ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

OFFLINE STATOR RESISTANCE ESTIMATION FOR PERMANENT MAGNET SYNCHRONOUS MOTOR AND REAL-TIME IMPLEMENTATION USING EMBEDDED CODER

M.Sc. THESIS

Lutfu Emre EFE

Department of Electrical Engineering

Electrical Engineering Programme

JUNE 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

OFFLINE STATOR RESISTANCE ESTIMATION FOR PERMANENT MAGNET SYNCHRONOUS MOTOR AND REAL-TIME IMPLEMENTATION USING EMBEDDED CODER

M.Sc. THESIS

Lutfu Emre EFE
(504191028)

Department of Electrical Engineering

Electrical Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Salih Baris OZTURK

JUNE 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

KALICI MIKNATISLI SENKRON MOTORLAR İÇİN OFFLINE STATOR DİRENCİ TAHMİNİ VE EMBEDED KODER KULLANARAK GERÇEK ZAMANLI UYGULAMASI

YÜKSEK LİSANS TEZİ

Lutfu Emre EFE
(504191028)

Elektrik Mühendisliği Anabilim Dalı

Elektrik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Salih Barış ÖZTÜRK

HAZİRAN 2023

Lutfu Emre EFE, a M.Sc. student of ITU Graduate School student ID 504191028, successfully defended the thesis entitled "OFFLINE STATOR RESISTANCE ESTIMATION FOR PERMANENT MAGNET SYNCHRONOUS MOTOR AND REAL-TIME IMPLEMENTATION USING EMBEDDED CODER", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Salih Baris OZTURK** Istanbul Technical University	....
Jury Members :	**Assoc. Prof. Dr. Derya Ahmet KOCABAS** Istanbul Technical University	....
	Assoc. Prof. Dr. Omer Cihan KIVANC Istanbul Okan University	....

Date of Submission : 14 May 2023
Date of Defense : 19 June 2023

To Melisa,

FOREWORD

First of all, I would like to thank my advisor, Assoc. Prof. Dr. Salih Baris Ozturk, for his continuous support, patience, help, understanding, and willingness throughout the period of the research. I am profoundly grateful for the valuable time that my advisor has dedicated to me. Working alongside such a highly knowledgeable and perceptive professor has been an incredible privilege. His wisdom and insight have been invaluable to me, and I am truly thankful for the opportunity to collaborate with him.

I would also like to thank my thesis committee members Assoc. Prof. Dr. Omer Cihan Kivanc and Assoc. Prof. Dr. Derya Ahmet Kocabas for generously dedicating their valuable time to attend my presentation.

I also need to thank my dear friends Ilayda DEMIRCIOGLU, Ayse YILMAZ, Kubra Nur KULAOGLU, Ilknur AKTEMUR, Esen Bahar ASLANTEPE, Beste GER, Hilal OZTURK, and Melih BOZ for their endless support and our amazing friendship.

I would also like to thank Selim Taskin TEMIZSOY and Sade TEMIZSOY, who has a very special place for me, for making me feel that they are always by my side.

I would like to express heartfelt gratitude to my dear family members, my mother Saniye EFE, My father Oguz EFE, and my little sister Cemre EFE, for their endless support and presence in my life. I consider myself incredibly fortunate to have you by my side. Your endless support and existence mean the world to me.

June 2023

Lutfu Emre EFE
(Mechatronics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

PMSM	**:**	Permanent Magnet Synchronous Motor
BLDC	**:**	Brushless DC Motor
PU	**:**	Per-Unit System
SI	**:**	International System of Units
SVPWM	**:**	Space Vector Pulse Width Modulation
ANN	**:**	Artificial Neural Network
AC	**:**	Alternating Current
DC	**:**	Direct Current
ADC	**:**	Analog-to-Digital Converter
FOC	**:**	Field Oriented Control

SYMBOLS

$\mathbf{R_s}$	: Stator Resistance
L	: Inductance
$\mathbf{V}_{abc}$	: Phase voltages
$\mathbf{V}_{\alpha\beta}$	: $\alpha\beta$-axis voltages
$\mathbf{V}_{dq}$	: dq-axis voltages
$\mathbf{i}_{abc}$	: Phase currents
$\mathbf{i}_{\alpha\beta}$	: $\alpha\beta$-axis currents
$\mathbf{i}_{dq}$	: dq-axis currents
$\omega_{\mathbf{e}}$	: Electrical angular velocity
$\omega_{\mathbf{r}}$	: Mechanical angular velocity
$\Theta_{\mathbf{e}}$	: electrical angular position
$\mathbf{R_0}$	: Resistance of a conductor at the temperature T_0
$\mathbf{R}$	: Resistance of a conductor at the temperature T
$\boldsymbol{\alpha}$	: Temperature Coefficient
λ	: Permanent magnet flux linkage
p	: Number of pole pairs for a PMSM

LIST OF TABLES

Page

LIST OF FIGURES

Page

OFFLINE STATOR RESISTANCE ESTIMATION FOR PERMANENT MAGNET SYNCHRONOUS MOTOR AND REAL-TIME IMPLEMENTATION USING EMBEDDED CODER

SUMMARY

This thesis presents offline stator resistance estimation for PMSM to design a sensorless control algorithm using a sliding mode observer (SMO) at low speeds. The precise value of permanent magnet synchronous motor (PMSM) parameters is crucial for modern electrical drives to design high-performance controllers, estimators, and observers. Stator resistance can be easily measured using an ohmmeter, but this is not preferred due to the advantages of the automated test method. Testing stator resistance using an inverter is beneficial since it also considers cable and power semiconductor resistances. Those resistances physically exist and affect the drive operation. Due to the inherent error-proneness of commonly used speed/position, sensorless control systems at low speeds are especially crucial for low-speed operation. The stator resistance may be measured by injecting a known, constant voltage and then monitoring the resulting current amplitude. It can also be achieved by injecting a known constant current and observing the resulting voltage. While an optimized current controller is not a condition for open-loop constant voltage injection, it does have drawbacks. Choosing a proper voltage amplitude for the test is an essential practical issue. If the resistance is too low, the test voltage may result in an over-current problem. It may also result in a very low current, preventing reliable data collection. Because the current controller automatically determines the injection voltage, injecting a DC current does not have these issues or related time-consuming iterations. Estimation processes typically aim to estimate parameters at a standstill. Since the rotor movement might not be permitted in some practice applications, this is highly desired. The calculation of stator resistance involves steady-state current excitation, which can cause rotor movement and torque generation. Therefore, in synchronous motors, stator resistance is estimated by injecting the constant current in the motor d–axis. It does not create torque and hence an undesirable and uncontrolled rotor motion. The only condition is to keep the angle at a constant position throughout the test. Offline stator resistance estimation for a sliding mode observer-based PMSM speed control algorithm has been experimentally verified using MATLAB®/Simulink® and Simscape Electrical™. The experimental kit includes a low-cost evaluation and development board LAUNCHXL-F28069M with a C2000 real-time microcontroller. DRV8301 has been chosen as an inverter board due to the compatibility of the nominal current and voltage ranges of the 52.5 Watt PMSM. Thanks to the code generation future of the Embedded Coder® toolbox, experimental results show that the estimated resistance is more accurate compared with the stator resistance value measured by an expensive RLC meter.

KALICI MIKNATISLI SENKRON MOTORLAR İÇİN OFFLINE STATOR DİRENCİ TAHMİNİ VE EMBEDED KODER KULLANARAK GERÇEK ZAMANLI UYGULAMASI

ÖZET

Bu tez, düşük hızlarda kayan mod gözlemcisi (SMO) kullanarak sensörsüz bir kontrol algoritması tasarlamaak gerektiği zaman gerekli olan stator direncinin tahmin edilmesi için stator direnci tahmin algoritmasını sunar. Kalıcı mıknatıslı senkron motor (PMSM) parametrelerinin kesin değeri, modern elektrikli sürücülerin yüksek performanslı kontrolörler, tahmin ediciler ve gözlemciler tasarlamnası için çok önemlidir. Stator direnci bir ohmmetre kullanılarak veya RLC metre kullanarak kolayca ölçülebilir, ancak otomatik test yönteminin avantajları nedeniyle bu tercih edilmez. Çünkü stator direncinin motorun sıcaklığının değişmesi ile sürekli değişmesi, direnç değerinin zaman zaman tekrar alınması gerekliliğini ortaya koyar. Bu ölçüm cihazları yerine, stator direncini bir invertör kullanarak test etmek, kablo ve güç yarı iletken dirençlerini de dikkate aldığından faydalıdır. Bu dirençler fiziksel olarak mevcuttur ve sürücü çalışmasını etkiler. Bu sebeple stator direnci tahmin algoritması önem arz etmektedir.

Senkron motor kullanımındaki amaç ise yüksek tork hacim oranına sahip olması, yüksek verimliliğe sahip olması, fırçalı DC motora göre daha az bakım gerektirmesi gibi avantajlar sayılabilir.

Stator direnci tahmini farklı yöntemler kullanılarak yapılabilir. Bu yöntemler genel hatlarıyla çevrimiçi tahmin algoritmaları ve çevrimdışı tahmin algortimaları olmak üzere iki farklı gruba ayrılmaktadır. Çevrimiçi tahmin algoritmaları, kontrol sisteminin her çevrimininde anlık olarak stator direncini ölçmektedir. Ölçülen bu stator direnç değeri, sistemin sahip olduğu kontrolcüleri otomatik olarak güncellemekte ve varsa sensörsüz algoritmaların katsayılarını da otomatik olarak güncelleyerek, sistemin veriminin yükseltilmesi amaçlanmaktadır. Bütün bunlar büyük bir avantaj oluşturmasına karşın, bu algoritmalar aynı zamanda yüksek işlem gücüne sahip mikroişlemcilere ihtiyaç duyarlar. Buna ek olarak da çok karmaşık kontrol yapılarına sahiptirler. Tüm bu özellikler göz önüne alındığında çevrimdışı stator tahmini yöntemine göre dezavantajları ortaya çıkmaktadır. Çevrimdışı stator tahmini yöntemleri ise, çevrimiçi stator tahmini yöntemlerine göre daha basit kontrol algoritmalarına sahiptirler. Stator direnci tahminini genellikle motor ilk başta çalışacağında hareket halinde değilken veya motor dur kalk yaptığı durumlarda tahmin yapmaktadır. Bu algoritmalar çamaşır makinesi, elektirkli veya hibrit araba gibi sistemlerde oldukça kullanışlıdır. Ayrıca yüksek işlem gücüne sahip mikroişlemcilere ihtiyaç duymazlar. Ölçülen stator direnci değeri, sıcaklıkla değişmektedir ve kontrolcü parametrelerinin belirlenmesinin yanısıra sensörsüz uygulamarda da önem arz etmektedir. Başlangıçta kontrol algoritmasının tasarımını yapmak içinde de bu bilgiye ihtiyaç vardır. Bu sebeple algoritmayı tasarlamak için stator direnci değeri RLC metre ile ölçülerek kullanılmıştır.

Yaygın olarak kullanılan hızın/konumun doğasında bulunan hataya yatkınlığı nedeniyle, düşük hızlardaki sensörsüz kontrol sistemleri, özellikle düşük hızlı çalışma için çok

önemlidir. Stator direnci, bilinen, sabit bir voltaj enjekte edilerek ve ardından ortaya çıkan akım genliği izlenerek ölçülebilir. Bilinen bir sabit akım enjekte edilerek ve ortaya çıkan voltajı gözlemleyerek de elde edilebilir. Bilinen bir gerilim enjekte edilmesinin açık çevrim olması sebebi ile, yani bir kontrolcüye ihtiyaç duymaması sebebi ile avantaj sahibi olduğu düşünülebilir ancak, sahip olunan sisteme zarar verebilecek büyük dezavantajlara sahiptir. Dezavantajı detayllıbir şekilde açıklamak gerekirse test için uygun bir voltaj genliği seçmek, önemli bir pratik konudur. Eğer direnç çok düşükse, test voltajı aşırı akım sorununa neden olabilir ve sistemde kalıcı hasara sebep olabilir. Eğer direnç çok yüksek ise güvenilir veri toplamayı engelleyen çok düşük bir akıma neden olabilir. Bu sebeple açık çevrim gerilim enjekte etme yöntemi mantıklı bir seçenek değildir. Öte yandan akım enjekte etme yöntemi kullanıldığı zaman, akım kotrolcüsü enjeksiyon voltajını otomatik olarak belirlediğinden, bu sorunlar meydana gelmemektedir. Bu sebeple çevrimdışı stator direnci ölçmek için, DC akım enjekte etme yöntemi tercih edilmiştir. Bu yöntemin getirdiği en önemli dezavantaj ise yukarıda da bahsedildiği gibi kontrolcü tasarlanması gerekmesidir.

Kalıcı mıknatıslı senkron motorları kontrol etmek için alan yönlendirmeli kontrol algoritması kullanılmaktadır. Bu yöntemde trapezioidal kontrol yönteminin aksine, stator ve rotor manyetik alanlarının arasındaki açının sabit 90 derece olması sebebiyle motordan daima maksimum tork çıkışı elde edilmektedir.Bu alan yönlendirmeli kontrol yönteminin önemli avantajlarından bir tanesidir. Bu yöntemin en önemli avantajlarından bir tanesi de üç fazlı olarak üretilen bir AC motorun, bu algoritma kullanılarak DC bir motor gibi kontrol edilebilmesini sağlamaktır.Bunun için farklı dönüştürme yöntemleri kullanılır. İlk olarak clark dönüşümü kullanılarak, motorun çıkışında ölçülen üç fazlı ve aralarında 120 derece faz farkı bulunan sinus formuna sahip akım sinyali, aralarında 90 derece faz farkı bulunan ve iki fazlı sinus formuna sahip bir sinyale dönüşmüş olur. daha sonra park dönüşümü kullanılarak 2 fazlı sinus formuna sahip bu sinyal DC bir sinyale dönüşmektedir. park dönüşümünde ek olarak motorun konum bilgisi de kullanılmaktadır. Bu sebeple motorun konumunun elde edilmesi çok önemli bir konudur. DC sinyal d ve q ekseni olarak adlandırılmaktadır. d ekseni akıyı temsil ederken q ekseni ise akımı yani torku temsil etmektedir. Akım kontrolcüsü tasarlamak istediğimiz için d eksenine referans değeri olarak sıfır değeri verilmektedir. Hem d hem de q ekseni kontrolcü parametreleri kullanılan simulasyon ortamı sayesinde ayarlandıktan sonra DC sinyaller tekrar AC sinyallere çevrilmelidir. Bunun için öncelikle ters park dönüşümü kullanılır. Normal park dönüşümünde olduğu gibi ters park dönüşümünde de motor konum bilgisi önemlidir. Ardından kullanılan PWM üretme yöntemine göre ters clarke dönüşümü kullanılır. Eğer sinusoidal darbe genişlik modulasyon yöntemi kullanılıcak ise ters, ters clarke dönüşümü kullanılır. Eğer uzay vektör darbe genişlik modulasyon yöntemi kullanılacak ise ters clarke dönüşümü yöntemi kullanılmasına gerek yoktur. Uzay vektor darbe genişlik modulasyonu kullanıldığında DC barada bulunan gücün tamamının kullanılmasına olanak sağladığı için bu tezde bu yöntem tercih edilmiştir. Sinusoidal PWM yöntemi kullanıldğında, DC baradaki gücün yaklaşık yüzde 86'lık bir kısmı kullanılabilmektedir. Bu durum sebebiyle bu yöntem tercih edilmemiştir. SVPWM yönteminin tüm basamakları detaylı bir şekilde açıklanmıştır.

Akım kontrolcüsü tasarımı bittikten sonra hız kontrolcüssü tasarımı zorunlu değildir. Ancak gelecekteki çalışmalarda kullanılmak üzere bu tezde hız kontrolcüsü tasarımı da gerçekleştirilmiştir. Bu kontrolcüler kaskat yöntemine göre tasarlanmaktadır. Akım kontrolcsünün hızı, hız kontrolcüsünün hızından en az 5 kat fazla olması gerekmektedir.

Bu tezde tasarlanmamış olup, eğer bir pozisyon kontrolcüsüne ihtiyacımız olsaydı, pozisyon kontrolcüsnün hızı da, hız kontrolcsünün hızından en az 5 kat daha hızlı olması gerekirdi. Hız kontrolcüsünün çıkışı q ekseninin girişi olmaktadır. Alan yönlendirmeli kontrol algoritmasının yanı sıra d eksenine negatif referans vererek motorun nominal hızından daha yüksek hızlara çıkması da sağlanabilir. Bu duruma alan zayıflatma yöntemi denir. Hızdan kazanç vardır ancak torktan kayıp olur.

Stator direnci tahmini kısmına geldiğimizde ise tahmin süreçleri tipik olarak motorun durağan durumundaki parametreleri tahmin etmeyi amaçlar. Bazı pratik uygulamalarda rotor hareketine izin verilmeyebileceğinden, bu çok tercih edilir. Stator direncinin hesaplanması, rotor hareketine ve tork oluşumuna neden olabilecek sabit durum akım uyarımını içerir. d ekseni akıyı temsil ederken q ekseni tork olarak temsil edilir. Bu nedenle senkron motorlarda stator direnci, d eksenine sabit akım enjekte edilerek tahmin edilir. q eksenine referans değeri olarak motorun hareket etmesini engellemek amacı ile sıfır değeri verilir. Tork ve dolayısıyla istenmeyen ve kontrolsüz bir rotor hareketi yaratmaz. d eksenine verilen test akımı sonucunda bir gerilim elde edilir. Bu gerilim değeri sistem gerçek zamanlı bir deney kitinde gerçeklendiği zaman gürültülere sahip olacaktır. Bu sebeple gürültülü sinyalden maksimum derece de doğtu ölçü alabilmek adına çok fazla sayıda örnek alınır ve alınan örneklerin toplamı, alınan örnek sayısına bölünür ise elde edilen değerin ortalaması alınmış olur ve gerçeğe en yakın değer elde edilmiş olunur. Burada dikkat edilen önemli bir kısım ise alınan örneklerin gerilimin transient zamanda değil steady-state zamanda alınmasıdır. Çünkü transient zamanda alınan veriler ile doğru bir tahmin yapılması mümkün değildir. Bir diğer önemli koşul ise, test boyunca açıyı sabit bir konumda tutmaktır. Motor zaten durağan konumda olacağı için bu koşul kesinlikle sağlanmak zorundadır. Simulasyon ortamında rotor açısı 0 derece olarak verilmiştir.

Kayan kipli gözlemci tabanlı bir PMSM hız kontrol algoritması için çevrimdışı stator direnci tahmini, MATLAB®/Simulink® ve Simscape Electrical™ kullanılarak deneysel olarak doğrulanmıştır. Daha sonra ise kod üretme desteği sayesinde üretilen kod mikroişlemiciye gömülmüştür. Seçilen deney kiti ile, C kodu üretecek yazılımın uyumlu bir şekilde çalışması gerekmektedir. Bu sebeple gerçek zamanlı uygulama yapmak için deney seti olarak, Texas Instruments C2000 gerçek zamanlı mikrodenetleyici ile düşük maliyetli bir değerlendirme ve geliştirme kartı olan LAUNCHXL-F28069M seçilmiştir. 52,5 Watt PMSM'nin nominal akım ve gerilim aralıklarının uyumlu olması nedeniyle inverter kartı olarak BoostXL-DRV8301 seçilmiştir. Seçmiş olduğumuz motorun datasheet'inde stator direnç değeri bulunmamatadır. Bu da yapılan çalışmanın önemini ortaya koymaktadır. Ayrıca Seçilen mikroişlemci kartı ile aynı anda iki motor kontrol etmek mümkündür. C kode üretim aşamasının önemli avantajlarından bir tanesi de manuel olarak kod yazma sırasında meydana gelebilecek hata riskini ortadan kaldırmaktır. Bu avantajı da kullanarak, Embedded Coder® araç kutusunun kod oluşturma yeteneği sayesinde, deneysel sonuçlar, tahmini direncin, hassas bir RLC ölçer tarafından ölçülen stator direnci değerine kıyasla daha doğru olduğunu göstermektedir. Daha doğru olmasının sebebi ise sistemin sadece stator direnci içermemesi, buna ek olarak kablo direnci, inverter üzerindeki mosfetlerin direnci gibi farklı dirençlerin de ölçülüyor olmasıdır. Sonuç olarak RLC metre ile ölçülen ve tasarlanan algoritma ile ölçülen stator direnci değerleri karşılaştırılmış olup, akım çevrimi kontrolcü tasarımı ve gözleyici tasarımı gibi adımlarda kullanılabilir.

1. INTRODUCTION

The introduction part includes the purpose of the thesis, literature review, and thesis organization respectively.

1.1 Purpose of the Thesis

Nowadays, self-commissioning of AC machines is a crucial topic since some of the motor parameters can not be known exactly sometimes and these parameters are important in order to design control algorithms and observers. Therefore, one of the steps of self-commissioning is stator resistance estimation. The reason for being crucial is that, by updating controller parameters or/and updating observer gain parameters, Thanks to this way, the efficiency of the motor can be increased. To estimate stator resistance there are different ways but to implement it in real hardware, the method should be easy and cost-effective. So, a cost-effective development kit has been used to estimate the stator resistance. Firstly, the algorithm has been designed in a simulation area. Then the experiments has been made by using this development kit and the real stator resistance value, datasheet value, and the value which is measured by RLC has been compared.

1.2 Literature Review

Nowadays, because of the electric vehicle and renewable energy industries, e-drives play an important role in energy conversions. They are responsible for almost half (%46) of the energy produced [1]. Due to the increasing importance of green energy and reducing carbon emissions [2] [3] [4] regulatory bodies have imposed stricter efficiency standards. This has resulted in manufacturers of electric drives being compelled to enhance the efficiency of their devices [5]. Consequently, power converter manufacturers are focusing more on energy-efficient algorithms and ensuring that control systems are properly adjusted. This necessitates accurate knowledge about the parameters of motor drives [6], [7].

