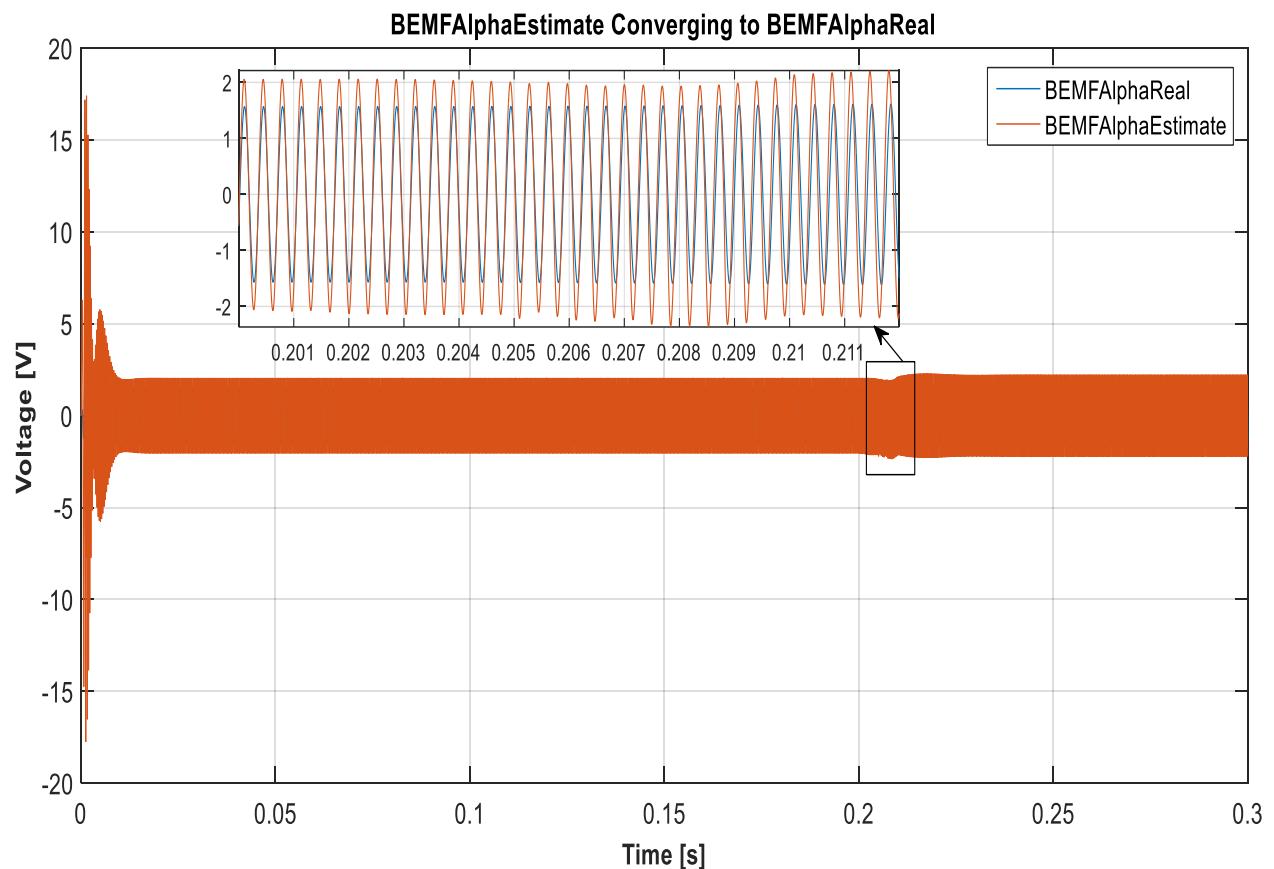
**Figure 4.19:** Rotor reference frame current  $i_d$  during transition at  $\omega = 1300 \text{ rad/s}$



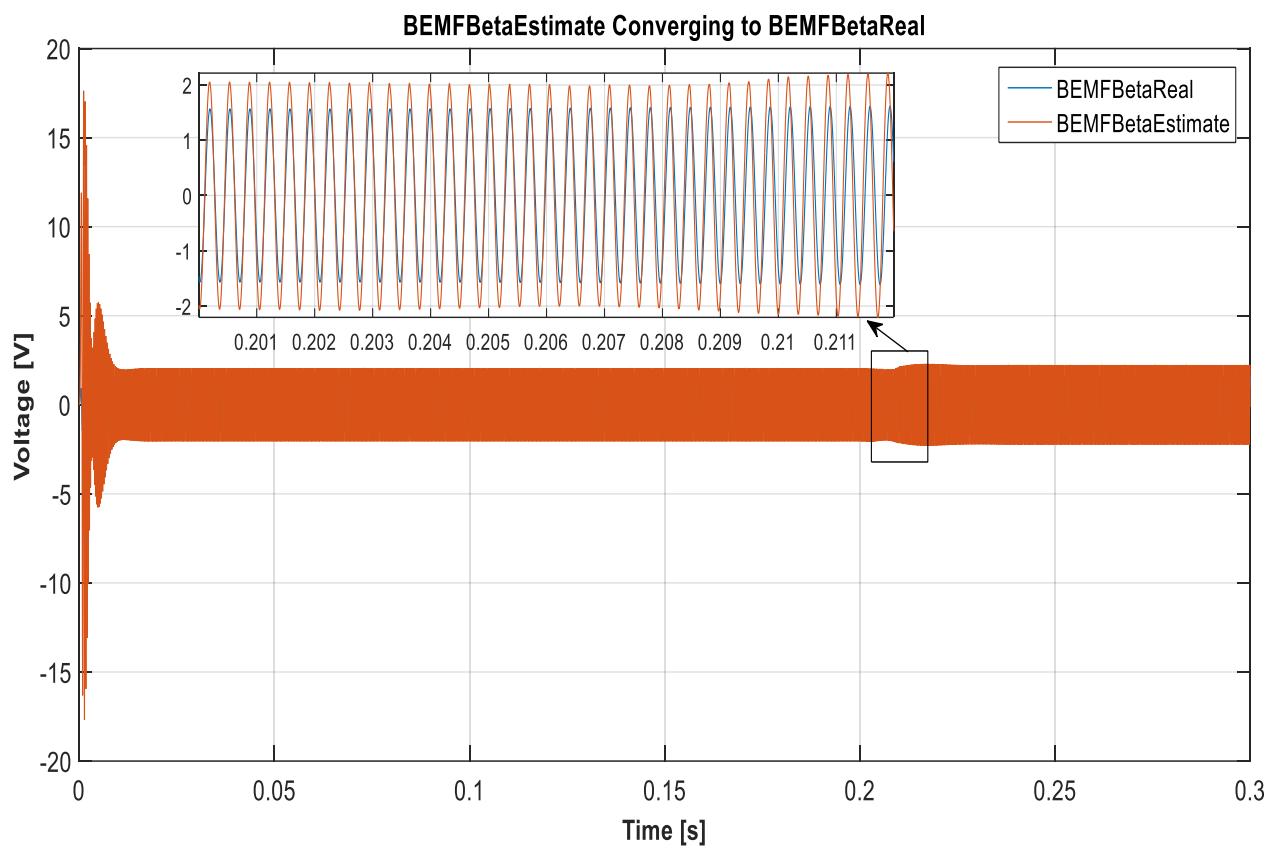
**Figure 4.20:** Rotor reference frame current  $i_q$  during transition at  $\omega = 1300 \text{ rad/s}$



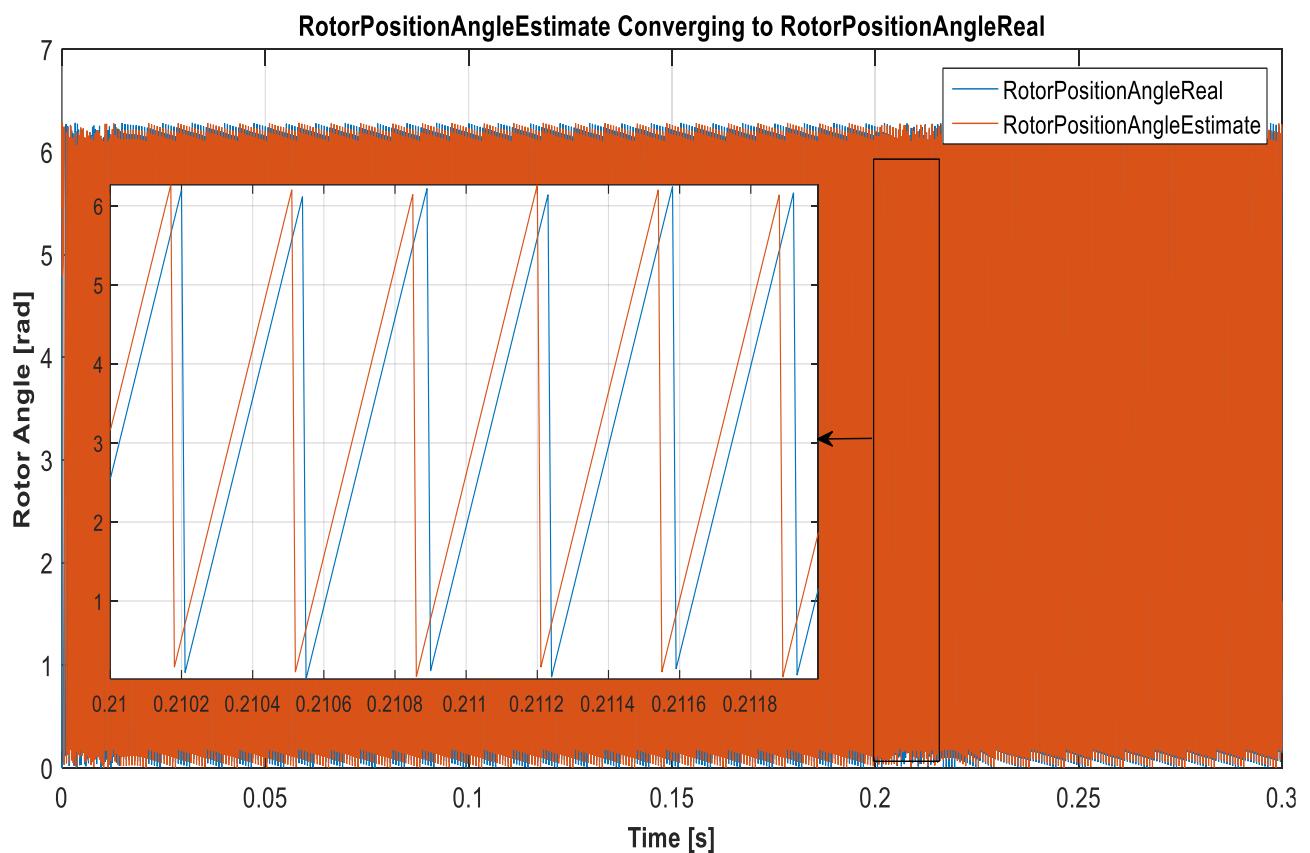
**Figure 4.21:** Rotor speed during transition with a speed reference input of  $\omega = 1300$  rad/s



**Figure 4.22:**  $v_\alpha$  converging to  $\widehat{v_\alpha}$  during transition at  $\omega = 1300$  rad/s



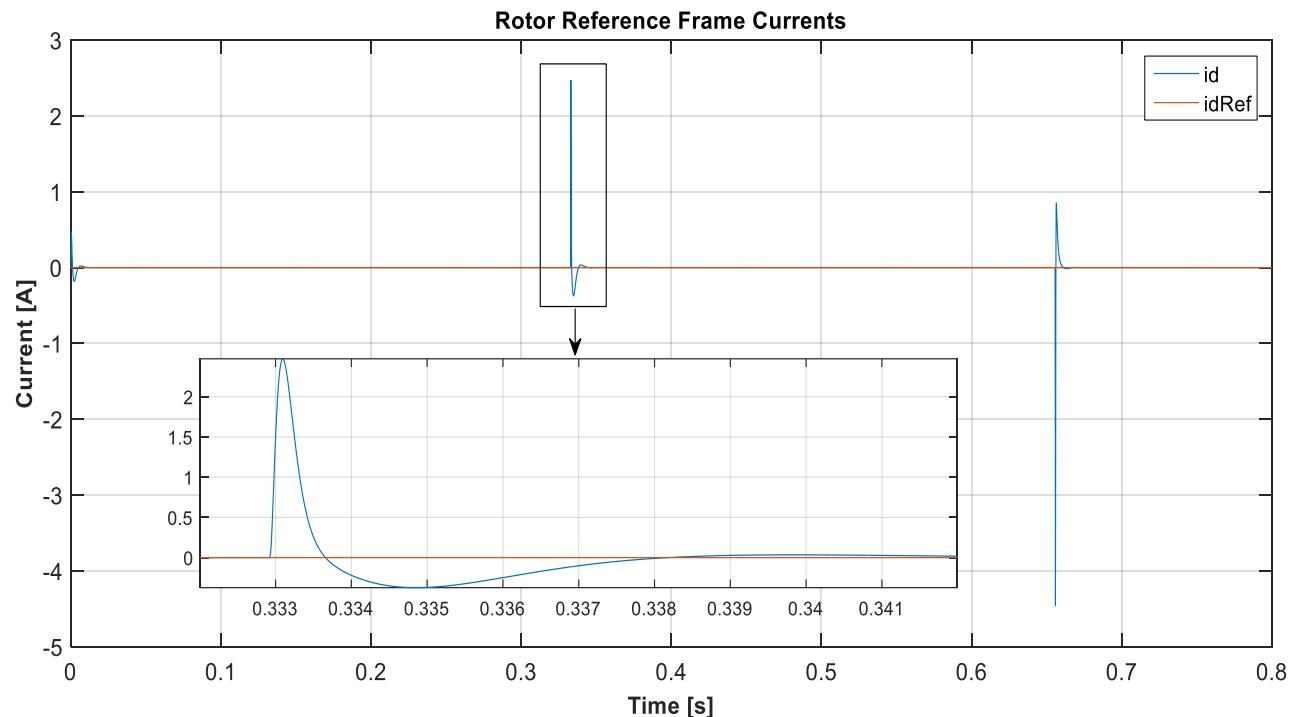
**Figure 4. 23:**  $v_\beta$  converging to  $\hat{v}_\beta$  during transition at  $\omega = 1300 \text{ rad/s}$



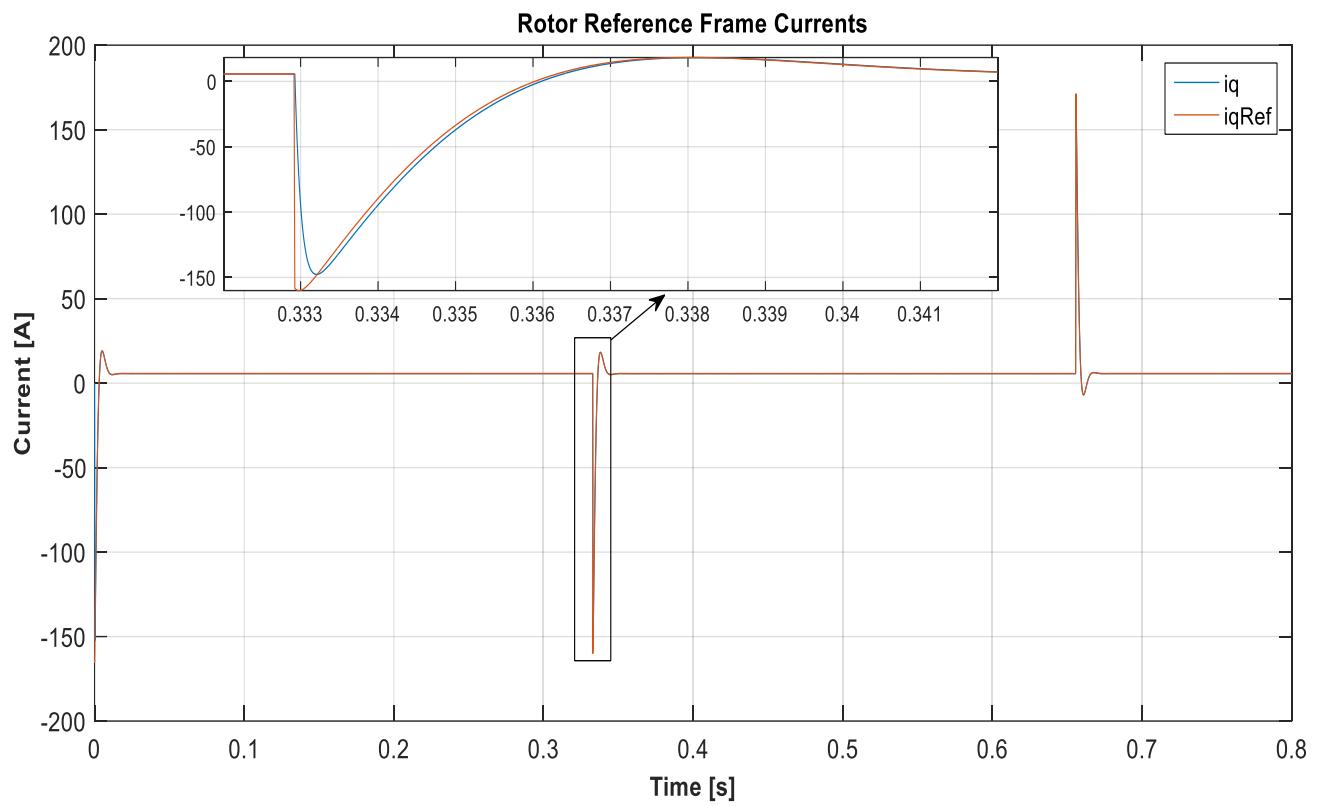
**Figure 4.24:** Rotor position angle during transition

## 4.5 Speed Control at Sensorless Operation

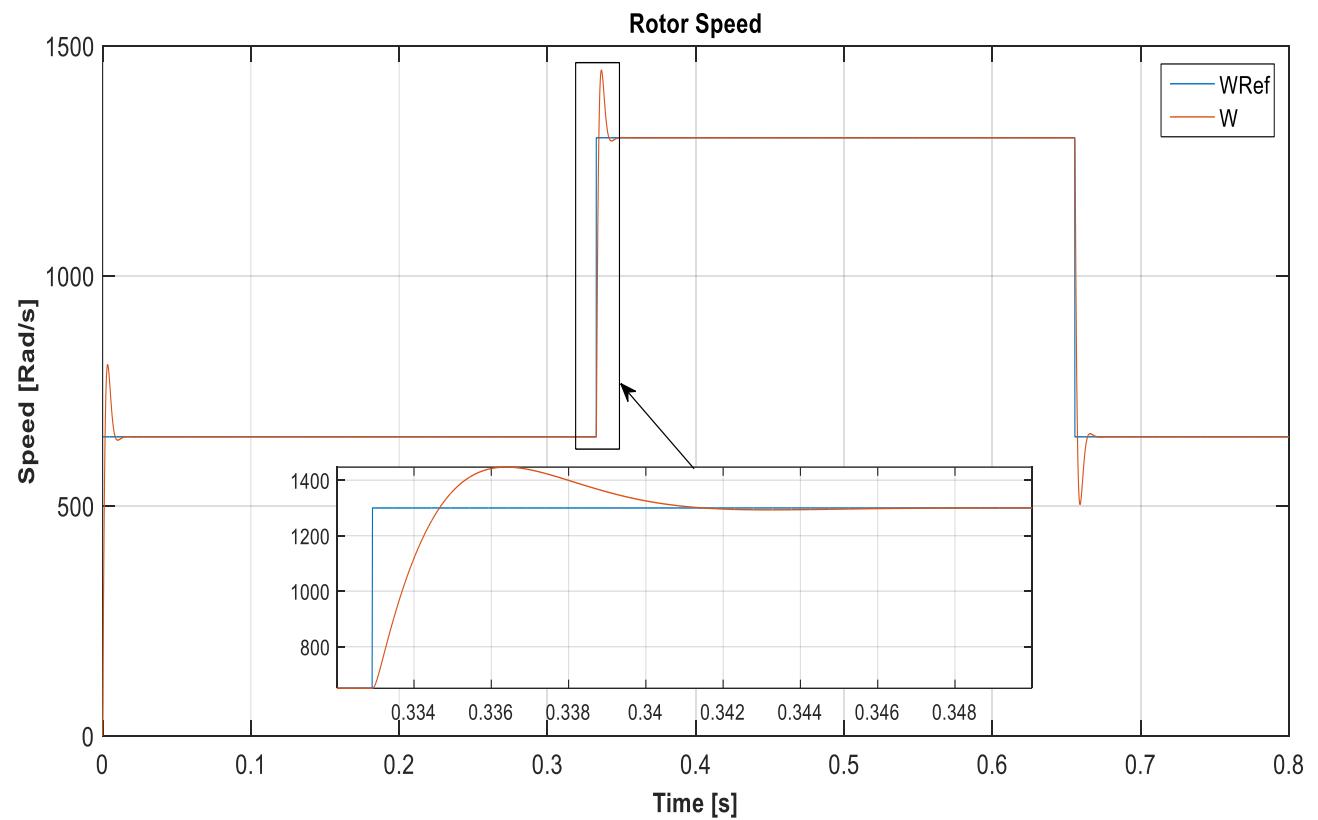
Figures 4.25, 4.26 and 4.27 show the control of the synchronous reference frame currents, and speed of the BLDC machine at sensorless operation respectively. The input speed reference is changed from half the rated speed to the rated speed at 0.333s. After a further 0.322s the speed reference is changed again to half the rated speed. During the change in speed reference currents  $i_d$  and  $i_q$  experience some oscillations. Current  $i_d$  peaks at +2.5 A when the speed of the machine is changed from half the rated speed to the rated speed and it peaks at -4.5 A when the speed of the machine is changed from rated speed to half the rated speed. Furthermore, current  $i_q$  peaks at +152 A when the speed of the machine is changed from half the rated speed to the rated speed and it peaks at -150 A when the speed of the machine is changed from rated speed to half the rated speed. Both currents then settle very quickly after the current spikes. The speed of the machine also experiences some oscillations during the speed change. Speed  $\omega$  peaks at +1400 rad/s when the speed of the machine is changed from half the rated speed to the rated speed and it peaks at +500 rad/s when the speed of the machine is changed from rated speed to half the rated speed. Speed  $\omega$  settles back after 0.011s



**Figure 4.25: Rotor reference frame current  $i_d$  during sensorless operation for  $\omega$  from 650 rad/s to 1300 rad/s and  $\omega$  from 1300 rad/s to 650 rad/s**



**Figure 4.26:** Rotor reference frame current  $i_q$  during sensorless operation for  $\omega$  from 650 rad/s to 1300 rad/s and  $\omega$  from 1300 rad/s to 650 rad/s



**Figure 4.27:** Rotor speed during sensorless operation for  $\omega$  from 650 rad/s to 1300 rad/s and  $\omega$  from 1300 rad/s to 650 rad/s

## 4.6 Conclusion

Throughout this chapter the simulation results for the BLDC machine currents, voltage, speed and rotor electrical angle were plotted in order to demonstrate the validity and functionality of a model based back-emf observer on a high-speed low power BLDC machine. From tabulated simulation results it can be concluded that the model based back-emf observer works well in estimating the rotor electrical angle during sensored operation throughout the whole machine's speed range. Furthermore, successful transition from sensored to sensorless operation was achieved with minimal transition spikes and deterioration to the estimates for both back-emf and rotor electrical angle. Lastly successful speed control during sensorless operation was also achieved with minimum settling time and overshoot.

## 5. Experimental Setup

### 5.1 Introduction to Experimental Setup

This chapter illustrates the implementation of the electrical and mechanical hardware and software designs used as part of an experimental setup to test the RFOC scheme on a BLDC machine. Furthermore, detailed analysis about the schematic diagrams, PCB designs and the software algorithm used will be discussed.

### 5.2 The BLDC Machine

#### 5.2.1 Construction of the BLDC Machine

The BLDC machine together with the PMSM, is a synchronous machine (Derammelaere, et al., 2016). The stator of a BLDC machine is practically identical to the stator of an induction machine where several thin laminations of steel are stacked together forming a hollow cylinder (Chau, 2015). Coils of insulated wire are then inserted into slots inside the stator core. The only purpose of the stator in a BLDC machine is to generate the magnetic field necessary for speed/position control. The rotor of the BLDC machine is made up of powerful permanent magnets grouped together as pole pairs (Tsai, et al., 2011). The only difference between the BLDC machine and the PMSM is the type of back-emf waveform generated; a BLDC machine has a trapezoidal back-emf while a PMSM has a sinusoidal back-emf (Derammelaere, et al., 2016). An outrunner BLDC machine, as the one used in this dissertation is shown in Figure 5.1, has its stator coils placed on the inside of the machine where usually the rotor would be situated, while the permanent magnets are situated on the outer body of the machine where usually the stator coils would be situated. This makes the whole outer body of the machine rotate instead of just having the rotor shaft rotate.



**Figure 5.1: BLDC machine used for the experimental setup**

### 5.2.2 Electrical and Mechanical Parameters of the BLDC Machine

Table 5.1 below shows the most important electrical and mechanical parameters of the BLDC machine used for the experimental setup.

**Table 5.1: BLDC machine parameters**

Parameter	Unit	Value
<b>Unit Weight</b>	g	58
<b>Width</b>	mm	45
<b>Length</b>	mm	112
<b>Power</b>	W	187
<b>Maximum Speed (electrical)</b>	Rad/s	18200
<b>Maximum Speed (mechanical)</b>	Rad/s	1300
<b>No. of Poles</b>	N.A.	14
<b>Maximum Voltage</b>	V	11
<b>Maximum Current</b>	A	20
<b>Phase Resistance</b>	$\Omega$	0.0815
<b>Phase Inductance</b>	$\mu$ H	6.5
<b>Full Load Torque</b>	Nm	0.1432

## 5.3 Hardware Design

### 5.3.1 Introduction

The hardware design consists of a SSTP inverter made up from five printed circuit boards (PCB), each of which performs a specific function of the inverter.

### 5.3.2 Power Supply Circuit

Figure 5.2 shows the schematic diagram of the power supply circuit. A +12V and Ground are the inputs of a positive linear regulator whose output is a +5V and Ground. The +5V and Ground output of the linear regulator are supplied to six isolated DC to DC converters. The outputs of the six DC to DC converters are six electrically isolated +12V and Grounds, one for each MOSFET switch. All inputs and outputs of the power supply circuit are listed in Table 5.2. The power supply PCB design and hardware implementation can be seen in Appendix A.1.

**Table 5.2: Power supply circuit inputs and outputs**

Inputs	
Input Name	Input Circuit Designator
Input power for linear voltage regulator	+12V_C/GND_C
Input power for potentiometer (POT)	+3.3V_MC/GND_MC
Outputs	
Output Name	Output Circuit Designator
Linear voltage regulator output	+5V/GND_C
High side 1 isolated +12V output	+12V_H1/GND_H1
High side 2 isolated +12V output	+12V_H2/GND_H2
High side 3 isolated +12V output	+12V_H3/GND_H3
Low side 1 isolated +12V output	+12V_L1/GND_L1
Low side 2 isolated +12V output	+12V_L2/GND_L2
Low side 3 isolated +12V output	+12V_L3/GND_L3
Potentiometer value output from 0-3.3V	POT

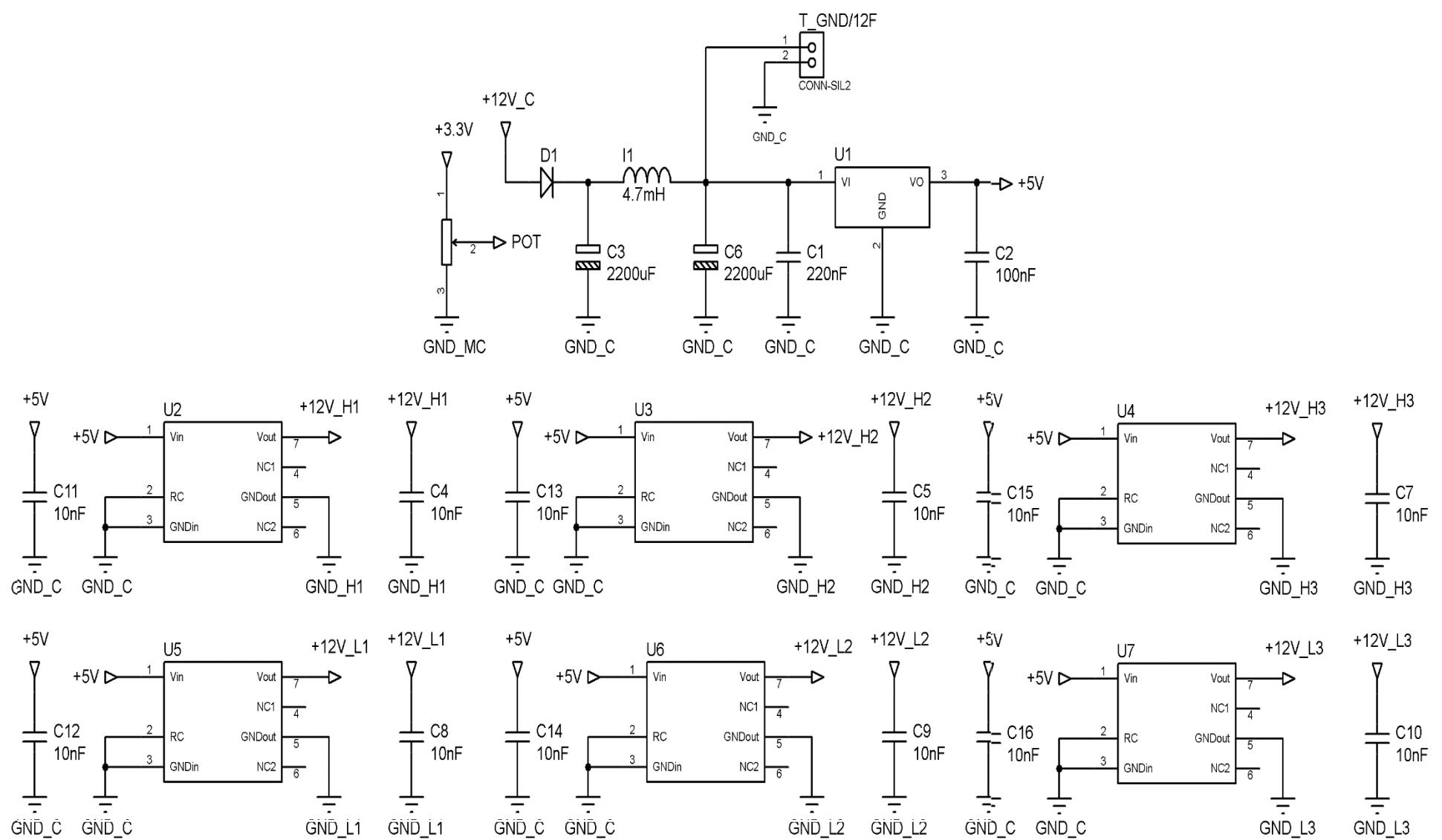


Figure 5.2: Power supply circuit diagram

### 5.3.3 Gate Driver Circuit

Figure 5.3 shows the schematic diagram of the Gate driver circuit. The six electrically isolated +12V and Grounds from the power supply circuit are inputs to the gate driver Integrated Circuits (IC). Each gate driver IC can fire the gate of two MOSFETS; one high side MOSFET and one low side MOSFET, thus each gate driver IC takes two of the six electrically isolated +12V and Grounds, two logic inputs and a +3.3V and Ground as input power from a microcontroller. When one of the logic inputs goes high, its respective +12V and Ground from the electrically isolated DC to DC converter is outputted. All inputs and outputs of the gate driver circuit are listed in Table 5.3. The gate driver PCB design and hardware implementation can be seen in Appendix A.2.

**Table 5.3: Gate driver circuit inputs and outputs**

Inputs	
Input Name	Input Circuit Designator
Input power for gate driver IC	+3.3V_MC/GND_MC
High side 1 isolated +12V output	+12V_H1/GND_H1
High side 2 isolated +12V output	+12V_H2/GND_H2
High side 3 isolated +12V output	+12V_H3/GND_H3
Low side 1 isolated +12V output	+12V_L1/GND_L1
Low side 2 isolated +12V output	+12V_L2/GND_L2
Low side 3 isolated +12V output	+12V_L3/GND_L3
Outputs	
Output Name	Output Circuit Designator
High side 1 gate input voltage	+12V_H1_O/GND_H1
High side 2 gate input voltage	+12V_H2_O/GND_H2
High side 3 gate input voltage	+12V_H3_O/GND_H3
Low side 1 gate input voltage	+12V_L1_O/GND_L1
Low side 2 gate input voltage	+12V_L2_O/GND_L2
Low side 3 gate input voltage	+12V_L3_O/GND_L3

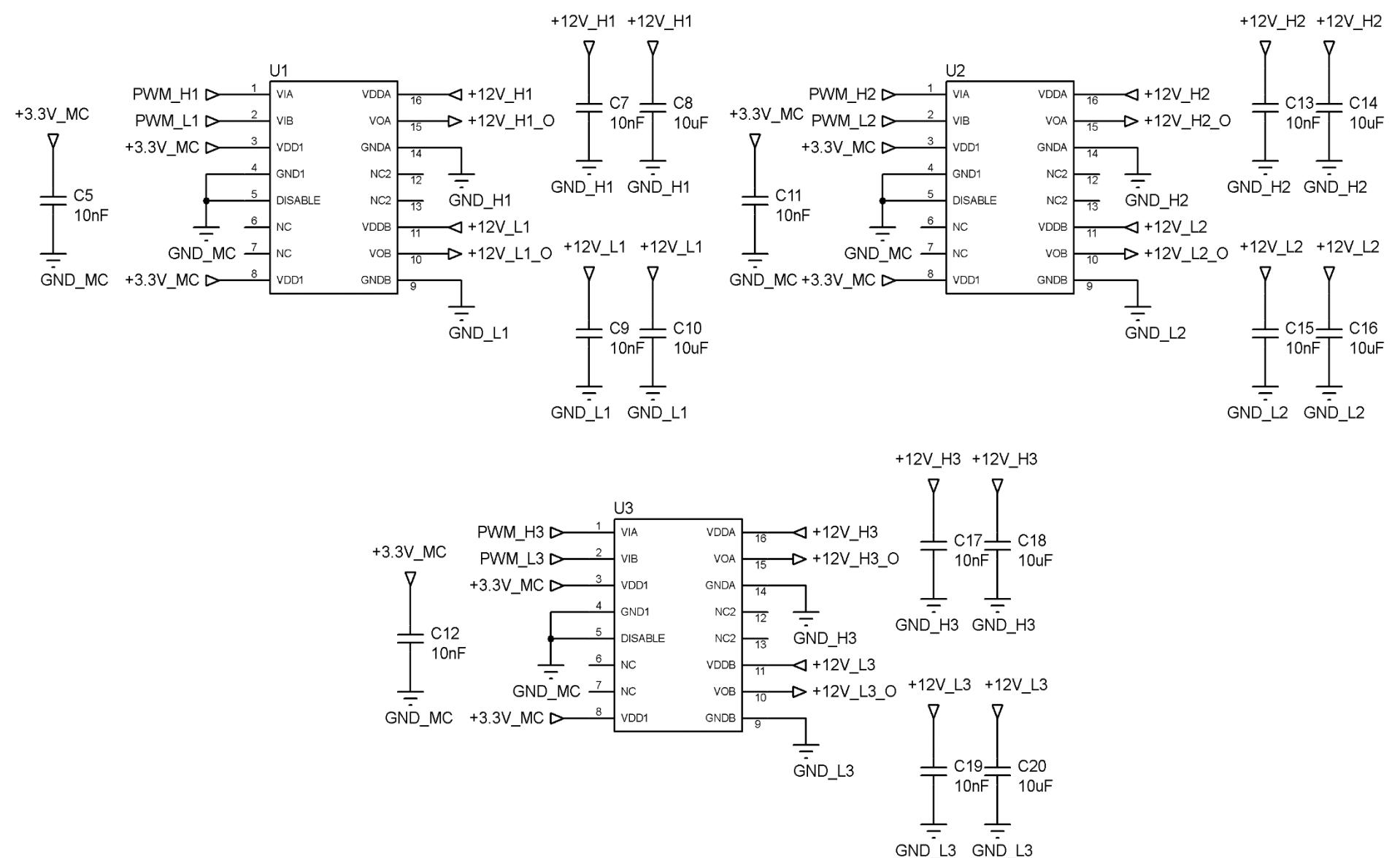


Figure 5.3: Gate driver circuit diagram

### 5.3.4 Three Phase Inverter Circuit

Figure 5.4 shows the schematic diagram of the three-phase inverter circuit. Each gate driver IC output is supplied to its respective MOSFET gate pin. Therefore, if a 1 kHz, 70% duty cycle PWM signal is applied on one of the logic input pins of the gate driver IC the MOSFET connect with the output of that gate driver IC will be switched ON for 0.0007s and switched OFF for the remaining of the cycle i.e. 0.0003s. A Further +12V and Ground are inputs of the three-phase inverter circuit as power input of the three-phase inverter. All inputs and outputs of the three-phase inverter circuit are listed in Table 5.4. The three-phase inverter PCB design and hardware implementation can be seen in Appendix A.3.

