

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**SAYISAL KONTROLCÜ KULLANARAK BUCK KONVERTÖR
TASARIMI**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Merve TÜRK TAM
DANIŞMAN: Doç. Dr. M.Nuri ALMALI

VAN-2020

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**SAYISAL KONTROLCÜ KULLANARAK BUCK KONVERTÖR
TASARIMI**

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Merve TÜRK TAM

VAN-2020

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.


Merve TÜRK TAM

ÖZET

SAYISAL KONTROLCÜ KULLANARAK BUCK KONVERTÖR TASARIMI

TÜRKTAM, Merve
Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
Tez Danışmanı: Doç. Dr. M. Nuri ALMALI
Ocak 2020, 69 sayfa

Bu çalışmada, 60W 12V-5V bir buck konvertör devresi sayısal kontrol yöntemi ile tasarlanmıştır. Bu amaçla öncelikle buck konvertörün matematiksel modeli elde edilmiş ve dönüştürücünün sürekli akım modunda çalışabilmesi için kritik endüktans değeri hesaplanmış, ardından dalgalanma faktörü için kritik kapasitans değeri hesaplanmıştır. Çalışma frekansı ve devredeki diğer faktörler göz önünde bulundurularak diyot ve MOSFET seçilmiştir. Sayısal kontrolcü olarak Arduino mikrodeneleyici kartı kullanılmıştır. Sistem çalışırken kaynak gerilimi aniden değiştirilerek sistem tepkisi izlenmiştir. Bu işlem hem PID hem PI kontrol yöntemleri ile denenmiştir. Kaynak gerilimi değişikliklerinde sistem tepkisi ölçüldükten sonra referans değerleri üzerinde değişiklik yapılmıştır. Ani referans değişimleri karşısında sistem tepkisi izlenmiştir. Bu işlemler Arduino mikrodeneleyici karta yazılan kodlar ile gerçekleştirilmiştir. Sonuç olarak sayısal kontrolcü kullanımında PID-PI kontrol yöntemleri karşılaştırılmıştır. Arduino kullanılarak yapılan buck konvertör tasarımında PID kontrolcünün oturma süresi açısından daha başarılı olduğu görülmüştür.

Anahtar kelimeler: Arduino, Buck konvertör, PI, PID



ABSTRACT

BUCK CONVERTOR DESIGN USING DIGITAL CONTROLLER

TÜRKTAM, Merve
M.Sc. Thesis, Electrical-Electronics Engineering
Supervisor : Assoc. Prof. Dr. M. Nuri ALMALI
January 2020, 69 pages

In this study, a buck converter circuit of 60W 12V-5V is designed with digital control method. For this purpose, firstly the mathematical model of buck converter is obtained and critical inductance value is calculated for the converter to work in continuous current mode, then the critical capacitance value is calculated for the fluctuation factor. Diode and MOSFET were selected considering the operating frequency and other factors in the circuit. Arduino microcontroller board is used as digital controller. The system response was monitored by changing the welding voltage suddenly while the system was operating. This process was tested with both PID and PI control methods. Reference values were changed after measuring the system response in welding voltage changes. These operations were performed with the codes written on the Arduino microcontroller board. As a result, PID-PI control methods were compared in the use of digital controllers. In the buck converter design using Arduino, the PID controller was found to be more successful in terms of seating time.

Keywords: Arduino, Buck konvertör, PI, PID



ÖN SÖZ

Bu tez çalışmasında, her türlü ilgi ve yardımlarını esirgemeyen danışmanım Sayın Doç. Dr. M. Nuri ALMALI'ya, bu satırları okuyamayacak olsa da hissettiğinden emin olduğum ve daima kalbimde olan canım babam Metin GÜLEŞ'e, fedakâr annem Mine GÜLEŞ'e, hayatın her anında yol göstericim abim Mert GÜLEŞ ve kız kardeşim Şevval GÜLEŞ'e, ilgisini, desteğini anlayışını bir an eksik etmeyen sevgili eşim Ufuk TÜRKTAM'a ve son olarak gelişi ile hayatımı anlamlandıran canım oğlum Metin Aras TÜRKTAM'a teşekkürlerimi sunarım.

2020

Merve TÜRKTAM



İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT	iii
ÖN SÖZ.....	v
İÇİNDEKİLER.....	vii
ÇİZELGELER LİSTESİ	ix
ŞEKİLLER LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xiii
EKLER DİZİNİ.....	xv
1. GİRİŞ	1
2. KAYNAK BİLDİRİŞLERİ.....	3
3. MATERYAL ve YÖNTEM.....	5
3.1. Materyal.....	5
3.1.1. Buck konvertör.....	5
3.1.1.1. Buck konvertörün transfer fonksiyonunun elde edilmesi.....	14
3.1.1.2. Devre elemanlarının hesaplanması.....	16
3.1.2. Arduino	21
3.2. Yöntem.....	27
3.2.1. PID kontrol.....	28
3.2.2. PI kontrol	31
3.2.3. MATLAB ortamında buck konvertör tasarımı	31
3.2.4. Yazılım tasarımı.....	33
3.2.4.1. Mikrodenetleyici programlama.....	33
4. BULGULAR VE TARTIŞMA	37
4.1. Giriş Geriliminin Değiştirilmesi.....	37
4.2. Referans Değerinin Değiştirilmesi	40
5. SONUÇ	45
KAYNAKLAR.....	47
EKLER	51
ÖZ GEÇMİŞ	69



ÇİZELGELER LİSTESİ

Çizelge	Sayfa
Çizelge 3. 1. B20100G Schottky bariyer diyotun genel ve mekanik özellikleri	18
Çizelge 3. 2. IRFZ44N MOSFET karaktetik özellikleri.....	19
Çizelge 3. 3. Buck konvertör parametreleri.....	19
Çizelge 3. 4. Arduino Mega 2560 kartın temel özellikleri (Anonim, 2015).....	23



ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 3. 1. Temel buck konvertör şeması	6
Şekil 3. 2. Anahtarın iletimde olduğu durum	7
Şekil 3. 3. Anahtarın kesimde olduğu durum	8
Şekil 3. 4. İdeal buck konvertör dönüştürücü dalga şekilleri (Rashid, 2014)	11
Şekil 3. 5. 0-150 Ω Sürgülü reosta	17
Şekil 3. 6. Uygulama devresi.....	21
Şekil 3. 7. Arduino Mega 2560 mikrodenetleyici kartı.....	23
Şekil 3. 8. Arduino IDE platformunun açılış görüntüsü.....	25
Şekil 3. 9. Arduino kartı seçimi.....	26
Şekil 3. 10. COM port seçimi.....	27
Şekil 3. 11. PID kontrol mekanizması.....	30
Şekil 3. 12. Klasik dijital PID denetleyici şeması (Özdemir,2018).....	30
Şekil 3. 13. PI kontrol mekanizması.....	31
Şekil 3. 14. Buck konvertör MATLAB modeli	32
Şekil 3. 15. Simülasyon çıkış gerilimi.....	33
Şekil 3. 16. Arduino IDE kod editörü ekran görüntüsü.....	34
Şekil 4.1. PID kontrol yöntemi ile değiştirilmiş giriş değerine sistemin tepkisi	37
Şekil 4.2. Aniden değiştirilen giriş gerilimine karşı sistemin tepkisi (PID kontrol).....	38
Şekil 4.3. PI kontrol yöntemi ile değiştirilmiş giriş değerine sistemin tepkisi.....	39
Şekil 4.4. Aniden değiştirilen giriş gerilimine karşı sistemin tepkisi (PI kontrol).....	39
Şekil 4.5. PID ve PI kontrol karşılaştırması	40
Şekil 4.6. Referans değeri 7V olarak değiştirildiğinde osiloskop sistem tepkisi	41

Şekil 4.7. Referans değeri 7V olarak değiştirildiğinde sistem tepkisi (MATLAB) 41

Şekil 4.8. Referans değeri 3V olarak değiştirildiğinde osiloskop sistem tepkisi 42

Şekil 4.9. Referans değeri 3V olarak değiştirildiğinde sistem tepkisi (MATLAB) 42



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
A	Amper
C	Kondansatör
C_K	Kritik kondansatör değeri
d	Görev Çarpanı
D	Diyot
F	Farad
f_s	Anahtarlama frekansı
gr	Gram
i_c	Kondansatör akımı
i_g	Giriş gerilimi
i_d	Diyot üzerinden geçen akım
i_L	Ortalama bobin akımı
i_o	Çıkış akımı
I_{omin}	Yük akımının minimum akım değeri
K_d	Türevsel katsayı
K_i	İntegral katsayı
K_p	Oransal katsayı
L	Bobin
L_K	Kritik bobin değeri
mΩ	Miliohm
nF	Nanofarad
R_L	Bobin eşdeğer direnci
S	Anahtar
ton	İletim süresi
toff	Kesim süresi
T	Periyot

Simgeler	Açıklama
μF	Mikrofarad
μH	Mikrohenry
V	Volt
V_c	Kondansatör gerilimi
V_d	Diyot gerilimi
V_i	Giriş gerilimi
V_L	Bobin gerilimi
V_o	Çıkış gerilimi
W	Watt
ΔI_L	Bobin akımı değişimi
ΔI_C	Kondansatör akım değişimi

Kısaltmalar	Açıklama
AA	Alternatif Akım
BJT	Bipolar Junction Transistor (Çift kutuplu eklemlı transistör)
CCM	Continuous Current Mode (Sürekli akım modu)
DA	Doğru Akım
DCM	Discontinuous Current Mode (Süreksiz akım modu)
FET	Field-Effect Transistor (Alan Etkili Transistör)
IDE	Integrated Development System (Bütünleşik Geliştirme Ortamı)
MOSFET	Metal Oxide Semiconductor FET (Metal oksit yarıiletkenli FET)
PI	Proportional-Integral (Oransal-İntegral Kontrol)
PID	Proportional-Integral-Derivative (Oransal-İntegral Türevsel)
PWM	Pulse Width Modulation (Darbe genlik modülasyonu)
RAM	Random Access Memory (Rasgele Erişimli Bellek)
USB	Universal Serial Bus (Evrensel Seri Veriyolu)

EKLER DİZİNİ

Ek	Sayfa
Ek 1. PID kontrol yöntemi ile referans değerini değiştiren Arduino kodları.....	51
Ek 2. PID kontrol yöntemi ile elde edilen verilerin grafik oluşturma kodları.....	60
Ek 3. PI kontrol yöntemi referans değerini aniden değiştiren Arduino kodları.....	62
Ek 4. PID ve PI kontrol karşılaştırması yapan verilerin grafik oluşturma kodları.....	62
Ek 5. Referans değerini 7V yapan Arduino kodları.....	64
Ek 6. Referans değerini 3V yapan Arduino kodları.....	68





1. GİRİŞ

Teknolojinin hızla gelişmesinin neticesinde doğru akım-dođru akım (DA-DA) konvertörler önemli bir hal almaya başlamıştır. DA-DA konvertörler sadece akademik camiada güç elektroniđi uygulamaları kapsamında deđil aynı zamanda endüstri, sanayi ve askeri havacılık sektörlerinde de yükseliştir. DA-DA konvertörlerin bu denli önem kazanmasında yüksek verim, küçük hacim, düzgün gerilim regölasyonu, dinamik cevabın iyi olması ve esnek bir kontrol yapısına sahip olmaları gibi özellikleri etkilidir (Erfidan ve ark., 2008). Dinamik cevabın iyi olmasından kasıt sistemin ani deđişimlere hızlı bir şekilde adapte olabilmesidir. Ayrıca Sistemlerin amaçladığı yüksek hassasiyete sahip besleme voltajının kısa sürede ve yüksek hassasiyetle gerçekleştirilmesi noktasında DA-DA konvertörler etkin rol oynamaktadırlar. Bu tarz konvertörler yapıları geređi anahtarlama elemanının tetiklenmesi için bir sinyale ihtiyaç duyarlar. Bu sinyal analog ya da dijital kontrol yöntemleri ile üretilebilir.

Teorik olarak tasarlanan kontrol teknikleri analog ve dijital kontrol teknikleri olarak ikiye ayrılır. Her iki kontrol tekniđi için aynı isimlerle anılan temel kontrol yöntemi tasarımı yapılabilir. Örnek verecek olursak, Oransal-Türevsel-İntegral (Proportional-Integral-Derivative-PID) kontrol her iki teknikle de tasarlanabilir. Analog kontrol yöntemleri pasif devre elemanlarına ek olarak transistörler ve işlemsel yükselteçler ile gerçekleştirilir. Dijital kontrol yöntemlerinde ise mikroişlemciler ile kodlar ön plandadır. Ayrıca dijital kontrol yöntemlerinin daha çok tercih edilmesinde tasarımlarının daha kolay olması bilgilerin bir yere konması gerektiğinde bu bilgilerin çağırılması ve istenildiđi kadar tutulabilmesi, gürültüden az etkilenmesi ve gerektiğinde üzerinde deđişiklik yapılmasının daha kolay olması gibi avantajları da etkili olmuştur.

Yapılan tez çalışmasında, öncelikle 60W 12V-5V bir DA-DA Düşürücü Tip (buck) konvertör tasarımı gerçekleştirilmiştir. Devredeki elektronik anahtarın tetiklenmesi için gerekli sinyal hem PID kontrol yöntemi, hem de PI (Proportional-Integral) kontrol yöntemi kullanılarak üretilmiştir. Fakat burada tasarlanan PID ve PI kontrol yöntemleri analog olarak deđil temelinde mikroişlemci mantığı bulunan

Arduino adı verilen fiziksel bir programlama platformunda tasarlanmıştır. Kodlar üzerinde yapılan değişikliklerde (örneğin aniden değişen giriş ve referans değerlerine karşılık sistemin tepkisi) sistemin nasıl etkilendiği ne tür geri dönüşler sağladığı analiz edilmiştir. Analog kontrolcüler kullanarak tasarlanan devrelerde sistemdeki değişikliklerde devre elemanlarının yeniden hesaplanması ya da sistemin komple yeniden tasarlanması gerekirken Arduino ile yapılan bu kontrolde kodlar üzerinde değişiklik yapmak yeterli olmuştur. Bu da bize diğer avantajlarının yanı sıra zaman kazancı sağlamıştır.

Bu konu ile ilgili yapılan çalışmalar incelenerek kaynak bildirişi kısmında sunulmuştur. Buck konvertör tasarımı için gerekli olan devre denklemleri çıkarılmış ve devre elemanlarının değerleri hesaplanmıştır. Hesaplanan bu değerlere bağlı olarak 12V-5V bir buck konvertör tasarımı MATLAB simülasyon programında gerçekleştirilmiştir. Simülasyon sonucunda elde edilen veriler MATLAB ortamında grafiklendirilmiştir. Seçilen devre elemanlarının özellikleri, çalışma prensipleri ve nasıl elde edildikleri de çalışma içerisinde olarak anlatılmıştır. Çalışmadaki PID ve PI kontrol yöntemlerinde seçilen parametrelerde yine açıklanmış olup iki kontrol yöntemi arasındaki farklarda değerlendirilmiştir. Çıkış gerilimini sabit tutabilmek ve elektronik anahtarın tetiklenmesi için gerekli darbe genişlik modülasyonu (Pulse Width Modulation-PWM) sinyali her iki kontrol yöntemi ile üretilmiştir. Hesaplanan devre elemanlarına uygun olarak uygulama devresi gerçekleştirilmiştir. Uygulama devresinde PID kontrolcüsünün yazılımı Arduino adı verilen mikrodenetleyici kartı kullanılarak gerçekleştirilmiştir. Seçilen Arduino kartının temel özellikleri, kullanımı ve programlanması hakkında bilgilere de çalışma içerisinde ayrıca yer verilmiştir. Son olarak çalışma sonucunda elde edilen bulgular verilmiş olup sonuç bölümünde bu bulguların değerlendirilmesi yapılmıştır.

2. KAYNAK BİLDİRİŞLERİ

Bu bölümde tez çalışması için yararlanılan kitap, makale ve tez çalışmaları gibi kaynaklara yer verilmiştir.

Ürgün ve ark. (2008), mikrodenetleyici kontrollü buck dönüştürücü devresi yapmışlardır. Çalışma sonucunda daha düşük maliyet ve yüksek hızlarda çalışabilmesi gibi özelliklerinden dolayı mikrodenetleyici kullanımının buck konvertörlerde kullanılması gerektiğini önermişlerdir.

Yurdakul ve ark. (2009), Yaptıkları çalışmada kendinden ayarlamalı bir PID kontrol tasarımı yapmışlardır. Elde ettikleri sonuçlara göre kendinden ayarlamalı kontrolde taşmaların ciddi bir şekilde azaldığını, sürücü sistem hızının yükselme süresi ve yatışma sürelerinin azalmasıyla geliştirildiği ve Ziegler-Nichols ve Cohen-Coon gibi geleneksel yöntemlerden daha iyi sonuç verdiğini belirtmişlerdir.

