

**DA/DA DÖNÜŞTÜRÜCÜLER İÇİN SEZGİSEL TABANLI Pİ, PD VE
PID DENETLEYİCİLİ BİLGİSAYAR ARAYÜZÜ PROGRAMININ
GERÇEKLEŞTİRİLMESİ**

Ahmet IRMAK

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

MART 2013

ANKARA

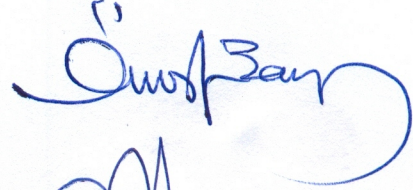
Ahmet IRMAK tarafından hazırlanan DA/DA DÖNÜŞTÜRÜCÜLER İÇİN SEZGİSEL TABANLI PI, PD ve PID DENETLEYİCİLİ BİLGİSAYAR ARAYÜZÜ PROGRAMININ GERÇEKLEŞTİRİLMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.



Yrd. Doç. Dr. İsmail ATACAK

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Elektronik Ve Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : Prof. Dr. Ömer Faruk BAY



Üye : Doç. Dr. M. Rahmi CANAL



Üye : Yrd. Doç. Dr. İsmail ATACAK

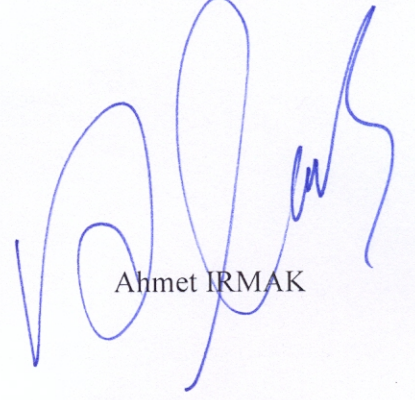


Tarih : 05/03/2013

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

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



Ahmet IRMAK

**DA/DA DÖNÜŞTÜRÜCÜLER İÇİN SEZGİSEL TABANLI PI, PD VE PID
DENETLEYİCİLİ BİLGİSAYAR ARAYÜZÜ PROGRAMININ
GERÇEKLEŞTİRİLMESİ
(Yüksek Lisans Tezi)**

Ahmet IRMAK

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

Mart 2013

ÖZET

Bu çalışmada, DA/DA dönüştürücülerin farklı çalışma şartları altında kapsamlı analizlerine olanak sağlayan yeni bir tool geliştirilmiştir. Geliştirilen tool; temel DA/DA dönüştürücü topolojilerini, temel/gelişmiş kontrol tekniklerini, ayrık zamanlı denetleyicileri, sezgisel tabanlı denetleyicileri ve doğrusal/doğrusal olmayan yük tiplerini kapsamlı bir grafik ara yüzü ile bileştiren tek bir ekrandan oluşur. Dönüştürücülerin zaman domenindeki performansları tool üzerinden nominal durum ve yük değişim analiz modları kullanılarak yapılabilir ve bu performanslar grafiksel ve numerik olarak ekranda görüntülenebilir. Analizlerden elde edilen grafikler rapor hazırlama amaçlı farklı resim/text formatlarında kaydedilebilir. Gerçekleştirilen tool aynı zamanda simülasyonla ilgili tüm parametreleri kullanıcıya kolayca değiştirebileceği esnek bir yapı sunar.

Bilim Kodu : 702.1.012
Anahtar Kelime : DA/DA dönüştürücü, PI, PD ve PID denetleyici, genetik algoritma, parçacık sürü optimizasyonu, yapay arı algoritması.
Sayfa Adedi : 76
Tez Yöneticisi : Yrd. Doç. Dr. İsmail ATACAK

**REALIZATION OF A COMPUTER INTERFACE PROGRAM WITH
HEURISTIC-BASED PI, PD AND PID CONTROLLER FOR DC/DC
CONVERTERS
(M.Sc. Thesis)**

Ahmet IRMAK

**GAZİ UNIVERSITY
INFORMATICS INSTITUTE**

March 2013

ABSTRACT

In this study, a new tool is developed to perform the comprehensive analyses of DC/DC converters under various operating conditions. The developed tool consists of one screen integrated main converter topologies, basic/advanced control techniques, discrete-time controllers, heuristic-based controller and linear/nonlinear loads with friendly and easy-to-use graphical interface. The converter performances in time domain can be analyzed through the tool by using nominal case analysis and voltage/load change analysis modes, and these performances can be observed both graphically and numerically on the screen. The graphics obtained from the analyses can be saved in different graphical/text file format for the reporting activities. Also, it has a flexible structure that user can change easily all parameters used in the simulations.