**Table 5.4: Three phase inverter circuit inputs and outputs**

Inputs	
Input Name	Input Circuit Designator
Input power for the three-phase inverter	+12V_M/GND_M
High side 1 gate input voltage	+12V_H1_O/GND_H1
High side 2 gate input voltage	+12V_H2_O/GND_H2
High side 3 gate input voltage	+12V_H3_O/GND_H3
Low side 1 gate input voltage	+12V_L1_O/GND_L1
Low side 2 gate input voltage	+12V_L2_O/GND_L2
Low side 3 gate input voltage	+12V_L3_O/GND_L3
Outputs	
Output Name	Output Circuit Designator
Current output phase A	PHASE_A_OUT
Current output phase B	PHASE_B_OUT
Current output phase C	PHASE_C_OUT

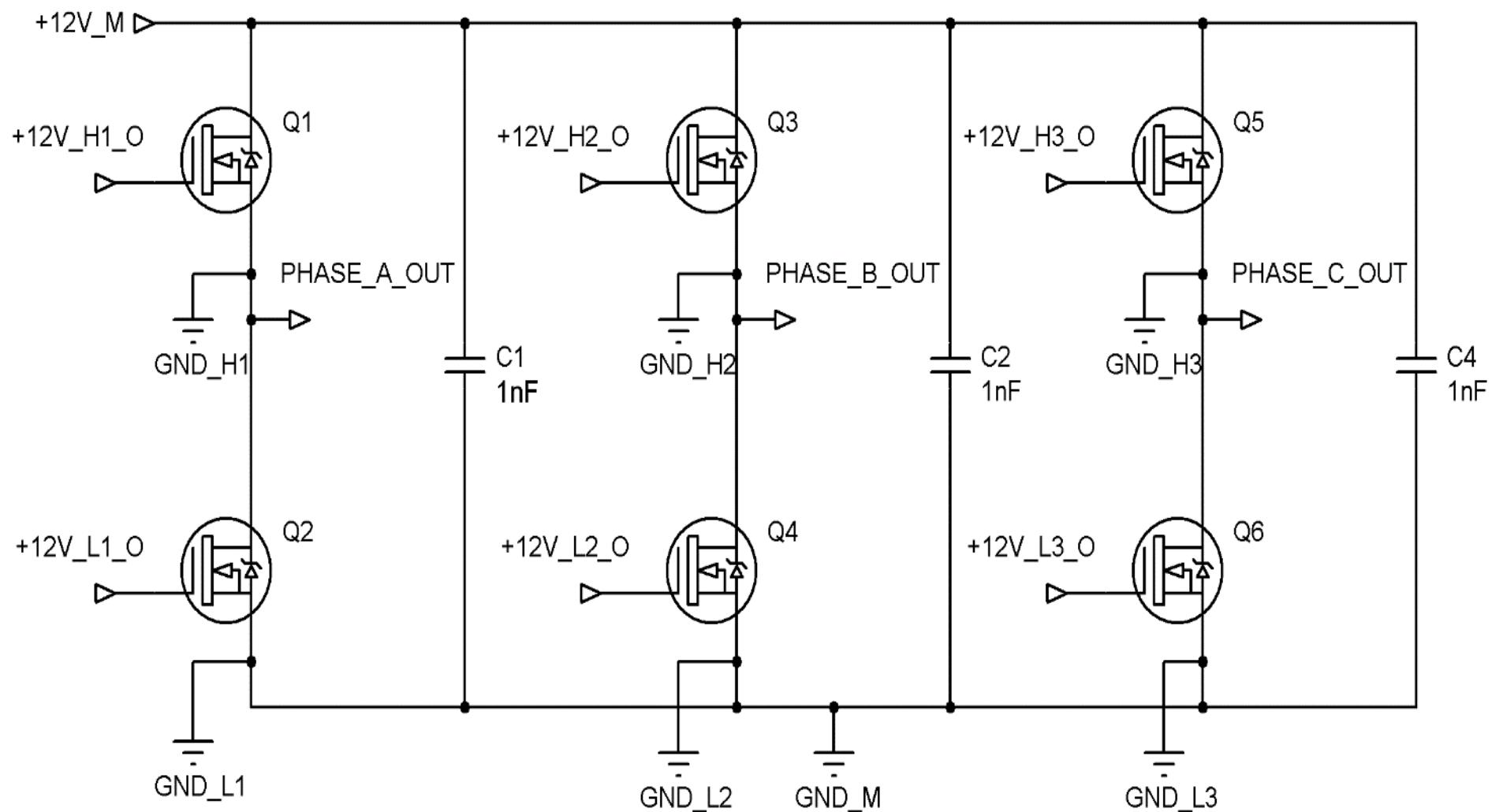


Figure 5.4: Three phase inverter circuit diagram

### 5.3.5 Current Sensing Circuit

Figure 5.5 shows the schematic diagram of the current sensing circuit. The current drawn from the BLDC machine as it exits the inverter, prior entering the BLDC machine, it goes inside the current sensing circuit. For each phase of the inverter there is a current sensing IC which translates the input current from the inverter to an output voltage. The output voltage from the current sensing IC is then given an offset and then buffered so as to have a suitable voltage for a microcontroller input. The current then resumes its path into the BLDC machine. The +5V and Ground generated on the power supply circuit are extended on the current sensing circuit and are used as a source of power for the current sensing ICs. All inputs and outputs of the current sensing circuit are listed in Table 5.5. The current sensing PCB design and hardware implementation can be seen in Appendix A.4.

**Table 5.5: Current sensing circuit inputs and outputs**

Inputs	
Input Name	Input Circuit Designator
Input power for the current sensing IC	+5V/GND_C
Current input from inverter leg phase A	PHASE_A_IN
Current input from inverter leg phase B	PHASE_B_IN
Current input from inverter leg phase C	PHASE_C_IN
Outputs	
Output Name	Output Circuit Designator
Current output to BLDC machine phase A	PHASE_A_OUT
Current output to BLDC machine phase B	PHASE_B_OUT
Current output to BLDC machine phase C	PHASE_C_OUT
Measured current in volts for phase A	iA
Measured current in volts for phase B	iB
Measured current in volts for phase C	iC

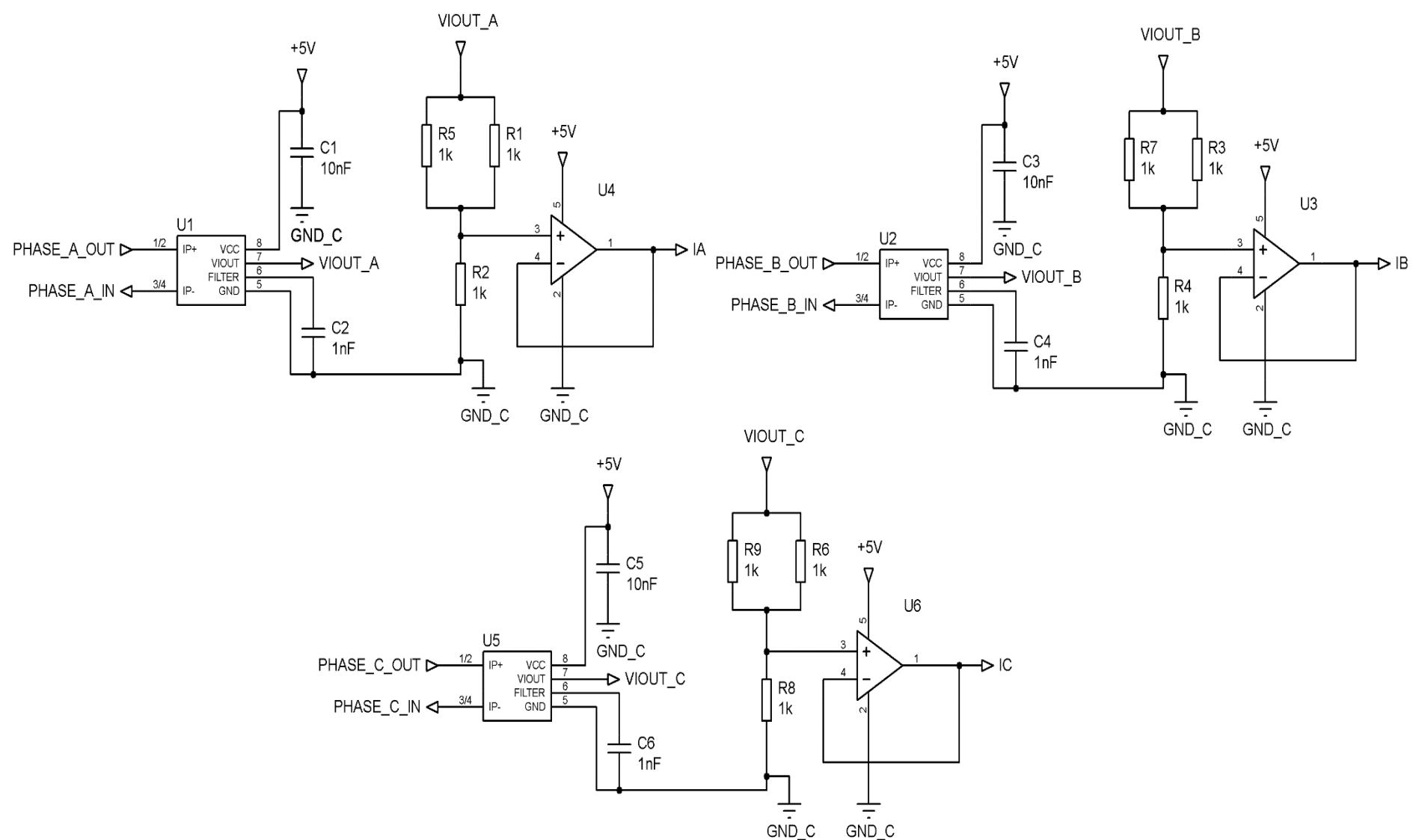


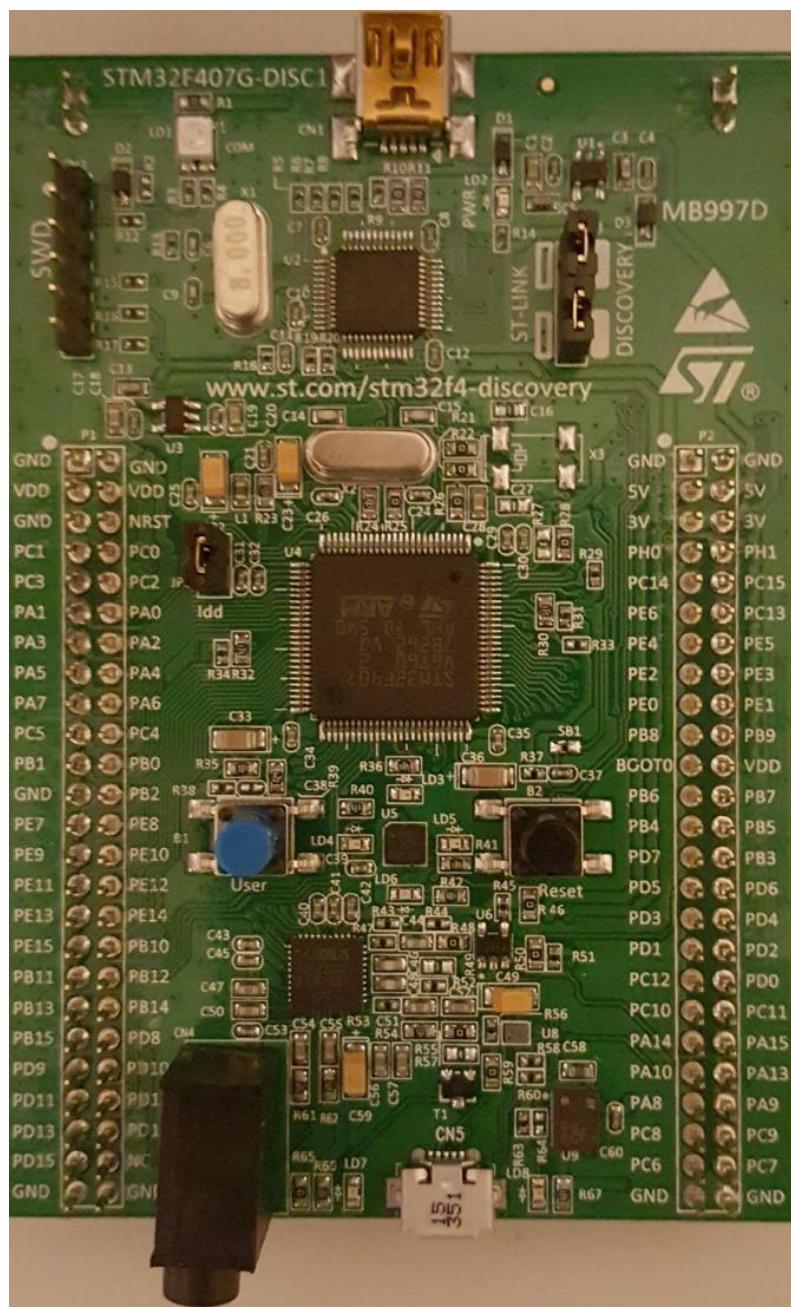
Figure 5.5: Current sensing circuit diagram

### 5.3.6 Microcontroller Circuit

The microcontroller used for the experimental setup is the STM32F407G from STMicroelectronics as shown in Figure 5.6. The microcontroller processes the measured currents output from the current sensing circuit together with the output value from the potentiometer connected on the power supply circuit. Depending on the measured currents, potentiometer reference and the programmed software algorithm (discussed in section 5.4) the microcontroller outputs three sets of PWM signals. The PWM signals are then fed back to the gate driver IC logic inputs, from which such a cycle is repeated. All inputs and outputs of the microcontroller circuit are listed in Table 5.6.

**Table 5. 6: Microcontroller circuit inputs and outputs**

Inputs	
Input Name	Input Circuit Designator
Measured current in volts for phase A	iA
Measured current in volts for phase B	iB
Measured current in volts for phase C	iC
Potentiometer value output from 0-3.3V	POT
Outputs	
Output Name	Output Circuit Designator
High side 1 PWM for gate driver IC logic input	PWM_H1
High side 2 PWM for gate driver IC logic input	PWM_H2
High side 3 PWM for gate driver IC logic input	PWM_H3
Low side 1 PWM for gate driver IC logic input	PWM_L1
Low side 2 PWM for gate driver IC logic input	PWM_L2
Low side 3 PWM for gate driver IC logic input	PWM_L3

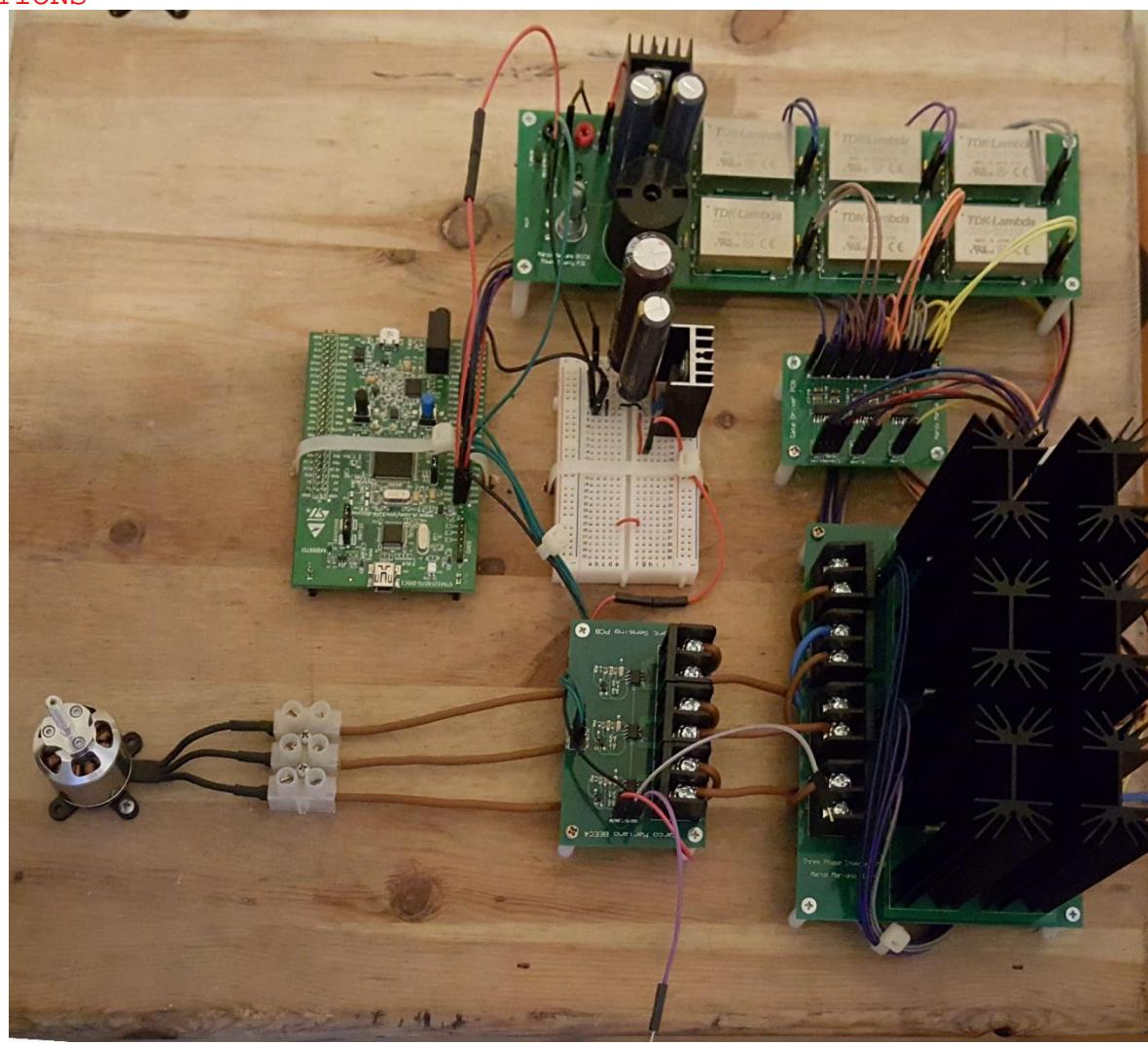


**Figure 5.6: STM32F407G microcontroller used for the experimental setup**

Further pictures showing the PCB design of each circuit schematic and the hardware implementation of each PCB design can be found in:

- Power Supply PCB – Appendix A.1 page 110
- Gate Driver PCB – Appendix A.2 page 111
- Three-phase Inverter PCB – Appendix A.3 page 112
- Current Sensing PCB – Appendix A.4 page 113

Figure 5.7 shows the complete experimental setup after connecting all PCBs together.



**Figure 5.7: Experimental setup**

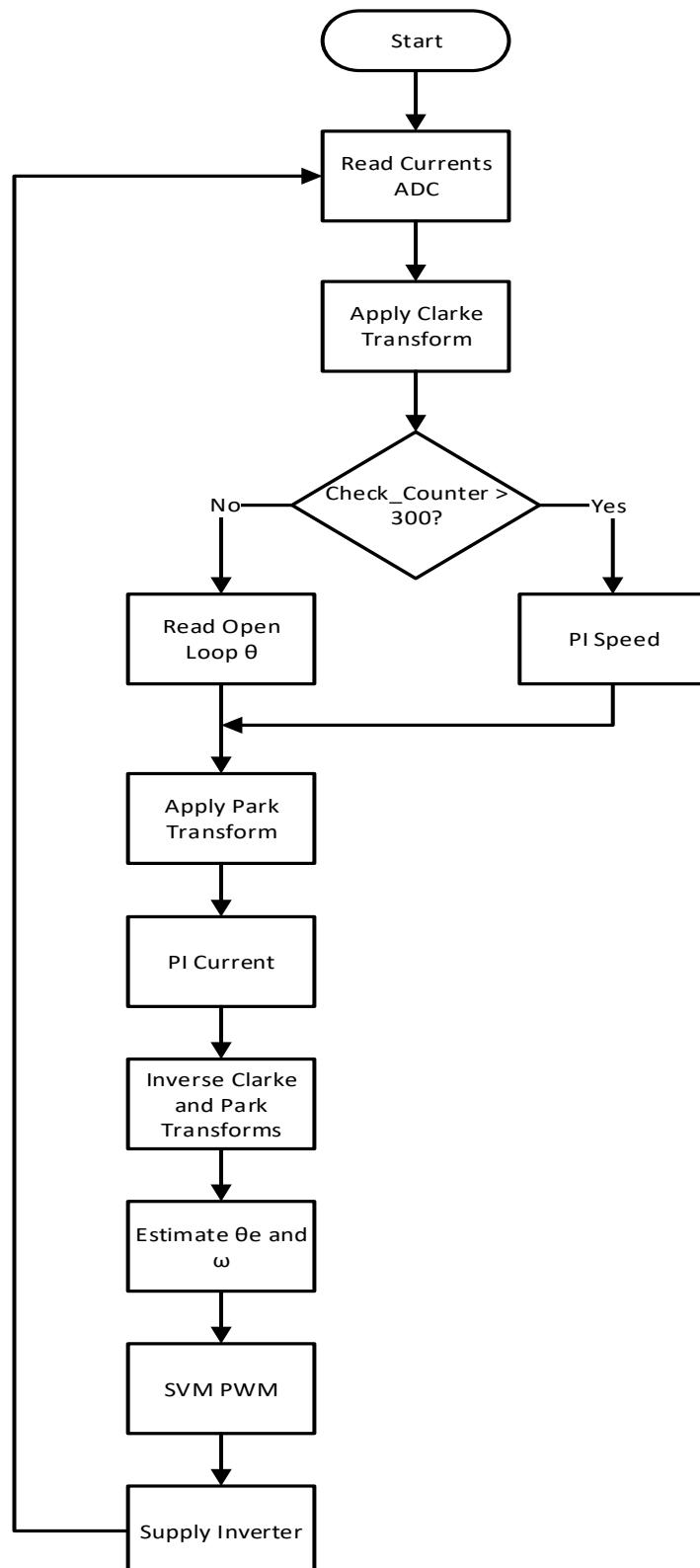
## 5.4 Software Design

### 5.4.1 Introduction

After receiving current and speed information from the hardware, the microcontroller performs an algorithm with which it decides the duty cycle requirement for the three sets of PWM outputs. The implementation of the software algorithm can be subdivided into two; the RFOC scheme software implementation, which performs the Clarke and Park transforms together with the speed and current control loops and back-emf observer, and the SVM software algorithm which decides the timing intervals for the three sets of PWM duty cycles.

#### 5.4.2 RFOC Software Algorithm

As mentioned in Chapter 3 the proposed control algorithm to control the BLDC machine is the RFOC scheme. Figure 5.8 shows a flow chart of the RFOC scheme software algorithm together with the back-emf observer.

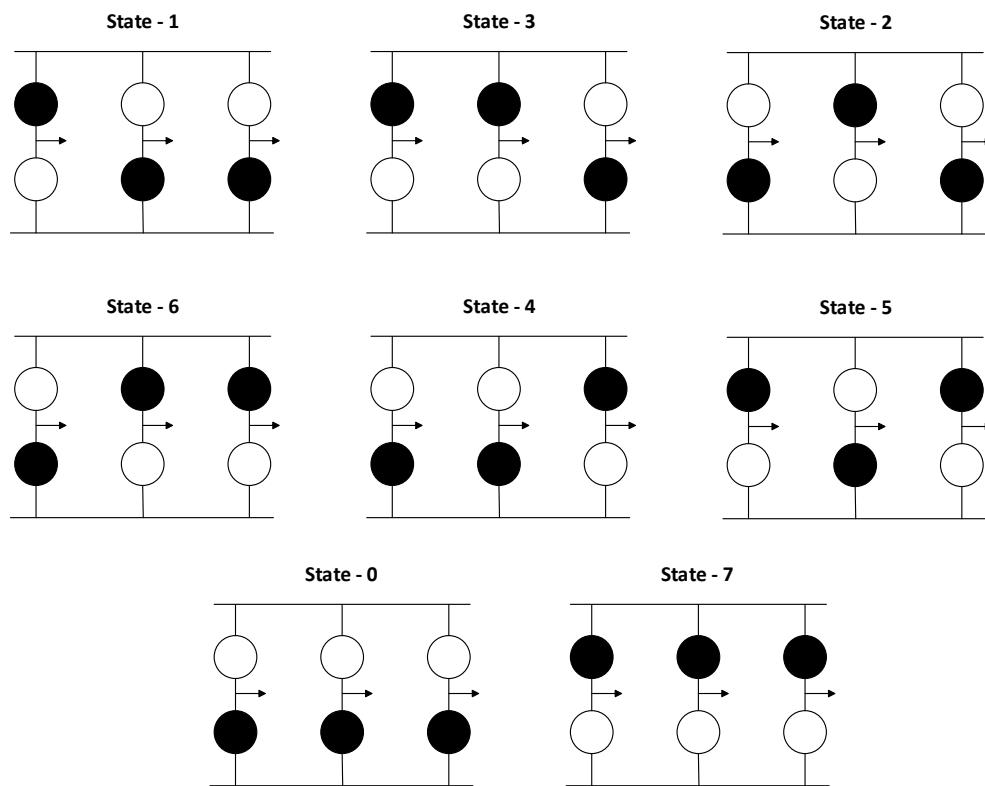


**Figure 5.8: Flow chart diagram of the RFOC scheme software algorithm**

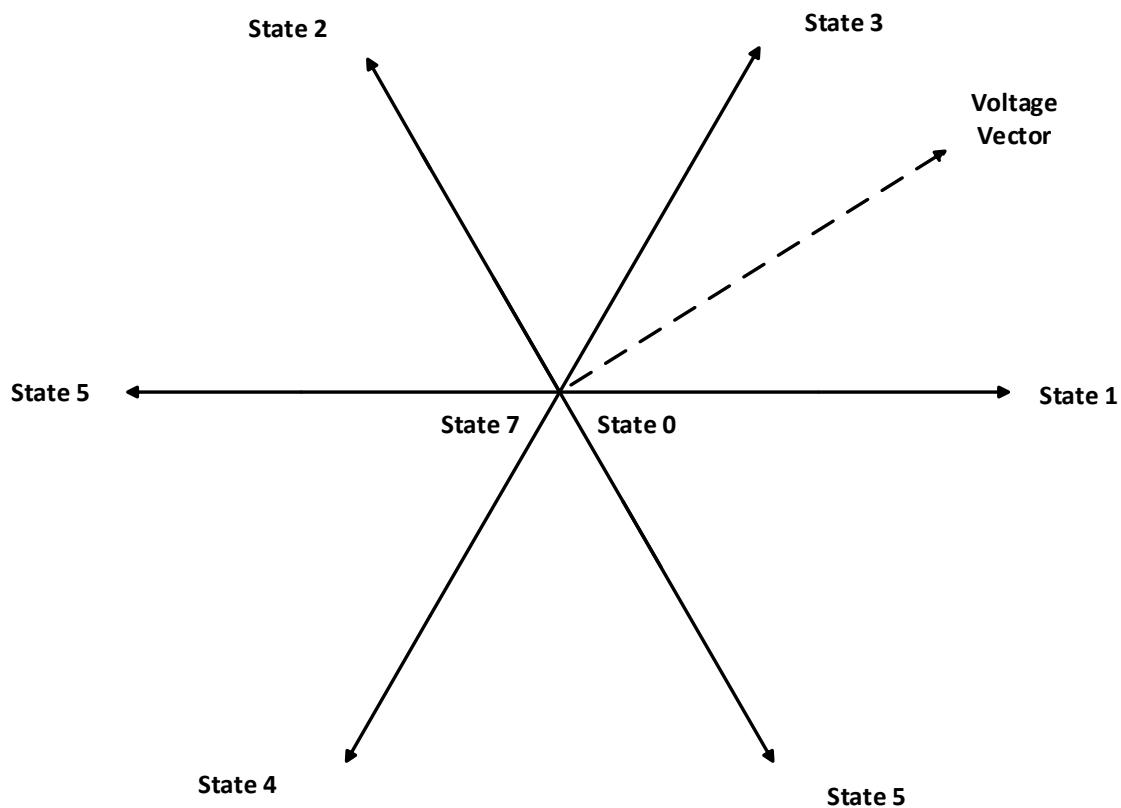
### 5.4.2 Space Vector Modulation Software Algorithm

After obtaining the three phase voltage vectors  $v_a$ ,  $v_b$  and  $v_c$  its the SVM software algorithm that decides the PWM duty cycles. Unlike conventional PWM techniques who only control the phase voltage  $120^0$  offsets between phases A, B and C the SVM controls the voltage vector directly (Wang, et al., 2017), (Zeng, et al., 2017). The inverter can be driven to eight states; 6 voltage vectors and two null vectors as shown in Figure 5.9. Each vector can be mapped on a vector diagram as shown in Figure 5.10 where:

- State “1” causes the voltage vector to have an angle of  $0^0$
- State “2” causes the voltage vector to have an angle of  $60^0$
- State “3” causes the voltage vector to have an angle of  $120^0$
- State “4” causes the voltage vector to have an angle of  $180^0$
- State “5” causes the voltage vector to have an angle of  $240^0$
- State “6” causes the voltage vector to have an angle of  $300^0$
- State “0” i.e. Null state causes the voltage vector to have an angle of  $0^0$
- State “7” i.e. Null state causes the voltage vector to have an angle of  $0^0$

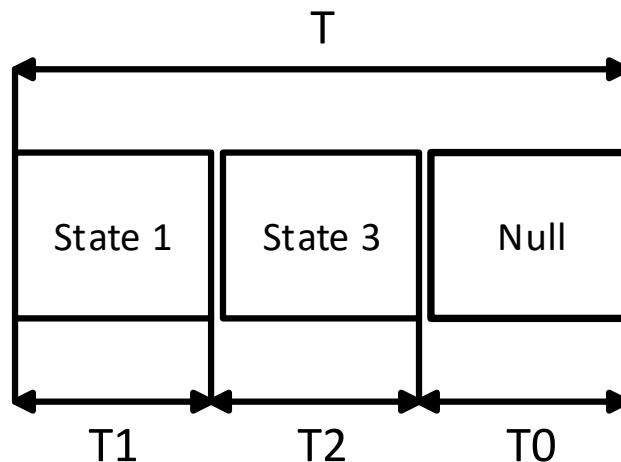


**Figure 5.9: Inverter eight useful switching states**



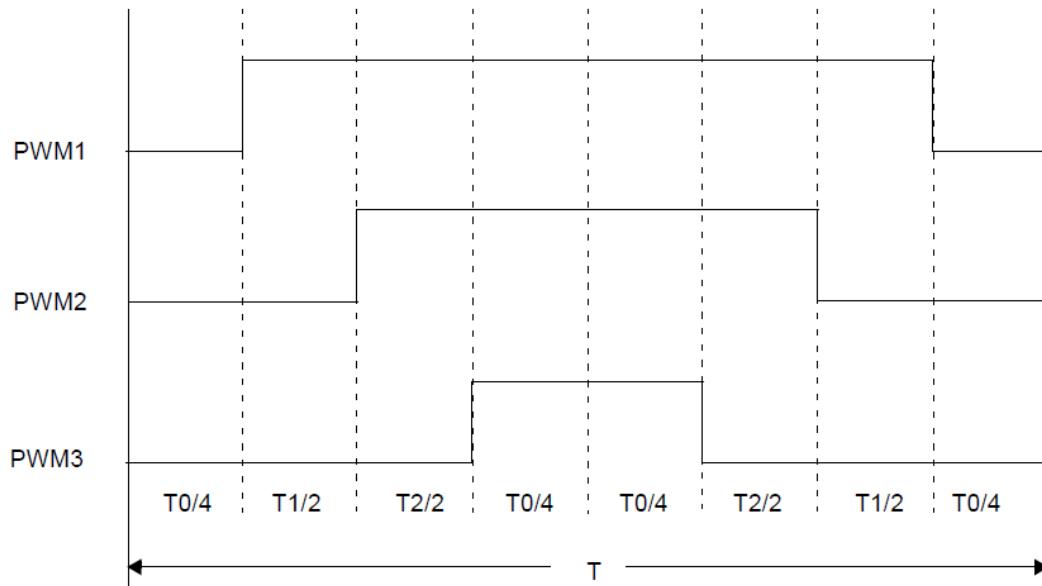
**Figure 5.10: Vector diagram**

The output voltage vector is created by repeatedly switching between adjacent vectors and the null vector. Therefore, the output angle  $\theta$  is determined by the relative “on” time between two adjacent vectors while the output magnitude of the voltage vector is determined by relative “on” time between two adjacent vectors and the null vectors (Ahmed, et al., 2016), (Deng, et al., 2016). For instance if an output magnitude of 0.5 with an output  $\theta$  of  $30^\circ$  is needed, apply an equal relative “on” time between state “1” and state “3” and the null vector as shown in Figure 5.11.



