

T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ



5V-12V YÜKSELTİCİ DC-DC DÖNÜŞTÜRÜCÜ İLE
DC MOTOR HIZ KONTROLÜ

LİSANS BİTİRME PROJESİ

Ahmet Nuri KESER

Kazım Miraç GÜNAY

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

T.C.
FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ



**5V-12V YÜKSELTİCİ DC-DC DÖNÜŞTÜRÜCÜ İLE
DC MOTOR HIZ KONTROLÜ**

LİSANS BİTİRME PROJESİ

Ahmet Nuri KESER

2021241009

Kazım Miraç GÜNAY

2121241357

Danışmanı: Prof. Dr. Hilmi ÜNLÜ

Bölümü Teslim Edildiği Tarih: 23 Haziran 2025

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

ÖNSÖZ

Günümüz mühendislik uygulamalarında enerji verimliliği ve hassas kontrol teknolojileri, modern sistemlerin geliştirilmesinde büyük önem taşımaktadır. Bu çalışma, düşük volajlı bir enerji kaynağını yüksek volaj seviyesine yükselten bir güç elektroniği devresinin tasarıımı ve bu devrenin, elektrik motorlarının hız kontrolü uygulamalarında kullanımını konu edinmiştir.

Bu tür sistemler, başta yenilenebilir enerji kaynaklarının yönetimi, endüstriyel otomasyon, elektrikli araçlar ve taşınabilir elektronik cihazlar olmak üzere, geniş bir kullanım alanına sahiptir. Verimliliğin artırılması ve kompakt tasarım gibi avantajları sayesinde bu devreler, enerji dönüşüm sistemlerinde vazgeçilmez bir rol oynamaktadır. Ancak, yüksek anahtarlama kayıpları, elektromanyetik parazitler ve karmaşık tasarım süreçleri gibi bazı dezavantajları, sistem tasarımında dikkat edilmesi gereken noktalar arasında yer alır. Bu bağlamda, çalışmada hem teorik analiz hem de pratik uygulama süreçlerine yer verilerek, bu zorlukların üstesinden gelinmesi hedeflenmiştir.

Proje sürecinde, güç elektroniği sistemlerinin temel prensipleri ve kontrol yöntemleri üzerine kapsamlı bir inceleme yapılmış; elde edilen teorik bilgiler, pratik uygulama ile birleştirilmiştir. Tasarlanan sistemde, verimliliğin artırılması, kararlılık sağlanması ve motor kontrolünün hassas bir şekilde gerçekleştirilmesi hedeflenmiştir. Bu bağlamda, çalışma hem akademik bir öğrenim sürecine katkıda bulunmuş hem de gelecekteki mühendislik uygulamaları için bir temel oluşturmuştur.

Bu çalışmayı gerçekleştirmeye sürecinde, bilgi ve deneyimleriyle bize yol gösteren danışman hocamız Sayın Prof. Dr. Hilmi ÜNLÜ 'ye teşekkür ederiz. Ayrıca, süreç boyunca desteklerini esirgemeyen ailelerimize ve ekip arkadaşlarımıza şükranlarımızı sunarız.

Haziran 2025

Ahmet Nuri KESER

Kazım Miraç GÜNAY

İÇİNDEKİLER

	<u>Sayfa</u>
KISALTMALAR.....	6
TABLO LİSTESİ	7
ŞEKİL LİSTESİ	8
ÖZET.....	10
ABSTRACT	11
1.GİRİŞ.....	12
1.1 Tezin Amacı	13
1.2 Literatür Özeti.....	13
1.2.1 DC-DC Boost Converter Tanımı	13
1.2.2 PID Controller Tanım	15
1.2.3 Arduino UNO Tanım	16
1.2.4 DC Motor Tanım.....	17
1.2.5 Encoder Tanımı.....	18
2. TEORİK ALTYAPI VE ANALİZLER.....	19
2.1 DC-DC Boost Converter	19
2.1.1 Boost Converter Çalışma Prensibi	19
2.1.2 Boost Converter Matematiksel Modeli	20
2.1.3 Boost Converter İçin Komponent Seçimi	25
2.1.3.2 Anahtar Seçiminde Dikkat Edilmesi Gerekilenler	26
2.1.3.3 Diyot Seçiminde Dikkat Edilmesi Gerekilenler	26
2.1.3.4 Kondansatör Seçiminde Dikkat Edilmesi Gerekilenler.....	27
2.1.4 Boost Converter Simülasyonu	27
2.2 PID Kontrol Sistemi	28
2.2.1 PID Kontrol Sisteminin Çalışma Prensibi	28
2.2.2 PID Kontrol Sisteminin Matematiksel Modeli	29
2.3 Arduino UNO	32
2.3.1 Arduino ile PWM Sürmek	32
2.3.2 Arduino UNO'nun Projedeki Kullanımı.....	32
2.4 Enkoderli DC Motor.....	34
2.4.1 DC Motor Kullanım Alanları.....	34
2.4.2 Enkoder Çalışma Prensibi.....	35
2.4.3 Arduinio ile Enkoderli DC motor ve L298N Motor Sürücüsü Kullanımı	36
3. PROJE UYGULAMASI.....	37
3.1 Hesaplamlar	37
3.1.1 Dönüşüm Oranı (Duty Cycle) Hesabı.....	37
3.1.2 Giriş Akımı Hesabı	37
3.1.3 Anahtarlama Frekansı ve Sürelerinin Hesabı	38
3.1.4 Endüktans Hesabı.....	38
3.1.4.1 Ripple Akımı Hedefinin Belirlenmesi	38
3.1.4.2 Endüktansdaki “Enerji Biriktirme” Akım Değişimi.....	39
3.1.4.3 Endüktanstaki “Enerji Verme” Akım Değişimi	39
3.1.5 Çıkış Kondansatörü Seçimi.....	39

3.1.5.1 Maksimum Kabul Edilebilir Çıkış Dalgalanması.....	40
3.1.5.2 Kondansatör Seçimi.....	40
3.2 Kullanılan Bileşenler ve Gerekçeleri.....	40
3.3 Blok Diyagram	42
3.4 Proteus Simülasyonu	43
3.5 Yazılım ve Kod Açıklamaları.....	45
3.5.1 Sabit Tanımlamaları ve Pin Atamaları.....	45
3.5.2 Interrupt Servis Rutinleri	47
3.5.3 DC Motor Kontrol Sisteminin Başlatılması ve Yükseltici Devrenin Yapılandırması..	48
3.5.4 Ana Döngüde Besleme Gerilimi Kontrolü	49
3.5.5 PID Döngüsü ve Hız Ölçümü	50
3.5.6 Seri Monitör ve Serial Plotter Çıktıları.....	51
3.6 Gerçek Sistem Uygulaması ve Sonuçlar	53
4. SONUÇ VE ÖNERİLER	61
4.1 Çalışmanın Uygulama Alanları	61
4.2 Çalışmanın Geliştirilmesi için Öneriler.....	62
KAYNAKLAR.....	63
ŞEKİL KAYNAKLARI.....	66
EKLER.....	67

KISALTMALAR

AC: Alternative Current (Alternatif Akım)

CCM: Continuous Current Mode (Sürekli İletim Modu)

C: Capacitor (Kapasite)

DC: Direct Current (Doğru Akım)

DCM: Discontinuous Current Mode (Kesintili İletim Modu)

EMI: Electromagnetic Interference (Elektromanyetik Girişim)

GND: Ground (Toprak)

I: Amper (Akım)

IDE: Integrated Development Environment (Entegre Geliştirme Ortamı)

IN: Input (Giriş Ucu)

L: Inductance (Endüktans)

LDO: Low Dropout Regulator (Düşük Düşümlü Regülatör)

MOSFET: Metal Oxide Semiconductor Field Effect Transistör

(Metal-Oksit Yarı İletken Alan Etkili Transistör)

PCB: Printed Circuit Board (Baskı Devre Kartı)

PID: Proportional-Integral-Derivative (Oransal-İntegral-Türevsel)

PWM: Pulse Width Modulation (Darbe Genişliği Modülasyonu)

RPM: Revolution Per Minute (Dakikadaki Devir ayısı)

V: Volt (Gerilim)

W: Watt (Güç, Enerji)

TABLO LİSTESİ

Sayfa

Tablo 1: K_P , K_I ve K_D Değerlerinin Grafik Üstündeki Yaklaşık Değişimleri.	31
---	----

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1: Temel bir DC-DC boost converter devre şeması.....	13
Şekil 2: Arduino UNO mikrodenetleyici kartı.....	16
Şekil 3: Enkoderli bir DC motor.....	17
Şekil 4: DC bir motor üzerindeki enkoder.	18
Şekil 5: Anahtarın iletim durumu (kapalı mod).....	19
Şekil 6: Anahtarın kesim durumu (açık mod).....	19
Şekil 7: $D = 2/3$ için (V) gerilimin ve (t) zaman aralığında çalışma periyodu grafiği.	21
Şekil 8: Endüktans akımının artma ve azalma grafiği.....	22
Şekil 9: Endüktans gerilim grafiği.....	22
Şekil 10: Anahtarlama esnasında, transistörün üzerindeki (V_T) geriliminin grafiği.....	23
Şekil 11: Anahtarlama esnasında, diyonet üzerindeki (V_D) geriliminin grafiği.....	23
Şekil 12: Kondansatördeki gerilimin dolma-boşalma grafiği.....	25
Şekil 13: Kondansatör üzerindeki (I_c) akım grafiği.....	25
Şekil 14: 5V-12V DC-DC boost converter grafiği (PSIM).....	27
Şekil 15: Giriş geriliği 5V DC olan boost converter simülasyonunun (I_g), ($I_{ç}$) ve ($V_{ç}$) değerlerinin grafiği.....	28
Şekil 16: PID kontrol sistem modeli.....	28
Şekil 17: Arduino UNO üzerindeki kullanılacak pinler.....	34
Şekil 18: Rotasyonel enkoder modeli.....	35
Şekil 19: L298N DC motor sürücü kartı.....	36
Şekil 20: Proje'nin blok diyagramı.....	42
Şekil 21: DC-DC boost converter devre şeması.....	43
Şekil 22: Geri besleme devre şeması.....	44
Şekil 23: Geri beslemeli DC-DC boost converter devresi şematiği.....	45

Şekil 24: DC-DC boost converter devre şematiği (Altium Designer).....	54
Şekil 25: PCB'nin 2D hali.....	54
Şekil 26: PCB'nin 3D hali (Yukardan bakış).....	55
Şekil 27: PCB'nin 3D hali (Ön açıdan bakış).....	55
Şekil 28: PCB'nin 3D hali (Arka açıdan bakış).....	55
Şekil 29: Üretilen PCB.....	56
Şekil 30: Dizgisi yapılan PCB (Ön yüz).....	56
Şekil 31: Dizgisi yapılan PCB (Arka yüz).....	56
Şekil 32: Boost converter çıkış gerilimi ve osiloskop çıktısı.....	57
Şekil 33: Delikli kart üzerinde bir araya getirilen proje (ön yüz	58
Şekil 34: Delikli kart üzerinde bir araya getirilen proje (arka yüz).....	58
Şekil 35: Giriş ve çıkışa ait gerilim çıktıları.....	59
Şekil 36: Osiloskop çıktısı.....	59
Şekil 37: 100 RPM dönmesi istenen DC motorun hedef ve anlık RPM grafiği	60
Şekil 38: Ters yönde 200 RPM dönmesi istenen DC motorun hedef ve anlık RPM grafiği	60

5V-12V YÜKSELTİCİ DC-DC DÖNÜŞTÜRÜCÜ İLE DC MOTOR HİZ KONTROLÜ

ÖZET

Bu çalışma, 5V-12V aralığında çalışan bir DC-DC boost converter tasarımları ve bu dönüştürücünün DC motor hız kontrolü üzerindeki potansiyelini araştırmaya yönelik teorik analizler ve hesaplamaları kapsamaktadır. Boost converter, düşük giriş gerilimini artırarak çeşitli uygulamalarda enerji verimliliği sağlayan bir devredir. Projede, bu sistemin temelleri araştırılarak enerji dönüşüm teknolojileri üzerine kapsamlı bir literatür çalışması gerçekleştirılmıştır.

Projenin ilk aşamasında boost converter devresinin çalışma prensipleri ve matematiksel modelleri detaylı şekilde incelenmiştir. Endüktans, MOSFET, diyon ve kondansatör gibi temel bileşenlerin seçiminde dikkat edilmesi gereken kriterler belirlenmiş ve teorik hesaplamalarla bileşen değerleri optimize edilmiştir. Ayrıca, sürekli iletim modu (CCM) ve kesintili iletim modu (DCM) altında devrenin davranışları analiz edilmiştir.

İkinci aşamada, DC motor hız kontrolüne yönelik PID kontrol teorisi incelenmiş ve bu kontrolcülerin matematiksel modelleri üzerinde çalışılmıştır. Arduino UNO mikrodenetleyici ve PWM sinyallerinin motor hız kontrolünde nasıl kullanılacağına dair teorik bir altyapı oluşturulmuştur. Enkoderlerden alınacak veriler ile kapalı döngü kontrol sistemlerinin sağladığı avantajlar araştırılmıştır.

Bu proje, boost converter ve PID kontrol sistemleri üzerine yapılan teorik çalışmalarla enerji dönüşümü ve hassas kontrol sistemlerinin temellerini araştırmayı amaçlamaktadır. Çalışma, ileride yapılacak olan uygulama ve prototip geliştirme süreçleri için sağlam bir teorik altyapı oluşturmayı hedeflemektedir.

DC MOTOR SPEED CONTROL WITH 5V-12V BOOST DC-DC CONVERTER

ABSTRACT

This study focuses on the design of a 5V-12V DC-DC boost converter and its potential application in DC motor speed control through theoretical analyses and calculations. The boost converter is a circuit that increases low input voltage, providing energy efficiency in various applications. Within the scope of this project, a comprehensive literature review on energy conversion technologies has been conducted to establish a strong theoretical foundation for the system.

In the first phase of the project, the operating principles and mathematical models of the boost converter were examined in detail. Key criteria for selecting components such as the inductor, MOSFET, diode, and capacitor were identified, and theoretical calculations were used to optimize their values. The behavior of the circuit under continuous conduction mode (CCM) and discontinuous conduction mode (DCM) was also analyzed.

In the second phase, the theory of PID control for DC motor speed regulation was studied, and the mathematical models of these controllers were explored. A theoretical framework was developed on how the Arduino UNO microcontroller and PWM signals can be used for motor speed control.

The advantages of closed-loop control systems utilizing encoder feedback were also investigated. This study aims to explore the fundamentals of energy conversion and precision control systems through theoretical research on Boost Converters and PID control systems. The findings provide a robust theoretical basis for future application and prototype development processes.

1.GİRİŞ

Son yıllarda, gömülü sistemlere ve benzer birçok cihazda enerji sağlayan uzun ömürlü bataryalar sayesinde güç dönüşümü önemli ölçüde gelişmiştir. Endüstriyel uygulamalar açısından bakıldığından, uzay aracı güç sistemleri, DC motor sürücüler, telekomünikasyon ekipmanları gibi birçok alanda ve kişisel uygulamalarda bilgisayarlar, ofis ekipmanları, elektrikli ev aletleri gibi modern cihazlarda yüksek güç kaynaklarına olan ihtiyaç giderek artmaktadır. Bu artan ihtiyaçla birlikte, kontrol sistemlerinin daha güvenilir, kararlı ve ekonomik olması gerekliliği ortaya çıkmaktadır.

DC-DC yükseltici dönüştürücüler, çıkış voltajının kaynak voltajından daha yüksek olması gereken uygulamalarda kullanılan önemli güç kaynaklarıdır. Verimli, kompakt ve maliyet etkin çözümler arayışıyla, yüksek voltaj kazancı gereksinimlerini karşılamak için son yıllarda anahtarlamalı DC-DC güç dönüştürücüler hızla gelişmiştir ve hem yerel hem de endüstriyel sektörlerin standartlarını karşılayacak seviyeye ulaşmıştır. Bununla birlikte, yüksek DC voltaj kazancı sağlama zorluğuya başa çıkabilmek ve genel tasarımlı ekonomik tutabilmek amacıyla, son yıllarda çeşitli yeni topolojiler önerilmiştir.

DC-DC dönüştürücüler, üç ana tipe ayrılır: buck, boost ve buck-boost dönüştürücüler. Boost dönüştürücüsünün çıkış voltagı her zaman giriş voltagından büyüktür, bu nedenle bir yükseltici dönüştürücü olarak işlev görür. Buck dönüştürücüsü ise çıkış voltagını giriş voltagından daha düşük tutarak alçaltıcı bir dönüştürücü olarak çalışır. Buck-boost dönüştürücüsünün çıkış voltagı ise, giriş voltagından ya daha yüksek ya da daha düşük olabilir.

DC-DC dönüştürücüler endüstriyel uygulamalarda oldukça yaygın kullanılmaktadır. Bu dönüştürücüler üzerine yapılan araştırmaların çoğu, en verimli anahtarlama yöntemlerini geliştirmeye odaklanmaktadır. Ayrıca, DC dönüştürücüler elektrikli otomobillerde, tramvaylarda, deniz vinçlerinde, forkliftlerde ve maden taşıyıcılarında tahrik motoru kontrolü için kullanılır. Yüksek verimlilik, iyi ivmelenme kontrolü ve hızlı dinamik yanıt sağlamakla birlikte, DC motorlarının rejeneratif frenlemesinde enerji geri kazanımını sağlayarak enerji tasarrufu sunar. Bu özellik, sık dur-kalk yapan ulaşım sistemlerinde önemli bir avantaj sağlar.

1.1 Tezin Amacı

Bu tezin amacı, 5V-12V aralığında çalışan bir yükseltici DC-DC dönüştürücü tasarlayarak, bu dönüştürücü ile DC motor hız kontrolü gerçekleştirmektir. Bu çalışma, özellikle endüstriyel uygulamalarda, elektrikli araçlarda, ofis ekipmanlarında ve diğer taşınabilir cihazlarda gerekli olan yüksek verimli, ekonomik ve stabil güç kaynaklarını sağlamak amacıyla, anahtarlamalı DC-DC dönüştürücülerinin potansiyelini araştırmaktadır.

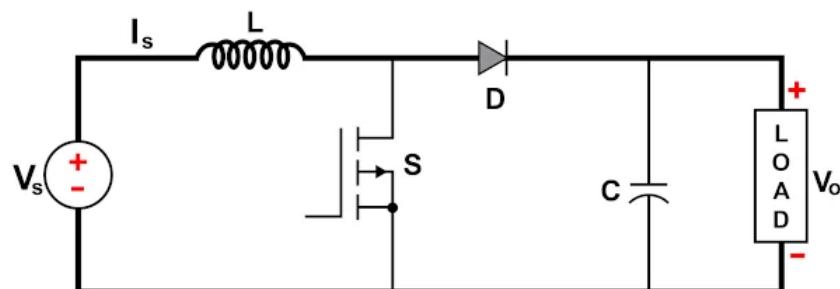
DC-DC dönüştürücüler, özellikle çıkış voltajının giriş voltajından daha yüksek olması gereken durumlarda önemli bir role sahiptir. Bu tezde, verimli bir güç dönüşümü sağlamak için kullanılan yükseltici dönüştürücülerin tasarımları ve uygulanması ele alınacak; aynı zamanda DC motor hız kontrolü için gerekli olan voltaj ve akım düzenlemeleri gerçekleştirilecektir. Tez, bu tür güç dönüştürücülerinin endüstriyel uygulamalarda ve yenilenebilir enerji sistemlerinde kullanılabilirliğini artırmaya yönelik yeni tasarım yaklaşımını ortaya koymayı hedeflemektedir.

1.2 Literatür Özeti

1.2.1 DC-DC Boost Converter Tanımı

DC-DC boost converter, girişinden aldığı bir doğru akım (DC) gerilimini çıkışında daha yüksek bir seviyeye yükselten bir güç elektroniği devresidir. Bu dönüştürücünün basitleştirilmiş devre yapısında, endüktans (L), çıkış kondansatörü (C), ideal bir anahtarlama elemanı (S) ve ideal bir diyon (D) bulunur. Giriş gerilimi genellikle bir DC kaynak olarak kabul edilir [1].

Temel olarak, voltajın yükseltilmesi amacıyla kullanılan bu dönüştürücü, enerji verimliliğini artırarak düşük volajlı güç kaynaklarını daha yüksek volaj gereksinimi duyan cihazlarla uyumlu hale getirir.



Şekil 1: Temel bir DC-DC boost converter devre şeması [1]

Bu dönüştürücüler, düşük güç tüketimli elektronik cihazlar, güneş enerjisi sistemleri, elektrikli araçlar gibi pek çok alanda, yüksek verimlilik ve kompakt tasarımlar sunarak tercih edilmektedir.

DC-DC boost converter'ın gelişimi, güç elektroniği alanındaki yeniliklerle birlikte ilerlemiştir. İlk kullanımı 1960'lara dayanan bu dönüştürücüler, başlangıçta havacılık endüstrisinde uçaklardaki elektronik sistemlere enerji sağlamak amacıyla kullanılmaktaydı [1]. O dönemde, bu cihazlardan beklenen en önemli özellikler yüksek verimlilik ve kompakt yapıydi, bu da tasarımların mümkün olduğunca basit ve verimli olmasını gerektiriyordu [2].

Zamanla, daha küçük ve daha verimli güç kaynaklarına olan ihtiyaç arttıkça, boost converter'ların popüleritesi de arttı. 1950'lerde ticari yarı iletken anahtarlarının devreye girmesiyle birlikte, bu dönüştürücüler daha etkili hale geldi. Yarı iletken anahtarlar, güç dönüştürücülerinde anahtarlama işlemlerinin daha hızlı ve verimli bir şekilde yapılmasını sağladı, bu da cihazların boyutlarını küçültüp verimliliklerini artırdı [3].