Science Code	: 702.1.012
Key Words	: DC/DC Converter, PI, PD and PID controller, genetic algorithm, particle swarm optimization algorithm, artificial bee algorithm.
Page Number	: 76
Advisor	: Asist. Prof. Dr. İsmail ATACAK

TEŞEKKÜR

Bu tez çalışması boyunca tevazu ve çalışkanlığıyla bana örnek olan, destek ve teşviklerini hiçbir zaman esirgemeyen danışmanım Yrd. Doç. Dr. İsmail ATACAK'a teşekkürlerimi arz ederim.

Ayrıca katkılarından dolayı değerli meslektaşım Engin CEYLAN'a ve Arş.Gör. Bayram KÜÇÜK'e teşekkür ederim.

Bu çalışma süresince gösterdiği anlayış ve desteklerinden dolayı âlembahâ eşime, manevi kuvvet vesilelerim başta çocuklarım olmak üzere anneme, babama ve kardeşlerime teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. DA/DA GÜÇ DÖNÜŞTÜRÜCÜLER	3
2.1. Doğrusal Güç Dönüştürücüler.....	3
2.2. Anahtarlama Mod Güç Dönüştürücüler	4
2.2.1. İzolesiz tip DA/DA güç dönüştürücüler	5
2.2.2. İzoleli tip DA/DA dönüştürücüler	14
2.2.3. DA/DA dönüştürücülerin denetimi.....	17
3. SEZGİSEL TABANLI OPTİMİZASYON ALGORİTMALARI.....	19
3.1. Parçacık Sürü Optimizasyonu	19
3.2. Yapay Arı Kolonisi Algoritması	21
3.3. Genetik Algoritma	25
3.4. Karınca Kolonisi Algoritması	33
3.5. Diferansiyel Gelişim Algoritması	34
4. DA/DA DÖNÜŞTÜRÜCÜLER İÇİN SEZGİSEL TABANLI PI, PD VE PID DENETLEYİCİLİ BİLGİSAYAR ARAYÜZÜ PROGRAMININ GERÇEKLEŞTİRİLMESİ	37
4.1. Kullanılan Güç Dönüştürücüler	37

Sayfa

4.2. Kullanılan Denetim Teknikleri.....	39
4.2.1. Gerilim mod kontrol	40
4.2.2. Tepe akım kontrol.....	40
4.2.3. Ortalama akım kontrol	41
4.2.4. Vadi akım kontrol	42
4.2.5. V^2 akım kontrol.....	43
4.2.6. Histerisiz kontrol.....	43
4.3. Kullanılan Sezgisel Tabanlı Denetleyiciler	44
4.4. Yazılımın Geliştirilmesi	49
4.5. Gerçekleştirilen Bilgisayar Ara Yüzü Yazılımı	50
4.6. Bilgisayar Ara Yüzü Yazılımı ile Yapılmış Benzetim Çalışmaları	64
5. SONUÇ VE ÖNERİLER	70
KAYNAKLAR	71
ÖZGEÇMİŞ	76

ÇİZELGELERİN LİSTESİ**Sayfa**

Çizelge 4. 1. Denetim tekniklerinin performans değerleri (sabit yük).....	64
Çizelge 4. 2. Denetim tekniklerinin performans değerleri (değişken yük)	66
Çizelge 4. 3. Denetleyicilere göre performans değerleri.....	67
Çizelge 4. 4. Sezgisel algoritmalara göre performans değerleri	68

