DESIGN OF A SYNCHRONOUS RELUCTANCE MOTOR DRIVE

A Project Report

submitted by

SAMEER KULKARNI

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY and MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2019

THESIS CERTIFICATE

This is to certify that the thesis titled **DESIGN OF A SYNCHRONOUS RELUCTANCE MOTOR DRIVE**, submitted by **SAMEER KULKARNI**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology and Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Krishna Vasudevan
Research Guide
Professor
Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai

Date: 3 May 2019

ACKNOWLEDGEMENTS

I thank my guide Prof. Krishna Vasudevan for his valuable suggestions and guidance during the course of this project. His course on Electrical Machines piqued my interest in the field of motor drives, and culminated in this project.

I thank my lab mates - especially Arun, Arjun, Deepthi, and Jayakumar - and the lab staff, without whom, the hardware implementation would not have been possible. I cannot thank them enough for their support in setting up the hardware and debugging the common hardware issues.

I also thank all the professors at IIT Madras who taught me the various concepts that were employed directly or indirectly in this project. Attending their classes was a pleasant and enriching experience.

Finally, I thank my parents and grandparents for their continued encouragement and support over the years. Without them, this project would not have been possible.

ABSTRACT

KEYWORDS: Synchronous Reluctance Motor; Space Vector Pulse Width Modulation; Motor Drive; Sensorless Control; Extended Kalman Filter; Flux Weakening

The Synchronous Reluctance motor is a variant of the Synchronous motor that relies on the principle of magnetic reluctance for generating torque. Due to its cost effective nature and the higher efficiency it offers compared to most conventional motors, it is being looked at as an alternative for the Induction Motor and Permanent Magnet Synchronous Motor for Electric Vehicles.

The design of a Synchronous Reluctance Motor drive using the Field Oriented Control method is discussed in this project. The motivation behind the use of sensorless control techniques is stated and the applicability of the Extended Kalman Filter for estimating the state of the motor is shown. The theory behind flux weakening control of the motor is also discussed.

Simulation studies are conducted in Simulink to show the working of Field Oriented Control and sensorless control using Extended Kalman Filter. It is shown that with proper tuning, the EKF can provide accurate estimates of the motor state variables and hence be used for sensorless control.

The project concludes with hardware implementation of the Field Oriented Control method on an actual Synchronous Reluctance Motor and demonstration of the starting, speed reversal and braking operation of the motor under rated speed and voltage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SyRM	Synchronous Reluctance Motor
EKF	Extended Kalman Filter
PMSM	Permanent Magnet Synchronous Motor
IM	Induction Motor
FOC	Field Oriented Control
DTC	Direct Torque Control
EV	Electric Vehicles
PWM	Pulse Width Modulation
SPWM	Sinusoidal Pulse Width Modulation
SVPWM	Space Vector Pulse Width Modulation
EPWM	Enhanced Pulse Width Modulation
VSI	Voltage Source Inverter
DSP	Digital Signal Processor
CCS	Code Composer Studio
TI	Texas Instruments
ADC	Analog to Digital Converter
GPIO	General Purpose Input Output
MSB	Most Significant Bit
FFT	Fast Fourier Transform
EMI	Electromagnetic Interference

NOTATION

B	Windage loss coefficient
θ_m	Mechanical angle of rotor
θ_e	Electrical angle of rotor
i_d, i_q	Currents in direct and quadrature axes
I_s	Motor current limit
J	Moment of inertia of rotor
K	Kalman Gain
L_d, L_q	Direct and Quadrature axis inductances
p	Number of pole pairs in the motor
$P_{k\|k}$	Confidence matrix at k^{th} time step
R_s	Stator coil resistance
τ_g	Generated torque
τ_L	Load torque
T_s	Time step
v_d, v_q	Voltages in direct and quadrature axes
v_α, v_β	Voltages in alpha and beta axes
V_{dc}	DC Bus voltage
$\overrightarrow{V}$	Voltage space phasor
V_s	Inverter voltage limit
ω_m	Mechanical rotor speed
ω	Electrical rotor speed
$x_{k\|k}$	Discretized state matrix at k^{th} time step
$x_{k+1\|k}$	Predicted state matrix at k^{th} time step

CHAPTER 1

INTRODUCTION

The electrical motor is an electromechanical device that has led to numerous advancements in the field of engineering and technology. From something as trivial as the vibration of the cellphone to something as complex as precision cutting tools, motors are used for a wide range of applications. Enabling conversion of electric energy to mechanical energy has led to motors being an indispensable device in the advancement of technology and thus it is no wonder that the Electric motor market was valued at $91 billion as of 2015 and is expected to be valued at $117 billion by 2022 [25]. Amounting to 40%[19] of all electric consumption, motors and generators are indeed the backbone of the modern society.

1.1 Relevance of Motor Control

With motors becoming an integral part of all industries, it became a necessity to devise means to control their speed and torque so as to make them suitable for specialized applications. Various open loop and closed loop control strategies have evolved since motors first started being used in industries and the goal for each advancement has been the improvement of performance, precision and efficiency.

In early 20th century, DC motors had been the mainstay for variable-speed applications due to the ease of controlling them and the AC motors were preferred for constant speed applications. However with the advancements in semiconductor technology and improvements in microcomputers, by 1980s[34], the AC motors were also being employed for variable speed applications.

The earliest and the most basic variable frequency control strategy developed for AC motors was the V/f control. Given the semiconductor technology at that time, it was not very difficult to implement and it worked for a wide range of applications. However, it offered sub-optimal performance in applications involving low speed or demanding

dynamic speed regulation or reversing load requirements. The research in the field of motor control in the subsequent years led to the introduction of vector control and soon, AC motors surpassed DC motors in terms of precision, speed range and response.

Vector control opened up several new areas of applications for motors, the most prominent ones being precision machinery and process automation. Motor control now involved, not only the motor itself but also the inverter that controlled the voltage input to the motor, the switching technique, and the microcontroller, which in turn controlled the inverter. This entire system consisting of the motor, inverter and the microcontroller is essentially what a 'Motor Drive' is.

In the initial stages, vector control relied heavily on sensors, requiring accurate measurements of rotor position and speed. However, usage of sensors also led to various practical problems. The sensors required a complex hardware setup and the mechanical attachment of the sensor to the machine was not easy even in a lab environment. Most sensors were difficult to use in rough environments and needless to say, they also led to increased cost of the system. Furthermore, the reliability of the readings was always a problem and being prone to electromagnetic noise interference only aggravated it.

Although one could use scalar V/f control to get rid of these problems, the demand of high performance would not be met. As a result, research in sensorless control schemes gained popularity and by the 1990s, sensorless control of AC drives had improved dramatically[29].

However, there was only so much an AC motor could do even when equipped with vector control. Although it worked flawlessly for the applications it was designed for, it did not perform as well for power generation and did not have good power density. Soon, the need to look at different types of motors for more specific applications became evident.

1.2 Synchronous Motors

Synchronous Motors are a variety of AC motors in which the rotation of the shaft is exactly synchronized with the frequency of the supplied current. Unlike the Induction motor which relies on current induction and hence rotates slightly slower than the input current frequency, a synchronous motor always rotates at the supply frequency. The fact that their speed is independent of the load over the operating range makes them preferable in operations where precise speed or position control is required. One of the most popular applications of this property is in power generation where the machine works as a generator rather than a motor. These machines can also work at leading or unity power factor and thus provide power factor correction. In terms of motoring applications, synchronous machines are a viable alternative to Induction machines as they generally provide better performance for low speed applications and actuation.

1.2.1 Synchronous Reluctance Motor

Figure 1.1: ABB's Synchronous Reluctance Motor[1]

The Synchronous Reluctance Motor (SyRM) is a variety of the Synchronous motor that does not have current flowing in the rotor and as the name implies, the torque in this motor is produced through the phenomenon of magnetic reluctance. In terms of construction, this motor is similar to the Induction motor. The major difference between the two being the rotor.

While an induction machine usually has a squirrel cage rotor or a wound rotor, a synchronous reluctance motor has a cylindrical rotor made of a magnetic material which makes it capable of producing reluctance torque. From Figure 1.1, one can see that the rotor shape is quite different from what one usually finds in an Induction motor and that it does not have any windings.

SyRMs provide high power density at low cost making them ideal for many applications. Although the Permanent Magnet Synchronous Motor (PMSM) has the highest power density among all the motors and is typically the best suited for most motoring applications, it uses rare earth materials. As the name suggests, these elements are rare (and costly) and thus PMSMs may not be a sustainable option in the long run.

The torque density offered by the SyRM is about $\frac{2}{3}^{rd}$ of that offered by the PMSM which makes the SyRM a successful competitor for the PMSM[12]. SyRMs also have a higher efficiency compared to Induction motors and have better thermal properties[4] which makes them a strong competitor for the induction machines.

However, there are drawbacks that SyRMs suffer from. The first is the problem of ripple torque. Since these machines rely on magnetic reluctance for torque generation, they possess a higher torque ripple compared to the Induction motor and thus may not be suitable for torque-sensitive applications. Secondly, the construction of the rotor for the SyRM is rather complicated. As a result these machines are not manufactured as widely as other AC machines. It has also been observed that they have a poorer efficiency and a lower power factor compared to Induction Machine drives when operating below rated conditions[24].

However, with the introduction of FOC and other vector control techniques, these motors are garnering attention once more. Companies like ABB already have their eyes on the SyRM technology and offer a complete drive based solution as one of their products[18]. They have also successfully demonstrated the advantages of this motor in the winch mooring system for M/S Gabriella[2]. Therefore, it would not be long before these motors begin seeing a more frequent industrial and domestic use.

Motor	Advantage	Disadvantage	Typical Application
Brushed DC Motor	Simple speed control Low initial cost	Maintenance (brushes) Medium lifespan Costly commutator and brushes	Steel mills Paper making machines Automotive accessories
Brushless DC Motor	Long lifespan Low maintenance High efficiency	Higher initial cost Requires closed-loop control	Rigid ("hard") disk drives CD/DVD players Electric vehicles RC Vehicles UAVs
Squirrel Cage Induction Motor	Self-starting Low cost Robust Reliable Ratings to 1+ MW Standardized types.	High starting current Lower efficiency due to need for magnetization.	Low-performance applications of all types
Wound Rotor Synchronous Motor	Synchronous speed Inherently more efficient than induction motor	Higher cost	Industrial motors
Synchronous Reluctance Motor	Equivalent to IM except more robust, more efficient, runs cooler, smaller footprint Competes with PM synchronous motor without demagnetization issues	Requires a controller Not widely available High cost	Appliances Electric vehicles Textile mills Aircraft applications
Permanent Magnet Synchronous Motor	High power density High efficiency High power factor	High cost Requires rare earth elements Prone to demagnetization by overheating	Machine tools Industrial drives Electric Vehicles

Table 1.2: Brief Comparison of the major categories of motors[33]

1.3 Motivation and Scope

As mentioned in Section §1.2, the SyRM is one of the best choices for Electric Vehicle (EV) applications. It offers a higher power density and efficiency compared to the Induction Motor and does not require the rare-earth materials that PMSM requires.

Since their invention, SyRMs have been of considerable academic interest and have been explored to a great extent by various researchers. The technology is far from dead as the major conferences still see papers on SyRM and its applications on a regular basis[6, 17, 10]. In the initial years, the focus of the research was mainly on exploring the properties of the motor, followed by building suitable drives for it. In recent years, however, the focus has shifted to designing controllers that also consider the effects of non-linearities which are usually ignored when modeling the motor[13] and improving the control performance using advanced control techniques[14] or neural networks[15]. In terms of sensorless control, the most common techniques involve either flux observers or high frequency signal injection[16]. Some studies have also combined the desirable properties of these two control techniques and introduced hybrid schemes[35] that work for the entire range of operation. There have also been studies published using Kalman Filters[23], which are also the estimation tools used in this project, but they do not appear to be as popular in literature as the former two schemes.

Apart from this, active research is also being carried out on optimizing the rotor design[28, 27, 22] and developing better manufacturing techniques[36, 9] to reduce the cost associated with making the rotor. However, discussion of rotor design and manufacturing techniques is beyond the scope of this project.

1.3.1 Outline of the report

The goal of the project is to explore the control requirements of a synchronous reluctance motor and test the applicability of the extended Kalman Filter as a state estimation technique for sensorless control.

Chapter 2 presents the mathematical model of the system. Section §2.1 includes the dynamic equations that govern the electrical and mechanical behavior of the synchronous reluctance motor. This is followed by a discussion of the Field Oriented Control technique in Section §2.2 which also includes a discussion on the choice of controller gains for the PI controller. Following this, details of the SVPWM technique used in the inverter operation are presented in Section §2.3. Section §2.4 presents the approach to state estimation using the Extended Kalman Filter and the chapter concludes with a discussion on the field weakening operation of the SyRM in Section §2.5.

Chapter 3 presents the motor parameter estimation methods in Section §3.1. Section §3.2 presents the approach for building the system model in Simulink, and discusses the results obtained through the simulation studies.

Chapter 4 involves a discussion of the hardware implementation of the controller. Section §4.1 presents the details of the hardware setup used for testing the controller. This includes a brief description of the major components involved in the hardware implementation. Section §4.2 discusses the programming aspects of the hardware implementation, and includes details of the data capturing and pulse generation processes. The chapter concludes with the discussion of the results obtained using hardware implementation of the controller, in Section §4.3.

CHAPTER 2

MODELING THE SYSTEM

2.1 Motor Model

For a synchronous motor, it is particularly useful to model the motor in the DQ reference frame aligned to the rotor. This is because in a synchronous machine, the DQ frame aligned to the rotor sees the sinusoidal currents flowing in the stator and the stator voltages as constant quantities and thus it becomes easier to build and analyze the model. Of course, using the Park and Clarke transformation and their inverses, one can always transform the relevant quantities - usually voltages and currents - from the DQ frame to the natural ABC frame and vice-versa.

The model can be derived from scratch by obtaining an expression for the self and mutual inductances between the stator phases in the natural frame and then transforming them to the rotor oriented DQ frame. The final model equations are as follows:

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} R_s & -\omega L_q \\ \omega L_d & R_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} L_d \dot{i}_d \\ L_q \dot{i}_q \end{bmatrix} \tag{2.1}$$

The subscript d indicates a component along the d axis, that is, aligned with the axis of the rotor which has the maximum percentage of iron along it, while a subscript q indicates a component along the q axis, that is, leading the d axis by 90° (electrical). The q axis has the minimum iron content along it. A graphical representation of the axes is shown in Figure 2.1

In equation (2.1), v_d, v_q are the instantaneous voltages. R_s is the stator resistance. L_d, L_q are stator inductances, and i_d, i_q are the instantaneous currents flowing into the motor. ω is the instantaneous electrical speed of the motor.

From the equations, one can see that there are only two voltage components involved in the model as opposed to 3 (as in the case of a wound field Synchronous motor) or

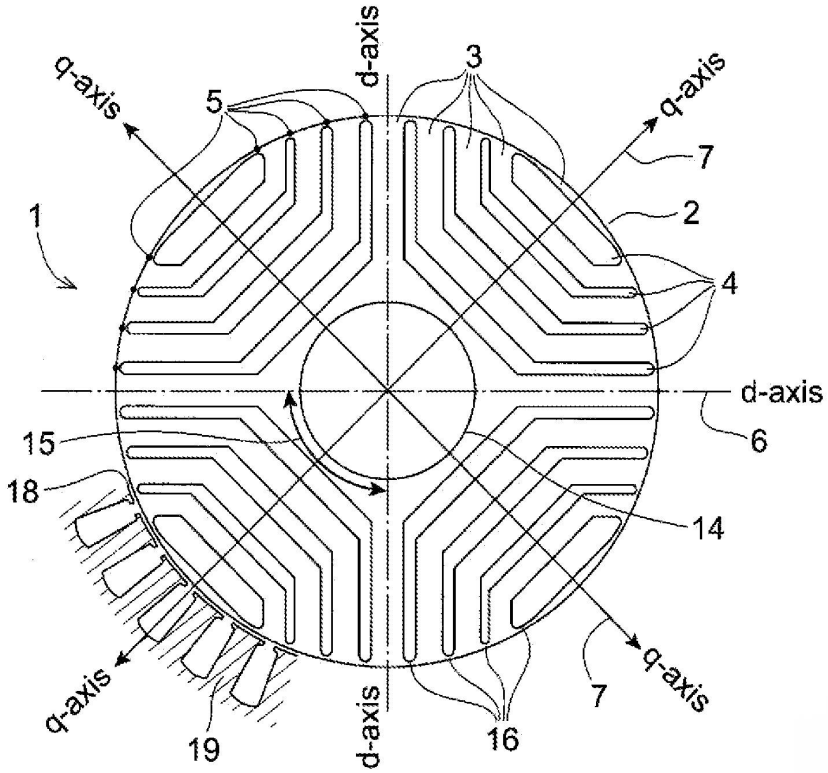


Figure 2.1: Rotor axes of the SyRM[21]

4 (as in the case of a Squirrel Cage Induction motor). A reduced order of the system essentially adds to the ease of building a controller for the machine.

$$\begin{bmatrix} \psi_d \\ \psi_q \end{bmatrix} = \begin{bmatrix} L_d i_d \\ L_q i_q \end{bmatrix} \tag{2.2}$$

ψ_d and ψ_q are the stator flux components. Note that they are only dependent on the inductances and the currents flowing in the respective axes. For this project, we align the D-axis of the DQ frame to the magnetic rotor axis. Due to the geometry of the machine, it turns out that $L_d > L_q$. Of course, since it is only notation, one is free to adopt any other alignment as well. However, the alignment needs to be consistent throughout for getting meaningful results.

$$\tau_g = \frac{3}{2} p \left(L_d - L_q \right) i_d i_q \tag{2.3}$$

τ_g is the generated electromagnetic torque and p represents the number of pole pairs of the rotor. Note that this equation is valid only if one uses the power variant Park

transform. In case of the power invariant transform, the constant $\frac{3}{2}$ would be replaced by 1.

One can see that the generated torque depends on the difference between the values L_d and L_q. In most literature on this topic, a quantity called saliency ratio, given by $\epsilon = \frac{L_d}{L_q}$, is defined. It is easy to see that a higher saliency ratio also results in a higher generated torque if L_q is maintained at the same value, and thus, maximization of the saliency ratio is an important objective in rotor design. While most SyRMs have a saliency ratio of about 6-8, values as high as 21[5] have been reported for an axially laminated anisotropic rotor in unsaturated operation.

Another point worth noting is that the generated torque requires both i_d and i_q to be non-zero to produce a non-zero value. This essentially means that one cannot force i_q or i_d to zero for simplifying the machine control as is done with most other motors like IMs and PMSMs. However, one can hold either i_d or i_q at a fixed reference value and manipulate the other variable to simplify the torque control of the motor.

$$J\dot{\omega_m} = \tau_g - B\omega_m - \tau_L \tag{2.4}$$

The equation above represents the mechanical characteristics of the motor. J is the rotational inertia of the shaft and B is the coefficient of friction. τ_g is the generated electromagnetic torque and τ_L is the load torque applied on the shaft. ω_m is the mechanical speed of the motor and is related to the electrical speed by $\omega = p\omega_m$.

Finally, note that the model does not make any assumptions about the waveforms or the magnitudes of the stator voltages and currents.

2.2 Field Oriented Control

The goal of most motor drives is to ensure that the motor speed is maintained at a user-defined set-point and does not deviate from the set-point under the rated operating conditions.

There are various techniques discussed in literature that achieve this goal. The most common ones are Direct Torque Control (DTC) and Field Oriented Control (FOC). DTC involves estimation of the generated torque and stator flux, and comparison of these quantities with their reference values so as to generate a voltage vector that reduces the error between the estimates and the references. On the other hand, FOC involves resolving the machine currents into a Field component and an armature component and thus implementing a control method similar to that of the DC machine.

For a SyRM, since torque ripple is already a concern, DTC may not be the best solution as it generates a significant torque ripple by design. Instead, the FOC approach is more suited for this machine as it results in simpler equations and is easier to implement using PI controllers.

As required by the FOC method, the dq model used so far resolves current into two components - i_d and i_q - which are independent of each other and can be assumed to be governing the field and torque respectively. Note that i_d governing field is just the convention used for the purpose of this project. One is free to assign i_q as the field current as well.

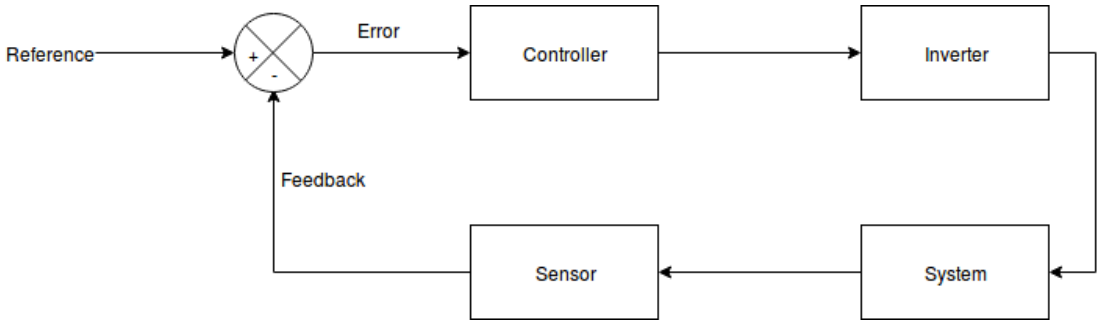


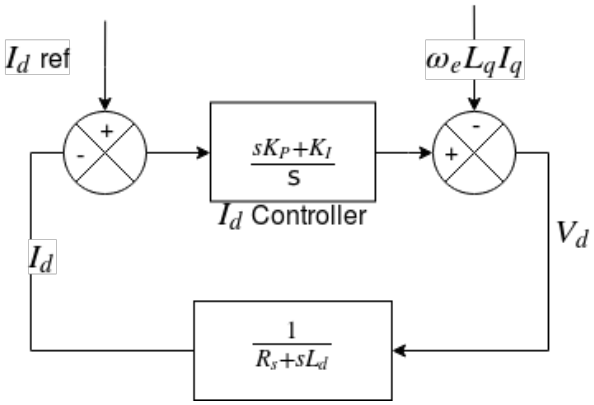
Figure 2.2: Control loop

Figure 2.2 shows a system level diagram of the FOC control loop. The controller used is a PI controller. The inverter can be modeled as a delay of magnitude equaling the switching time period of the inverter. The system response can be determined using

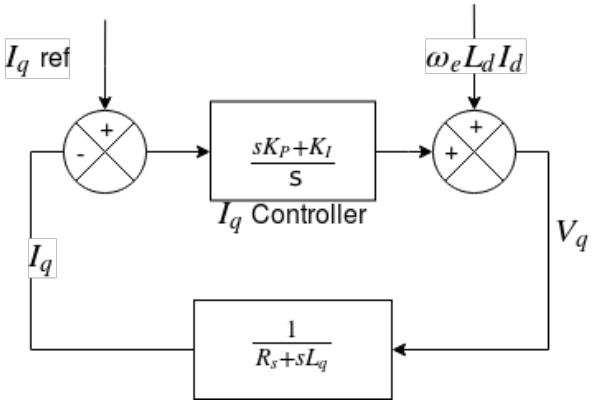
the machine equations in Section §2.1. The sensor can also be modeled as a lag element but is usually treated as a unity gain to simplify the controller design and analysis.

2.2.1 Current controller

The controller design method for the currents i_d and i_q is similar. In both cases, a PI controller is used in order to achieve the set-point and the design aspect involves choosing the K_P and K_I parameters of the controller. To determine these quantities, it is worth looking at the closed loop transfer function of the currents. With the understanding that the two controllers - i_d and i_q - will have similar transfer functions and responses, the analysis is carried out for i_d controller alone and then extended to the i_q controller. The block diagrams for the two controllers are shown in Figure 2.3



(a) D-axis Current control loop



(b) Q-axis Current control loop

Figure 2.3: Current control loops

The closed loop transfer function for i_d, given by $I_D(s)$ as shown in equation (2.5). For arriving at the transfer function, it is assumed that the inverter and sensor transfer functions are simply unit gains. This is not true in general but is a reasonable assumption

if the bandwidth of the controller is at least one decade smaller than the bandwidths of the sensor and inverter. A simplified representation of the i_d control loop is shown in Figure 2.3a

$$\begin{aligned} G(s) &= \frac{sK_P + K_I}{s} \times \frac{1}{sL_d + R_s} \\ I_D(s) = \frac{G(s)}{1 + G(s)} &= \frac{sK_P + K_I}{s(sL_d + R_s) + sK_p + K_I} \end{aligned} \tag{2.5}$$

If one chooses K_p and K_I such that $\frac{K_P}{K_I} = \frac{L_d}{R_S}$, it is easy to see that the transfer function in equation (2.5) reduces to a first order system. One can then tune K_P(or K_I) to achieve the appropriate bandwidth as shown in equation (2.6)

$$\begin{aligned} \frac{1}{1 + \frac{s}{f_{id}}} = \frac{1}{1 + s\frac{L_d}{K_p}} &\implies K_P = f_{id} * L_d \\ \frac{K_P}{K_I} = \frac{L_d}{R_s} &\implies K_I = f_{id} * R_s \end{aligned} \tag{2.6}$$

The drawback of this approach is that it assumes that L_d and R_s would be constant throughout the operation. This need not be true as heating of the stator and the saturation of the stator field can change these quantities, albeit not significantly.

Another possible approach to choose the K_P and K_I values is to employ the properties of a second order transfer function and set ζ to a value of $\frac{1}{\sqrt{2}}$ to achieve a critically damped response. The bandwidth of a critically damped system is equal to its natural frequency ω_n and is chosen to be at least one decade lower than the switching frequency of the inverter. Comparing the coefficients of the transfer function obtained in equation (2.5) with the standard form of a 2^{nd} order system, one can obtain K_P and K_I as shown in equation (2.7)

$$\begin{aligned} \frac{K_I}{L_d} = \omega_n^2 &\implies K_I = L_d\omega_n^2 \\ \frac{K_P + R_s}{L_d} = 2\zeta\omega_n = \sqrt{2}\omega_n &\implies K_P = \left(\sqrt{2}\omega_n L_d - R_s\right) \end{aligned} \tag{2.7}$$

One can notice that the numerator of the closed loop transfer function does not conform to the standard form of a second order system. However, from the properties of the Laplace transform, it isn't difficult to see that it is the scaled derivative of the time response of the standard form of the second order system. The scaling factor being

equal to $\frac{K_P}{L_d}$.

The equations leading to this result are shown in equation (2.8).

$$\begin{aligned}
\frac{s\frac{K_p}{L_d} + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} &= \frac{s\frac{K_p}{L_d}}{s^2 + 2\zeta\omega_n s + \omega_n^2} + \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\
\mathcal{L}^{-1}\left(\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right) &= x(t) \\
\mathcal{L}\left(\frac{d}{dt}x(t)\right) = sX(s) - x(0) \implies \frac{K_P}{L_d}\left(\frac{d}{dt}x(t)\right) &+ x(t)
\end{aligned} \tag{2.8}$$

Due to the additional time derivative term introduced in the time response of the system, which was not accounted for in deriving the equations in equation (2.6), the K_P and K_I values may need to be recomputed if the system does not respond as intended.

Once the K_P and K_I values have been set, the decoupling terms are added to the output of i_d controller to make the i_d response independent of i_q. The final expression after the addition of the decoupling terms gives the input voltage v_d which is supplied to the motor.