**Figure 5.11: Switching times between two adjacent vector and the null vector**

By shifting the position of which vector is switched first or which null switching state is used, various advantages on the inverter drive could be achieved. For example, the choice of only using the switching state 0 i.e. null vector with bottom MOSFETS on, can reduce the amount of switching losses by 33% thus increasing the inverter lifetime (Zeng, et al., 2017), (Ahmed, et al., 2016). The switching algorithm used in this dissertation is the alternate-reversed sequence. This switching algorithm method first runs the sequence forward and then the next time it runs, it is run in reverse while alternating between null states “0” and “7”. A pattern sequence of the alternate-reversed switching algorithm is shown in Figure 5.12. The complete STM32 software algorithm code can be found attached in Appendix B page 114.



**Figure 5.12: Alternate-reverse switching sequence (Zambada & Deb, 2010)**

## 5.5 Conclusion

Throughout this chapter, a comprehensive explanation of the implementation of the hardware and software used as an experimental setup to test the RFOC scheme of a BLDC machine was discussed. The construction and characteristics of the BLDC machine used in this dissertation were analysed, furthermore, all the necessary circuit schematic diagrams, PCB designs, PCB design hardware implementation and software design were explained.

## 6. Experimental Results

### 6.1 Introduction to Experimental Results

This chapter illustrates the results obtained from different tests conducted on the experimental setup. Results for the BLDC stator currents, stator voltages, speed and rotor angular position are presented for the BLDC machine operation during open-loop, sensorless changeover, and sensorless operation.

### 6.2 Experimental Results – Open Loop Operation

#### 6.2.1 Introduction

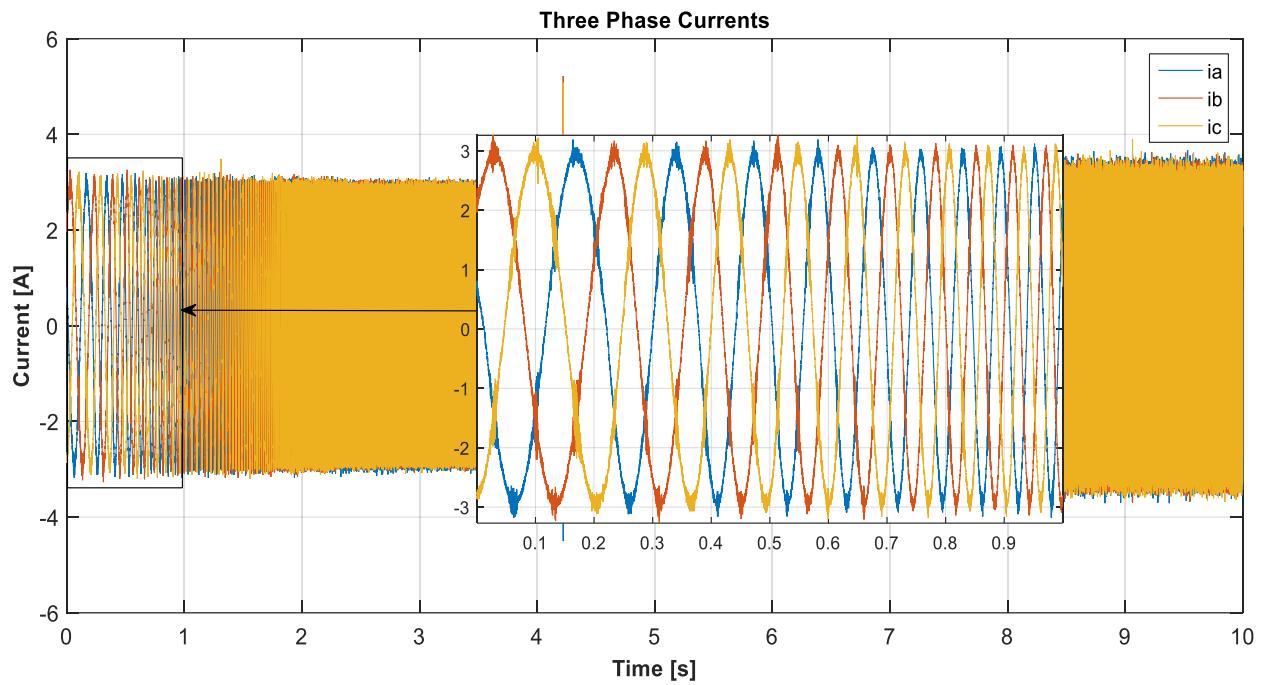
In order for the back-emf observer to start estimating the rotor angular position, the BLDC machine must first accelerate in such a way that the back-emf of the BLDC is sufficient for estimation purposes. This is achieved in the open loop operation where a fictitious rotor angular position is used to generate the sinusoidal based PWM to start the BLDC machine from standstill. During open loop operation the BLDC machine reaches:

1. A maximum mechanical speed of 75 rad/s or 1050 rad/s of electrical speed
2. Open loop operation takes 4.22s (Time required to reach sufficient speed for estimation)

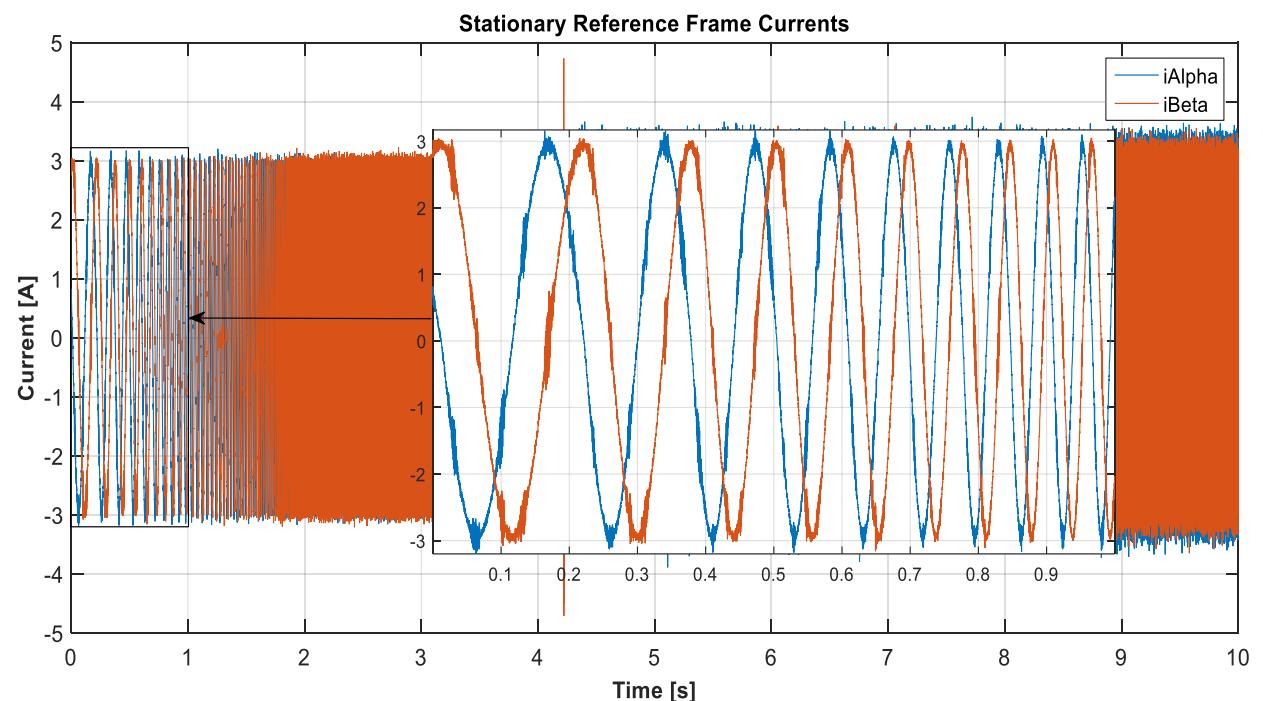
Figures 6.1 to 6.5 show the experimental results obtained for the three-phase currents, stationary reference frame currents  $i_\alpha$  and  $i_\beta$ , synchronous reference frame currents  $i_d$  and  $i_q$  and rotor speed during open loop operation respectively. Furthermore, Figures 6.6 to 6.9 show the experimental results obtained for the back-emf observer during open loop operation for  $i_\alpha$  and  $\hat{i}_\alpha$  convergence,  $i_\beta$  and  $\hat{i}_\beta$  convergence, BEMF $\alpha$  together with BEMF $\beta$  and the rotor angle generation respectively.

## 6.2.2 Open Loop Operation Results

Figure 6.1 shows stator three phase currents  $i_a$ ,  $i_b$  and  $i_c$  where they reach a peak of  $\pm 3$  A. As the machine is building up speed during the open loop operation, the frequency of the three phase currents increases. Figure 6.2 shows the generation of the stationary reference frame currents  $i_\alpha$  and  $i_\beta$  after applying the Clarke transform on the three phase currents. The frequency of currents  $i_\alpha$  and  $i_\beta$  can also be seen increasing as the machine's speed increases. They also reach a peak of  $\pm 3$  A.

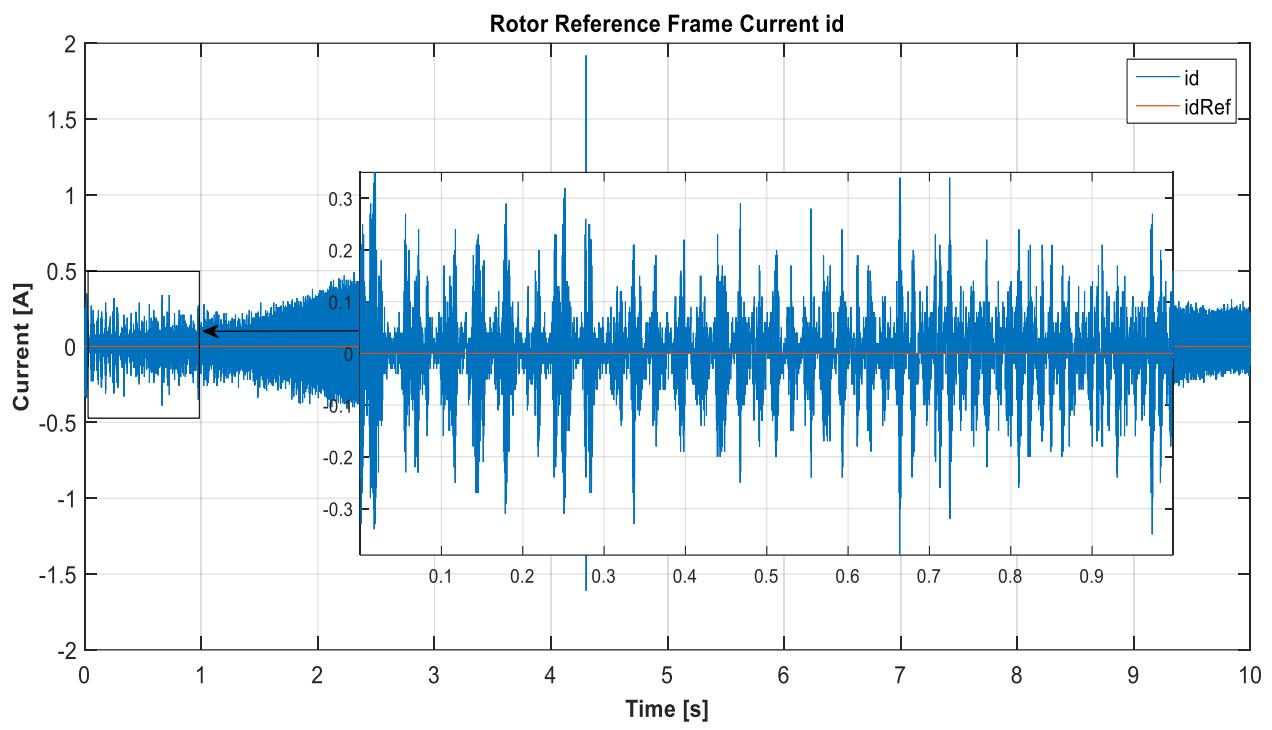


**Figure 6.1: Three phase currents during open loop operation**

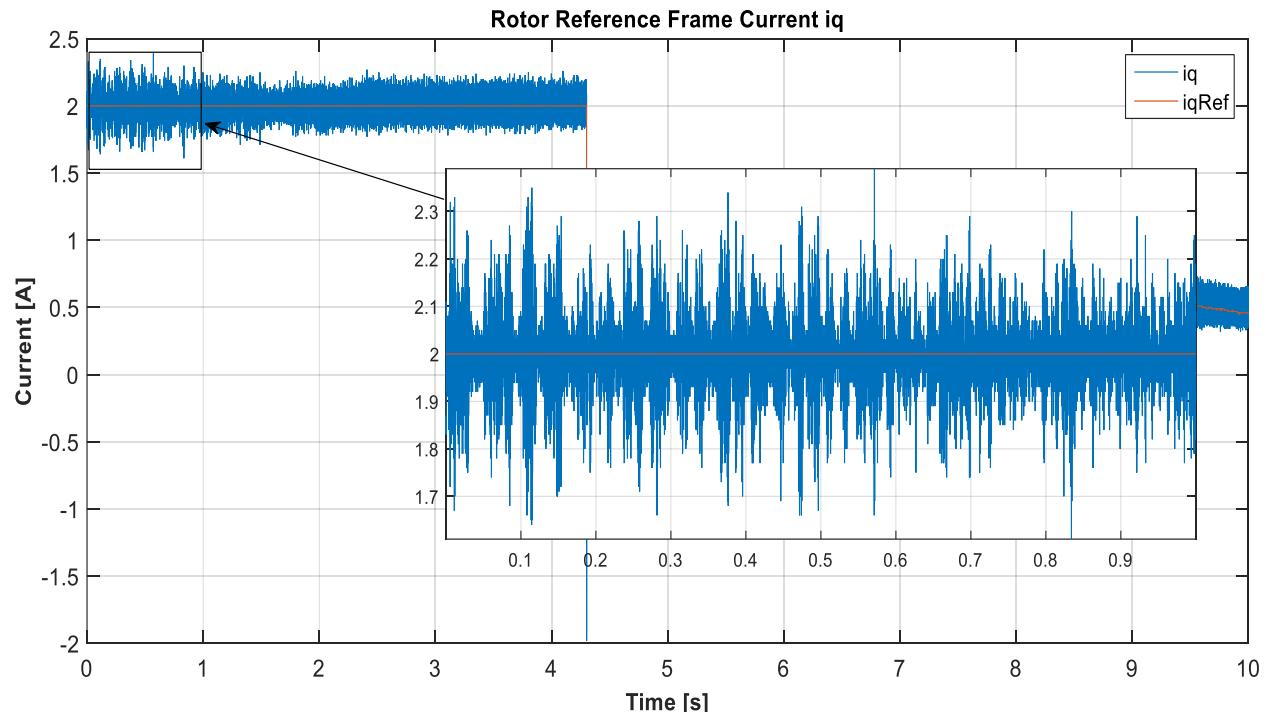


**Figure 6.2: Stationary Reference frame currents during open loop operation**

Figures 6.3 and 6.4 show the generation of the synchronous reference frame currents  $i_d$  and  $i_q$  after applying the park transform and the fictitious rotor electrical angle on the stationary reference frame currents  $i_\alpha$  and  $i_\beta$  respectively. Current  $i_d$  can be seen to settle around the 0 A current reference with peaks reaching  $\pm 0.3$  A while current  $i_q$  can be seen to settle around the 2 A current reference with peaks reaching  $\pm 0.3$  A.

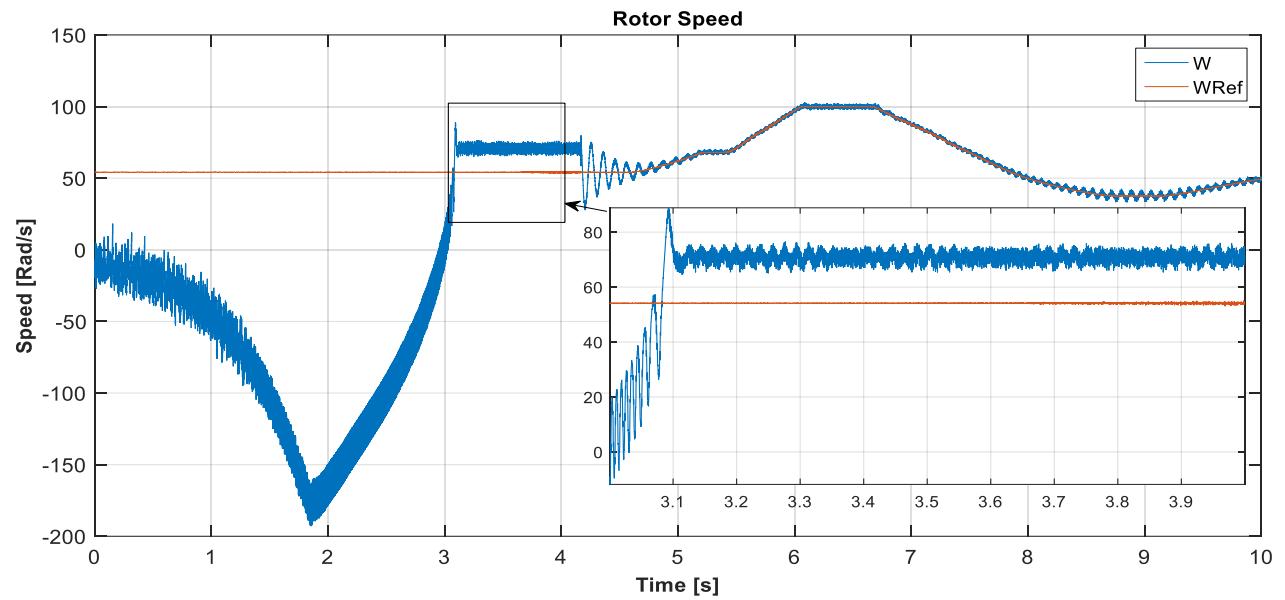


**Figure 6.3: Rotor reference frame current  $i_d$  during open loop operation**

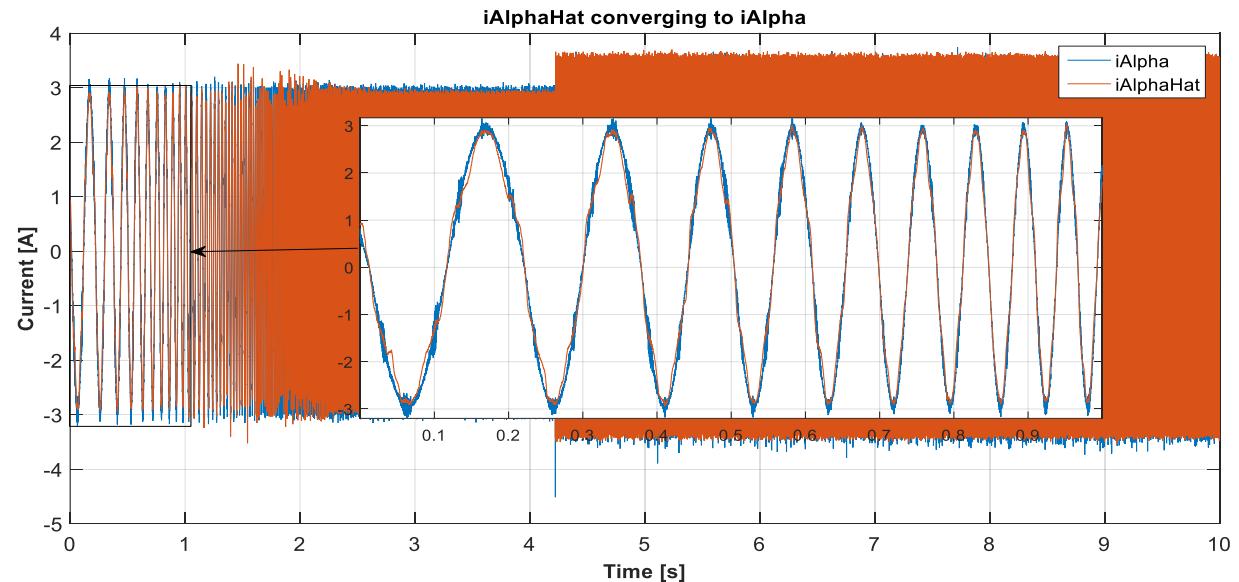


**Figure 6.4: Rotor reference frame current  $i_q$  during open loop operation**

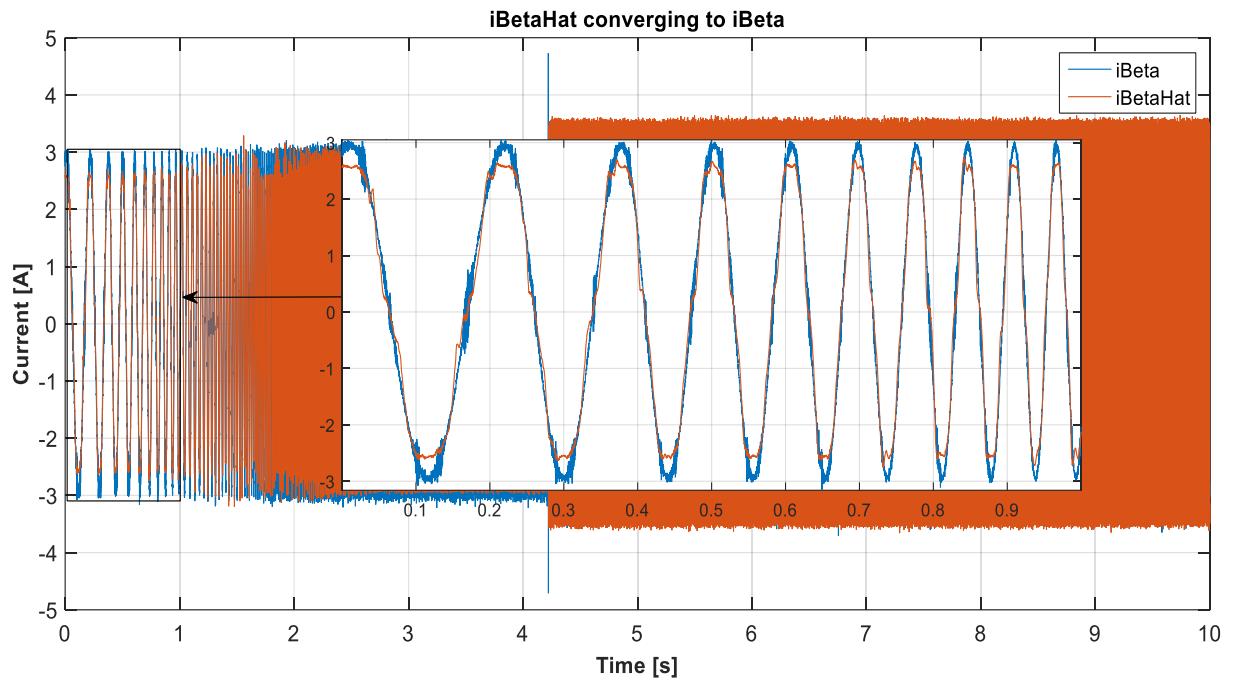
Figure 6.5 shows the rotor speed estimation as it increases from 0 rad/s to 75 rad/s during the open loop operation. The first 2 s in Figure 6.5 show the estimated speed of the machine increasing in the opposite direction, however, this is not actually occurring on the BLDC. Since the machine's speed during start-up is very low, the magnitude of the back-emf is insufficient for an accurate estimate. After 3s sufficient speed is reached such that the estimated speed reflects the operation of the machine. The speed of the machine's rotor reaches 75 rad/s with peaks peaking at  $\pm 77.5$  rad/s. Figures 6.6 and 6.7 show the ability of the back-emf observer PI controller in correcting the error between  $i_\alpha$  and  $\hat{i}_\alpha$  and  $i_\beta$  and  $\hat{i}_\beta$  respectively. As it can be seen, during open loop operation, and between 0 rad/s to 250 rad/s, the observer generates an  $\hat{i}_\alpha$  and an  $\hat{i}_\beta$  identical to  $i_\alpha$  and  $i_\beta$ .



**Figure 6.5: Rotor speed estimation during open loop operation**

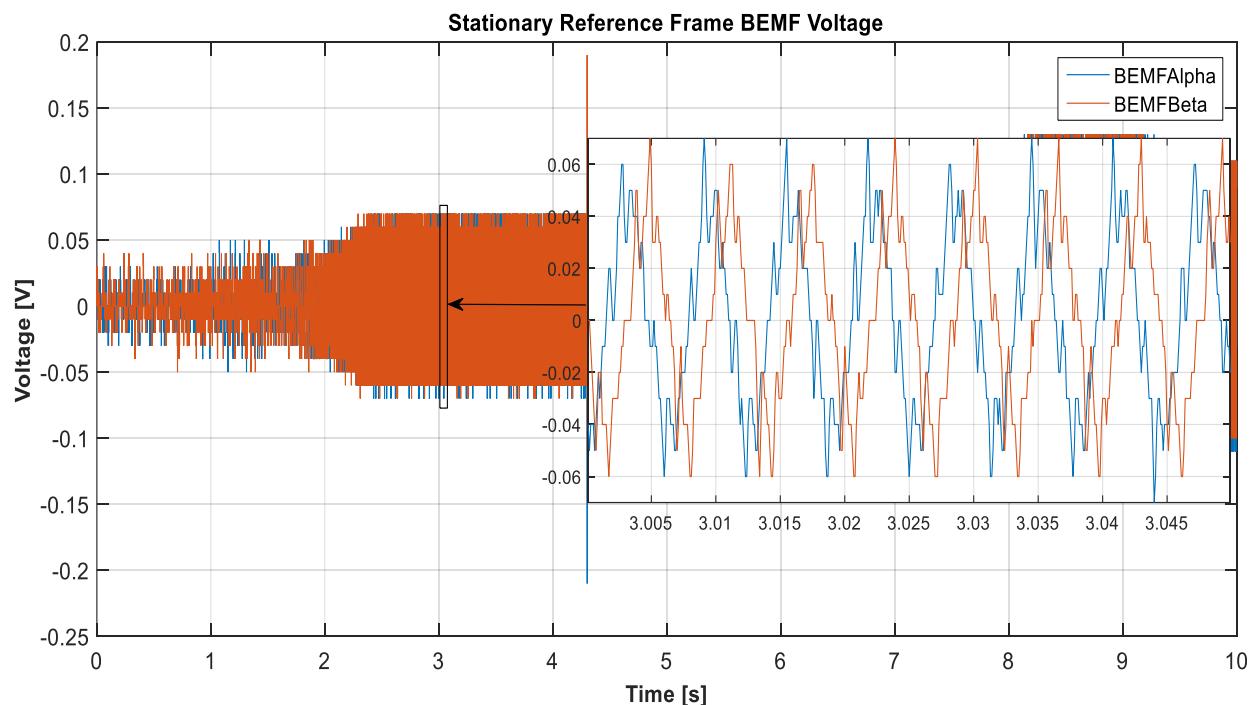


**Figure 6.6:  $i_\alpha$  converging to  $\hat{i}_\alpha$  during open loop operation**

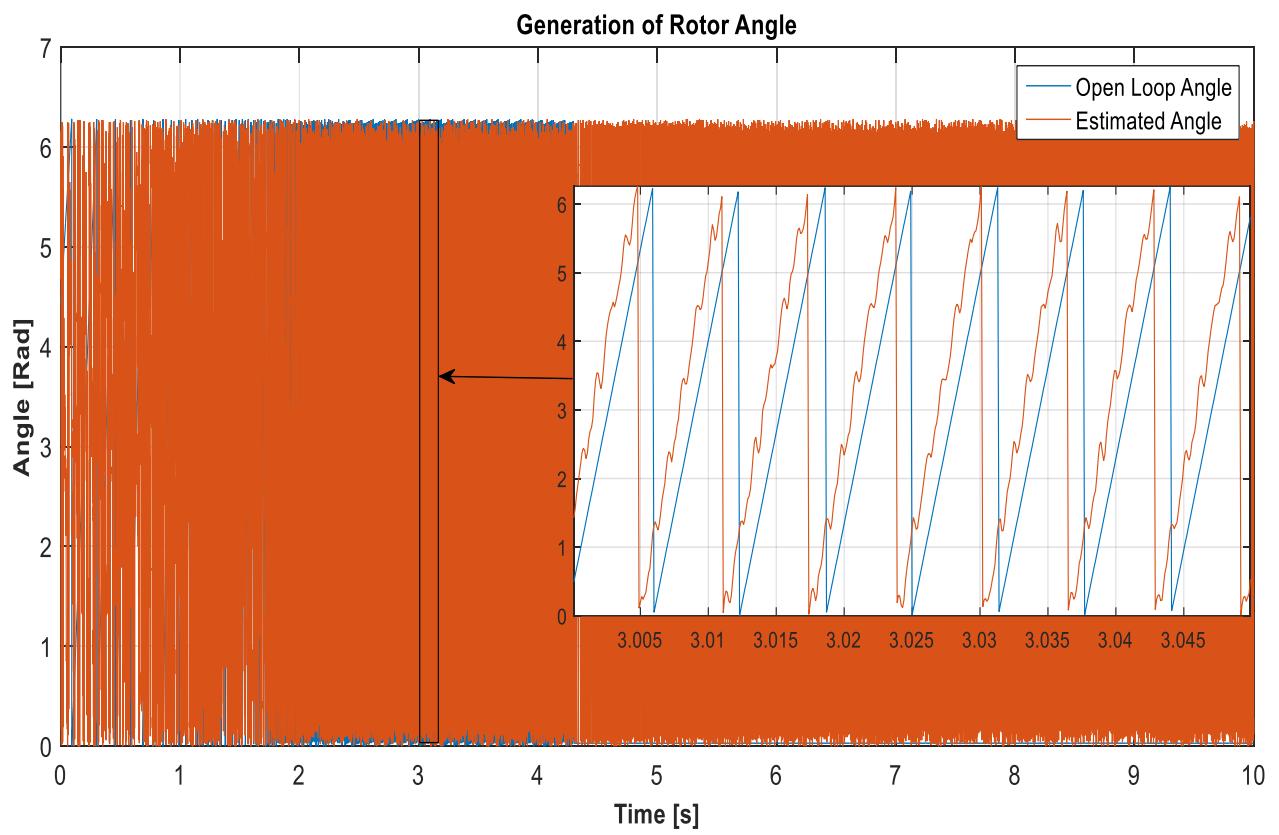


**Figure 6.7:**  $i_\beta$  converging to  $\hat{i}_\beta$  during open loop operation

Figure 6.8 shows the generation of the estimated BEMF $\alpha$  and BEMF $\beta$  from the observer during open loop operation. At the maximum open loop speed of 75 rad/s the estimate back-emf generated reaches a peak of  $\pm 0.06$  V. Figure 6.9 shows the generated rotor position angle from the estimated back-emf. Since the maximum speed reached during open loop operation is only 75 rad/s, the quality of the estimated back-emf is deteriorated, this causes the generation of the rotor position angle to also deteriorate.



**Figure 6.8:** Estimated Back-emf voltages during open loop operation



**Figure 6.9: Rotor position angle during open loop operation**

## 6.3 Experimental Results – Transition

### 6.3.1 Introduction

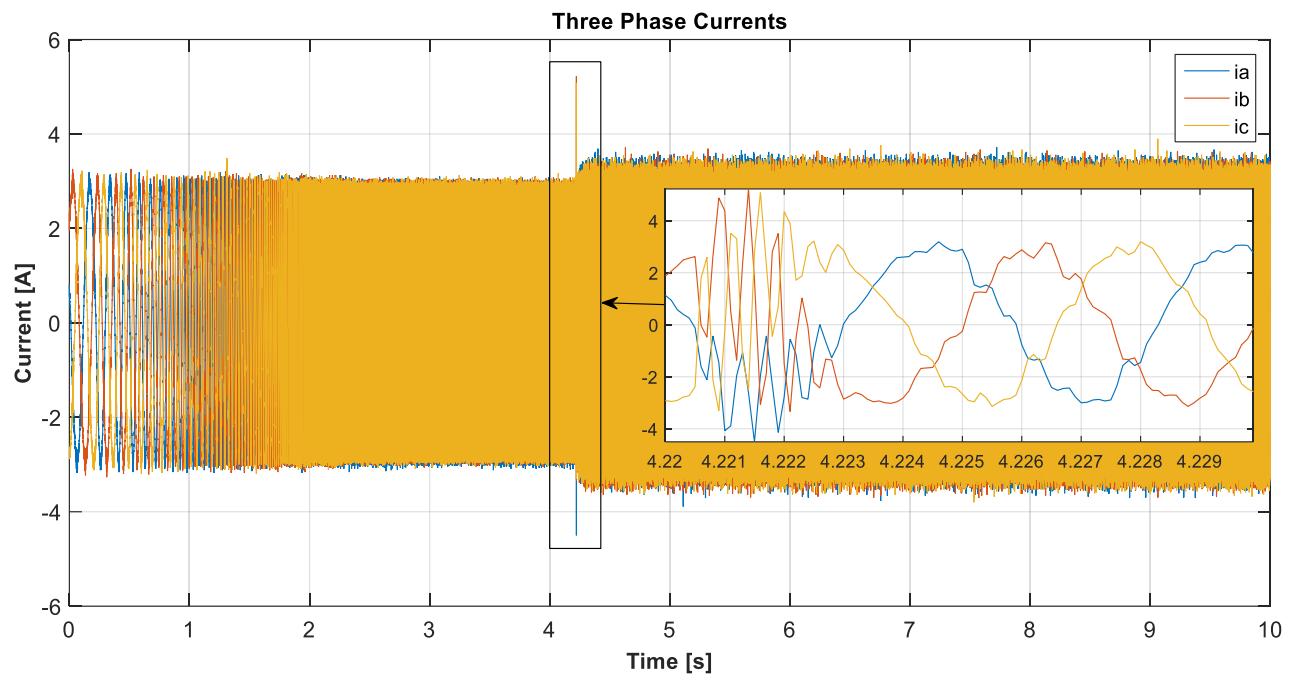
After gaining enough speed from which a correct rotor angle position can be estimated, the BLDC machine is set to transition from using the fictitious rotor angle to the estimated rotor angle. The transition occurs at:

1. Time 4.22s
2. Machine's maximum speed reference set at 100 rad/s

Figures 6.10 to 6.13 show the experimental results obtained for the three-phase stator currents, synchronous reference frame currents  $i_d$  and  $i_q$  and rotor speed during the transition respectively. Furthermore, Figures 6.14 to 6.16 show the experimental results obtained for  $i_\alpha$  and  $i_\beta$  convergence, BEMF $\alpha$  together with BEMF $\beta$  and the rotor angle generation during transition respectively.