Furthermore, the effectiveness of sensorless control methods, which have become increasingly popular in recent years, is heavily reliant on the accuracy of motor parameters. If the provided motor parameters are imprecise, the estimators used in these techniques may reduce the overall efficiency of the system [8]. The conventional approach of using a predetermined set of constants may not be suitable when the parameters of motor drives fluctuate due to changing operating conditions. This issue is particularly significant for self-commissioning drives that must operate motors with unknown parameters and establish them through a sequence of tests [9], [10]. Therefore, the identification of motor drive parameters has garnered increasing attention from researchers, and various techniques in this field have been recently proposed.

The primary factor distinguishing parameter estimation algorithms is their mode of operation, which can be classified as online or offline methods. Online algorithms function simultaneously with device operation and can monitor changes in parameters during the working cycles of motor drives. They are well-suited for systems that operate continuously for extended periods, but their implementation and tuning are challenging tasks. Additionally, they typically necessitate powerful microcontrollers and may not be feasible for low-cost systems.

In contrast, offline algorithms operate before the start of working cycles and determine system parameters only for conditions at that specific moment in time. These techniques cannot monitor changes in motor parameters during operation, making them suitable for motor drives that operate in start/stop modes, such as lifting mechanisms or washing machines. Offline algorithms are simpler than online techniques and usually do not require complex tuning and adjustments.

When estimating motor parameters, two critical ones are the stator resistance and the voltage drop across inverter switches. These parameters are particularly important for motor drives that operate at low speeds or frequently start. Accurate knowledge of the stator resistance reduces speed and position estimation errors of non-injective estimators at lower speeds where back electromotive force is low. This allows for open-loop starting and earlier switching to closed-loop mode, leading to an overall increase in drive efficiency [11], [12], [13].

Motor temperature monitoring is another difficult task that requires accurate measurement of stator resistance and the voltage drop across switches [14], [15]. Several algorithms have been proposed in recent research [16], [17], [18] to address this challenge, but they all rely on the principle that the resistance of a conductor varies with temperature, as described by the Equation 1.1.

$$R = R_0(1 + \alpha(T - T_0)) \tag{1.1}$$

According to Equation 1.1, the increase in resistance with temperature is relatively small, at around 4% per 10 degrees centigrade. This highlights the importance of precise measurement of stator resistance, which can lead to a more accurate estimation of its temperature.

Various techniques have been proposed to estimate stator resistance, with a primary focus on induction motors (IMs) [19] and permanent magnet synchronous motors (PMSMs). While these techniques were developed for these motor types, they may also be applicable to other types of motors with appropriate adaptations.

The papers referenced from [20]–[27] discuss different methods for estimating the resistance of induction motors (IMs) online. Two of these papers, [20] and [21], suggest using different types of estimators, with Kenné et al. [20] proposing a simpler online resistance estimator. Other papers, such as Holakooie et al. [21], and Davari et al. [22], propose using a model reference adaptive system (MRAS)-based resistance estimation in motors controlled by direct torque control, and deadbeat control, respectively.

Papers [23] and [24] suggest using adaptive observers for estimating resistance and analyzing their stability. On the other hand, papers [25] and [26] describe different injection-based methods, with Lee et al. [25] examining the method's applicability during standstill. Additionally, Karanayil et al. [27] used artificial neural networks (ANNs) to estimate resistance and reported low estimation errors.

The methods described in references [16], [17], and [28]–[39] were developed to estimate the resistance of permanent magnet synchronous motors (PMSMs) in real-time.

The studies published in references [27]–[31] suggested different estimation techniques, such as an adaptive estimator [29] and an estimator that takes into account machine saturation. Sangsefidi et al. [30], [31] specifically designed resistor estimators for motors operating under direct torque control (DTC), while the article mentioned in reference [32] proposed an estimator that analyzes higher harmonics of the back electromotive force (EMF). The research papers cited as [18], [35], and [36] have put forward the idea of utilizing the nonlinear Luenberger observer and MRAS-based techniques for online operations. In addition, other interesting methods for online estimation of stator resistance have been proposed in papers [37]–[39]. Liu et al. [37] introduced quantum genetic algorithms, while articles [38] and [39] employed the affine projection of motor electrical signals. Pulvirenti et al. [40] presented a specialized current vector control strategy that allows for the independent determination of stator resistance and permanent magnet flux linkage. This particular method is designed for open-end winding motors.

Simultaneously, another set of techniques described in papers [19] and [41] introduced different injection methods. Antonello et al. [19] presented an algorithm that enables accurate measurement of stator resistance, which is particularly useful for temperature estimation purposes. On the other hand, the method outlined in the paper [41] proposed injections along the $d-$ and $q-$axes, with adaptability for both offline and online operations.

All of these techniques have the ability to estimate stator resistance in real-time, but their implementation and fine-tuning can be challenging. Additionally, some of these methods may not provide precise results in the low-speed range.

To simplify the process of stator resistance estimation, an alternative approach involves offline techniques. Although these methods cannot continuously track resistance changes during motor operation, such functionality is not always necessary in many applications. Consequently, the offline estimation of stator resistance has garnered significant attention, leading to the publication of new studies in the form of papers [42]–[46]. The methodology presented in the paper [42] suggests injecting a high-frequency signal while compensating for dead time. However, this approach shares a drawback commonly found in online methods: it is challenging to implement and fine-tune.

Additionally, it introduces unpleasant noise that can be bothersome to users, making it more suitable for home appliances. In contrast, an alternative approach was explored in

[43], which involves adjusting each motor parameter by modifying the current controller structure and employing a hill-climbing algorithm to seek the minimum point of the current norm. Nevertheless, implementing and fine-tuning this method is also complex. Moreover, it demands a significant amount of time for implementation and fine-tuning, which is inconvenient for factory-oriented projects with stringent deadlines. Long et al. [44] demonstrated that the sequence of parameter estimation impacts precision and proposed an optimal algorithm for identifying motor parameters.

The study documented in [45] introduced offline techniques for accurately measuring model parameters of induction motors (IMs). However, this method requires specialized laboratory equipment and cannot be employed in self-commissioning drives. To address this drawback, Marcetic et al. [46] proposed a straightforward algorithm for self-commissioning IM drives using the injection of direct current (dc) signals. While this technique estimates stator resistance, it cannot measure the voltage drop across inverter switches. Furthermore, it does not offer guidance on parameter selection for the estimation algorithm, making its tuning more complicated.

Simultaneously, the measurement of voltage drop across inverter switches has received limited attention, and only a few studies have addressed this issue. Wang et al. [47] specifically focused on compensating for inverter nonlinearities and put forth a straightforward estimation method for the voltage drop across inverter switches. However, this method requires knowledge of the stator resistance. The proposed algorithm relies on a predetermined constant value, disregarding variations in resistance, thereby leading to reduced accuracy.

In a similar vein, the approach discussed in [48] also focuses on compensation techniques but relies on pre-defined constants for the voltage drop values. This method treats the voltage drop across inverter switches as fixed values that can be obtained from the power device specifications. However, it does not account for variations in voltage drop caused by temperature changes or discrepancies in production. The research presented in [53] suggests injecting two multilevel direct current (dc) sequences with approximately 20-40 different dc levels, each at different pulse width modulation (PWM) frequencies.

The authors then calculate the inverter characteristic at the corresponding injection levels. By injecting more levels, a more detailed curve of the characteristic can be obtained. While this approach enables obtaining detailed inverter characteristics, it requires around 2 minutes to execute, making it impractical for execution before each motor start. Additionally, the authors do not take into account parameter variations caused by temperature changes, which renders it unsuitable for usage in electric drives.

After evaluating the existing methods and assessing their advantages and disadvantages, it became evident that they either lacked sufficient precision or were overly complex. As a result, the decision was made to develop a straightforward offline technique for estimating the stator resistance. This technique aims to provide ease of implementation and tuning. All details are explained in Chapter 4.

1.3 Thesis Organization

This thesis comprises 6 chapters which are related to stator resistance estimation algorithm design and its real-time implementation steps. The first chapter is related to the purpose of the thesis and literature review.

As we know, knowing the plant that we want to control is a crucial part of designing a control system. The second chapter involves information about electrical motors and their control methodologies. The chosen development kit is introduced in detail in this part and gives a brief explanation of the estimation method.

The third chapter is related to the mathematical equations of the implemented theories such as permanent magnet synchronous motor dynamics equations and field-oriented control equations. Moreover, Moreover, this chapter involves equations and mathematical theories to obtain PI controller parameters, These calculated controller parameters are optimized in chapter five thanks to the response optimizer toolbox. Lastly, Two different groups of sliding mode observer equations also are in this chapter. The difference between them is just a discrete part. The first group does not include discrete equations. The second part includes discrete equations for both PMSM and sliding mode observer. To generate C code, the first group can not be used. So discrete equations are used to design simulations and generate C code.

Chapter 4 includes detailed information about stator resistance estimation. Estimation types and the reason for choosing the implemented method are explained in detail. Moreover, the stator resistance value which is measured by the RLC meter is in this chapter.

Chapter 5 comprises both simulation design and real-time implementation results. Firstly, the per unit system and code generation background are explained in detail. Then designing the plant model for the motor and inverter by using a physical modeling strategy is explained. In addition, the controller design for both the current and speed parts by using the cascade method is explained. For the current controller part, the space vector pulse width modulation technique is explained in detail. A sliding mode observer is designed too. The stator resistance estimation method does not need rotation. In contrast, the rotor position has to be constant. But, SMO is designed for future work. After designing field oriented control algorithm, test signals to inject the $d-$axis is prepared and the stator resistance estimation value is estimated in the real system.

Chapter 6 is the conclusion part. The datasheet value, and estimated and measured value of the stator resistance information have been compared. Then the future works have been explained in brief.

2. ELECTRICAL MOTORS AND FIELD ORIENTED CONTROL

This part includes DC motors and their problems, Permanent magnet synchronous motors, field-oriented control, and our development kit.

2.1 DC Motors and Their Problems

DC motors have gained significant popularity in high-performance systems due to their ability to easily and independently control torque and flux. In DC brush machines, the magnetizing current, which is responsible for field excitation, can be supplied by an external source. When this occurs, the machine is referred to as separately excited. Separately excited DC motors have been primarily utilized in applications that demand a rapid response, four-quadrant operation, and exceptional performance at low speeds.

In DC brush motors, the flux is controlled by adjusting the current in the field winding, while the torque is modified by changing the current in the armature winding. However, this design choice comes with certain drawbacks. The motor construction is less sturdy, requiring frequent maintenance and eventual replacement of brushes and commutators. It also limits the use of DC motors in hazardous environments where sparking is prohibited. Additionally, there is a potential drop known as the "contact potential difference," typically ranging from 1 to 1.5 V, which results in a reduction of the effective input voltage.

The DC brush motor is composed of a stator and rotor. The stator (stationary part) contains a magnetic field, which can be created by either permanent magnets or excited field windings on the stator poles. The rotor, on the other hand, consists of the armature winding, a commutator (a mechanical switch that rotates), an armature core, and a rotor shaft. The commutator segments are isolated from one another and from the clamp holding them. In addition to brushes, which are the stationary external components of the rotor, the commutator and brushes serve as rotary contacts between the coils of the rotating armature and the stationary external circuit.

The DC brush motor achieves maximum torque when the stator and rotor magnetic fields are positioned perpendicular to each other. The commutator is responsible for ensuring that the stator and rotor magnetic fields remain perpendicular. Commutation plays a crucial role in the operation of the DC brush motor as it causes the current flowing through the loop to reverse precisely when unlike poles face each other. This reversal in the current direction changes the polarity of the magnetic field, transforming the attractive magnetic force into a repulsive one. Consequently, this continuous rotation of the loop is sustained.

Applying a voltage at the brushes of the DC motor results in current flowing through two coils. This current interacts with the magnetic field of the permanent magnet, generating torque that initiates movement. As the motor moves, the brushes automatically switch to a different coil, causing the rotor to continue turning. Increasing the voltage (armature) leads to a higher rotation speed, while a stronger magnetic field from the permanent magnet produces greater torque.

Due to the short-circuiting of the back-EMF generated in the coil by the brush, a significant current flows, resulting in sparking at the interface between the commutator and the brushes. This process also leads to heating and the production of braking torque. To mitigate this issue, commutation is performed in the region where the magnetic field crosses over. However, despite these measures, the effective magnetic flux is distorted due to armature reaction, leading to some residual back-EMF generated in the coils within the magnetic field crossover region. Minimizing the crossover region is desirable in order to maximize the motor's efficiency and utilization.

In DC motors, the applied voltage, also known as EMF, is always lower than the back-EMF. This difference between the applied EMF and back-EMF ensures that current can flow through the conductor, enabling motion to be produced.

2.2 Permanent Magnet Synchronous Motors (PMSMs)

There are two main types of three phase permanent magnet synchronous motors: those that employ rotor windings supplied by the stator and those that utilize permanent magnets. In motors with rotor windings, brushes are necessary to obtain the current supply and create rotor flux.

These brushes make contact with rings that have multiple commutator segments. However, this type of structure has certain drawbacks, including increased maintenance requirements and reduced reliability.

By substituting pole structure and the conventional rotor field windings with permanent magnets, the motor falls into the category of brushless motors. Brushless permanent magnet motors can be constructed with any even number of magnet poles. Utilizing magnets allows for efficient utilization of radial space and eliminates the need for rotor windings, thereby reducing losses in the copper. The use of advanced magnet materials enables significant size reduction of the motor while maintaining a high power density.

The recent emergence of high energy density permanent magnet (PM) materials at affordable prices, along with significant advancements and cost reductions in microcontrollers and powerful fast digital signal processors (DSPs), as well as breakthroughs in modern control technologies and semiconductor switches, has created exciting opportunities for brushless motor drives utilizing permanent magnets. These developments aim to meet the competitive demands of the global market [1].

Permanent Magnet Synchronous Motors (PMSMs) have gained popularity due to their attractive characteristics:

- High efficiency
- High torque to volume ratio
- High torque to inertia ratio
- High power factor
- High air gap flux density
- Lower maintenance cost
- High acceleration and deceleration rates
- Compact structure
- Simplicity and ruggedness
- Linear response in the effective input voltage

Nevertheless, certain limitations, such as higher initial expenses, temperature constraints during operation, and the risk of demagnetization (primarily attributed to the presence of permanent magnets) can impose restrictions on certain applications.

In permanent magnet synchronous motors (PMSMs), the rotor accommodates permanent magnets, either inside or outside. Unlike DC brush motors, both PMSMs and brushless DC (referred to as BLDC) motors require a "drive" to deliver commutated current. This is achieved by pulse width modulation of the DC bus through a DC-AC inverter connected to the motor windings. Synchronization of the windings with the rotor position is essential and can be achieved through position sensors or sensorless position estimation techniques. By activating specific windings in the stator based on the rotor's position, a rotating magnetic field is created.

In synchronous motors, permanent magnets are securely attached to the rotating axis to establish a consistent rotor flux with a constant strength. When the stator windings are energized, they generate a rotating electromagnetic field. To regulate this rotating magnetic field, it is crucial to control the currents flowing through the stator.

The specific configuration of the rotor in synchronous machines varies based on the machine's power range and rated speed. Permanent magnets are typically utilized in synchronous machines with power ratings up to a few Kilowatts. However, for higher power ratings, the rotor is commonly composed of windings through which a direct current (DC) circulates. The mechanical structure of the rotor is designed to accommodate the desired number of poles and achieve the desired flux gradients.

The torque is generated by the interaction between the rotor and stator fluxes. The stator is stationary, while the rotor can rotate freely. As a result of this interaction, the rotor rotates, producing a mechanical output that can be used for various applications.

To maximize torque and enhance electromechanical energy conversion efficiency, it is crucial to precisely control the angle between the stator field and the rotor magnetic field. Fine-tuning is necessary after implementing the field-oriented control algorithm to close the speed loop. This fine-tuning aims to minimize the current drawn under identical speed and torque conditions.

In order to achieve maximum torque production, the rotor field and the stator field need to be perpendicular to each other. This synchronization relies on determining the position of the rotor to generate the appropriate stator field.

2.3 Field Oriented Control

To enhance the dynamic performance of a PM motor, a sophisticated control scheme is required. Leveraging the computational capabilities of microcontrollers, advanced control strategies can be implemented, utilizing mathematical transformations to separate torque generation and magnetization functions in PM motors. This separation, known as rotor flux-oriented control (FOC), enables decoupled control of magnetization and torque for improved motor performance. To comprehend the essence of the FOC method, it is necessary to begin with an introduction to the separately excited DC motor. In this motor configuration, the excitation for both the stator and rotor can be controlled independently. When studying the electrical characteristics of the DC motor, it becomes evident that the flux produced and the torque generated can be adjusted independently of each other. The field excitation strength, determined by the magnitude of the excitation current, directly influences the level of flux. Meanwhile, the amount of torque generated is dictated by the current flowing through the rotor windings. Notably, the commutator located on the rotor plays an interesting role in the production of torque. The commutator makes contact with the brushes, and the mechanical structure is designed to activate the windings aligned in a way that maximizes torque production. The important aspect to note is that the management of the windings ensures that the flux generated by the rotor windings remains perpendicular to the stator field. Unlike DC motors, AC machines lack some crucial characteristics. In both types of motors, the stator currents are the only controlled source. In the case of a synchronous machine, the rotor excitation is provided by the permanent magnets mounted on the shaft, whereas for the synchronous motor, the stator phase voltage is the sole source of power and magnetic field. Notably, unlike the DC motor, the flux, and torque in AC machines are dependent on each other.

The objective of FOC, also known as vector control, in both synchronous and asynchronous machines is to independently control the magnetizing flux and torque-producing components. The aim of this control technique is to replicate the operational characteristics of a DC motor.

FOC enables the separation of torque and magnetizing flux components of the stator current, effectively achieving independent torque control. To achieve this decoupling, various mathematical transformations are employed, and this is where microcontrollers play a crucial role, providing significant value. The computational power of microcontrollers facilitates the swift execution of the necessary mathematical transformations. As a result, the entire motor control algorithm can operate at a rapid pace, leading to enhanced dynamic performance. Furthermore, besides decoupling, a dynamic model of the motor is employed for calculating various quantities such as rotor flux angle and rotor speed. This approach ensures that the impact of these factors is considered, resulting in improved overall control quality.

According to the principles of electromagnetism, the torque generated in a synchronous machine can be understood as the outcome of a vector cross-product involving two magnetic fields:

$$T_{em} = \vec{B}_{stator} x \vec{B}_{rotor} \tag{2.1}$$

Equation 2.1 indicates that the torque reaches its maximum value when the stator and rotor magnetic fields are perpendicular, which corresponds to maintaining a load at a 90° angle. By consistently achieving this condition and correctly orienting the flux, torque ripple can be minimized and a more favorable dynamic response can be ensured. However, the challenge lies in determining the rotor position, which can be accomplished using a position sensor like an incremental encoder. In cost-effective applications where direct access to the rotor is not feasible, alternative strategies using rotor position observers such as sliding mode observers are employed to eliminate the need for a position sensor.

In summary, the objective is to keep the stator and rotor flux at a 90-degree angle with each other. This is done by aligning the stator flux with the $q-$axis of the rotor flux, which is perpendicular to the rotor flux. This is achieved by controlling the stator current component that is perpendicular to the rotor flux, thereby generating the desired torque, while the direct component is kept at zero.

In certain situations, the direct component of the stator current can be utilized for field weakening, which opposes the rotor flux and reduces back electromotive force (back EMF), allowing for operation at higher speeds.

All mathematical equations that are used in field-oriented control algorithms are explained in the next chapter. Generating space vector pulse width modulation is explained by creating its model in the next chapter.

3. MATHEMATICAL MODELING OF PMSM

This chapter includes mathematical modeling of PMSM, PI controller design for current control of PMSM, PI controller design for speed control of PMSM, sliding mode observer equations and its tuning.

3.1 Mathematical Model of PMSM

$$\frac{di_d(t)}{dt} = \frac{1}{L_d}\left(V_d(t) - R_s i_d(t) + \omega_e(t) L_q i_q(t)\right) \tag{3.1}$$

$$\frac{di_q(t)}{dt} = \frac{1}{L_q}\left(V_q(t) - R_s i_q(t) + \omega_e(t) L_d i_d(t) - \omega_e(t)\phi_{mg}\right) \tag{3.2}$$

$$\frac{\omega_e(t)}{dt} = \frac{Z_p}{J_m}\left(T_e - \frac{B_v}{Z_p}\omega_e(t) - T_L\right) \tag{3.3}$$

$$T_e = \frac{3}{2} Z_p \left[\phi_{mg} i_q + (L_d - L_q) i_d(t) i_q(t)\right] \tag{3.4}$$

where ;

- i_d and i_q values are $d-$ and $q-$axes currents,
- u_d and u_q values are $d-$ and $q-$axes stator voltages,
- L_d and L_q values are $d-$ and $q-$axes inductances,
- R_s is stator resistance,
- ψ_f is magnetic flux,
- P is pole pairs,
- ω is speed,
- J is the moment of inertia,
- f_c viscous coeffient,
- T_L is torque load

3.2 PI Controller Design for Current Control of PMSM

$d-$axis current i_d and $q-$axis current i_q provide the controller with a feedback signal. If the main goal is to control the electrical velocity, one of the feedback signals should be ω_e as electrical velocity. A number of procedures to bridge sensor measurements and feedback signals are used. Three current sensors are employed to measure the currents in all three phases as shown in Figure 3.1.