The PI controller parameters for the Q-axis current loop can be obtained in the same way as the parameters of the i_d loop. The difference being that the L_d value would need to be replaced with L_q and the decoupling term would be different. The resultant output of the controller would be the v_q voltage that is fed into the motor. For completeness, the controller gain equations for the Q-axis controller are shown in equation (2.9) and equation (2.10) for the pole-zero cancellation and second order approach respectively.

$$\begin{aligned}
\frac{1}{1 + \frac{s}{f_{iq}}} = \frac{1}{1 + s\frac{L_q}{K_p}} &\implies K_P = f_{iq} * L_q \\
\frac{K_P}{K_I} = \frac{L_q}{R_s} &\implies K_I = f_{iq} * R_s
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
\frac{K_I}{L_q} = \omega_n^2 &\implies K_I = L_q\omega_n^2 \\
\frac{K_P + R_s}{L_q} = 2\zeta\omega_n = \sqrt{2}\omega_n &\implies K_P = \left(\sqrt{2}\omega_n L_q - R_s\right)
\end{aligned} \tag{2.10}$$

2.2.2 Speed controller

For speed control, two loops are required. An outer loop for speed control, the output of which is analogous to the motor's generated electromagnetic torque, and an inner loop for current control, which essentially controls the value of i_q.

The open loop transfer function for the speed controller with the PI controller is as shown in equation (2.11). The $I_q(s)$ term in the transfer function corresponds to the closed loop transfer function of I_q controller.

$$G_\omega(s) = \frac{K_I\left(1 + s\frac{K_P}{K_I}\right)}{s} \times \frac{1}{B\left(1 + s\frac{J}{B}\right)} \times I_q(s) \tag{2.11}$$

Employing the fact that the mechanical time constant of the motor is significantly larger than the electrical time constant, one can treat the $I_q(s)$ term as a unity gain if the bandwidth chosen for the speed controller is at least a decade smaller than the bandwidth of the current controller. The closed loop transfer function obtained after treating $I_q(s)$ as unity gain is as shown in equation (2.12).

$$\omega(s) = \frac{G_\omega(s)}{1 + G_\omega(s)} = \frac{sK_P + K_I}{s(sJ + B) + sK_p + K_I} \tag{2.12}$$

Unlike the case with the current controllers, the B and J parameters generally do not vary during the operation of the motor and hence one can equate the ratio of the K_P and K_I terms with the values of J and B to obtain a first order transfer function of required bandwidth(f_ω). The equations leading to K_P and K_I values in this case are as shown in equation (2.13).

$$\begin{aligned} \frac{1}{1 + \frac{s}{f_\omega}} = \frac{1}{1 + s\frac{J}{K_p}} &\implies K_P = f_\omega * J \\ \frac{K_P}{K_I} = \frac{J}{B} &\implies K_I = f_\omega * B \end{aligned} \tag{2.13}$$

Of course, one can still follow the second order transfer function approach described in 2.2.1. Since the form of the transfer function in equation (2.12) is similar to equation (2.5), one can replace L_d with J and R_s with B to arrive at the gains as shown in equation (2.14)

$$\begin{aligned} \frac{K_I}{J} = \omega_n^2 \implies K_I = J\omega_n^2 \\ \frac{K_P + B}{J} = 2\zeta\omega_n = \sqrt{2}\omega_n \implies K_P = \left(\sqrt{2}\omega_n J - B\right) \end{aligned} \tag{2.14}$$

The output of the outer loop of the speed controller is interpreted as generated torque. From equation (2.3), it is easy to see that one can scale this value to obtain the reference for the i_q controller. Although the load torque is not usually known, if one can estimate or measure it, the load torque itself can be added as a feed-forward term to the output of the speed controller before scaling. The advantage offered by addition of the feed-forward term is that it improves the transient response of the system by accounting for the known 'disturbances' and hence aiding the PI controller. The output of the speed controller thus obtained is then used as the reference for the inner PI controller. Figure 2.4 shows a simplified representation of the speed control loop along with the addition of the feed-forward term.

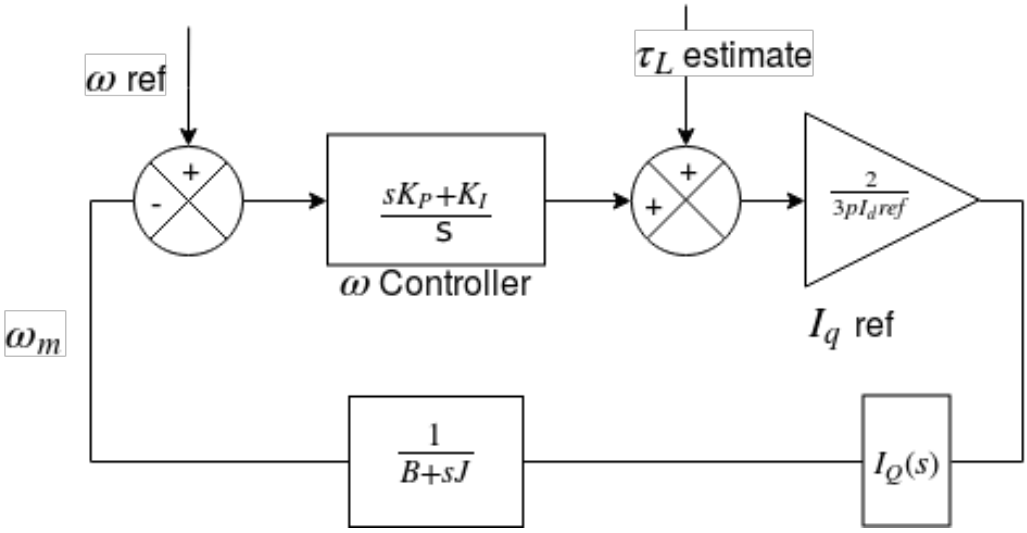


Figure 2.4: Speed control loop

While implementing the PI controller in simulation or in hardware, one also needs to account for the saturation of values. This will be discussed in 3.2.2.

2.3 PWM techniques

Now that the model is established, the next important part involves sending the pulses to generate the required voltages at the machine terminals. This is usually done with the help of a 3-phase voltage source inverter (VSI).

A 3-phase VSI consists of a DC source (called the DC bus) which is usually a capacitor and 6 electrical switches which usually employ IGBTs. IGBTs are preferred over other switches due to the fact that they have the lowest losses among all the switches that operate in the range required for most motor drives - which is a voltage between 600V and 1200V and a current of 10-20A.

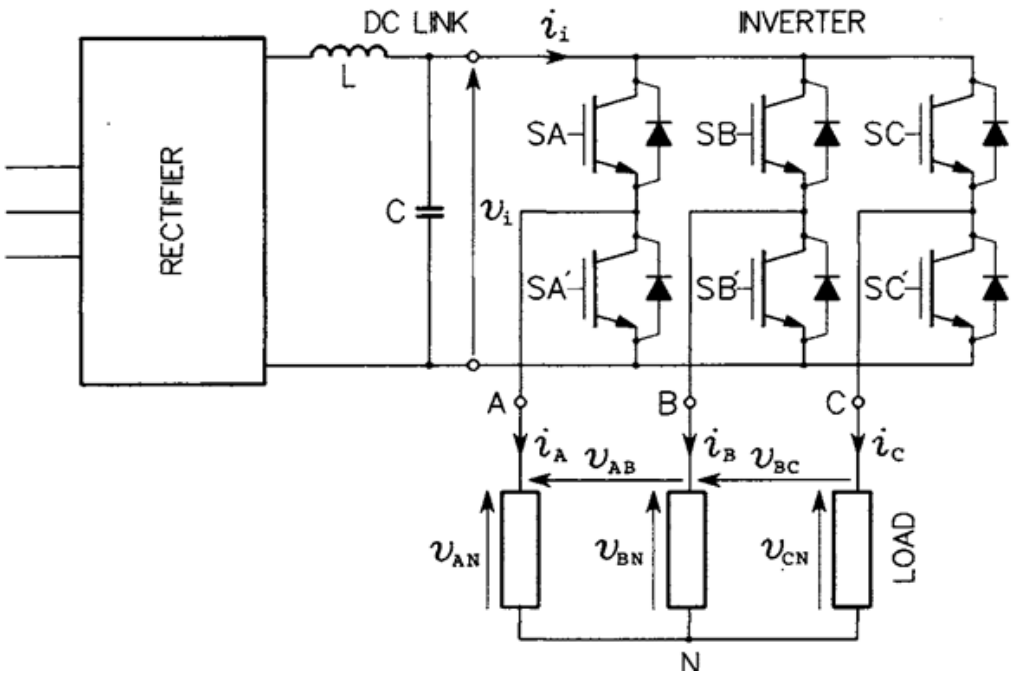


Figure 2.5: Voltage Source Inverter block diagram[32]

The basic principle behind Pulse-width modulation is to use a rectangular pulse whose width can be adjusted (modulated) resulting in the variation of the average value of the output. For a motor drive application, the requirement is to generate a sinusoidal waveform with little to no harmonics and which maximizes the utilization of the DC bus voltage.

Mainly two PWM techniques are used in drives: SPWM (Sinusoidal Pulse Width Modulation) and SVPWM (Space Vector Pulse Width Modulation). Of course, specialized techniques[31, 8, 11] exist which are slight modifications of the above two but

those would not be discussed here.

In SPWM, the reference input is compared with a sawtooth waveform. When the reference value is higher than the value of the sawtooth, the switch is turned ON whereas when the reference value is lower than the value of the sawtooth, the switch is turned OFF. In a 3-phase inverter with 6 switches, the two switches on a given leg have complementary operation and thus when one switch is turned OFF, the other is ON and vice versa. As a result, the pole voltage (that is, the voltage at the midpoint of the leg w.r.t. the voltage at the midpoint of the DC bus) varies between $+\frac{V_{dc}}{2}$ (when the top switch is ON for the full period of the sawtooth) to $-\frac{V_{dc}}{2}$ (when the bottom switch is ON for the full period of the sawtooth).

If the reference voltage, scaled to lie between -1 and 1 is given by V_{ref} and the sawtooth varies between -1 and 1, the pole voltage at a given leg can be obtained by:

$$V_{pole} = V_{ref} \times \frac{V_{dc}}{2} \tag{2.15}$$

The quantity of interest for the motor drive is the phase voltage that appears at the terminals of the motor and it can be shown that this voltage has the value shown below

$$\begin{aligned} V_{an} &= V_{aref}\tfrac{V_{dc}}{3} - V_{bref}\tfrac{V_{dc}}{6} - V_{cref}\tfrac{V_{dc}}{6} \\ V_{bn} &= V_{bref}\tfrac{V_{dc}}{3} - V_{aref}\tfrac{V_{dc}}{6} - V_{cref}\tfrac{V_{dc}}{6} \\ V_{cn} &= V_{cref}\tfrac{V_{dc}}{3} - V_{bref}\tfrac{V_{dc}}{6} - V_{aref}\tfrac{V_{dc}}{6} \end{aligned} \tag{2.16}$$

One can see that in the SPWM technique the maximum magnitude of voltage that can appear at any phase is $\frac{2}{3}rd$ of the DC bus voltage. Can one do better? As it turns out, it is indeed possible to increase the voltage by using the SVPWM technique.

For the 3 phases in a 6-pulse VSI, there are 8 possible switch configurations. The table below shows the voltage vectors associated with each of the switch configurations.

Voltage vector {A, B, C}	V_{ab}	V_{bc}	V_{ca}	V_{an}	V_{bn}	V_{cn}	$V_{\alpha\beta}$
V_0={0, 0, 0}	0	0	0	0	0	0	0
V_1={1, 0, 0}	V_{dc}	0	$-V_{dc}$	$\frac{2V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{j0°}$
V_2={1, 1, 0}	0	V_{dc}	$-V_{dc}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$-\frac{2V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{j60°}$
V_3={0, 1, 0}	$-V_{dc}$	V_{dc}	0	$-\frac{V_{dc}}{3}$	$\frac{2V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{j120°}$
V_4={0, 1, 1}	$-V_{dc}$	0	V_{dc}	$-\frac{2V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{j180°}$
V_5={0, 0, 1}	0	$-V_{dc}$	V_{dc}	$-\frac{V_{dc}}{3}$	$-\frac{V_{dc}}{3}$	$\frac{2V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{-j120°}$
V_6={1, 0, 1}	V_{dc}	$-V_{dc}$	0	$\frac{V_{dc}}{3}$	$-\frac{2V_{dc}}{3}$	$\frac{V_{dc}}{3}$	$\sqrt{\frac{2}{3}}V_{dc}e^{-j60°}$
V_7={1, 1, 1}	0	0	0	0	0	0	0

Note that the magnitude of the voltage vectors in the $\alpha\beta 0$ frame is obtained from the power invariant Clarke transform.

One can see that the Clarke-transformed $\alpha - \beta$ components of the voltages result in Figure 2.6.

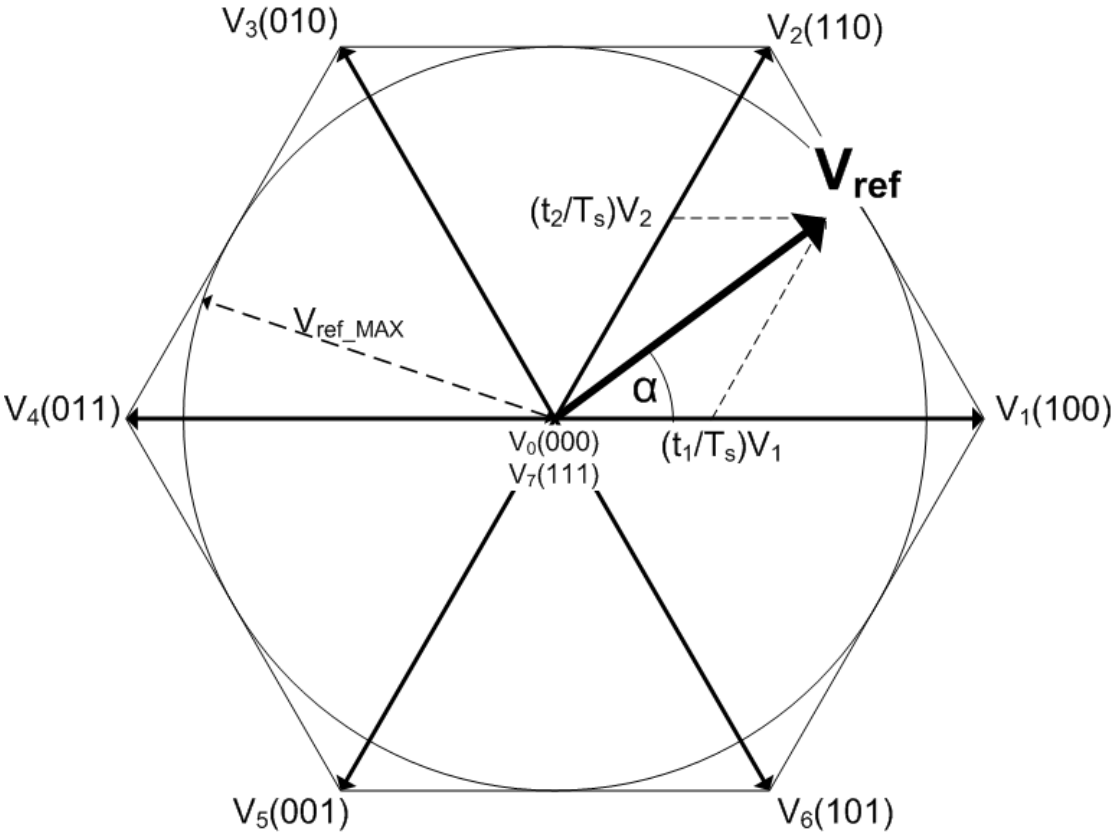


Figure 2.6: Space Vector diagram[20]

The possible space vectors form the vertices of a hexagon and that any voltage of

a magnitude equal to the radius of the in-circle of this hexagon can be obtained by a superposition of the 6 possible space vectors.

Note that the maximum voltage magnitude obtainable in this technique is higher than that obtained using SPWM. However, SVPWM also requires a more sophisticated switching algorithm.

To generate a given space vector, one requires appropriate timing of the zero-vector as well as 2 of the 6 voltage vectors than enclose the given space vector. The formula to obtain the timing is as shown below.

Let T_s be the switching period of the inverter, T_1 and T_2 the duration for which voltage vectors $\overrightarrow{V_1}$ and $\overrightarrow{V_2}$ are generated respectively, and T_0 being the duration for which the zero vector is generated.

If $\overrightarrow{V}$ is the space vector to be generated, we have the following equations:

$$\begin{aligned} \overrightarrow{V} &= \overrightarrow{V_1}T_1 + \overrightarrow{V_2}T_2 \\ T_s &= T_0 + T_1 + T_2 \end{aligned} \tag{2.17}$$

Once the switching durations for the voltage vectors are obtained, one can generate the given voltage vector $\overrightarrow{V}$.

However, in practice, the voltages aren't generated one after the other for the entire duration. Rather, the zero vector (V_0) is generated for $\frac{T_0}{4}$, followed by $\overrightarrow{V_1}$ for $\frac{T_1}{2}$ and $\overrightarrow{V_2}$ for $\frac{T_2}{2}$ and then zero vector (V_7) again for a duration of $\frac{T_0}{4}$. The same cycle is repeated in reverse. As a result, one can reduce the magnitude of the harmonics generated while generating the given voltage vector.

Due to the various advantages[30] offered by the SVPWM technique over SPWM, the former is employed for generating the voltages in this project.

2.4 State Estimation

2.4.1 Introduction

In control theory, a state observer (or estimator) is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. For motor drives, the state observer is particularly useful for implementing sensorless control as it provides accurate estimates of quantities that are not directly measured using sensors.

Various techniques exist for state estimation ranging from something as simple as ordinary least squares to algorithms like Particle filters. For this particular application, the Extended Kalman Filter (EKF) method is employed for estimating the state.

The basic algorithm of a Kalman filter involves two steps. The first step, called the *prediction* step, as the name suggests, involves prediction of the next state. This usually involves building a mathematical model of the dynamics of the system and substituting values of the current state to obtain the prediction for the next state. The second step is called the *correction* step and involves updation of the state estimate obtained from the prediction step based on actual measurements obtained from the system. This step requires a measurement model, which relates the state of the system to the measurements, using mathematical equations. The estimate thus obtained at the end of these two steps is then used as the input for the next time step and the process is repeated.

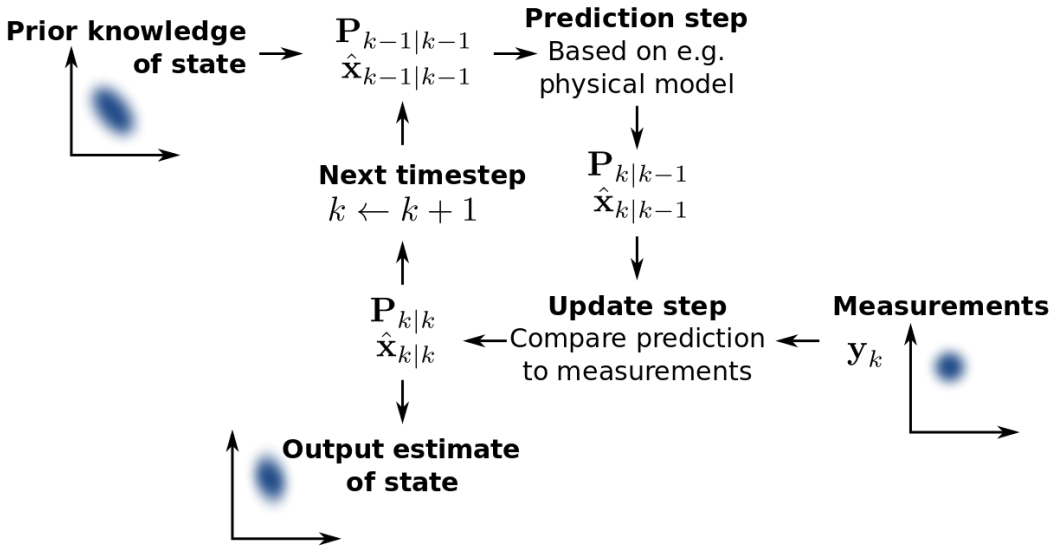


Figure 2.7: A brief explanation of the working of the Kalman Filter algorithm[26]

Since their introduction in early 1960s, Kalman filters have been known to work remarkably well in most applications and a lot of research has been conducted in making minor improvements to the original Kalman Filter, which mainly involved a linear model of the system. One such improvement is the EKF, which takes away the linearity constraint on the system model and introduces a way to make the Kalman Filter algorithm applicable for non-linear systems.

2.4.2 EKF equations

The prediction and correction equations of the EKF for a general system are as follows

Prediction step:

$$\hat{\boldsymbol{x}}_{k+1|k} = f\left(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{u}_k\right) \tag{2.18}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{A}_k \boldsymbol{P}_{k|k} \boldsymbol{A}_k^T + \boldsymbol{Q}_k \tag{2.19}$$

where $\boldsymbol{x}_{k+1} = f\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$, $\boldsymbol{A}_k = \left.\dfrac{\partial f}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{k|k}, \boldsymbol{u}_k}$ and $\boldsymbol{Q}_k$ is the model covariance matrix of appropriate size. $f(x_k, u_k)$ is obtained by discretizing the equations of a continuous time system. The P matrix corresponds to estimation covariance and the smaller the determinant of this matrix, the better the quality of estimate. In general, the prediction step increases the covariance of the P matrix and the measurement step reduces it.

Correction step:

$$\hat{\boldsymbol{y}}_{k+1} = h\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{u}_k\right) \tag{2.20}$$

$$\boldsymbol{S}_k = \boldsymbol{C}_k \boldsymbol{P}_{k+1|k} \boldsymbol{C}_k^T + \boldsymbol{R}_k \tag{2.21}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k+1|k} \boldsymbol{C}_k^T \boldsymbol{S}_k^{-1} \tag{2.22}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_k \left(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1} \right) \tag{2.23}$$

$$\boldsymbol{P}_{k+1|k+1} = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{C}_k \right) \boldsymbol{P}_{k+1|k} \tag{2.24}$$

where $\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{u}_k)$, $\boldsymbol{C}_k = \left. \dfrac{\partial h}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x}_{k+1|k}, \boldsymbol{u}_k}$ and $\boldsymbol{R}_k$ is the measurement covariance matrix. As before, $h(x_k, u_k)$ is obtained by discretizing the model (if required). However, in most cases, the continuous time equations can be used as is for h.

The above equations do not make any assumptions about the model or the states and thus, the implementation of the EKF for the same physical system can vary depending on the states and measurements chosen by the designer.

2.4.3 EKF for SyRM

For this project, a 5x2 EKF model was used with the state and measurement variables as listed below

$$\boldsymbol{x}_k = \begin{bmatrix} i_{d_k} & i_{q_k} & \omega_{m_k} & \theta_k & \tau_{L_k} \end{bmatrix}^T$$

$$\boldsymbol{y}_k = \begin{bmatrix} i_{\alpha_k} & i_{\beta_k} \end{bmatrix}^T$$

$$\boldsymbol{u}_k = \begin{bmatrix} v_{d_k} & v_{q_k} \end{bmatrix}^T$$

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \left(\frac{v_{d_k} - R_s i_{d_k} + p \omega_{m_k} i_{q_k} L_q}{L_d} \right) T_s + i_{d_k} \\ \left(\frac{v_{q_k} - R_s i_{q_k} - p \omega_{m_k} i_{d_k} L_d}{L_q} \right) T_s + i_{q_k} \\ \left(\frac{1.5p(L_d - L_q) i_{d_k} i_{q_k} - B\omega_{m_k} - \tau_L}{J} \right) T_s + \omega_{m_k} \\ T_s p \omega_{m_k} + \theta_k \\ \tau_{L_k} \end{bmatrix}$$

$$\boldsymbol{y}_k = \begin{bmatrix} \cos(\theta_k) i_{d_k} - \sin(\theta_k) i_{q_k} \\ \sin(\theta_k) i_{d_k} + \cos(\theta_k) i_{q_k} \end{bmatrix}$$

$$\boldsymbol{A}_k = \boldsymbol{I} + T_s \begin{bmatrix} \frac{-R_s}{L_d} & \frac{p\omega_{m_k}}{L_d} & \frac{p i_{q_k}}{L_d} & 0 & 0 \\ -\frac{p\omega_{m_k}}{L_q} & \frac{-R_s}{L_q} & -\frac{p i_{d_k}}{L_q} & 0 & 0 \\ \frac{1.5p(L_d - L_q) i_{q_k}}{J} & \frac{1.5p(L_d - L_q) i_{d_k}}{J} & \frac{-B}{J} & 0 & \frac{-1}{J} \\ 0 & 0 & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{C}_k = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) & 0 & -\sin(\theta_k) i_{d_k} - \cos(\theta_k) i_{q_k} & 0 \\ \sin(\theta_k) & \cos(\theta_k) & 0 & \cos(\theta_k) i_{d_k} - \sin(\theta_k) i_{q_k} & 0 \end{bmatrix}$$

R_k and Q_k are diagonal matrices of appropriate dimensions which need to be tuned appropriately depending on the motor parameters. T_s is the time step. This EKF model estimates 5 variables with the measurements of only 2.

In the equations above, i_α and i_β, the measured currents are obtained by performing a Clarke's transform on i_a, i_b and i_c .

2.5 Field Weakening operation

In variable frequency drives, field weakening performance is of great interest as it offers higher speeds without the requirement of a higher inverter rating. The downside to it, however, is that the maximum torque that can be produced at higher speeds is lower than the rated torque of the machine. Similar to the operation of a DC motor, where reduction in field current leads to an increase in speed, even in the case of SyRM, reduction in the direct (or quadrature) axis reference enables the motor to achieve speeds beyond the rated speed.

Before moving further, the system limits are defined mathematically.

$$v_d^2 + v_q^2 \leq V_s^2 \tag{2.25}$$

The equation above corresponds to the inverter limits and V_s in this case is the

maximum voltage obtainable from the inverter. When using SVPWM and the power variant transforms, $V_s = \frac{V_{dc}}{\sqrt{3}}$.

$$i_d^2 + i_q^2 \leq I_s^2 \tag{2.26}$$

The equation above represents the motor limits and puts an upper limit on the magnitude of current that can be supplied to the motor. This value is also equal to the rated current of the motor. However, for short intervals, typically representative of the controller bandwidths, the current supplied to the motor can exceed the rated current without actually damaging the motor.

Of course, the machine equations listed in Section §2.1 still hold and govern the value of the reference current for speeds beyond the rated speed.

2.5.1 Modes of operation

From equation (2.1), assuming steady state and negligible resistive drop, results in the following

$$v_d = p\omega_m L_q i_q, v_q = p\omega_m L_d i_d$$

and substituting these values in equation (2.25) gives

$$p^2\omega_m^2 L_q^2 i_q^2 + p^2\omega_m^2 L_d^2 i_d^2 \leq V_s^2 \tag{2.27}$$

In the equations above, ω_m corresponds to the instantaneous mechanical speed of the motor.

It is easy to see that while equation (2.26) is the equation of a region enclosed in a circle, equation (2.27) is the equation of a region enclosed in an ellipse whose major and minor axes vary with speed.

Note that as the speed rises, the current limit circle remains unchanged while the area under the voltage limit circle shrinks. As a result, three distinct modes of operation can be identified as shown in Figure 2.8.