Gidemen ve ark. (2015), PID parametrelerinin ayarlama yöntemleri üzerinde bir çalışma yapmışlardır. Çalışma içerisinde açık çevrim Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswick, Wong-Juang-Chan ve $1/4$ bozunma oranlı yöntemler incelenmiştir. Bu çalışma sonucunda endüstriyel uygulamalarda PID denetleyicinin yaygın olarak kullanılmasına rağmen uygulamalardaki bozucu, belirsizlik veya modellenemeyen diğer etkenlerden dolayı bu çalışmada incelenen yöntemlerin yeterli performans veremeyeceği bu nedenle daha iyi performans için parametreler üzerinde elle ayarlama yapmanın daha iyi olabileceğini belirtmişlerdir.

Yalduz (2015), yaptığı tez çalışmasında düşürücü tip dönüştürücünün sayısal kontrollü tasarımını gerçekleştirmiştir. Yaptığı çalışma sonucunda dönüştürücünün sayısal kontrolünün analog kontrole göre boyut, maliyet ve değişen ortam şartlarına karşı esneklik gibi avantajları olduğunu belirtmiştir.

Zaman (2016), çalışmasında sabit ve bozulmamış bir DA çıkış gerilimi elde edebilmek için geniş bir değişken giriş voltajı aralığında çalışabilen bir voltaj düşürücü-yükseltici DA-DA çevirici tasarlamıştır. Bu çalışması ile dönüştürücü tipleri arasında verime dayalı bir karşılaştırma yapmıştır.

Mondal (2014), bir buck dönüştürücü için sayısal PID kontrol çalışması yapmıştır. Çalışma sonucunda bilgisayar kontrollü DA-DA dönüştürücünün daha verimli çalışması, ortam değişikliklerine karşı daha dayanıklı olması ve kolayca arayüz programlayabilme özelliklerinden dolayı daha sık kullanılması gerektiği belirtilmiştir.

Trip ve ark. (2010), yaptıkları çalışmada DA-DA Buck konvertör için sayısal denetleyicili bir uygulama yapmışlardır. Önerilen sistemin avantajı farklı algoritmaları donanım değişikliği olmadan uygulanabilmesidir. Makalede sunulan tasarım yönteminin eğitim süreçlerinde laboratuvar derslerinde kullanılabileceğini belirtmişlerdir.

Rao ve ark. (2014), yaptıkları çalışmada PID kontroldeki üç değişkenin belirli oranlardaki toplamından elde edilen kontrol sinyali sistemin sonucunu daha başarılı yaptığını belirtmişlerdir.

Müller ve ark. (2015), dönüştürücü tasarımında Arduino kullanımının Arduino'nun yeteneklerinin fazlalığı sayesinde arttığını, daha sofistike kontrol uygulamaları için bir basamak taşı olduğunu belirtmişlerdir.

Evrar (2019), azaltan DA-DA dönüştürücüsü için kayma kip denetleyici tasarımı yapmıştır. PSIM programında yapılan benzetim sonuçları neticesinde ani değişen yük ve giriş geriliminde referans değere hızlı bir şekilde ulaşıldığı ve ani değişikliklerin az olmasını sağladığı görülmüştür.

Al-Juboori (2017), yaptığı tez çalışmasında Arduino ile kontrol edilen AA (alternatif akım) voltaj regülatörü tasarlamıştır. Bu çalışmada Arduino kullanımının verimli ve hızlı bir kontrol sağladığını belirtmiştir.

Köse ve ark. (2016), yaptıkları çalışmada buck konvertörün çıkış gerilimini ayırık zamanlı PI kontrol algoritması kullanarak optimize etmişlerdir. PI kontrolcünün parametreleri Harmony Arama Algoritması (HSA) ile ayarlanmıştır. Kontrolcü yanıtları Matlab/Simulink programında incelenmiş ve başarısı araştırılmıştır.

Özdemir (2018), yaptığı tez çalışmasında buck konvertör için ayırık zamanda üç farklı kontrolör tasarımı yapmış olup bunların performansını değerlendirmiştir. Çalışma sonuçlarına göre, kayan kip kontrolörlü buck konvertörün tüm aşamalarda daha iyi performans gösterdiğini belirtmiştir.

3. MATERYAL ve YÖNTEM

3.1. Materyal

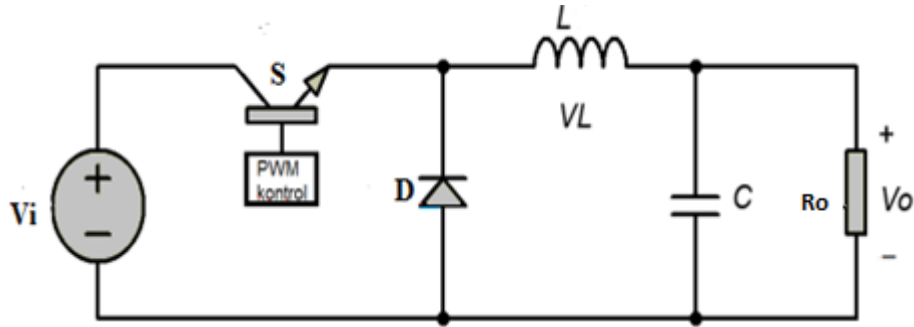
Bu bölümde çalışmada tasarlanan buck konvertörün devre şeması, çalışma prensibi, çalışma modları, durum denklemleri, konvertörün sürekli akım modunda çalışabilmesi için gerekli endüktans ve kondansatör değerlerinin hesabı, durum uzay modeli, transfer fonksiyonu, devre elemanlarının hesabı, küçük işaret transfer fonksiyonu ile birlikte mikrodenetleyici olarak kullanılan Arduino kartı tanıtılmış olup kartın nasıl programlandığına dair bilgilere yer verilmiştir.

3.1.1. Buck konvertör

Güç elektroniği kısaca gerilim kaynağından gelen enerjinin çeşitli devre elemanları (endüktans, kondansatör, direnç) yardımıyla dönüştürülerek sistem çıkışında istenen sinyale dönüştürülmesi olarak tanımlanabilir. Güç elektroniği kavramı, güç, elektronik ve kontrol kavramlarını bünyesinde barındırır. Burada güç kavramı enerji üretimi, iletimi, dağıtımı için kullanılan güç cihazlarını, elektronik kavramı yarı iletken devre elemanları ve kontrol amacına ulaşabilmek için kullanılan devrelerini içerir. Kontrol kavramının amacı ise kapalı çevrim sistemlerde durum ve dinamik özellikleri ile ilgilenmektir. Aynı zamanda güç elektroniği denilince akıllara elektrik enerjisinin verimli, temiz ve güvenilir bir formdan başka bir forma dönüştürme işlemi gelir (Rashid, 2004).

Günümüzde kullanılan güç kaynaklarında yüksek verim, küçük hacim, düşük fiyat ve sabit çıkış gerilimi gibi özellikler istenmektedir. Bu özellikler doğrusal elektronik elemanlarla sağlanamamaktadır. Bu nedenle anahtarlamalı güç kaynaklarına ihtiyaç duyulmaktadır. Anahtarlamalı güç kaynaklarının sayısal kontrolü ile yapılan çalışmalar yalnızca akademisyenler tarafından değil ayrıca güç elektroniği endüstrisinde de yoğun olarak kullanılmaktadır (Erfidan ve ark., 2008).

Güç elektroniğinde sıklıkla kullanılan buck konvertörler DA giriş geriliminden yine DA formda daha küçük ve kararlı bir çıkış gerilimi vermektedir. Buck konvertörler güç elektroniğinin farklı uygulama alanlarında geniş yer bulmaktadır. Çoğunlukla regüle edilmiş güç kaynakları ve DA motor hız kontrol devrelerinde kullanılmaktadır (Mohan ve ark., 2010). Şekil 3.1. de temel bir buck konvertör şeması gösterilmiştir.



Şekil 3. 1.Temel buck konvertör şeması.

Bu devrede V_i giriş gerilim kaynağına seri bir anahtar (S) bağlanmıştır. Girişteki kaynak gerilimi anahtar ile bobin ve kondansatörden oluşan LC alçak geçiren filtre üzerinden çıkıştaki yükü besler (Kamil, 2017).

Buck konvertörün temelinde S anahtarının açık ve kapalı olduğu durumlara göre ortalama bir çıkış gerilimi elde etme amacı vardır. Bu ortalama gerilim doluluk oranına (d) bağlı olarak değişir. Doluluk oranı anahtarın açılıp kapanması sırasında girişteki gerilimin yüke aktarılma oranıdır. Buck konvertörde anahtarlama elemanının kesimde olduğu sürede endüktans akımının sıfır değerine ulaşması durumunda süreksiz akım modu (Discontinuous Current Mode-DCM), sıfır değerine ulaşmaması durumunda sürekli akım modu (Continuous Current Mode-CCM) oluşur (Mamur,2012).

Devredeki S anahtarı uygulanan PWM işaretine bağlı olarak iletim veya kesim modunda çalıştırılır. Burada anahtar iletim süresi $t_{on} = dT$ ve kesimde olduğu süre $t_{off} = (1-d)T$, anahtarlama periyodu ise iletimde ve kesimde olduğu sürelerin toplamına eşittir ($T = t_{off} + t_{on}$) (Yalduz, 2015). Doluluk oranının olması gereken aralık “Eş. 3.1” de verilmiştir.

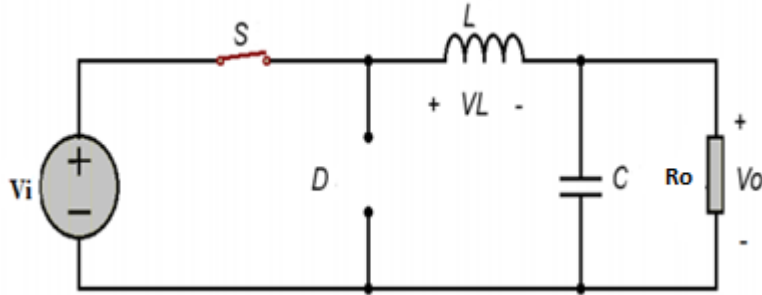
$$0 < d < 1 \quad (3.1)$$

Devredeki tüm elemanlar ideal kabul edilerek anahtarın Şekil 3.2'deki gibi kapalı yani iletimde olduğu durum için devreye Kirchoff'un gerilim yasası uygulanırsa "Eş. 3.2" elde edilir.

$$V_i + L \frac{di_L}{dt} + V_o = 0 \quad (3.2)$$

Bu durumda V_i kaynak gerilimi t_{on} süresi boyunca çıkışı besler. Bu durumda bobin üzerindeki gerilim giriş gerilimi ile çıkış gerilimi V_o arasındaki farka eşit olur. Kondansatör üzerinde oluşan gerilim ise çıkış gerilimine eşit olur. Bobin üzerindeki gerilim "Eş. 3.3" te verilmiştir.

$$V_L = V_i - V_o \quad (3.3)$$



Şekil 3.2. Anahtarın iletimde olduğu durum.

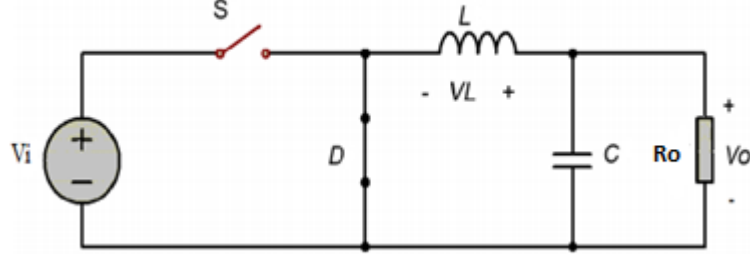
Bobinde oluşan gerilim bobin akımını lineer olarak artırır. Bobin geriliminin akıma bağlı değişimi "Eş. 3.4" te verilmiştir.

$$V_L = L \frac{di_L}{dt} \quad (3.4)$$

İletim durumu için bobin akımındaki yükselme “Eş. 3.5” te verilmiştir.

$$\Delta I_L(+)=\frac{V_i-(V_o-I_L R_L)-V_o}{L}t_{on} \quad (3.5)$$

Bu formülde R_L bobin eşdeğer devresindeki direnci, V_o anahtar iletimde iken yarı iletken üzerindeki gerilim düşümünü ve $\Delta I_L(+)$ bobin akımındaki dalgalanmayı gösterir (Ürgün ve ark., 2008).



Şekil 3.3. Anahtarın kesimde olduğu durum.

S anahtarının açık olduğu yani kesime gittiği t_{off} süresinde bobin üzerinde biriken enerji yük ve kondansatörden geçerek diyot üzerinden devreyi tamamlar (Şekil 3.3). Aynı şekilde Şekil 3.3’deki devreye de Kirchhoff’un gerilim yasası uygulanırsa “Eş. 3.6” ve “Eş. 3.7” elde edilir.

$$L\frac{di_L}{dt}+V_o=0 \quad (3.6)$$

$$\frac{dV_o}{dt}=\frac{i_L}{C}-\frac{V_o}{R_o C} \quad (3.7)$$

Bu durumda bobin gerilimi çıkış gerilimine eşit olur. Bobin gerilimi “Eş. 3.8” de verilmiştir.

$$V_L = -L \frac{di_L}{dt} \quad (3.8)$$

Şekil 3.2 ve Şekil 3.3'e Kirchoff'un akım yasası uygulanırsa "Eş. 3.9" elde edilir.

$$i_C = i_L - i_0 \quad (3.9)$$

Buradaki i_C kondansatör akımı, i_L bobin akımı, i_0 yük akımını ifade eder.

Kesim durumu için bobin akımındaki azalma "Eş. 3.10" da verilmiştir.

$$\Delta I_L(-) = \frac{V_i - (V_o - I_L R_L) - V_o}{L} t_{on} \quad (3.10)$$

Buradaki $\Delta I_L(-)$ anahtar kesimde iken bobin akımındaki dalgalanmayı gösterir. Kararlı durum çalışma şartlarında $\Delta I_L(+)$ ve $\Delta I_L(-)$ eşit kabul edilir. Bundan dolayı denklem (8) ve denklem (10) eşitlenerek sürekli durum için buck konvertörün çıkış gerilimi V_o bulunabilir (Ürgün ve ark., 2008).

Bobin akımı x_1 ve çıkış gerilimi x_2 ve anahtarın açık kapalı olma durumları olmak üzere, buck konvertörün durum uzay modeli "Eş. 3.11", Eş. "3.12" ve Eş. "3.13" ile verilmiştir (Özdemir, 2018).

$$x_1 = I_L \quad \text{ve} \quad x_2 = V_o \quad (3.11)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{V_{in}}{L} \\ 0 \end{bmatrix} u \quad (3.12)$$

$$x_2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.13)$$

Buck konvertörün küçük işaret transfer fonksiyonları “Eş. 3.14”, “Eş. 3.15”, “Eş. 3.16” ve “Eş. 3.17” ile verilmiştir (Hekimoğlu ve ark., 2018).

$$G_{v_i}(s) = \frac{\hat{V}_o}{\hat{V}_i} = \frac{D/LC}{s^2 + S/R_oC + 1/LC} \quad (3.14)$$

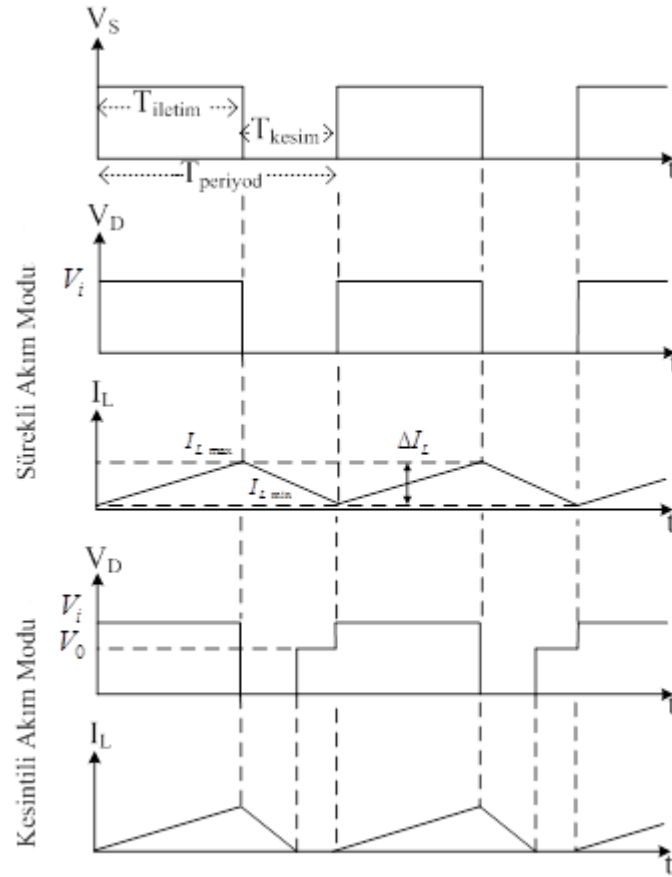
$$G_{v_d}(s) = \frac{\hat{V}_o}{\hat{d}} = \frac{V_i/LC}{s^2 + s/R_oC + 1/LC} \quad (3.15)$$

$$G_{i_g}(s) = \frac{\hat{i}_L}{\hat{V}_i} = \frac{D/L(S + 1/R_oC)}{s^2 + S/R_oC + 1/LC} \quad (3.16)$$

$$G_{i_d}(s) = \frac{\hat{i}_L}{\hat{d}} = \frac{V_i/L(S + 1/R_oC)}{s^2 + s/R_oC + 1/LC} \quad (3.17)$$

“Eş. 3.14” ve “Eş. 3.15” sırasıyla kondansatör geriliminin çıkış olarak seçildiği durum için girişten çıkışa küçük işaret cevabı ve kontrolden çıkışa küçük işaret cevabıdır. “Eş. 3.16” ve “Eş. 3.17” ise sırasıyla öz endüktör akımının çıkış olarak seçildiği durum için girişten çıkışa küçük işaret cevabı ve kontrolden çıkışa küçük işaret cevabıdır. Eşitliklerde kullanılan i_g giriş gerilimi, i_d diyot akımı, V_d diyot gerilimini ifade eder (Hekimoğlu ve ark., 2018).