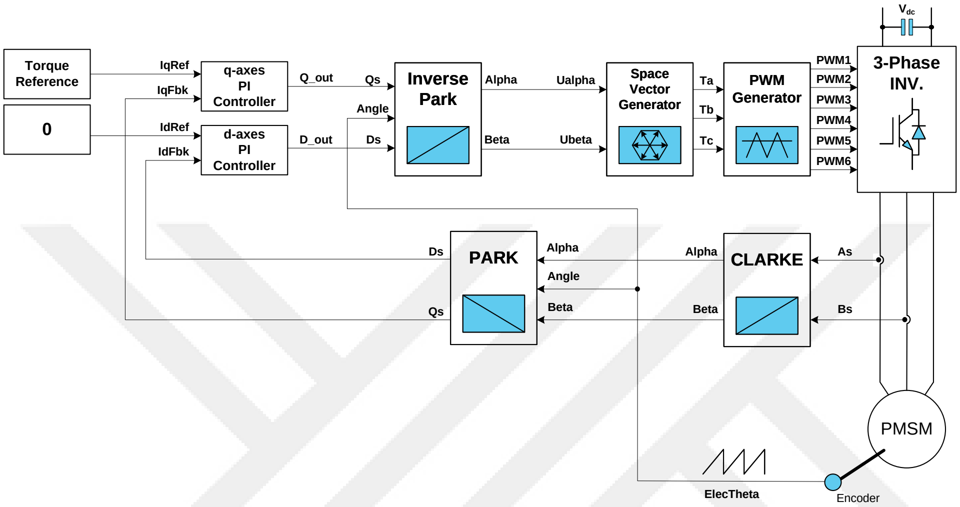

Figure 3.1 : Current control algorithm of PMSM drive.

Using Clarke's transformation theory, three-phase sinusoidal current signals which have 120° phase difference among them are transformed into two-phase sinusoidal signals which have 90° phase difference between them according to Equation 3.5.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt(3)}{2} & -\frac{\sqrt(3)}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{3.5}$$

The aim of these transformations is making easier the control current by reducing the number of currents and by converting the shape of the current. Thanks to Clarke transformation, the number of signals is reduced, and then the sinusoidal current shape is transformed into DC signals by using Park transformation Equation 3.6.

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta_e & \sin\theta_e \\ -\sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{3.6}$$

where the AC sinusoidal signals i_α and i_β are transformed to the i_d and i_q currents. The Park transformation is Equation 3.6. In a PMSM normal steady-state operation, the i_d and i_q currents should be constant so that using PI controllers, i_d and i_q currents are regulated.

In order to control the velocity of PMSM, obtaining velocity signal ω_e is required. To measure ω_e, an encoder is used. Moreover, sensorless methods can be used such as sliding mode observer, Luenberger observer, etc. These methods measure the mechanical angle θr and then this angle is multiplied by the number of pole pairs to obtain electrical angle θ_e. By taking the derivative of the electrical angle θ_e, electrical velocity ω_e is obtained. This part is crucial because the electrical angle is used in Park transformations and electrical speed is used as a feedback signal for the speed controller.

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta_e & -\sin\theta_e \\ \sin\theta_e & \cos\theta_e \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix} \tag{3.7}$$

The output of the controllers is V_d and V_q are DC voltages. In order to obtain sinusoidal three-phase voltages again, inverse park transformation in Equation 3.7 and inverse Clarke transformation in Equation 3.8 are used.

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 \end{bmatrix} \begin{bmatrix} V_\alpha \\ V_\beta \\ V_0 \end{bmatrix} \tag{3.8}$$

After this transformation process, three-phase Ac signals are connected to the space vector pulse width modulation algorithm. This algorithm is used to generate control signals to control the inverter. A DC bus is connected to an inverter and DC voltage is translated into AC voltage to drive the PMSM motor. Generating space vector pulse width modulation is explained in future chapters by examining in detail all equations and theories. The electromagnetic torque $T_e(t)$ in Equation 3.4 is made up of two components, with the first component being directly proportional to the q-axis current, while the second component is proportional to the product of $i_d(t)$ and $i_q(t)$.

Therefore, to control the electromagnetic torque, it is necessary to implement closed-loop control of the $d-$axis and $q-$axis currents.

During the implementation of torque control, the set-point signals for the current loops are typically either constant or piece-wise constant. Normally, the reference signal for d-axis current is given as zero so that maximizing the generated torque.

This approach effectively plays roles in the second term in Equation 3.4 to be zero during steady-state operation. Failure to do so would result in a nonlinear relationship between the electromagnetic torque T_e and i_q. As a result, the d-axis current represents the flux and the q-axis current represents the torque.

Since inductances L_d and L_q are identical for a surface-mounted PMSM, the second part of the torque in Equation 3.4 becomes zero. To sum up, for a surface-mounted PMSM, the torque equation is

$$T_e = \frac{3}{2} Z_p \phi_{mg} i_q(t) \tag{3.9}$$

As a result, q-axis current control is necessary to control the electromagnetic torque. The set point signal of current i_q^* is calculated as shown in Equation 3.10.

$$i_q^* = \frac{2}{3} \frac{T_e^*}{Z_p \phi_{mg}} \tag{3.10}$$

where T_e is the electromagnetic torque reference signal. If the motor is an interior-mounted PMSM, the set point signal is,

$$i_q^* = \frac{2}{3} \frac{T_e^*}{Z_p \phi_{mg}(1 + i_d^*)} \tag{3.11}$$

Thanks to the current control system decoupling method, by using Equation 3.1 and Equation 3.2, the electrical part of the machine dynamics is represented by a first-order model as shown in Equation (3.12), Equation (3.13).

$$\frac{di_d}{dt} = \frac{R_s}{L_d} i_d + \frac{1}{L_d} \hat{v}_d \tag{3.12}$$

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q + \frac{1}{L_q} \hat{v}_q \tag{3.13}$$

By utilizing Equation 3.12 and Equation 3.13, it is possible to design two feedback controllers that can be employed to regulate the stator current through the manipulation of auxiliary stator voltages in the dq frame.

To develop proportional-integral (PI) controllers for the current control loops, the Laplace transfer functions for the electric system are determined using Equation 3.12 and Equation 3.13 resulting in a transfer function denoted as,

$$\frac{I_d(s)}{\hat{V}_d(s)} = \frac{\frac{1}{L_d}}{s + \frac{R_s}{L_d}} \tag{3.14}$$

$$\frac{I_q(s)}{\hat{V}_q(s)} = \frac{\frac{1}{L_q}}{s + \frac{R_s}{L_q}} \tag{3.15}$$

Laplace transforms of the $d-$axis current $q-$axis current are $I_d(s)$ and $I_q(s)$, respectively; auxilary voltage variables laplace transforms are $\hat{V}_d(s)$ and $\hat{V}_q(s)$.

The transfer function for the PI controller associated with the $d-$axis current loop is expressed in the following manner:

$$C(s) = K_c^d \left(1 + \frac{1}{\tau_I^d s}\right) \tag{3.16}$$

The integral time constant τ_I^d or τ_I^q and proportional gain K_c^d or K_c^q are calculated by using pole-assignment controller design. By selecting $a = \frac{R_s}{L_d}$ and $b = \frac{1}{L_d}$, integral time constant and proportional control gain are calculated as

$$K_c^d = L_d \left(2\zeta\omega_n - \frac{R_s}{L_d}\right) = 2\zeta\omega_n L_d - R_s \tag{3.17}$$

$$\tau_I^d = \frac{2\zeta\omega_n - \frac{R_s}{L_d}}{\omega_n^2} = \frac{2\zeta\omega_n L_d - R_s}{L_d\omega_n^2} \tag{3.18}$$

The PI controller parameters for the $q-$axis current control shall also be calculated as follows:

$$K_c^q == 2\zeta\omega_n L_q - R_s \tag{3.19}$$

$$\tau_I^q = \frac{2\zeta\omega_n L_q - R_s}{L_q\omega_n^2} \tag{3.20}$$

In the control system design, the damping coefficient ζ is chosen as 0.707. The other parameter, natural frequency ω_n is evaluated to specify the closed-loop control system settling time. This parameter also is related to the desired bandwidth of the closed-loop control system. If the value of ω_n is increased, the desired settling time of the closed-loop system is reduced. In other words, a larger ω_n leads to a shorter settling time for the closed-loop control system. Selecting this parameter according to open-loop system bandwidth is useful and selected as either $\frac{R_s}{L_q}$ or $\frac{R_s}{L_d}$. For use in current control of $d-$axis, natural frequency is,

$$\omega_n = \frac{1}{1-\gamma}\frac{R_s}{L_d} \tag{3.21}$$

where γ is a normalized parameter as $0 < \gamma < 1$. For the current control of q-axis

$$\omega_n = \frac{1}{1-\gamma}\frac{R_s}{L_q} \tag{3.22}$$

When using such formulas, the value of γ is typically chosen to be approximately 0.9 to achieve satisfactory system performance. It is worth noting that the desired closed-loop performance in the d-axis of the system may not necessarily be the same as that specified for the q-axis current control system. For natural frequency $\omega_n = 68.5 rad/s$ and damping ratio $\zeta = 0.707$ current loop controllers are calculated as $K_p = 0.72$ and $K_i = 3.75$. These controller parameters can be used for both the $d-$axis and $q-$axis current controllers.

3.3 PI Controller Design for Speed Control of PMSM

Since the Proportional-Integral current controller, for the current control system, the steady-state error is eliminated. This type of controller leads to the robustness of the control system in the presence of parameter variation. According to the cascade current controller design method, after describing the PI current control in the previous part, we will now design the outer loop which is the velocity control.

$q-$axis current dynamic system is described in Equation 3.23.

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q + \frac{1}{L_q} \hat{v}_q \tag{3.23}$$

where $\hat{V}_q$ is related to the original q-axis voltage by the following equation,

$$V_q = \hat{v}_q + \omega_e L_d i_d + \omega_e \phi_{mg} \tag{3.24}$$

The closed-loop transfer function between the feedback signal $I_q(s)$ and the reference signal $I_q^2(s)$ is required in order to design the speed controller. Normally, Equation 3.23 has nonlinear terms as shown in Equation 3.25.

Laplace transformation is taken on both sides, we obtain

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q(t) + \frac{1}{L_q} K_c^q \left(i_q^*(t) - i_q(t) \right) + \frac{K_c^q}{\tau_I^q L_q} \int_0^t (i_q^*(\tau) - i_q(\tau)\, d\tau) \tag{3.25}$$

Following relationships on Equation 3.19 and Equation 3.20 are true.

$$sI_q(s) = -\frac{R_s}{L_q} I_q(s) + \frac{1}{L_q} \left(K_c^q (I_q^*(s) - I_q(s) \right) + \frac{K_c^q}{\tau_I^q L_q s} (i_q^*(s) - i_q(s)) \tag{3.26}$$

$$\frac{K_c^q}{\tau_I^q} = L_q \omega_n^2 \,;\ \frac{K_c^q}{L_q} = 2\zeta\omega_n - \frac{R_s}{L_q} \tag{3.27}$$

By evaluating all these equations, the closed-loop transfer function is obtained as shown in Equation 3.28.

$$\frac{I_q(s)}{I_q^*(s)} = \frac{\left(2\zeta\omega_n - \frac{R_s}{L_q}\right) s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.28}$$

The current loop poles are located with $\zeta = 0.707$ at $s_{1,2} = -\zeta\omega_n \pm \omega_n j\sqrt{1-\zeta^2}$. ω_n is a tunable parameter for the closed-loop controller system. Closed loop transfer function steady-state gain is unity setting $s = 0$ in Equation 3.28.

To design a speed loop PI control system, the transfer function between the $I_q(s)$ $q-$axis current and $\Omega_e(s)$ electrical velocity is governed by

$$\left(s + \frac{B_v}{J_m}\right)\Omega_e(s) = \frac{3}{2}\frac{Z_p^2\phi_{mg}}{J_m}I_q(s) \tag{3.29}$$

By substituting Equation 3.28, Equation 3.29, transfer function between the $\Omega_e(s)$ electrical velocity and $I_q^*(s)$ q-axis current as

$$\frac{\Omega_e(s)}{I_q^*(s)} = \left(\frac{\frac{3}{2}\frac{Z_p^2\phi_{mg}}{J_m}}{s+\frac{B_v}{J_m}}\right)\left(\frac{\left(2\zeta\omega_n - \frac{R_s}{L_q}\right)s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right) \tag{3.30}$$

A first-order model is required in order to design a PI controller by using the pole assignment controller design method. The Equation 3.31 is a third-order model and it is required to be linearized into the first-order model. The natural frequency ω_n is for the current loop q- axis controller, and this parameter is selected in the design. In order to get a first order approximation, ω_n is selected as $\omega_n >> \frac{B_v}{J_m}$, for example, $\omega_n = 10\frac{B_v}{J_m}\ rad/s$. By selecting large ω_n and neglecting the dynamics of inner loop current control, the first-order model is obtained to design a PI controller for speed loop as shown in Equation 3.31.

$$\frac{\Omega_e(s)}{I_q^*(s)} \approx \frac{\frac{3}{2}\frac{Z_p^2\phi_{mg}}{J_m}}{s+\frac{B_v}{J_m}} \tag{3.31}$$

To calculate PI controller parameters using pole assignment method α and β is selected as shown in Equation 3.32.

$$a = \frac{B_v}{J_m}\ ;\ b = \frac{3}{2}\frac{Zp^2\phi_{mg}}{J_m} \tag{3.32}$$

And then, selecting the desired closed loop poles $s_1, 2 = -\zeta\omega \pm j\omega_n\sqrt{(1-\zeta^2)}$, where $\zeta = 0.707$ or 1, the proportional gain K_c is as shown in Equation 3.33.

$$K_c = \frac{2\zeta\omega_n - a}{b} \tag{3.33}$$

Lastly, the integral time constant is as shown in Equation 3.34.

$$\tau_I = \frac{2\zeta\omega_n - a}{\omega_n^2} \tag{3.34}$$

The current (inner-loop) loop PI control systems natural frequency is selected as higher than the speed loop PI control systems natural frequency. Because the current loop speed should be greater than the speed loop. The difference should be at least 5 times to 10 times. To calculate control parameters for the speed control loop, natural frequency ω_n is 6.85 rad/sec and damping ratio ζ is 0.707. Then, controller parameters are calculated as $K_p = 1.38$ and $K_i = 6.65$. For this calculation, the moment of inertia is 0.0047 $kg.m^2$ and permanent magnet flux is 0.033 Wb. To improve control performance, during the simulation phase, thanks to the Simulink response optimizer toolbox and control system toolbox, all controller parameters will be updated. All these parameters can be calculated using these toolboxes, but knowing the theory is crucial in deciding whether these parameters are applicable in a real system.

3.4 Sliding Mode Observer Equations

The sliding-mode observer (SMO) estimates the stator currents at the stationary reference coordinate, taking into account the estimated currents and the measured actual current values, with a control strategy in which a specified switching function is present. In stator current estimation, back-EMF voltages are obtained. These back-EMF signals carry rotor position information necessary for Clarke and Park transformations.

3.4.1 Mathematical equations of sliding mode observers

Field oriented control method is used to control a PMSM. Thanks to this method, an AC motor is controlled like a DC motor.

For this reason, it is thought that the electrical equations from the conventional brushed motor differential equations can be used to obtain the estimated currents for the estimation of the stator currents needed within the sliding mode observer. The motor dynamics in $\alpha\beta$ reference system is

$$\frac{d}{dt}i_\alpha = -\frac{R_s}{L_s}i_\alpha - \frac{1}{L_s}e_\alpha + \frac{1}{L_s}u_\alpha \tag{3.35}$$

$$\frac{d}{dt}i_\beta = -\frac{R_s}{L_s}i_\beta - \frac{1}{L_s}e_\beta + \frac{1}{L_s}u_\beta \tag{3.36}$$

Back EMF signals are described as shown in Equation 3.37 and Equation 3.38.

$$e_\alpha = -\psi_f \omega sin(\theta) \tag{3.37}$$

$$e_\beta = -\psi_f \omega cos(\theta) \tag{3.38}$$

Therefore, sliding mode observer dynamic equations are shown in Equation 3.39, Equation 3.40.

$$L_s \frac{d}{dt}\hat{i_\alpha} = -R_s\hat{i_a} + u_\alpha - e_\alpha - ksign(\hat{i_\alpha} - i_\alpha) \tag{3.39}$$

$$L_s \frac{d}{dt}\hat{i_\beta} = -R_s\hat{i_\beta} + u_\beta - e_\beta - ksign(\hat{i_\beta} - i_\beta) \tag{3.40}$$

In Equation 3.39, Equation 3.40, $\hat{i_\alpha}$ and $\hat{i_\beta}$ estimated stator currents and k is the sliding mode observer gain. The criteria for k is as shown in Equation 3.41.

$$k > max(\|e_\alpha||e_\beta|) \tag{3.41}$$

$ksign(\hat{i_\alpha} - i_\alpha)$ and $ksign(\hat{i_\beta} - i_\beta)$ terms try to make the error value between the estimated and measured stator current signals zero.

Here, instead of the $signum(x)$ function, the $sigmoid(x)$ or $tanh(x)$ functions can be used as an alternative. $ksign(\hat{i_\alpha} - i_\alpha)$ and $ksign(\hat{i_\beta} - i_\beta)$ terms are passed through a low-pass filter, e_α and e_β signals, which are back-EMF signals at the stationary reference coordinate, are obtained.

The rotor flux position is obtained by using the estimated back-EMF signals as given by Equation 3.42.

$$\hat{\theta} = \arctan\left(-\frac{\hat{e_\alpha}}{\hat{e_\beta}}\right) \tag{3.42}$$

This estimated rotor position is used in Park and inverse Park transformation. Moreover, this position information is also used to get speed information by taking the derivative.

The Sliding Mode Observer is a component that determines the electrical position and mechanical speed of a Permanent Magnet Synchronous Motor (PMSM). It achieves this by analyzing the voltage and current values present on the $\alpha-$ and $\beta-$axes of the stationary $\alpha\beta$ reference frame.

Discrete-time operation of a PMSM is shown in Equation 3.43–3.49.

$$i_{\alpha\beta(k+1)} = Ai_{\alpha\beta(k)} + Bv_{\alpha\beta(k)} - Be_{\alpha\beta(k)} \tag{3.43}$$

$$e_{\alpha\beta(k+1)} = e_{\alpha\beta(k)} + T_s\omega_{e(k)}Je_{\alpha\beta(k)} \tag{3.44}$$

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{3.45}$$

$$\Phi = \begin{bmatrix} -\frac{R}{L} & 0 \\ 0 & -\frac{R}{L} \end{bmatrix} \tag{3.46}$$

$$A = e^{\Phi T_s} \tag{3.47}$$

$$B = \begin{bmatrix} b & 0 \\ 0 & b \end{bmatrix} \tag{3.48}$$

$$b = \frac{1 - e^{-RT_s/L}}{R} \tag{3.49}$$

Discrete-time sliding mode observer operation of a PMSM is shown in Equation 3.50–3.53.

$$\hat{i}_{\alpha\beta(k+1)} = A\hat{i}_{\alpha\beta(k)} + Bv_{\alpha\beta(k)} - B\hat{e}_{\alpha\beta(k)} - \eta Sign(\tilde{i}_{\alpha\beta(k)}) \tag{3.50}$$

$$\hat{e}_{\alpha\beta(k+1)} = \hat{e}_{\alpha\beta(k)} + B^{-1}g(\tilde{i}_{\alpha\beta(k)} - A\tilde{i}_{\alpha\beta(k-1)} + \eta Sign(\tilde{i}_{\alpha\beta(k-1)}) \tag{3.51}$$

$$\tilde{i}_{\alpha\beta(k)} = \hat{i}_{\alpha\beta(k)} + i_{\alpha\beta(k)} \tag{3.52}$$

$$\tilde{e}_{\alpha\beta(k)} = \hat{e}_{\alpha\beta(k)} + e_{\alpha\beta(k)} \tag{3.53}$$

If the back electromotive force observer fulfills the conditions on Equation 3.54, Equation 3.55

$$|e_{\alpha\beta(k+1)} - e_{\alpha\beta(k)}| \leq m \tag{3.54}$$

$$g\varepsilon(0,1) \tag{3.55}$$

there exist a k_0, such that :

$$\tilde{e}_{\alpha\beta(k)} < \frac{m}{g} \tag{3.56}$$

if the sliding mode observer fulfills below conditions:

- $g\varepsilon(0,1)$
- $|e_{\alpha\beta(k+1) - e_{\alpha\beta(k)}}| \leq m$
- $\eta > b\frac{m}{g}$

then there exist a $K = k_0$, such that for $k \geq$:

$$|\tilde{i}_{\alpha\beta(k)}| \leq \eta + b\frac{m}{g} \tag{3.57}$$

where;

- i_α and e_α are the current and stator back EMF for the α axis

- i_β and e_β current and stator back EMF for the β axis
- $\tilde{i}_\alpha$ and $\tilde{e}_\alpha$ are the errors ins current and stator back EMF for the α axis
- $\tilde{i}_\beta$ and $\tilde{e}_\beta$ are the errors ins current and stator back EMF for the β axis
- v_α, v_β are supply voltages of stator
- L is the stator inductance
- R is the stator resistance
- g is the back EMF observer gain
- η is the current observer gain
- T_s is the sampling period
- ω_e is the electrical angular speed
- k is the sample count

3.4.2 Sliding mode observer tuning

To tune the Back-emf observer gain g and current observer gain $zeta$ parameters steps below are used.

- Back-emf observer gain g value is selected such that $g \in (0, 1)$. if g close to the value 1, estimated back-emf is resulted in less error. However, this makes convergence slow.
- Choose a suitable value for the parameter m by considering the block sample time and the maximum slope of the operating back-electromotive force. (Such that $|e_\alpha\beta(k+1) - e_\alpha\beta(k)| \leq m$)
- Determine an appropriate value for the current observer gain (η) by taking into account the values of b, m, g. such that $\eta > b\frac{m}{g}$

4. STATOR RESISTANCE ESTIMATION METHODS OF PMSM

For optimal dynamic performance of a drive system, it is crucial to accurately estimate machine parameters. While some parameters, such as stator resistance, are typically provided by manufacturers, they may vary under different operating conditions. Therefore, precise estimation of these parameters is necessary to ensure optimal performance of the drive system [32].