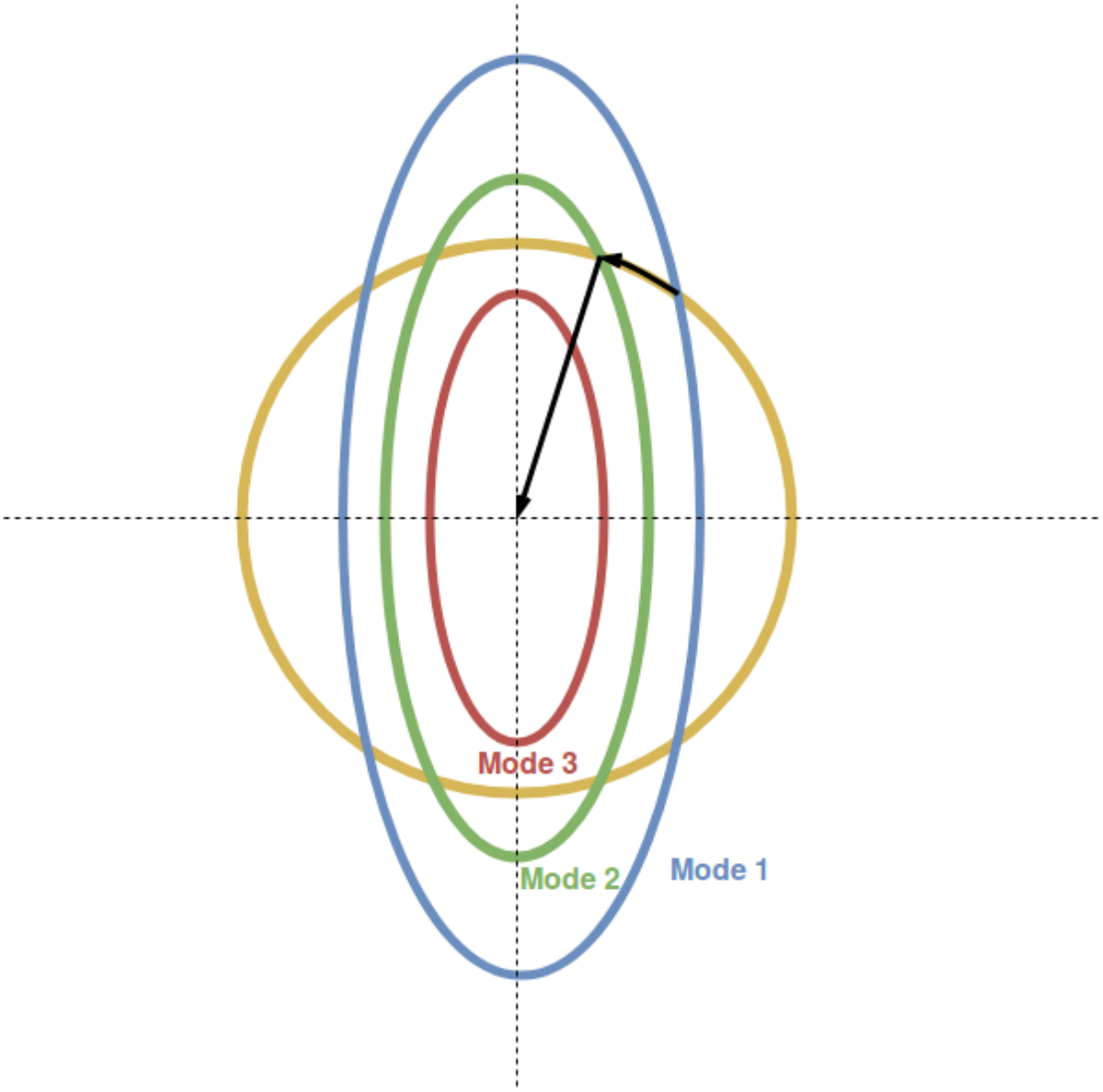


Figure 2.8: Flux weakening modes of operation

Mode 1 - Maximum Torque mode [$0 \leq |\omega_m| \leq \omega_{rated}$**]** In this mode, both, the motor as well as the inverter operate within their rated conditions and the reference field current is set to its rated value which was obtained using the MATLAB code in Section §A.1. The motor speed is within its rated value and the rated torque can be generated at all speeds in this range. This is also the region in which the motor can generate the rated output power. The desired d-axis current in this region is constant and is the same as the rated current for that axis.

$$I_{d_{desired}} = I_{d_{rated}} \tag{2.28}$$

$$\tau_{max} = \tau_{rated} \tag{2.29}$$

Mode 2 - Current limit mode [$\omega_{rated} \leq |\omega_m| \leq \omega_{climit}$] As the speed rises beyond the rated speed, the ellipse corresponding to the voltage limit starts contracting in size. The operating point, therefore, is obtained from the intersection of the current limit circle and the voltage limit ellipse. Since the operating point is dependent on the speed, the desired d-axis current is also a function of the reference speed in this mode. Since the motor is no longer operating at the rated condition, the maximum torque that can be obtained is also a function of the reference speed.

The operating point can be derived as follows

$$p^2\omega_m^2 L_q^2 i_q^2 + p^2\omega_m^2 L_d^2 i_d^2 = V_s^2$$

$$i_q^2 + i_d^2 = I_s^2$$

$$p^2\omega_m^2 \left(L_d^2 - L_q^2\right) i_d^2 = V_s^2 - p^2\omega_m^2 L_q^2 I_s^2$$

$$\implies I_{d_{desired}} = \sqrt{\frac{V_s^2 - p^2\omega_m^2 L_q^2 I_s^2}{p^2\omega_m^2 \left(L_d^2 - L_q^2\right)}} \tag{2.30}$$

$$\tau_{max} = \frac{3}{2} p(L_d - L_q) I_{d_{desired}} \times \sqrt{I_s^2 - I_{d_{desired}}^2} \tag{2.31}$$

Note that in the expressions above, ω_m is the instantaneous mechanical speed. Also note that $I_{d_{desired}}$ does not exhibit a discontinuity when the mode of operation changes at ω_{rated}.

As the speed continues to increase, the voltage limit ellipse shrinks and at some point, it becomes completely enclosed in the current limit circle. The speed at which this happens is denoted by ω_{climit} and at this speed, the mode of operation changes.

Mode 3 - Inverter limit mode [$\omega_{climit} \leq |\omega| \leq \omega_{mlimit}$] In this mode, the voltage limit ellipse is completely enclosed within the current limit circle. The operating point in this region is obtained by finding the point where the voltage limit ellipse and the torque curve are tangential to each other. This also means that for a given speed, the torque is generated with the minimum current.

It can be shown that the condition mentioned above corresponds to $L_d i_d = L_q i_q$

Substituting this in the equation for the voltage limit gives

$$2p^2\omega_m^2 L_d^2 i_d^2 = V_s^2$$

$$\implies I_{d_{desired}} = \frac{V_s}{\sqrt{2}p\omega_m L_d} \tag{2.32}$$

As before, the maximum possible torque can be found from equation (2.31)

Noting that the quantity $I_{d_{desired}}$should be continuous even when the mode changes, we can obtain ω_{climit} by equating the reference current equation for mode 2 and mode 3 and solving for ω_{climit}

$$\frac{V_s}{\sqrt{2}p\omega_{climit}L_d} = \sqrt{\frac{V_s^2 - p^2\omega_{climit}^2 L_q^2 I_s^2}{p^2\omega_{climit}^2\left(L_d^2 - L_q^2\right)}}$$

$$\frac{V_s^2}{2L_d^2} = \frac{V_s^2 - p^2\omega_{climit}^2 L_q^2 I_s^2}{\left(L_d^2 - L_q^2\right)}$$

$$2p^2\omega_{climit}^2 L_d^2 L_q^2 I_s^2 = V_s^2\left(L_d^2 + L_q^2\right)$$

$$\omega_{climit} = \frac{V_s}{pL_d L_q I_s}\sqrt{\frac{\left(L_d^2 + L_q^2\right)}{2}} \tag{2.33}$$

As the reference speed of the motor increases, the torque producing capability of the motor goes down. At the same time, the windage losses of the motor grow. Thus, at a particular speed, say ω_{mlimit}, the generated torque will no longer be capable of

overcoming the mechanical losses and the motor would be unable to accelerate further. It is also possible that the value of ω_{mlimit} can be smaller than ω_{climit}, in which case, the motor would never enter the 3^{rd} mode of operation. In practice, the value of ω_{mlimit} would be lower than the one calculated theoretically due to the saturation of the motor, resistive losses which were neglected while arriving at the equations above, and inverter losses.

2.6 Summary

This chapter presents the theory behind the design of the SyRM motor drive.

Section §2.1 introduces the motor model and the relevant electrical and mechanical equations that govern the motor operation.

Section §2.2 involves a discussion of the motor control loop. The closed loop transfer function equations are presented and the PI controller parameters are derived.

Section §2.3 focuses on the PWM techniques used for voltage generation and introduces the theory behind the SVPWM technique.

Section §2.4 presents the Extended Kalman Filter technique and discusses the way it can be used for sensorless control of the SyRM.

Section §2.5 discusses the field weakening operation of the SyRM and presents the equations involved in each region of operation.

With the theory of the SyRM drive presented, Chapter 3 delves into the simulation studies performed on the SyRM.

CHAPTER 3

PARAMETER ESTIMATION AND SIMULATION STUDIES

3.1 Estimating the Motor Parameters

The main component of this project, the motor, was manufactured by Mark Elektriks. The nameplate details of the motor are listed in Table 3.1

Parameter	Value
Rated power	2KW
Rated speed	1500rpm
Rated Voltage	415V
Rated Current	7.32A
Number of poles	4
Frame	132S

Table 3.1: Motor nameplate details

The motor supports star and delta connections and the results in this report were obtained from the star connected configuration.

Unfortunately, the motor nameplate data does not provide information about the internal parameters of the motor which are crucial for ensuring that the control loop works well. These parameters include the stator resistance R_s, the inductances L_d and L_q, the moment of inertia J and the windage coefficient B

3.1.1 Measuring the electrical parameters

The electrical parameters of the motor include R_s, L_d and L_q.

R_s can be measured using a multimeter. Since this motor supports both star and delta connections, 6 terminals are available for measurements. R_s can be obtained by measuring resistance between any two terminals corresponding to the same phase.

Inductance measurement can be done using an LCR meter or by plotting the rise or fall of the current on supplying a step input. Using an LCR meter, one can measure the inductance between two phases for various angles and by obtaining the maximum and minimum values of the inductance readings, one can estimate the values of L_d and L_q respectively. In fact, $L_d = \frac{L_{AB}max}{2}$ and $L_q = \frac{L_{AB}min}{2}$. In the case of this machine, however, using the LCR meter led to unreliable results as the difference between maximum and minimum values was negligible - suggesting that $L_d = L_q$.

The current rise method involves exciting the motor phases with a DC supply and fitting a curve of the form $I_0(1 - e^{-\frac{(t-t_0)R}{L}})$ to estimate the inductance of the resulting circuit. On exciting the motor with a DC voltage, the rotor orients itself in the direction of the magnetic field. As a result, the equivalent inductance seen by the voltage source terminals is equal to L_d and as long as the rotor position does not change, the time constant is determined by the ratio of L_d and the equivalent resistance. In this case, the current rise test was performed by shorting the B and C phases of the motor and applying a DC voltage between the A and B phases. The equivalent resistance of this configuration was $\frac{3R_s}{2}$. The current rise was plotted on an oscilloscope and the inductance was obtained by using the *cftool* utility in MATLAB. The results of this exercise are shown in Figure 3.1.

For measuring L_q using this method, one needs to clamp the rotor at a position that is electrically 90° away from the L_d position. However, due to the absence of a clamping mechanism in the workspace, L_q was estimated from the motor equations using the rated operation test data supplied by the manufacturer [See Section §A.2].

The power invariant Park transform was used for transforming the abc reference frame variables to the $dq0$ reference frame and solving equation (3.1) using MATLAB's symbolic solver, the value of L_q was obtained.

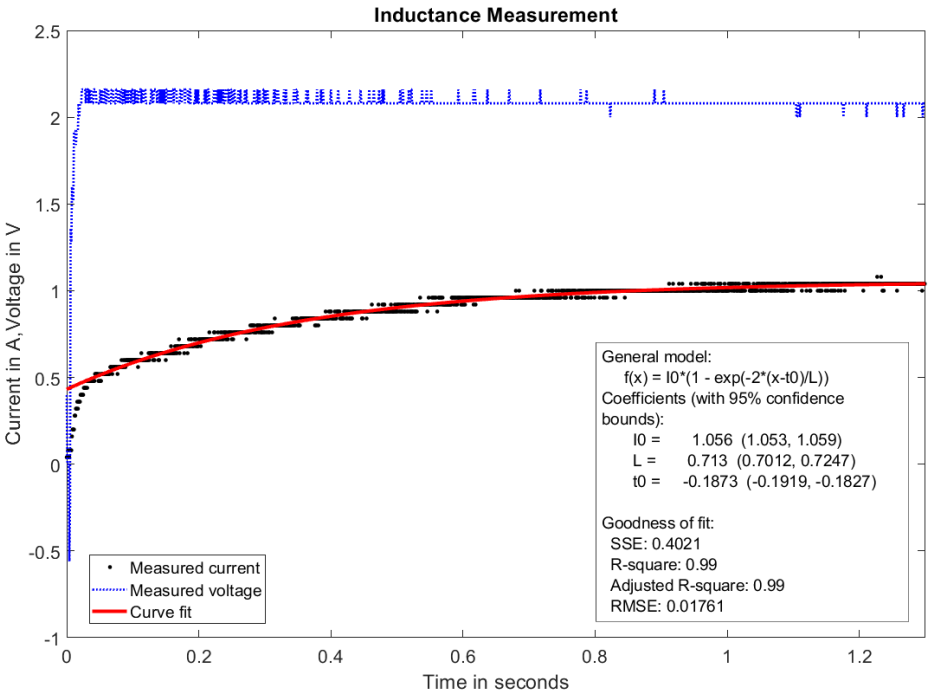


Figure 3.1: L_d estimation

$$
\begin{aligned}
v_{d\_rated} &= R_s i_{d\_rated} - \omega_{e\_rated} L_q i_{q\_rated} \\
v_{q\_rated} &= R_s i_{q\_rated} + \omega_{e\_rated} L_d i_{d\_rated} \\
v_{q\_rated}^2 + v_{d\_rated}^2 &= V_{line\_rms}^2 \\
i_{q\_rated}^2 + i_{d\_rated}^2 &= 3I_{line\_rms}^2 \\
v_{q\_rated} i_{q\_rated} + v_{d\_rated} i_{d\_rated} &= P_{rated}
\end{aligned}
\tag{3.1}
$$

3.1.2 Measuring the mechanical parameters

The mechanical parameters of the motor are B and J .

With the electrical parameters derived, the windage loss coefficient B was estimated using the solution of equation (3.1) in equation (3.2).

$$
\tau_{rated} = p(L_d - L_q) i_{d\_rated} i_{q\_rated} - B\omega_{e\_rated} \tag{3.2}
$$

The value of the rotor's moment of inertia, J was estimated by first finding out the mechanical time constant of the motor and then substituting the value of B in it. The

mechanical time constant was obtained by plotting the rotor speed as a function of time and fitting an exponential decay curve to it using MATLAB's *cftool* utility. The result of the curve fitting exercise is shown in Figure 3.2

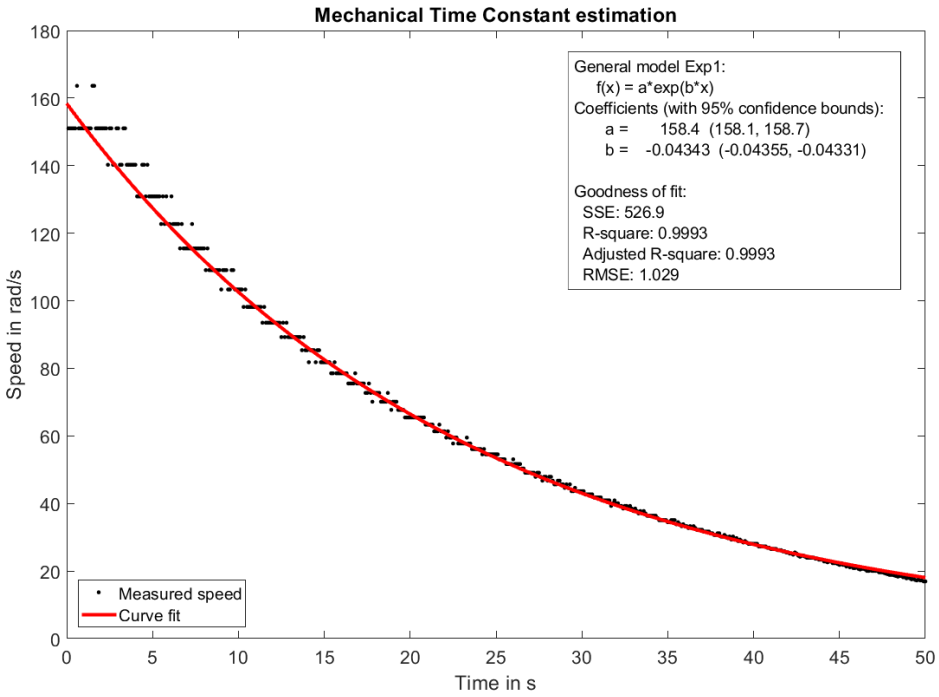


Figure 3.2: Mechanical Time constant estimation

3.1.3 Estimated motor parameters

The parameters obtained from the exercise carried out in 3.1.1 and 3.1.2 are listed in Table 3.2

Parameter	Value
R_s	1.3Ω
L_d	0.713 H
L_q	0.09 H
B	0.00675 Nms
$\frac{B}{J}$	0.04343 s^{-1}
J	0.1554kgm^2

Table 3.2: Motor parameters

The MATLAB solver also computed the value of i_d at rated operation - 0.876A - and this value of i_d was used as the fixed reference for the I_D controller.

From the estimated parameters, one can see that the saliency ratio of the motor is close to 8. The values of the rotor inertia and the mechanical time constant, however, are on the higher side and even physically, one can see that the motor is extremely bulky for its rating.

3.2 Simulations

Before implementing the drive on an actual SyRM, the operation of drive is simulated in Simulink to verify its functionality.

3.2.1 System model

The system model and its components are as shown in Figure 3.3

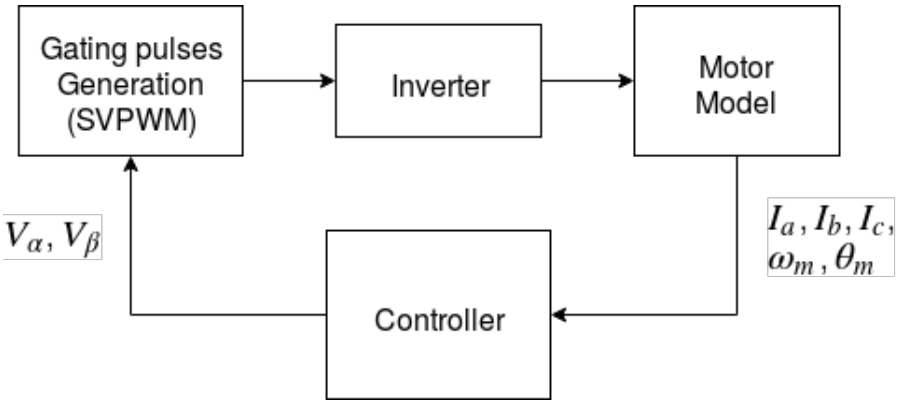
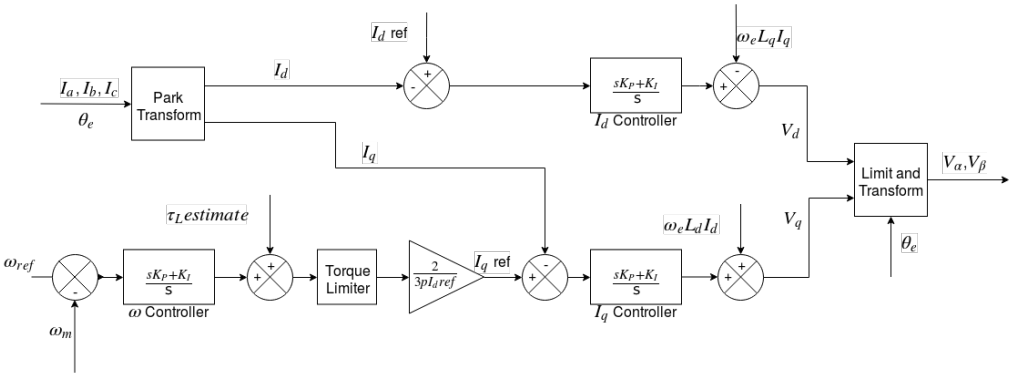


Figure 3.3: Top-level model

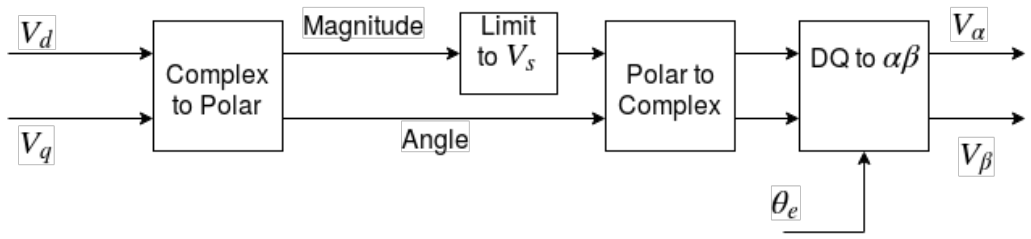
Simulink provides a module for simulating the PMSM. Noting that the SyRM equations are the same PMSM's when the magnet linked flux is set to 0, the PMSM module is used to simulate the SyRM response.

The inverter control block is responsible for generation of the gating pulses with the reference voltage as the input. Simulink provides a module for simulating the inverter response with the gating signals as input and this module is used to power the motor.

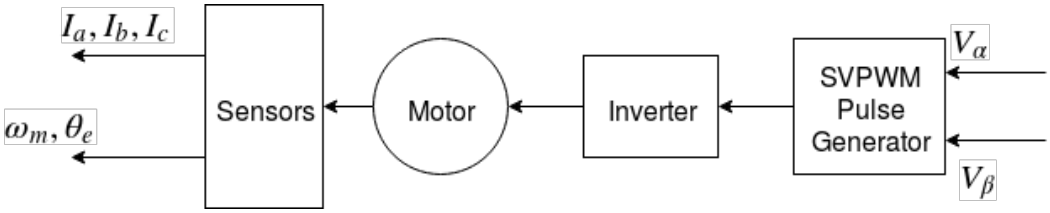
The controller block handles the control logic which takes in the stator currents, electrical angle, mechanical speed, reference speed and reference current i_d as the input and generates the reference voltages which are then fed into the limit and transform



(a) Controller block model



(b) Limiter model



(c) Inverter and sensor model

Figure 3.4: Flow of Control in simulation

block and finally into the SVPWM generator. The SVPWM generator takes in the $\alpha - \beta$ transformed voltage and produced the gating pulses that drive the inverter and the motor. The sensors capture the stator currents, electrical angle and the mechanical speed and feed it into the controller, thus closing the loop. Figure 3.4 depicts the flow of control graphically.

The i_d and i_q controllers, as explained in 2.2.1, consist of a PI controller followed by addition of the decoupling term. Saturation was enabled in the PI block to ensure that the integrator does not attain absurd values and the saturation limits were set to be equal to the maximum voltage that the inverter could generate. The PI gains were chosen using the second order design method and the natural frequency was set to 100Hz, which was about 2 decades away from the 10kHz operating frequency of the inverter.

The bode plots of the closed loop controllers thus obtained are shown in Figure 3.5. From the plots one can see that the controllers have a similar magnitude and phase

response.

The limiting factor in choosing the bandwidth of the speed controller is the mechanical time constant. For the SyRM in the lab, the mechanical time constant is about 25s. This value is rather large and hints that the motor would have a sluggish response. Since the motor would take a long time to settle at its value, using a high bandwidth would only lead the controller into saturation. Therefore, a bandwidth of 0.2Hz was chosen and the PI gains were computed using the pole zero cancellation approach.

The resultant bode plot is shown in Figure 3.6. As expected, the -3dB bandwidth in the plot corresponds to the frequency of just over 1 rad/s. The downside of this low bandwidth is that the controller has a settling time of 3.11s for reaching 90% of its final value. This is extremely slow for a controller but with the given mechanical parameters, it's much better than what the performance would be like without the controller.

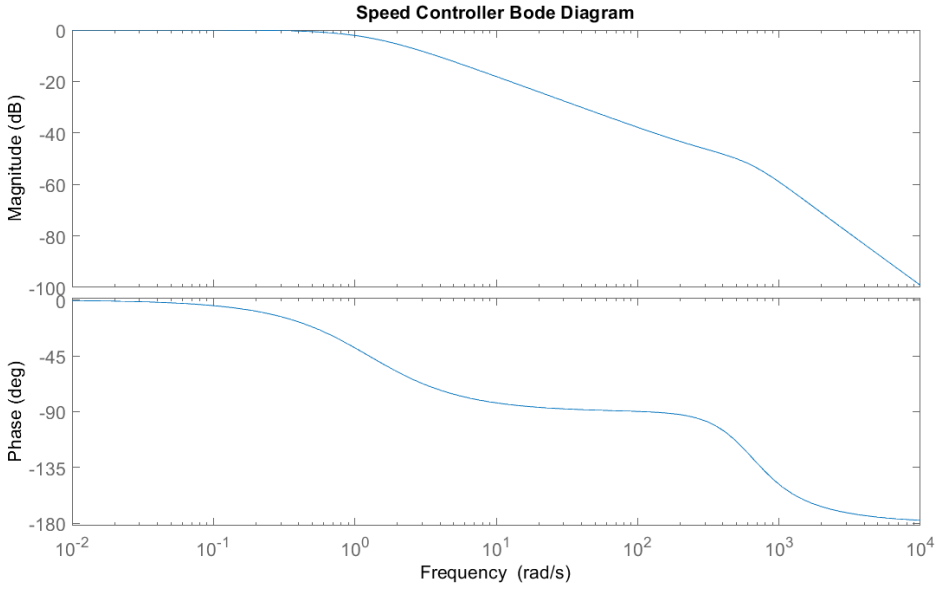
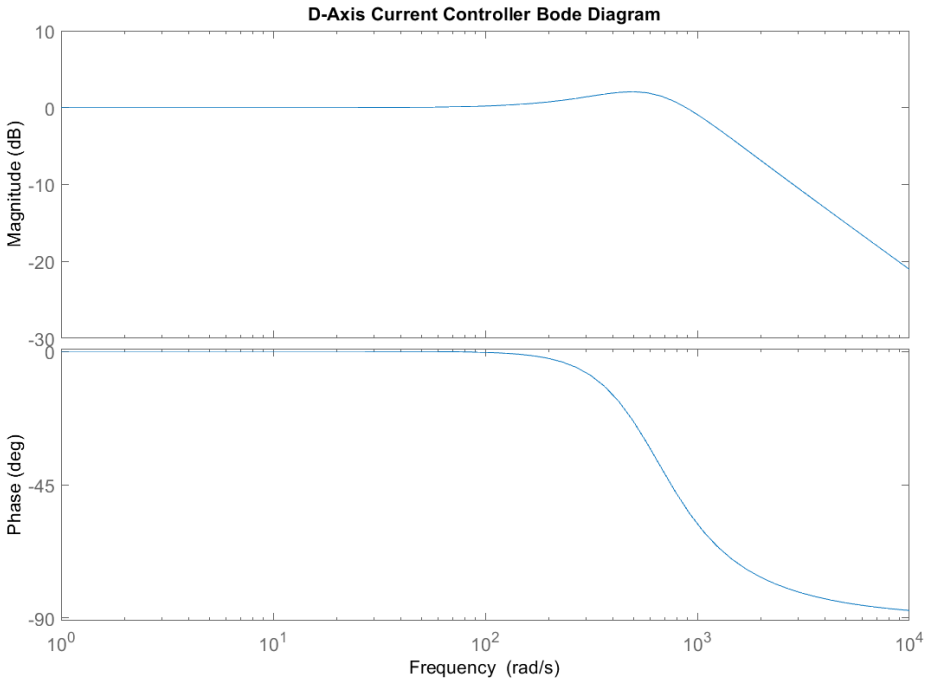


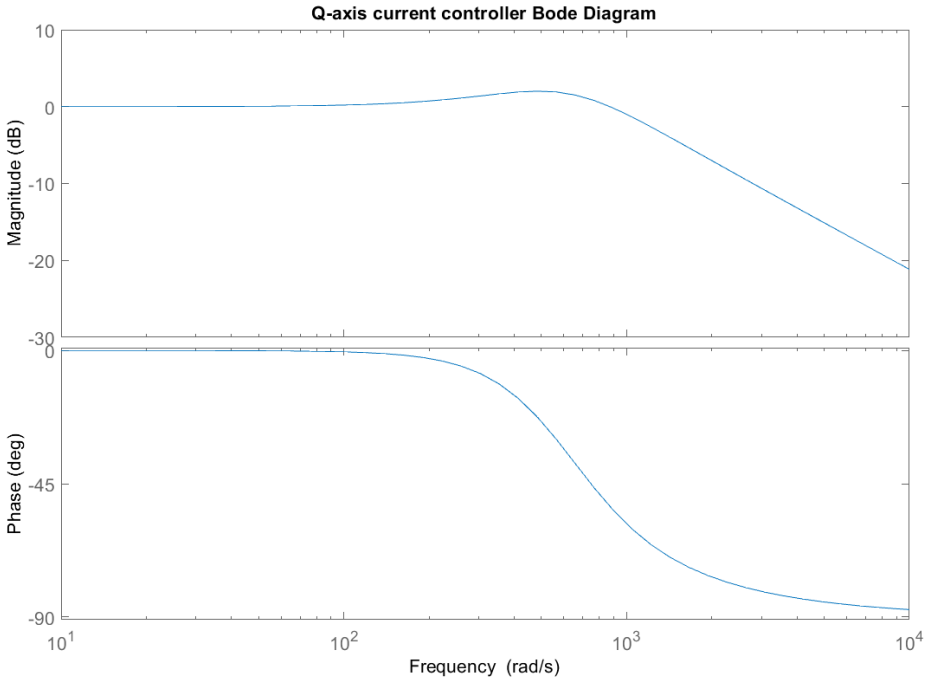
Figure 3.6: Bode plot of ω controller

The feed-forward load torque term was only added to the output when the EKF was used for state estimation, as it involved estimation of the load torque. For the simulations that involved FOC using sensors, the feed-forward term was not added.