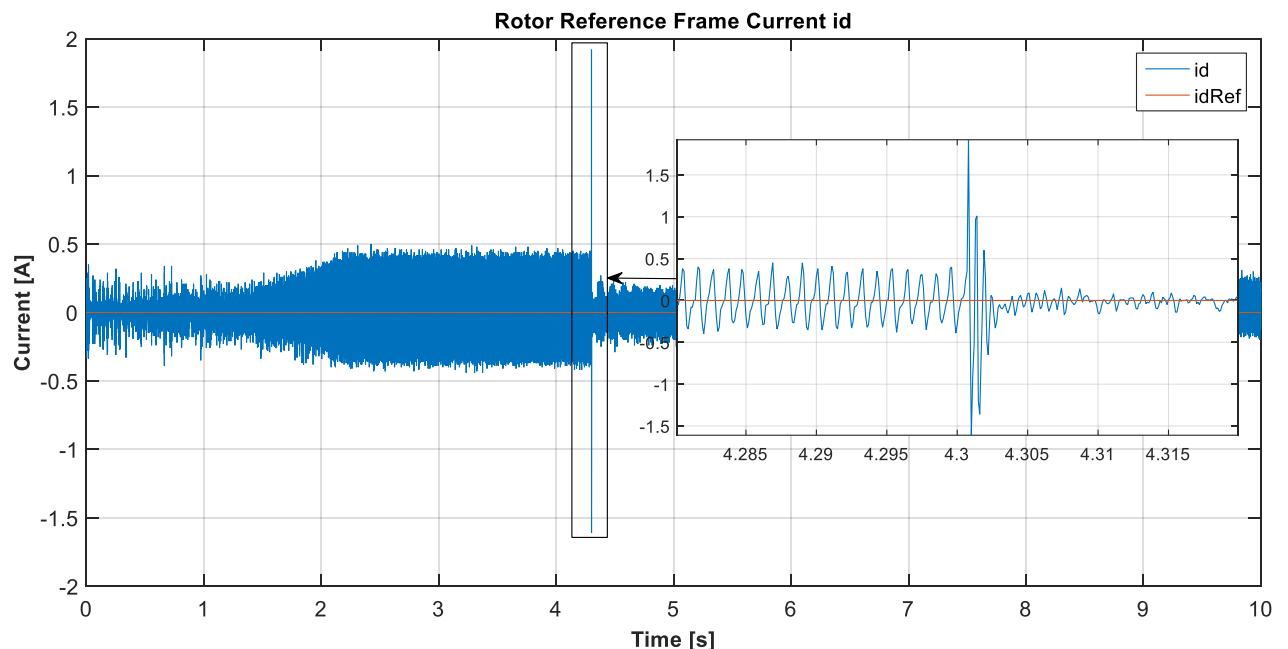
### 6.3.2 Transition Results

Figure 6.10 show the transition effect on the three-phase currents. During transition the quality of the three-phase currents deteriorates reaching peaks of  $\pm 5$  A. The transition dynamics settle after 0.003s.

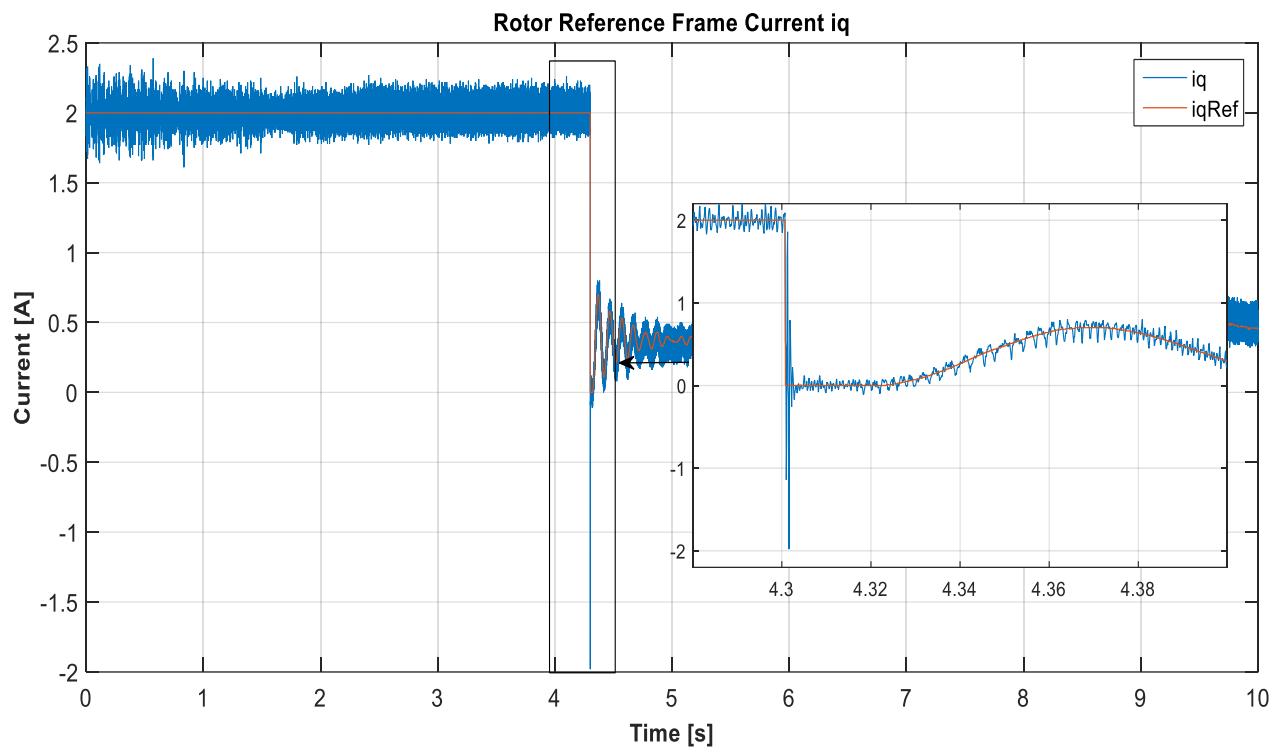


**Figure 6.10: Three phase currents during transition**

Figures 6.11 and 6.12 show the effects of the transition on the reference frame currents  $i_d$  and  $i_q$  respectively. During transition current  $i_d$  peaks at  $\pm 1.5$  A and settles back to 0 A after 0.003s, while current  $i_q$  which now depends on the speed PI controller peaks at  $-2$  A and settles back to its reference position after 0.003s.

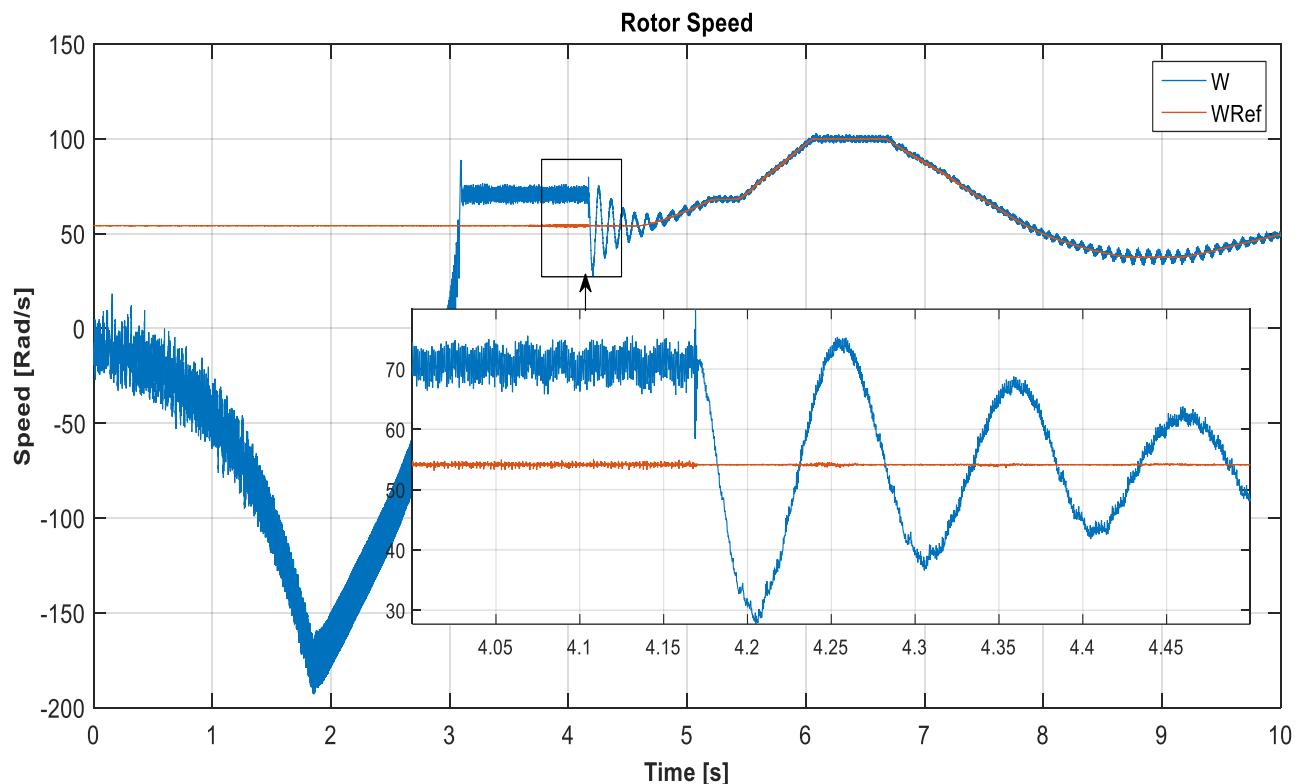


**Figure 6.11: Rotor reference frame current  $i_d$  during transition**



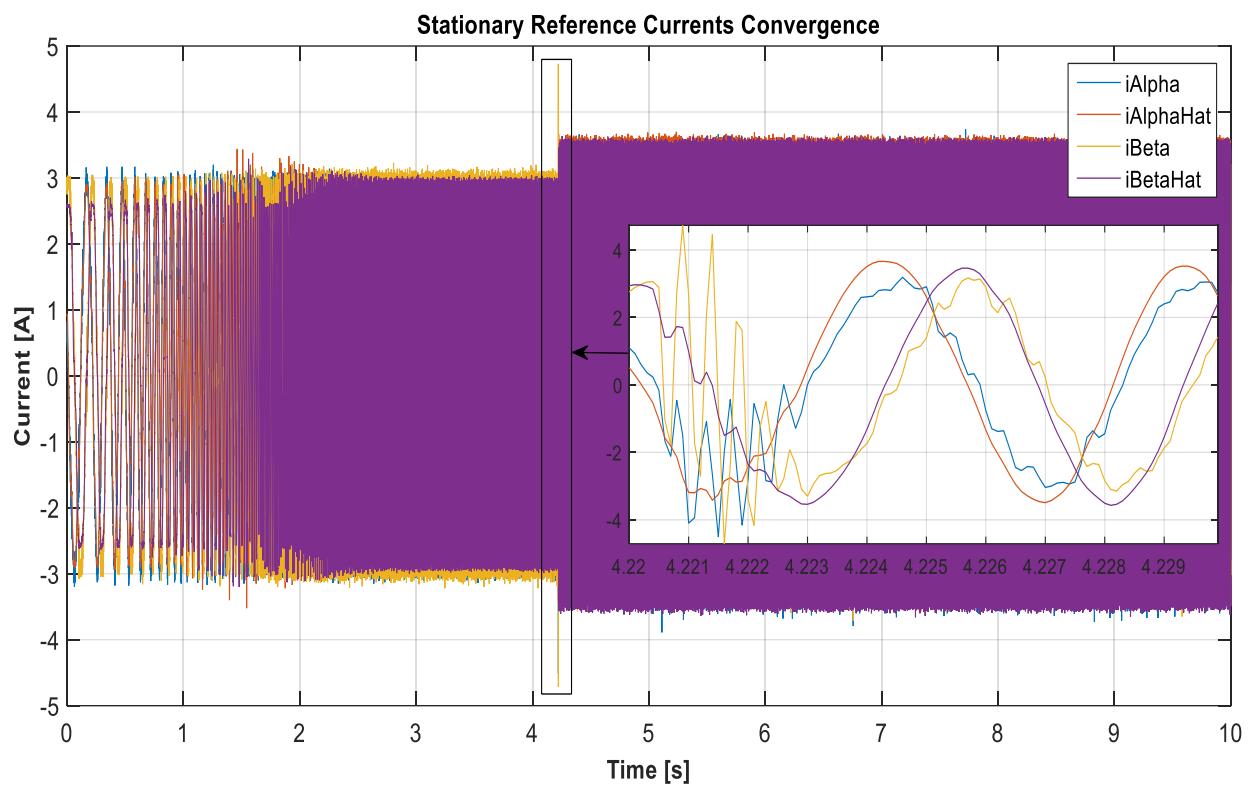
**Figure 6.12: Rotor reference frame current  $i_q$  during transition**

Figure 6.13 show the transition dynamics of the machine rotor speed. After transition the machine's speed reference is set to 52 rad/s. During transition the machine's speed oscillates from a maximum of +75 rad/s to a minimum of +28 rad/s. The transition incurred oscillations decrease until settling at 52 rad/s after 0.6s.

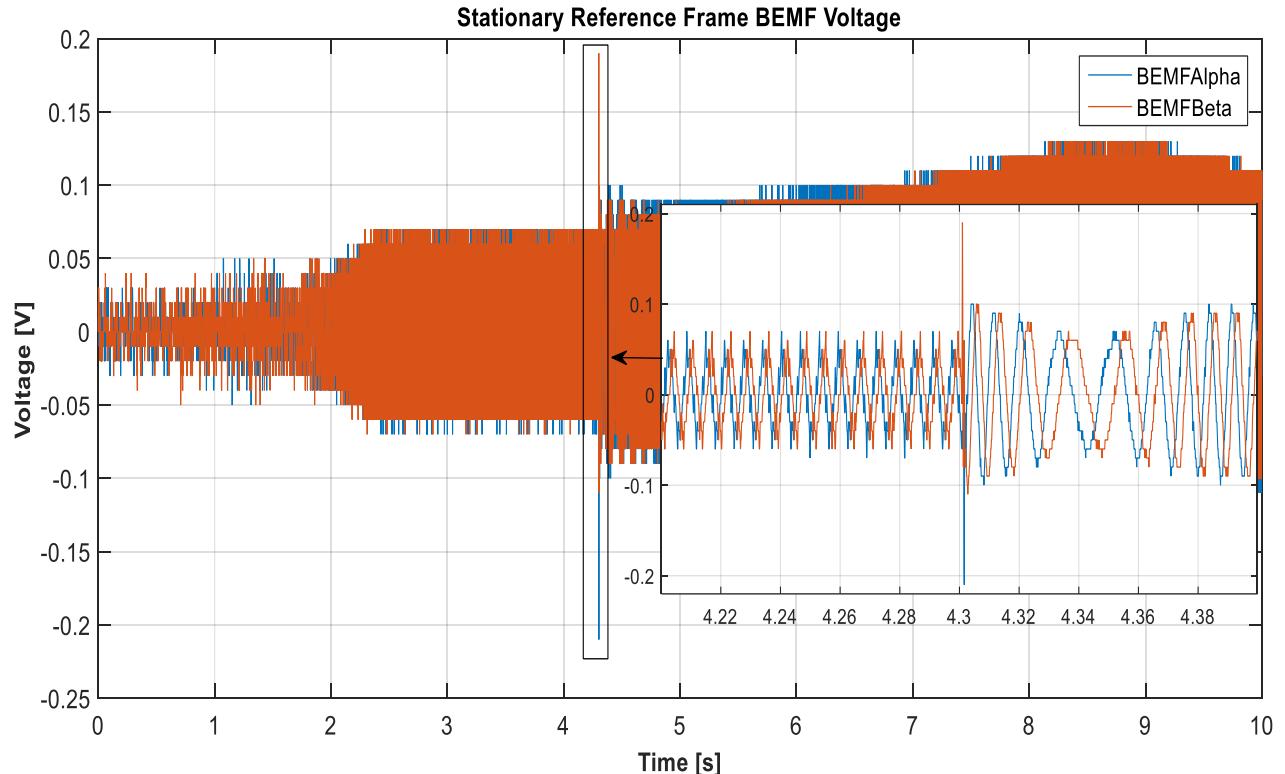


**Figure 6.13: Rotor speed during transition**

Figure 6.14 and 6.15 show the effect of the transition on the back-emf observer in estimating  $\hat{i}_\alpha$  and  $\hat{i}_\beta$  and BEMF $\alpha$  and BEMF $\beta$ . During transition the estimates for  $\hat{i}_\alpha$  and  $\hat{i}_\beta$  and BEMF $\alpha$  and BEMF $\beta$  deteriorate reaching peaks of  $\pm 4.8$  A and  $\pm 0.2$  V respectively. The transition dynamics takes 0.003s.

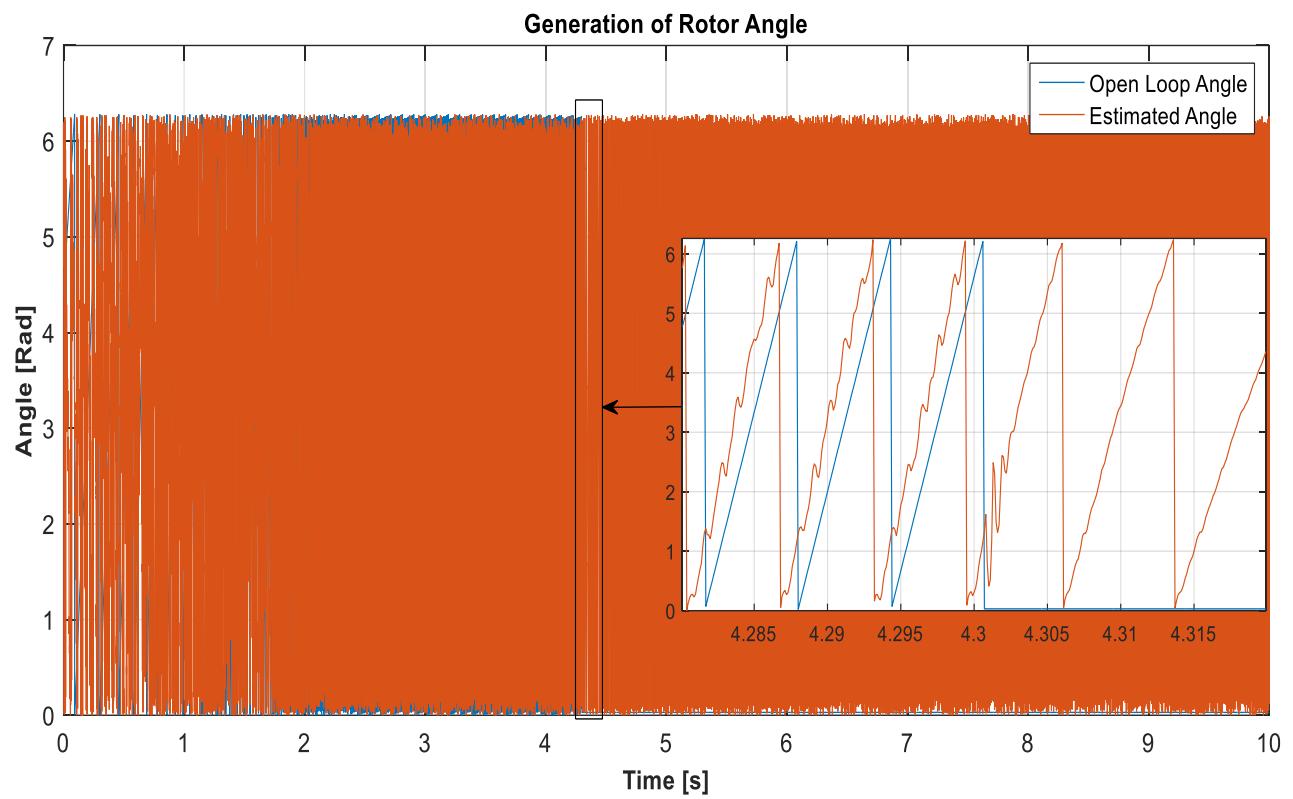


**Figure 6.14: Stationary reference frame current convergence during transition**



**Figure 6.15: Back-emf voltages during transition**

Figure 6.16 shows how the estimated rotor position angle is affected during transition. As it can be seen, during transition, the estimate of the rotor position angle is deteriorated since the back-emf produced deteriorated. The transition dynamics take 0.003s.



**Figure 6.16: Rotor position angle during transition**

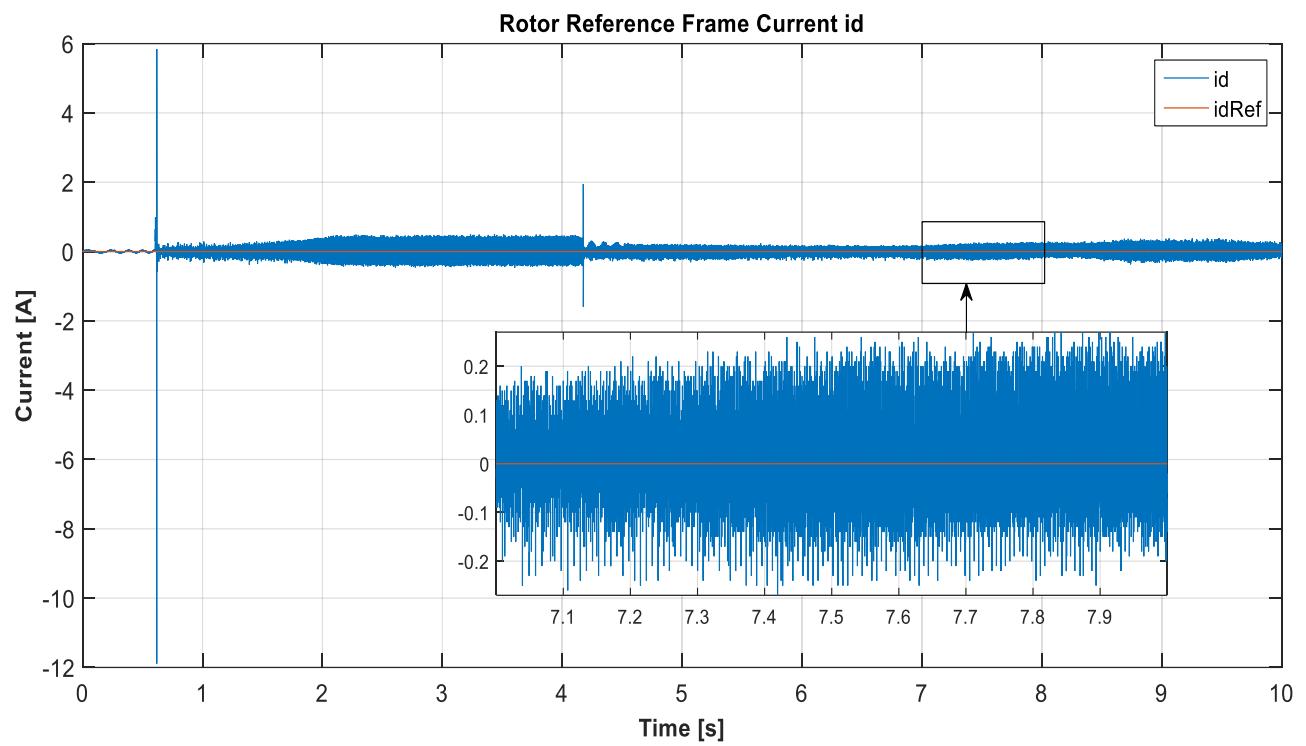
## 6.4 Experimental Results – After Transition

### 6.4.1 Introduction

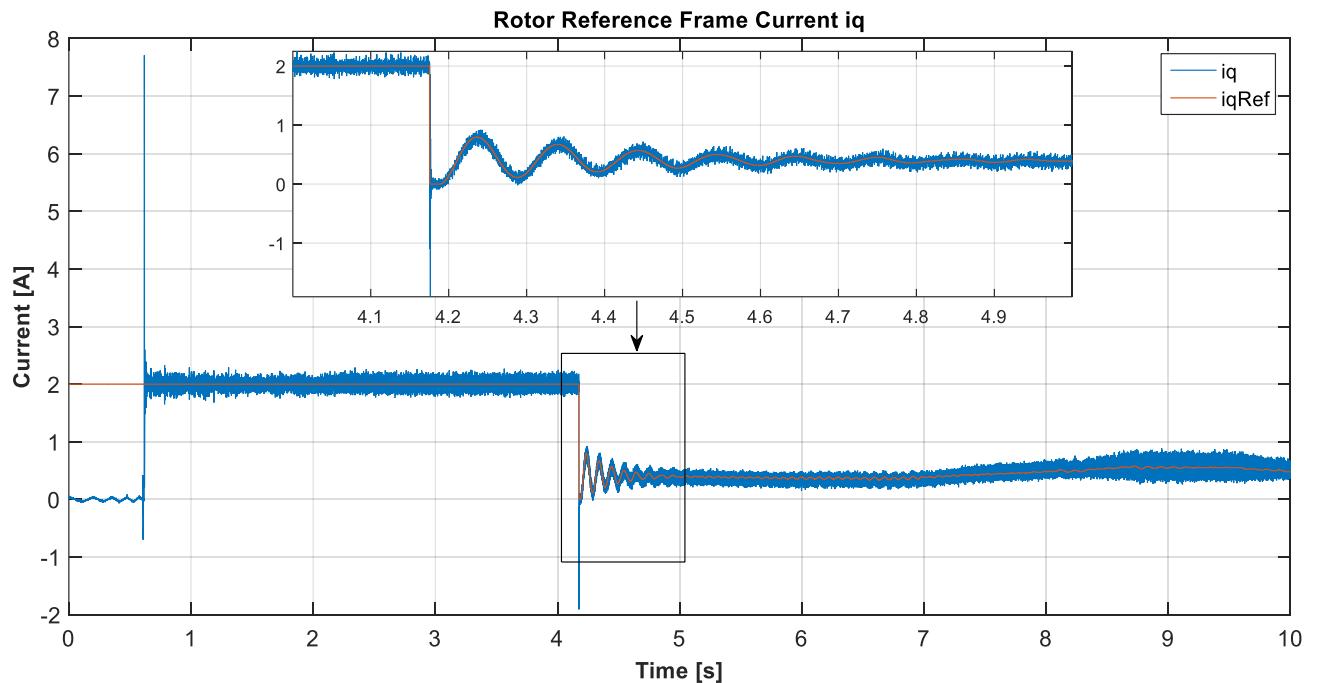
After transition, the sensorless closed loop system is tested for stability by varying the machine's mechanical speed, starting from 40 rad/s, and then gradually increasing it until a maximum of 1300 rad/s i.e. rated speed at a load torque of approximately 0.1437 Nm i.e. full load torque. Two sets of after transition results are presented. The first set shown in Figures 6.17 to 6.21 show the experimental results obtained for the synchronous reference frame currents  $i_d$  and  $i_q$ , rotor speed,  $i_\alpha$  and  $i_\beta$  convergence and rotor angle for a BLDC mechanical speed reference from 40 rad/s to 100 rad/s. The second set shown in Figures 6.22 to 6.26 show the experimental results obtained for the synchronous reference frame currents  $i_d$  and  $i_q$ , rotor speed,  $i_\alpha$  and  $i_\beta$  convergence and rotor angle respectively for the machines mechanical speed from 100 rad/s up to 250 rad/s.

#### 6.4.2 After Transition Results – Set 1

Figures 6.17 and 6.18 show currents  $i_d$  and  $i_q$  during sensorless operation for a speed range between 40 rad/s to 100 rad/s. In this speed range Current  $i_d$  can be seen to settle around the 0 A current reference with peaks reaching  $\pm 0.2$  A while current  $i_q$  can be seen settling around the 0.5 A current reference with peaks reaching  $\pm 0.2$  A.

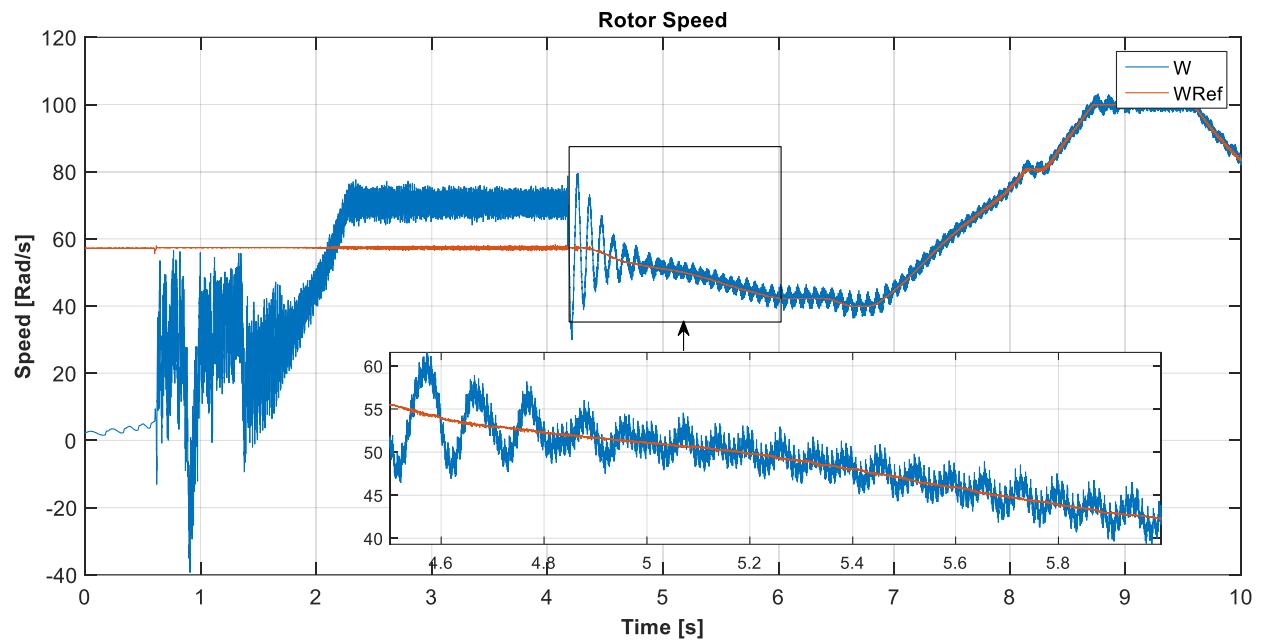


**Figure 6.17: Rotor reference frame current  $i_d$  after transition from 40 rad/s to 100 rad/s**

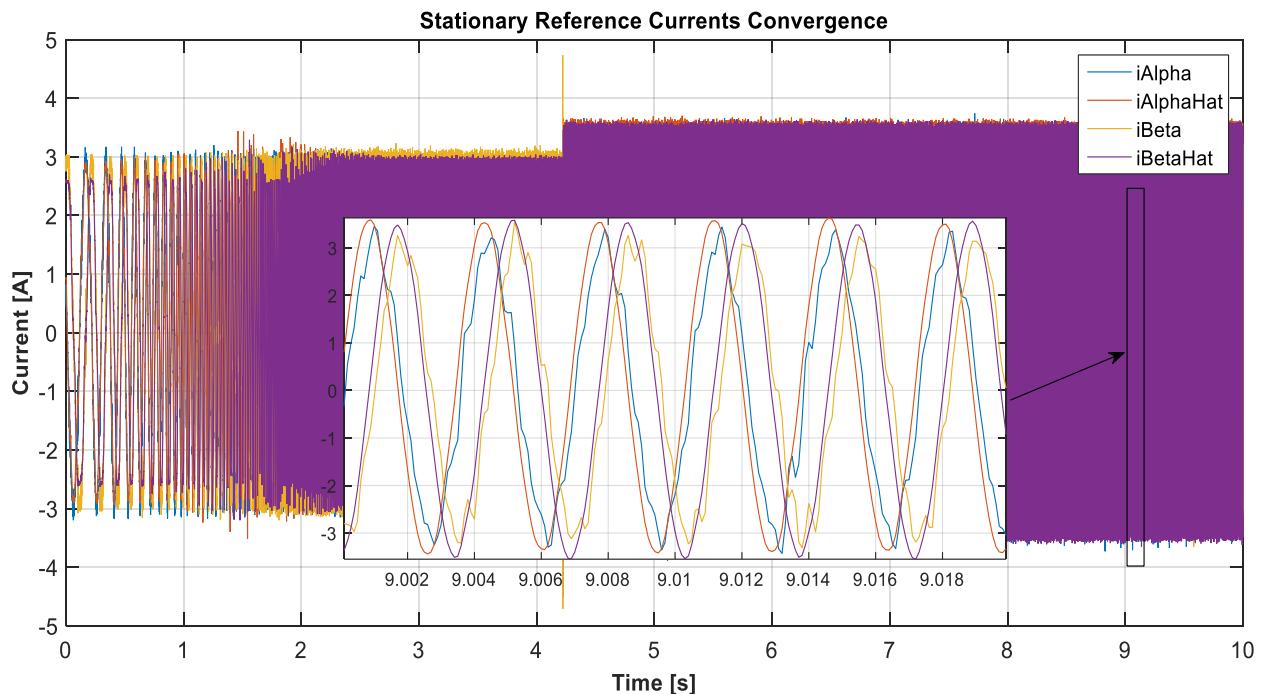


**Figure 6.18: Rotor reference frame current  $i_q$  after transition from 40 rad/s to 100 rad/s**

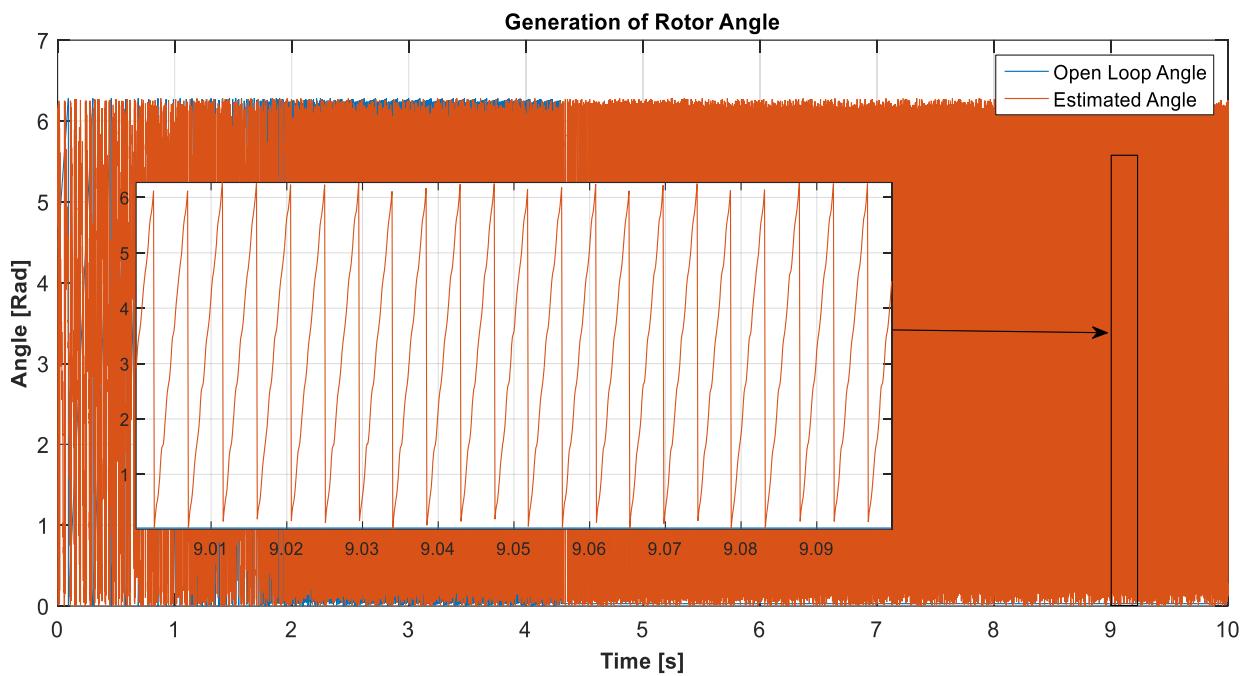
Figure 6.19 shows the rotor speed during sensorless operation for a speed range between 40 rad/s to 100 rad/s. In this speed range the machine's speed can be seen to track the reference while having maximum peak oscillations of  $\pm 3$  rad/s about the reference. Figure 6.20 shows how the estimates for  $\hat{i}_\alpha$  and  $\hat{i}_\beta$  continue to track  $i_\alpha$  and  $i_\beta$  for a maximum speed of 100 rad/s. For a speed range between 40 rad/s to 100 rad/s the back-emf observer can still generate a correct estimate of the rotor position angle as seen in Figure 6.21.