Buck konvertörün CCM ve DCM durumlarında akım gerilim grafiği Şekil 3.4'te verilmiştir (Özdemir, 2018).



Şekil 3.4. İdeal buck konvertör dönüştürücü dalga şekilleri (Özdemir, 2018).

Bobin akımının CCM ve DCM için olması gereken sınır değerleri sırasıyla “Eş. 3.18” ve “Eş. 3.19” da verilmiştir (Rashid, 2014; Arifoğlu, 2018).

$$\Delta I_L < I_{o \min} \quad (3.18)$$

$$\Delta I_L > I_{o \min} \quad (3.19)$$

ΔI_L ; bobin akımının maksimum ve minimum değerlerini $I_{o \min}$ ise yük akımının minimum değerini ifade etmektedir. Anahtarlama frekansı f_s olmak üzere, bobin

akımının deęişim miktarı, giriş ve çıkış gerilimi cinsinden ifadesi “Eş. 3.20” de verilmiştir.

$$\Delta I_L = \frac{V_i - V_o}{Lf_s} d \quad (3.20)$$

“Eş. 3.21” de bobin akımının deęişiminin sadece çıkış gerilimi cinsinden ifadesi verilmiştir.

$$\Delta I_L = \frac{V_o}{Lf_s} (1-d) \quad (3.21)$$

“Eş. 3.21” nin giriş gerilimi cinsinden ifadesi aşağıda (“Eş. 3.22”) verilmiştir.

$$\Delta I_L = \frac{V_i d (1-d)}{Lf_s} \quad (3.22)$$

Endüktans akımının sürekli ya da süreksiz olması devredeki endüktans (L) deęerine baęlı olduğundan kritik endüktans deęerinin (L_K) doęru hesaplanması önemlidir. L_K deęeri anahtarlama frekansına, doluluk oranına ve yük deęerine baęlıdır. L_K deęeri “Eş. 3.23” te verilmiştir.

$$L_K = \frac{R_0 (1-d)}{2f_s} \quad (3.23)$$

Bu eşitlikte R_0 yük deęeri, f_s anahtarlama frekansını ifade eder. Ayrıca sürekli akım modunda $L > L_K$ olmalıdır.

İdeal bir buck konvertör devresinde yük akımı üzerinde dalgalanma olmayacağından kondansatör akımındaki dalgalanma endüktans akımı üzerinde oluşacak dalgalanmaya eşit olur (“Eş. 3.24”) (Rashid, 2014).

$$\Delta I_C = \Delta I_L \quad (3.24)$$

Kapasite üzerinden $(T_{\text{off}}/2 + T_{\text{on}}/2) = T/2$ ’lik zaman boyunca pozitif yönde akan akımın ortalama değeri” Eş. 3.25” te verilmiştir.

$$I_C = \frac{\Delta I_L}{4} \quad (3.25)$$

Kapasite gerilimi V_C aşağıdaki şekilde hesaplanır (“Eş. 3.26”).

$$V_C = \frac{1}{C} \int I_C dt + V_C(t=0) \quad (3.26)$$

“Eş. 3.27” de kapasite geriliminin tepeden tepeye dalgalanma miktarı verilmiştir.

$$\Delta V_C = \frac{1}{C} \int_0^{T/2} \frac{\Delta I_L}{4} dt + V_C(t=0) = \frac{\Delta I_L}{8f_s C} \quad (3.27)$$

veya;

$$\Delta V_C = \frac{V_i d(1-d)}{8LCf_s^2} \quad (3.28)$$

Kondansatör gerilimini sürekli yapacak kritik kondansatör değeri (C_K) “Eş. 3.29” da ifade edildiği gibi anahtarlama frekansına, PWM işaretinin doluluk oranına ve endüktans değerine bağlıdır (Mamur, 2012).

$$C_K = \frac{(1-d)}{16Lf_s^2} \quad (3.29)$$

Ortalama gerilim V_o darbeleme oranına bağlı olarak doluluk oranı “Eş. 3.30” ve “Eş. 3.31” ile elde edilir.

$$V_o = \frac{1}{T_s} \int_0^T V_o dt = dV_i \quad (3.30)$$

$$d = \frac{V_o}{V_i} \quad (3.31)$$

Burada d değeri değiştirilerek V_o gerilimi kontrol edilebilir. V_o kontrol gerilimine bağlı olarak doğrusal bir değişim gösterir. Ancak pratik olarak yük tamamen omik değil endüktif olabilir. Bu durumda anahtarlama elemanı endüktif enerjiyi tüketip bozulabilir. Ayrıca çıkış geriliminin 0 ile V_d arasında fazla salınım yapması birçok uygulama için istenmeyen bir sonuçtur. Devredeki diyot sayesinde depolanan enerji problemi giderilir. Çıkış gerilimindeki bozukluklar alçak geçiren filtre yardımıyla yok edilebilir.

3.1.1.1. Buck konvertörün transfer fonksiyonunun elde edilmesi

Transfer fonksiyonu kısaca impuls cevabının laplace dönüşümü olarak tanımlanır. Yani zamanla değişmeyen doğrusal sistemlerde sistemin girişi ile çıkışı arasındaki laplace dönüşüm oranına transfer fonksiyonu denir (Özkan, 2007).

Direnç, bobin ve kondansatör için gerilim ve akım değerleri sırasıyla “Eş. 3.32”, “Eş. 3.33”, “Eş. 3.34” te verilmiştir (Özdemir, 2018).

$$v(t) = Ri(t) \text{ ve } i(t) = \frac{v(t)}{R} \quad (3.32)$$

$$v(t) = \frac{Ldi(t)}{dt} + v(t=0) \text{ ve } i(t) = \frac{1}{L} \int_0^t v(t)dt + i(t=0) \quad (3.33)$$

$$v(t) = \frac{1}{C} \int_0^t i(t)dt + v(t=0) \text{ ve } i(t) = \frac{Cdv(t)}{dt} + i(t=0) \quad (3.34)$$

Kirchhoff yasalarına göre (Bir kapalı çevrim sistemde gerilimler toplamı ve bir noktaya gelen ve giden akımların toplamı sıfırdır.) devrenin diferansiyel denklemi yazılır (“Eş. 3.35”, “Eş. 3.36”). Bu ifadenin laplace dönüşümü yapılarak (“Eş. 3.37”, “Eş. 3.38”) sistemin transfer fonksiyonu elde edilir (“Eş. 3.39”) (Rashid, 2014; Özdemir, 2018).

$$V_i(t) = \frac{Ld_i(t)}{dt} + V_o(t) \quad (3.35)$$

$$\frac{V_i(t) - V_o(t)}{\frac{Ldi(t)}{dt}} = \frac{CdV_o(t)}{dt} + \frac{V_o(t)}{R_o} \quad (3.36)$$

$$V_i(s) = Ls + V_o(s) \quad (3.37)$$

$$\frac{V_i(s) - V_o(s)}{Ls} = CsV_o(s) + \frac{V_o(s)}{R_o} \quad (3.38)$$

$$\frac{V_o(s)}{V_i(s)} = \frac{1}{LCs^2 + \frac{L}{R_o}s + 1} \quad (3.39)$$

3.1.1.2. Devre elemanlarının hesaplanması

Güç elektroniği uygulamalarında kullanılan devre elemanlarının seçimi oldukça önemlidir. Bu seçim yapılırken sistemin amacı göz önünde bulundurulmalıdır. Devrede kullanılan devre elemanları direnç, kondansatör, bobin, diyot ve mosfettir.

Elektrik akımına karşı ortaya çıkan zorluk için elektriksel direnç tanımı yapılır. Elektronik uygulamalarda en sık kullanılan devre elemanı dirençlerdir. Birimi ohm, simgesi Ω 'dur. Dirençlerin çok farklı çeşitleri bulunmaktadır. Ölçümleri renkli çizimlerle anlaşılır. Farklı amaçlar için doğru direnç elemanı seçilmesi önemlidir. Dirençler sabit dirençler ve ayarlı dirençler olmak üzere ikiye ayrılır. Sabit değerli dirençlerde gerilim belirli bir değerde sabittir. Bu yüzden direnç değerinde herhangi bir değişim olmaz. Değişken değerli dirençler ise farklı aralıklarda direnç değerinin ayarlanması mantığına dayanır. Bu yüzden kullanım alanları çok daha kapsamlıdır. Direncin bağlı olduğu noktadaki gerilim ayarlanabildiği gibi geçen akımın da ayarlanabilmesi oldukça avantaj sağlamaktadır. Ayarlı dirençlerin ilk örneği ve en popüler olanları reostalardır. Üzerlerinde mevcut olan kol yardımı ile direnç değeri artırılıp azaltılabilir. Birden fazla reostayı seri ya da paralel bağlama yolu ile de istenilen direnç değeri elde edilebilir.



Şekil 3.5. 0-150 Ω Sürgülü reosta.

Bobinler elektrik enerjisini manyetik akıya çevirerek depolama yapabilen devre elemanlarıdır. En önemli özellikleri üzerlerinden geçen akımın bobin üzerinde bir gerilim üretmesidir. Meydana gelen bu olaya indükleme, bobinin bu özelliğine ise endüktans adı verilir. Kısaltması L, birimi Henry'dir.

Kondansatörler elektrik enerjisini elektrik alan olarak depolayan iki uçlu devre elemanlarıdır. Kısaltması C, birimi Farad (F)'dir. Kondansatörler güç kaynağı devrelerinde filtrelemede, rezonans devrelerinde istenilen frekansı üretmede ve güç aktarım hatlarında gerilim düzenlemesi ve güç akışını kontrol etmek amacıyla kullanılır. Kapasitöre gerilim uygulandığında iletken plakalar birbirine göre ters ve eşit değerlerdeki elektrik yükü ile yüklenir. Aslında bu durum elektrik alan oluşmasına sebep olur. İki plaka arasındaki yalıtkan madde yük akışına engel olur. Uygulamalarda kondansatör seçerken dikkat edilmesi gereken bazı hususlar vardır. Çalışma gerilimi polarizasyon, tolerans, sıcaklık katsayısı ve kaçak akım bunlardan bazılarıdır.

Diyotlar elektrik akımının sadece bir yönde geçişine imkân veren yarı iletken devre elemanlarıdır. Anot ve katot olmak üzere iki bacağı vardır. Akım, diyot üzerinden anot bacağından katot bacağına doğru iletilir. Diyotlar p-tipi ve n-tipi olmak üzere iki farklı yarı iletken malzemenin birleştirilmesi ile elde edilir. Diyotun anot ucuna bir

gerilim uygulanıp katot ucuna negatif gerilim uygulanması işlemine ileri kutuplama denir. Bu sayede p tipi yarı iletken malzemede bulunan pozitif yükler n-tipi yarı iletken malzemenin negatif yüklerine doğru hareket eder ve bu sayede elektrik akımının geçişi başlamış olur. Bu işlemin tam tersi olarak ters kutuplama yapıldığında herhangi bir yük hareketi olmaz ve diyot üzerinden akım geçmez. Bir diyotun davranışı yani karakteristiğini akım-gerilim grafiğine bakarak anlamak mümkündür. Köprü, led, zener, schottky, lazer ve fotodiyot olmak üzere birçok çeşitleri vardır. Bu çalışmada düşük voltaj uygulamalarında sık sık kullanılan B20100G schottky bariyer diyot kullanılmıştır. Kullanılan bu diyotun genel ve mekanik özellikleri Çizelge 3.1'de belirtilmiştir.

Çizelge 3. 1. B20100G Schottky bariyer diyotun genel ve mekanik özellikleri

Genel Özellikler	Mekanik Özellikler
Yüksek kararlı pasifleştirilmiş bağlantı	Epoksi kalıplıdır.
Düşük ileri gerilim düşümü	Genelde ağırlığı 1.9 gr'dır.
Ters yönlü enerji geçişine yüksek dayanım	Tüm dış yüzeyler korozyona karşı dayanıklıdır.
Ayrıca elektrik izolasyonuna gerek yoktur.	Lehimleme için kurşun sıcaklığı 260 ° C (maksimum)

Çalışmada kullanılan en önemli devre elemanlarından biri MOSFET'tir. (Metal Oxide Field Effect Transistor-Metal oksit yarıiletkenli FET). MOSFET'ler transistör adı verilen devre elemanının geliştirilmiş halidir. Transistörler bir gerilim ya da akım kaynağı ile başka bir gerilim ya da akım kaynağını kontrol etmek amacıyla kullanılan devre elemanlarıdır. En çok kullanılan çeşitleri BJT (Bipolar Junction Transistor - Çift kutuplu eklemli transistör) ve Field-Effect Transistor (FET-Alan etkili transistör)'lerdir. Bu iki transistör arasındaki fark, BJT'lerin akımın oluşturduğu elektrik alanla, FET'lerin gerilimin oluşturduğu elektrik alan ile çalışmasıdır. Transistörler temelde üç bacağı sahiptir. FET'lerde bu bacaklar gate, drain, source olarak adlandırılır.

MOSFET'lerin gate bacağı izoleli olduğundan giriş empedansı son derece yüksek, elektrotlar arası iç kapasitansı çok düşüktür. Bu özelliklerinden dolayı MOSFET'ler daha çok tercih edilir. Ayrıca MOSFET'lerde gürültü daha az ve bant genişliği daha fazladır. Bu çalışmada karakteristik özelliklerinin uygunluğundan dolayı IRFZ44 kodlu N tipi bir MOSFET kullanılmıştır. IRFZ44N MOSFET'in bazı karakteristik özellikleri Çizelge 3.2'de gösterilmektedir.

Çizelge 3. 2. IRFZ44N MOSFET karakteristik özellikleri

Sembol	Parametre	Maksimum Değer	Birim
VDS	Drain-Source Voltajı	55	V
ID	Drain Akımı	49	A
Ptot	Toplam güç dağılımı	110	W
TJ	Bağlantı Sıcaklığı	175	° C
RDS	D-S Durum Direnci	22	mΩ

Çizelge 3.3'te tasarlanan buck konvertörün yük değeri, ilk durumdaki giriş gerilimi ve daha sonra kodlar ile değiştirilen giriş gerilimi, ilk durumdaki çıkış gerilimi ve daha sonra yine kodlarla değiştirilen çıkış gerilimleri ve anahtarlama frekansı değerleri verilmiştir.

Çizelge 3. 3. Buck konvertör parametreleri

Parametre	Açıklama	Değer
R_o	Yük Değeri	50 Ω
V_i	Giriş Gerilimi	12-10V
V_o	Çıkış Gerilimi	5-3-7V
f_s	Anahtarlama Frekansı	50 kHz

Çizelge 3.3'teki değerlere göre doluluk oranı "Eş. 3.40" da verilmiştir.

$$d = \frac{V_o}{V_i} = \frac{5}{12} = 0,42 \quad (3.40)$$

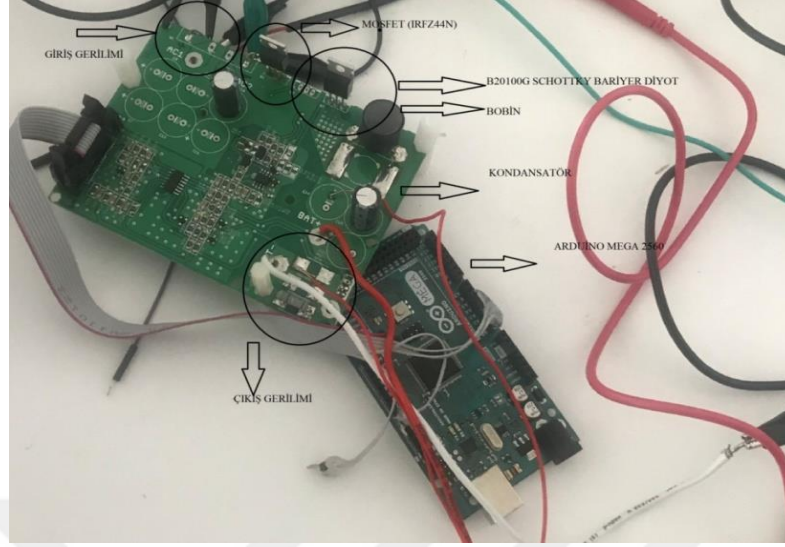
Endüktansın alabileceği minimum değer "Eş. 3.23" ten yararlanılarak "Eş. 3.41" deki şekilde hesaplanır.

$$L_K = \frac{R_o(1-d)}{2f_s} = \frac{50(1-0,42)}{2(50.10^3)} = 290\mu H \quad (3.41)$$

Kondansatörün minimum değeri "Eş. 3.29" dan yararlanılarak "Eş. 3.42" deki gibi hesaplanır.

$$C_K = \frac{1-d}{16Lf_s} = \frac{1-0,42}{16.0,33.10^{-3} \cdot (50.10^3)^2} = 43,9nF \quad (3.42)$$

Çalışmada yük akımının sürekli olmasını garanti etmek amacıyla 330μH değerinde bir bobin kullanılmıştır. Ayrıca yük üzerindeki gerilim dalgalanmasının düşük olması için 1000μF değerinde bir kondansatör kullanılmıştır. Anahtarlama frekansı 50 kHz ve devrenin gücü 60W olarak seçilmiştir.