ŞEKİLLERİN LİSTESİ

Sayfa

Şekil 2. 1. Doğrusal güç dönüştürücülerinin genel yapısı.....	3
Şekil 2. 2. Gerilim düşüren dönüştürücünün güç devresi	6
Şekil 2. 3. Gerilim düşüren dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri.....	7
Şekil 2. 4. Gerilim yükselten dönüştürücünün güç devresi.....	8
Şekil 2. 5. Gerilim yükselten dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri.....	9
Şekil 2. 6. Gerilim düşüren/yükselten dönüştürücünün güç devresi.....	10
Şekil 2. 7. Gerilim düşüren/yükselten dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri	11
Şekil 2. 8. CÜK dönüştürücünün güç devresi	12
Şekil 2. 9. SEPIC dönüştürücünün güç devresi	13
Şekil 2. 10. Flyback DA/DA dönüştürücü güç devresi.....	14
Şekil 2. 11. Forward dönüştürücü güç devresi	15
Şekil 2. 12. Yarım köprü dönüştürücü	16
Şekil 2. 13. Tam köprü dönüştürücü	16
Şekil 2. 14. Push-Pull dönüştürücü	17
Şekil 3. 1. Rulet çember yöntemi	30
Şekil 4. 1. Gerilim mod kontrol tekniği	40
Şekil 4. 2. Tepe akım kontrol tekniği	41
Şekil 4. 3. Ortalama akım kontrol tekniği	42
Şekil 4. 4. Vadi akım kontrol tekniği	42
Şekil 4. 5. V^2 kontrol tekniği.....	43
Şekil 4. 6. Histerisiz akım kontrol tekniği.....	44

Sayfa

Şekil 4. 7 Başlangıç popülasyonu	47
Şekil 4. 8. Yazılımın setup dosyaları	51
Şekil 4. 9. Kurulum ekranı	51
Şekil 4. 10. Başlat menüsü kısayolu	52
Şekil 4. 11. Masaüstü kısayolu.....	52
Şekil 4. 12. Yazılım çalıştığında gelen açılış penceresi	52
Şekil 4. 13. Simple Multi-Plot türü çizim örneği	55
Şekil 4. 14. Multi-Y Coordinates türü çizim örneği	55
Şekil 4. 15. Grafik üzerinde sağ tuş yapıldığında açılan menü.	57
Şekil 4. 16. Verilin aktarıldığı bir txt dosyası örneği.....	58
Şekil 4. 17. V_{in} için değişim parametrelerini ayarlama penceresi.....	59
Şekil 4. 18. Algoritmanın uygulanacağı denetleyici seçimi.....	60
Şekil 4. 19. Yük değişimi parametreleri için ayar penceresi.....	60
Şekil 4. 20. Normal durum analizi için performans parametreleri.....	61
Şekil 4. 21. Giriş gerilimi veya yük değişimi analiz parametreleri.....	62
Şekil 4. 22. Ön değer (defaults) kaydetme ve çağırma seçenekleri	63
Şekil 4. 23. Karşılaştırmada kullanılan çalışma şartları	64
Şekil 4. 24. Karşılaştırmada kullanılan devre elemanları	64
Şekil 4. 25. Denetim tekniklerinin çıkış değerleri (sabit yük)	65
Şekil 4. 26. Denetim tekniklerinin çıkış değerleri (sabit yük)	66
Şekil 4. 27. Denetleyicilere göre çıkışların grafiksel gösterimi	67
Şekil 4. 28. Sezgisel algoritmalar ile elde edilen çıkışların grafiksel gösterimi	69

SİMGELER VE KISALTMALAR

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

Simgeler	Açıklama
accdb	MS Access 2010 veri tabanı dosyası uzantısı
Bmp	Resim dosyası uzantısı (Nokta haritası resmi)
Emf	Resim dosyası uzantısı (Gelişmiş Meta Dosyası)
Gif	Resim dosyası uzantısı (Değişebilir Grafik formatı)
jpg	Resim dosyası uzantısı (Birleşik Fotoğraf Uzmanları Grubu)
Png	Resim dosyası uzantısı (Taşınabilir Ağ Grafiği)
sln	MS Visual Studio çözüm dosyası
Tiff	Resim dosyası uzantısı (Etiketlenmiş Resim dosyası formatı)
txt	Metin dosyası uzantısı
V_{in}	Giriş gerilimi

