

T.C.
KIRKLARELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**AKIM MODLU BUCK TİPİ DÖNÜŞTÜRÜCÜ İÇİN PARAMETRELERİ YAK İLE
OPTİMİZE EDİLMİŞ MIT KURALINA DAYALI MRAC UYGULAMASI**

YÜKSEK LİSANS TEZİ

Halil LEYLAK

TEMMUZ - 2021

T.C.
KIRKLARELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

**AKIM MODLU BUCK TİPİ DÖNÜŞTÜRÜCÜ İÇİN PARAMETRELERİ YAK
İLE OPTİMİZE EDİLMİŞ MIT KURALINA DAYALI MRAC UYGULAMASI**

YÜKSEK LİSANS TEZİ

Halil LEYLAK

Tez Danışmanı: Dr. Öğr. Üyesi Aydın MÜHÜRÇÜ

TEMMUZ – 2021

ETİK BEYAN

Kırkırelı Üniversitesi Fen Bilimleri Enstitüsü Tez ve Proje Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduđum bilgileri, verileri ve dokümanları, deđişik sonuç verebilecek şekilde araştırma araç gereçleri kullanmadan, işlem veya kayıt sonuçlarını deđiřtirmeden akademik ve etik kurallar çerçevesinde elde ettiđimi, bilimsel etik ve ahlak kurallarına uygun olarak sunduđumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiđimi, bu tezde sunduđum çalışmanın özgün olduđunu bildirir, aksi bir durumda aleyhime dođabilecek tüm hak kayıplarını kabullendiđimi beyan ederim.

İmza
Halil LEYLAK
04/07/2021

ÖZET

AKIM MODLU BUCK TİPİ DÖNÜŞTÜRÜCÜ İÇİN PARAMETRELERİ YAK İLE OPTİMİZE EDİLMİŞ MIT KURALINA DAYALI MRAC UYGULAMASI

Halil LEYLAK

Yüksek Lisans Tezi

Kırklareli Üniversitesi

Fen Bilimleri Enstitüsü

Danışman: Dr. Öğr. Üyesi Aydın MÜHÜRÇÜ

Temmuz 2021,47 sayfa

Bu çalışmada Buck tipi dönüştürücünün MIT Kurallı MRAC ile kontrolü yapılmıştır ve kontrol parametreleri Yapay Arı Kolonisi algoritmasıyla optimize edilmiştir. Sistemin sürekli zaman matematiksel ifadesi ayrık zamana çevrilmiştir. Ayrık zamandaki sistem Matlab ortamında matematiksel algoritmaya dönüştürülmüştür. Devre; çıkış akımı, giriş gerilimi, yük akımı, örnekleme süresi vb. parametrelerine göre Yapay Arı Kolonisi algoritmasıyla optimizasyonu yapılmıştır. Optimize olan sistem Simulink ortamında yapılan simülasyonda zamanla değişen referans akımına göre devre çıkışına en optimum hızda tepki vermekte ve değer aşımı olmadan referans akıma en yakın değerde çıkış vermektedir.

Anahtar Kelimeler: ABC Algoritması, MRAC, MIT Kuralı, Adaptif Kontrol.

ABSTRACT

MIT RULE BASED MRAC APPLICATION WITH OPTIMIZED PARAMETERS FOR BUCK TYPE CONVERTER WITH CURRENT MODE

Halil LEYLAK

MSc Thesis

Kirklareli University

Graduate School of Natural and Applied Sciences

Supervisor: Asst. Prof. Aydın MÜHÜRÇÜ

Temmuz 2021, 47 pages

In this study, the Buck type converter was controlled with MIT Ruled MRAC and control parameters optimized with Artificial Bee Colony algorithm. The continuous time mathematical expression of the system is converted to discrete time. Circuit optimization was made with the Artificial Bee Colony algorithm according to output current, input voltage, load current, sampling time, etc. parameters. The optimized system responds to the circuit output at the most optimum speed according to the reference current that changes over time in the simulation made in the Simulink environment and gives the output at the closest value to the reference current without exceeding the value.

Keywords: ABC Algorithm, MRAC, MIT Rule, Adaptive Control.

TEŐEKKÜR

Tez alıŐmamn her aŐamasında desteęini esirgemeyen danıŐman hocam Dr. Öğr. Üyesi Aydın MÜHÜRÇÜ'ye;

Lisans ve Lisansüstü eğitimimde desteęi ve katkısı olan Kırklareli Üniversitesi Elektrik Elektronik Mühendislięi öğretim üyelerine;

Lisansüstü eğitimim boyunca sorularımı hiç cevapsız bırakmayan ve manevi destek veren dostum ArŐ. Gör. Alperen KIZILAY'a;

Hayatımın her anında bana destek ve yardımcı olan aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

| | |
|--|------|
| ÖZET..... | iv |
| ABSTRACT | v |
| TEŞEKKÜR..... | vi |
| İÇİNDEKİLER..... | vii |
| RESİMLERİN LİSTESİ..... | viii |
| SİMGELER VE KISALTMALAR..... | xiii |
| 1. GİRİŞ..... | 1 |
| 2. KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI..... | 3 |
| 2.1. DC-DC DÖNÜŞTÜRÜCÜLER | 3 |
| 2.1.1. Buck tipi DC-DC dönüştürücü..... | 3 |
| 2.1.2. Buck dönüştürücü devre denklemleri ve simulink simülasyonu..... | 6 |
| 2.2. ADAPTİF KONTROL..... | 9 |
| 2.2.1. MRAC..... | 10 |
| 2.2.2. MIT kuralı..... | 11 |
| 2.2.3. MIT kuralı MRAC tasarımı | 13 |
| 2.3. OPTİMİZASYON | 16 |
| 2.3.1. Yapay arı kolonisi algoritması | 17 |
| 3. MATERYAL VE DENEYSEL YÖNTEM..... | 21 |
| 4. BULGULAR VE TARTIŞMA | 43 |
| 5. SONUÇ VE ÖNERİLER | 47 |
| KAYNAKLAR..... | 48 |
| ÖZGEÇMİŞ..... | 49 |

RESİMLERİN LİSTESİ

| | |
|---|----|
| Resim 2.1 Buck tipi dönüştürücü topolojisi | 4 |
| Resim 2.2 Devre akımlarının durumu | 4 |
| Resim 2.3 Simulink buck devresi | 6 |
| Resim 2.4 Buck dönüştürücü yük gerilimi grafiği | 7 |
| Resim 2.5 Buck dönüştürücü devresi | 7 |
| Resim 2.6 Buck devre denklemleri ve devresinin simulink çizimleri | 8 |
| Resim 2.7 Devre denklemleri ve devrenin çıkış gerilimi kıyaslaması | 9 |
| Resim 2.8 devre denklemleri ve devrenin çıkış gerilimi kıyaslaması (Yakınlaştırılmış). 9 | |
| Resim 2.9 MRAC kontrolün blok diyagramı | 10 |
| Resim 2.10 MIT kuralı blok diyagramı | 12 |
| Resim 2.11 MIT kuralı uygulanmış mrac sistem diyagramı | 15 |
| Resim 2.12 Optimizasyon algoritma sınıfları | 17 |
| Resim 2.13 Arıların dansı | 17 |
| Resim 2.14 Arıların yön tayini | 18 |
| Resim 3.1 Örnek denklem simulink blok diyagramı | 22 |
| Resim 3.2 Devre denklemleri simulink blok diyagram | 23 |
| Resim 3.3 Blok diyagram sonuçları ve devrenin sonuçlarının kıyaslanması | 24 |
| Resim 3.4 Blok diyagram sonuçları ve devrenin sonuçlarının kıyaslanması(Yakınlaştırılmış) | 24 |
| Resim 3.5 Simulink gecikme(delay) bloğu | 24 |
| Resim 3.6 Örnek sistem matlab kodu | 25 |
| Resim 3.7 PWM sinyali oluşturulması | 26 |
| Resim 3.8 Buck dönüştürücü matlab algoritması | 26 |
| Resim 3.9 Buck devre ve matlab yazılım sonuç kıyaslaması(yakınlaştırılmış) | 27 |
| Resim 3.10 MIT kuralıyla tasarlanmış mrac kontrollü buck dönüştürücü | 28 |
| Resim 3.11 Kontrollü ve kontrolsüz buck devresi sonuçları | 29 |
| Resim 3.12 MIT kurallı mrac kontrollü buck devresi | 30 |
| Resim 3.13 Ayrık zaman ve sürekli zaman mrac karşılaştırması | 31 |
| Resim 3.14 MIT kuralları MRAC kontrollü buck dönüştürücü sistem parametreleri.... | 32 |

| | |
|---|----|
| Resim 3.15 Kontrol mekanizması kısım 1 | 32 |
| Resim 3.16 Kontrol mekanizması kısım 2 | 33 |
| Resim 3.17 Optimizasyon akış diyagramı | 34 |
| Resim 3.18 YAK algoritması sonucu iterasyon-hata toplamı grafiği | 35 |
| Resim 3.19 simulasyon sonuçları | 36 |
| Resim 3.20 Referans akım ve yük akımı karşılaştırması..... | 36 |
| Resim 3.21 Sistemin değişen referans akımlara tepkisi | 37 |
| Resim 3.22 Çıkış akımı ve referans model kıyaslaması..... | 37 |
| Resim 2.23 Çıkış akımı ve referans model kıyaslaması(yakınlaştırılmış) | 38 |
| Resim 2.24 Yük akımı hatası(%). | 38 |
| Resim 3.25 Düzeltme katsayısı +100 ile -100 aralığında hata çıktısı | 39 |
| Resim 3.26 Yeni katsayılarla yük çıkış değerleri | 40 |
| Resim 3.27 Sistem çıkış akımı ve referans model kıyaslaması yeni katsayılar | 40 |
| Resim 3.27 Sistem çıkış akımı ve referans model kıyaslaması yeni katsayılar(yakınlaştırılmış) | 41 |
| Resim 3.28 Yeni katsayılı sistem çıkış akım hatası(%). | 41 |
| Resim 4.1 Zamanla değişen referansa buck dönüştürücü tepkisi | 43 |
| Resim 4.2 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları 1 | 44 |
| Resim 4.3 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları(yakınlaştırılmış) 1 | 44 |
| Resim 4.4 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları 2 | 44 |
| Resim 4.5 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları(yakınlaştırılmış) 2 | 45 |

SİMGELER VE KISALTMALAR

Simgeler

Açıklamalar

| | |
|------------|-------------------------|
| s | Saniye |
| A | Amper |
| V | Volt |
| ms | Milisaniye |
| I_L | Endüktans Akımı |
| I_C | Kondansatör Akımı |
| $I_{Ryük}$ | Yük Akımı |
| V_C | Kondansatör Gerilimi |
| V_{out} | Yük Gerilimi |
| γ | MRAC Düzeltme Katsayısı |

Kısaltmalar

Açıklamalar

| | |
|---------------|---|
| DC | Direct Current |
| MRAC | Model Reference Adaptive Control |
| DA | Doğru Akım |
| MIT | Massachusetts Institute Of Technology |
| YAK | Yapay Arı Kolonisi |
| ABC | Artificial Bee Colony |
| PWM | Pulse Width Modulation |
| EMI | Electromagnetic Interference |
| MOSFET | Metal Oxide Semiconductor Field Effect Transistor |
| PID | Proportional–İntegral–Derivative |
| IAE | Integral Absolute Error |
| ISE | Integral Square Error |
| ITSE | Integral Time Square Error |
| ITAE | Integral Time Absolute Error |

1. GİRİŞ

Dijital teknoloji insanlar arasından yaygınlaştıkça sektör de büyümekte ve gelişmektedir. Gelişen elektronik endüstrisiyle beraber gereklilikler de değişmeye başlamıştır. Günümüzde yaygın olarak kullanılan cep telefonu, bilgisayar vb. cihazların gelişimiyle anahtarlama DA-DA dönüştürücülerin de önemi artmıştır.

“Anahtarlama DA-DA dönüştürücülerin kayıpları çok az olduğu için ve fiziksel boyut olarak küçültemeye uygun yapıda olduklarından elektronik sektöründe çokça kullanılmaktadır (Rasid, 2001, s. 211-215)”. Temel olarak 3 tipi vardır, Buck, Boost ve Buck-Boost. Buck dönüştürücü giriş gerilimini düşürürken, Boost dönüştürücü yükseltmeye yarar. Buck-Boost dönüştürücü ise giriş gerilimini hem düşürüp hem de yükseltme yapmaya olanak tanır.

Anahtarlama DA-DA dönüştürücüler regülasyon yaparken, anahtarlama başladığı ilk anda giriş geriliminin istenilen çıkış gerilimine regülasyonu sırasında bir oturma zamanı ve aşım değeri vardır. Bu oturma zamanı ve aşım değeri sistemde kullanılan elemanlara göre değişiklik gösterir. Oturma zamanı ve aşım değerini kontrol edebilmek için kapalı çevrim denilen yöntemler kullanılmaktadır. Kapalı çevrim yöntemlerinden biri de MIT kuralıyla tasarlanmış MRAC'dır.

MRAC, adaptif kontrol yöntemlerinden birisidir. Adaptif kontrol doğrusal olmayan sistemlerin değişkenleri bilinmediği zaman veya çok değişiklik gösterdiği zaman kullanılan bir kontrol yöntemidir. Sistemin dinamik yapısı değiştiğinde veya değişkenlerin bilinmediği zaman gerçek zamanda otomatik kontrole imkan sağlar.

MRAC, kontrol edilecek devrenin çıkış cevabını girişinden bir referans değer olarak kontrol eden bir mekanizmadır. Bu yapıya sahip olmasının en büyük avantajı kontrol edilen sistemin matematiksel modeli bilmeden kontrole imkan sağlamasıdır. En büyük dezavantajı ise referans değer sistem çalışmadan önce girilmesi gerektiğinden kontrol edilecek sistemin dinamiklerinin bilinmesi gereklidir.

“MIT Kuralı, Massachusetts Institute of Technology tarafından 1960’da hava araçlarının otomatik pilot özelliği için geliştirilen bir kuraldır (Barghandan ve DaeiFarshchi, 2018, s. 7-9)”. Temel olarak “sistem çıkışını model referansı ile karşılaştırarak hatayı alır ve bu hatayı sistemi içindeki uygun parametreleri ayarlamak için kullanır. Amacı çıkışta hesaplanan hatayı sifıra yakınsatmaktır (Barghandan ve DaeiFarshchi, 2018, s. 7-9)”.

MIT kuralıyla tasarlanmış MRAC sisteminde iki parametre mevcuttur. Bu parametreler kontrol edilecek devre elemanlarına uygun seçilmelidir ki sistem çalışırken kararsızlığa gitmesi engellensin. En uygun değerleri bulmak amacıyla sistem bir optimizasyon algoritmasında çalıştırılıp değerler tespit etmek mümkündür. Bu çalışmada sistem için en uygun değerlerin bulunmasında Yapay Arı Kolonisi algoritması kullanılmıştır.

İnsanlık tarih boyunca karşılaştıkları sorunları çözmek için doğada gözlemledikleri olayları taklit etmişlerdir. Bu taklitlerden biri de Yapay Arı Kolonisi algoritmasıdır. “Artificial Bee Colony(Yapay Arı Kolonisi) algoritması arıların doğada yiyecek bulmasını olayının gözlemlenerek, bu olayı matematiksel olarak ifade edilmesini esas alır (Karaboğa, 2005)”. Bu matematik ifadeleri bir algoritmaya çevrilebilir. Temel olarak alt ve üst sınır değerleri belli, rastgele oluşturulmuş bir dizideki değerleri sistemde deneyerek en az hataya sahip olan değer bulunması şeklinde ifade edilebilir. Sistem için en optimum değer bu şekilde bulunmuş olur.

Bu çalışmada Buck tipi bir DA-DA dönüştürücünün MIT kuralıyla tasarlanmış MRAC kontrolü üzerine çalışılmıştır. MRAC içindeki düzeltme katsayı parametreleri YAK algoritmasıyla bulunmuştur. Bu sayede dönüştürücünün oturma zamanı ve aşım değeri optimum seviyede tutulmuştur. Bunun yanında sürekli zamanda değişen referans değerine de devrenin tepkisi optimum seviyede olmaktadır.

2. KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI

2.1. DC-DC Dönüştürücüler

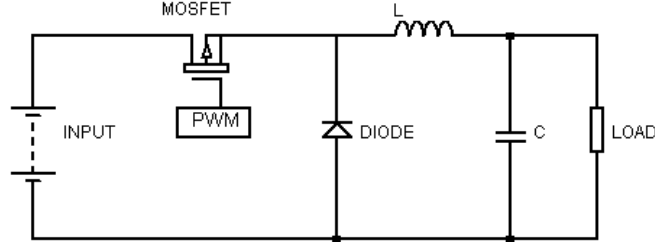
Modern elektronik sistemleri yüksek kalite, küçük boyut, devamlılık, verimliliğe ihtiyaç duyar. Cihaza güç sağlamak adına akım veya gerilimi regüle etmeye yarayan doğrusal güç çeviricileri istenilen sonucu verse de ihtiyaç duyulan bu özelliklerin birçoğunu karşılayamaz. “Bunun sebebi 50-60Hz frekansa sahip transformatör ve filtre kullanmalarından kaynaklı güç yoğunluklarının düşük olması ve genel olarak yüksek giriş gücünü düşük çıkış gücüne çevirmelerindedir (Rasid, 2001, s. 211-215)”.

