



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



HAFİF ELEKTRİKLİ ARAÇLAR İÇİN **AKILLI, FSDAM** **SÜRÜCÜ VE KONTROLÖR TASARIMI**

GÜNER TATAR

YÜKSEK LİSANS TEZİ

Elektrik ve Elektronik Mühendisliği
(Türkçe) ABD

DANIŞMAN

Doç. Dr. Hayriye KORKMAZ

EŞ-DANIŞMAN

Prof. Dr. Füsun SERTELLER

İSTANBUL, 2017



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



HAFİF ELEKTRİKLİ ARAÇLAR İÇİN **AKILLI, FSDAM** **SÜRÜCÜ VE KONTROLÖR TASARIMI**

523114006 GÜNER TATAR

YÜKSEK LİSANS TEZİ

Elektrik ve Elektronik Mühendisliği
(Türkçe) ABD

DANIŞMAN

Doç. Dr. Hayriye KORKMAZ

EŞ-DANIŞMAN

Prof. Dr. Füsun SERTELLER

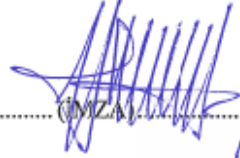
İSTANBUL, 2017

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Güner TATAR'ın “**Hafif Elektrikli Araçlar İçin Akıllı, FSDAM Sürücü ve Kontrolör Tasarımı**” başlıklı tez çalışması, 11 Ocak 2018 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Doç. Dr. Hayriye KORKMAZ (Danışman)

Marmara Üniversitesi (İMZA) 

Prof. Dr. Burhanettin CAN (Üye)

Fatih Sultan Mehmet Vakıf Üniversitesi (İMZA) 

Prof. Dr. Hasan ERDAL (Üye)

Marmara Üniversitesi (İMZA) 

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 22.01.2018 tarih ve 2018/02-02 sayılı kararı ile **Güner TATAR**'ın **Elektrik ve Elektronik Mühendisliği (Türkçe) Anabilim Dalı Yüksek Lisans** derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü
Prof. Dr. Bülent EKİCİ




ÖNSÖZ

Bu tez çalışmasında güç elektroniği ve motor sürücü devrelerinde sıklıkla kullanılan Uzak Vektör Modülasyon Teorisinin (SVPWM) Fırçasız Doğru Akım Motorlarına (FSDAM), “LabVIEW FPGA Module” araç kutusu kullanılarak uygulanması incelenmiştir.

Bu çalışmanın gerçekleştirilmesinde, değerli bilgilerini benimle paylaşan, kendisine ne zaman danışsam bana kıymetli zamanını ayırıp sabırla ve büyük bir ilgiyle bana faydalı olabilmek için elinden gelenden fazlasını sunan, her sorun yaşadığımda yanına çekinmeden gidebildiğim, güler yüzünü ve samimiyetini benden esirgemeyen ve gelecekteki mesleki hayatımda da bana verdiği değerli bilgilerden de faydalanacağımı düşündüğüm kıymetli ve danışman hoca statüsünü hakkıyla yerine getiren Doç. Dr. Hayriye KORKMAZ ve Prof. Dr. Füsun SERTELLER hocalarıma teşekkürü bir borç biliyor ve şükranlarımı sunuyorum. Yine çalışmamda konu, kaynak ve yöntem açısından bana sürekli yardımda bulunarak yol gösteren ve gelecekteki hayatında çok daha başarılı olacağına inandığım kıymetli Arş. Gör. Kenan TOKER’e de sonsuz teşekkürlerimi sunuyorum.

Teşekkürlerin az kalacağı diğer üniversite hocalarımda bana 7 yıllık üniversite (lisans ve lisansüstü) hayatım boyunca kazandırdıkları her şey için ve beni gelecekte söz sahibi yapacak bilgilerle donattıkları için hepsine teker teker teşekkürlerimi sunuyorum.

Son olarak beni bu günlere sevgi ve saygı kelimelerinin anlamlarını bilecek şekilde yetiştiren ve benden hiçbir zaman desteğini esirgemeyen bu hayattaki en büyük şansım olan Aileme, Zübeyir BİLGE ve Taner AYDIN kardeşime sonsuz teşekkür ederim.

İÇİNDEKİLER

1. GİRİŞ.....	1
1.1. Amaç ve Gerekçe.....	1
1.2. Benzer Çalışmalar.....	2
2. FSDAM'LER İÇİN TEMEL KAVRAMLAR, SÜRME VE KONTROL YÖNTEMLERİ.....	8
2.1. Giriş	8
2.2. FSDA Motorların Yapısı	8
2.3. DA Motorların Çalışma Prensipleri	9
2.4. FSDAM Matematiksel Modeli	12
2.5. Sürme ve Kontrol Yöntemleri.....	15
2.6. Uzay Vektör Darbe Genişlik Modülasyon Teorisi (SVPWM).....	23
2.7. Sonuç	35
3. FSDA MOTORLAR İÇİN SÜRÜCÜ ve KONTROLÖR TASARIMI	36
3.1. Giriş	36
3.2. Denetleyici	37
3.3. FSDA MOTOR.....	39
3.4. Anahtarlama ve Sürücü Devresi	40
3.5. ADC Bloğu	46
3.5.1. Akım Algılayıcı Çıkışların FPGA karta Uygulanması.....	47
4. SİSTEM YAZILIMI VE UYGULAMA	48
4.1. Giriş	48
4.2. LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench).....	48
4.2.1. sbRIO Mimarisi	49
4.2.2. LabVIEW FPGA Modül Proje Ekranına Genel Bakış	51
4.2.3. Katmanlar Arası Haberleşme Mimarisi	52

4.2.4.	Kullanıcı Bilgisayarı (Host PC) ve FPGA Arası Veri Transferi	53
4.2.5.	FPGA Hedeften Kullanıcı Bilgisayara Veri Transferi	54
4.2.6.	DMA FIFO Yapısı.....	54
4.2.7.	Blok Hafıza (RAM) Yapısı.....	55
4.3.	FPGA İçerisinde Yazılım Tasarımı	56
4.3.1.	Clarke Dönüşümü	59
4.3.2.	Park Dönüşümü	59
4.3.3.	Tez Çalışmasında Kullanılan Kontrolörler.....	60
4.3.4.	Ters Park Dönüşümü	67
4.3.5.	Ters Clarke Dönüşümü.....	67
4.3.6.	Uzay Vektör PWM Üretici	68
4.3.7.	Motor Akımlarının Analog Giriş Terminalinden Okunması.....	70
4.3.8.	Motor Konum Bilgisinin Okunması.....	70
4.3.9.	Motor Hızının Hesaplanması (RPM).....	72
4.3.10.	PWM Sinyallerinin Oluşturulması.....	73
4.4.	FPGA Hedef Ana Uygulaması	75
4.4.1.	PI Parametrelerinin Belirlenmesine Yönelik Test Programı	79
4.4.1.	I_q Parametresinin Birim Basamak Cevabı	81
4.4.2.	V_d / V_q Ölçekleme Faktörü.....	84
4.5.	Kullanıcı Bilgisayarında Gerçekleştirilen İşlemler.....	85
5.	ALAN YÖNLENDİRMELİ KONTROL ALGORİTMASI KULLANAN FSDA MOTORU SÜRÜCÜ SİSTEMİNİN BENZETİMİ	90
5.1.	Matlab / Simulink	90
5.2.	Motor Modeli	90
5.2.1.	Alan Yönlendirmeli Kontrol Algoritmasının Matlab Uygulaması.....	92
5.2.2.	İleri Beslemeli Kontrol Algoritması Matlab Uygulaması	95

6. DENEY SONUÇLARI.....	98
6.1. Test ve Doğrulama.....	98
6.1.1. Alan Etkili Algılayıcı Verilerinin Doğrulanması	99
6.1.2. Geri Besleme Akımı Verisinin Doğrulanması	100
6.1.3. FFC Algoritmasının Doğrulanması	102
6.1.4. FOC Algoritmasının Gerçek zamanlı LabVIEW uygulaması ile Matlab/Simulink Benzetimi Sonuçlarının Karşılaştırılması.....	103
6.1.5. PWM Anahtarlama Frekansı	111
7. SONUÇ.....	114
7.1. Giriş	114
7.2. Donanım.....	114
7.3. Yazılım.....	115
7.4. Gelecek Çalışmalar	116

ÖZET

FSDA motorlar yüksek verimlilik ve kolay kontrol stratejileri nedeniyle çok geniş bir uygulama alanına sahiptir. Bu tip motorların kontrolünde, 6 konumlu anahtar yapısı (invertör) kullanılır. Kontrol anında rotorun konumuna göre üç fazdan herhangi ikisi aktif diğeri sıfır konumunu alır. Rotor konumuna göre altı konumlu anahtar yapısı her 60° için değişmektedir. Rotorun konumu alan etkili algılayıcılar (hall effect sensor), kodlayıcılar (decoder) veya çözücüler (resolver) yardımıyla tayin edilir. Altı konumlu anahtar yapısının gerilimini ve frekansını kontrol etmek için Darbe Genişlik Modülasyonu (PWM) tekniği kullanılır. Sinüsoidal darbe genişlik modülasyonu (SPWM) ve Uzay vektör Darbe Genişlik modülasyonu (SVPWM) günümüzde yaygın olarak kullanılan yöntemlerdir. SPWM yöntemi en çok kullanılmasına karşın; anahtarlama kayıpları, yüksek harmonik içeriği ve dc-hat geriliminden tam olarak faydalanılamama gibi problemleri vardır. Diğer taraftan SVPWM tekniği yukarıda belirtilen problemleri daha aza indirmek adına oldukça başarılıdır.

Bu tez çalışması SVPWM tekniğini kullanan bir FSDA motorun kontrolü için Alan Yönelimli Kontrol Algoritmasının (FOC) LabVIEW FPGA Modülü araç kutusu kullanılarak, grafiksel geliştirme ortamında sürücü ve kontrol sistemi geliştirmeye yönelik çalışmasını anlatmaktadır. Sistemin amacı kontrolü yapılacak motorun parametrelerinin istenilen şekilde girilmesini ve görseller kullanarak davranışını incelemektir. Kontrol kısmında I_d ve I_q referans akımlarının denetimi için PI denetleyicinin yanı sıra, İleri Besleme Kontrol (Feed Forward Control) algoritması paralel olarak bağlanmıştır. PI denetleyiciler motorun davranışını iyileştirmek ve kararlı duruma getirmek için, FFC ise motor parametrelerinin girilmesi ve PI denetleyici hatalarının giderilmesi için kullanılmıştır. FSDA motorun kontrolü için FOC algoritması ile birlikte SVPWM tekniği ve bu motorun Matlab/Simulink ile benzetimi gerçekleştirilmiştir.

ABSTRACT

BLDC motors have a wide range of applications due to their higher efficiency and easy control strategies. The 6-position switch structure (inverter) is used in controlling of this kinds of motors. While controlling, any two of the three phases were set as active; the other was set to zero position depending on the motor position. At the same time, the six-position switch structure varies for every 60° depending on the rotor position. The position of the rotor is determined by using hall effect sensors, encoders or resolvers. Pulse Width Modulation (PWM) technique is used to control the voltage and frequency of the inverter. Sinusoidal Pulse Width Modulation (SPWM) and Space Vector Pulse Width Modulation (SVPWM) are techniques widely used today. Although SPWM method is used the most often, it suffers from switching losses, high harmonic content and inability to fully utilize dc-line stress. On the other hand, the SVPWM technique is quite succesful in reducing the problems mentioned above.

This study explains the development of a Field Oriented Control Algorithm (FOC) by using LabVIEW FPGA Module toolbox for a BLDC motor controller based on SVPWM technique. The aim of the system is to enter the motor parameters to be controlled and to monitor the its behavior. In the controller, beside a PI controller, a Feedforward Control algorithm operates in parallel in order to control I_d and I_q reference currents. The PI controller was used to improve and stabilize the behavior of the motor, while the FFC was used to adjust the motor parameters and for removing PI controllers' errors. In addition, a simulation for the BLDC motor by using FOC algorithm based on the SVPWM technique was developed and run in Matlab / Simulink.

SEMBOL LİSTESİ

Sembol	Anlamı	Birimi
$I_{d,q}$	Park Transformasyonu	A
$I_{\alpha,\beta}$	Clarke Transformasyonu	A
V_d/V_q	Ölçekleme Faktörü	V
B	Sürtünme Katsayısı	Nms/rad
J	Atalet Momenti	kgm ²
V	Uygulanan Gerilim	V
$e_b(t)$	Geri Elektromotor Kuvvet	V
I	Akım	A
$\Re$	Relüktans	ATur /Wb = 1/H
R	Rezistans	ohm
L	Endüktans	H = Wb/ ATur
F	Manyeto Motor Kuvvet	ATur
λ	Stator Akı Bağlantısı	WbTur
ψ_s	Toplam Stator Akı Bağlantısı	WbTur
ψ_R	Rotor-Stator Akı Bağlantısı	WbTur
Φ	Manyetik Akı	Wb
B	Manyetik Akı Yoğunluğu	Wb/m ² = T
H	Manyetik Alan	A/m
μ	Manyetik Geçirgenlik	H/m
T_e	Elektromanyetik Moment	Nm

T_L	Yük Momenti	Nm
t	Zaman	sn
$\omega(t)$	Elektriksel Aç1	rad/sn
$\theta(t)$	Elektriksel Yerdeęiřtirme	rad/sn
n_p	Stator Kutup Çifti	



KISALTMALAR

AA	: Alternatif Akım
AASM	: Alternatif Akım Senkron Makina
ACSM	: Alternative Current Sencronous Machines
ADC	: Analog to Digital Convertor
APKD	: Alanda Programlanabilir Kapı Dizileri
BLDC	: Brushless Direct Current
CMOS	: Complementary Metal Oxide Semiconductor
DA	: Doğru Akım
DAC	: Digital to Analog Convertor
DMA	: Direct Memory Acces
DSP	: Digital Signal Processor
FFC	: Feed Forward Control
FIFO	: First In First Out
FOC	: Field Oriented Control
FPGA	: Field Programmable Gate Array
FSDAM	: Fırçasız Doğru Akım Motoru
FSDA	: Fırçasız Doğru Akım
IGBT	: Insulated Gate Bipolar Transistor
LabVIEW	: Laboratory Virtual Instrumentation Engineering Workbranch
MOSFET	: Metal Oxide Semiconductor Field Effect Transistor
OEM	: Original Equipment Manufacturer
PC	: Personal Computer
PI	: Proportional Integral
RAM	: Read Acces Memory
RTOS	: Real Time Operating System

SVPWM	: Space Vector Pulse Width Modulation
sbRIO	: Single Board Reconfigurable Input Output
VHDL	: Very High Speed Integrated Circuit Hardware Description Language
VI	: Virtual Instrumentation



ŞEKİL LİSTESİ

Şekil 2.1 Üç fazlı 26 kutuplu FSDA motor yapısı [18].....	9
Şekil 2.2 Fırçalı DC motor çalışma prensibi [22]	10
Şekil 2.3 Stator sargılarının uyartılması ve kutup oluşturmaları	11
Şekil 2.4 FSDA motor elektronik komütasyon yapısı [19]	11
Şekil 2.5 FSDA motor eşdeğer devresi (a) d eşdeğer devresi, (b) q eşdeğer devresi [20,21]	12
Şekil 2.6 FSDA Motoru Kontrol Yöntemleri.....	16
Şekil 2.7 Clarke dönüşümü vektör diyagramı [25]	18
Şekil 2.8 Clarke dönüşüm modeli $I_a, I_b, I_c \rightarrow I_\alpha, I_\beta$	19
Şekil 2.9 Park Dönüşümü [25]	20
Şekil 2.10 Park dönüşüm modeli $\vec{I}_\alpha, \vec{I}_\beta \rightarrow \vec{I}_d, \vec{I}_q$	21
Şekil 2.11 FSDA motorlar için üç faz altı konumlu anahtar yapısı	24
Şekil 2.12 Altı Konumlu Anahtarlama Yapısına Göre Temel Gerilim Vektörleri	27
Şekil 2.13 Uzak vektör modülasyon geometrisi [15].....	28
Şekil 2.14 Sektör belirleme algoritması akış diyagramı [8].....	32
Şekil 2.15 Sektör 1 için anahtarlama deseni [10].....	33
Şekil 2.16 Bütün sektörler için uzak vektör modülasyonu anahtarlama desenleri [7]...	34
Şekil 2.17 Taşıyıcı filtreli Uzak vektör modülasyon çıkışı [7]	35
Şekil 2.18 Bir faz için Uzak Vektör Modülasyonu Osiloskop çıktısı	35
Şekil 3.1 FSDA motor sürücü ve kontrol devresi blok diyagramı	37
Şekil 3.2 NI sbRIO-9637 kart [26].....	38
Şekil 3.3 FSDA motoru iç yapı ve tasarım [28].....	39
Şekil 3.4 FSDA Motor sürücü devresi açık devre şeması.....	41
Şekil 3.5 IRFP064 N kanallı MOSFET'in yapısı ve teknik özellikleri.....	42
Şekil 3.6 IR2113 Mosfet Sürücü ve teknik detaylar	43
Şekil 3.7 ACS758u alan etkili algılayıcı [29].....	43
Şekil 3.8 ACS758 tipik bağlantı şeması [29]	44
Şekil 3.9 Soğutucu ve MOSFET'lerin kart üzerine yerleştirilmesi.....	44
Şekil 3.10 Sürücü devresi a) Akım sınırlama için karşılaştırmalı devre, b) Bir faz için açık devre şeması, c) Sürücü devresi üstten görünüm	46

Şekil 3.11 Bir faz için ölçülen akım değerinin FPGA kartın analog girişine uygulanması	47
Şekil 4.1 Blok diyagram ve Ön panel görseli.....	49
Şekil 4.2 sbRIO mimarsı [26]	50
Şekil 4.3 Proje ekranı genel görünümü	51
Şekil 4.4 Katmanlar arası haberleşme blok diyagramı.....	53
Şekil 4.5 DMA FIFO özellik penceresi.....	55
Şekil 4.6 Blok RAM özellik penceresi	56
Şekil 4.7 FOC kontrol yapısı [21]	57
Şekil 4.8 Sistem tasarımı	58
Şekil 4.9 Sabit Nokta Aritmetiği hesaplaması [10].....	58
Şekil 4.10 Clarke dönüşümü	59
Şekil 4.11 Park dönüşümü.....	60
Şekil 4.12 FFC algoritması LabVIEW uygulaması blok diyagram görüntüsü a) Motor parametrelerinin girilmesi, b) Eşitlik 4.6 ve 4.7 ye ait hesaplamalar	63
Şekil 4.13 İleri beslemeli kontrol algoritması için motor parametrelerinin girildiği ön panel görüntüsü	65
Şekil 4.14 Ters Park dönüşümü blok diyagramı	67
Şekil 4.15 Ters Clarke dönüşümü blok diyagramı	68
Şekil 4.16 Uzay vektör modülasyon üretici.....	69
Şekil 4.17 Fazlardan okunan analog verilerin ölçeklendirilmesine ait blok diyagram ..	70
Şekil 4.18 Theta (θ) açısının belirlenmesi	71
Şekil 4.19 Radyan cinsinde hesaplanan Rotor açısının derece cinsine çevrilmesine ait blok diyagram	72
Şekil 4.20 Hall a algılayıcı verisinin okunması.....	73
Şekil 4.21 Kullanıcı bilgisayarında FIFO verilerinin okunması ve RPM değerine dönüştürülmesi	73
Şekil 4.22 PWM Sinyalinin sıralı yapı (sequenced structure) içinde üretilmesi.....	74
Şekil 4.23 Ölü zaman süresinin belirlenmesine ait blok diyagram	75
Şekil 4.24 Anahtarlama işareti a) Ölü zamansız, b) Ölü zamanlı	75
Şekil 4.25 Alan Yönlendirmeli Kontrol Algoritması Blok diyagramı	76
Şekil 4.26 FPGA içerisinde çalışan tüm programlara ait blok diyagram	78

Şekil 4.27 FPGA Hedef ana ve alt program bileşenlerine ait ön panel görünümü	79
Şekil 4.28 PI test programı a) blok diyagram b) kullanıcı arayüzü.....	80
Şekil 4.29 I_q parametresi basamak cevabı ($f_s=20000$ Hz)	81
Şekil 4.30 Değiştirilmiş PI parametreleri ile sistemin yeni cevabı	82
Şekil 4.31 1A referans değeri için farklı K_p ve K_i değerlerinde sistemin PI cevabı	83
Şekil 4.32 I_α, I_β Maksimum genlik vektörü gösterimi	84
Şekil 4.33 Kullanıcı Bilgisayar ile FPGA hedefi arası haberleşme [24]	85
Şekil 4.34 FPGA VI üzerinde veri okuma/yazma [24]	87
Şekil 4.35 Gerçek Zaman (Real Time) kullanıcı ara yüzü blok diyagramı.....	88
Şekil 4.36 Gerçek zamanlı kullanıcı ara yüzü ön panel gösterimi	89
Şekil 5.1 FFC İle Birlikte Alan Yönlendirmeli Kontrol algoritması Matlab modeli	93
Şekil 5.2 Ters park dönüşümü Matlab uygulaması	94
Şekil 5.3 İnvertör yapısı ve FSDA motora bağlanması.....	95
Şekil 5.4 FFC algoritması Matlab uygulaması a) FFC genel blok yapısı, b) FFC algoritması iç yapısı.....	96
Şekil 6.1 1kW FSDA motor deney düzeneği	98
Şekil 6.2 Alan etkili algılayıcılara ait aç-kapa süreleri Matlab benzetim sonucu	100
Şekil 6.3 Alan etkili algılayıcılara ait aç-kapa sürelerinin osiloskop ekranı üzerinden izlenmesi.....	100
Şekil 6.4 50 rpm için geri besleme I_d ve I_q akım sinyalleri a) Gerçek sistem LabVIEW arayüzü üzerinden ölçülen değer, b) Matlab/Simulink benzetim sonucu.....	101
Şekil 6.5 Farklı akım değerleri için FFC Algoritması LabVIEW Tabanlı Gerçek Kontrolör ile Matlab/Simulink Benzetim Sonuçlarının Karşılaştırılması	107
Şekil 6.6 $K_p=2.6$ ve $K_i=0.2$ parametre değerlerine göre PI ve PI+FFC algortimasının birlikte kullanılmasına ait Matlab/Simulink benzetim sonuçları.....	108
Şekil 6.7 $K_p=5.2$ ve $K_i=0.2$ parametre değerleri için sadece PI ve PI+FFC kullanılırken Matlab/Simulink benzetim sonucu	110
Şekil 6.8 FOC Algortiması LabVIEW tabanlı gerçek uygulamaya ait deney sonuçları	111
Şekil 6.9 MOSFET anahtarlama sinyali [30]	112
Şekil 6.10 16.6 KHz (2400 ticks) için MOSFET anahtarlama sinyalleri a) Matlab/Simulink benzetim sonucu, b) Gerçek uygulama sonucu	113

TABLO LİSTESİ

Tablo 1.1 Önceki çalışmalar	7
Tablo 2.1 Altı konumlu anahtar çıkış gerilimleri ($\vec{V}_{A-0}, \vec{V}_{B-0}, \vec{V}_{C-0}$) [7].....	24
Tablo 2.2 Faz ($\vec{V}_{AN}, \vec{V}_{BN}, \vec{V}_{CN}$) ve anahtarlama ($\vec{V}_{AB}, \vec{V}_{CB}, \vec{V}_{CA}$) gerilimleri [7]	26
Tablo 2.3 α, β gerilimleri ve vektör pozisyonları [7]	27
Tablo 2.4 Belirlenen altı sektör için t_1 ve t_2 sürelerine karşılık gelen gerilim değerlerinin tanımlanması [7].....	30
Tablo 2.5 İkili sayı sistemine göre karşılık gelen sektör numaraları [7]	31
Tablo 2.6 Döngü çevrim atamaları [7].....	33
Tablo 3.1 sbRIO-9637 karta ait teknik özellikler [26].....	38
Tablo 3.2 FSDA Motora ait teknik bilgiler [28]	40
Tablo 4.1 Veri transfer yöntemlerinin karşılaştırılması.....	54
Tablo 4.2 FFC Algoritmasına ait LabVIEW Uygulama Sonuçları	64
Tablo 4.3 Durum yapısı sıralı içeriği ve uygulanması.....	70
Tablo 4.4 Alan etkili algılayıcı değerlerinin açısal ifadesi	71
Tablo 5.1 FFC Algoritması Matlab Benzetim Sonuçları	97
Tablo 6.1 Kontrol Parametreleri	99
Tablo 6.2 İleri Besleme kontrol algoritması için LabVIEW tabanlı Gerçek kontrolör ile Matlab/Simulink benzetim sonuçlarının karşılaştırılması	103
Tablo 6.3 Farklı akım değerleri için FOC algoritması Matlab/Simulink Benzetimi sonuçları	104
Tablo 6.4 FOC Algoritması Sadece PI ve PI ile FFC'nin birlikte kullanılması ile yapılan gerçek zamanlı LabVIEW denetleyici sonuçları	105

1. GİRİŞ

Endüstriyel uygulamalarda kullanılan elektrik motorlarının çoğu üç fazlı indüksiyon motorları veya fırçalı doğru akım (DC-Direct Current) motorlarıdır. Bu motor türleri düşük verim ve yüksek bakım maliyet gerektiren motorlar iken; fırçasız doğru akım (BLDC-Brushless Direct Current Machine) motorları ise yüksek verim ve düşük bakım maliyeti gerektiren motorlardır [1]. Bu motorların davranışını incelemek amacıyla bilgisayar modeli geliştirilmelidir. Aynı zamanda yüksek hızlarda güç yoğunluğunun artmasından dolayı motorun davranışını test etmek için bir test platformu da gereklidir. Bu tez çalışması, bu tip motorların davranışlarını incelemek amacıyla böyle bir test platformunun geliştirilmesini anlatmaktadır. Bunun için kullanılan platform National Instruments firmasının geliştirdiği sbRIO-9637 kartıdır. sbRIO-9637, çift çekirdek 667MHz NI Linux Real-Time çalıştıran gerçek zamanlı bir işlemci, kullanıcı tarafından yeniden yapılandırılabilen bir Zynq-7020 FPGA (Field Programmable Gate Array, Alanda Programlanabilir Kapı Dizileri) ve giriş/çıkış (G/Ç) ile tek bir baskı devre kartına entegre edilmiş tümleşik devredir. Oluşturulan platformun programlanması için National Instruments firmasının LabVIEW grafiksel programlama dili kullanılmıştır.

1.1. Amaç ve Gerekçe

Anahtarlama frekansı, güç yoğunluğu ve hızın önemli olduğu bu ve buna benzer sistemlerde FPGA tabanlı kontrol devreleri kullanılmaktadır. FPGA'ler şematik, donanım tanımlama dili olan VERILOG, VHDL (Very High-Speed Integrated Circuit Hardware Description Language- Çok Yüksek Hızlı Entegre Devre Donanım Tanımlama Dili) ve birçok uygulamanın geliştirilmesine olanak sağlayan sanal enstrümanlar (VI- Virtual Instruments) ile geniş fonksiyon araçları ve çeşitli çevresel arabirimlerle iletişim kuracak şekilde geliştirilen LabVIEW ile programlanabilmektedir. Projede LabVIEW ortamında söz konusu algoritmalara ait blok diyagramlar oluşturulduktan sonra FPGA karta aktarılması için LabVIEW FPGA modülü araç kutusundan faydalanılmıştır. Bu modül, LabVIEW ortamında hazırlanmış grafiksel kod bloklarının VHDL koda çevrilerek FPGA ortama aktarılmasını sağlayan yazılım paketidir. Bu modül;

- 1- Grafiksel Sistem Tasarımı; Geleneksel FPGA tabanlı sistemler alt seviye yazılım araçlarını ve donanım tanımlama dillerine (HDLs Hardware Description Languages) ihtiyaç duymaktadır. Donanım tanımlama dillerini öğrenmek ve etkili bir şekilde kullanmak sıkıcı ve zaman alan bir süreç olabilir. LabVIEW FPGA modülü giriş/çıkış arabirim oluşturma ve veri iletme görevini basitleştiren, tasarım verimliliğini önemli ölçüde artıran ve ürünün ticarileşmesini hızlandıran grafiksel bir programlama yaklaşımı sunar [2].
- 2- IP Kütüphanesi ve HDL Kodun Yeniden Kullanımı; FPGA'lar ile tasarım yaparken yazılım geliştirme verimliliğinin artması için verimli kodların tekrar kullanılması şarttır. LabVIEW FPGA modülü sayıcılar gibi temel fonksiyonlar veya video işleme gibi daha gelişmiş algoritmalar için NI ve Xilinx tarafından geliştirilen IP'leri sağlar. Bu durumda var olan HDL kodlar IP uyum düğümü ile alınabilir ve tekrar kullanılabilir yapıda olmaktadır [2].
- 3- Hızlı Algoritma Geliştirme; LabVIEW FPGA tamamen yerleşik benzetim yetenekleri ve hata ayıklama araçları ile donatılmış; bu sayede derlemeden önce var olan hataları görebilir ve düzeltilebilir yapı sunmaktadır. Benzetim anında temel LabVIEW hata ayıklama özellikleri ile (Highlight execution, breakpoints, probes) hatalar ayıklanabilir ve düzeltilebilir [2].

Çalışmada ilk olarak fırçasız doğru akım motorlarının teorisi ve bu motorların sürme yöntemleri, İkinci olarak FSDA motorlar için sürücü ve kontrol tasarımı, üçüncü olarak sistem yazılımı ve uygulama, dördüncü olarak FOC kullanılarak FSDA motor sürücü sisteminin benzetimi, beşinci olarak sistemin test edilmesi ve son olarak sonuçlar kısmı açıklanmıştır.

1.2. Benzer Çalışmalar

Motor sürücü ve kontrol yöntemlerini ele alan çalışmalarda ifade edilen önemli sorunlarının başında, farklı güçteki motorlar için farklı sürücü devrelerinin kullanımı gelmektedir. Aynı şekilde performans, maliyet ve verim gibi durumlar da sürücü devre tasarımında göz önünde bulundurulduğunda, tek bir sürücü devresi ile farklı güçteki aynı tip motorların sürülebilmesi önemli bir avantaj ve kolaylık sağlayacaktır. Bu sebeple bu tez çalışmasında, farklı güçteki fırçasız DA motorları için bir sürücü tasarımı konusu ele alınmıştır. Bu bölümde, güç elektroniği alanında FSDA motorlar için

geliştirilmiş sürücüler ve kontrol yöntemleri hakkında kapsamlı bir araştırma yapılarak; tez çalışmasında ele alınan içerik ile literatürde yer alan benzer çalışmalar karşılaştırılmış ve üstün tarafları veya aralarındaki farklılıklar vurgulanmıştır.

Hashimoto ve arkadaşları [3] yaptıkları çalışmada fırçasız Servo motorların özellikleri hakkında bilgiler vermiş, çalışmalarında bu motorların konum kontrolünde yüksek moment ağırlık oranına sahip olduğu, bilgisayar kontrolüne olanak verdiği, verimliliğin yüksek olduğu, gürültüsüz çalıştığı ve komütatör bakımı gerektirmediği konusunda bilgiler vermiştir. Bu çalışmada, bu tezde de kullanılan algoritmaya benzer bir vektör kontrol algoritması kullanılmıştır. Vektör kontrolü için 8 durum arasından 6 aktif durumun d ve q uzayında yerleşimini ve komütasyon yapısının nasıl gerçekleştiğinden bahsetmişlerdir. Hashimoto ve arkadaşlarının çalışmasında sadece tek motor için hız kontrolü gerçekleştirilmiştir. Ayrıca kullanıcı tarafından parametre girişine uygun herhangi bir arayüz de tasarlanmamışlardır.

O Al-Ayasrah, T. Alukaidey, ve G. Pessinadis'in [4] çalışmalarında, endüstriyel uygulamalar için n sayıda özdeş fırçasız DA motorunun hız kontrolünü ele almışlardır. Özdeş 2 motorun hız kontrol sisteminde harici bir FPGA tamponlama tasarımı, karma işaretli ADSP-21992'nin PWM üretim birimi ile arabirim oluşturarak gerçekleştirilmiştir. Bu çalışmada DC hat geri besleme akımını ölçmek için, üç fazlı motorun iki fazından iki adet akım sensörü ile ölçüm yapmak yerine; her motor için sadece tek bir faz üzerinden bir tane akım sensörünün kullanımı tercih edilmiştir. Böylelikle altı konumlu anahtar tekniğinde, her bir motor için yüksek maliyetli dekodek kullanmak yerine, üç konumlu alan etkili algılayıcı kullanılarak maliyet düşürülmüştür. Her bir motor hızı, takometre yerine alan etkili algılayıcı sinyalleri temel alınarak belirlenip, hesaplanmıştır.

Ehsan Drayabeigi ve Gholamreza Arab Markadeh'in [5] çalışmalarında fırçasız DA motorunun (FSDAM) hız kontrolü için bir duygusal denetleyici sunmuşlardır. Önerilen bu denetleyici, duygusal öğrenme tabanlı akıllı bir denetleyicidir. Yani denetleyicinin kullanılması beyindeki duygu işleme mekanizmasına dayanmaktadır. Bu akıllı kontrolör, Matlab benzetim ortamında trapezoidal geri-emk'lı fırçasız DA sürücüsü kullanılarak başarıyla uygulanmıştır. Bu çalışma, içinde FSDAM tahrik sisteminin yenilikçi ve basit bir şekilde uygulanmasına olanak sağlamaktadır. Bu duygusal zeki

kontrol sistemi yüksek otomatik öğrenme özelliğine sahip bir yapıdır. Benzetim sonuçları sabit durum ve hızlı geçiş hız yanıtlarının, 20 ile 300 [dev/dk] arasında geniş bir aralıkta elde edilebileceğini göstermektedir. Kullanılan kontrol yöntemi zeki bir sistem olmasına rağmen, aynı cins farklı güçteki motorların sürülebilmesi yeteneği bu çalışmada tartışılmamıştır.

I. Topaloglu ve arkadaşlarının [1] yaptıkları çalışmada, altı konumlu anahtar ile beslenen fırçasız DA motorların kapalı çevrim hız kontrolü ele alınmış; masa üstü bir CNC makinesinde sürülen motorun cinsi veya gücü değiştirilmeden sadece atalet değişimlerinin (yük değişimi) sürme işlemi üzerindeki etkileri incelemişlerdir. Bu tez çalışmasında kullanılan kontrol yöntemi ile bahsedilen makalede kullanılan kontrol yöntemi (kapalı çevrim hız kontrolü) benzerdir.

M. A. Bevilaqua, Ademir Nied ve José de Oliveira'nın [6] çalışmalarında bu tez çalışmasında da tercih edildiği gibi sabit mıknatıslı senkron motorların hız kontrolü için Alan Yönlendirmeli Kontrol (FOC) algoritması kullanılmıştır. Bu çalışmada kullanılan motor, sinüzoidal geri emk'ya sahip ve doğrudan tahrik tipi (DD_Direct Drive) fırçasız AA senkron (BLAC) motordur ve bu tezde kullanılan motor ile de benzerlik göstermektedir. Diğer taraftan kontrolör tasarımı da -bu tezde de olduğu gibi- NI firmasına ait Compact RIO üzerinde LabVIEW FPGA Modül araçları kullanılarak geliştirilmiştir. Bu sistem FPGA mantık sentezi için yüksek düzeyde/yüksek verimlilik sağlayan bir araçtan oluşmaktadır. Sözkonusu çalışma ile bu tezde önerilen sistemin arasındaki farklardan biri, kontrolörün geliştirme platformudur. Bu tezde sbRIO-9637 geliştirme kartı kullanılmıştır. Diğerisi ise, farklı tip motorların kullanılmasıdır. Makale, FOC algoritması için tasarım ilkelerini ve daha sonra LabVIEW-FPGA platformda tasarlanmış denetleyicilerin uygulamasını açıklamaktadır. Ayrıca bu çalışmada FOC algoritması LabVIEW'e ait hazır sanal enstrümanlardan yararlanılarak oluşturulmuşken; bu tezde tüm sanal enstrümanlar tarafımızdan geliştirilmiştir.

Anders Kronberg [7]'e ait yüksek lisans tezinde, uzay vektör modülasyonun uygulandığı çıkıntılı kutuplu (Salient pole) motorların invertör tasarımı, Compact RIO da gerçekleştirilmiş ve bu tip motorlar için bir düzenek oluşturulmuştur. Bu çalışmada, yeni motor tasarımını volan temelli tüm elektrikli aktarma organlarına dahil etmek için kullanabilecek bir invertör tipi tasarlanmıştır. Çalışmanın amaçlarından biri daha

önceden yapmış oldukları “Design and Simulation of Field Oriented Control and Direct Torque Control for a Permanent Magnet Synchronous Motor with Positive Saliency” başlıklı tezdeki tork dalgalanmalarını azaltmak için daha karmaşık bir kontrol yöntemi uygulamışlar ve sonuçları karşılaştırmışlardır. Bizim tez çalışmasında da bu konu ele alınmıştır.

Geoffrey De Smaele ve Matthias Verschelden [8] sabit mıknastıslı elektrik motorların kontrolü isimli çalışmalarında FOC algoritmasını NI firmasına ait cRIO kullanarak gerçekleştirmişlerdir. Ayrıca söz konusu algoritmayı daha somut hale getirip görselliği artırmak adına bir kullanıcı arayüzü de tasarlamışlardır. Yapılan bu çalışmada, bu tezde de olduğu gibi, kullanıcının istediği motor parametrelerini sisteme girip sonuçlarını izleyebilmesi mümkündür. Bu çalışmada tasarlanan düzenek ve kullanıcı arayüzü aynı cins farklı güçteki motorların sürülmesine izin vermesine rağmen bu durum ele alınmamıştır.

Benzer olarak Magnus Fredrixon [9] Fırçasız DA Motorları için LabVIEW-FPGA tabanlı “Evaluation of a System for Rapid Control Prototyping in the Field of Robot Control” başlıklı bir tez yapmıştır. Yaptığı çalışmanın amacı ise, yeni donanım ve yazılımları değerlendirmek ve başlatmaktır. Değerlendirme kısmında; kullanılan donanımdaki zaman tüketimi ve kaynak kullanımı ile, yazılımın kullanılabilirliği üzerinde durmaktadır. Donanım olarak NI firmasına ait sbRIO 9631 kartı kullanmıştır. Bizim yaptığımız tezdekine benzer olarak FOC algoritmasını kullanmış, ama İleri Besleme Kontrol (FFC-Feed Forward Control) algoritmasını kullanmamıştır. Sürücü katında ise piyasada hazır olarak bulunan “Single Drive Circuit” kartını kullanmıştır. Çalışmasında tek bir motor için tork ve hız kontrolü yapmıştır.

Bengi Tolunay [10] “Space Vector Pulse Width Modulation for Three Level Converters- a LabVIEW Implementation” adlı yüksek lisans tezinde; üç seviyeli dönüştürücülere bir dalga güç dönüşümü bakış açısı ile yaklaşan ve arzulanan üç seviyeli dalga biçimini elde etmek için bir boşluk vektörünü referans olarak alan darbe genişlik modülasyon sisteminin hesaplanmasını ve uygulanmasını kapsayan bir sistem geliştirmiştir. Belirtilen sistem yüksek güç ve gerilim gerektiren üç fazlı sistemler için özel olarak uyarlanmış ve bu nedenle her türlü yenilenebilir enerji kaynağına uygun prototip bir yapı sunmuştur. Bu çalışmada sürücü katında çok seviyeli evirici ve kaskat

H köprüsü ile sürücü katı beslemesi için üç seviyeli nötr noktası sıkıştırılmış gerilim kaynağı kullanılmıştır. Yük olarak ise orta gerilim yüksek güç üç fazlı bir “dalga enerji dönüştürücü” sistemi beslenmiştir. Söz konusu çalışmanın kontrol bölümü, bu tezde önerilen sistem ile benzerlik göstermesine rağmen; yük ve besleme katları tamamen farklıdır.

Vom Fachbereich [11] “Rapid Prototyping System for Control of Inverters and Electrical Drives” başlıklı doktora tezinde deneysel amaçlı olarak üç fazlı servo ve sabit mıknatıslı senkron makineler için FPGA ortamında sensörsüz konum kontrolü gerçekleştirmiş ve sonuçlarını karşılaştırmıştır. Bu tezde de önerilen sisteme benzer olarak kontrol bölümünde, uzay vektör modülasyon (SVPWM) ile birlikte FOC algoritmasını kullanmıştır. Yük katı, algoritmanın geliştirildiği donanım ile yazılım geliştirme dili bakımından bu tez ile farklılık göstermektedir. Yazılım dili olarak metin tabanlı bir donanım tanımlama dili olan VHDL tercih edilmiştir.

Bu tezde önerilen sistem ile yukarıda sıralan çalışmalar arasında ayrıntılı bir karşılaştırma yapılmıştır. Ancak literatürde yer alan tüm çalışmalar bunlarla sınırlı değildir. Benzerlik oranı nispeten daha düşük olan çalışmalara ise Tablo 1’de yer verilmiştir. Burada benzer çalışmalar, kullanılan kontrol yöntemi, donanım yapısı, yazılım ve geliştirme platformu bakımından değerlendirilmiş ve tabloda sıralanmıştır.

Tablo 1’de sıralanan motor kontrol uygulamalarında yaygın olarak iki yazılım geliştirme dilinin kullanıldığı görülmektedir: Birincisi NI LabVIEW, diğeri ise Matlab/Simulink. Bu tezde önerilen sistemde de benzer olarak LabVIEW grafiksel geliştirme platformu kullanılmış; yazılım LabVIEW-FPGA modül aracılığı ile derlenerek FPGA üzerine gömülecek bit dosyası oluşturulmuştur. Bu tezde klasik kontrol algoritmalarından farklı olarak FOC ve SVPWM birlikte kullanılarak elektrik makineleri geçici rejim aşamasındaki aşırı yüklenme sonucu oluşan harmoniklerin giderilmesi veya azaltılmasına yönelik akıllı sürücü ve kontrolör içeren bir test düzeneği geliştirilmiştir. Gerçekleştirilen proje aynı zamanda Matlab/Simulink paketi ile benzetimi yapılmış ve desteklenmiştir.

Tablo 1.1 Önceki çalışmalar

Kaynak Numarası	Başlık	Kullanılan Yöntem	Yazılım Geliştirme Dili	Donanım
[12]	A new method for tuning PI gains for position control of BLDC motor based wing morphing actuators	Internal Model Control Methodology	C Matlab/Simulink	DSP Personal Computer
[13]	Brushless DC Motor Speed Control System Simulink Simulation	PI Controller for Speed and Current	Matlab/Simulink	Personal Computer
[14]	Design and Modelling of BLDC Motor for Automotive Application	Finite Element and Numerical Methods	Finite Element Analysis software and Matlab/Simulink	Personal Computer
[15]	Modelling and Implementation of Intellegent Commutation System for BLDC Motor Underwater Robotics App.	Fuzzy Logic Control	Matlab/Simulink Microchip	Personal Computer dsPIC33FJ
[16]	Trajectory based Predictive Current Control for Permanent Magnet Brushless DC Motor	Stationary Plane Based Vector Control	Matlab/Simulink	Personal Computer

2. FSDAM'LER İÇİN TEMEL KAVRAMLAR, SÜRME VE KONTROL YÖNTEMLERİ

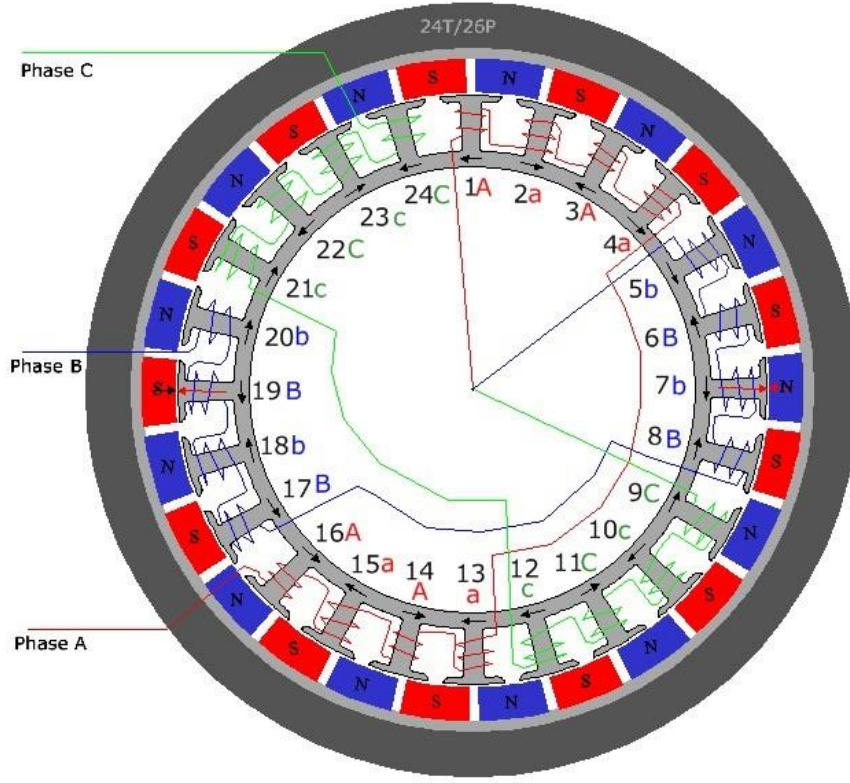
2.1. Giriş

Bu bölümde FSDA motorların yapısı, çalışması, matematiksel modeli ile sürme ve kontrol yöntemlerinden bahsedilmiştir.