Kısaltmalar	Açıklama
C	Kondansatör kapasitansı
D	Diyot
DA	Doğru Akım
DA/DA	Doğru Akımdan Doğru Akıma
Da Motor	Doğru Akım Motoru
DGA	Diferansiyel Gelişim Algoritması

Kısaltmalar	Açıklama
DGM	Darbe Genlik Modülasyonu
DLL	Dinamik Bağlantı Kütüphanesi
DNA	Deoksiribo Nükleik Asit
GA	Genetik Algoritma
Kd	Türevsel Denetleyici Katsayısı
Ki	İntegral Denetleyici Katsayısı
KKA	Karınca Kolonisi Algoritması
Kp	Oransal Denetleyici Katsayısı
L	Bobin indüktansı
MSSQL	Microsoft Yapılandırılmış Sorgu Dili
P	Oransal
PD	Oransal ve Türevsel
PI	Oransal ve Integral
PID	Oransal, Integral ve Türevsel
PSO	Parçacık Sürü Optimizasyonu
R	Direnç
RC	Direnç – Kondansatör
RL	Direnç – Bobin
T	Peryot
V	Volt
YAK	Yapay Arı Kolonisi

1. GİRİŞ

Son yıllarda anahtarlama mod doğru akımdan doğru akıma (DA/DA) dönüştürücülerin tasarımına olan ihtiyaç, onların uzay araçlarından değişik tüketici ürünlerine kadar birçok uygulamada çok yaygın kullanımından dolayı gittikçe artmaktadır [1]. Genel olarak bu dönüştürücüler farklı topolojilere sahip güç devreleri ve bunların denetimine olanak sağlayan kontrol ünitelerinden oluşurlar. Bu yüzden DA/DA dönüştürücülerin tasarımı bahsedilen bu iki bileşen üzerinden yerine getirilir. Dönüştürücü tasarımında; kullanıldıkları uygulama içerisindeki ihtiyaç duyulan güç ve regülasyon gereksinimlerini yerine getirebilmesi için, kontrol ünitesi içerisindeki denetim teknikleri ve denetleyicilerin seçim ve tasarımının önemli bir rolü vardır. Geçmişten günümüze DA/DA dönüştürücülerin denetimi için gerilim mod kontrol, akım mod kontrol, histerisiz kontrol, vadi akım kontrol ve V^2 kontrol gibi çok sayıda gerilim ve akım tabanlı analog ve sayısal kontrol tekniği önerilmiş ve onların farklı uygulamalar içerisindeki performansları analiz edilmiştir [2–10]. Güç sistemi uygulamalarında, bu tekniklerin içerisinde yapılarının basit ve tasarımlarının kolay olmaları sebebi ile PI, PD ve PID denetleyiciler yaygın olarak kullanılırlar. Bu denetleyiciler kazanç parametreleri iyi bir şekilde ayarlandığında birçok sistemin denetiminde iyi bir performans sergilese de, aynı performansı farklı çalışma şartları altında DA/DA dönüştürücülerde sağlayamazlar. Bunun nedeni bu dönüştürücülerin doğrusal olmayan karakteristiklerinden dolayı denetleyici kazanç parametreleri değerlerinin ayarlanmasındaki zorluğa dayanır. Ziegler-Nichols ve Cohen-Coon gibi kurallar PI, PD ve PID denetleyicilerin kazanç parametrelerini ayarlayabilmek için kullanılan genel metotlardır. Bununla birlikte, denetleyiciler ayar işleminde denetlenen sistem dinamik bilgisine ihtiyaç duydukları için denetleyicinin performansı üzerinde arzu edilen etkiyi sağlayamazlar [11–13]. Bu yüzden sistemin çalışma şartlarına göre kendi kazanç parametrelerini çevrim içi olarak kendi kendine ayarlayabilen bir denetleyici geliştirmek bu problemin çözümüne önemli katkılar sağlayacaktır.