Günümüzde yüksek güç gerektiren sistemlerde anahtarlama regülasyonu devreleri kullanılmaktadır. Anahtarlama regülatörler yarı-iletken elemanı iletim ve kesim durumunda kullanır. Çünkü bu 2 durumda güç kaybı düşüktür. Anahtarlama regülatörler çevrim yaparken yüksek verimliliğine sahip olabilirler. Ek olarak yüksek anahtarlama frekansın çalışabildiklerinden daha küçük boyutlu transformatör, kapasitör ve endüktans ile kullanılabilirler. Bu çalışmada son 30 yıldır çok popüler olan ve hemen her güç seviyesinde kullanılabilen PWM kontrollü Buck tipi DA-DA çeviricisi kullanılmıştır. PWM kontrollü çevirilerin avantajları, yüksek verimliliğe sahip olmaları, sabit frekans ile yönetilmeleri, nispeten kolay kontrol edilmeleri ve az elemana sahip yapılarıdır. “Dezavantajları ise açma-kapama işleminde yarı-iletken kaybı olmasıdır ki bu sebeple sadece birkaç yüz kilohertz frekansa kadar anahtarlama yapılabilir, aksi halde kayıplar artar, ve PWM için oluşturulan üçgen sinyal EMI(Elektromanyetik Girişim) üretir (Rasid, 2001, s. 211-215)”.

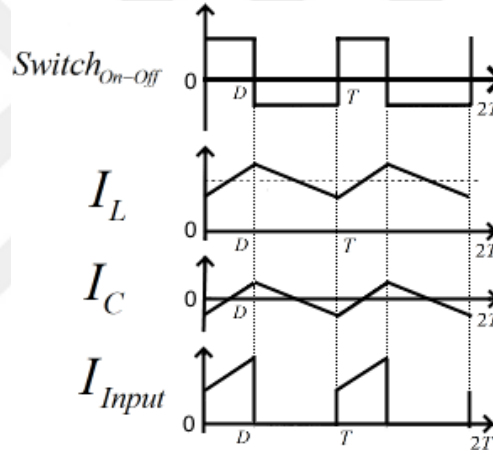
2.1.1. Buck tipi DC-DC dönüştürücü

Buck tipi dönüştürücüler endüstriyel uygulamalarda yaygın olarak kullanılan dönüştürücü tipidir. Resim 2.1’de Buck Dönüştürücü topolojisi görülmektedir. Devredeki MOSFET’in iletme girmesiyle giriş gerilimi endüktansa ve çıkışa uygulanır. Endüktans akımı artmaya ve kondansatör şarj olmaya başlar. MOSFET’in iletimden çıkmasıyla ise diyot iletme girer ve endüktansta depolanan enerji çıkışa aktarılır. Çıkışa paralel kondansatör ile çıkış filtrelenir ve sabit DC gerilimi elde edilir. İki çalışma şekli vardır;

Sürekli Akım Modu (Continuous Current Mode) ve Kesikli Akım Modu(Discontinuous Current Mode). Kararlı hal için, sürekli akım modunda bobin içinden devamlı olarak akım geçer. Kesikli akım modunda ise bobin akımı sıfırdan başlar, tepe değerine kadar çıkar ve periyot bitmeden sıfıra düşer. Ripple(Dalgalanma) gerilimi küçük olan dönüştürücü iyi tasarlanmış bir dönüştürücüdür.



Resim 2.1 Buck tipi dönüştürücü topolojisi



Resim 2.2 Devre akımlarının durumu

$$V_L(t) = L \frac{di_L}{dt} \quad (2.1)$$

Eş. 2.1'de bobin üzerinde depolanan gerilim hesaplanabilir. Eş. 2.2 ise bobin üzerinden geçen akım ifadesidir. T_{sw} anahtarlama periyodu olmak üzere,

$$i_L(T_{sw})i_L(0) = \frac{1}{L} \int_0^{T_{sw}} V_L(t)dt \quad (2.2)$$

$$A = \frac{1}{L} \int_0^{T_{sw}} V_L(t) dt = D(V_G - V_0) + D'(-V_0) \quad (2.3)$$

V_L alanının altında kalan alan bir anahtarlama periyodu boyunca sıfır olduğundan Eş. 2.3 eşitliği kurulabilir. Bu eşitlikte D , PWM doluluk oranını temsil etmektedir. V_G , giriş gerilimini ve V_0 ise yük gerilimini temsil eder. $D' = 1 - D$ ile ifade edilmektedir.

$$V_L = \frac{A}{T_{sw}} = D(V_G - V_0) + D'(-V_0) \quad (2.4)$$

$$V_0 = DV_G \quad (2.5)$$

Eş. 2.4'te $D' + D = 1$ eşitliği kullanılırsa ortaya Eş. 2.5 çıkar. Kapasite gerilimi ile kapasite akımı ilişkisi ise Eş. (2.6) gibi ifade edilir.

$$i_C(t) = C \frac{dV_C}{dt} \quad (2.6)$$

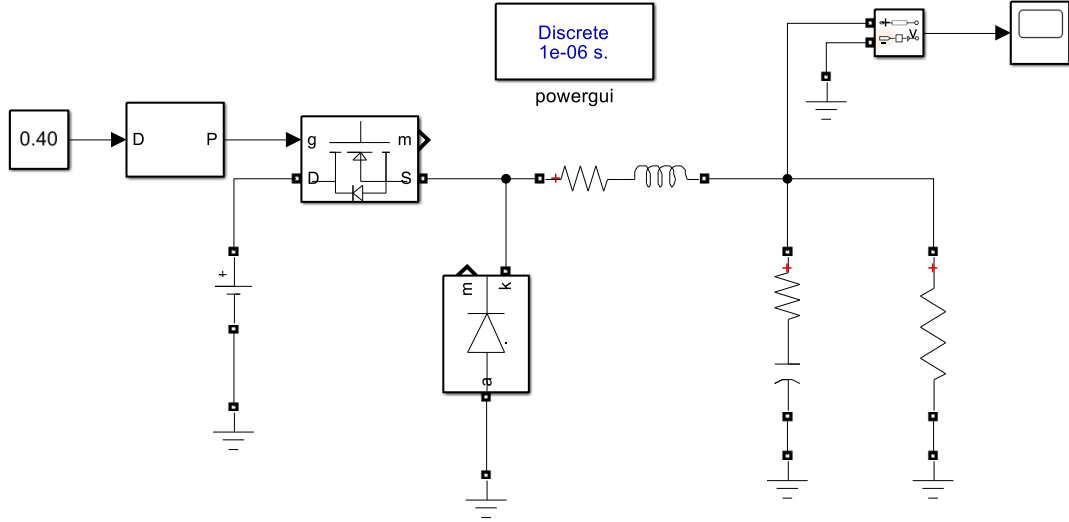
PWM doluluk oranını temsil Kapasitör gerilimi bir anahtarlama periyodundan sonra Eş. (2.7) ile ifade edilir.

$$V_C(T_{sw}) - V_C(0) = \frac{1}{C} \int_0^{T_{sw}} i_C(t) dt \quad (2.7)$$

Bir anahtarlama frekansı bittiğinde kalıcı durumda gerilim değişimi sıfırdır. Bu sebeple Eş. (2.7)'nin sol tarafı sıfıra eşit olur. Bu durum kapasitörün DC gerilimi sıfır demektir. Eş. (2.8) sayesinde ise kapasitör akımı hesaplanabilir.

$$i_C = \frac{1}{T_{sw}} \int_0^{T_{sw}} i_C(t) dt = 0 \quad (2.8)$$

2.1.2. Buck dönüştürücü devre denklemleri ve simulink simülasyonu



Resim 2.3 Simulink buck devresi

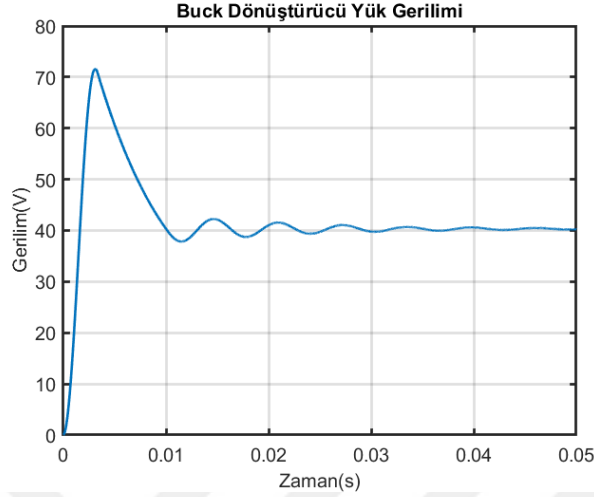
Resim 2.3’de görülen Buck devresi Simulink simülasyon programında kurulmuştur. Tablo 2.1’de devre elemanlarının parametreleri görülmektedir.

| Eleman | Değer |
|----------------|---------------|
| L | 1mH |
| C | 1mF |
| R_c | 0.01 Ω |
| R_L | 0.01 Ω |
| R_{Load} | 10 Ω |
| Frekans | 10kHz |
| PWM Doluluk | %40 |
| Giriş Gerilimi | 100V |

Tablo 2.1 Buck Devresi Eleman Değerleri

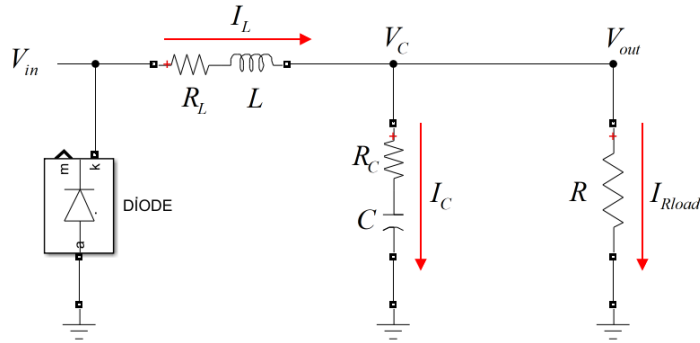
Gerçek bir Buck devresi elemanlarında iç direnç değeri mevcuttur. Bu değerler devrede kayıplara sebep olmaktadır. Simülasyonun olabildiğince gerçek uygulamaya benzemesi

için bu iç direnç değerleri devreye ve devre denklemlerine eklenmiştir. Tablo 2.1’de görülen değerler Resim 2.3’deki devre elemanlarında kullanılmıştır. Resim 2.4’te bu simülasyonun sonuçları görülmektedir.



Resim 2.4 Buck dönüştürücü yük gerilimi grafiği

Resim 2.4’te görüldüğü üzere Buck tipi dönüştürücü ilk çalışma anında yaklaşık 30V’luk hata yaparak %75’lik hata değerine ulaşmaktadır. Bu durum 10ms içinde toparlanmaktadır lakin hassas devreler için bu durum kabul edilemez bir hata miktarıdır.



Resim 2.5 Buck dönüştürücü devresi

Resim 2.5’de verilen devrede V_{in} , giriş gerilimidir. MOSFET, PWM sinyaline göre devreye giriş gerilimini veya sıfır değeri uygulanır. Devre şemasında I_L , I_C ve I_{Rload} ile ifade edilen akım değerleri kırmızı ile belirtilen okların referansında S-Domeni’de,

$$X_L(s) = sL \quad (2.9)$$

$$X_C(s) = \frac{1}{sC} \quad (2.10)$$

olmak üzere devre denklemleri şu şekilde çıkartılır,

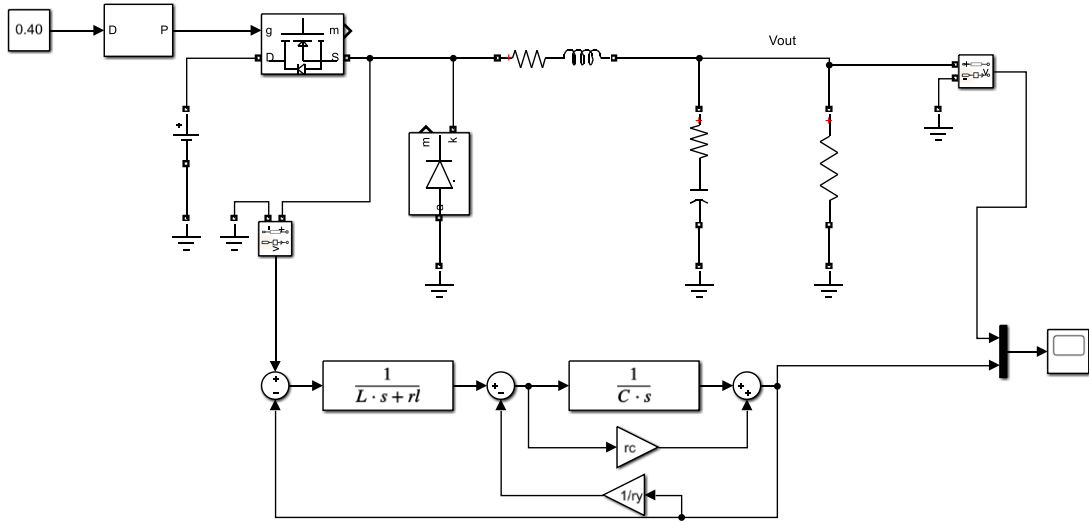
$$I_L = (V_{in} - V_{out}) \frac{1}{R_L + LS} \quad (2.11)$$

$$I_C = I_L - I_{Rload} \quad (2.12)$$

$$V_C = I_C(R_C + \frac{1}{CS}) \quad (2.13)$$

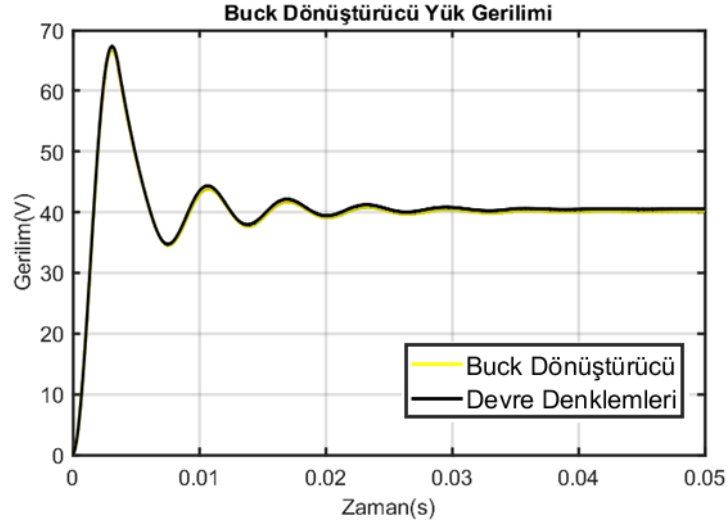
$$I_{Rload} = V_C \frac{1}{R} = V_{out} \frac{1}{R} \quad (2.14)$$

Eş. (2.11), Eş. (2.12), Eş. (2.13) ve Eş. (2.14) içinde farklı elemanların gerilimlerini ve akımlarını da bulmayı sağlayan denklemlerdir. Bu denklemleri Simulink ortamında blok diyagram olarak tasarlanıp sonuçlar paylaşılmıştır.

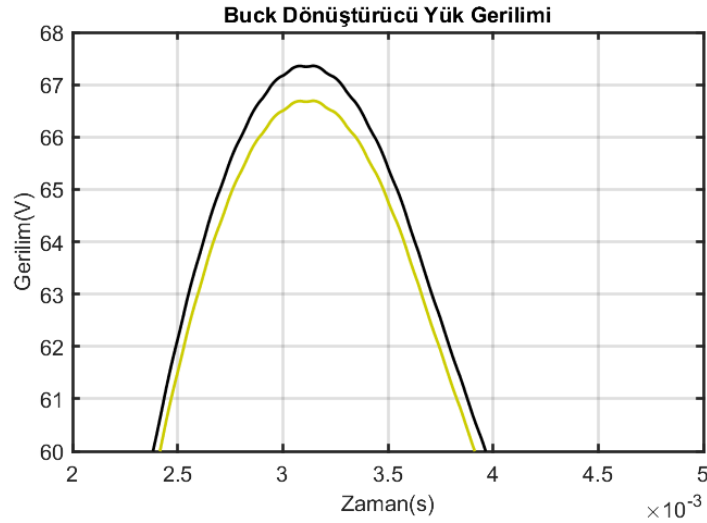


Resim 2.6 Buck devre denklemleri ve devresinin simulink çizimleri

Resim 2.7’de ve Resim 2.8’de görüldüğü üzere Eş. (2.11), Eş. (2.12), Eş. (2.13) ve Eş. (2.14) referansında blok diyagram oluşturulmuş diyagram ile devre elemanlarıyla oluşturulmuş Buck devresi sonuçları üst üste binmektedir. Yani denklemler ve devre aynı sonuçları ve aynı tepkileri vermektedir.