Şekil 3.6. Uygulama devresi.

Sisteme giriş gerilimi DA gerilim kaynağından verilmiştir. Böylelikle giriş gerilimi üzerinde değişiklik ve hassas ayar yapmak kolaylaşmıştır. Sistem çıkışını görebilmek amacıyla çıkış uçlarına osiloskop bağlanmıştır. Dijital çıkış alabilmek amacıyla Arduino'nun A0, A1 ve A2 numaralı pinleri kullanılmıştır. Arduino'da kodlar yardımı ile üretilen PWM sinyali ise Arduino'nun 9 numaralı pininden verilmiştir.

3.1.2. Arduino

Günümüzde dijital elektronik işlemesi yapmak için mikroişlemciler kullanılmaktadır. Mikroişlemci sektörüne ilk adımı Intel firması yapmıştır. Daha sonraki yıllarda teknolojinin gelişmesi ile birlikte tümleşik devre teknolojisi mikrodenetleyici olarak adlandırılmaya başlandı (Türkmen, 2017).

Mikrodenetleyici, mikroişlemci, bellek ve bazı diğer birimlerin tek bir entegre içinde toplanmış halidir. Bu özelliği tasarımlarda boyuttan tasarruf edilmesi sebebi ile önemlidir. Mikrodenetleyiciler özellikle denetim teknolojisi gerektiren uygulamalarda sık sık kullanılırlar. Mikroişlemcilere göre boyutunun küçük olmasının dışında mikroişlemcilere göre daha basit kullanımlı ve daha ucuzdur. Mikrodenetleyiciler ile

tasarım yapmadan önce bazı özelliklerinin bilinmesi gereklidir. Bunlardan bazıları programlanabilir dijital ve analog giriş/çıkış bacakları, seri girişlerde motor veya servo kontrol için tetik sinyali çıkışı, dahili bellek tipi seçenekleri, dahili RAM (Random Access Memory- Rasgele Erişimli Bellek) seçeneği gibi özelliklerdir (Megep, 2012).

Son yıllarda özellikle veri görselleştirme alanında programlar yazmayı oldukça kolaylaştıran ATMEL firması tarafından üretilen Arduino mikrodeneleyiciler oldukça yaygın kullanım alanına sahiptir. Genellikle elektronik mühendisleri tarafından kullanılan bu mikrodeneleyici sayesinde çok az teknik bilgiye sahip olan insanlara da programlamanın kapıları açılmış oldu. Arduino mikrodeneleyiciler, günümüzde çok sık kullanılan bir mikrodeneleyici çeşitidir. Arduino'yu bu kadar popüler yapan bazı özellikleri vardır. Bunlar geliştirme ortamının ve sürücülerinin kurulumunun kolay olması, bütün platformlarda çalışabilir olması (Windows, Linux, Mac), geniş kütüphanesi, hızlı çalışabilmesi, birlikte çalışmaya uyumlu birçok ek donanım desteği, fiyatının uygun olması, açık kaynaklı olduğundan isteyen herkesin kullanımına açık olması gibi özelliklerdir. Arduino'nun Mega, Uno, Nano, Mini, Mini Pro, Due gibi birçok çeşidi vardır. Tasarlanacak sisteme göre bunlardan herhangi biri tercih edilebilir (Taşdemir, 2015). Arduino devre kiti dijital çıkışları TTL şeklinde olan ve yapılan çalışmaya bağlı olarak PWM olarak da kullanılabilen çıkışlara sahiptir. 10 bitlik analog girişleri olan bu devre kiti USB üzerinden haberleşmeye de uygundur (Gupta ve ark., 2018).

Bu çalışmada, buck konvertörün çalışabilmesi için gereken tetikleme sinyalini üretmek amacı ile Arduino Mega 2560 mikrodeneleyici kartı kullanılmıştır (Şekil 3.7).



Şekil 3.7. Arduino Mega 2560 mikrodenetleyici kartı (Türkmen, 2017).

Arduino Mega 2560 mikrodenetleyici kartın temel özellikleri Çizelge 3.4'te verilmiştir.

Çizelge 3. 4. Arduino Mega 2560 kartın temel özellikleri (Anonim, 2015)

Özellik	Değer
Mikrokontrolcü	ATmega 2560
Çalışma gerilimi	5V
Önerilen giriş gerilimi aralığı	7 – 12V
Giriş / Çıkış pinleri	54 (15'i PWM)
Analog giriş pinleri	16
Giriş / Çıkış pinleri akım	40 mA
Flash bellek	256 KB
SRAM	8 KB
EEPROM	4 KB
Saat frekansı	16 MHz

Arduino Mega 2560 mikrodenetleyici kartı ATmega 2560 mikrokontrolcüyü kullanmaktadır. Atmel firması tarafından geliştirilen bu mikrokontrolör, AVR ailesine üye RISC mimarisi ile geliştirmiş 8 bitlik, 16 MHz frekansı ile çalışan bir karttır. ATmega 2560 mikrokontrolcü temeli üzerine tasarlanan Arduino Mega 2560 mikrodenetleyici kartı,

üzerinde bulunan bellek ve giriş çıkış pinleri gibi donanımları yardımıyla, çevresel elamanlardan veri alabilme aldıkları verileri değerlendirebilme ve çıkışa sayısal sinyaller verebilme özelliğine sahiptir (Türkmen, 2017).

Arduino Mega 2560, USB'den ve harici bir adaptör ile beslenebilir. Kart 6-20V harici kaynaktan beslenebilir. Ancak burada dikkat edilmesi gereken 7V aşağısı beslemelerde çıkış pini vermesi gereken 5V çıkışı veremez ve aynı şekilde 12V üstü gerilimlerde voltaj regülatörü fazla ısınarak karta zarar verir.

Kart üzerindeki 54 adet dijital giriş çıkış pini `pinmode()`, `digitalWrite()`, `digitalRead()` fonksiyonları ile giriş ve çıkış olarak atanabilir. 2, 3, 18, 19, 20 ve 21 numaralı pinler harici kesmeler için kullanılır ve `attachInterrupt()` fonksiyonu ile hangi kesmenin hangi şekilde kullanılacağı belirtilir. 0-13 arasındaki pinler 8-bit PWM çıkış verir. Bu çıkış için `analogWrite()` fonksiyonu kullanılır.

Arduino Mega 2560 mikrodenetleyici kartını programlayabilmek için gerekli arayüz internet ortamından kolayca ve ücretsiz bir biçimde indirilebilir. Bu arayüzün adı Arduino IDE (Integrated Development System- Bütünleşik Geliştirme Ortamı)'dir. 256 KB kapasiteye sahip flash belleğin 8 KB'lık kısmı, kartı programlamak için gerekli olan kod parçasına (bootloader) ayrılmıştır.

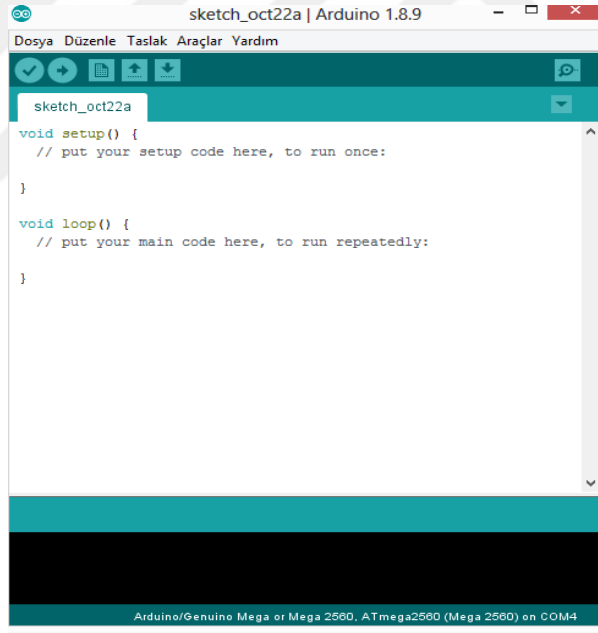
Arduino platformu iki bileşenden oluşur. Bunlardan ilki üzerinde program yazdığımız, derleme ve program yükleme işlemlerini yapmamıza olanak sağlayan IDE, diğeri ise programın içinde çalıştığı mikrodenetleyici, çevresel bileşenleri ve giriş çıkış portlarını barındıran Arduino kartıdır. Arduino ile bilgiler www.arduino.cc adresinden kolaylıkla indirilebilir.

Arduino mikrodenetleyici kartları üzerinde, önceden yazılmış bir bootloader programı vardır. Bootloader diğeri bir ismiyle önyükleyici Arduino içerisine derlenen programların yüklenmesini sağlayan küçük bir programcıdır. Bu özelliğin sağladığı avantaj kartı programlamak için dışarıdan bir programcıya ihtiyaç duyulmamasıdır.

Arduino Mega 2560 mikrodenetleyici kartına bakıldığında üzerinde USB port girişi bulunduğu görülmektedir. Bu port sayesinde IDE kod editöründe yazılan satırlar kartın flash belleğine aktarılır. Bu sayede programlama da yapılmış olur. Ayrıca yine bu USB

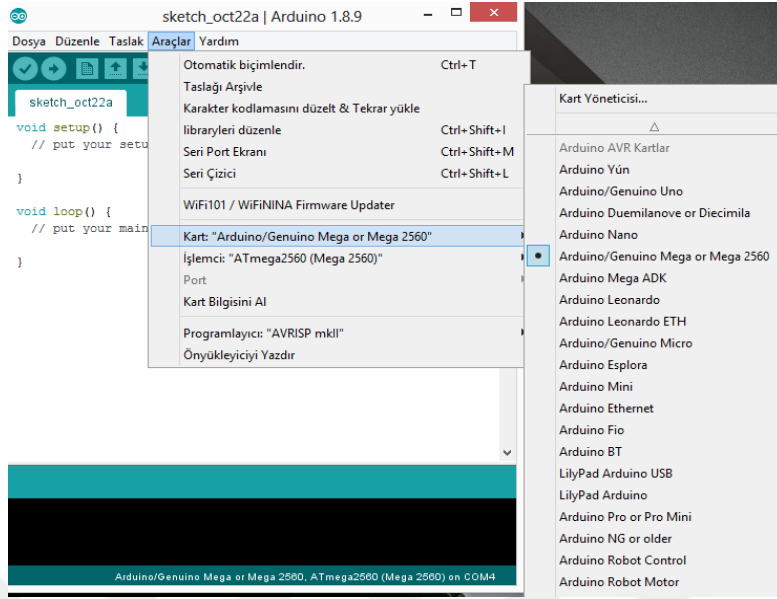
portu sayesinde Arduino IDE yardımıyla programlanmış mikrodenetleyici kartları başka bir bilgisayar ya da mikrodenetleyici ile seri olarak haberleşme yapılabilir.

Arduino IDE bilgisayara kurulduktan sonra çalıştırıldığında “.ino” uzantılı, “sketch” isimli dosyalar açar. Bu dosyaların içerisinde varsayılan olarak belirtilen iki adet fonksiyon bulunmaktadır. Yazılmak istenen kod veya komutlar, editör üzerinde “setup()” ve “loop()” olarak isimlendirilmiş fonksiyonların içerisindeki bloklara yazılmaktadır (Şekil 3.8). Setup fonksiyonu Arduino’yu çalıştırdıktan sonra bir kez çağrılır ve başlangıç ayarlarını yapar. Loop fonksiyonu ise devamlı çalışan fonksiyondur. Arduino yeniden başlatılmadıkça çalışmaya devam eder. Program açıldığında karşımıza ilk çıkan void fonksiyonu değer döndürmeyen fonksiyonlar olduğunu belirtmektedir. Süslü parantezler içinde kalan bölümlerde fonksiyon gövdeleri bulunur.



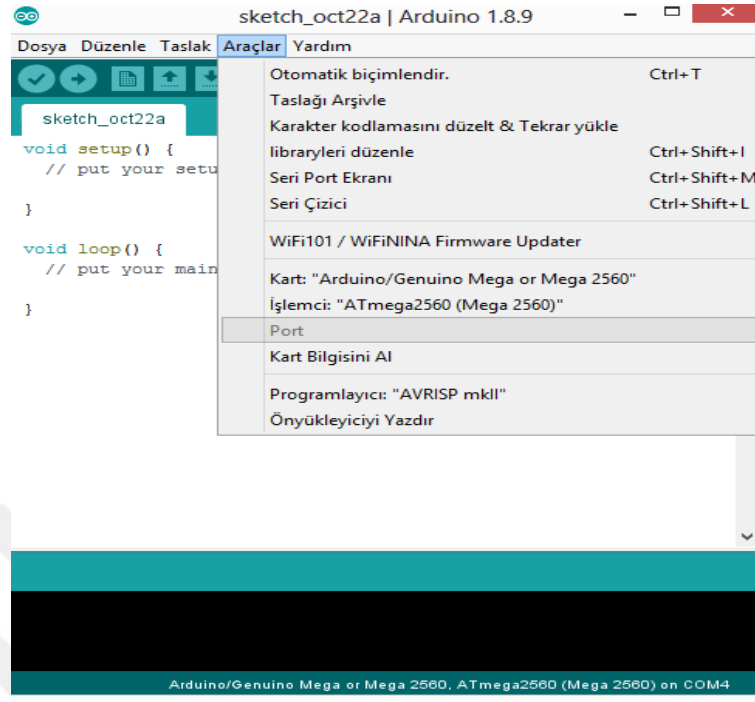
Şekil 3.8. Arduino IDE platformunun açılış görüntüsü.

Bu ekran açıldıktan sonra bazı ayarlamalar yapılması gerekir. Öncelikle Araçlar>Kart menüsünden kullanılan Arduino kartının çeşidi seçilir (Şekil 3.9).



Şekil 3.9. Arduino kartı seçimi.

Daha sonra Arduino ile iletişim kurmayı sağlayan USB üzerinden çalışan bir COM port tanımlanır (Şekil 3.10).



Şekil 3.10. COM port seçimi.

Ayrıca Arduino IDE içerisinde bazı kütüphaneler bulunmaktadır. Bu kütüphaneler, çeşitli programlar için kullanılmak istenen birçok özelliği içerisinde bulunudur ve çalışma sayfasının ilk satırına gerekli komutlar eklenerek program içerisine aktarılır.

3.2. Yöntem

Çalışmada ilk olarak kontrol yöntemi seçimi yapılmıştır. Bu çalışmada kontrol yöntemi olarak PID ve PI kontrol yöntemleri seçilmiştir. Bu bölüm içinde iki kontrol yöntemine ait formüller ve parametre seçimlerine ilişkin bilgi verilmiştir. Daha sonra uygulamadan önce tasarlanan simülasyon programı hakkında bilgi verilmiş olup simülasyona ait görüntülere yine bu bölüm içinde yer verilmiştir.

Kontrol, herhangi bir sistemin çıktılarının önceden referans olarak belirlenmiş kriterlere göre belirli bir zaman aralığında davranışını göstermesi anlamına gelir. Kontrol sistemlerinin amaçları, bir takım süreçlerin (denetlenen sistemlerin) çıktısını, istenen sabit

bir deęerde kontrol altına almak (ısıtma, soęutma sistemleri gibi), sreęlerin ıktısının belirli bir deęişim formunu takip etmesini saęlamak malzeme (ısıl işlemler sistemleri gibi), olayların belirli bir sıra ile oluşmasını saęlamak (amaşır makinesi, trafik ışıkları gibi), kontrol edemediğimiz deęişkenlerin bozucu etkisini düzeltmek, rutin, tekrara dayalı işlerin insan yerine makine ile yapılması, insan tarafından yapılamayacak hızda, hassaslıkta ve hatasız yapılması, sürekli deęişen durumlara karşın, en doęru/uygun aksiyonun yerine getirilmesi, emniyetinin saęlanması, minimum harcama ile maksimum faydanın/retim saęlanması olarak ele alınabilir (Arifoęlu, 2005).