1960'ların başında, büyük DC-DC dönüştürücüler yarı iletken anahtarların yardımıyla geliştirilmeye başlandı. Havacılık endüstrisinin küçük, hafif ve verimli güç dönüştürücülerine olan ihtiyacı, bu teknolojinin hızla gelişmesini sağladı. Aynı dönemde, endüstriyel ve askeri uygulamalar için daha güçlü ve güvenilir dönüştürücüler tasarlandı [2].

1977'de, Caltech'ten RD Middlebrook'un geliştirdiği "durum-uzay ortalaması" tekniği, DC-DC dönüştürücüler için önemli bir dönüm noktası olmuştur. Bu teknik, her anahtar durumu için devre yapılandırmalarını ortalamayı içermekteydi ve devre tasarımını daha basit hale getirmiştir. Middlebrook'un yaptığı bu geliştirme, DC-DC dönüştürücülerin tasarımını daha verimli hale getirmiş ve günümüzdeki modern anahtarlamalı güç kaynaklarının temelini atmıştır [4].

Günümüzde, DC-DC boost converter'lar, enerjinin verimli bir şekilde dönüştürülmesi gereği her alanda yaygın olarak kullanılmaktadır. Elektrikli ve hibrit araçlar, enerji depolama sistemleri, telekomünikasyon altyapıları ve taşınabilir elektronik cihazlar gibi çeşitli uygulamalarda önemli bir rol oynamaktadırlar. Ayrıca, yenilenebilir enerji sistemlerinde, güneş panelleri ve rüzgâr türbinleri gibi enerji kaynaklarından elde edilen enerjiyi etkin bir şekilde yönetmek ve dönüştürmek için kullanılmaktadırlar.

1.2.2 PID Controller Tanım

PID (Proportional – Integral – Derivative) kontrolcü, kontrol mühendislerinin endüstriyel kontrol sistemlerinde sıcaklık, akış, basınç, hız ve diğer süreç değişkenlerini düzenlemek için kullandığı bir cihazdır. PID kontrolcüler, süreç değişkenlerini kontrol etmek için bir geri besleme mekanizması kullanır ve en doğru ve en kararlı kontrolcülerdir.

PID kontrolcü, en yaygın geri besleme türündür. İlk hız düzenleyicilerinde temel bir unsur olarak kullanılmış ve 1940'larda süreç kontrolünün ortaya çıkmasıyla standart bir araç haline gelmiştir. Günümüzde süreç kontrolede kullanılan kontrol döngülerinin %95'inden fazlası PID türündedir ve bu döngülerin çoğu aslında PI kontrolüdür. PID kontrolcüler, kontrolün kullanıldığı tüm alanlarda bulunmaktadır [5].

Kontrolcüler, birçok farklı formda gelir. Yıllık yüz binlerce üretilen ve bir veya birkaç döngü için kutu şeklinde bağımsız sistemler mevcuttur. PID kontrolü, dağıtılmış kontrol sistemlerinin önemli bir bileşenidir. Ayrıca, birçok özel amaçlı kontrol sisteme gömülüdür. PID kontrolü, genellikle enerji üretimi, ulaşım ve imalat gibi karmaşık otomasyon sistemlerini oluşturmak için mantık, sıralı işlevler, seçiciler ve basit işlev bloklarıyla birleştirilir [5].

Model öngörülü kontrol gibi birçok gelişmiş kontrol stratejisi de hiyerarşik olarak organize edilir. PID kontrolü en alt seviyede kullanılır ve çok değişkenli kontrolcü, alt seviyedeki kontrolcülere referans değerlerini sağlar. Bu nedenle PID kontrolcü, kontrol mühendisliğinin "ekmek ve tereyağı" olarak adlandırılabilir. Her kontrol mühendisinin araç kutusunda önemli bir bileşendir.

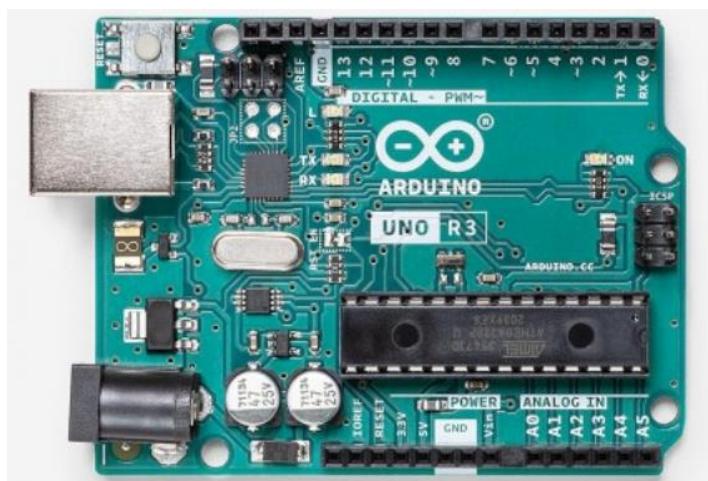
PID kontrolcüler, mekanikten, elektronik tüpler, transistörler, bütünlüksüz devreler üzerinden mikroişemicilere kadar teknolojideki birçok değişime rağmen varlığını sürdürmüştür. Mikroişlemci, PID kontrolcü üzerinde dramatik bir etki yaratmıştır. Günümüzde üretilen neredeyse tüm PID kontrolcüler mikroişemicilere dayanmaktadır. Bu durum, otomatik ayar, kazanç planlaması ve sürekli adaptasyon gibi ek özelliklerin sağlanması olanak tanımıştir.

1.2.3 Arduino UNO Tanım

Arduino.cc tarafından geliştirilen Arduino, düşük maliyetli, açık kaynaklı ve basit bir donanım-yazılım yapısına sahip bir elektronik geliştirme platformudur. Atmega328P mikrodenetleyici çipini temel alan bu geliştirme kartıdır [6, 7].

Dijital ve analog giriş/çıkış pinlerine sahip olan Arduino Uno, diğer elektronik devrelerle kolayca entegrasyon sağlayabilir. Kart üzerinde 14 dijital giriş/çıkış pini bulunur ve bunlardan 6 tanesi PWM çıkışını verebilir. Ayrıca, 40 mA DC akım sağlayabilen ve 3,3V'ta 50 mA yük sürebilen 6 analog giriş pini vardır [8]. Bu pinlerle bir sensörden alınan ışık verisi veya bir düğmeye yapılan basınç gibi girdileri algılayabilir ve bu girdilere karşılık olarak bir LED'in yanması ya da bir motorun çalışması gibi çıktılar üretебilir. Ayrıca kart üzerinde seri haberleşme pinlerinde bulunmaktadır [7].

Programlama işlemleri için Arduino IDE kullanılmakta ve kart, USB kablosu ya da harici bir güç kaynağı ile çalıştırılabilir. Çalışma voltajı 7 ila 20 volt arasında değişmekle birlikte, genellikle 9 voltluk bir pil veya USB bağlantısı ile güç sağlanır. Kartın öne çıkan diğer özellikleri arasında 32 kB Flash bellek, 2 kB SRAM ve 16 MHz saat frekansı bulunmaktadır. Bu özellikler, Arduino UNO'nun çeşitli elektronik projelerde etkili bir şekilde kullanılmasını sağlar [7, 8].



Şekil 2: Arduino UNO mikrodenetleyici kartı [2]

Arduino, 2005 yılında İtalya'da, Etkileşim Tasarımı Enstitüsü İvrea'da geliştirilmeye başlanmıştır. Massimo Banzi tarafından başlatılan bu proje, mühendislik bilgisi olmayan kişilerin de elektronik projeler tasarlayabilmesi için düşük maliyetli ve kolay kullanılabilir bir platform sunmayı

amaçlamıştır [9, 10]. Platformun kurucuları arasında Massimo Banzi'nin yanı sıra David Cuartielles, Tom Igoe ve David Mellis de yer almaktadır [10]. Arduino'nun açık kaynaklı yapısı, Creative Commons lisansı altında paylaşılmasını sağlamış ve bu sayede bireylerin kendi Arduino kartlarını üretmelerine olanak tanınmıştır [9]. Arduino, günümüzde robotik, drone tasarımları ve çeşitli otomasyon projelerinde sıkça kullanılan bir platform haline gelmiştir. İşminin anlamı İtalyanca'da "Güvenilir Dost" veya "Arkadaş" olarak çevrilmekte olup, bu güvenilirlik, platformun uygulamaları eksiksiz yerine getirme konusundaki başarısını yansıtmaktadır. Kodlama ve programlama bilgisiyle desteklenen Arduino, günlük hayatı kolaylaştıran teknolojik çözümler sunmakta ve yaşamı kolaylaştıran birçok projeye olanak tanımaktadır [10].

1.2.4 DC Motor Tanım

DC (Doğru Akım) motoru, elektriksel enerjiyi mekanik enerjiye dönüştüren ilk motor türlerinden biridir ve elektrik mühendisliğinin en temel buluşlarından biridir. DC motorlarının geliştirilmesi, elektrikle ilgili pek çok buluşun önünü açarak modern sanayi devriminde önemli bir rol oynamıştır. DC motorunun tarihçesi, elektrik mühendisliğinin gelişimiyle paralel olarak ilerlemiştir [11].



Şekil 3: Enkoderli bir DC motor [3]

DC motorunun temelleri, 19. yüzyılın başlarına kadar gitmektedir. 1821 yılında Michael Faraday, manyetik alanların elektrik akımını etkileyebileceğini keşfederken elektromanyetik induksiyon prensibini bulmuştur. Faraday'ın bu buluşu, elektromıknatıslar ve elektrik motorlarının ilk prototiplerinin gelişimine olanak sağlamıştır. Bu dönemde, doğru akım ile çalışan ilk elektrik motoru, bilim insanları ve mühendisler tarafından deneysel olarak yapılmış, fakat pratikte kullanımı yaygınlaşmamıştır [12].

Günümüzde, DC motorları özellikle düşük güçlü, hassas hız kontrolü gerektiren alanlarda yaygın olarak kullanılmaktadır. Elektrikli arabalar, robotik sistemler, bilgisayar donanımları ve HVAC sistemlerinde (Isıtma, Havalandırma ve Klima) DC motorları önemli bir yer tutmaktadır. Ayrıca,

günümüz teknolojileriyle entegre edilen firçasız DC motorlar (DC motorlar), daha verimli ve dayanıklı motorlar olarak elektrikli araçlar ve diğer endüstriyel uygulamalarda kullanılmaktadır [13].

DC motorlarının tarihi, elektrik mühendisliğinin ilerlemesiyle derinden bağlantılıdır. Faraday’ın keşiflerinden Edison’un ticari uygulamalarına, Tesla’nın AC sistemlerinden günümüzdeki modern uygulamalara kadar DC motorları, teknolojik ilerlemenin öncüsü olmuştur. Yüksek verimlilik, hassas hız kontrolü ve düşük maliyetli üretim avantajları ile DC motorları, endüstriyel sistemlerden kişisel cihazlara kadar geniş bir kullanım alanına sahiptir. Bu nedenle, DC motorları mühendislik ve sanayi tarihinin önemli bir parçası olmaya devam etmektedir [14].

1.2.5 Encoder Tanımı

DC motor enkoderleri, DC motorlarda hız kontrol geri bildirimi için kullanılır. Bu motorlarda, sarılmış tellerden oluşan bir armatür veya rotor, bir stator tarafından oluşturulan manyetik alanın içinde döner. DC motor enkoderi, rotorun hızını ölçmek ve sürücüye hassas hız kontrolü için kapalı döngü geri bildirimi sağlamak amacıyla bir mekanizma sunmaktadır.

Endüstriyel kontrol sistemleri için bir enkoder, bir motor milinin veya diğer mekanik hareketlerin konumunu, hızını ve yönünü algılayan ve bu veriyi diğer cihazlara iletten özel bir sensördür. Konum bilgisi, optik, manyetik veya kapasitif gibi üç farklı teknoloji kullanılarak belirlenebilir. Bu cihazlar, döner masaların konumlandırılması, taşıma ve yerleştirme işlemleri, makine montajı, ambalajlama, robotik uygulamalar gibi birçok alanda hassas kontrol sağlamak için gerekli verileri temin eder. Enkoderler, konum kontrolü için bir referans noktası olarak kullanılan yön tespiti sağlamak amacıyla benzer bir yöntem sunar [15].



Şekil 4: DC bir motor üzerindeki enkoder [3]

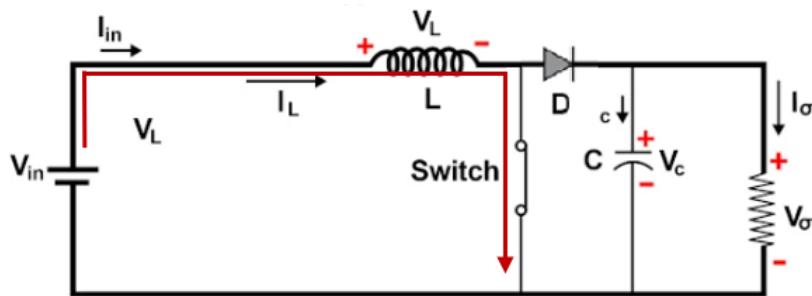
2. TEORİK ALTYAPI VE ANALİZLER

2.1 DC-DC Boost Converter

2.1.1 Boost Converter Çalışma Prensibi

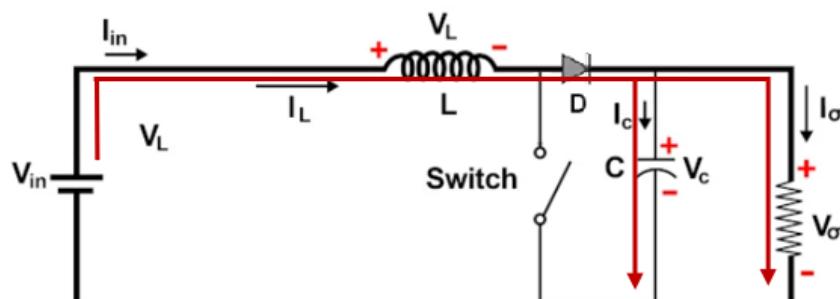
Yükseltici dönüştürücüler, tamamlayıcı anahtarlama prensibiyle çalışmaktadır, açık ve kapalı olmak üzere iki temel çalışma modu bulunmaktadır [16]. Giriş voltajı, anahtar elemanı (MOSFET veya benzeri bir transistör) tarafından belirli aralıklarla kesilip açılarak, endüktansta manyetik alan oluşturulur. Transistörün kesilip açılması modları oluşturmaktadır [16, 17].

Kapalı mod sırasında transistör iletimdedir, Giriş kaynağından endüktans beslenir. Endüktanstan geçen akım lineer olarak artar bu endüktansın bir manyetik alan oluşturmasına, dolayısıyla da endüktansta enerji depolamasına yol açar, bu enerji zamanla yükselir ve bu aralıkta kondansatör, depoladığı enerji yükün üzerine aktararak yükü besler [16, 17].



Şekil 5: Anahtarın iletim durumu (kapalı mod) [1]

Açık modda ise, transistörün kesime girmesiyle endüktansta biriken enerji serbest bırakılır ve bu enerji diyon ile iletme girer. Açık modda diyon iletimdedir. Güç kaynağı ve endüktansta biriken enerji kondansatör tarafından depolanır ve yük beslenir. Aynı anda endüktanstan geçen lineer akımda azalma meydana gelir ve endüktansın enerji seviyesi düşüş gösterir. Ayrıca, açık modda güç elemanları çıkış gerilimine maruz kalmaktadır [16, 17].



Şekil 6: Anahtarın kesim durumu (açık mod) [1]

Kapalı modda giriş gerilimi ile beslenen endüktans, açık modda akım sürdürübilmek üzere çıkış gerilimini eksi giriş gerilimi şeklinde bir emk (elektromotor kuvveti) üretir [17].

2.1.2 Boost Converter Matematiksel Modeli

Analiz için, devre elemanlarının ideal olduğu kabul edilir, hesaplamalar bu kabule göre gerçekleştirilir. Anahtarın yani transistörün iletim olduğu durumda üzerindeki gerilim düşümü sıfırdır, iletimde veya kesimde olması durumlarında gerilimlerde çakışmalar meydana gelmemektedir. Aynı şekilde diyonetin iletimde olması durumunda diyonet üzerindeki gerilim düşümü sıfırdır ve endüktans ile kapasite kayıpsızdır [18].

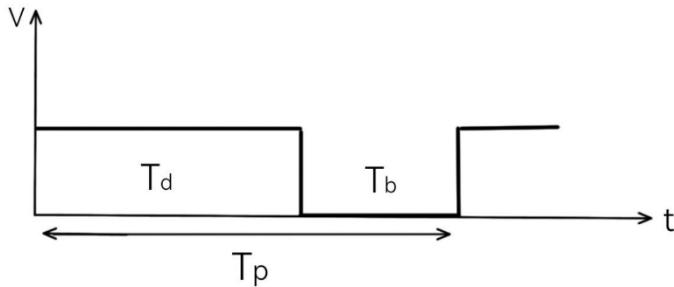
Ayrıca, sistemin kararlı olarak çalıştığı kabul edilir. Kararlı çalışma rejiminde, devre çıkışında yeterince büyülüklükte bir kondansatör olduğu ve bir periyot boyunca kondansatör geriliminin (V_c) sabit kaldığı kabul edilmektedir. Aynı zamanda çıkış gerilimi (V_o) kondansatör üzerindeki gerilime (V_c) eşit olduğu için çıkış akımı (I_o) sabit kabul edilir [17].

Karalı rejimde; endüktansın ve kondansatörün anahtarlama esnasında aldığı ve verdiği enerji miktarları, endüktans akımındaki artma ve azalma miktarları, kondansatörün gerilimindeki artma ve azalma miktarları eşittir [17].

Devrenin kararlı rejimde çalışması durumunda, endüktansın akım değerine bağlı olacak şekilde iki çalışma modu tanımlanabilir. Endüktansın akımı hiçbir zaman sıfıra düşmediği bu moda CCM (Sürekli İletim Modu) denir. Endüktansın akımının aralıklarla sıfıra düştüğü moda ise DCM (Kesintili İletim Modu) denir [2]. Bu çalışma modu, frekans ve endüktans değerine bağlı olacak şekilde oluşur veya oluşturulur. Genellikle CCM modunda çalışmakta olan bu dönüştürücü devre, bazen de DCM modıyla da çalıştırılmaktadır [17].

Devrenin çalışması periyodiktir. Periyot ($T_{periyot}$), transistörün iletimde olduğu aralık ($T_{doluluk}$), transistörün kesimde diyonetin ise iletimde olduğu aralık ($T_{boşluk}$) şeklinde belirtilmektedir. Periyot, transistörün iletimde ve kesimde olduğu aralıkların toplamına eşittir [17].

$$T_p = T_d + T_b \quad (1)$$



Şekil 7: $D = 2/3$ için (V) gerilimin ve (t) zaman aralığında çalışma periyodu grafiği [4]

DC dönüştürücülerdeki çalışma frekansı şebeke frekansına göre çok yüksektir. Çalışma frekansı (f) aşağıdaki bağıntıyla elde edilir [17].

$$f = 1/T_p \quad (2)$$

Duty cycle yani doluluk oranı (λ veya D) aşağıdaki bağıntılı elde edilir.

$$\lambda = D = \frac{T_d}{T_p}, \quad 0 < D < 1 \quad (3)$$

Ayrıca, (λ) ve (T_p) değerleri kullanılarak (T_d) ve (T_b) elde edilebilir.

$$T_d = D T_p \quad (4)$$

$$T_b = (1 - D)T_p \quad (5)$$

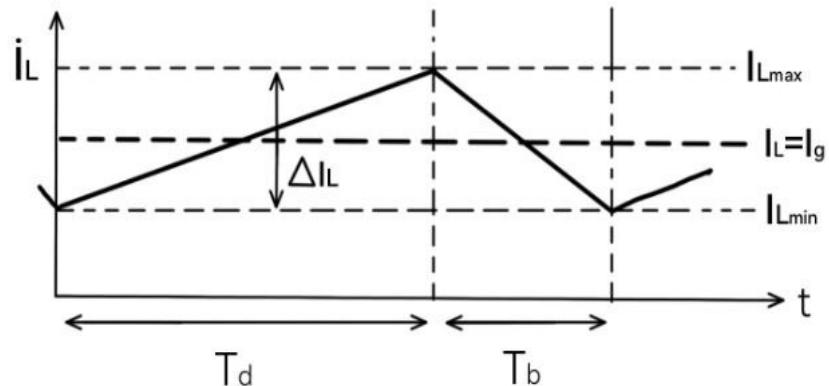
Olarak tanımlanır. Doluluk oranındaki değişiklik ile güç kontrolü sağlanır. Genellikle doluluk oranının kontrolünü, frekansın sabit tutulması ve PWM darbe genişliğinin değiştirilmesiyle sağlanmaktadır bu PWM (Darbe Genişliği Modülasyonu) olarak adlandırılır. Çok yaygın olmaya da bazı durumlarda darbe genişliği sabit tutularak frekans değiştirilmektedir. Bu kontrol türü ise FM (Frekans Modülasyonu) olarak adlandırılır. FM'e göre daha kolay bir kontrol sağlayan PWM kontrolü daha yaygın olarak tercih edilmektedir. Endüktansta geriliminin dalga şeklinde göre aşağıdaki bağıntılar yazılabilir [17].