There are two types of machine parameters, namely electrical parameters and mechanical parameters. Electrical parameters include the winding resistance and inductance of the stator and rotor in both $dq-$axes. On the other hand, mechanical parameters consist of angular velocity, rotor position in mechanical degrees, moment of inertia, and viscous friction coefficient [33].

During this thesis, we will consider estimating the electrical parameter, the stator resistance. Based on the operational condition, identification methods can be categorized into two types: offline identification method and online identification method [34]. The online identification method is capable of identifying real-time stator resistance and adjusting parameter changes during different operation conditions. Conversely, the offline method is commonly employed to determine the initial resistance parameters and serve as a point of reference for the online identification method [35].

This approach, the dq injection method, demonstrated here can be applied to all types of electric motors, including synchronous motors such as IPMSM, SynRM, and SPMSM, as well as induction motors. The information about the stator resistance obtained through offline estimation is utilized in the design of observers for the PMSM, which is essential for accurate motor control. Additionally, this information may be used to improve the tuning of the current controller for the motor.

Electrical parameter estimation typically involves injecting a test signal into the studied object and observing the resulting signal. If the test signal is voltage, the resulting signal is typically current, or vice versa. By analyzing the amplitude and/or phases of these signals, electrical parameters can be identified. This process is commonly used in electrical engineering for identifying parameters such as resistance. The accuracy of the estimation depends on factors such as the quality of the test signal and the measurement equipment, as well as the complexity of the system being studied.

While stator resistance can be measured manually using an ohmmeter or calculated using Ohm's law, automated testing is highly desirable for stator resistance measurement for several reasons. Firstly, since the stator resistance can change with especially temperature as shown in Equation 4.1 and other operating conditions. Automated testing can provide more reliable and accurate measurements under different conditions. Consequently, it is crucial to accurately measure the stator resistance.

$$R = R_0(1+\alpha(T-T_0)) \tag{4.1}$$

Secondly, automated testing can be used to design motor control systems and is used to continually monitor the stator resistance during operation, improving efficiency and enabling more precise control. Thirdly, automated testing can save time and increase efficiency compared to manual testing, which can decrease downtime and increase productivity. However, implementing and developing an automated testing system requires careful attention to various factors, including measurement safety, accuracy, and system integration. As a result, using an electrical drive for automated testing is advantageous and desirable and there are multiple factors which is related to cable, power semiconductor resistance, inverter voltage distortion, etc., that need to be taken into consideration when designing such a system.

4.1 Cable and Power Semiconductor Resistance

The use of an inverter for stator resistance measurement is crucial as it accounts for the resistances of the cables and power semiconductors, which are physically present and can affect the operation of the drive. They are particularly important at low-speed operation because widely used speed/position sensorless control methods are inherently error-prone at low speed [36].

By using an inverter to measure stator resistance, it is possible to compensate for these resistances and improve the accuracy of the control system. This can lead to better overall performance and quality of the electric drive system. Additionally, inverter-based stator resistance measurement can provide a more convenient and efficient method for measuring resistance, since it can be done without the need for additional measurement equipment.

Overall, the use of inverter-based stator resistance measurement can be a valuable tool for improving the performance and quality of electric drive systems.

4.2 Current vs. Voltage Injection Based Tests

To measure the stator resistance, one can either inject a constant voltage and measure the resulting current amplitude or inject a constant current and observe the voltage required to maintain this current.

While the first approach of measuring stator resistance by open-loop constant voltage injection does not necessarily require a well-tuned current controller, it does have some inherent drawbacks. The main practical challenge is selecting an appropriate voltage amplitude for the test. If the resistance is too low, the test voltage can cause an over-current fault. Conversely, if the voltage is too low, it may result in a very low current, which could hinder accurate data collection. Therefore, the open-loop voltage approach is generally not the preferred method in practical applications.

In contrast, when measuring stator resistance by injecting DC current, there are no issues related to selecting the appropriate voltage amplitude or dealing with over-current faults. This method also eliminates the need for time-consuming iterations, as the current controller automatically determines the injection voltage. Moreover, many motor parameter estimation procedures are better suited for use with a current controller. With a well-designed current controller, the current settles to the reference value in a predictable time, which makes the procedure more reliable and suitable for automation in the final product.

Finally, after trying the open loop method, the results showed that the current is too high to implement on our motor driver board whose name is BOOSTXL-DRV8301. Therefore, in this thesis, first, a closed loop field-oriented control algorithm has been developed by using sliding mode observer, then injecting DC current method has been selected. Offline stator resistance estimation for a sliding mode observer-based PMSM speed control algorithm has been experimentally verified using MATLAB®/Simulink®

and Simscape Electrical$^{\text{TM}}$. The experimental kit includes a low-cost evaluation and development board LAUNCHXL-F28069M with a C2000 real-time microcontroller. DRV8301 has been chosen as an inverter board due to the compatibility of the nominal current and voltage ranges of the 52.5 Watt PMSM. Thanks to the code generation future of the Embedded Coder® toolbox, experimental results show that the estimated resistance is more accurate compared with the stator resistance value measured by an expensive RLC meter.

4.3 Injection DC Currents

Parameter estimation algorithms can be divided into two groups online and offline methods [37].

Online algorithms are designed to operate concurrently with the devices they control, enabling them to track changes in parameters during the working cycles of motor drives. These algorithms are particularly well-suited for systems that operate continuously over extended periods, without the need for downtime. However, implementing and tuning these algorithms can be a challenging task, requiring significant expertise and resources. Moreover, they typically require powerful microcontrollers, which may not be feasible for use in low-cost systems.

In contrast, offline algorithms operate before the start of working cycles and define system parameters based only on the conditions at that specific moment in time. As a result, these techniques are not suitable for tracking changes in motor parameters during operation and are better suited for motor drives that operate in start/stop modes, such as those used in lifting mechanisms or washing machines. Offline algorithms are generally simpler than online techniques and typically do not require complex tuning or adjustment. Therefore, self-commissioning procedures are typically designed to estimate motor parameters while the motor is at a standstill. In procedures for estimating stator resistance, steady-state current excitation can create torque generation and unwanted rotor movement. As a result, in synchronous motors, the stator resistance is typically estimated by injecting a constant current in the actual motor d–axis, as this will not create torque and hence will not cause undesirable or uncontrolled rotor motion. Because for a Permanent magnet synchronous motor, d–axis represents the motor flux and q–axis represents the torque. Importantly, The critical requirement is to maintain a constant angle during the test. Injecting at a constant spatial angle is a must for estimating stator resistance, whether it is on the α–axis, β–axis, or any other angle.

4.4 Inverter Voltage Distortion

In the electric drive industry, it is common practice to measure the phase currents and dc-link voltage instead of the actual motor voltages. Measuring the inverter phase voltages is not preferred due to the high level of contamination by PWM harmonics, which can make it difficult to accurately measure the voltage waveform. Additionally, measuring phase voltages during low phase voltages, such as during stator resistance estimation, may not be helpful due to isolation and low-pass filtering requirements and undesirable measurement phenomena like offset and gain nonlinearity. Therefore, a common approach is to estimate the phase voltages roughly using the PWM voltage references and dc-link voltage measurement.

However, due to inverter distortion voltage, the real inverter output voltage is different from the reference value. The causes include power semiconductor voltage drops, power semiconductor output capacitance charging and discharging time (more noticeable at low phase currents), dead-time insertion to prevent shoot-through in the inverter leg, turn-on and turn-off signal delays in the control hardware such as gate drivers and isolators, and power semiconductor voltage drops.

Since the inverter distorts the synthesized voltage as a result of dead time, device voltage drops, and signal delays, the influence of the distortion voltage must be taken into account while estimating stator resistance. To achieve this, it is preferable to first evaluate the distortion voltage, an example of which is shown in Figure 4.1 The distortion voltage is obviously a nonlinear function of phase current. Furthermore, it primarily consists of two regions: one with a linear voltage-current connection at low current and the other with a flat distortion voltage at high current. The exact characteristics of the distortion voltage shouldn't be necessary for a straightforward, reliable solution. Without knowing the distortion voltage, a reliable solution should be able to determine the stator resistance value. Even with the best of intentions, it is almost impossible to detect the distortion voltage at the low current zone with great

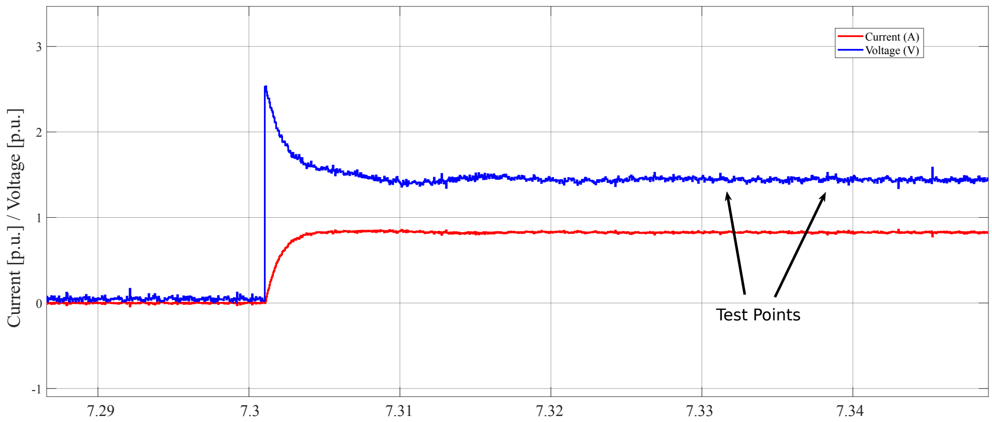

Figure 4.1 : Test points.

accuracy. Even with the best of measurements, it is almost impossible to detect the distortion voltage at the low current zone with great accuracy.

The use of a flat distortion voltage region at a high current as a solution for stator resistance estimation in an inverter-based system can be a reliable approach. This is because measuring the distortion voltage in this region is relatively easy, and the voltage remains constant over a wide current range. By injecting two distinct current values, the redundancy in the distortion voltage can compensate for any potential inaccuracies, and the stator resistance estimation can still be accurate even in the presence of inverter distortion voltage.

This approach can help mitigate the effects of inverter distortion voltage, which can introduce errors in stator resistance estimation due to nonlinearity and non-idealities in the inverter operation. By utilizing a flat region of distortion voltage at high current levels, where the voltage remains relatively constant, the accuracy of stator resistance estimation can be improved.

It's important to note that proper calibration and validation of the measurement setup and algorithm are essential for accurate stator resistance estimation. Additionally, other factors such as temperature variations, harmonics, and noise should also be considered and mitigated to ensure reliable and accurate stator resistance estimation in an inverter-based system.

In a steady-state situation, the voltage reference becomes as Equation 4.2. If the two test currents are chosen to be sufficiently high so that they fall within the flat region of the distortion voltage, then the distortion voltage values at these currents will be approximately equal to each other as Equation 4.3.

The difference in voltage between the two test points is provided in Equation 4.4. By using these equations, the stator resistance of PMSM can be calculated as Equation 4.5.

$$V_d = V_d I_d + V_{Dist} I_d \tag{4.2}$$

$$V_{Dist}(I_{d1}) = V_{Dist}(I_{d2}) \tag{4.3}$$

$$V_{d2} - V_{d1} = R_S(I_{d2}) + V_{dist}(I_{d2}) - V_{dist}(I_{d1}) = R_S(I_{d2} - I_{d1}) \tag{4.4}$$

$$R_s = \frac{(V_{d2} - V_{d1})}{(I_{d2} - I_{d1})} \tag{4.5}$$

4.5 Transients

Ohm's law can only be applied when steady-state constant current and voltage coexist. All electrical machines have resistive and inductive components in their stator windings, thus every change in current or voltage causes a transient. Therefore, as illustrated in Figure 4.2, the stator resistance estimation algorithm should be started once the stator flux reaches steady state. Therefore, enough waiting time should be allotted before activating the stator resistance estimator to take into account this flux settling period. The stator flux settling time and current settling time for synchronous motors are proportional. The stator flux settles when the current reaches its steady-state constant value. To determine the waiting time, the current loop time constant can be used. There isn't a set upper limit on this waiting period in actuality. However, lower than-necessary waiting time could lead to inaccurate parameter estimation. Therefore, selecting conservative waiting times is safe. For instance, it is possible to use several multiples of the current loop settling time in PMSMs.

4.6 Measurement Noise

In real life, measurement noise is inescapable. Therefore, employing noisy voltage and current while directly applying Ohm's law may result in some estimation errors. Voltage and current should therefore be filtered in some way. In this study, we use N samples of simple averaging. As stated in Equation 4.6, this equates to Figure 4.2. The mathematical definition of the average is Equation 4.7 for voltage and Equation 4.8 for current.

$$t_{test} = N.T_s \tag{4.6}$$

$$V_{d1} = \frac{1}{N}\sum_{n=1}^{N} V_{d1}[n] \tag{4.7}$$

$$I_{d1} = \frac{1}{N}\sum_{n=1}^{N} i_{d1}[n] \tag{4.8}$$

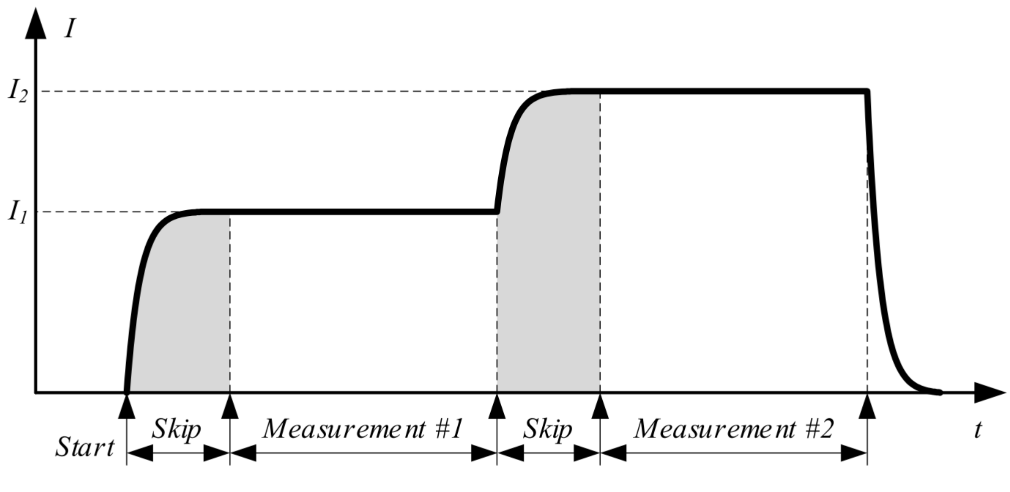

Figure 4.2 : Two stepped measurement of resistance [37].

4.7 Experimental Results

For the real-time verification of the stator resistance estimation, the structure in Figure 4.3 is used. The field-oriented control algorithm is used, and constant rotor position is used in Clarke/Park transformations. In addition to structure, Figure 4.3 includes a test current waveform.

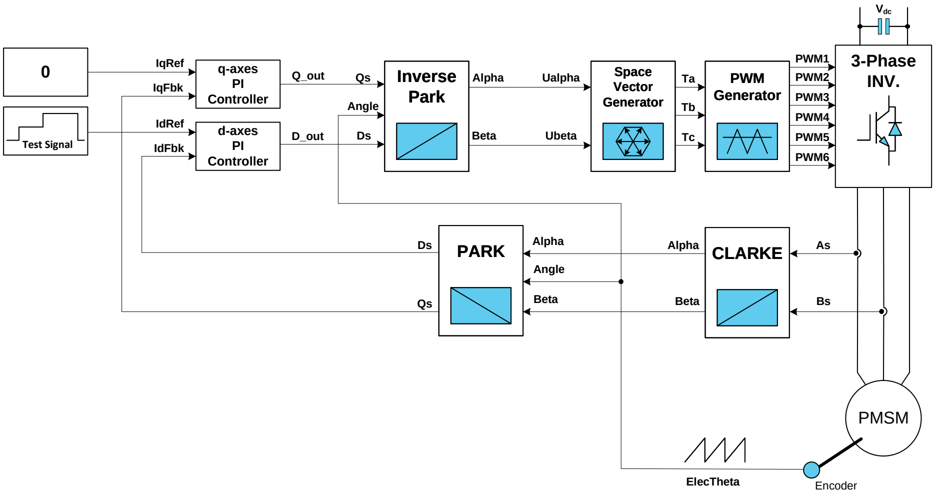

Figure 4.3 : Test structure for stator resistance estimation of PMSM.

Current and voltage amplitudes are filtered and detected by using Equation 4.7 and Equation 4.8. The stator resistance is then calculated in Equation 4.4 using those values. Real-time experiment results are shown in Figure 4.4. The whole algorithm is summarized in Figure 4.5. By using a real-time development kit, an experiment has been carried out with the motor in Table 5.2. The stator resistance estimation result is 1.64 Ω.

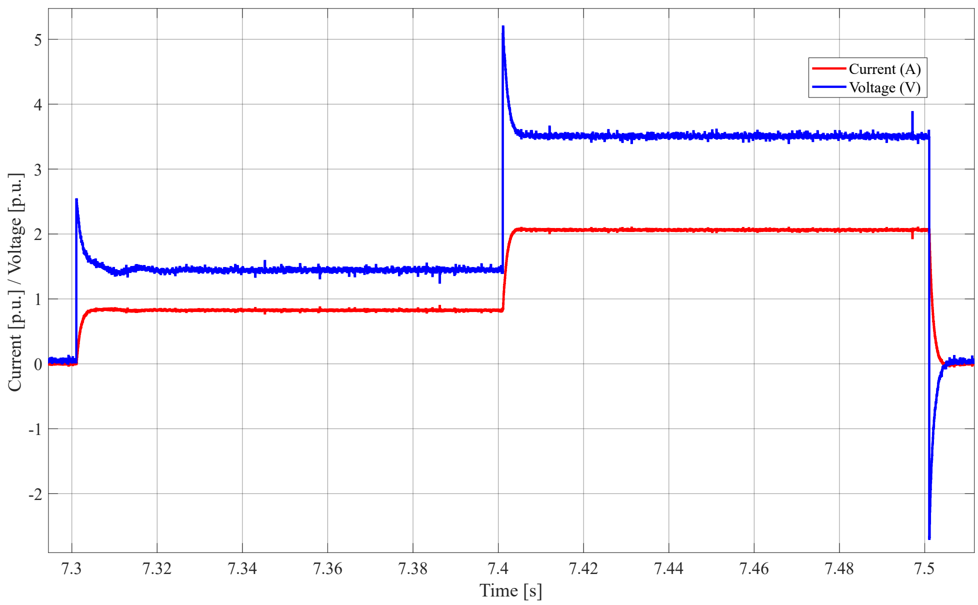

Figure 4.4 : Real-time experimental result.

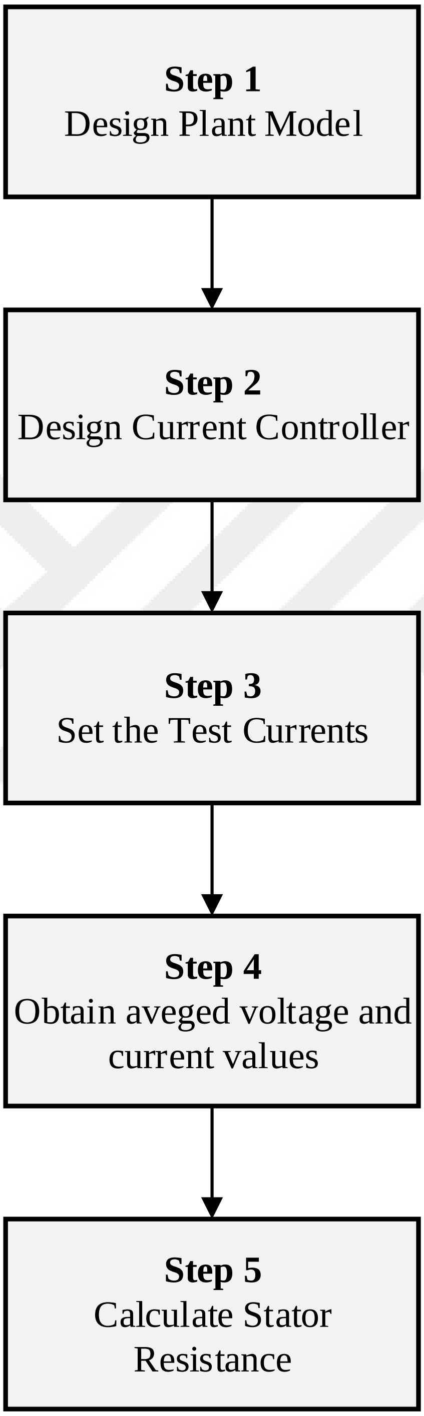

Figure 4.5 : Stator resistance estimation algorithm.

5. SIMULATION AND EXPERIMENTAL RESULTS

The simulation and implementation part includes two main parts, designing the control system simulation and designing the stator resistance estimation algorithm respectively. In addition, the simulation phase and implementation phase includes both the same and different parts. For instance, the control system design part is a common part for both computer simulation and real-time code generation part. But for the implementation phase, the model that will generate C code does not have a plant model subsystem. Instead of a plant subsystem, the code generation model will have blocks which are related to microcontroller peripherals like ADC, ePWM, eQEP, etc.

Since the PU system is used in our simulations, this topic was explained at the beginning of this part briefly. Then to understand the background of the code generation process, detailed information was given under the title "Embedded Code Generation Background". Following these parts, firstly, the electrical plant model and control system was developed respectively. Subsequently, the developed model was adjusted to code generation, and the stator resistance estimation algorithm was added to the real-time code generation model. In the end, the estimation result by using software and the measured result by using an RLC meter were compared.