Şu ana kadar farklı alanlarda optimizasyon problemlerini çözümü üzerine, genetik algoritma (GA), karınca kolonisi algoritması (KKA), parçacık sürü optimizasyon

algoritması (PSO), diferansiyel gelişim algoritması (DGA) ve yapay arı kolonisi algoritması (YAK) gibi sezgisel algoritmaları içeren çok sayıda çalışma yapılmıştır. Bu çalışmalardan özellikle PI ve PID denetleyici kazanç parametrelerinin optimizasyonu üzerine yapılmış olanlar, genetik ve parçacık sürü optimizasyon algoritmalarının dinamik ve doğrusal olmayan sistemlerin optimizasyon sürecinde daha hızlı bir çözüme ulaştıkları için diğer algoritmalarından daha başarılı sonuçlar elde ettiğini göstermektedir [14–23]. Bu yüzden her iki algoritmada birçok kontrol probleminin çözümünde yaygın bir şekilde kullanılmaktadırlar.

Bu çalışmada, DA/DA dönüştürücülerin denetimi için sezgisel tabanlı ayırık zamanlı bir PI, PD ve PID denetleyici toolbox' ı geliştirilmiştir. Geliştirilen toolbox; gerilim düşüren, gerilim yükselten, gerilim düşüren/yükselten, CÜK ve SEPIC dönüştürücü topolojilerini, gerilim ve akım tabanlı kontrol tekniklerini, ayırık zamanlı PI, PD ve PID denetleyicileri, sezgisel tabanlı denetleyicileri ve dönüştürücü çıkışında kullanılan doğrusal ve doğrusal olmayan yükleri içeren tek bir ara yüzden oluşur. Yük olarak R, RC, RL, Evirici ve DA Motor olmak üzere farklı yük kombinasyonlarında bu dönüştürücülerin çalışmasına olanak sağlar. Bu haliyle toolbox, DA/DA dönüştürücülerin farklı şartlar altında denetleyiciler ve kontrol tekniklerinin sistem performansı üzerindeki etkilerini incelenme ve analiz etmeye önemli bir katkı sağlar.

Bu çalışma giriş bölümünden sonra dört bölümden oluşmaktadır. Bu bölümden sonra gelen ikinci bölümde DA/DA Güç Dönüştürücüler hakkında Doğrusal ve Anahtarlama mod güç dönüştürücüler olmak üzere iki ana başlık altında bilgi verilmiştir. Üçüncü bölümde uygulamalarda sıkça kullanılan sezgisel tabanlı optimizasyon tekniklerinden beş tanesi çalışma algoritmalarıyla birlikte anlatılmıştır. Dördüncü bölümde tool ile ilgili temel bileşenlerin tasarımı, tool'un tanıtımı ve tool üzerinden yapılan analizler sunulmaktadır. Beşinci ve son bölümde ise sonuç ve önerilere yer verilmiştir.