Endüktans akımdaki artma oranı:

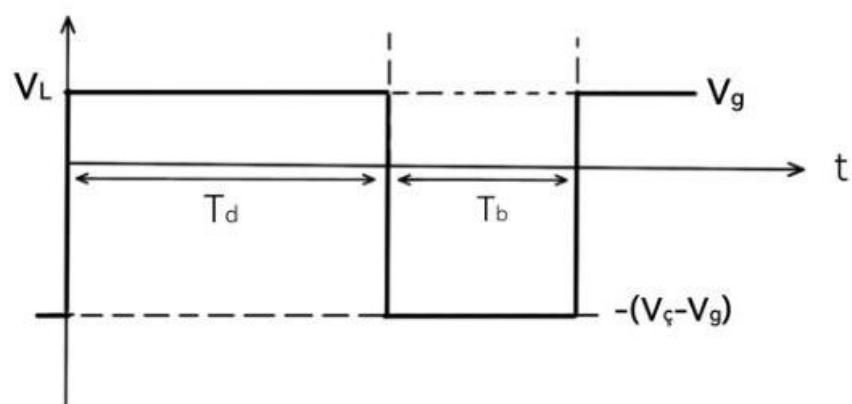
$$\Delta I_{Lp} = \frac{V_g}{L} T_d \quad | \quad 0 < t < T_d \quad (6)$$

Endüktans akımdaki azalma oranı:

$$\Delta I_{Ln} = \frac{V_c - V_g}{L} T_b \quad | \quad T_d < t < T_p \quad (7)$$



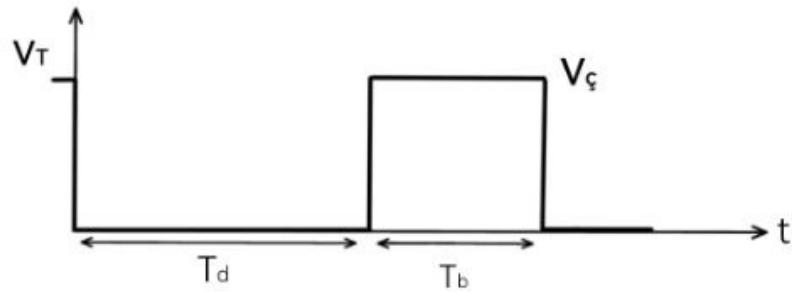
Şekil 8: Endüktans akımının artma ve azalma grafiği [4]



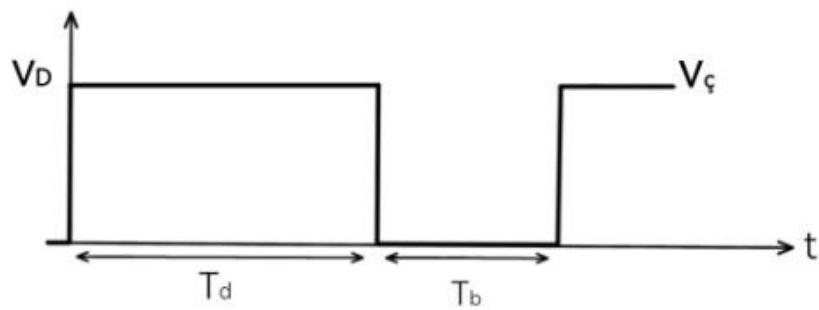
Şekil 9: Endüktans gerilim grafiği [4]

Kararlı rejimde endüktans geriliminin pozitif ve negatif alanları birbirine eşittir. Dolayısıyla, ortalama DC çıkış gerilimi (V_c) aşağıdaki şekilde bulunabilir ve bu devrede transistör ile diyor çıkış gerilimine maruz kalmaktadır [17].

$$V_c = \frac{1}{1-D} V_g \quad (8)$$



Şekil 10: Anahtarlama esnasında, transistörün üzerindeki (V_T) geriliminin grafiği [4]



Şekil 11: Anahtarlama esnasında, diyon üzerindeki (V_D) geriliminin grafiği [4]

Girişteki ve çıkıştaki güçlerin eşitliğinden, giriş akımı (I_g) ve çıkış akımının (I_c) bağıntısı aşağıdaki şekilde elde edilebilir. Ayrıca, endüktans akımı (I_L) ve giriş akımı (I_g) birbirine eşittir [17].

$$I_g = \frac{1}{1-D} I_c \quad (9)$$

Kararlı rejimde, $\Delta I_{Lp} = \Delta I_{Ln} = \Delta I_L$ eşitliği kabul edilerek (8) numaralı bağıntılarının ortak çözümünden, endüktans akımındaki dalgalanma miktarı hesaplanabilir.

$$\Delta I_L = D(1 - D) \frac{V_c}{f_p L} \quad (10)$$

Yukarıdaki (10) numaralı bağıntıda D 'ye göre türev alınıp sonuç sıfır'a eşitlenirse $D = \frac{1}{2}$ için olan endüktans akımının maksimum dalgalanma miktarı aşağıdaki gibi hesaplanır.

$$\Delta I_{Lmax} = \frac{V_c}{4f_p L} \quad (11)$$

Bu dönüştürücü devresindeki belli bir akımın dalgalanma miktarı için gerekli olan endüktans değeri, çıkış gerilime bağlıdır ve bu endüktans değeri frekansla ters orantılıdır [17].

Dönüştürücünü sürekli, sınırda ve kesikli çalışma modları giriş akımı ile endüktansın dalgalanma akımı arasındaki ilişkiyle açıklanır. Sırasıyla sürekli, sınırda ve kesikli modda çalışma:

$$I_g > \Delta I_L / 2$$

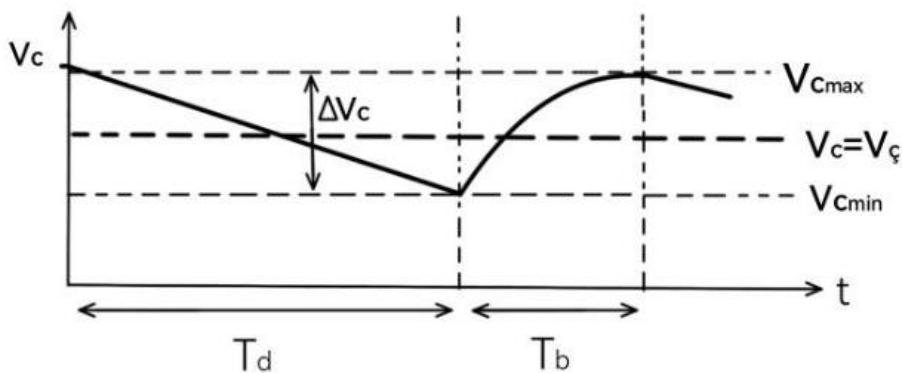
$$I_g = \Delta I_L / 2 \quad (12)$$

$$I_g < \Delta I_L / 2$$

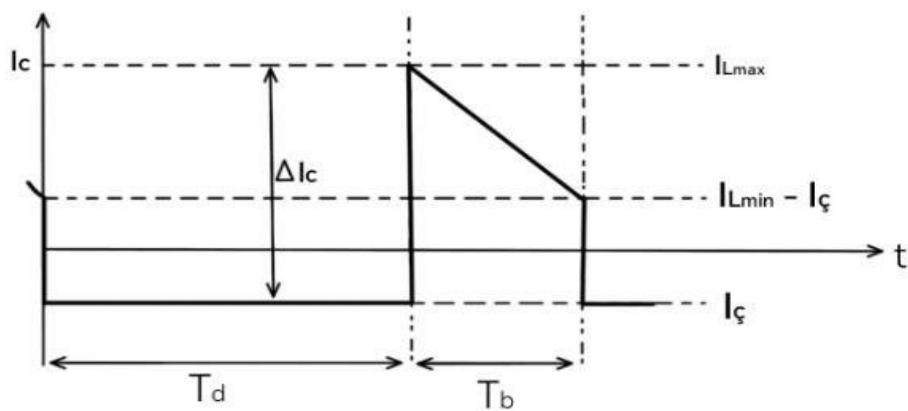
Kararlı rejimde, transistörün iletimde olduğu aralıkta çıkış akımının tamamını kondansatör üstlenmektedir. Transistörün kesimde olduğu aralıkta boşalan kondansatör transistörün iletme geçmesiyle tekrar dolar. Ayrıca, kondansatör üzerindeki gerilim dalgalanması $\Delta V_{cp} = \Delta V_{cn} = \Delta V_c$ kabul edilir ve (13) numaralı denklemle hesaplanmaktadır.

$$V_c = \frac{1}{C} \int i_c dt \rightarrow \Delta V_{cp} = \Delta V_{cn} = \Delta V_c = \frac{I_c}{C} T_d = \frac{I_c}{C} D T_p$$

$$\Delta V_c = \frac{D I_c}{f_p C} \quad (13)$$



Şekil 12: Kondansatördeki gerilimin dolma-boşalma grafiği [4]



Şekil 13: Kondansatör üzerindeki (I_c) akım grafiği [4]

2.1.3 Boost Converter İçin Komponent Seçimi

2.1.3.1 Endüktans (Bobin) Seçiminde Dikkat Edilmesi Gerekilenler

DC-DC boost converter'ların çalışmasında endüktans, enerjinin depolanması ve aktarılması için kritik bir role sahiptir. Endüktans seçiminde dikkat edilmesi gereken en önemli parametre, Şekil 8'deki grafikte yer alan akım dalgalanması (ΔI_L) miktarıdır. Bu dalgalanma, dönüştürücünün verimliliğini doğrudan etkiler. Bu sebeple düşük akım dalgalanması için daha büyük değerli bir endüktans seçilmelidir.

Ayrıca, endüktans değerini seçerken anahtarlama frekansı (f_p), giriş gerilimi (V_g), çıkış gerilimi (V_c), ve yük akımı (I_c) dikkate alınmalıdır. Endüktans seçimi sırasında, seçilen bileşenin akım taşıma kapasitesi değerine dikkat edilmelidir. Endüktansın, devrede oluşabilecek maksimum akımı güvenli bir şekilde taşıyabilmesi gereklidir.

2.1.3.2 Anahtar Seçiminde Dikkat Edilmesi Gerekilenler

DC-DC boost converter'lar, kHz ve MHz gibi yüksek frekanslarda çalışıkları için hızlı bir anahtarlama elamanını ihtiyaç duymaktadır bu sebeple anahtarlama elamanı olarak bir MOSFET seçilmesi devrenin verimliliği açısından daha faydalı olacaktır.

Yükseltici dönüştürücüler, diğer dönüştürücülerde olduğu gibi anahtarlama sinyalini bir kontrol devresinden alır. Günümüzde bu kontrol devrelerinde, mikrodenetleyiciler yaygın olarak kullanılmaktadır. Bunun sebebi, mikrodenetleyicilerin sayısal ve analog yapılarının birlikte gerçekleştirerek denetimi kolaylaştırmalarıdır [18]. Bu projede, kontrol kolaylığı açısından Arduino UNO mikrodenetleyicisini kullanmayı uygun gördük.

MOSFET seçiminde dikkat edilmesi gereken diğer husus ise MOSFET'in Gate (G) ucuna verilecek olan PWM sinyalinin voltaj değeridir. PWM sinyalini, Arduino UNO ile vereceğimiz ve Arduino UNO mikrodenetleyicisinin maksimum 5V çıkış verdieneni için MOSFET seçimi 5V ile tamamen iletme girecek şekilde seçilmelidir. Ayrıca, bu devredeki anahtarlama elamanı, **Şekil 10**'deki grafikte görüldüğü üzere (V_c) çıkış gerilimine maruz kalacaktır, bu sebeple MOSFET seçilirken maksimum dayanım gerilimi, dönüştürücünün çıkış geriliminin altında kalmayacak şekilde seçilmelidir.

2.1.3.3 Diyot Seçiminde Dikkat Edilmesi Gerekilenler

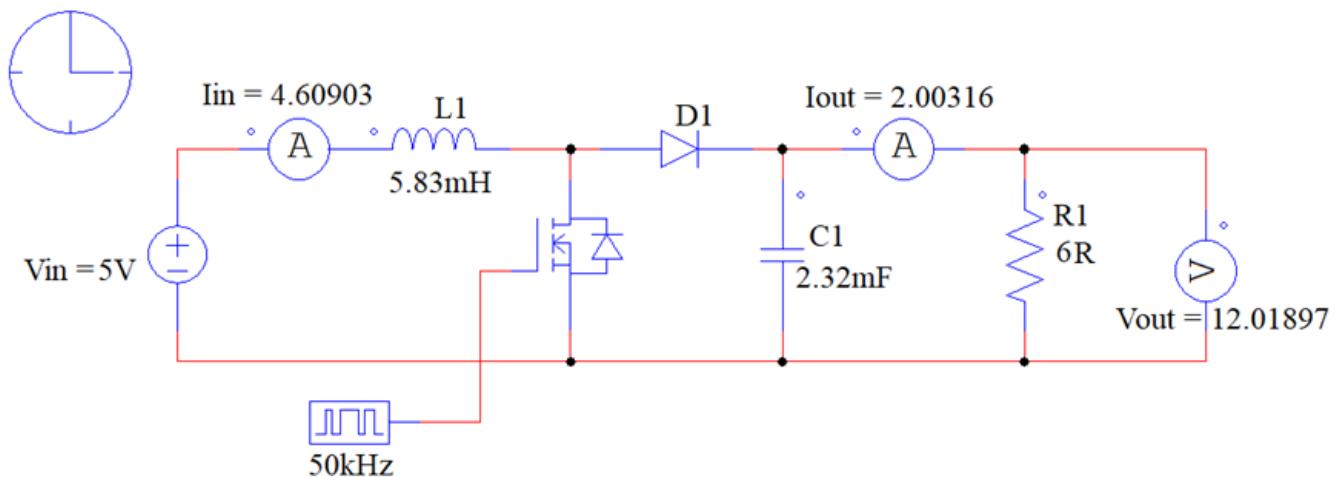
Boost converter'lar, yüksek frekansta çalışıkları için hızlı anahtarlama yapabilen bir diyot kullanımı gereklidir. Sık kullanılan silikon diyotlar bu tür devrelerde verimsiz olabilir. Bu nedenle, Schottky diyotlar tercih edilir. Schottky diyotların düşük ileri gerilim düşümü, daha az güç kaybına yol açarak devrenin verimliliğini artıracaktır. **Şekil 11**'deki grafikte görüldüğü üzere de diyot, (V_c) çıkış gerilimine maruz kalmaktadır. Bu sebeple, diyotun geri dönüş gerilimi, dönüştürücünün çıkış geriliminden daha yüksek olmasına dikkat edilmelidir. Ayrıca, ileri akım kapasitesi de maksimum yük akımını güvenli bir şekilde taşıyabilecek bir değer olmalıdır.

2.1.3.4 Kondansatör Seçiminde Dikkat Edilmesi Gerekilenler

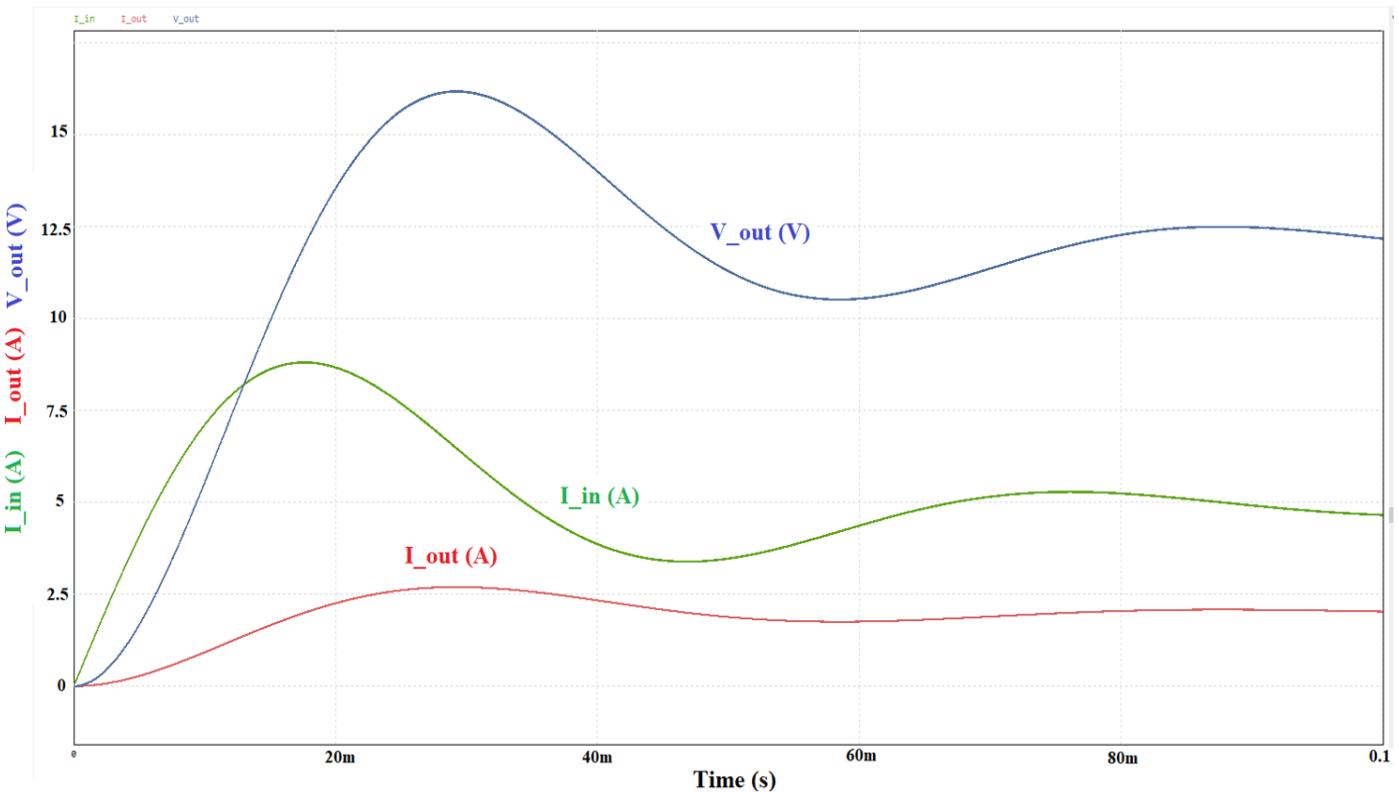
Kondansatör seçiminde, dikkat edilmesi kapasite değer (C), çalışma gerilimi (V_c), kondansatörün iç direncine dikkat edilmelidir. Dönüştürücü çıkıştaki geriliminin dalgalanmasını azaltmak ve stabil bir çıkış sağlamak için yüksek kapasite değerine sahip bir kondansatör seçimine dikkat edilmelidir. Çıkış kondansatörünün kapasitesi, devredeki yük akımı ve gerilim dalgalanması hedeflerine göre seçilir. Burada yüksek kapasiteli bir kondansatör seçmek gerilim dalgalanmalarını azaltacaktır. Devredeki verimliliğe etkisi olan kondansatörün iç direnç değeri, güç kaybı yaşamamak için düşük seçilmelidir.

Ayrıca, **Şekil 12**'daki grafikte görüldüğü üzere kondansatörün en yüksek gerilimi ($V_{c_{max}}$), devrenin çıkış geriliminden yüksektir. Bu sebeple, kondansatörün çalışma gerilimi (V_c), çıkış gerilimi (V_c) değerinden daha yüksek seçilmelidir. Bu, kondansatörün aşırı gerilim nedeniyle zarar görmesini önlemek için önemlidir.

2.1.4 Boost Converter Simülasyonu



Şekil 14: 5V-12V DC-DC boost converter grafiği (PSIM)



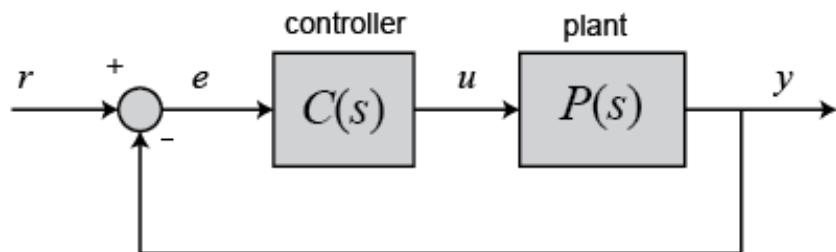
Şekil 15: Giriş gerilimi 5V DC olan boost converter simülasyonunun (Ig), (İç) ve (Vç) değerlerinin grafiği

2.2 PID Kontrol Sistemi

2.2.1 PID Kontrol Sisteminin Çalışma Prensibi

PID kontrol, sıcaklık da dahil olmak üzere çeşitli süreçleri kontrol etmek için çok basit ve güçlü bir yöntemdir [19].

Diyelim ki bir süreciniz var (örneğin ısıtıcı ve kompresörlü bir sıcaklık odası) ve bu süreç Değişkeni y (örneğin odadaki sıcaklık ölçümü) üretiyor. Süreç, kontrol cihazından gelen bir sürücü sinyali (u) ile kontrol ediliyor ve amacınız PV 'yi hedef değere, yani ayar noktası olarak da bilinen değere eşleştirmek [2].



Şekil 16: PID kontrol Sistem modeli [5]

PID kısaltması, Proportional (Oransal), Integral (İntegral) ve Derivative (Türev) terimlerinin baş harflerinden oluşur. Her döngüde PID kontrolcü, yukarıdaki diyagramda gösterildiği gibi, Ayar Noktası (*Setpoint*) ile ölçülen Süreç Değişkeni (*Process Variable*, PV) arasındaki hatayı kullanarak bir sonraki çıkış değerini hesaplar. Çıkış değerini aşağıdaki üç değerin toplamı olarak hesaplar:

1. **Oransal terim:** Hatayı alır ve bir sabit K_P ile çarpar.
2. **Integral terim:** Birikmiş toplam hatayı alır ve bir sabit K_I ile çarpar.
3. **Türev terim:** Hatanın değişim hızını alır ve bir sabit K_D ile çarpar.

Son olarak, bu üç değeri toplar ve o döngü için nihai çıkış değeri U'yu üretir.

2.2.2 PID Kontrol Sisteminin Matematiksel Modeli

Bir PID kontrolcüsünün çıkışı $u(t)$, Oransal, İntegral ve Türev terimlerinin toplamı kullanılarak hesaplanır. Burada K_P , K_I ve K_D sabitleri, kontrolcünün performansını ince ayar yapabilmek için ayarlanabilir.