5.1 Per-Unit System

The per-unit (PU) system is a widely adopted method in electrical engineering to express the values of various quantities such as current, voltage, speed, power, and torque. This system is particularly useful in power system analysis for AC machines and transformers. This system is utilized by embedded system engineers, for scalability and optimized code generation, particularly when working with fixed-point targets. In the PU system, a value of a specific quantity (such as current, voltage, speed, power, and torque) is expressed in terms of a base quantity in Equation 5.1, facilitating easy comparison and standardization across different systems and applications.

$$quantity\ \ expressed\ \ in\ \ PU = \frac{quantity\ \ expressed\ \ in\ \ SI\ \ units}{base\ \ value} \tag{5.1}$$

Selecting the appropriate base value is of great importance in the PU system as it establishes the scaling factor for representing other quantities in the system. In most cases, the base values in the per-unit (PU) system are chosen as the nominal values of the system. However, there are situations where the maximum measurable value of a quantity can also be considered as the base value. Therefore, In order to determine the base values, inverter, and motor parameters should be known.

In our simulation, the base voltage value is selected as the highest phase voltage that the inverter can supply. Mostly, for space vector pulse width modulation technique, it is;

$$Base\ Voltage = \frac{Maximum\ Inverter\ Supply\ Voltage}{\sqrt{3}} = \frac{24}{\sqrt{3}} = 13.8564\ V \tag{5.2}$$

for sinusoidal pulse width modulation technique, it is;

$$Base\ Voltage = \frac{Maximum\ Inverter\ Supply\ Voltage}{\sqrt{2}} = \frac{24}{\sqrt{2}} = 16.9706\ V \tag{5.3}$$

Since we use space vector pulse width modulation Equation (5.2), the base value of voltage is selected as 13.854 V.

The base current value is the highest current that the inverter's current sensing circuit can measure. Typically, although not always, this value corresponds to the maximum current of the inverter. According to the inverter data sheet;

$$Base\ Current = 4.125\ A \tag{5.4}$$

The base speed of a motor refers to its rated or nominal speed, which is also the highest achievable speed under normal operating conditions, specifically at the nominal load and voltage, without requiring any field-weakening operation. According to the motor data sheet;

$$Base\ Speed = 5000\ rpm \tag{5.5}$$

The base torque Equation 5.6 of a motor is calculated based on the base current value. However, it is important to note that the motor may or may not have the ability to actually produce this amount of torque.

$$Base\ Torque = \frac{3}{2}\ x\ p\ x\ \lambda\ x\ Base\ Current \tag{5.6}$$

The base power of a system is obtained through the multiplication of the base current and base voltage values Equation 5.7.

$$Base\ power = \frac{3}{2}\ x\ Base\ Current\ x\ Base\ Voltage \tag{5.7}$$

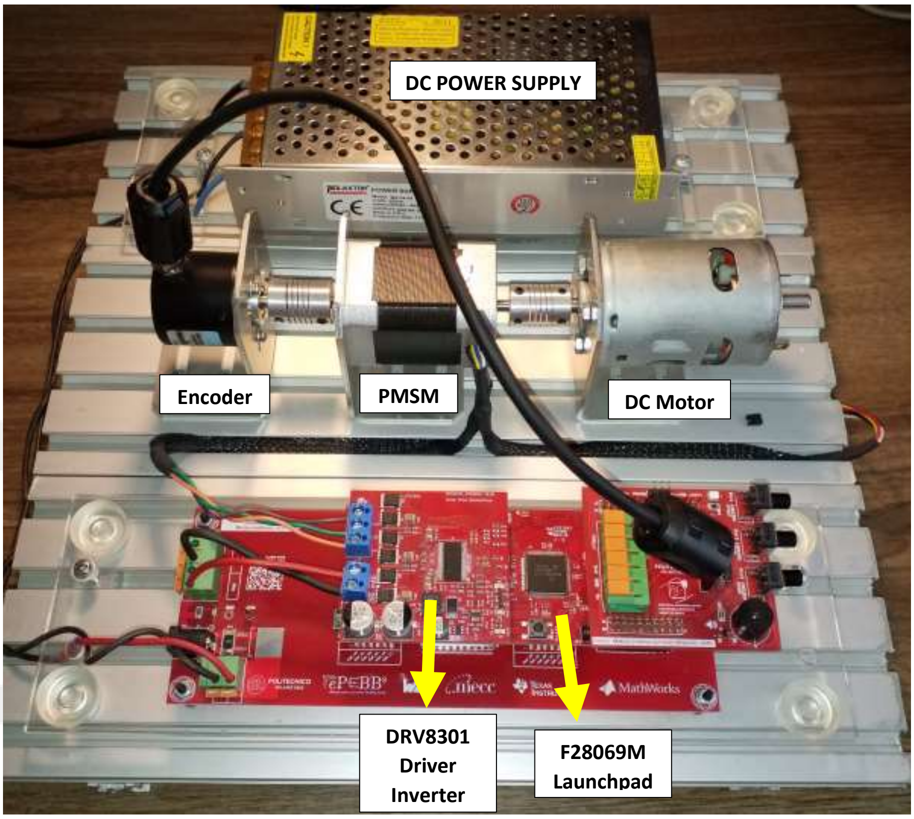

Figure 5.1 : Real-time development kit.

Figure 5.1 shows the real-time development kit.

5.2 Embedded Code Generation Background

In order to generate C code and deploy it to the Texas Instruments motor control algorithm development kit, Simulink® software, and its tools such as the controls system toolbox, Simscape Electrical Toolbox, Motor control Blockset, and Embedded Coder® Toolbox have been used in this thesis.

Simulink® possesses versatile features, allowing for the modeling of linear, nonlinear, continuous-time, discrete-time, multi-rate, and hybrid systems. Moreover, It provides work in various fields such as signal processing, controls, physical modeling, communications, and other systems engineering fields.

5.2.1 Generating embedded code

The purpose of Embedded Coder® is to automatically generate C code from a Simulink® model, which can be utilized for developing an application specific to a particular platform. Thanks to the automatic code generation process,

- The need for manual coding of algorithms in a Simulink® model is bypassed.
- The risk of coding errors due to the manual application is reduced.
- Sustain code quality consistently.
- Generate code with a single click increases productivity.

5.2.1.1 Code generation products

There are three different coder types: MATLAB® Coder™, Simulink® Coder™, and Embedded Coder®. MATLAB Coder is capable of generating standalone C and C++ code from MATLAB® code, which is portable and easy to read. It encompasses a subset of essential MATLAB® language functionalities, such as matrix operations, functions, and program control constructs. Figure 5.2 shows the Embedded Coder® classification.

Simulink® Coder™, previously known as Real-Time Workshop, executes and generates C and C++ code from Stateflow charts, Simulink® diagrams, and MATLAB functions. The generated code can be used for various applications, including non-real-time and real-time scenarios such as rapid prototyping, simulation acceleration, and hardware-in-the-loop testing. Code generated by Simulink® Coder can be tuned and monitored using Simulink®, or executed and interacted with independently outside of MATLAB®/Simulink®.

Embedded Coder® is an add-on product that enhances Simulink® Coder™, specifically designed for embedded software development. By utilizing Embedded Coder®, users can leverage the full capabilities of Simulink® Coder™ to generate code that possesses the clarity and efficiency equivalent to expertly crafted manual code.

Embedded Coder® provides extended capabilities for configuring MATLAB® Coder™ and Simulink® Coder™, allowing for advanced optimizations that offer precise control over the generated code's files, functions, and data. These optimizations not only

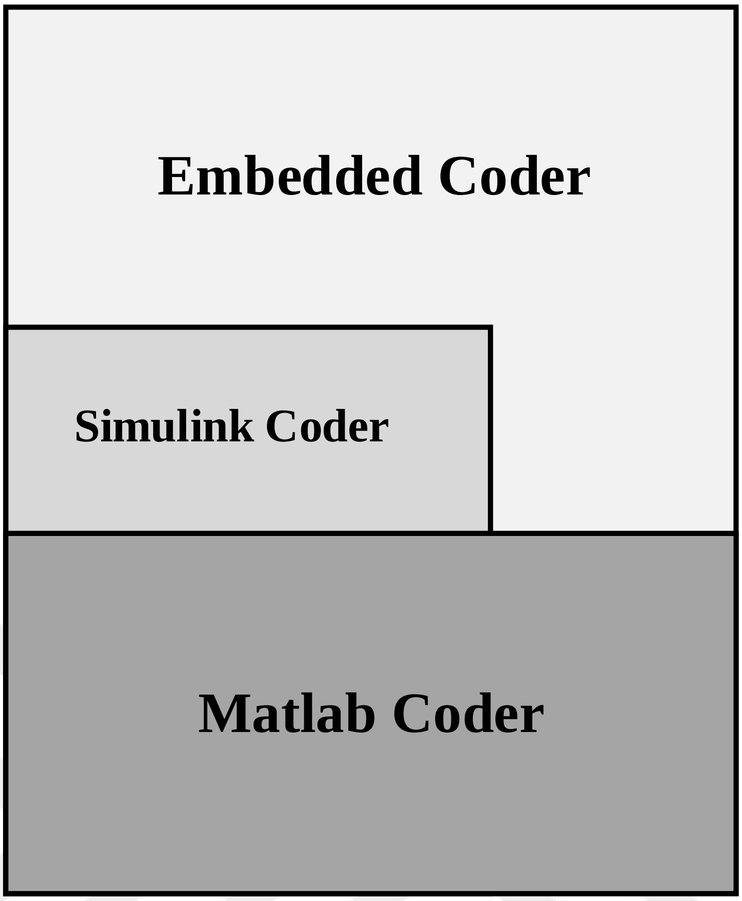

Figure 5.2 : Embedded coder® classification.

enhance code efficiency but also facilitate seamless integration with legacy code, calibration parameters, and data types used in production environments.

5.2.1.2 Capabilities of embedded coder®

Embedded Coder® offers a comprehensive framework for creating production code that is optimized for factors such as memory usage, speed, and simplicity.

Embedded Coder® generates highly optimized ISO-C or ANSI-C code that is tailored for floating-point and fixed-point microprocessors. It builds upon the features offered by Simulink® Coder™, providing enhanced support for specification, integration, deployment, and testing of production applications on embedded targets. It extends the features offered by Simulink® Coder™, providing enhanced support for integration, specification, testing, and deployment of production applications on embedded targets.

Embedded Coder® specifically caters to targeting considerations, including constraints related to RAM, ROM, CPU resources, code configuration, and code verification, to ensure optimal performance and efficient utilization of resources in embedded systems.

The Embedded Real-Time (ERT) target, which is a part of Embedded Coder®, is designed to be customizable. Many users require code generation for specific microprocessors or development boards, along with deployment on target hardware using a cross-development system. In order to achieve this, certain modifications to the ERT target files may be necessary.

- Generate C/C++ code with memory usage, readability, and execution speed comparable to handwritten code.
- Verification of code by automatically importing it into Simulink® for SIL testing.
- Generate code from Simulink® data objects, including user-defined types, storage classes, and aliases.
- Package generated code to comply with specific standards and software styles.
- Generate, deploy, and test embedded systems on specific hardware boards such as Texas Instruments development boards.

5.2.1.3 Architecture of an embedded application

The following components constitute an embedded application:

- a model application that encompasses the algorithmic logic of the system
- a main program that invokes or calls the model application.
- a run-time library, which gathers commonly used operation utilities that the model application can access.
- data logging interface
- a data exchange interface that enables the model application to interact with the host during runtime

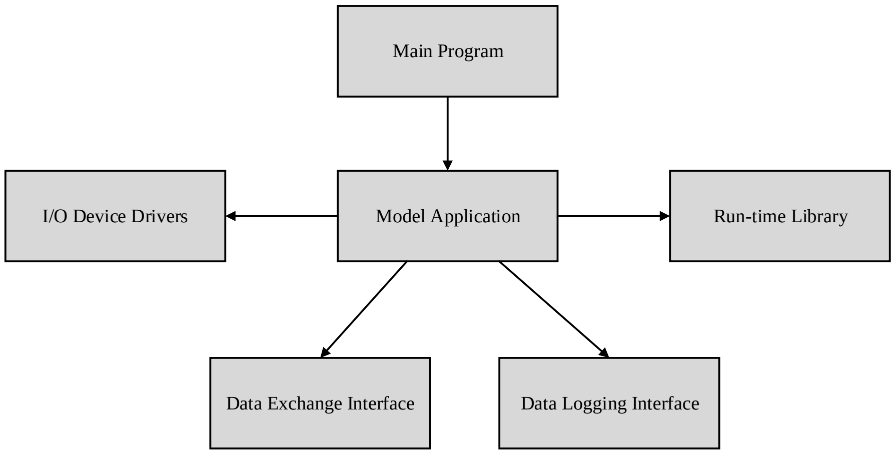

Figure 5.3 : Embedded application structure.

- a collection of driver routines used by the model application to communicate with I/O devices.

Figure 5.3 shows the embedded application structure. Embedded Coder® generates only the model application among these components. Additionally, Embedded Coder® has the capability to create an optional data logging interface called MAT-file logging. The remaining components are provided as a run-time interface during the link time.

5.2.1.4 Code modules

Code modules are shown in Figure 5.4. Embedded Coder® generates a build directory within your current working directory where it stores the source code that it generates. This build directory also includes a makefile, object files, and other files that are created during the code generation. By default, the build directory is named as "model_ert_rtw", where "model" refers to the name of your Simulink® model. Here is a summary of the files that are generated:

- The file "*model.c*" serves as an entry point for all the code that implements the algorithm of the model.
- The file "*model.h*" declares the data structures of the model and provides a public interface to the data structures and entry points of the model. It also includes an interface to the real-time model data structure (*model*_M) with accessor macros.

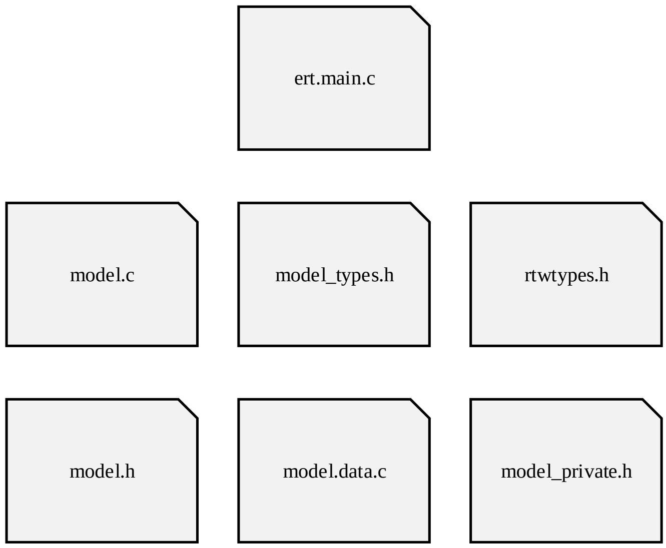

Figure 5.4 : Code modules.

- The "*model_private.h*" file includes data and local macros that are necessary for the model and its subsystems, as well as externally defined data that is given by the model code. When the model has nontrivial content, this file is included by the generated source files in the model.

- The "*model_types.h*" file includes forward declarations for the parameters data structure and the real-time model data structure. These declarations may be necessary for function declarations of reusable functions.

- The "*rtwtypes.h*" file defines various structures, macros, and data types that are necessary for Embedded Coder® These definitions are required by most other generated code modules

- The file "*ert_main.c*" is a default example main program that is generated by Embedded Coder®.

- The file "*model_data.c*" is generated conditionally, meaning it is only created when certain conditions are met. It includes declarations for constant block I/O data structure and the parameters data structure. If these structures are not used in the model, "*model_data.c*" will not be generated.

5.2.1.5 The main routine

The main routine implemented in *ert_main.c* file is an example main program, which Embedded Coder® generates by default. This main routine calls for three major functions in the following order:

- *model_initialize* is the first major function and called the beginning of execution. This function is implemented in *model.c*. Furthermore, it contains initialization code for:
  - error status
  - external outputs and inputs
  - States (DW), the initial condition of states
- The second major function called in the *ert_main.c* file is the *"rt_OneStep"* function. This function is called at every timer interrupt and is implemented within the *ert_main.c* file. The *"rt_OneStep"* function calls the *"model_step"* function, which performs a one-time step of the model. By default, the call to *"rt_OneStep"* is commented out and needs to be added to an interrupt-service routine or timer with a period that corresponds to the model's sample time.
- The function *"model_terminate"* is invoked at the end of the program's execution. It is implemented in the *"model.c"* file and includes all essential tasks related to the program's termination.

5.2.1.6 Simulink® data in the generated code

Simulink® data is composed of the following elements:

- External inputs
- External outputs
- Intermediate signals
- Discrete states (delays)

- Shared memory (data store memory)
- Parameters
- Timing information

Symbols of data structures are as shown Table 5.1.

Table 5.1 : Symbols of data structures.

	Data type naming	Variable name
Model Data	RT_MODEL_*model*_T	*model*_M
Parameters	P_*model*_T	*model*_P
External Inputs	ExtU_*model*_T	*model*_U
External Outputs	ExtY_*model*_T	*model*_Y
Block Signals	B_*model*_T	*model*_B
Block States	DW_*model*_T	*model*_DW

5.2.1.7 Build process

In order to build a program or code Generation process, an Embedded Coder® carries out the following build process:

- Model Compilation
- Code Generation
- Executable generation (optional)

The block diagram in Figure 5.5 illustrates how Embedded Coder® controls the process of automatic program building.

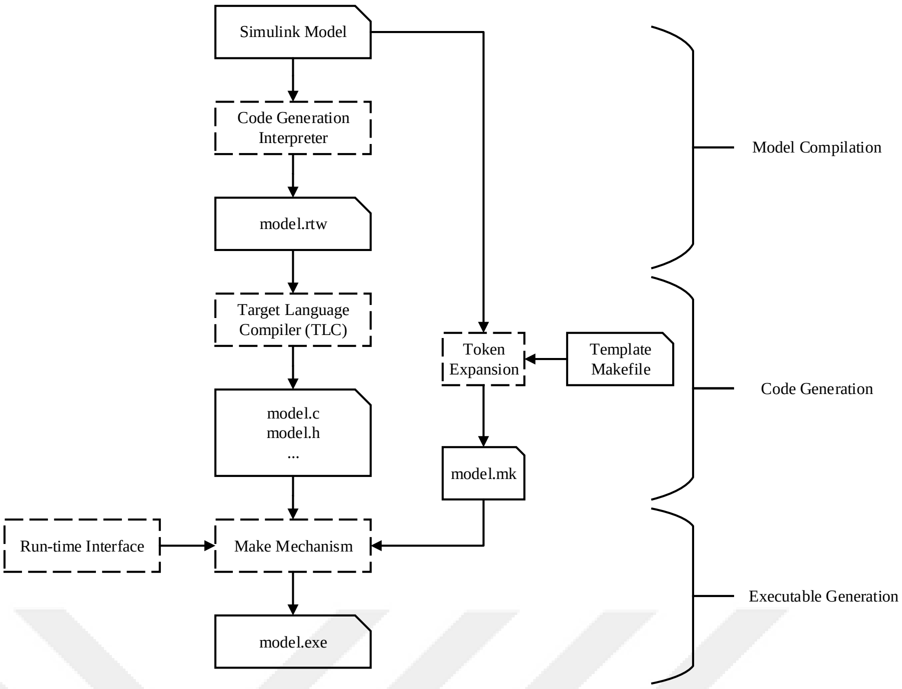

Figure 5.5 : Build process.

5.2.1.8 Model compilation

The process of building starts when the block diagram is compiled by the Embedded Coder® and the process is shown as in Figure 5.6. At this point, Simulink® carries out the following tasks:

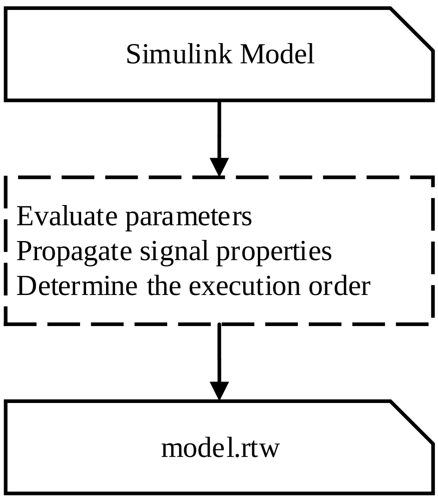

Figure 5.6 : Build process: model compilation.

- Evaluates block and simulation parameters

- Propagates sample times and signal widths
- Specifies the execution order of blocks inside of the model

After completing the aforementioned tasks, Simulink® compiles an intermediate representation of the model. The compiled model, which includes information on inputs, outputs, blocks, parameters, storage, states, and other components and properties, is stored in the *"model.rtw"* file in a format that is not specific to any programming language. The input of the next step of the build process is this *"model.rtw"* file. Figure 5.6 shows the model compilation process.

5.2.1.9 Code generation

To generate code that is specific to the target platform, the Embedded Coder® code generator employs a Target Language Compiler (TLC) and a TLC function library. The TLC and its associated function library are responsible for transforming the intermediate model description, which is stored in the model.rtw file, into code that is compatible with the target hardware and software environment. Figure 5.7 shows the code generation part of the system.

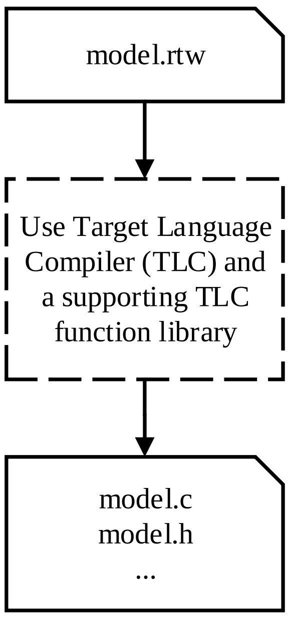

Figure 5.7 : Build process: code generation.