Resim 2.7 Devre denklemleri ve devrenin çıkış gerilimi kıyaslaması



Resim 2.8 devre denklemleri ve devrenin çıkış gerilimi kıyaslaması (Yakınlaştırılmış)

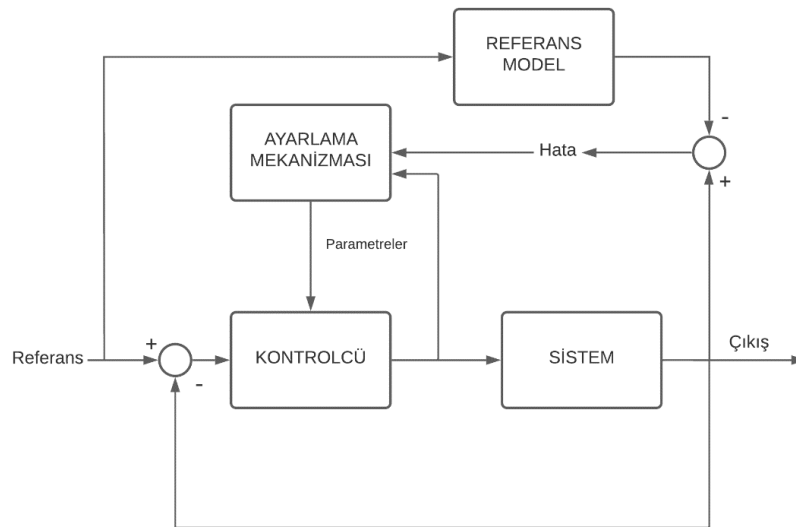
2.2. Adaptif Kontrol

“Endüstriyel uygulamaların yüksek performans vermesine yönelik yapılan kontrol sistemleri araştırmalarının sonucunda modern kontrol yöntemi olan adaptif kontrol bulundu” (Swarnkar, Jain ve Nema, 2010, s. 1-2). İlk kullanım amacı hava araçları olan bu kontrol yöntemi zamanla gelişerek yaygınlaştı. Adaptif kontrol sisteme verilen bir referansa göre sistem sonucunu ayarlamaya yaran bir yöntemdir. “Sistemi referans modele göre davranması için zorlar ve ayarlar” (Swarnkar ve diğerleri, 2010, s. 1-2). Adaptif kontrolcüler doğrusal olmayan sistemlerin parametreleri bilinmediği zaman veya çok değişkenlik gösterdiği zaman tercih edilen bir kontrol yöntemidir.

Adaptif kontrolün en büyük avantajı gürültüsü yüksek ortamlarda sistemin dinamiğini en az derecede bu durumdan etkilenecek şekilde kontrolünü sağlar. Bunun sebebi ise aslında bu sistemin dezavantajıdır. Sistemin dezavantajı sisteme ait model bilinmese bile sistemin dinamiği bilinmelidir ki normalde sistemin vermesi gereken tepki dışında bir olay gerçekleşmesin. Adaptif kontrolün 2 çevrimi vardır. Birincisi PID kontrolden bilinen kapalı çevrim. İkincisi ise parametre ayarlama çevrimidir. MRAC’da bir adaptif kontrol yöntemidir.

2.2.1. MRAC

“Kontrol sistemleri, uygulamaları ve işlemleri regüle etmek ve kontrol etmek için olan sistemlerdir” (Swarnkar ve diğerleri, 2010, s. 1-2). “Adaptif kontrol ise modern kontrol sistemleri arasında en iyi sonuç ve performans veren kontrol yöntemidir” (Barghandan ve DaeiFarshchi, 2018, s. 7-9). “MRAC, en çok kullanılan adaptif kontrol yaklaşımıdır. MRAC istenilen performansı vermek adına sistemde kullanılan bir referans modeli baz alır” (Garikayi, Nyanga, Mushiri, Mhlanga, Muzinda ve Mutangi ,2013). Detaylı olarak şu şekilde ifade edilebilir; bir model referans, referans değeri giriş olarak alır. Model referans çıkışında sistem dinamiğinin sonucunda vermesi gereken çıkış elde edilir. Bu değer sistemin çıkışından beklenen değerdir. Daha sonra bu çıkış değeri sistemin gerçek zaman çıkışıyla işleme alınır ve hata miktarı bulunur. Bu hata, ayarlama mekanizmasına girer ve normal kapalı çevrim ile kıyaslaması yapılır. Parametreler bu şekilde ayarlanır. MRAC sistemini MIT kuralıyla modifiye ederek daha performanslı hale getirmek mümkündür.



Resim 2.9 MRAC kontrolün blok diyagramı

2.2.2. MIT kuralı

MIT Kuralı 1960'da Massachusetts Institute of Technology tarafından tasarlanmıştır. Otomatik pilot sistemi kontrolü için tasarlanan bu sistem “MRAC kontrolü kullanılacak sistemler için de uygundur” (Barghandan ve DaeiFarshchi, 2018, s. 7-9). $y_m(t)$ referans model, $y(t)$ kontrol edilen sistem ve $e(t)$ hata olmak üzere (Barghandan ve DaeiFarshchi, 2018, s. 7-9),

$$e(t) = y(t) - y_m(t) \quad (2.15)$$

e , model referansın çıkışı ile kontrol edilen sistemin çıkışının farkıdır yani sistem hatasıdır.

Maliyet fonksiyonu şu şekilde tanımlanırsa,

$$J(\theta) = \frac{e^2}{2} \quad (2.16)$$

θ ayarlama parametresi olmak üzere, bu parametre hata sıfır olana kadar ayarlanır. “Bu sebeple θ parametresini negatif yönlü değiştirerek J ayarlanır” (Barghandan ve DaeiFarshchi, 2018, s. 7-9).

$$\frac{d\theta}{dt} = \gamma \frac{\partial J}{\partial \theta} \quad (2.17)$$

Eş. (2.15)'e göre,

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2.18)$$

“ $\frac{\partial e}{\partial \theta}$ türev ifadesine sistemin hassaslık türevi denir” (Swarnkar ve diğerleri, 2010, s. 1-2).

Bu şunu gösterir, hata θ ayarlama parametresine bağlıdır, $J(\theta)$ maliyet fonksiyonu sıfıra yakınsar.” γ ise kontrol sisteminin düzeltme kazancını temsil eden bir değerdir” (Barghandan ve DaeiFarshchi, 2018, s. 7-9).

Şöyle bir işlem varsayarsak; $KG(s)$ doğrusal transfer fonksiyonu olsun. K parametreleri bilinmesin ve $G(s)$ ikinci dereceden bilinen bir transfer fonksiyonu olsun. Amaç, K_0 bilinen bir parametre olmak üzere $G_m(s) = K_0 G(s)$ şeklinde oluşturulan bir referansı takip edecek bir kontrolcü tasarlamak olsun (Duvanov, Kudinov, Pashchenko ve Ponomarev, 2020).

Eş. (2.15)'ten yola çıkarak,

$$E(s) = KG(s)U(s) - K_0G(s)U_c(s) \quad (2.19)$$

u_c adaptif kontrol çıktısı, $u(t)$ kontrolcü çıktısı olmak üzere kontrol kuralının tanımı, (Duvanov ve diğerleri, 2020).

$$u_c u(t) = \theta^* \quad (2.20)$$

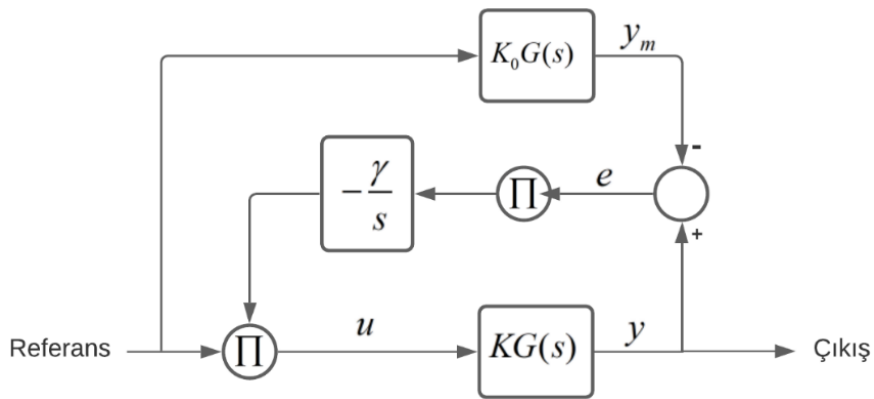
Eş. (2.19) ve Eş. (2.20)'ye göre ve kısmi diferansiyel göre (Duvanov ve diğerleri, 2020),

$$\frac{\partial E(s)}{\partial \theta} = KG(s)U_c(s) = \frac{K}{K_0} Y_m(s) \quad (2.21)$$

Eş. (2.18) ve Eş. (2.21) göz önüne alındığında Eş. (2.22) elde edilir (Duvanov ve diğerleri, 2020).

$$\frac{d\theta}{dt} = -\gamma e \frac{K}{K_0} y_m = -\gamma' e y_m \quad (2.22)$$

“Varsayımsal sistem göstermektedir ki kontrolü yapılacak sistem γ' adaptasyon kazanç değerine bağlıdır” (Barghandan ve DaeiFarshchi, 2018, s. 7-9). Eğer γ' çok büyük değerlerde alınır ya da bulunursa sistem bozulmaya doğru gider. Bu sebeple MIT Kurallı sistemlerde γ' katsayısı çok büyük öneme sahiptir. Resim 2.10'da MIT Kurallıyla tasarlanmış bir sistemin diyagramı görülmektedir.



Resim 2.10 MIT kuralı blok diyagramı

Resim 2.10'dan anlaşılacağı üzere düzeltme kazanç parametresi sistem döngüsünde önemli bir yere sahiptir. Bu derece önemli bir parametre ne kadar hassas seçilirse sistem o kadar güvenilir ve stabil olacaktır.

2.2.3. MIT kuralı MRAC tasarımı

$y_m(t)$ referans model, $y(t)$ kontrol edilen sistem ve $e(t)$ hata olmak üzere,

$$e(t) = y(t) - y_m(t) \quad (2.23)$$

“ e , model referansın çıkışı ile kontrol edilen sistemin çıkışının farkıdır yani sistemin hatasıdır.” (Barghandan ve DaeiFarshchi, 2018, s. 7-9).

Maliyet fonksiyonu şu şekilde tanımlanırsa,

$$J(\theta) = \frac{e^2}{2} \quad (2.24)$$

θ ayarlama parametresi olmak üzere, bu parametre hata sıfır olana kadar ayarlanır. Bu sebeple θ parametresini negatif yönlü değiştirerek J ayarlanır” (Barghandan ve DaeiFarshchi, 2018, s. 7-9).

$$\frac{d\theta}{dt} = \gamma \frac{\partial J}{\partial \theta} \quad (2.25)$$

Eş. (2.24)’e göre,

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta} \quad (2.26)$$

“ $\frac{\partial e}{\partial \theta}$ türev ifadesine sistemin hassaslık türevi denir.” (Swarnkar, Jain ve Nema, 2010, s. 1-2). Bu şunu gösterir, hata θ ayarlama parametresine bağlıdır, $J(\theta)$ maliyet fonksiyonu sifıra yakınsar. γ ise kontrol sisteminin düzeltme kazancını temsil eden bir değerdir (Barghandan ve DaeiFarshchi, 2018, s. 7-9).

MIT kuralı MRAC sistemine şu şekilde uygulanabilir;

Modeli Eş. (2.27)’deki gibi birinci derecen bir model düşünölsün.

$$\frac{dy}{dt} = -ay + bu \quad (2.27)$$

u , kontrol deęişkeni, y ise çıkış ve kapalı sistemin modeli Eş. (2.28)’daki gibi belirlenmiş olsun.

$$\frac{dy_m}{dt} = -a_m y_m + b_m u_c \quad (2.28)$$

$$u(t) = \theta_1 u_c(t) - \theta_2 y(t) \quad (2.29)$$

Sistem Eş. (2.29) ile kontrol edilirse kontrolcünün 2 parametresi olur ve onlarda şu şekilde seçilmiş olsun,

$$\theta_1 = \frac{b_m}{b} \quad (2.30)$$

$$\theta_2 = \frac{a_m - a}{b} \quad (2.31)$$

Bu sisteme MIT Kuralı'nı uygularsak.

$$e = y - y_m \quad (2.32)$$

Eş. (2.28) ve Eş. (2.29) denklemlerine göre Eş. (2.33) denklemi bulunur.

$$y = \frac{b\theta_1}{p + a + b\theta_2} u_c \quad (2.33)$$

$p = d/dt$ türev operatörüdür.

$$\frac{\partial e}{\partial \theta_1} = \frac{b}{p + a + b\theta_2} u_c \quad (2.34)$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{b}{p + a + b\theta_2} y \quad (2.35)$$

$$p + a + b\theta_2 \approx p + a_m \quad (2.36)$$

$$\frac{d\theta_1}{dt} = -\gamma \left(\frac{a_m}{p + a_m} u_c \right) e \quad (2.37)$$

$$\frac{d\theta_2}{dt} = -\gamma \left(\frac{a_m}{p + a_m} y \right) e \quad (2.38)$$

Eş. (2.34) ve Eş. (2.35) denklemlerine Eş. (2.36) yaklaşımı yapılırsa, Eş. (2.37) ve Eş. (2.38) denklemleri elde edilir. Bu denklemler ışığında $a = b = a_m = b_m = \tau$ eşitliğine göre çizilmiş blok diyagram Resim 2.11'de gösterilmektedir.

2.3. Optimizasyon

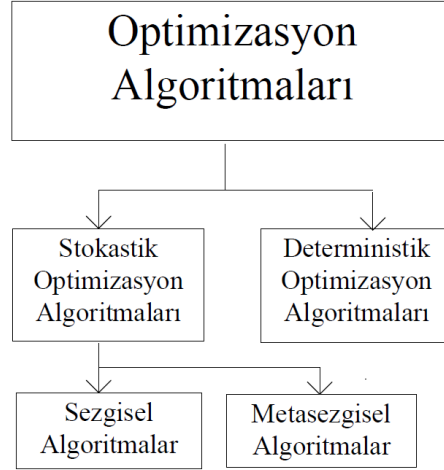
Gelişen teknolojiyle birlikte nüfusun artmasından dolayı dünya kaynaklarının verimli kullanılması neredeyse zorunlu hale geldi. Bu bağlamda bir sorunu çözerken ya da bir ihtiyacı karşılarken en uygun çözümden ziyade verimliliği en yüksek olan uygun çözüm arayışı başladı. Bu arayış için en önemli araç optimizasyondur. “Problemi çözmeye kaynakları en verimli şekilde kullanılması işlemine optimizasyon denir. Fonksiyondan istenilen sonuca göre minimum veya maksimum çıktı verecek şekilde yapılan işlemler bütünü olarak da tanımlamak mümkün” (Nocedal, Wright, 2006). “Optimizasyon teknolojisi, karar verme işlemlerini hızlandırma ve karar kalitesini artırma süreçlerinde kullanılarak gerçek hayatta karşılaşılan problemlerin etkin, doğru ve gerçek zamanlı olarak çözümlenmesini sağlar” (Winston, Wayne ve Goldberg, 2004). Bilgisayar ve teknolojinin yaygınlaşması ve gelişmesiyle beraber problemler için optimizasyon yapılması daha kolay geldi.

Optimizasyona giren fonksiyonun çıktısının değeri algoritma için önemsizdir. Önemli olan hatanın sifira yakınsamadır. Lakin hata toplanırken negatif ve pozitif değerler birbirlerini sıfırlar ve optimizasyon başarısı oldukça düşer. “Optimizasyon da önemli noktalardan biri de “Amaç Fonksiyonu” denilen fonksiyondur. Amaç fonksiyonu zamanla meydana gelen hatanın büyüklüğünü vermektedir” (Maiti, Acharya, Chakraborty, Konar ve Janarthanan, (2008). “Literatüre geçen bazı Amaç Fonksiyonları şunlardır; Mutlak Hatanın Toplamı(IAE), Hata Karelerinin Toplamı(ISE), Zaman Ağırlıklı Hata Karelerinin Toplamı(ITSE) ve Zaman Ağırlıklı Mutlak Hatanın Toplamı(ITAE)” (Campo, 2012).

$$IAE(e) = \sum_{k=0}^T |e_k| \quad (2.39)$$

$$ISE(e) = \sum_{k=0}^T e_k^2 \quad (2.40)$$

$$ITAE(e) = \sum_{k=0}^T k|e_k| \quad (2.41)$$



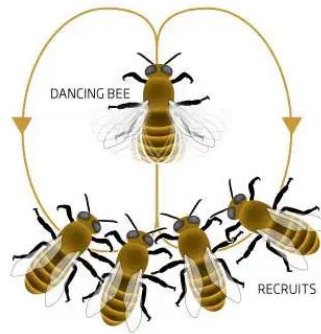
Resim 2.12 Optimizasyon algoritma sınıfları

Her yöntemin kullanım amacı, avantajları ve dezavantajları değişiklik göstermektedir. Sezgisel optimizasyonlar sonucun doğruluğunun kanıtlanabilir olup olmadığını önemsemezler fakat en iyiye yakın çözüm yolları elde ederler. En mükemmel çözümden ziyade mükemmel en yakın çözüme en hızlı şekilde gitmeyi hedefler. Büyük boyutlu optimizasyon problemleri için kullanılması daha verimli olur. Bu sebeple yapılan çalışmada sezgisel optimizasyon yöntemlerinden Yapay Arı Kolonisi optimizasyon yöntemi kullanılmıştır.