PID kontrolrn grevi, lme zerinden geri beslenen ıkış byklęn referans olarak verilen giriş byklę ile karşılaştırarak aradaki farkın yani hata deęerinin miktarına kendi kontrol sisteminin etkisine uygun bir şekilde kumanda denetim sinyali reterek ıkışa tekrar sunmaktır (Ogata, 2009).

nceleri manuel olarak yapılan kontroller gnmzde yerini otomatik kontrol sistemlerine bırakmıřtır. Kontrolclerin grevi sistem davranışını biimlendirmektir. Yani bařka bir deyiřle kontrolc bir sistemin işlevini bir ama doęrultusunda deęiřtirir. Bu amala girdi olarak sistemin o anki durumunun bir gstergesine ihtiya duyar. Uygulamalarda bu girdi sistem hatasıdır. Sistem hatası bařlangıtaki giriş verisinin referans olarak belirlenen deęerden ne kadar uzak olduęunu gsterir (Gren, 2016).

Bu tez alıřmasında kontrolc olarak PID ve PI kontrol yntemleri denenmiřtir. İki kontrol yntemi iin sistem kararlılıęı incelenmiř ve bu sayede oturma sresi bakımından karşılaştırma yapılmıřtır.

3.2.1. PID kontrol

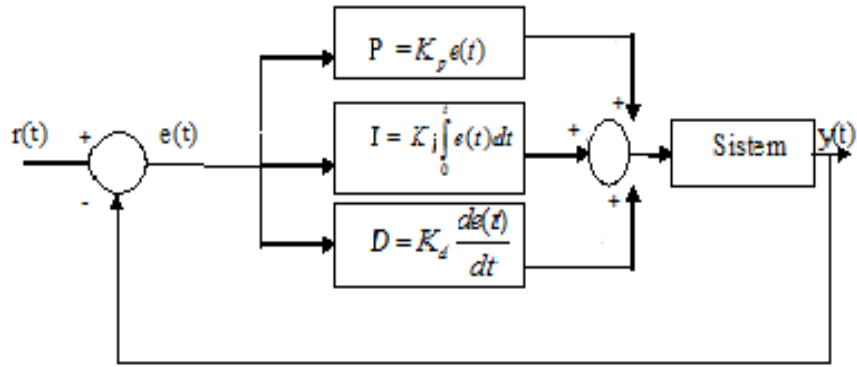
PID kontrol oransal, integral ve trev kavramlarının bir araya gelmesi ile oluşan bir kontrol yntemidir. G elektronięi ve endstriyel uygulamaların oęunda bu yntemi grmek mmkndr.

Burada oransal etki hata değeri azaldıkça daha az bir kontrol etkisi göstererek sistem çıkışının belirlenen referans değerine yumuşak bir şekilde yaklaşmasını sağlar. Yani salınım etkisini en aza indirmek için kullanılır. Fakat hata değeri azaldıkça kontrol etkisi zayıflayacağından dolayı belirli bir değerin altına gelindiğinde sistem çıktısı referans değere asla tam oturmayacaktır. Bu sorunu gidermek için genelde kontrolcü çıkışına sabit bir değer eklenir. Ayrıca oransal etki sistem tepkisini hızlandırır ve aşma değerini artırır (Megep, 2009).

Türevsel etki sistem hatasının değişimine göre orantılı olarak belirlenir. Buradan yola çıkarak türevsel etki bir öngörüdür diyebiliriz. Sistemdeki salınımlar büyüdükçe türevsel kontrol etkisi de artar. Dolayısı ile sistem çıktısı daha kararlı bir davranış gösterir. Ayrıca türevsel etki oransal etki ile birleştiğinde aşma değerini azaltır, sistemin reaksiyon hızını artırır. Ancak türevsel etkinin bir dezavantajı gürültülere karşı sistem hassasiyetini arttırmasıdır.

İntegral etkisi her ne kadar oransal kontrol etkisine benzesede fark olarak burada anlık hata değeri değil onun yerine onun yerine sistem çalışmaya başladığı andan itibaren olan tüm anlardaki hataların toplamı bir kazanç değeri ile çarpılır. Burada bahsi geçen toplam sürekli sistemler için hatanın integralini ifade eder. Sistem cevabının referans olarak belirlenen değere oturması geciktikçe integral etkisi daha fazla artar (Gören, 2016).

PID kontrol bu üç bileşeni bünyesinde bulundurur. Bu sayede hem daha düzenli rejim hem de geçici rejim davranışı iyileşmiş olur. PID kontrol yöntemi aslında bir kontrol geri bildirim sistemidir. Bir PID denetleyici belirli bir zaman aralığında değişen ve önceden belirlenen referans değeri arasındaki farkı alır ve bir hata değeri hesaplar. Bu kontrol yöntemi sayesinde hata en aza indirilir. Genel bir PID kontrol mekanizması aşağıda gösterilmiştir (Şekil 3.11).

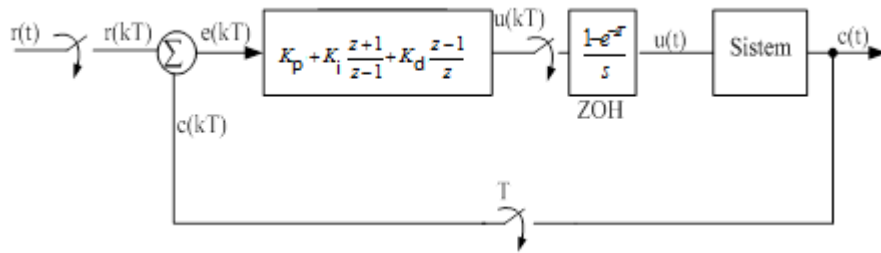


Şekil 3.11. PID kontrol mekanizması.

Burada $r(t)$ referans gerilimini, $e(t)$ hata değerini, $u(t)$ kontrol etkisini $y(t)$ ise çıkış gerilimini ifade eder. Kontrol etkisi $u(t)$ “Eş. 3.43” teki şekilde hesaplanır.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3.43)$$

Dijital kontrol sistemlerin tasarımı sürekli kontrol sistemlerin tasarımına büyük ölçüde benzerlik gösterir. Şekil 3.12’de dijital PID kontrol için blok diyagramı verilmiştir. Dijital kontrol sistemlerinde Z-dönüşümü yapılır. “Eş. 3.44”de Z domeninde PID kontrol etkisi formülü verilmiştir.



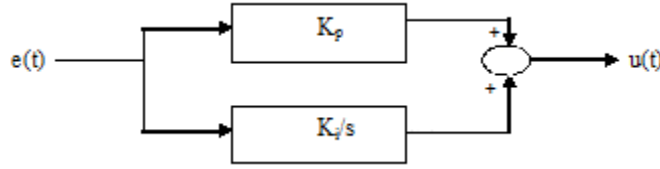
Şekil 3.12. Klasik Dijital PID denetleyici şeması (Özdemir,2018).

$$u(t)_{(z)} = K_p + K_i \frac{z+1}{z-1} + K_d \frac{z-1}{z} \quad (3.44)$$

Bu çalışmada PID kontrol yöntemi kodlar yardımıyla Arduino platformunda tasarlanarak yapılmıştır. Bu özelliği sayesinde uygulama değiştirilmeden parametreler sürekli değiştirilerek en ideal değerler bulunmuştur.

3.2.2. PI kontrol

PI kontrol, kontrol sistemlerinde sıkça kullanılan kontrol tekniklerinden biridir. PI denetleyicinin modeli Şekil 3.13'te verilmiştir.



Şekil 3.13. PI kontrol mekanizması.

PI denetleyicinin genel ifadesi “Eş. 3.45” te verilmiştir.

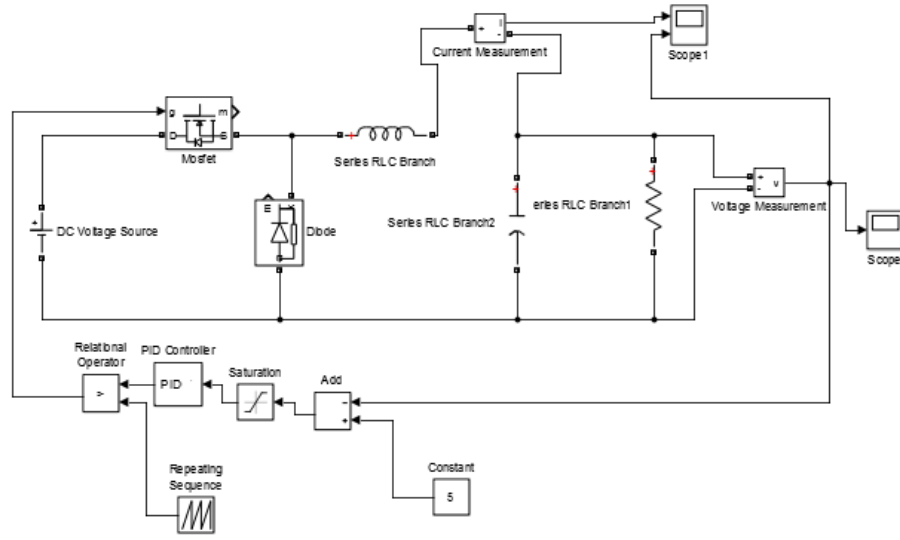
$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (3.45)$$

PID kontrol yönteminde kullanılan türev etkisinin kullanılmaması ile PI kontrol yöntemi elde edilmiş olur. Bu tez çalışmasında daha önce PID kontrol yöntemi için yazılan kodlarda türev etkisi yani K_d değeri sıfır alınarak PI kontrol yöntemi çalışılmış olup daha sonra oturma süreleri bakımından iki yöntem arasında karşılaştırma yapılmıştır.

3.2.3. MATLAB ortamında buck konvertör tasarımı

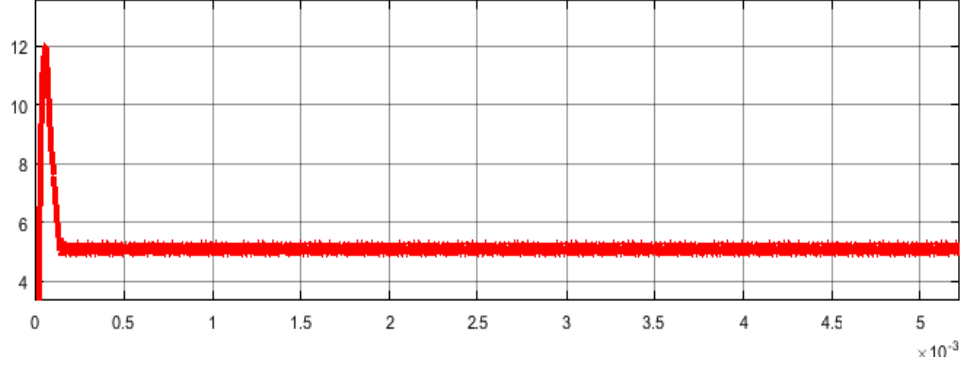
MATLAB, Mathworks şirketi tarafından geliştirilmiş sayısal işlem, grafiksel veri işlemleri, matris işlemleri, fonksiyonel işlemler gibi kabiliyetlere sahip üst düzey

bir yazılımdır. Son yıllarda özellikle mühendislik alanında oldukça yaygın bir kullanım alanına sahiptir. Mühendislik uygulamalarında tasarlanmış sistemi gerçek ortamda denemeden önce MATLAB ortamında simülasyonunu yapmak sistemle ilgili oluşabilecek birtakım sıkıntıları önceden kestirebilmek açısından kolaylık sağlar. Bu çalışmada da sistem gerçek ortamda yapılmadan önce belirlenmiş değerler MATLAB ortamında denenmiştir. MATLAB ortamında tasarlanan PID kontrollü buck konvertör Şekil 3.14'te gösterilmiştir.



Şekil 3. 14. Buck konvertör MATLAB modeli.

Simülasyonda önce PID kontrol yöntemi ile anahtarlama elemanını tetiklemek için PWM sinyali üretilmiştir. Elde edilen çıkış gerilimi Şekil 3.15'te gösterilmiştir.



Şekil 3.15. Simülasyon çıkış gerilimi.

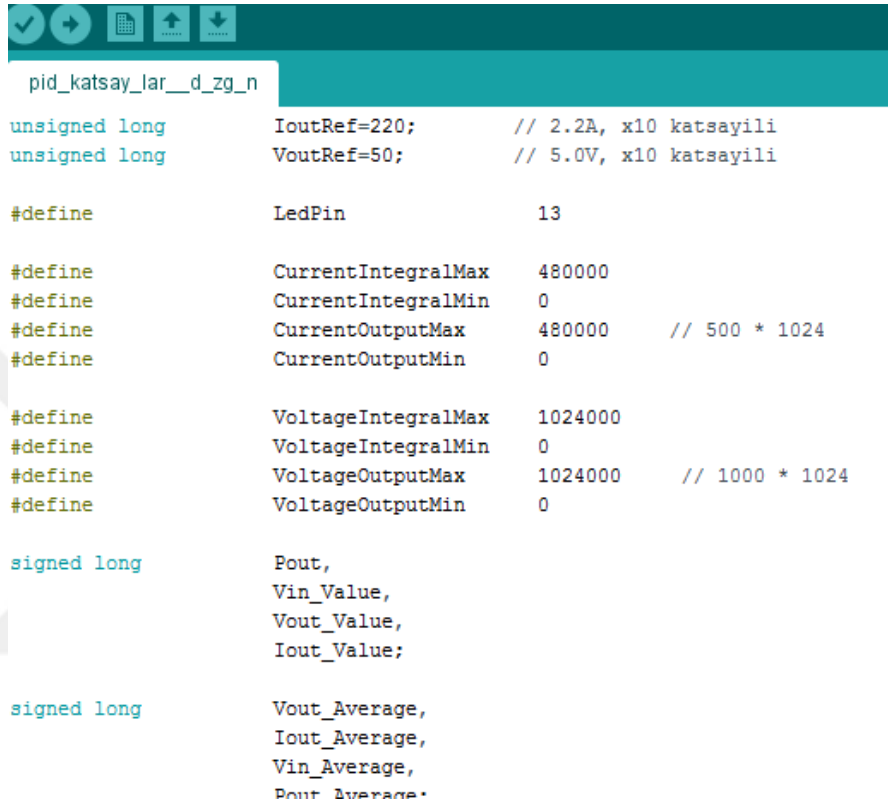
3.2.4. Yazılım tasarımı

3.2.4.1. Mikrodenetleyici programlama

Çalışma içerisinde, buck konvertörün kontrolü için sinyal üreten ve gelen verileri değerlendirme işlemini yapan Arduino Mega 2560 mikrodenetleyici kartın programlanması daha önce bahsedilen Arduino IDE arayüzü ile gerçekleştirilmiştir. Arduino IDE arayüzünde yazılan yazılım, mikrokontrolcünün içerisinde bulunan ve programlama işlemi için ayrılmış olan bootloader kısmı ile mikrodenetleyici karta aktarılır. Yazılan kodlar USB kablo yardımı ile bilgisayarın "COM" portundan mikrodenetleyici karta aktarılır. Bu yüzden ilk ayarlamalar yapılırken port seçimini doğru yapmak oldukça önemlidir. Port seçimi yanlış yapıldığında yazılan yazılım ya da kod parçaları mikrodenetleyiciye aktarılamamış olur.

Buck konvertörün işlevini yapabilmesi için üretilen PWM sinyali PID kontrol yöntemi ile Arduino platformunda kodlanmıştır. Bu işlem yapılırken bilgisayar tarafından gönderilen komutlar mikrodenetleyiciye ulaşır. Kodlar yardımı ile üretilen PWM sinyali sistemin çıkış gerilimini önceden belirlenen referans değerine indirir. Bilgisayardan gönderilen komutların mikrodenetleyiciye ulaşması seri haberleşme yöntemi ile sağlanır. Bu yüzden Arduino kartı seri haberleşmeye uygun şekilde programlanmıştır.

Arduino kartına aktarılmak için yazılan program daha önce de bahsedildiği gibi Arduino IDE editör arayüzünde yazılmıştır (Şekil 3.16). Yazılan bu kodlar açılan editör sayfasında "void setup()" ve "void loop()" yöntemleri içine yazılır.



```

pid_katsay_lar__d_zg_n
unsigned long      IoutRef=220;          // 2.2A, x10 katsayili
unsigned long      VoutRef=50;          // 5.0V, x10 katsayili

#define            LedPin                13

#define            CurrentIntegralMax    480000
#define            CurrentIntegralMin    0
#define            CurrentOutputMax     480000 // 500 * 1024
#define            CurrentOutputMin     0

#define            VoltageIntegralMax    1024000
#define            VoltageIntegralMin    0
#define            VoltageOutputMax     1024000 // 1000 * 1024
#define            VoltageOutputMin     0

signed long        Pout,
                  Vin_Value,
                  Vout_Value,
                  Iout_Value;

signed long        Vout_Average,
                  Iout_Average,
                  Vin_Average,
                  Pout_Average;

```

Şekil 3.16. Arduino IDE kod editörü ekran görüntüsü.