5.2.1.10 Executable generation

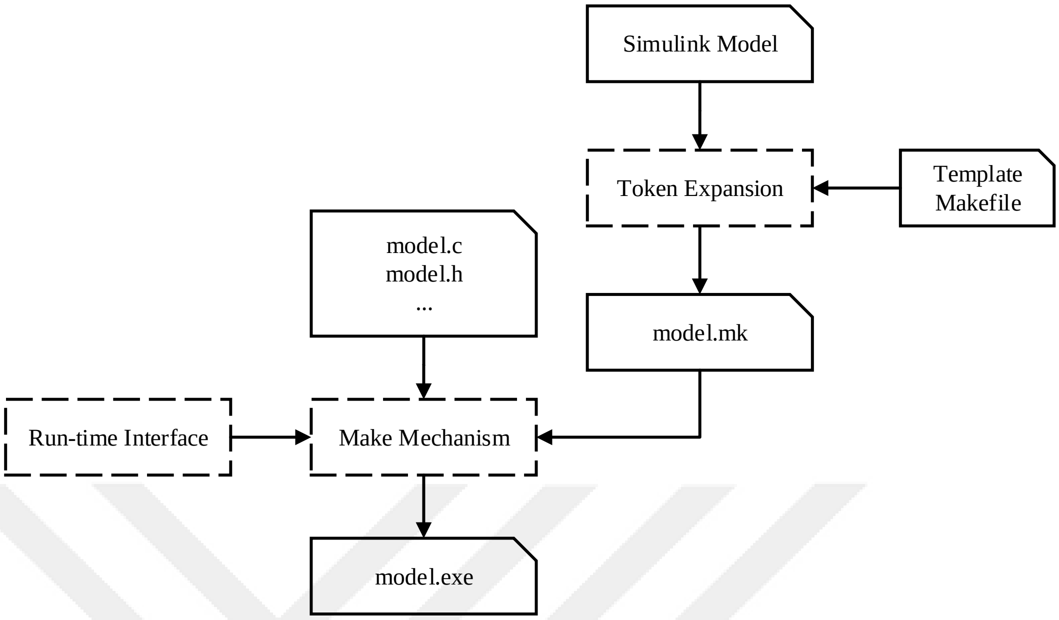

Figure 5.8 : Build process: executable generation.

The details of the executable generation are shown in Figure 5.8. After explaining the background of the code generation process by using Simulink® and Embedded Coder® Toolbox, I can explain real-time speed control system design and offline stator resistance estimation design by using these tools step by step.

5.3 Plant Model Creation

Creating an accurate plant model is crucial in developing a motor control system. This allows for verification of the control system's functionality, conducting tests through closed-loop model-in-the-loop simulations, tuning controller gains, and optimizing the algorithm before deploying it in the actual plant. In summary, an accurate plant model is a necessary component in motor control system development to ensure efficiency and effectiveness in its implementation.

When you want to create a plant model, there are two ways, mathematical modeling, and physical modeling. During this thesis physical modeling approach will be used due to its advantages over mathematical modeling while creating code generation. However, to understand the main philosophy of the system design, the mathematical modeling of some systems was explained in the previous chapters. For the physical modeling approach, Simulink® includes different libraries such as Motor Control Blockset, Simscape Electrical, etc.

These libraries include electrical motors such as DC motors, permanent magnet synchronous motors, and induction motors. They also have inverters, signal conditioning circuits, processor peripherals such as Pulse width modulators (PWM), analog to digital converters (ADC), etc.

To ensure the accuracy and functionality of the plant model, there are several steps to follow. Firstly, the normalized PWM duty cycle from the control algorithm needs to be read. Secondly, simulating the motor for the connected load is essential. Finally, the output motor phase current and the output motor position from the simulation, in terms of ADC counts and encoder pulse counts respectively, can be obtained. By following these steps, the plant model can be thoroughly verified, ensuring that it meets the necessary requirements for effective implementation in motor control system development. To create a plant model for motor control system development, there is a specific workflow that should be followed. This workflow can be identified as developing a plant model and then developing its software part, which has field-oriented control.

5.3.1 Motor and inverter design

During this thesis, Simscape Electrical is used. This library has a PMSM motor which has all nonlinear specifications. The main speed control algorithm is shown in Figure 5.9.

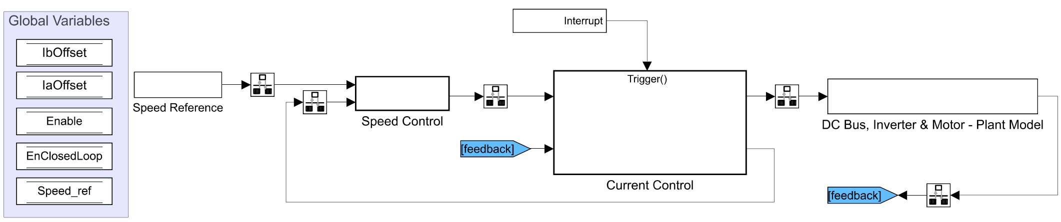

Figure 5.9 : Main speed control algorithm.

The motor parameters that we need to design PMSM can be found by looking at the motor datasheet in Table 5.2.

Table 5.2 : PMSM datasheet values.

Electrical Specifications	Value	Unit
Phase	3	Phase
Pole	8	Poles
Rated Voltage	24	Volt
Rated Speed	4000	Rpm
No-load Speed	6500	Rpm
Rated Torque	0.125	Newton-meter
Rated Current	3.4	Ampere
No Load Current	0.7	Ampere
Rated Power	52.5	Watt

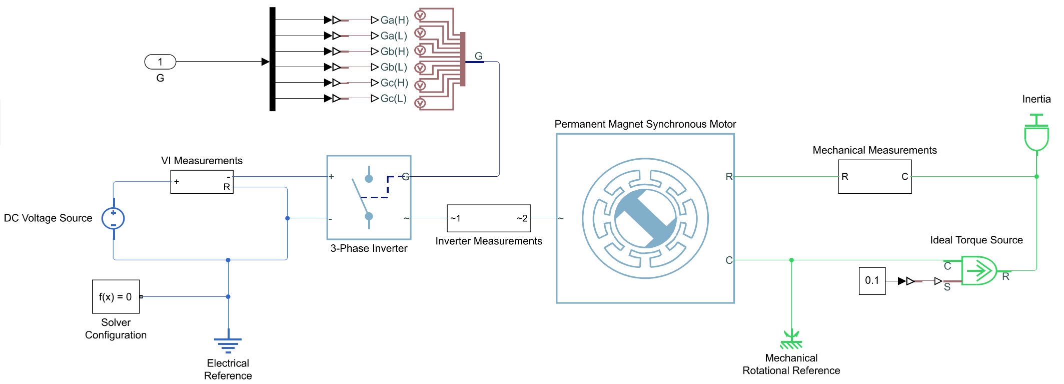

Figure 5.10 : DC bus inverter PMSM plant model.

As we have seen in Table 5.2 there is no stator resistance information. In order to continue to design the control algorithm we need this value. So, as stator resistance, we gave an approximate value by looking at the same power rate motors data sheets. Because we need a controller to estimate stator resistance in the future. At the same time, we also can obtain stator resistance value by using an RLC meter.

Main speed control system which is shown in Figure 5.9 involves a DC Bus, an inverter, a motor plant model, a current controller, and a speed controller. To control the motor, a field-oriented control algorithm has been selected.

DC Bus, inverter, and motor plant subsystem which is shown in Figure 5.10 have a 24-volt DC voltage source, an inverter, a PMSM, an ideal torque source, an inertia block, and sensors. In order to both measure inverter (Figure 5.11) and DC Bus (Figure 5.12) currents and voltages, there are current and voltage sensors as well as mechanical sensors (Figure 5.13) such as an ideal torque sensor and an ideal rotational motion sensor to measure torque and speed. The input of the inverter is 24-volt DC voltage and the output of the inverter is connected to the PMSM. In order to control the inverter by using space vector pulse width modulation, there is a control input. Figure 5.11 measures the 3-phase inverter voltage and current.

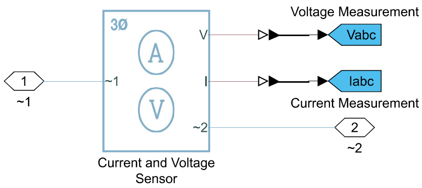

Figure 5.11 : Inverter measurements.

DC Bus Measurements subsystem measures the DC Bus voltage and current as shown in Figure 5.12.

In the mechanical measurements subsystem, motor mechanical outputs are measured such as angle, torque, and speed as shown in Figure 5.13.

5.3.2 Motor phase current and position sensing subsystem

The output of the physical system measurements is analog signals. In order to use these values they should be converted to digital parameters by using analog-to-digital converters. This operation is named sensing. In our model, there are two scaling parts as shown in Figure 5.14

The first part is current scaling. In real physical hardware, the PMSM current read by the sensors is first filtered and then scaled to an analog-to-digital converter measurable range. The analog-to-digital converter peripheral of the processor reads the current signal and then outputs the analog-to-digital converter counts as a feedback signal for the current control algorithm. In order to determine the ADC counts corresponding to the current flowing through the motor, the maximum measurable peak current is first

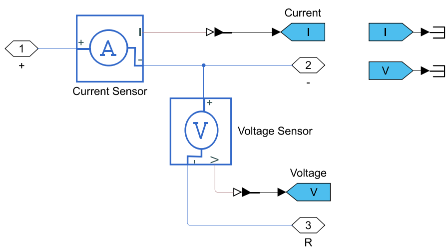

Figure 5.12 : DC bus measurements.

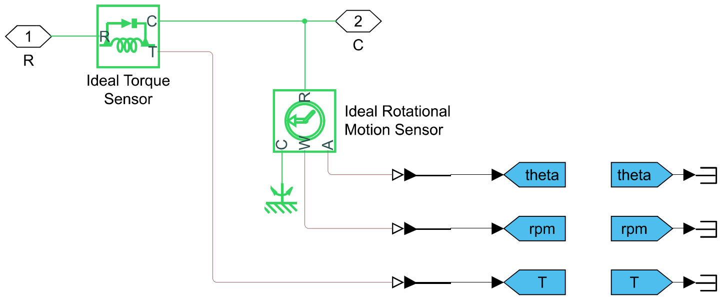

Figure 5.13 : Mechanical measurements.

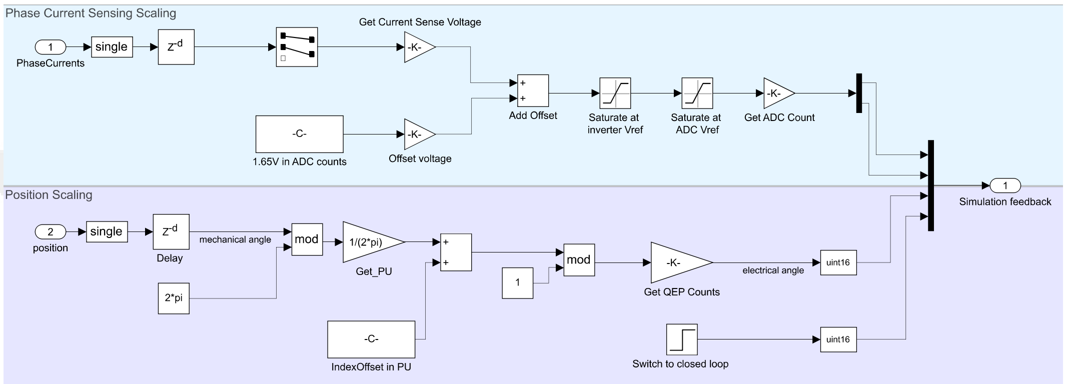

Figure 5.14 : Current and position sensing subsystem.

established as a reference point. Using this base value, along with the full-scale ADC values and ADC offset, the ADC counts can then be calculated using (5.8).

For our inverter, BOOSTXL-DRV8301, the ADC offset values for both phase-A and phase-B are 2048 according to its datasheet.

$$ADC\ counts = \left(\frac{\frac{Full\ scale\ ADC\ counts}{2}}{Base\ current} \right) + ADC\ offset \tag{5.8}$$

The second part is the position scaling. The motor position is read as the mechanical position of PMSM in the unit of rad/s by using a rotational motion sensor. Then this position information is converted from $0-2\pi\ rad/s$ to QEP encoder counts as shown in Figure 5.14.

Adding delays in the plant model is crucial to increase the reality of the simulation. Therefore, pulse width modulation switching delay and control algorithm processing delay that we encounter in the real system can be added to the simulation as a delay block, as shown in Figure 5.14.

Up to this part, all physical system has been designed. This side of the simulation just will be used to design the controller and estimation algorithm. For the code generation part, this part will not be used. Instead of this part, microcontroller peripheral blocks are used, and how to use this peripherals part will be explained in the future.

5.4 Current control subsystem development

This subsystem includes current and position sensing as well as field oriented control algorithm. SVPWM design and PWM scale block is in this subsystem, and all these block is explained one by one in detail. After designing these controllers, a stator resistance estimation algorithm is added. Following this part, a sliding mode observer is added instead of an encoder sensor.

5.4.1 Current and position sensing subsystem

In the first part of the subsystem, current information is converted from ADC counts to per-unit value. IaOffset and IbOffset data store memory blocks are the Analog to digital converter offsets for current measurement, and they are hardware specific.

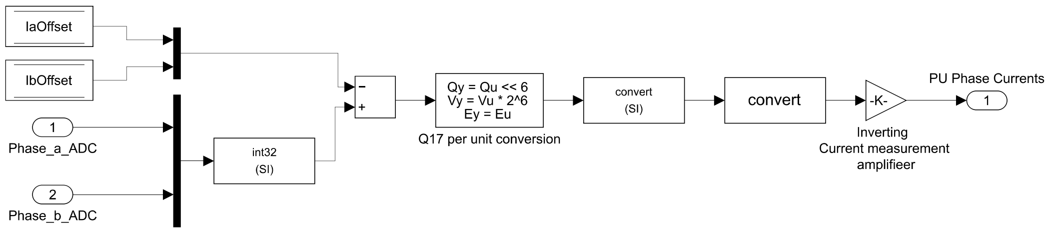

Figure 5.15 : Current sensing.

These blocks as shown in Figure 5.15 are also used to share data between the current and position subsystems.

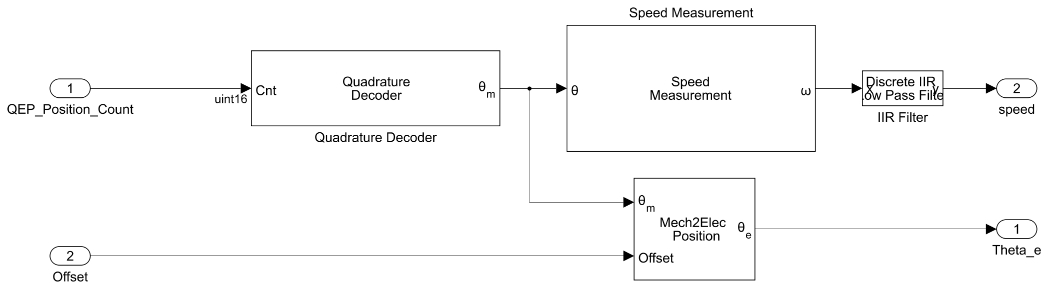

Figure 5.16 : Position sensing.

Thanks to the subsystem in Figure 5.16, the rotor position is read from the QEP pulse count. The quadrature decoder reads the position count from the hardware driver block or plant model. Then rotor mechanical position is converted in encoder position counts to the rotor mechanical angle in PU.

5.4.2 Field oriented control subsystem design

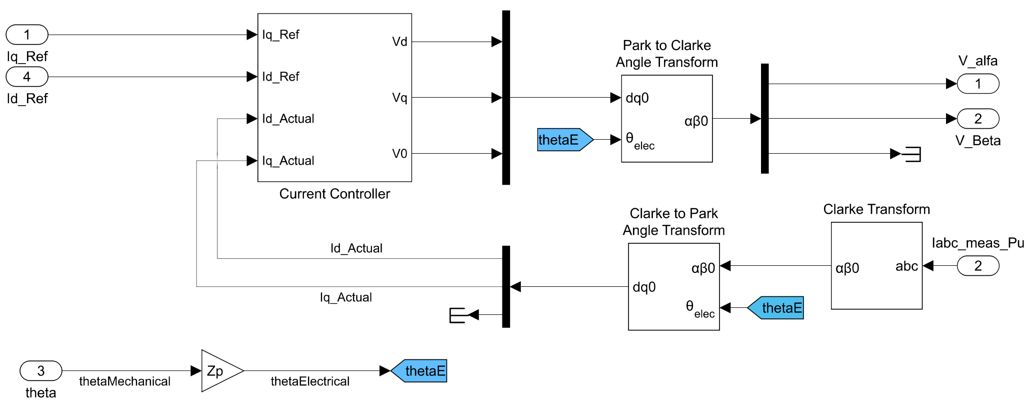

Figure 5.17 : Current control subsystem.

Details of field-oriented control have been explained in the previous chapters in detail. The algorithm has a nested structure that has a current controller, speed controller, and position controller if there is one. Therefore the design of the algorithm should start from the current control subsystem and continue outward. Therefore, the current control subsystem and speed control subsystem are designed, respectively.

The current controller subsystem which is shown in Figure 5.17 has a Park, Clarke, and inverse Park to utilize field-oriented control algorithms. In addition, there are two controllers to control $d-$axis and $q-$axis. While the $d-$axis represents the flux, the $q-$axis represents the torque, as we explained earlier.

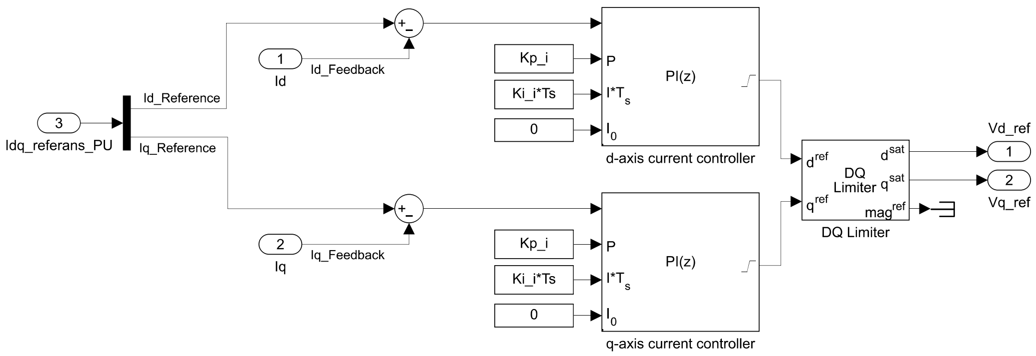

Figure 5.18 : Current controller.

The reference of the $d-$axis is zero, and the reference of the torque comes from the speed controller output. Thanks to this way, controlling the $q-$axis current, PMSM torque can be controlled. In addition, to obtain the electrical angle which is used in park transformations, the mechanical angle is multiplied by the pole pair. In order to control currents, $d-$axis current controller and $q-$axis current controller are shown in Figure 5.18.

Controller parameters have been designed in Chapter 3 according to Equation 3.19, Equation 3.20. In the first step, these parameters will be used. After finishing the whole simulation, by using Response Optimizer Toolbox, these parameters will be updated.

5.4.3 Space vector pulse width modulation design

Space vector pulse width modulation (SVPWM) has become the widely used and significant pulse width modulation (PWM) technique for three-phase voltage source inverters [38]. It is used to control various types of motors such as switched reluctance, brushless DC, AC induction, and permanent magnet synchronous motors [39]. This modulation technique is the crucial part of the field-oriented control algorithm [40]. The whole algorithm of SVPWM is shown in Figure 5.19.

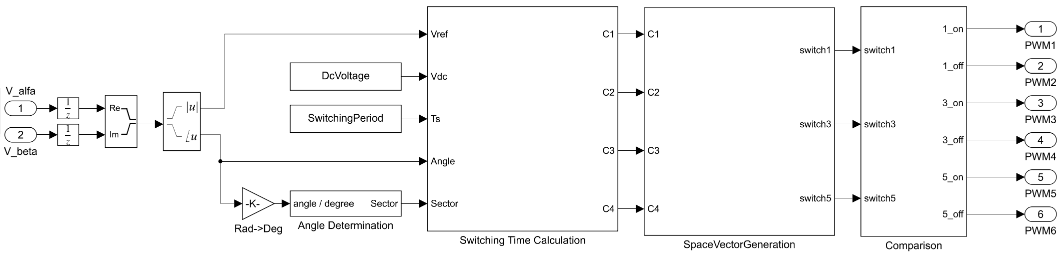

Figure 5.19 : Space vector pulse width modulation.

The realization of space vector pulse width modulation includes three steps. Firstly, V_d, V_q, V_{ref}, and $angle\ \alpha$ should be determined according to Equation 5.9, Equation 5.10. V_d and V_q are generated after the current controller subsystem. By using V_d and V_q, V_{Ref} can be determined according to Equation 5.9.

$$|V_{Ref}| = \sqrt{(V_d)^2 + (V_q)^2} \tag{5.9}$$

After obtaining reference voltage, by using V_d and V_q, angle α can be determined according to Equation 5.10.

$$\alpha = \arctan\left(\frac{V_q}{V_d}\right)^2 \tag{5.10}$$

Sector determination is also crucial and required to calculate switching time. According to the angle which is calculated in Equation 5.10 it is determined as shown in Figure 5.20.