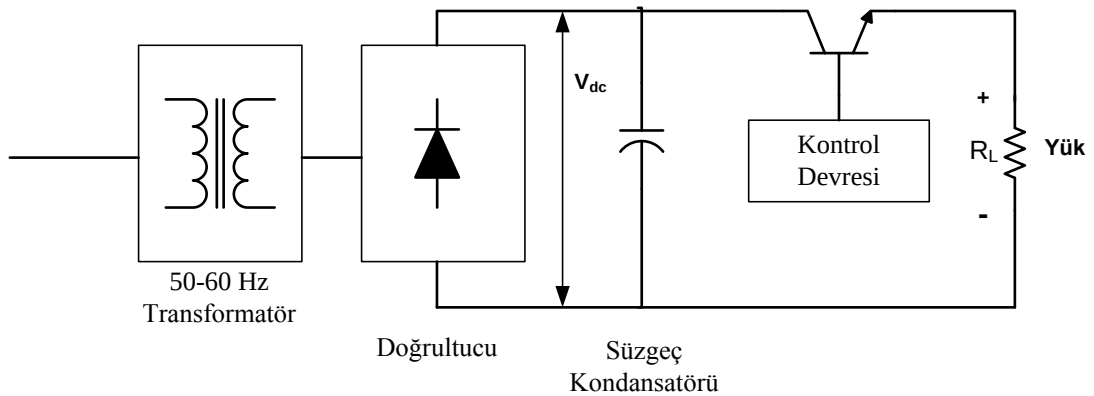
2. DA/DA GÜÇ DÖNÜŞTÜRÜCÜLER

DA/DA Dönüştürücüler yüksek frekansta çalışan, DA gerilimden farklı seviyede yine DA gerilimi üreten güç elektroniği devreleridir. Yüksek frekanslarda çalışması fiziksel özellikleri açısından hafif ve küçük olarak tasarlanabilmelerine imkân verir. Kolay uygulanabilir olmasını da sağlayan bu özellik onların yaygın bir şekilde kullanımını da beraberinde getirmiştir.

DA/DA dönüştürücüleri genellikle iki tipe ayrılır: Doğrusal güç dönüştürücüleri ve anahtarlama mod güç dönüştürücüleri. Sayısal sistemlerin minyatürleşme ve daha büyük akım-daha az güç kaybı ihtiyacı nedeniyle, doğrusal güç dönüştürücüleri yerini anahtarlama mod güç dönüştürücüleri bırakmaya başlamıştır.

2.1. Doğrusal Güç Dönüştürücüleri

Doğrusal güç dönüştürücülerinin genel yapıları Şekil 2.1’de verilmiştir. Bu dönüştürücüler hat gerilimini hat frekansında çalışan bir transformatör ile istediğimiz gerilim seviyesine düşürür. Düşürülen bu gerilim diyotlar yardımıyla doğrultulduktan sonra süzgeç kondansatörüne uygulanır. Seri bağlı anahtar ile yüke verilir. Seri anahtar olarak genellikle bir transistör kullanılır. Burada transistör ayarlı bir direnç gibi kullanılarak istenilen çıkış gerilimi elde edilir. Transistör aktif bölgede çalıştığı için üzerinde güç kayıpları oluşur [24–26].



Şekil 2. 1 Doğrusal güç dönüştürücülerinin genel yapısı

Bu yapıda kullanılan transformatörün düşük frekansta çalışmasından dolayı, boyutu ve ağırlığı oldukça büyüktür. Doğrusal güç dönüştürücülerinin diğer önemli bir eksiği ise seri bağlı eleman üzerinde oluşan kayıptır. Bütün yük akımı seri bağlı eleman üzerinden geçtiği için burada oluşan güç kaybı $(V_d - V_o) \times (I_o)$ 'dır. Minimum fark bir NPN transistörde yaklaşık 2,5 V' tur. AC giriş gerilimi arttığında seri eleman üzerindeki kayıp artar ve dolayısıyla bu eleman üzerindeki sıcaklık artar. Bu güç dönüştürücüsünün verimini azaltır.

2.2. Anahtarlama Mod Güç Dönüştürücüleri

Devrede, aktif ve pasif olmak üzere iki anahtarlama elemanı olan dönüştürücü tipleridir. Aktif anahtar dışarıdan uygulanan bir sinyal ile kontrol edilir. Pasif anahtar ise devrenin çalışması sırasında, çalışma şartlarına bağlı olarak konum değiştiren bir anahtardır. Aktif anahtar olarak transistör, IGBT vb. bir devre elemanı kullanılabilir. Uygulanan harici sinyal ile bu eleman iletime ya da kesime sokulabilir. Anahtarlama ile elde edilen darbeleme oranı girişteki DA geriliminden elde edilecek çıkış DA gerilimini belirler. Buradaki darbeleme oranı:

$$D = \frac{t_k}{t_a + t_k} = \frac{t_k}{T} = t_k \cdot f_a \quad (2.1)$$

ile hesaplanır. Burada t_a anahtarın açık kaldığı süreyi, t_k ise anahtarın kapalı olduğu süreyi gösterir.