Program yazılırken öncelikle veri boyutuna göre değişken tanımlamaları yapılmıştır. Hemen ardındaki satırlarda #define komutu ile hangi değerin kaç alınacağı ve nerede kullanılacağı tanımlanmıştır.

Arduino IDE kod editörü ekranında görülen “void setup()” yöntemi içerisine, programda bir kez çalıştırılacak olan kodlar eklenmektedir. Tanımlama işlemleri bittikten sonra zaman kesmesi (interrupt) ayarlaması yapılmıştır. Zaman kesmeleri belirlenen süre aralıklarında belirlenmiş görevlerin yapılabilmesi için kullanılır. Zaman kesmelerinin kodda kullanılmasının amacı Arduino programını rahatlatmaktır.

Zaman kesmesi ayarlandıktan sonra çıkış pininin Arduino'nun çıkış pini olarak ayrılmış pinlerinden hangisi olacağı sisteme tanıtılmıştır. Bu adımdan sonra sistemin gerilim ya da akım modlarından hangisinde çalışacağı belirlenmiştir. Bu işlemlerden sonra kodun ilk satırlarında sisteme tanıtılmış referans değeri sistemin çıkış gerilimi ile karşılaştırılmış ve daha önceden deneme yöntemi ile bulunmuş K_p , K_i , K_d parametreleri sisteme tanıtılmıştır. Bu kodlar yardımı ile referans değeri ile sistem çıkışı karşılaştırmasından elde edilen hata değeri en aza indirilmiştir.





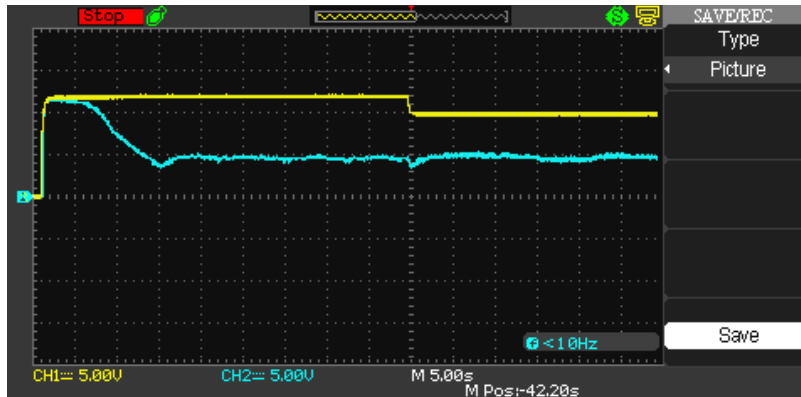
4. BULGULAR VE TARTIŞMA

Bu tez çalışması kapsamında Arduino kullanılarak PID ve PI kontrol yöntemleri ile bir buck konvertör tasarımı yapılmıştır. Sistemin MATLAB ortamında simülasyonu yapıldıktan sonra uygulama devresi yapılmıştır.

Sistemde ilk olarak hedeflenen 12V'luk giriş gerilimini 5V'a düşürmekti. Bu adım gerçekleştirildikten sonra sistem çalışmaya devam ederken 12V olan giriş gerilimi aniden 10V'a düşürülmüş ve sistemin tepkisi ölçülmüştür. Bu işlem her iki kontrol yönteminde de denenmiştir. Giriş geriliminde yapılan değişikliklerden sonra 5V olan referans değeri 7V ve 3V olacak şekilde değiştirilip sistemin ne kadar sürede oturduğuna bakılmıştır. Elde edilen sonuçlar osiloskop yardımı ile alınmış ve daha sonra MATLAB ortamında grafikleri elde edilmiştir.

4.1. Giriş Geriliminin Değiştirilmesi

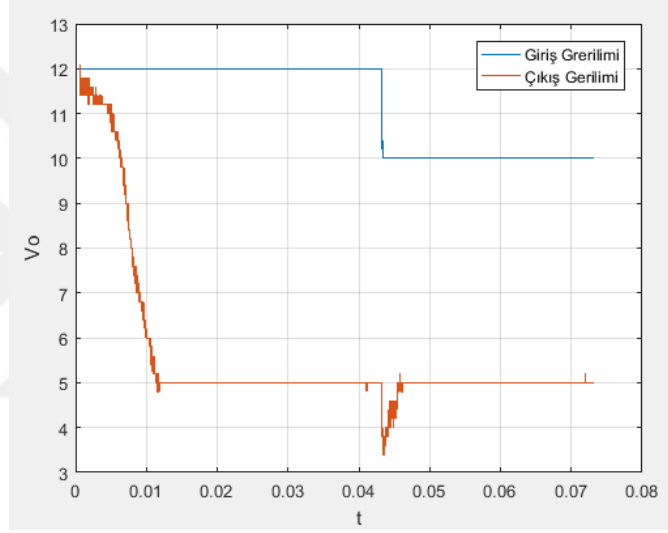
Sistemin giriş gerilimi sistem çalışırken 12V'dan 10V'a düşürülerek sistem tepkisine bakılmıştır. Giriş gerilimi 10V'a düşerken çıkış geriliminin 5V olarak kalması istenmiştir. İlk olarak PID kontrol yöntemi ile denenmiştir. Ek 1'de bu işleme ait Arduino kodları verilmiştir. Osiloskop görüntüsü Şekil 4.1'de verilmiştir.



Şekil 4.1. PID kontrol yöntemi ile değiştirilmiş giriş değerine sistemin tepkisi.

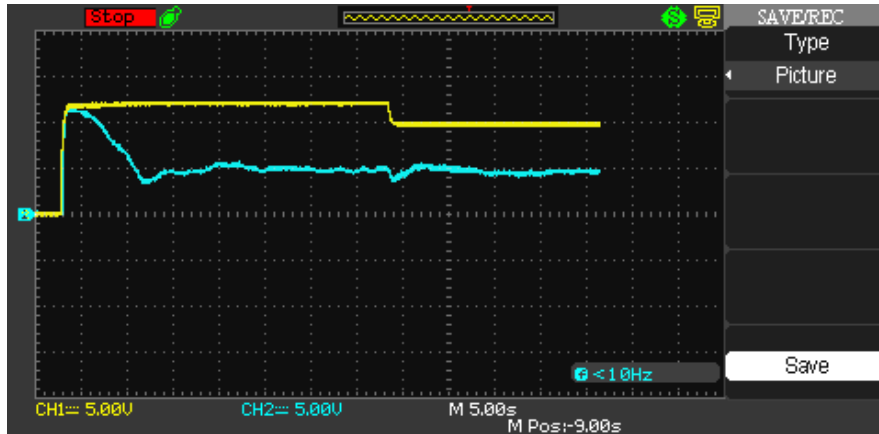
Bu çalışma sonucunda elde edilen veriler MATLAB ortamına aktarılmış ve daha anlaşılır olması açısından orada da grafiklendirilmiştir. Verilerin MATLAB'a aktarılmasına yardımcı kodlar Ek 2'de verilmiştir.

Osiloskoptan alınan excel dosyası formatındaki verilerin 'filename' komutu ile dosya yolu oluşturulur. Daha sonra 'xlsread' komutu ile veriler MATLAB'a tanıtılır. Grafikte görülmesi istenen verilerde tanıtıldıktan sonra 'plot' komutu ile grafik çizdirilir. Aniden değiştirilen giriş gerilimine karşı sistemin zaman-çıkış gerilimi grafiği Şekil 4.2'de verilmiştir.



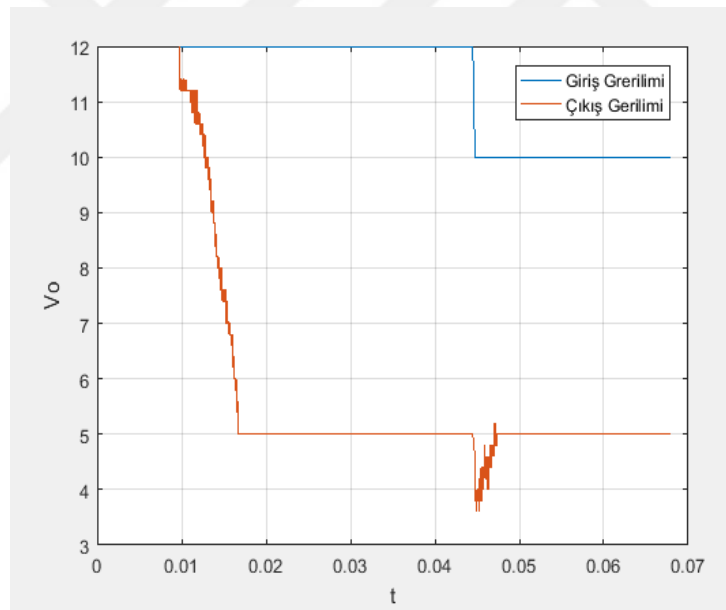
Şekil 4.2. Aniden değiştirilen giriş gerilimine karşı sistemin tepkisi (PID kontrol).

Grafikten görüldüğü üzere sistem 12V'dan 5V'a düşürülmüş oturması beklendikten sonra aniden giriş gerilimi 10V'a çekilmiştir. 0.04 ile 0.05. saniyeler arasında sistem çıkışı bu ani değişime tepki vermiş, değişimi algıladıktan sonra hızlı bir şekilde toparlanıp yeniden 5V'a oturmuştur. Kıyas yapılabilmesi açısından aynı işlem PI kontrol yöntemi ile de yapılmıştır. Bu yöntemle ait Arduino bloğu Ek 3'te verilmiştir. PI kontrol yöntemi kullanılarak elde edilen osiloskop görüntüsü Şekil 4.3'te verilmiştir.



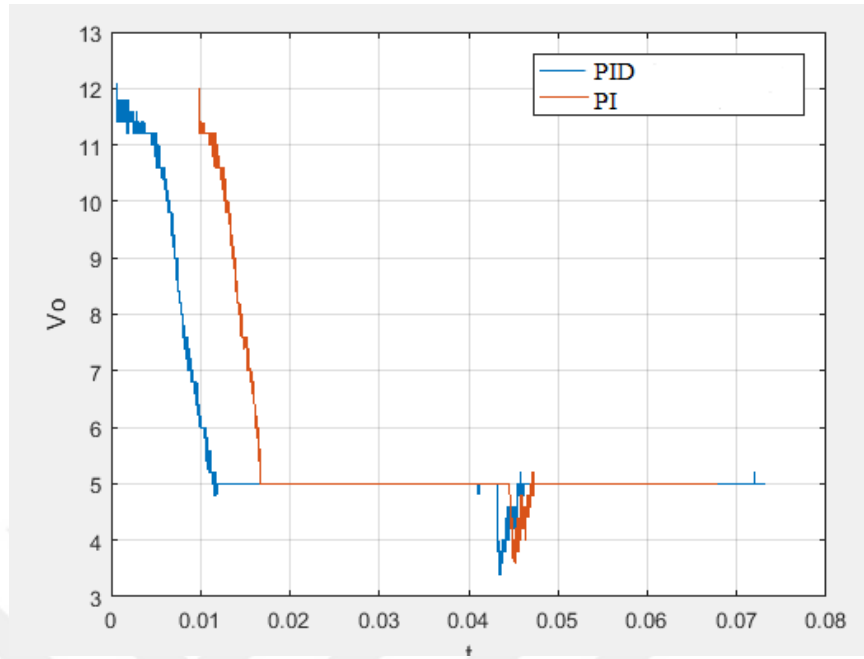
Şekil 4.3. PID kontrol yöntemi ile değiştirilmiş giriş değerine sistemin tepkisi.

Osiloskoptan alınan veriler yine aynı şekilde MATLAB ortamına aktarılmış olup elde edilen grafik Şekil 4.4'te verilmiştir.



Şekil 4.4. Aniden değiştirilen giriş gerilimine karşı sistemin tepkisi (PI kontrol).

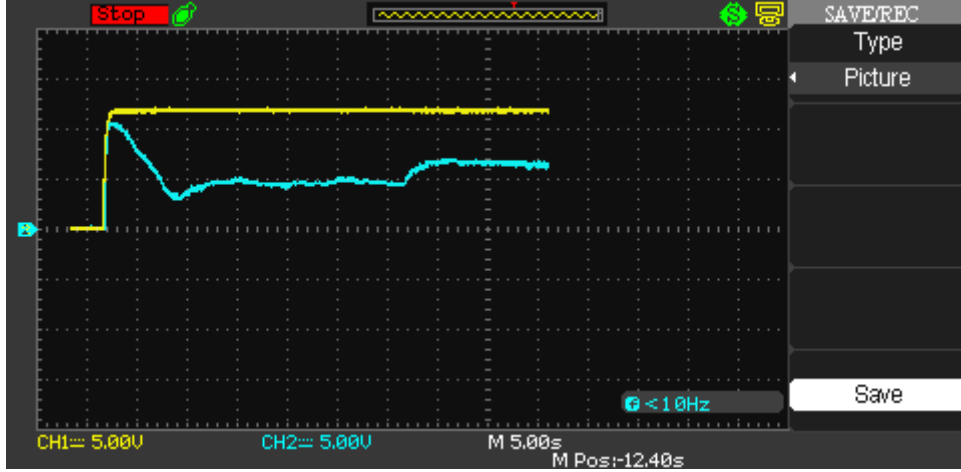
İki grafik incelendiğinde PID kontrol yöntemi ile sistem daha hızlı toparlanmış ve istenen referans değerine daha çabuk oturmuştur. Daha rahat bir karşılaştırma için Şekil 4.5'de Şekil 4.2 ve 4.4 üst üste çizdirilmiştir. Bu işleme ait MATLAB kodları Ek 4'de verilmiştir.



Şekil 4.5. PID ve PI kontrol karşılaştırması.

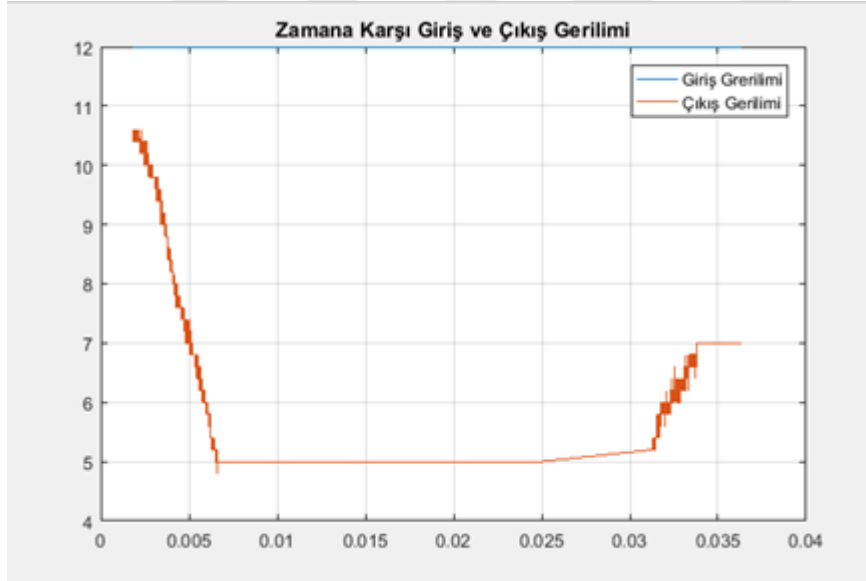
4.2. Referans Değerinin Değiştirilmesi

Giriş gerilimindeki değişikliklere karşı sistemin nasıl değiştiği incelendikten sonra referans değerlerinde bir takım değişiklikler yapılmıştır. Referans değeri 5V iken önce 7V olarak değiştirilmiş ve sistem incelenmiştir. Şekil 4.6'da yapılan değişikliği gösteren osiloskop görüntüsü verilmektedir.



Şekil 4.6. Referans değeri 7V olarak değiştirildiğinde osiloskop sistem tepkisi.

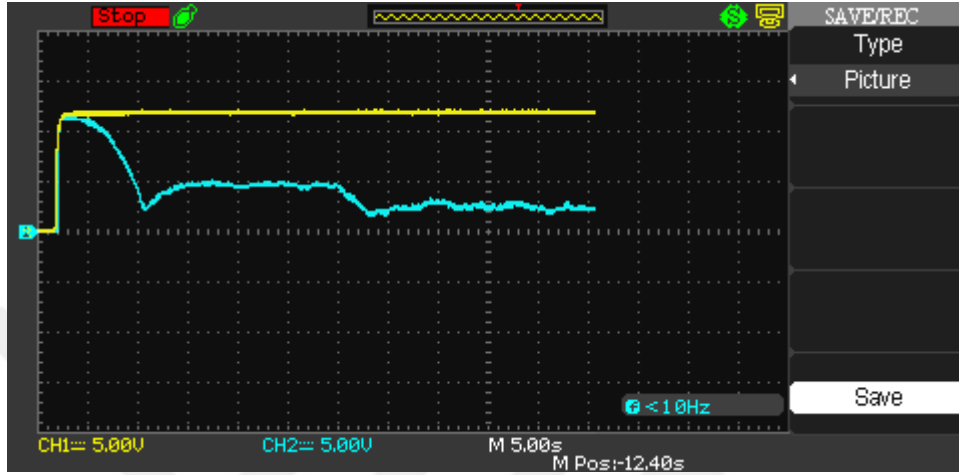
Osiloskoptan alınan Excel formatındaki veriler daha önce yapıldığı gibi MATLAB ortamına aktarılmış ve elde edilen sonuçlar MATLAB ortamında grafiklendirilmiştir (Şekil 4.7).