2.3.1. Yapay arı kolonisi algoritması

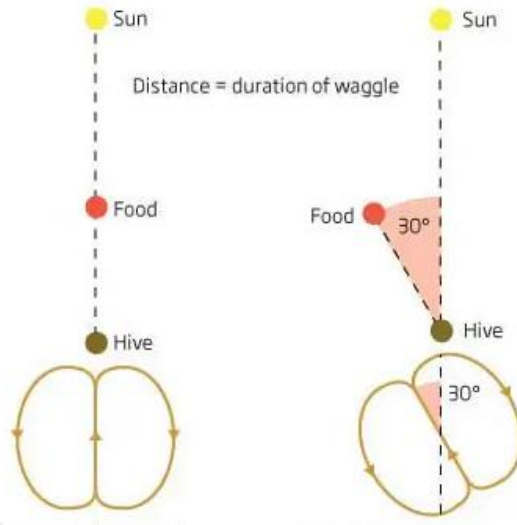
Yapay Arı Kolonisi algoritması arıların doğal davranışlarından esinlenerek tasarlanmış bir optimizasyon yöntemidir. 2005 yılında Derviş Karaboğa tarafından geliştirilmiştir (Karaboğa, 2005).

Temelde kovandaki görevli arıların nektar aramasıyla olay başlar ve nektar bulan görevli arı kovana dönerek titreşim dansı yapar.



Resim 2.13 Arıların dansı

Bu dansın frekansı ne kadar çoksa o kadar çok nektar var demektir. Dans eden arıları nektarın yönünü de güneşe göre tayin ederler.



Resim 2.14 Arıların yön tayini

Bu dans sayesinde kovandaki arılar sürekli olarak nektar bulur ve her seferinde daha iyi bir nektar kaynağı varsa o kaynağa yönelirler.

YAK algoritmasında 3 çeşit arı vardır. İlk olarak kaşif arılar kovandan çıkıp rastgele yemek kaynakları ararlar. Kaynak bulan kaşif arı işçi arıya dönüşür. İşçi arı mevcut yiyecek kaynağının komşuluğunda yeni kaynak arar. Bulunan komşuların kalitesini değerlendirir ve en iyi seçimi hafızaya alır ve kovana döner. Kovandaki gözcü arılar tüm işçi arıların bilgilerini kıyaslarlar ve en iyi kaynağı seçer. Seçilen kaynak tükendiğinde ise işçi arı kaşif arıya dönüşür ve aramaya devam eder (Karaboga, Akay, 2009).

Algoritma akışı şu şekildedir.

1. Başlangıç koşulları oluşturulur.
2. İşçi arılar yiyecek kaynağına gönderilir.
3. Yiyecek miktarına göre gözcü arılar kaynaklara yerleşir.
4. Kaşif arılar yeni kaynak bulmak için gönderilir.
5. En iyi yiyecek kaynağı hafızaya alınır.
6. Bitirme şartları sağlanırsa adım 2.'ye gidilir.
7. Şartlar sağlanıyorsa bitirilir

Algoritma matematiksek olarak ise şu şekilde işler,

Rasgele yiyecek kaynağı oluşturulur.

$$x_{ij} = x_j^{min} + rand(0,1) \times (x_j^{max} - x_j^{min}) \quad (2.42)$$

x konumu, i kaynak sayısını ve j ise parametre sayısını gösterir. Eş. (2.42) sayesinde belirlenen şartlar aralığında rastgele yiyecek kaynağı oluşturulur. Sonrasında $i=1$ 'den N 'ye kadar Eş. (2.42)'de görevli arı için yeni kaynak üretilir (Karaboga, Akay, 2009)..

$$v_{ij} = x_{ij} + \phi_{ij} \times (x_{ij} - x_{kj}) \quad (2.43)$$

Bu basamakta v_{ij} , daha önce belirtilmiş alt ve üst sınırları aşması durumunda Eş. (2.43)'ü kullanarak j parametresine ait alt ve üst sınırları ötelenir (Karaboga, Akay, 2009).

$$v_{ij} = \begin{cases} x_j^{min} & v_{ij} < x_j^{min} \\ v_{ij} & x_j^{min} \leq v_{ij} < x_j^{max} \\ x_j^{max} & v_{ij} > x_j^{max} \end{cases} \quad (2.44)$$

Bu kaynağın maliyet değeri $f(v_i)$, Eş. (2.45)'de yerine koyulur ve uygunluk değeri hesaplanır. v_i ile x_i arasında seçim yapılır. Maliyet değeri hangi seçim için daha kazançlıysa o çözüm hafızaya alınır (Karaboga, Akay, 2009).

$$uygunluk_i = \begin{cases} 1/(1 + f_i), & f_i \geq 0 \\ 1 + |f_i|, & f_i < 0 \end{cases} \quad (2.45)$$

Gözcü arılar seçim yaparken uygunluk değerine bağlı olasılık hesabı $i=1, \dots, SN$ olmak üzere Eş. (2.46)'da gösterilmiştir (Karaboga, Akay, 2009).

$$p_i = \frac{uygunluk}{\sum_{i=1}^{SN} uygunluk} \quad (2.46)$$

“Seçim işleminde her kaynak için [0,1] aralığından üretilen p_i değeri, rastgele üretilen değerden büyükse gözcü arı tarafından Eş. (2.43)’ü kullanarak yeni kaynak üretilir ve üretilen v_i ile x_i arasında seçim işlemi yapılır” (Karaboga, Akay, 2009). Daha iyi olan seçilir. Bu adım gözcü arılar tüm kaynaklara dağılıncaya kadar tekrar edilir. Sonlandırma koşulu sağlanıyorsa algoritma bitirilir. Bu akış ile parametreler en iyi değere yakınsamış olur.

Yapay Arı Kolonisi optimizasyonuna giren fonksiyonun çıktı değeri bu algoritma için önemsizdir. Önemli olan hatanın sifira yakınsamadır. Hata toplanırken negatif ve pozitif değerler birbirlerini sıfırlar. Bu yüzden optimizasyon başarısı düşer. Başarı yükseltmek adına bir önceki bölümde gösterilen Eş. (2.39), Eş. (2.40) ve Eş. (2.41) Amaç Fonksiyonları hata toplamada kullanılabilirler.

3. MATERYAL VE DENEYSSEL YÖNTEM

Bu bölümde çalışmanın ana materyalleri detaylı olarak tanımlanmıştır ve tanıtılmıştır. Bölüm 2.1.2’de çıkarılan Buck Dönüştürücü denklemleri, Forward Euler yönetimi ile Z-Domeni’ne çevrilmiştir. Z-Domeni’ne çevrilme sebebi, örnekleme tabanlı yapılacak kontrol sistemine yazılımsal entegrasyonunun eklenecek olmasıdır. Z-Domeine çevrilen dönüştürücünün yazılımsal algoritması detaylı olarak tanıtılmış ve açıklanmıştır. Dönüştürücüye MIT Kurallı MRAC tasarımı ayrık zamana çevrilip algoritmik olarak eklenmiştir. Sistem Matlab yazılım ortamında çalıştırılıp test edilmiştir. Daha sonra maliyet fonksiyonu olarak bu sistem atanarak YAK algoritması çalıştırılmıştır. YAK algoritmasından çıkan değerler MRAC içindeki devre çalışması için gereken optimum parametrelerini vermektedir. Bu sayede devredeki çıkış değerinin oturma süresi ve aşım değeri en uygun ve en optimum şekilde hareket etmektedir.

Bölüm 2.1.2’de çıkarılan Eş. (2.11), Eş. (2.12), Eş. (2.13) ve Eş. (2.14) sürekli zaman ifadeleridir. Yazılımsal entegrasyonu yapılacak bu denklemler ayrık zamana çevrilmelidir ki entegrasyon sağlanabilsin. Bu bağlamda Buck tipi dönüştürücü devre denklemleri Forward Euler yöntemi ile ayrık zamana çevrilmiştir.

$T(s) \rightarrow T(z)$ dönüşümü için Forward Euler yöntemi Eş. (3.1)’de tanımlanmıştır.

$$s \rightarrow \frac{1}{T}(z - 1) \quad (3.1)$$

T, örnekleme süresi olmak üzere temel olarak ayrık zaman, sürekli zamanda belirli aralıklarla alınan örneklemlerden oluşan bir yapıdır. Bu bağlamda örnekleme alınacak süre aralığı belirlenmelidir. Eş. (3.2) verilmiş olsun ve bu deklemler, Eş. (3.1) yaklaşımıyla ayrık zamana çevrilirse Eş. (3.3) ortaya çıkar.

$$\frac{1}{\tau s + 1} \quad (3.2)$$

$$\frac{T}{\tau z + (T - \tau)} \quad (3.3)$$

Çıkarılan Eş. (3.3) Z Domain'de olmasına karşın Simulink'de blok diyagram şekline çevrilmeye ve algoritma olarak yazılmaya uygun değildir. Bu denklemde en büyük kuvvetli z değişkeninin katsayısını z^0 olacak şekilde sadeleştirme yapılırsa sistem entegrasyona uygun hale gelmektedir. Eş. (3.3)'e bu işlem yapılırsa ortaya Eş. (3.4) çıkar.

$$\frac{Tz^{-1}}{\tau + (T - \tau)z^{-1}} \quad (3.4)$$

Eş. (3.4) bir transfer fonksiyonudur. Transfer fonksiyonu temelde çıkışın girişe oranı şeklinde ifade edildiğinden bu denklem, Eş. (3.5) şeklinde yazılırsa bir çıkış bir de giriş ifadesi elde edilir ve blok diyagram düzenlemesi için kolaylık sağlanmış olur.

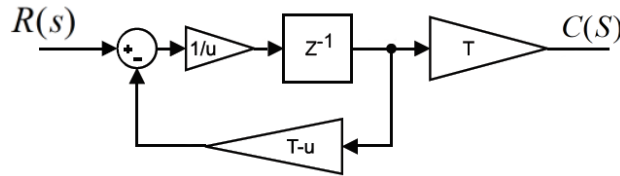
$$\frac{C}{G} = \frac{X(Tz^{-1})}{X(\tau + (T - \tau)z^{-1})} \quad (3.5)$$

Eş. (3.5)'den görüldüğü üzere bir X değişkeni ifadeyi matematiksel olarak değiştirmedeği gibi C ve G değerlerini birbirine benzetme açısından bir araçtır. Bu denklem Eş. (3.6) ve Eş. (3.7) gibi düzenlenirse,

$$C = TXz^{-1} \quad (3.6)$$

$$X = \frac{G - Xz^{-1}(T - \tau)}{\tau} \quad (3.7)$$

Eş. (3.6) ve Eş. (3.7), Eş. (3.5)'in pay ve paydasının parçalanıp düzenlenmiş halidir. Eş. (3.5)'i Eş. (3.6) ve Eş. (3.7) referansında Simulink ortamında blok diyagrama çevrilirse Resim 3.1 ortaya çıkar.



Resim 3.1 Örnek denklem simulink blok diyagramı

Bu bağlamda Forward Euler yöntemi Bölüm 2.1.2'deki sürekli zaman devre denklemlerine uygulanırsa ve yukarıda belirtildiği gibi düzenlenirse devre denklemleri şu şekilde ortaya çıkar;

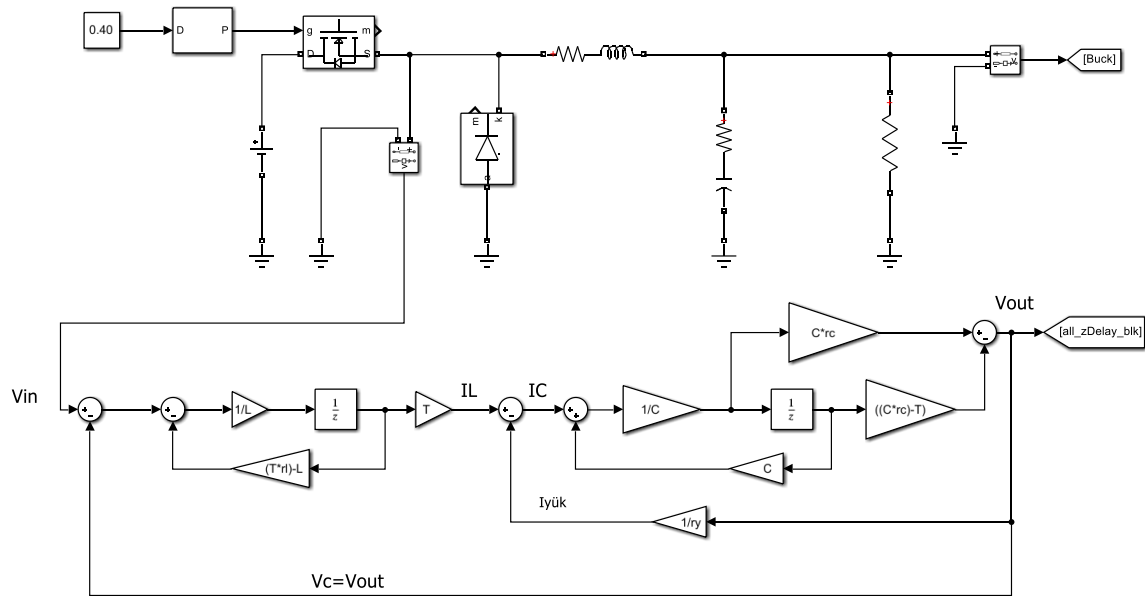
$$I_L = (V_{in} - V_{out}) \frac{Tz^{-1}}{(TR_L - L)z^{-1} + L} \quad (3.8)$$

$$I_C = I_L - I_{Ryük} \quad (3.9)$$

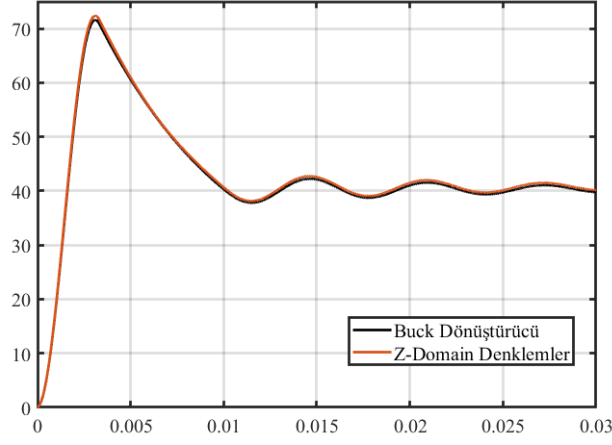
$$\frac{V_C}{I_C} = \frac{(CR_C - (CR_C - T)z^{-1})}{C - Cz^{-1}} \quad (3.10)$$

$$I_{Ryük} = V_C \frac{1}{R_{yük}} = V_{out} \frac{1}{R_{yük}} \quad (3.11)$$

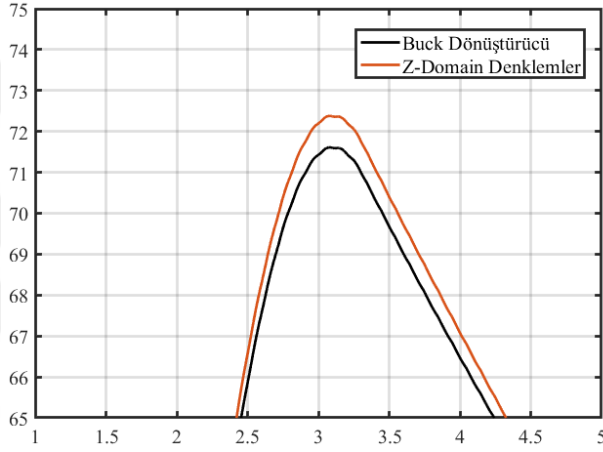
Çıkarılan Eş. (3.8), Eş. (3.9), Eş. (3.10) ve Eş. (3.11)'in Resim 5.1'deki yöntem ile Simulink ortamında blok diyagrama çevrilmiş hali Resim 3.1'de görülmektedir.