2.2. FSDA Motorların Yapısı

FSDA motorlar içyapı olarak Alternatif Akım Senkron Motorlara (Alternative Current Synchronous Machines- ACSM) ve karakteristikleri açısından ise klasik fırça ve kolektör düzeneğine sahip DC motorlara benzerler [15,17]. FSDA motorlar komütasyon işlemini mekanik olarak değil; elektronik olarak sağlayan bir motor türüdür. Fırça içeren DC motorlarda, rotordaki sargılara elektrik iletimi fırça ve kolektör yapısı ile sağlanır. Dilimli yapıda olan fırça ve kolektör düzeneği sayesinde rotor sargılarından geçen akımın yönü motor hareket ederken kendiliğinden değişir. Bu sistemin kıvılcım oluşturma, bakım gerektirme ve fırçalarda aşınma gibi bazı problemleri bulunmaktadır. FSDA motorlarda ise fırça kolektör düzeneğinin görevini elektronik bir denetleyici ve algılayıcılar üstlenir [4,15,17]. Bir FSDA motor kısaca üç ana bölümden oluşmaktadır: bunlar sabit mıknatlara sahip rotor, sargıların bulunduğu stator ve rotor konumunu belirleyen algılayıcılardır.

Rotorda bulunan sabit mıknatıslar neodmiyum, samaryum, nikel, kobalt, demir gibi elementlerin farklı oranda karışımıyla oluşmaktadır. Bu mıknatıslar, rotorlarda kutup sayısını belirlemek ve manyetik alan oluşturmak için kullanılırlar. Motorlardaki kutup sayısı motorun hassasiyetini belirlemektedir. Oluşturulan kutupların sayısı arttıkça motordaki adım aralığının açısal hızı düşük olacağı için motor hızı ve konum takibi daha kolay ve kararlı olacaktır. FSDA motorlarda rotor, statorun içinde, dışında veya statorla karşı karşıya olacak şekilde üretilmektedir. Şekil 2.1'de statoru 24 oluktan, rotoru 26 kutuptan oluşan 3 fazlı bir FSDA motor görülmektedir.



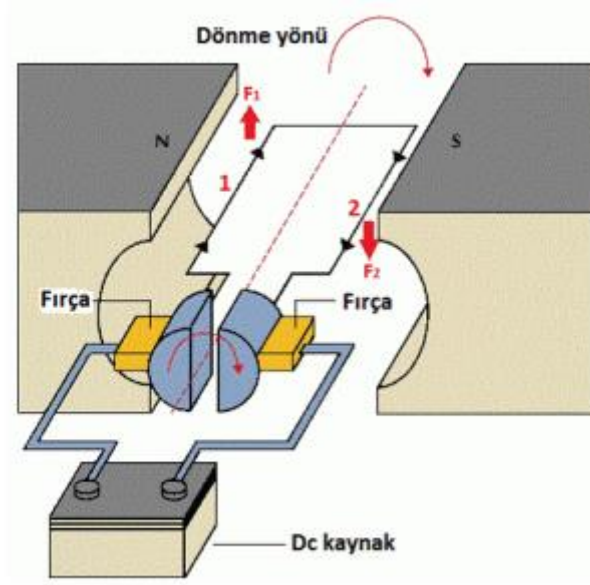
Şekil 2.1 Üç fazlı 26 kutuplu FSDA motor yapısı [18]

FSDA motorlardaki stator kısmı, kullanım yerine ve amacına göre değişik fazlarda olmaktadır. Hassas çalışılması istenilen yerlerdeki motorlar genelde üç fazlıdır. Maliyet ve hassasiyet göz önüne alındığında motorlardaki kutup sayısı da farklılık göstermektedir. Çok kutuplu FSDA motorlarda bir faza ait kutup sargıları seri bağlanmakta ve stator üzerine sırayla yerleştirilmektedir. Bu tez çalışmasında kullanılan motor üç fazlı 24 kutuplu stator ve 18 oluklu rotordan oluşmaktadır. Motorun konumunu ve dönüş yönünü belirlemek için alan etkili algılayıcılardan (Hall-Effect Sensor) faydalanılır.

2.3. DA Motorların Çalışma Prensipleri

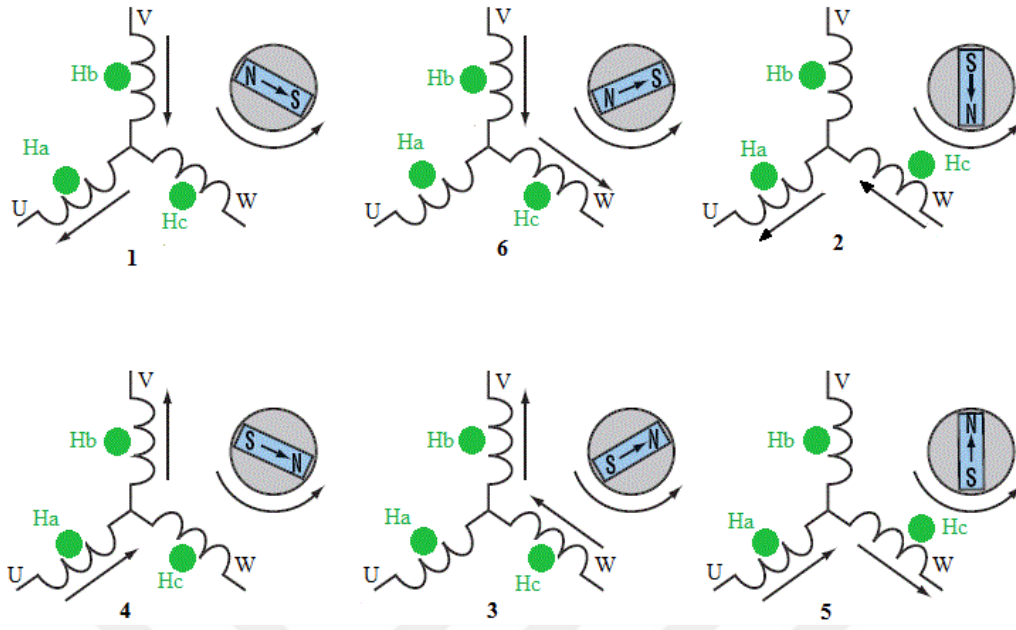
Manyetik alan içerisinde akım taşıyan telde meydana gelen kuvvet mekanik hareketin temel kaynağıdır. Bu hareket sağ el kuralı dediğimiz Akım ($\vec{i}$), Manyetik alan ($\vec{B}$) ve Kuvvet ($\vec{F}$) elemanlarının birbirine dik olduğu sistem ile açıklanır. Şekil 2.2 de görüldüğü gibi manyetik alan çizgileri kuzey kutbundan (N) güney kutbuna (S) sabit ve sürekli. Fakat 1 ve 2 numaralı kollarda akımların yönünün birbirine göre ters olması

meydana getirecekleri kuvvetlerin de ters yönde olmalarına neden olur (F_1 ve F_2). Bu ters yönlü kuvvetler devir daim sağlayarak sürekli bir dönme meydana getirirler. Bu fırçalı DC motorların çalışma prensibidir ve dönme hareketi mekanik bir komütatör ve fırçalar yardımıyla olur [17,22].

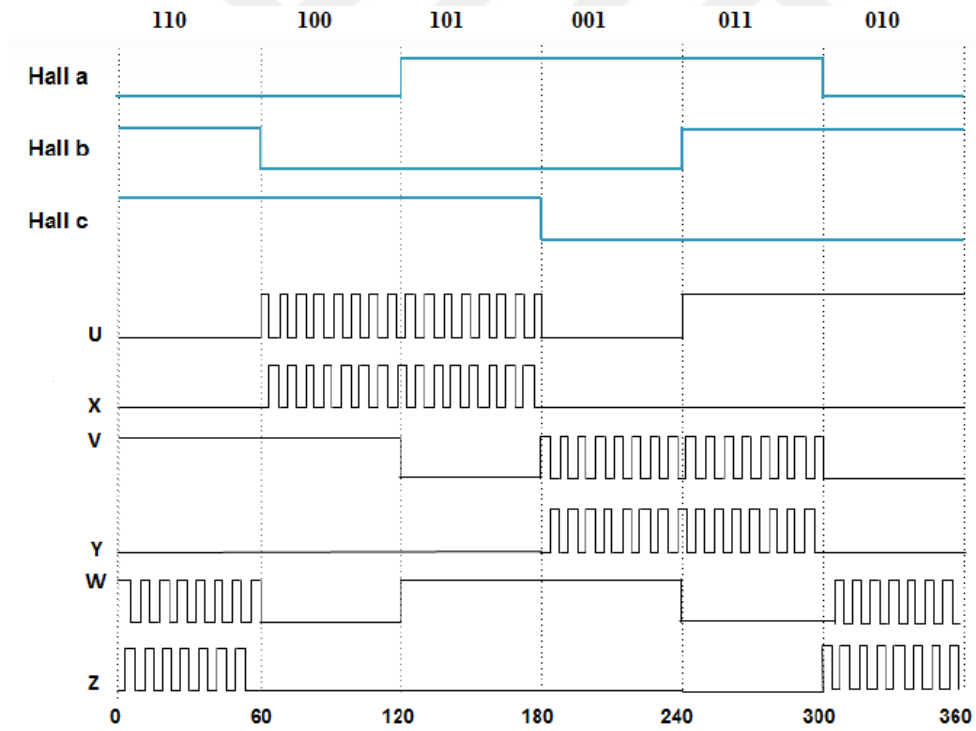


Şekil 2.2 Fırçalı DC motor çalışma prensibi [22]

FSDA motorların çalışma prensipleri fırçalı DC motorlar ile benzerdir. Fakat bu motorlarda dönme çoklu geri bildirim algılayıcıları (alan etkili algılayıcılar, optik kodlayıcılar, çözücüler- resolver) kullanılarak gerçekleştirilir [4,15]. Bu tip motorlarda en çok kullanılan sensörler alan etkili algılayıcılar ve optik kodlayıcılardır. Hall sensörler, bir akım taşıyan iletken manyetik alana maruz kaldığında yük taşıyıcılarının iletkenin iki yüzeyi boyunca oluşan gerilime dayanan bir kuvvetle karşılaştığı alan etkili algılayıcılar prensibine göre çalışır [11,17]. Sargılarda oluşan manyetik alanın yönü ters çevrilirse indüklenen gerilimde ters dönecektir [11-14,17]. FSDA motorlarda kullanılan alan etkili algılayıcılar için, rotor manyetik kutupları N veya S belirli algılayıcının yanından geçtiğinde konum belirlemek için kullanılacak yüksek veya düşük seviyeli lojik bir sinyal üretir. Geri besleme algılayıcıları kullanılarak tespit edilen motor konumuna dayanan bir komütasyon sistemi Şekil 2.3 ve Şekil 2.4'te gösterilmiştir.



Şekil 2.3 Stator sargılarının uyartılması ve kutup oluşturma



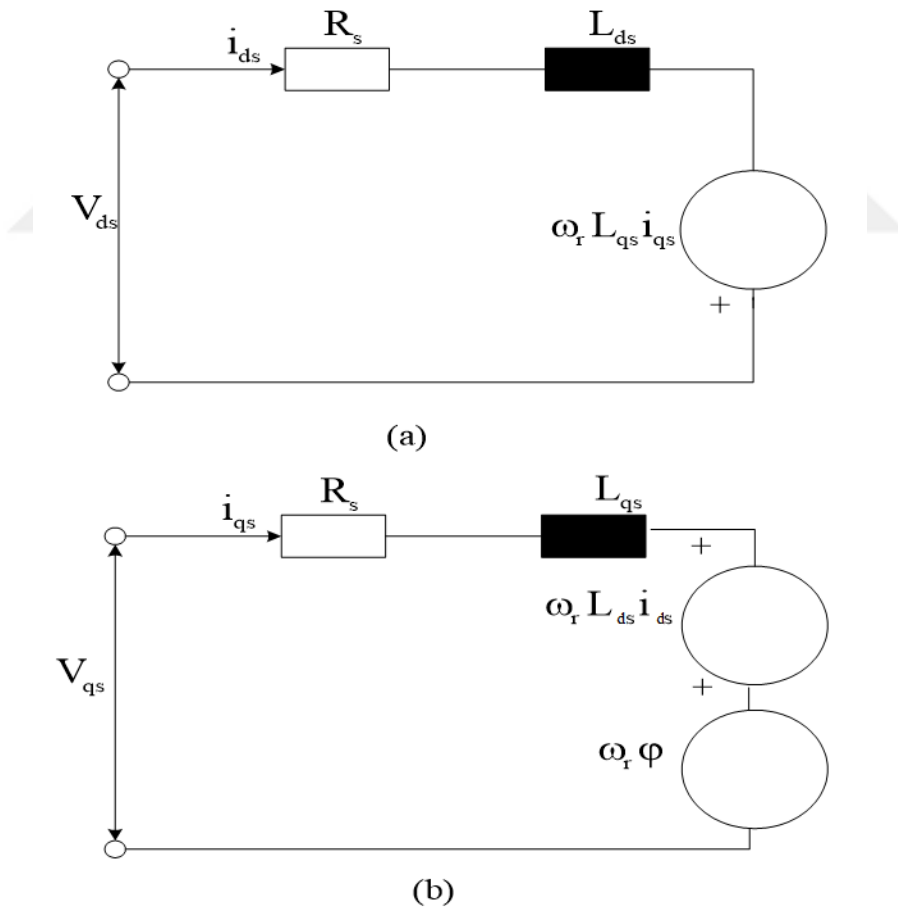
Şekil 2.4 FSDA motor elektronik komütasyon yapısı [19]

Geri besleme algılayıcıları kullanılarak motor konumu tespit etme prensibine dayanan bir komütasyon sisteminde Şekil 2.3'te gösterildiği gibi üç stator sargısının ikisi

uyartılmaktadır. Şekil 2.4'te lojik “001” olarak uygulandığında Şekil 2.3 1. durum için V faz stator sargısı kuzey kutbu olarak uyartılırken, W faz stator sargısı güney kutbu olarak uyartılmaktadır. Bu uyarılma nedeniyle rotorun güney kutbu V sargısı ile hizalanır, kuzey kutbu W sargısı ile hizalanır. Motorun her hareketi bu 6 durum için sargı ve kutup başları değişerek motora dönme hareketini oluşturur.

2.4. FSDAM Matematiksel Modeli

Bu bölümde FSDA motora ait matematiksel model, $\alpha\beta \leftrightarrow dq$ çevirimi ve Kirchoff denklemleri üzerinde durulmuştur. Bu tez çalışmasında kontrol yöntemi olarak SVPWM kullanılmıştır. Bu teknik, 2.6 bölümde Uzay Vektör Darbe Genişlik Modülasyon Teorisi başlığı altında anlatılmıştır. Şekil 2.5'te FSDA motora ait dq eşdeğer devreleri gösterilmiştir.



Şekil 2.5 FSDA motor eşdeğer devresi (a) d eşdeğer devresi, (b) q eşdeğer devresi [20,21]

FSDA motorların vektör denetimi için sabit düzlemdeki stator değişkenlerinin dönen rotor düzlemine aktarılması gerekmektedir. [6,17,20] Bu sayede eşitliklerin dereceleri indirgenerek çözümlerinin daha hızlı olması mümkün olmaktadır [17,20]. Ancak ele alınacak sistemin referans düzleminde hızının bilinmesi gerekmektedir. Referans düzlemlerinin incelenmesi Park ve Clark dönüşümleri ile mümkün olmaktadır. eşdeğer devrede akım ve gerilim eşitliklerinde üç fazdan iki faza Park dönüşümü kullanılarak dönüştürülmektedir. Bu durumda elde edilen FSDA motorun referans düzleminde stator gerilim eşitlikleri Kirchhoff yasasından Eşitlik 2.1 ve Eşitlik 2.2’de ki gibi elde edilmektedir [20,21].

$$V_{ds} = R_s I_{ds} - \omega_r L_{qs} I_{qs} + L_{ds} \frac{d(I_{qs} + \varphi)}{dt} \quad (2.1)$$

$$V_{qs} = R_s I_{qs} + \omega_r L_{ds} I_{ds} + L_{qs} \frac{dI_{qs}}{dt} + \omega_r \varphi \quad (2.2)$$

$\frac{d\varphi}{dt} = 0$ Olduğundan dolayı Eşitlik.2.1 ve Eşitlik.2.2 yeniden düzenlenirse Eşitlik.2.3 ve 2.4 elde edilir [20,21].

$$V_{ds} = R_s I_{ds} - \omega_r L_{qs} I_{qs} + L_{ds} \frac{dI_{qs}}{dt} \quad (2.3)$$

$$V_{qs} = R_s I_{qs} + \omega_r L_{ds} I_{ds} + L_{qs} \frac{dI_{qs}}{dt} + \omega_r \varphi \quad (2.4)$$

Eşitlik 2.3 ve 2.4 yeniden düzenlenerek durum uzay formunda yazılırsa Eşitlik 2.5 elde edilir [20,21].

$$\frac{d}{dt} \begin{pmatrix} I_{ds} \\ I_{qs} \end{pmatrix} = \begin{pmatrix} \frac{1}{L_{ds}} & 0 \\ 0 & \frac{1}{L_{qs}} \end{pmatrix} \left\{ \begin{bmatrix} V_{ds} \\ V_{qs} \end{bmatrix} - \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix} + \omega_r \begin{bmatrix} 0 & L_{qs} \\ L_{ds} & 0 \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \end{bmatrix} - \frac{np\omega_r}{2} \begin{bmatrix} 0 \\ \varphi \end{bmatrix} \right\} \quad (2.5)$$

Bu durumda motorun ürettiği Elektromanyetik Moment ise Eşitlik 2.6 teki gibi olur [20,21].

$$T_e = 0,75np \left(\phi I_{qs} + (L_{ds} - L_{qs}) I_{qs} I_{ds} \right) \quad (2.6)$$

FSDA motorların rotorlarında oluşan manyetik akı sabit mıknatıs tarafından oluşturulduğundan ve rotor direnci çok yüksek olduğundan akımın mıknatıslanma bileşeni $I_{ds} = 0$ olur [19,20,21]. Ayrıca mıknatısların rotor yüzeyine yerleştirildiği motorlarda salınım olmadığından dolayı d ve q eksenli endüktansları birbirine eşittir. Buna göre Eşitlik.2.6 yeniden düzenlenirse motorun ürettiği elektromanyetik moment Eşitlik 2.7’de ki gibi elde edilir [20,21].

$$T_e = 0,75np\phi I_{qs} \quad (2.7)$$

Motorun elektrik açısal hızı ve konumu Eşitlik 2.8 ve Eşitlik 2.9 kullanılarak bulunur [20,21].

$$\frac{d\omega_r}{dt} = \frac{(T_e - B\omega_r - T_L)}{J} \quad (2.8)$$

$$\frac{d\theta_r}{dt} = \omega_r \quad (2.9)$$

Son olarak motorların rotor referans düzlem modeli, durum uzay formunda ise Eşitlik 2.10’da ki gibi ifade edilir [20,21].

$$\frac{d}{dt} \begin{bmatrix} I_{qs} \\ I_{ds} \\ \omega_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_{qs}} & -\omega_r \frac{L_{ds}}{L_{qs}} & -\frac{\phi}{L_{qs}} & 0 \\ \omega_r \frac{L_{ds}}{L_{qs}} & \frac{R_s}{L_{ds}} & 0 & 0 \\ \frac{3np^2\phi}{8J} & 0 & -\frac{B}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} I_q \\ I_d \\ \omega_r \\ \theta_r \end{bmatrix} + \begin{bmatrix} \frac{1}{L_q} & 0 & 0 \\ 0 & \frac{1}{L_d} & 0 \\ 0 & 0 & -\frac{np}{2J} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_q \\ V_d \\ T_L \end{bmatrix} \quad (2.10)$$

Burada;

R_s : Stator hat direnci (Ω)

L_{sq}, L_{sd} : Stator endüktansları (H)

i_{sq}, i_{sd} : $d-q$ eksenli stator akımları (A)

ω_r : Rotorun mekanik hızı (rad/sn)

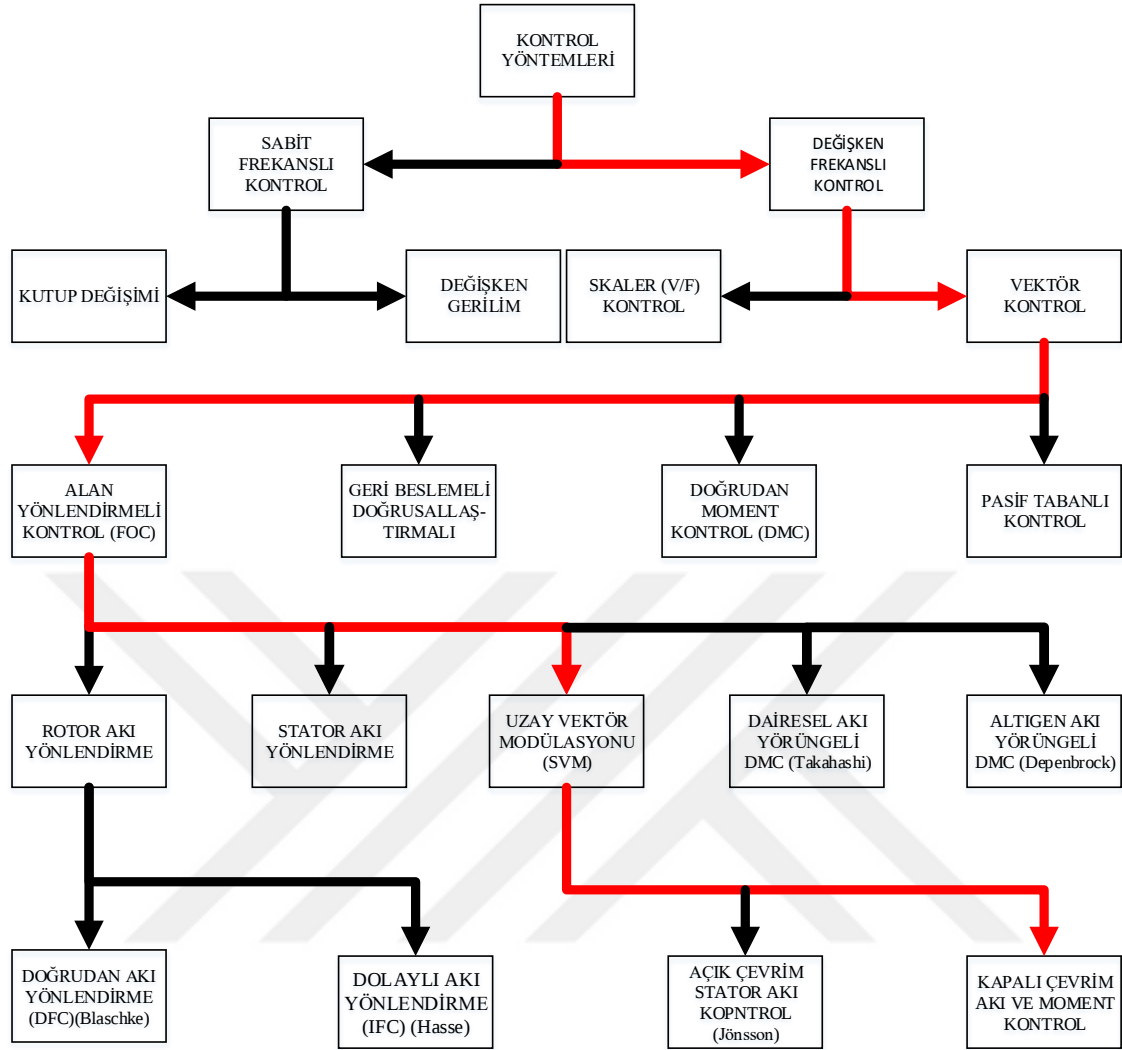
Φ : Motorun manyetik akısı (Wb)

np : Kutup çifti sayısı

T_e : Elektromanyetik moment (Nm)

2.5. Sürme ve Kontrol Yöntemleri

FSDA motorlarda kontrolü, elektronik bir denetleyici üstlenir. Denetleyicide yüksek akım anahtarlama görevini yürüten yarı iletken devre elemanları ve anahtarlama ile ilgili zamanlamayı sağlayan bir mikro denetleyici bulunur. Motorun dönüşünde aksama olmaması için denetleyicinin uygun bir hızda rotoru takip etmesi gerekmektedir. Bunun için rotor pozisyonunun kesin ve hassas olarak tayin edilmesi gerekir. Bu süreçte kullanılan birçok kontrol yöntemi mevcuttur. Bu yöntemler genel olarak motordaki konum tespit mantığına göre belirlenmektedir ve uygun bir yazılımla bu konum tespitleri mümkün olmaktadır. Kullanılan bu yazılımlar direk motor parametrelerini etkilediğinden hassasiyet, maliyet, verim ve hız gibi kriterler göz önünde bulundurulmalıdır. Şekil 2.6'te kontrol yöntemlerine ait sınıflandırma gösterilmiştir. Kırmızı oklar bu tezde izlenen yöntemi ifade etmektedir.



Şekil 2.6 FSDA Motoru Kontrol Yöntemleri

Şekilde gösterilen mevcut kontrol yöntemlerinden FOC son zamanlarda yukarıda belirtilen kriterleri, diğer algoritmalarla göre daha verimli bir şekilde sağladığından yaygın olarak kullanılmaktadır [5,8-10,22]. FOC yöntemi uzay vektörlerinin konumlandırılması temeline dayanmaktadır. Bu ve buna benzer vektör kontrol algoritmalarının arkasındaki temel mantık, senkron veya asenkron motorlardaki manyetik akı ve momenti oluşturan akım bileşenlerinin ayrı ayrı kontrolüne olanak sağlamasıdır [22]. Tork ve akı üretiminin bağımsız kontrolünü belirtildiği gibi yapabilmek için stator akımı gereklidir. Stator akım vektörü önce statora sabitlenmiş statik referans sistemi $\alpha\beta$ ya; daha sonra ise dönel dinamik bir referans sistemi olan dq ya yansıtılır. Belirtilen kavramların anlaşılabilmesi için rotorun elektriksel ve

mekanik konumu arasındaki farkın bilinmesi gereklidir. Mekanik konum rotor şaftının gerçek pozisyonu, elektriksel konum ise rotorun manyetik alanının dönüşüyle ilgilidir ve kutup çifti sayısına bağlıdır. Her iki terim arasındaki bağıntı Eşitlik. 2.11 ile ifade edilebilir.

$$\theta_e = \theta_m np \quad (2.11)$$

Burada np kutup çiftini belirtmektedir.

Dönme hızı rotor konumunun türevi olduğu için elektriksel ve mekanik dönme hızı arasındaki bağıntı Eşitlik 2.12 gibi olmalıdır.

$$\omega_e = \omega_m np \quad (2.12)$$

Hız ve rotor pozisyonu alan etkili algılayıcılardan gelen geri besleme bilgisi ile hesaplanır. Akım değerleri ise algılayıcılar yardımıyla hesaplanmaktadır. Simetrik akımlar $i_a + i_b + i_c = 0$ olduğu için herhangi iki stator faz akımlarının ölçülmesi yeterli olacaktır.

İlk adımda üç boyutlu koordinat sisteminde i_a, i_b, i_c fazlarının herhangi ikisinden geçen akımlar ölçülür (örnek olarak U ve V fazları) ve iki boyutlu α, β düzlemine yansıtılır. Bu işlem Şekil 2.8 ve Şekil 2.9 da gösterilen Clarke dönüşümü ile aşağıdaki eşitlikler kullanılarak elde edilir. Şekil 2.9'daki dalga formları Mathematica analiz programı ile elde edilmiştir.

$$\vec{I}_s(t) = \vec{I}_a(t) + \vec{I}_b(t)e^{i\frac{2\pi}{3}} + \vec{I}_c(t)e^{i\frac{4\pi}{3}} \quad (2.13)$$

$$\vec{I}_\alpha = \vec{I}_a - \vec{I}_b \sin(30) - \vec{I}_c \sin(30) \quad (2.14)$$

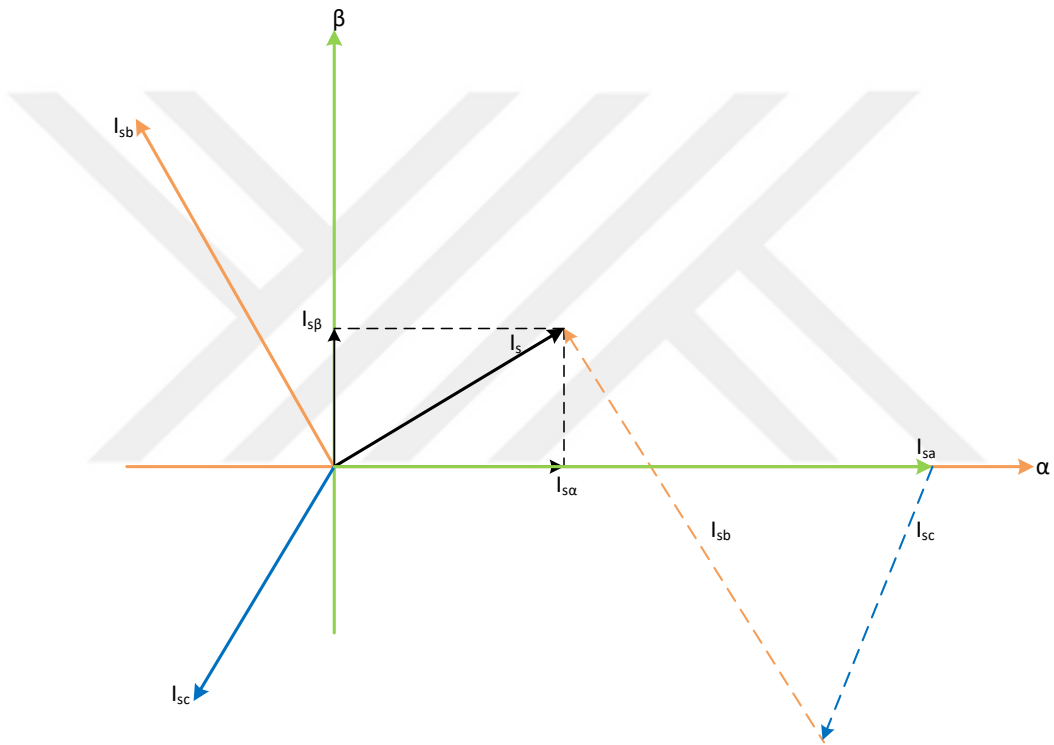
$$I_\alpha = I_a - \frac{I_b + I_c}{2} \quad (2.15)$$

$$\vec{I}_\beta = 0 + \vec{I}_b \cos(30) - \vec{I}_c \cos(30) \quad (2.16)$$

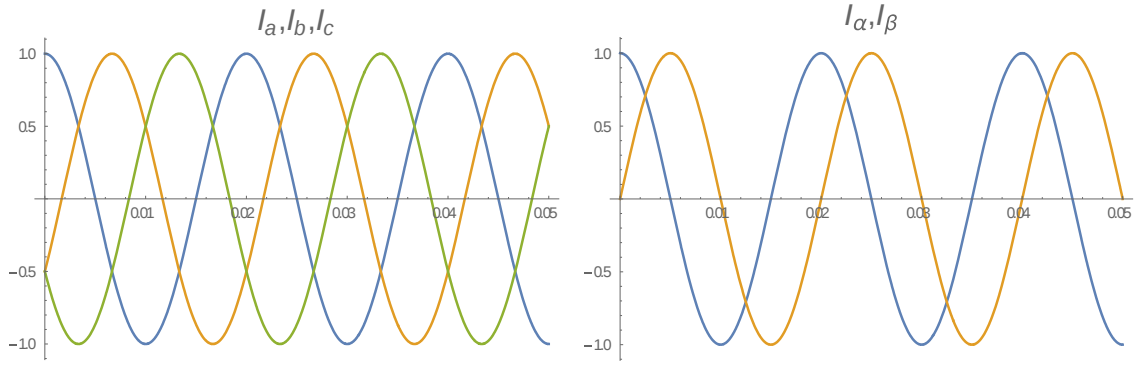
$$\begin{bmatrix} \vec{I}_\alpha \\ \vec{I}_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \vec{I}_a \\ \vec{I}_b \\ \vec{I}_c \end{bmatrix} \quad (2.17)$$

$$\vec{I}_\alpha = \vec{I}_a \quad (2.18)$$

$$\vec{I}_\beta = \left(\frac{\vec{I}_a + 2\vec{I}_b}{\sqrt{3}} \right) \quad (2.19)$$



Şekil 2.7 Clarke dönüşümü vektör diyagramı [25]



Şekil 2.8 Clarke dönüşüm modeli $I_a, I_b, I_c \rightarrow I_\alpha, I_\beta$

Geçiş fazörlerini α ve β eksenine üzerine yansıtan Clarke dönüşümünün bu şekilde uygulanması orijinal faz akımının $\frac{3}{2}$ katı olan toplam I_α, I_β akım vektör genliği olarak elde edilir. Ayrıca bunun dışında iki farklı uygulamada mümkündür: Birincisi toplam akım vektörünün genliğinin Clarke formüllerinde $\frac{2}{3}$ ile çarpılması; diğeri ise Clarke dönüşümünde gücün aynen korunmasıdır. Bu tezde, elde edilen Clarke formüllerinin $\sqrt{\frac{2}{3}}$ ile çarpılması gerekmektedir.

Ayrıca FOC algoritmasında Ters Clarke ve Ters Park dönüşümlerine ihtiyaç duyulmaktadır. Ters Clarke dönüşümü için Clarke dönüşümünün tersinin alınması gerekmektedir. Bu işlem Eşitlik 2.20’de verilen matrisin tersi alınarak elde edilir. Sonuç, Eşitlik 2.21, 2.22 ve 2.23’te verilmiştir.

$$\begin{bmatrix} \vec{I}_a \\ \vec{I}_b \\ \vec{I}_c \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \vec{I}_\alpha \\ \vec{I}_\beta \end{bmatrix} \quad (2.20)$$

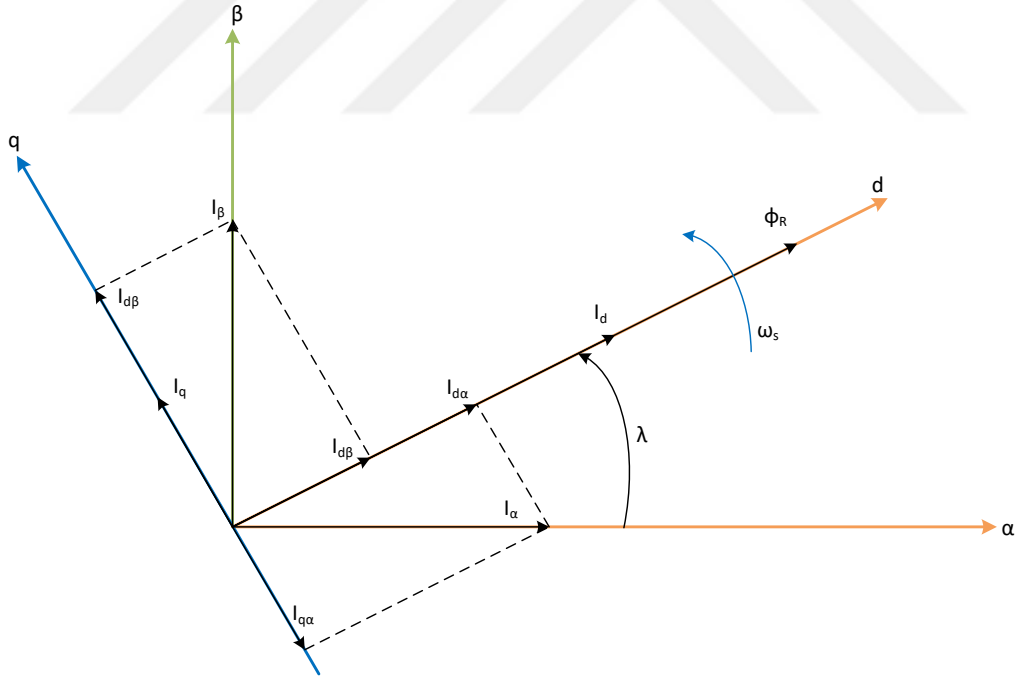
Bu durumda Eşitlik 2.21’deki gibi olmaktadır.

$$\vec{I}_a = \vec{I}_\alpha \quad (2.21)$$

$$\vec{I}_b = \frac{-\vec{I}_\alpha + \sqrt{3}\vec{I}_\beta}{2} \quad (2.22)$$

$$\vec{I}_c = -\vec{I}_a - \vec{I}_b = \frac{-\vec{I}_\alpha - \sqrt{3}\vec{I}_\beta}{2} \quad (2.23)$$

İkinci adımda ise akım vektörleri denklem 2.14 ve 2.15 te açıklanan ve Şekil 2.10 ve şekil 2.11 de gösterilen Park dönüşümü kullanılır. Park dönüşümünde, akımlar durağan α, β referans düzleminden dönel d, q referans sistemine dönüştürür. Bu dönüşümün yapılması için rotorun elektriksel konumunun (θ) bilinmesi gerekir. Bu işlemden sonra I_{sd} ve I_{sq} akımları rotora sabitlenir ve DA olarak görülür. Akımlar (I_{sd} ve I_{sq}) hesaplandığından referans akımları I_{dref} (moment referansı) ve I_{qref} (tork referansı) ile karşılaştırılabilir ve bu durumda tork ve akı değerleri ayrı ayrı kontrol edilmiş olur.



Şekil 2.9 Park Dönüşümü [25]

$\vec{I}_\alpha$ Akım değerleri d, q ekseninde Eşitlik 2.24 ve Eşitlik 2.25 deki gibi ifade edilir.

$$d_{axis} : \vec{I}_{d\alpha} = \vec{I}_\alpha \cos \lambda \quad (2.24)$$

$$q_{axis} : \vec{I}_{q\alpha} = -\vec{I}_\alpha \sin \lambda \quad (2.25)$$

$\vec{I}_\beta$ Akım değerleri ise d, q ekseninde Eşitlik 2.26 ve Eşitlik 2.27 teki gibi ifade edilir.

$$d_{axis} : \vec{I}_{d\beta} = \vec{I}_\beta \sin \lambda \quad (2.26)$$

$$q_{axis} : \vec{I}_{q\beta} = \vec{I}_\beta \cos \lambda \quad (2.27)$$

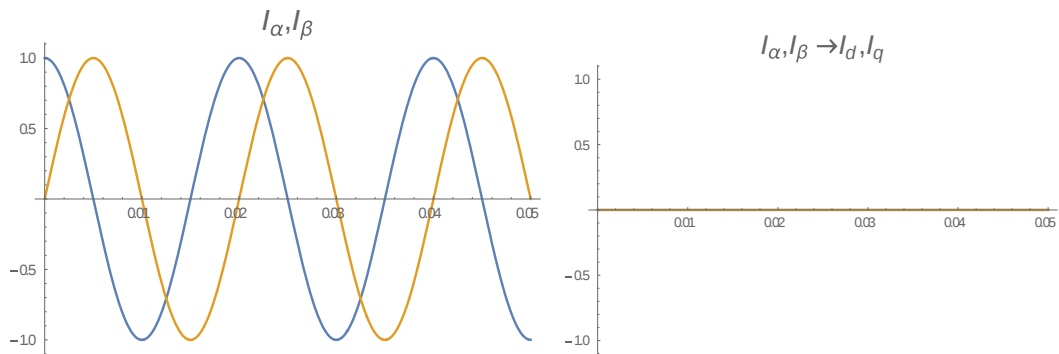
Bileşenlerin d ve q eksenini boyunca kombinasyonu Park dönüşümü için denklemlerle ifade edilir. Bu denklemler matrislerle gösterilebilir.

$$\begin{bmatrix} \vec{I}_d \\ \vec{I}_q \end{bmatrix} = \begin{bmatrix} \cos \lambda & \sin \lambda \\ -\sin \lambda & \cos \lambda \end{bmatrix} \begin{bmatrix} \vec{I}_\alpha \\ \vec{I}_\beta \end{bmatrix} \quad (2.28)$$

Bu durumda $\vec{I}_d$ ve $\vec{I}_q$ referans denklemleri aşağıdaki gibi olur.

$$\vec{I}_d = \vec{I}_\alpha \cos \lambda + \vec{I}_\beta \sin \lambda \quad (2.29)$$

$$\vec{I}_q = -\vec{I}_\alpha \sin \lambda + \vec{I}_\beta \cos \lambda \quad (2.30)$$



Şekil 2.10 Park dönüşüm modeli $\vec{I}_\alpha, \vec{I}_\beta \rightarrow \vec{I}_d, \vec{I}_q$

I_α ve I_β AA'ların ortalama değeri sıfır olduğundan bu akımların dc ye dönüştürülmüş değeri olan I_d ve I_q akımları, şekil 2.11 deki grafikte sıfır olarak görülür.

Aynı şekilde Ters Park dönüşümü için Park dönüşümünün tersinin alınması yeterlidir.

$$\begin{bmatrix} \vec{I}_\alpha \\ \vec{I}_\beta \end{bmatrix} = \begin{bmatrix} \cos \lambda & -\sin \lambda \\ \sin \lambda & \cos \lambda \end{bmatrix} \begin{bmatrix} \vec{I}_d \\ \vec{I}_q \end{bmatrix} \quad (2.31)$$

Bu durumda eşitlik aşağıdaki gibi olmaktadır.

$$\vec{I}_\alpha = \vec{I}_d \cos \theta - \vec{I}_q \sin \theta \quad (2.32)$$

$$\vec{I}_\beta = \vec{I}_d \sin \theta + \vec{I}_q \cos \theta \quad (2.33)$$

Park dönüşümü için rotorun elektriksel konumu θ_r , bu bölümde λ olarak alınmıştır.

Kullanılan kontrol yönteminin yanı sıra, darbe genişlik modülasyon (PWM) yöntemi de sürücü performansını dolayısıyla da motorun performansını etkilemektedir. PWM'in amacı, istenilen çıkış dalga şeklini üretmenin yanında üretilen dalga şeklinin en az harmonik içermesini de sağlamaktır. Oluşan bu harmonikler motorun demir ve bakır kayıplarının artmasına sebep olduğu gibi; bunun yanında motorun ısınmasına da yol açmaktadır. Ayrıca elektrik motor sürücülerinde güç kayıpları, elektromanyetik girişim ve darbe momentlerine neden olmaktadır. Bunun sonucunda da motorun verimi düşmektedir.

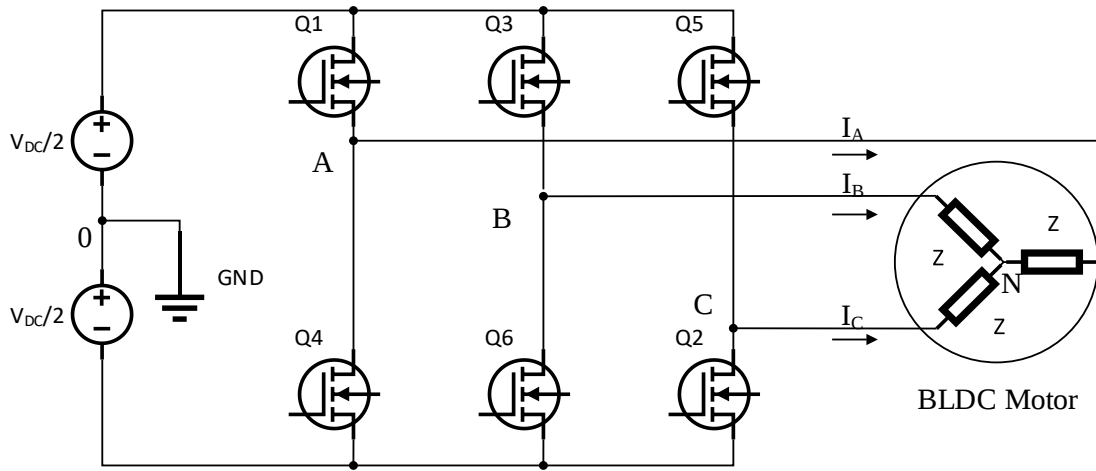
Klasik PWM'e göre SVPWM yöntemi ile sürülen anahtarlama devrelerinde, giriş geriliminin şeklinden dolayı daha az harmonik içeriğine sahip bir çıkış gerilimi elde edilmektedir. Anahtarlama devreleri, girişteki DA gerilimi çıkışında istenilen frekans ve gerilim değerinde AA gerilimine dönüştüren güç elektroniği düzenekleridir. Altı konumlu anahtar çıkışındaki akım veya gerilim eğrilerinin sinüzoidal olması arzu edilen bir durum olmasına rağmen; çoğunlukla çıkış sinüs sinyali bozan harmoniklerin oluşması kaçınılmazdır. Yüksek hıza sahip yarı iletken devre elemanları ve anahtarlama teknikleri kullanılarak söz konusu çıkış geriliminde oluşan harmoniklerin en aza indirgenmesi sağlanır. Güç çeviricilerinde kullanılan yarı iletken elemanların ve hızlı

işlem yapabilen sayısal sinyal işlemcilerinin gelişmesi karışık dönüşüm ve kontrol algoritmalarının uygulanabilmesine olanak sağlamıştır. Altı konumlu anahtar girişine yukarıda bahsedilen teknikler kullanılarak oluşturulmuş uygun bir PWM işareti uygulanarak, motor sargılarına uygulanan gerilimin genliği, fazı ve frekansı değiştirilebilir. PWM algoritmalarının oluşturduğu kontrol işareti yüksek frekanslı taşıyıcı işaretlerle karşılaştırılarak, eviricide yarı iletken anahtarlama elemanlarının “aç-kapa” sürelerini ayarlar. Böylelikle temel frekansı istenilen değerde olan bir gerilim şekli üretilmiş olur [5,22].