Simulink contains a module to generate the SVPWM gating pulses from the V_α and V_β components with the requirement being that the reference voltage magnitude is less than 1. The reference signal generated in the controller block (in $dq0$ reference frame) was converted to $\alpha\beta0$ reference frame and then scaled down so as to meet the magnitude



(a) Bode plot of I_D controller



(b) Bode plot of I_Q controller

Figure 3.5: Bode plots

requirement.

For sensorless control, the states were estimated by the EKF which was implemented as shown below.

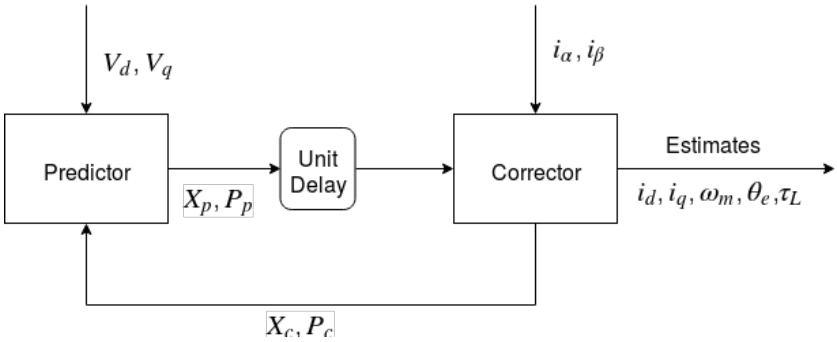


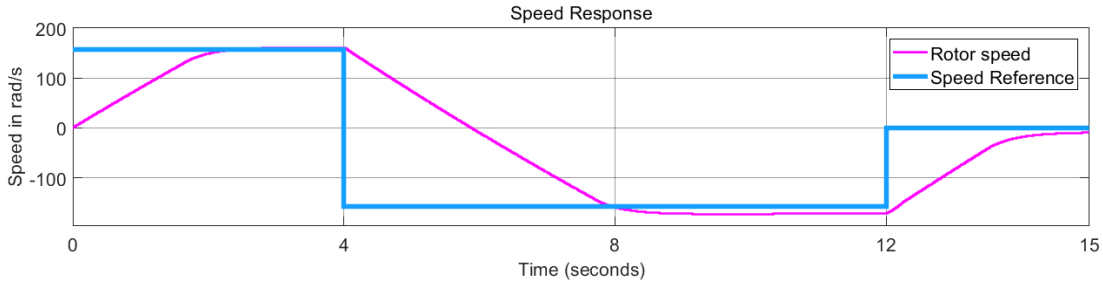
Figure 3.7: EKF state estimator model

3.2.2 Simulation results

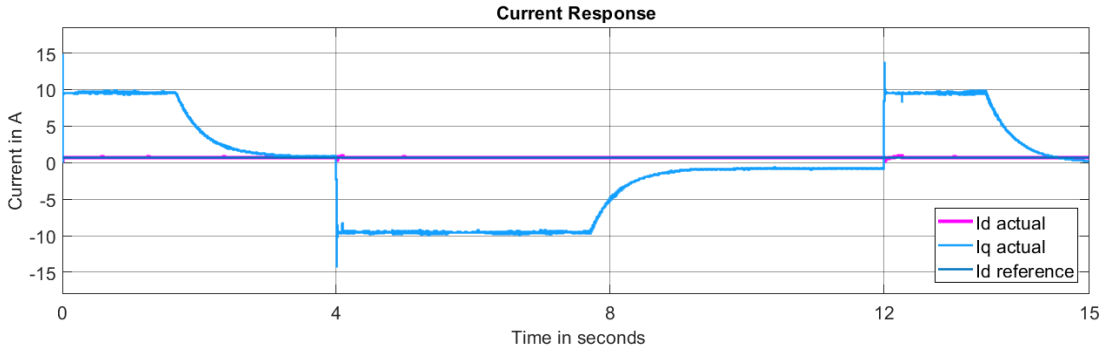
3.2.2.1 FOC simulation

The FOC simulations were performed using a discrete solver of step time $1e^{-5}s$. Two separate simulations were performed, one for demonstrating the motor starting, speed reversal and braking operation at no load condition, and the other for demonstrating the behavior under loading.

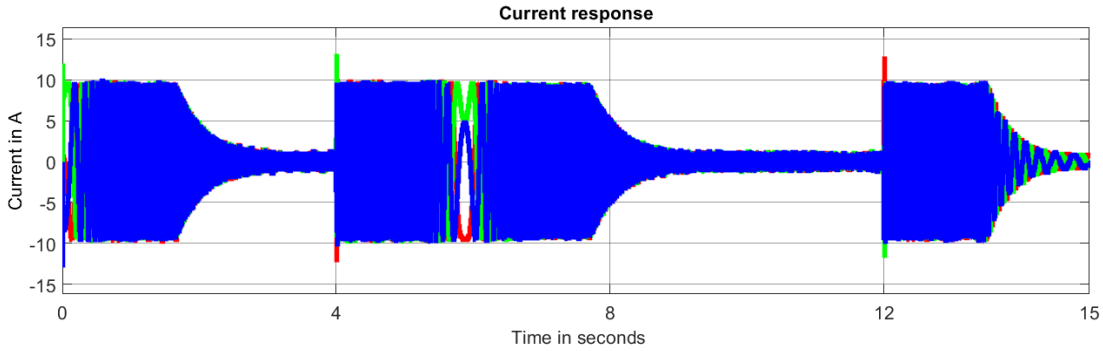
No Load simulation For the purpose of the no-load simulation, the load torque was kept at 0 at all times and the speed reference was given to be positive rated speed followed by negative rated speed and then 0. These references would facilitate the demonstration of the starting, speed reversal and braking operations of the motor. The simulation results are shown in Figure 3.9.



(a) Speed response



(b) DQ axis Current response



(c) 3 phase Current response

Figure 3.8: Operation under no load

During the starting operation, the rotor speed has a linear rise and the i_q current reference is held constant. This is due to the fact that the speed controller output is in saturation and the rated torque is being generated by the machine. There is a sharp peak in the 3 phase current waveform as soon as the motor starts which exceeds the current limit of the motor. However, the effect of this overshoot are only temporary and the current soon attains a value within the rated limit. As the error in speed decreases, the speed control gets out of the saturation and enters the control region. At that point, the i_q current starts falling and the rise in rotor speed is no longer linear. Even the magnitude of the 3 phase currents starts falling at the same rate as i_q. Once the speed

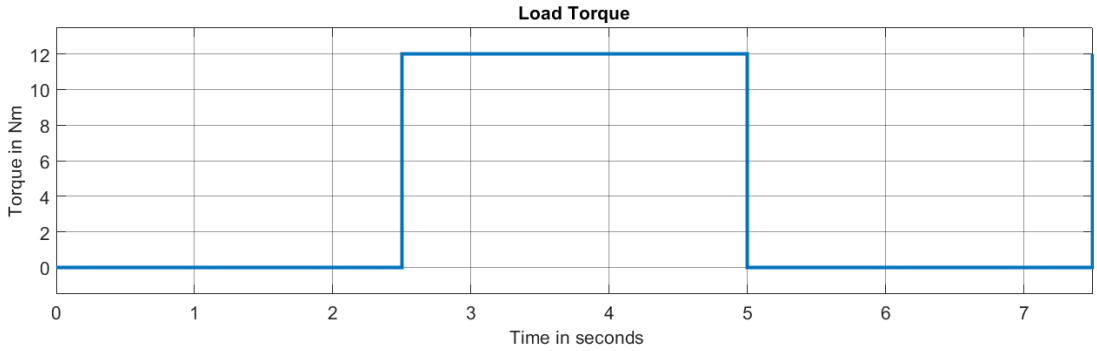
comes very close to the set-point, the value of i_q becomes almost constant and just enough torque is produced to overcome the windage losses. Due to the low bandwidth of the speed controller, the speed does not immediately settle down to the set-point and takes a considerably long duration(over 20s) to settle down at the reference. During this process, the value of i_d remains unperturbed from its reference.

At 4s, there is a step change in the speed reference and a corresponding step change in the i_q value. Once again, there is a sharp peak observed in the 3 phase stator currents which overshoots the ratings. One can notice that the i_q magnitude briefly overshoots the rated limit and that the i_d value also shows a change at that point. This is because the step change in speed temporarily sends the inverter into saturation and the limiter action affects the voltage generation process in such a way that both i_d and i_q are no longer in the control ranges of their respective controllers. At about 6s, one can see that the 3 phase currents switch the sequence from a,b,c(or R,G,B in this case) to a,c,b(or R,B,G in this case). This change is observed because the motor reverses the direction of rotation at that point.

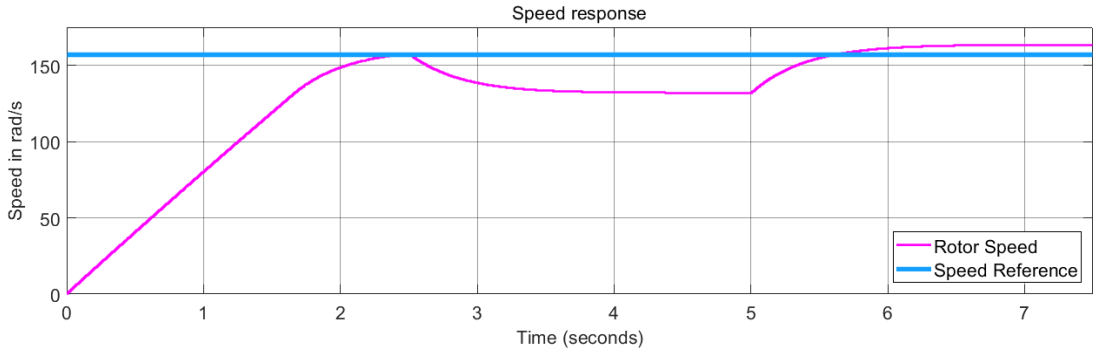
After this, until close to 8s, the negative rated torque is generated and the machine reverses its speed completely. Near 8s, once the speed error makes the controller enter the control region, the i_q value begins to rise and the magnitude of speed, which has overshot the reference, begins to fall. However, the rate at which the magnitude of speed decreases is extremely low due to the low bandwidth of the controller and hence the speed does not achieve its reference within the simulation duration.

At 12, a phenomenon similar to what was observed at 4s is observed. The value of i_q changes abruptly and i_d as well shows a visible change. Similarly, the 3 phase currents also display the sharp peak. This, again, is due to the action of the limiter.

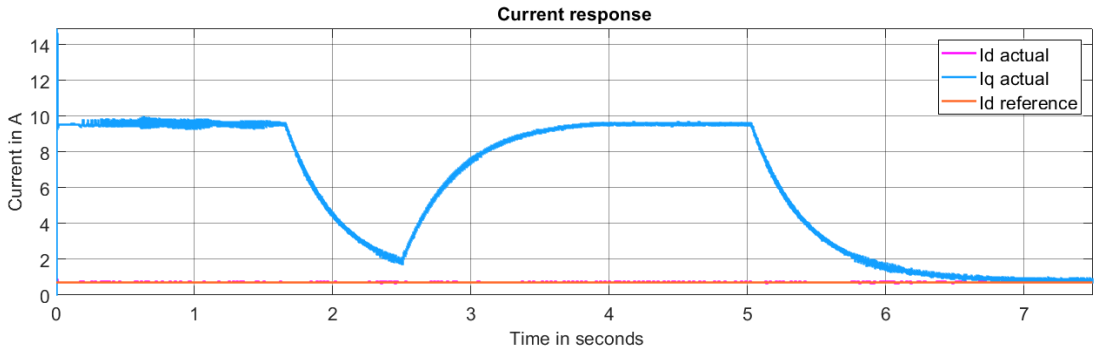
As was seen during the starting and reversal operations, the speed rises linearly until it enters the control region and thereafter the i_q value begins to fall and the change in speed is no longer linear. Again, similar to the previous cases, the speed does not achieve the set-point, however, it can be seen that the error between the actual speed and the reference is indeed getting smaller over time and based on the step response of the controller, would go close to 0 in about 20s.



(a) Load Torque



(b) Speed response

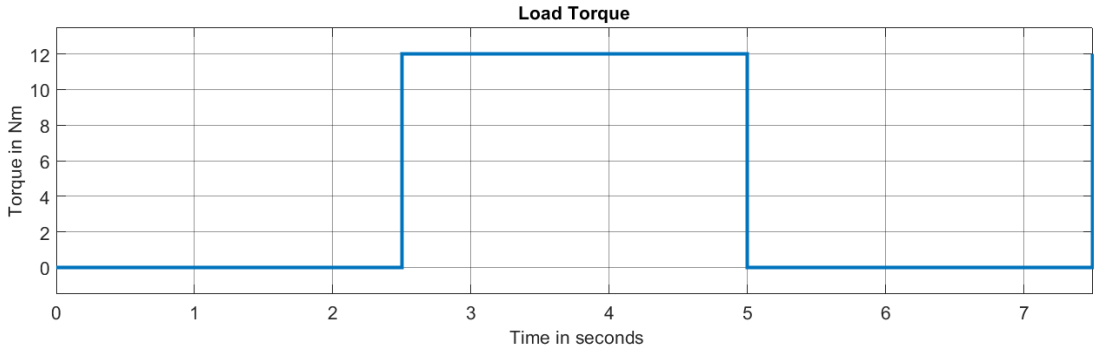


(c) Current response

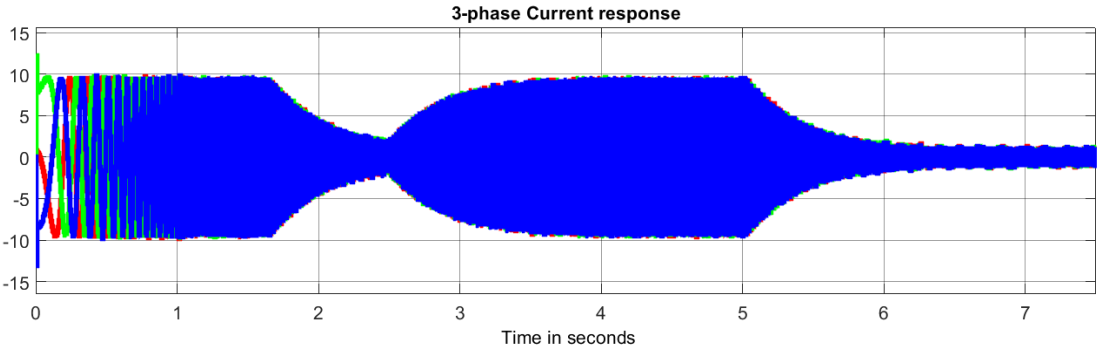
Figure 3.9: Operation under loading

Loading behavior A step load was applied to the motor at 2.5s for a duration of 2.5s and the speed and current responses for the motor were plotted. The results are as shown in Figure 3.9 and Figure 3.10.

From Figure 3.9b one can see that the speed rise is almost linear until 1.5s. This is a result of the PI controller for speed being in saturation. The speed error being high, the entire rated torque is demanded by the speed controller and hence the speed rises linearly. At about 2s, the speed falls in the control range and the rise is no longer linear. At 2.5s, the step load is applied and as a result, the speed begins to fall. Although the speed does rise again to achieve the set-point, the time taken for rising is rather



(a) Load Torque



(b) 3 phase Current response

Figure 3.10: Operation under loading

high(above 20s) and hence not observable in this simulation result. At 5s, once the torque is set to zero, the speed rises again and overshoots the set-point at about 5.5s. Once again, the speed shows a slow fall and gets closer to the set-point, however, due to the high settling time, the fall of speed and achievement of the set-point was not demonstrated in this simulation.

The current response in Figure 3.9c shows that the value of i_d is always at its set-point and remains unaffected by the step changes in torque. The i_q waveform on the other hand is closely linked to the speed waveform due to the fact that it governs the torque being generated by the motor. One can see that i_q is almost constant in the region where the speed has a linear rise and as explained earlier, this is because the reference current in that region is equal to the rated current of the motor. As the speed enters the control region, the i_q value starts falling. However, the fall is cut short by the loading that occurs at 2.5s. The current then begins to rise and at 4s, achieves the rated value again. Although the motor is producing the rated torque at that point, the total load that the generated torque needs to counter is so high that the speed rises extremely slowly. At 5s, once the load is removed, the current starts falling again as there is no longer a

need to produce the rated torque. Even after the set-point is overshot, the current keeps falling and settles at about the same reference value as i_d. Although the load torque on the motor is 0, the motor still faces a torque due to windage and hence the value of i_q does not go down to 0 but rather settles at a finite non-zero value in the steady state.

Comparing the results of Figure 3.8c and Figure 3.10b one can see that the latter does not see any sharp peaks after the initial peak seen during the starting of the motor. From this, one can conclude that it is only the step changes in the speed reference, and not the loading, that causes the sharp peaks to occur in the current waveform. One can, therefore, provide a gradually rising(or falling) speed input to the motor in order to eliminate the presence of these peaks.

3.2.2.2 Analysis of the FOC simulation results

- The low bandwidth of the controller and the high mechanical time constant of the motor result in the motor speed not settling to its set-point within the duration of the simulation. From the step response of the controller, the expected settling time to achieve the reference is in the order of 10s of seconds.
- The transient speed response of the motor is limited by the torque generation capability. As a result, the speed waveform during step changes in input is a straight line. Further, the value of i_q also remains constant in this region.
- The effective control range of the speed controller is limited to about 30 rad/s around the set-point. Beyond this range, the speed controller remains saturated.
- Despite the currents being decoupled using the decoupling terms, the responses of i_d and i_q still showed hints of interdependence in certain conditions. The main cause for this was the voltage limiter action which causes the voltage inputs corresponding to the two currents to be interdependent. However, in regions where the voltage is well within limits, i_d and i_q responses can be seen to be completely decoupled.
- During loading, the motor shows a deceleration of speed and the rate at which the speed rises thereafter is low.
- The 3-phase waveform shows sharp peaks that exceed the current rating of the motor when a step change is applied to the speed reference. Since the duration of these peaks is in the order of milliseconds, one may choose to ignore them as they would not damage the motor. One may also provide a gradually varying reference to the motor instead of a step reference to overcome this problem.

The FOC operation, although greatly limited by the mechanical time constant, supports the theory presented in Section §2.2.

3.2.2.3 EKF simulation

The EKF simulations were performed for two different set of tuned parameters in order to demonstrate the trade off between the estimation accuracy among the estimated states, and the importance of proper tuning.

Loading at rated speed For the first simulation, the EKF parameters were tuned so as to achieve a good high speed performance and the EKF was subjected to a step load at 2.5s for a duration of 2.5s. This is an exact replication of the FOC simulation in 3.2.2.1, except that it also considers the effect of estimated torque

The tuned parameters are as follows:

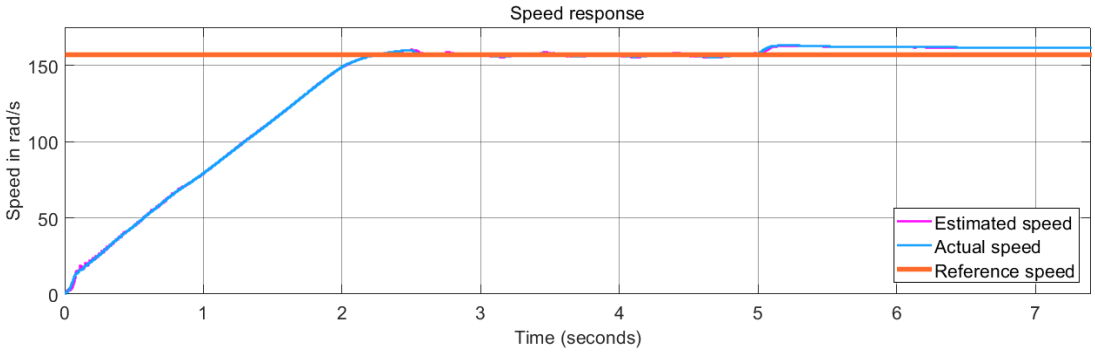
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

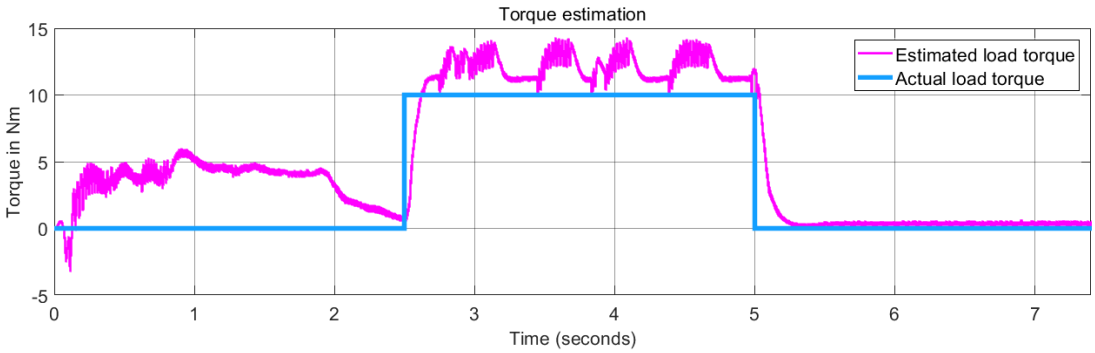
It is possible to further tune these values in order to obtain a better performance but the values used above provided the best response among the several values tried by the author. In general, one would require a grid search to narrow down on the parameters that provide the best results.

Figure 3.11, Figure 3.12 and Figure 3.13 demonstrate the responses obtained on running the EKF in simulations.

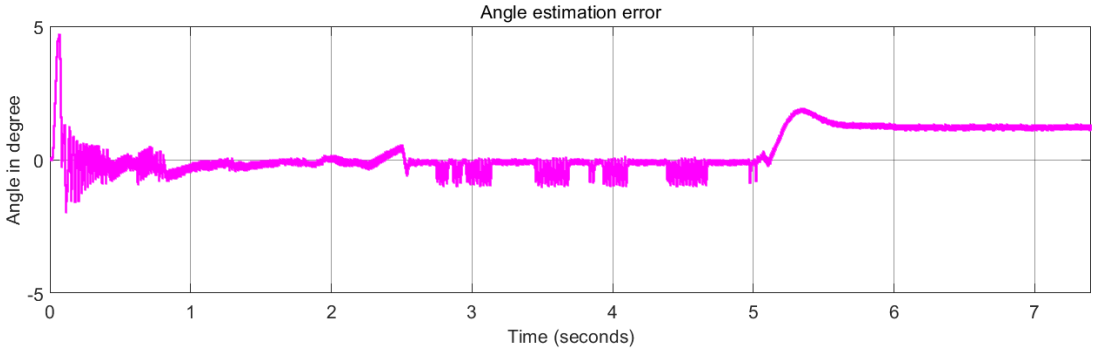
Comparing the response obtained in Figure 3.11a with Figure 3.9b, one can see that the addition of the torque estimate has significantly improved the motor performance under loading. While the FOC case showed a dip in speed that would have taken over 20s to overcome, the EKF manages to maintain the set-point throughout the duration of the loading. Once the load is removed, the FOC case showed a positive deviation from the set-point which would have taken 10s of seconds to overcome. On the other



(a) Speed response with EKF tuned for rated operation

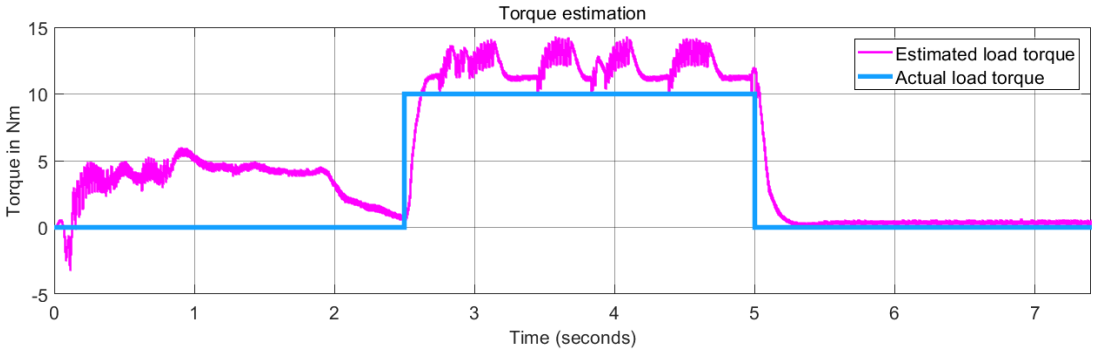


(b) Torque estimate with EKF tuned for rated operation

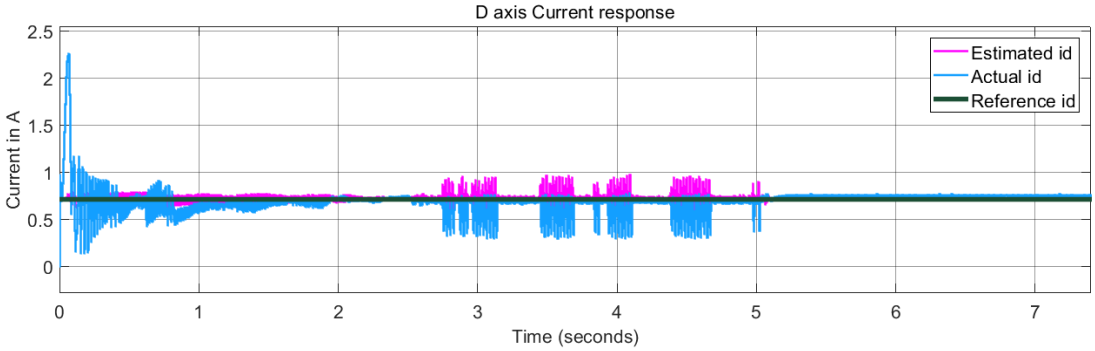


(c) Angle estimation error with EKF tuned for rated operation

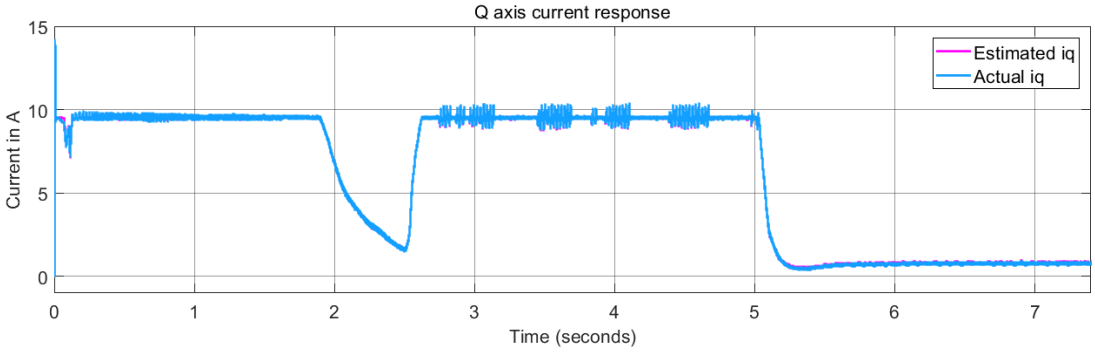
Figure 3.11: EKF for rated loading - speed and torque estimation