**Figure 6.19: Rotor speed after transition from 40 rad/s to 100 rad/s**



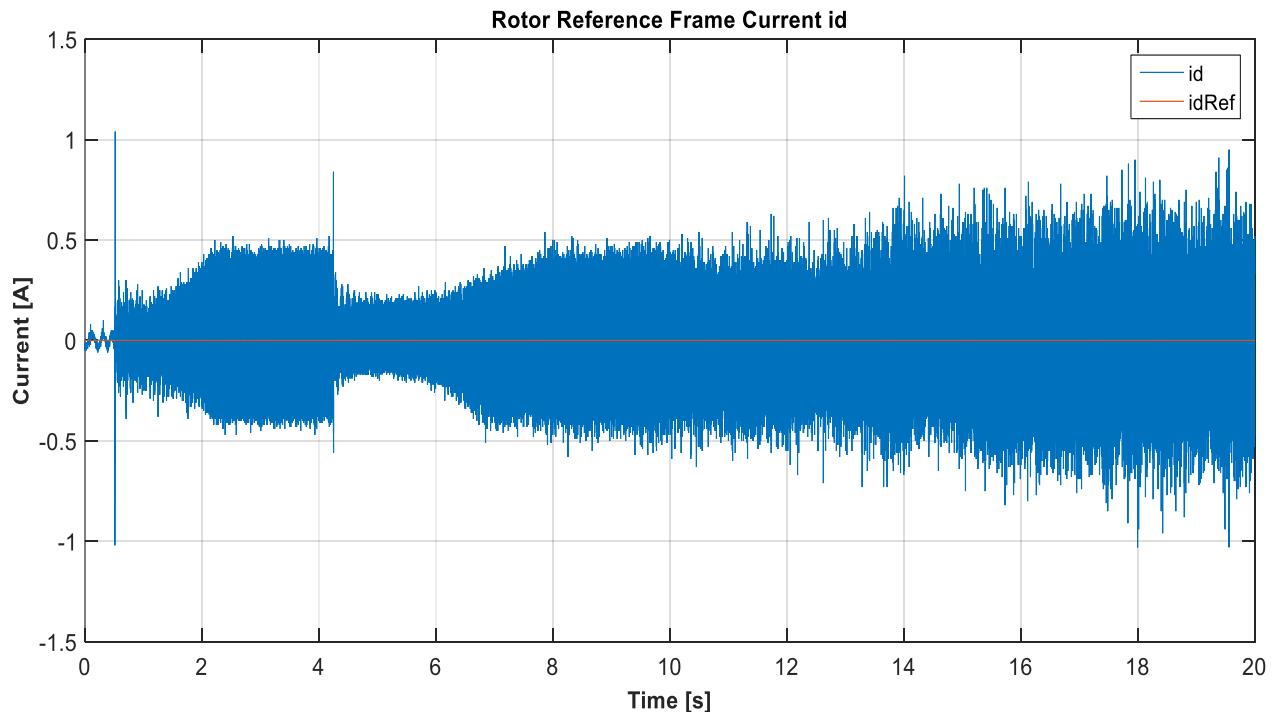
**Figure 6.20: Stationary reference frame current convergence after transition from 40 rad/s to 100 rad/s**



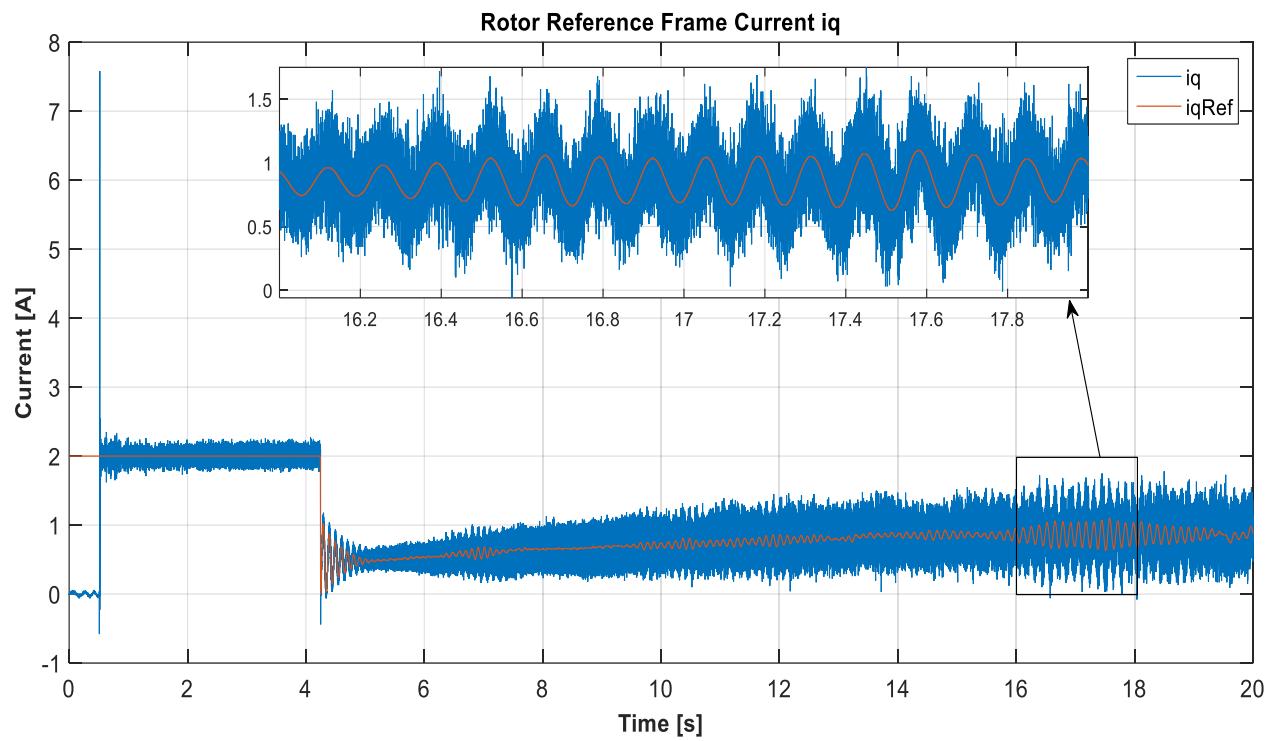
**Figure 6.21:** Rotor position angle after transition from 40 rad/s to 100 rad/s

#### 6.4.3 After Transition Results – Set 2

Figures 6.22 and 6.23 show currents  $i_d$  and  $i_q$  during sensorless operation for a speed range between 100 rad/s to 300 rad/s. In this speed range current  $i_d$  can be seen to go deteriorating reaching peaks of  $\pm 1$  A while current  $i_q$  can be seen oscillating around the reference while reaching peaks of  $\pm 1.5$  A.

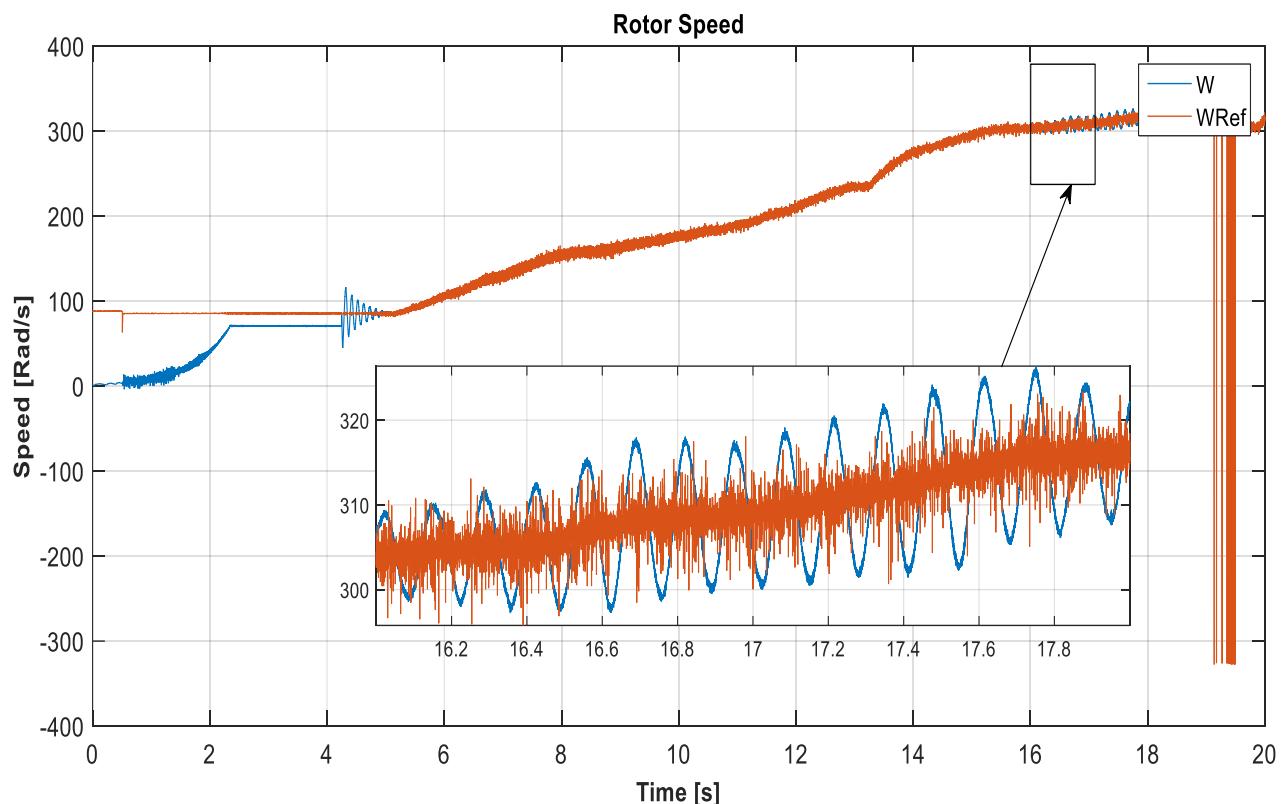


**Figure 6.22:** Rotor reference frame current  $i_d$  after transition from 100 rad/s to 300 rad/s



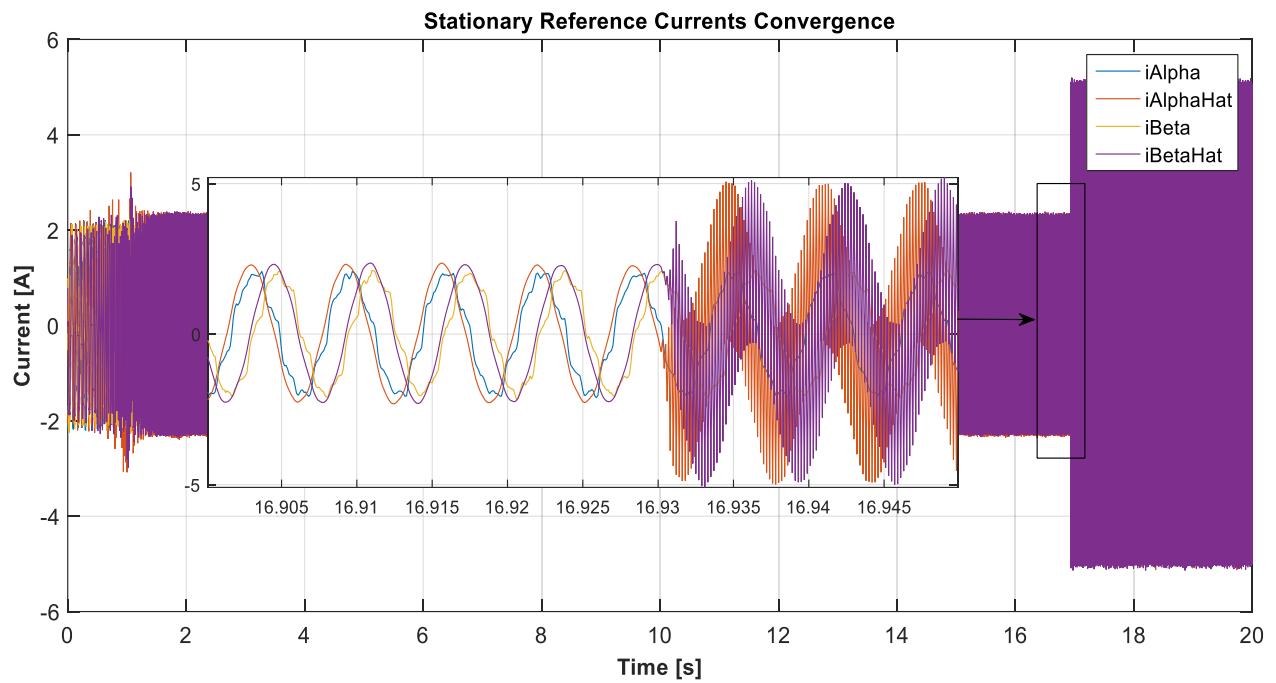
**Figure 6.23: Rotor reference frame current  $i_q$  after transition from 100 rad/s to 300 rad/s**

Figure 6.24 shows the rotor speed during sensorless operation for a speed range between 100 rad/s to 300 rad/s. At approximately 300 rad/s the machine speed can be seen sustaining low frequency oscillations that can overshoot to a maximum of  $\pm 20$  rad/s about the reference. At higher speeds the machine's rotor stops tracking the speed reference.

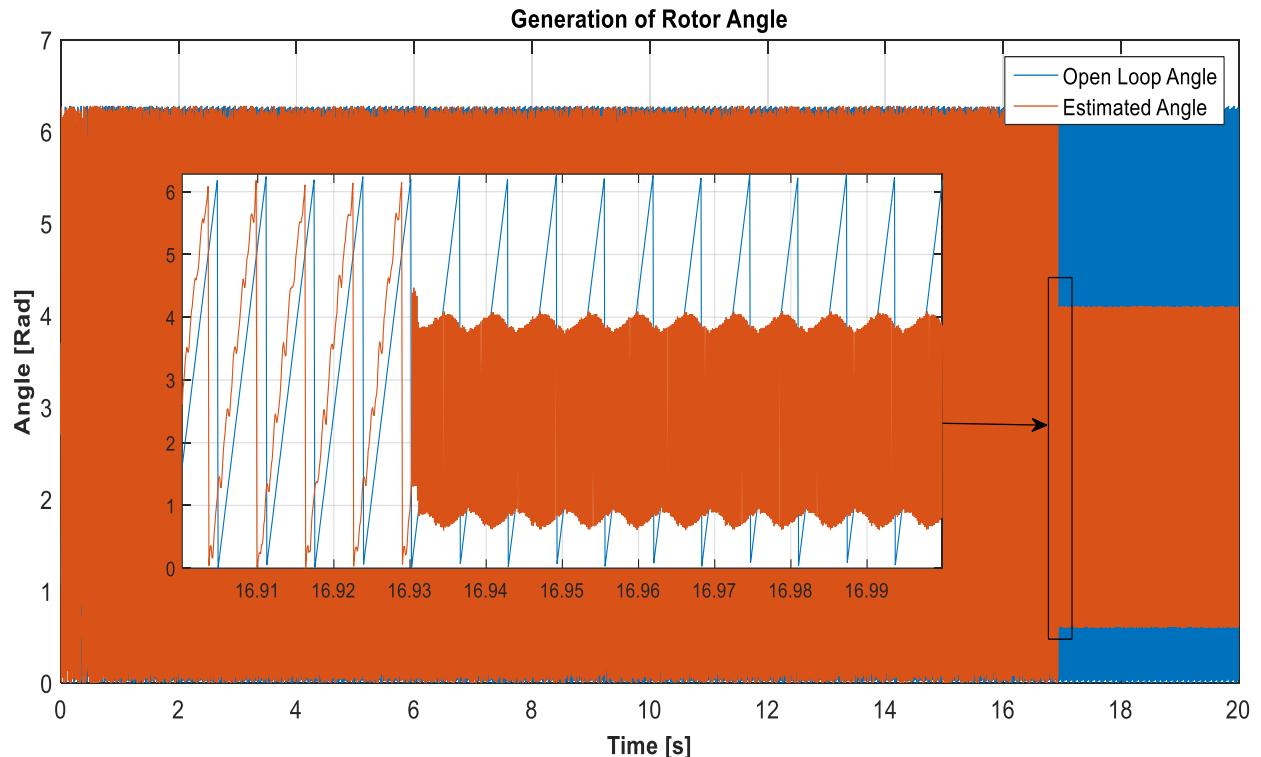


**Figure 6.24: Rotor speed after transition from 100 rad/s to 300 rad/s**

Figures 6.25 and 6.26 show how the back-emf observer generated currents  $i_\alpha$  and  $i_\beta$  and the rotor position angle generation during sensorless operation for a speed range between 100 rad/s and 300 rad/s. At time 16.93s it can be seen that the back-emf observer generated currents stop tracking  $i_\alpha$  and  $i_\beta$  and become unstable. This causes BEMF $\alpha$  and BEMF $\beta$  to distort thus creating an estimate for the rotor position angle that does not represent the actual rotor position angle.



**Figure 6.25: Stationary reference frame current convergence after transition from 100 rad/s to 300 rad/s**



**Figure 6.26: Rotor position angle after transition from 100 rad/s to 300 rad/s**

## 6.5 Conclusion

Throughout this chapter the experimental results for the BLDC machine currents, voltage, speed and rotor electrical angle were plotted in order to demonstrate the validity and functionality of a model based back-emf observer on a high-speed low power BLDC machine. From the tabulated experimental results it can be concluded that the model based back-emf observer works well in estimating the rotor electrical angle during the open loop operation. Furthermore, successful transition from open loop to sensorless operation was achieved with minimal deterioration to the estimates for both back-emf and rotor electrical angle. Lastly, speed control during sensorless operation was tested from 40 rad/s to the machine's rated speed i.e. 1300 rad/s. From experimental results speed control was only achieved from 40 rad/s to 250 rad/s, while at higher speeds the system become unstable.

## 7. Conclusion

### 7.1 Introduction

This chapter discusses the differences observed between the simulation (Chapter 4) and experimental results (Chapter 6). The performance of both closed loop sensorless speed control and the model based back-emf observer are analysed and discussed. The differences between simulation and experimental results are identified and justified. At the end of this chapter further improvements to system simulated/implemented are given.

### 7.2 Speed Control

When comparing the results obtained in simulation to the results obtained in practice, it can be noted that there are some differences in the sensorless speed control of the BLDC machine. Table 7.1 highlights the differences between the simulation results (Figures 4.25 – 4.27) and experimental results (Figures 6.17 – 6.19, 6.22 – 6.24), for the synchronous reference frame currents  $i_d$  and  $i_q$  and the BLDC machine speed with respect to the settling time, overshoot and steady state error.

**Table 7.1: Comparison between simulated and experimental results on speed control**

Simulation Results (Low Speed) – From 0 rad/s to 650 rad/s			
	Settling Time (s)	Overshoot (%)	Steady State Error (%)
<b>Current <math>i_d</math></b>	0.018	47	0
<b>Current <math>i_q</math></b>	0.012	8.56	0
<b>Machine's Speed <math>\omega</math></b>	0.022	24.15	0

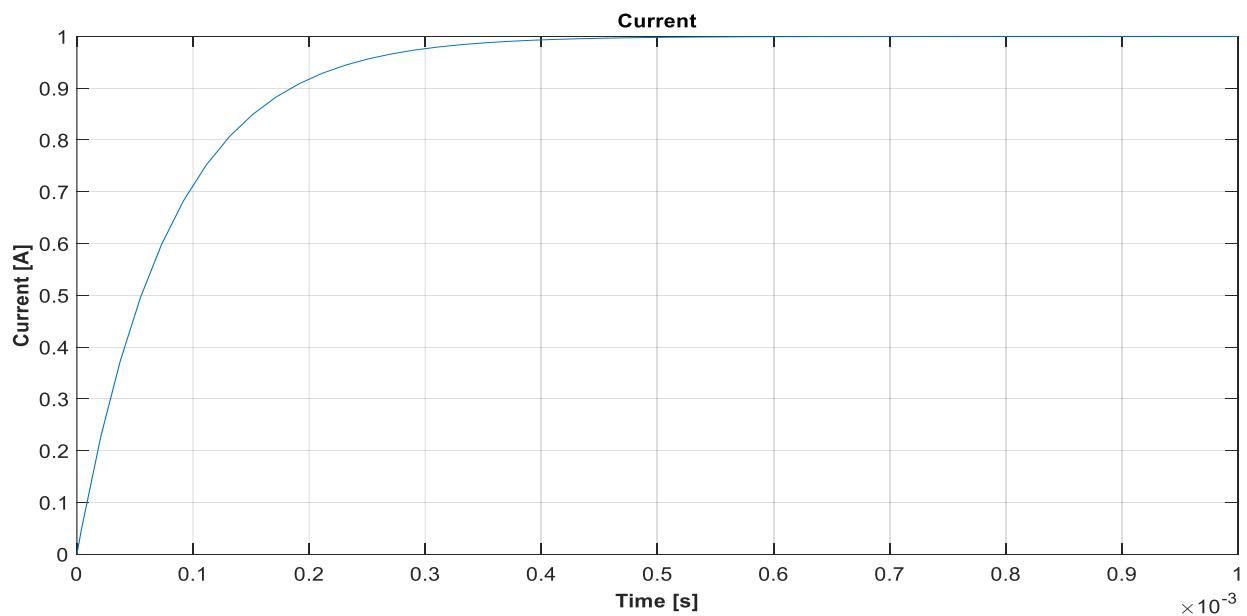
Simulation Results (High Speed) – From 650 rad/s to 1300 rad/s			
<b>Current <math>i_d</math></b>	0.019	147.2	0
<b>Current <math>i_q</math></b>	0.012	8.84	0
<b>Machine's Speed <math>\omega</math></b>	0.022	11.3	0
Experimental Results (Low Speed) – From 0 rad/s to 250 rad/s			
<b>Current <math>i_d</math></b>	0.2	26.72	0
<b>Current <math>i_q</math></b>	0.2	9.3	0
<b>Machine's Speed <math>\omega</math></b>	0.811	50	0
Experimental Results (High Speed) – From 250 rad/s to 1300 rad/s			
<b>Current <math>i_d</math></b>	Unstable	Unstable	Unstable
<b>Current <math>i_q</math></b>	Unstable	Unstable	Unstable
<b>Machine's Speed <math>\omega</math></b>	Unstable	Unstable	Unstable

The simulation results show zero steady state error, very low settling times and an accepted percentage overshoot for both synchronous reference frame currents and machine's speed from 0 rad/s to 1300 rad/s. While the experimental results show zero steady state error, considerably low settling times and an accepted percentage overshoot for both synchronous reference frame currents and machine's speed below 250 rad/s. From 250 rad/s to 1300 rad /s i.e. high-speed operation, both synchronous reference frame currents and machine's speed go unstable. Such differences could be the result of various factors some of which are discussed in the following sections.

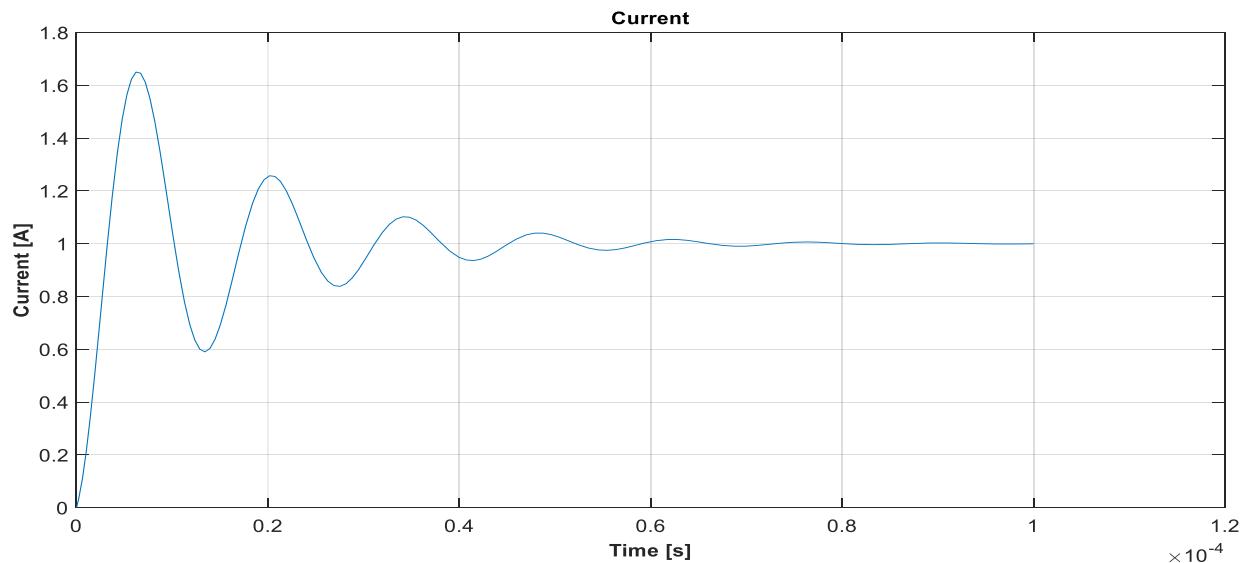
### 7.2.1 Current PI controller Design

As discussed in Section 3.4, RFOC controls stator synchronous currents  $i_d$  and  $i_q$  through current PI controllers and the speed  $\omega$  through a speed PI controller in cascaded form. Hence, the performance of the current PI controllers is critical to the overall system response. For simulation, it was assumed that the phase resistances and inductances are equal to each other and also equal to the two-axis dq frame resistance and inductance. The current PI controller was then designed such that

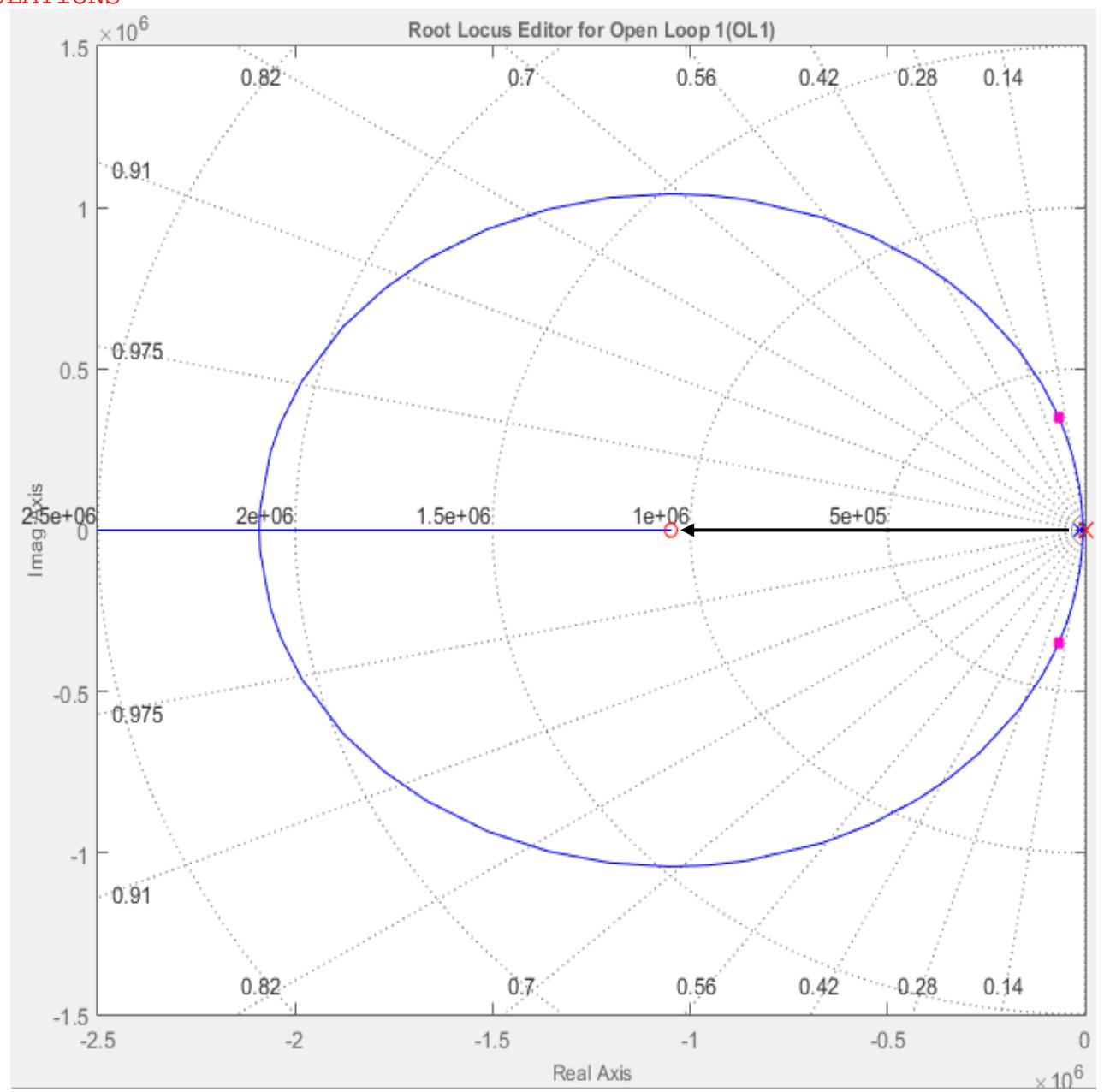
it's zero cancels out the machine's  $\frac{I_d(s)}{V_d(s)}$  pole. In practice, this can be approximated, however, pole-zero cancellation cannot be performed exactly due to parameter variations in the BLDC. Thus, the design of the current PI controller in practice might not have eliminated the machine's  $\frac{I_d(s)}{V_d(s)}$  pole causing further oscillations in practice than in simulation. Figure 7.1 shows the effect of having pole-zero cancellation when applying a step input to the simulated current transfer function while Figures 7.2 and 7.3 show the effect of losing pole-zero cancellation because of the misalignment between the machine's pole and the PI controller's zero respectively.



**Figure 7.1: Effect of having pole-zero cancellation when applying a step input**



**Figure 7.2: Effect of not having pole-zero cancellation when applying a step input**



**Figure 7.3: Misalignment between the machine's pole and the PI controller's zero**

This can be further observed by comparing the synchronous reference frame currents settling time and overshoot from Figure 4.26 and Figure 6.18. It can be seen that the settling time and overshoot for the simulated synchronous reference frame currents is much lower than that of the experimental counterpart. Moreover, heat generated inside the machine's coil and the frequency of operation affects the value of the two-axis dq frame resistance and inductance (Hu & Hameyer, 2015), (Kim, et al., 2016), (Wang, et al., 2016) causing the machine's  $\frac{I_d(s)}{V_d(s)}$  pole to move as these parameters change value.

## 7.2.2 Speed PI controller Design

The speed PI controller depends on the machine's moment of inertia J and the vicious coefficient of friction B, (Zhiligotov & Frolov, 2017), (Song, et al., 2017) both of which had to be estimated via calculations. In practice, such values might be very different thus effecting the overall speed PI controller design. This affect can be seen by comparing the machine's speed settling time from Figure 4.27 and Figure 6.19. It can be seen that in practice the speed dynamic response takes 0.789 s more time to settle than the simulation speed dynamic response.

## 7.3 Observer

Since the closed-loop response of the BLDC is dependent on the estimated position/speed of the sensorless observer, it is critical that the estimated parameters are sufficiently accurate within a wide range of speed. The most significant difference between the performance of the BLDC in the simulation environment and the experimental setup is that the sensorless estimates deteriorate at above a certain threshold in practice. In simulation, the speed range of the machine's rotor can vary from 0 rad/s to 1300 rad/s while in practice the speed range can only go to a maximum of 250 rad/s. The error in the estimated electrical angle on the experimental setup is significantly high such that the BLDC is stalled due to incorrect alignment to the synchronous frame of reference. The deterioration of the sensorless performance of the experimental BLDC is associated with variations between simulation/experimental environments which become significant at higher speeds. Table 7.2 compares the errors between the actual and estimate of the stationary reference frame currents and rotor electrical position between the simulated results (Figures 4.10 – 4.17) and experimental results (Figures 6.20, 6.21 and 6.25, 6.26).