2.6. Uzay Vektör Darbe Genişlik Modülasyon Teorisi (SVPWM)

Statora, doğru faz geriliminin uygulanması için altı konumlu anahtarların uygun bir darbe genişlik modülasyonu sinyaline ihtiyacı vardır. Bu PWM sinyalleri Uzay Vektör Modülasyonu tarafından elde edilir. SVPWM, iki referans vektörünü ($V_{\alpha ref}$, $V_{\beta ref}$) darbe genişlik modülasyonunu kullanarak altı konumlu anahtarın sürülmesi için uygun ikili anahtar desenlerine dönüştürür [5,8-10,22]. Bu anahtarlama desenleri, istenilen akı vektörünü oluşturacak stator akımlarını elde etmek için stator fazlarında gerilim indüklenmesine yol açar.

Uzay vektör darbe genişlik modülasyonun daha iyi anlaşılması için 3 fazlı altı konumlu anahtar yapısının bilinmesi gereklidir. Şekil 2.11 den de görüldüğü gibi altı konumlu anahtarın her bir fazı (kolu) iki adet alt ve üst anahtar gruplarından oluşur. Üst anahtar grubu ($Q1, Q2, Q5$) pozitif DA hattına bağlanırken; alt anahtar grubu ($Q2, Q4, Q6$) negatif DA hattına bağlanır. Uzay vektör PWM modülasyon tekniğini kullanan bu yapı, “180°-type” modunu kullanır. Yani aynı anda her kolun üst veya alt anahtar gruplarından sadece biri aktif olacaktır. Üç fazlı altı konumlu anahtar devrelerinde, her bir kolda 2 ve toplamda 6 anahtar bulunur ve bu anahtarlar $2^3 = 8$ olası kombinasyona göre motoru sürer.



Şekil 2.11 FSDA motorlar için üç faz altı konumlu anahtar yapısı

Anahtarlama kombinasyonu ve bunlara karşılık gelen evirici çıkış gerilimleri Tablo 2.1 de verilmiştir.

Tablo 2.1 Altı konumlu anahtar çıkış gerilimleri ($\vec{V}_{A-0}, \vec{V}_{B-0}, \vec{V}_{C-0}$) [7]

A	B	C	V_{A-0}	V_{B-0}	V_{C-0}
0	0	0	$-\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$
0	0	1	$-\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$
0	1	0	$-\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$
0	1	1	$-\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$
1	0	0	$+\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$
1	0	1	$+\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$
1	1	0	$+\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$
1	1	1	$+\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$	$+\frac{V_{DC}}{2}$

Şekilde verilen anahtarlama çıkış gerilimlerinden, kaynak geriliminin sıfır noktası ile yıldız bağlı yükün sıfır noktası arasındaki gerilimler hesaplanabilir.

$$V_{ON} = V_{A-0} + ZI_A \quad (2.34)$$

$$V_{ON} = V_{B-0} + ZI_B \quad (2.35)$$

$$V_{ON} = V_{C-0} + ZI_C \quad (2.36)$$

Verilen üç denklem toplanır;

$$3V_{ON} = V_{A-0} + V_{B-0} + V_{C-0} + Z(I_A + I_B + I_C) \quad (2.37)$$

Denklemleri ayrıca aşağıdaki gibi yazabiliriz:

$$V_{AN} = V_{ON} - V_{A-0} = \frac{V_{A-0} + V_{B-0} + V_{C-0}}{3} - V_{A-0} \quad (2.38)$$

Eşitlik 2.38 düzenlenirse

$$V_{AN} = \frac{1}{3}(2V_{A-0} - V_{B-0} - V_{C-0}) \quad (2.39)$$

V_{BN} ve V_{CN} değerleri de benzer şekilde elde edilebilir. Bunun sonucunda 3 faz için oluşacak gerilim değerleri, Eşitlik 2.40, 2.41 ve 2.42 kullanılarak hesaplanır.

$$V_{AN} = \frac{1}{3}(2V_{A-0} - V_{B-0} - V_{C-0}) \quad (2.40)$$

$$V_{BN} = \frac{1}{3}(2V_{B-0} - V_{A-0} - V_{C-0}) \quad (2.41)$$

$$V_{CN} = \frac{1}{3}(2V_{C-0} - V_{A-0} - V_{B-0}) \quad (2.42)$$

Eşitlik 2.40, 2.41 ve 2.42 deki denklemler Tablo 2.2'deki gibi invertör çıkış geriliminde kullanılırsa, her anahtarlama sırasına karşılık gelen gerilim değerini verir.

Tablo 2.2 Faz ($\vec{V}_{AN}, \vec{V}_{BN}, \vec{V}_{CN}$) ve anahtarlama ($\vec{V}_{AB}, \vec{V}_{CB}, \vec{V}_{CA}$) gerilimleri [7]

A	B	C	V_{AN}	V_{BN}	V_{CN}	V_{AB}	V_{BC}	V_{CA}
0	0	0	0	0	0	0	0	0
0	0	1	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$+\frac{2V_{DC}}{3}$	0	$-\frac{V_{DC}}{2}$	$\frac{V_{DC}}{2}$
0	1	0	$-\frac{V_{DC}}{3}$	$+\frac{2V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{2}$	$\frac{V_{DC}}{2}$	0
0	1	1	$-\frac{2V_{DC}}{3}$	$+\frac{V_{DC}}{3}$	$+\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{2}$	0	$\frac{V_{DC}}{2}$
1	0	0	$+\frac{2V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$\frac{V_{DC}}{2}$	0	$-\frac{V_{DC}}{2}$
1	0	1	$+\frac{V_{DC}}{3}$	$-\frac{2V_{DC}}{3}$	$+\frac{V_{DC}}{3}$	$\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$	0
1	1	0	$+\frac{V_{DC}}{3}$	$+\frac{V_{DC}}{3}$	$-\frac{2V_{DC}}{3}$	0	$\frac{V_{DC}}{2}$	$-\frac{V_{DC}}{2}$
1	1	1	0	0	0	0	0	0

Eşitlik 2.40, 2.41, 2.42 ve Tablo 2.2 de verilen her bir anahtarlama durumu için U_{xxx} gerilim vektörü ve Clarke dönüşümünden elde edilen U_α ve U_β iki faz gerilim değerlerini hesaplamak mümkündür. Dengeli bir sistem için Eşitlik 2.43 ve 2.44'teki gibi sadeleştirilebilir. Gerilim değerleri Tablo 2.3'te verilmiştir. Burada $\vec{U}_{000}$ ve $\vec{U}_{111}$ sıfır vektörleri olarak alınır ve bu durumda çıkışta gerilim oluşmaz.

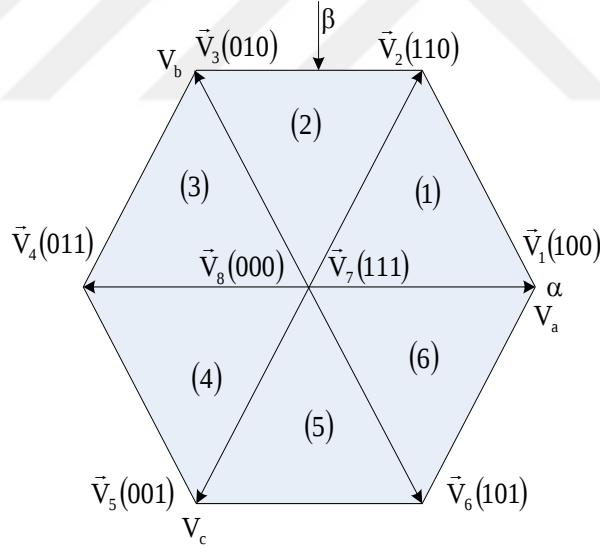
$$V_\alpha = V_{AN} \quad (2.43)$$

$$V_\beta = \frac{V_{AN} + 2V_{BN}}{\sqrt{3}} \quad (2.44)$$

Tablo 2.3 ve Şekil 2.12 üç fazlı altı konumlu anahtarlar için dönüştürülmüş α, β gerilimleri ile olası anahtar pozisyonlarını göstermektedir.

Tablo 2.3 α, β gerilimleri ve vektör pozisyonları [7]

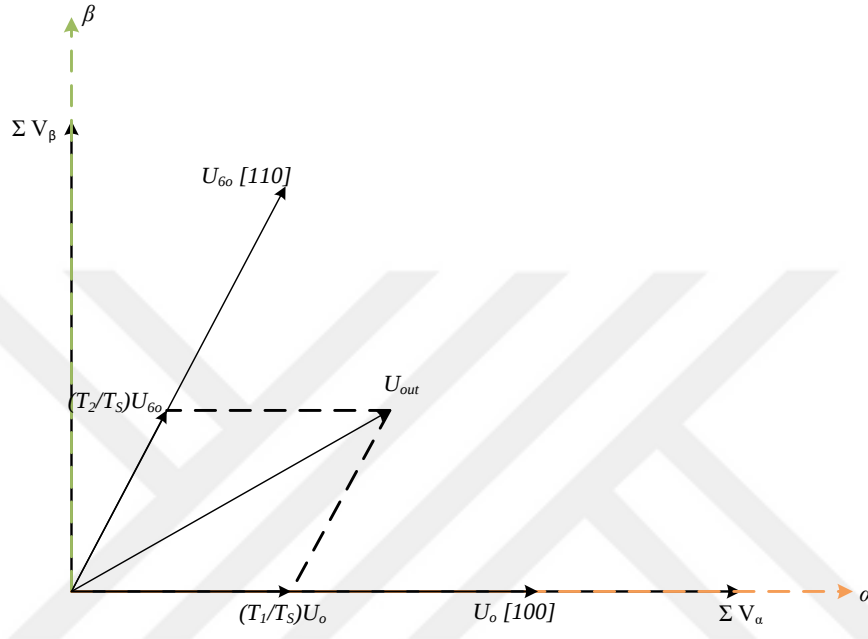
A	B	C	V_α	V_β	Vektör
0	0	0	0	0	$\vec{U}_{111}$
0	0	1	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$\vec{U}_{240}$
0	1	0	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\vec{U}_{120}$
0	1	1	$-\frac{2V_{DC}}{3}$	0	$\vec{U}_{180}$
1	0	0	$\frac{2V_{DC}}{3}$	0	$\vec{U}_0$
1	0	1	$\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$\vec{U}_{300}$
1	1	0	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\vec{U}_{60}$
1	1	1	0	0	$\vec{U}_{000}$



Şekil 2.12 Altı Konumlu Anahtarlama Yapısına Göre Temel Gerilim Vektörleri

Uzay vektörü modülasyonunun amacı yukarıda belirtilen olası 8 durumdan uygun olanı kullanarak bir çıkış gerilimi U_{xxx} elde etmektir. Bu modülasyon tekniği kullanılırken kontrol algoritması, motorun bir elektriksel çevrimini 60° 'lik sektörlere böler ve her biri için U_{xxx} değerinde bir gerilim vektörü üretir. Her sektörde yalnızca dört geçerli

anahtar durumu vardır ve bunlar ilgili sektöre bitişik iki aktif vektör ile iki sıfır vektöründen oluşur. Her bir anahtarlama durumu için zaman hesaplanarak herhangi bir $\vec{U}_{xxx}$ çıkış gerilim vektörü, $0^\circ - 360^\circ$ ve 0 ile $\frac{2V_{DC}}{3}$ aralığında sentezlenir.



Şekil 2.13 Uzay vektör modülasyon geometrisi [15]

Şekil 2.13'te gösterilen örnek 0° ile 60° arasında kalan sektör için çıkış vektör gerilimi U_{out} aşağıdaki denklem ile tanımlanır.

$$T_S = T_1 + T_2 + T_0 \quad (2.45)$$

$$U_{out} = \frac{T_1}{T_S} U_{0^\circ} + \frac{T_2}{T_S} U_{60^\circ} \quad (2.46)$$

Burada T_S , PWM anahtarlama frekansının periyodudur. T_0, T_1, T_2 ise U_0, U_{60} ve uygulanan sıfır vektörü ile harcanan zamanın genişliğini temsil eder. U_{out} çıkış gerilimi için U_α ve U_β değerleri Eşitlik 2.47 ve 2.48'de ki gibi hesaplanır.

$$U_\alpha = \frac{T_1}{T_S} |U_0| \cos(0) + \frac{T_2}{T_S} |U_{60}| \cos(60) \quad (2.47)$$

$$U_{\beta} = \frac{T_1}{T_s} |U_0| \sin(0) + \frac{T_2}{T_s} |U_{60}| \sin(60) \quad (2.48)$$

$\frac{2V_{DC}}{3}$ değerinin maksimum faz geriliminin $\frac{V_{DC}}{\sqrt{3}}$ 'e göre normalleştirilmesi sonucu,

$\frac{2}{\sqrt{3}}$ değerine sahip normalize edilmiş bir uzay vektör büyüklüğünü (U_{out}) verir. 2.47

ve 2.48 denklemleri süreler cinsinden çıkış geriliminin U_{α} ve U_{β} cinsinden hesaplanmasında kullanılır. İstenilen U_{α} ve U_{β} gerilim değerlerini üretmek için T_1 ve T_2 sürelerinin neye eşit olacağını hesaplamak için 2.49 ve 2.50 denklemleri kullanılır. Bunun için 0 ve 60 derece değerlerinde U_{α} ve U_{β} hesaplanarak; buradan T_1 ve T_2 süreleri çekilir.

$$T_1 = \frac{T_s}{2} (\sqrt{3}U_{\alpha} - U_{\beta}) \quad (2.49)$$

$$T_2 = T_s U_{\beta} \quad (2.50)$$

$$T_0 = T_s - T_1 - T_2 \quad (2.51)$$

T_1 ve T_2 normalize edilmiş U_{α} ve U_{β} nın fonksiyonu olarak uygulanan aktif vektörlerde harcanan süreler olup, T_0 periyodunun kalan kısmı ise uygulanan sıfır vektörlerden biri ile harcanır. Aktif vektörler için t_1 ve t_2 görev çevrimi (duty cycle) süresine karşılık gelen gerilim değerleri, Eşitlik 2.52 ve 2.53 kullanılarak hesaplanır.

$$t_1 = \frac{T_1}{T_s} = \frac{1}{2} (\sqrt{3}U_{\alpha} - U_{\beta}) \quad (2.52)$$

$$t_2 = \frac{T_2}{T_s} = U_{\beta} \quad (2.53)$$

Aynı hesaplama yöntemi diğer sektör açıları için (60° ve 120°) yapılırsa;

$$t_1 = \frac{T_1}{T_s} = \frac{1}{2} (-\sqrt{3}U_{\alpha} + U_{\beta}) \quad (2.54)$$

$$t_2 = \frac{T_2}{T_s} = \frac{1}{2}(\sqrt{3}U_\alpha + U_\beta) \quad (2.55)$$

Olarak elde edilir.

Eşitlik 2.52 e 2.53'te verilen formül kullanılarak her bir sektör için X, Y ve Z olarak tanımlanmış altı mümkün duruma karşılık gelen gerilim değerleri hesaplanmış olacaktır.

$$X = U_\beta \quad (2.56)$$

$$Y = \frac{T_1}{T_s} = \frac{1}{2}(\sqrt{3}U_\alpha + U_\beta) \quad (2.57)$$

$$Z = \frac{T_2}{T_s} = \frac{1}{2}(-\sqrt{3}U_\alpha + U_\beta) \quad (2.58)$$

Tüm sektörler için aktif vektörlerin görev döngüsü Tablo 2.4'te verilmiştir.

Tablo 2.4 Belirlenen altı sektör için t_1 ve t_2 sürelerine karşılık gelen gerilim değerlerinin tanımlanması [7]

Sektör	$0^\circ - 60^\circ$	$60^\circ - 120^\circ$	$120^\circ - 180^\circ$	$180^\circ - 240^\circ$	$240^\circ - 300^\circ$	$300^\circ - 0^\circ$
t_1	$-Z$	Z	X	$-X$	$-Y$	Y
t_2	X	Y	Y	Z	$-Z$	$-X$

Modifiye edilmiş Clarke dönüşümü kullanılarak ve basit bir karşılaştırma işlemi ile hesaplanan referans gerilim değerlerinin hangi sektöre karşılık geldiği belirlenmektedir.

$$V_{ref1} = U_\beta \quad (2.59)$$

$$V_{ref2} = \frac{1}{2}(\sqrt{3}U_\alpha - U_\beta) \quad (2.60)$$

$$V_{ref3} = \frac{1}{2}(-\sqrt{3}U_\alpha - U_\beta) \quad (2.61)$$

Eşitlik 2.59, 2.60 ve 2.61 Clarke dönüşümüne göre 90° ileri faz farklı 3 fazlı gerilim vektörlerini (V_{ref1} , V_{ref2} , V_{ref3}) vermektedir. Verilen gerilim değerlerini sıfır ile karşılaştırırsak;

$$\left\{ \begin{array}{l} V_{ref1} \leq 0, a' = 0 \\ V_{ref2} \leq 0, b' = 0 \\ V_{ref3} \leq 0, c' = 0 \end{array} \right\} \text{ve} \left\{ \begin{array}{l} V_{ref1} > 0, a' = 1 \\ V_{ref2} > 0, b' = 1 \\ V_{ref3} > 0, c' = 1 \end{array} \right\} \quad (2.62)$$

a' , b' ve c' değişkenlerinin ikili sayı sistemindeki karşılıklarından, bu duruma ait sektörleri hesaplamak mümkündür. İkili sayı sistemindeki sektör numarası N_{bin} ve gerçek sektör numarası S arasındaki bağıntı Eşitlik 2.63 ile tanımlanabilir. Bu durumda Tablo 2.5'te gösterilen ilişki bize sektör numarasını verir.

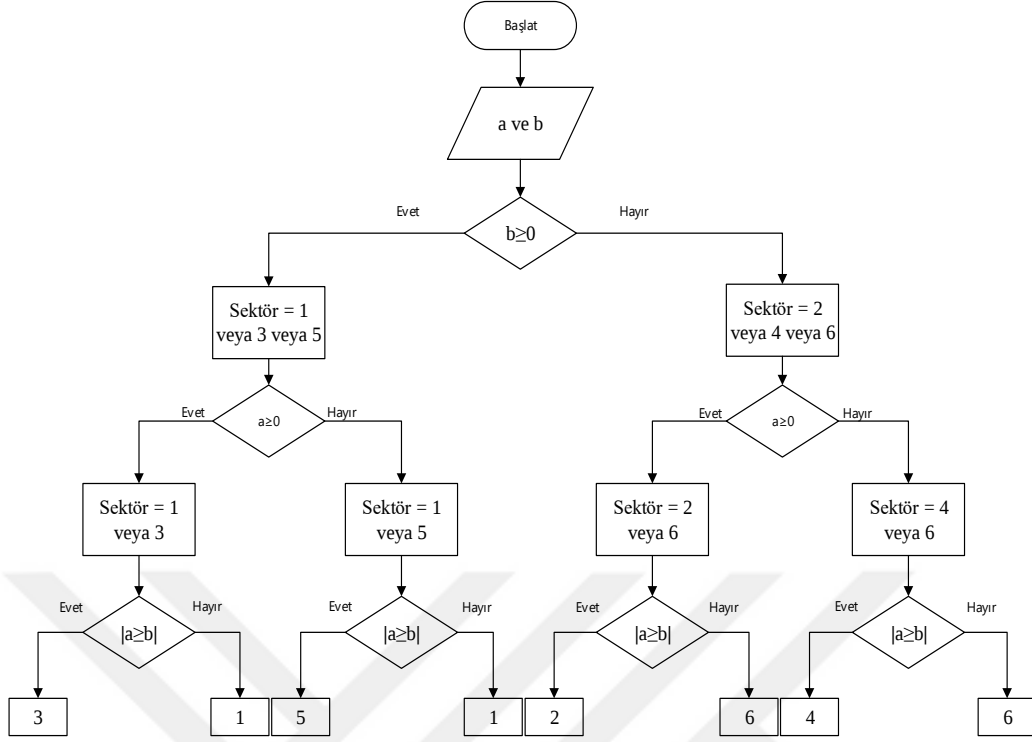
Tablo 2.5 İkili sayı sistemine göre karşılık gelen sektör numaraları [7]

a'	b'	c'	N_{bin}	Sektör S
0	0	0	0	<i>kullanılmaz</i>
0	0	1	1	2
0	1	0	2	6
0	1	1	3	1
1	0	0	4	4
1	0	1	5	3
1	1	0	6	5
1	1	1	7	<i>kullanılmaz</i>

N_{bin}	3	1	5	4	6	2
S	1	2	3	4	5	6

$$N_{bin} = 4c' + 2b' + a' \quad (2.63)$$

$V_a = U_\alpha$ ve $V_b = \frac{U_\beta}{\sqrt{3}}$ şeklinde tanımlandığında sektör belirleme algoritmasına ait akış şeması Şekil 2.15'te verilmiştir.



Şekil 2.14 Sektör belirleme algoritması akış diyagramı [8]

SVPWM'in amaçlarından biri belirlenen zamanda anahtar durumunu değiştirmektir. Bu nedenle ilk sektör için açma kapama süresi aşağıdaki şekilde tanımlanır:

$$t_{aon} = \frac{T_{PWM} (1 - T_1 - T_2)}{2} \quad (2.64)$$

$$t_{bon} = t_{aon} + T_{PWM} T_1 \quad (2.65)$$

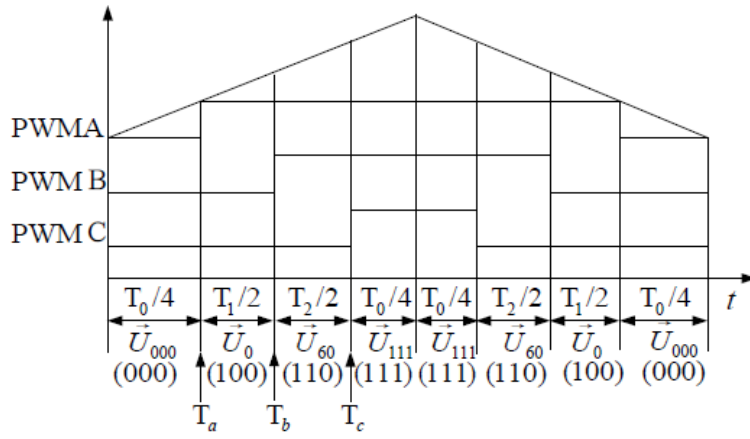
$$t_{con} = t_{bon} + T_{PWM} T_2 \quad (2.66)$$

Burada, t_{aon} V_{ref1} geriliminin “on” süresini, t_{bon} V_{ref2} geriliminin “on” süresini ve t_{con} V_{ref3} geriliminin “on” süresini ifade etmektedir. Tablo 2.6 her faz ve tüm sektörler için döngü çevrimi durumları verilmiştir.

Tablo 2.6 Döngü çevrim atamaları [7]

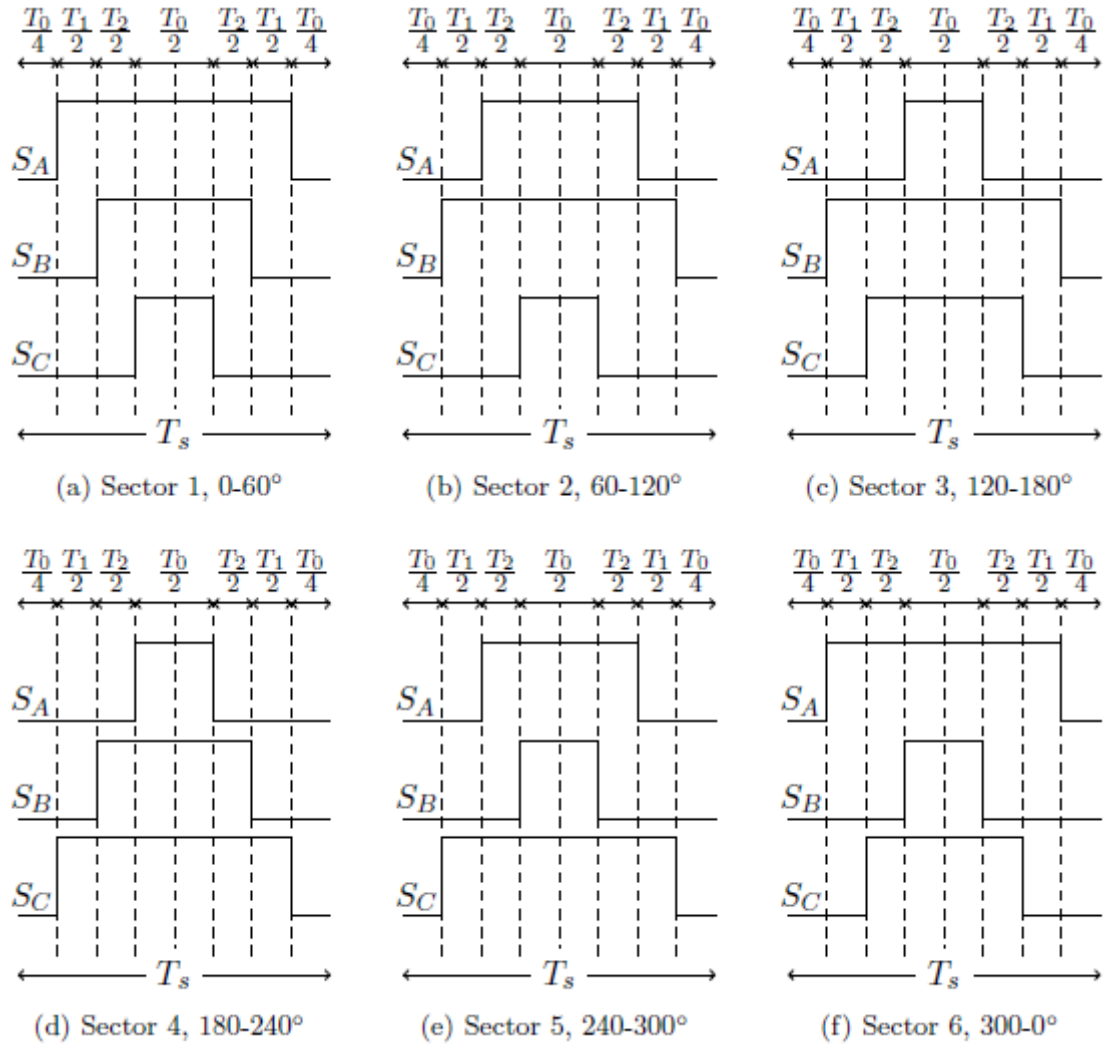
Sektör	$0^\circ - 60^\circ$	$60^\circ - 120^\circ$	$120^\circ - 180^\circ$	$180^\circ - 240^\circ$	$240^\circ - 300^\circ$	$300^\circ - 0^\circ$
T_a	t_{aon}	t_{bon}	t_{bon}	t_{con}	t_{con}	t_{aon}
T_b	t_{bon}	t_{aon}	t_{con}	t_{bon}	t_{aon}	t_{con}
T_c	t_{con}	t_{con}	t_{aon}	t_{aon}	t_{bon}	t_{bon}

Şekil 2.15 Sektör 1'e ait T_a, T_b ve T_c sürelerini ve bu sürelerle karşılık gelen açı değerlerini ($0^\circ - 60^\circ$) göstermektedir.



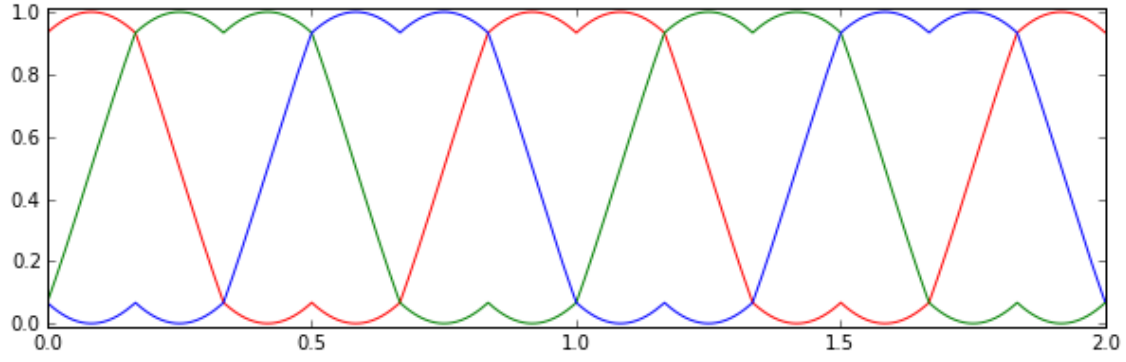
Şekil 2.15 Sektör 1 için anahtarlama deseni [10]

$0^\circ - 360^\circ$ Aralığında ve 1'den 6'ya kadar değer alabilen 6 sektörlü bir modelde, her bir konum değişiminde üç fazda oluşacak gerilim değerlerinin “On” süreleri Şekil 2.16'da görülmektedir.

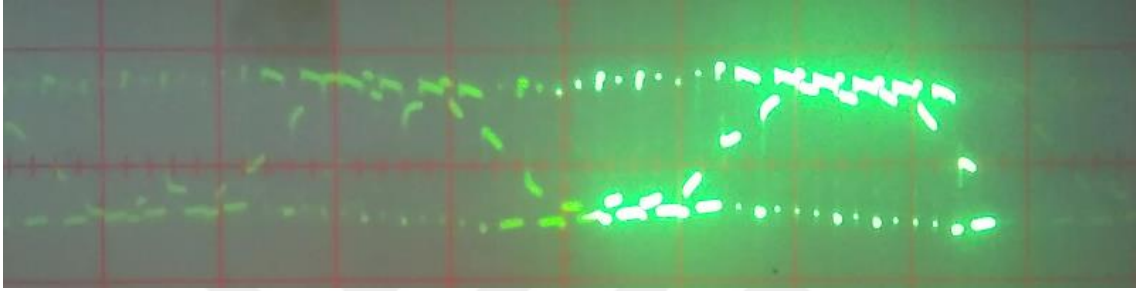


Şekil 2.16 Bütün sektörler için uzay vektör modülasyonu anahtarlama desenleri [7]

Yüksek frekanslı bir üçgen dalganın sinüs sinyali ile modüle edilmesi ile oluşan modüleli çıkış dalgası formu, SVPWM çıkışı için de benzer şekilde olacaktır. Üç faz için SVPWM sinyali şekil 2.17 ve tek faz için LabVIEW tabanlı gerçek model 2.18 de görülmektedir.



Şekil 2.17 Taşıyıcı filtreli Uzay vektör modülasyon çıkışı [7]



Şekil 2.18 Bir faz için Uzay Vektör Modülasyonu Osiloskop çıktısı

2.7. Sonuç

Bu bölümde ilk olarak FSDA motorların yapısı, çalışması, sürücü ve kontrol yöntemleri, mekanik ve elektriksel açı arasındaki farklılık ve matematiksel denklemler verilmiştir. Ardından FSDA motorlar ile FOC kontrol yapısı arasındaki ilişki ve teori tartışılmıştır. Bu bağlamda Clarke dönüşümü, Park dönüşümü, alan etkili algılayıcılar ve SVPWM gibi kavramlar üzerinde durulmuştur. Temelde bu bilgiler, belirtilen kontrol yöntemini gerçek FSDA motorda uygulamak ve kontrol yazılımı için gereklidir. Söz konusu kontrol yöntemine ait teori, çok iyi bilinmekle birlikte uygulamada birtakım zorluklar ile karşılaşılacaktır. Bu nedenle belirtilen kavramlar, sadece kâğıt üzerinde incelenmemiş; aynı zamanda bir sonraki bölümde de anlatıldığı gibi gerçek zamanlı uygulaması yapılmış; elde edilen sonuçlar benzetim programları ile de desteklenmiştir.

3. FSDA MOTORLAR İÇİN SÜRÜCÜ ve KONTROLÖR TASARIMI

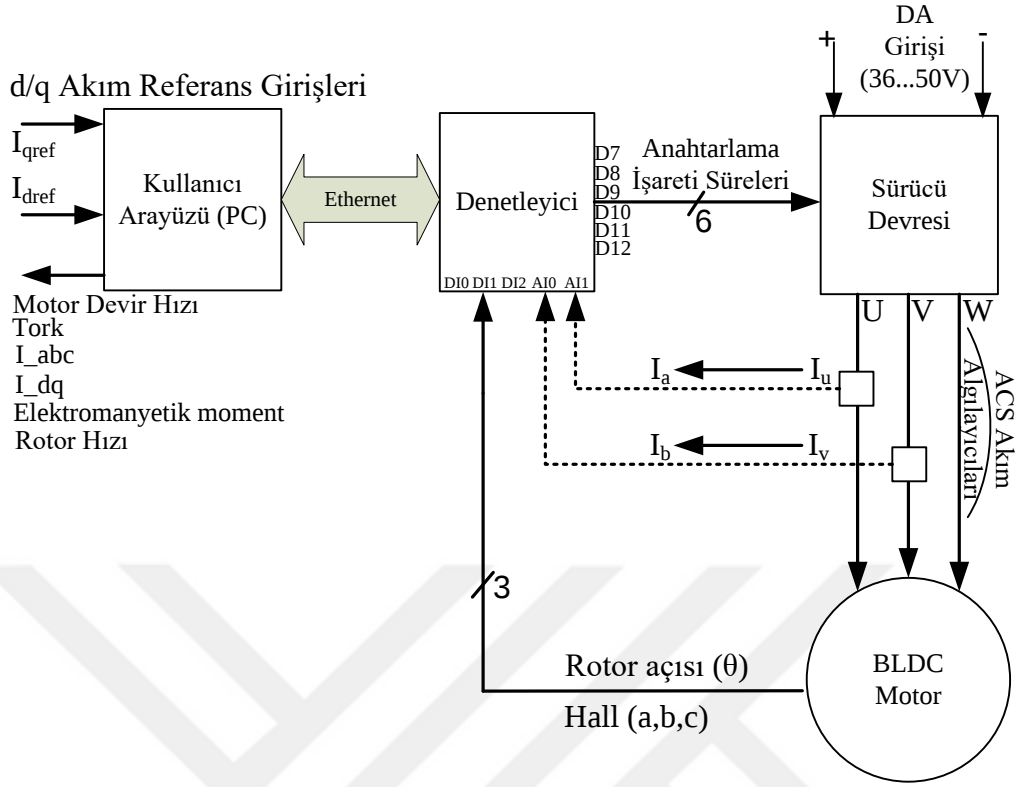
3.1. Giriş

Bu tezde FPGA tabanlı bir kontrolör kartı üzerinden amaca uygun tasarlanan sürücü devresi kullanılarak FSDA motorların sürülmesi konusu ele alınmıştır. Anahtarlama frekansı, güç yoğunluğu ve hızın önemli olduğu bu ve buna benzer uygulamalarda FPGA tabanlı kontrolörlerin kullanımı önemlidir. Bu tezde bisiklet gibi hafif elektrikli araçlara yönelik, endüstriyel uygulamalarda sıklıkla kullanılan fırçasız doğru akım motoru için bir sürücü tasarlanarak; maksimum verim elde etmek için uygun yazılım ve donanımların belirlenmesi amaçlanmıştır.

Tez sürecinde geliştirilen FSDA motor sürücüsü ve kontrolöre ait blok şema Şekil 3.1 de verilmiştir. Şekilden de görüldüğü üzere sistemin donanım yapısı dört ana bölümden oluşmaktadır:

- Denetleyici (FPGA tabanlı geliştirme kartı, NI sbRIO 9637)
- Fırçasız DA motoru
- Altı konumlu anahtarlama ve sürücü devresi
- Motorun devir, tork ve akı değerlerinin ölçümü ve kontrolöre aktarılması

Bu bölümde sistemde yer alan tüm donanım bileşenleri ve kurulumundan bahsedilmiştir.



Şekil 3.1 FSDA motor sürücü ve kontrol devresi blok diyagramı

3.2. Denetleyici

Bu tezde geliştirme kartı olarak National Instruments (NI) firmasının üretmiş olduğu sbRIO-9637 isimli FPGA kartı kullanılmıştır. Bu kart, Linux Real-Time işletim sistemini çalıştıran gerçek zamanlı bir işlemci ile kullanıcı tarafından yeniden yapılandırılabilen APKD ve giriş/çıkış üniteleri tek bir baskı devre karta entegre edilmiş gömülü bir denetleyicidir [23]. Toplam 48 adet Analog ve Sayısal giriş/çıkış kanalı, 667 MHz çift çekirdekli işlemci, 512 MB dinamik RAM belleği, 512 MB dahili depolama birimi ile Zynq-7020 FPGA CompactRIO tek kartlı denetleyici ile yüksek performans ve güvenilirlik isteyen uygulamalar için güçlü bir çözüm haline gelmiştir [23,24]. sbRIO 9637 kartı esneklik, güvenilirlik ve yüksek performans gerektiren yüksek hacimli OEM (Original Equipment Manufacturer – Orijinal Ekipman Üreticisi) uygulamalarını kolayca programlayabilecek şekilde tasarlanmıştır. Bu kontrol kartı, Gigabit Ethernet, CAN, USB, seri ve SDHC portlarına ve en zorlu ortamları karşılayabilecek bir çalışma sıcaklık aralığına sahiptir. sbRIO-9637 de 16 adet 16 bit çözünürlüğe sahip analog giriş

(AI), 4 adet 16 bit çözünürlüğüne sahip analog çıkış (AO) ve 28 adet 3.3V sayısal giriş/çıkış hattı bulunmaktadır. Şekil 3.2 sbRIO-9637 FPGA kartı gösterilmiştir.



Şekil 3.2 NI sbRIO-9637 kart [26]

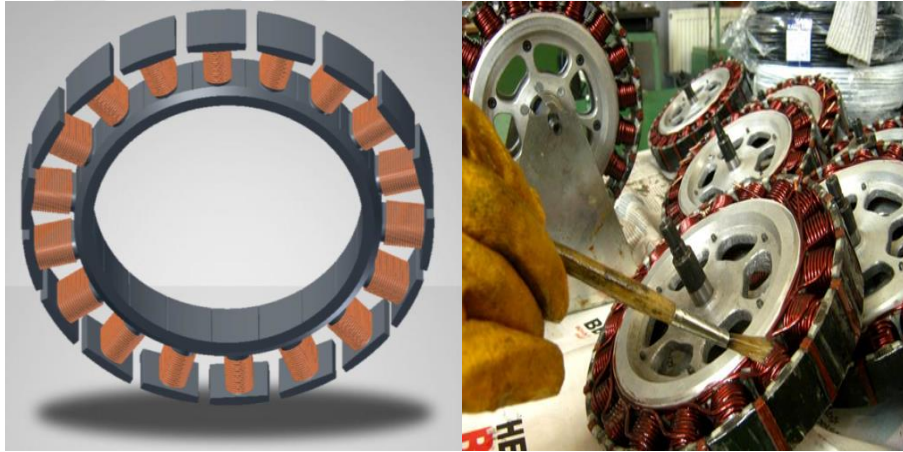
Tablo 3.1 sbRIO-9637 karta ait teknik özellikler [26]

Güç Yönetimi	
Çalışma Gerilim Aralığı	19-30 volt
Güç Tüketimi (Yüksüz)	7.75 watt
Ağ Yapısı	
Arayüz	10BASE-T ve 100BASE-T Ethernet
Uygunluk	IEEE 802.3
Haberleşme Oranı	10Mb/s, 100Mb/s
Zynq 7020 yapılandırılabilir FPGA	
Programlanabilir Lojik Hücre	85.000
Look Up Tablo (LUTs)	53.200
Flip-Flop	106.400
Gömülü RAM bloğu	512MB Dinamik
İşlemci Hızı	Çift Çekirdek ARM-Cortex A9
DSP Bölümler (18x25 MACCs)	220
DSP Performans (Simetrik FIR)	276 (GMACs)
Haberleşme	
FPGA ile PCI Haberleşme	32 bit Adres/Veri yolu
Giriş / Çıkış portlar	
Dijital Giriş / Çıkış Kanal Sayısı	16 bit 28 adet 3.3v
Analog Giriş	16 bit 16 adet $\pm 10v$
Analog Çıkış	16 bit 4 adet $\pm 10v$

sbRIO-9637 kartta bulunan FPGA ve işlemci NI LabVIEW grafiksel programlama araçları ile sentezlenebilmekte ve programlanabilmektedir. Kart ile bilgisayar arasındaki iletişim 10/100BASE-T Ethernet portu üzerinden ya da bir RS-232 seri iletişim portu üzerinden sağlanmaktadır. sbRIO-9637 karta ait bazı özellikler Tablo 3.1 de verilmiştir.

3.3. FSDA MOTOR

Bir yandan DC motorlardaki moment-devir karakteristiği avantajını koruyarak; diğer taraftan da fırça ve kolektör düzeneğinin getirdiği dezavantajlardan kurtulmak için FSDA motorların gerekliliği ilk bölümde vurgulanmıştı. Bu tezde kullanılan motorun iç yapısı ve tasarımı Şekil 3.3'te verilmiştir. Görüldüğü gibi motor 18 oluk ve 12 kutup çiftinden (24 kutup) oluşmaktadır. Motora ait teknik bilgi ve özellikler Tablo 3.2 de verilmiştir.



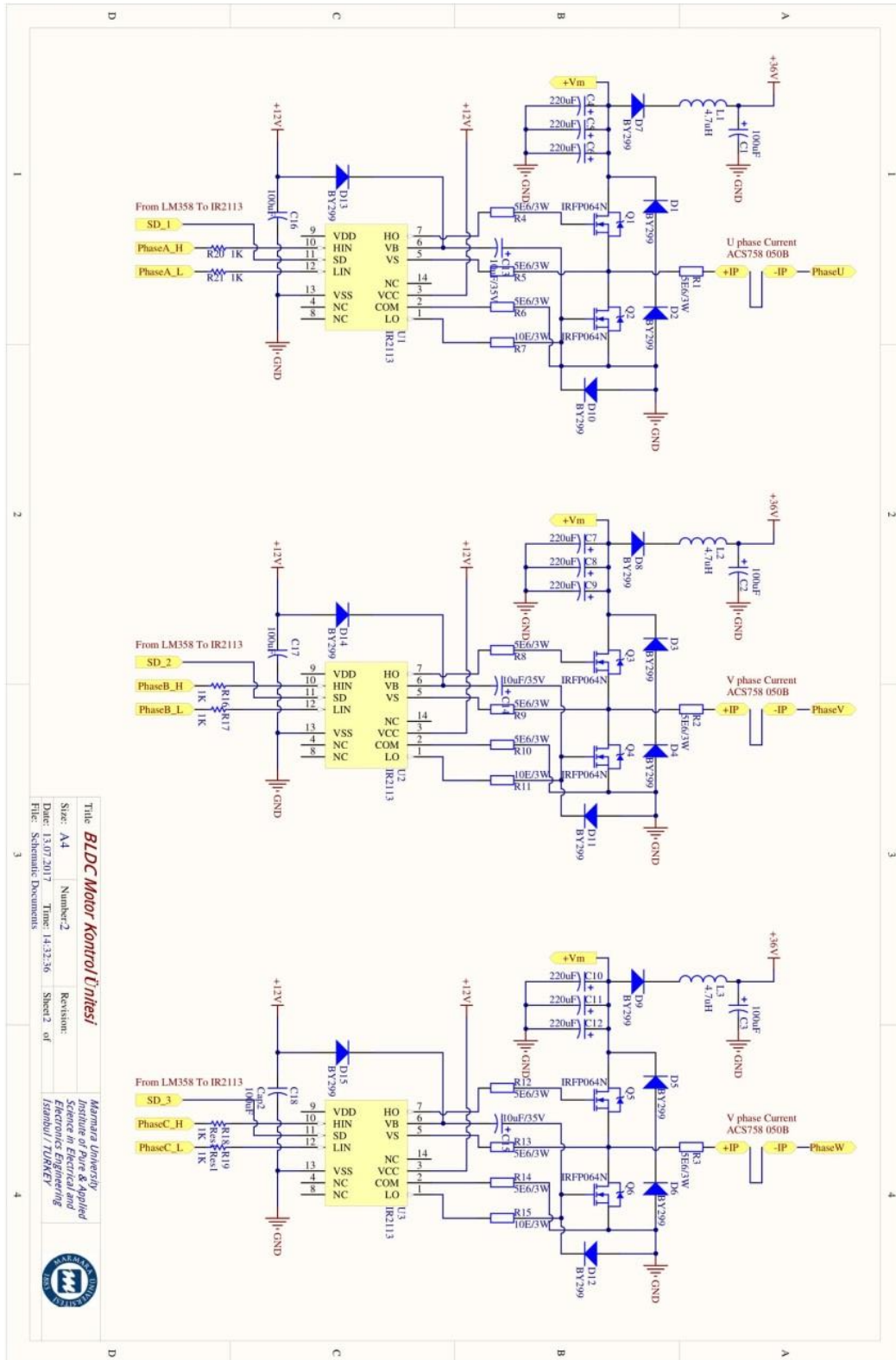
Şekil 3.3 FSDA motoru iç yapı ve tasarım [28]

Tablo 3.2 FSDA Motora ait teknik bilgiler [28]

Genel Veri		Tam Yükte		Yüksüz Manyetik Veris	
Omik Direnç Değeri (Ω)	0.08	Ortalama Giriş Akımı (A)	8.1	Stator Akı Yoğunluğu (Tesla)	1.788
Endüktans Değeri (mH)	2.716	Armatür Akımı (A)	7.2	Rotor Akı Yoğunluğu (Tesla)	2.8057
Motorun Manyetik Akı Bağlantısı (mNΦ)	44.648	Giriş Gücü (W)	405	Stator Amper Tur (A/T)	378
Ortalama çıkış gücü (kW)	1	Çıkış Gücü (W)	380	Kilitli Rotor Akımı (A)	1.64
Ortalama Gerilim (V)	50	Verimlilik (%)	94	Kaçak Akı Faktörü	1
Kutup Sayısı	24	Ortalama Hız (RPM)	300	Stator By-Pass Faktörü	0.0268
Oluk Sayısı	18	Ortalama Tork (N.m)	12.12	Rotor By-Pass Faktörü	0.096
Maksimum Anma Akımı (Amper)	30	Kilitli Rotor Torku (N.m)	27.25	Ortalama Tork (N.m)	0.203
Minimum Anma Akımı (Amper)	10	Armatür Akım Yoğunluğu (A/mm^2)	1,728	Ortalama Hız (RPM)	341
Maksimum Çalışma Sıcaklığı ($^{\circ}C$)	75	Bakır Kayıpları (W)	12.27	Başlangıç Anında Armatür Reaktif Amper Tur Değeri (A/T)	173.95
Ortalama Hız (RPM)	300	Toplam kayıplar (W)	24.13	Manyetik Amper Tur Değeri (A/T)	-1678

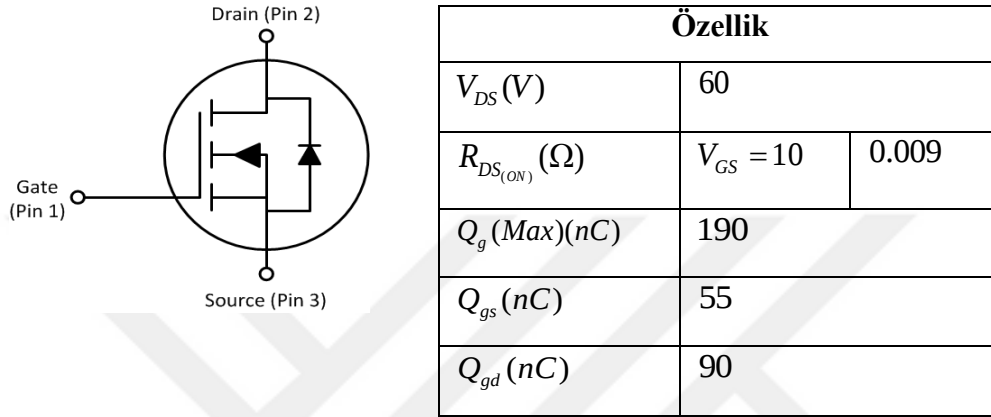
3.4. Anahtarlama ve Sürücü Devresi

Bu tezde FSDA motoru sürmek için altı konumlu anahtar (invertör) yapısından oluşan ve Şekil 3.4'te açık devre şeması verilen bir sürücü devresi tasarlanmıştır.