Resim 3.2 Devre denklemleri simulink blok diyagram



Resim 3.3 Blok diyagram sonuçları ve devrenin sonuçlarının kıyaslanması



Resim 3.4 Blok diyagram sonuçları ve devrenin sonuçlarının kıyaslanması (Yakınlaştırılmış)

Resim 3.3 ve Resim 3.4'te görüldüğü üzere Resim 3.2'deki devre sonuçları Resim 2.4'teki sonuçlar ile kıyaslanınca aynı değer çıktıkları olduğu görülmektedir. Bu bağlamda Z-Domain'de çıkarılan denklemler gerçeğe uygun ve stabil çalıştığı ispatlanmıştır. Bir sonraki aşama olarak bu çıkarın denklemler Matlab ortamında hiçbir özel fonksiyon kullanılmadan yazılım haline çevrilmesi anlatılmıştır.

$$z^{-1}$$

Resim 3.5 Simulink gecikme(delay) bloğu

Resim 3.5'te görülen z^{-1} bloğu simulink'de gecikme bloğu olarak da bilinmektedir. Bu blok çalışma dinamiği olarak girişinden gelen sinyali hafızada tutar ve ayrık zaman bloğu olduğundan içine yazılan parametre kadar sonraki adımda hafızasındaki çıkışına verir ve bu sırada çalışan adımdaki değeri bir sonraki adımda çıkışına vermek amacıyla hafızada tutar. Bu çalışma şekli algoritmik olarak şöyle özetlenebilir,

1. Sistem çalışır. İlk çalışmada Gecikme Bloğu girişinden aldığı değeri hafızaya alır ve çıkışına sıfır değerini verir.
2. Sistem sonraki adıma geçer. Gecikme bloğu çıkışa bir önceki adımda hafızada tuttuğu değeri verir ve girişinden o anki aldığı değeri hafızada tutar.
3. Toplam örnekleme bitene kadar işlemler tekrarlanır.

Akıştan görüldüğü üzere ilk çalışmada sıfır değerini çıkışına vermesine karşın girişine gelen değer hesaplanıp adım sonunda hafızada tutulması gereklidir. Bu bağlamda Resim 5.1'deki örnek sistem yazılıma aktarılırsa şu şekilde bir algoritma ortaya çıkmaktadır;

```
1   xctf1=(giris-((T-u)*xctf1Old))*(1/u);  
2   cikis1tf1=T*xctf1Old;  
3   xctf1Old=xctf1;
```

Resim 3.6 Örnek sistem matlab kodu

“cikis1tf1” değişkeni Res 3.1'de görülen $C(s)$ çıkışıdır. “giris” değişkeni sisteme giren $R(s)$ giriş değeridir. “xctf1Old” ise “xctf1” değerinin bir önceki örneklemedeki değerini hafızada tutacak şekilde akışa eklenmiştir. Bu şekilde bir akış yapılmasının amacı blok diyagramda bir gecikme bloğu olmasındandır. Tek gecikme bloğu dinamik olarak işlenen adımda çıkışa bir önceki adımdaki değeri vermektedir. Resim 3.6'da görülen algoritma ile bu işlem sağlanmış olur.

Bu örnekler referansında Buck çevirici Matlab ortamında algoritmaya çevrilmeye çalışılırsa öncelikle giriş sinyalini taklit edecek bir yapı kurulması gereklidir. Buck çevirici önceki bölümlerde anlatıldığı gibi girişine bir MOSFET sayesinde sinyali kontrol sinyalinin doluluk oranını baz alarak giriş gerilimi uygular yada uygulanan gerilimi keser. Aslında giriş gerilimi büyüklüğünde bir PWM sinyali oluşturmak bu olayı simüle etmeye olanak sağlar.

```

1   vin=100;                               %Giriş gerilimi
2   t = 0:1e-6:0.05;                       % 1us örnekleme ile 0.05s çalıştır
3   f=10000;                               % 10kHz anahtarlama frekansı
4   dutyC=0.4;                             %Doluluk Oranı sabiti
5   compare=-1+(2*dutyC);
6   x = sawtooth(2*pi*f*t);               %Testeredişli
7   pwm=vin*(x<compare);

```

Resim 3.7 PWM sinyali oluşturulması

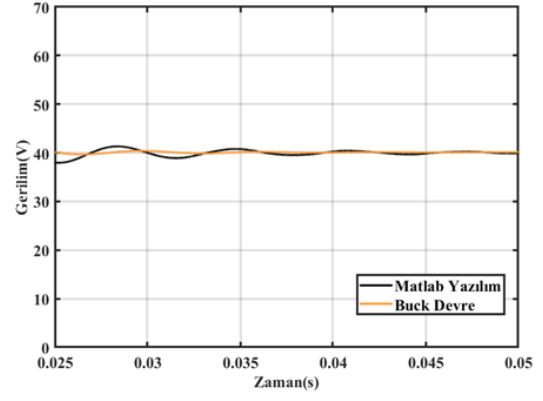
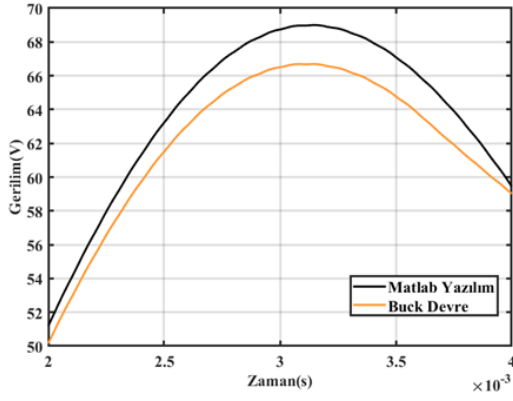
Resim 3.7’de görülen algoritma giriş gerilimi 100V olan bir sistemde MOSFET’in iletim ve kesim durumunun değerlerini ayırık zaman çalışmasına göre ayarlayan bir yapıdır. Bir PWM oluşturmak için DC bir gerilim ile testere dişli sinyali karşılaştırmak gereklidir. Bu bağlamda Resim 3.7’de görülen yapıda bir testere dişli sinyali oluşturulmuş ve bu sinyal PWM doluluk oranına göre aritmetik işlem yaparak uygun karşılaştırmayı yapmaktadır. Son olarak “pwm” değişkeni giriş gerilimi ile çarpılır ve tıpkı gerçek devredeki MOSFET’in sisteme uyguladığı gibi ya sıfır ya da giriş gerilimi değerlerini uygun adımlara atamaktadır.

```

1   for i=1:1:tts
2       %1.Denklem
3       il=T*xilOld;
4       xil=(Vio-((T*rl)-L)*xilOld)*(1/L);
5       xilOld=xil;
6       %2.Denklem
7       ic=il-iy;
8       %3.Denklem
9       xvc=((ic+(C*xvcOld))*(1/C));
10      vc=(C*rc*xvc)-((C*rc)-T)*xvcOld;
11      xvcOld=xvc;
12      %4.Denklem
13      iy=vc*(1/(ry+rvri));
14      Vout(i)=vc;
15      Vio=pwm(i)-Vout(i);
16   end

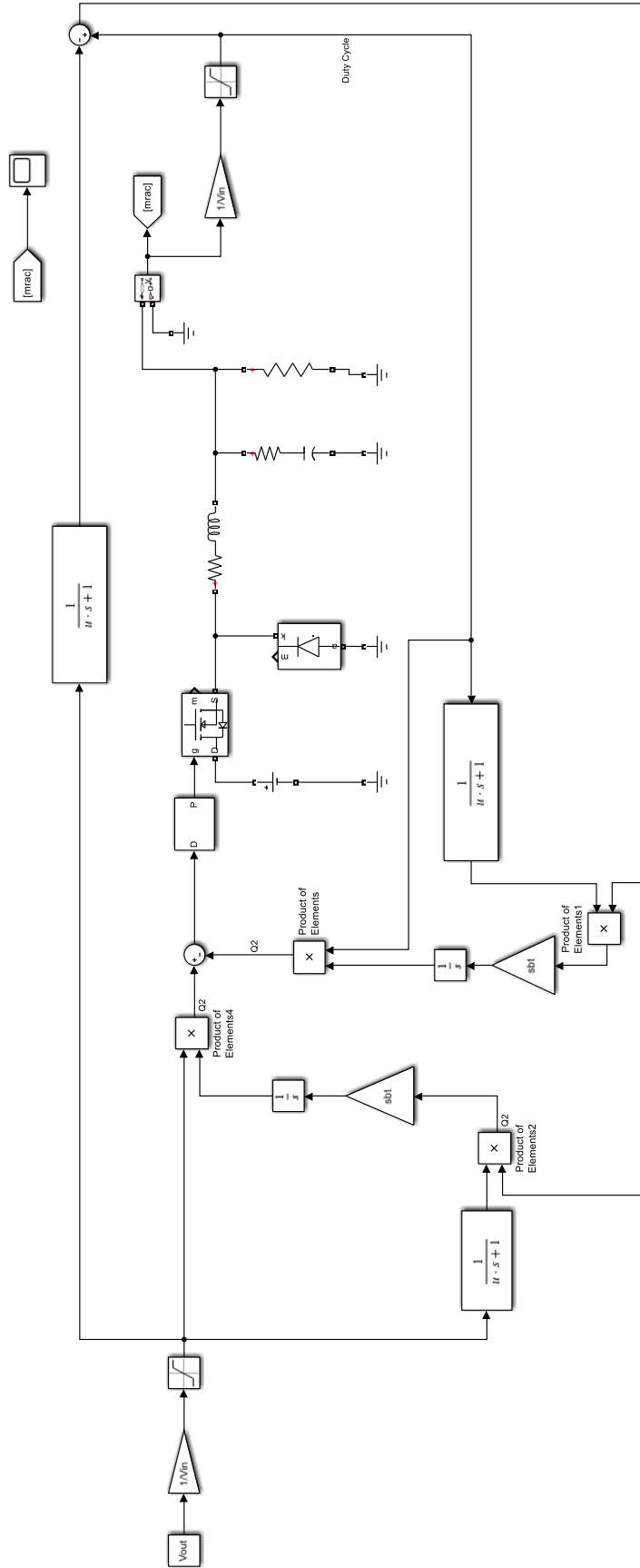
```

Resim 3.8 Buck dönüştürücü matlab algoritması



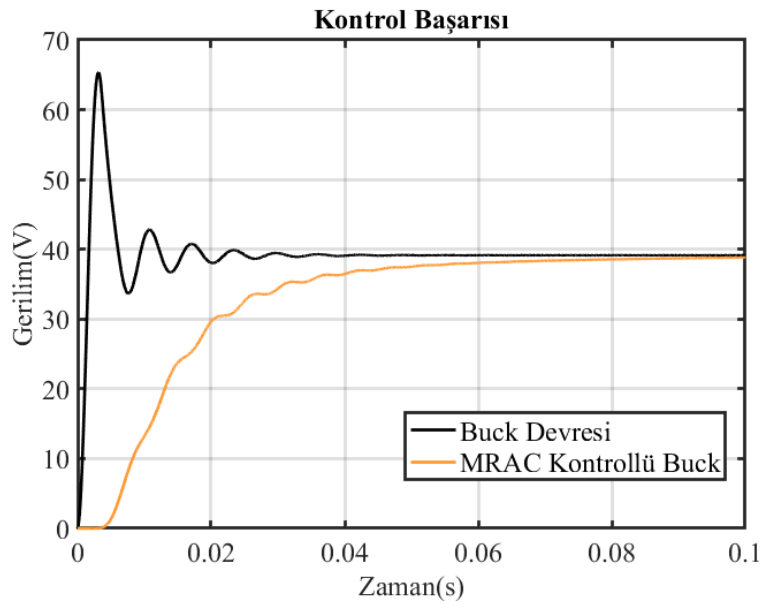
Resim 3.9 Buck devre ve matlab yazılım sonuç kıyaslaması(yakınlaştırılmış)

Resim 3.9’da görüldüğü üzere aynı değerlere sahip Buck devresi Matlab yazılımı ve Simulink simülasyonunda kalkış değerleri ve oturma değerleri aynıdır. Optimizasyon yapılacak sistem için aradaki dalgalanma farkları sorun teşkil etmemektedir.



Resim 3.10 MIT kuralıyla tasarlanmış mrac kontrollü buck dönüştürücü

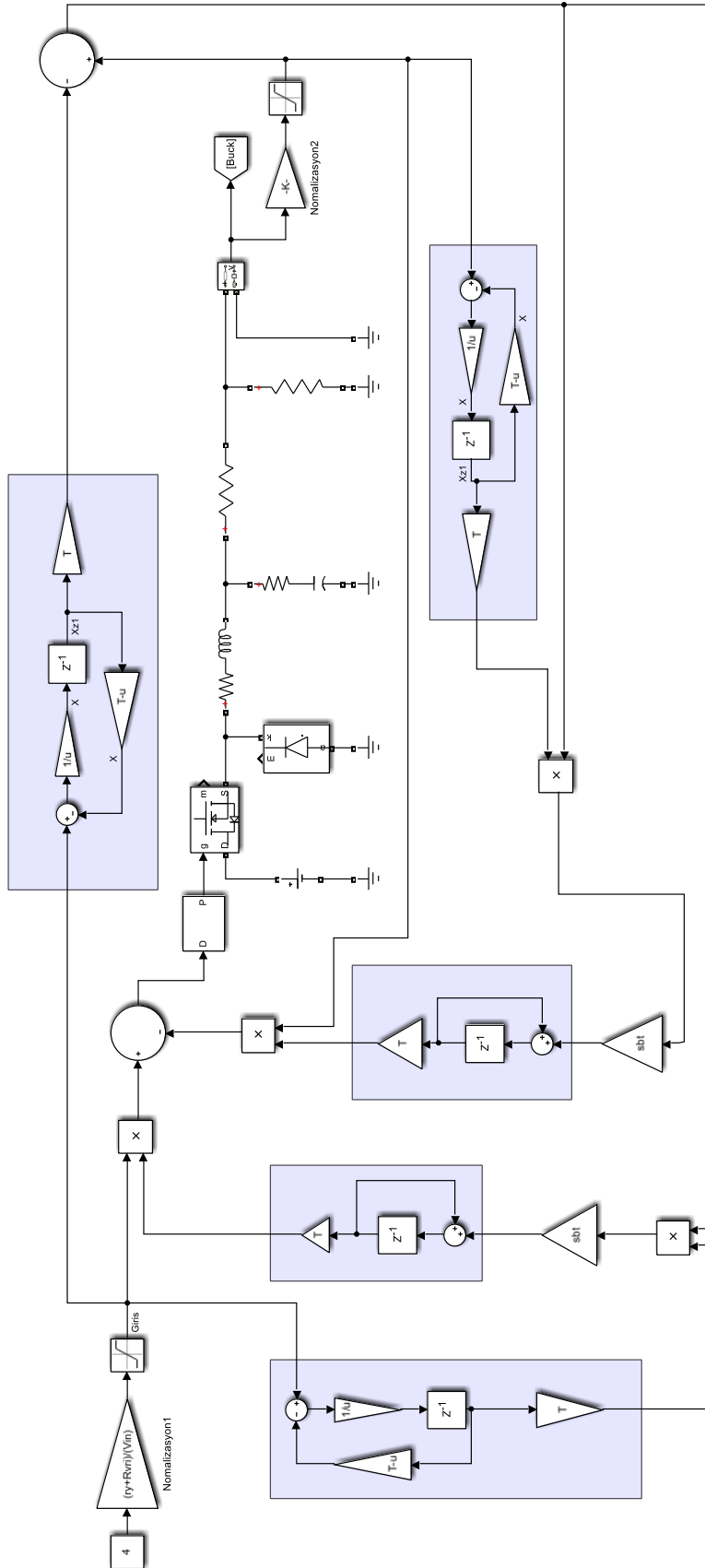
Resim 3.10'da Resim 2.11'de tasarlanan kontrol sistemi Buck dönüştürücüye uygulanmıştır. Ek olarak Simulink'te kurulan bu diyagramda kontrol sistemi PWM doluluk oranının kontrolünü sağlamaktadır bu sebeple Buck devre çıkışında alınan gerilim değeri giriş gerilimin nominal değerine bölünerek normalize edilmektedir. Düzeltme parametresi henüz optimizasyon yapılmadığı için deneme yanılma yoluyla bulunmuştur. Bu bağlamda γ değerine -2000 sayısı atandığında ve sistem çalıştırıldığında Resim 3.11'deki sonuçlar elde edilmektedir.