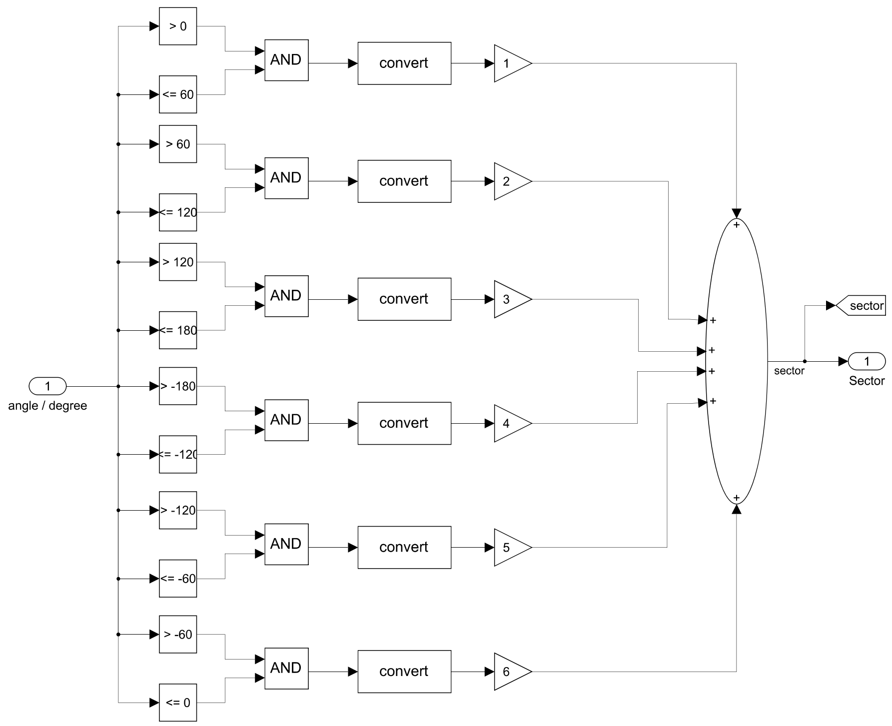

Figure 5.20 : Sector determination.

The second step of the SVPWM generation is determining the switching time duration at any section. These times are calculated according to Equations 5.11–5.13.

$$T_1 = \frac{\sqrt{3}.T_z.|V_{Ref}|}{V_{DC}}.\sin\left(\frac{n}{3}\pi - \alpha\right) \tag{5.11}$$

$$T_2 = \frac{\sqrt{3}.T_z.|V_{Ref}|}{V_{DC}}.\sin\left(\alpha - \frac{n-1}{3}\pi\right) \tag{5.12}$$

$$T_0 = T_Z - T_1 - T_2 \tag{5.13}$$

For the above equations, n is the sector. The switching time calculation design is shown in Figure 5.21.

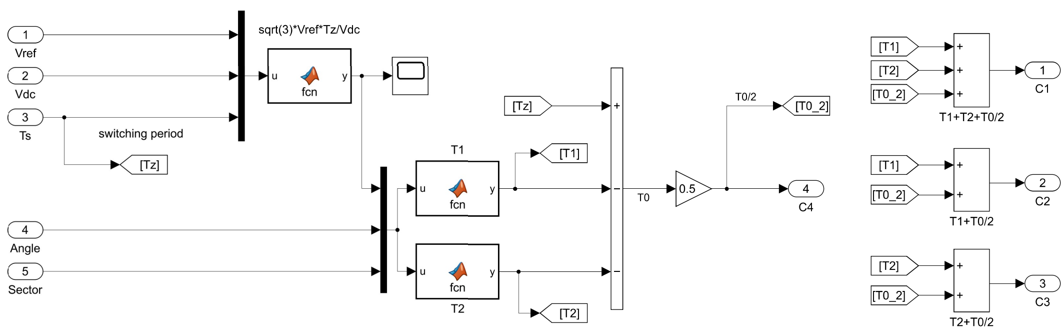

Figure 5.21 : Switching time calculation.

Thirdly, the switching time of each transistor should be determined according to the Table 5.3.

Table 5.3 : Switching sequence table.

Sector	Upper Switches	Lower Switches
1	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
2	$S_1 = T_1 + T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_0/2$	$S_4 = T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_2 + T_0/2$
3	$S_1 = T_0/2$ $S_3 = T_1 + T_2 + T_0/2$ $S_5 = T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_0/2$ $S_2 = T_1 + T_0/2$
4	$S_1 = T_0/2$ $S_3 = T_1 + T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_2 + T_0/2$ $S_6 = T_2 + T_0/2$ $S_2 = T_0/2$
5	$S_1 = T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_2 + T_0/2$	$S_4 = T_1 + T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_0/2$
6	$S_1 = T_1 + T_2 + T_0/2$ $S_3 = T_0/2$ $S_5 = T_1 + T_0/2$	$S_4 = T_0/2$ $S_6 = T_1 + T_2 + T_0/2$ $S_2 = T_2 + T_0/2$

The simulation of this each transistor switching time is shown in Figure 5.22.

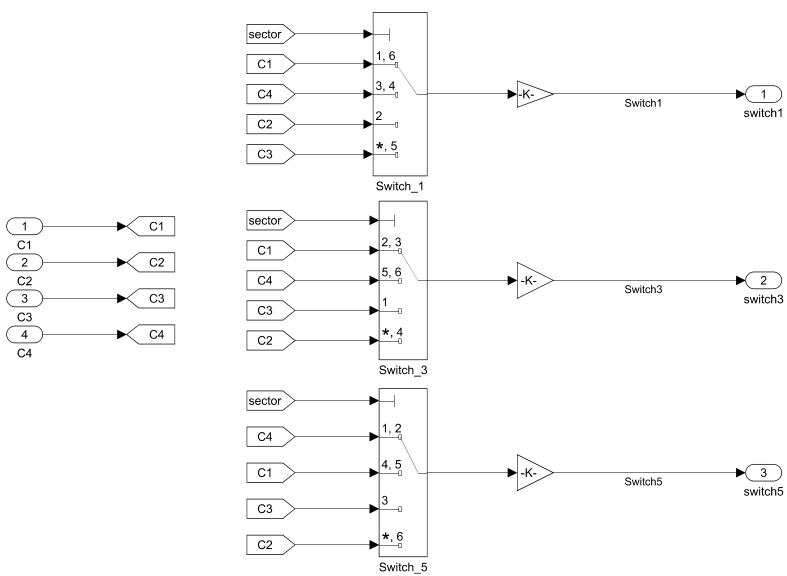

Figure 5.22 : Determination of each transistor switching time.

Lastly to control PMSM by generating space vector pulse width modulation signal, triangular and modulation waves should compare as shown in Figure 5.23.

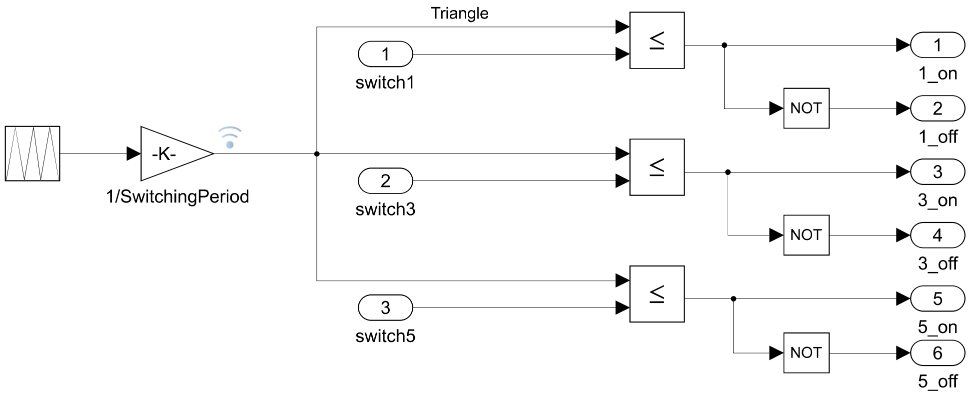

Figure 5.23 : PWM generation.

As a result, the generated space vector pulse width modulation signal is shown in Figure 5.24.

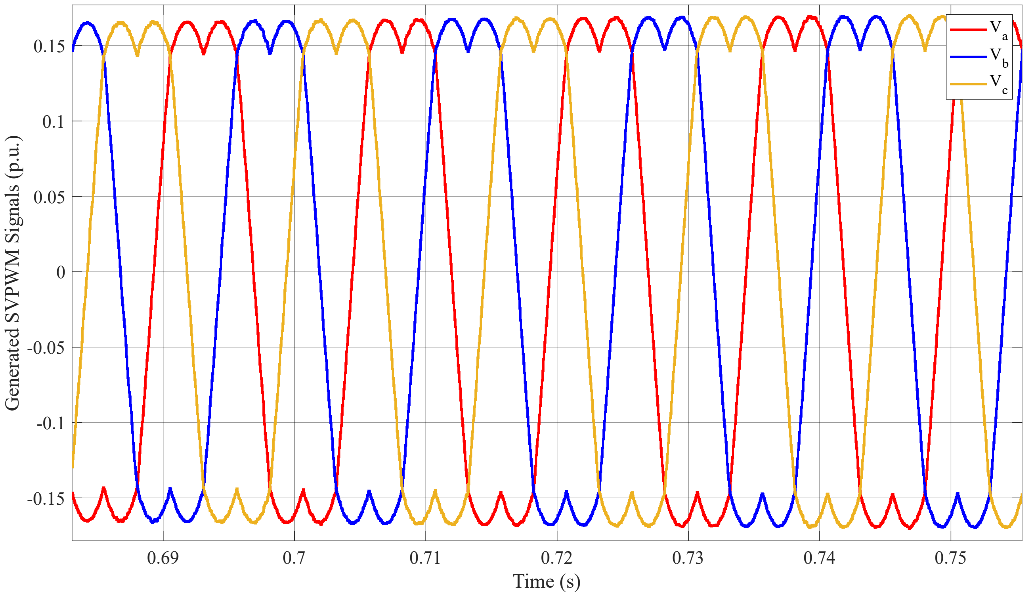

Figure 5.24 : Generated SVPWM.

5.4.4 Current controller parameters

As we introduced in Chapter 3, PMSM current controller can be tuned by using Equation 3.19 and Equation 3.20.

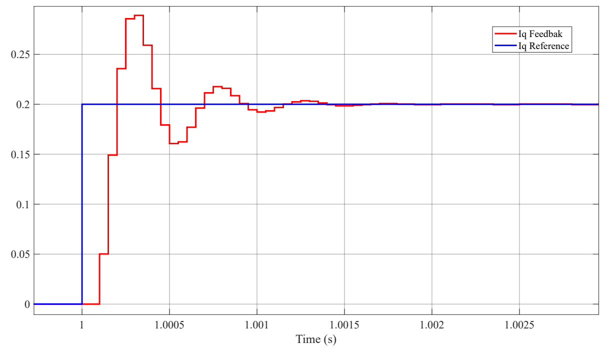

Figure 5.25 : $q-$axis current controller response.

In addition to this method, Simulink® allows us different methods to tune these parameters. By using field-oriented control autotuner block or Simulink® control design toolbox or model initialization script, control loop gains can be calculated effectively. If you are using a simulation environment like Simulink®, to tune the current controller there is an attractive way. If you lock the rotor and give the current reference value, you can adjust the controller parameters by examining the current response of the $d-$ and $q-$axes controllers. The current response of the $q-$axis current controller is as shown in Figure 5.25

The output of the current control subsystem will be the input of the space vector pulse width modulation, the output of the SVPWM is connected to the inverter. For the simulation part, this output is connected to the inverter which is shown in Figure 5.10. For the code generation model, the output of the SVPWM block is connected to the ePWM (Enhanced Pulse Width Modulation) blocks in order to generate C code which is available for the LAUNCHXL-F28069M evaluation board.

5.4.5 PWM scaling

In the space vector pulse width modulation subsystem PWM is generated. Then this subsystem outputs the normalized pulse width modulation cycles (0-1) for the plant model.

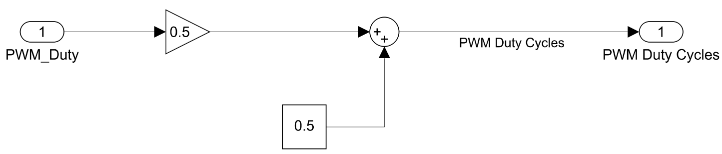

Figure 5.26 : PWM scaling.

5.5 Speed controller design

The output of the speed controller block is the reference of the $q-$axis current. The reference speed value is taken from the host model and it is limited to the nominal value of the motor. And then actual speed value is taken thanks to the sensor. The subtraction of the actual speed from the reference speed gives the error signal and it is the input of the speed controller. In addition, the reference speed value is filtered by using a low pass filter. This filter's duty is to reduce the torque ripple. Finally, controller parameters are calculated according to Equation 3.33 and Equation 3.34. Our controller is as shown in Figure 5.27.

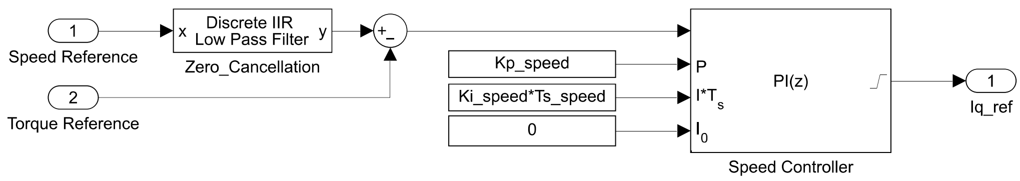

Figure 5.27 : Speed controller subsystem.

The speed response of our speed controller is shown in Figure 5.28.

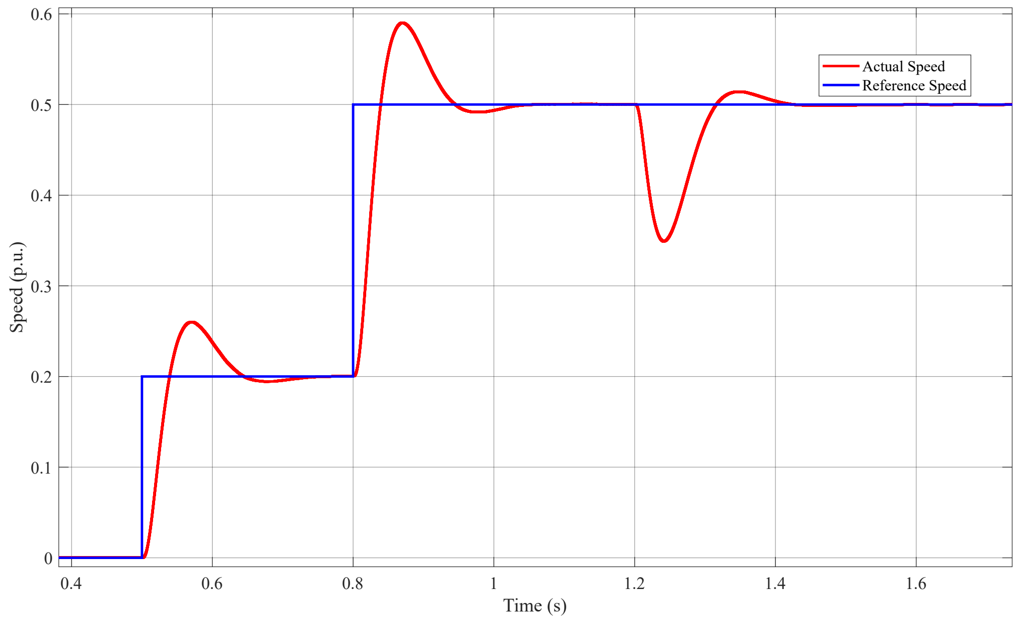

Figure 5.28 : Speed controller response.

The q-axis current response of our system according to speed reference is shown in Figure 5.29

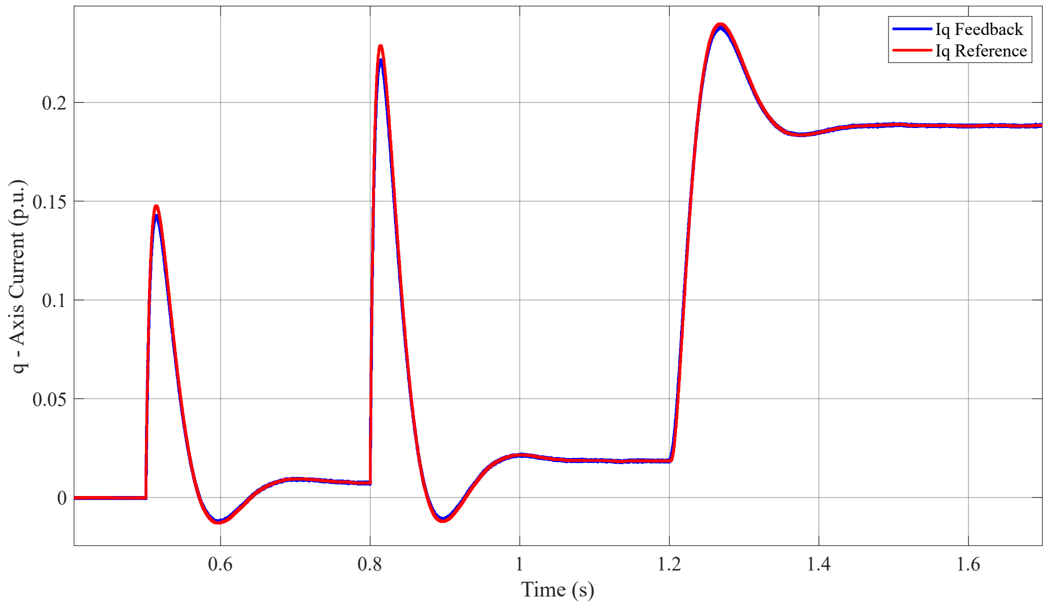

Figure 5.29 : $q-$axis controller response.

The d-axis current response of our system according to speed reference is shown in Figure 5.30

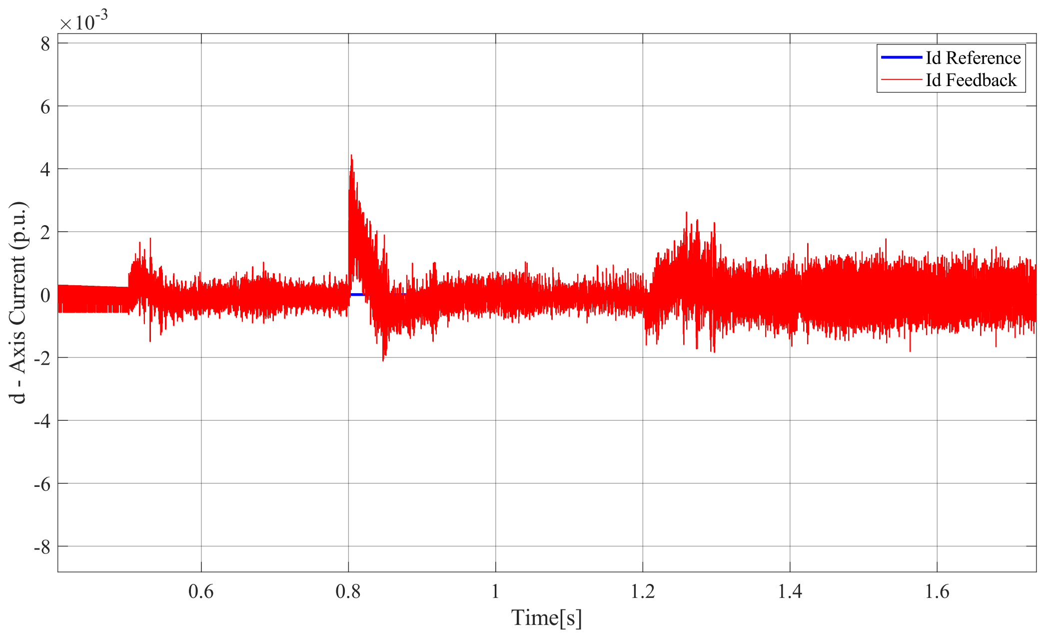

Figure 5.30 : $d-$axis controller response.

The abc phase currents according to speed reference are shown in Figure 5.31. These currents are generated after the space vector pulse width modulation technique.

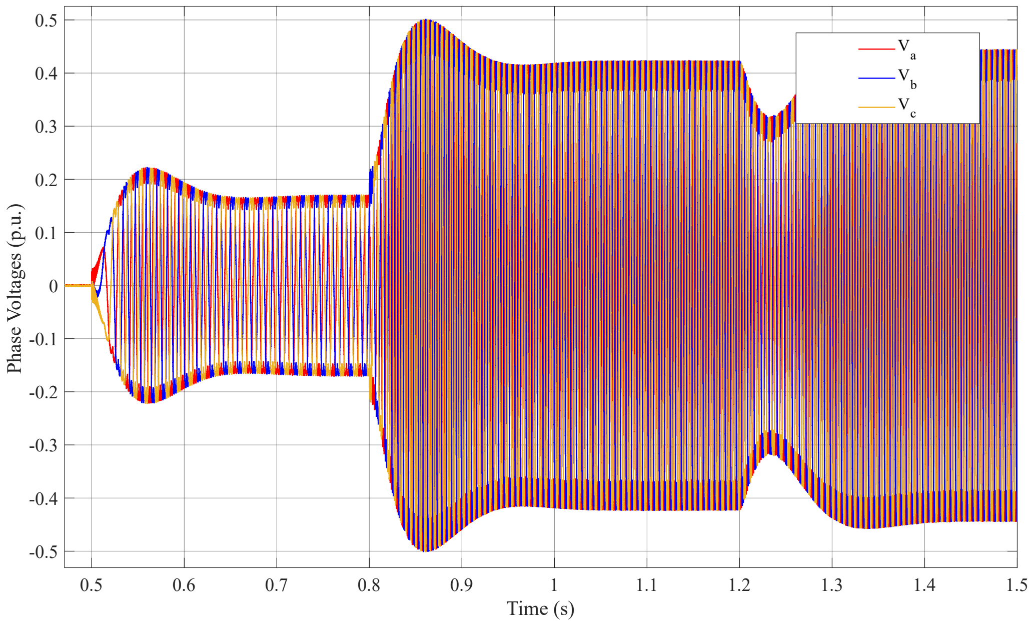

Figure 5.31 : abc phase voltages.