Şekil 3.4 FSDA Motor sürücü devresi açık devre şeması

Tasarımda 6 adet Vishay Siliconix N kanallı IRFP064 tipi MOSFET (Metal Oxide Semiconductor Field Effect Transistor, Metal Oksit Yarıiletken Alan Etkili Transistör) kullanılmış; bunlar üç fazlı köprü tipinde bağlanmıştır. Üçüncü nesil güç MOSFET'i olarak adlandırılan IRFP064; tasarımcıya hızlı anahtarlama, sağlamlaştırılmış gövde, yüksek dayanıklılık ve düşük maliyet gibi üstünlükler sunmaktadır. Şekil 3.5 kullanılan transistöre ait paket yapısı ve bazı teknik özellikleri verilmiştir.



Şekil 3.5 IRFP064 N kanallı MOSFET'in yapısı ve teknik özellikleri

Transistörlere ait her kapı, International Rectifier firmasının yüksek gerilim üç fazlı IR2113 tümleşik devre (Integrated Circuit IC) sürücüsü tarafından uyartılır. IR2113 entegresi; bağımsız lojik 1 ve lojik 0 çıkış kanallarına sahip olup; yüksek gerilim, yüksek hızlı güç MOSFET'i ve IGBT sürücüleri için uygundur. Ayrıca aşırı akım ve aşırı sıcaklıkta kapanmalara karşı dâhili bir koruma özelliğine sahiptir. Yüksek frekanslı devrelerde kullanımı basitleştirmek için yayılma gecikme zamanı CMOS (Complementary Metal Oxide Semiconductor Bütünleyici Metal Oksit Yarıiletken) teknolojisi ile en aza indirilmiştir. Şekil 3.6 sürücü entegresi IR2113'e ait kılıf yapısı ve teknik detayları göstermektedir.



Özellik	
V_{OFFSET}	600V
$I_{OUT} \pm$	2A / 2A
V_{OUT}	10 – 20V
$t_{ON/OFF}$	94 / 120ns
Gecikme uyumu	20ns Max

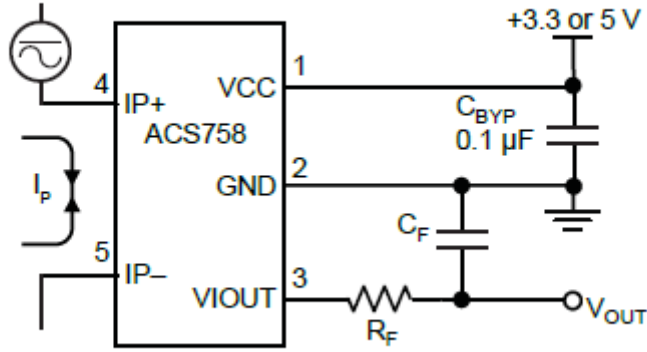
Şekil 3.6 IR2113 Mosfet Sürücü ve teknik detaylar

Devrede yer alan 2 adet ACS758 Hall etkisine dayalı doğrusal akım algılayıcısı, motorun U ve V fazlarına bağlı 6-konumlu anahtarlama köprüsünün çıkışlarına seri olarak bağlanmıştır. Geri bildirim için gerekli faz akımları ACS758 akım algılayıcısı ile Volt başına 1 mA olarak ölçülmektedir. Ölçülen akım değerleri sbRIO karta ait iki adet analog giriş ile 16 bit çözünürlükte sayısal bir işarete dönüştürülür. Elde edilen sayısal işaretler sbRIO kartta işlenir. ACS758 alan etkili akım algılayıcısı Allegro firması tarafında tek ve çift yönlü olarak (ACS758U, ACS758B) üretilmektedir. Bu tümleşik devreye ait paket yapısı ve ayak bağlantıları Şekil 3.7 de gösterilmiştir.



Şekil 3.7 ACS758u alan etkili algılayıcı [29]

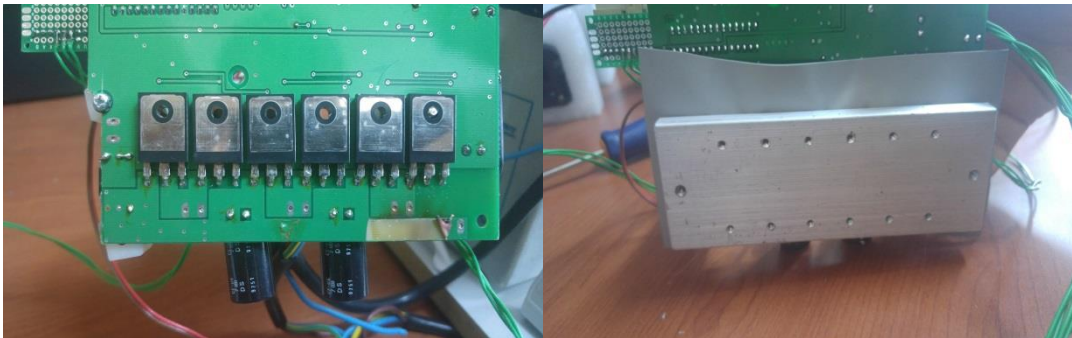
ACS758 giriş ile doğrusal olarak değişen tek veya çift yönlü AA veya DA bir çıkış gerilimi, V_{OUT} üretir. Algılayıcıya ait tipik bağlantı şeması Şekil 3.8 de gösterilmiştir.



Şekil 3.8 ACS758 tipik bağlantı şeması [29]

ACS758 Hall etkisine dayalı doğrusal akım algılayıcısı, ayrıca sürücü devresinde aşırı akım koruma görevini de yerine getirmektedir. Bu tezde tasarlanan sürücü devresinde kaynak gerilimine bağlı olarak anma akımı, 8 A dir. Bu bakımdan ACS758 devresi ile aşırı akım koruma 8 A'e göre ayarlanmıştır. Yani devrede herhangi bir kısa devre olduğunda veya malzemelerden biri bozulduğunda kaynaktan çekilebilecek akım 8 A ile sınırlandırılmıştır. Bu da devrenin nominal akımı olduğu için devreye herhangi bir hasar vermeyecektir. Sürücü devresinde Şekil 3.9'den de görüldüğü gibi yukarıda belirtilen malzeme dışında aktif ve pasif (direnç, kapasite, transistör diyot, vb.) başka elemanlar da kullanılmıştır.

Şekil 3.9'da güç mosfetleri ana kart üzerine tam köprü olarak bağlanmış ve termal ped, MOSFET'ler ile soğutucu arasına gelecek şekilde yerleştirilmiştir.

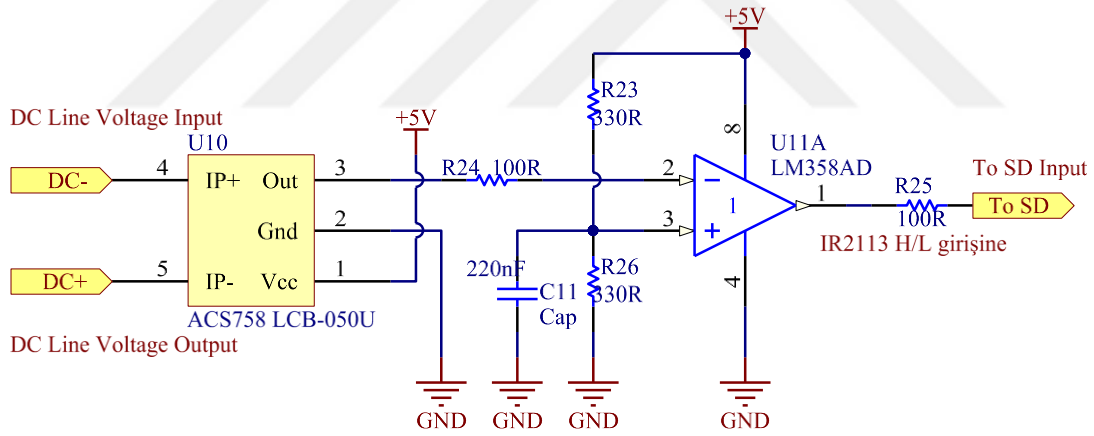


Şekil 3.9 Soğutucu ve MOSFET'lerin kart üzerine yerleştirilmesi

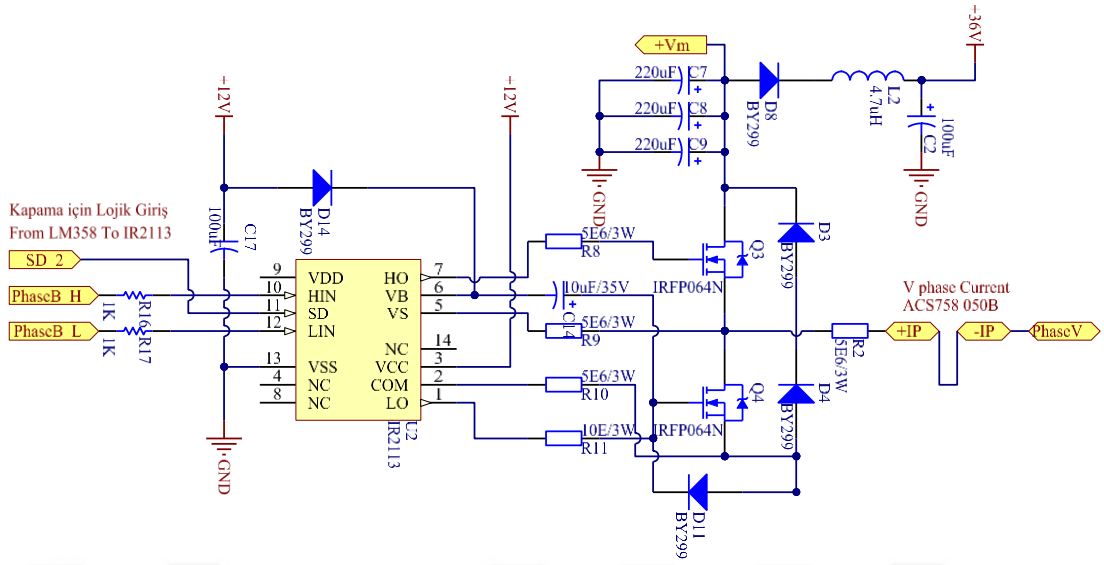
Şekil 3.10 a'da ACS758 Hall tabanlı akım algılayıcısı, anlık akım değerlerinin ölçümü için kullanılmıştır. Ölçülen akım değeri, 8 A (veya $60\text{mV/A} \times 8\text{A} = 480\text{ mV}$) değeri ile

LM358 devresi üzerinden karşılaştırılır. ACS758 algılayıcısının maksimum akım çektiği kısa devre anında, karşılaştırıcı çıkışındaki gerilim değeri “High” seviyeye dönüştürülür. Lojik seviye H olduğunda yani akım algılayıcı 8 A ve üzeri akım tespit etmişse, IR2113 entegresi kesime giderek; devreyi koruyacaktır. Eğer karşılaştırma devresi çıkış “Low” seviyesinde ise devre çalışmaya devam edecektir.

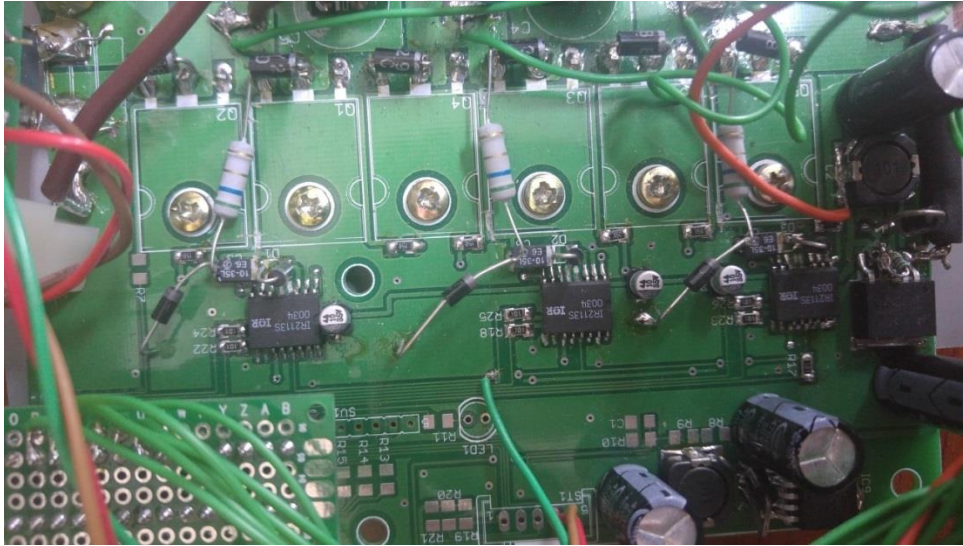
Şekil 3.10 b’de devrenin herhangi bir fazı için (u, v, w) açık devre şeması ile ana kart üzerindeki IR2113 MOSFET sürücü entegresinin yüksek ve düşük (H_{out} / L_{out}) çıkışları mosfetlerin kapı girişine bağlanmış ve Mosfetler ile sürücü entegreleri toprak hatları birleştirilmiştir. Şekil 3.10 b’de görüldüğü gibi BY299 tipi yüksek hızlı kurtarma doğrultucu diyotları IR2113 entegrelerinin, ters kırılma gerilimi anında bozulmasını önlemek amacıyla 6 ve 13. ayakları (V_B, V_{SS}) arasına seri olarak 100µF lık bir kondansatör bağlanmıştır. Aynı şekilde bu diyotlardan birer tane de her bir seri MOSFET çiftlerinin Akaç (Drain) ve Kaynak (Source) pinlerinin arasına bağlanarak; açma-kapama sırasında oluşan ters akımlardan korumak için de kullanılmıştır.



a)



b)



c)

Şekil 3.10 Sürücü devresi a) Akım sınırlama için karşılaştırmalı devre, b) Bir faz için açık devre şeması, c) Sürücü devresi üstten görünüm

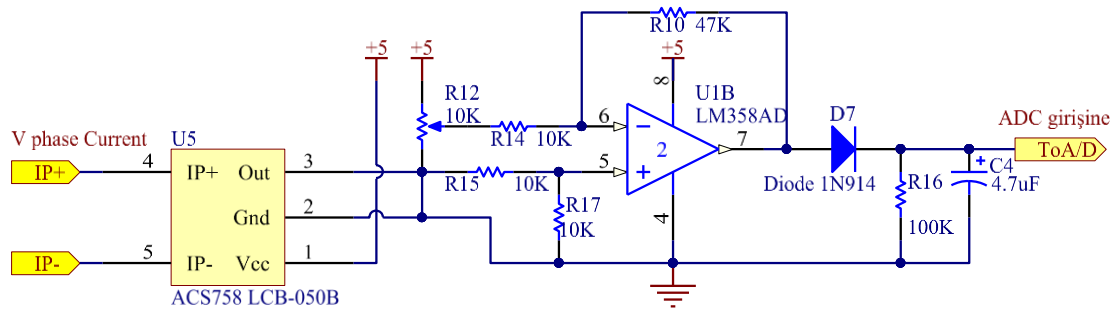
3.5. ADC Bloğu

Çevremizdeki sıcaklık, ağırlık, basınç vb. büyüklükler, analog olarak ortaya çıkmaktadır. Bu durumda bu büyüklükler, sadece 0 ve 1 gibi ikili değerler değil; belli

bir aralıkta (minimum ve maksimum arasında) çok geniş bir yelpazede çeşitli değerler alabilmektedirler. Denetlenecek fiziksel büyüklükler daha çok analog değişkenler olmasına karşın; bilgiyi işleyen denetleyici donanımlar (sayısal sistemler, mikroişlemciler, bilgisayarlar) sayısal veriler ile çalışmaktadır. Bu sistemler bilgiyi daha güvenli ve daha hızlı işler ve değerlendirirler. Denetleyici çıkışında oluşan sayısal bilgi dış dünyaya aktarılırken tekrar analoğa çevrilmesi gerekebilir. Bahsedilen bu fiziksel büyüklükler algılayıcı ve dönüştürücüler yardımıyla elektriksel formata dönüştürülürler. Bu işaretler bilindiği gibi analog formdadır. Daha sonra bu işaret bir ADC yardımıyla sayısal forma çevrilir. Sayısal sistem bu bilgiyi istenilen bir biçimde işler ve bir sonuç elde eder. Bu sonuç analog veya sayısal olarak değerlendirilebilir. Eğer elde edilen sonuç analog olarak değerlendirilecekse tekrar analoğa çevrilmesi gerekir. Sayısal işareti analog işarete çevirme işlemi ise DAC ile mümkündür.

3.5.1. Akım Algılayıcı Çıkışların FPGA karta Uygulanması

sbRIO-9637 FPGA kartı yukarıda da bahsedildiği gibi 16 bitlik 16 analog giriş/çıkış kanalına sahip bir denetleyicidir. Şekil 3.11’de ACS758 akım algılayıcısından gelen analog verileri denetleyicinin doğrudan analog girişlerine (AI1 ve AI2) iletmektedir. Arada herhangi bir dönüştürücü (DAC veya ADC) kullanılmamıştır. Denetleyicide bulunan AI1 ve AI2 girişlerine uygulanan analog veriler denetleyici içerisinde 16 bitlik sayısal verilere dönüştürülerek LabVIEW FPGA kodlarında sentezlenir. Denetleyiciye ait 16 bitlik 16 analog g/ç kanalları $\pm 10V, \pm 5V, \pm 2V$ şeklinde ölçeklenebilir bir yapıya sahiptir. Giriş gerilimine göre kanallar ölçeklendirilebilmektedir.



Şekil 3.11 Bir faz için ölçülen akım değerinin FPGA kartın analog girişine uygulanması

4. SİSTEM YAZILIMI VE UYGULAMA

4.1. Giriş

Bu bölüm motor kontrol sisteminin tamamının FPGA üzerinde uygulamasını açıklamaktadır. Bu tezin yazılım kısmı, kontrol algoritması ve kullanıcı arayüzü olmak üzere iki bölümden oluşmaktadır. Uygulamada kullanılan VI'ların bir kısmı (PI ve Hall Angle Calculator) hazır olarak LabVIEW de bulunan Motion paletinin altında bulmak mümkündür.

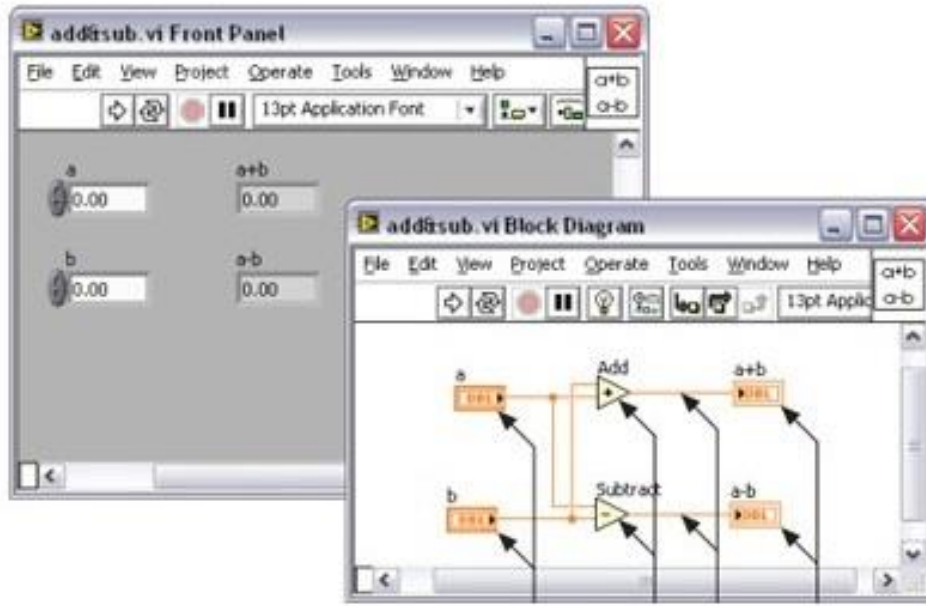
4.2. LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench)

Motor kontrol yazılımı için LabVIEW ortamı seçilmiştir. LabVIEW National Instruments firmasının tasarlamış olduğu grafiksel programlama platformudur. Ağırlıklı olarak ölçümleme ve prototipleme için kullanılır. Diğer metin tabanlı yazılım dillerine nazaran kod yazma işlemi grafiksel simgelerden ve bunları birbirine bağlayan hatlardan oluşmaktadır. LabVIEW, işlerinin bir parçası olarak bir programa ihtiyaç duyan bilim insanları ve mühendisler gibi kişiler için tasarlanmış interaktif bir program geliştirme ve uygulama sistemidir. LabVIEW uygulamaları Windows, Mac OSX veya Linux işletim sistemleri altında çalışır. LabVIEW ile Microsoft Pocket PC, Microsoft Windows CE, Palm OS gibi platformlarda ve FPGA, Sayısal Sinyal İşlemciler, Mikroişlemciler gibi değişik gömülü platformları içeren sistemlerde çalışabilen programlar yapılabilir. Programlama esnasında yardımcı olabilecek ve hafıza işgal etmeden kullanılabilecek çok geniş bir fonksiyon ve alt program kütüphanelerine sahiptir. Geleneksel programlama dillerinde karşılaşılabilecek gizli programlama problemleri ile LabVIEW de daha az karşılaşılabilmektedir. LabVIEW, Data Acquisition (DAQ) General Purpose Interface Bus (GPIB) için kod kütüphaneleri gibi çok özel kütüphane uygulamalarını ve seri aygıt kontrolü, veri analizi, veri sunumu, veri depolama ve internet üzerinden haberleşme gibi farklı uygulamaları da içermektedir. Analiz kütüphanesi, sinyal üretme, sinyal işleme filtreleri, Windows istatistik ve regresyonları, lineer cebir ve dizi aritmetiği gibi çok yönlü ve kullanışlı fonksiyonları bünyesinde barındırmaktadır. Verileri istenilen herhangi bir formda (çizelge, grafik ve kullanıcı tanımlı grafik gibi) kullanılabilecek çıkış seçeneklerine sahiptir. Bu tez çalışmasında oluşturulan kod

blokları LabVIEW 2015'te hazırlanmış ve bir Ethernet kablosuyla bilgisayara bağlı bir NI sbRIO 9637 gömülü geliştirme kartı üzerinde çalıştırılmıştır. LabVIEW ön panel ve blok diyagram kısmı olarak iki bölümden oluşmaktadır.

Ön panel, kullanıcı ara yüzüdür ve oluşturulacak uygulamayı kullanacak operatörün sisteme değerler girmesine ve çıkışlarını görmesine yardımcı olur.

Blok diyagram, program komutlarının koşulduğu yerdir. Ön panelde kullanıcı bir taraftan kontrolü sağlarken diğer taraftan blok diyagramda sanal enstrümanlar çalışmaktadır. Ayrıca NI firmasının geliştirdiği donanım ile gerçek sistemlere hükmedebilmektedir. Blok diyagram görsel programlama dillerinde kod yazma bölümüne, ön panel ise kullanıcı ara yüzünün hazırlandığı formlara karşılık gelmektedir. Şekil 4.1 blok diyagram ve ön panele ait görsellerdir.

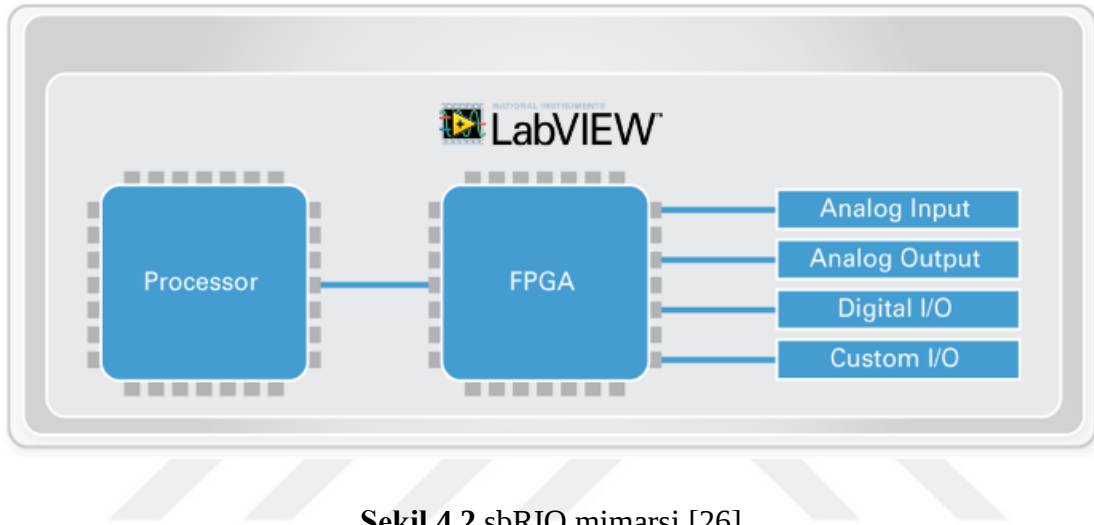


Şekil 4.1 Blok diyagram ve Ön panel görseli

4.2.1. sbRIO Mimarisi

Yeniden yapılandırılabilir bir gömülü sistem mimarisine sahip olan sbRIO, Şekil 4.2 de de görüldüğü gibi üç ana kısımdan meydana gelmektedir: İşlemci, FPGA ve Giriş/Çıkış terminalleri

İşlemci (Processor), daha önceden de bahsedildiği gibi NI Linux Real-Time çalışan 667 MHz çift çekirdekli ARM A9 dan, NI Linux Real-Time veya WES7 işletim sistemlerini çalıştıran 1.9 GHz 4 çekirdekli Intel Atom'a kadar çok çeşitli yüksek performanslı gömülü işlemciler sunmaktadır. İşlemci, uygulamaları çalıştırma, veri setleri düzenleme, sinyalleri işleme, veri günlüğü oluşturma gibi ortak görevleri yerine getirmek için LabVIEW'in entegre Eclipse desteği aracılığıyla veya sezgisel LabVIEW G veri akışı (dataflow) programlama dili C/C++ ile programlanabilmektedir.



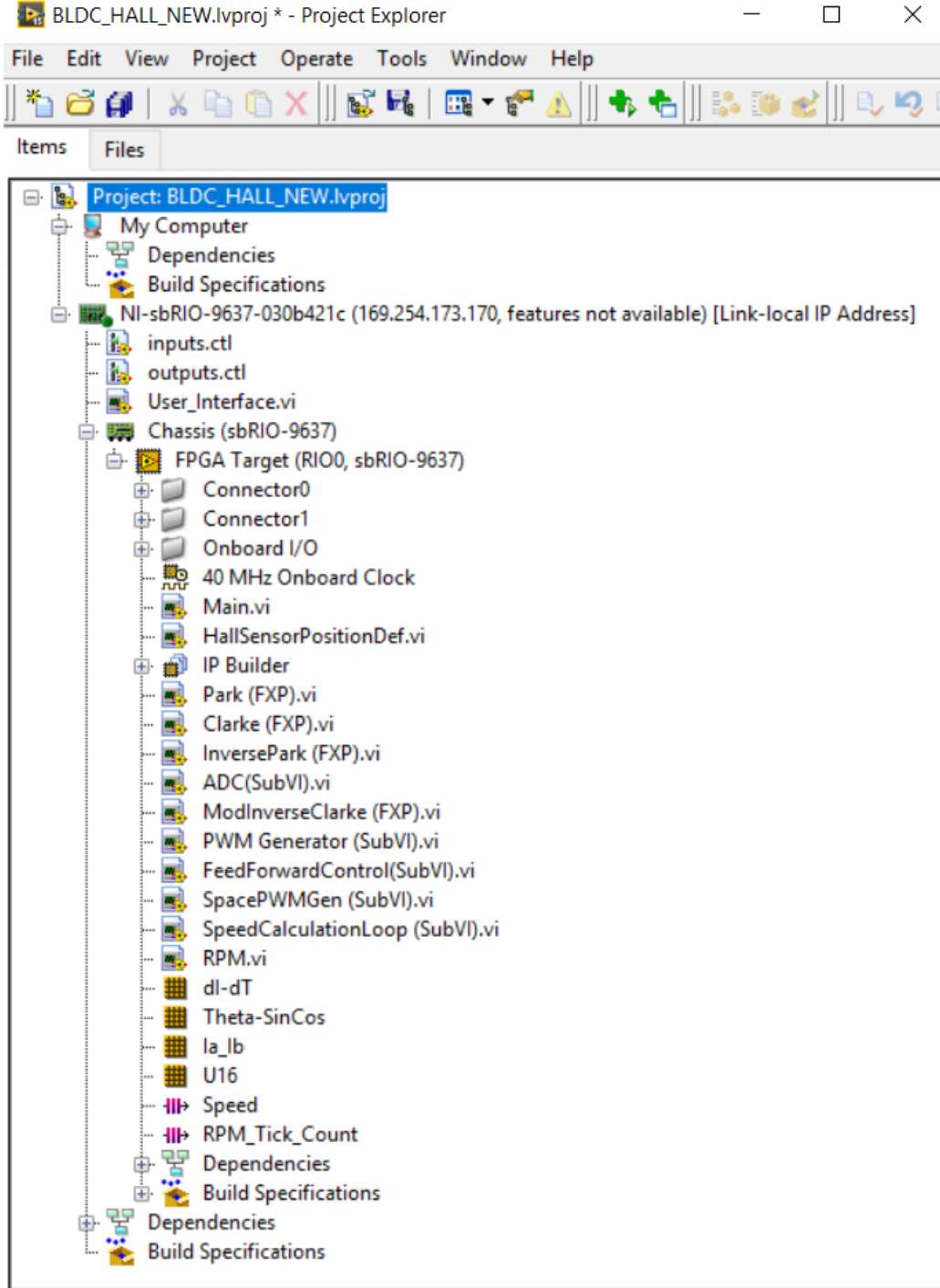
Şekil 4.2 sbRIO mimarsı [26]

FPGA, alanda programlanabilir kapı dizileri LabVIEW RIO mimarisinin çekirdeğini oluşturur. Kritik ve yoğun görevleri işlemciden almak ve son derece hızlı çıktı vermek için kullanılır. FPGA, yüksek frekanslı sinyal ve görüntü işleme, özelleştirilebilir zamanlama ve senkronizasyon için doğrudan Giriş/Çıkış terminallerine bağlıdır. Doğrudan G/Ç terminaline bağlı olduğundan neredeyse sıfır kontrol döngü gecikmesi ile, kontrol algoritmaları için gerekli yüksek performansı sağlamaktadır. Örneğin, FPGA kullanılarak tek bir sbRIO ile 100 KHz hızında aynı anda oransal, integral ve türev (PID) kontrol döngüsünü yürütebilir.

Giriş / Çıkış, sbRIO serisi g/ç ünitesi dahili yalıtım, dönüştürme devresi, sinyal denetimi ve endüstriyel algılayıcılar, kameralar, motorlar, sürücüler vb. endüstriyel iletişim protokollerine doğrudan bağlantı sunmaktadır. Ayrıca modül geliştirme seti ile birlikte uygulamaya özel ihtiyaçları karşılamak için özel modüller geliştirebilir ve g/ç ünitesine doğrudan bağlanabilir bir yapı sunmaktadır.

4.2.2. LabVIEW FPGA Modül Proje Ekranına Genel Bakış

Proje, iki farklı platformda (PC ve sbRIO) çalışan, üç ana VI'dan ve eş zamansız çalışan üç farklı işlem (PC, RTOS ve FPGA) katmanından oluşmaktadır. Şekil 4.3'te projeye ait görünüm verilmiştir.



Şekil 4.3 Proje ekranı genel görünümü

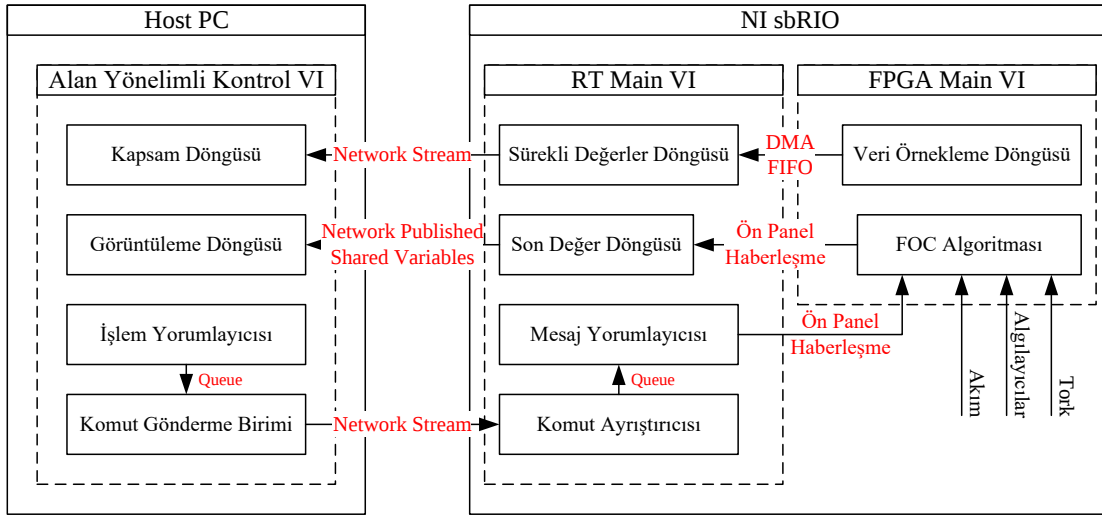
Burada birinci kısım olan ana bilgisayar (My Computer), kullanıcı arayüzünün çalıştırıldığı kısımdır. Kullanıcı ile sistem arasındaki veri giriş-çıkışını sağlar. Projede FOC algoritması, FPGA içerisinde çalışmaktadır. Diğer katmanlarda da program çalışmakta; ama kullanıcı bu katmanlara doğrudan erişim yapamamaktadır. Bu katmanlar kullanıcı ara yüzünden gelen komutları takip eder ve çalıştırır.

İkinci kısım ise RTOS (Real Time Operating System, NI-sbRIO-9637-030b421c (160.224.173.170) [Link Local IP Address]) katmanıdır. Burada proje uzantılar RT main VI olarak adlandırılır. Bu katmanın projedeki işlevi ana bilgisayardan gelen komutları FPGA hedefine göndermek ve bazı veri işlemleri yapmaktır. RTOS kesme (interrupt) tabanlı olmadığı için belirlenen sürede verilerin alınması, işlenmesi ve gönderilmesi ve ekstra gecikmelerin olmaması için yeterince hızlı bir işlemciye sahip olması gereklidir. İşlemci hızı yavaş olursa windows tabanlı bir makinada kesintisiz olarak belli bir oranda çalışması gereken işlemler için sorun olabilir. RTOS çalışma anında ana bilgisayarda anti virüs programları gibi programların çalışması RTOS çalışmasını yavaşlatabilir. Bu nedenle RTOS ile birlikte farklı programların çalıştırılmaması gerekmektedir.

Üçüncü işlem katmanı ise üst düzey VI ların çalıştırıldığı FPGA (FPGA Target) hedefidir. Motor kontrol algoritması ile birlikte gerekli tüm ölçümler ve hesaplamalar FPGA hedefinde gerçekleştirilmektedir. 40 MHz lik bir saat frekansına sahiptir. Bu FOC gibi yüksek işlem hızı gerektiren dinamik motor kontrol algoritması için ideal bir hızdır. Gerekli olduğu takdirde saat hızı sağ tıklanarak değiştirilebilir ve yeni bir saat hızı elde edilir. Örneğin 40 MHz lik bir saat frekansından 2.5 MHz lik bir saat hızı elde etmek mümkündür. Giriş çıkış terminalleri sbRIO karta sabitlenmiş bir şekilde sunulmaktadır ve Şekil 4.3'te de görüldüğü gibi bu terminaller FPGA hedefi altında bağlantı 0 ve bağlantı1 (Connector0, Connector1) olarak yer almaktadır.

4.2.3. Katmanlar Arası Haberleşme Mimarisi

Üç farklı işlem katmanı üzerinde program geliştirirken, projede yer alan tüm görev ve iletişim fonksiyonları göz önünde bulundurularak, hangi işlevin hangi katmanda yapılacağından önceden belirlenmesi önemlidir. Şekil 4.4'te sistemin katmanlarına ait mimari verilmiştir.



Şekil 4.4 Katmanlar arası haberleşme blok diyagramı

4.2.4. Kullanıcı Bilgisayarı (Host PC) ve FPGA Arası Veri Transferi

Aşağıda kullanıcı bilgisayarı (Host PC) ile FPGA arasındaki haberleşme yöntemleri sıralanmıştır.

- Doğrudan Hafıza Erişimi:** FPGA da kod bloğu çalışırken ana bilgisayarda programa ait ön panel penceresinin görüntülenmesi, sinyal üretimi ve veri kaydı için kullanılır.
- Programatik Ön Panel Haberleşmesi:** FPGA ile veri kontrolü izlenmesi ve aktarılması için harici bir bilgisayarda çalışan ayrı bir VI kullanılır.
- Kullanıcı Tanımlı G/Ç Değişkenleri (RT-Host VI Sadece):** Donanım aygıtları arası veri aktarımı için kullanılır.

Veri transfer yöntemleri arasında Tablo 4.1'deki gibi farklılıklar bulunmaktadır.

Tablo 4.1 Veri transfer yöntemlerinin karşılaştırılması

Data Transfer Yöntemi	İş Hacmi Oranı	Transfer Hızı	Host CPU Kullanımı	Zamanlama Modeli	Kullanım Yeri
Direct Memory Access	Yüksek	Yüksek	Düşük	Kullanıcı Tanımlı	Veri depolama veya sinyal üretme
Programatik Ön Panel Hab.	Düşük	Düşük	Yüksek	Kullanıcı Tanımlı	Kontrol, Benzetim
Kullanıcı Tanımlı Giriş/Çıkış Değişkenleri	Düşük	Düşük	Düşük	NI Tarama Motoru	Uyumlu veri kümelerinin aktarılmasını gerektiren uygulamalar

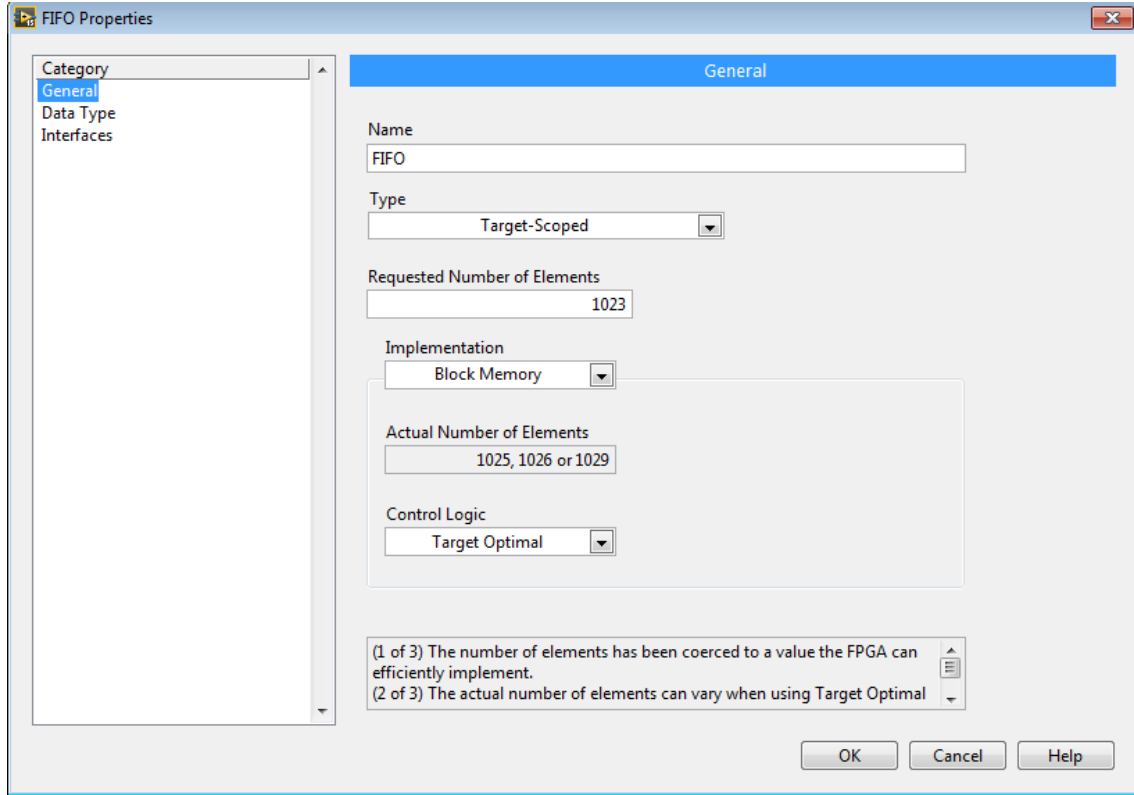
4.2.5. FPGA Hedeften Kullanıcı Bilgisayara Veri Transferi

Kullanıcı bilgisayarı ekranında grafikleri ve çizelgeleri görüntüleyebilmek için büyük miktarda verinin doğru ve kesintisiz bir şekilde gönderilmesi gerekmektedir. Burada ön panel haberleşmesi kullanılamaz; çünkü veri alma oranını kontrol etmek mümkün değildir. Bu nedenle “Streams” (sürekli veri akışı veya akışları) kullanılmalıdır. FPGA üzerinden veri alımı ve alınan verinin kullanıcı bilgisayarında bulunan grafik ve çizelgelerde gösterilmesi için birkaç farklı adım uygulanmalıdır. Bunlardan birincisi, FPGA üzerinde bir döngü ve bu döngü içerisinde DMA FIFO yapısını kullanmaktır. DMA FIFO daki veriler RTOS da bulunan bir döngü içerisinde okunur ve *Network Stream (Ağ Akışı)* ile kullanıcı bilgisayara gönderilir. *Network Stream* den gelen veriyi okumak ve göstermek için kullanıcı bilgisayarında bir döngü bulunur ve alınan veriler anlık olarak gösterilir.

4.2.6. DMA FIFO Yapısı

Projeye DMA FIFO eklemek için proje ekranında FPGA Hedefe sağ tıklanarak “New” ve sonra “FIFO” seçilir. Böylelikle seçilen pencerede “FIFO özellikleri” penceresi aktif olur. Burada istenilen iletişim yönüne bağlı olarak FIFO türü (“Host To Target-DMA” veya “Target To Host-DMA”) olarak ayarlanır. FIFO türü seçildikten sonra eleman

sayısı ve veri türü de belirlenmelidir. Şekil 4.5 DMA-FIFO oluşturma ekranını göstermektedir.