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t te(t)dt + K_D \cdot \frac{de(t)}{dt} \quad (14)$$

Output = Proportional + Integral + Derivative

Proportional (P) kontrol: Bu bileşen, setpoint ile süreç değişkeni (PV) arasındaki hata temelinde sürecin çıktısını ayarlar. Hata ne kadar büyükse, uygulanan düzeltme de o kadar büyük olur.

$$\text{Proportional} = K_P e(t) \quad (15)$$

Integral (I) kontrol: Bu bileşen, zamanla biriken hata temelinde çıktıyı ayarlar. Kararlı durum hmasını ortadan kaldırırmaya yardımcı olur ve kontrol sisteminin stabilitesini iyileştirebilir.

$$\text{Integral} = K_I \int_0^t te(t)dt \quad (16)$$

Derivative (D) kontrol: Bu bileşen, hatanın değişim hızına dayalı olarak çıktıyı ayarlar. Salınımları azaltmaya ve kontrol sisteminin stabilitesini iyileştirmeye yardımcı olur. Türev terimi, ölçüm gürültüsünü (rastgele dalgalanmalar) artırabilir ve aşırı çıkış değişikliklerine neden olabilir. Daha iyi bir süreç değişkeni değişim hızı tahmini elde etmek için filtreler önemlidir.

$$Derivative = K_D \cdot \frac{de(t)}{dt} = -K_D \cdot \frac{d(PV)}{dt} \quad (17)$$

Bir PID kontrolcüsünün çıkışı, sürece uygulanan kontrol girişine eşit olan, geribildirim hatasından zaman domaininde şu şekilde hesaplanır [3].

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t te(t)dt + K_D \cdot \frac{de(t)}{dt} \quad (18)$$

Öncelikle, yukarıda gösterilen şemayı kullanarak PID kontrolcüsünün kapalı döngü sisteminde nasıl çalıştığını inceleyelim. Değişken (e), izleme hatasını, istenen çıkış (r) ile gerçek çıkış (y) arasındaki farkı temsil eder. Bu hata sinyali (e), PID kontrolcüsüne iletılır ve kontrolcü, bu hata sinyalinin türevini ve integralini zamanla hesaplar.

Sürece uygulanan kontrol sinyali (U), orantısal kazanç (K_P) ile hatanın büyüklüğünün çarpımı, artı integral kazanç (K_I) ile hatanın integralinin çarpımı, artı türevsel kazanç (K_D) ile hatanın türevine çarpımıdır [3].

Bu kontrol sinyali (U), bitkiye iletılır ve yeni bir çıkış (y) elde edilir. Yeni çıkış (y) daha sonra geri beslenir ve yeni hata sinyalini (e) bulmak için referansla karşılaştırılır. Kontrolcü, bu yeni hata sinyalini alır ve kontrol girişini günceller. Bu süreç, kontrolcü aktif olduğu sürece devam eder [3].

PID kontrolcüsünün transfer fonksiyonu, yukarıdaki Denklemin Laplace dönüşümü alınarak bulunur [20].

$$K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \quad (19)$$

Burada K_P = orantısal kazanç, K_I = integral kazanç ve K_D = türevsel kazançtır.

Orantısal kazancın (K_P) artırılması, aynı hata seviyesi için kontrol sinyalinin orantılı olarak artırılması etkisini gösterir. Kontrolcünün belirli bir hata seviyesinde daha "sert" bir şekilde tepki vermesi, kapalı döngü sisteminin daha hızlı tepki vermesine neden olur, ancak aynı zamanda daha fazla aşım yapmasına yol açar. K_P 'nin artırılmasının bir diğer etkisi, durağan hata miktarını azaltma eğiliminde olmasıdır, ancak bu hatayı tamamen ortadan kaldırılmaz [4].

Kontrolcuya türev teriminin (K_D) eklenmesi, kontrolcünün hatayı "öngörme" yeteneğini artırır. Basit orantısal kontrol ile K_P sabitse, kontrolün artmasının tek yolu hatanın artmasıdır. Ancak türevsel kontrol ile hata yukarı doğru eğilmeye başladığında, hata büyülüüğü nispeten küçük olsa bile kontrol sinyali büyük hale gelebilir. Bu öngörüleme, sisteme sönüm ekleyerek aşımı azaltma eğilimindedir. Ancak, türev teriminin eklenmesi durağan hataya herhangi bir etkide bulunmaz.

Kontrolcuya integral teriminin (K_I) eklenmesi, durağan hatayı azaltmaya yardımcı olur. Eğer kalıcı ve sabit bir hata varsa, integratör bu hatayı biriktirerek kontrol sinyalini artırır ve hatayı azaltır. Ancak integral teriminin bir dezavantajı, sistemi daha yavaş (ve dalgalı) hale getirebilmesidir, çünkü hata sinyali işaret değiştirdiğinde, integratörün birikimi "geri alması" zaman alabilir [4].

Kapalı döngü bir sistemde her bir kontrol parametresinin (K_P, K_D, K_I) genel etkileri aşağıdaki tabloda özetlenmiştir. Bu yönergeler birçok durumda geçerli olsa da her zaman doğru olmayabilir. Bireysel kazançların ayarının etkisini gerçekten anlamak istiyorsanız, daha fazla analiz yapmamız veya gerçek sistem üzerinde testler gerçekleştirmemiz gereklidir [21].

Kapalı Döngü Yanıtı	Yükselme zamanı	En büyük aşım	Yerleşme süresi	Hata
K_P	Düşüş	Artış	Ufak değişiklik	Düşüş
K_I	Düşüş	Artış	Artış	Düşüş
K_D	Ufak değişiklik	Düşüş	Düşüş	Değişim yok

Tablo 1: K_P, K_I ve K_D Değerlerinin Grafik Üstündeki Yaklaşık Değişimleri

2.3 Arduino UNO

2.3.1 Arduino ile PWM Sürmek

Arduino, 2005 yılında İtalya'da, Etkileşim Tasarımı Enstitüsü Ivrea'da geliştirilmeye başlanmıştır. Massimo Banzi tarafından başlatılan bu proje, mühendislik bilgisi olmayan kişilerin de elektronik projeler tasarlayabilmesi için düşük maliyetli ve kolay kullanılabilir bir platform sunmayı amaçlamıştır [9, 10].

Platformun kurucuları arasında Massimo Banzi'nin yanı sıra David Cuartielles, Tom Igoe ve David Mellis de yer almaktadır [10]. Arduino'nun açık kaynaklı yapısı, Creative Commons lisansı altında paylaşılmasını sağlamış ve bu sayede bireylerin kendi Arduino kartlarını üremelerine olanak tanınmıştır [9].

Arduino, günümüzde robotik, drone tasarımları ve çeşitli otomasyon projelerinde sıkça kullanılan bir platform haline gelmiştir. İsminin anlamı İtalyanca'da "Güvenilir Dost" veya "Arkadaş" olarak çevrilmekte olup, bu güvenilirlik, platformun uygulamaları eksiksiz yerine getirme konusundaki başarısını yansıtmaktadır. Kodlama ve programlama bilgisiyle desteklenen Arduino, günlük hayatı kolaylaştıran birçok projeye olanak tanımaktadır [10].

2.3.2 Arduino UNO'nun Projedeki Kullanımı

Arduino UNO, projenin hem birinci hem de ikinci kısmında temel bir kontrol birimi olarak görev yapmaktadır. Projenin birinci kısmında, Arduino UNO, DC-DC boost converter'in MOSFET sürücüsünü kontrol etmek için PWM sinyali üretmek amacıyla kullanılacaktır. Bu işlem için Arduino UNO'nun 9 numaralı dijital pini tercih edilmiştir. Bu pinin seçilme sebebi, Arduino UNO'daki Timer 1 ile doğrudan ilişkili olmasıdır. Timer 1, 16-bit çözünürlüğe sahip olduğundan, daha yüksek doğruluk ve hassasiyet gerektiren PWM sinyallerini üretmek için idealdir. Bu özellik, özellikle kHz düzeyinde hassas frekans kontrolü gerektiren uygulamalarda önemli bir avantaj sağlamaktadır. Ayrıca Timer 1'in diğer dahili zamanlayıcılarla kıyasla daha fazla yapılandırma seçeneği sunması, PWM sinyallerinin frekans ve görev döngüsünü daha esnek bir şekilde ayarlamaya olanak tanır [22, 23].

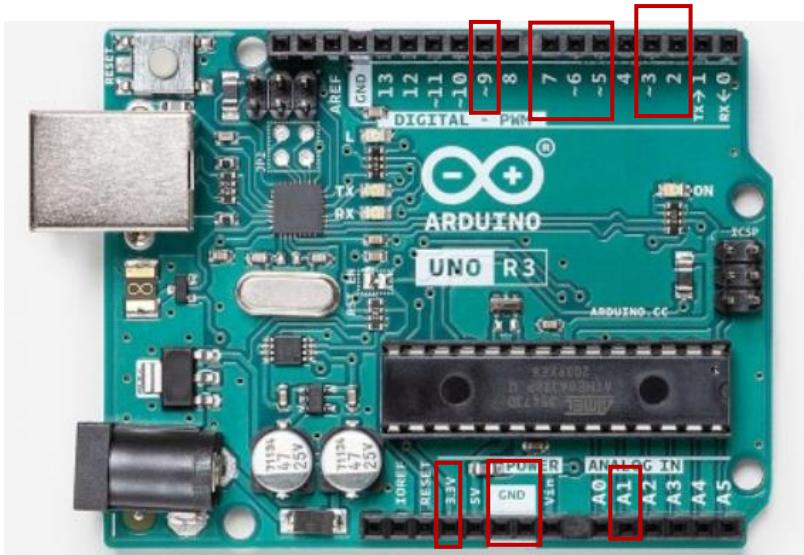
Ayrıca, boost converter çıkış geriliminin sabit tutulabilmesi için Arduino UNO'nun A1 analog pini üzerinden gerilim geri beslemesi sağlanacaktır. Bu geri besleme, boost converter çıkışına bağlı bir gerilim bölücü devresi aracılığıyla yaklaşık 3V seviyesine indirilen gerilim sinyalinin ölçülmesiyle gerçekleştirilir. Arduino, bu analog gerilim değerini okuyarak çıkış gerilimindeki sapmaları algılar ve PWM sinyalinin duty cycle'ını (görev döngüsü) ayarlayarak çıkış gerilimini kararlı tutar. Böylece, gerilim değişimlerine karşı dinamik bir tepki verilerek hedeflenen sabit çıkış gerilimi sağlanmış olur.

Projenin ikinci kısmında, Arduino UNO, PID kontrol sistemi kullanarak enkoderli bir DC motorun hız kontrolünü gerçekleştirecektir. Bu aşamada, Arduino'nun belirli pinleri motorun hızını düzgün bir şekilde kontrol etmek için kullanılacaktır. Arduino UNO'nun 3V3 ve GND pinleri ile beslemesi sağlanan enkoderli DC motorun dönüş hızını ölçmek için Enkoder A ve Enkoder B pinleri kullanılacaktır. Bu pinler, motorun dönüşünü izlemek için gerekli olan enkoder sinyallerini alacaktır. Pin 2 ve Pin 3'ün seçilmesinin nedeni, bu pinlerin dijital dış kesme (external interrupt) özelliklerine sahip olmalarıdır [24]. Bu özellik, yüksek hızda gelen sinyalleri hızlı ve kesintisiz bir şekilde işleyememizi sağlar. Bu sayede, motorun dönüş sayısı doğru bir şekilde hesaplanarak hız bilgisi alınabilir. Enkoder A (Pin 2), enkoderin A kanalından gelen sinyali alacak, Enkoder B (Pin 3) ise B kanalından gelen sinyali alacaktır. Bu iki kanal arasındaki faz farkı, motorun dönüş yönünü belirlemek için kullanılacaktır.

Motorun hızını ayarlamak için PWM sinyali kullanılacaktır. Bu PWM sinyali, Pin 5 ile sağlanacaktır. Pin 5, Arduino UNO'nun PWM çıkışı sağlama yeteneğine sahip Timer 0 pinlerinden biridir. PWM çıkışı sayesinde motorun hızını hassas bir şekilde ayarlanacak ve hız kontrolünü gerçekleştirilecektir [25].

Motorun yönünü kontrol etmek ise Pin 6 ve Pin 7 kullanılacaktır. Bu pinler, motor sürücüsüne bağlı olan IN_1 ve IN_2 pinleridir. IN_1 ve IN_2 arasındaki mantıksal düzey değişimleri motorun yönünü belirler. IN_1 yüksek ve IN_2 düşük olduğunda motor ileri yönde donecek, tersi durumda ise geri yönde donecektir [26].

Sonuç olarak projenin ikinci kısmında, Arduino UNO kartı üzerinde yer alan 3V3, GND, 2, 3, 5, 6, 7, 9 ve A1 pinler kullanılacaktır.



Şekil 17: Arduino UNO üzerindeki kullanılacak pinler [2]

2.4 Enkoderli DC Motor

2.4.1 DC Motor Kullanım Alanları

Doğru akım (DC) motor, drone'lar, robotik sistemler ve elektrikli cihazlarda kritik bir rol oynar.

Fırçalı DC motor, en yaygın kullanılan DC motor türüdür ve motorun dönen parçası olan armatüre doğru akım elektrik iletmek için fırçalar kullanır. Bu süreç, mekanik hareket üreten elektrik akışını sürekli olarak değiştirir. Fırçalı DC motor, basit tasarımlı, geniş bulunabilirliği, kolay kontrol edilebilirliği ve iyi güvenilirliği sayesinde çeşitli uygulamalarda cazip bir seçenektedir [27].

DC motorlar, AC motorlar ile kıyaslandığında maliyetleri düşüktür. Ayrıca DC motorlar, çok basit ve kararlı kontrol yöntemleri ile uyum sağlamaktadırlar. Diğer bir avantajı ise yüksek verimlik ve ani oluşan yük artışlarında yüksek başlama torkudur. Fakat periyodik bakım ihtiyaçları, çıkışların mekanik olarak kısa zamanda aşınması, akustik gürültü, parlama, fırçanın verime etkisi gibi bazı alanlarda yetersiz kalan DC motorlara alternatif olarak fırçasız DC motorlar ortaya çıkmıştır. Gürültülü çalışmamaları, bakım gerektirmemeleri, dinamik cevaplarının hızlı olması, tork karakteristiğinin iyi olması, verimli çalışmaları gibi avantajları sayesinde fırçasız DC motorlar tercih edilebilmektedir [27].

DC motorlarının yüksek verimlilik ve kontrol edilebilirlik sunduğunu, ayrıca uzun bir çalışma ömrüne sahip olduklarını gördük. Peki, bunlar ne için kullanılır? Verimlilikleri ve uzun ömürleri

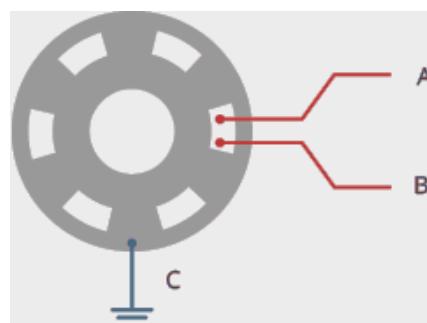
sayesinde, sürekli çalışan cihazlarda yaygın olarak kullanılırlar. Uzun bir süredir çamaşır makineleri, klima sistemleri ve diğer tüketici elektroniği cihazlarında kullanılmıştır; daha yakın zamanda ise, yüksek verimlilikleri sayesinde enerji tüketimini önemli ölçüde azaltmaya katkıda bulunan fanlarda görülmektedir [27].

Ayrıca, vakum makinelerini çalıştmak için de kullanılmaktadırlar. Bir durumda, kontrol programındaki bir değişiklik, dönme hızında büyük bir artışa yol açtı bu motorların sunduğu mükemmel kontrol edilebilirliğinin bir örneğidir [27].

DC motorları, sabit disk sürücülerini döndürmek için de kullanılmaktadır. Dayanıklılıkları sayesinde sürücüler uzun vadede güvenilir bir şekilde çalışmaya devam ederken, güç verimlilikleri de bu alanda giderek daha önemli hale gelen enerji tasarrufuna katkıda bulunmaktadır [28].

2.4.2 Enkoder Çalışma Prensibi

Enkoderin içinde, ortak topraklama olan C pinine bağlı yivli bir disk bulunur. Ayrıca, A ve B olmak üzere iki temas pimi vardır. Aşağıda gösterildiği gibi, düğmeyi çevirdiğinizde A ve B, dönme yönüne bağlı olarak sırayla ortak topraklama pini C ile temas eder.



Şekil 18: Rotasyonel enkoder modeli [6]

Ortak topraklama ile temas ettiklerinde, iki sinyal üretilir. Bu sinyaller 90° faz farkına sahiptir çünkü bir pin, diğerinden önce topraklama ile temas eder. Düğme saat yönünde çevrildiğinde, A pimi B piminden önce topraklama ile temas eder. Düğme saat yönünün tersine çevrildiğinde ise B pimi, A piminden önce topraklama ile temas eder.

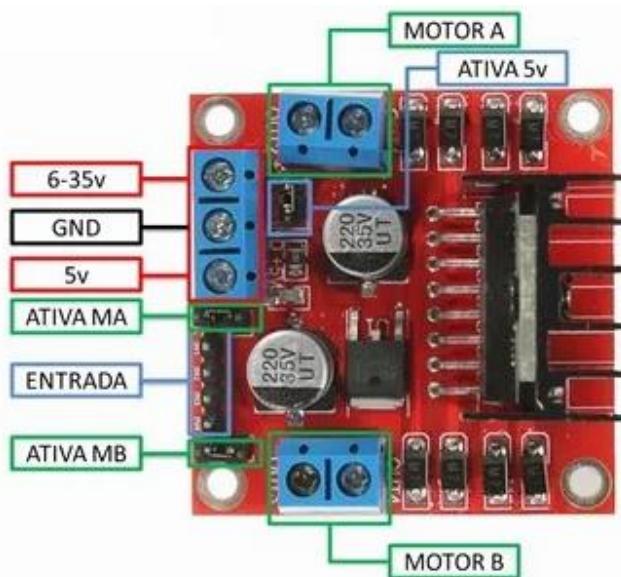
Her bir pinin topraklama ile ne zaman bağlantı kurduğunu veya bağlantıyı kestiğini izleyerek, düğmenin hangi yönde döndüğünü belirleyebiliriz [29].

2.4.3 Arduino ile Enkoderli DC motor ve L298N Motor Sürücüsü Kullanımı

L298N motor sürücüsü, Arduino UNO ile kullanılarak motorun hareketini kontrol etmek ve enkoder'den gelen sinyallerle hassas geri bildirim almak için tasarlanmıştır. Bu bağlantınlarda, motorun pozitif ve negatif terminalleri sırasıyla L298N motor sürücüsü üzerindeki OUT1 ve OUT2 pinlerine bağlanır. 12V bataryanın pozitif ucu, motor sürücünün VCC pinine, negatif ucu ise motor sürücünün GND pinine bağlanarak gücün sürücüyü beslemesi sağlanır. Arduino UNO'nun 5 numaralı pini, L298N motor sürücüsü üzerindeki ENA pinine bağlanır ve bu bağlantı PWM sinyali kullanılarak motorun hızını kontrol etmeye olanak tanır. Yön kontrolü için IN1 ve IN2 pinleri sırasıyla Arduino'nun 6 ve 7 numaralı pinlerine bağlanır.

Motor üzerindeki enkoder güç ve sinyal pinleri de uygun şekilde Arduino'ya entegre edilir. Enkoder gücü sağlamak için motorun enkoder VCC pini, Arduino'nun 5V pinine bağlanırken, enkoderin GND pini Arduino'nun GND pinine bağlanır. Sinyal pinlerinden Encoder A, Arduino'nun D2 pinine; Encoder B ise 6 numaralı pini bağlanır. Bu bağlantılar, Arduino'nun interrupt özelliklerini kullanarak enkoder sinyallerini hassas bir şekilde okuyabilmesini sağlar.

Son olarak, 12V batarya hem motor hem de motor sürücü için güç kaynağı olarak kullanılır. Bataryanın pozitif ucu L298N'in VCC pinine, negatif ucu ise GND pinine bağlanarak tüm sistemin enerji ihtiyacı karşılanır [30].



Şekil 19: L298N DC motor sürücü kartı [7]

3. PROJE UYGULAMASI

3.1 Hesaplamalar

Bu bölümde, 5 V'den 12 V ve 2 A çıkış elde etmek üzere tasarlanan DC-DC Boost dönüştürücünün bileşen seçiminde ve performans analizinde kullanılan temel hesaplamalar açıklanmaktadır. Hesaplamalar, “Teorik Altyapı ve Analizler” kısmında bahsedilen formüller doğrultusunda, gerçek donanım değerlerinin belirlenmesini sağlayacak şekilde adım adım anlatılmıştır.

- Giriş Gerilimi: $V_g = 5 \text{ V}$
- İstenilen çıkış gerilimi: $V_c = 12 \text{ V}$
- Çıkış akımı (istenilen): $I_c = 2 \text{ A}$
- Anahtarlama Frekansı: $f_p = 24 \text{ kHz}$

3.1.1 Dönüşüm Oranı (Duty Cycle) Hesabı

Daha önceden Teorik Altyapı ve Analizler bölümünde bahsedilen (8) numaralı bağıntı ile aşağıdaki gibi hesaplanmıştır.

$$V_c = \frac{V_g}{1 - D}, \quad D = 1 - \frac{V_g}{V_c} = 1 - \frac{5}{12} = 0,5833 \approx \%58,33$$

Sonuç olarak MOSFET, her anahtarlama döngüsünün yaklaşık %58,33'ü boyunca kapalı (enerji biriktiren) konumda çalışacaktır.