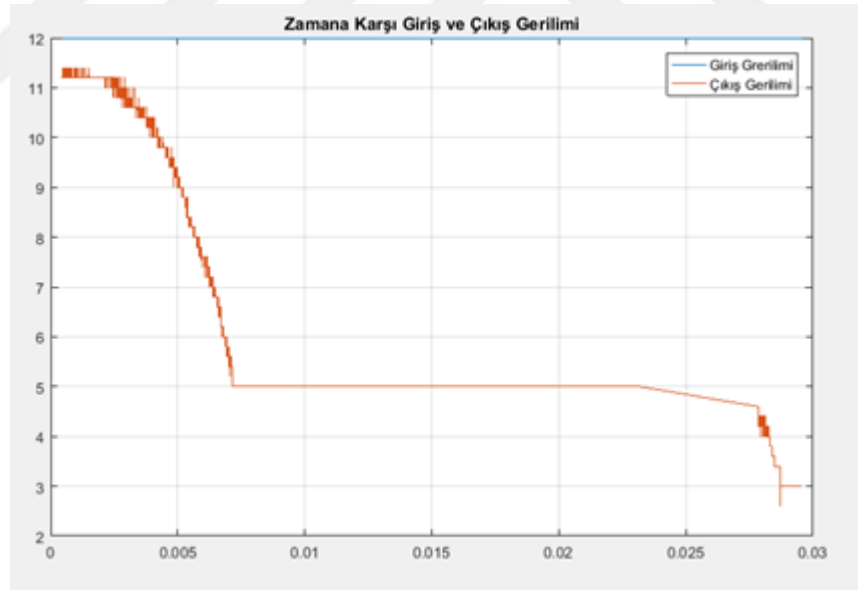
Şekil 4.7. Referans değeri 7V olarak değiştirildiğinde sistem tepkisi (MATLAB).

Grafikte görüldüğü üzere sistem 0.005 ile 0.01saniyeler arasında ilk referans değeri olan 5V'a oturmuş daha sonra sistem çalışırken referans değeri 7V olarak değiştirildiğinde sistem önce bozulmalar yapmış daha sonra toparlayarak 7V'a oturmuştur. Bu değişikliği yapmayı sağlayan Arduino kodları Ek 5'te verilmiştir. Aynı şekilde referans değeri 3V olarak değiştirilmiş ve sistem tepkisi incelenmiştir. Referans

değerinin değişimine karşı sistemin tepkisinin osiloskop görüntüsü Şekil 4.8’de verilmiştir. Referans 3V iken MATLAB ortamından alınan grafik ise Şekil 4.9’da gösterilmektedir. Bu işleme ait Arduino kodları Ek 6’da verilmiştir.



Şekil 4.8. Referans değeri 3V olarak değiştirildiğinde sistem tepkisi (osiloskop).



Şekil 4.9. Referans değeri 3V olarak değiştirildiğinde sistem tepkisi (MATLAB).

Ürgün ve ark. (2008), yaptıkları çalışmada maliyeti düşürmek amacıyla dsPIC mikrodenetleyici kullanmışlardır. Bu çalışmadan farklı olarak tezimizde mikrodenetleyici olarak Arduino kullanılmıştır. Bu sayede üzerinde değişiklik yapılmasının daha rahat olduğu bir sistem ortaya çıkmıştır.

Al-Juboori (2017), yaptığı çalışmada Arduino kullanarak AA bir gerilim regülatörü tasarlamıştır. Bu çalışmadan farklı olarak tezimizde AA yerine DA gerilim üzerinde çalışılmıştır. Böylelikle DA uygulamalarda da Arduino ile iyi bir sonuç alınabileceği görülmüştür.





5. SONUÇ

Bu tez çalışmasında 12V ile 5V aralığında çalışan bir buck konvertör tasarımı yapılmıştır. Buck konvertör tasarımı önce MATLAB ortamında simule edilmiş parametrelerin uygunluğu kontrol edilmiştir. Daha sonra uygulama devresi yapılmış, bu devreye Arduino eklenmiştir. Sistem tepkilerini inceleyebilmek adına ilk olarak sistem çalışır haldeyken gerilim kaynağı kullanılarak giriş gerilimi 12V'dan aniden 10V'a düşürülmüştür. Bu işlem yapılırken 9V'a kadar gerilim düşümü sağlanmış ancak 9V'dan sonraki gerilimlerde sistemde kullanılan anahtarlama elemanlarının sınır gerilimlerinden dolayı devrenin çalışmadığı gözlemlenmiştir. Bu ani gerilim değişimleri hem PID hem PI kontrol yöntemi ile denenmiş ve sonuç olarak sistemin PID kontroldeki türev etkisinden dolayı PID kontrol yöntemi ile istenen referans değerine daha çabuk oturduğu gözlemlenmiştir. Ayrıca gerilim ile oynamalar yapıldığı sırada sistemin belirli bir noktada zıplama yaptığı fark edilmiş ve bu istenmeyen etkinin güç kaynağından kaynaklı olduğu görülmüştür. Bu sorun daha yüksek güçte güç kaynağı kullanılarak çözülmüştür.

Sistem tepkisini izleyebilmek için yapılan bir diğer işlem ise referans değerini değiştirmek olmuştur. Daha önce 5V olarak belirlenen referans değeri önce 7V olarak değiştirilmiş ve sistemin ne kadar sürede referans değerine ulaştığı takip edilmiştir. Referans değeri değiştirilirken 4V ve 6V gerilimlerde denenmiş fakat ilk olarak belirlenen referans değerine çok yakın olduklarından net bir grafik elde edilememiştir. Daha belirgin bir farklılık izleyebilmek adına 7V ve 3V seçilmiştir. Arduino'da yazılan kodlarda bir takım değişiklikler yapılarak uygulama devresi değiştirilmeden referans değerleri kolayca değiştirilebilmiştir. Bu da bu tarz uygulamalarda Arduino kullanmanın rahatlığını ispatlar niteliktedir.

Bu çalışma neticesinde dönüştürücü tasarımında dijital yöntemlerin kullanılmasının maliyet, ebat, kolay programlanabilme gibi avantajları görülmüş ayrıca Arduino kullanılarak tasarlanan sistemlerde PID ve PI kontrol yöntemlerinden PID kontrolün daha başarılı olduğu görülmüştür.



KAYNAKLAR

- Al-Juboori, A.A.J., 2017. *Arduino Controlled AC Voltage Regulator* (yüksek lisans tezi). THKÜ, Fen Bilimleri Enstitüsü, Ankara.
- Anonim, 2015. Arduino Mega 2560 mikrodenetleyici kartı. <https://arduino.cc/en/Main/arduinoBoardMega2560>. Erişim tarihi: 18.10.2019.
- Anonim, 2015. IRFZ44 mosfete ait datasheet. <http://datasheetcafe.databank.netdna-cdn.com/wp-content/uploads/2017/03/B20100G.pdf>. Erişim tarihi: 21.10.2019.
- Arifoğlu, B., 2005. Otomasyon sistemleri. <https://www.scribd.com/document/399142479/OTOMASYON-SİSTEMLERİ>. Erişim tarihi: 22.10.2019.
- Arifoğlu, U., 2018. *Güç Elektroniği Endüstriyel Uygulamaları Ders Notu*. Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.
- Bodur, H., 2010. *Güç Elektroniği*. Birsen Yayınevi, İstanbul.216.
- Dutta, R., Kumar, N., Pankaj, D. 2014. PID control for ambulatory gait orthosis: application of different tuning methods. *Advances in Biomedical Engineering Research*, 2: 44-49.
- Evrar, F., 2019. Azaltan DA-DA dönüştürücüsü için kayma kip denetleyici tasarımı. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 7 (1): 519-530.
- Giaouris, D., Banerjee, S., Zahawi, B., Pickert, V., 2008. Stability analysis of the continuous-conduction-mode buck converter via Filippov's method. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55, (4): 1084-1096.
- Gidemen, G., Furat, M., 2015. PID parametrelerini ayarlama yöntemleri: 2. derece sistem modeline uygulanması ve karşılaştırmalı olarak değerlendirilmesi. *Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 30,(2): 355-362.
- Gören, A., 2016. Kontrol sistemleri. <http://kisi.deu.edu.tr/aytac.goren/MAK3026/h5.pdf>. Erişim Tarihi: 01.11.2019.
- Guo, L., Hung, J. Y., Nelms, R. M., 2003. Digital controller design for buck and boost converters using root locus techniques. *In IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society* 2:1864-1869.
- Gupta, S., Kaur, A., Garg, A., Verma, A., Bansal, A., Singh, A., 2013. Arduino based smart cart. *International Journal of Advanced Research in Computer Engineering & Technology*, 2: 12.
- Hekimoğlu, B., Ekinci, S., Kaya S., Balina optimizasyon algoritması kullanılarak DA-DA düşürücü dönüştürücünün optimum PID denetleyicili tasarımı. *International Conference on Artificial Intelligence and Data Processing (IDAP)*, 28-30 September, 2018. Malatya.
- Kamil, M., 2007. Switch mode power supply topologies. *Microchip Technology Inc. AN1114*.
- Kovačević, H., Stojanović, Ž., 2016. Buck converter controlled by Arduino Uno. *39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 1638-1642.
- Köse, E., Mühürücü, A., Mühürücü, G., Temiz, M. N., 2016. Optimal control of a buck converter by using PI controller based on HSA algorithm. *1 st International*

- Mediterranean Science and Engineering Congress (IMSEC 2016)*, October 26-28, 2016, Adana.
- Mahdavi, J., Emadi, A., Toliyat, H. A., 1997. Application of state space averaging method to sliding mode control of PWM DC/DC converters. *IAS'97. Conference Record of the 1997 IEEE Industry Applications Conference Thirty-Second IAS Annual Meeting* October 1997, (2): 820-827.
- Mamur A. 2012. *FPGA Denetimli Düşürücü DA-DA Dönüştürücünün Tasarımı ve Gerçekleşmesi* (yüksek lisans tezi). FÜ, Fen Bilimleri Enstitüsü, Eskisehir.
- Megep., 2009. *PID kontrol*. Milli Eğitim Bakanlığı Mesleki Eğitim ve Öğretim Sistemlerinin Güçlendirilmesi Projesi. Endüstriyel Otomasyon Teknolojileri.
- Megep., 2012. *Mikroişlemci ve mikrodenetleyiciler*. Milli Eğitim Bakanlığı Mesleki Eğitim ve Öğretim Sistemlerinin Güçlendirilmesi Projesi. Endüstriyel Otomasyon Teknolojileri.
- Mondal, A. 2014. *Digital PID Controller Design For DC-DC Buck Converter* (doctoral thesis). National Institute of Technology, Rourkela.
- Müller, L., Mohammed, M., Kimball, J. W., 2015. Using the Arduino Uno to teach digital control of power electronics. *In 2015 IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL)*:1-8.
- Ogata, K., 2009. *Modern Control Engineering*, Upper Saddle River, NJ: Prentice Hall. 6142-6143.
- Özdemir, Ö., 2018. *Sürekli Akım Modunda Çalışan Alçaltıcı Tip DA-DA Dönüştürücülerin Gerilim Kontrolü* (yüksek lisans tezi). SAÜ, Fen Bilimleri Enstitüsü, Sakarya.
- Özkan, G. 2007. *Yakut Pilleri Uygulamaları İçin Dijital Kontrollü DA-DA Dönüştürücü Devre Tasarımı* (yüksek lisans tezi). Eskisehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskisehir.
- Rashid, M. H., 2004. *Power Electronics: Circuits, Devices and Applications*. In University of West Florida. Pearson Prentice Hall.
- Rashid, M. H., 2014. *Power Electronics: Circuits, Devices, and Applications*., Front Cover, 4th Edition.
- Rao, P. V. G. K., Subramanyam, M. V., Satyaprasad, K., 2014. Study on PID controller design and performance based on tuning techniques, *International Conference on Control, Instrumentation, Communication and Computational Technologies*, India, 1411-1417.
- Taşdemir, C., 2015. *Arduino*. Dikeyksen, İstanbul.264.
- Trip, N. D., Dale, S., Popescu, V., 2010 Digital control for switched mode DC-DC buck converters. *In 2010 9th International Symposium on Electronics and Telecommunications IEEE*. November. 2010, 99-102.
- Türkmen, O., 2017. *Bilgisayar Kontrollü Rehabilitatif Kol Ortezi Tasarımı* (yüksek lisans tezi). YYÜ, Fen Bilimleri Enstitüsü, Van.
- Ürgün, S., T. Erfidan, N. Çoruh. DA-DA buck dönüştürücü tasarımı ve gerçekleşmesi. *Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu* 2008,:358-361.
- Yalduz, H., 2015. *DA-DA Düşüren Dönüştürücü Tasarımı ve PI Kontrollü* (yüksek lisans tezi). YYÜ, Fen Bilimleri Enstitüsü, Van.

- Yurdakul, S.S., Eker, İ., 2009. Kendinden ayarlamalı PID kontrol ve deneysel uygulamaları. *V.OTOMASYON SEMPOZYUMU, EMO*, 17-10 Mayıs 2009, İzmir.
- Zaman, R.U., 2014. *Design of a Buck-Boost Converter and Its Implementation in a Real System* (yüksek lisans tezi). YÜ, Fen Bilimleri Enstitüsü, İzmir.





EKLER

Ek 1. . PID kontrol yöntemi ile referans değerini değiştiren Arduino kodları.

```
unsigned long    IoutRef=220;    // 2.2A, x10 katsayili
unsigned long    VoutRef=50;     // 5.0V, x10 katsayili

#define          LedPin          13

#define          CurrentIntegralMax  480000
#define          CurrentIntegralMin  0
#define          CurrentOutputMax    480000 // 500 * 1024
#define          CurrentOutputMin    0

#define          VoltageIntegralMax  1024000
#define          VoltageIntegralMin  0
#define          VoltageOutputMax    1024000 // 1000 * 1024
#define          VoltageOutputMin    0

signed long      Pout,
                Vin_Value,
                Vout_Value,
                Iout_Value;

signed long      Vout_Average,
                Iout_Average,
                Vin_Average,
                Pout_Average;

signed long      Vout_Temp,
```

```
Iout_Tem
Vin_Temp,
Pout_Temp;

signed short    Average_Counter;

unsigned char   StartStop =0;

const signed long  Ki_P = 20,
                  Ki_I = 1,
                  Ki_D = 0,
                  Kv_P = 1000,
                  Kv_I = 5,
                  Kv_D = 0;

unsigned short   Uart_Tick_Counter,
                  Meas_Tick_Counter;

signed long      CurrentRef,
                  CurrentError,
                  CurrentPreError,
                  CurrentPD_Term,
                  CurrentI_Term,
                  CurrentIntegral,
                  CurrentDerivative,
                  CurrentOutput;
```

```

signed long      VoltageError,
                VoltagePreError,
                VoltagePD_Term,
                VoltageI_Term,
                VoltageIntegral,
                VoltageDerivative,
                VoltageOutput;

long            ControlOutput;

// PWM 16 bit Timer1 birimi kullanılarak yapılacak
// Frekans 16kHz olarak ayarlanacak
// Bu sekilde kullanmak icin datasheet kullanip islemcinin register degerlerine
ulasiliyor
void PWM_Init(void)
{
    TCCR1B &= ~(1 << CS12);
    TCCR1B &= ~(1 << CS11);
    TCCR1B |= (1 << CS10);

    ICR1=800;

    TCCR1B |= (1 << WGM13); // Timer B clear bit 4
    TCCR1B |= (1 << WGM12); // set bit 3

    TCCR1A |= (1 << WGM11); // Timer A set bit 1
    TCCR1A &= ~(1 << WGM10); // set bit 0

    pinMode(9, OUTPUT);
    analogWrite(9, 0);

```

```

}

// PID ayarlari icin 1kHz'lik Timer kuruluyor!
// (16.000.000Hz/64)/250=1000Hz
void Timer0_Init(void)
{
  TCCR0A = 0; // Basta ayarlar sifirlaniyor
  TCCR0B = 0; // Basta ayarlar sifirlaniyor
  TCNT0 = 0; // Basta ayarlar sifirlaniyor

  OCR0A = 250; // 1kHz'lik sinyal icin timer degeri

  TCCR0A |= (1 << WGM01);
  TCCR0B |= (1 << CS01) | (1 << CS00); // On bolucu 1:64
  TIMSK0 |= (1 << OCIE0A); // Timer compare interrupt'ı aciliyor
}

// Bu kisimda Arduino'nun ilk ayarlari yapiacal
// UART BaudRate 19200 olarak ayarlaniyor, Timer0 1kHz olacak, PWM
frekans 16kHz
void setup()
{
  Serial.begin(1000000);
  while(!Serial);

  PWM_Init();
  Timer0_Init();

  pinMode(13, OUTPUT); // Gerilim ya da akim modunda calisildigini
gosteriyor
}