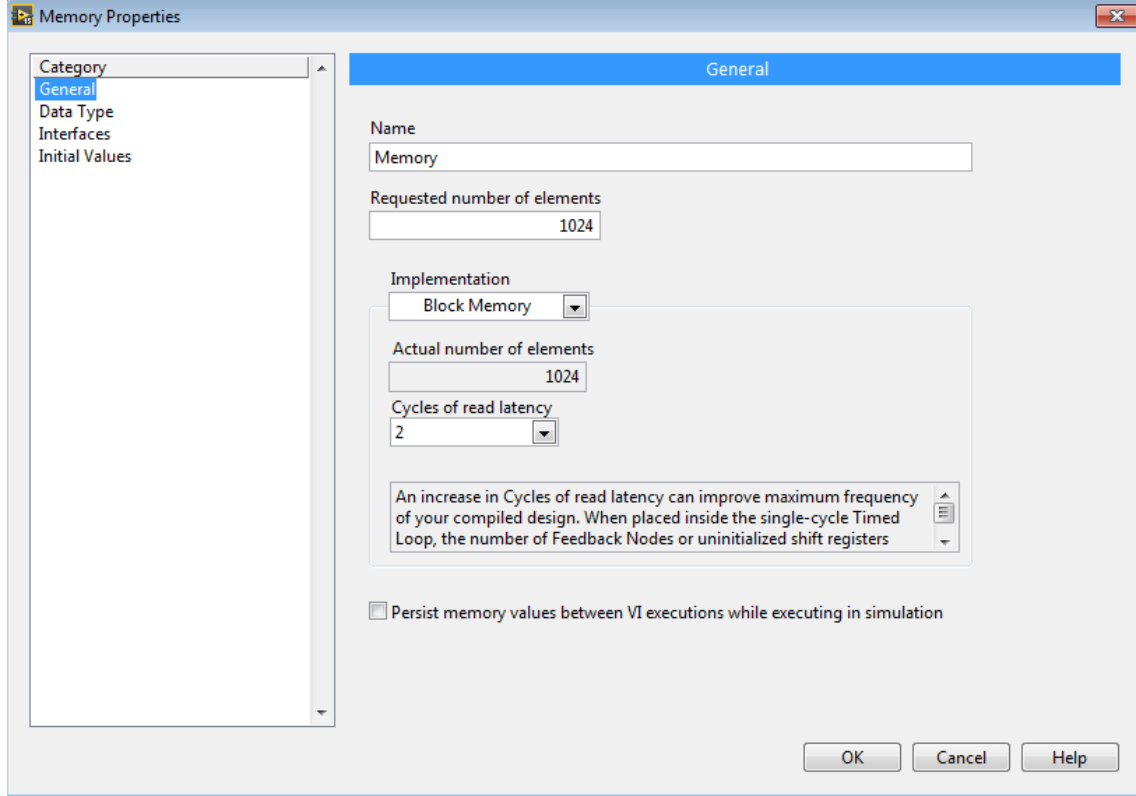
Şekil 4.5 DMA FIFO özellik penceresi

FIFO projeye eklendikten sonra veri elde edilebilir ve FIFO'ya yazılabilir. Ön panel kontrollerine ait yerel değişkenler kullanılarak, FPGA da bulunan veri örnekleme döngüsünden FIFO daki veriler alınır.

4.2.7. Blok Hafıza (RAM) Yapısı

Blok rastgele erişimli bellek olarak ta bilinen “Blok RAM” veya BRAM, veri depolamak için dâhili bir FPGA kaynağıdır. Blok RAM belleği kullanan hafıza ögeleri diğer bellek ögelerine göre daha yüksek bir saat hızında derlenir. Bu bellek tipi okuma-yazma erişimi veya çift bağlantı noktası okuma erişimi için yapılandırılabilir. Ayrıca veriyi farklı bir saat bölgesinden okumak ve farklı bir saat bölgesinde yazmak için blok hafıza ögeleri kullanılabilir. Bu uygulamada her bellek ögesi için bir yazıcı düğümü ve bir okuyucu düğümü kullanılmıştır. Bu tip hafıza ögeleri FPGA ya ait kaynakları tüketmediğinden, diğer bellek türlerine (Handshake, Register, FIFO vb.) ihtiyaç

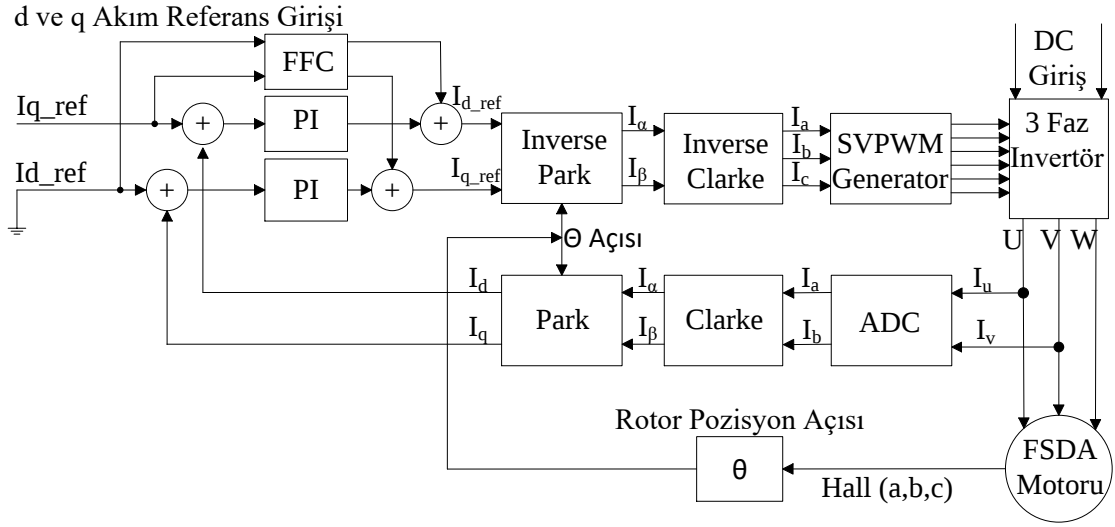
duyulmadığı müddetçe blok hafıza ögesinin kullanımı tercih edilmelidir. Bir projeye blok hafıza ögesi eklemek için DMA-FIFO’da da olduğu gibi FPGA hedefine sağ tıklanarak yeni penceresinden “*memory*” seçilir ve açılan pencerede istenilen özelliklerde Blok RAM belleği oluşturulur. Şekil 4.6 blok hafıza ögesinin yapılandırılmasına ait pencereyi göstermektedir.



Şekil 4.6 Blok RAM özellik penceresi

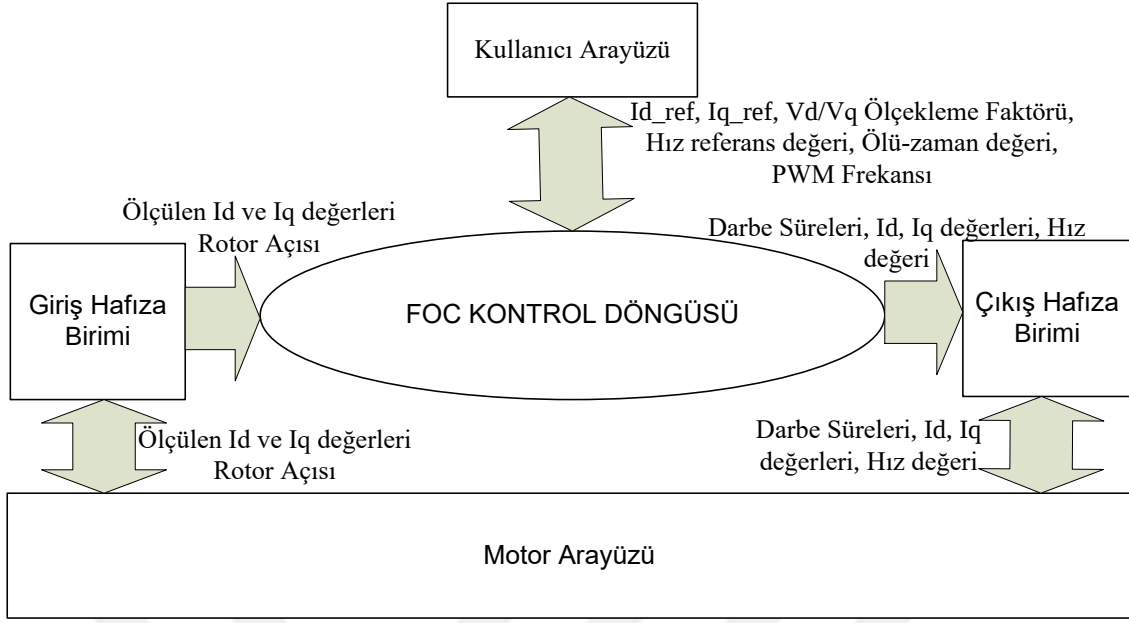
4.3. FPGA İçerisinde Yazılım Tasarımı

Şekil 4.7 FPGA’da koşturulacak FOC algoritmasına ait blok diyagram ve bu diyagramda kullanılan SVPWM birlikte verilmiştir.



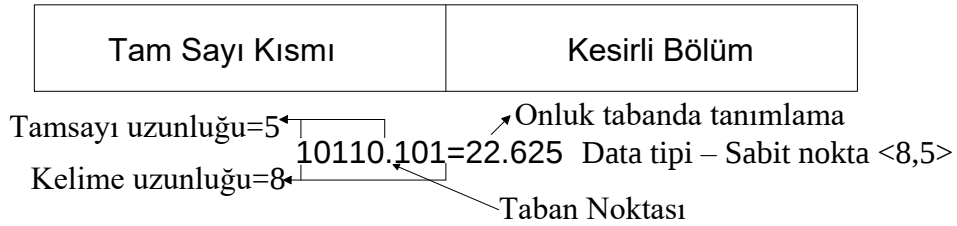
Şekil 4.7 FOC kontrol yapısı [21]

Bütün sistem, bölüm 4.3.1 ile 4.3.10 arasında açıklanan bileşenlerin kullanılmasıyla FPGA içerisinde tasarlanmıştır. Sistemin kontrolü ve parametre değişimi ise “*Network Published Shared Variables*” ve “*Network Stream*” kullanılarak Host PC de gerçekleştirilmiştir. FPGA içerisinde belirtilen bileşenler, LabVIEW blok diyagramında paralel olarak yürütülen birkaç döngü olacak şekilde tasarlanmıştır. Bu paralel döngüler arasındaki iletişim blok RAM belleğinde bulunan özel bellek alanları tarafından gerçekleşmektedir. Hızlı bir değerlendirme için kullanıcının denetleyici ile etkileşime girmesi önemlidir. Bu yüzden PI ve ileri beslemeli (Feed Forward) kontrolör gibi kullanıcı verilerinin girildiği ve saklandığı bir kullanıcı veri hafızası alanı eklenmiştir. Ayrıca I_d , I_q , hız, rpm gibi verilerin izlenmesi için görüntüleme alanı da eklenmiştir. Şekil 4.8 sistem tasarımına ait blok diyagramı göstermektedir.



Şekil 4.8 Sistem tasarımı

Sistemdeki tüm matematiksel hesaplamalar FPGA karta ait sabit nokta (Fixed Point Representation) aritmetiği kullanılarak gerçekleştirilmiştir. Bu sabit nokta veri tipi, veriyi temsil etmek için gerekli doğruluğu sağlamak üzere yapılandırılabilir. Yani Şekil 4.9 da görüldüğü gibi Integer ve Fraction kısımlarından oluşan toplam veri bit sayısı ve tam sayı (Integer) bit sayısı değiştirilebilir. Değişkenlerin her biri için sabit nokta veri gösterimi, Ekler kısmında sunulmuştur.

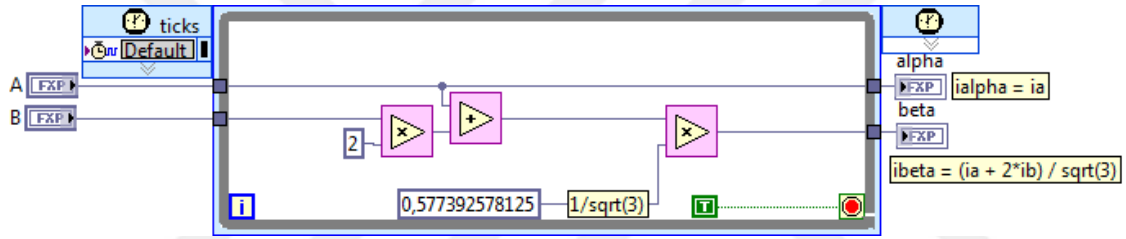


Şekil 4.9 Sabit Nokta Aritmetiği hesaplaması [10]

Şekil 4.9’da da görüldüğü gibi verilen sayının kelime uzunluğu (Word length) 8, tam sayı uzunluğu (integer length) 5 tir. Onluk sayı sistemi olarak sonuç ise; tam sayı için $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 = 0 + 2 + 4 + 0 + 16 = 22$, ondalık kısmı için; $1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 0.5 + 0 + 0.125 = 0.625$ ve genel olarak verilen sayının değeri $22 + 0.625 = 22.625$ olmaktadır.

4.3.1. Clarke Dönüşümü

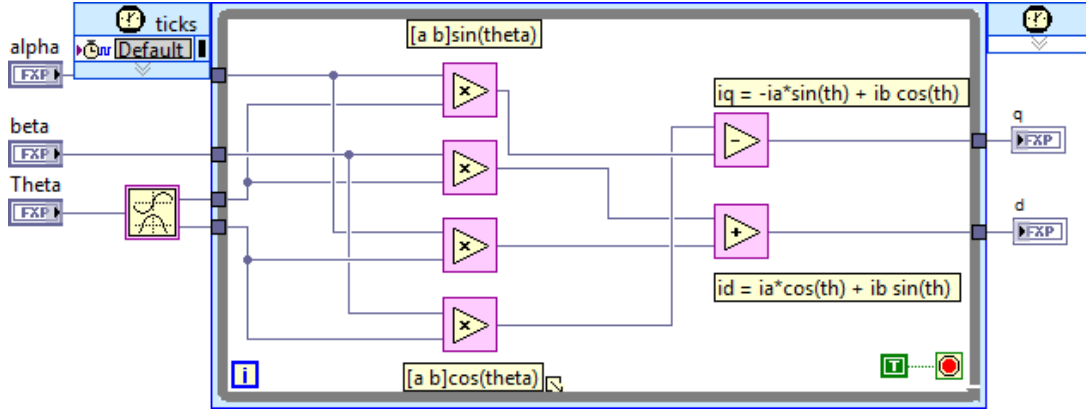
Clarke dönüşümüne ait formüller, Bölüm 2 de 18 ve 19 numaralı eşitliklerde daha önceden verilmiştir. LabVIEW nesneleri kullanılarak oluşturulan bu eşitliklere ait işlemler, Şekil 4.10'da görülen blok diyagramda gerçekleştirilmiştir. Burada görülen “A” ve “B” değerleri, eşitliklerdeki u ve v sargılarından ölçülen faz akımları I_a ve I_b değerlerini ifade etmektedir. Clarke dönüşümünde büyüklük değişmez, yani α, β düzlemine dönüştürülen akım değerlerinin genlikleri korunur. Burada dikkat edilmesi gereken nokta; döngüler FPGA içerisinde koşacağından ve hesaplama işlemleri de 1 tick (FPGA klok frekansının zaman değeri) sürede gerçekleşmesi gerektiğinden, döngü parametresi olarak *Timed Loop* seçilmelidir. Tick kavramı 4.4 nolu başlık altında ayrıntılı bir şekilde verilmiştir.



Şekil 4.10 Clarke dönüşümü

4.3.2. Park Dönüşümü

Park dönüşümü ise Clarke dönüşümünden elde edilen α (bakınız Şekil 4.10 da alpha) ve β (bakınız Şekil 4.10 da beta) değerlerini, iki boyutlu d, q dönel referans düzlemine geçirmek için kullanılır. Burada Clarke dönüşümünden elde edilen I_α, I_β ve motorun elektrik açısının (bakınız Şekil 4.11 de Theta) sin ve cos bileşenleri kullanılarak I_d, I_q elde edilmiştir. Bölüm 2 de daha önceden verilmiş Eşitlik 2.26 ve 2.27 ya ait hesaplamalar, Şekil 4.11'de görülen blok diyagramda gerçekleştirilmiştir.



Şekil 4.11 Park dönüşümü

4.3.3. Tez Çalışmasında Kullanılan Kontrolörler

Park dönüşümünden elde edilen I_d, I_q bileşenlerini kontrol etmek için PI ve İleri besleme kontrol olmak üzere iki tip kontrol yöntemi kullanılmıştır. İleri yönlü kontrol yöntemi motor sürekli durumda çalışırken PI denetleyicilerinin yükünü hafifletmek için kullanılır. PI denetleyicisi, ileri yönlü kontrol algoritması tarafından üretilen hatayı düzenlemek için ve hız veya yükteki ani artışlarla bağlantılı geçişleri kontrol etmek için tasarlanmıştır. PI ve ileri yönlü kontrol algoritmaları bu işlevselliği denetlemek için paralel kullanılmalıdır. Bu tez çalışmasında da her iki yöntemin paralel çalışması gerçekleştirilmiştir.

4.3.3.1. PI Denetleyicisi

Bu tezde Park dönüşümünden elde edilen I_d ve I_q akımlarının kontrol edilmesi için toplamda iki adet PI denetleyici kullanılmıştır. Bunun için FPGA içerisinde bulunan dâhili PID bloğundan yararlanılmıştır. Denetleyici çıkışı aşağıdaki eşitlikliklerin toplamı olarak ifade edilmiştir.

PID denetleyici t uzayında;

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{e(t)}{dt} \quad (4.1)$$

Bu tez çalışmasında türevsel kazanç kullanılmamıştır. Bu durumda eşitlik tekrardan yazılırsa;

$$u(t) = K_p e(t) + K_i \int e(t) dt \quad (4.2)$$

Burada;

$u(t)$: Etkiyen işaret

$e(t)$: Hata işareti

K_p : Oransal kazanç

K_i : İntegral kazancı

K_d : Türevsel kazanç

PID denetleyici yazılımsal formda ise;

$$U_{p(k)} = K_p e_{(k)} \quad (4.3)$$

$$U_{i(k)} = K_p K_i \sum_{i=1}^k \left[\frac{e_{(i)} + e_{(i-1)}}{2} \right] \quad (4.4)$$

$$U_{d(k)} = -K_p K_d [PV_{(k)} - PV_{(k-1)}] \quad (4.5)$$

Şeklinde yazılır.

Burada;

$e(k)$: Başlangıç noktası $(k) - PV(k)$

$PV_{(k)}$: k. tekrarlama işlem değişkeni değeri

Uygulamada türevsel denetleyici olmadığından çıkış işareti, $U_{p(k)}$ ve $U_{i(k)}$ değerlerinin toplamına eşittir.

4.3.3.2. İleri Beslemeli Kontrol (Feed Forward Control) Algoritması

İleri Beslemeli Kontrol (FFC), bir açık çevrim kontrol yöntemidir ve sistemde herhangi bir geri besleme kullanılmaz. İleri yönlü kontrol denetleyicisi, Eşitlik. 4.6 ve Eşitlik. 4.7 ile tanımlanan motordaki elektrik dinamiklerinin bir modelini temsil etmektedir.

$$U_{sd} = R_s i_{sd} + L_d \frac{di_{sd}}{dt} - \omega_s L_{sq} i_{sq} \quad (4.6)$$

$$U_{sq} = (L_s di_{sd} + \lambda) \omega_s + L_{sq} \frac{di_{sq}}{dt} + R_s i_{sq} \quad (4.7)$$

Burada;

R_s : Motor hat direnci (Ω)

L_{sq}, L_{sd} : Stator endüktansları (H)

i_{sq} : Motor tork referans değeri (N.m)

i_{sd} : Motor Akı referans değeri (Φ)

ω_s : Motorun elektriksel hızı (rad/sn)

λ : Motorun manyetik akı bağlantısı- magnetic flux linkage, (N. Φ)

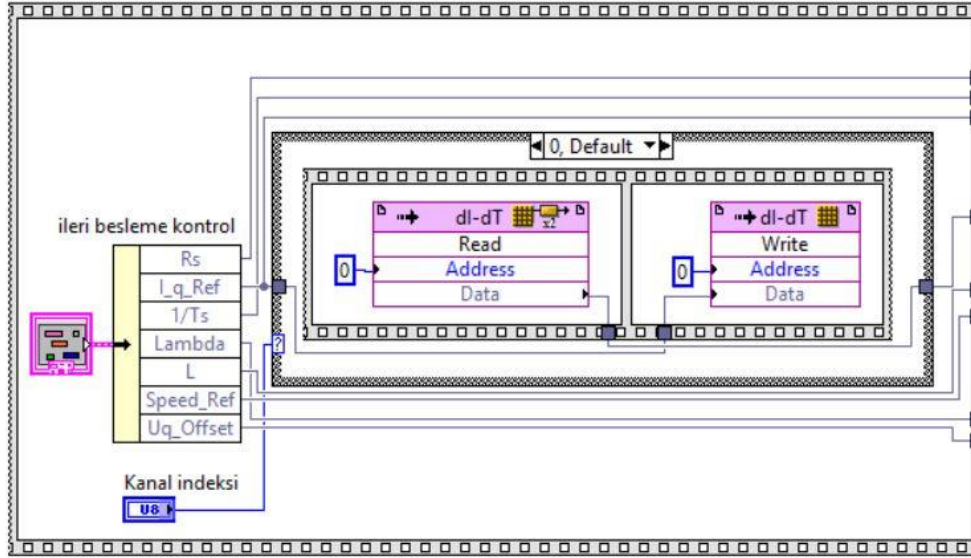
Bu tez çalışmasında kullanılan motorun devri, nominal değerin üzerine çıkmadığından, alan zayıflatması gerekmemektedir. Yani motor akı referans değeri her zaman sıfır olacaktır. Dolayısıyla i_{sd} değeri yerine 0 yazılarak, Eşitlik 4.6 ve Eşitlik 4.7 yeniden düzenlenmiş ve Eşitlik 4.8 ve 4.9 elde edilmiştir.

$$U_{sd\ ref} = \omega_s L_{sq} i_{sq} \quad (4.8)$$

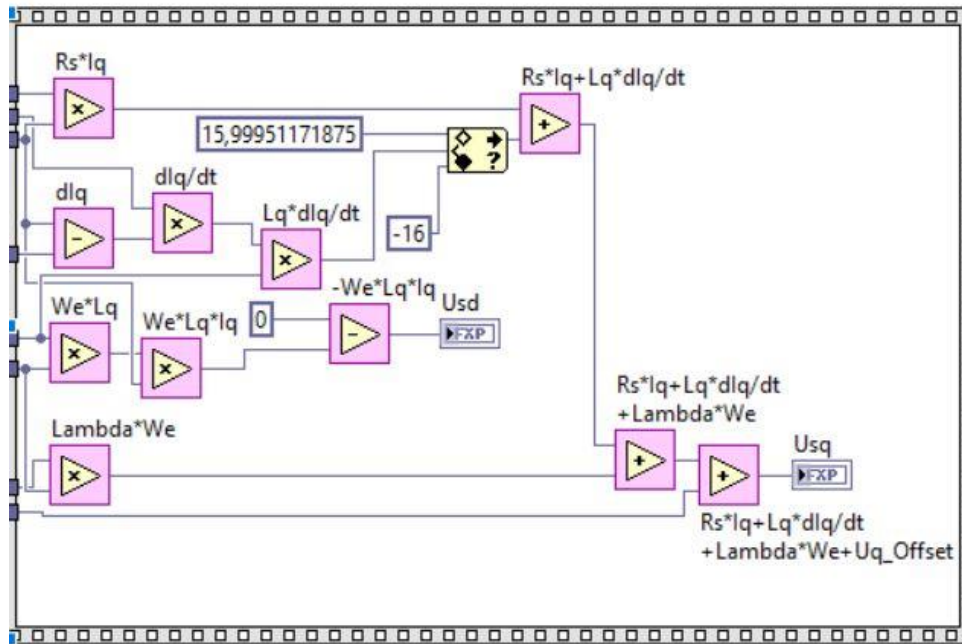
$$U_{sq} = \lambda \omega_s + L_{sq} \frac{di_{sq}}{dt} + R_s i_{sq} \quad (4.9)$$

FFC Algoritmasının LabVIEW uygulaması, Şekil 4.12’de verilmiştir. Şekil 4.12 a’da bu açık çevrim kontrolörün kullanıcı arayüzü üzerinden girilen parametreleri; Şekil 4.12 b’de ise Eşitlik 4.8 ve 4.9’da verilen diferansiyel denklemlerin çözümünü göstermektedir. Denetleyici çıkışında elde edilen U_{sd} ve U_{sq} değerlerinin hesaplanması için döngünün önceki tekrarlamasından elde edilen i_{sq} değeri, türev terimi için bir blok ram belleğinden okunur. Algoritmada verilerin sıralı bir şekilde işlenmesi için FPGA’ya

ait sıralı yapı (Sequence Structure) kullanılmış; önce RAM bellekten i_{sq} değeri okunmuş ardından da U_{sd} ve U_{sq} denetleyici çıkışları elde edilmiştir.



a)



b)

Şekil 4.12 FFC algoritması LabVIEW uygulaması blok diyagram görüntüsü a) Motor parametrelerinin girilmesi, b) Eşitlik 4.6 ve 4.7 ye ait hesaplamalar [30]

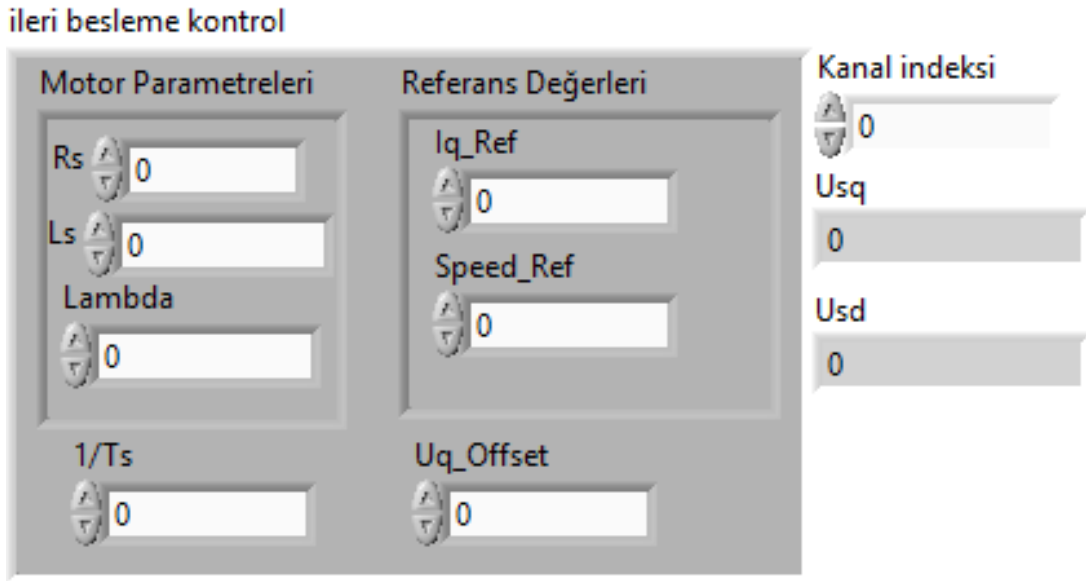
FFC algoritması LabVIEW uygulamasında farklı I_{q_ref} değerleri için yapılan deney sonuçları Tablo 4.2 de verilmiştir. Bu deneylerde $L_d=L_q=1.3508e-3H$, $R=0.08\Omega$, $\lambda=0.0446 N\Phi$, $K_e=1.36524 Vs/rad$, $V_d/V_q= 0.0721687$ olarak alınmıştır.

Tablo 4.2 FFC Algoritmasına ait LabVIEW Uygulama Sonuçları

Iq_ref (A)	Speed_ref (rpm)	Iq ölçülen (A)
1	20	0.121882
	40	0.22561
	60	0.335603
	80	0.4455
	100	0.5554
	120	0.6654
2	20	0.1217
	40	0.2318
	60	0.342026
	80	0.4521
	100	0.561261
	120	0.671207
3	20	0.1274
	40	0.2376
	60	0.3478
	80	0.4579
	100	0.5681
	120	0.6782
4	20	0.1327
	40	0.2434
	60	0.3535
	80	0.4637
	100	0.5738
	120	0.684
5	20	0.139
	40	0.2492
	60	0.3593
	80	0.4695
	100	0.5796
	120	0.6898

4.3.3.3. FFC Parametrelerinin Çıkarımı

Uygulamada kullanılacak motora ait parametreler, Şekil 4.13 de görülen arayüz üzerinden ileri beslemeli kontrol algoritmasına girilmektedir. Parametre çıkarımında kullanılacak veriler, motora ait veri sayfalarından elde edilmiştir[25]. R_s ve L_s parametreleri için hat (line to line) değerleri kullanılmalıdır. λ değeri ise, 4.10 nolu denklem ile elde edilmektedir. Burada anma akım ve gerilim değerleri, maksimum değerleri temsil etmektedir. İleri yönlü kontrol algoritmasında hız referans değeri (speed_ref) ise elektriksel açısal hızı temsil etmektedir.



Şekil 4.13 İleri beslemeli kontrol algoritması için motor parametrelerinin girildiği ön panel görüntüsü

$$\lambda = \frac{Rated_Torque}{np \cdot \sqrt{2} \cdot Rated_Current_{(rms)}} \quad (4.10)$$

Bununla birlikte eğer mekanik açısal hız belli ise Eşitlik 4.11 kullanılarak elektriksel açısal hız da hesaplanabilir.

$$\omega_e = \omega_m \cdot np \quad (4.11)$$

Burada;

np : Kutup sayısını ifade etmektedir.

İleri beslemeli kontrol algoritması Matlab benzetimi ise Bölüm 5.2.2’de ele alınmıştır. Ayrıca FFC algoritması LabVIEW uygulaması ve Matlab benzetimine ait deney sonuçlarının karşılaştırılması ve değerlendirilmesi, Bölüm 6’da ayrıntılı bir şekilde verilmiştir.

4.3.3.4. FFC Algoritmasında U_{sq} ’nin Türevsel Terimi

İleri besleme kontrol algoritmasının 1.derece kısmi türevi, $L_{sq} \frac{di_{sq}}{dt}$ dir. Eğer algoritma birden fazla yükü kontrol edecekse i_{sq} ’nin önceki değeri güvenli bir şekilde tutulmalıdır. Burada, farklı yük seçimi ve yüke ait parametrelerin girilmesine imkan tanıyan çok durumlu (case) bir yapı oluşturulmuştur. “Kanal indeksi” ismi ile anılan bir durum seçici girişi tanımlanmıştır. Mevcut uygulamada sadece “0,Default” seçeneği kullanılmıştır. Ancak ileride farklı yüklerin kullanılması durumunda farklı durumların eklenmesine imkan tanıyan esnek bir yapı oluşturulmuştur. Güvenli bir depolama sağlamak için, bir blok ram belleği ile birlikte bir sıralı yapı kullanılmıştır. Her durumda i_{sq} ’nin bir önceki değeri, bellekte tanımlanan kanala ait seçilen motor verisini istenilen adresten alır. Alınan i_{sq} değeri toplandıktan sonra bir sonraki geçerli i_{sq} değeri aynı bellek adresine yazılır. i_{sq} Akımının türevini hesaplamak için önceki değer, mevcut değerden çıkartılır ve sonuç $\frac{1}{T_s}$ ile çarpılır (bakınız Şekil 4.11). Böylelikle $\frac{\Delta i_{sq}}{\Delta t}$ ye karşılık gelen değer elde edilir. Eğer bir önceki i_{sq} değeri ile güncel i_{sq} değeri arasındaki fark, -16 ile +16 Amper aralığının dışında ise probleme neden olmaktadır. Fark değeri bu aralığın dışında olduğunda türev değeri de bu aralığın dışında olacağından; sabit nokta gösteriminde sayısal sorunlara neden olmaktadır.

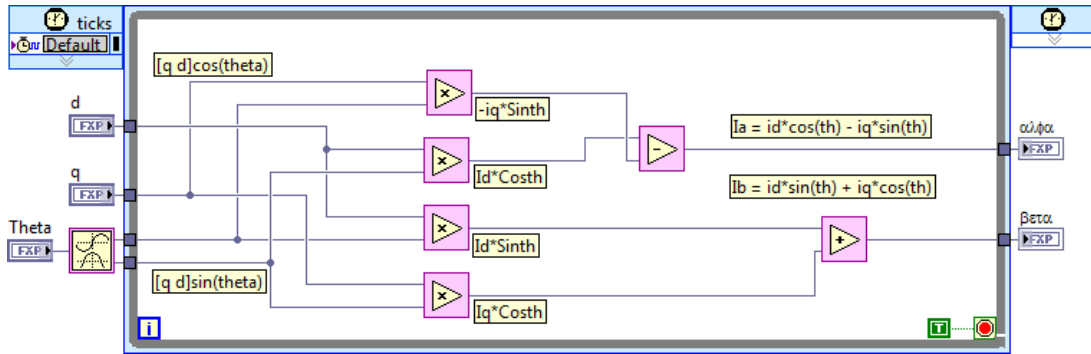
Bu tezde, ileri beslemeli kontrol algoritması PI denetleyici ile paralel çalışarak PI denetleyicisinin farklı sürme koşullarındaki dayanıklılığını iyileştirmek amacı ile kullanılmıştır. Diğer taraftan FFC algoritması iyileştirme destekleyicisi olarak değil de sistemin ana bileşeni olarak ta düşünülebilir. Bu durumda bu yöntem hız, kontrol ve yol düzenleyicisi olarak ele alınabilir.

Dayanıklılığı iyileştirmek amacı ile kullanılan FFC algoritmasında, dış kontrol döngüleri bir tork değerine göre i_{sq} değerini yavaş yavaş azaltır veya artırır. Yani i_{sq} değerinde büyük bir sıçramanın olmasına izin vermez. Referans değeri kademeli olarak arttığında, türev terimi herhangi bir probleme neden olmaz. Sistemi dış döngüler olmadan test ederken oluşacak problemleri önlemek amacı ile türev teriminin çıkışını, -16 ile +16 Amper arasında tutacak bir doygunluk fonksiyonu, Şekil 4.11 de görülen blok diyagrama eklenmiştir. Böylece türev terimini, $R_s i_{sq}$ değerine eklerken ondalık kısımda iyi bir çözünürlük sağlanarak sabit nokta gösteriminde tam sayı bölümünde oluşabilecek taşmalar engellenmiştir.

4.3.4. Ters Park Dönüşümü

PI denetleyicisinden elde edilen düzenlenmiş akım veya gerilim değerlerini tekrar statik α, β referans düzlemine geri döndürmek için ters park dönüşümünü kullanılmaktadır. Kullanılan sin ve cos değerleri Park dönüşümündeki değerler ile aynı olmalıdır. Şekil 4.14 ve Eşitlik 2.27 denkleminin uygulamasını göstermektedir.

Burada, $I_d, I_q, \sin, \cos \rightarrow I_\alpha, I_\beta$ şeklinde dönüşüm gerçekleştirilmektedir.



Şekil 4.14 Ters Park dönüşümü blok diyagramı

4.3.5. Ters Clarke Dönüşümü

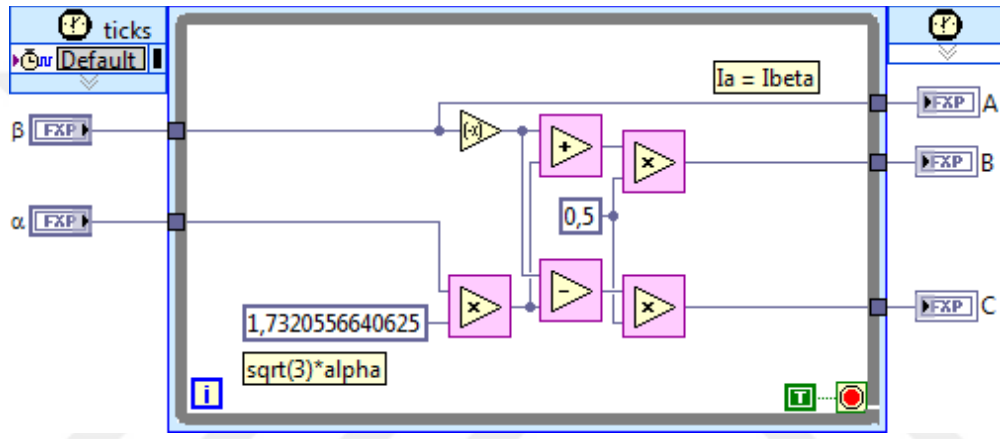
Bu tez çalışmasında modifiye edilmiş ters Clarke dönüşümü kullanılmıştır. Yani Eşitlik 2.21 de I_α ve I_β bileenleri ters çevrilmiştir. Bu durum, referans vektörünün saat yönünde 90° dönmesine neden olur. Bu dönüşüm kullanılarak SVPWM üretimi daha basit hale gelir. Clarke dönüşümünde olduğu gibi verilerin genliklerinde değişme olmaz. Şekil

4.15 ve Eşitlik. 4.12, 4.13, 4.14 modifiye edilmiş ters Clarke dönüşümünü göstermektedir.

$$I_a = I_\beta \quad (4.12)$$

$$I_b = \frac{-I_\alpha + \sqrt{3}I_\beta}{2} \quad (4.13)$$

$$I_c = -I_a - I_b = \frac{-I_\alpha - \sqrt{3}I_\beta}{2} \quad (4.14)$$



Şekil 4.15 Ters Clarke dönüşümü blok diyagramı

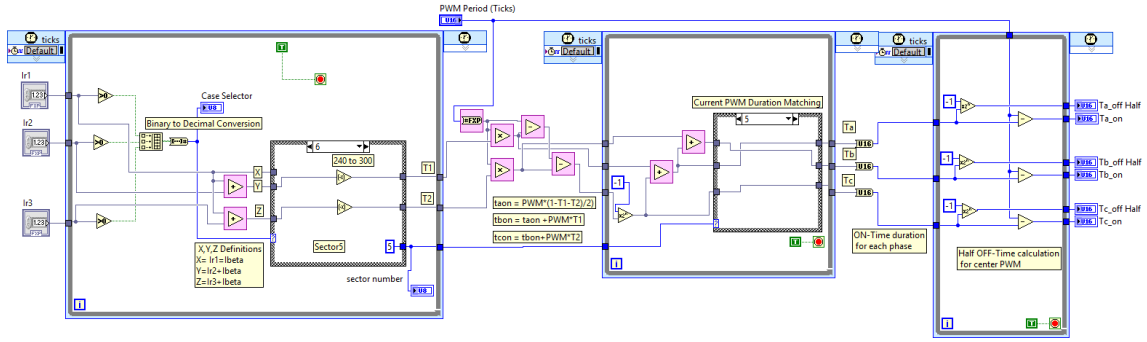
4.3.6. Uzak Vektör PWM Üretici

Uzak vektör PWM üretici, gerilim vektörünün hangi sektörde bulunduğunu belirtmek için Ters Clarke Dönüşümünün çıkışını kullanmaktadır. Sektörlerin belirlenmesi için I_a, I_b, I_c olarak ifade edilen I_{r1}, I_{r2} ve I_{r3} referansları kullanılır. Değer 0'dan büyük ise ilgili bit 3 bitlik sayı (101 gibi) ile ifade edilir. Belirlenen bu sayı daha sonra lojik uzayında hangi sektörde vektörün olduğunu bilmemizi sağlar. Sektöre bağlı olarak PWM zamanları T_1 ve T_2 , X, Y ve Z den türetilir ve Eşitlik 4.15, 4.16, 4.17 ve Şekil 4.16'ya göre hesaplanır.

$$X = I_{r1} \quad (4.15)$$

$$Y = I_{r1} + I_{r2} \quad (4.16)$$

$$Z = I_{r1} + I_{r3} \quad (4.17)$$



Şekil 4.16 Uzay vektör modülasyon üretici [30]

Üç faz için döngü çevrim süreleri (t_{aon} , t_{bon} , t_{con}) T_1 ve T_2 , PWM periyodunun karşılık gelen oranına ölçeklendirilir ve Eşitlik 4.18, 4.19 ve 4.20 ile hesaplanır.

$$t_{aon} = \frac{T_{PWM} - T_1 T_{PWM} - T_2 T_{PWM}}{2} \quad (4.18)$$

$$t_{bon} = \frac{T_{PWM} + T_1 T_{PWM} - T_2 T_{PWM}}{2} = t_{aon} + T_1 T_{PWM} \quad (4.19)$$

$$t_{con} = \frac{T_{PWM} + T_1 T_{PWM} + T_2 T_{PWM}}{2} = t_{bon} + T_2 T_{PWM} \quad (4.20)$$

Geçerli PWM sürelerinin belirlenmesi için fazlara karşılık gelen darbe süreleri durum yapısı (Case Structure) içeriği sıralı bir şekilde Tablo 4.2 de gösterilmiştir.

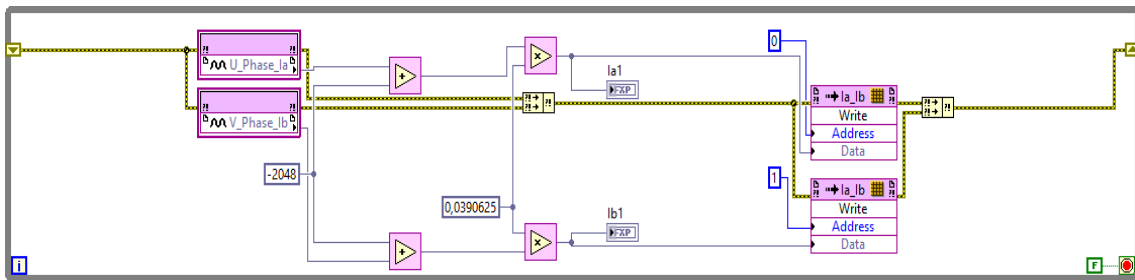
Fazların döngü çevrim süreleri belirlendikten sonra vektörlerin hangi sektöre yerleştiğine bağlı olarak değerler doğru faza uygulanır. SVPWM'in işlevselliğini doğrulamak için ters clark dönüşümü ile birlikte kullanılmalıdır. Her bir temel vektör için α, β koordinatlarını manuel olarak girip üreticinin çıkışını izleyerek doğru PWM modellerini oluşturabiliriz. Uygun PWM modellerinin sağlanması için sektör çıkışları ile Şekil 2.16 daki modellerin birbirini tutması gerekmektedir.

Tablo 4.3 Durum yapısı sıralı içeriği ve uygulanması

Ir1>0	Ir2>0	Ir3>0	Durum Seçici (Case Selector)	Sektör Numarası T ₁ ve T ₂ cinsinden sürelerin hesaplanması	Sektöre göre darbe sürelerinin Eşitleştirilmesi
				t _{aon} , t _{bon} , t _{con}	
0	0	0	0	off	0
0	0	1	1	2	T _c , T _b , T _a
0	1	0	2	6	T _c , T _a , T _b
0	1	1	3	1	T _a , T _c , T _b
1	0	0	4	4	T _a , T _b , T _c
1	0	1	5	3	T _b , T _a , T _c
1	1	0	6	5	T _b , T _c , T _a
1	1	1	7	off	0

4.3.7. Motor Akımlarının Analog Giriş Terminalinden Okunması

Motor hızının denetlenmesi için, motor akımlarının ölçülmesi gerekmektedir. Herhangi iki fazdan (u, v, w) geçen anlık akım değerleri, FPGA kartın analog giriş terminalinden (AI0, AI1) okunur ve 16 bitlik bir sayısal veriye dönüştürülür. Daha sonra bu sayı -16 ile +16 Amper arasındaki bir akımı temsil edecek şekilde ölçeklendirilir ve blok ram belleğe yazılır. Bunun için gerekli LabVIEW uygulamasına ait blok diyagram Şekil 4.17 de gösterilmiştir.



Şekil 4.17 Fazlardan okunan analog verilerin ölçeklendirilmesine ait blok diyagram

4.3.8. Motor Konum Bilgisinin Okunması

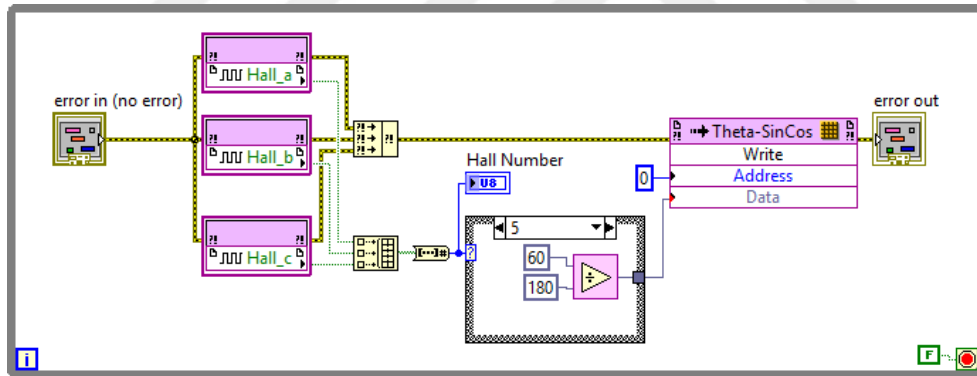
Motor konum bilgisinin okunması için motor içerisinde bulunan alan etkili algılayıcılardan yararlanılmıştır. Motordan gelen bilgiler 1, 0 şeklinde lojik olarak karşılaştırılır ve Tablo 4.3'den de görüleceği gibi mevcut 8 durum için çıkış açısı

belirlenir. Burada alan etkili algılayıcıların hepsinin aynı anda bir stator kutbunu göremeyeceğinden; 000 ve 111 durumlarında herhangi bir açı değeri okunmaz.