Resim 3.11 Kontrollü ve kontrolsüz buck devresi sonuçları

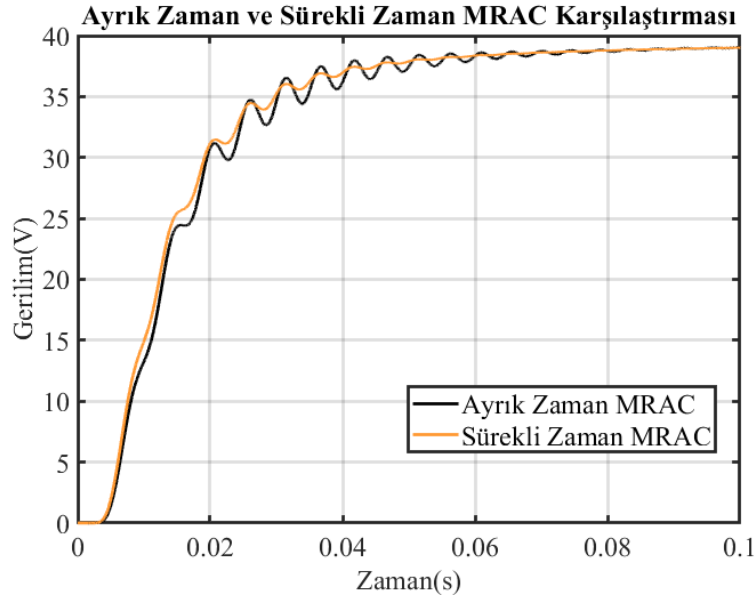
Resim 3.11'de MIT Kurallı MRAC kontrolün başarısı görülmektedir. MRAC ile kontrol edilen Buck devresi aşım ve dalgalanma yapmamaktadır.

Resim 3.10'da oluşturulan sistem sürekli zaman transfer fonksiyonlarının yer aldığı sistemdir. Gerçekte devrenin kontrol mekanizmasını bir bilgisayar, bir mikrodenetleyici ya da bir mikro-işlemci yapmaktadır. Yani kontrol sistemi ayrık zamanda belirli örnekleme aralığında çalışmaktadır. Bu bağlamda gerçeğe en uygun simülasyonu yapmak amaçlı Resim 3.12'deki sistem oluşturulmuş. Bu sistem sürekli zamanda çalışan bir devrede ayrık zamanlı bir kontrol yapmaktadır. Resim 3.10'daki sistemde kontrol mekanizması ayrık zamana çevrilip bu bölümün başında gösterilen işlemlere tabi tutulduğunda Resim 3.12'deki Simulink diyagramı elde edilir.



Resim 3.12 MIT kurallı mrcak kontrollü buck devresi

Resim 3.12’de görüldüğü üzere giriş referansı bazı aritmetik işlemlere tabi tutulmuştur. Bunun sebebi devreyi akım kontrollü olmasının istenmesidir. Yükün üzerinden geçmesi istenen akım değeri sisteme giriş olarak verilir. Aslında sistemdeki PWM üretici de kontrolü sağlayan cihazdır. Bu bağlamda istenilen yük akımı algoritmaya verilmesi sadece aritmetik işlemler ile PWM doluluk oranını sisteme verir ve sistem, içinde bu değeri çevirir. Yine devrenin olabildiğince gerçeğe uygun olması açısından çıkışa yüke seri bir 0.05Ω ’luk bir direnç yerleştirilmiştir ve bu direnç üzerine düşen gerilim ölçülüp yine aritmetik işlemler sayesinde bu değer PWM doluluğuna çevrilmektedir. Yine sisteme deneysel olarak elde edilmiş γ değerine -2000 atanırsa sonuçlar Resim 3.13’deki gibi olmaktadır.



Resim 3.13 Ayrık zaman ve sürekli zaman mrac karşılaştırması

Resim 3.13’deki sonuçlardan görüldüğü üzere ayırık zamanlı MRAC kontrolü sürekli zamandaki transfer fonksiyonuyla örtüşmektedir. Ayırık zamandaki daha yüksek dalgalanma miktarının sebebi ayırık zamandaki kontrolcü $20\mu s$ örnekleme zamanı seçilmesidir. $20\mu s$ seçilmesinin sebebi ise günümüzde kontrolcü olarak kullanılacak bir mikro-denetleyici ya da mikro-işlemci için daha gerçekçi bir örnekleme süresi olmasıdır. Sistemdeki bu dalgalanmaları en aza indirmek amaçlı bu sistem yukarıda gösterildiği üzere Matlab algoritmasına çevrilip optimizasyon algoritmasına tabi tutulursa çok daha başarılı bir sinyal elde etmek mümkün.

```

1 %DEVRE DEĞİŞKENLERİ
2 cs=0.5; %Simulasyon Çalışma Süresi
3 ry=100; %Yük direnci
4 rvri=0.05; %Akım ölçmek için direnç
5 vin=100; %Giriş gerilimi
6 iouts=5; %İstenilen yük akımı
7 ysbtl=BestSol.Position(1); %Y1 değeri
8 ysbtl=BestSol.Position(2); %Y2 değeri
9 u=1e-2; %TF mekanizması içindeki sabit.u=1e-2;
10 tsample=20e-6; %Örnekleme süresi
11
12 %Buck Dönüştürücü Parametreleri*****
13 L=1e-3;
14 rl=0.01;
15 C=1000e-6;
16 rc=0.01;
17 T=tsample;

```

Resim 3.14 MIT kuralları MRAC kontrollü buck dönüştürücü sistem parametreleri

```

132 % 1.TF Mekanizması*****
133 - xctf1=(iout-((T-u)*xctf1Old))*(1/u);
134 - cikis1tf1=T*xctf1Old; % 1.TF Mekanizması Çıkışı
135 - xctf1Old=xctf1;
136 %*****
137
138 %Hata e Değeri bulma
139 - hata=dutyccikis-cikis1tf1;
140
141 % 2.TF Mekanizması*****
142 - xctf2=(dutyccikis-((T-u)*xctf2Old))*(1/u);
143 - cikis1tf2=T*xctf2Old; % 2.TF Mekanizması Çıkışı
144 - xctf2Old=xctf2;
145 %*****
146
147 % 1.Y İntegratör Kısmı*****
148 %1. Y/s Girişi
149 - yintlq=cikis1tf2*hata; % 1.İntegratörün girişi.
150 - yintlq=ysbtl*yintlq; %1.İntegratör girişi y sabitiyle çarpılır.
151
152 %1. Y/s Mekanizması
153 - xcyintl=yintlq+xcyintlOld;
154 - cikisyintl=T*xcyintlOld; %1.İntegratörün çıkışı.
155 - xcyintlOld=xcyintl;
156
157 %*****
158
159 - q2=cikisyintl*dutyccikis;
160
161 % 3.TF Mekanizması*****
162 - xctf3=(iout-((T-u)*xctf3Old))*(1/u);
163 - cikis1tf3=T*xctf3Old; % 3.TF Mekanizması Çıkışı
164 - xctf3Old=xctf3;
165 %*****

```

Resim 3.15 Kontrol mekanizması kısım 1

```

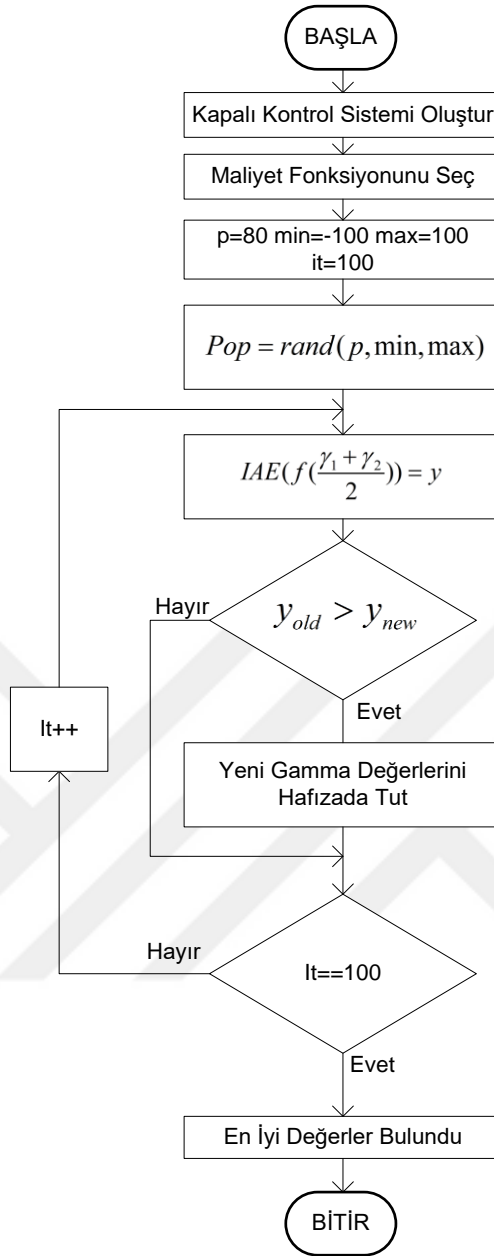
169 % 2.Y İntegratör Kısmı*****
170 % 2.Y/s Girişi
171 - yint2g=cikis1tf3*hata; %İntegratörün girişi.
172 - yint2g=ysbt2*yint2g; %İntagrator girişi y sabitiyle çarpılır.
173
174
175 % 2.Y/s Mekanizması
176 - xcyint2=yint2g+xcyint2Old;
177 - cikisyint2=T*xcyint2Old; % 2.İntegratörün çıkışı.
178 - xcyint2Old=xcyint2;
179 %*****
180
181 - q1=cikisyint2*iout;

```

Resim 3.16 Kontrol mekanizması kısım 2

Bu bölümün başında anlatılan ayırıklaştırma işlemleri Resim 3.12'deki sisteme uygulanmıştır. Sistem parametreleri Resim 3.14'de görülmektedir. Bu parametreler optimizasyona girecek olan sisteme ait olan parametrelerdir. Devre dinamiği, eleman değerleri ve kontrolcü değişkenlerine göre farklılık gösterdiğinden dolayı farklı özellikli MIT Kurallı MRAC kontrollü Buck dönüştürücüler için bu algoritma ile optimizasyon yapılmasına olanak sağlar. Resim 3.15 ve Resim 3.16 ise Resim 3.12'de görülen MIT Kurallı MRAC kontrolcü mekanizmasının Matlab algoritmasıdır.

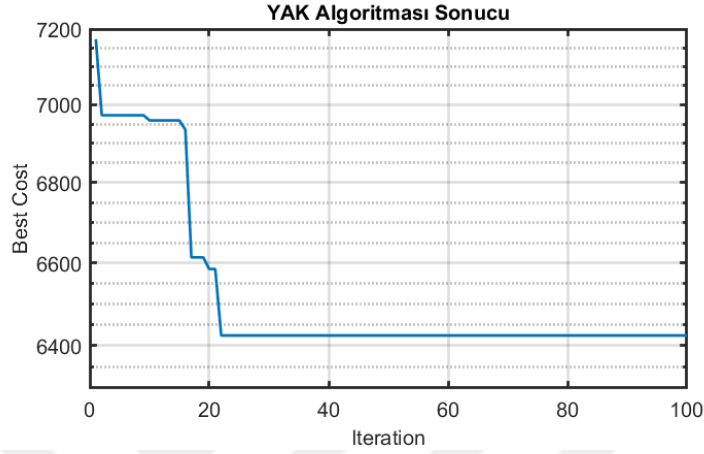
Tasarlanan sistemin normal MIT Kuralı'yla tasarlanmış MRAC'a göre farkı düzeltme katsayısının bulunma yöntemidir. MIT Kuralı'yla tasarlanmış bir MRAC sistemde gamma sabiti tek bir değer olarak bulunur ve sisteme atanır. Yapılan bu çalışmada 2 farklı gamma değeri optimizasyona dahil edilmiştir. Optimizasyon çıkışında 2 gamma değeri çıkışı alınmaktadır. Daha sonra bu gamma değerlerinin ortalaması alınıp sistem çalıştırılmaktadır.



Resim 3.17 Optimizasyon akış diyagramı

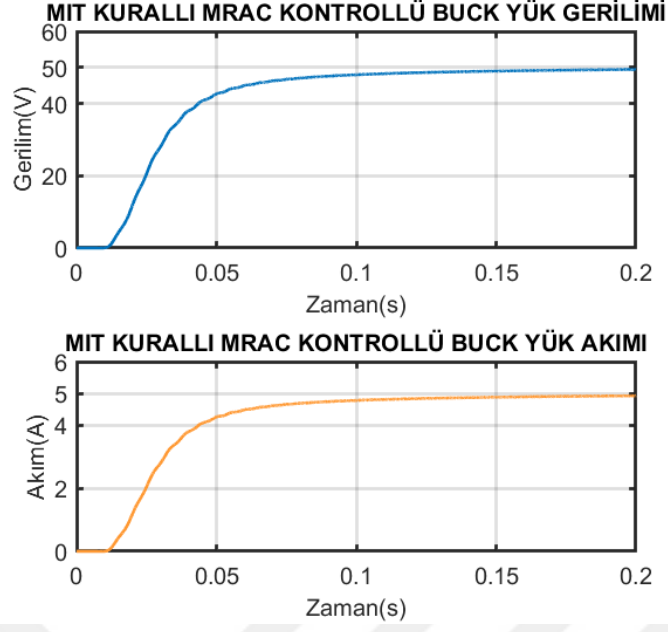
Resim 3.17’de sisteme yapılan optimizasyon algoritmasının akış diyagramı görülmektedir. Optimizasyon için maliyet fonksiyonu olarak Buck dönüştürücü atandıktan sonra YAK algoritmasının parametreleri seçilmiştir. “p” popülasyon, “min” optimize edilecek parametrenin en küçük değeri, “max” optimize edilecek parametrenin en büyük değeri, “it” ise iterasyon sayısıdır. Bir sonraki adımda YAK algoritmasının rastgele popülasyonu oluşturulur. Oluşturulan popülasyon değerleri sırasıyla belirlenen maliyet fonksiyonu içinde çalıştırılır. Tasarlanan Buck dönüştürücünün sonunda hata toplayan IAE fonksiyonunun sonucu optimizasyon algoritmasına geri döndürülür. Geri dönen değer bir önceki değerden daha küçükse hata miktarı azalmış demektir ve daha az

hata miktarını veren parametre hafızada tutulur. Geri dönen değer bir önceki değerden daha büyükse hata miktarı büyümüş ki bu da yeni parametre hafızada tutulmaya layık değil demektir. İterasyon sayının sınırına ulaşıldığında algoritma kendini tamamlar. Bütün popülasyon değerleri ve bu değerlerin belirlenen genişleme miktarları denenmiş olur. Algoritma son bulur ve en iyi sonuç çıktısını verir.

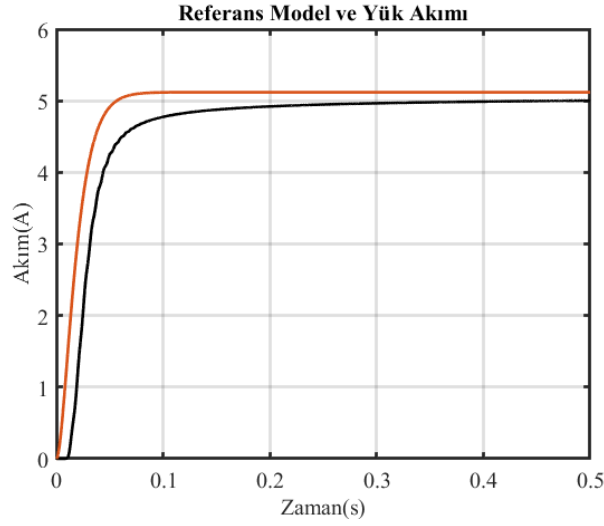


Resim 3.18 YAK algoritması sonucu iterasyon-hata toplamı grafiği

Resim 3.18 Rastgele oluşturulmuş popülasyon değerlerinin MIT Kurallı MRAC Kontrollü Buck Dönüştürücün YAK algoritmasına tabi tutulup γ değerlerinin denemesinden oluşan hata toplamlarıdır. Bu çalışma için maksimum değer 1000 ve minimum değer -1000 olmak üzere optimizasyon yapılmıştır. Optimizasyon algoritması γ_1 değerini -422.68 ve γ_2 değerini -878.84 olarak bulmuştur. Bu 2 değerın ortalaması alınır ve Resim 3.12'deki sistemde yerine atanır ve sonuçlar aşağıdaki gibi olmaktadır.