**Table 7.2: Comparison between simulated and experimental results on observer performance**

Simulation Results (Full Speed Range) – From 0 rad/s to 1300 rad/s		
Error in $\hat{i}_\alpha$	Error in $\hat{i}_\beta$	Error in $\hat{\theta}$
$\pm 0.3 \text{ A}$	$\pm 0.3 \text{ A}$	$-0.36 \text{ rad}$

<b>Experimental Results (Low Speed) – From 0 rad/s to 250 rad/s</b>		
<b>Error in <math>\hat{i}_\alpha</math></b>	<b>Error in <math>\hat{i}_\beta</math></b>	<b>Error in <math>\hat{\theta}</math></b>
$\pm 0.35 \text{ A}$	$\pm 0.35 \text{ A}$	N. A.
<b>Experimental Results (High Speed) – From 250 rad/s to 1300 rad/s</b>		
<b>Error in <math>\hat{i}_\alpha</math></b>	<b>Error in <math>\hat{i}_\beta</math></b>	<b>Error in <math>\hat{\theta}</math></b>
Unstable	Unstable	Unstable

In simulation, the model designed to estimate currents  $\hat{i}_\alpha$  and  $\hat{i}_\beta$  has identical parameters to that BLDC machine model simulation as seen in the errors between the actual and estimate of the stationary reference frame currents and rotor electrical position (Figures 4.10 – 4.17) thus such an observer could theoretically work in all speed ranges. In practice at low speeds the errors between the actual and estimate of the stationary reference frame currents (Figure 6.20) is as low as the simulated results. However, as speed is increased above 250 rad/s the errors between the actual and estimate of the stationary reference frame currents (Figure 6.25) increase significantly indicating that the observer's PI controller does not track the actual values. The deterioration in the performance of the closed-loop PI trackers in the experimental setup is associated with:

1.  $L_d$  and  $L_q$  having different values in practice such that the gain/phase difference introduced in the closed-loop will be different to that designed/simulated while also tending to increase at higher frequency. In simulation the values of  $L_d$  and  $L_q$  used for tuning the observer were identical to those of the BLDC model.
2. Parameter variation because of heat generated and frequency of operation (Wang, et al., 2016) (Hu & Hameyer, 2015). In simulation parameter variation was not considered and modelled for.
3. The bandwidth of the PI controllers required in the observer loop to produce the estimated back-emf output is higher than the 1 kHz limit possible with a 10 kHz sampling frequency (bilinear transform requirements).

## 7.4 Further Improvements

Significant differences could be noted between the simulated and experimental results as mentioned in sections 7.2 and 7.3. Some of which could be reduced or eliminated in future work. In this section, further suggestions are proposed to improve the performance of the experimental setup.

Due to time constraints associated with the project, experimental tests to determine the real values of the synchronous reference frame resistances  $R_d$  and  $R_q$  and inductances  $L_d$  and  $L_q$  were not conducted. Differences between  $R_d$  and  $R_q$  and  $L_d$  and  $L_q$  would affect the bandwidth of the current controllers. An experimental test could have been done in order to estimate the values of the synchronous reference frame resistances and inductances as shown in (Ji & Noguchi, 2015), (Bobek, 2013) where they propose on-line identification of the q-axis inductance and resistance on the basis of the relationship between the mismatch of  $L_q$  and the d-axis feedback current. Also, the implementation of a SMC instead of the traditional PI controller inside the model based back-emf observer was not implemented. Because of the multiple input nature of the model based back-emf observer a non-linear controller such as the SMC is more effective than the PI controller (Jena, et al., 2015) as used in this dissertation. An SMC based controller could have possibly improved tracking of stationary reference frame currents  $i_\alpha$  and  $i_\beta$  by having near to zero overshoot and faster response both in steady and transient states. This type of controller could also remove any phase delays in the estimations of the observer as implemented in (Zheng & Li, 2016), (Mehta, et al., 2016), (Nigam, et al., 2016).

Due to financial constraints, high quality BLDC machines that provide all the necessary electrical and mechanical parameters could not be bought. Instead, a low cost BLDC machine was bought from an on-line hobbyist's store and electrical/mechanical parameters use for modelling/control where estimated. Better experimental results could have been obtained if a better microprocessor was used such as a Field Programmable Gate Array (FPGA) or a Digital Signal Processor (DSP) as suggested in (Darba, et al., 2013), (Reddy & Murali, 2016). Such controllers could provide higher processing speeds which could increase sampling frequency and therefore

observer bandwidth accordingly. Optimal signal conditioning circuits for improved current measurement could also be implemented. Since current measurement is the first step when implementing the RFOC scheme, improved signal conditioning circuits such as input offset compensation, amplification, filtering and buffering would have improved the overall quality of the sensorless BLDC drive. Better experimental results could also have been obtained if the estimated rotor position angle was compared with the actual BLDC machine's rotor position angle. This would have given better indication to how accurate the model based back-emf observer output is at different speed ranges.

## 7.5 Conclusion

This dissertation has reviewed the performance and validity of a model based back-emf observer on a low power, high-speed BLDC machine using the RFOC scheme with a PWM inverter through a simulated environment and an experimental setup. In Chapter 3 the BLDC machine mathematical model, the current and speed PI controllers and the model based back-emf observer mathematical model were designed, while in Chapter 5 an experimental setup consisting of a SSTP inverter hardware together with the RFOC scheme and SPWM software algorithms were designed. From the simulation results presented in Chapter 4 it was noted that the sensorless observer designed and simulated tracks currents in the  $\alpha\beta$ -frame from 0 rad/s to 1300 rad/s and produce an estimate of the rotor position angle that meets the actual rotor position angle with a maximum error of 0.1 rad. Furthermore, from simulation results, it was noted that accurate speed control during sensorless operation is also achieved having zero steady state error, less than 25% of overshoot and settling times of less than 0.025s. From the experimental results presented in Chapter 6 it was noted that the sensorless observer designed could only successfully track with a  $\pm 0.35$  A error the current transients in the  $\alpha\beta$ -frame from 0 rad/s to 250 rad/s while failing in tracking at higher speeds. Furthermore, from experimental results, it was noted that accurate speed control from 0 rad/s to 250 rad/s during sensorless operation is also achieved having zero steady state error, less than 25% of overshoot and settling times of less than 0.3s, while for speed higher than 250 rad/s speed control fails.

The differences between the simulated and experimental results were discussed in Chapter 7

which show that parameter variation, assumptions in the synchronous reference frame resistances and inductances and limited observer's PI controller bandwidth could have produced such discrepancies. Furthermore, improvements in the observer's controller, microprocessors, parameter variation and calculated estimates for the synchronous reference frame resistances and inductances were discussed. The sensorless observer presented in this dissertation has the disadvantage of having to estimate the back-emf quantities unlike the method in a six-step commutation that reads the floating phase (Pola & Vittal, 2006). However, since it can be used with a PWM inverter it has the advantage for improved efficiency in battery-operated applications (Lee, et al., 2003), (Krishnan & Arjun, 2014). Moreover, from results obtained such an experimental setup would be more suitable for BLDC machines having a lower Kv rating and a higher rotor inertia such that transients are physically limited and the observer's PI controller bandwidth limitation is not exceeded.

Please find attached in Appendix C page 124 a paper entitled "Modelling of a Sensorless Rotor Flux Oriented BLDC machine" submitted for peer review at the 19th International Conference on ELECTRICAL DRIVES and POWER ELECTRONICS October 04 - 06, 2017, Dubrovnik, CROATIA

## References

- Abdulmelik, M. B. & Hassan, T. K., 2016. Sensorless Vector Controlled Multilevel Inverter Fed BLDC Motor. *International Journal of Computer Applications*, 141(11), pp. 16 - 20.
- Ahmed, I. et al., 2016. Simplified Space Vector Modulation Techniques for Multilevel Inverters. *IEEE Transactions on Power Electronics*, 31(12), pp. 8483 - 8499.
- Benito, J. A., Glez-de-Rivera, G., Garrido, J. & Ponticelli, R., 2014. Design considerations of a small UAV platform carrying medium payloads. *Design of Circuits and Integrated Systems*, pp. 1-6.
- Blaschke, F., 1972. The Principle of Field Orientation Applied to The New Transvector Closed-Loop Control System for Rotating Field Machines. *Siemens-Rev*, 39(1), pp. 217-220.
- Bobek, V., 2013. *PMSM Electrical Parameters Measurement*, s.l.: NXP.
- Bostanci, E., Neuschl, Z. & Plikat, R., 2015. Influence of phase magnetic couplings on phase current characteristics of multiphase BLDC machines with overlapping phase windings. *IEEE Transactions on Magnetics*, 51(9), pp. 1-8.
- Chau, K. T., 2015. Permanent Magnet Brushless Motor Drives. In: K. T. Chau, ed. *Electric Vehicle Machines and Drives: Design, Analysis and Application*. 1st ed. s.l.:Wiley-IEEE Press, pp. 69-106.
- Chun, T. W., Tran, Q. V., Lee, H. H. & Kim, H. G., 2014. Sensorless Control of BLDC Motor Drive for an Automotive Fuel Pump Using Hysteresis Comparator. *IEEE Transactions on Power Electronics*, Volume 29, pp. 1382-1391.
- Darba, A., Belie, F. D., Salem, A. & Melkebeek, J., 2013. FPGA-based implementation of the back-EMF symmetric-threshold-tracking sensorless commutation method for Brushless DC-machines. *2013 IEEE International Symposium on Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE)*, pp. 1 - 6.
- Deng, Y., Wang, Y., Teo, K. H. & Harley, R. G., 2016. A Simplified Space Vector Modulation Scheme for Multilevel Converters. *IEEE Transactions on Power Electronics*, 31(3), pp. 1873 - 1886.
- Derammelaere, S. et al., 2016. A quantitative comparison between BLDC, PMSM, brushed DC and stepping motor technologies. *2016 19th International Conference on Electrical Machines and Systems (ICEMS)*, pp. 1-5.
- Ejlali, A. & Soleimani, J., 2012. Sensorless vector control of 3-phase BLDC motor using a novel Extended Kalman. *2012 International Conference on Advances in Power Conversion and Energy Technologies (APCET)*, pp. 1 - 6.

Firmansyah, E., Wijaya, F. D., Aditya, W. P. R. & Wicaksono, R., 2014. Six-steo commutation with round robin state machine to alleviate error in hall-effect-sensor reading for BLDC motor control. *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*, pp. 251-253.

Girija, P. K. & Prince, A., 2014. Robustness evaluation of SMO in sensorless control of BLDC motor under DTC scheme. *2014 International Conference on Power Signals Control and Computations (EPSCICON)*, pp. 1-6.

Gonzalez, I. et al., 2011. Attitude control of a quad-rotor using speed sensing in brushless DC motors. *Electrical Engineering Computing Science and Automatic Control*, pp. 1-6.

Harrison, D. B. & Rye, N. Y., 1955. *Commutatorless direct current motor*. United States, Patent No. 2,719,944.

Horvat, R., Jezernik, K. & Cukovic, M., 2014. An Event-Driven Approach to the Current Control of a BLDC Motor Using FPGA. *IEEE Transactions on Industrial Electronics*, 61(7), pp. 3719 - 3726.

Huang, X. et al., 2006. Fault-Tolerant Brushless DC Motor Drive For Electro-Hydrostatic Actuation System In Aerospace Application. *Conference Records of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting*, pp. 473-480.

Huang, X. et al., 2012. A Single Sided Matrix Converter Drive for a Brushless DC Motor in Aerospace Applications. *IEEE Transactions on Industrial Electronics*, Volume 59, pp. 3542-3552.

Huang, X. et al., 2012. Design of a Five-Phase Brushless DC Motor for a Safety Critical Aerospace Applications. *IEEE Transactions on Industrial Electronics*, Volume 59, pp. 3532-3541.

Hu, Z. & Hameyer, K., 2015. Robust predictive current control for performance improvement of induction motors with parameter variation. *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 000451 - 000456.

Imoru, O. & Tsado, J., 2012. Modelling of an electronically commutated (Brushless DC) motor drives with back-emf sensing. *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*, pp. 828-831.

Itoh, J.-i., Nomura, N. & Ohsawa, H., 2002. *A Comparison between V/f Control and Position-Sensorless Vector Control for the Permanent Magnet Synchronous Motor*. Tokyo, Power Conversion Conference, 2002. PCC-Osaka 2002..

Jahns, T. M. & Soong, W. L., 1996. Pulsating torque minimization techniques for permanent magnet AC motor drives-a review. *IEEE Trans. on Industrial Electronics*, Volume 43, pp. 321-330.

Jena, N. K., Mohanty, K. B., Pradhan, H. & Sanyal, S. K., 2015. A comparison between PI & SMC used for decoupled control of PMSG in a variable speed wind energy system. *2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, pp. 1 - 6.

Jianwen, S., Nolan, D., Teissier, M. & Swanson, a. D., 2003. A novel microcontroller-based sensorless brushless DC (BLDC) motor drive for automotive fuel pumps. *IEEE Transactions on Industry Applications*, Volume 39, pp. 1734-1740.

Ji, X. & Noguchi, T., 2015. Online q-axis inductance and resistance identification of IPM synchronous motor based on relationship between its parameter mismatch and current. *2015 18th International Conference on Electrical Machines and Systems (ICEMS)*, pp. 648 - 653.

Joice, C. S., Paranjothi, S. R. & Kumar, V. J. S., 2013. Digital Control Strategy for Four Quadrant Operation of Three Phase BLDC Motor With Load Variation. *IEEE Transactions on Industrial Informatics*, 9(2), pp. 974-982.

Joseph, T. E. & Sreethumol, M. V., 2015. Sensorless control of BLDC motor drive under DTC scheme with modified integrator. *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, pp. 1-7.

Kim, D.-Y., Moon, J.-J., Kim, J.-M. & Ji, J.-S., 2016. Sensorless control method of 3-Phase BLDC motor through the real time compensation of back-EMF constant. *Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), 2016 IEEE 8th International*, pp. 1-7.

Kim, D.-Y., Moon, J.-J., Kim, J.-M. & Ji, J.-S., 2016. Sensorless control method of 3-Phase BLDC motor through the real time compensation of back-EMF constant. *2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*, pp. 3361 - 3367.

Kim, H., Son, J. & Lee, J., 2011. A High-Speed Sliding-Mode Observer for the sensorless Speed Control of a PMSM. *IEEE Transactions on Industrial Electronics*, Volume 58, pp. 4069-4077.

Korkmaz, F., Topaloglu, I., Cakir, M. F. & Gurbuz, R., 2013. Comparative performance evaluation of FOC and DTC controlled PMSM drives. *4th International Conference on Power Engineering, Energy and Electrical Drives*, pp. 705-708.

Krishnan, P. H. & Arjun, M., 2014. *Control of BLDC motor based on adaptive fuzzy logic PID controller*. s.l.:IEEE.

Kumar, R. & Singh, B., 2016. BLDC Motor-Driven Solar PV Array-Fed Water Pumping System Employing Zeta Converter. *IEEE Transactions on Industry Applications*, 52(3), pp. 2315 - 2322.

Kye-Lyong, K., Jang-Mok, K., Keun-Bae, H. & Kyung-Hoon, K., 2004. Sensorless control of PMSM in high speed range with iterative sliding mode observer. *Applied Power Electronics Conference and Exposition, 2004. APEC '04. Nineteenth Annual IEEE*, Volume 2, pp. 1111-1116.

Lazor, M. & Stulrajter, M., 2014. *Modified Field Oriented Control for Smooth Torque Operation of a BLDC Motor*. Roznov pod Radhostem, ELEKTRO, 2014.

Lee, B.-K., Kim, T.-H. & Ehsani, M., 2003. On the Feasibility of Four-Switch Three-Phase BLDC Motor Drives for Low Cost Commercial Applications: Topology and Control. *IEEE Transactions on Power Electronics*, 18(1), pp. 164 - 172.

Lin, C., hung, C. & Liu, C., 2006. Sensorless Control for Four-Switch Three-Phase Brushless DC Motor Drives. *Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE*, pp. 2048-2053.

Liu, X. et al., 2016. Analytical solution of the magnetic field and EMF calculation in ironless BLDC motor. *IEEE Transactions on Magnetics*, 52(2), pp. 1-8.

Lizuka, K. et al., 1985. Microcomputer Control for Sensorless Brushless Motor. *IEEE Transactions on Industry Applications*, IA-21(3), pp. 595 - 601.

LoVetri, J. & Makaran, J. E., 2001. Prediction of conducted RFI emissions in BLDC motors for automotive applications. *IEEE EMC International Symposium. Symposium Record. International Symposium on Electromagnetic Compatibility (Cat. No.01CH37161)*, Volume 1, pp. 21-25.

Mashhadany, Y. I. A., 2015. High-Performance Multilevel Inverter Drive of Brushless DC Motor.

*International Journal of Sustainable and Green Energy, Special Issue*, 4(3 - 1), pp. 1 - 7.

Mehta, H., Joshi, V. & Kurulkar, P., 2016. Implementation issues of sliding mode observer for sensorless field oriented control of PMSM using TMS320F2812. *2016 IEEE Symposium on Sensorless Control for Electrical Drives (SLED)*, pp. 1 - 6.

Murai, Y. et al., 1989. Torque ripple improvement for brushless DC miniature motors. *IEEE Transactions on Industry Applications*, 25(3), pp. 441-450.

Nair, D. S., Jagadanand, G. & George, S., 2016. Direct Torque Control with duty cycle control strategy for permanent magnet motors with non-sinusoidal EMF. *2016 IEEE International Conference on Industrial Technology (ICIT)*, pp. 141-146.

Nigam, V., Hussain, S. & Agarwal, S. N., 2016. A hybrid fuzzy sliding mode controller for a BLDC motor drive. *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1 - 4.

Ozturk, S. B., Alexander, W. C. & Toliat, H. A., 2009. Direct Torque Control of Four-Switch Brushless DC Motor With Non-Sinusoidal Back EMF. *IEEE Transactions on Power Electronics*, 25(2), pp. 263 - 271.

Parasiliti, F., Petrella, R. & Tursini, M., 2001. Sensorless speed control of a PM synchronous motor based on sliding mode observer and Extended Kalman filter. *Conference Record of the 2001 IEEE Industry Applications Conference 36th IAS Annual Meeting (Cat. No.01CH37248)*, Volume 1, pp. 533-540.

Pola, U. V. S. & Vittal, K. P., 2006. Recent Developments in Control Schemes of BLDC Motors. *IEEE International Conference on Industrial Technology*, pp. 477-482.

Rao, A. C., Y.P.Obulesh & Babu, C. S., 2013. A Five Level Cascaded Multilevel Inverter FED Brushless DC Motor With Phase Shifted Carrier PWM Techniques. *International Journal of Electrical and Electronics Engineering Research*, 3(1), pp. 231 - 240.

Rao, A. P. C., Obulesh2, Y. & Babu3, C. S., 2013. High Performance Cascaded Multilevel Inverter Fed Brushless DC Motor Drive. *International Journal of Engineering Sciences & Emerging Technologies*, 5(2), pp. 88 - 96.

Rau, D., Rodina, J., Palkovic, L. & Hubinsky, P., 2015. Sensorless Field Oriented Control of BLDC Motors for MAVs. *Transactions on Electrical Engineering*, 4(4), pp. 91-96.

Reddy, B. P. & Murali, A., 2016. SoC FPGA-based field oriented control of BLDC motor using low resolution Hall sensor. *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 2941 - 2945.

Saleh, K. I., Badr, M. A. & Wahsh, S. A., 2001. Perfect field oriented brushless DC motor. *2001 Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.01CH37262)*, Volume 3, pp. 1433 - 1438.

Sanchez, A. et al., 2011. Hovering flight improvement of a quad-rotor mini UAV using brushless DC motors. *Journal of Intelligent & Robotic Systems*, Volume 61, pp. 85-101.

Scicluna, K., Staines, C. S. & Raute, R., 2016. Sensorless position control of a PMSM for steer-by-wire applications. *International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 046-051.

Shanmugapriya, M. & Michael, P. A., 2012. Sensorless control of an four switch three phase inverter using FPGA. *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference*, pp. 1-7.

Singh, B., Bist, V., Chandra, A. & Al-Haddad, K., 2015. Power Factor Correction in Bridgeless-Luo Converter-Fed BLDC Motor Drive. *IEEE Transactions on Industry Applications*, 51(2), pp. 1179 - 1188.

Song, X., Han, B., Zheng, S. & Fang, J., 2017. High-precision Sensorless Drive for High-speed BLDC Motors based on the Virtual 3rd harmonic back-EMF. *IEEE Transactions on Power Electronics*, PP(99), pp. 1 - 1.

Staines, C. S., Caruana, C. & Raute, R., 2014. A Review of Saliency-based Sensorless Control Methods for Alternating Current Machines. *IEEE Journal of Industry Applications*, Volume 3, pp. 86-94.

Takahashi, I. N. T., 1986. A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor. *IEEE Transactions on Industry Application*, 5(1), pp. 820-827.

Tsai, M.-F., Quy, T. P., Wu, B.-F. & Tseng, C.-S., 2011. Model construction and verification of a BLDC motor using MATLAB/SIMULINK and FPGA control. *2011 6th IEEE Conference on Industrial Electronics and Applications*, pp. 1797 - 1802.

Waide, P. & Brunner, C. U., 2011. *Energy-Efficiency Policy Opportunities for Electric Motor-Driven Systems*. Paris: International Energy Agency.

Wang, Y., Li, H. & Ren, D., 2016. Characteristics of BLDC motor drive system at high temperature. *2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia)*, pp. 856 - 862.

Wang, Y. et al., 2017. Deadbeat Model-Predictive Torque Control With Discrete Space-Vector Modulation for PMSM Drives. *IEEE Transactions on Industrial Electronics*, 64(5), pp. 3537 - 3547.

Xia, C.-I., 2012. *Permanent Magnet Brushless DC Motor Drives and Controls*. Singapore: John Wiley & Sons Singapore Pte. Ltd..

Xia, C., Xiao, Y. & Shi, T., 2014. Boost Three-Effective-Vector Current Control Scheme for a Brushless DC Motor With Novel Five-Switch Three-Phase Topology. *IEEE Transactions on Power Electronics*, 29(12), pp. 6581 - 6592.

Xia, K. et al., 2015. Research of four-switch three-phase BLDC motor control scheme based on quasi Z-source converter. *Electrical Machines and Systems (ICEMS), 2015 18th International Conference*, pp. 1-6.

Zambada, J. & Deb, D., 2010. *Sensorless Field Oriented Control of a PMSM*, s.l.: Microchip.

Zeng, Z. et al., 2017. Hybrid Space Vector Modulation Strategy for Torque Ripple Minimization in Three-Phase Four-Switch Inverter-Fed PMSM Drives. *IEEE Transactions on Industrial Electronics*, 64(3), pp. 2122 - 2134.

Zheng, C. & Li, Y., 2016. Sensorless Speed Control for Brushless DC Motors System Using Sliding-Mode Controller and Observers. *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Volume 1, pp. 216 - 220.

Zhiligotov, R. I. & Frolov, V. Y., 2017. Development of the sensorless control system BLDC motor. *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1109 - 1111.

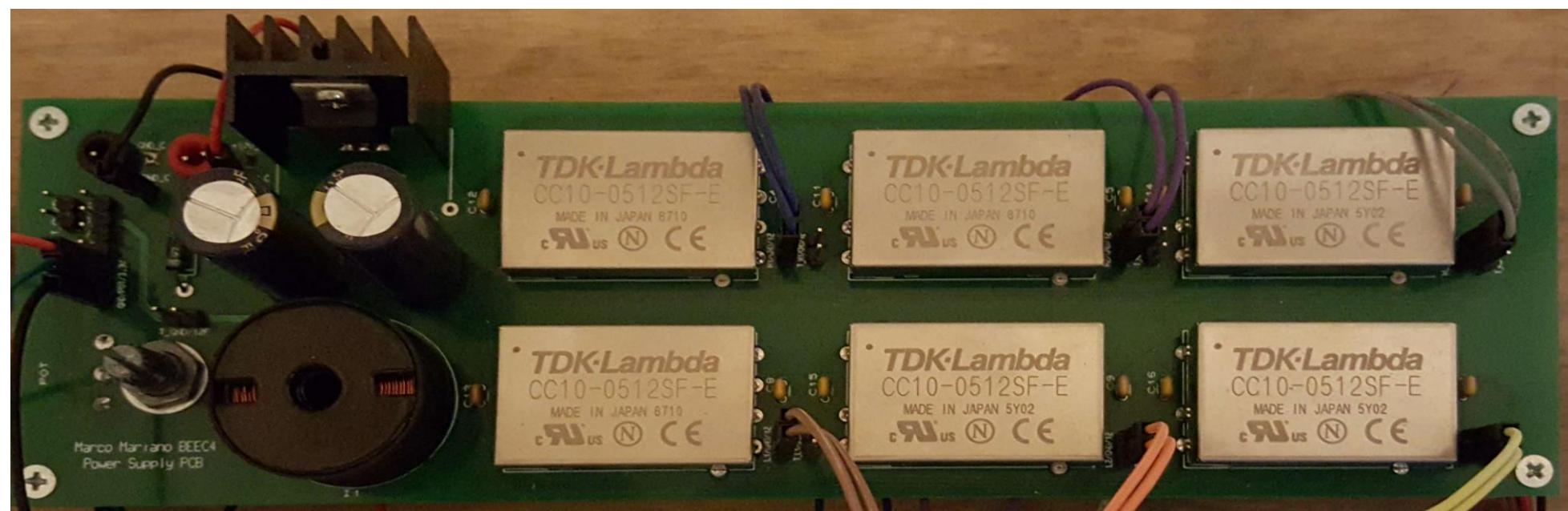
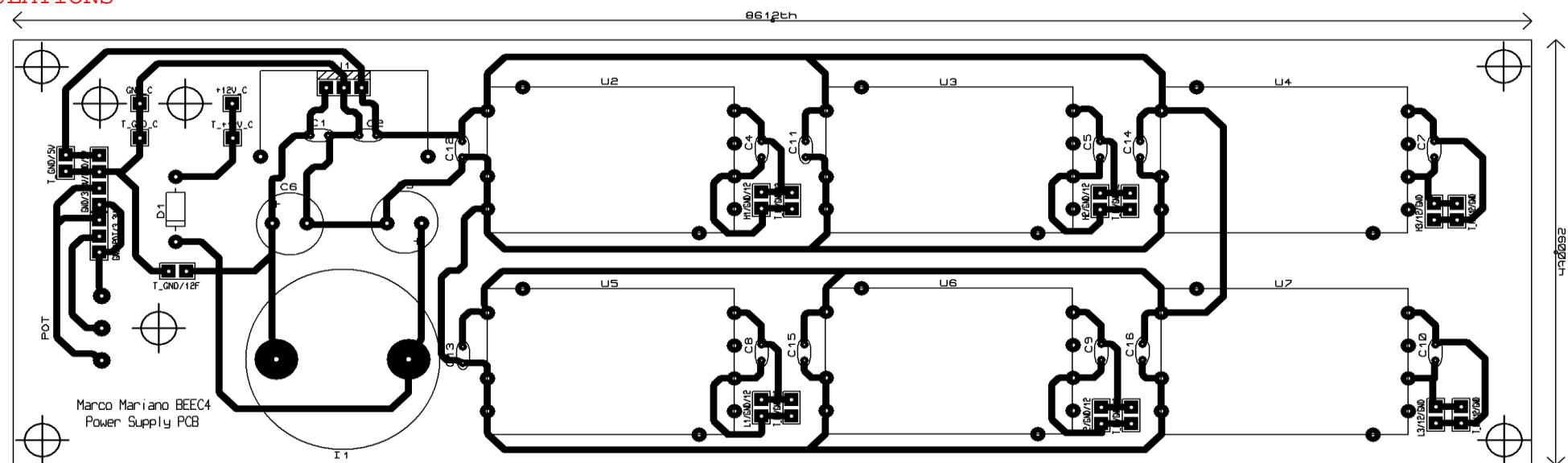
# Appendix A

Power Supply PCB – Appendix A.1

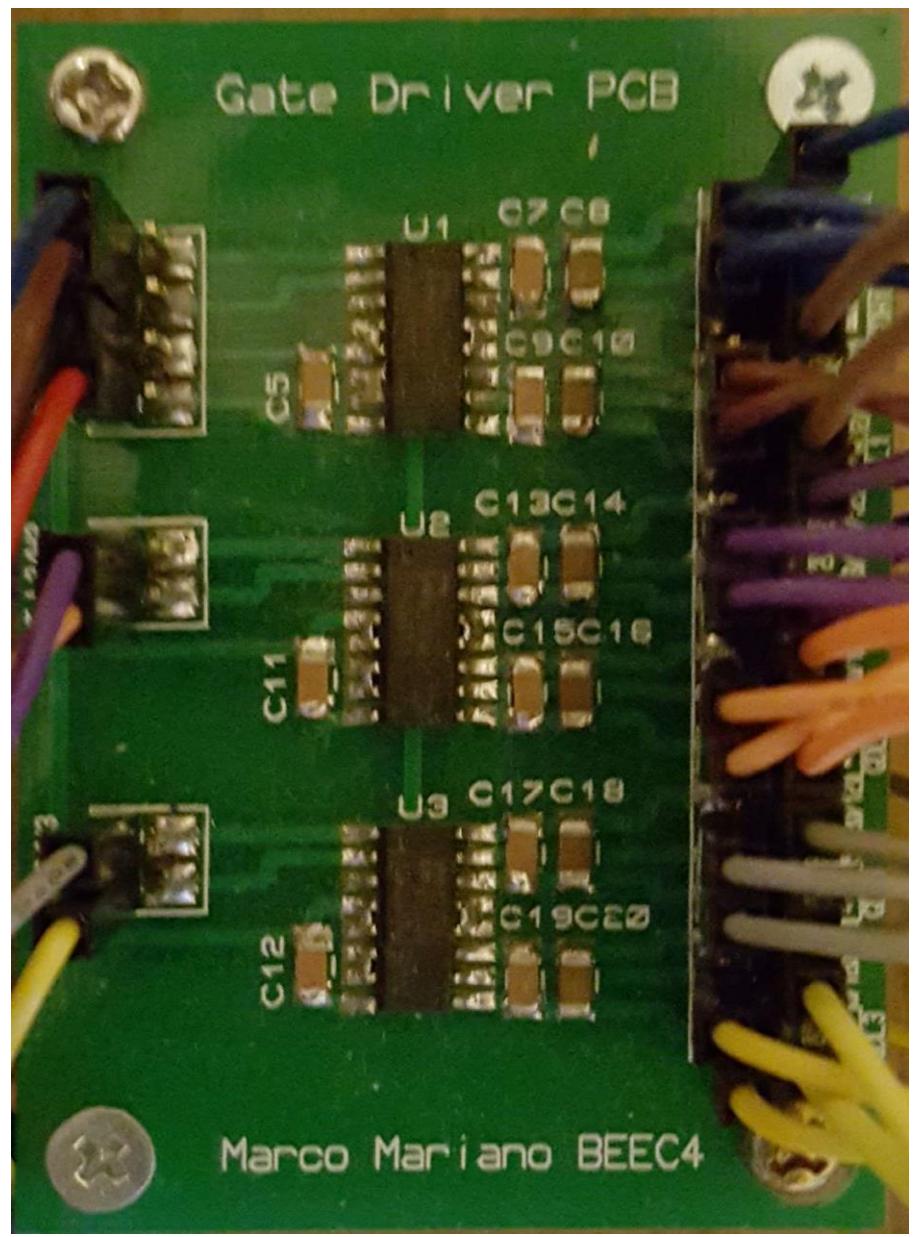
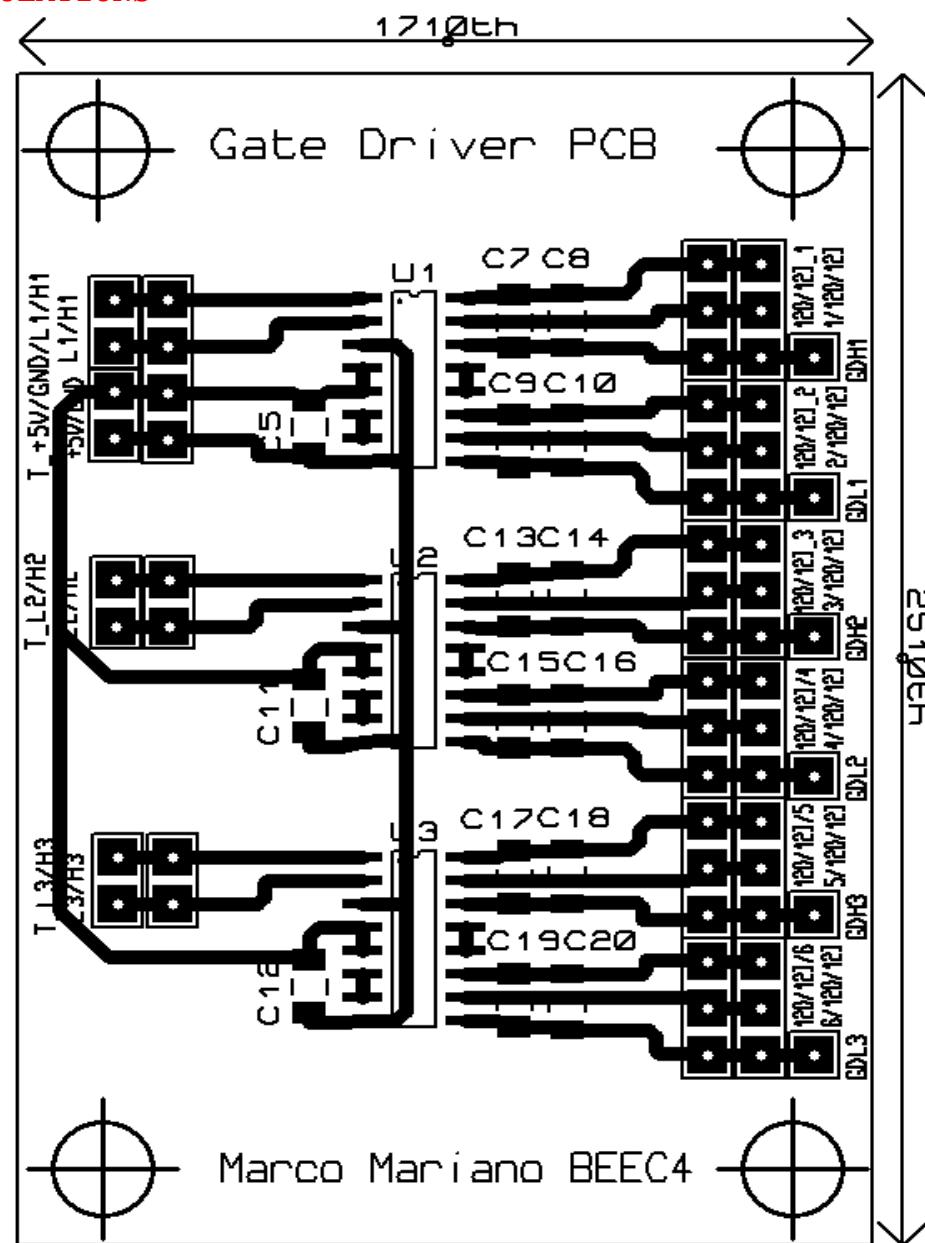
Gate Driver PCB – Appendix A.2