Tablo 4.4 Alan etkili algılayıcı değerlerinin açısal ifadesi

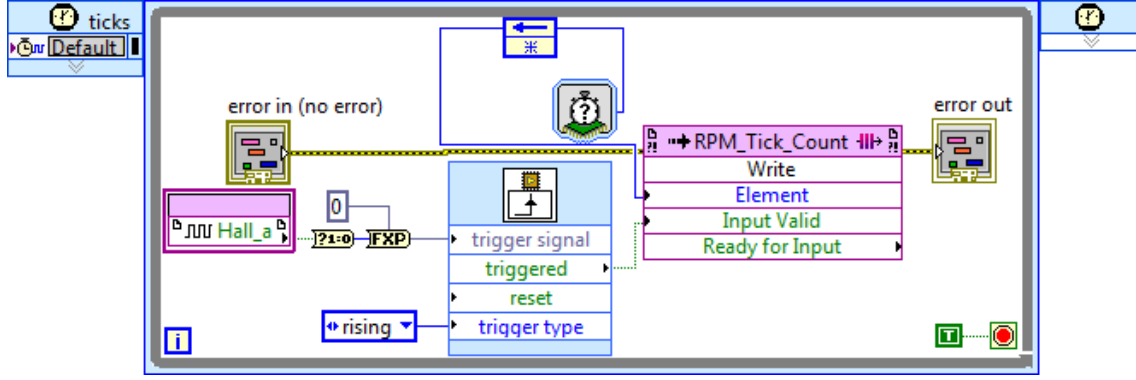
Hall a	Hall b	Hall c	Hall numarası	Radyan cinsinden açı değeri
0	0	0	0	-
0	0	1	1	2
0	1	0	2	1.33333
0	1	1	3	1.66666
1	0	0	4	0.66666
1	0	1	5	0.33333
1	1	0	6	1
1	1	1	7	-

Park ve Ters Park Dönüşümü için gerekli Theta açısının ve Tablo 4.4 te gösterilen açı değerlerinin elde edilmesi için, Şekil 4.87 de görülen blok diyagram oluşturulmuştur.



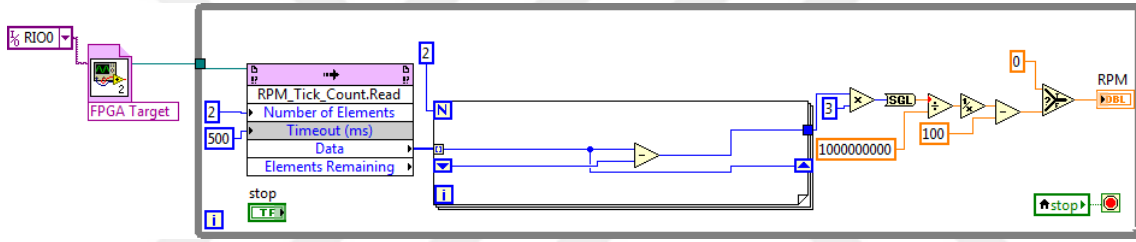
Şekil 4.18 Theta (θ) açısının belirlenmesi

Elde edilen açı verisinin, sin ve cos değerleri hesaplanarak bir blok ram belleği yardımıyla park ve ters park dönüşümleri için ana programa aktarılır. LabVIEW FPGA modülünde yer alan “yüksek işlem hacmi (High Throughput Sin, Cos)” nesnesi, açı değerlerini radyan cinsinden kabul ettiğinden açıların radyana çevrilmesi gerekmektedir. Şekil 4.18 $x(Pi)$ açı değeri radyan cinsinden olmalıdır ve bu işlem Eşitlik 4.21 ile hesaplanır.



Şekil 4.20 Hall a algılayıcı verisinin okunması

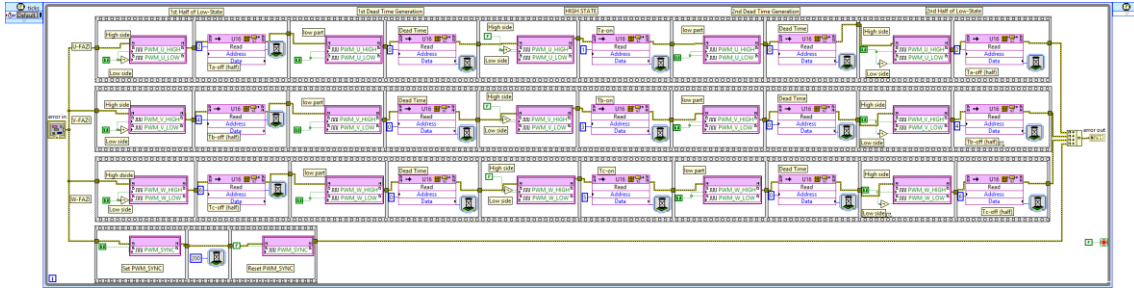
FIFO değerlerinin işlenmesi ve RPM'e dönüştürülmesi işlemi ise, Şekil 4.21 de görüldüğü gibi kullanıcı bilgisayarında oluşturulan bir blok diyagramda gerçekleştirilmiştir.



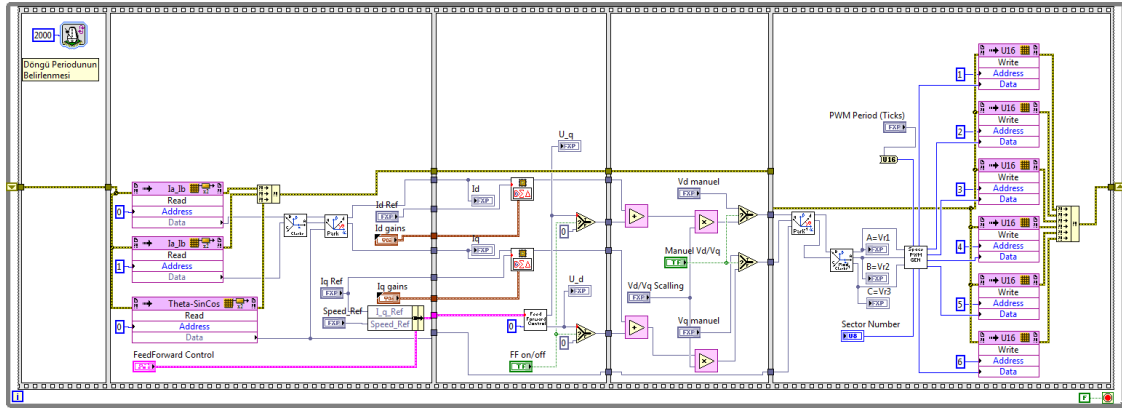
Şekil 4.21 Kullanıcı bilgisayarında FIFO verilerinin okunması ve RPM değerine dönüştürülmesi

4.3.10. PWM Sinyallerinin Oluşturulması

Altı konumlu anahtar için gerekli PWM işaretlerini üretmek için her fazda bir, toplamda üç sıralı yapı kullanılmalıdır. PWM sinyalleri, çıkışlara bir boolean (true, false) değerinin atanması ve ardından SVPWM üretici bloğunun çıkışında (Bölüm xxx e bakınız) üretilen bekleme sürelerinin bu sayısal çıkışlara (DO9..14) uygulanması ile oluşur. Atanan bekleme süresi dolduktan sonra, çıkışlara yeni bir değer atanır; ardından başka bir bekleme süresi eklenerek PWM sinyalleri oluşturulur. Şekil 4.22 LabVIEW ortamında hazırlanmış u, v ve w fazları için PWM sinyallerinin üretilmesini göstermektedir.



programları (Sub VI) da içermektedir. Tüm alt programlara (vi) ait giriş-çıkış ve göstergeler ana programda birbirine bağlanmıştır.



Şekil 4.25 Alan Yönlendirmeli Kontrol Algoritması Blok diyagramı

Uygulamadaki blok RAM bellekler programlar arası döngü içine giren ve çıkan verileri tutmak için kullanılmıştır. Verilerin eşitlik zamanlı çalışması için sıralı bir dizi yapısı kullanılmıştır. Bu yapı, Şekil 4.25'te de görüldüğü gibi senkronizasyon için 5 bölüme ayrılmıştır. Sıralı yapının ilk bölümü, döngü periyodunun (T_s) tanımlandığı yerdir. Döngü periyodu FPGA içerisinde bulunan osilatör frekansına göre tick cinsinden girilmelidir. FPGA da tick değeri, saat frekansının zaman cinsinden değerini ifade etmektedir. Döngü periyodunun tick cinsinden değerinin hesaplanmasını Eşitlik 4.15 de gösterilmiştir.

$$T_s : \text{GirilenSayı} \times \text{SistemTickPeriyodu} \quad (4.19)$$

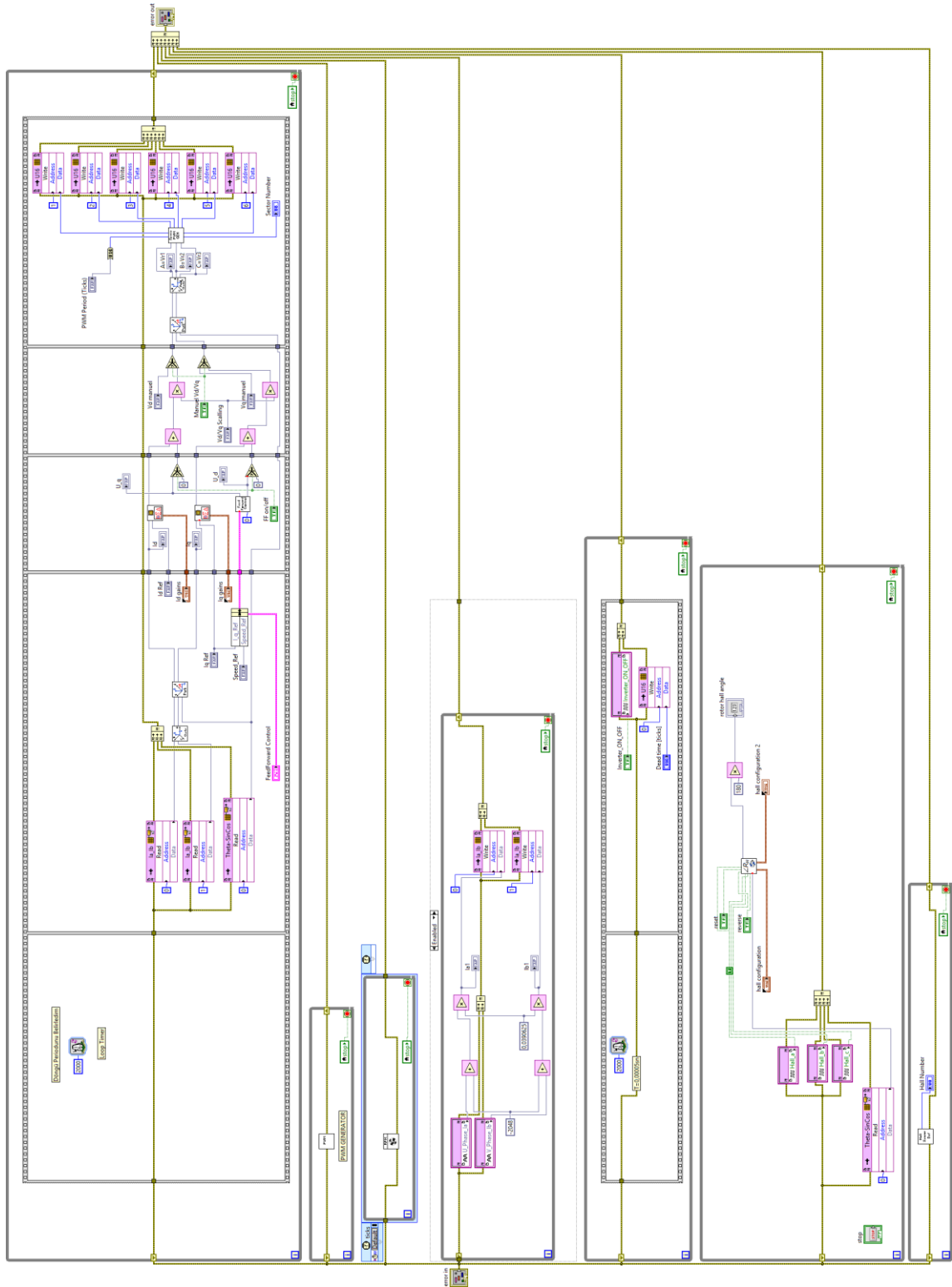
40MHz saat frekansına sahip bir FPGA için tick değeri:

$$\text{Tick} = \frac{1}{40 \times 10^6} = 25ns \quad \text{Olarak bulunur.} \quad (4.20)$$

Sıralı yapının ikinci bölümünde, hesaplamada kullanılacak tüm veriler (alan etkili algılayıcı verileri, kullanıcı verileri, geri besleme verileri) RAM belleği üzerinden okunur. Clarke ve Park dönüşümü alt programları da bu bölümde çalıştırılır. Üçüncü bölümde, veri çıkışları RAM bloğa yazılmadan önce kontrolörler, ters park ve Clarke dönüşümleri ve SVPWM tarafından işlenir ve düzenlenir. Dördüncü bölümde ise, V_d / V_q ölçekleme faktörü, vektörü altı konumlu anahtara iletilecek şekilde DA besleme

gerilimine normalleřtirmek iin kullanılır. V_d / V_q lekleme faktr Blm 4.4.1 de ayrıntılı bir řekilde verilecektir. Son blmde ise dzenlenmiř ve leklenmiř gerilim vektrleri blok RAM belleęe SVPWM kullanılarak yazılır. Altı konumlu anahtar ve algılayıcıların FPGA ya baęlanması iin oluřturulan bileřenler alt VI'lar gibi paralel olarak programa dhil edilmiřtir. řekil 4.26 FPGA ierisinde alıřan tm programlara ait blok diyagramı gstermektedir.

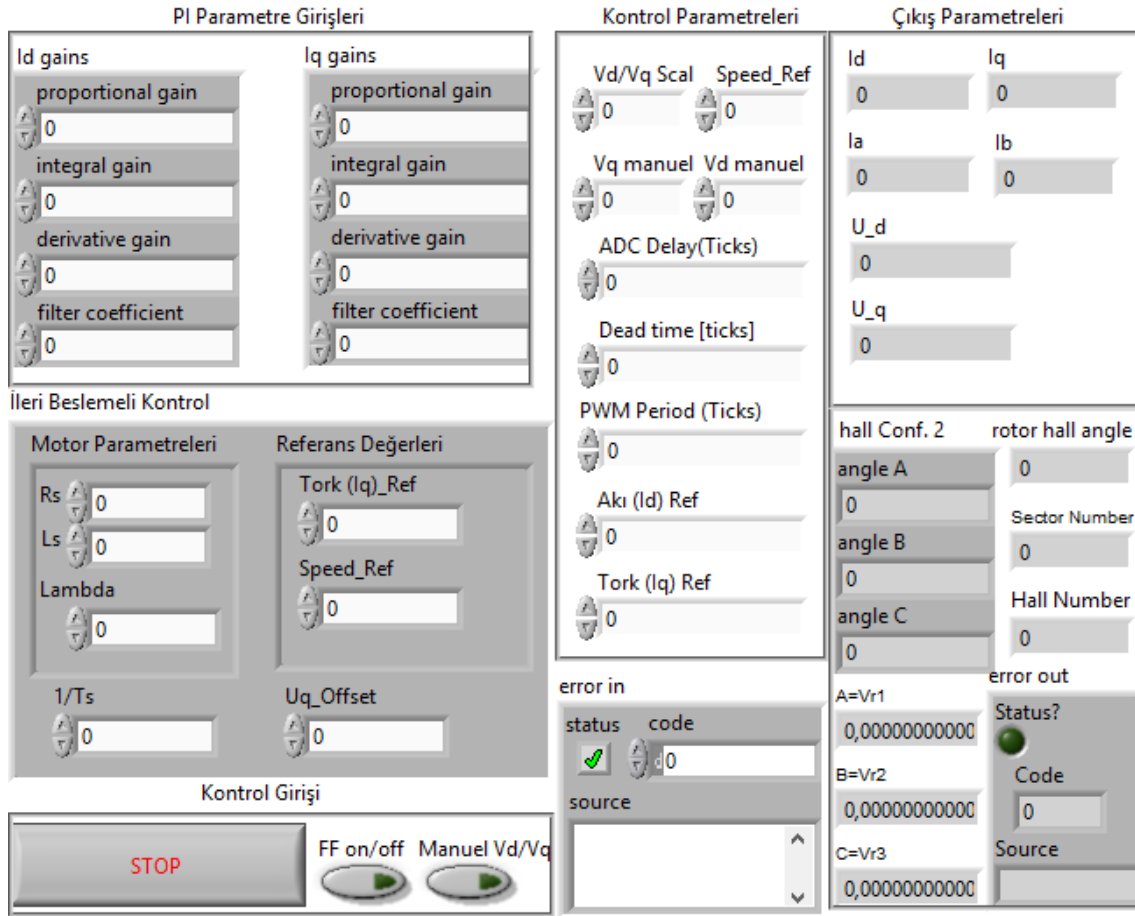




Şekil 4.26 FPGA içerisinde çalışan tüm programlara ait blok diyagram

Şekil 4.26 da FPGA hedef ana program ve bileşenlerine ait ön panel görülmektedir. Çalışma anında sürücü devresinin açılıp kapanması ve ölü zamanın değiştirilmesi için

bir döngü kullanılmıştır. Verilerin izlenmesi ve parametrelerin değiştirilmesi için ön panelde gerekli göstergeler ve kontrolörler bulunmaktadır. LabVIEW FPGA modülü, grafiksel olarak veri sunma olanağını da desteklemektedir. Verilerin grafik üzerinden izlenmesi isteniyorsa, bu verilerin başka bir ortama aktarılması gerekmektedir. Söz konusu veriler DMA (Direct Memory Acces) FIFO'lar kullanılarak kullanıcı bilgisayarına (Host PC) aktarılabilir.

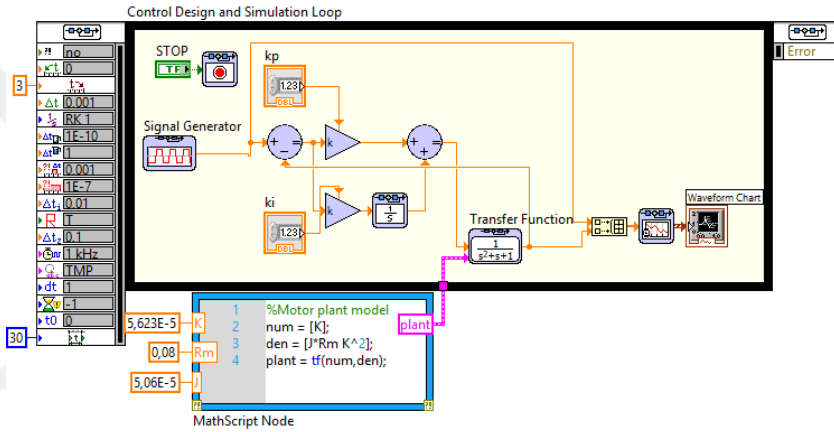


Şekil 4.27 FPGA Hedef ana ve alt program bileşenlerine ait ön panel görünümü

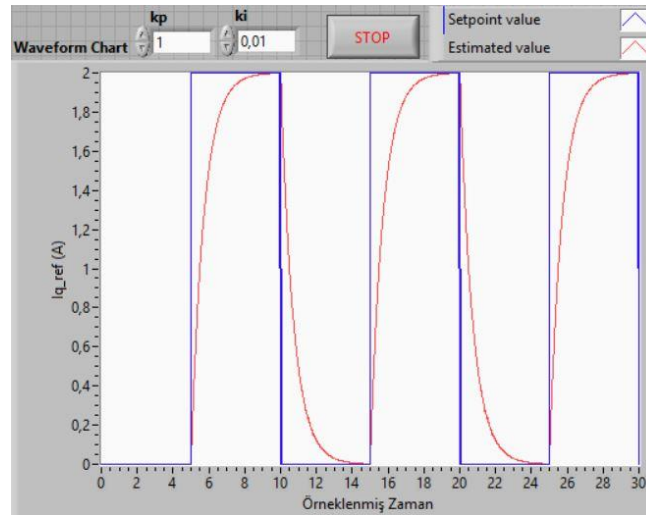
4.4.1. PI Parametrelerinin Belirlenmesine Yönelik Test Programı

Alan yönlendirmeli kontrol algoritmasının uygulanması için döngüdeki hesaplamaların, döngünün önceki tekrarlamalardan herhangi bir etkileşim olmadan gerçekleştirilmesi önemlidir. Bunun için döngünün ayrılmaz bir parçası PI denetleyiciler kullanılmıştır. Kullanılan PI'lar LabVIEW de hazır olarak verilmiş ve birden çok sistemin aynı anda kontrol edilebileceği şekilde yapılandırılmıştır. Kontrol edilecek sistem seçimi kanal seçimi ile sağlanmaktadır ve her kanal için parametreler ayrı ayrı belirlenebilmektedir.

PI bloğundaki integral kısmı, önceki tekrarlamalardan oluşan hata verilerini depolar ve tüm girişlerden gelen verilerin bağımsız olarak çalışmasını ve çıkışa aktarılmasını sağlar. PI bloklarının işlevselliğini doğrulamak için geliştirilen PI test programı blok diyagramı ve kullanıcı ara yüzü, Şekil 4.28’de görülmektedir. LabVIEW ortamında hazır olarak verilen PID bloğu, uygulamada PI bloğu olarak kullanılması için türevsel (Derivative “D”) değeri sıfır olarak alınmıştır. Uygulama, bu program ile test edilmiş ve belirlenen motor parametrelerine göre en iyi değerler K_i için 5.2, K_p için 0.2 olarak bulunmuştur. Gerçek uygulamada da bu değerler I_d ve I_q kazançları için PI düzenleyicilerde kullanılmıştır.



a)

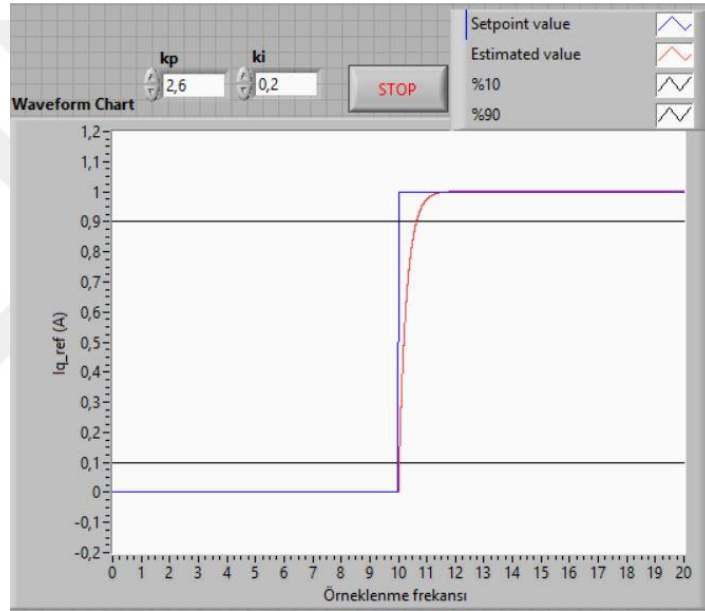


b)

Şekil 4.28 PI test programı a) blok diyagramı b) kullanıcı arayüzü

4.4.1. I_q Parametresinin Birim Basamak Cevabı

I_{q_ref} değerine genliği 2 A ve frekansı 1 Hz olan bir kare dalga fonksiyonu uygulanarak, çıkışta I_q değerinin zamana göre değişimini ekranda gösterecek şekilde bir test programı hazırlanmıştır. Bu yöntem PI parametrelerinin değerlerini belirlemek ve birim basamak cevabı üzerindeki etkilerini göstermek için kullanılır. Bu testlerin amacı, sistemin ne kadar hızlı ve kararlı olduğu hakkında bir fikir elde etmektir. I_d parametresinin basamak cevabına gerek yoktur; çünkü her koşulda I_d değeri sıfıra eşittir.



Şekil 4.29 I_q parametresi basamak cevabı ($f_s=20000$ Hz)

Şekil 4.29 I_{q_ref} değerinin “0” olduğu durumda I_q akımının birim basamak cevabını göstermektedir. $K_p=1$ ve $K_i=0.01$ alınması durumunda integral süresi, Eşitlik 4.21 denklemi ile hesaplanmaktadır.

$$T_i \cong K_p \times \frac{T_s}{K_i} = 1 \times \frac{1/20000}{0.01} = 0.005 = 5ms \quad (4.21)$$

Bu durumda sistemin yerleşme zamanı;

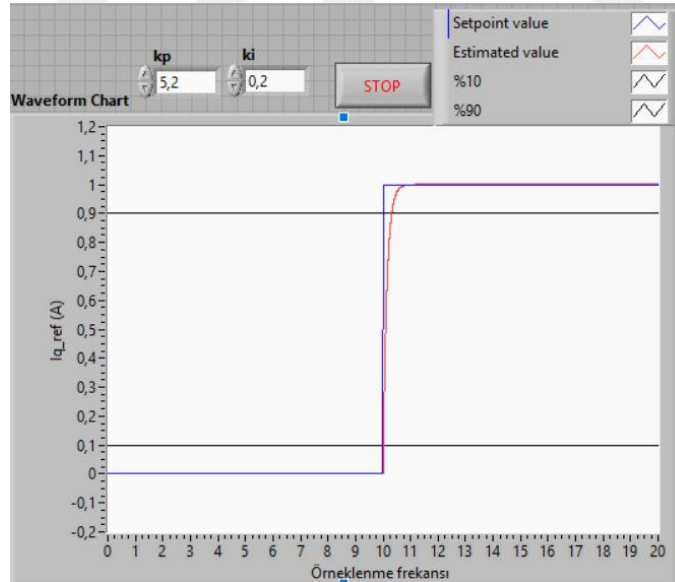
$$T_{Yerleşme_Zamanı} \cong \frac{T_{Yerleşme_Noktası} - T_{Başlama_Noktası}}{f_s} = \frac{12250 - 10250}{20000} = 100ms \quad (4.22)$$

Basamak sinyalinin başlama noktası 5. sn, sisteme oturma noktası ise 10. sn ye karşılık gelendir. Eşitlik 4.22’de verildiği gibi yerleşme süresi oldukça uzundur. Bunun nedeni K_p ve K_i değerlerinin oldukça küçük seçilmesidir. Kararlı durum hata bandı $[+0.1, -0.1]$ aralığında olmaktadır ve bu da görelî hata oranını $\pm\%10$ değerinden daha az değerde tutmaktadır.

K_p ve K_i değerlerine göre sinyalin yükselme zamanı Eşitlik 4.23 denklemi ile hesaplanmaktadır.

$$T_R \cong \frac{T_{\%90} - T_{\%10}}{f_s} = \frac{11000 - 10250}{20000} = 37.5ms \quad (4.23)$$

PI parametrelerinin değiştirilmesi durumunda ($K_p = 5.2$ ve $K_i = 0.2$) sistemin birim basamak cevabı Şekil 4.30’da ki gibi olmaktadır.



Şekil 4.30 Değiştirilmiş PI parametreleri ile sistemin yeni cevabı

Bu durumda sistemin integral süresi aşağıdaki gibi olmaktadır.

$$T_i \cong K_p \times \frac{T_s}{K_i} = 5.2 \times \frac{1/20000}{0.2} = 1.3ms \quad (4.24)$$

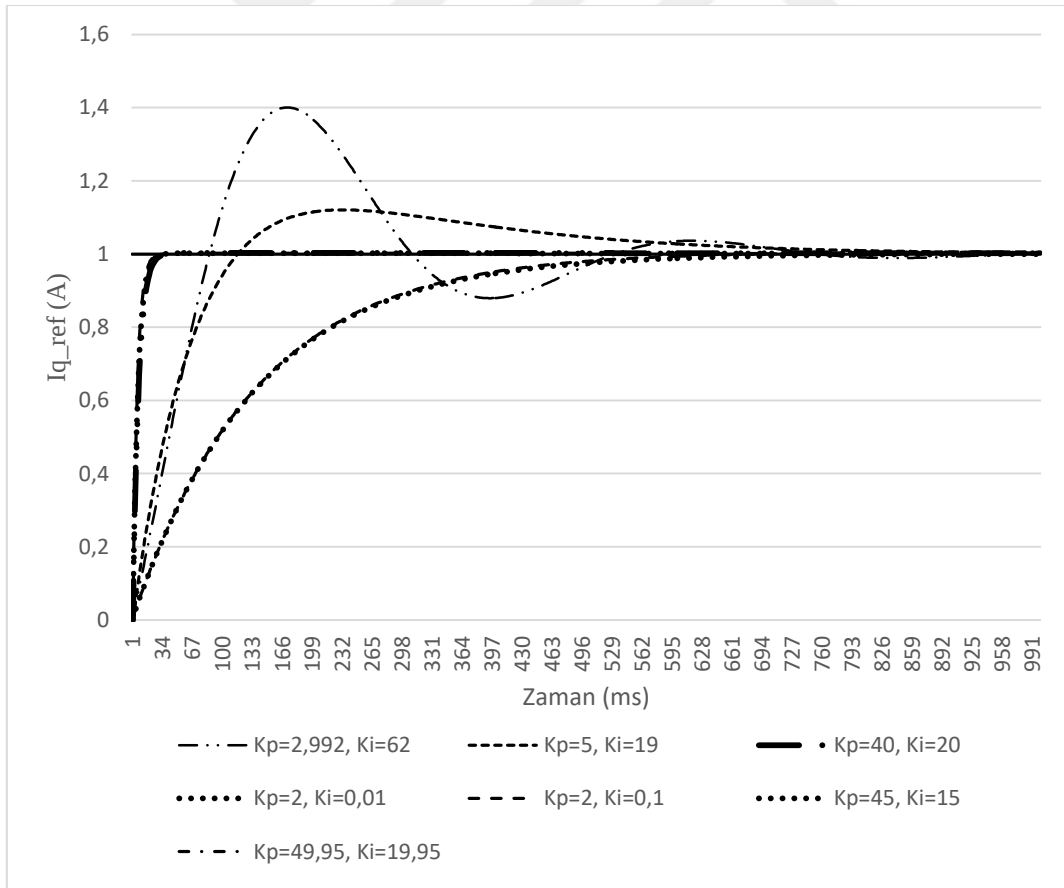
Sistemin oturma süresi;

$$T_{Oturma_Suresi} \cong \frac{T_{Yerleşme_Noktası} - T_{Başlama_Noktası}}{f_s} = \frac{11250 - 10250}{20000} = 50ms \quad (4.25)$$

Sinyalin yükselme zamanı ise aşağıdaki gibi olmaktadır.

$$T_R \cong \frac{T_{\%90} - T_{\%10}}{f_s} = \frac{10500 - 10000}{20000} = 25ms \text{ olmaktadır.} \quad (4.26)$$

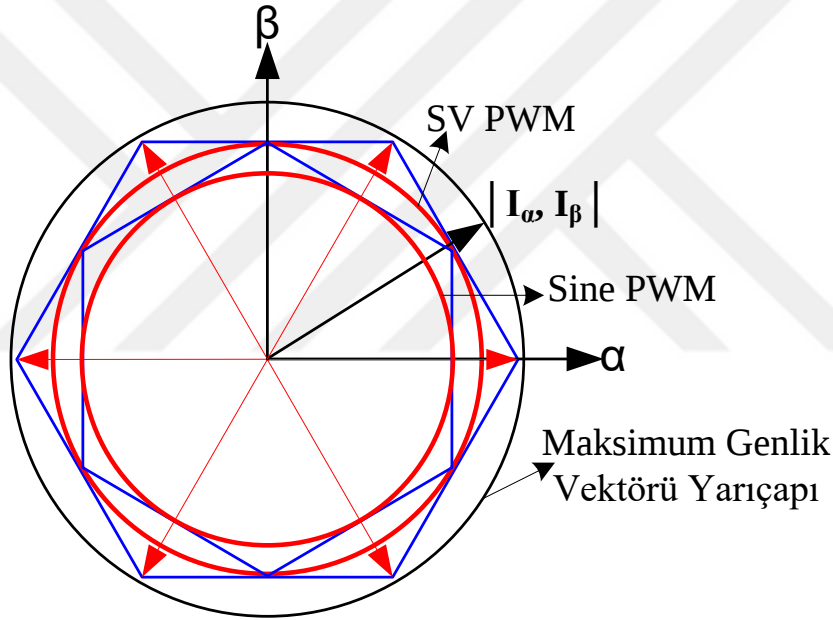
Süreç boyunca farklı PI parametreleri kullanılarak yapılan deneylere ait sonuçlar Şekil 4.31 de karşılaştırılmıştır. Ancak buradan edinilen bilgiler ışığında $K_p=5,2$ ve $K_i=0,2$ değerleri ile deneylere devam edilmiştir.



Şekil 4.31 1A referans değeri için farklı K_p ve K_i değerlerinde sistemin PI cevabı

4.4.2. V_d/V_q Ölçekleme Faktörü

PI denetleyicilerinin çıkışı motora uygulanan gerilime karşılık gelen V_d ve V_q bileşenlerini temsil etmektedir. SVPWM'in düzgün bir şekilde çalışabilmesi için I_α, I_β 'nin büyüklüğü SVPWM altıgeninin içine yerleştirilmiş dairenin yarıçapından daha büyük olmamalıdır. (Bakınız Şekil 4.31). Motora maksimum gerilim uygulandığında I_α, I_β büyüklüğü, daire yarıçapına eşitlenir. Eğer genlik daire yarıçapından büyük olursa; PWM uygulanacak çıkışın bir veya birkaçının görev döngüsü normal değerden büyük veya negatif olacaktır.



Şekil 4.32 I_α, I_β Maksimum genlik vektörü gösterimi

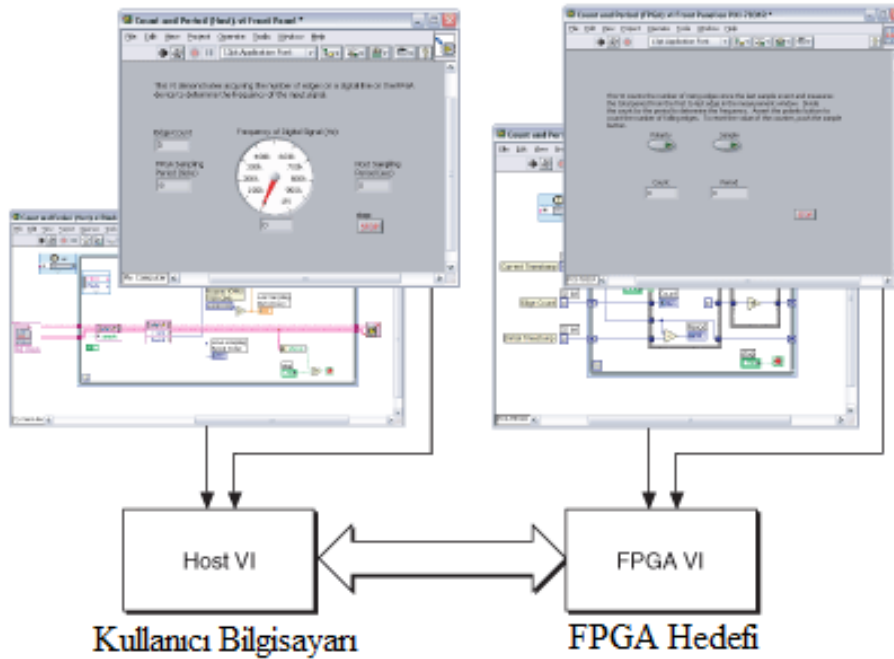
Burada maksimum genlik vektörünün yarıçapı $\frac{2}{3}V_{DC}$, Sine PWM genlik vektörü yarıçapı $\frac{1}{2}V_{DC}$ iken SVPWM uygulamasında daire $\frac{1}{\sqrt{3}}V_{DC}$ yarıçapa sahiptir. Vektörün genliğinin daire içerisine sığması için normalleştirme işlemi uygulanmalıdır. Bu V_d/V_q bileşenlerini DA besleme gerilimine normalleştiren bir faktör ile ölçeklendirilerek yapılır. Bu ölçekleme faktörü Eşitlik 4.27 de gösterilmiştir.

$$\text{Ölçekleme_Faktörü} = \frac{\sqrt{3}}{V_{DC}} \quad (4.27)$$

Ölçekleme işlemi ters park dönüşümünden önce veya sonra yapılabilir. Bu tez çalışmasında ölçekleme işlemi ters park dönüşümünden önce yapılmaktadır. Bilindiği gibi bu dönüşüm sinyalin genliğinin korunduğu dönüşümdür.

4.5. Kullanıcı Bilgisayarında Gerçekleştirilen İşlemler

FPGA ortamında çalıştırılan kod blokları, döngüler ve algılayıcılardan alınan verilerin görüntülenmesi ve değişiklik yapılması için bir kullanıcı bilgisayara ihtiyaç vardır. Bu bilgisayar bir Network hattı üzerinden sbRIO karta bağlanmıştır. (Bakınız Şekil 4.4) Network hattı sayesinde kullanıcı bilgisayarda çalışan VI ile FPGA hedefte çalışan VI arasında programatiksel olarak etkileşim kurulmaktadır. Bu programatik FPGA arayüz haberleşmesi, interaktif ön panel haberleşmesinden farklıdır. Çünkü etkileşim için bir ana VI ile bir de FPGA VI gereklidir. Programatik FPGA arayüz haberleşmesi kullanılırken, FPGA VI kodu FPGA hedefi üzerinde; Kullanıcı VI ise kullanıcı bilgisayarında çalışmaktadır. Şekil 4.32 programatik haberleşmede kullanıcı bilgisayar ile FPGA arası örnek bir bağlantıyı göstermektedir.



Şekil 4.33 Kullanıcı Bilgisayar ile FPGA hedefi arası haberleşme [24]

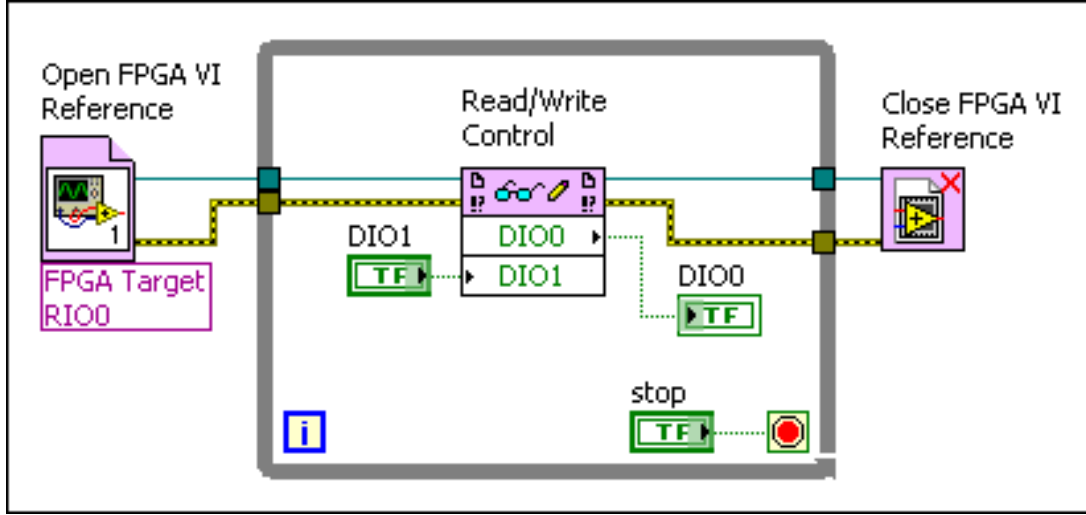
Ana bilgisayar ile FPGA hedef arasında veri alışverişinin nedenleri aşağıda sıralanmıştır.

- 1) FPGA hafızasının yetersiz kalması
- 2) FPGA içerisinden matematiksel hesaplamaların kısıtlı olması ve hassas hesaplama yapma isteği
- 3) Çok katmanlı uygulama oluşturma isteği
- 4) Verilerin anlık olarak kaydedilmesi isteği
- 5) Veri aktarımının zamanlanması ve kontrol edilmesi
- 6) FPGA için *test bench* oluşturmak

Burada kullanıcı bilgisayarı Windows tabanlı bir bilgisayar veya bir gerçek zamanlı işletim sistemi olabilir. Hem Windows işletim sistemi hem de RT işletim sistemi, FPGA arabirim işlevlerini desteklemektedir. Ayrıca buna ek olarak RT hedefi ile FPGA hedefi arasında iletişim kurmak için FPGA arabirim fonksiyonlarını kullanabiliriz. Aşağıdaki adımlar kullanıcı bilgisayarı VI için genel programlama sırasının ana hatlarını göstermektedir.

- 1) Kullanıcı bilgisayarında FPGA VI referansı açılır ve bit dosyası oluşturulur.
- 2) FPGA arabirim fonksiyonları (Read/Write Control, Invoke Method) kullanılarak veri gönderimi veya alımı yapılır.
- 3) Close FPGA VI Reference ile açılan FPGA VI referansı kapatılır.

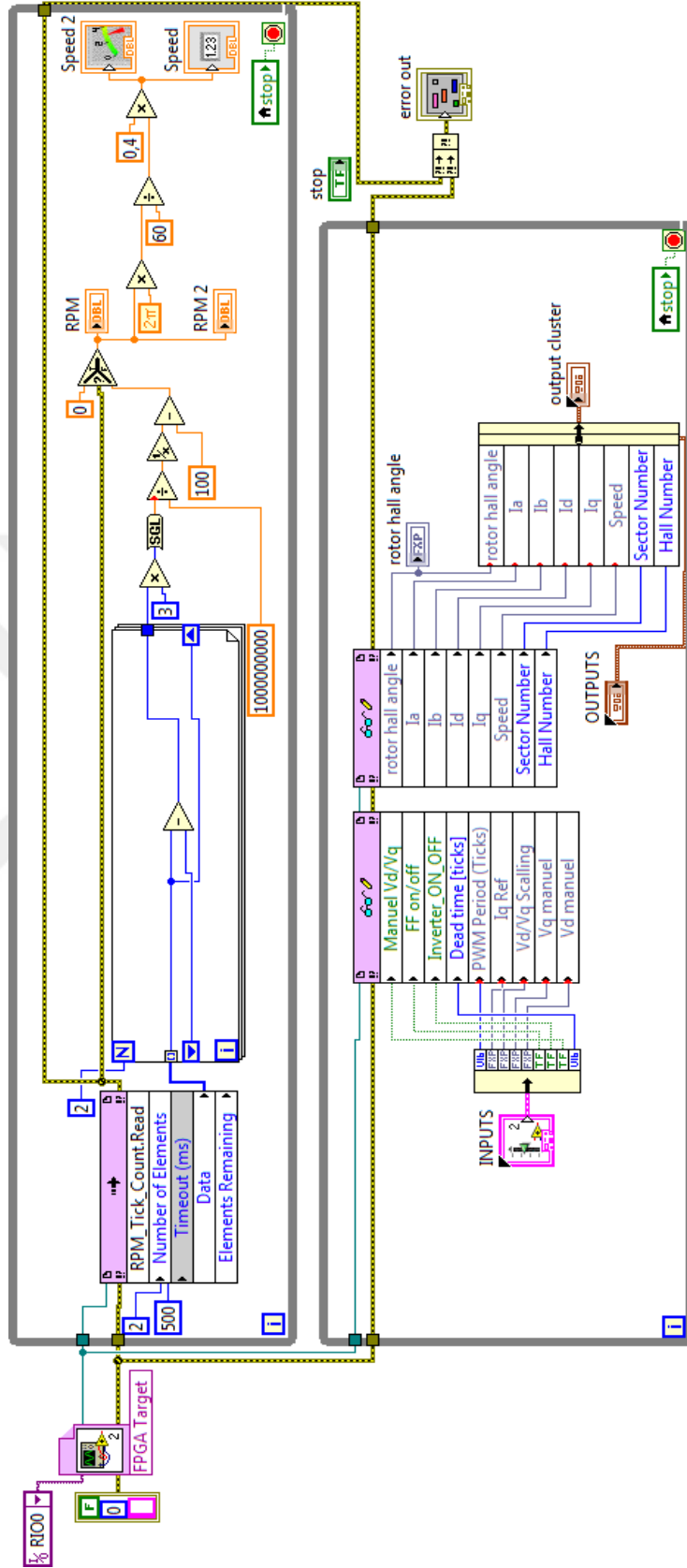
Şekil 4.30 daki blok diyagram, yukarıda belirtilen adımların gerçekleştiği kullanıcı bilgisayarındaki LabVIEW kodunu göstermektedir. Bu kullanıcı bilgisayarındaki VI örneği FPGA üzerinde DIO1' doğru /yanlış değerinin yazılması ve DIO0'dan bu değerin okunmasını göstermektedir.



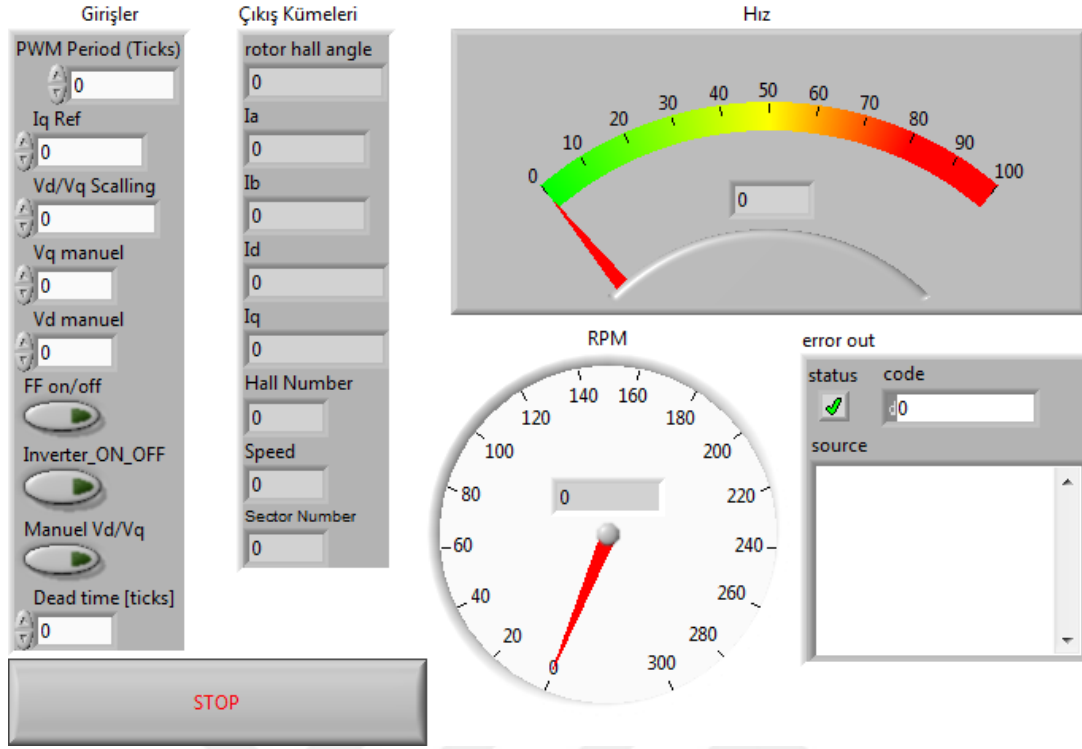
Şekil 4.34 FPGA VI üzerinde veri okuma/yazma [24]

Bu tez çalışmasında ise motor kontrol algoritmasından alınan verilerin tamamının grafik ve çizelgelerde gösterildiği, kontrol parametrelerinin değiştirildiği kullanıcı bilgisayar VI ön paneli, Şekil 4.35'te görülmektedir. Burada motor parametrelerine ait değerler değiştirilip, girilebilir veya hız, rpm gibi çıkışlar görsel nesneler ile izlenebilir.