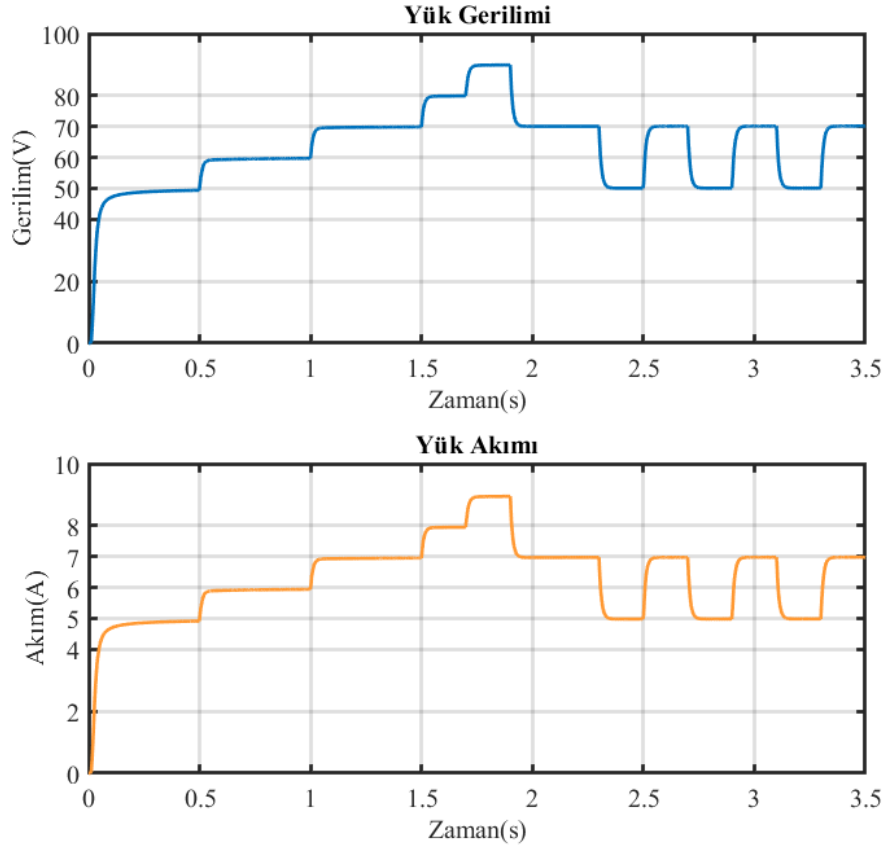
Resim 3.19 simulasyon sonuçları



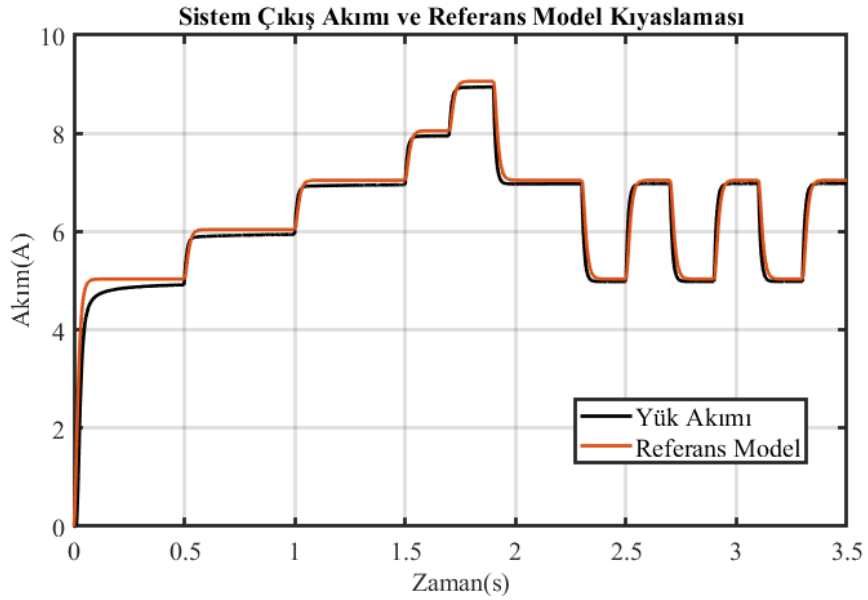
Resim 3.20 Referans akım ve yük akımı karşılaştırması

Resim 3.20’de referans modeli takip eden sistemin başarısı görülmektedir.

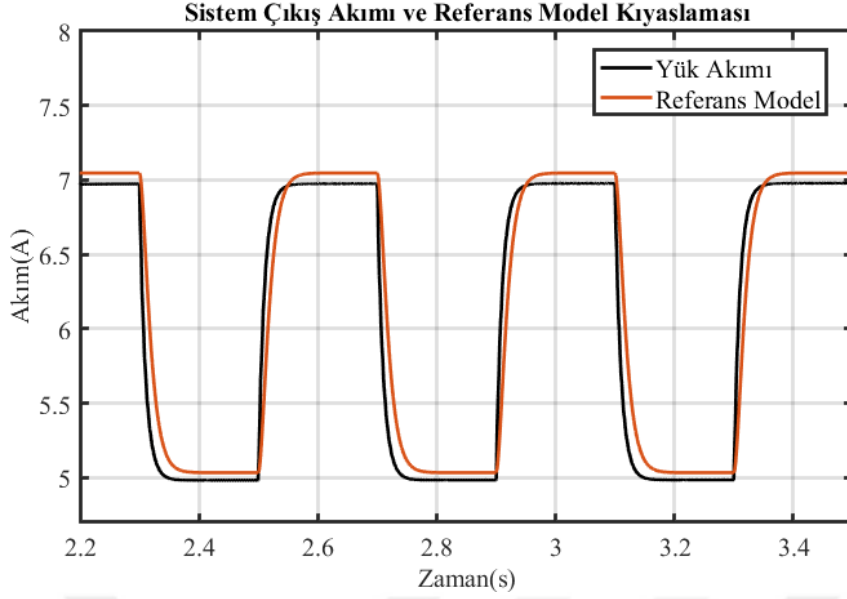
Zamanla akım ihtiyacı değişen ya da ayarlanmak istenen bir yük olduğunda MIT Kurallı MRAC kontrolü ile referans akım zamanla değiştirilerek istenilen çıkış akımı elde edilebilir. 500ms ve 200ms’de değişen referans akımlarına sistemin verdiği tepki Resim 3.21’de, Resim 3.22’de ve Resim 3.23’de görülmektedir.



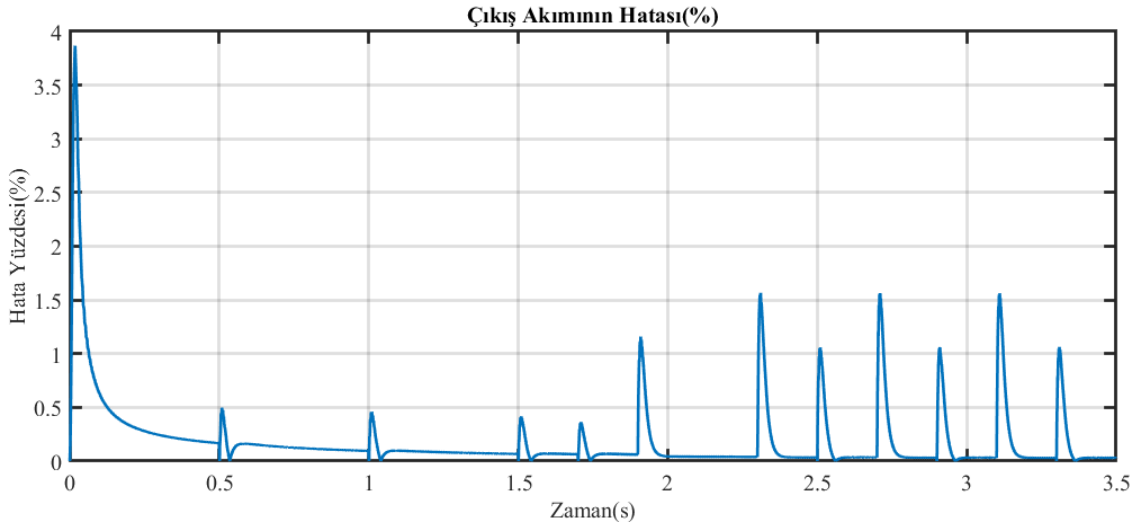
Resim 3.21 Sistemin değişen referans akımlara tepkisi



Resim 3.22 Çıkış akımı ve referans model kıyaslaması



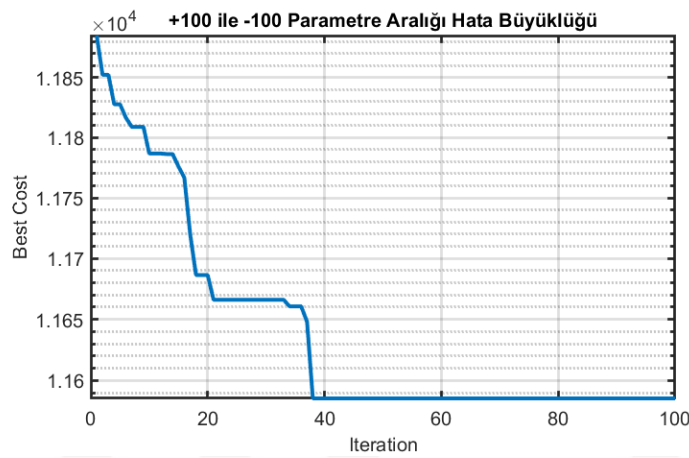
Resim 2.23 Çıkış akımı ve referans model kıyaslaması(yakınlaştırılmış)



Resim 2.24 Yük akımı hatası(%)

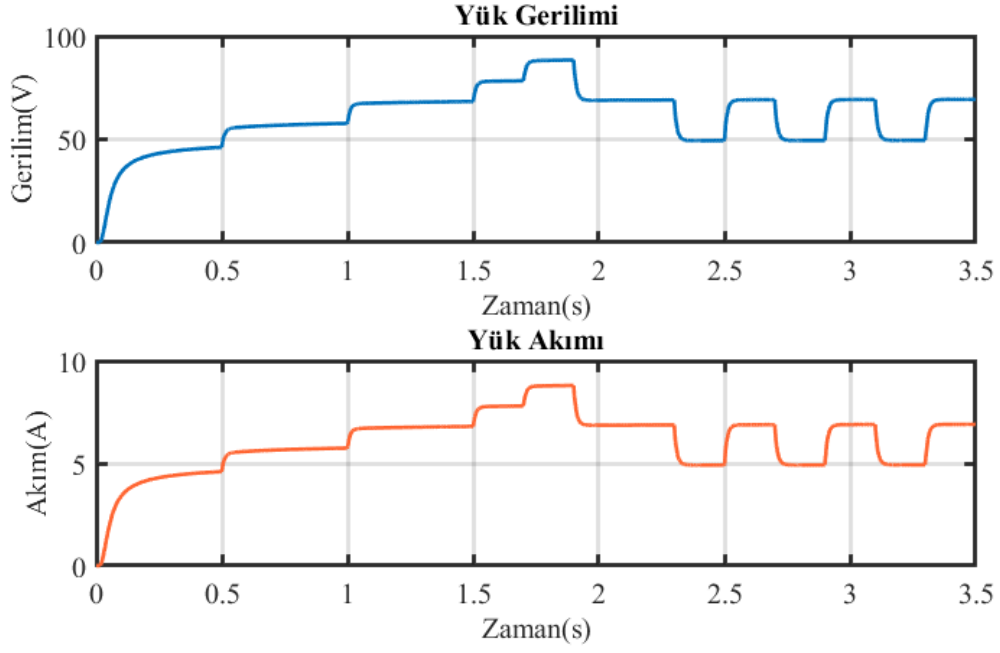
Sistem zamanla değişen referans akımlarına 100ms altında tepki verebilmektedir. Resim 2.24'de sisteme verilen referans akım ile yük akım farkının yüzdesi görülmektedir. İlk çalışma anında %4'ten küçük bir hata olmaktadır ve 100ms civarında bu hata %1'in altına düşmektedir. Daha sonra değişen referans akımlara sistem tepkisi geciktiğinden anlık olarak bile %1.5'in altındadır.

Sistem başarısı yukarıdaki resimlerde görölse de eęer bu sistem geręeklenirse bazı sorunlar ortaya ıkabilir. Optimizasyon algoritmasında buldurulan en iyi γ parametresi nceden belirtildięi gibi +1000 ile -1000 deęerleri arasında alınmaktadır. Bu deęerler sistem alıřması iin bařarılı bir iř ortaya koysa da sonu olarak bir dzeltme parametresidir. Eęer sisteme bu dzeltme iřleminde nce herhangi bir istenmeyen giriřim olursa, bu istenmeyen harmonięin de katı alınacaktır. Katı alınan istenmeyen harmonik giriřim sinyali bozacak ve sistemi dzensizlięe srkleyecektir. Bu durumun nne gemek amalı olarak γ parametresinin alt ve st deęerlerini +100 ile -100 arasında alınması sistem kararlılıęı iin daha doęru olacaktır.

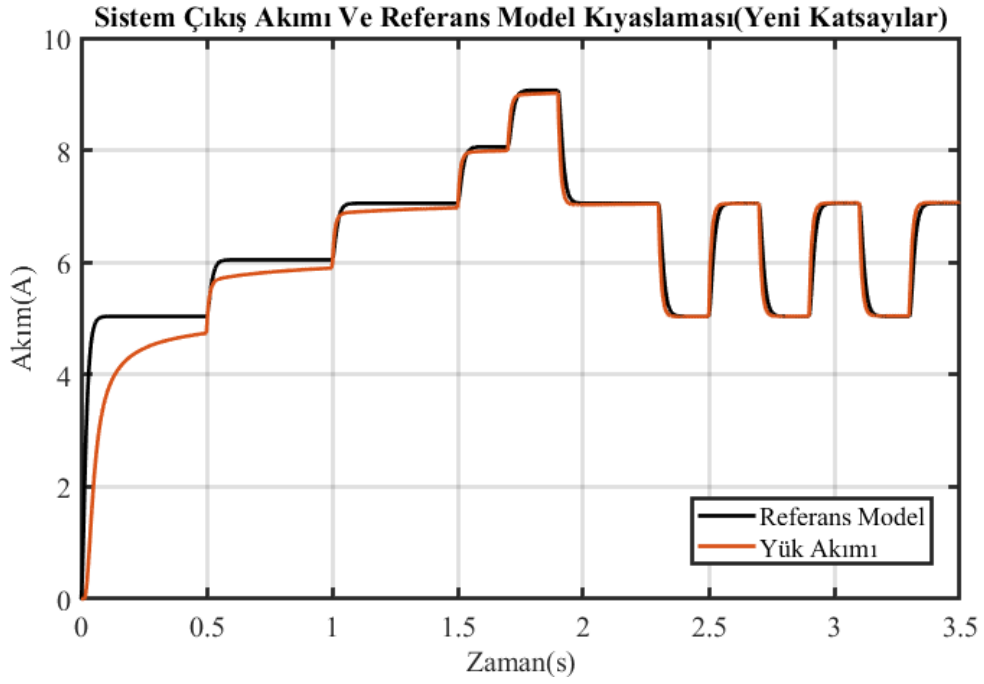


Resim 3.25 Dzeltme katsayısı +100 ile -100 aralıęında hata ıktısı

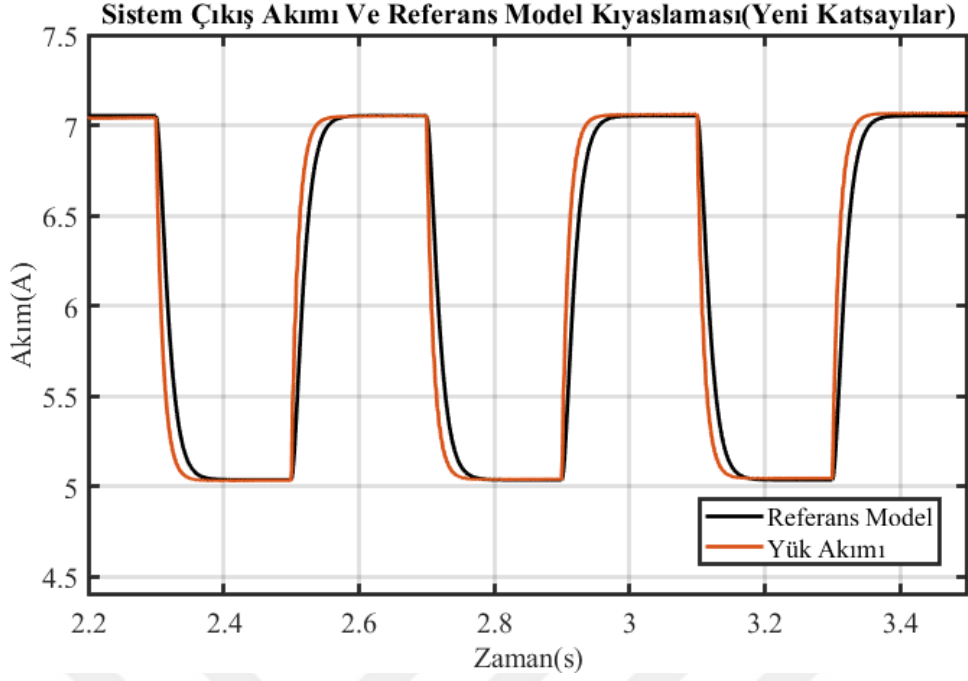
Resim 3.25 ile Resim 3.18 kıyaslandıęında +100 ile -100 aralıęındaki dzeltme katsayıları kullanıldıęında toplam hata miktarı daha artmaktadır. Bu artıřın sebebi sistemin geici hal durumudur. Sistemdeki endktans ve kondansatr zamanla dolan ve zamanla bořalan zellięe sahip oldukları iin toparlanma sresi deęiřiklik gstermektedir. Optimizasyon bu doldur bořalt olayını dengelerken yeni katsayılarda bu iři yapması biraz daha zorlařmıřtır. Bu yeni parametreler ile devre alıřtırıldıęında, devre alıřmasındaki yaklařık 1 saniyelik sre kontrolcnn devreyi nasıl daha kt toparladıęı grlmektedir.