(a) Torque estimate with EKF tuned for rated operation

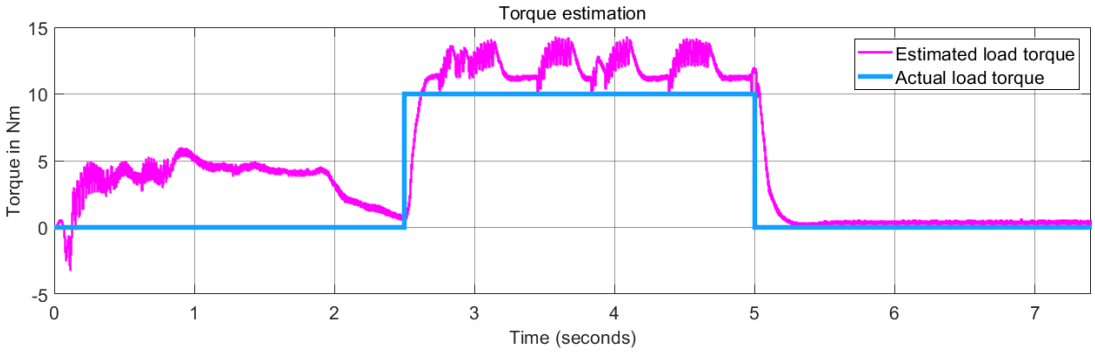


(b) Direct axis current response with EKF tuned for rated operation

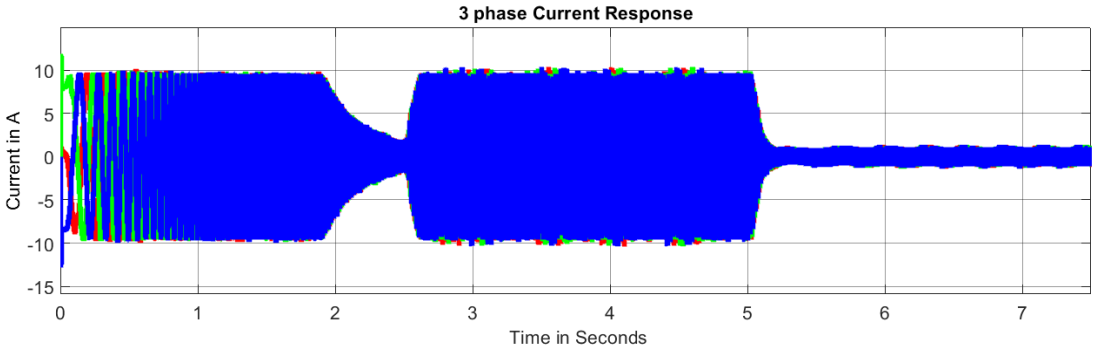


(c) Quadrature axis current response with EKF tuned for rated operation

Figure 3.12: EKF for rated loading - current and torque estimation



(a) Torque estimate with EKF tuned for rated operation



(b) 3-phase current response with EKF tuned for rated operation

Figure 3.13: EKF for rated loading - torque estimation and 3-phase currents

hand, the rate of fall in rotor speed is comparatively faster in the EKF simulation. The reason for this, however, is the erroneous overestimation of the load torque which in turn causes the motor decelerate faster than it would without the estimate.

Figure 3.11b shows the torque estimated by the EKF as compared to the actual torque. One can see that the torque estimates until about 2s are terrible. The estimated value is bout 5Nm while the actual torque is 0. Despite this, the speed response is unaffected. The reason for this can be seen from Figure 3.12b. The i_d estimation also has an error during the same period and the two errors cancel out each other, thus giving the same speed response as one would get from a sensored controller. Even during the loaded condition, that is, between 2.5s and 5s, the torque estimates as well as the i_d estimates are both error prone and based on the speed response, the two errors appear to cancel each other out. The final 2.5s show much better estimation in terms of both i_d as well as the load torque. However, even that comes with a price. Figure 3.11c shows that although the torque estimate is a lot more accurate in the last 2.5s, the angle estimation has a high error.

The currents in Figure 3.13b show that the 3-phase currents occasionally exceed

the current limit of the motor and that these instants correspond to the instants when both the torque and current estimates of the motor have a large error. Unlike the case with FOC, the duration for which these currents exceed their limits is not as short and hence one would require a finer tuning of the EKF to ensure that the estimates are more accurate.

Although the errors in the estimates of load torque, i_d and angle are interrelated and in each 2.5s period, at least one of them has significant error, i_q and rotor speed estimates are almost completely error free.

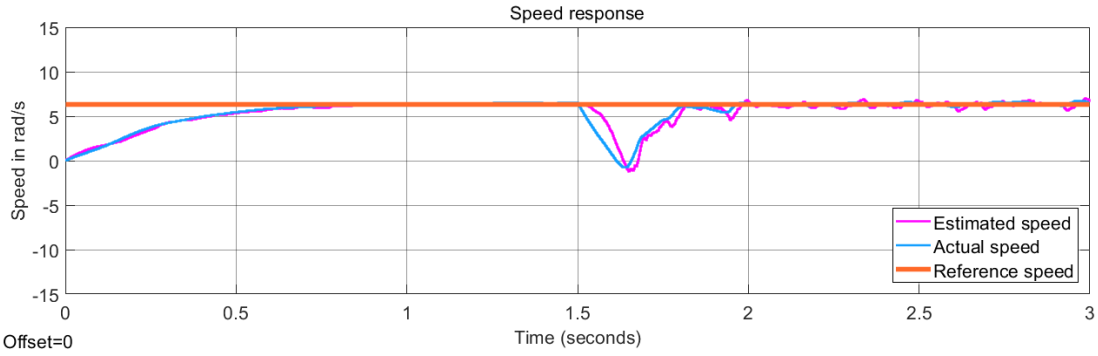
Low speed operation In most sensorless control approaches, accurate estimation of low speeds is a hard nut to crack. Therefore, for the second demonstration, the hyper-parameters (Q and R) were tuned such that the low speed performance was improved. Once more, the hyper-parameters used here have not been claimed to be optimal. However, among all the values that the author has tried, these gave the best results.

The hyper-parameters used for this simulation are as mentioned below:

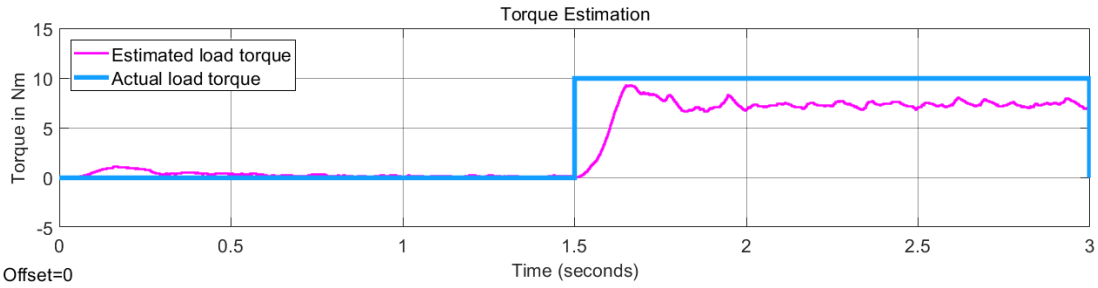
$$Q = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0.009 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

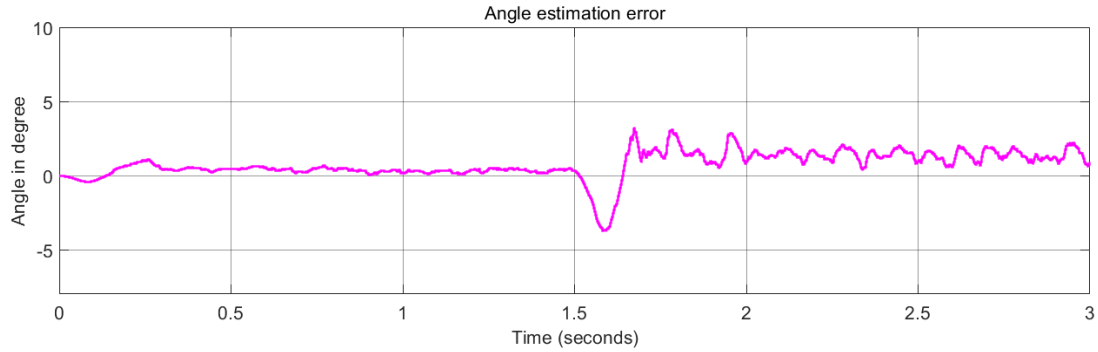
For the simulations, a reference speed of 1Hz, that is 2π rad/s was given to the motor and a step load was applied at 1.5s.



(a) Speed response with EKF tuned for low speed operation

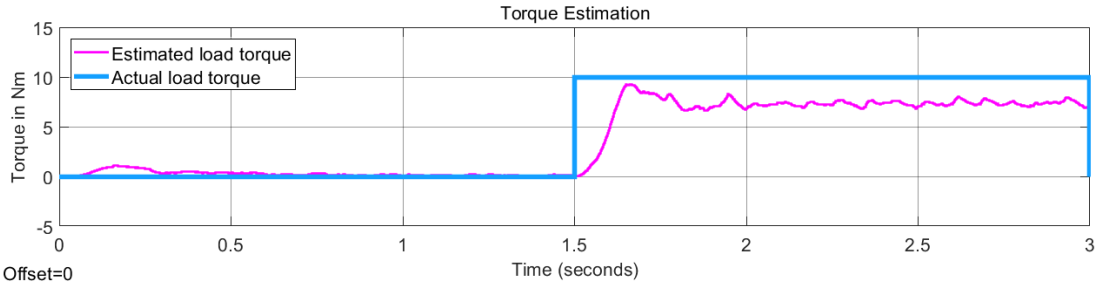


(b) Torque estimate with EKF tuned for low speed operation

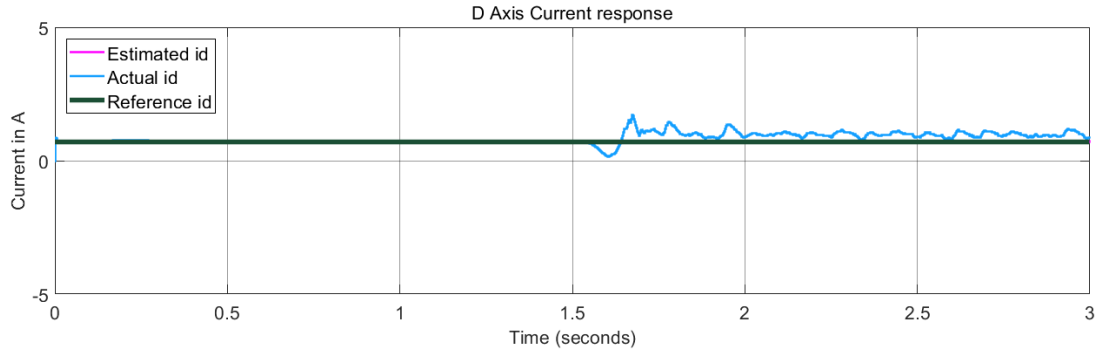


(c) Angle estimation error with EKF tuned for low speed operation

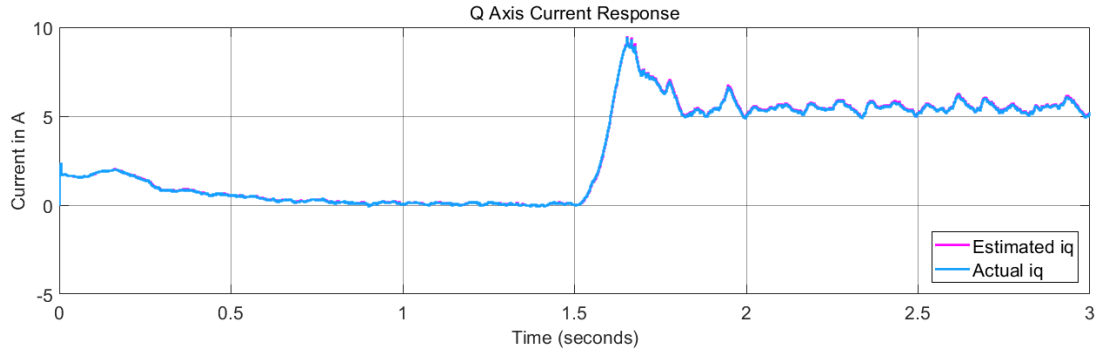
Figure 3.14: EKF for low speed - speed and torque estimation



(a) Torque estimate with EKF tuned for low speed operation

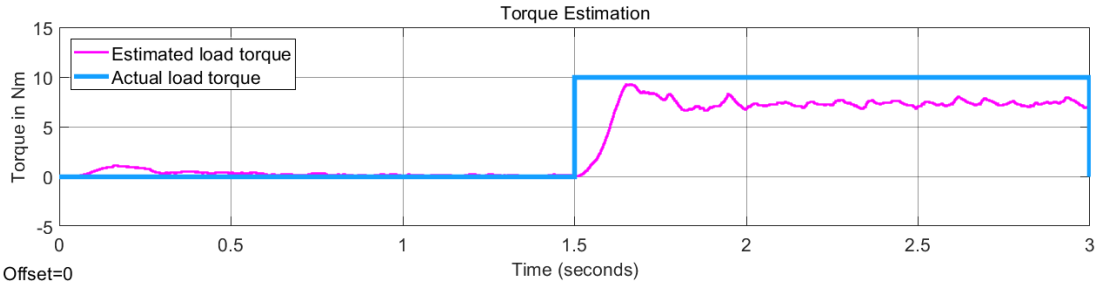


(b) Direct axis current response with EKF tuned for low speed operation

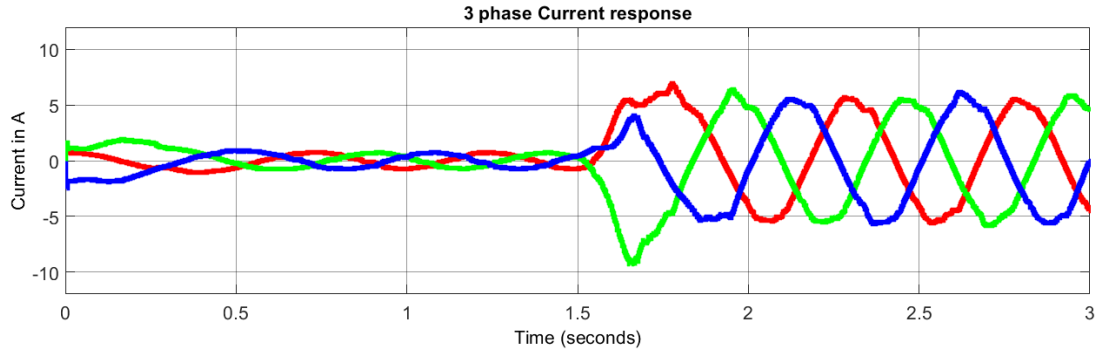


(c) Quadrature axis current response with EKF tuned for low speed operation

Figure 3.15: EKF for low speed - current and torque estimation



(a) Torque estimate with EKF tuned for low speed operation



(b) 3 phase current response with EKF tuned for low speed operation

Figure 3.16: EKF for low speed - torque estimation and 3 phase currents

The results in Figure 3.14, Figure 3.15 and Figure 3.16 show that the EKF is reasonable for sensorless operation even at low speeds.

Figure 3.14a shows that in the first 1.5s of operation under no load, the speed estimate is reasonably accurate although after the application of the step load, there is a significant drop in the speed after which one can see oscillations about the set-point in the estimate as well as the actual speed.

As seen in 3.2.2.3, the errors in estimation of load torque, angle and i_d are interrelated. Although in the first 1.5s all the estimates are reasonably accurate, the next 1.5s where the load is applied show that while the load torque and the d-axis current are being underestimated, the angle is being overestimated. Also, these errors seem to cancel out each other and as a result the speed and i_q responses have a much higher accuracy in comparison.

Being a low speed case, Figure 3.16b shows that despite the error in the estimates, the current limit is no longer a concern as the motor operates well within its rated current limit even under a load of 10Nm.

It is worth noting here that the load torque estimate takes about 0.2s to achieve a

stable value after the application of step load and it is in this period that the angle and i_d estimates have high errors. Even the speed estimate which appears to be largely independent of the errors in the angle, torque and i_d, becomes error prone in this duration. On the other hand, i_q has perfect estimates for the entirety of the operation and although the actual value of i_q shows a step change on the application of load, the accuracy of estimates remains completely unaffected.

3.2.2.4 Analysis of the EKF simulation results and guidelines for using the EKF

- The results obtained in the simulations indicate that with proper tuning, the EKF can be used for achieving position sensorless control of the SyRM. Although the EKF parameters were not fine-tuned, the EKF estimates were still reasonable and the end result, that is the speed reference, was still achieved without much deviation from the 'sensored' FOC response.
- The speed response under the load case with the EKF was significantly better than the speed response achieved in the FOC. This is mainly due to the fact that the load torque was added as a feed-forward term and no longer considered a disturbance. However, the downside to using the EKF is that without proper tuning, the estimates may not be accurate and one may overshoot the rated limits of the motor.
- Tuning of the hyper-parameters has a major impact on the accuracy of the estimates. Unfortunately, there are no strict rules for determining the value for the parameters and a grid search is usually the best way to narrow down on the values which give the best results
- The accuracy of estimation of one state variable is generally not independent of the other state variables. As a result, even a slight change in the hyper-parameter value of one of the variables can affect the accuracy of the entire simulation.
- At lower speeds or standstill condition, adding a high frequency voltage signal to the input is recommended to improve the accuracy of estimates[7]. However, due to time constraints, this method was not tested in the simulations.
- The EKF is prone to numerical instabilities. Care needs to be taken while implementing the algorithm on fixed point or limited precision systems. In particular, the matrix inversion operation in the correction step needs to be handled carefully.
- The EKF estimates improve if the sampling time interval is reduced. This is because the filter can undergo more iterations between successive inputs to the motor and hence can converge to the true value faster. However, as the algorithm involves numerous matrix multiplication operations, one is constrained by the hardware resources and processor clock speed. On an intel i5 processor, a single iteration of the EKF required 3×10^{-5}s without optimization.
- The TI F28335 board used for the tests demonstrated in this report required 0.16ms for the execution of a single iteration of the EKF algorithm when an optimization level of 2 was used. To be able to support the switching frequency

of the inverter, an iteration of the EKF must complete in the duration between successive pulses.

- Increasing the number of measurement variables improves the accuracy of the estimates as long as they are independent of each other. However, adding more measurement variables also makes the algorithm more computationally expensive and there is a trade off between these values.

With the simulations supporting the theory, the next step is to confirm the results of the simulation on the actual motor. This is covered in Chapter 4

CHAPTER 4

HARDWARE IMPLEMENTATION

4.1 Hardware setup

4.1.1 Components required

Testing out the actual motor requires more than just the motor itself. The drive system, as explained in Section §1.1, consists of a microcontroller, an inverter, sensors, and the motor itself.

Apart from this, for running tests on the motor at voltages lower than the rated voltage, an auto-transformer is also required. When testing the drive, it is usually preferable to test with low voltages before implementing the control loop at rated voltage. This is because failure of the control loop at higher voltages can damage both, the motor as well as the inverter, and during the testing phase, this could lead to considerable loss of time and money.

A signal conditioning board, which is often considered to be a part of the controller, is also required for operating the drive. This board is used for level shifting the inverter PWM signals, filtering out the sensor readings, and providing trip protection in case of overloading of the motor.

On the mechanical side, a coupling between the motor shaft and the encoder shaft is required along with a mechanical assembly on which the encoder is mounted so as to align its axis with the motor shaft. Although this task seems trivial, it is quite crucial to align the axes perfectly in order to prevent damage to the encoder during the operation and to ensure that the sensor readings are reliable.

4.1.2 Component description

4.1.2.1 Inverter

The inverter used for this project was manufactured by Semikron and supports a 600V DC bus with a 75A current rating. It can accept upto 415 V of 3phase AC input at the rectifier. The inverter also has ports for giving supply to a cooling fan contained within the housing. The gating signals to the inverter are required to be 15V and apart from the gating signal inputs for the 3 phases of the motor, the inverter also supports gating pulses to regulate the DC bus voltage that is obtained as the output of the rectifier. The rectifier output and the inverter DC bus input terminals are available as external ports. For this application, however, the two terminals were shorted using wires as the rectifier output is also required to be the input for the inverter DC bus.

4.1.2.2 Microcontroller

TI's TMS320F28335 peripheral explorer DSP was the used as the microcontroller for this project. The salient features of the board are as follows:

- Operating voltage of 5V with support for 3.3V I/O
- Clock rate of 150MHz
- Supports single precision floating point operations
- 64 GPIO pins which include 10 pins specialized for generation of PWM signals.
- 12-bit, 16 channel ADC with 80ns conversion rate capable of reading voltages between 0 and 3V

The programs were loaded onto the board's ROM via a USB cable which also powered the board. TI's Code Composer Studio (CCS) software was used to write, debug and load the programs onto the microcontroller.

4.1.2.3 Current sensor

On the sensor side, the current sensing was done using the LA25-NP sensor. It is a Hall effect based sensor and has a conversion ratio of 1:1000. That is, it generates

a signal of 1mA if the current in the wire is 1A. Two of these sensors were used for current measurement. This was because the motor being star connected, the sum of the currents is guaranteed to be zero and hence, measuring the currents in any two phases also provides information about the current in the third phase. The outputs of the sensor are fed into the signal conditioning board which further processes these readings before passing them on to the microcontroller.

4.1.2.4 Rotary encoder

The EP50 S8-1024-3F-N-5 rotary encoder was used for measuring the angles. It is an absolute angle encoder which outputs a value in 10 bit gray code and operates at a voltage of 5V. The encoder is NPN transistor based and thus the sensor outputs need to be pulled up via an external resistor before being read. The sensor supports a maximum current of 32mA and therefore resistors of 1kΩ were used for pull up.The encoder output consists of 10 wires, each containing the information of one of the 10 bits, and 2 wires corresponding to ground and 5V respectively. Being a 10-bit output sensor, the resolution it offers is of $0.703^\circ \pm 15'$ which is sufficient to accurately control the motor. The gray coded sensor readings are read into the microcontroller by configuring the GPIO pins as inputs. The gray code data thus read is converted to binary code, normalized between 0 and 1, and then converted into radians.

4.1.2.5 Signal conditioning board

The signal conditioning board is an important sub-component of the drive that is often not talked about much when discussing the theoretical aspects. The board carries out 3 major tasks - level shifting the PWM, tripping the inverter in case of overloading, and conditioning the sensor signals.

The PWM pulses generated by the F28335 microcontroller have levels of 0V and 5V. The inverter gating pulses on the other hand are required to be at 0V or 15V levels. The signal conditioning board enables the conversion of 5V level PWM signal to a 15V level PWM signal. It is achieved using the CD4504B hex voltage level-shifter IC present on the board. These level shifted signals are then fed into the inverter as the gating pulses.

The overload protection involves using a comparator to check if the measured current is within the thresholds set by the user and using the comparator output to set the enable signal of the PWM being fed into the inverter. If the measured current exceeds the thresholds, the enable signal is turned off and consequently the generation of level shifted PWM signals is terminated.

Since the current sensor's output is a current signal, it is necessary to convert the signal into an equivalent voltage before being used further. This conversion is done by passing the current to the ground through one of the on-board resistors. The signal conditioning board used in this project provides the user the option to pass the current through a 300Ω resistor or a $3k\Omega$ resistor. In this case, choosing the 300Ω resistor is preferred. The reason being, the motor is rated for a peak AC current of $7.32\sqrt{2}$A, that is, $10.35A$. This current, on being sensed by the LA25-NP would generate a current signal of 10.35mA peak. Since the board is powered by a +15/-15V supply, using a $3k\Omega$ resistor would mean that the peak voltage sensed is beyond 15V and thus, further operations on the signal would become meaningless. The 300Ω resistor on the other hand would lead to a peak voltage that is within 4V and thus easy to handle on the board.

The thresholds for the tripping of the inverter are set by adjusting the resistances of the on-board potentiometers. There are two potentiometers for each signal input which correspond to the positive peak and the negative peak. An LM339 quad differential comparator is used to compare the signal voltage with the thresholds. The output of this comparison operation is passed on as an ENABLE signal to the HEF4081B quad 2-input AND gate IC which performs an AND operation of the PWM signal and the ENABLE signal. If the voltage corresponding to the sensed current is beyond the threshold set by the user, a 0 is passed to the ENABLE and the PWMs are effectively turned OFF. On the other hand, if the voltage is within the thresholds, the ENABLE has logic level 1 and PWM pulses are passed on to the inverter.

The final operation that the signal conditioning board enables is the analog conditioning of the voltage signal corresponding to the sensed current (referred to as sensor voltage henceforth). Since the voltage obtained across the 300Ω resistor is just a scaled version of the actual current passing through the motor, it can be expected to vary between +3.135V ($300\Omega \times 10.35mA$) and -3.135V when operating at the rated condition.

Unfortunately, the ADC on the F28335 only supports voltages between 0 and 3V. The analog circuitry on the signal conditioning board scales down the sensor voltage and adds a DC offset of 1.5V to it so as to make the resultant voltage vary between 0 and 3V. The circuit which enables this operation is shown in Figure 4.1. Note that the scaled down sensor voltage is also filtered before finally being made available for being read by the ADC.