Blok diyagram program kodlarının yazıldığı yerdir ve burada FPGA VI da koşulan program komutlarının verileri FPGA VI referanstan alınmaktadır. Değerlerin değiştirildiği ve göstergelerin bulunduğu ön panel Şekil 4.34 de gösterilmiştir.



Şekil 4.35 Gerçek Zaman (Real Time) kullanıcı ara yüzü blok diyagramı



Şekil 4.36 Gerçek zamanlı kullanıcı ara yüzü ön panel gösterimi

5. ALAN YÖNLENDİRMELİ KONTROL ALGORİTMASI KULLANAN FSDA MOTORU SÜRÜCÜ SİSTEMİNİN BENZETİMİ

Uygulama sonuçlarının işlevselliğinin doğrulanması için sistemin Matlab/Simulink modeli oluşturulmuş ve benzetim sonuçları elde edilmiştir. Bu sonuçlar bir önceki bölümde anlatılan LabVIEW FPGA modül kullanılarak sürülen gerçek motor üzerinden elde edilen sonuçlar ile karşılaştırılmıştır. Uygulama aşamasında tüm sistem bileşenleri test edilmiş ve bileşenlerin doğruluğunu artırmak için bazı parametreler yeniden belirlenmiştir.

5.1. Matlab / Simulink

Sistemin Simulink benzetim programında modellenmesinin amacı, LabVIEW ortamında hazırlanmış kontrol algoritmasının doğruluğunun test edilmesidir. Sistemin üç farklı fiziksel alana (elektrik, manyetik ve mekanik) yayılmış olması sebebi ile eksiksiz ve hatasız modelinin türetilmesi gerekmektedir. Kullanılan motor ve parametrelerin iyi anlaşılmış olması, denklem ve çözümler ile desteklenmelidir. Bu nedenle Matlab ortamında kontrol algoritmasının basitleştirilmiş halinin çıkarılması ve SimPowerSystem araç kutusunun sağlamış olduğu bir FSDA bloğunun kontrolü sağlanmıştır. SimPowerSystem, elektrik ve güç elektroniği uygulamalarının benzetimi için Simulink te sıklıkla kullanan bir araç kutusudur.

5.2. Motor Modeli

Bölüm 2.4'te gerilim cinsinden verilen motora ait matematiksel denklemler, burada SimPowerSystem deki gibi d, q çerçevesinde Eşitlik 5.1 ve 5.2'deki görüldüğü üzere I_d ve I_q akımları cinsinden ifade edilmiştir.

$$\frac{di_d(t)}{dt} = \frac{1}{L_d} V_d(t) - \frac{R}{L_d} i_d(t) + \frac{L_q}{L_d} \omega(t) i_q(t) \quad (5.1)$$

$$\frac{di_q(t)}{dt} = \frac{1}{L_d} V_q(t) - \frac{R}{L_d} i_q(t) + \frac{L_d}{L_q} \omega(t) i_d(t) - \frac{\lambda \omega(t)}{L_q} \quad (5.2)$$

Burada:

I_d : d eksenli akım bileşeni

I_q : q eksenli akım bileşeni

R : Armatür direnci

L_d : İndüktansın Akı bileşeni

L_q : İndüktansın tork bileşeni

$\omega(t)$: Rotorun elektriksel hızı (rad/sn)

λ : Motorun akı bağlantısı ($N\phi$)

d, q Referans çerçevesi için endüktanslar Eşitlik 5.3'teki gibi hesaplanır:

$$L_d = L_q = \frac{L_{ab}}{2} \quad (5.3)$$

Elektriksel ve mekanik açı arasındaki bağıntı, Eşitlik 5.4 ve 5.5'te verilmiştir.

$$\omega(t) = \frac{d\theta(t)}{dt} \quad (5.4)$$

$$\frac{d\omega(t)}{dt} = \frac{1}{J} [T_M - T_E - B\omega(t)] \quad (6.5)$$

Burada:

J : Atalet momenti

T_M : Mekanik tork

T_E : Elektriksel tork

B : Sürtünme katsayısı

Eşitlik 5.1- 5.3, dq referans çerçevesinde FSDA motorlar için genel dinamik motor denklemleridir.

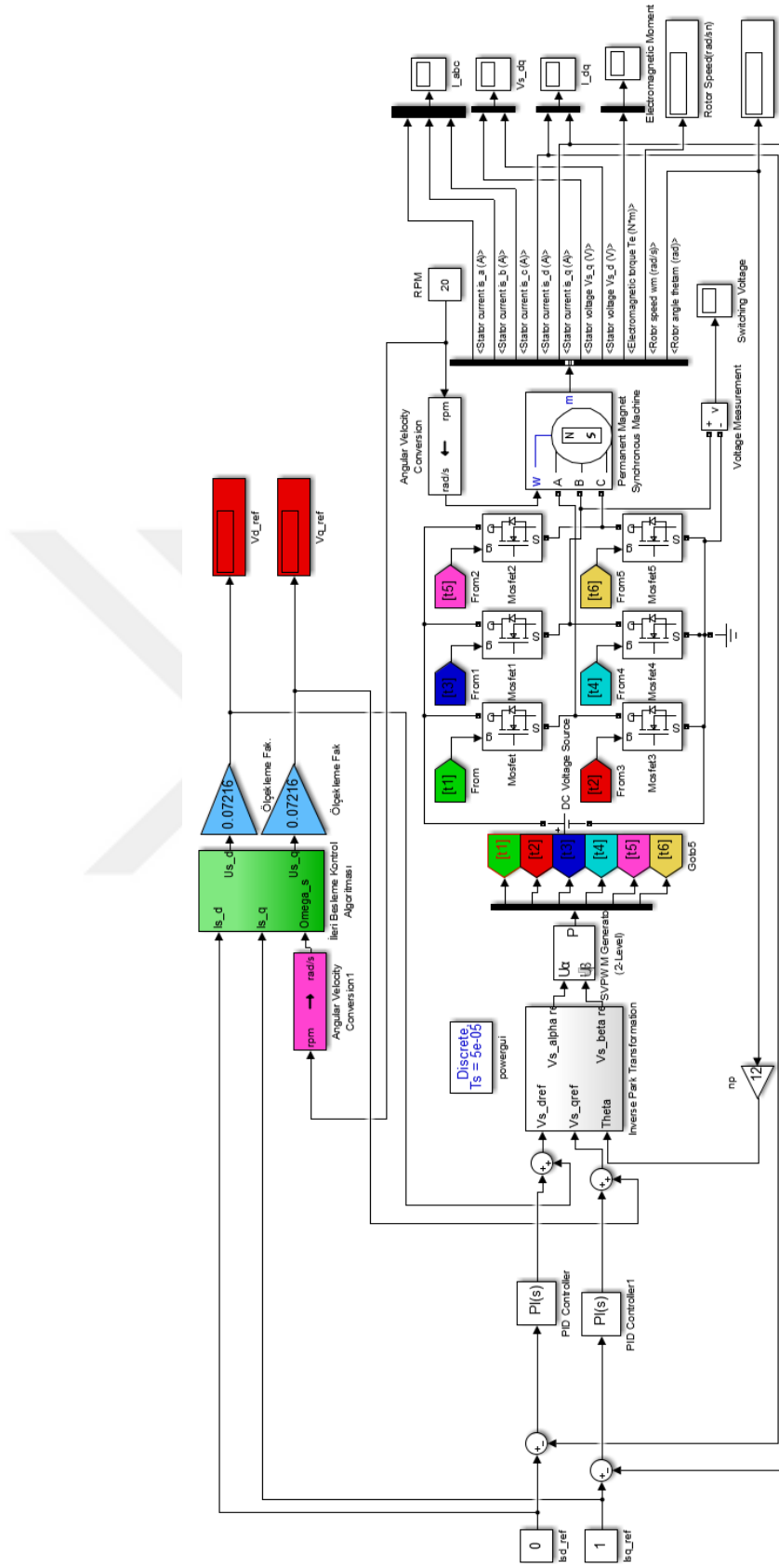
Benzetimde kullanılan motora ait parametre listesi Tablo 3.2 de verilmiştir. B (sürtünme katsayısı) haricindeki tüm veriler FSDA motor veri sayfasından alınmış değerlerdir. B değeri motor veri sayfasında verilmediği için 0.008 olarak alınmıştır. B değeri motorun uygun mekanik özellikleri elde etmek için sonra tekrar belirlenmelidir.

5.2.1. Alan Yönlendirmeli Kontrol Algoritmasının Matlab Uygulaması

Şekil 5.1, Simulink ortamında uygulamaya ait basitleştirilmiş FOC algoritmasını göstermektedir. Uygulamaya ait kod bloğunda toplamda 4 giriş ve kullanıcı tanımlı 13 çıkış bulunmaktadır. ω ile gösterilen ilk giriş motorun belirli bir hızda rad/sn çalışıp çalışmayacağını veya belirli bir mekanik tork (T_M) ile yükleneceğini göstermektedir. Geriye kalan üç giriş, üç fazlı invertör çıkışlarının bağlandığı motor sargıları u, v ve w dir. Belirtilen 13 çıkış ise isteğe bağlıdır ve bir veri yolu ile seçilebilir. Çıkış değerleri; u, v, w stator akımları, I_d/I_q , V_d/V_q , rotor hızı (rad/sn), rotor açısı (rad/sn), alan etkili algılayıcılar (halla, hallb, hallc), elektromanyetik tork (Nm) şeklinde sıralanabilir. Ayrıca kullanıcı motorun geri elektromanyetik kuvvetini de belirleyebilir. Geri elektromanyetik kuvvet FSDA için sinüzoidal veya trapez biçimli bir dalga şekli olabilir. Standart motor parametrelerinde, stator direnç (R_s), dq çerçevesi endüktans değerleri (L_d , L_q), tork ve gerilim sabitleri (K_t , K_e) ve kutup çifti (n_p) değerlerinin belirlenmesi gerekmektedir.

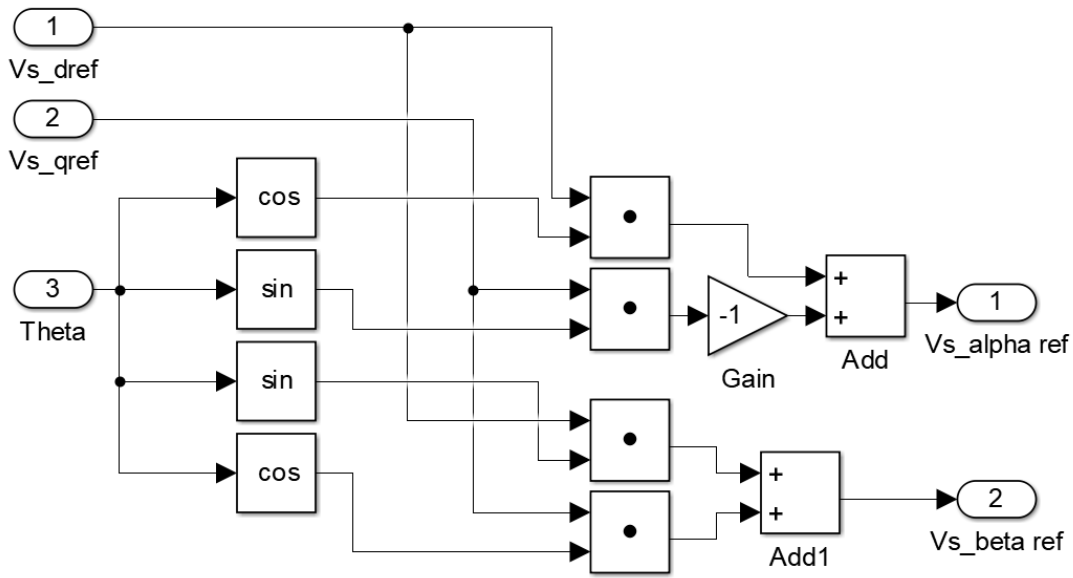
PI denetleyicide kullanılan parametre değerleri uygulamada da aynı şekilde kullanılır. Uygulamada oransal kazanç $K_p=5.2$ ve integral kazancı $K_i=0.2$ olarak alınmıştır. Denetleyicilerin örnekleme frekansları her biri için 16.6KHz olarak ayarlanmıştır.

Benzetim ortamında DSDA bloğu, I_d ve I_q cinsinden akım değerleri üretmektedir. Dolayısıyla üç fazlı akımların d , q referans çerçevesine dönüştürmek için Clarke ve Park dönüşümü kullanmaya gerek yoktur. Geri besleme akımları motor çıkışından doğrudan elde edilebilir. Rotorun konumu da motor bloğu tarafından doğrudan elde edilebilir. Dolayısı ile konum tespiti için algılayıcı modellemesine gerek yoktur.



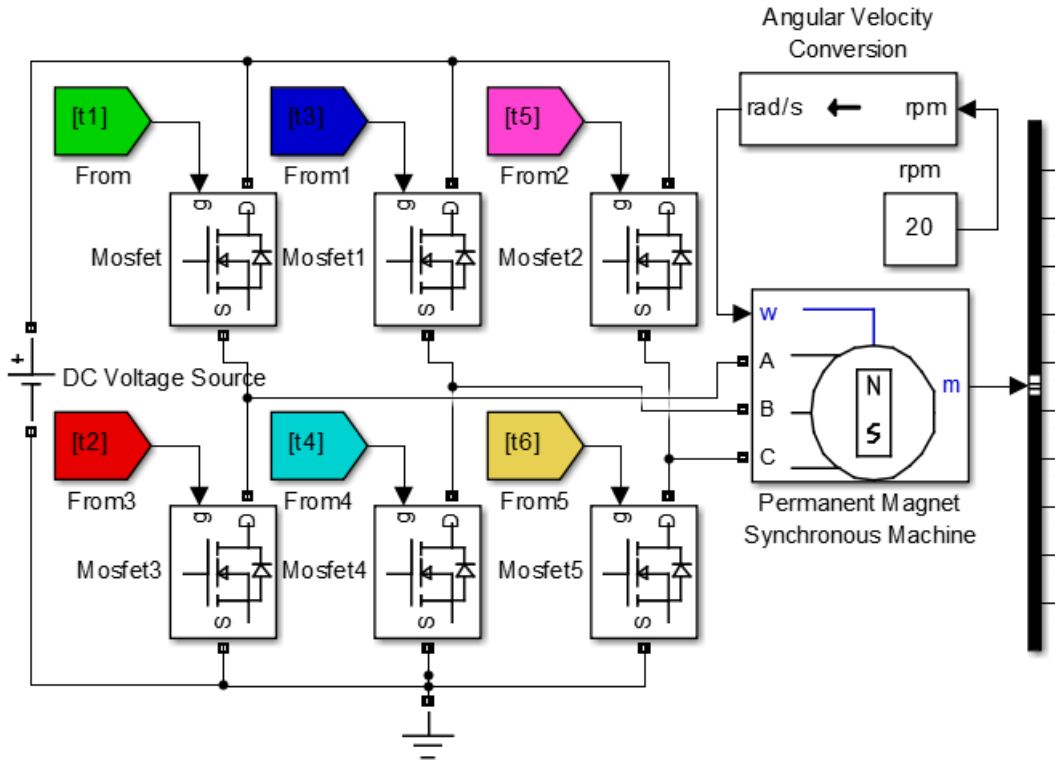
Şekil 5.1 FFC İle Birlikte Alan Yönlendirmeli Kontrol algoritması Matlab modeli

Şekil 5.1 uzay vektörü modülasyon ve altı konumlu anahtar yapısını içeren blok yapıyı göstermektedir. SVPWM, Matlab SimPowerSystem tarafından sağlanmaktadır. Modülasyon indeksi ve referans vektör tanımlaması FPGA içerisinde yazılan kod bloğundan biraz farklı ama temelde aynı işlevleri yerine getirmektedirler. Diğer fark ise ters Clarke dönüşümü tarafından elde edilen I_d, I_q bileşenleri yerine I_α, I_β bileşenleri doğrudan girdi olarak kullanılmıştır. Bu durumda benzetim modelinde ters Clarke dönüşümü kullanılmamıştır. Şekil 5.2 ters park dönüşümünde durağan referans çerçevesinden döner referans çerçevesine dönüşüme ait blok yapıyı göstermektedir.



Şekil 5.2 Ters park dönüşümü Matlab uygulaması

İnvertör yapısında ise altı adet ideal MOSFET kullanılmıştır. Burada, Şekil 5.1 ve 5.3 te de görüldüğü gibi denetleyici katından gelen etiketler ([t1]'den [t6]'ya kadar olan ve sıralı olarak) altı adet MOSFET'in kapı (gate) uçlarına uygulanmıştır. Üç faz çıkışları ise FSDA motorun 3 faz girişine sıralı bir şekilde bağlanmış ve gerilim kaynağı 48 V'a ayarlanmıştır. (Bakınız Şekil 5.3)

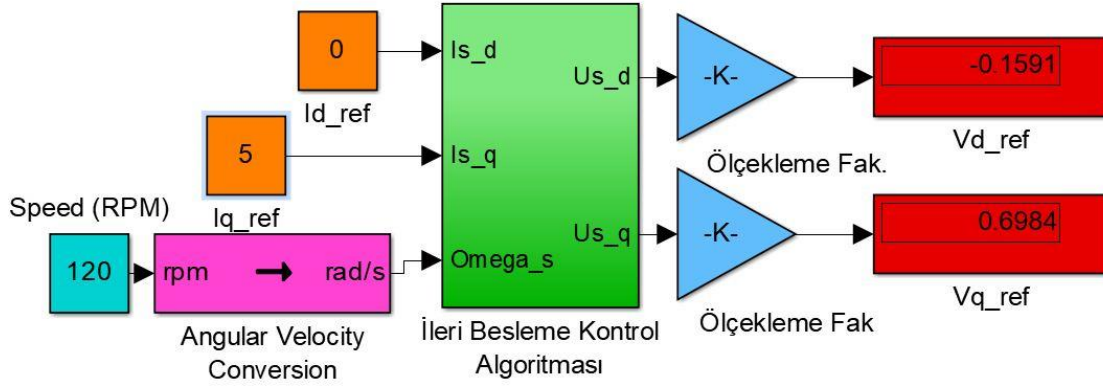


Şekil 5.3 İnvörtör yapısı ve FSDA motora bağlanması

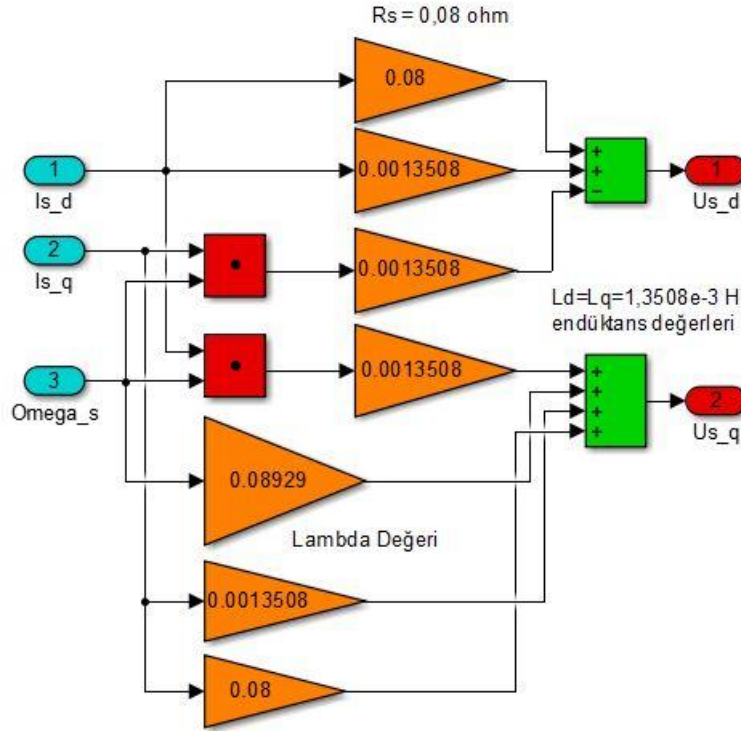
5.2.2. İleri Beslemeli Kontrol Algoritması Matlab Uygulaması

İleri beslemeli kontrol yapısı bölüm 4.3.3.2’de verilen denklemler ile elde edilir. Bu kontrol yöntemi hız ve farklı I_{q_ref} değerleri için uygulamanın V_d/V_q açısından benzetimi için basit bir modeldir. V_d ve V_q bileşenleri aynı girişler için uygulanan sisteme karşı belirlenecektir. İleri besleme kontrol algoritması Matlab uygulamasına ait blok diyagram Şekil 5.4 a ve b’de gösterilmiştir.

FFC algoritması Matlab uygulamasında farklı I_{q_ref} değerleri için yapılan deney sonuçları Tablo 5.1 de verilmiştir. Bu deneylerde $L_d=L_q=1.3508e-3H$, $R=0.08\Omega$, $\lambda=0.0446 \text{ N}\Phi$, $K_e=1.36524 \text{ Vs/rad}$, $V_d/V_q= 0.0721687$ olarak alınmıştır. Ayrıca FFC algoritması LabVIEW uygulaması ve Matlab benzetimine ait deney sonuçlarının karşılaştırılması ve değerlendirilmesi, Bölüm 6’da ayrıntılı bir şekilde verilmiştir.



a)



b)

Şekil 5.4 FFC algoritması Matlab uygulaması a) FFC genel blok yapısı, b) FFC algoritması iç yapısı

Tablo 5.1 FFC Algoritması Matlab Benzetim Sonuçları

Iq_ref	Speed_ref	Vd-Matlab
1	20	0.1211
	40	0.1937
	60	0.2611
	80	0.3286
	100	0.3961
	120	0.4636
2	20	0.1849
	40	0.2524
	60	0.3198
	80	0.3873
	100	0.4548
	120	0.5223
3	20	0.2436
	40	0.3111
	60	0.3785
	80	0.446
	100	0.5135
	120	0.5811
4	20	0.3023
	40	0.3698
	60	0.4372
	80	0.5047
	100	0.5722
	120	0.6397
5	20	0.3611
	40	0.4285
	60	0.4959
	80	0.5634
	100	0.6309
	120	0.6984

6. DENEY SONUÇLARI

Bu bölümde yapılan deneylerin amacı, sistemin daha iyi anlaşılmasını sağlamak, sistemi değerlendirmek ve sistemin performansını net bir şekilde ortaya koymaktır. Deneylerde tüm süreçler hem gerçek zamanlı olarak LabVIEW ortamında hemde benzetim olarak Matlab/Simulink ortamında gerçekleştirilmiştir. Ayrıca denetleyici tasarımında 3 farklı durum ele alınmıştır;

- 1- Sadece PI denetleyicisi kullanılarak akım cinsinden tork kontrolü
- 2- Sadece FFC algoritması kullanılarak PI hatalarının tespiti
- 3- PI ve FFC algoritmaları aynı anda paralel kullanılarak elde edilen iyileştirmelere yönelik karşılaştırmalar

6.1. Test ve Doğrulama

Şekil 6.1’de görüldüğü gibi bisiklet tekeri üzerine monte edilmiş 1 kW lık bir motordan oluşan bir deney düzeneği oluşturulmuştur.



Şekil 6.1 1kW FSDA motor deney düzeneği

Deneyler sırasında başka bir şey belirtilmediği sürece Tablo 6.1 de verilen parametreler kullanılmıştır.

Tablo 6.1 Kontrol Parametreleri

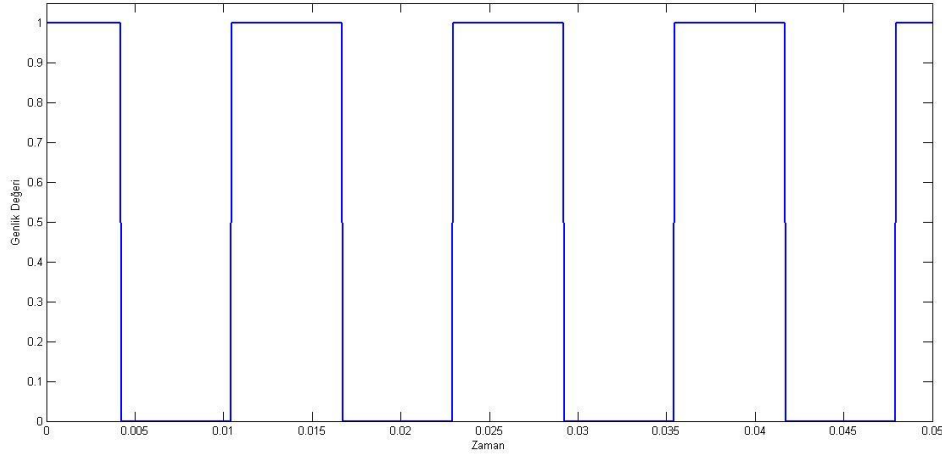
Parametre	Birimi	Değeri
Kontrol Döngüsü Örnekleme Frekansı	Ticks	2000
PWM Periyodu	Ticks	2400 (veya 16.6 KHz)
Vd/Vq Ölçekleme faktörü	-	0,721687
Ölü Zaman	Ticks	5
Iq kazancı (Kp)	-	5.2
Iq kazancı (Ki)	-	0.2
Id kazancı (Kp)	-	5.2
Id kazancı (Ki)	-	0.2

Testlerde, hat gerilimi ve akımları (u, v, w faz) 1 kW lık motor için osiloskop ve analizör yardımıyla izlenmiştir. Kontrolörden verileri örneklemek ve bunları grafikler veya göstergeler ile görüntülemek için basit bir VI oluşturulmuştur. Denetleyicilerdeki dâhili veriler, LabVIEW kod blokları ile izlenebilirken; FPGA üzerindeki veriler ise doğrudan bellek erişimi (DMA-FIFO) blokları ile gerçek zamanlı olarak kullanıcı bilgisayarına aktarılmıştır. Kullanıcı bilgisayarı FPGA ile ethernet ve DMA-FIFO'lar yardımıyla iletişim kurmaktadır. Daha sonra bu veriler kullanıcı bilgisayarındaki grafik ve çizelgeler yardımıyla yorumlanmıştır.

6.1.1. Alan Etkili Algılayıcı Verilerinin Doğrulanması

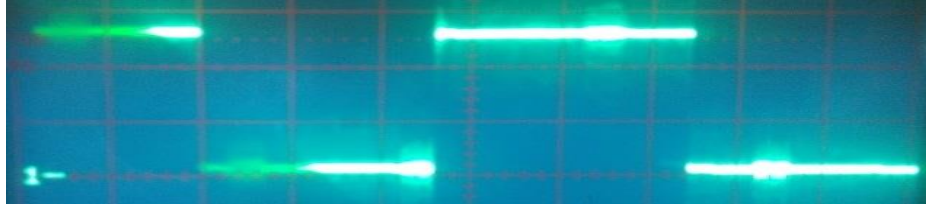
Motor ve alan etkili algılayıcıların birbirleri ile doğru bir şekilde kalibre edildiğini göstermek için motor pozisyonunun manuel olarak testi yapılmıştır. Motor kalibresi için herhangi bir faz (u, v, w burada u fazı referans alınmıştır) DC beslemeye; diğer fazlardan birisi de (v, w burada v fazı referans alınmıştır) toprağa bağlandığında rotor kutup noktasından herhangi birine kilitlenir. Bu durumda motora bağlanmış alan etkili algılayıcılar test edilir. Motorda 3 adet alan etkili algılayıcısı 120° lik açılar ile yerleştirilmiştir. Bu algılayıcılara ait her bir çıkış için dalga formu Bölüm 2.3'te verilmiştir. Alan etkili algılayıcıların (Halla, Hallb, Hallc) frekansı, motorun elektriksel açısal hızına bağlıdır. 16.6 KHz PWM frekansında, sistem parametreleri Tablo 6.1 de

verilen deęerlere set edildięinde, motor 120 rpm hızında alıřmıřtır. Bu deneye ait Matlab benzetim sonucu, řekil 6.2 de verilmiřtir.



řekil 6.2 Alan etkili algılayıcılara ait a-kapa sreleri Matlab benzetim sonucu

Aynı řartlarda test sisteminden elde edilen alan etkili algılayıcıların sinyal ıkıřı řekil 6.3 de verilmiřtir.

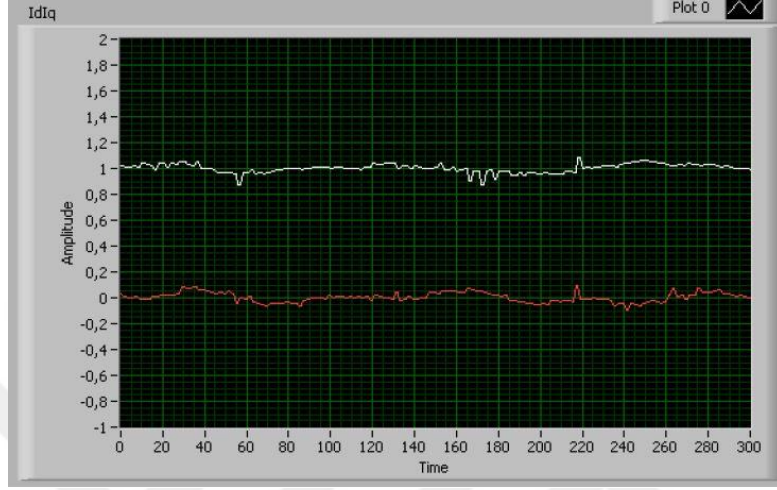


řekil 6.3 Alan etkili algılayıcılara ait a-kapa srelerinin osiloskop ekranı zerinden izlenmesi

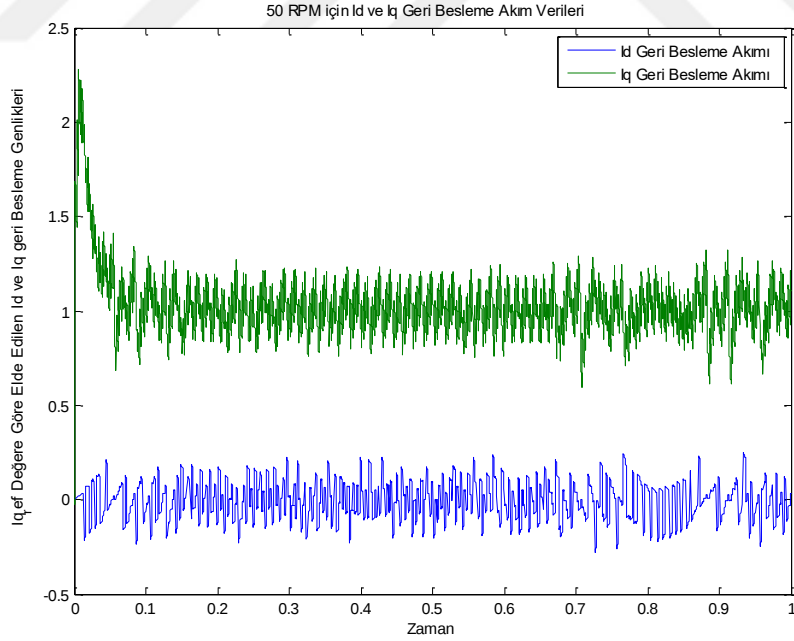
6.1.2. Geri Besleme Akımı Verisinin Doęrulanması

Geri besleme akımları U ve V fazlarından alınan ve rneklenen anlık deęerlerdir. Teorik olarak geri besleme akımı I_{q_ref} 'e eřit olmalıdır. nk dnřmlerde (Clarke, Park) genlik deęiřimi olmaz; sinyallerin faz aıları ve sayıları deęiřir. LabVIEW tabanlı gerek kontrolrden elde edilen geri besleme akım deęerleri, Matlab/Simulink benzetim sonuları ile karřılařtırılarak doęrulanmıřtır. Akımlar aynı I_{q_ref} (1A) deęeri ve hızı iin Matlab Simulink test edilmiř ve LabVIEW VI test programından elde edilen akımlar ile karřılařtırılmıřtır. řekil 6.3, altı konumlu anahtar yapısına giden 3 faz akımlarından elde

edilen I_d ve I_q akım değerini göstermektedir. I_q akımının ortalama değeri 1A dir ve bu değer Matlab benzetim programında ölçülen değere çok yakındır. Ancak Matlab programında gerçek sistem üzerinden alınan değer zaman örneği kadar örnek alınamamıştır. Nedeni ise Matlab programında uzun süreli analizlerin yapılamamasıdır.



a)



b)

Şekil 6.4 50 rpm için geri besleme Id ve Iq akım sinyalleri a) Gerçek sistem LabVIEW arayüzü üzerinden ölçülen değer, b) Matlab/Simulink benzetim sonucu

6.1.3. FFC Algoritmasının Doğrulanması

FFC algoritması PI denetleyiciler çalıştırılmadan ayrı olarak test edilmiş, farklı I_{q_ref} değer ve hızlar için V_q bileşeni bir dizi ölçümleri yapılmıştır. İleri yönlü kontrol algoritması hem LabVIEW platformunda gerçekleştirilmiş hem de Matlab ortamında benzetimi yapılmıştır. Her iki duruma ait sonuçlar Tablo 6.3'te de görüldüğü gibi karşılaştırılarak; doğruluğu test edilmiştir. Test sonuçlarının hata analizi, Eşitlik 6.1 de verilen denkleme göre yapılmıştır.

$$V_{hata} = \frac{|V_{Benzetim}| - |V_{Kontrolör}|}{|V_{Benzetim}|} \times 100 \quad (6.1)$$

Tablo 6.2 deney sonuçları dikkate alındığında ortalama hata oranı V_q değeri için **%21,7588** olarak hesaplanmıştır. Hatanın nedeni çoğunlukla LabVIEW FPGA modül kullanılarak geliştirilen gerçek zamanlı ileri besleme denetleyicisinden kaynaklanmaktadır. Çünkü V_q ölçekleme değerlerinin veri tipi tanımlamasında yeterli hassasiyetin elde edilememesidir. V_d / V_q parametresinin ölçekleme değeri, FPGA modülde “Fixed Point Configuration” tipinde ve Word Length=42 bits, Integer Word length=2 bits olarak tanımlanmıştır. Bu tanımlamadaki hata, genel gerilim düşümü hata oranını da artırmaktadır.

Tablo 6.2 İleri Besleme kontrol algoritması için LabVIEW tabanlı Gerçek kontrolör ile Matlab/Simulink benzetim sonuçlarının karşılaştırılması

I_{q_ref}	Hız RPM	$V_q(V)$ Gerçek kontrolör	$V_q(V)$ Benzetim	V_q % Hata
1	20	0.1219	0.1211	0.6457
	40	0.2256	0.1937	16.4739
	60	0.3356	0.2611	28.5343
	80	0.4455	0.3286	35.5752
	100	0.5554	0.3961	40.2171
	120	0.6654	0.4636	43.5289
2	20	0.1217	0.1849	34.1806
	40	0.2318	0.2524	8.616
	60	0.3420	0.3198	6.9500
	80	0.4521	0.3873	16.7312
	100	0.5613	0.4548	23.4083
	120	0.6712	0.5223	28.5099
3	20	0.1274	0.2436	47.7011
	40	0.2376	0.3111	23.6258
	60	0.3478	0.3785	8.1110
	80	0.4579	0.446	2.6682
	100	0.5181	0.5135	0.8958
	120	0.6111	0.5811	5.1626
4	20	0.1327	0.3023	56.1032
	40	0.2434	0.3698	34.1806
	60	0.3535	0.4372	19.1446
	80	0.4637	0.5047	8.1236
	100	0.5738	0.5722	0.2796
	120	0.6840	0.6397	6.9251
5	20	0.1390	0.3611	61.5065
	40	0.2492	0.4285	41.8436
	60	0.3593	0.4959	27.5459
	80	0.4695	0.5634	16.6667
	100	0.5796	0.6309	8.1312
	120	0.6898	0.6984	1.2314
Ortalama Hata Oranı (%)				21.7588

6.1.4. FOC Algoritmasının Gerçek zamanlı LabVIEW uygulaması ile Matlab/Simulink Benzetimi Sonuçlarının Karşılaştırılması

Alan yönlendirmeli kontrol algoritması için matlab/Simulink benzetiminden ve LabVIEW tabanlı gerçek sistem üzerinden elde edilen veriler karşılaştırılmıştır. Sistem ilk olarak FFC algoritması olmadan test edilmiş, daha sonra FFC algoritması sisteme

dahil edilerek test edilmiş ve sonuçlar karşılaştırılmıştır. Tablo 6.4 sistemin farklı hızlarda, farklı I_{q_ref} değerlerde ve farklı K_p ve K_i parametrelerde Matlab/Simulink benzetiminden elde edilen sonuçları göstermektedir. Bu deneylerde $L_d=L_q=1,3508e-3H$, $R=0,08\Omega$, $\lambda=0,0446 N\Phi$, $K_e=1,36524 Vs/rad$, $V_d/V_q= 0,0721687$ olarak alınmıştır.

Tablo 6.3 Farklı akım değerleri için FOC algoritması Matlab/Simulink Benzetimi sonuçları

I_{q_ref} (A)	$Speed_ref$ (rpm)	$K_p=2,6, K_i=0,2$		$K_p=5,2, K_i=0,2$	
		Sadece PI	PI+ FFC	Sadece PI	PI+ FFC
		I_q (A)	I_q (A)	I_q (A)	I_q (A)
1	20	1.003	1.069	1.021	1.029
	40	0.8261	0.8673	0.9786	1.067
	60	0.8086	0.8524	1.072	1.055
	80	0.7044	0.7161	0.9059	1.055
	100	0.625	0.6588	1.011	1.053
	120	0.3984	0.6497	0.9007	1.023
	140	0.1677	0.493	0.6125	1.001
2	20	1.965	2.05	1.986	2.1
	40	1.901	1.941	2.098	2.07
	60	1.79	1.905	2.039	2.06
	80	1.623	1.865	1.913	2
	100	1.536	1.73	2.091	2.09
	120	1.513	1.454	2.082	2
	140	1.065	1.111	1.723	2.05
3	20	2.94	3.117	3.083	3.01
	40	2.876	2.986	3.051	3.061
	60	2.861	2.931	2.905	3.041
	80	2.841	2.774	2.97	3.056
	100	2.591	2.72	3.117	3.097
	120	2.844	2.535	2.994	3.05
	140	1.594	1.621	2.503	2.59
4	20	3.913	4.215	4.068	4
	40	3.908	4.096	3.985	4.02
	60	3.684	4.07	3.965	4
	80	3.656	3.993	3.921	4.02
	100	3.644	3.825	3.828	4
	120	3.54	3.756	4.077	4.08
	140	1.707	2.079	2.018	3.56
5	20	492	5.175	4.783	5.01
	40	4.839	4.971	5.047	5.09

	60	4.797	4.965	4.931	5.34
	80	4.654	4.925	5.131	5.115
	100	4.554	4.873	5.079	5
	120	4.462	4.864	4.934	5
	140	1.9	2.128	1.755	2.93

Tablo 6.4 FOC Algoritması Sadece PI ve PI ile FCC'nin birlikte kullanılması ile yapılan gerçek zamanlı LabVIEW denetleyici sonuçları

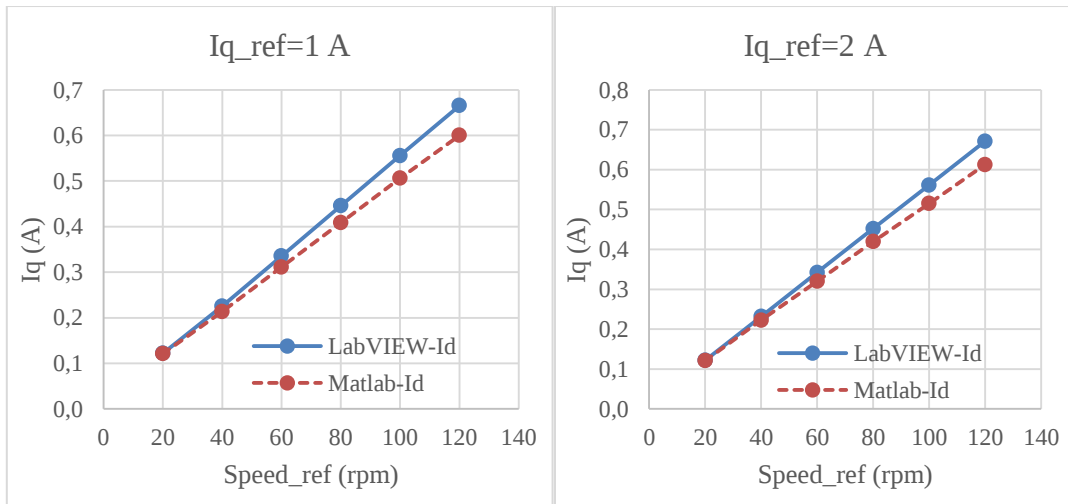
<i>I_{q_ref}</i> (A)	<i>Speed_ref</i> (rpm)	<i>Sadece PI</i>	<i>FCC+PI</i>
		<i>I_q (A)</i>	<i>I_q (A)</i>
1	20	0.9628	1.014
	40	0.99	1.015
	60	1.06	1.018
	80	1.077	1.038
	100	1.01	1.045
	120	0.94	1.044
2	20	1.98	2.17
	40	1.8715	2.014
	60	1.7387	1.8755
	80	1.7664	1.8273
	100	1.724	1.7698
	120	1.7114	1.7275
3	20	3.6	3.64
	40	3.64	3.63
	60	3.63	3.625
	80	3.64	3.662
	100	3.63	3.625
	120	3.64	3.63
4	20	3.635	3.87
	40	3.635	3.7885
	60	3.6407	3.65
	80	3.645	3.625
	100	3.6342	3.625
	120	3.6342	3.625
5	20	3.7526	4.2156
	40	3.6442	3.9287
	60	3.6342	3.8785
	80	3.64	3.8578
	100	3.6384	3.8445
	120	3.6366	3.81

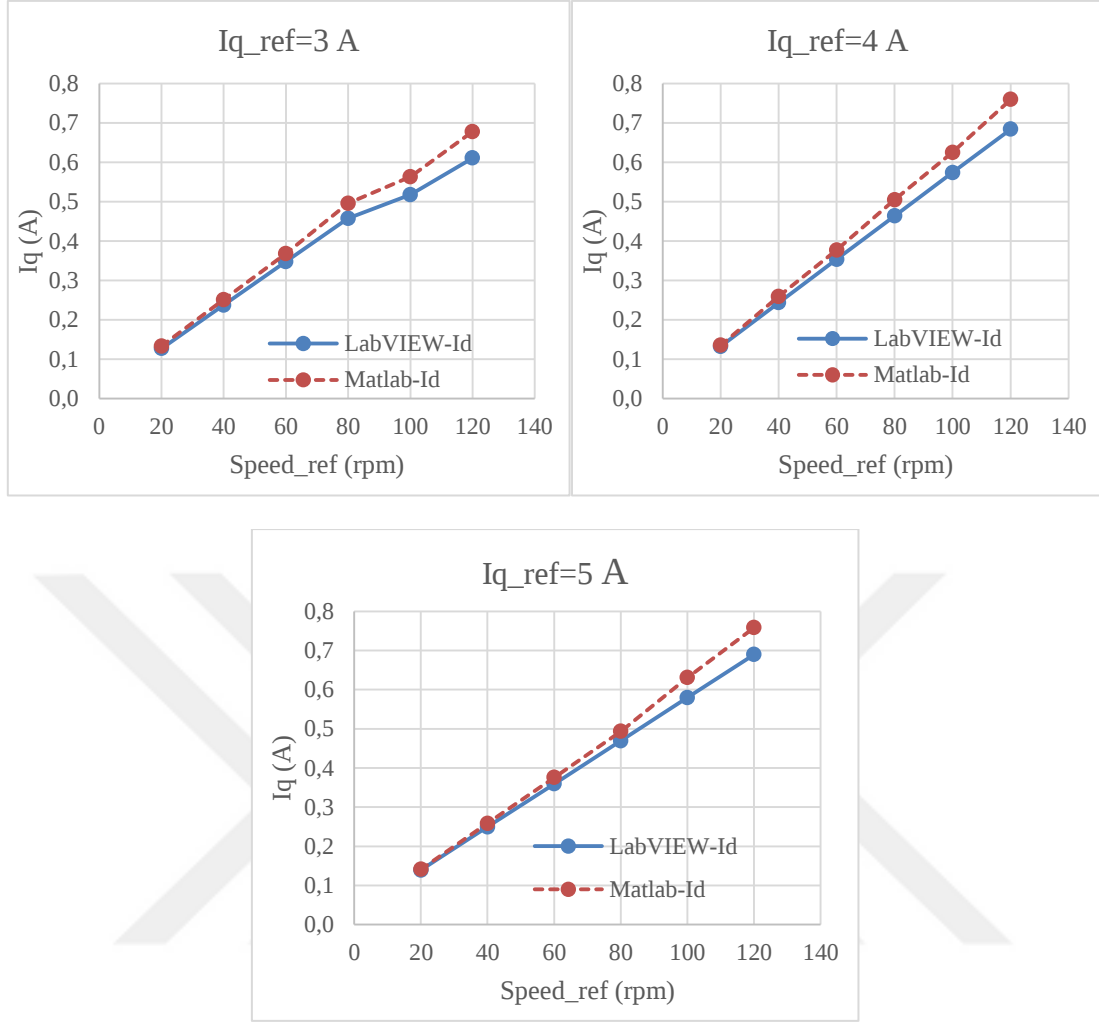
Tablo 6.4 için PI parametreleri ($K_p=5.2$, $K_i=0.2$) olarak alınmış, değerlendirilmesi Sonuçlar bölümünde ayrıntılı bir şekilde anlatılmıştır.