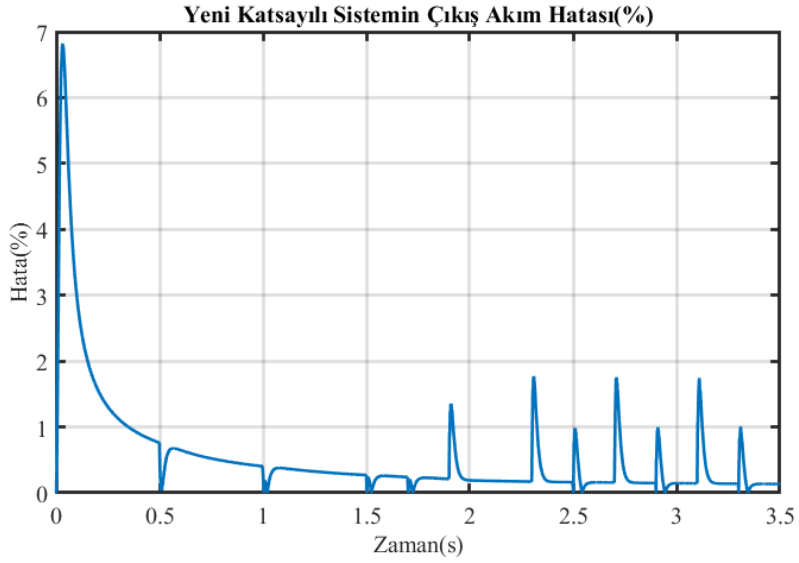
Resim 3.26 Yeni katsayılarla yük çıkış değerleri



Resim 3.27 Sistem çıkış akımını ve referans model kıyaslaması yeni katsayılar



Resim 3.27 Sistem çıkış akımını ve referans model kıyaslaması yeni katsayılar(yaklaştırılmış)



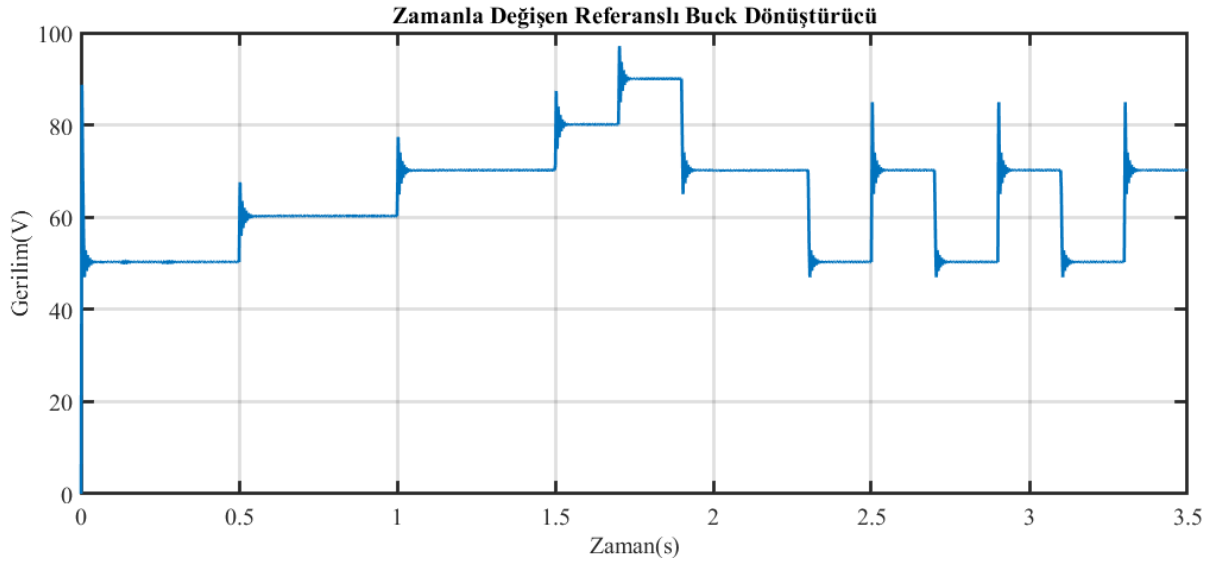
Resim 3.28 Yeni katsayılı sistem çıkış akım hatası(%)

Resim 3.28'den görüldüğü üzere ilk çalışma hatasını kontrol sisteminin sıfırlaması daha uzun sürmektedir. Bu 1 saniyelik kayıp gerçek bir devrede beklenmedik girişimlerin sistem sürekliliğini ve kararlılığını bozmasını engeller. Bu bağlamda yeterli fiziksel koruması olmayan sistemlerde katsayı aralığı azaltılarak sistem kararlılığı korunabilir.



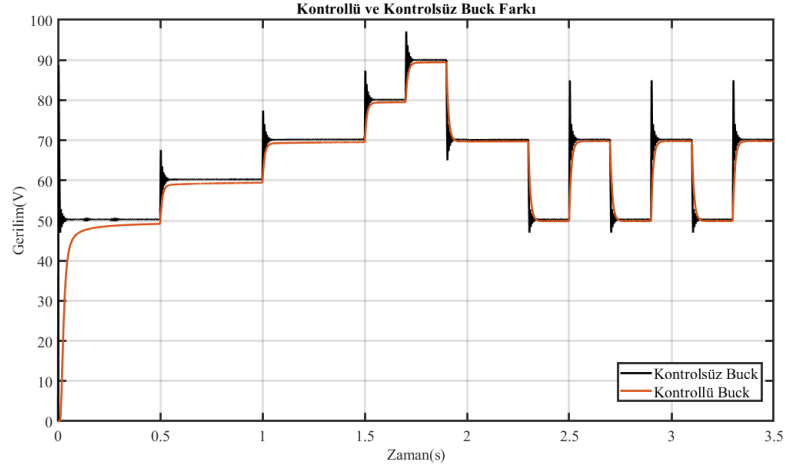
4. BULGULAR VE TARTIŞMA

Buck tipi dönüştürücü kontrolsüz şekilde kullanılmak istenirse gerekli hesaplamalar yapıp ihtiyaca uygun endüktans ve kondansatör devrede kullanılır. Bu sayede tek tip devre dinamiği ile değişimsiz düz bir çıkış almak mümkün. Lakin çıkışa farklı zamanlarda farklı yük değerleri vermek istenirse yada yük kendi doğal çalışması olarak farklı yük akımlarına ihtiyaç duyarsa normal bir buck dönüştürücü iş görmez hale gelmektedir.

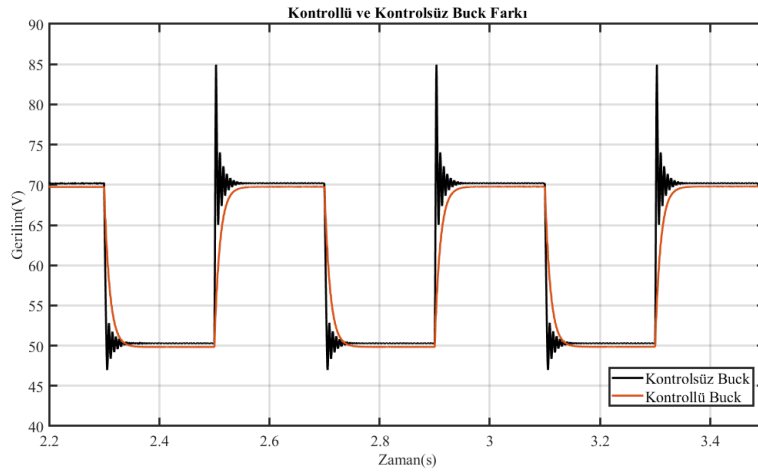


Resim 4.1 Zamanla değişen referansa buck dönüştürücü tepkisi

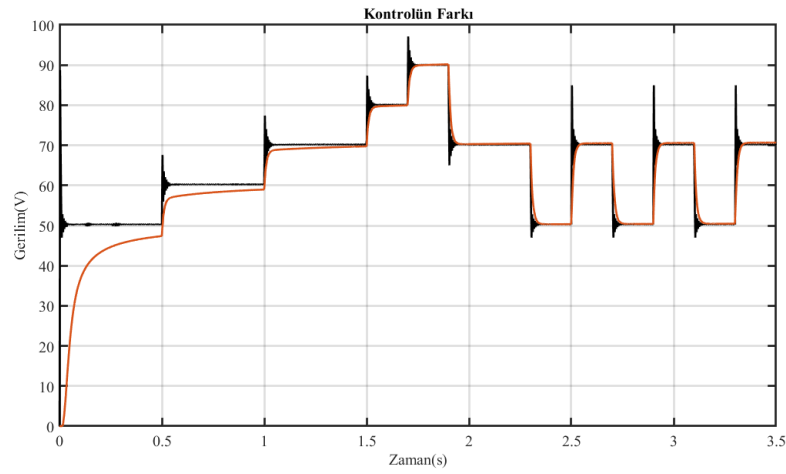
Resim 4.1'den görüldüğü üzere zamanla değişen referans değerlere kontrolsüz bir Buck dönüştürücünün verdiği tepki devreye fiziksel olarak hasar verecek derecede harmoniklere sahiptir. Referans değişimi esnasında devre istenilen gerilimden çok daha yüksek değerlerde gerilimi devreye verip toparlamaya çalışmaktadır lakin bu toparlamada bile dalgalanmalar istenilen değer altına ve üstüne çıkmaktadır. Bu bağlamda yapılan bu çalışmada MIT Kurallı MRAC ile kontrol edilen Buck dönüştürücü daha düzgün, harmonikleri yok denecek kadar az ve sistemi kararsızlığa sürüklemeyecek şekilde kontrolü sağlamaktadır.



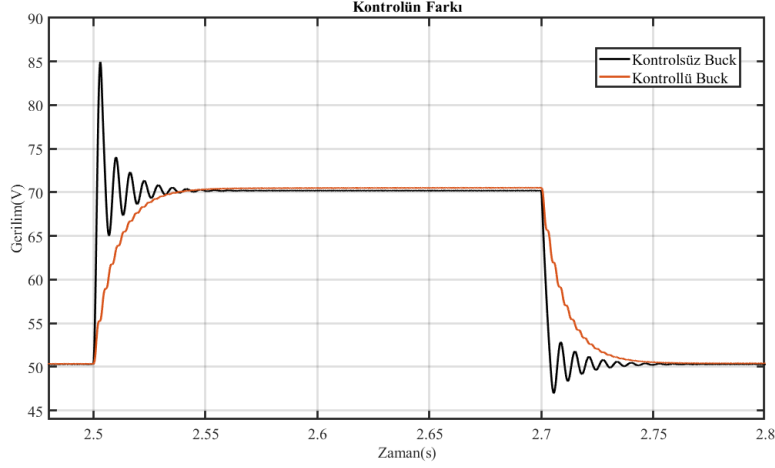
Resim 4.2 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları 1



Resim 4.3 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları(yakınlaştırılmış) 1



Resim 4.4 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları 2



Resim 4.5 Kontrollü ve kontrolsüz buck dönüştürücü sonuçları(yakınlaştırılmış) 2

Resim 4.2 ve Resim 4.3'teki grafiklerde kontrolü yapılan dönüştürücünün düzeltme katsayıları +1000 ile -1000 aralığında optimizasyonu yapılmış halidir. Resim 4.4 ve Resim 4.5 ise düzeltme katsayıları +100 ile -100 aralığında optimizasyonu yapılmış halidir. Düzeltme katsayısı büyük olması simülasyon sonuçlarında ilk çalışma anındaki toparlamayı daha çabuk yapıyor olmasına karşın simülasyonda sanallaştırılamayan girişimler gerçekte olursa devre kararsızlığa gitmesi söz konusu olacaktır. Katsayıları +100 ile -100 arasında optimize edilen sistem ise dışarıdan bir girişim olsa dahi bunu biraz daha kontrol altında tutabilecek seviyede bir kontrol sağlar. Her 2 durum içinde karalı hal durumlarına bakıldığında, ki bu yakınlaştırılmış grafiklerden görülmektedir, devre referansa uygun 70V ve 50V aralığında aşım yapmaksızın hareketini sürdürmektedir. Kontrolü yapılmayan dönüştürücüde ise 50V'dan 70V seviyesine çıkarken yaptığı dalgalanmayı 70V'dan 50V seviyesine inerken de yapmaktadır. Bu 4 grafikten MIT Kurallı MRAC kontrolünün Buck dönüştürücü üzerindeki kontrol başarısı görülmektedir.



5. SONUÇ VE ÖNERİLER

Bu çalışmada Buck tipi bir dönüştürücünün MIT Kuralı'yla tasarlanmış MRAC ile kontrolü sağlanmıştır. Buck dönüştürücü devresi matematiksel olarak modellenip bir algoritmaya çevrilmiştir. Daha sonra bu model MIT Kurallı MRAC tasarımıyla birleştirilmiş ve sistem bütünüyle modellenmiştir. Modellenen bütün sistem matematiksel algoritma olarak yazılmış ve MRAC içindeki düzeltme parametrelerini bulmak amaçlı optimizasyona tabi tutulmuştur. MRAC düzeltme parametreleri olağanın aksine tek değil iki parametre buldurulmuş ve bu iki değerlerin ortalaması sistemde yerine konmuştur. Optimizasyon algoritmasının bulunduğu değerler yine Simulink ortamında hazırlanan sistemde yerlerine atanış ve sonuçlar verilmiş ve yorumlanmıştır. MIT Kuralıyla tasarlanmış MRAC'ın düzeltme parametreleri optimizasyon algoritmasında buldurulurken farklı değer aralıklarının ne gibi sonuçlara yol açacağı, avantajları ve dezavantajları gösterilmiştir. Sonuç olarak kontrolsüz bir Buck dönüştürücü istenilen değerlerin üzerine çıkarak ve yoğun dalgalanmalı bir sinyal şeklinde tepki vermektedir. Bu çalışmada bu soruna bir çözüm önerilmiş ve MIT Kurallı MRAC kontrolcü tasarımının başarısı gösterilmiştir.

Bu çalışmayı geliştirmek amaçlı MIT Kuralı'yla tasarlanmış MRAC kontrolcüsüyle farklı dönüştürücü tipleri kontrol edilmeye çalışılabilir. Bu çalışmadaki sistem tümüyle farklı tip optimizasyon yöntemlerine tabi tutulabilir. Optimizasyondaki değer aralıkları daha farklı seçilip değerlendirilebilir. Matematiksel algoritmaya dökülmüş sistemin algoritma akışı geliştirilebilir ya da kısaltılabilir.

KAYNAKLAR

- Rasid, M.H.** (2001). Power Electronic Handbook. University Of Florida Florida:Academic Press(s. 211-215).
- Barghandan, S., DaeiFarshchi, A.** (2018).Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System. 2018, s. 7-9.
- Karaboga, D.** (2005). An Idea Based On Honey Bee Swarm For Numerical Optimization. Erciyes University, Engineering Faculty Computer Engineering Department Kayseri/Türkiye.
- Swarnkar, P., Jain, S. ve Nema, R.K.** (2010). “Effect of Adaptation Gain on system Performance for Model Reference Adaptive Control Scheme using MIT Rule”, *International Journal of Electrical and Computer Engineering* , s. 1-2.
- Duvanov, E.S., Kudinov, Y.I., Pashchenko, F.F. ve Ponomarev, A.A** (2020). Analysis and Synthesis of the Modified MRAC-MIT System and the MRAC-Lyapunov System. 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, Kas. 2020, ss. 521-526. doi: 10.1109/SUMMA50634.2020.9280614.
- Garikayi, T., Nyanga, L., Mushiri, T., Mhlanga, S., Muzinda, A. ve Mutangi, T.** (2013) Model Reference Adaptive Control System For Moisture Regulation In Cotton Ginning”, s. 19, 2013.
- Nocedal, J., Wright, S. J.** (2006). Numerical Optimization, New York: Springer.
- Winston, W.L. ve Goldberg, J.B.,** (2004). Operations Research: Applications And Algorithms. Belmont: Thomson Brooks/Cole. 4th Edition. ISBN:0534423620.
- Campo, A.B.** (2012). PID Control Design. İçinde MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1. InTech. DOI: 10.5772/2557. ISBN: 978-953-51-0750-7.
- Maiti, D., Acharya, A., Chakraborty, M., Konar A. ve Janarthanan R.,** (2008). Tuning PID and Fractional PID Controllers using the Integral Time Absolute Error Criterion. İçinde Proceedings of the 2008 4th International Conference on Information and Automation for Sustainability, ICIAFS 2008. (ss. 457–462).
- Karaboga D., Akay B.,** (2009). Artificial Bee Colony (ABC). Harmony Search and Bees Algorithms on Numerical Optimization. IPROMS 2009 Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.