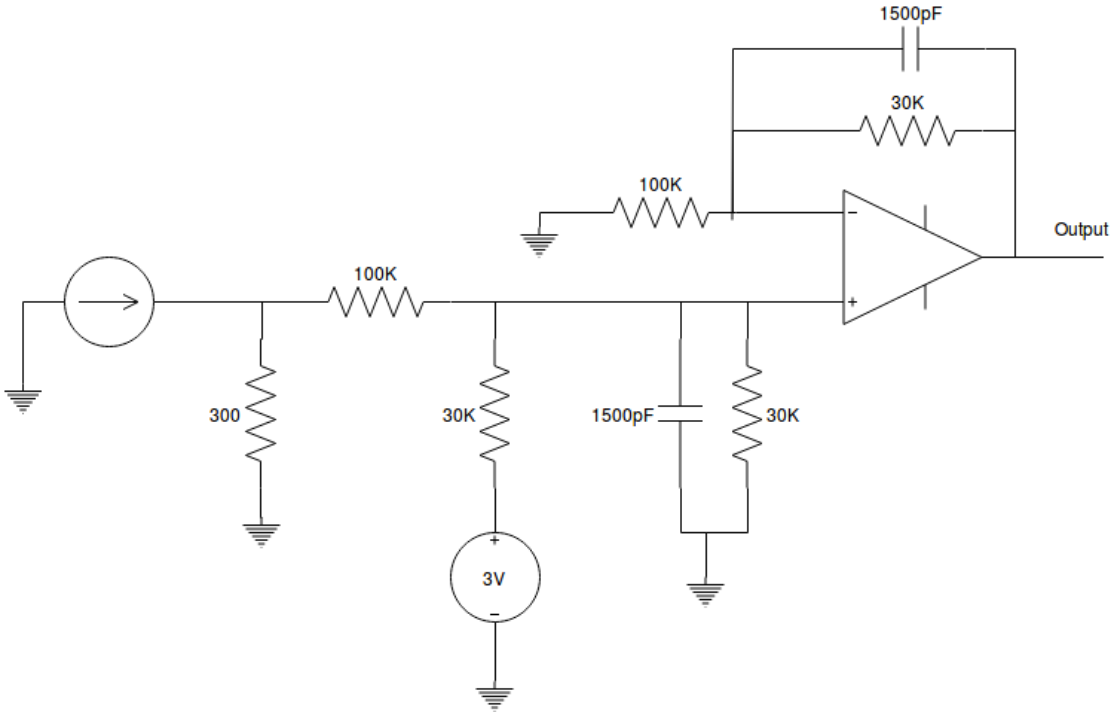


Figure 4.1: Analog conditioning circuit

4.1.2.6 Autotransformer

Two different autotransformers were used during different phases of testing the controller. The first few tests were carried out using a single phase autotransformer rated for 380V and 50A that was supplied from the mains with a phase-to-neutral supply. The output of this was fed into the rectifier and provided a DC bus voltage upto 320V. In the second stage of testing, a 3 phase autotransformer was used which was rated for 380V and 30A. In general, one can just stick to the 3 phase autotransformer and keep the voltage at sufficiently low levels during initial phases, raising it upto the rated voltage only when the controller is tested to work at lower voltages (100V or lower).

4.1.2.7 Final hardware setup

The block diagram of the complete hardware setup is as shown below

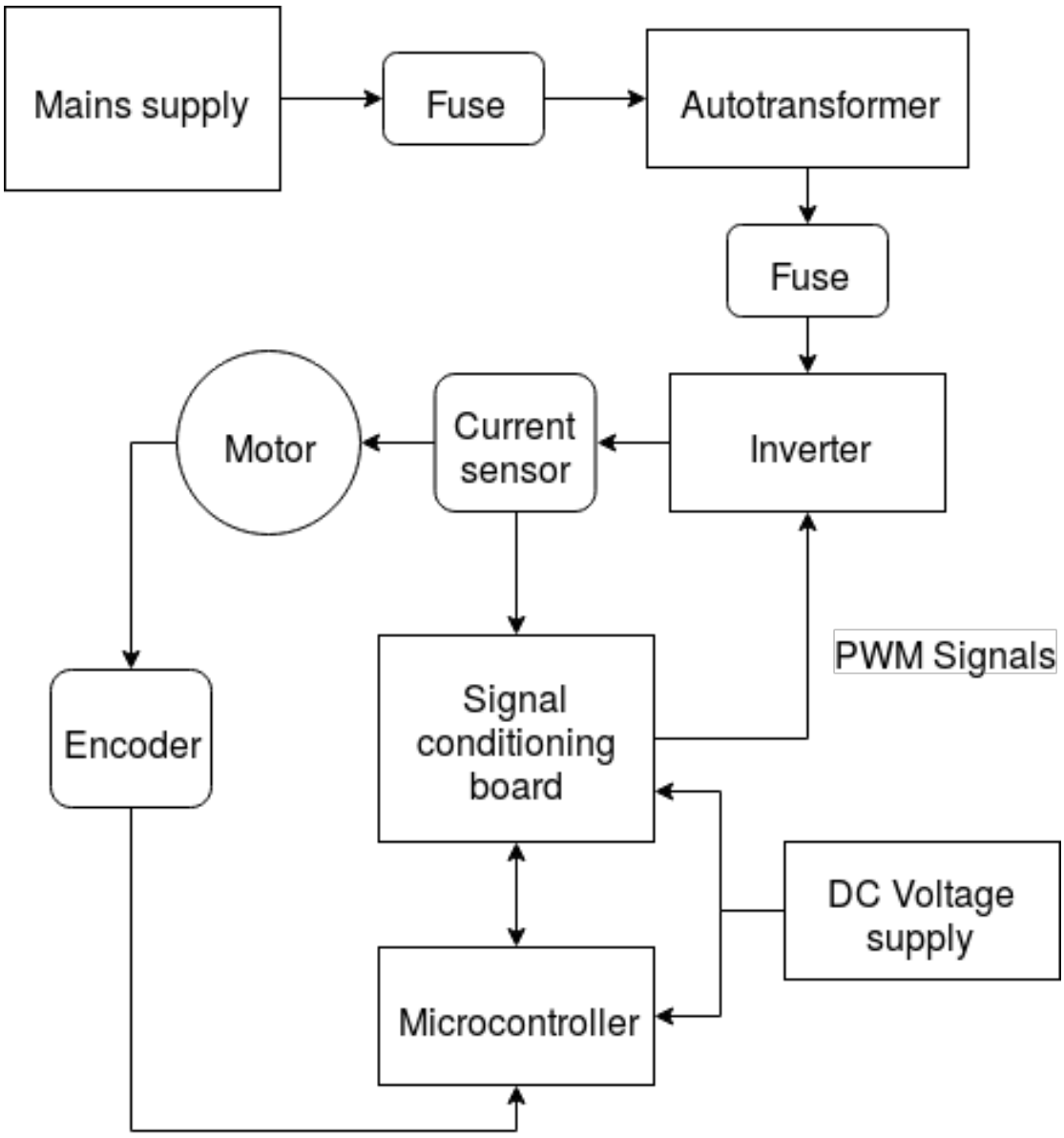


Figure 4.2: Hardware setup

Shielded cables were used for reading the outputs of the current sensor as well as for feeding the conditioned readings to the ADC. This was done to avoid corruption of the signal due to external interference. A connection was also made from the ground of the signal conditioning board to the ground of the microcontroller ADC in order to prevent problems associated with the DC offset.

In case of the rotary encoder, since the readings were digital, using jumper wires for making the connections and mounting the circuit for pull-up on a breadboard was

sufficient. It is recommended to use wires as short as possible for the connections of the encoder in order to ensure minimum latency. In this case, since the frequency of recording the encoder readings was 100Hz, which is sufficiently low, the length of the wires used did not affect the readings much.

4.2 Programming the Microcontroller

The flow of control in the microcontroller program is the same as the Simulink model described in 3.2.1. The Control Suite libraries provided by TI have utilities which implement the CLARKE and PARK transform and their inverses, PI controllers, SVPWM and PWM generators, ADC measurements, etc. These utilities were used to implement the control loop in the microcontroller. It is worth noting that the transforms in MATLAB as well as in Control Suite are power variant.

The major steps involved in the microcontroller implementation are as follows.

4.2.1 Sampling and recording data

The micrcontroller implements timed interrupts using the EPWM modules. The skeleton code for the EPWM interrupts was used for this purpose. The interrupt rate was chosen to be 10kHz, which coincides with the PWM frequency. Counters of the *unsigned int* data type were used to keep track of time for time-based control.

The microcontroller had enough memory to support the storage of 1000 floating point data samples. Therefore, the measured data was recorded at 50Hz with 200 data samples allocated to the measurements of i_a, i_b, i_d, i_q, and ω_m. As a result, the data over a period of 4s could be gathered. Note that the low sampling rate was unable to effectively capture the harmonics in currents. These were captured separately, that is, by ignoring all the measurements, and allocating the entire memory used for measurements to a single array instead. The data was then captured at 1kHz for a time interval of 0.1s

4.2.2 Reading the sensor data

The current sensor data is read from the ADC while the angle data is read from the GPIO pins on the microcontroller.

The ADC readings are prone to noise and hence one must be careful when interfacing the signal conditioning board with the microcontroller. The use of shielded cables is recommended while connecting the outputs of the signal conditioning board to the ADC. Care must also be taken when connecting the GND of the signal conditioning board and the GND of the ADCs as it is one of the major sources of noise.

The encoder output is in gray code and the data read by the microcontroller is converted into binary code using standard conversion algorithms. The binary coded data is then converted into an angle value in radian and used for the transforms.

4.2.2.1 Speed estimation

While the encoder gives accurate angle data, the speed estimation from the angle data is not as easy. The obvious method of estimating speed would be to use discrete differentiation. However, the results obtained from this method are extremely noisy and require low pass filtering in order to suppress the high frequency noise that the discrete differentiators are prone to. The other method is to measure the time between successive bit flips for a GPIO pin and using the angle information to obtain the speed. E.g. the 5th most significant bit flips on traversing an angle of $\frac{\pi}{8}$. Thus, the speed can be obtained by $\frac{\pi}{8\triangle T}$ where $\triangle T$ is the time taken between successive bit flips. Since the signals are sampled at $10kHz$, one can use a counter to keep track of the time and obtain the speed estimate. Although the noise present in this method is lower than using the differentiator, the trade off is its update rate. The speed estimate is only updated after a particular angle - dependent on which bit is being tracked - has been traversed by the rotor, and hence, if the speed is low, the controller performance would be limited by the speed estimation itself. Unfortunately, blindly using the LSBs does not help the cause either as the high speed performance suffers when using these bits - in this case the angle is traversed before the counter updates - thus leading to absurd or wrong values. It was observed that the 5^{th} or 6^{th} MSB provided the best speed estimates for a wide operation range. For this project, the 6^{th} MSB was tracked.

4.2.3 PI Controller

The current and the angle readings are passed on to the CLARKE and PARK transform modules to obtain the values of i_d and i_q. These values, along with the speed estimate are then used as the feedback terms for the PI controller. Since the PI controller utility implements a discrete time PI controller, the K_I value needs to be scaled appropriately depending on the sampling time used.

Since the Simulink model does not account for effects like saturation and inaccuracies in the estimation of motor parameters, one may need to tune the PI controller bandwidths and the gain values in order to achieve a reasonable performance.

In the case of this motor, it was observed that the decoupling terms caused high frequency spikes in the current waveforms which in turn caused the signal conditioning board to trip the inverter. One way to get around this problem was to ignore the decoupling terms altogether. The other way was to use the reference values of currents and speed instead of the measured values in order to generate the decoupling terms. These approaches are compared in Section §4.3.

4.2.4 Generating the gate pulses

The pulse generation process is simplified a great deal by the utilities provided in Control Suite. The v_d and v_q values obtained as the outputs of the PI controllers are scaled and converted to v_α and v_β using the inverse Clarke transform (*ICLARKE*) utility. These values are then scaled and fed into the SVPWM generator utility, the output of which gives the duration of the pulses for the PWM signals to be supplied to the motor. These pulse durations are then passed on to the PWM module which generates the gating signals that are finally passed on to the signal conditioning board.

Note that the inputs to the SVPWM module are expected to lie between -1 and 1. For ensuring that the module functions properly, one also needs to ensure that the magnitude of the voltage, that is, $\sqrt{v_\alpha^2 + v_\beta^2}$ does not exceed 1. To ensure this, the v_d and v_q values are scaled such that the angle between them is preserved and the overall magnitude is suppressed so as to not exceed 1. This method was used as it is similar to the way MATLAB handles over voltages while generating the SVPWM pulses.

The scaling factor is equal to $\frac{\sqrt{3}}{V_{DC}}$ and is dependent on the DC bus voltage. As a result, the DC bus voltage needs to be measured and declared as a constant in the microcontroller program.

4.3 Results

The FOC algorithm was implemented on the SyRM. In the simulations carried out in Section §3.2, the bandwidths were set at 100Hz for the current controller and 0.2Hz for the speed controller. For the hardware implementation, however, the bandwidths had to be recomputed. The Simulink model used for the simulations turned out to be an insufficient representation of the motor dynamics. Factors like inaccuracies in parameter estimation, saturation not being accounted for, friction torque not being considered, inductances being assumed constant, and harmonics introduced by rectifier among others led to a significant difference between the simulations and the results obtained on hardware.

Another of the causes for this discrepancy between the hardware implementation and the simulations was found to be the presence of a peak near 250 Hz frequency in the spectrum of the d-axis currents when the current controller was tuned for 100Hz bandwidth.

The FFT spectrum of the i_d current response is as shown in Figure 4.3.

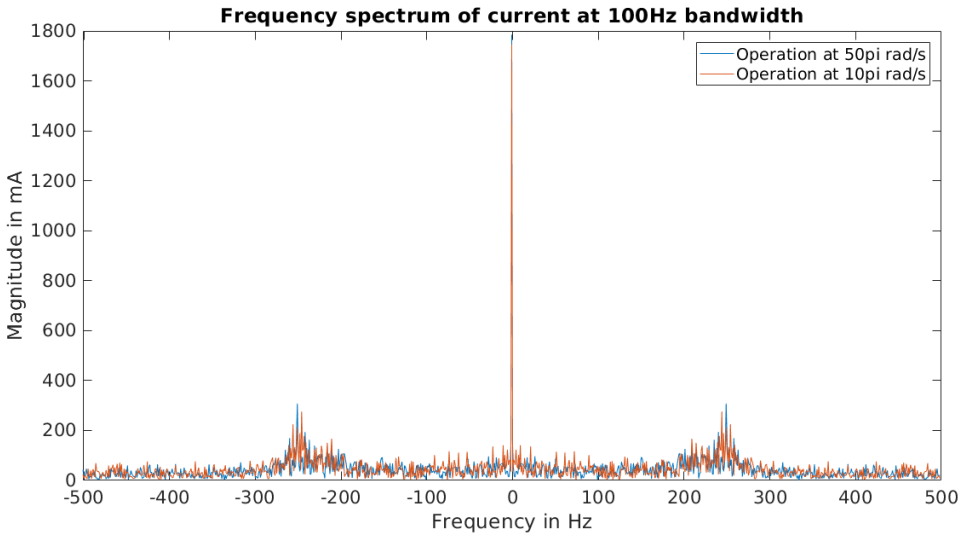


Figure 4.3: FFT spectrum of direct axis current response at 100Hz bandwidth

It was also observed that increasing the reference i_d current to 1.68A resulted in a better response in hardware.

After running a number of tests and some manual tuning, the controller gains that were finally arrived at gave reasonable results in the tests run on the actual motor but failed to follow the reference when used in simulations. Thus, a clear discrepancy exists between the actual motor and the model, addressing which would need a reasonable amount of time.

An additional note was that while computing the decoupling terms the actual measured currents were not added and rather the reference values were used along with the measured speed. This was done because adding the actual currents amplified the noise in the system and also caused the inverter to trip almost immediately after the motor started. Further, for the purpose of comparing the two responses, the output of the ω_m controller was saturated so as to limit i_{q_ref} to within $\pm 4A$. This, again, was done to ensure that the inverter does not trip during operation.

Due to the absence of a loading mechanism, the values of rotor speed, i_d and i_q were recorded at 50Hz sampling frequency for the starting, reversal and braking operation, at rated speed and voltage, and no load condition. The motor response was also compared for the cases when the decoupling terms were added to the PI controller outputs and when they weren't.

4.3.1 Starting operation

In the first test, the starting operation of the motor was compared with and without the presence of the decoupling terms.

4.3.1.1 With decoupling terms

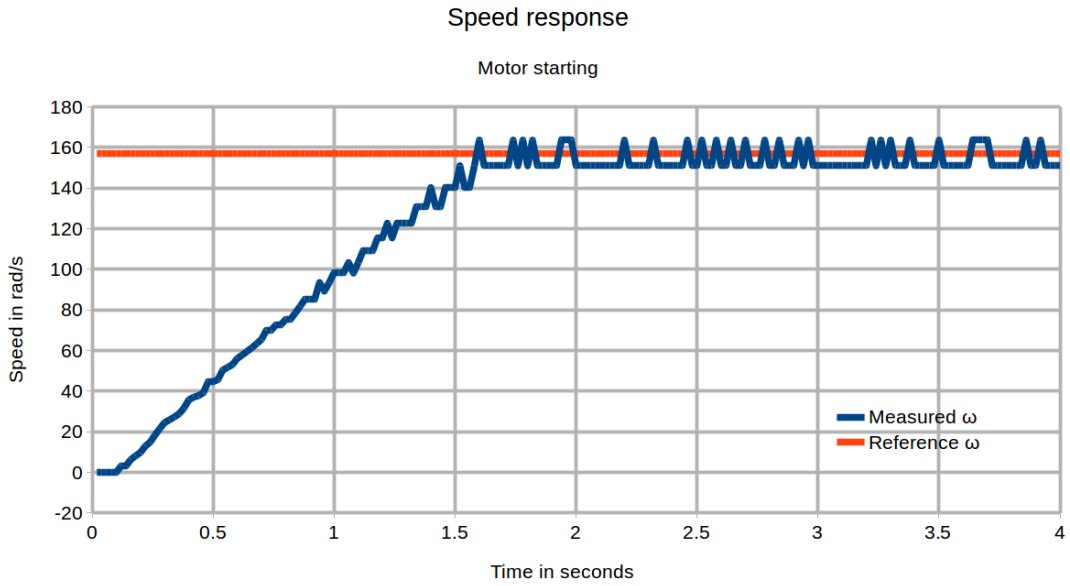


Figure 4.4: Speed response for starting operation without decoupling terms

As seen in Figure 4.4, the motor speed reaches the set-point at 1.56s which is less than the time it took for reaching the set-point in simulations. This is despite the current in the quadrature axis is being limited to about half of its rated value. The cause for this is not completely clear but could be a result of an inaccurate estimation of the mechanical parameters. It is also worth observing that the speed rise occurs in a linear manner, much like what was observed in simulations, due to the constraints on the torque generation.

One can also see that on reaching the set-point, the rotor speed takes the values of 163.625 rad/s and 151.038 rad/s. This is mainly because of the way the speed is estimated - tracking the number of 10kHz pulses between successive bit flips of the 6^{th} MSB. Therefore, the resolution of speed estimation is limited.

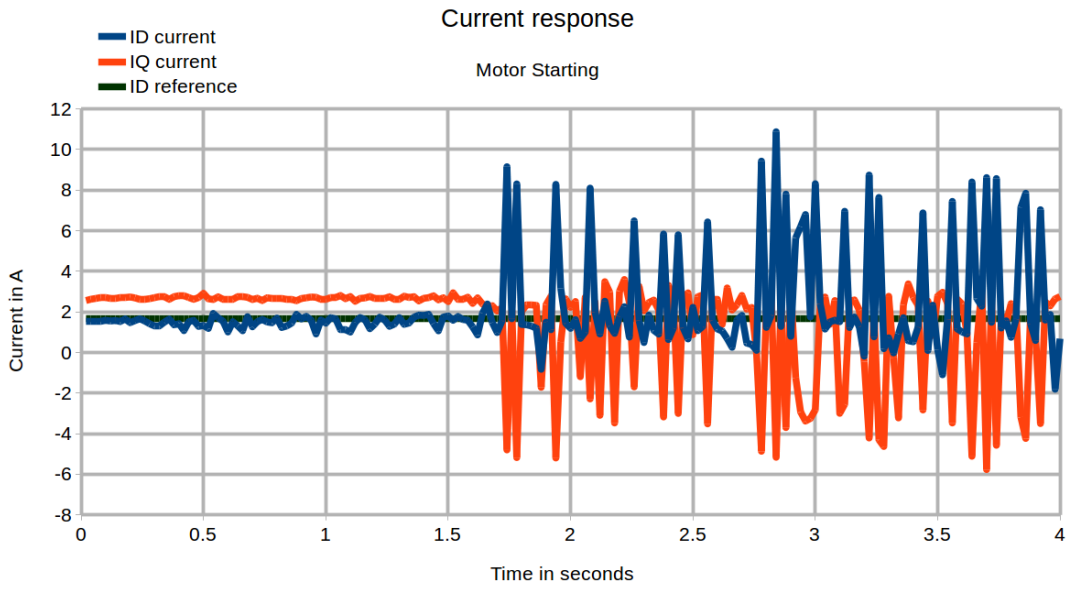


Figure 4.5: Current response for starting operation with decoupling terms

The current response in Figure 4.5 shows that the i_d and i_q values stay stable at 1.68A and 2.5A respectively until the set-point is achieved and thereafter, have large oscillations. While i_q oscillation are about 0A, the oscillations of i_d consist of high frequency peaks superposed onto the 1.68A reference that the current follows. These oscillations are likely a result of the speed estimate not obtaining the set-point.

4.3.1.2 Without decoupling terms

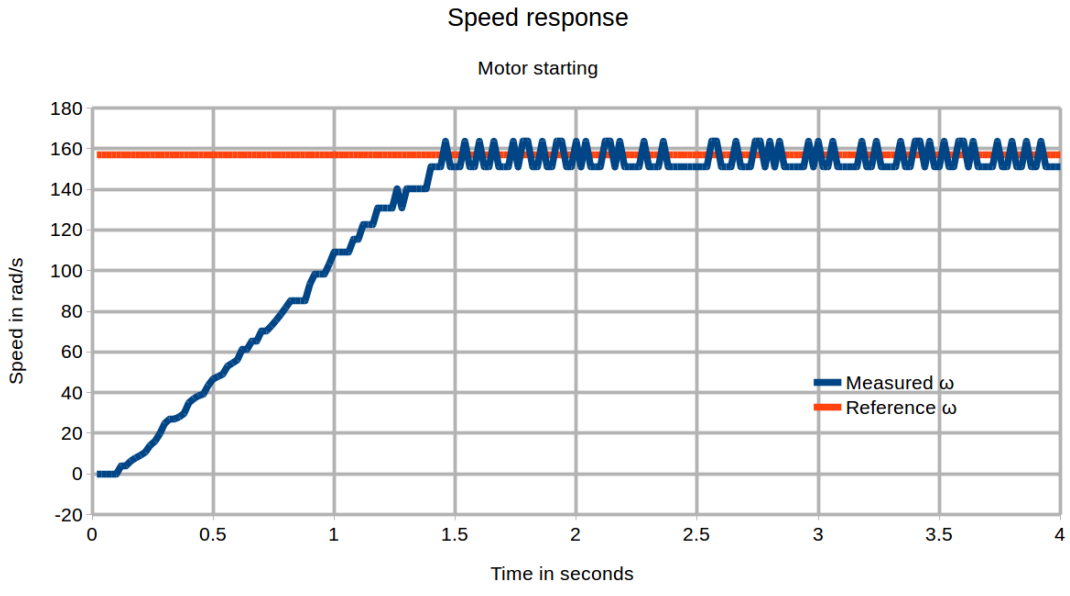


Figure 4.6: Speed response for starting operation without decoupling terms

In the case of the decoupling terms being neglected, Figure 4.6 shows that the rotor speed achieves the set-point before 1.5s, as opposed to 1.56s as seen in 4.3.1.1. This difference, however, is small enough to be attributed to the noise in the system. Apart from this, no significant difference is observed between the speed response of the two cases.

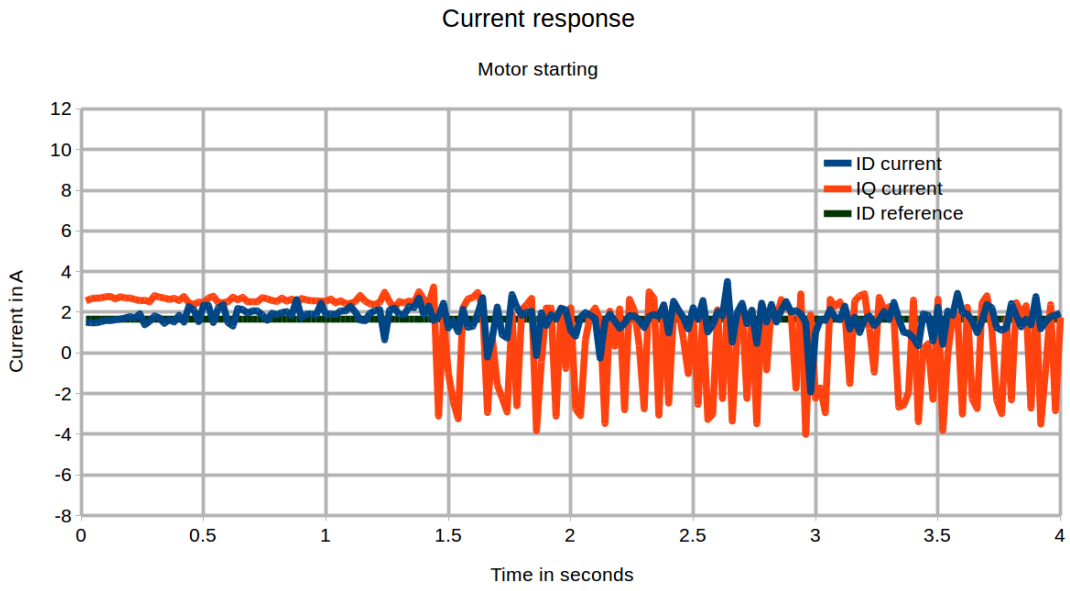


Figure 4.7: Current response for starting operation without decoupling terms

While the response before achieving set-point is similar for both the cases, it is easy to see from Figure 4.7 that the magnitude of oscillations in the current waveform are much smaller when the decoupling terms are not used.

The i_q values oscillated between 3A and -5A in the case of decoupling while in this case the oscillations are between 3A and -3A. The most significant difference, however, is in the i_d response. In the results obtained in 4.3.1.1, the current contained high frequency peaks with magnitude as high as 10A while in this case, the peaks are absent and the current itself does not deviate from the set-point by more than 4A.

The reason for the decoupled controller operation containing high frequency harmonics can be attributed to the fact that the v_d and v_q values have terms that are added onto the low pass filtered outputs of the PI controllers. As a result, the actual voltage that gets applied to the motor does not suppress the 250Hz frequencies observed in Figure 4.3 which in turn ends up amplifying the current.

4.3.2 Speed reversal

4.3.2.1 With decoupling terms

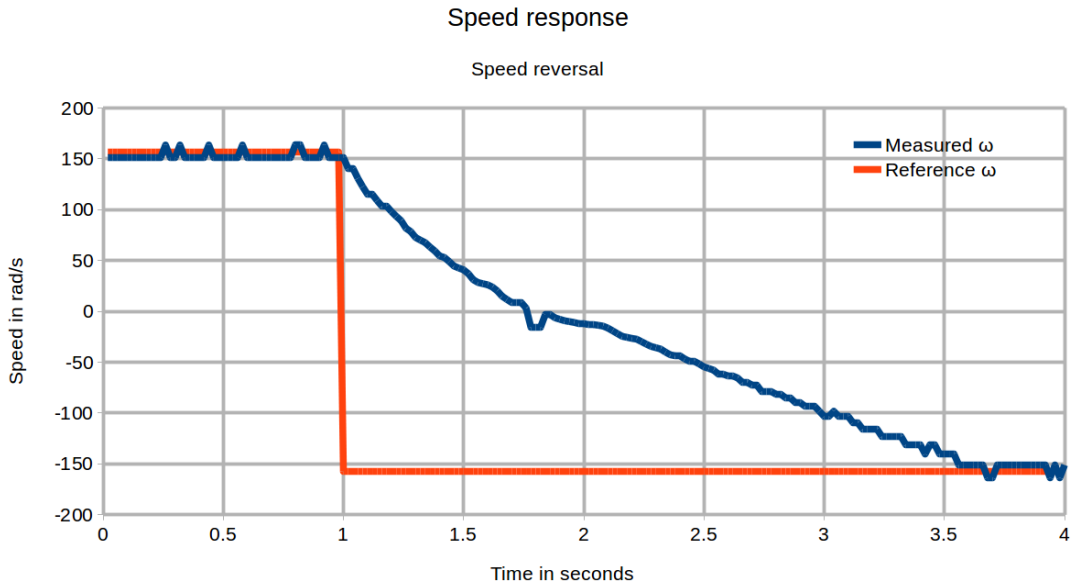


Figure 4.8: Speed response for speed reversal operation with decoupling terms

Figure 4.8 shows that the speed reversal happens in 2.6s. One can see that the speed response is largely a straight line except when the speed is near 0 rad/s. This can be attributed to the way the speed is estimated. Since the sign of the speed is determined based on the angle information as well as the past speed information, the speed estimation at low speeds is not as accurate as it is in the high speed region and is prone to errors in sign.