3.1.2 Giriş Akımı Hesabı

Kararlı durumda Boost dönüştürücüde giriş gücü ile çıkış gücü yaklaşık olarak eşit kabul edilir. Dolayısıyla (9) bağıntısı ile I_g değeri bulunur.

$$I_g = \frac{1}{1 - D} I_c = \frac{1}{1 - 0,5833} \times 2 \approx 4,8 \text{ A}$$

3.1.3 Anahtarlama Frekansı ve Sürelerinin Hesabı

Daha önce projenin gerektirdiği hızlı tepki, küçük boyutlar ve yeterli verim değerleri dikkate alınarak anahtarlama frekansı olarak $f_p = 24 \text{ kHz}$ belirlenmiştir. Buna karşılık bir tam periyot (anahtar açma–kapama döngüsü) süresi aşağıdaki gibidir.

$$T_p = \frac{1}{f_p} = \frac{1}{24 \times 10^3} = 41,666 \mu\text{s}$$

Buradan yola çıkarak dolum ve boşaltım anlarını hesaplayabiliriz. Anahtarın kapalı kaldığı yanı dolum anında kalma süresi hesaplanır,

$$T_d = D \times T_p = 0,5833 \times 41,666 \mu\text{s} = 24,304 \mu\text{s}$$

Anahtarın boşaltım esnasında kaldığı süre ise aşağıdaki gibidir.

$$T_b = T_p - T_d = 41,666 \mu\text{s} - 24,304 \mu\text{s} = 17,362 \mu\text{s}$$

3.1.4 Endüktans Hesabı

Endüktans değeri hem dalgalanmaları kontrol etmek hem de CCM (sürekli iletim modu) şartlarını korumak açısından kritik öneme sahiptir. Hesaplamlar aşağıdaki adımlar izlenerek yapılmıştır.

3.1.4.1 Ripple Akımı Hedefinin Belirlenmesi

Endüktans dalgalanma akımı için çıkış akımının %20 ila %40'ı arasında bir değer olarak tahmin edilir. Bu projede %40 olarak hesaplarımıza devam ettik.

$$\Delta I_{L,\text{hedef}} = 0,4 \times I_c \times \frac{V_c}{V_g} = 0,4 \times 2 \times \frac{12}{5} = 1,92 \text{ A}$$

Bu değer, endüktansın her anahtarlama periyodunda akımının yaklaşık olarak 1,92 A kadar dalgalanacağı anlamına gelir.

3.1.4.2 Endüktansdaki “Enerji Biriktirme” Akım Değişimi

Anahtar kapalıken (MOSFET off konumunda) endüktansa uygulanan voltaj

$$\Delta I_{L,p} = \frac{V_g}{L} \times T_d = \frac{5}{L} \times 24,304 \times 10^{-6}$$

İle hesaplanır, burada $\Delta I_{L,p}$ değerini 1.92 A değerine eşitlediğimizde ardından L değerini hesaplayabiliriz.

$$L = \frac{5 \times 24,304 \times 10^{-6}}{1,92} = 63,297 \mu\text{H}$$

Hesaplama sonucunda pratikte kolay bulunabilen ve bir miktar güvenlik marjı sağlayan 100 μH değeri seçilmiştir.

3.1.4.3 Endüktanstaki “Enerji Verme” Akım Değişimi

Anahtar açıkken (MOSFET on konumunda) endüktansa uygulanan voltaj

$$\Delta I_{L,n} = \frac{V_c - V_g}{L} \times T_b = \frac{7}{L} \times 17,362 \times 10^{-6}$$

İle hesaplanır. Bu proje dahilinde endüktans değerimiz $L = 100 \mu\text{H}$. Denklemde yerine koyduğumuz takdirde

$$\Delta I_{L,n} = \frac{7}{100 \times 10^{-6}} \times 17,362 \times 10^{-6} \approx 1,215 \text{ A}$$

3.1.5 Çıkış Kondansatörü Seçimi

Çıkış tarafında kullanılan kondansatör, anahtarlama sırasında ortaya çıkan gerilim dalgalanmalarını belli bir seviyenin altına sınırlamaktadır.

3.1.5.1 Maksimum Kabul Edilebilir Çıkış Dalgalanması

Projeye göre çıkış gerilimi 12 V iken, %5'lik toleransla en fazla 0,6 V dalgalanma kabul edilecektir.

$$\Delta V_{C,\max} = 12 \text{ V} \times 0,05 = 0,6 \text{ V}$$

3.1.5.2 Kondansatör Seçimi

Kabul edilen maksimum çıkış voltajı dalgalanması ve diğer değerler aşağıdaki bağıntıda yerine yazılarak çıkış kondansatörünün kapasitesi hesaplanmaktadır.

$$\Delta V_C = \frac{D \times I_\zeta}{f_p \times \Delta V_C} \Rightarrow C = \frac{D \times I_\zeta}{f_p \times \Delta V_C}$$

Değerleri yerine koyulduğunda aşağıdaki sonuç elde edilmektedir.

$$C = \frac{0,5833 \times 2}{24 \times 10^3 \times 0,6} = \frac{1,1666}{14,400} \approx 0,00008201 \text{ F} = 82,01 \mu\text{F}$$

Bu değer, teorik olarak minimum gereklili kapasitans değerini verir.

3.2 Kullanılan Bileşenler ve Gerekçeleri

Giriş Kondansatörü (100 μF / 35V, Elektrolit)

Boost converter devresinin girişine paralel olarak yerleştirilen 100 μF /25 V elektrolit kondansatör, girişte oluşabilecek ani gerilim değişimlerini (voltage ripple) sönmeyerek güç kaynağından gelen gerilimin daha kararlı bir şekilde dönüştürücüye ulaşmasını sağlar. Bu sayede devreyi besleyen kaynaktaki parazitlerin ve ani yük geçişlerinin etkisi azaltılırken, MOSFET'in anahtarlama anlarında oluşabilecek gerilim düşümlerinin önüne geçilmiştir.

Endüktans (100 μH , 10 A)

Sistemde kullanılan 100 μH değerindeki endüktans, teorik olarak hesaplanan 63 μH değerinin üzerinde seçilerek çıkıştaki akım dalgalanmasının azaltılması hedeflenmiştir. Ayrıca 10 A'lık

akım taşıma kapasitesine sahip olması sayesinde, sistemdeki 4,8 A ortalama ve 1,8 A dalgalanma içeren çalışma koşulları altında güvenli ve kararlı bir şekilde çalışması sağlanmaktadır.

Projenin test aşamalarında, MOSFET anahtarlama sinyalinde gözlemlenen gürültü nedeniyle, piyasada yaygın olarak şok bobini (choke coil) olarak adlandırılan ve elektromanyetik girişimi (EMI) bastırma özelliği bulunan bir endüktans kullanılması uygun görülmüştür. Bu sayede devredeki gürültü kaynaklı bozulmaların önüne geçilerek, sistemin kararlılığı ve elektromanyetik uyumluluğu artırılmıştır.

NMOS Sürücü Entegresi (TC4427)

MOSFET'in gate kapasitansının yüksek frekansta hızlı bir şekilde şarj ve deşarj edilmesi için yüksek akım sağlayabilen bir gate sürücü kullanılması gereklidir. Bu amaçla tercih edilen TC4427 entegresi, sistemin ilk çalışmasında 5 V'luk giriş gerilimi ile IRFZ44N MOSFET'in gate'ine yeterli gerilim sağlayarak MOSFET'in kararlı anahtarlama yaparak 24 kHz frekansında verimli ve düşük geçiş kayıpları bir sürüsü sağlamaktadır.

Zener Diyot (1 W, 18 V)

Devrede koruma elemanı olarak kullanılan 18 V/1 W Zener diyot, MOSFET gate sürücü hattının gerilimini güvenli sınırlar içinde tutmak amacıyla devreye dahil edilmiştir. Ayrıca ters yönlü gerilim durumlarında hassas bileşenlerin zarar görmesinin önlenmesine katkıda bulunarak sistemin güvenliği artırılmıştır.

MOSFET (IRFZ44N)

Anahtarlama elemanı olarak kullanılan IRFZ44N MOSFET, 49 A sürekli ve 160 A tepe akım taşıma kapasitesiyle sistemdeki maksimum akım gereksinimlerinin oldukça üzerinde bir dayanıklılık sunmaktadır. 55 V'luk maksimum drain-source gerilim sınırı sayesinde, sistemin 12 V çıkış gerilimi ve yük kaynaklı ani gerilim artışlarına karşı yeterli koruma sağlamaktadır.

Ayrıca, düşük giriş kapasitansı sayesinde, TC4427 sürücü ile 24 kHz frekansında hızlı ve güvenli anahtarlama yapılmaktadır, böylece geçiş kayıpları en aza indirilmektedir.

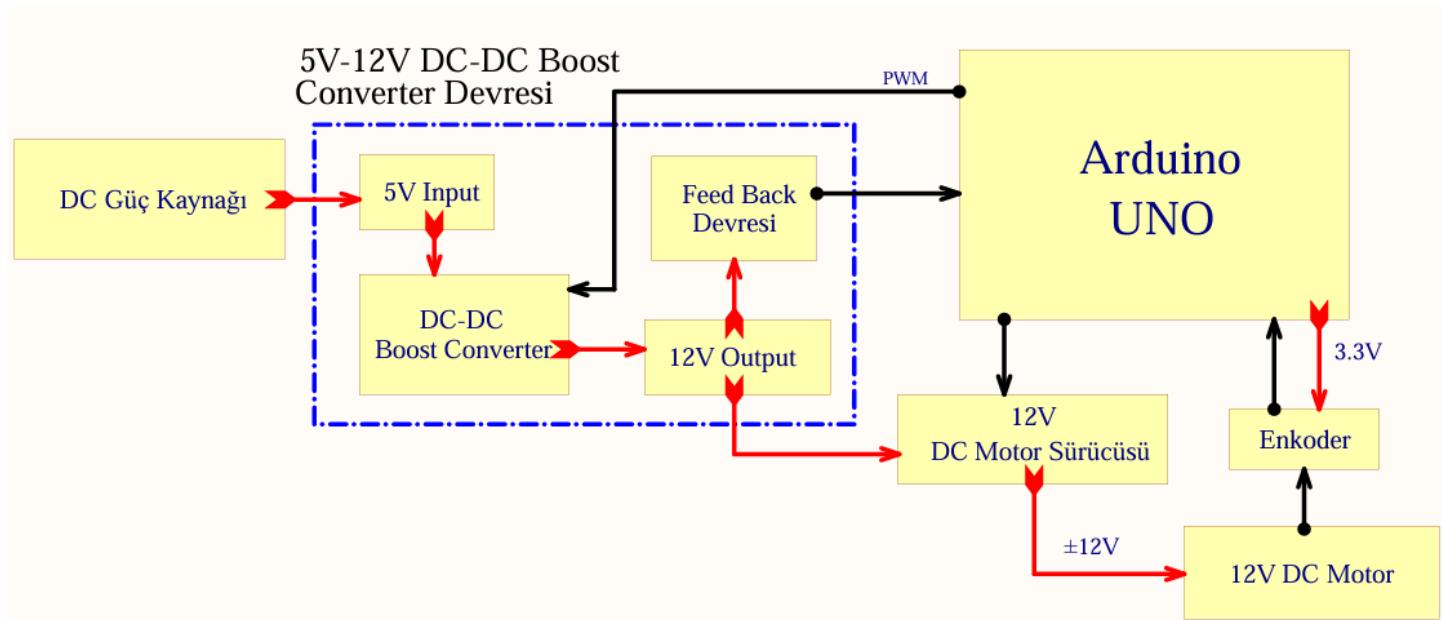
Schottky Diyot (MBR2060CT)

Çıkış doğrultma ve ters akım koruması amacıyla kullanılan MBR2060CT Schottky diyot, düşük ileri gerilim düşüşü (forward voltage drop) ve hızlı anahtarlama özellikleri sayesinde yüksek verimli bir doğrultma sağlamaktadır. 20 A sürekli akım ve 60 V ters gerilim dayanımı ile sistemin çıkış akımı ve gerilim seviyelerine uygun bir çalışma güvenliği sunmaktadır. Düşük iletim kaybı sayesinde ısınma azaltılarak verimlilik artırılmış, aynı zamanda ani yük değişimlerinde çıkış geriliminin kararlılığı korunmuştur.

Çıkış Kondansatörü (470 μ F / 35V, Elektrolit)

Çıkış kondansatörü olarak 470 μ F/35 V elektrolit tipi tercih edilmiş; bu değer, teorik olarak hesaplanan 82 μ F'lik minimum değerin oldukça üzerinde tutulmak suretiyle özellikle düşük frekanslı çıkış dalgalanmalarının etkin bir biçimde bastırılması sağlanmıştır. Ükte anı değişimler meydana geldiğinde kondansatörün depoladığı enerjinin devreye hızlıca aktarılmasıyla çıkış gerilimindeki anı düşüşlerin önlenmesi, sistemin dinamik tepkisinin ve kararlılığının iyileştirilmesi mümkün kılınmıştır.

3.3 Blok Diyagram



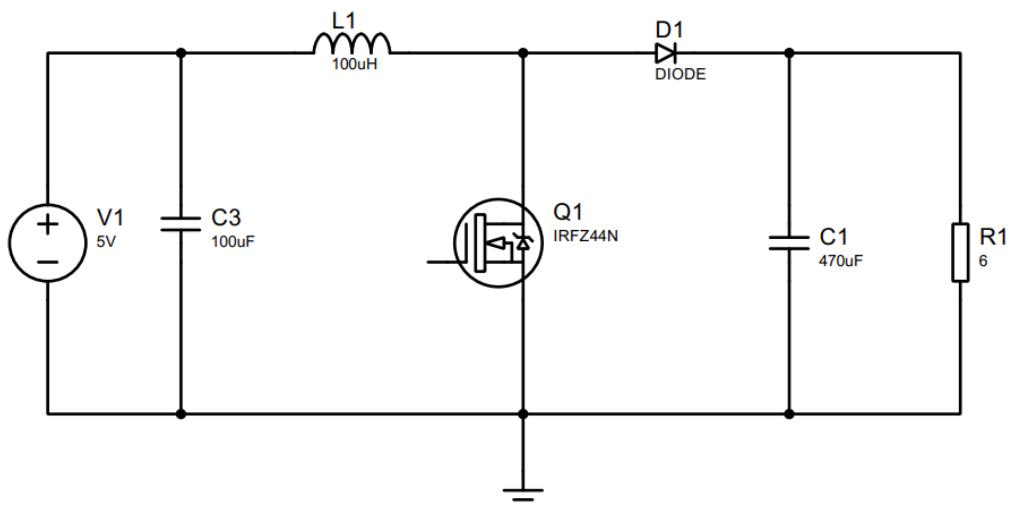
Şekil 20: Proje'nin blok diyagramı

3.4 Proteus Simülasyonu

Bu bölümde, bir önceki “Kullanılan Bileşenler ve Gerekçeler” ile “Hesaplamalar” başlıklarında yer alan teorik değerleri kullanarak Proteus ortamında oluşturulan simülasyon anlatılacaktır.

Başlangıç olarak devrede bir adet 5 V DC kaynak devre girişine bağlanarak hem Arduino Uno'ya hem de devre elemanlarına güç sağlanmıştır. Güç kaynağın paralel olacak şekilde filtre amacı ile 470 μ F değerinde bir adet kondansatör bağlandı. Sırası ile bobin, mosfet ve diyon bağlantıları yapıldı. Yük paralel olacak şekilde 470 μ F kondansatör bağlandı. “

Başlangıç olarak devrede bir adet 5 V DC kaynak devre girişine bağlanarak devre elemanlarına güç sağlanmıştır. Güç kaynağın paralel olacak şekilde filtre amacı ile 470 μ F değerinde bir adet kondansatör bağlandı. Daha öncesinde seçmiş olduğumuz 100 μ H değerindeki bobin elemanı, IRFZ44N'in “Drain” bacağına doğrudan seri bağlanmıştır. Drain ile Bobin arasındaki bu bağlantı, MOSFET kapalı konumdayken bobinin enerji depolamasını sağlar. Bobinin boşaltma ucu bir adet Schottky diyot üzerinden çıkış kondansatörüne paralel bağlanır. Bu diyot bize endüktansdan gelen enerjinin yalnızca yük yönüne akmasını sağlar ve ters yönde geri besleme olmasını engeller. Çıkış kondansatörlerinin pozitif ucu 60 Ω 'luk bir dirence bağlanmıştır. Yük direncinin negatif bacağı, giriş ve çıkış kondansatörlerinin negatif bacakları, MOSFET'in source bacağı devrenin toprak hattına bağlandı.

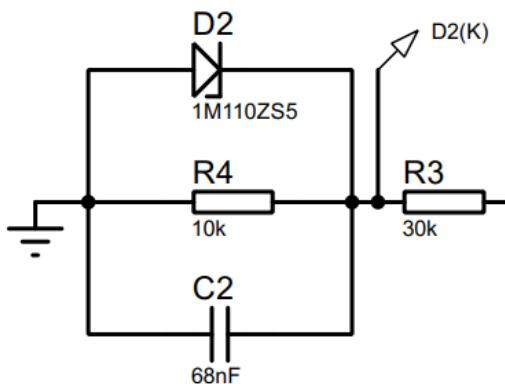


Şekil 21: DC-DC boost converter devre şeması

Boost dönüştürücü çıkış geriliminin takip edilmesi ve bu izlenen bu değer ile duty cycle değerinin kendisini düzeltmesi için yapılmıştır. Bu ağıda $R3 = 30 \text{ k}\Omega$ ve $R4 = 10 \text{ k}\Omega$ değerli dirençler kullanılarak, çıkıştaki maksimum 12 V'luk gerilim yaklaşık olarak 3 V seviyesine düşürülmüştür. Aşağıdaki bağıntı ile hesaplanmıştır.

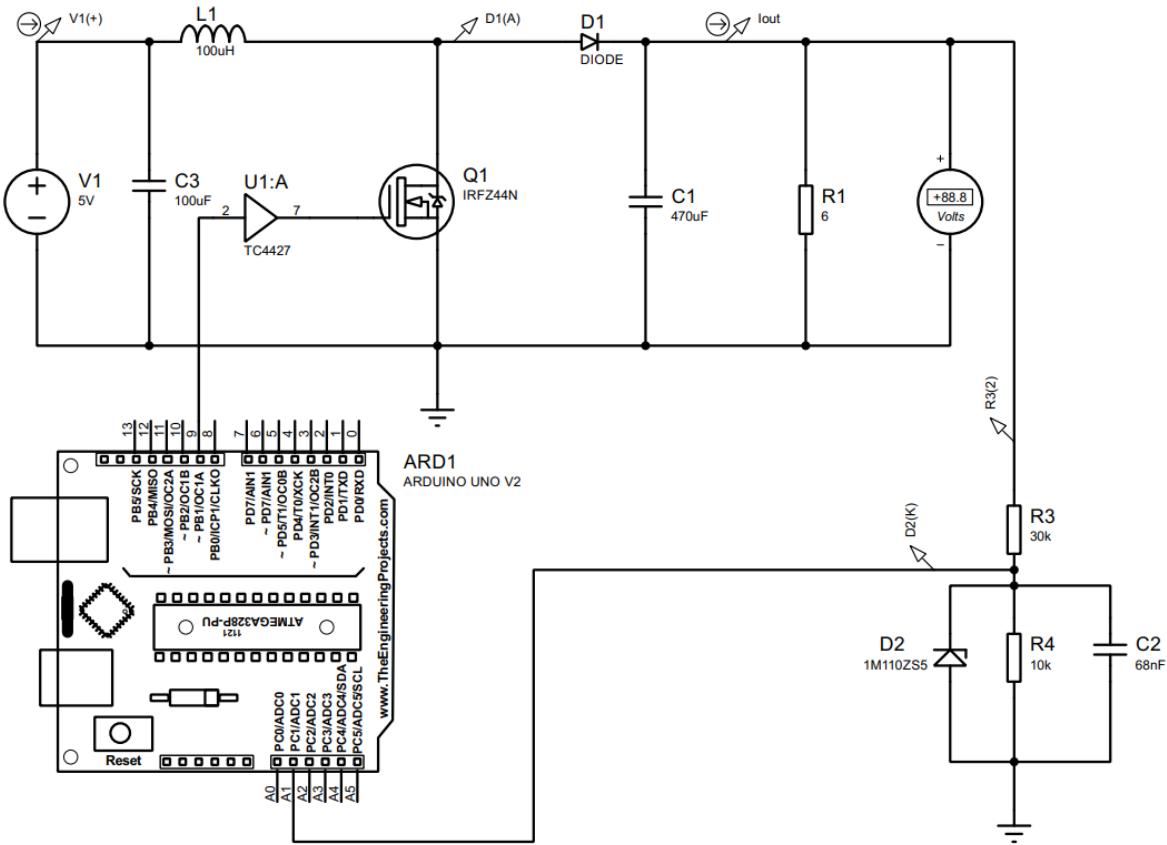
$$V = V_{\text{out}} \times \frac{R4}{R3 + R4} = 12 \text{ V} \times \frac{10 \text{ k}\Omega}{30 \text{ k}\Omega + 10 \text{ k}\Omega} \approx 3 \text{ V}$$

Gerilim bölücünün çıkışına, yüksek frekanslı parazitleri azaltmak amacıyla 68nF kapasiteli bir kondansatör paralel bağlanmıştır. Arduino'yu korumak için devreye paralel bir Arduino UNO'nun okuyabileceği değer olan 5 volta sınırlamak için $5.1\text{V} / 1\text{W}$ değerinde bir zener diyon eklenmiş ve ani gerilim yükselmelerine karşı koruma sağlanmıştır. Geri besleme devresine eklenen diyon, direnç ve kondansatör ise toprağa bağlanarak devrenin kararlılığı artırılmıştır.



Şekil 22: Geri besleme devre şeması

Sonuç olarak, Arduino UNO kartından oluşturulan PWM sinyali TC4427 entegre üzerinden IRFZ44N MOSFET'e aktarılır. Bu durum bize MOSFET'in çok kısa aralıklarla aç kapa işlemini yapmasını sağlar. Girişten sağlanan 5 V'luk kaynak önce bobine sonrasında MOSFET, schottky diyon, kondansatör ve çıkış yükünün üzerinden geçer. Çıkış yükünün üzerinden alınan geri besleme gerilimi Arduino kart üzerinden okuması yapılır. Simülasyon üzerinde hazırlanana devrenin son hali aşağıdaki şekildedir.