The α β phase currents according to speed reference are as shown in Figure 5.32. These currents are generated after inverse Park transformation.

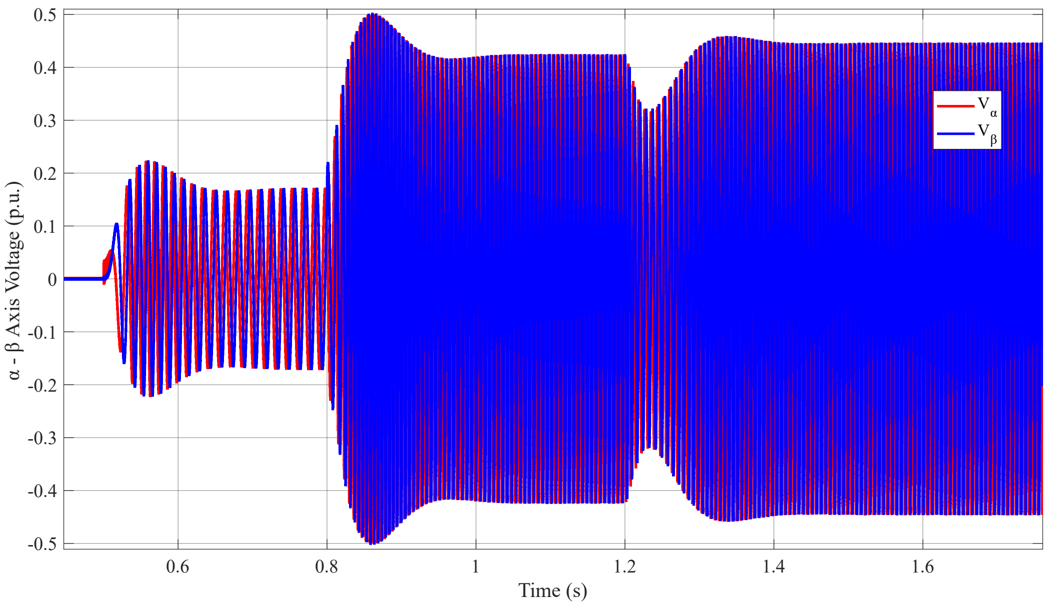

Figure 5.32 : $\alpha\beta$ phase voltage.

5.6 Real Time Implementation Subsystems

When we want to generate C code for our development kit, the plant model is not used. Instead of a plant model, microcontroller peripheral blocks are used. In Figure 5.33, ePWM blocks are used in order to send generated space vector pulse width modulation control signals to the real inverter. The most important thing is here switching frequency. In order to determine the switching frequency Equation 5.14 is used

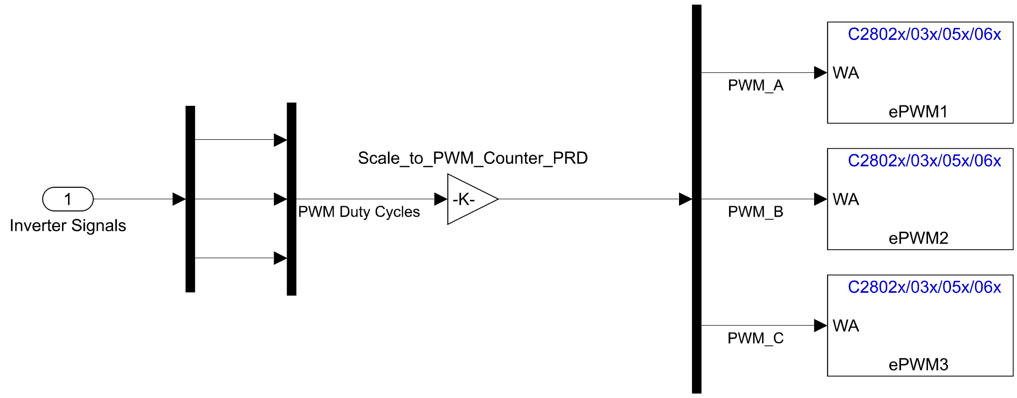

Figure 5.33 : ePWM blocks.

$$TBPRD = PWM_{CounterPeriod} = \frac{CPU_{Frequency}}{2.PWM_{Frequency}} \tag{5.14}$$

In our control system, the switching frequency is selected as 20 kHz. In order to give this information to the code generation toolbox, we need to calculate $PWM_{CounterPeriod}$ which is given in Equation 5.14. Our development board CPU frequency is 90 MHz. Therefore the TBPRD is calculated as 2250 in Equation 5.15.

$$PWM_{CounterPeriod} = \frac{90\ Mhz}{2x20\ Khz} = 2250 \tag{5.15}$$

Lastly, to get feedback signals, current and position values from the real system, ADC and eQEP blocks are used as shown in Figure 5.34.

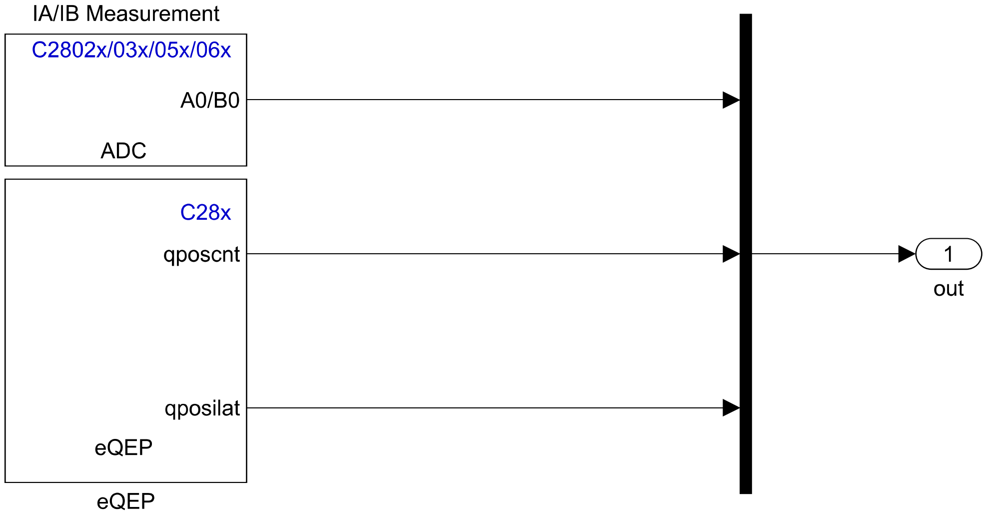

Figure 5.34 : ADC and eQEP blocks.

This subsystem is used before the current and position scaling subsystem in order to get real-time current and position information.

5.7 Stator Resistance Estimation

As we talked about in Chapter 4, the algorithm has been applied according to Figure 4.3.

The red line signal is the injected current. The blue one is the voltage value. Thanks to the current controller the voltage value is adjusted automatically. During this process the rotor is constant and in order to obtain the exact stator resistance value, the samples are taken from the steady state part from the signals. By using these samples and Equation 4.7 and Equation 4.8, stator resistance is obtained as 1.64 Ω. By using an RLC meter, the stator resistance is obtained as 1.59 Ω. When measuring the stator resistance by using an RLC meter, we should pay attention to the stator resistance winding shape. Because the result changes according to delta or star winding types.

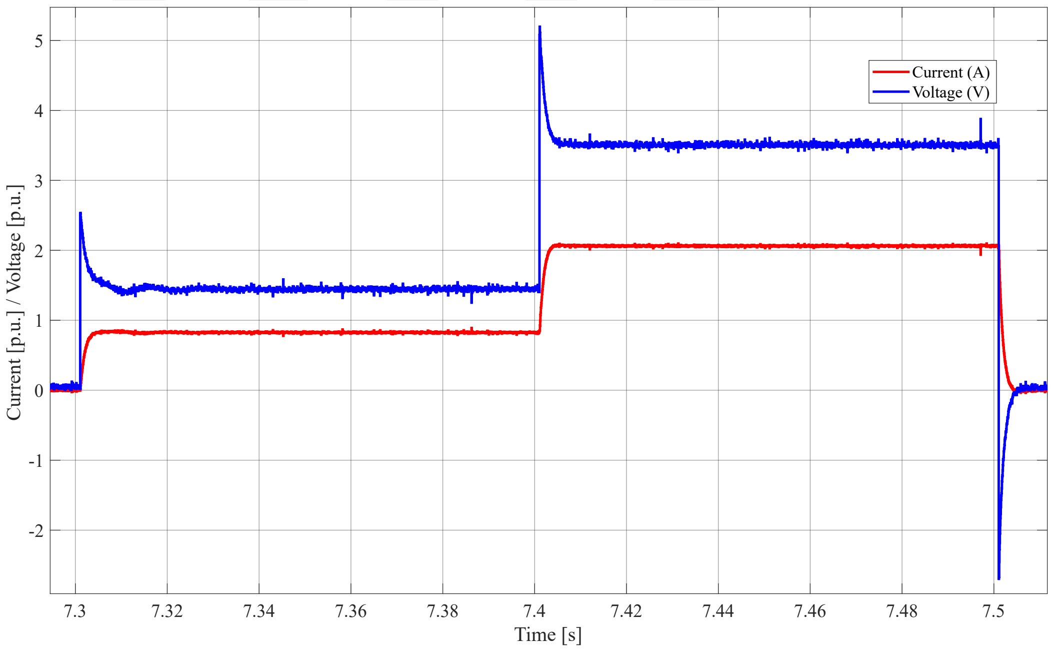

Figure 5.35 : Injected DC current.

6. CONCLUSIONS AND RECOMMENDATIONS

As a result, The stator resistance value is obtained as 1.64 Ω by using our algorithm and, 1.59 Ω by using an RLC meter. The difference is related to the cable and power semiconductor resistors. We know that when the rotor temperature increases, the stator value changes and this value affects the controller parameters. By using an offline stator resistance algorithm, the stator resistance value can be obtained when the rotor stops and it can be used in order to update the controller parameter if there is an autotuner algorithm, etc. Also, this value is critical for observer design, and when the temperature changes, all the parameters can be updated automatically. This algorithm is especially useful for washing machines and electric vehicles.

This study will be continued by researching the voltage distortion effect of the stator resistance estimation algorithm and estimating other motor parameters such as inductance, flux, etc.

REFERENCES

[1] **Ferreira, F.J.T.E. and de Almeida, A.T.** (2016). Overview on energy saving opportunities in electric motor driven systems - Part 1: System efficiency improvement, *2016 IEEE/IAS 52nd Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pp.1–8.

[2] **Goman, V.**, **Prakht, V.**, **Kazakbaev, V. and Dmitrievskii, V.** (2020). Comparative Study of Induction Motors of IE2, IE3 and IE4 Efficiency Classes in Pump Applications Taking into Account CO2 Emission Intensity, *Applied Sciences*, *10*(23), `https://www.mdpi.com/2076-3417/10/23/8536`.

[3] **Lukin, A.**, **Demidova, G.**, **Rassõlkin, A.**, **Lukichev, D.**, **Vaimann, T. and Anuchin, A.** (2022). Small Magnus Wind Turbine: Modeling Approaches, *Applied Sciences*, *12*(4), `https://www.mdpi.com/2076-3417/12/4/1884`.

[4] **Goman, V.**, **Prakht, V.**, **Kazakbaev, V. and Dmitrievskii, V.** (2021). Comparative Study of Energy Consumption and CO2 Emissions of Variable-Speed Electric Drives with Induction and Synchronous Reluctance Motors in Pump Units, *Mathematics*, *9*(21), `https://www.mdpi.com/2227-7390/9/21/2679`.

[5] **de Almeida, A.**, **Ferreira, F. and Duarte, A.** (2014). Technical and Economical Considerations on Super High-Efficiency Three-Phase Motors, *Industry Applications, IEEE Transactions on*, *50*, 1274–1285.

[6] **Iqbal, M.N.**, **Kütt, L.**, **Lehtonen, M.**, **Millar, R.J.**, **Püvi, V.**, **Rassõlkin, A. and Demidova, G.L.** (2021). Travel Activity Based Stochastic Modelling of Load and Charging State of Electric Vehicles, *Sustainability*, *13*(3), `https://www.mdpi.com/2071-1050/13/3/1550`.

[7] **Demidova, G.**, **Lukichev, D. and Kuzin, A.** (2019). A Genetic Approach for Auto-Tuning of Adaptive Fuzzy PID Control of a Telescope's Tracking System, *Procedia Computer Science*, *150*, 495–502.

[8] **Rassõlkin, A.**, **Rjabtšikov, V.**, **Kuts, V.**, **Vaimann, T.**, **Kallaste, A.**, **Asad, B. and Partyshev, A.** (2022). Interface Development for Digital Twin of an Electric Motor Based on Empirical Performance Model, *IEEE Access*, *10*.

[9] **Pellegrino, G.**, **Boazzo, B. and Jahns, T.** (2015). Plug-in, Direct Flux Vector Control of PM Synchronous Machine Drives, *IEEE Transactions on Industry Applications*, *51*, 1–1.

[10] **Pellegrino, G.**, **Bojoi, R.**, **Guglielmi, P. and Cupertino, F.** (2010). Accurate Inverter Error Compensation and Related Self-Commissioning Scheme in Sensorless Induction Motor Drives, *Industry Applications, IEEE Transactions on*, *46*, 1970 – 1978.

[11] **Dianov, A.** (2021). Instant and seamless closing of control system of IPMSM after open-loop starting, pp.1–6.

[12] **Goman, V.**, **Oshurbekov, S.**, **Kazakbaev, V.**, **Prakht, V. and Dmitrievskii, V.** (2019). Energy Efficiency Analysis of Fixed-Speed Pump Drives with Various Types of Motors, *Applied Sciences*, *9*(24), `https://www.mdpi.com/2076-3417/9/24/5295`.

[13] **Dianov, A.**, **Kim, Y.K.**, **Lee, S.**, **Lee, S. and Yoon, T.H.** (2008). Sensorless IPMSM based drive for reciprocating compressor.

[14] **Kudelina, K.**, **Vaimann, T.**, **Asad, B.**, **Rassõlkin, A.**, **Kallaste, A. and Demidova, G.** (2021). Trends and Challenges in Intelligent Condition Monitoring of Electrical Machines Using Machine Learning, *Applied Sciences*, *11*(6), `https://www.mdpi.com/2076-3417/11/6/2761`.

[15] **Rassõlkin, A.**, **Orosz, T.**, **Demidova, G.**, **Kuts, V.**, **Rjabtšikov, V.**, **Vaimann, T. and Kallaste, A.** (2021). Implementation of Digital Twins for electrical energy conversion systems in selected case studies, *Proceedings of the Estonian Academy of Sciences*, *70*, 19–39.

[16] **Wilson, S.**, **Stewart, P. and Taylor, B.** (2010). Methods of Resistance Estimation in Permanent Magnet Synchronous Motors for Real-Time Thermal Management, *Energy Conversion, IEEE Transactions on*, *25*, 698 – 707.

[17] **Badini, S.S. and Verma, V.** (2018). A New Stator Resistance Estimation Technique for Vector Controlled PMSM Drive, pp.1–6.

[18] **Antonello, R.**, **Ortombina, L.**, **Tinazzi, F. and Zigliotto, M.** (2017). Online stator resistance tracking for reluctance and interior permanent magnet synchronous motors, *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp.5861–5868.

[19] **Tang, J.**, **Yang, Y.**, **Blaabjerg, F.**, **Chen, J.**, **Diao, L. and Liu, Z.** (2018). Parameter Identification of Inverter-Fed Induction Motors: A Review, *Energies*, *11*(9), `https://www.mdpi.com/1996-1073/11/9/2194`.

[20] **Simo, R.**, **Lamnabhi-Lagarrigue, F.**, **Arzandé, A. and Vannier, J.** (2010). An Online Simplified Rotor Resistance Estimator for Induction Motors, *IEEE Transactions on Control Systems Technology*, *18*.

[21] **Ozkop, E. and Okumuş, H.** (2005). Direct Torque Controlled Induction Machine with MRAS Based Stator Resistance Estimation.

[22] **Davari, S.**, **Wang, F. and Kennel, R.** (2017). Robust Deadbeat Control of Induction Motor by Stable MRAS Speed and Stator Estimation, *IEEE Transactions on Industrial Informatics*, *PP*, 1–1.

[23] **Chen, J. and Huang, J.** (2016). Online Decoupled Stator and Rotor Resistances Adaptation for Speed Sensorless Induction Motor Drives by a Time Division Approach, *IEEE Transactions on Power Electronics*, *32*, 1–1.

[24] **Zaky, M.** (2012). Stability Analysis of Speed and Stator Resistance Estimators for Sensorless Induction Motor Drives, *Industrial Electronics, IEEE Transactions on*, *59*, 858–870.

[25] **Lee, S.H.**, **Yoo, A.**, **Lee, H.J.**, **Yoon, Y.D. and Han, B.** (2017). Identification of Induction Motor Parameters at Standstill Based on Integral Calculation, *IEEE Transactions on Industry Applications*, *PP*, 1–1.

[26] **Hasanzadeh, A.**, **Reed, D. and Hofmann, H.** (2019). Rotor Resistance Estimation for Induction Machines Using Carrier Signal Injection With Minimized Torque Ripple, *IEEE Transactions on Energy Conversion*, *34*, 942–951.

[27] **Karanayil, B.**, **Rahman, F. and Grantham, C.** (2007). Online Stator and Rotor Resistance Estimation Scheme Using Artificial Neural Networks for Vector Controlled Speed Sensorless Induction Motor Drive, *Industrial Electronics, IEEE Transactions on*, *54*, 167 – 176.

[28] **Xu, Y.**, **Parspour, N. and Vollmer, U.** (2014). Torque Ripple Minimization Using Online Estimation of the Stator Resistances With Consideration of Magnetic Saturation, *Industrial Electronics, IEEE Transactions on*, *61*, 5105–5114.

[29] **Rashed, M.**, **MacConnell, P.**, **Stronach, A. and Acarnley, P.** (2007). Sensorless Indirect-Rotor-Field-Orientation Speed Control of a Permanent-Magnet Synchronous Motor With Stator-Resistance Estimation, *Industrial Electronics, IEEE Transactions on*, *54*, 1664 – 1675.

[30] **Sangsefidi, Y.**, **Ziaeinejad, S.**, **Mehrizi-Sani, A.**, **Pairo, H. and Shoulaie, A.** (2015). Estimation of Stator Resistance in Direct Torque Control Synchronous Motor Drives, *Energy Conversion, IEEE Transactions on*, *30*, 626–634.

[31] **Sangsefidi, Y.**, **Ziaeinejad, S. and Mehrizi-Sani, A.** (2015). Sensorless Speed Control of Synchronous Motors: Analysis and Mitigation of Stator Resistance Error, *IEEE Transactions on Energy Conversion*, *31*, 1–9.

[32] **Pilla, R.**, **Mary, K.**, **Kalavathi, M. and Swathi, A.** (2015). PARAMETER ESTIMATION OF PERMANENT MAGNET SYNCHRONOUS MOTOR-A REVIEW By, *i-manager's Journal on Electrical Engineering*, *9*, 49–59.

[33] **Zhang, H.**, **Wu, W. and Wang, L.** (2017). An improved off-line identification technology for parameters of surface permanent magnet synchronous motors, pp.1–4.

[34] **Odhano, S.A.** (2014). Self-Commissioning of AC Motor Drives.

[35] **Wu, X.**, **Lin, M.**, **Wang, P.**, **Jia, L. and Fu, X.** (2019). Off-line Stator Resistance Identification for PMSM with Pulse Signal Injection avoiding the Dead-time Effect, *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, pp.1–5.

[36] **Erturk, F.** *et al.* (2019). Self-Commissioning of Sensorless AC Motor Drives, *Ph.D. Dissertation*.

[37] **Dianov, A.** (2022). An Algorithm for Offline Measurement of Motor Stator Resistance and Voltage Drop Across Inverter Switches for Washing Machine Drives, *IEEE Journal of Emerging and Selected Topics in Power Electronics*, *10*, 1–1.

[38] **Chung, D.W.**, **Kim, J.S. and Sul, S.K.** (1998). Unified voltage modulation technique for real-time three-phase power conversion, *IEEE Transactions on Industry Applications*, *34*(2), 374–380.

[39] **Rathnakumar, D.**, **LakshmanaPerumal, J. and Srinivasan, T.** (2005). A new software implementation of space vector PWM, *Proceedings. IEEE SoutheastCon, 2005.*, pp.131–136.

[40] **Nazlee, A.M.**, **Hamid, N.H.**, **Hussin, F.A. and Ali, N.B.Z.** (2010). Space Vector PWM for PMSM simulation using Matlab Simulink, *2010 IEEE Asia Pacific Conference on Circuits and Systems*, pp.1127–1130.

CURRICULUM VITAE

Name Surname : Lutfu Emre Efe

EDUCATION :

- **B.Sc.** : 2019, Istanbul Okan University, Faculty of Engineering, Department of Mechatronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- Figes Engineering
- Akim Metal
- FEMSAN Electric Motors

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **L. E. Efe, S. B. Ozturk** (2023). Offline Stator Resistance Estimation For Permanent Magnet Synchronous Motor And Real-Time Implementation Using Embedded Coder. *International Graduate Research Symposium*, May 23, Istanbul, Turkey.

LUTFU EMRE EFE

OFFLINE STATOR RESISTANCE ESTIMATION FOR PERMANENT MAGNET SYNCHRONOUS MOTOR AND REAL-TIME IMPLEMENTATION USING EMBEDDED CODER

2023