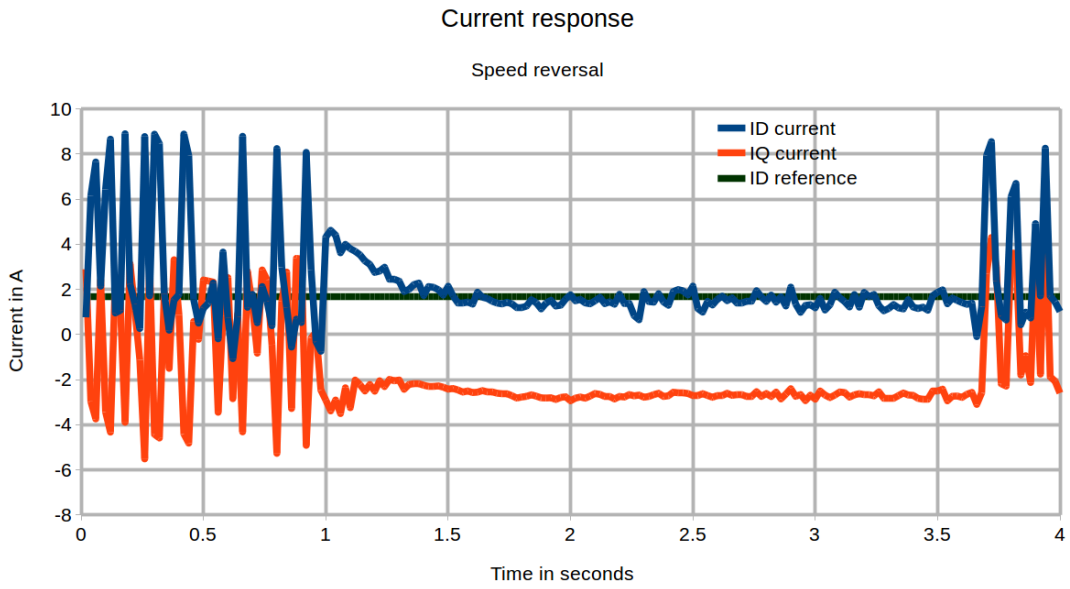


Figure 4.9: Current response for speed reversal operation with decoupling terms

The current response observed in Figure 4.9 is not very different from what was observed in the motor starting case. Both the current waveforms have large oscillations when the motor is at the set-point while the values taken by the currents are constant at 1.68A and about -2.6A for i_d and i_q respectively when the speed is in transition. As before, the high frequency oscillations can be attributed to the fact that the 250Hz peak in the current response is not completely suppressed when the decoupling terms are added.

4.3.2.2 Without decoupling terms

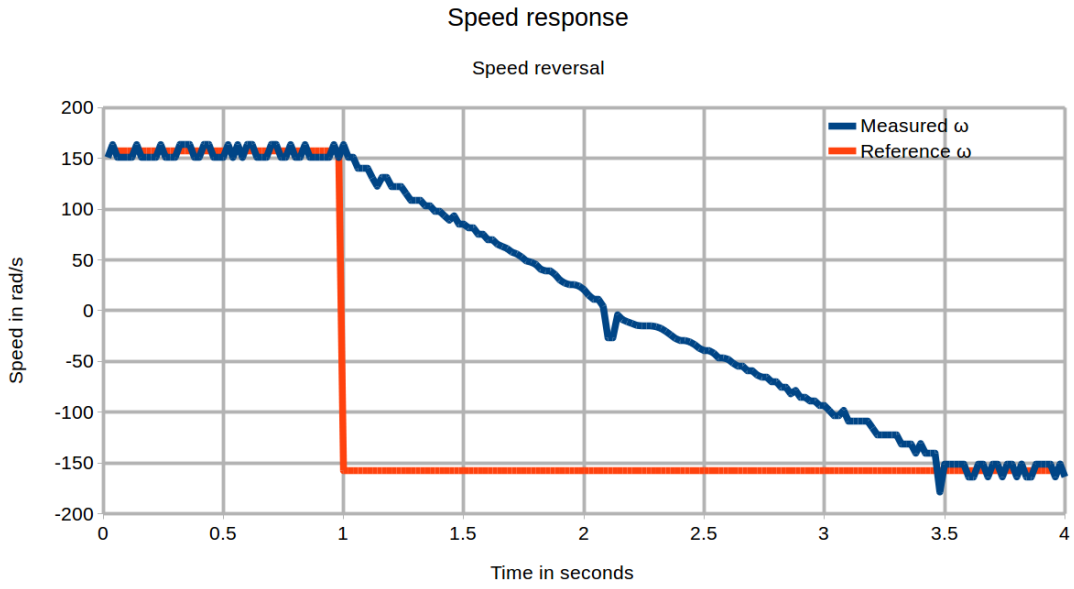


Figure 4.10: Speed response for speed reversal operation without decoupling terms

Similar to the starting case, the speed response without decoupling as seen in Figure 4.10 is not any different from the one seen in 4.3.2.1. As seen in 4.3.2.1, the speed estimate in the near-zero speed region seems to have a sign error that can be attributed to the estimation technique used.

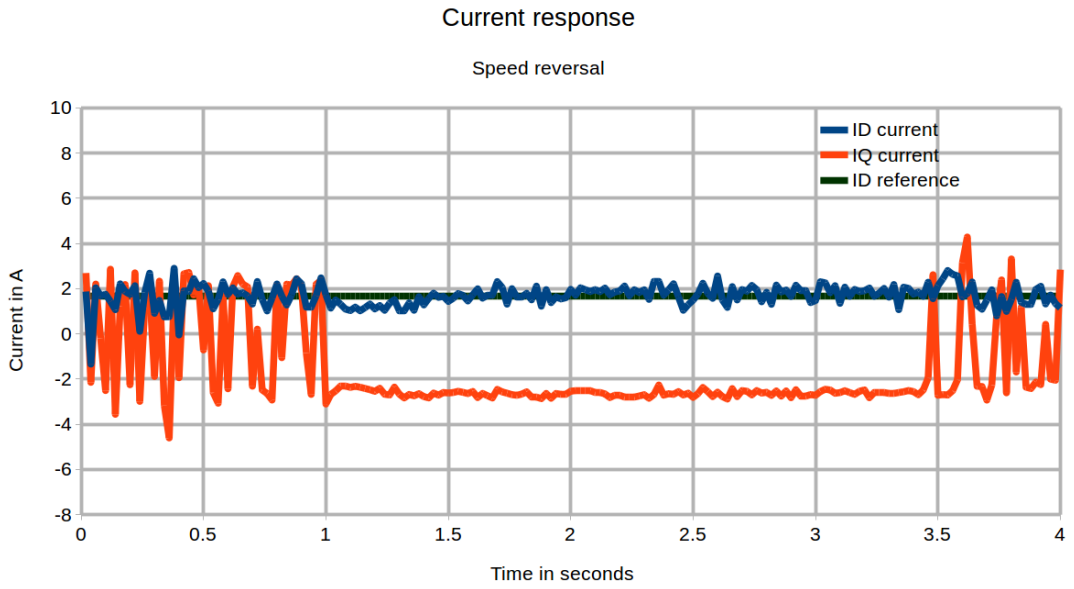


Figure 4.11: Current response for speed reversal operation without decoupling terms

Figure 4.11 shows that much like the observation during the motor starting case, the removal of decoupling terms suppresses the oscillations in the current waveforms. The value of i_q oscillates about 0A when the speed is at the set-point while it obtains the value of -2.6A when the speed is in transition. The i_d waveform closely follows the reference of 1.68A and the magnitude of oscillations is much smaller compared to the results obtained in 4.3.2.1.

4.3.3 Braking

4.3.3.1 With decoupling terms

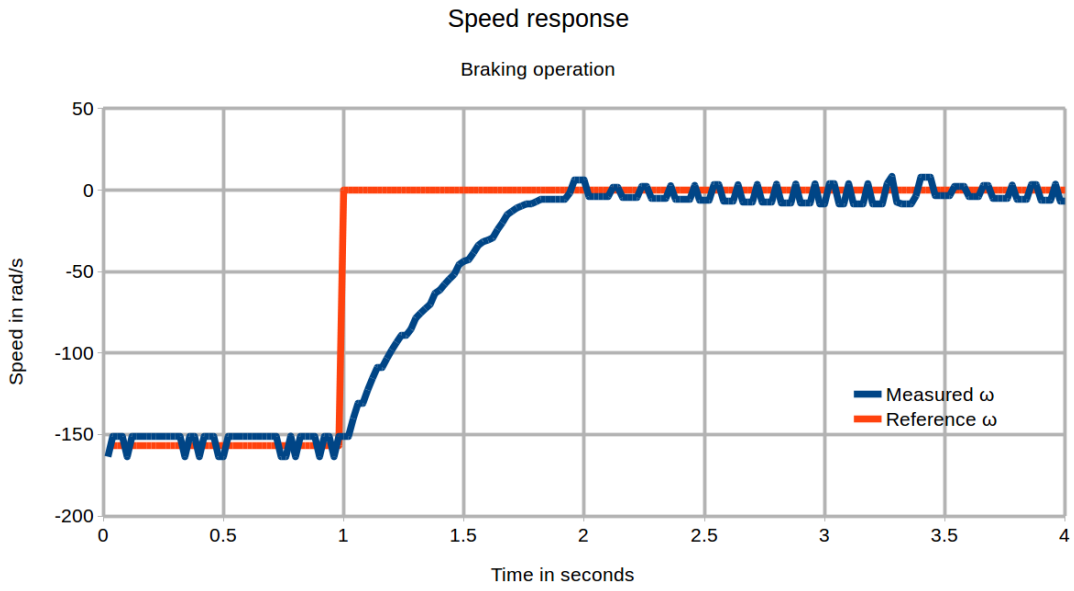


Figure 4.12: Speed response for braking operation with decoupling terms

Figure 4.12 shows that the braking operation occurs in 0.9s. However, it can be seen that despite the set-point being at 0, the recorded speed does not stay constant at 0 rad/s. This is due to the way speed is estimated, that is, with the flipping of the 6^{th} MSB. Obviously, to update the speed estimate, the rotor needs to have moved by a certain angle. And as a result, the speed oscillates about 0 instead of settling at that value. Recalling the results of 4.3.1, one can see that the time taken for braking is considerably lower than the starting time despite the step change in speed being the same in both cases. This difference can be attributed to the differences in the i_d and i_q waveforms in the two cases. During the starting operation, both i_d and i_q were initially

at 0 and a finite amount of time was required for these values to obtain their set-points. In the braking operation, however, the i_d value was already at the set-point and the i_q value was oscillating about 0. This, along with the minor difference in the windage loss due to the cooling fan of the motor, and the friction aiding the braking, caused the braking operation to conclude faster than the starting operation.

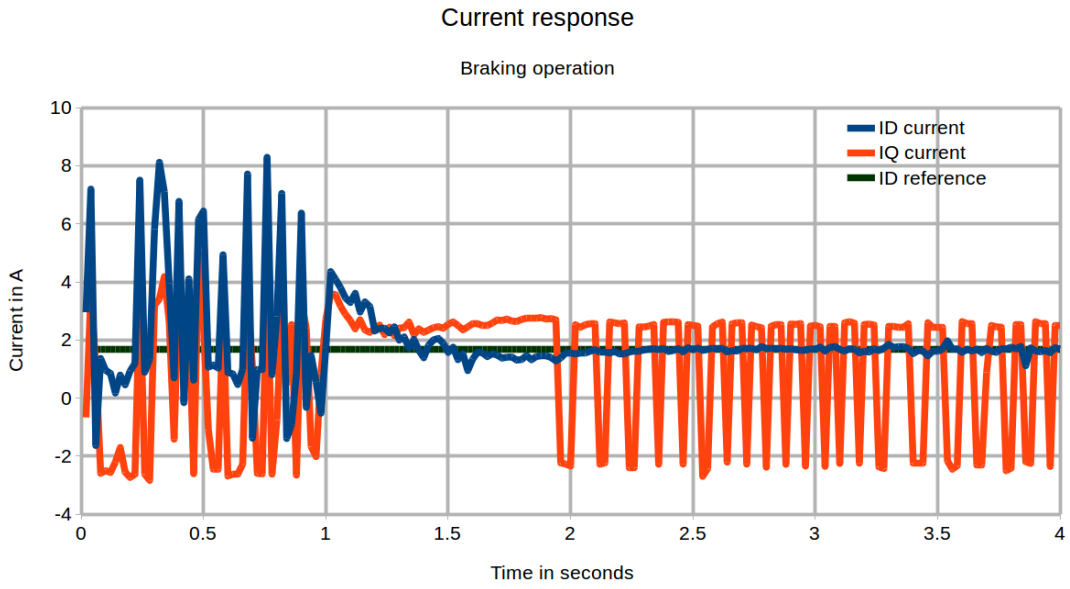


Figure 4.13: Current response for braking operation with decoupling terms

Similar to the results obtained in 4.3.1 and 4.3.2, the currents, as shown in Figure 4.13 oscillate when they are at the set-point of -50π rad/s. However, when the set-point of 0 rad/s is reached, the oscillations in i_d die down almost completely while i_q value continues to oscillate about 0A. The reason for this is that the decoupling terms themselves have a very low magnitude when the speed magnitude is small and hence resonant frequencies in the control loop do not affect the current response.

4.3.3.2 Without decoupling terms

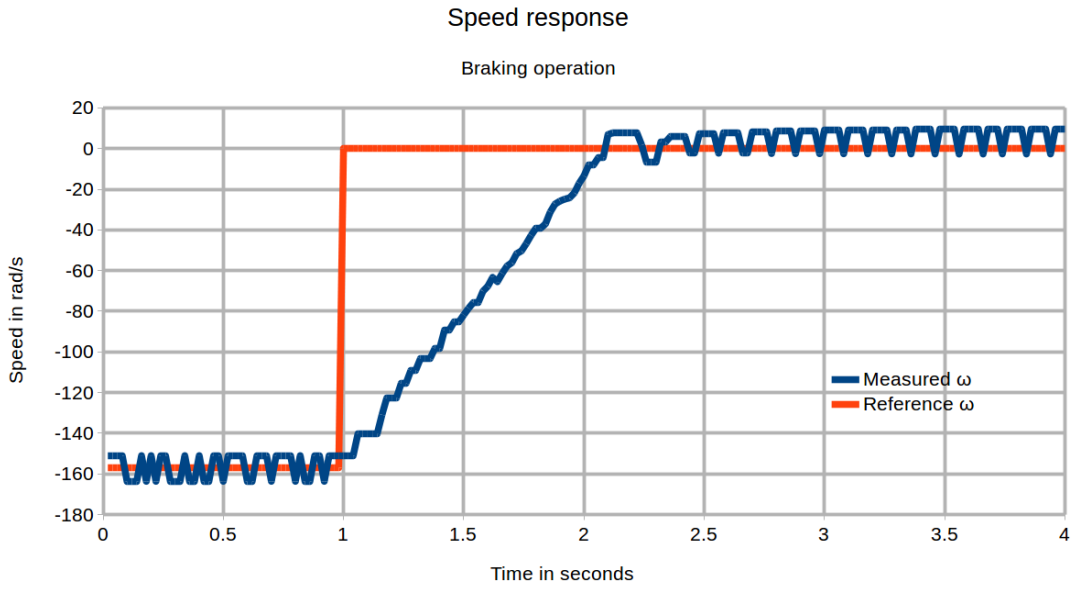


Figure 4.14: Speed response for braking operation without decoupling terms

The braking operation in the no decoupling case takes 1.2s to complete as shown in Figure 4.14 while the results in 4.3.3.1 required 0.9s. This can be attributed to the difference in current waveforms between the two cases. In the results obtained in 4.3.3.1, the currents are slightly on the higher side due to the disturbances in the system and hence the torque generated is higher. As a result, the set-point is achieved faster. Apart from this, however, there are no major differences between the two cases.

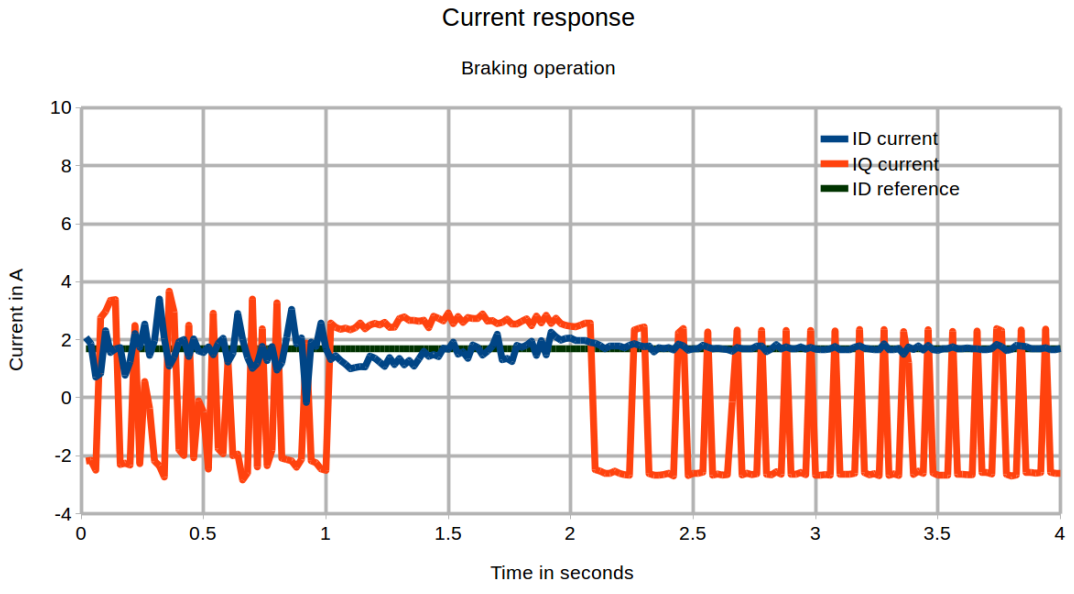


Figure 4.15: Current response for braking operation without decoupling terms

Once again, it can be seen from Figure 4.15 that the oscillations in the currents are much smaller in the case when decoupling terms are not added, when the speed is at its rated value. Further, as mentioned above, the magnitude of currents is slightly lower during the transition duration and as a result the braking operation takes longer compared to the results obtained in 4.3.3.1. One can also see that the waveforms obtained during the near 0 speed operation are very similar to the ones obtained in 4.3.3.1. The explanation for this similarity is that the magnitude of the decoupling terms is so small that there isn't any observable difference between the two cases.

4.3.4 Results with rated current limit

From the results obtained in 4.3.1, 4.3.2 and 4.3.3, it is easy to conclude that not using the decoupling terms results in a better current response in terms of noise and harmonic content.

The testing in 4.3.1, 4.3.2 and 4.3.3, however, involved limiting the i_{q_ref} value to within $\pm 4A$ and hence did not provide an accurate depiction of what the motor is actually capable of. Therefore, tests were conducted after resetting PI saturation limit to the rated value.

In the following results, the quantities i_a and i_b were not measured and the measurements were recorded at 100Hz for a period of 3s.

4.3.4.1 Starting operation

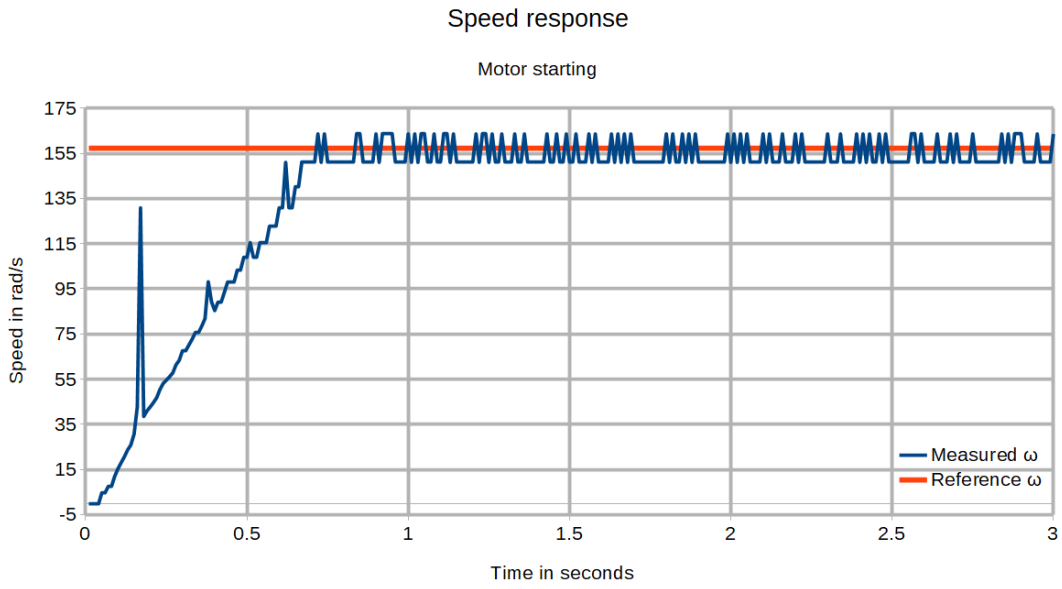


Figure 4.16: Speed response for rated current limit starting operation

Contrary to the simulations in 3.2.2, Figure 4.16 shows that the speed achieves the set-point within 0.8s. Additionally, much like the results in 4.3.1, the speed does not settle at the reference but instead oscillates about the set-point due to the way estimation takes place. The sharp peak in speed at about 0.2s, is likely the result of the encoder bit switching signs due to electromagnetic interference (EMI).

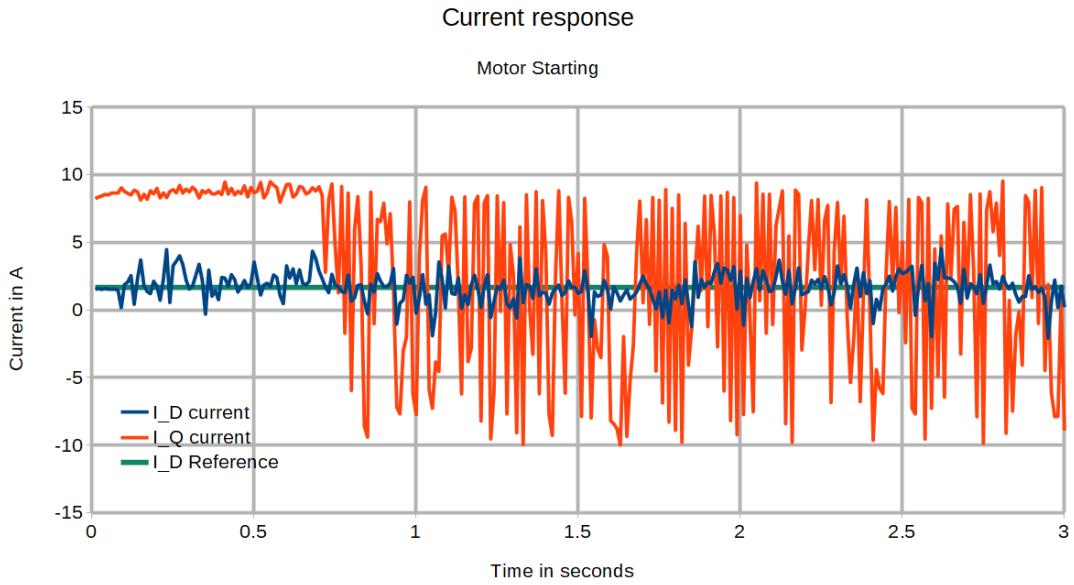


Figure 4.17: Current response for rated current limit starting operation

As can be seen in Figure 4.17, the i_d value follows the reference as expected while i_q obtains a value of about 8.5A until the set-point is achieved, following which it oscillates between -10A and 10A. The inverter trip limit was set to 12A which is slightly higher than the peak of the rated current (10.6A) , to account for these oscillations in i_q.

4.3.4.2 Speed Reversal

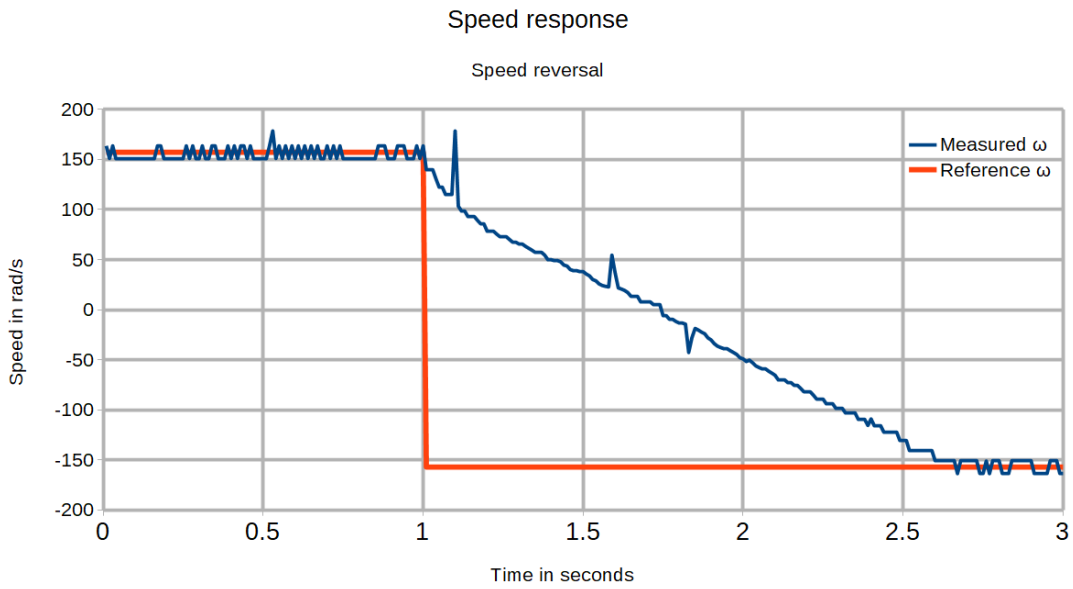


Figure 4.18: Speed response for rated current limit reversal operation

The speed reversal operation with the rated current limit, as shown in Figure 4.18 completes in 1.6s. The peaks in the waveform are likely due to EMI from the motor supply which introduces noise in the encoder reading and causes the bit to flip erroneously.

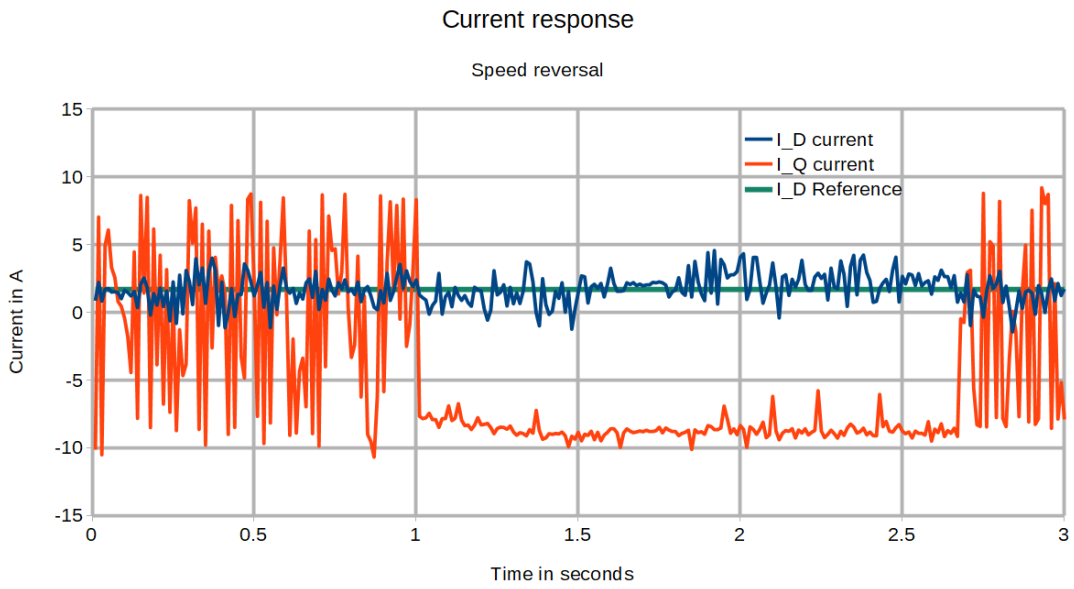


Figure 4.19: Current response for rated current limit reversal operation

Much like in 4.3.4.1, Figure 4.19 shows that the i_d value follows the reference while i_q achieves -8.5A during speed transition and oscillates about 0A on achieving set-point.