Şekil 23: Geri beslemeli DC-DC boost converter devresi şematiği

3.5 Yazılım ve Kod Açıklamaları

Bu kısımda Arduino üzerinden kullanılan yazılım ve kodları açıklanacaktır. Kullanılan sabitler ve değişkenler, interrupt servis rutinleri, PWM sürücü ayarları, ölçüm ve kontrol döngüleri ayrıntılı şekilde anlatılacaktır.

3.5.1 Sabit Tanımlamaları ve Pin Atamaları

```

const uint16_t FREQUENCY_ICR1 = 662;
const float V_THRESH_V = 3.0;
const int FB_PIN = A1;
const int PWM_SUPPLY_PIN = 9;
const int RAW_THRESH = int(V_THRESH_V * 1023.0 / 5.0 + 0.5);
const uint16_t MAX_COUNT = uint16_t(0.7 * (FREQUENCY_ICR1 + 1) + 0.5);
const unsigned long INIT_TIME = 5000;

```

Başlangıçta, FREQUENCY_ICR1=665 değeri ile sistemin PWM frekansını yaklaşık 24 kHz seviyesinde sabit tutmak amacıyla Timer1 modülü üzerinden PWM periyodu ayarlanmıştır. Gerilim bölgücü çıkışı için tanımlanan V_THRESH_V = 3.0 değeri, sistemin referans eşik gerilimi

olarak belirlenmiş ve geri besleme yoluyla elde edilen ölçümelerin bu değerle karşılaştırılması sağlanmıştır. Geri besleme geriliminin ölçülmesi amacıyla FB_PIN olarak A1 numaralı Arduino analog girişi atanmıştır. PWM_SUPPLY_PIN olarak belirlenen 9 numaralı dijital pin, boost dönüştürücüye PWM sinyali gönderilmesinde kullanılmıştır.

Analog dijital çevirici (ADC) üzerinden okunan değerin, V_THRESH_V seviyesine karşılık gelen ham veri eşğini temsil eden RAW_THRESH hesaplanarak değerlendirmeler bu değere göre yapılmıştır. Ayrıca, duty cycle (D) değerinin ulaşabileceği en yüksek seviye MAX_COUNT ile sınırlanmıştır. Sistemin başlangıcında stabil bir çalışma ortamı sağlamak amacıyla INIT_TIME değişkeniyle 5 saniyelik bir gecikme süresi tanımlanmıştır.

```
float kp = 0.04;  
float ki = 0.00030;  
float kd = 0.005;
```

Kodun bu kısmında, kp, ki ve kd sırasıyla oransal, integral ve türevsel kazançları belirlenmiştir. Bu parametreler, motor hızı hatasının işlenerek uygun sürücü gerilimi oluşturulmasında kullanılır. Kullanılmış olan DC motorun ilgili parametrelerine sahip olmadığımız için sistemin dinamik davranışına uygun bir yanıt elde edilebilmesi amacıyla Ziegler–Nichols yöntemi referans alınarak deneysel olarak, deneme-yanılma yoluyla optimize edilmiştir.

```
const byte Encoder_A = 2;  
const byte Encoder_B = 3;  
const byte PWMPin = 5;  
const byte IN_1 = 6;  
const byte IN_2 = 7;
```

Bu kısımda, motorun konum ve hız bilgisinin elde edilebilmesi amacıyla, enkoderden gelen A ve B faz sinyallerinin okunacağı pinler sırasıyla Encoder_A=2 ve Encoder_B=3 olarak tanımlanmıştır. Motorun hız kontrolü için gerekli olan PWM sinyalinin çıkış yaptığı pin olarak PWMPin=5 numaralı dijital pin kullanılmıştır.

Ayrıca, L298N motor sürücü entegresinin yön kontrolünü sağlayan giriş pinleri olan IN_1 ve IN_2, sırasıyla 6 ve 7 numaralı dijital pinlere atanarak, motorun ileri ve geri yöndeği hareketleri bu pinler üzerinden kontrol edilebilir hâle getirilmiştir.

```

volatile uint16_t dutyCount;
unsigned long startMillis;
volatile long EncoderCount = 0;
unsigned long t, t_prev = 0;
int dt;
float Theta, RPM, RPM_target = 0;
float Theta_prev = 0;
float e, e_prev = 0, inte, inte_prev = 0;
float V, Vmax = 12.0, Vmin = -12.0;
unsigned long count = 0;
bool useSerialPlotter = true;

```

Bu kod bloğunda, dutyCount ve EncoderCount, sırasıyla PWM sinyalinin darbe genişliği ve enkoder sayısını tutmak için kullanılırken, startMillis ve t ve t_prev zaman takibini sağlamaktadır. Theta ve RPM motorun konum ve hızını temsil ederken, RPM_target hedef hızı belirtir. e, inte, intre_prev ve V gibi değişkenler PID kontrol hesaplamaları için kullanılmaktadır. Son olarak, Vmax ve Vmin değerleri çıkış geriliminin sınırları belirlenmiştir.

3.5.2 Interrupt Servis Rutinleri

```

void ISR_EncoderA() {
    bool PinB = digitalRead(Encoder_B);
    bool PinA = digitalRead(Encoder_A);
    EncoderCount += (PinB == LOW) ? ((PinA == HIGH) ? 1 : -1) : ((PinA == HIGH) ? -1 : 1);
}

void ISR_EncoderB() {
    bool PinA = digitalRead(Encoder_A);
    bool PinB = digitalRead(Encoder_B);
    EncoderCount += (PinA == LOW) ? ((PinB == HIGH) ? -1 : 1) : ((PinB == HIGH) ? 1 : -1);
}

```

Enkoderden alınan sinyallerin doğru şekilde işlenebilmesi amacıyla, A ve B fazlarındaki her değişim anında ilgili kesme servis rutinleri (ISR) devreye alınmıştır. ISR_EncoderA() ve ISR_EncoderB() fonksiyonlarında, karşı fazın lojik durumu okunarak motorun dönüş yönüne göre EncoderCount değeri artırılmakta veya azaltılmaktadır. Bu sayede, yalnızca faz değişimini değil aynı zamanda yön bilgisini de içeren bir sayımla mekanizması elde edilmiştir.

3.5.3 DC Motor Kontrol Sisteminin Başlatılması ve Yükseltici Devrenin Yapılandırması

```
void setup() {
    Serial.begin(115200);
    while (!Serial);

    pinMode(PWM_SUPPLY_PIN, OUTPUT);
    pinMode(Encoder_A, INPUT_PULLUP);
    pinMode(Encoder_B, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(Encoder_A), ISR_EncoderA, CHANGE);
    attachInterrupt(digitalPinToInterrupt(Encoder_B), ISR_EncoderB, CHANGE);
    pinMode(IN_1, OUTPUT);
    pinMode(IN_2, OUTPUT);
    pinMode(PWMPin, OUTPUT);

    cli();
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
    TCCR1A |= (1 << WGM11) | (1 << COM1A1);
    TCCR1B |= (1 << WGM13) | (1 << WGM12) | (1 << CS10);
    ICR1 = FREQUENCY_ICR1;
    TIMSK1 |= (1 << OCIE1A);
    dutyCount = uint16_t(0.25 * (FREQUENCY_ICR1 + 1) + 0.5);
    OCR1A = dutyCount;
    sei();
}
```

Bu kod bloğunda, sistemin temel donanım yapılandırmalarını gerçekleştirerek motor kontrolüne hazır hâle getirilmesini sağlamaktadır. Seri haberleşme 115200 baud hızında başlatılmış, PWM çıkışı ve motor kontrol pinleri çıkış olarak tanımlanmıştır. Enkoder pinleri INPUT_PULLUP modunda giriş olarak ayarlanmıştır. Ardından, attachInterrupt() fonksiyonları ile, Encoder_A ve Encoder_B pinlerine ait kesmeler CHANGE modu ile tanımlanmış, böylece hem yükselen hem de alçalan kenarlarda kesme yapılması sağlanmıştır. Elde edilen EncoderCount değişkeni, motorun açısal konumu (θ) ve dönüş hızı (RPM) gibi büyüklüklerin hesaplanmasıında temel veri olarak kullanılmaktadır. Bu yapı, motorun hem yön hem de hızının hassas biçimde belirlenmesi sağlamaktadır. PWM sinyalinin üretileceği Timer1 konfigürasyonu, hızlı PWM modunda çalışacak şekilde yapılandırılmış ve FREQUENCY_ICR1 değeri referans alınarak frekans belirlenmiştir. PWM çıkışının başlangıç duty cycle değeri %25 olarak ayarlanmış ve OCR1A register'ına atanmıştır. Global kesmeler etkinleştirilerek sei() zamanlamalı işlemlerin yürütülmesi sağlanmış; son olarak, sistemin çalışmaya başladığı an startMillis değişkenine kaydedilmiştir.

3.5.4 Ana Döngüde Besleme Gerilimi Kontrolü

```
void loop() {
    unsigned long now = millis();

    if (now - startMillis < INIT_TIME) {
        OCR1A = dutyCount;
    } else {
        static unsigned long lastMicros = 0;
        unsigned long m = micros();
        if (m - lastMicros >= 1000) {
            lastMicros += 1000;
            int raw = analogRead(FB_PIN);
            if (raw > RAW_THRESH) {
                if (dutyCount > 1) dutyCount--;
            } else {
                if (dutyCount < MAX_COUNT) dutyCount++;
            }
            OCR1A = dutyCount;
        }
    }
}
```

Bu kod bloğu, boost dönüştürücünün çıkış gerilimini izleyerek PWM duty cycle'ını dinamik biçimde ayarlamakta ve böylece çıkış gerilimini hedeflenen değerde sabit tutmayı amaçlamaktadır. Sistem ilk başlatıldığında, tanımlanan INIT_TIME süresi boyunca herhangi bir kontrol gerçekleştirilmeden dutyCount değeri doğrudan Timer1'in karşılaştırma kayıtçısına (OCR1A) yüklenmekte ve PWM çıkışı sabit bir seviyede tutulmaktadır. Bu başlangıç süresi, sistemin ve geri besleme devresinin kararlı bir duruma geçişini sağlamak amacıyla gereklidir.

Açılış evresi tamamlandığında, her yaklaşık 1 milisaniyede bir ADC üzerinden geri besleme pini (FB_PIN) okunarak ölçülen ham değer, daha önce belirlenmiş olan RAW_THRESH eşiği ile karşılaştırılmaktadır. Ölçülen değer bu eşiğin üzerinde ise, çıkış gerilimini azaltmak amacıyla dutyCount bir birim düşürülmektedir; tersi durumda ise, çıkış gerilimini artırmak amacıyla dutyCount bir birim artırmaktadır. Duty cycle değerinin aşırı yükselmesini engellemek için MAX_COUNT sınırı göz önünde bulundurulmakta ve bu sınıra ulaşması durumunda artırma işlemi durdurulmaktadır. Güncellenen dutyCount değeri her döngüde OCR1A register'ına aktarılmakta ve böylece PWM çıkışı anlık olarak yeni duty cycle'a göre güncellenmektedir.

3.5.5 PID Döngüsü ve Hız Ölçümü

```
static unsigned long lastPidTime = 0;
if (now - lastPidTime >= 10) {
    lastPidTime = now;
    t = now;
    Theta = EncoderCount / 900.0;
    dt = t - t_prev;
    if (dt > 0) {
        RPM = (Theta - Theta_prev) / (dt / 1000.0) * 60.0;
        static float rpmBuffer[5] = {0};
        static int bufferIndex = 0;
        rpmBuffer[bufferIndex] = RPM;
        bufferIndex = (bufferIndex + 1) % 5;
        float rpmSum = 0;
        for (int i = 0; i < 5; i++) rpmSum += rpmBuffer[i];
        RPM = rpmSum / 5.0;

        e = RPM_target - RPM;
        inte = inte_prev + (dt * (e + e_prev) / 2.0);
        V = kp * e + ki * inte + kd * (e - e_prev) / dt;
        if (V > Vmax) {
            V = Vmax;
            inte = inte_prev;
        }
        if (V < Vmin) {
            V = Vmin;
            inte = inte_prev;
        }
    }

    WriteDriverVoltage(V, Vmax);
    Theta_prev = Theta;

    t_prev = t;
    e_prev = e;
    inte_prev = inte;
}
}
```

Bu kod bloğunda, motor hızını kapalı çevrimle kontrol etmek amacıyla klasik bir PID algoritmasının uygulanmasını sağlamaktadır. lastPidTime değişkeni, son PID hesaplamasının yapıldığı anı milisaniye cinsinden tutmakta ve böylece her PID döngüsünün 10 ms aralıklarla çalışması `if (now - lastPidTime >= 10)` koşuluyla garanti altına alınmaktadır. Zaman değişkeni olarak `t = now` ifadesi ile güncel zaman kaydedilirken, `dt = t - t_prev` işlemiyle bir önceki ölçüme göre geçen süre milisaniye olarak hesaplanmaktadır. Bu süre pozitif olduğu sürece işlem devam

etmekte ve olası sıfır'a bölme hatalarının önüne geçilmektedir. Motorun açısal konumu, enkoder sayımı olan EncoderCount üzerinden $\Theta = \text{EncoderCount} / 900.0$ ifadesiyle radyan cinsinden belirlenmektedir. İki konum arasındaki farkın zaman farkına bölünmesiyle elde edilen değer, saniye cinsinden tura dönüştürülüyor, ardından 60 ile çarpılarak dakikadaki devir sayısı (RPM) hesaplanmaktadır.

Elde edilen RPM değeri, kısa süreli dalgalanmaları azaltmak ve sistemin kararlılığını artırmak amacıyla beş örneklemeli bir tampon dizi olan rpmBuffer içerisinde kaydedilmekte ve ortalaması alınarak filtrelenmiş RPM değeri elde edilmektedir. Hedef hız ile ölçülen hız arasındaki fark $e = \text{RPM_target} - \text{RPM}$ ifadesiyle bulunmakta, bu hata değeri kullanılarak inte değişkeni üzerinden trapezoidal entegrasyon yöntemiyle birikimli hata hesabı yapılmaktadır.

Oransal, integral ve türevsel bileşenlerin birleşimiyle elde edilen V kontrol sinyali, klasik PID denklemi olan $V = kp * e + ki * \text{inte} + kd * (\text{e} - \text{e_prev}) / dt$ ile hesaplanmaktadır. Kontrol sinyalinin sınırlandırılması için Vmax ve Vmin limitleri uygulanmakta; bu sınırların aşılması durumunda integral terimi sabit tutularak denge korunmaktadır.

Hesaplanan V değeri, WriteDriverVoltage(V, Vmax) fonksiyonu aracılığıyla uygun bir PWM duty cycle değerine dönüştürüülerek motor sürücüsüne aktarılmaktadır. Son olarak, Theta_prev, t_prev, e_prev ve inte_prev gibi geçmiş veriler, bir sonraki kontrol çevriminde kullanılmak üzere güncellenmektedir. Bu yapı, motorun hem hızlı hem de kararlı bir şekilde hedeflenen hız değerine ulaşmasını sağlayan standart bir PID kapalı çevrim kontrol sistemini temsil etmektedir.

3.5.6 Seri Monitör ve Serial Plotter Çıktıları

```
static unsigned long lastPrint = 0;
if (now - lastPrint >= 100) {
    lastPrint = now;
    float dutyCyclePct = 100.0 * dutyCount / float(FREQUENCY_ICR1 + 1);
    if (useSerialPlotter) {

        // === Serial Plotter: Value1 = RPM_target, Value2 = RPM, Value3 = dutyCycle ===
        Serial.print("Target:");
        Serial.print(RPM target);
        Serial.print(" | Actual:");
        Serial.print(RPM);
        Serial.print("\t\t\tDutyCycle:");
        Serial.println(dutyCyclePct);
    } else {
```

```

Serial.print("Target:");
Serial.print(RPM_target);
Serial.print(" Actual:");
Serial.print(RPM);
Serial.print(" DutyCycle:");
Serial.println(dutyCyclePct);
}

}

static unsigned long lastVoltageTime = 0;
if (now - lastVoltageTime >= 100) {
    lastVoltageTime = now;
    float voltageADC = analogRead(FB_PIN) * 5.0 / 1023.0;
    float voltageOut = voltageADC * 4.0 * 1.0484;

    Serial.print(" \t\t\t\t Voltage: ");
    Serial.print(voltageOut, 2);
    Serial.println(" V");
}

```

Son olarak kodun bu kısmında, sistemin çalışma verilerinin belirli aralıklarla seri monitöre yazdırılması sağlanmaktadır. now - lastPrint >= 100 koşulu ile her 100 ms'de bir hedef hız (RPM_target), gerçek hız (RPM) ve PWM duty cycle değeri yüzde cinsinden hesaplanarak (dutyCyclePct) seri port üzerinden kullanıcıya iletilmektedir. useSerialPlotter bayrağı true ise bu veriler Serial Plotter formatında yazdırılır.

Ayrıca, now - lastVoltageTime >= 100 ifadesiyle yine her 100 ms'de bir geri besleme pininden (FB_PIN) alınan ADC değeri, analogRead(FB_PIN) * 5.0 / 1023.0 işlemiyle 0–5 V aralığına dönüştürülmektedir. Bu değer, gerilim bölücü oranı ve düzeltme katsayısı dikkate alınarak çıkış gerilimine çevrilmekte ve voltageOut olarak hesaplanmaktadır. Elde edilen çıkış gerilimi, iki ondalık hassasiyetle seri monitöre yazdırılarak kullanıcıya bilgi sunulmaktadır.

Arduino ile ilgili tüm kaynak kodlar için bkz. Ek-1

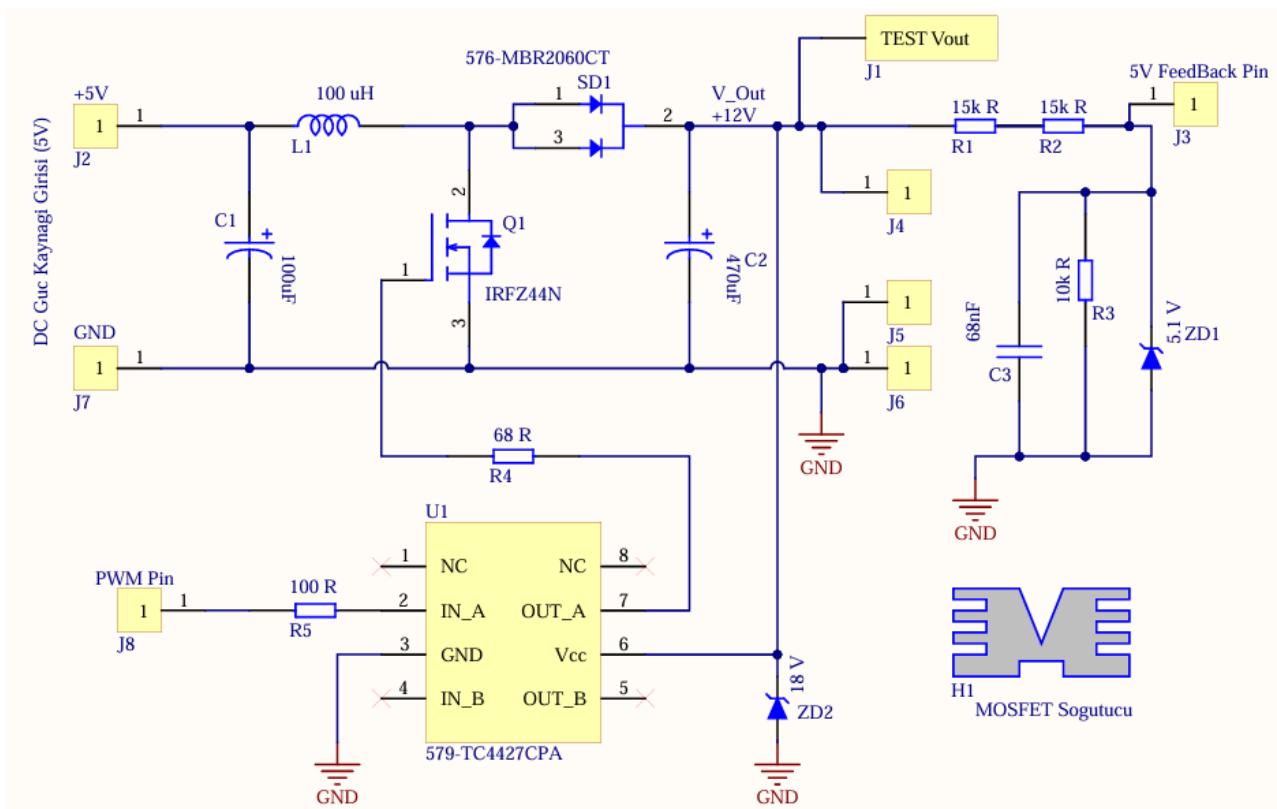
3.6 Gerçek Sistem Uygulaması ve Sonuçlar

Bu bölümde buraya kadar yapılan hesaplamalar, simülasyon ve yazılım adımlarını gerçek bir sistem üzerinde deneyip aldığımız sonuçları değerlendireceğiz.

Projenin uygulanabilirliğini değerlendirmek ve teorik olarak gerçekleştirilen hesaplamaların pratikteki karşılığını test edebilmek amacıyla ilk prototipleme çalışmaları breadboard üzerinde gerçekleştirilmiştir. Bu aşamada, önceden yapılan matematiksel hesaplar doğrultusunda belirlenen devre elemanları tedarik edilerek devre üzerinde yerleştirilmiştir. Breadboard üzerinde kurulan geçici devre düzeni ile, sistemin temel işlevselligi testlere tabi tutulmuştur. Özellikle boost converter devresinin anahtarlamalı çalışma yapısı incelenmiş; Arduino UNO üzerinden üretilen PWM sinyalinin IRFZ44N MOSFET’i sürme başarımı, MOSFET’in açma-kapama (anahtarlama) davranışları ve endüktans ile kondansatör arasındaki enerji aktarım döngüsü gözlemlenmiştir.