Three-phase Inverter PCB – Appendix A.3

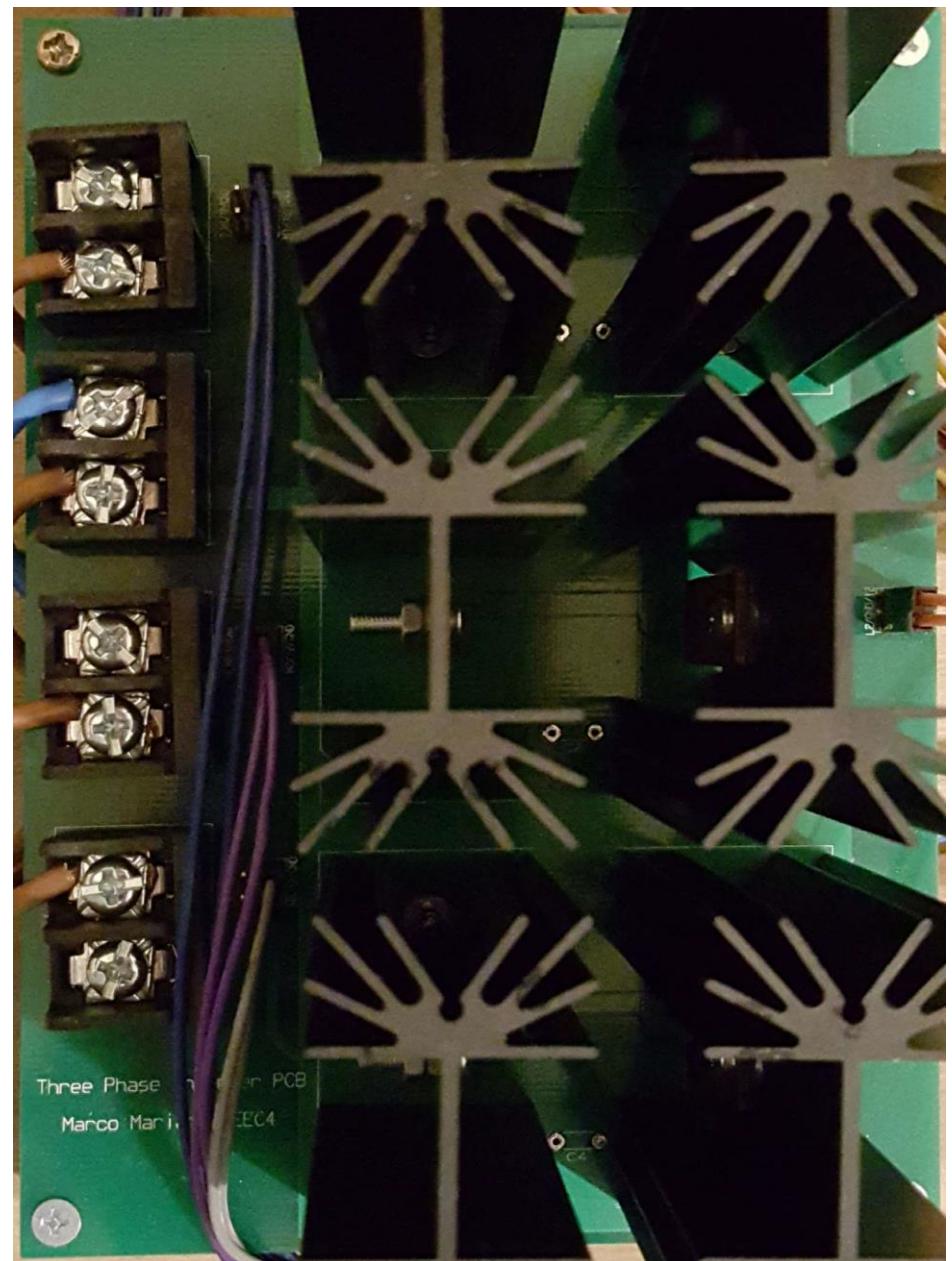
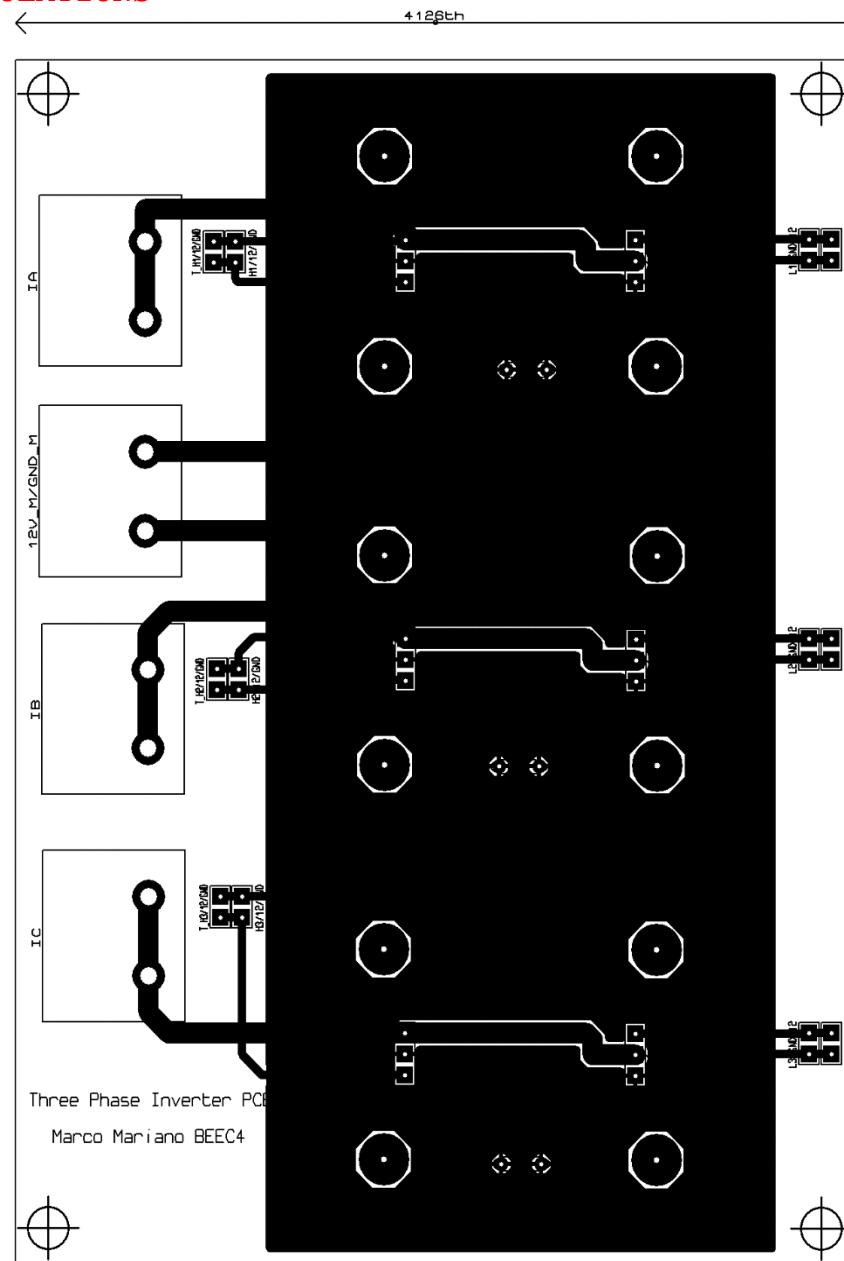
Current Sensing PCB – Appendix A.4



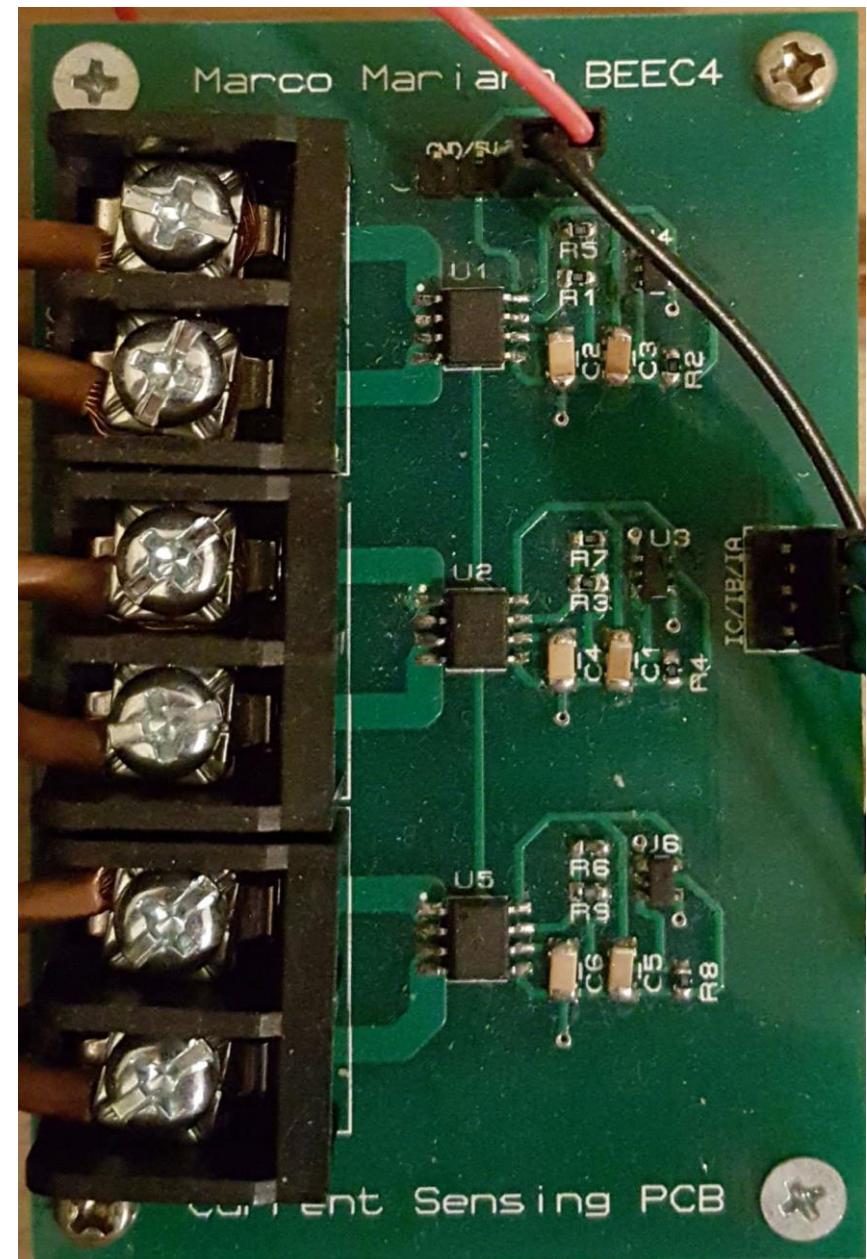
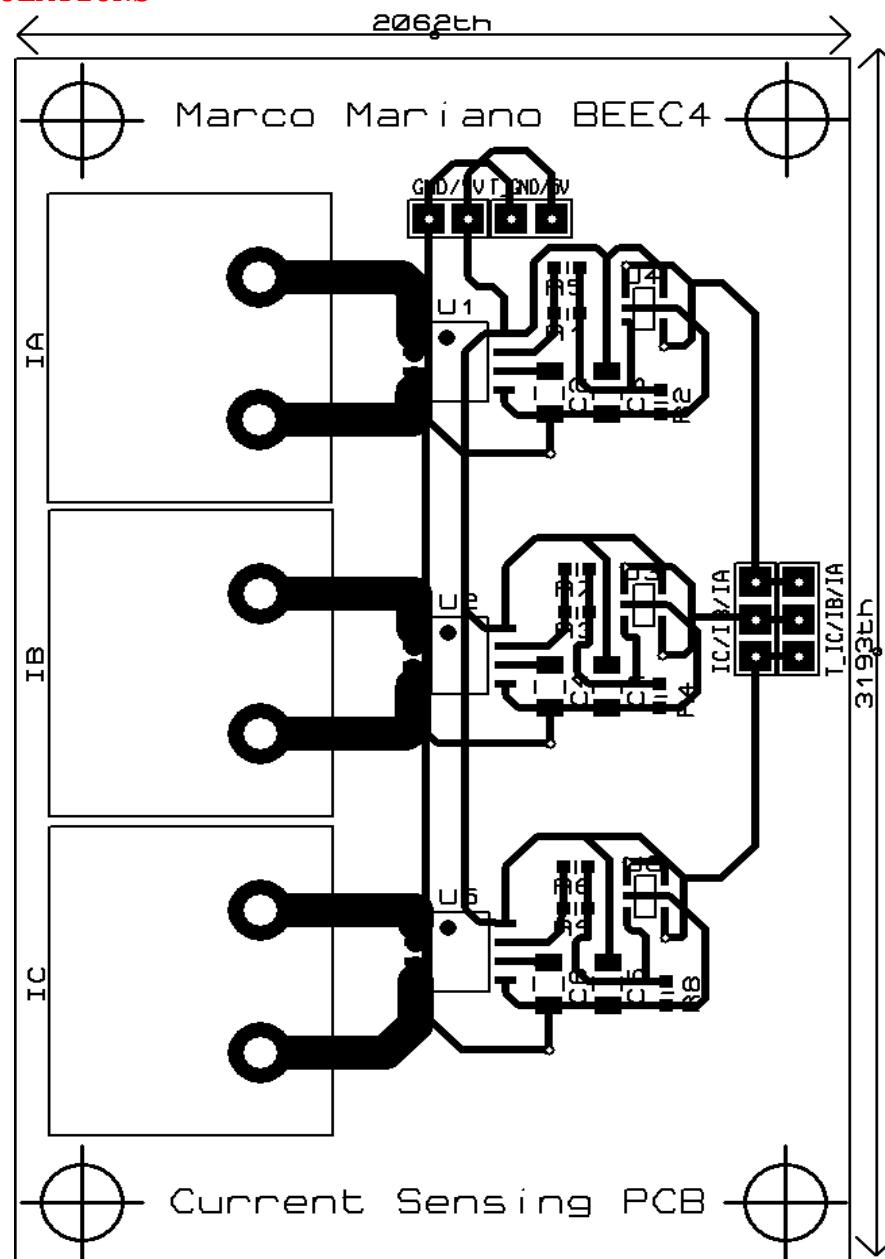
Appendix A.1: Top - Power supply circuit PCB design. Bottom - Power supply circuit hardware implementation



Appendix A.2: Left - Gate driver circuit PCB design. Right - Gate driver circuit hardware implementation



Appendix A.3: Left - Three phase inverter circuit PCB design. Right - Three phase inverter circuit hardware implementation



Appendix A.4: Left - Current sensing circuit PCB design. Right - Current sensing circuit hardware implementation

## Appendix B

Complete STM32 Software Algorithm Code

# RFOC Programming Code

```
1. //////////////////////////////////////////////////////////////////-Directives-////////////////////////////////////////////////////////////////
2. #include "usbd_cdc_core.h"
3. #include "math.h"
4. #include "global.c"
5. #include "usbd_usr.h"
6. #include "usb_conf.h"
7. #include "usbd_desc.h"
8. #include "usbd_cdc_vcp.h"
9. #include "stm32f4xx_gpio.h"
10. #include "NVIC_Configuration.c"
11. #include "GPIO.h"
12. #include "TIM1.h"
13. #include "LCDfunctions.h"
14. #include "OutputVoltageSVM_1a.h"
15. #include "CurrentController_2a.h"
16. #include "VoltageModel_1a.h"
17. #include "SpeedController_1a.h"
18. #include "position_data.c"
19.
20. #define ADC1_DR_ADDRESS ((uint32_t)0x4001204C)
21. #define VDC_LINK 12
22.
23. extern uint16_t VCP_DataTx (uint8_t* Buf, uint32_t Len);
24. extern void SendVariable2Host(int16_t variable);
25. extern uint8_t APP_Tx_Buffer [];
26. extern uint8_t APP_Rx_Buffer [];
27. extern uint32_t APP_Tx_ptr_in;
28. extern uint32_t APP_Rx_ptr_in;
29.
30. //////////////////////////////////////////////////////////////////-Function Prototypes-////////////////////////////////////////////////////////////////
31. void ADC1_DMA_Config(void);
32.
33. //////////////////////////////////////////////////////////////////-Variables-////////////////////////////////////////////////////////////////
34. int16_t ADCConvertedValue[64] = {0,0,0,0};
35. int32_t ADCConvertedValue_x[64] = {0,0,0,0};
36. float speed_counter = 1e-3;
37.
38. float iAlpha_hat = 0;
39. float iBeta_hat = 0;
40. float iAlpha_hat_km1 = 0;
41. float iBeta_hat_km1 = 0;
42. float Kp_o = 0;
43. float Ki_o = 0;
44. float BEMF_Alpha = 0;
45. float BEMF_Beta = 0;
46. float BEMF_Alpha_km1 = 0;
47. float BEMF_Beta_km1 = 0;
48. float iAlpha_Err = 0;
49. float iBeta_Err = 0;
50. float iAlpha_Err_km1 = 0;
51. float iBeta_Err_km1 = 0;
52. float theta_e = 0;
53. float T = 0.0001;
54.
55. float CosAngle_e = 0;
56. float SinAngle_e = 0;
57. int16_t check_counter = 0;
58. int16_t loop_counter = 0;
59. float Kp_i = 0;
60. float Ki_i = 0;
61. float Kp_i2 = 0;
62. float Ki_i2 = 0;
63. float vD_km1 = 0;
64. float vQ_km1 = 0;
65. float vD_km1_2 = 0;
66. float vQ_km1_2 = 0;
```

```
67.  
68. float vAlpha_km1 = 0;  
69. float vBeta_km1 = 0;  
70.  
71. float BEMF_Alpha_f = 0;  
72. float BEMF_Beta_f = 0;  
73. float BEMF_Alpha_f_km1 = 0;  
74. float BEMF_Beta_f_km1 = 0;  
75.  
76. float iAlpha_hat2 = 0;  
77. float iAlpha_hat2_km1 = 0;  
78. float iBeta_hat2 = 0;  
79. float iBeta_hat2_km1 = 0;  
80.  
81. float theta_e_new_km1 = 0;  
82. float theta_e_new = 0;  
83. float Omega_w = 0;  
84. float Omega_w_scale = 0;  
85. float Omega_w_km1 = 0;  
86. float Kp_PLL = 0;  
87. float Ki_PLL = 0;  
88. float PLL_Error = 0;  
89. float PLL_Error_km1 = 0;  
90.  
91. float W_Ref = 0;  
92. float Speed_Error = 0;  
93. float Kp_w = 0;  
94. float Ki_w = 0;  
95. float Speed_Error_km1 = 0;  
96.  
97.  
98. #ifdef USB_OTG_HS_INTERNAL_DMA_ENABLED  
99.     #if defined ( __ICCARM__ )  
100.         #pragma data_alignment=4  
101.     #endif  
102. #endif  
103.  
104.     __ALIGN_BEGIN USB_OTG_CORE_HANDLE      USB_OTG_dev __ALIGN_END ;  
105.  
106. //////////////////////////////////////////////////////////////////-Main Loop-////////////////////////////////////////////////////////////////  
107. int main(void)  
108. {  
109.  
110.     //////////////////////////////////////////////////////////////////-USBD Init-////////////////////////////////////////////////////////////////  
111.     USBD_Init(&USB_OTG_dev,  
112.     #ifdef      USE_USB_OTG_HS  
113.                 USB_OTG_HS_CORE_ID,  
114.             #else  
115.                 USB_OTG_FS_CORE_ID,  
116.             #endif  
117.                 &USR_desc,  
118.                 &USBD_CDC_cb,  
119.                 &USR_cb);  
120.  
121.     APP_Rx_ptr_in=0;  
122.  
123.     //////////////////////////////////////////////////////////////////-Function Calls-////////////////////////////////////////////////////////////////  
124.     GPIO_Setup();  
125.     Timer1Setup();  
126.     Timer10_Setup();  
127.     NVIC_Configuration();  
128.     ADC1_DMA_Config();  
129.  
130.     //////////////////////////////////////////////////////////////////-Main Output Enable-////////////////////////////////////////////////////////////////  
131.     TIM1->CCER |= 0x1555;  
132.     TIM_CtrlPWMOutputs(TIM1, ENABLE);  
133.  
134.     while (1)  
135.     {  
136.     }  
137. }
```

```
138.
139. #ifdef USE_FULL_ASSERT
140. void assert_failed(uint8_t* file, uint32_t line)
141. {
142.     while (1)
143.     {
144.     }
145. }
146. #endif
147.
148. void ADC1_DMA_Config(void)
149. {
150.     ADC_InitTypeDef      ADC_InitStructure;
151.     ADC_CommonInitTypeDef ADC_CommonInitStructure;
152.     DMA_InitTypeDef      DMA_InitStructure;
153.
154.     ///////////////////////////////////////////////////Enable peripheral clocks-///////////////////
155.     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_DMA2, ENABLE);
156.     RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
157.
158.     ///////////////////////////////////////////////////DMA2_Stream0 channel0 configuration-/////////////////
159.     DMA_DeInit(DMA2_Stream0);
160.     DMA_InitStructure.DMA_Channel = DMA_Channel_0;
161.     DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)ADC1_DR_ADDRESS;
162.     DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)&ADCConvertedValue;
163.     DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralToMemory;
164.     DMA_InitStructure.DMA_BufferSize = 64;
165.     DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
166.     DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
167.     DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
168.     DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
169.     DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
170.     DMA_InitStructure.DMA_Priority = DMA_Priority_High;
171.     DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Disable;
172.     DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_HalfFull;
173.     DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
174.     DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
175.     DMA_Init(DMA2_Stream0, &DMA_InitStructure);
176.     DMA_Cmd(DMA2_Stream0, ENABLE);
177.
178.     ///////////////////////////////////////////////////Enable DMA1 Channel1 complete transfer interrupt-///////////
179.     DMA_ITConfig(DMA2_Stream0, DMA_IT_TC , ENABLE);
180.
181.     ///////////////////////////////////////////////////ADC Common Init-///////////////////////////////
182.     ADC_CommonInitStructure.ADC_Mode = ADC_Mode_Independent;
183.     ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div4;
184.     ADC_CommonInitStructure.ADC_DMAAccessMode = ADC_DMAAccessMode_Disabled;
185.     ADC_CommonInitStructure.ADC_TwoSamplingDelay = ADC_TwoSamplingDelay_5Cycles;
186.     ADC_CommonInit(&ADC_CommonInitStructure);
187.
188.     ///////////////////////////////////////////////////ADC1 Init-///////////////////////////////
189.     ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
190.     ADC_InitStructure.ADC_ScanConvMode = ENABLE;
191.     ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
192.     ADC_InitStructure.ADC_ExternalTrigConvEdge = ADC_ExternalTrigConvEdge_None;
193.     ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
194.     ADC_InitStructure.ADC_NbrOfConversion = 4;
195.     ADC_Init(ADC1, &ADC_InitStructure);
196.     ADC_DMACmd(ADC1, ENABLE);
197.
198.     ///////////////////////////////////////////////////ADC1 regular channel18 (VBAT) configuration-///////////
199.     ADC-RegularChannelConfig(ADC1,ADC_Channel_1,4,ADC_SampleTime_3Cycles); //IQ_Ref
200.     ADC-RegularChannelConfig(ADC1,ADC_Channel_12,2,ADC_SampleTime_3Cycles); //iB
201.     ADC-RegularChannelConfig(ADC1,ADC_Channel_11,1,ADC_SampleTime_3Cycles); //iA
202.     ADC-RegularChannelConfig(ADC1,ADC_Channel_15,3,ADC_SampleTime_3Cycles); //iC
203.
204.     ///////////////////////////////////////////////////Enable DMA request after last transfer (Single-ADC mode)-/////////
205.     ADC_DMAResquestAfterLastTransferCmd(ADC1, ENABLE);
206.
207.     ///////////////////////////////////////////////////Enable ADC1-///////////////////////////////
208.     ADC_Cmd(ADC1, ENABLE);
```

```
209.  
210.    ADC_ITConfig(ADC1, ADC_IT_EOC, ENABLE);  
211. }  
212.  
213. void DMA2_Stream0_IRQHandler(void)  
214. {  
215.     GPIO_SetBits(GPIOOD, GPIO_Pin_13);  
216.  
217.     //////////////////////////////////////////////////////////////////-GET 3-phase Currents and Speed Info-////////////////////////////////////////////////////////////////  
218.     if (DMA_GetITStatus (DMA2_Stream0,DMA_IT_TCIF0) == SET)  
219.     {  
220.         DMA_ClearITPendingBit (DMA2_Stream0, DMA_IT_TCIF0);  
221.  
222.         ADCConvertedValue_x[0] = ADCConvertedValue[28] + ADCConvertedValue[24] +  
223.                                         ADCConvertedValue[20] + ADCConvertedValue[16] +  
224.                                         ADCConvertedValue[12] + ADCConvertedValue[8] +  
225.                                         ADCConvertedValue[4] + ADCConvertedValue[0] +  
226.                                         ADCConvertedValue[32] + ADCConvertedValue[36] +  
227.                                         ADCConvertedValue[40] + ADCConvertedValue[44] +  
228.                                         ADCConvertedValue[48] + ADCConvertedValue[52] +  
229.                                         ADCConvertedValue[56] + ADCConvertedValue[60];  
230.  
231.         ADCConvertedValue_x[1] = ADCConvertedValue[29] + ADCConvertedValue[25] +  
232.                                         ADCConvertedValue[21] + ADCConvertedValue[17] +  
233.                                         ADCConvertedValue[13] + ADCConvertedValue[9] +  
234.                                         ADCConvertedValue[5] + ADCConvertedValue[1] +  
235.                                         ADCConvertedValue[33] + ADCConvertedValue[37] +  
236.                                         ADCConvertedValue[41] + ADCConvertedValue[45] +  
237.                                         ADCConvertedValue[49] + ADCConvertedValue[53] +  
238.                                         ADCConvertedValue[57] + ADCConvertedValue[61];  
239.  
240.         ADCConvertedValue_x[2] = ADCConvertedValue[30] + ADCConvertedValue[26] +  
241.                                         ADCConvertedValue[22] + ADCConvertedValue[18] +  
242.                                         ADCConvertedValue[14] + ADCConvertedValue[10] +  
243.                                         ADCConvertedValue[6] + ADCConvertedValue[2] +  
244.                                         ADCConvertedValue[34] + ADCConvertedValue[38] +  
245.                                         ADCConvertedValue[42] + ADCConvertedValue[46] +  
246.                                         ADCConvertedValue[50] + ADCConvertedValue[54] +  
247.                                         ADCConvertedValue[58] + ADCConvertedValue[62];  
248.  
249.         ADCConvertedValue_x[3] = ADCConvertedValue[31] + ADCConvertedValue[27] +  
250.                                         ADCConvertedValue[23] + ADCConvertedValue[19] +  
251.                                         ADCConvertedValue[15] + ADCConvertedValue[11] +  
252.                                         ADCConvertedValue[7] + ADCConvertedValue[3] +  
253.                                         ADCConvertedValue[35] + ADCConvertedValue[39] +  
254.                                         ADCConvertedValue[43] + ADCConvertedValue[47] +  
255.                                         ADCConvertedValue[51] + ADCConvertedValue[55] +  
256.                                         ADCConvertedValue[59] + ADCConvertedValue[63];  
257.  
258.     iA = ADCConvertedValue_x[0]/16;  
259.     iA = iA - 2307;  
260.     iA = (iA/56) - 0.2;  
261.  
262.     iB = ADCConvertedValue_x[1]/16;  
263.     iB = iB - 2307;  
264.     iB = (iB/56) - 0.2;  
265.  
266.     iC = ADCConvertedValue_x[2]/16;  
267.     iC = iC - 2309;  
268.     iC = (iC/56) - 0.2;  
269.  
270.     //iQ_Ref = ADCConvertedValue_x[3]/16;  
271.     //iQ_Ref = iQ_Ref/4096;  
272.     //iQ_Ref = iQ_Ref*10;  
273.  
274.     W_Ref = ADCConvertedValue_x[3]/16;  
275.     W_Ref = W_Ref/4096;  
276.     W_Ref = W_Ref*100;  
277.  
278.     //////////////////////////////////////////////////////////////////-Alpha/Beta Reference Frame-////////////////////////////////////////////////////////////////  
279.     iAlpha = 0.6666666666666666*iA-0.3333333333333333*iB-0.3333333333333333*iC;
```

```
280.     iBeta = 0.577350269189626*iB-0.577350269189626*iC;
281.
282.     ///////////////////////////////////////////////////-OL Current Control-/////////////////////////////////////////////////
283.     if(check_counter < 300)
284.     {
285.         theta_r_sim = theta_r_sim + speed_counter;
286.
287.         while (theta_r_sim > 6.283185)
288.         {
289.             theta_r_sim = theta_r_sim - 6.283185;
290.             speed_counter = speed_counter + 1e-3;
291.
292.             if(speed_counter > 1e-1)
293.             {
294.                 speed_counter = speed_counter - 1e-3;
295.                 check_counter++;
296.             }
297.         }
298.
299.         while (theta_r_sim < 0)
300.         {
301.             theta_r_sim = theta_r_sim + 6.283185;
302.         }
303.
304.         CosAngle = cosf(theta_r_sim);
305.         SinAngle = sinf(theta_r_sim);
306.
307.         iD = ((CosAngle*iAlpha) + (SinAngle*iBeta));
308.         iQ = ((-SinAngle*iAlpha) + (CosAngle*iBeta));
309.
310.         iQ_Ref = 2;
311.         iD_Ref = 0;
312.
313.         iD_Err = iD_Ref - iD;
314.         iQ_Err = iQ_Ref - iQ;
315.
316.         Kp_i = 0;
317.         Ki_i = 2550;
318.
319.         vD=iD_Err*((Kp_i+(Ki_i*(T/2)))+iD_Err_km1*((-Kp_i+(Ki_i*(T/2))))+vD_km1;
320.         vQ=iQ_Err*((Kp_i+(Ki_i*(T/2)))+iQ_Err_km1*((-Kp_i+(Ki_i*(T/2))))+vQ_km1;
321.
322.         if (vD > VDC_LINK) vD = VDC_LINK;
323.         if (vD < -VDC_LINK) vD = -VDC_LINK;
324.         if (vQ > VDC_LINK) vQ = VDC_LINK;
325.         if (vQ < -VDC_LINK) vQ = -VDC_LINK;
326.
327.         iD_Err_km1=iD_Err;
328.         iQ_Err_km1=iQ_Err;
329.         vD_km1=vD;
330.         vQ_km1=vQ;
331.
332.         vAlpha = (vD*CosAngle)+(-vQ*SinAngle);
333.         vBeta = (vD*SinAngle)+(vQ*CosAngle);
334.     }
335.
336.     //////////////////////////////////////////////////-CL Speed and Current Control-/////////////////////////////////////////////////
337.     else
338.     {
339.         Speed_Error = W_Ref - Omega_w_scale;
340.
341.         Kp_w = 0;
342.         Ki_w = 1;
343.
344.         iQ_Ref = Speed_Error*((Kp_w+(Ki_w*(T/2)))+
345.                               Speed_Error_km1*((-Kp_w+(Ki_w*(T/2))))+iQ_Ref_km1;
346.
347.         if (iQ_Ref > 3) iQ_Ref = 3;
348.         if (iQ_Ref < 0) iQ_Ref = 0;
349.
350.         iD = ((CosAngle_e*iAlpha) + (SinAngle_e*iBeta));
```

```
351.     iQ = ((-SinAngle_e*iAlpha) + (CosAngle_e*iBeta));  
352.  
353.     iD_Ref = 0;  
354.  
355.     iD_Err = iD_Ref - iD;  
356.     iQ_Err = iQ_Ref - iQ;  
357.  
358.     Kp_i = 0;  
359.     Ki_i = 1000;  
360.  
361.     vD=iD_Err*((Kp_i+(Ki_i*(T/2))))+iD_Err_km1*((-Kp_i+(Ki_i*(T/2)))+vD_km1;  
362.     vQ=iQ_Err*((Kp_i+(Ki_i*(T/2))))+iQ_Err_km1*((-Kp_i+(Ki_i*(T/2)))+vQ_km1;  
363.  
364.     if (vD > VDC_LINK) vD = VDC_LINK;  
365.     if (vD < -VDC_LINK) vD = -VDC_LINK;  
366.     if (vQ > VDC_LINK) vQ = VDC_LINK;  
367.     if (vQ < -VDC_LINK) vQ = -VDC_LINK;  
368.  
369.     iD_Err_km1=iD_Err;  
370.     iQ_Err_km1=iQ_Err;  
371.     vD_km1=vD;  
372.     vQ_km1=vQ;  
373.     Speed_Error_km1 = Speed_Error;  
374.     iQ_Ref_km1 = iQ_Ref;  
375.  
376.     vAlpha = (vD*CosAngle_e)+(-vQ*SinAngle_e);  
377.     vBeta = (vD*SinAngle_e)+(vQ*CosAngle_e);  
378. }  
379.  
380. ///////////////////////////////////////////////////////////////////Observer/////////////////////////////////////////////////////////////////  
381. iAlpha_hat = (iAlpha_hat_km1*0.8) +  
382.             ((vAlpha-BEMF_Alpha)*(0.1+(0.1*vQ))) +  
383.             ((vAlpha_km1-BEMF_Alpha_km1)*(0.1+(0.1*vQ)));  
384.  
385. iBeta_hat = (iBeta_hat_km1*0.8) +  
386.             ((vBeta-BEMF_Beta)*(0.1+(0.1*vQ))) +  
387.             ((vBeta_km1-BEMF_Beta_km1)*(0.1+(0.1*vQ)));  
388.  
389. //iAlpha_hat = (iAlpha_hat_km1*0.8) +  
390. //             ((vAlpha-BEMF_Alpha)*(0.1+(0.05*iQ_Ref))) +  
391. //             ((vAlpha_km1-BEMF_Alpha_km1)*(0.1+(0.05*iQ_Ref)));  
392.  
393. //iBeta_hat = (iBeta_hat_km1*0.8) +  
394. //             ((vBeta-BEMF_Beta)*(0.1+(0.05*iQ_Ref))) +  
395. //             ((vBeta_km1-BEMF_Beta_km1)*(0.1+(0.05*iQ_Ref)));  
396.  
397. iAlpha_Err = iAlpha-iAlpha_hat;  
398. iBeta_Err = iBeta-iBeta_hat;  
399.  
400. Kp_o = 0;  
401. Ki_o = 1500;  
402.  
403. BEMF_Alpha = iAlpha_Err*((Kp_o+(Ki_o*(T/2)))) +  
404.                 iAlpha_Err_km1*((-Kp_o+(Ki_o*(T/2)))) +  
405.                 BEMF_Alpha_km1;  
406. BEMF_Alpha = BEMF_Alpha*(-1);  
407.  
408. BEMF_Beta = iBeta_Err*((Kp_o+(Ki_o*(T/2)))) +  
409.                 iBeta_Err_km1*((-Kp_o+(Ki_o*(T/2)))) +  
410.                 BEMF_Beta_km1;  
411. BEMF_Beta = BEMF_Beta*(-1);  
412.  
413. BEMF_Alpha_f = BEMF_Alpha_f_km1 +  
414.                 (0.0001*2*3.141592654*1000*(BEMF_Alpha-BEMF_Alpha_f));  
415. BEMF_Beta_f = BEMF_Beta_f_km1 +  
416.                 (0.0001*2*3.141592654*1000*(BEMF_Beta-BEMF_Beta_f));  
417.  
418. if (BEMF_Alpha > VDC_LINK) BEMF_Alpha = VDC_LINK;  
419. if (BEMF_Alpha < -VDC_LINK) BEMF_Alpha = -VDC_LINK;  
420. if (BEMF_Beta > VDC_LINK) BEMF_Beta = VDC_LINK;  
421. if (BEMF_Beta < -VDC_LINK) BEMF_Beta = -VDC_LINK;
```

```
422.  
423.         iAlpha_hat_km1 = iAlpha_hat;  
424.         iBeta_hat_km1 = iBeta_hat;  
425.         iAlpha_Err_km1 = iAlpha_Err;  
426.         iBeta_Err_km1 = iBeta_Err;  
427.         vAlpha_km1 = vAlpha;  
428.         vBeta_km1 = vBeta;  
429.         BEMF_Alpha_km1 = BEMF_Alpha;  
430.         BEMF_Beta_km1 = BEMF_Beta;  
431.         BEMF_Alpha_f_km1 = BEMF_Alpha_f;  
432.         BEMF_Beta_f_km1 = BEMF_Beta_f;  
433.  
434.         theta_e = atan2f((-1)*BEMF_Alpha_f),BEMF_Beta_f);  
435.  
436.         while (theta_e > 6.283185)  
437.         {  
438.             theta_e = theta_e - 6.283185;  
439.         }  
440.  
441.         while (theta_e < 0)  
442.         {  
443.             theta_e = theta_e + 6.283185;  
444.         }  
445.  
446.         CosAngle_e = cosf(theta_e);  
447.         SinAngle_e = sinf(theta_e);  
448.  
449.         ///////////////////////////////////////////////////////////////////Speed Estimator/////////////////////////////////////////////////////////////////  
450.         PLL_Error = (SinAngle_e*(cosf(theta_e_new)))-  
451.                 (CosAngle_e*(sinf(theta_e_new)));  
452.  
453.         Kp_PLL = 200;  
454.         Ki_PLL = 50000;  
455.  
456.         Omega_w = PLL_Error*((Kp_PLL+(Ki_PLL*(T/2)))+  
457.                         PLL_Error_km1*(-Kp_PLL+(Ki_PLL*(T/2))))+  
458.                         Omega_w_km1;  
459.         Omega_w_scale = Omega_w*0.07142857143;  
460.  
461.         theta_e_new = theta_e_new_km1 + (Omega_w*T);  
462.  
463.         while (theta_e_new > 6.283185)  
464.         {  
465.             theta_e_new = theta_e_new - 6.283185;  
466.         }  
467.  
468.         while (theta_e_new < 0)  
469.         {  
470.             theta_e_new = theta_e_new + 6.283185;  
471.         }  
472.  
473.         PLL_Error_km1 = PLL_Error;  
474.         Omega_w_km1 = Omega_w;  
475.         theta_e_new_km1 = theta_e_new;  
476.  
477.         ///////////////////////////////////////////////////////////////////SVM Output Function/////////////////////////////////////////////////////////////////  
478.         OutputVoltageSVM();  
479.  
480.         TIM1->CCR1 = (uint16_t) ((1-dA)*8400);  
481.         TIM1->CCR2 = (uint16_t) ((1-dB)*8400);  
482.         TIM1->CCR3 = (uint16_t) ((1-dC)*8400);  
483.  
484.         ///////////////////////////////////////////////////////////////////Send to MATLAB/////////////////////////////////////////////////////////////////  
485.         if(NumberOfSamplePointsToCapture)  
486.         {  
487.             SendVariable2Host((int16_t)(iA*100));  
488.             SendVariable2Host((int16_t)(iB*100));  
489.             SendVariable2Host((int16_t)(iC*100));  
490.  
491.             SendVariable2Host((int16_t)(iAlpha*100));  
492.             SendVariable2Host((int16_t)(iBeta*100));
```

```
493.     SendVariable2Host((int16_t)(iAlpha_hat*100));  
494.     SendVariable2Host((int16_t)(iBeta_hat*100));  
495.  
496.     SendVariable2Host((int16_t)(BEMF_Alpha*100));  
497.     SendVariable2Host((int16_t)(BEMF_Beta*100));  
498.     SendVariable2Host((int16_t)(theta_e*100));  
499.     SendVariable2Host((int16_t)(theta_r_sim*100));  
500.  
501.     SendVariable2Host((int16_t)(iD*100));  
502.     SendVariable2Host((int16_t)(iQ*100));  
503.     SendVariable2Host((int16_t)(iD_Ref*100));  
504.     SendVariable2Host((int16_t)(iQ_Ref*100));  
505.  
506.     SendVariable2Host((int16_t)(BEMF_Alpha_f*100));  
507.     SendVariable2Host((int16_t)(BEMF_Beta_f*100));  
508.  
509.     SendVariable2Host((int16_t)(theta_e_new*100));  
510.     SendVariable2Host((int16_t)(Omega_w_scale*100));  
511.     SendVariable2Host((int16_t)(W_Ref*100));  
512.     SendVariable2Host((int16_t)(Omega_w*100));  
513.  
514.     PWM_CycleCounter++;  
515.  
516.     if(PWM_CycleCounter>=NumberOfSamplePointsToCapture)  
517.     {  
518.         NumberOfSamplePointsToCapture=0;  
519.         PWM_CycleCounter=0;  
520.     }  
521. }  
522.  
523. else  
524. {  
525.  
526. };  
527. }  
528.  
529. GPIO_ResetBits(GPIOD, GPIO_Pin_13);  
530. }
```