4.3.4.3 Braking

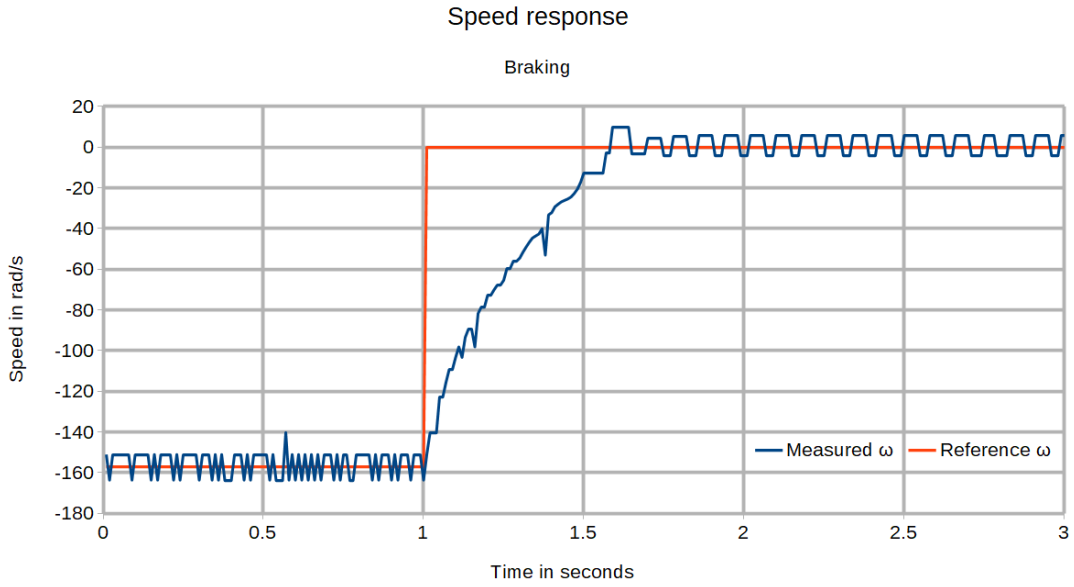


Figure 4.20: Speed response for rated current limit braking operation

As seen in the results obtained in 4.3.3, Figure 4.20 shows that the braking operation concludes faster than the starting operation due to the various factors mentioned earlier.

Further, the rotor does not settle at 0 rad/s due to the way speed is being estimated.

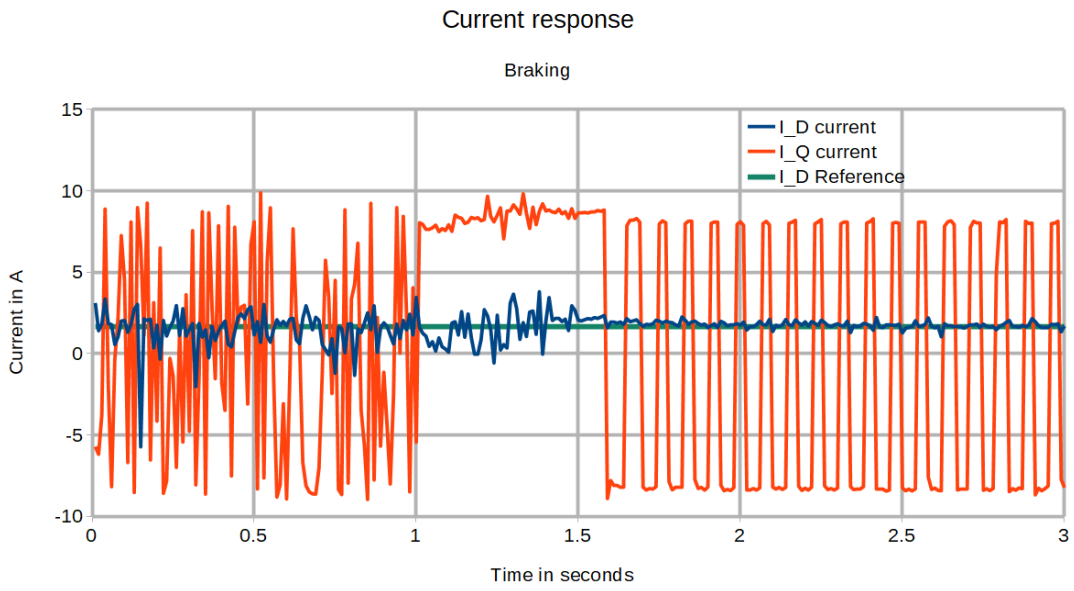


Figure 4.21: Current response for rated current limit braking operation

Much like the earlier cases, Figure 4.21 shows that the i_q current oscillates about 0A when the speed is at set-point while it takes the value of 8.5A when the speed is in transition.

4.3.4.4 Low speed operation

For testing the low speed operation of the motor, the speed reference was set to $\frac{1}{10}^{th}$ of the rated value for starting and the measurements were recorded for 2s after motor achieved set-point. For estimating speed for the low speed case, the 5^{th} MSB was tracked as it was found to give better results for lower speeds.

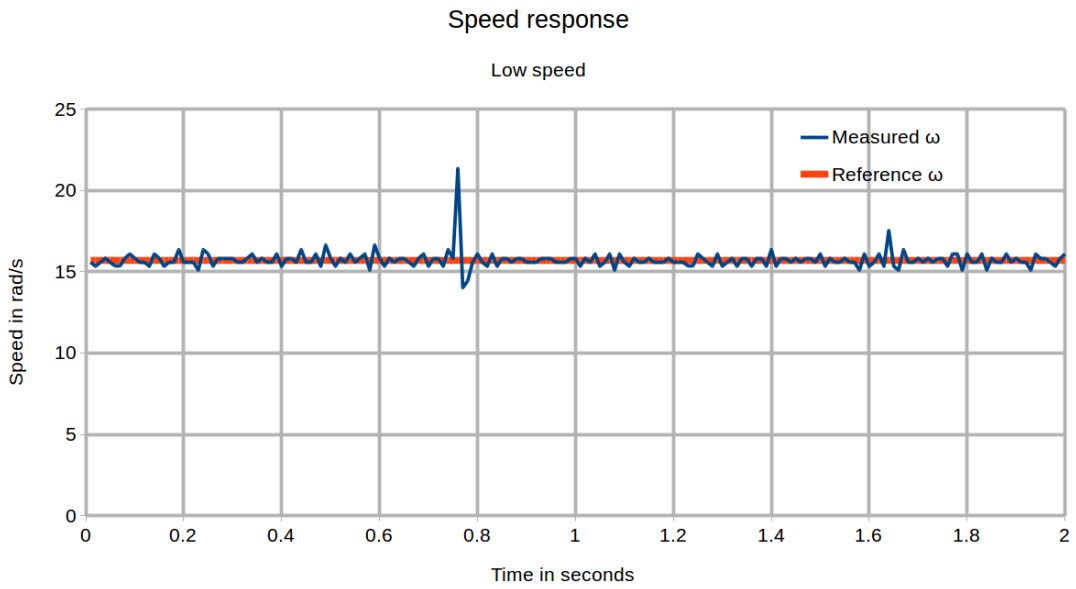


Figure 4.22: Speed response for rated current limit low speed operation

From Figure 4.22, one can see that the motor follows the reference speed quite closely. However, due to the resolution at which the speed is sensed, the motor does not rotate smoothly at the reference speed and one can see that the rotor jerks at regular intervals.

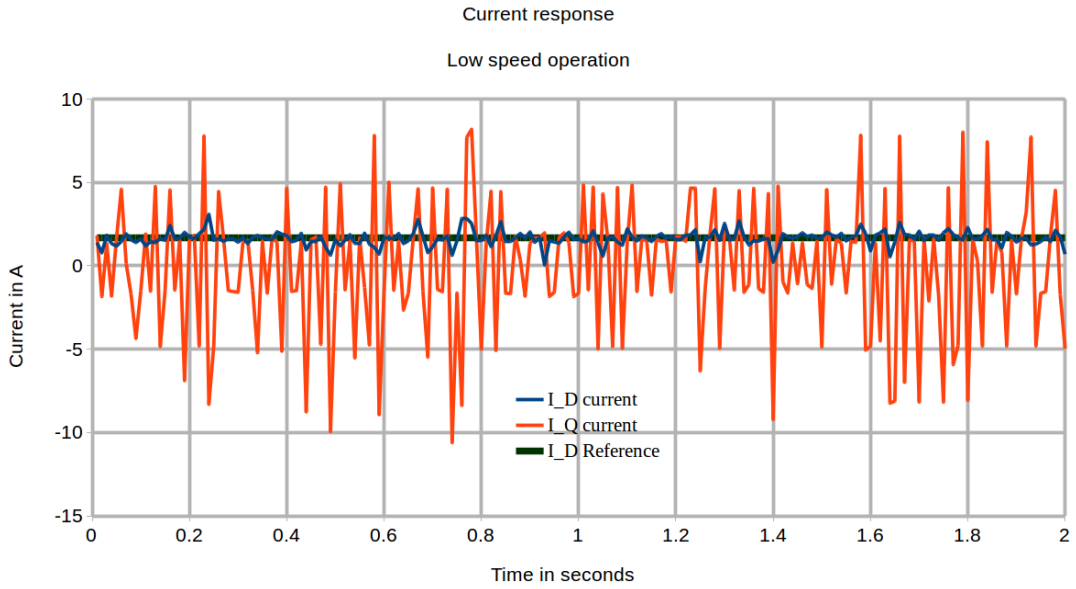


Figure 4.23: Current response for rated current limit low speed operation

The current response, as shown in Figure 4.23 is not very different from the earlier cases. The i_d value follows the reference while the i_q value oscillates about the 0A mark after the set-point is achieved. It is worth noting that the magnitude of oscillations is

lower in the case of low speed operation due to the fact that the speed estimation has a better resolution and that the achieved speed does not deviate from the set-point by a large margin.

4.3.5 Summary and analysis of the results

- The motor model used in simulations does not accurately model the actual motor. As a result, there is a considerable difference in the performance obtained in the simulations and on the actual motor.
- The speed estimation in hardware is not as straightforward as it appears to be and depending on the way the speed is estimated the controller performance can vary.
- Due to the limited resolution offered by the speed estimation method, the speed never stayed constant on reaching the set-point and instead oscillated about the set-point value.
- Although setting the bandwidths of the controllers to high values is recommended to ensure that the controller responds to abrupt changes quickly, it may lead to unexpected behavior if the controller isn't tuned well. Further, high bandwidth also worsens the acoustic performance of the motor.
- The braking operation on hardware occurred faster than starting due to factors like the motor being already energized, the cooling fan windage loss being different depending on the direction of rotation, and most importantly, friction aiding the process.
- From the results obtained, one can also conclude that addition of the decoupling terms does more harm than good to the current response of the controller. The main culprit behind this, however, was the resonant frequency of 250Hz.
- Due to the errors associated with the speed estimation, the low speed performance of this motor is significantly worse than its rated performance. A smarter speed estimation technique involving filter can also improve the low speed performance, however, due to lack of time, such techniques were not experimented with.
- There were differences observed between the simulations and the hardware implementation which can be attributed to the inaccuracies in the parameter estimates, friction being neglected, saturation not being accounted for, and resolution of speed estimation. However, overall, one can conclude that the FOC method itself works as intended and achieves its control objective in the hardware implementation.

CHAPTER 5

CONCLUSION

5.1 Summary of the report

An overview of the design aspects of a SyRM drive were covered in this report.

The report began with a discussion of the history of motor drives and the motivation behind using the SyRM. It was stated that the SyRM offers advantages over the induction motor in terms of efficiency and over the PMSM in terms of cost.

The theoretical aspects of the SyRM modeling were discussed. The SyRM has a very simple model compared to the induction motor and the synchronous motor and thus, in theory, it is easier to design a controller for this machine. The FOC method for designing the controller for the SyRM was discussed with the motivation behind using it being that it allows control of the two axes currents independent of each other and is in a way similar to the DC motor drive. The SVPWM technique was explained and the equations involved in the pulse generation were stated. The theory of the EKF was stated and the way it can be used for state estimation in SyRM was demonstrated. The Field Weakening control of the SyRM was mentioned along with an insight into the different modes of operation and the equations governing the choice of direct axis current reference for these modes.

The working of FOC was shown using MATLAB simulations and the results showed that the transient performance of the motor was limited by the rated torque - which in turn is dependent on the rated current of the motor, and the mechanical parameters. The simulation results also showed that voltage limit enforced by the inverter created a dependence between the decoupled currents when operating at the inverter limit. The sensorless control employing the EKF was simulated and even with limited tuning, reasonable results were obtained. The rated performance of the EKF was shown to be better than the FOC counterpart due to the added advantage of load torque estimation that the EKF offers. The low speed performance of the EKF was analyzed and it was

observed that with proper tuning, the EKF can be used effectively even for low speeds. The numerical and computational issues of implementing the SyRM on the microcontroller were discussed and some guidelines were given for choosing the size of the measurement vector and the sampling interval.

In the final chapter, the hardware implementation details of the controller were given and the importance of the auxiliary components like the rotary encoders, the sensors, the signal conditioning board and the autotransformer was discussed. The working of the signal conditioning board was mentioned in detail with stress upon the level shifting, inverter tripping and analog conditioning operations. The major steps involved in programming the microcontroller were mentioned with particular stress upon the way the PI controller was implemented and the speed estimation method.

During the testing of the FOC on the actual motor, it was noted that a discrepancy exists between the motor model used in Simulink and the actual motor. The controller parameters used for the Simulink failed to make the actual motor run. Similarly the controller parameters obtained after manually tuning the motor did not produce a reasonable response on the Simulink system. The no load tests on the motor at rated speed and voltage carried out by suppressing the current limits showed that not adding the decoupling terms in the current controllers resulted in a better current response compared to adding them. Finally, the starting, speed reversal, braking, and low speed operation of the motor in the no load condition and at rated speed using the FOC method were demonstrated with the current limits set to the rated values.

Despite the discrepancy between the motor model in Simulink and the actual motor, the FOC method was shown to work on the actual motor.

5.2 FUTURE SCOPE

Although the project covers a number of aspects related to the design of a SyRM drive, the insufficiency of time and resources allows for a lot of this work to be carried forward and improved upon.

- The theory behind the Field Weakening method was discussed however the simulations or hardware implementation of the field weakening control were not demonstrated in this project. Although the equations make the control appear trivial to implement, one faces a lot of unexpected results during the actual implementation due to the saturation not being accounted for, or not being dealt with as required.
- The EKF results were suboptimal due to insufficient tuning. One can do a grid search on the tuning parameters and obtain a set of parameters that work best for the motor. One can also try out the High Frequency voltage superposition method to improve the accuracy of the estimates at low speeds.
- Hardware implementation of the EKF, which was not covered in this project, is also a challenging problem and one needs to spend time to achieve the intended accuracy as well as latency when implementing this on hardware.
- The differences between the simulation model and the actual model led to a significant difference between the results obtained in the two cases. The cause behind this can be investigated thoroughly and guidelines can be established to improve the accuracy of the modeling process.
- In terms of research potential, the sensorless techniques for the SyRM have still not been fully explored and while the EKF appears to be a promising alternative, a proper comparison between the sensorless control methods for SyRM has - to the best of the author's knowledge - not appeared in any of the journals.
- Rotor design and manufacture of the SyRM remains a major research area. The motor used for this project was not well designed and hence was quite bulky for its power rating. However, if one can obtain a high saliency ratio while designing the rotor, the motor size can be reduced a great deal and introducing manufacturing techniques that can make the production of rotors easier would greatly boost the commercial prospects of the SyRMs.

APPENDIX A

APPENDIX

A.1 Symbolic math code for obtaining motor parameters

```
p = 2;  %pole pairs
wr = 50*pi;
we = p*wr;
Vs = 415;
Is = 7.32*sqrt(3);
pf = 0.452;
Pin = 2375;
T = 2000/wr;
rs = 1.3;
Ld = 0.713;

syms vd vq id iq Lq B

eq1 = (vd == rs*id - we*Lq*iq);
eq2 = (vq == rs*iq + we*Ld*id);
eq3 = (T == p*(Ld-Lq)*id*iq - B*wr);
eq4 = (vd*id + vq*iq == Vs*Is*pf);
eq5 = (vd^2 + vq^2 == Vs^2);
eq6 = (id^2 + iq^2 == Is^2);
eq7 = (Lq>0);

sol = solve([eq1, eq2, eq3, eq4, eq5, eq6,eq7], [vd vq id
    iq Lq B], 'Real',true);

```

```
24 disp(vpa(sol.vd));
25 disp(vpa(sol.vq));
26 disp(vpa(sol.id));
27 disp(vpa(sol.iq));
28 disp(vpa(sol.Lq));
29 disp(vpa(sol.B));
```

A.2 Manufacturer's test results

Voltage (V)	Current (A)	Power (W)	P.F.	Speed (RPM)
415 (λ)	4.325	228	0.0726	1500
230 ($\triangle$)	7.523	310	0.0982	1500

Table A.1: No Load Tests

Voltage (V)	Current (A)	Power (W)	P.F.	Speed (RPM)	Torque (Kgm)
415 (λ)	7.31	2375	0.452	1500	1.30
230 ($\triangle$)	12.887	2400	0.448	1500	1.32

Table A.2: Full Load Tests

A.3 Code repository

https://github.com/SAM10795/DDP_SyRM

REFERENCES

[1] **ABB** (Accessed Apr 2019). Synchronous reluctance motors. URL https://new.abb.com/motors-generators/iec-low-voltage-motors/process-performance-motors/synchronous-reluctance-motors. (document), 1.1

[2] **ABB** (Accessed Apr 2019). Viking line takes the tension out of harbor mooring for m/s gabriella. URL https://new.abb.com/drives/media/viking-line-takes-the-tension-out-of-harbor-mooring-for-m-s-gabr 1.2.1

[3] **Alves dos Santos Junior, J.**, **D. Alves Andrade**, **G. P. Viajante**, **M. A. Arantes de Freitas**, and **V. R. Bernadeli** (2015). Analysis and mathematical modeling of the synchronous reluctance motor. *IEEE Latin America Transactions*, **13**(12), 3820–3825. ISSN 1548-0992.

[4] **Boglietti, A.**, **A. Cavagnino**, **M. Pastorelli**, **D. Staton**, and **A. Vagati** (2006). Thermal analysis of induction and synchronous reluctance motors. *IEEE Transactions on Industry Applications*, **42**(3), 675–680. ISSN 0093-9994. 1.2.1

[5] **Boldea, I.**, **Z. X. Fu**, and **S. A. Nasar**, Performance evaluation of axially-laminated anisotropic (ala) rotor reluctance synchronous motors. *In Conference Record of the 1992 IEEE Industry Applications Society Annual Meeting*. 1992. 2.1

[6] **Carlet, P. G.**, **F. Tinazzi**, **S. Bolognani**, and **M. Zigliotto** (2019). An effective model-free predictive current control for synchronous reluctance motor drives. *IEEE Transactions on Industry Applications*, 1–1. ISSN 0093-9994. 1.3

[7] **Caro, R.**, **C. A. Silva**, **R. PÃ©rez**, and **J. I. Yuz**, Sensorless control of a synrm for the whole speed range based on a nonlinear observability analysis. *In 2017 IEEE International Conference on Industrial Technology (ICIT)*. 2017. 3.2.2.4

[8] **Du, P.**, **L. Li**, **J. Liu**, and **R. Yang**, A novel simplified 3-level svpwm modulation method based on the conventional 2-l svpwm modulation method. *In 2018 21st International Conference on Electrical Machines and Systems (ICEMS)*. 2018. 2.3

[9] **Hong, H.**, **H. Liu**, **S. Cho**, **J. Lee**, and **C. Jin** (2017). Design of high-end synchronous reluctance motor using 3-d printing technology. *IEEE Transactions on Magnetics*, **53**(6), 1–5. ISSN 0018-9464. 1.3

[10] **Jlassi, I.** and **A. J. Marques Cardoso**, Model predictive current control of synchronous reluctance motors, including saturation and iron losses. *In 2018 XIII International Conference on Electrical Machines (ICEM)*. 2018. ISSN 2381-4802. 1.3

[11] **Kun, W.**, **Y. Xiaojie**, **W. Chenchen**, and **Z. Minglei**, An equivalent dual three-phase svpwm realization of the modified 24-sector svpwm strategy for asymmetrical dual stator induction machine. *In 2016 IEEE Energy Conversion Congress and Exposition (ECCE)*. 2016. 2.3

[12] **Lavrinovicha, L.** and **J. Dirba**, Comparison of permanent magnet synchronous motor and synchronous reluctance motor based on their torque per unit volume. *In 2014 Electric Power Quality and Supply Reliability Conference (PQ)*. 2014. 1.2.1

[13] **Lee, J.**, **D. Jung**, **K. Lee**, **W. Jin**, **G. Lee**, **J. Lee**, and **Y. J. Oh** (2019). A study on analysis of synchronous reluctance motor considering axial flux leakage through end plate. *IEEE Transactions on Magnetics*, 1–4. ISSN 0018-9464. 1.3

[14] **Li, C.**, **G. Wang**, **G. Zhang**, **D. Xu**, and **D. Xiao** (2019). Saliency-based sensorless control for synrm drives with suppression of position estimation error. *IEEE Transactions on Industrial Electronics*, **66**(8), 5839–5849. ISSN 0278-0046. 1.3

[15] **Lin, F.**, **M. Huang**, **S. Chen**, and **C. Hsu** (2019). Intelligent maximum torque per ampere tracking control of synchronous reluctance motor using recurrent legendre fuzzy neural network. *IEEE Transactions on Power Electronics*, 1–1. ISSN 0885-8993. 1.3

[16] **Mabuchi, T.**, **A. Makimura**, **M. Tomita**, **S. Kondo**, **M. Hasegawa**, **S. Doki**, and **S. Kato**, Position sensorless control of synchronous reluctance motors at very low speeds region using high-frequency current control system. *In 2017 20th International Conference on Electrical Machines and Systems (ICEMS)*. 2017. 1.3

[17] **Manzolini, V.**, **D. Da Ru**, and **S. Bolognani** (2019). An effective flux weakening control of a syrm drive including mtpv operation. *IEEE Transactions on Industry Applications*, **55**(3), 2700–2709. ISSN 0093-9994. 1.3

[18] **Matyska, P.**, Advantages of synchronous reluctance motors. *In Transactions on Electrical Engineering*, volume 3. 2014. 1.2.1

[19] **Mikami, H.**, **K. Ide**, **Y. Shimizu**, **M. Senoo**, and **H. Seki**, Historical evolution of motor technology. Hitachi, 2011. URL http://www.hitachi.com/rev/pdf/2011/r2011_01_107.pdf. 1

[20] **Mikey likes mountains** (Accessed Apr 2019). Voltage phasors in the svpwm. URL https://commons.wikimedia.org/w/index.php?curid=5882951. (document), 2.6

[21] **Moghaddam, R. R.** (2011). Rotor for a synchronous reluctance machine. URL https://patents.google.com/patent/US20120062053. US Patent US20120062053A1. (document), 2.1

[22] **Mun, S. J.**, **Y. H. Cho**, and **J. H. Lee** (2008). Optimum design of synchronous reluctance motors based on torque/Volume using finite-element method and sequential unconstrained minimization technique. *IEEE Transactions on Magnetics*, **44**(11), 4143–4146. ISSN 0018-9464. 1.3

[23] **Nguyen, D.**, **L. Loron**, and **K. Dakhouche**, High-speed sensorless control of a synchronous reluctance motor based on an extended kalman filter. *In 2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe)*. 2015. 1.3

[24] **Niedworok, A.** and **Å. Orzech**, Determination of efficiency and thermal properties of drive equipped with induction motor and drive equipped with synchronous reluctance motor. *In 2017 19th European Conference on Power Electronics and Applications (EPE'17 ECCE Europe)*. 2017. 1.2.1

[25] **Pal, S.** and **K. Sonawane**, Electric motor market size, share & trends | industry analysis, 2022. Allied Market Research, 2017. URL https://www.alliedmarketresearch.com/electric-motor-market. 1

[26] **Petteri Aimonen** (Accessed Apr 2019). Graphical depiction of the kalman filter. URL https://commons.wikimedia.org/w/index.php?curid=17475883. (document), 2.7

[27] **Riley, C. P.**, **N. P. Sawant**, **D. Ilea**, **A. Venskus**, **A. M. Bedford**, **A. M. Michaelides**, **X. Vinamata**, **P. J. Topping**, **J. M. Soler-Vizan**, **E. Odvarka**, **J. D. Wale**, **D. J. Hawke**, and **J. M. Reeve**, Simulation based design of reluctance motors for traction applications in hybrid and electric vehicles. *In 7th IET International Conference on Power Electronics, Machines and Drives (PEMD 2014)*. 2014. 1.3

[28] **Son, B.**, **J. Kim**, **J. Kim**, **Y. Kim**, and **S. Jung** (2019). Adaptive particle swarm optimization based on kernel support vector machine for optimal design of synchronous reluctance motor. *IEEE Transactions on Magnetics*, 1–5. ISSN 0018-9464. 1.3

[29] **Sul, S.**, *Position/Speed Sensorless Control of AC Machines*. IEEE, 2011. ISBN 9780470876541. URL https://ieeexplore.ieee.org/document/5675910. 1.1

[30] **Vinoth Kumar, K.**, **P. Angel Michael**, **J. P. John**, and **D. S. Suresh Kumar**, Simulation and comparison of spwm and svpwm control for three phase inverter. *In ARPN Journal of Engineering and Applied Sciences*, volume 5. 2010. 2.3

[31] **Vivek, G.** and **J. Biswas**, Study on hybrid svpwm sequences for two level vsis. *In 2017 IEEE International Conference on Industrial Technology (ICIT)*. 2017. 2.3

[32] **what-when how.com** (). Inverters (induction motor). URL http://what-when-how.com/induction-motor/inverters-induction-motor. (document), 2.5

[33] **Wikipedia** (Accessed Apr 2019). Electric motor - comparison by major categories. URL https://en.wikipedia.org/wiki/Electric_motor#Comparison_by_major_categories. (document), 1.2

[34] **Yano, M.**, **S. Abe**, and **E. Ohno**, History of power electronics for motor drives in japan. Proc. IPEC Rec., 2005. URL https://ethw.org/w/images/4/49/Yano2.pdf. 1.1

[35] **Yousefi-Talouki, A.**, **P. Pescetto**, and **G. Pellegrino** (2017). Sensorless direct flux vector control of synchronous reluctance motors including standstill, mtpa, and flux weakening. *IEEE Transactions on Industry Applications*, **53**(4), 3598–3608. ISSN 0093-9994. 1.3

[36] **Zhang, Z.**, **K. J. Jhong**, **C. Cheng**, **P. Huang**, **M. Tsai**, and **W. Lee**, Metal 3d printing of synchronous reluctance motor. *In 2016 IEEE International Conference on Industrial Technology (ICIT)*. 2016. 1.3