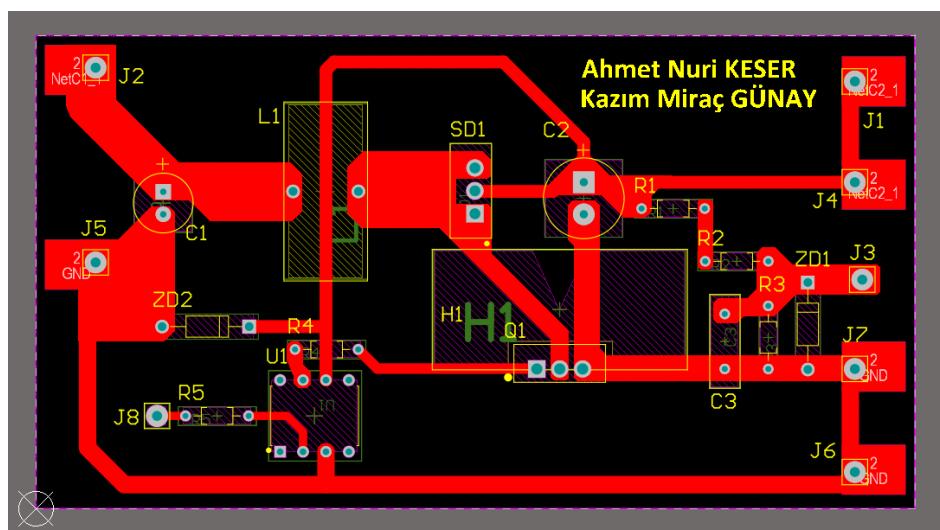
Bu testler sırasında, devre elemanlarının birbirleriyle uyumlu çalıştığı, bağlantıların doğru şekilde yapıldığı ve teorik modellemelere uygun bir gerilim yükseltmesinin sağlanmış olduğu tespit edilmiştir. Bununla birlikte breadboard (devre tahtası) üzerindeki geçici bağlantıların, sistemin uzun süreli çalışması için yeterli kararlılığı sağlayamadığı için temassızlık ve gürültü etkilerinin özellikle yüksek frekansta çalışan bir boost converter devresinde hata payını artırabileceği değerlendirilmiştir. Bu doğrultuda, testlerden elde edilen olumlu sonuçların ardından kalıcı ve güvenilir bir sistem altyapısı oluşturmak üzere PCB (baskı devre kartı) tasarıımı sürecine geçilmiştir.

PCB çizimi için öncelikle Altium programında her bir devre elemanı için standartlara uygun şekilde şematik ve PCB kütüphaneleri oluşturulmuştur. Oluşturulan bu kütüphanelerdeki komponentlerle aşağıdaki DC-DC boost converter devre şematiği oluşturulmuştur.

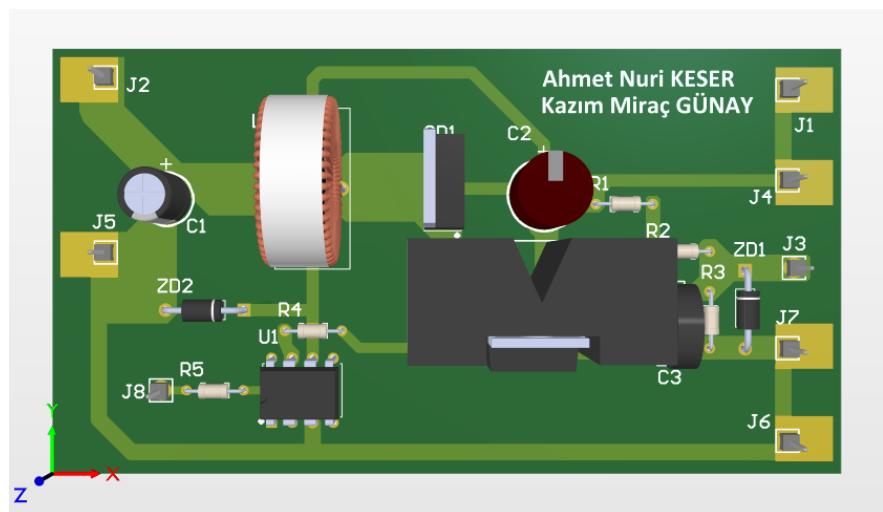


Şekil 24: DC-DC boost converter devre şematiği (Altium Designer)

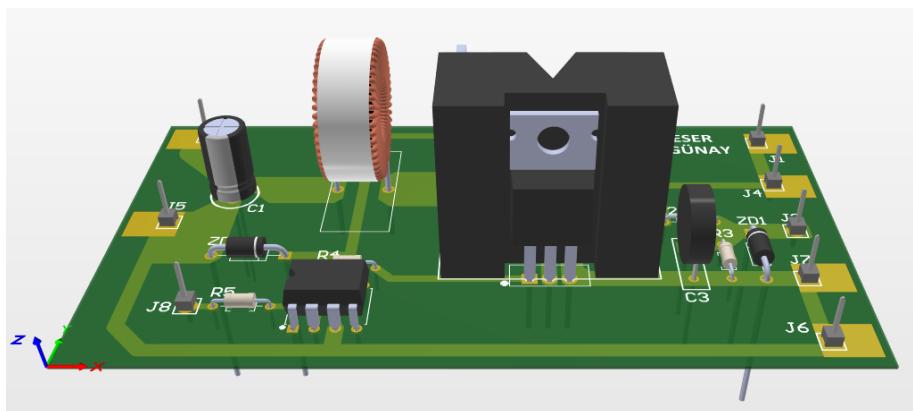
Ardından, çizilmiş olan bu devre şematiği boş bir PCB kart sayfasına aktarılarak komponentlerin konumlandırılması ve komponentler arası yollar çizilmiştir. Sonuç olarak, başarılı bir şekilde tek katmanlı bir PCB çizimi gerçekleştirilmiştir.



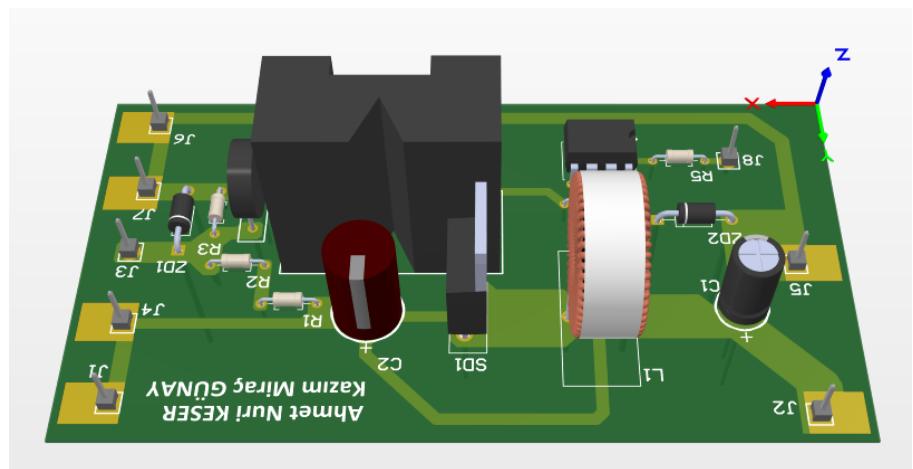
Şekil 25: PCB'nin 2D hali



Şekil 26: PCB'nin 3D hali (Yukardan bakış)

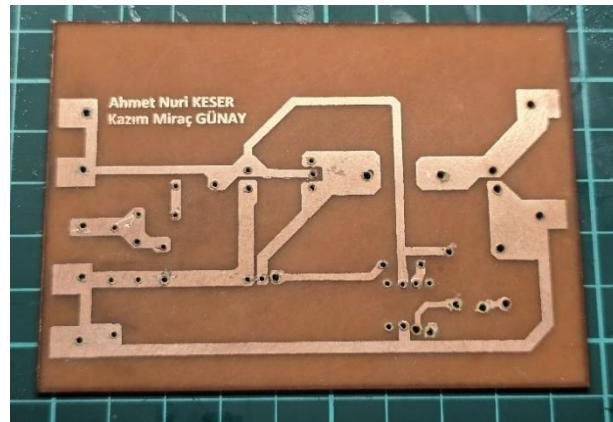


Şekil 27: PCB'nin 3D hali (Ön açıdan bakış)



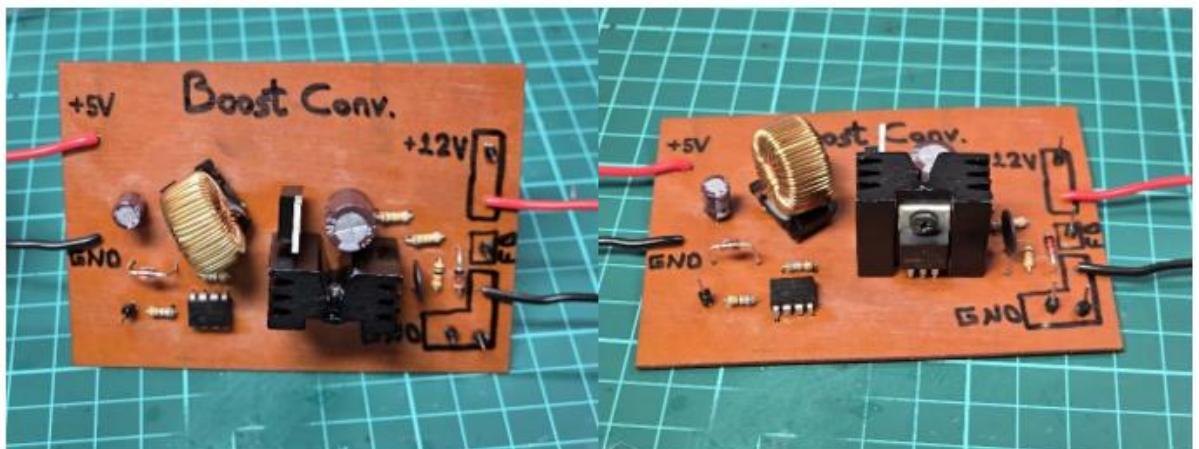
Şekil 28: PCB'nin 3D hali (Arka açıdan bakış)

Altium Designer'da şematik ve PCB tasarımları başarıyla hazırlandıktan sonra PCB'nin üretim işlemlerini gerçekleştirildi.



Şekil 29: Üretilen PCB

PCB üretildikten sonra ilgili komponentlerin dizgi işlemi manuel olarak başarıyla gerçekleştirilmiştir.

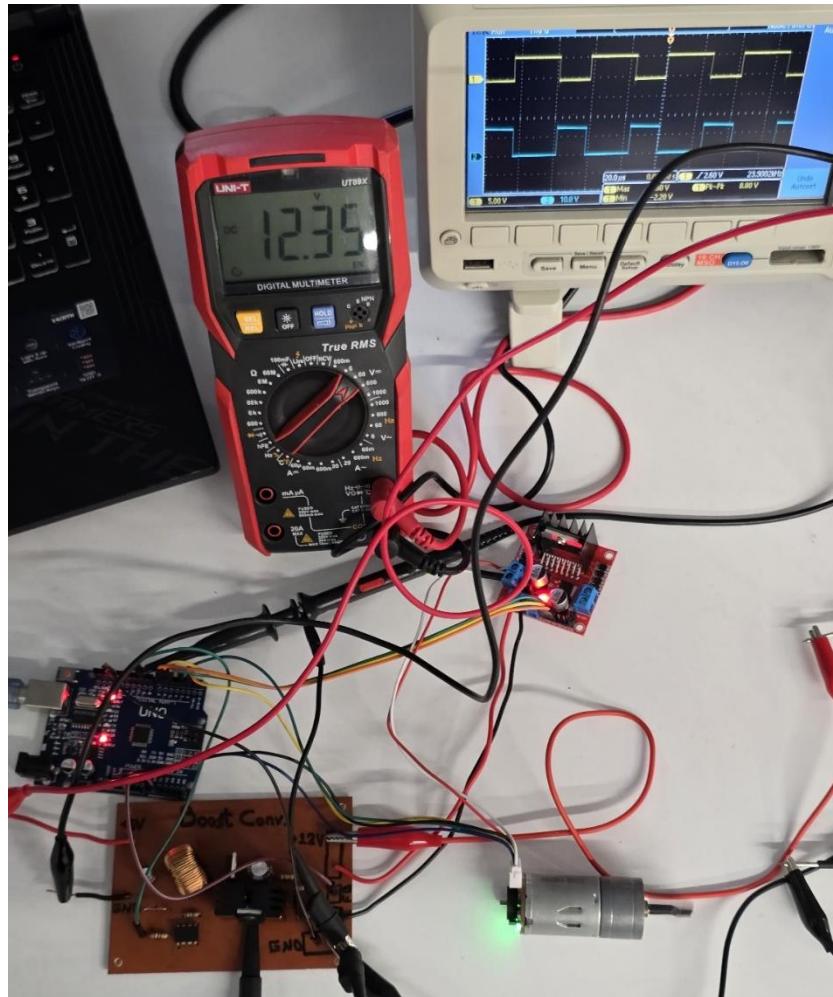


Şekil 30: Dizgisi yapılan PCB (Ön yüz)



Şekil 31: Dizgisi yapılan PCB (Arka yüz)

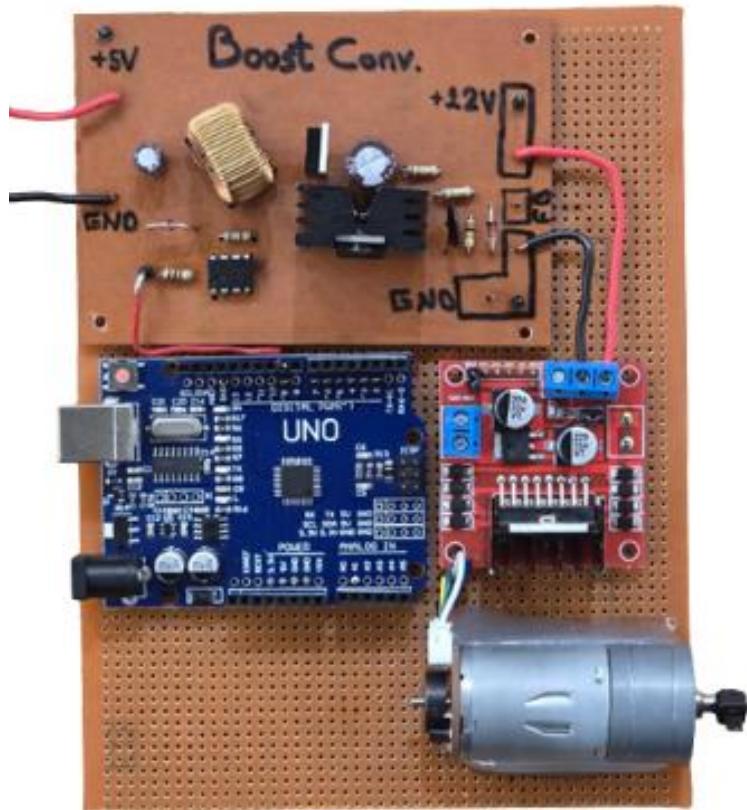
Boost converter'ın baskı devresi alındıktan sonra gerekli kontrol ve testleri yapılmıştır. İlk olarak multimetre ile yapılan ölçümlerde, çıkış geriliminin yaklaşık 12 V seviyesinde olduğu ve osiloskop çıktılarının değerlerinin istenilen kararlıktı oldukları gözlemlenmiştir.



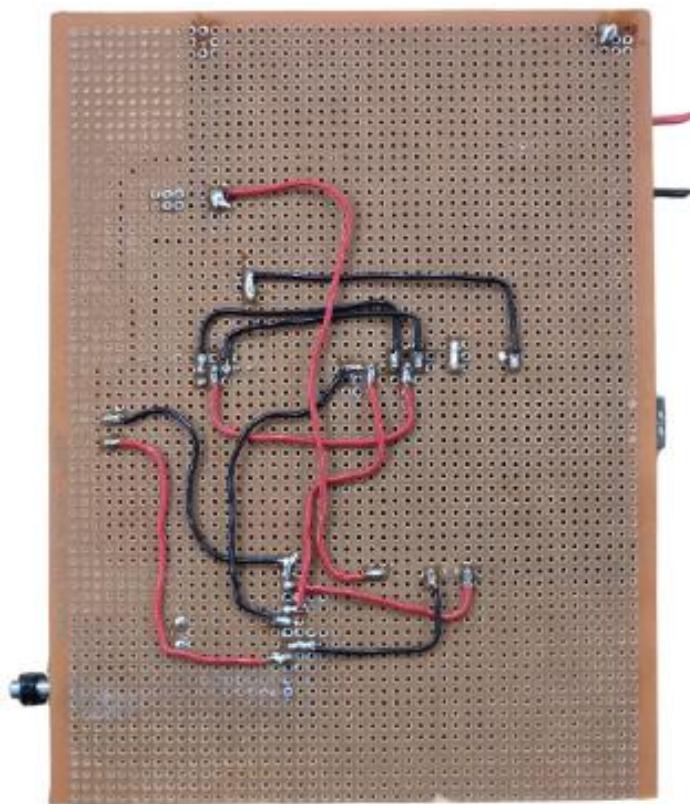
Şekil 32: Boost converter çıkış gerilimi ve osiloskop çıktısı

Baskı devre kartından elde edilen kararlı çıktılar sonrasında, jumper kablolarından kaynaklı oluşabilecek temassızlık sorunlarının önüne geçmek ve projeye bütünlük kazandırmak için tüm devre elemanları tek bir delikli kart üzerinde bir araya getirilmiştir.

Delikli kart üzerinde bir araya getirilen proje tekrardan testlerden geçirilerek kontrolleri sağlanmıştır. Giriş ve çıkışa ait gerilim değerleri istenilen seviyelerde gözlemlenmiştir.



Şekil 33: Delikli kart üzerinde bir araya getirilen proje (ön yüz)

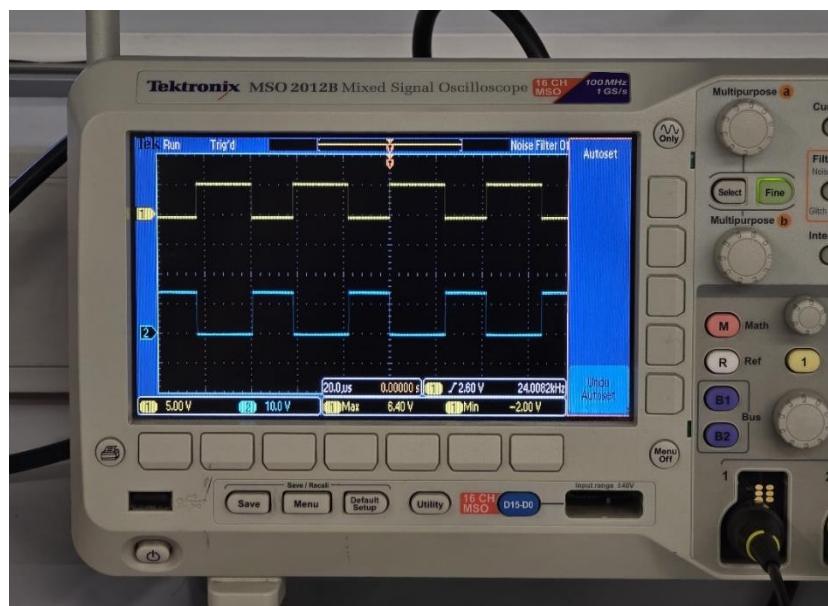


Şekil 34: Delikli kart üzerinde bir araya getirilen proje (arka yüz)



Şekil 35: Giriş ve çıkışa ait gerilim çıktıları

Ardından, osiloskop ile Arduino'nun 9 numaralı pininden sağlanan PWM ve MOSFET'in drain bacağına ait sinyaller incelenerek çıktı değerlerinin istenilen şekilde olduğu görülmüştür. Sinyallerde herhangi bir bozulma ya da gürültü gözlemlenmeyerek sistem kararlılığını teyit edilmiştir. Devrenin çalıştığı süre boyunca devre elemanlarında aşırı ısınma gözlemlenmemiştir.

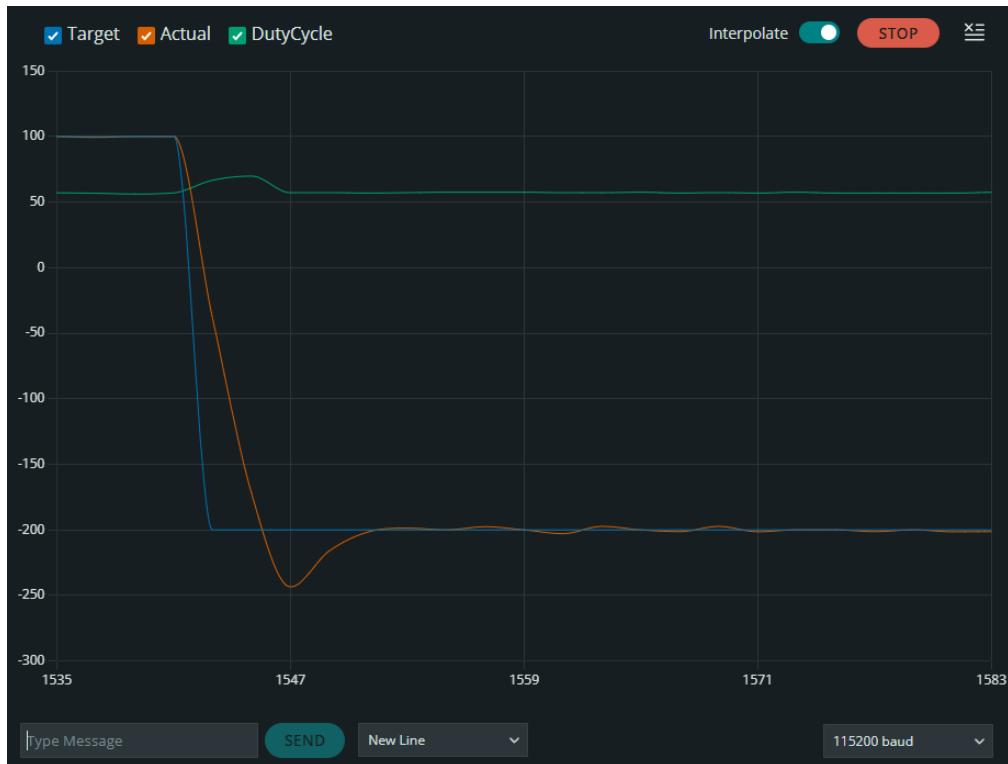


Şekil 36: Osiloskop çıktısı

Ayrıca farklı hedef RPM'ler girilerek devrenin PID ile motor kontrol kısmında verilen komutlara karşılık motorun doğru yönde ve doğru RPM değerlerinde döndüğü gözlemlenmiştir. Motorun enkoder verileri kullanılarak elde edilen RPM, PID kontrol algoritması ile hedeflenen hız değerlerine ulaşılmıştır.