## SVM Programming Code

```
1. #include "OutputVoltageSVM_1a.h"  
2. #include <stdint.h>  
3. #include "stm32f4xx_gpio.h"  
4. #include "GPIO.h"  
5.  
6. void OutputVoltageSVM(void)  
7. {  
8.     //Space Vector Modulation  
9.     Ua=(1.732050807568877*vAlpha)/12;  
10.    Ub=(-vBeta)/12;  
11.  
12.    X=Ub;  
13.    Y=(Ua+Ub)/2;  
14.    Z=(Ub-Ua)/2;  
15.  
16.    if(X<0)  
17.    {  
18.        if(Y>=0)  
19.        {  
20.            dA=(1-X+Y)/2;  
21.            dB=dA-Y;  
22.            dC=dB+X;  
23.        }
```

```
24.     else
25.     {
26.         if(Z<0)
27.         {
28.             dA=(1+Y-Z)/2;
29.             dB=dA-Y;
30.             dC=dA+Z;
31.         }
32.     else
33.     {
34.         dA=(1+X-Z)/2;
35.         dC=dA+Z;
36.         dB=dC-X;
37.     }
38. }
39. }
40.
41. else
42. {
43.     if(Y<0)
44.     {
45.         dA=(1-X+Y)/2;
46.         dB=dA-Y;
47.         dC=dB+X;
48.     }
49. else
50. {
51.     if(Z>=0)
52.     {
53.         dA=(1+Y-Z)/2;
54.         dB=dA-Y;
55.         dC=dA+Z;
56.     }
57. else
58. {
59.     dA=(1+X-Z)/2;
60.     dC=dA+Z;
61.     dB=dC-X;
62. }
63. }
64. }
65.
66. //End: Space Vector Modulation
67.
68. //limit duty cycles
69. if(dA>dMax) dA=dMax;
70. if(dA<dMin) dA=dMin;
71. if(dB>dMax) dB=dMax;
72. if(dB<dMin) dB=dMin;
73. if(dC>dMax) dC=dMax;
74. if(dC<dMin) dC=dMin;
75. }
```

## Appendix C

A paper entitled “Modelling of a Sensorless Rotor Flux Oriented BLDC machine” submitted for peer review at the 19th International Conference on ELECTRICAL DRIVES and POWER ELECTRONICS October 04 - 06, 2017, Dubrovnik, CROATIA

# Modelling of a Sensorless Rotor Flux Oriented BLDC machine

M. Mariano

Student, Institute of Engineering and Transport, Electrical and Electronics  
Malta College for Arts, Science and Technology

K. Scicluna, J. Scerri

Institute of Engineering and Transport, Electrical and Electronics  
Malta College for Arts, Science and Technology

**Abstract**— This paper presents the modelling and simulation of a low-power Brushless DC machine in a Rotor Flux Oriented mode in MATLAB/Simulink. Sensorless simulation estimates obtained from a back-Emf observer for both electrical angle and rotor speed are presented. The dynamic response of the model is also shown in sensorless closed-loop whereby the estimates are used for control purposes.

**Keywords**—BLDC; Modelling, Simulink; sensorless;

## I. INTRODUCTION

The Brushless DC (BLDC) machine has been widely used in applications where high-speed and low maintenance are critical requirements. Such applications include unmanned vehicles [1, 2], automotive [3-5] and aerospace [6-8]. Typically to control BLDC machines in closed-loop shaft speed/position measurements are required. While these can be obtained through dedicated sensors such as encoders and tachogenerators, typically such an approach is not feasible due to limited application area and additional cost.

Hence BLDC machines are usually combined with sensorless speed/position algorithms. Sensorless algorithms are subdivided into two main categories: model based and non-model based. Model based sensorless algorithms estimate the position/speed of the machine through electrical or mechanical models and are particularly suited for high speed operation [9-11]. Non-model based algorithms are used at close to zero speeds and rely on injection additional carrier signals to the fundamental ones required for control [12, 13]. In speed controlled systems such as the one modelled in this paper the back-emf of the machine is usually sufficient for accurate speed/position estimates to be obtained for sensorless closed-loop purposes.

In low-cost Electronic Speed Controllers (ESCs) used with low to medium power BLDC machines a six step commutation sequence is used [14]. In this method one of the three phases of the machine is left floating allowing for a back-emf voltage measurement to be taken. The drawback of the six step approach is that it has a high harmonic content which would result in higher power losses [15, 16]. Since the aim of this paper is to model and simulate a low-power BLDC motor in MATLAB/Simulink which is typically used in

battery powered applications; a Rotor Flux Oriented (RFO) control topology is to be used assuming a PWM inverter control scheme instead of the six-step commutation approach. Therefore the back-emf is estimated through current measurements rather than measured through the floating phase.

## II. ROTOR FLUX ORIENTED MODELLING

### A. Modelling of the BLDC in the DQ Frame

In the modelling presented in this paper vector control is assumed with the synchronous frame d-axis oriented on the rotor flux. This approach has been widely used to control synchronous machines and has been shown to be robust under varying load conditions [17, 18]. Using this representation the stator voltages in dq frame are expressed in (1) and (2).

$$v_{sd} = R_{isd} + L_d \frac{di_{sd}}{dt} - \omega L_q i_{sq} \quad (1)$$

$$v_{sq} = R_{isq} + L_q \frac{di_{sq}}{dt} + \omega L_d i_{sd} + \omega \varphi_{rd} \quad (2)$$

Where:

$v_{sd}$ / $v_{sq}$	are the stator voltages in the dq-frame
$R$	is the stator resistance
$i_{sd}$ / $i_{sq}$	are the stator currents in the dq-frame
$L_d$ / $L_q$	are the stator inductances in the dq-frame
$\omega$	is the synchronous frequency
$\varphi_{rd}$	is the rotor flux aligned to the d-axis

The mechanical differential equation relating rotor shaft speed to the BLDC electromagnetic torque is expressed in (3).

$$T_e = J\omega_m + B\omega_m \quad (3)$$

Where:

$T_e$	is the electromagnetic torque of the BLDC
$J$	is the moment of inertia of the BLDC rotor
$\omega_m$	is the rotor speed
$B$	is the viscous friction coefficient of the BLDC rotor

The necessary parameters used for modelling were based on the experimental BLDC machine shown in Fig. 1 and shown in Table I.



Fig. 1. Experimental BLDC used for modelling.

TABLE I. BLDC MACHINE PARAMETERS

Symbol	Description	Value	Unit
P	Power Rating	187	W
V <sub>rms</sub>	RMS Voltage	11	V
T <sub>rated</sub>	Rated Load Torque	0.1437	Nm
L <sub>d</sub>	D frame stator inductance	6.5	μH
L <sub>q</sub>	Q frame stator inductance	6.5	μH
R	Stator Resistance	81.5	mΩ
J	Moment of Inertia	7.312x10 <sup>-6</sup>	kg m <sup>2</sup>
B	Coefficient of Friction	6.511x10 <sup>-7</sup>	Nms
p	Number of Pole Pairs	14	-

### B. Sensored Control of the BLDC

By applying Laplace transformations to (1), (2) and (3) and using frequency domain tools the Proportional-Integral (PI) controllers of the cascaded RFO control topology shown in Fig. 2 were tuned with gains and bandwidths stated in Table II.

TABLE II. CONTROLLER GAINS

Loop	K <sub>p</sub>	K <sub>i</sub>	Closed-Loop Bandwidth [rad/s]	Damping Ratio
Current	0.08125	1018.75	12.5x10 <sup>3</sup>	-
Speed	0.25223	111.826	625	0.707

No field weakening is assumed for the operation of the BLDC such that the d-axis reference  $i_d^* = 0$  A. Hence the torque of the electrical machine is a function of the q-axis current  $i_q$  as shown in (4).

$$T_e = 3\pi i_q \varphi_{rd} \quad (4)$$

Where:

T <sub>e</sub>	is the electromagnetic torque of the BLDC
P	is the number of pole pairs
i <sub>q</sub>	is the stator currents in the q axis
φ <sub>rd</sub>	is the rotor flux aligned to the q-axis

The reference for the q-axis current  $i_q^*$  is generated from the error of the speed reference  $\omega_m^*$  and actual speed  $\omega_m$ . The outputs of the current PI controllers are the references for the stator voltages in the dq-frame  $v_{sd}^*$  and  $v_{sq}^*$ . In the simulation results presented in this paper the generation of pulse width modulation signals from the stator voltage references has not been considered as ideal three phase voltages  $v_a$ ,  $v_b$  and  $v_c$  are applied to the model of the BLDC.

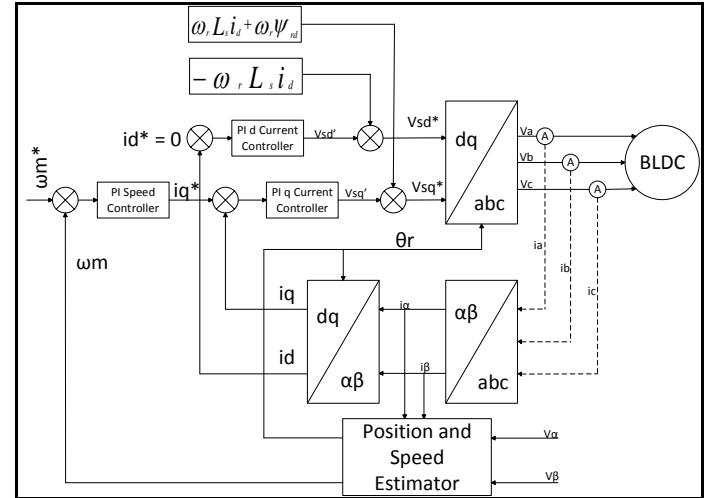


Fig. 2. RFO control of BLDC.

### III. SENSORLESS POSITION/SPEED ESTIMATION

For correct operation of the RFO control topology presented in Section II accurate measurements are required for the rotor position and speed. At high speeds the electrical angle of the BLDC can be obtained from trigonometry as shown in Fig. 3. Since the vector control algorithm will keep the back-emf vector  $E_s$  at an angle of 90° shifted from the d-axis the electrical angle  $\theta_e$  can be found from the back-emfs in the stationary αβ-frame through equation (5).

$$\theta_e = \tan^{-1} \left( \frac{E_\alpha}{E_\beta} \right) \quad (5)$$

Where:

θ <sub>e</sub>	is the electromagnetic angle
E <sub>α</sub> / E <sub>β</sub>	are the back-emfs in the αβ-frame

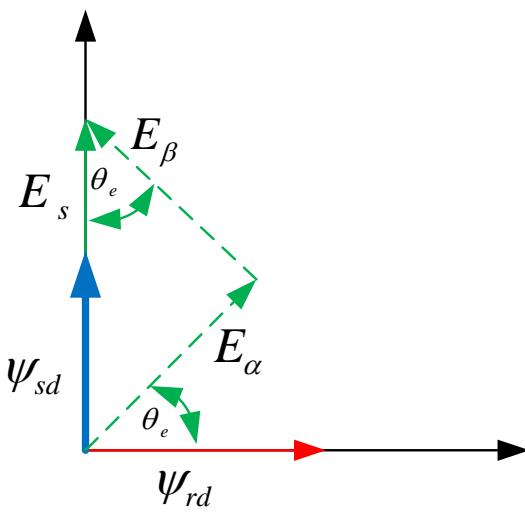


Fig. 3. Caclulation of the electrical angle from back-emf quantities.

As discussed previously in Section I the back-emf quantities required for the estimation of the electrical angle can be obtained through a direct measurement in the floating phase when using the six step commutation. Another approach which can be used with a PWM inverter is to estimate  $E_\alpha$  and  $E_\beta$  through current measurements  $i_\alpha$  and  $i_\beta$ . The actual currents  $i_\alpha$  and  $i_\beta$  are compared to estimates generated from the difference of the phase voltages  $v_\alpha$  and  $v_\beta$  subtracted from the back-emf estimates as shown in Fig. 4. The error in the back-emf estimates are compensated with a suitable PI controller such that the estimated currents track the actual currents measured. The calculation of (5) with the estimated back-emf quantities produces the electrical angle estimate which is differentiated to obtain the electrical frequency. Successful sensorless closed-loop was obtained with the tracking PI gains set at  $K_p = 0.08125$  and  $K_i = 1018.75$ .

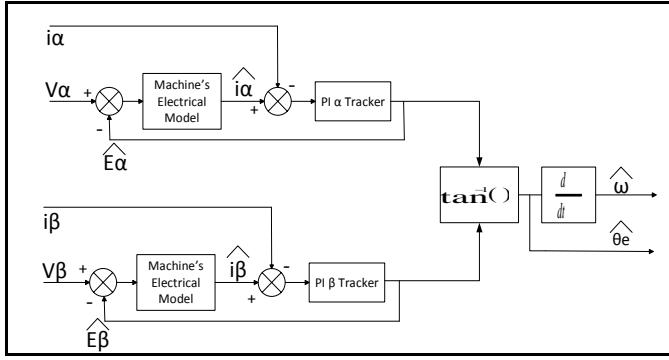


Fig. 4. Model-based sensorless observed for BLDC

#### IV. SIMULATION RESULTS

The simulation results shown in this section are for the BLDC described in Section II with a reference speed of 400 rad/s (3820 rpm). A constant rated load torque of 0.1437 Nm is applied at the output shaft of the BLDC. The BLDC is started from 0 rad/s up to the specified reference value in open-loop in such a way that the back-emf is of sufficient magnitude to be estimated. The sensorless speed and position estimates are used for a sensorless closed-loop operation after 0.2 s. Simulation results for the actual/estimated electrical angles required for vector control purposes is shown in Fig. 5 while the error between these two quantities is shown in Fig. 6. The reference/actual rotor speeds are shown in Fig. 7. The actual and estimated currents in the  $\alpha\beta$ -frame are shown in Figs. 8 and 9 respectively. The estimated back-emf from which the electrical angle is calculated is shown in Fig. 10. The synchronous frame dq currents are shown in Fig. 11.

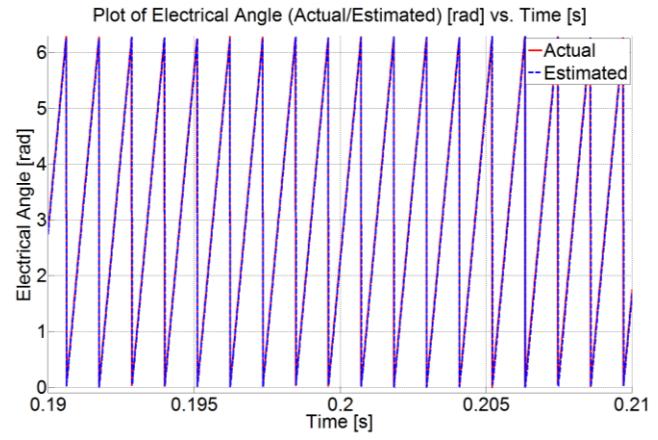


Fig. 5. Plot of Electrical Angle [rad] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at t=0.2 s.

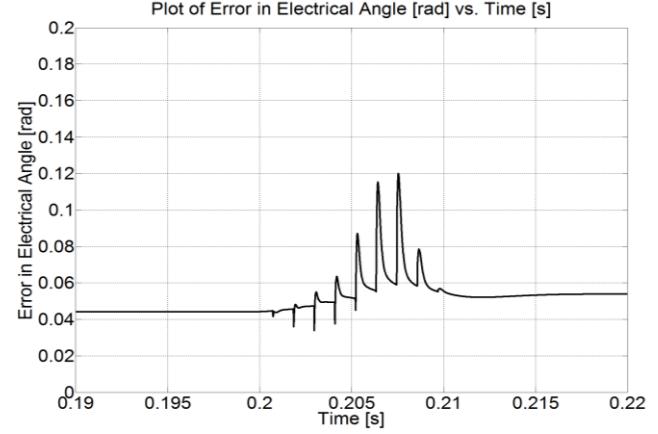


Fig. 6. Plot of Error in Electrical Angle [rad] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at t=0.2 s.

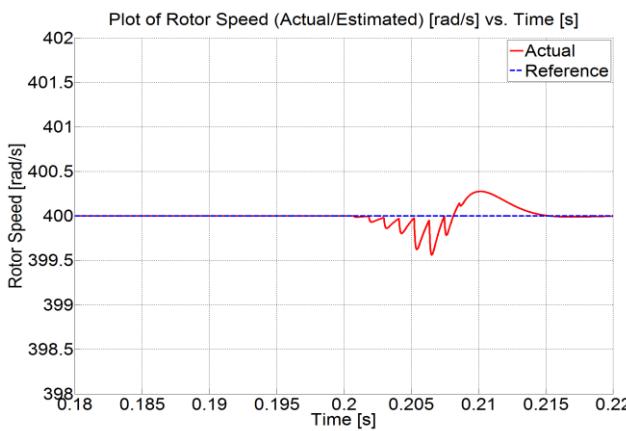


Fig. 7. Plot of Rotor Speed [rad/s] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at  $t=0.2$  s.

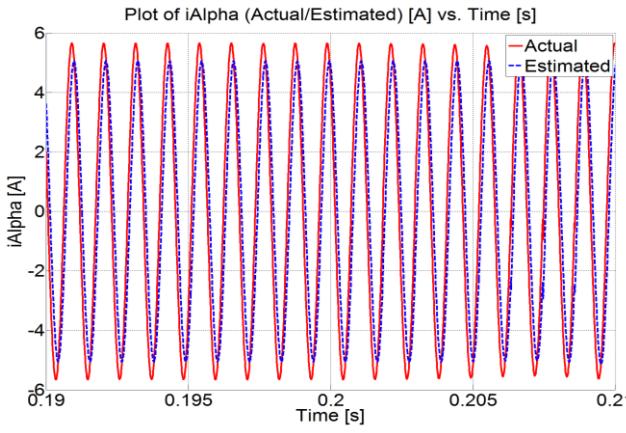


Fig. 8. Plot of  $i_a$  [A] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at  $t=0.2$  s.

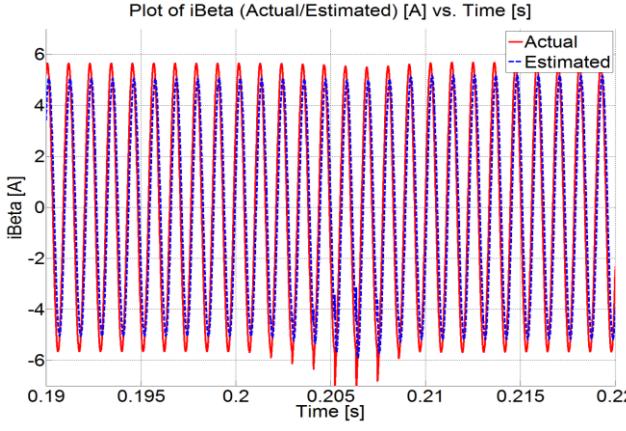


Fig. 9. Plot of  $i_b$  [A] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at  $t=0.2$  s.

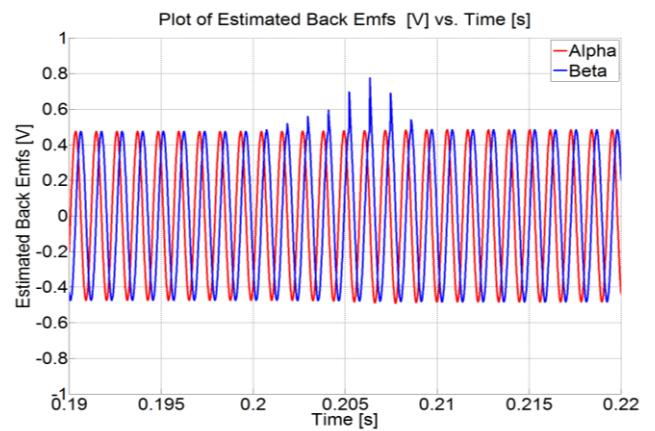


Fig. 10. Plot of Estimated  $E_a/E_\beta$  [V] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at  $t=0.2$  s.

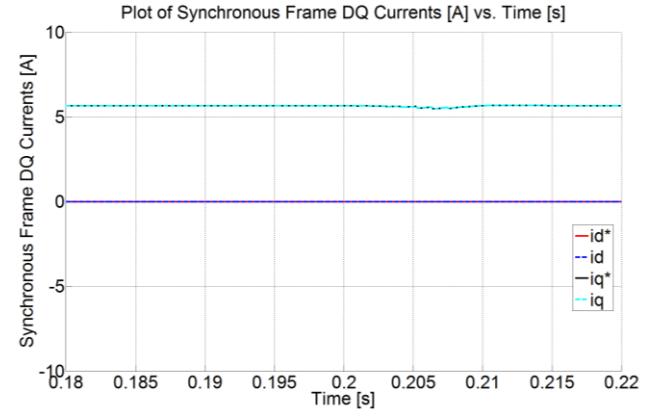


Fig. 11. Plot of  $i_d/i_q$  [A] vs. Time [s] with Load Torque of 0.1437 Nm and sensorless changeover at  $t=0.2$  s.

The estimated electrical angle is shown to track the actual electrical angle in Fig. 5. The error in the estimated angle is shown in Fig. 6 and has a maximum value of 0.12 rad (6.87 degrees), which is within 1.9 % of the actual value. The speed control loop is shown to be stable after sensorless changeover as the maximum error in the actual speed with respect to the reference is 0.4 rad/s (0.1 % of the reference value). Figs. 8 and 9 show that the sensorless estimation loops result in  $i_a$  and  $i_b$  estimates which are attenuated and phase shifted with respect to the actual quantities; this is expected due to the frequency response of the PI controllers. However since both  $\alpha$  and  $\beta$  quantities are tracked with identical controllers the effects on the estimated electrical parameters are negligible. The estimated back-emfs shown in Fig. 10 are of a sinusoidal nature with transients about the instant of sensorless changeover at 0.2 s. This is a result of the transients on  $i_b$  shown in Fig. 9 which are not tracked by the sensorless observer. This is due to the limited bandwidth of the PI tracker. The synchronous frame currents in Fig. 11 show that the d-axis current  $i_d = 0$  A while the q-axis current  $i_q$  has a positive value to produce a torque in the positive direction as required in the speed control loop.

## V. CONCLUSIONS

This paper has reviewed the modelling of the BLDC machine for vector control purposes in the dq-frame so as to facilitate modelling in MATLAB/Simulink. The stator voltage electrical dynamics and rotor mechanical dynamics were presented in Section II. These equations are required to implement the necessary PI controllers required for speed RFO control of the BLDC machine. A model-based sensorless observer which is applicable at high speeds was shown in Section III and the dependence of the electrical angle  $\theta_e$  discussed. The proposed sensorless observer estimates the back-emf quantities  $E_a$  and  $E_b$  by comparing estimates for the currents in the  $\alpha\beta$ -frame  $i_a$  and  $i_b$ . The tuning of a suitable observer to track these estimates was also presented.

From the simulation results presented in Section IV it was noted that the sensorless observer designed and simulated fails to track current transients in the  $\alpha\beta$ -frame. However the effect of this limitation associated with the bandwidth of the PI controllers has a negligible effect on the operation of the BLDC as the error in the estimated angle is a maximum of 1.9 % while the error in the actual speed is 0.1 %. The tuning of these PI controllers was set such that a sensorless operation is possible over a wide range of loading conditions. The bandwidth of the observer can theoretically be increased indefinitely to track the transients however this tends to result in an unstable control loop at the instant of sensorless changeover.

The sensorless observer presented in this paper has the disadvantage of having to estimate back-emf quantities unlike that associated with reading the floating phase in a six-step commutation. However it has the advantage that it can be used with a PWM inverter for improved efficiency in battery operated applications. Hence it is the opinion of the authors that such a sensorless observer is more suitable for BLDC machines with a lower Kv rating and a higher rotor inertia such that the transients are physically limited. The advantages of the modelling presented in this paper is that the various controllers within both the cascaded RFO oriented topology and the sensorless observer can be tested simultaneously for stability under various load conditions. The model has also been designed in such a way that it can be modified to test machines of different power ratings and saliency types.

## REFERENCES

- [1] J. A. Benito, G. Glez-de-Rivera, J. Garrido, and R. Ponticelli, "Design considerations of a small UAV platform carrying medium payloads," in *Design of Circuits and Integrated Systems*, 2014, pp. 1-6.
- [2] A. Sanchez, L. García Carrillo, E. Rondon, R. Lozano, and O. Garcia, "Hovering flight improvement of a quad-rotor mini UAV using brushless DC motors," *Journal of Intelligent & Robotic Systems*, vol. 61, pp. 85-101, 2011.
- [3] J. E. Makaran and J. LoVetri, "Prediction of conducted RFI emissions in BLDC motors for automotive applications," in *2001 IEEE EMC International Symposium. Symposium Record*.
- [4] T. W. Chun, Q. V. Tran, H. H. Lee, and H. G. Kim, "Sensorless Control of BLDC Motor Drive for an Automotive Fuel Pump Using a Hysteresis Comparator," *IEEE Transactions on Power Electronics*, vol. 29, pp. 1382-1391, 2014.
- [5] S. Jianwen, D. Nolan, M. Teissier, and D. Swanson, "A novel microcontroller-based sensorless brushless DC (BLDC) motor drive for automotive fuel pumps," *IEEE Transactions on Industry Applications*, vol. 39, pp. 1734-1740, 2003.
- [6] X. Huang, K. Bradley, A. Goodman, C. Gerada, P. Wheeler, J. Clare, *et al.*, "Fault-Tolerant Brushless DC Motor Drive For Electro-Hydrostatic Actuation System In Aerospace Application," in *Conference Record of the 2006 IEEE Industry Applications Conference Forty-First IAS Annual Meeting*, 2006, pp. 473-480.
- [7] X. Huang, A. Goodman, C. Gerada, Y. Fang, and Q. Lu, "Design of a Five-Phase Brushless DC Motor for a Safety Critical Aerospace Application," *IEEE Transactions on Industrial Electronics*, vol. 59, pp. 3532-3541, 2012.
- [8] X. Huang, A. Goodman, C. Gerada, Y. Fang, and Q. Lu, "A Single Sided Matrix Converter Drive for a Brushless DC Motor in Aerospace Applications," *IEEE Transactions on Industrial Electronics*, vol. 59, pp. 3542-3552, 2012.
- [9] H. Kim, J. Son, and J. Lee, "A High-Speed Sliding-Mode Observer for the Sensorless Speed Control of a PMSM," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 4069-4077, 2011.
- [10] K. Kye-Lyong, K. Jang-Mok, H. Keun-Bae, and K. Kyung-Hoon, "Sensorless control of PMSM in high speed range with iterative sliding mode observer," in *Applied Power Electronics Conference and Exposition, 2004. APEC '04. Nineteenth Annual IEEE*, 2004, pp. 1111-1116 vol.2.
- [11] F. Parasiliti, R. Petrella, and M. Tursini, "Sensorless speed control of a PM synchronous motor based on sliding mode observer and extended Kalman filter," in *Conference Record of the 2001 IEEE Industry Applications Conference. 36th IAS Annual Meeting (Cat. No.01CH37248)*, 2001, pp. 533-540 vol.1.
- [12] C. S. Staines, C. Caruana, and R. Raute, "A Review of Saliency-based Sensorless Control Methods for Alternating Current Machines," *IEEJ Journal of Industry Applications*, vol. 3, pp. 86-96, 2014.
- [13] K. Scicluna, C. S. Staines, and R. Raute, "Sensorless position control of a PMSM for steer-by-wire applications," in *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, 2016, pp. 046-051.
- [14] U. V. S. Pola, and K. P. Vittal, "Recent Developments in Control Schemes of BLDC Motors," in *2006 IEEE International Conference on Industrial Technology*, 2006, pp. 477-482.
- [15] L. Byoung-Kuk, K. Tae-Hyung, and M. Ehsani, "On the feasibility of four-switch three-phase BLDC motor drives for low cost commercial applications: topology and control," *IEEE Transactions on Power Electronics*, vol. 18, pp. 164-172, 2003.
- [16] R. Krishnan and S. Lee, "PM brushless DC motor drive with a new power converter topology," in *Industry Applications Conference, 1995. Thirtieth IAS Annual Meeting, IAS '95., Conference Record of the 1995 IEEE*, 1995, pp. 380-387 vol.1.
- [17] M. Lazor and M. Štrajter, "Modified field oriented control for smooth torque operation of a BLDC motor," in *2014 ELEKTRO*, 2014, pp. 180-185.
- [18] N. Matsui, "Sensorless PM brushless DC motor drives," *Industrial Electronics, IEEE Transactions on*, vol. 43, pp. 300-308, 1996.