6.1.4.1. Sonuçlar

FFC algoritması için Matlab/Simulink benzetim modelinden ve gerçek sistem üzerinden ölçülerek elde edilen I_{q_ref} değerleri karşılaştırılmıştır. Tablo 6.3'ten de görüldüğü gibi genel olarak hız arttıkça tüm akım değerleri için hata oranı da artmaktadır. Bunun nedeni FFC algoritması bir açık çevrim kontrol algoritması olup; herhangi bir geri besleme kullanmamasıdır. Bu algoritma uygulamada PI denetleyiciye yardımcı olarak hız, kontrol ve yol düzenleyicisi olarak ele alınmış ve PI denetleyicilere paralel bir şekilde bağlanmıştır. PI denetleyici çıkışlarına FFC algoritmasının çıkışları eklenerek clark dönüşümünün girişlerine uygulanmaktadır. Diğer taraftan, V_q hata % değeri, akım değeri ile ters orantılıdır. Akım değeri arttıkça hata değeri azalmaktadır. Motor düşük akımlarda sürülürken, ataleti yenemediğinden geçici rejim süresi artmaktadır. Dolayısı ile, geçici rejim anında üretilen darbe momentleri hata oranını olumsuz etkilemektedir.

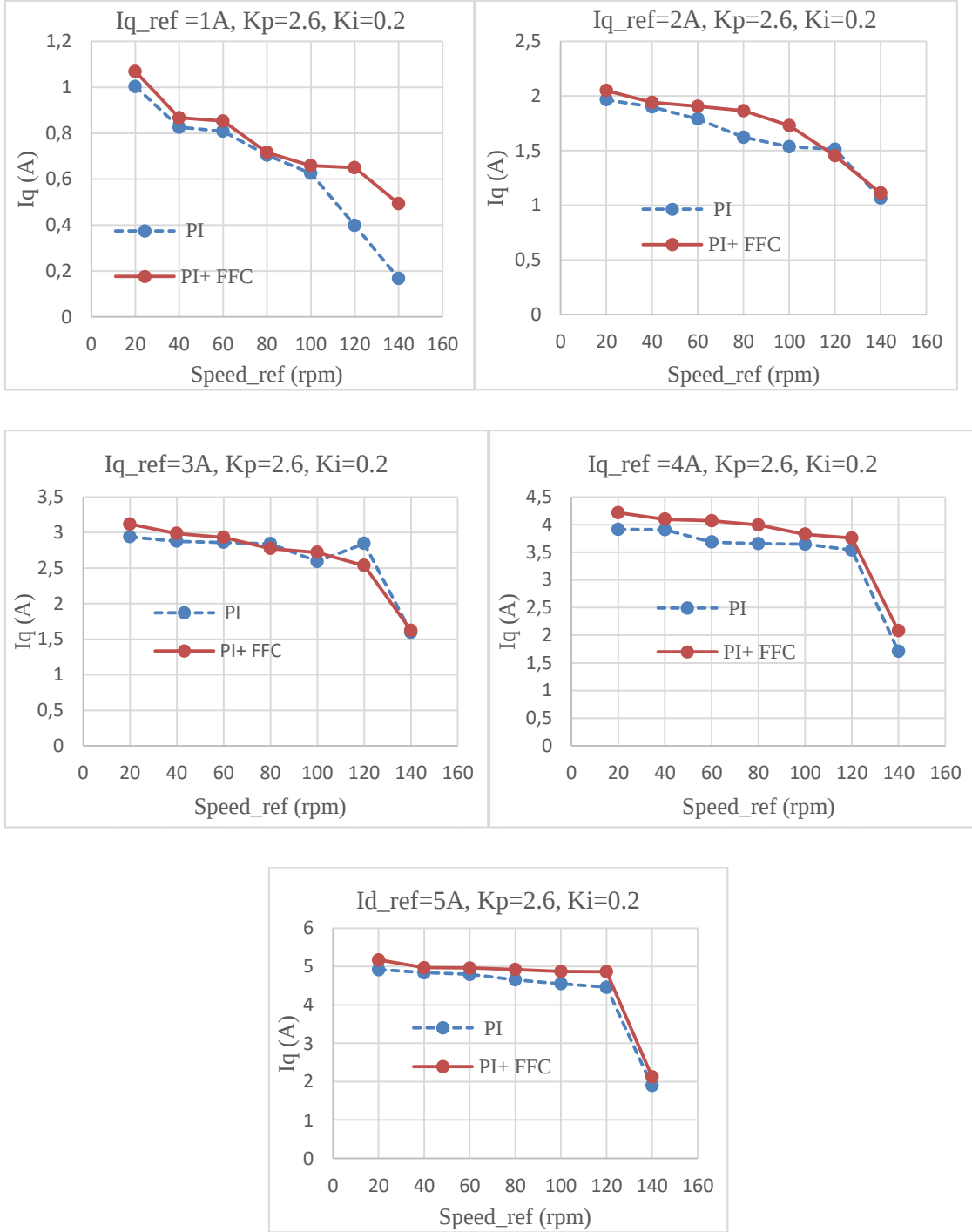
Sistemdeki V_d % hata değeri ise motor dönüş hızı veya sürme akım değerinden bağımsızdır. $I_{d_ref} = 0$ A olarak arayüz üzerinden girildiğinden; V_d % hata değeri üzerinde bir etki yaratmamaktadır. V_d % hata değerini etkileyen tek faktör, sistemin tamamından elde edilen I_d geribesleme akım değeridir. Şekil 6.5 farklı I_{q_ref} değeri ve farklı hızlardaki FFC algoritmasının davranışını göstermektedir.





Şekil 6.5 Farklı akım değerleri için FFC Algoritması LabVIEW Tabanlı Gerçek Kontrolör ile Matlab/Simulink Benzetim Sonuçlarının Karşılaştırılması

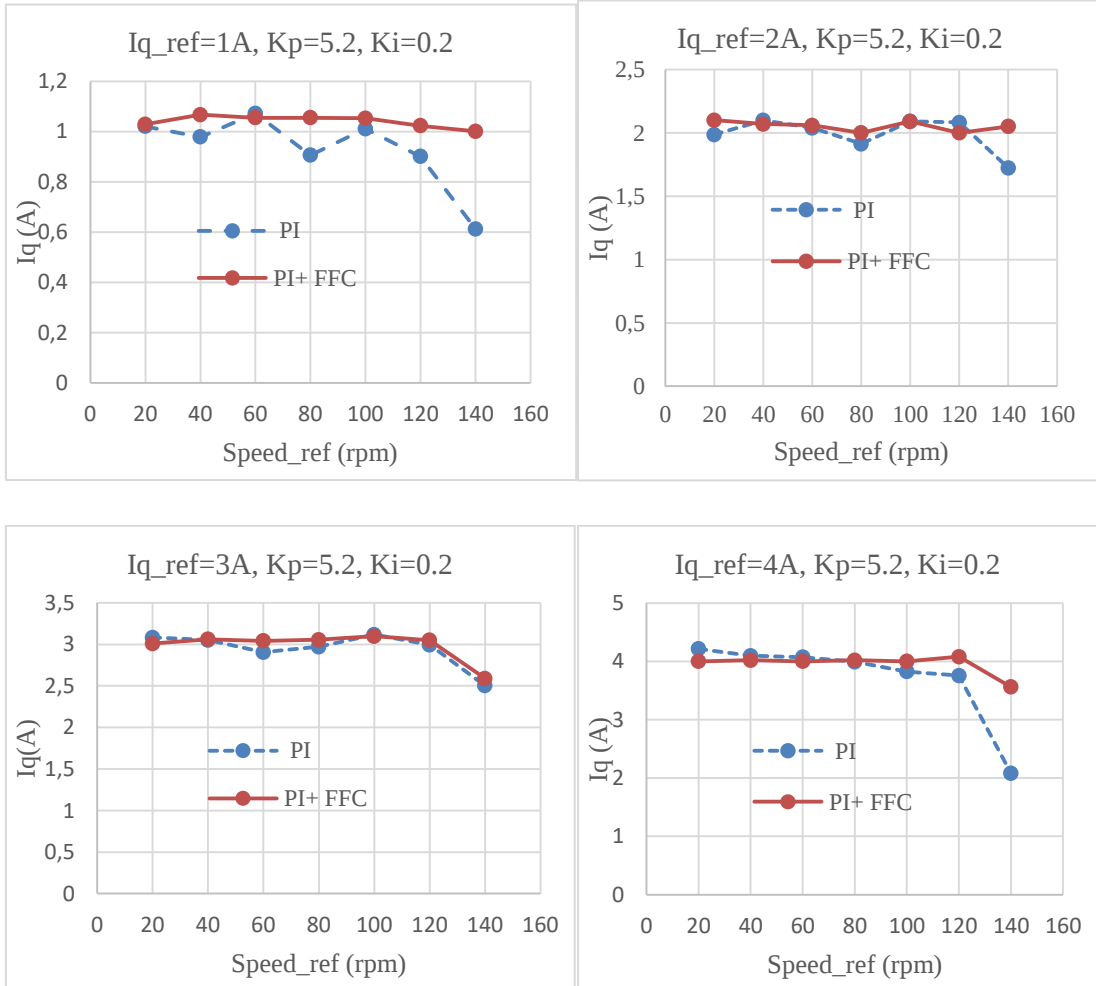
Sadece PI ve PI+FFC algoritmasının birlikte kullanılmasına ait Matlab/Simulink benzetimi Tablo 6.4'te verilmişti. Sistemin benzetimi uygulamadaki verilerin doğruluğu için önemlidir. Bu nedenle tüm sisteme ait (FOC algoritması) uygulama ve benzetim sonuçları grafiklerle analiz edilmiş ve doğruluğu ispatlanmıştır. Şekil 6.6 a'da $K_p=2.6$ ve $K_i=0.2$ değerleri kullanılarak; Şekil 6.6 b de ise $K_p=5.2$ ve $K_i=0.2$ kullanılarak sistemin Matlab/Simulink ortamındaki benzetim sonuçları verilmiştir.

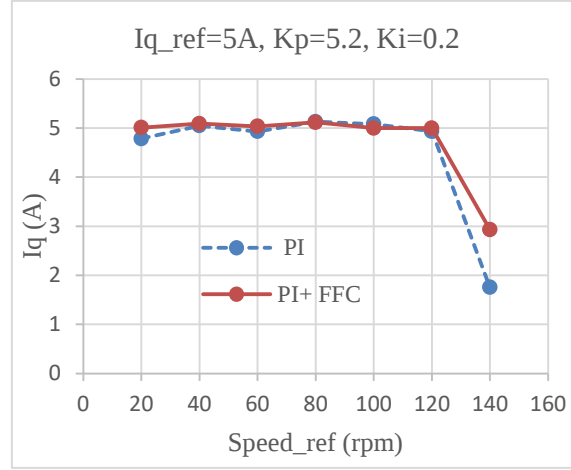


Şekil 6.6 $K_p=2.6$ ve $K_i=0.2$ parametre değerlerine göre PI ve PI+FFC algortimasının birlikte kullanılmasına ait Matlab/Simulink benzetim sonuçları

Şekil 6.6 da verilenler göz önüne alındığında FOC algortimasında yalnızca PI kullanılması yetersiz kalmıştır. 0-120 rpm aralığında PI parametrelerinin değeri, referans akım değerlerine göre değişmektedir. İlk durumlarda ($I_{q_ref}=1A$ ve $I_{q_ref}=2A$) sistemin cevabı yalnız PI varken beklenildiği gibi değildir. Hatta FFC ile PI

birlikte kullanılmasına rağmen sonuç biraz düzelse de yine istenildiği gibi verilen referans değerini karşılamamıştır. Akım referans değerlerinin artırılması ile ($I_{q_ref}=3$, $I_{q_ref}=4$, $I_{q_ref}=5A$) sadece PI varken sistemin cevabı istenilen değere yaklaşmaktadır ama yinede istenilen referans değerini karşılamamaktadır. Aynı durumda ($I_{q_ref}=3$, $I_{q_ref}=4$, $I_{q_ref}=5A$) FFC algoritmasının devreye alınması ile sistem cevabı referans değerini iyi bir şekilde karşılamaktadır. Şekil 6.6 a'dan gözlemlenen bir diğer sonuç ise, PI parametrelerinin belirlenen hız değerleri için uygun olmadığıdır. Bu nedenle Şekil 6.7'de değiştirilmiş PI parametreleri ile FOC algoritmasının benzetimi tekrarlanmıştır.



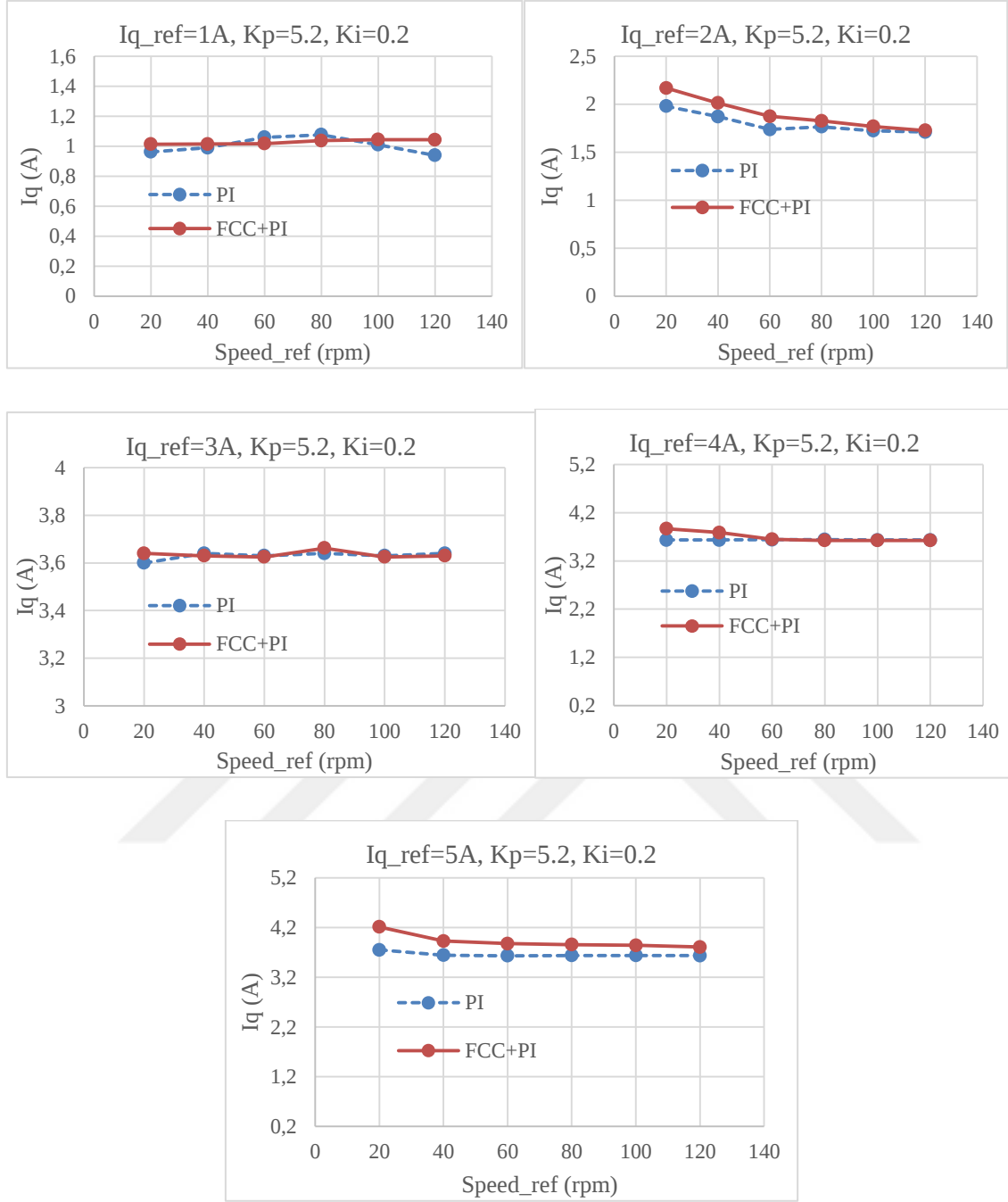


Şekil 6.7 Kp=5.2 ve Ki=0.2 parametre değerleri için sadece PI ve PI+FFC kullanılırken Matlab/Simulink benzetim sonucu

Şekil 6.7, PI parametrelerinin değiştirilmesi sonucu elde edilen FOC algoritmasına ait benzetimi göstermektedir. PI parametrelerinin değiştirilmesi ile 0-120 rpm aralığında motorun kontrolü test edilmiştir. Şekil 6.7 den de görüldüğü gibi yalnızca PI düzenleyicisinin kullanılması Şekil 6.6 a'da olduğu gibi kontrol için yetersiz kalmıştır. Buna karşın PI ile birlikte kullanılan FFC algoritması belirtilen Iq_ref ve rpm aralığında motorun kontrolünü tam olarak sağlamıştır. Tüm Iq_ref değerleri için PI+FFC birlikte kullanımı yalnız PI kullanımına göre oldukça başarılıdır. Sistemin 120 rpm değerini aşınca çalışmaması, yine Kp ve Ki parametrelerine bağlıdır.

Bu tezde farklı PI parametreleri kullanılarak 0 ile 120 rpm aralığında istenilen hızlarda ve 1 ile 5 Nm aralığında istenilen tork değerinde motorun kontrolü sağlanabilmektedir.

FOC algoritmasına ait LabVIEW tabanlı gerçek kontrole ait sonuçlar ise Şekil 6.8'de grafikler halinde verilmiştir. Burada, Matlab/Simulink benzetiminden farklı olarak sadece benzetimde en iyi sonucu veren PI parametre değerleri (Kp=5.2 ve Ki=0.2) LabVIEW tabanlı gerçek uygulamada kullanılmıştır. Kp=2.6 ve Ki=0.2 için test işlemi yapılmamıştır.

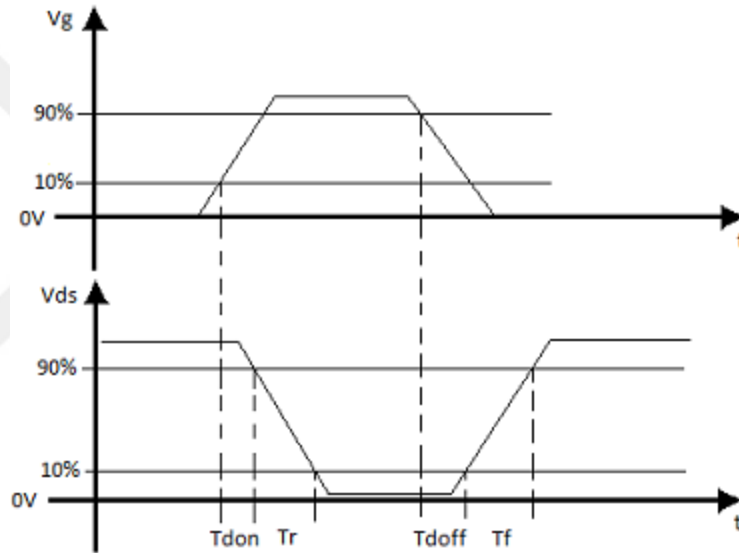


Şekil 6.8 FOC Algoritması LabVIEW tabanlı gerçek uygulamaya ait deney sonuçları

6.1.5. PWM Anahtarlama Frekansı

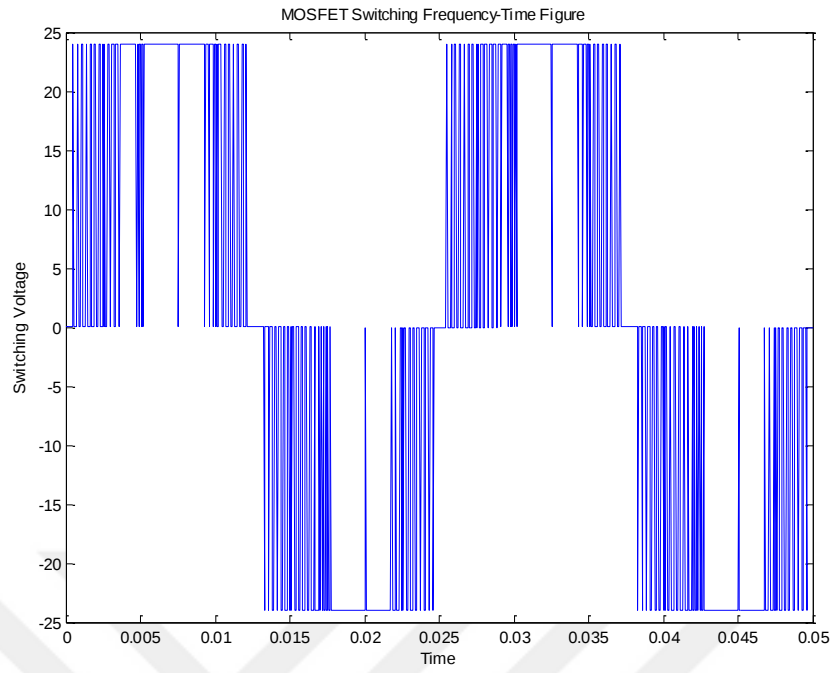
Motor, yükte çalışırken yüksek I_{q_ref} değerinde sistemin distorsiyonlu akımlar ürettiği gözlemlenmiştir. Bu duruma yol açan problem, üç fazlı altı konumlu anahtar köprüsünde kullanılan MOSFET'lerin yüksek verim oranını sağlayabilmeleri için gerekli zamana ihtiyaç duymalarından kaynaklanmaktadır. MOSFET kapı girişlerine

uygulanan iki darbe arasındaki süre, MOSFET'in tamamen açılıp tekrar tamamen kapanması gereken süreden küçük olursa, yükün ihtiyaç duyduğu toplam akım tam olarak motora iletilmez. Şekil 6.9'da bir MOSFET'e ait anahtarlama sinyali görülmektedir. Uygulamada PWM periyodu 2400 ticks olarak ayarlanmıştır. Bu SVPWM üreticinden alınan darbelerin 16.6 KHz olduğu anlamına gelir. En küçük darbe süresine sahip işaretin darbe genişliği 16.6 KHz den çok daha kısa olacaktır. MOSFET'lerin bir anahtarlama periyodu boyunca yeterli miktarda akım sağlayabilmesini garanti etmek için $T_{d(on)}$ süresi; V_g geriliminin yükselmeye başlayarak %10 seviyesine ulaştığı nokta ile Drain-Source geriliminin düşmeye başlayarak %90 seviyesine ulaştığı nokta arasında geçen süre kadar olmalıdır.

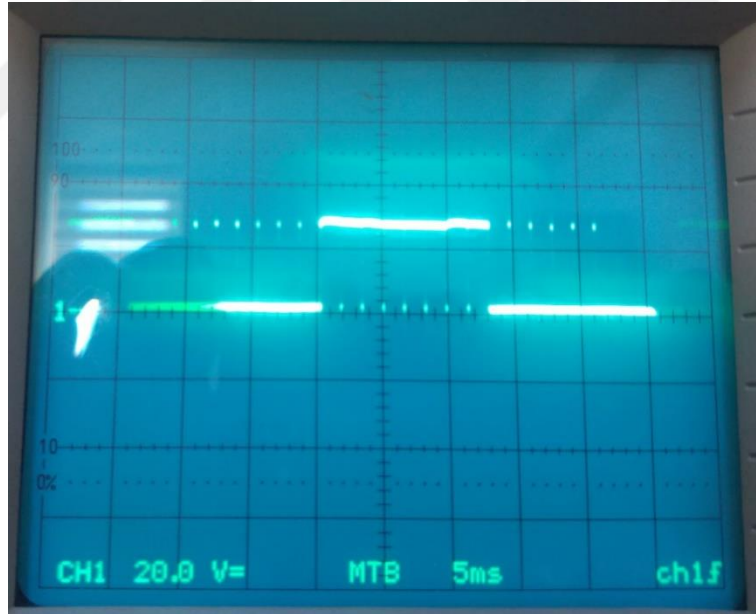


Şekil 6.9 MOSFET anahtarlama sinyali [30]

PWM sinyalinin periyodu, ana programın ön panelinden belirlenmektedir. Şekil 6.10 16.6 KHz lik anahtarlama frekansı için IRFP06 MOSFET'in benzetim ortamında ve gerçek uygulama üzerinden elde edilen anahtarlama sinyalini göstermektedir. Şekil 6.10 den de görüldüğü gibi anahtarlama frekansı 16.6 KHz iken, motor frekansı benzetim ve uygulama için 40 Hz ($1/\approx 5 \text{ div} \cdot 5\text{ms}$) dir.



a)



b)

Şekil 6.10 16.6 KHz (2400 ticks) için MOSFET anahtarlama sinyalleri a) Matlab/Simulink benzetim sonucu, b) Gerçek uygulama sonucu

7. SONUÇ

7.1. Giriş

Bu yüksek lisans tezinde, FSDA motorların akı ve tork kontrolünde FOC algoritmasına ait bir uygulama geliştirilmiştir. İlk olarak FSDA motorlarının çalışması, diğer motorlardan farkı, üstünlükleri ve zayıflıkları, ikinci olarak FOC algoritmasının teorisi ve gerçek uygulamaya aktarılması ve son olarak bu teorinini LabVIEW ortamında geliştirilmesi açıklanmıştır. Yapılan çalışma Matlab/Simulink benzetimi ile desteklenmiş ve doğruluğu test edilmiştir. NI firmasına ait sbRIO 9637 kartı üzerinde gerçekleştirilen bu proje, hızlı bir şekilde birkaç motor parametresi değiştirilerek; farklı motorların sürülmesine yönelik gelecek projelerde bir şablon olarak ta kullanılabilecektir. FOC algoritmasının tamamı kart üzerinde bulunan FPGA’de yürütülmekte, ana bilgisayar (Host PC) kullanıcı arayüz olarak, gerçek zamanlı işletim sistemi (RTOS) ise FPGA ile Host PC arasında veri aktarımı için kullanılmıştır.

7.2. Donanım

Kurulum yapılırken, donanım bileşenlerinin iletişim kurmasının her zaman kolay bir iş olmadığı ağıktı. sbRIO sayısal çıkış terminalinden sürücü katında (inverter) MOSFET leri sürmek için 3.3 V luk seviyenin TTL seviyeye dönüştürülmesi gerekmektedir. Bu yüzden tasarlanan sürücü katına, bir seviye uygunlaştırma devresi de eklenmiştir. PCB tasarımında, farklı akım ve güç değerlerine göre yol kalınlıkları değişmektedir. Bu yüzden özellikle motor akımlarının geçtiği yolların yeterince kalın olmasına, daha zayıf akımlarla beslenen diğer elektronik malzemelerin de fazla yer kaplamaması açısından yollarının yeterince ince olmasına özen gösterilmiştir. Kontrol için gerekli olan rotor açısı (θ) FSDA motorun içerisine yerleştirilmiş alan etkili algılayıcılar yardımıyla elde edildiğinden, bu algılayıcıların düzgün şekilde yerleştirilmesi önemlidir. Sistemde kullanılan motor, algılayıcılar önceden üzerine yerleştirilmiş durumda hazır olarak alınmış ve ilk denemeler de mevcut duruma göre yapılmıştır. Ancak motor üzerine algılayıcılarının tam olarak doğru konumda yerleştirilmemesinden ötürü, motor konumu tayin edilememiş ve sistem çalıştırılamamıştır. Bu alan etkili algılayıcıların konumları gerekli ölçüm ve hesaplamalardan sonra düzeltilmiş ve sistem düzgün çalışır duruma getirilmiştir. Donanım kısmında SVPWM üreticinden alınan DA gerilimler, sürücü kata

uygulanırken; yarı iletken doğrultucu ve çevirici modülü arasındaki ani akımları sınırlamak için bir seri direnç yerleştirilmesi çok önemlidir. Bu yüzden sistemin tamamının korunması için seri 60mV- 25 A lik bir direnç bağlanmış; buna ek olarak bir de akım sınırlama devresi eklenmiştir. Sürücü katındaki MOSFET elemanları, yüksek frekanlarda anahtarlama yaptıklarından dolayı ısınmaktadır. Bu ısınmalar elemanların performansını olumsuz yönde etkilemekte ve motora iletilen gücü düşürmektedir. Bu yüzden bütün anahtarlama elemanlarının soğutulması için PCB tasarımı yapılırken buna dikkat edilmiş ve soğutucular bağlanmıştır.

7.3. Yazılım

Motor kontrolü için gerekli kontrolör algoritmasının tamamı LabVIEW ortamında gerçekleştirilmiş ve “LabVIEW FPGA Module” araçkutusu (Toolbox) yardımıyla .bit file dosyası oluşturularak FPGA ‘ya yüklenmiştir. LabVIEW, grafiksel bir programlama platformu olduğundan, oluşturulan tasarım (blok diyagram) geniş yer kaplamaktadır. Bu nedenle ana program, bölümlere ayrılarak alt programlar (sub vi) şeklinde oluşturulmuştur. Alt programlar ana program (main vi) altına eklenerek, program koşturulmuştur. FOC algoritmasının tamamı FPGA içerisinde koşturulmuştur. Ancak FPGA da ondalık sayıların kullanımındaki kısıtlamalar yüzünden FPGA de kullanılan simgelerin kelime uzunluğu (ondalık ayırıcın öncesi ve sonrası dahil) bit cinsinden (Word length) ve tam sayı kısmının kelime uzunluğu (Integer Word length) iyi bir optimizasyon gerektirmektedir. Bu simgelerin kelime ve tam sayı kelime uzunluğuna ait veriler Ek A’ da verilmiştir. Bu uzunlukların kullanım yerine göre en kısa tutulması gereklidir. Aksi takdirde FPGA bellek hatası (memory out error) verecektir. Bu tezde FPGA ait bu uzunluklar max akı ve tork durumuna göre belirlenmiş ve optimizasyonu yapılmıştır. Bilindiği gibi LabVIEW ve FPGA paralel işlem yeteneğine sahip platformlardır. Bu yüzden sıralı bir program yürütmek isteniyorsa LabVIEW içerisinde bulunan sıralı yapı (Sequence Structure) kullanılmalıdır. Şekil 2.7 de verilen blok diyagramdaki işlemler, sıralı bir şekilde yürütülmesi gerektiğinden ana programda sıralı yapı kullanılmıştır. FPGA giriş/çıkış portlarına ve veri aktarımı için gerekli bellek yapısına erişim, yazılımsal olarak zor olmasına karşın LabVIEW ortamında sürüklenip bırak mantığına dayanan bir yapı bulunmaktadır. İstenilen giriş ya da çıkışa ait portlar uygulama ekranına sürüklenip, bırakılabilmektedir.

7.4. Gelecek Çalışmalar

Tezin geri kalanında, FSDA motorların modellenmesi, benzetimi ve sentezleme konusundaki zorluklar düşünüldü, bu nedenle bu konulara odaklanma kararı ileriki çalışmalara kapı açtı. Bu tezi temel alan bir okuyucu aşağıdaki noktalarda kendisine bu konuda biraz yardım bulacaktır.

İlk olarak; bu tezde kullanılan alan etkili algılayıcılar (hall effect sensors) yerine daha hassas olan kodlayıcılar (encoder) veya çözücüler (resolver) kullanabilir. Tabi bu donanım kullanılırken yazılım kısmında da bazı değişiklikler yapılması gerekmektedir. Çözücülerden alınan veriler, motor konumunun daha hassas bir şekilde elde edilmesini sağlamaktadır. Bu bilginin doğruluğu için güvenilen bir ortamda (Matlab/Simulink gibi) sistemin modellemesini yapmak, çalışmaya başlamadan önce iyi bir hazırlık olabilir. Çözücüler sin ve cos cinsinden iki adet çıkış oluşturmaktadır. Analog olarak elde edilen bu veriler akımlar söz konusu olduğu için I_a ve I_b ye karşılık gelecek şekilde sentezlenir. Bu iki akım, referans değerlerle karşılaştırılması ve PI denetleyiciye iletilmesi için geri besleme olarak d ve q eksene dönüştürülmelidir. Bu dönüşüm sin ve cos gibi trigonometrik fonksiyonları da gerektirdiği için basit bir şekilde yapılacaktır.

İkinci olarak ise; PI'lar için gerekli olan d ve q referans akım değerlerinin trigonometrik dönüşümlerinde olası hatalarından dolayı PI düzenleyicilerini gözden geçirilebilir ve ona göre PI katsayı değerlerini belirleyebilir. Bu bağlamda PI katsayısının belirlenmesi için tekrardan bir birim basamak cevabı veya farklı algoritmalar (yapay sinir ağları, parçacık sürü optimizasyon yöntemi gibi) denenebilir.

Son olarak; bu tezde geliştirilen LabVIEW kodu değiştirilebilir parametrelere ve arayüze sahip olduğu için kullanıcı, sistem hakkında daha fazla bilgi edinmek için bazı parametrelerin değiştirilmesi ile farklı testler yapabilir. Örneğin besleme gerilimini değiştirme, SVPWM de sıfır vektör yerine sağa veya sola hizalama olarak tekrardan kodlar oluşturulabilir.

FSDA motor kontrolünün karmaşıklığı ve bu tezde kullanılan SVPWM yönteminin zor olması nedeniyle, bu konudaki problemlerin çözümünde veya mevcut durumların iyileştirilmesinde bilimsel topluluğun ortak çabası gerekmektedir. Çok farklı makale ve benzer yazılarda sunulan teoriyi temel alan sonuçlara göre bu tezin, farklı benzetim

programları ile doğrulanmış olması ve farklı güçteki aynı tip motorlara uygulanılabilme potansiyelinin görülmesi nedeniyle bu ortak çabaya bir miktar katkıda bulunabilme iddiası vardır. Kontrol edilen motoru görmek ve mühendisin nihai hedefi olan denetleyicinin sentezine adım atmak için iyi bir referans sağlayabilecektir.



Kaynaklar

- [1] I. Topaloglu, F. Korkmaz, H. Mamur, R. Gurbuz, "Closed-Loop Speed Control of PM-BLDC Motor Fed by Six Step Inverter and Effects of Inertia Changes for Desktop CNC Machine", *elektronika ir elektrotechnika*, issn 1392-1215, vol. 19, no. 1, 2013
- [2] National Instruments," The LabVIEW Fixed-Point Data Type Part 1- Fixed-Point 101", *forums.ni.com*, Via <http://www.ni.com/newsletter/50303/en/#toc1>, Consulted on 22 April 2015
- [3] H. Hashimoto, T. Nakayama, S. Kondo, F. Harashima, "Practical variable structure approach for brushless servo motor control-practical implementation of DSP", *Power Electronics Specialist Conference*, vol. 1, pp. 207-213, 1988
- [4] O. Al-Ayasrah, T. Alukaidey, G. Pissanidis, "DSP Based N-Motor Speed Control of Brushless DC Motors Using External FPGA Design", *Industrial Tecnology IEEE International Conference*, Dec.15-17pp. 627-631, 2006
- [5] Ehsan Drayabeigi and Gholamreza Arab Markadeh, "Speed Control of Brushless DC Motors Using Emotional Intelligent Controller", *Intelligence Systems in Electrical Engineering Journal*, 4th year, No. 4, Winter 2014
- [6] Matheus Alexandre Bevilaqua and Ademir Nied, José de Oliveira, "Labview FPGA FOC Implementation for SynchronousPermanent Magnet Motor Speed Control", *IEEE Conference*, 2014
- [7] Anders KRONBERG, "Salient Pole Motor InverterDesign with Implementation of Space Vector Modulation", *Master Theses, Uppsala Universitet, Department of Engineering Sciences* December 2012
- [8] Geoffrey De Smaele, Matthias Verschelden, "Control of an electric machine with permanent magnets", *Master Theses, Faculty of Engineering and Architecture, Universiteit Gent*, Academic year 2014-2015
- [9] Magnus Fredrixon, "Evaluation of a system for Rapid Control Prototyping in the field of robot control", *Master of Science Thesis, KTH Industrial Engineering and Management Machine Design*, Stockholm, Sweden 2009.
- [10] Bengi Tolunay, "Space Vector Pulse Width Modulation for Three-Level Converters- a LabVIEW Implementation" *Electrotechnic Master Programme in Electrical Engineering, Uppsala Universitet*, February 2012.
- [11] Vorgelet Von mgrinz, Pawel Szczupak ausWarschau, Polen, "Rapid Prototyping System for Control of Inverters and Electrical Drives" *Master Thesis of Science, Wuppertal*, 2008
- [12] Shehryar Khan, Ruxandra Mihaela Botez and Teodor Lucian Grigorie, "A new method for tuning PI gains for position control of BLDC motor based wing morphing

actuators”, AIAA Modeling and Simulation Technologies Conference, Dallas, TX, 22-26 June 2015

[13] Ling Xu, Jian-Guo Song, Qiang-Qiang Lin, “Brushless DC Motor Speed Control System Simulink Simulation” IEEE International Conference on Power and Renewable Energy College of Electronic Information and Control Engineering Beijing University of Technology Beijing, China 2016

[14] E Elakkia, S Anita, R Girish Ganesan and S Saikira “Design and Modelling of BLDC Motor for Automotive Application” National Level Technical Conference P&E-BiDD, 1 March 2015

[15] Surendra Singh Pate I; B.A. Botre; Krishan K.; Kausha, Samarth S; S.A. Akbar; Yogesh Biradar and Prabhu K.R, “Modelling and Implementation of Intellegent Commutation System for BLDC Motor Underwater Robotics App.” 1st IEEE International Conference on Power Electronics. Intelligent Control and Energy Systems ICPEICES-2016

[16] Manish S Trivedi and Ritesh Kumar Keshri, “Trajectory based Predictive Current Control for Permanent Magnet Brushless DC Motor”, Visvesvaraya National Institute of Technology, Nagpur, India, IEEE, 2016

[17] Salih Baris Öztürk, “MODELLING, SIMULATION AND ANALYSIS OF LOW-COST DIRECT TORQUE CONTROL OF PMSM USING HALL-EFFECT SENSORS”, Submitted to the Office of Graduate Studies of Texas A&M University in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE, December 2005.

[18] <https://tr.pinterest.com/pin/608689705856865397/> 10-10-2017.

[19] Digi-Key, “Why and How to Sinusoidally Control Three-Phase Brushless DC Motors”, Contributed by Digi-Key’s North American Editors 17/01/2017

[20] Çetin GENÇER, “vektör kontrollü fırçasız da motorun iki seviyeli evirici ile uzay vektör modülasyonu”, Doğu Anadolu Araştırmaları, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2015.

[21] Vahid Behjat , Mehrdad Hamrahi, “Dynamic modeling and performance evaluation of axial flux PMSG based wind turbine system with MPPT control” Department of Electrical Engineering, Engineering Faculty, Azarbaijan Shahid Madani University, Tabriz, Iran. Ain Shams Engineering Journal 5, 1157–1166, 2014.

[22] Texas Instruments Europe, “Field Oriented Control of 3-Phase AC-Motors”, Literature Number: BPRA073, February 1998

[23] J.L. Kirtley. Jr, “Permanent Magnet brushless DC Motors”, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science 6.061 Introduction to Power Systems, 2003

- [24] Maria BORGSTRÖM, “Case study a Rapid Control Prototyping System Based on Xilinx System Generator”, Master’s Thesis at KTH/ICT and ABB AB, 2010
- [25] Pollefliet, J. Elektronische vermogencontrole. Volume 2: Elektronische motorcontrole. Ghent: Academia Press, seventh edition, 2011.
- [26] NI, <http://search.ni.com/nisearch/app/main/p/bot/no/ap/global/lang/en/pg/1/q/foc/>, <http://www.ni.com/en-tr/support/model.sbrio-9637.html>
- [27] http://zone.ni.com/reference/en-XX/help/371599M-01/lvfpgaconcepts/pfi_comm/
- [28] Binge Motor Technical Document, “www.bingez.com”, Sales@bingez.com, 2011.
- [29] Allegro, “Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor”, Date of status change: June 5, 2017
- [30] Joakim Eriksson, Luciano Hermansen, “Rapid Prototyping development and evaluation of Field Oriented Control using LabVIEW-FPGA” , Master Thesis ABB Corporate Research, Maladelen University, IDT, 28.01.2011

EK A:**Sabit Nokta Gösterimi (Fixed Point Representation)**

	Veri Uzunluğu	Tamsayı Uzunluğu	Maks/Min	Delta Değeri
$I_{A(\text{memory})}$	12	12	4095 / 0	1
$I_{B(\text{memory})}$	12	12	4095 / 0	1
θ	28	2	4 / 0	1.49012E-8
$\sin(\theta)$	16	2	1.999939 / -2	6.10352E-5
$\cos(\theta)$	16	2	1.999939 / -2	6.10352E-5
$I_{A(\text{ölçeklenen})}$	20	5	15.999969 / -16	3.05176E-5
$I_{B(\text{ölçeklenen})}$	20	5	15.999969 / -16	3.05176E-5
$V_{d(\text{ölçeklenen})}$	42	2	2 / -2	9.09495E-13
$V_{q(\text{ölçeklenen})}$	42	2	2 / -2	9.09495E-13
I_{α}	20	5	15.999969 / -16	3.05176E-5
I_{β}	20	5	15.999969 / -16	3.05176E-5
I_d	20	5	15.999969 / -16	3.05176E-5
I_q	20	5	15.999969 / -16	3.05176E-5
V_d	16	3	3.999878 / -4	0.00012207
V_q	16	3	3.999878 / -4	0.00012207
I_{r1}	20	5	15.999969 / -16	3.05176E-5
I_{r2}	20	5	15.999969 / -16	3.05176E-5
I_{r3}	20	5	15.999969 / -16	3.05176E-5
$\text{Speed}_{\text{Ref}}$	32	16	32767,9998 / -32768	1.52588E-5
I_{q_ref}	20	5	15,999969 / -16	3.05176E-5
I_{d_ref}	20	5	15,999969 / -16	9.53674E-7
$I_a_I_b$	20	5	15.999969 / -16	3.05176E-5
V_d/V_q	42	2	2 / -2	9.09495E-13
V_{d_manuel}	16	3	3.999878 / -4	0.00012207
V_{q_manuel}	16	3	3.999878 / -4	0.00012207
$\text{PWM}_{\text{PeriodTicks}}$	16	16	65535 / 0	1
U_{16}	16	16	65535 / 0	1
$dI-dT$	32	12	2047.999999 / -2048	9.53674E-7
$\text{Speed}_{\text{FIFO}}$	16	16	65535 / 0	1

EK B:**sbRIO-9637 Pin Atamaları**

İsim	Port	Açıklama
Hall_a	DI0	Alan Etkili Algılayıcı Okuma
Hall_b	DI1	Alan Etkili Algılayıcı Okuma
Hall_c	DI2	Alan Etkili Algılayıcı Okuma
U_Phase_Ia	AI0	ACS758 ile Geri emk değerinin okunması
V_Phase_Ib	AI1	ACS758 ile Geri emk değerinin okunması
Rotor Açısı (θ)	AI2	Rotor açısının okunması
PWM_SYNC	DIO3	Akım okuma ve PWM Senkronizasyonu
PWM_U_HIGH	DIO9	U fazı üst transistör bağlantısı
PWM_U_LOW	DIO10	U fazı alt transistör bağlantısı
PWM_V_HIGH	DIO11	V fazı üst transistör bağlantısı
PWM_V_LOW	DIO12	V fazı alt transistör bağlantısı
PWM_W_HIGH	DIO13	W fazı üst transistör bağlantısı
PWM_W_LOW	DIO14	W fazı alt transistör bağlantısı
Inverter_ON_OFF	DIO15	Sürücü devresini açma kapama anahtarı

ÖZGEÇMİŞ

Güner TATAR; 1989 Kahramanmaraş/Elbistan doğumludur. İlkokulunu Geçit köyü ilk öğretim okulunda, Orta ve Lise öğrenimini Elbistanda tamamlamıştır. Elbistan Mesleki ve Teknik Anadolu Lisesi (Elektronik) bölümünü 2007 yılında tamamladıktan sonra 2009 yılında Marmara Üniversitesi Elektronik ve Haberleşme Eğitimi (İngilizce) Bölümünü kazanmış ve 2014 yılında mezun olmuştur. Aynı yıl içerisinde Marmara Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümünde Yüksek Lisansa başlamış, ders dönemini bitirdikten sonra 2015-2016 yılları arasında Malatyada İnönü Üniversitesi Elektrik ve Elektronik Mühendisliği bölümünde ikinci lisansını bitirdikten sonra Marmara üniversitesine dönmüş ve yüksek lisans eğitimine tez döneminde devam etmiştir. 2017 yılında yüksek lisans eğitimini tamamlayan Güner TATAR iyi sayılabilecek derecede İngilizce ve LabVIEW, Altium, FPGA gibi bilgisayar ve gömülü sistem programları bilmektedir. Hali hazırda Fatih Sultan Mehmet Vakıf Üniveristesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır. Yarı iletken güç elemanları, Güç Elektroniği, Elektrik Makinaları, Yenilenebilir Enerji Sistemleri, Elektrikli Araçlar ilgi alanlarıdır.