```

```

// Bu fonksiyonda daha dogru olcum icin ortalama alinacak
// Degerlerin ortalamasi aliniyor
void GetMeasurements(void)
{
    // read the analog in value:
    Vout_Value = map(analogRead(0), 0, 1023, 0, 240); // Direnc oranlarindan
dolayi Vout gerilimi 0 ile 23.5V araliginda olculebiliyor
    Vin_Value = map(analogRead(1), 0, 1023, 0, 240); // Direnc oranlarindan
dolayi Vin gerilimi 0 ile 23.5V araliginda olculebiliyor
    Iout_Value = map(analogRead(2), 0, 1023, 0, 320); // Direnc oranlarindan
dolayi Iout akimi 0 ile 10.6A araliginda olculebiliyor
    Pout = Vout_Average*Iout_Average;

    if(Average_Counter<128)
    {
        Vout_Temp +=Vout_Value;
        Iout_Temp +=Iout_Value;
        Vin_Temp +=Vin_Value;

        Average_Counter++;
    }
    else
    {
        Vout_Average =((Vout_Temp>>7)+Vout_Average)>>1;
        Iout_Average =((Iout_Temp>>7)+Iout_Average)>>1;
        Vin_Average =((Vin_Temp>>7)+Vin_Average)>>1;

        Vout_Temp =0;
        Iout_Temp =0;
        Vin_Temp =0;
    }
}

```



```

    Average_Counter =0;
}
}

// Akim ve gerilimin PI hesapları yapılıp, minimum deger cikis olarak yaziliyor
void PI_Control(void)
{
    if(StartStop==1) // Cevrim baslatiliyor
    {
        DoVoltageControl();
        VoltageError =VoutRef-Vout_Value;

        if(VoltageError<-5)
        {
            CurrentRef = 0;
        }
        else
        {
            CurrentRef = (IoutRef*VoltageOutput)>>10;
        }

        DoCurrentControl();

        ControlOutput=CurrentOutput;
    }
    else // Cevrim durduruluyor baslatiliyor
    {
        CurrentIntegral =0;
        VoltageIntegral =0;
        CurrentOutput =0;
    }
}

```

```

VoltageOutput =0;
ControlOutput =0;
}

if(ControlOutput>600)
    ControlOutput=600;

analogWrite(9, ControlOutput);
}

// Bu kisimda cikis akiminin PI hesaplamasi yapiliyor
void DoCurrentControl(void)
{
/*-----Current PI Control Algorithm-----*/
// Calculate Error Value
CurrentError =CurrentRef-Iout_Value;

if(CurrentError>20) CurrentError=20;
if(CurrentError<-20) CurrentError=-20;

// P-D degerleri hesaplaniyor
CurrentPD_Term = (CurrentError*Ki_P) + (CurrentError-
CurrentPreError)*Ki_D;

CurrentI_Term = CurrentError*Ki_I;

// I degeri hesaplaniyor
CurrentIntegral = CurrentIntegral + CurrentI_Term;

if(CurrentIntegral>=CurrentIntegralMax) CurrentIntegral=CurrentIntegralMax;
if(CurrentIntegral<=CurrentIntegralMin) CurrentIntegral=CurrentIntegralMin;

```

```

CurrentOutput = CurrentPD_Term + CurrentIntegral;

if(CurrentOutput>=CurrentOutputMax) CurrentOutput=CurrentOutputMax;
if(CurrentOutput<=CurrentOutputMin) CurrentOutput=CurrentOutputMin;

CurrentOutput=CurrentOutput>>10;

CurrentPreError = CurrentError;
}

// Bu kisimda cikis akiminin PI hesaplamasi yapiliyor
void DoVoltageControl(void)
{
/*-----Voltage PI Control Algorithm-----*/
// Calculate Error Value
VoltageError =VoutRef-Vout_Value;

if(VoltageError>20) VoltageError=20;
if(VoltageError<-20) VoltageError=-20;

// P-D degerleri hesaplaniyor
VoltagePD_Term = (VoltageError*Kv_P) + (VoltageError-
VoltagePreError)*Kv_D;

VoltageI_Term = VoltageError*Kv_I;

// I degeri hesaplaniyor
VoltageIntegral = VoltageIntegral + VoltageI_Term;

```

```

    if(VoltageIntegral>=VoltageIntegralMax)
VoltageIntegral=VoltageIntegralMax;
    if(VoltageIntegral<=VoltageIntegralMin) VoltageIntegral=VoltageIntegralMin;

VoltageOutput = VoltagePD_Term + VoltageIntegral;

if(VoltageOutput>=VoltageOutputMax) VoltageOutput=VoltageOutputMax;
if(VoltageOutput<=VoltageOutputMin) VoltageOutput=VoltageOutputMin;

VoltageOutput = VoltageOutput>>10;

VoltagePreError = VoltageError;
}

// UART'tan gönderilecek bilgiler buradan gonderiliyor
void UART_Send(void)
{
    Serial.print("Vin = ");          Serial.print((float)Vin_Average/10,1);
Serial.print("V");
    Serial.print("\t Vout = ");      Serial.print((float)Vout_Average/10,1);
Serial.print("V");
    Serial.print("\t Iout = ");      Serial.print((float)Iout_Average/100,2);
Serial.print("A");
    Serial.print("\t          Pout          =          ");
Serial.print((float)(Vout_Average*Iout_Average)/1000,1); Serial.print("W");
    Serial.print("\t Duty = ");      Serial.print(ControlOutput);
    Serial.print("\t Start/Stop = "); Serial.println(StartStop);

    StartStop=1;
}

```

```

void loop()
{
  StartStop = 1;

  if(Uart_Tick_Counter>499)    {UART_Send();    Uart_Tick_Counter=0;
digitalWrite(LedPin, !digitalRead(LedPin)); } // Her 500ms'de bir bu kısma giriliyor
  if(Meas_Tick_Counter>0)    {GetMeasurements(); Meas_Tick_Counter=0; }
// Her 1ms'de bir bu kısma giriliyor
}

// Timer0 kesmesi bu kısma geliyor, kesme frekansı 1kHz
ISR(TIMER0_COMPA_vect)
{
  Uart_Tick_Counter++;
  Meas_Tick_Counter++;

  noInterrupts();
  interrupts();
}

```

Ek 2. PID kontrol yöntemi ile elde edilen verilerin grafik oluşturma kodları.

```
clear all;close all;clc;
```

```
filename='pid.xlsx'; %Dosya yolu
```

```
VERI1=xlsread(filename); %Dosyanın matlabe tanıtılması
```

```
%VERI1 matrisinin tanıtılması
```

```
ZAMAN1=VERI1(:,1);
```

```
GIRIS_GERILIM1=VERI1(:,2);
```

```
CIKIS_GERILIM1=VERI1(:,3);
```

```
filename='pi.xlsx'; %Dosya yolu
```

```
VERI2=xlsread(filename); %Dosyanın matlabe tanıtılması
```

```
%VERİ1 matrisinin tanıtılması
```

```
ZAMAN2=VERI2(:,1);
```

```
GIRIS_GERILIM2=VERI2(:,2);
```

```
CIKIS_GERILIM2=VERI2(:,3);
```

```
figure()
```

```
plot(ZAMAN1,GIRIS_GERILIM1);hold on;
```

```
plot(ZAMAN1,CIKIS_GERILIM1);
```

```
title('Zamana Karşı Giriş ve Çıkış Gerilimi (pid.xlsx)');
```

```
legend('Giriş Gerilimi','Çıkış Gerilimi');
```

```
figure()
```

```
plot(ZAMAN2,GIRIS_GERILIM2);hold on;
```

```
plot(ZAMAN2,CIKIS_GERILIM2);
```

```

title('Zamana Karşı Giriş ve Çıkış Gerilimi (pi.xlsx)');
legend('Giriş Grerilimi','Çıkış Gerilimi');
figure()
plot(ZAMAN1,GIRIS_GERILIM1);hold on;
plot(ZAMAN2,GIRIS_GERILIM2);hold on;
title('Zamana Karşı Çıkış Gerilimi Karşılaştırılması');
legend('Giriş Grerilimi(pid.xlsx)','Giriş Gerilimi(p.xlsx)');
figure()

```

Ek 3. PI kontrol yöntemi referans değerini aniden değiştiren Arduino kodları.

```

const signed long   Ki_P = 20,
                   Ki_I = 1,
                   Ki_D = 0,
                   Kv_P = 1000,
                   Kv_I = 5,
                   Kv_D = 0;

```

Ek 4. PID ve PI kontrol karşılaştırması yapan verilerin grafik oluşturma kodları.

```

clear all;close all;clc;

filename='pid.xlsx'; %Dosya yolu

VERI1=xlsread(filename); %Dosyanın matlabe tanıtılması

%VERI1 matrisinin tanıtılması

ZAMAN1=VERI1(:,1);

GIRIS_GERILIM1=VERI1(:,2);

CIKIS_GERILIM1=VERI1(:,3);

```

```
filename='pi.xlsx'; %Dosya yolu
VERI2=xlsread(filename); %Dosyanın matlabe tanıtılması
%VERİ1 matrisinin tanıtılması
ZAMAN2=VERI2(:,1);
GIRIS_GERILIM2=VERI2(:,2);
CIKIS_GERILIM2=VERI2(:,3);
figure()
plot(ZAMAN1,GIRIS_GERILIM1);hold on;
plot(ZAMAN1,CIKIS_GERILIM1);
title('Zamana Karşı Giriş ve Çıkış Gerilimi (pid.xlsx)');
legend('Giriş Gerilimi','Çıkış Gerilimi');
figure()
plot(ZAMAN2,GIRIS_GERILIM2);hold on;
plot(ZAMAN2,CIKIS_GERILIM2);
title('Zamana Karşı Giriş ve Çıkış Gerilimi (pi.xlsx)');
legend('Giriş Gerilimi','Çıkış Gerilimi');
figure()
plot(ZAMAN1,GIRIS_GERILIM1);hold on;
plot(ZAMAN2,GIRIS_GERILIM2);hold on;
title('Zamana Karşı Giriş Gerilimi Karşılaştırılması');
legend('Giriş Gerilimi(pid.xlsx)','Giriş Gerilimi(p.xlsx)');
figure()
plot(ZAMAN1,CIKIS_GERILIM1);hold on;
plot(ZAMAN2,CIKIS_GERILIM2);hold on;
```



```

legend('Çıkış Gerilimi(pid.xlsx)', 'Çıkış Gerilimi(p.xlsx)');
xlabel('t');
ylabel('Vo');
grid;

```

Ek 5. Referans değerini 7V yapan Arduino kodları.

```

unsigned long      IoutRef=220;      // 2.2A, x10 katsayili
    unsigned long      VoutRef=50;      // 5.0V, x10 katsayili
    unsigned long      VoutRef2=70;      // 7.0V, x10 katsayili
if(VoltageError<-7)
    {
        CurrentRef = 0;
    }
else
    {
        CurrentRef = (IoutRef*VoltageOutput)>>10;
    }

DoCurrentControl();

ControlOutput=CurrentOutput;
}
else      // Cevrim durduruluyor baslatiliyor
{
    CurrentIntegral =0;
    VoltageIntegral =0;
    CurrentOutput =0;
    VoltageOutput =0;
}

```

```

    ControlOutput =0;
}

if(ControlOutput>600)
    ControlOutput=600;

analogWrite(9, ControlOutput);
// Bu kisimda cikis akiminin PI hesaplamasi yapiliyor
void DoCurrentControl(void)
{
/*-----Current PI Control Algorithm-----*/
// Calculate Error Value
CurrentError =CurrentRef-Iout_Value;

if(CurrentError>20) CurrentError=20;
if(CurrentError<-20) CurrentError=-20;

// P-D degerleri hesaplaniyor
CurrentPD_Term = (CurrentError*Ki_P) + (CurrentError-
CurrentPreError)*Ki_D;

CurrentI_Term = CurrentError*Ki_I;

// I degeri hesaplaniyor
CurrentIntegral = CurrentIntegral + CurrentI_Term;

if(CurrentIntegral>=CurrentIntegralMax) CurrentIntegral=CurrentIntegralMax;
if(CurrentIntegral<=CurrentIntegralMin) CurrentIntegral=CurrentIntegralMin;

CurrentOutput = CurrentPD_Term + CurrentIntegral;

```

```

if(CurrentOutput>=CurrentOutputMax) CurrentOutput=CurrentOutputMax;
if(CurrentOutput<=CurrentOutputMin) CurrentOutput=CurrentOutputMin;

CurrentOutput=CurrentOutput>>10;

CurrentPreError = CurrentError;
}

// Bu kisimda cikis akiminin PI hesaplamasi yapiliyor
void DoVoltageControl(void)
{
/*-----Voltage PI Control Algorithm-----*/
// Calculate Error Value
VoltageError =VoutRef-Vout_Value;

if(VoltageError>20) VoltageError=20;
if(VoltageError<-20) VoltageError=-20;

// P-D degerleri hesaplaniyor
VoltagePD_Term = (VoltageError*Kv_P) + (VoltageError-
VoltagePreError)*Kv_D;
VoltageI_Term = VoltageError*Kv_I;

// I degeri hesaplaniyor
VoltageIntegral = VoltageIntegral + VoltageI_Term;

if(VoltageIntegral>=VoltageIntegralMax)
VoltageIntegral=VoltageIntegralMax;
if(VoltageIntegral<=VoltageIntegralMin) VoltageIntegral=VoltageIntegralMin;
VoltageOutput = VoltagePD_Term + VoltageIntegral;
if(VoltageOutput>=VoltageOutputMax) VoltageOutput=VoltageOutputMax;

```

```

if(VoltageOutput<=VoltageOutputMin) VoltageOutput=VoltageOutputMin;

VoltageOutput = VoltageOutput>>10;

VoltagePreError = VoltageError;
}
// UART'tan gönderilecek bilgiler buradan gonderiliyor
void UART_Send(void)
{
    Serial.print("Vin = ");      Serial.print((float)Vin_Average/10,1);
Serial.print("V");
    Serial.print("\t Vout = ");   Serial.print((float)Vout_Average/10,1);
Serial.print("V");
    Serial.print("\t Iout = ");   Serial.print((float)Iout_Average/100,2);
Serial.print("A");
    Serial.print("\t Pout = ");
Serial.print((float)(Vout_Average*Iout_Average)/1000,1);  Serial.print("W");
    Serial.print("\t Duty = ");   Serial.print(ControlOutput);
    Serial.print("\t Start/Stop = ");  Serial.println(StartStop);
    StartStop=1;
}
void loop()
{
    StartStop = 1;

    if(Uart_Tick_Counter>499) {UART_Send(); Uart_Tick_Counter=0;
digitalWrite(LedPin, !digitalRead(LedPin)); } // Her 500ms'de bir bu kisma giriliyor
    if(Meas_Tick_Counter>0) {GetMeasurements(); Meas_Tick_Counter=0; }
// Her 1ms'de bir bu kisma giriliyor
}

```

```
// Timer0 kesmesi bu kisima geliyor, kesme frekansi 1kHz
ISR(TIMER0_COMPA_vect)
{
  Uart_Tick_Counter++;
  Meas_Tick_Counter++;

  interrupts();
}
```

Ek 6. Referans deęerini 3V yapan Arduino kodları.

```
void PI_Control(void)
{
  if(StartStop==1) // Cevrim baslatiliyor
  {
    Zaman=Zaman+1;
    if(Zaman<45000)
    {
      VoutRef=50;
    }
    else
    {
      VoutRef=30
    }
  }
}
```

ÖZ GEÇMİŞ

1992 yılında Van'da doğdu. İlk, orta ve lise öğrenimini Van'da tamamladı. 2010 yılında öğrenime başladığı Van Yüzüncü Yıl Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü'nden 2014 yılında mezun oldu. 2018 yılında çalışmaya başladığı Van Yüzüncü Yıl Üniversitesi Erciş Meslek Yüksek Okulu'nda öğretim görevlisi olarak görev yapmaktadır.



T.C
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORJİNALLİK RAPORU

Tarih: 20/01/2020

Tez Başlığı / Konusu: Sayısal Kontrolcü Kullanarak Buck Konvertör Tasarımı

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 69 sayfalık kısmına ilişkin, 20/01/2020 tarihinde şahsım/tez danışmanım tarafından Turmitin intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezim benzerlik oranı % 4 (dört) tür.

Uygulanan filtreler aşağıda verilmiştir:

- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimedenden daha az örtüşme içeren metin kısımları hariç (Limit inatch size to 7 words)

Van Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

20.01.2020
Faruk ve İntza

Adı Soyadı: Merve TÜRK TAM
Öğrenci No: 149101004
Anabilim Dalı: Elektrik-Elektronik Mühendisliği
Programı: Elektrik-Elektronik Mühendisliği Bölümü
Statüsü: Y. Lisans Doktora

DANIŞMAN ONAYI
UYGUNDUR



Doç. Dr. Mehmet Nuri ALMALI

FEN BİLİMLERİ ENSTİTÜSÜ
ONAY
UYGUNDUR