Şekil 37: 100 RPM dönmesi istenen DC motorun hedef ve anlık RPM grafiği



Şekil 38: Ters yönde 200 RPM dönmesi istenen DC motorun hedef ve anlık RPM grafiği

Bu sonuçlar, sistemin hem güç dönüşümü hem de kapalı çevrim hız kontrolü bakımından istenilen değerleri sağladığını göstermektedir.

4. SONUÇ VE ÖNERİLER

Bu çalışmada, 5V giriş gerilimini 12V'a yükselten bir boost converter tasarlanmış ve bu dönüştürücü devresi, PID tabanlı bir hız kontrol sistemiyle entegre edilerek DC motor sürme uygulamasında başarıyla kullanılmıştır. Sistem, Arduino UNO ile yönetilmiş; PWM sinyali üretimi, gerilim izleme ve PID algoritması yazılımsal olarak uygulanmıştır. Enkoder geri bildiriminiyle motorun yön ve hızı gerçek zamanlı olarak takip edilmiş; PID döngüsüyle hedef hızı kararlı şekilde ulaşılmıştır.

Donanım bileşenleri, teorik hesaplamalar ve pratik güvenlik kriterleri dikkate alınarak seçilmiştir. $100\ \mu\text{H}$ endüktör dalgalanmayı azaltmış, $470\ \mu\text{F}$ Kondansatör yük geçişlerinde kararlılığı sağlamış, IRFZ44N MOSFET ve TC4427 sürücü ise hızlı anahtarlama ile verimliliği artırmıştır. Sistem, geri besleme mekanizması sayesinde çıkış gerilimini stabil tutmuş ve duty cycle'ı dinamik biçimde ayarlayarak motora doğru sürüş sağlamıştır.

Sonuçlar, sistemin hem enerji verimliliği hem de kontrol doğruluğu açısından başarılı olduğunu göstermektedir. İlerleyen çalışmalarında baskın devre tasarıımı yapılması, farklı kontrol algoritmalarının uygulanması ve sistemin farklı yük senaryolarında test edilmesi önerilmektedir. Ayrıca, kullanıcı arayüzü entegrasyonu ile sistemin etkileşim düzeyi artırılabilir ve endüstriyel uygulamalara uygun hale getirilebilir.

4.1 Çalışmanın Uygulama Alanları

Boost converter destekli motor kontrol sistemleri, birçok alanda yaygın ve etkin bir şekilde kullanılmaktadır. Özellikle endüstriyel üretim hatlarında yer alan konveyör sistemlerinde, motorların hız ve yön kontrolü PWM ve PID temelli sistemler ile sağlanmaktadır. Bu tür sistemlerde hem saat yönünde hem de saat yönünün tersi yönde hareket gerektiren uygulamalarda kullanılan motorlar, parça taşıma, ayrıştırma ve geri yünlü malzeme döndürme işlemleri için kritik bir rol oynamaktadır. Benzer amaçlarla ev otomasyon sistemleri ve otomatik kapı-bariyer sistemleri, boost converter destekli DC motor kontrolünün yaygın olarak kullanıldığı önemli uygulama alanlarıdır. Akıllı ev çözümleri kapsamında yer alan perde açma-kapama mekanizmaları, pencere havalandırma sistemleri ve otomatik stor sistemleri gibi uygulamalarda, düşük gerilimli bataryalarla çalışan DC motorlar tercih edilmektedir. Bu motorların hızlı ve doğru çalışabilmesi için boost converter devreleri ile gerilim seviyesi artırılarak ihtiyaç duyulan güç

sağlanmaktadır. Ayrıca, motorların yön değiştirebilme özelliği sayesinde perdelerin her iki yöne açılması veya pencere mekanizmalarının hem açma hem kapama işlevlerini yerine getirmesi sağlanmaktadır. Benzer şekilde, alışveriş merkezleri, otoparklar ve fabrika girişlerinde yer alan otomatik kapı ve bariyer sistemleri de çift yönlü motor sürücülere ihtiyaç duyar. Bu sistemlerde motorların hareket yönü, gelen sinyale göre değiştirilebilir; örneğin araç girişinde kapı açılırken saat yönünde dönerken, çıkışta ters yöne dönerek kapatılır. Bu gibi uygulamalarda hız kontrolü, güvenli geçiş için kritik olduğundan, PID destekli PWM motor sürücülerini kullanılarak sistemin tepkisel ve kararlı çalışması sağlanır.

4.2 Çalışmanın Geliştirilmesi için Öneriler

DC motor hız kontrolü başarıyla gerçekleştirılmıştır; ancak sistemin çeşitli yönlerden geliştirilmesi önerilmektedir. İlk olarak, PID kontrol yerine veya tamamlayıcısı olarak Fuzzy Logic (Bulanık Mantık) temelli bir kontrol sisteminin kullanılması tavsiye edilmektedir [31]. Bu yaklaşım, daha karmaşık çalışma koşullarında dahi uyumlu ve esnek performans sağlar. Ayrıca, sistemin güvenlik ve kararlılığının artırılması amacıyla gerilim ve akım sensörlerinin entegrasyonu sağlanabilir. Bu sensörler aracılığıyla anlık gerilim ve akım değerlerinin izlenmesi, aşırı akım veya düşük gerilim durumlarının tespit edilerek koruma mekanizmalarının devreye alınması mümkün olacaktır. Böylece, gerilim ve akım takibi ile arıza tespit süreçlerine de doğrudan katkı sağlanacaktır.

Boost converter devresinin 5 V giriş gerilimini 12 V seviyesine yükselterek motor sürme uygulamaları için yeterli güç sağladığı görülmüştür; ancak daha yüksek enerji dönüşümleri için farklı tasarım yaklaşımlarının uygulanması önerilmektedir. Kullanılan endüktans ve kondansatörlerin enerji depolama kapasitelerinin artırılması; anahtarlama elemanı olarak tercih edilen MOSFET'lerin ise daha yüksek gerilim dayanıklı modellerle güncellenmesi sistem verimliliğini yükseltecektir. Gelişmiş MOSFET sürücü entegreleri aracılığıyla anahtarlama işlemlerinin hız ve verimlilik bakımından iyileştirilmesi mümkün olacaktır. Ek olarak, birden fazla yükseltici devrenin paralel ve senkronize çalışmasını temel alan çok fazlı (multi-phase) boost converter tasarımları sayesinde giriş akımının devreler arasında paylaşılması ve çıkış akımının artırılması sağlanabilir [32].

Sistem, sensör entegrasyonu, fuzzy logic kontrol, çok fazlı yapı ve kablosuz izleme gibi geliştirmelerle daha güçlü ve endüstriyel uygulamalara uygun hâle getirilebilir.

KAYNAKLAR

- [1] "Components101", 2019. [Çevrimiçi]. Available: <https://components101.com/articles/boost-converter-basics-working-design>. [Erişildi: 10 12 2024].
- [2] A. Ramanath, "EEPower", 2022. [Çevrimiçi]. Available: <https://eepower.com/technical-articles/dc-dc-converters-devices-for-converting-to-a-higher-voltage/#>. [Erişildi: 10 12 2024].
- [3] "Wikipedia, The free Encyclopedia", 2024. [Çevrimiçi]. Available: https://en.wikipedia.org/w/index.php?title=Boost_converter&action=history. [Erişildi: 10 12 2024].
- [4] Wikipedia contributors, "Wikipedia, The Free Encyclopedia", 2024. [Çevrimiçi]. Available: https://en.wikipedia.org/w/index.php?title=Boost_converter&action=history. [Erişildi: 10 12 2024].
- [5] J. S. Ali, "Design of PID controller for firstorder and second order systems", 2024.
- [6] J. R. S. Fernando, Y. S. P. T, V. Mohanraj, "Design of Engine Control Unit with Arduino Board for Unmanned Aerial Vehicles to control fuel flow", 2022 8th International Conference on Smart Structures and Systems (ICSSS), 2022.
- [7] "VikipediÖzgür Ansiklopedi", 2024. [Çevrimiçi]. Available: https://tr.wikipedia.org/w/index.php?title=Arduino_Uno&action=history. [Erişildi: 24 11 2024].
- [8] N. Shwetha, L. Niranjan, V. Chidanandan ve N. Sangeetha, "Advance System for Driving Assistance Using Arduino and Proteus Design Tool", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021.
- [9] "Makerspaces.com", 2017. [Çevrimiçi]. Available: <https://web.archive.org/web/20171106020554/https://www.makerspaces.com/arduino-uno-tutorial-beginners/>. [Erişildi: 24 11 2024].
- [10] "Robocombo.com", 2021. [Çevrimiçi]. Available: <https://www.robocombo.com/blog/icerik/arduino-yu-kim-buldu?srsltid=AfmBOoqer13FybUSkRRSgbHnU3G2IT8RZvVGLCNiu8HHTVbk8FQVbiN>. [Erişildi: 15 12 2024].

- [11] B. S. Guru, H. R. Hiziroglu, "Electric Machinery and Transformers", Goodreads, 1988.
- [12] P. C. Sen, "Principles of Electric Machines and Power Electronics", Wiley, 1989.
- [13] A. Hughes, "Electric Motors and Drives: Fundamentals, Types, and Applications", 1990.
- [14] D. J. Howell, The History of the Electric Motor.
- [15] J. Kimbrell, "Fundamentals of Industrial Encoder Sensing Technologies, Motion Detection Theory and", Automation Direct.
- [16] U. Saha, A. Jawad, S. Shahria, A. H.-U. Rashid, "Proximal policy optimization-based reinforcement learning", Heliyon, 2024.
- [17] H. Bodur, "Güç Elektroniği", Birsen Yayıncılık, 2010.
- [18] N. Çoruh, T. Erfidan, S. Ürgün, "DA-DA Boost Dönüştürücü Tasarımı ve Gerçeklenmesi", Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu (ELECO-2008), 2008.
- [19] S. Zhuge, "Crystal Instruments", 2024. [Çevrimiçi]. Available: <https://www.crystalinstruments.com/blog/2020/8/23/pid-control-theory>. [Erişildi: Ocak 2025].
- [20] D. Puangdownreong, A. Nawikavatan ve C. Thammarat, "Optimal Design of I-PD Controller for DC Motor Speed Control", Procedia Computer Science, 2016.
- [21] CONTROL TUTORIALS MATLAB & SIMULINK. [Çevrimiçi].
- [22] W. Ewald, "Wolles Elektronikkiste", 2020. [Çevrimiçi]. Available: <https://wolles-elektronikkiste.de/en/timer-and-pwm-part-2-16-bit-timer1>. [Erişildi: 18 12 2024].
- [23] "Generating an Arduino 16 Bit PWM" 02 01 2020. [Çevrimiçi]. Available: https://www.teachmemicro.com/generate-arduino-16-bit-pwm/#google_vignette. [Erişildi: 18 12 2024].
- [24] S. Karaca, "Arduino Dış (Harici) Kesmeler", Koddefteri, 2018. [Çevrimiçi]. Available: <https://koddefteri.net/arduino/temel-arduino-dersleri/arduino-dis-harici-kesmeler.html>. [Erişildi: 22 12 2024].
- [25] W. Ewald, "Wolles Elektronikkiste", 2020. [Çevrimiçi]. Available: <https://wolles-elektronikkiste.de/en/timer-and-pwm-part-1-8-bit-timer0-2>. [Erişildi: 18 12 2024].
- [26] D. H. B. MACİT, " Arduino ve L298N ile Motor Kontrolü", Hüseyin Bilal Macit PhdComputerengineerasstprof, 2020. [Çevrimiçi]. Available:

<https://www.hbmacit.com/2020/09/13/arduino-l298n-ile-motor-kontrolu/>. [Erişildi: 04 01 2025].

- [27] D. Lin, F. Deng, W. Hua, M. Cheng, Z. Chen, Z. Wang, "High-performance photon-driven DC", *Nature Communications*, 2024.
- [28] R. Bitriá, J. Palacín, "Optimal PID Control of a Brushed DC Motor with an Embedded Low-Cost Magnetic Quadrature Encoder for", Multidisciplinary Digital Publishing Institute, 2022.
- [29] "Last Minute Engineering", 2024. [Çevrimiçi]. Available: <https://lastminuteengineers.com/rotary-encoder-arduino-tutorial/>.
- [30] "Cirkit Designer", [Çevrimiçi]. Available: <https://docs.cirkitdesigner.com/project/published/805e5e61-fdeb-4cc5-80d3-d550dc988d3c/arduino-uno-and-l298n-motor-driver-controlled-dc-motor-with-encoder>. [Erişildi: 17 11 2024].
- [31] "ENCODER", ENCODER PRODUCTS COMPANY, [Çevrimiçi]. Available: https://www.encoder.com/hubfs/Motor_Encoders_article_08012024_FINAL.pdf?hsCtaTracking=18c71b0d-ec96-41d6-a676-ed55ab7e7ee3%7Ceca1270e-d7af-478a-a5e6-e26db2340bb0.
- [32] Abdullah J. H. Al Gizi, "Fuzzy Logic Control Design and Implementation with DC-DC Boost Converter," *EAI Endorsed Transactions on Context-aware Systems and Applications*, vol. 8, 2022. [Çevrimiçi]. Available: Fuzzy_Logic_Control_Design_and_Implementation_with.pdf. [Erişildi: 20 06 2025].
- [33] Fernando Sobrino-Manzanares, Ausias Garrigós, "Interleaved, Multi-Switch, Multi-Phase Boost Converter for Battery Discharge Regulators," *E3S Web of Conferences* (ESPC 2016), EDP Sciences, 2017. [Çevrimiçi]. Available: Interleaved_Multi-Switch_Multi-Phase_Boost_Convert.pdf. [Erişildi: 20 06 2025].

ŞEKİL KAYNAKLARI

- [1] "Boost Converter | Step Up Chopper", Electrtical 4 U, 2024.
<https://www.electrical4u.com/boost-converter-step-up-chopper/>
- [2] C. Torun, "Arduino Kartı Tanıyalım", 2021.
<https://bilimgenc.tubitak.gov.tr/makale/arduino-karti-taniyalim-0>
- [3] <https://www.robotzade.com/urun/enkoderli-25mm-reduktorlu-dc-motor-12v-1000rpm?q=>
- [4] H. Bodur, "Güç Elektroniği", Birsen Yayıncılık, 2010.
- [5] "DC Motor Modelleme PID Kontrolü ve Simülasyonu", Mekatronik ve Bilim, 2014.
<https://mekatronikbilim.wordpress.com/2014/05/17/dc-motor-modelleme-pid-kontrolu-ve-simulasyonu/>
- [6] M. Karthick, "Intergacing Rotary Encoder With Bharat", Autodesk Instructables.
- [7] "L298N Motor Sürücü Devresi", Diyot.net, 2022.
<https://diyot.net/l298n-motor-surucu-devresi/>

EKLER

Ek-1: Arduino Kaynak Kodları

// BİTİRME PROJESİ, Ahmet Nuri KESER & Kazım Miraç GÜNAY

```
const uint16_t FREQUENCY_ICR1 = 662;
const float V THRESH_V = 3.0;
const int FB_PIN = A1;
const int PWM_SUPPLY_PIN = 9;
const int RAW_THRESH = int(V_THRESH_V * 1023.0 / 5.0 + 0.5);
const uint16_t MAX_COUNT = uint16_t(0.7 * (FREQUENCY_ICR1 + 1) + 0.5);
const unsigned long INIT_TIME = 5000;

float kp = 0.04;
float ki = 0.00030;
float kd = 0.005;

const byte Encoder_A = 2;
const byte Encoder_B = 3;
const byte PWMPin = 5;
const byte IN_1 = 6;
const byte IN_2 = 7;

volatile uint16_t dutyCount;
unsigned long startMillis;
volatile long EncoderCount = 0;
unsigned long t, t_prev = 0;
int dt;
float Theta, RPM, RPM_target = 0;
float Theta_prev = 0;
float e, e_prev = 0, inte_prev = 0;
float V, Vmax = 12.0, Vmin = -12.0;
unsigned long count = 0;
bool useSerialPlotter = true;

void ISR_EncoderA() {
    bool PinB = digitalRead(Encoder_B);
    bool PinA = digitalRead(Encoder_A);
    EncoderCount += (PinB == LOW) ? ((PinA == HIGH) ? 1 : -1) : ((PinA == HIGH) ? -1 : 1);
}

void ISR_EncoderB() {
    bool PinA = digitalRead(Encoder_A);
    bool PinB = digitalRead(Encoder_B);
    EncoderCount += (PinA == LOW) ? ((PinB == HIGH) ? -1 : 1) : ((PinB == HIGH) ? 1 : -1);
}

void WriteDriverVoltage(float V, float Vmax) {
```

```

int PWMval = int(255 * abs(V) / Vmax);
PWMval = constrain(PWMval, 0, 255);
if (V > 0) {
    digitalWrite(IN_1, HIGH);
    digitalWrite(IN_2, LOW);
} else if (V < 0) {
    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, HIGH);
} else {
    digitalWrite(IN_1, LOW);
    digitalWrite(IN_2, LOW);
}
analogWrite(PWMPin, PWMval);
}

ISR(TIMER1_COMPA_vect) {
    count++;
}

void setup() {
    Serial.begin(115200);
    while (!Serial);

    pinMode(PWM_SUPPLY_PIN, OUTPUT);
    pinMode(Encoder_A, INPUT_PULLUP);
    pinMode(Encoder_B, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(Encoder_A), ISR_EncoderA, CHANGE);
    attachInterrupt(digitalPinToInterrupt(Encoder_B), ISR_EncoderB, CHANGE);
    pinMode(IN_1, OUTPUT);
    pinMode(IN_2, OUTPUT);
    pinMode(PWMPin, OUTPUT);

    cli();
    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
    TCCR1A |= (1 << WGM11) | (1 << COM1A1);
    TCCR1B |= (1 << WGM13) | (1 << WGM12) | (1 << CS10);
    ICR1 = FREQUENCY_ICR1;
    TIMSK1 |= (1 << OCIE1A);
    dutyCount = uint16_t(0.25 * (FREQUENCY_ICR1 + 1) + 0.5);
    OCR1A = dutyCount;
    sei();

    startMillis = millis();
}

void loop() {
    unsigned long now = millis();

```

```

if (now - startMillis < INIT_TIME) {
    OCR1A = dutyCount;
} else {
    static unsigned long lastMicros = 0;
    unsigned long m = micros();
    if (m - lastMicros >= 1000) {
        lastMicros += 1000;
        int raw = analogRead(FB_PIN);
        if (raw > RAW_THRESH) {
            if (dutyCount > 1) dutyCount--;
        } else {
            if (dutyCount < MAX_COUNT) dutyCount++;
        }
        OCR1A = dutyCount;
    }
}

if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim();
    float newRPM = input.toFloat();
    if (newRPM != 0 || input == "0") {
        RPM target = constrain(newRPM, -400, 400);
        if (!useSerialPlotter) {
            Serial.print("Target RPM set to: ");
            Serial.println(RPM target);
        }
    }
}

static unsigned long lastPidTime = 0;
if (now - lastPidTime >= 10) {
    lastPidTime = now;
    t = now;
    Theta = EncoderCount / 900.0;
    dt = t - t_prev;
    if (dt > 0) {
        RPM = (Theta - Theta_prev) / (dt / 1000.0) * 60.0;
        static float rpmBuffer[5] = {0};
        static int bufferIndex = 0;
        rpmBuffer[bufferIndex] = RPM;
        bufferIndex = (bufferIndex + 1) % 5;
        float rpmSum = 0;
        for (int i = 0; i < 5; i++) rpmSum += rpmBuffer[i];
        RPM = rpmSum / 5.0;

        e = RPM_target - RPM;
        inte = inte_prev + (dt * (e + e_prev) / 2.0);
    }
}

```

```

V = kp * e + ki * inte + kd * (e - e_prev) / dt;
if (V > Vmax) {
    V = Vmax;
    inte = inte_prev;
}
if (V < Vmin) {
    V = Vmin;
    inte = inte_prev;
}

WriteDriverVoltage(V, Vmax);
Theta_prev = Theta;

t_prev = t;
e_prev = e;
inte_prev = inte;
}
}

static unsigned long lastPrint = 0;
if (now - lastPrint >= 100) {
    lastPrint = now;
    float dutyCyclePct = 100.0 * dutyCount / float(FREQUENCY ICR1 + 1);
    if (useSerialPlotter) {

        // === Serial Plotter: Value1 = RPM_target, Value2 = RPM, Value3 = dutyCycle ===
        Serial.print("Target:");
        Serial.print(RPM target);
        Serial.print(" | Actual:");
        Serial.print(RPM);
        Serial.print("\t\t\tDutyCycle:");
        Serial.println(dutyCyclePct);

    } else {
        Serial.print("Target:");
        Serial.print(RPM target);
        Serial.print(" Actual:");
        Serial.print(RPM);
        Serial.print(" DutyCycle:");
        Serial.println(dutyCyclePct);
    }
}

static unsigned long lastVoltageTime = 0;
if (now - lastVoltageTime >= 100) {
    lastVoltageTime = now;
    float voltageADC = analogRead(FB_PIN) * 5.0 / 1023.0;
    float voltageOut = voltageADC * 4.0 * 1.0484;
}

```

```
    Serial.print(" \t \t \t \t Voltage: ");
    Serial.print(voltageOut, 2);
    Serial.println(" V");
}
}
```