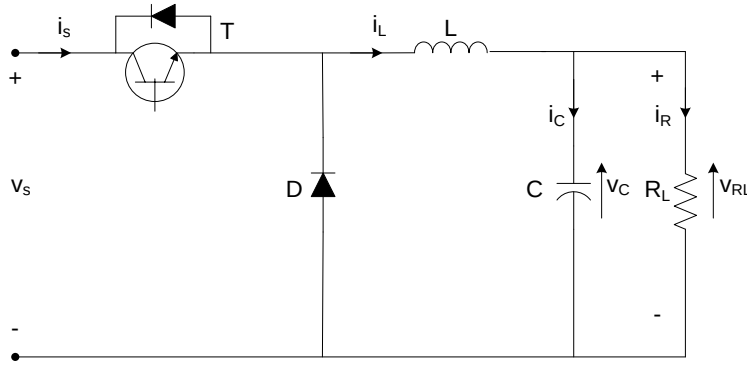
Bu anahtarlama davranışından dolayı çıkış gerilimlerine eklenmiş gürültüler oluşur. Çıkış gerilimini kontrol etmekte yaygın kullanılan metot darbe genişlik modülasyonu (DGM)' dur. Bu metot, sabit bir anahtarlama frekansında görev saykılının değiştirilmesi mantığına dayanır. Anahtarlama frekansı sabit olduğu için, modülasyon çıkış gerilimindeki gürültüyü hızlı bir şekilde azaltabilmek için basit bir alçak geçiren filtreye izin veren göreceli olarak dar bir gürültü spektrumuna sahip olur. Bu gereksinim dönüştürücü çıkışında bir bobin ve kondansatör kullanılarak oluşturulan alçak geçiren filtre ile yerine getirilir. Anahtarlama mod güç dönüştürücüleri aşağıdaki alt başlıklarda anlatıldığı gibi izoleli ve izolesiz olmak üzere iki kategoride ele alınırlar.

2.2.1. İzolesiz tip DA/DA güç dönüştürücüleri

Alt harmonik osilasyonları izole edilemediği için çıkışında sürekli osilasyon gözlenebilen dönüştürücü topolojisidir. Bununla birlikte yapılarının basit, analiz ve tasarımlarının kolay ve ebatlarının küçük olması sebebi ile özellikle düşük güç uygulamalarında yaygın olarak kullanılırlar.

Gerilim düşüren DA/DA dönüştürücüler

Gerilim düşüren (Buck) dönüştürücüler, giriş çıkış gerilimleri arasında $V_{giriş} > V_{çıkış}$ ilişkisi olan izolesiz tip dönüştürücü topolojisidir. Bu dönüştürücüler adından da anlaşıldığı gibi girişine uygulanan DA gerilimi istenilen referans gerilime düşürerek çıkışına aktarırlar. Yaygın olarak kullanıldığı yerler; regüleli DA güç kaynakları, DA motor hız denetim devreleridir. Şekil 2.2, saf bir direnci besleyen gerilim düşüren dönüştürücünün güç devresini göstermektedir. Bu yapısal olarak DA gerilim kaynağı, güç anahtarı, hızlı diyot, filtre bobini ve filtre kondansatöründen oluşmaktadır. Bobin ve kondansatör çıkış gerilimi üzerindeki dalgalanma miktarını belirleyen parametrelerdir. Bu parametrelerin değeri dönüştürücünün hangi modda çalışacağına karar verir. Şayet bobinin değeri yeterince büyük ise akım hiçbir zaman sıfır olmayacağı için dönüştürücü sürekli iletim modunda çalışır. Ters durumda ise dönüştürücünün çalıştığı mod süreksiz akım modu olarak isimlendirilir. Yani bir anahtarlama periyodu süresinde bobin üzerinden geçen akım sıfırdan başlar ve tepe değerine ulaşarak tekrar sıfıra düşer. Pratikte bu dönüştürücüler her iki durumda da çalıştırılabileceğinden analizleri her iki mod dikkate alınarak yapılır [25].



Şekil 2. 2. Gerilim düşüren dönüştürücünün güç devresi

Güç anahtarı beyzine bir tetikleme sinyali geldiği zaman iletme geçer. Akım devresini DA gerilim kaynağının (+) noktasından çıkıp bobin, kondansatör ve direnç üzerinden kaynağın (–) noktasına gelerek tamamlar. Güç anahtarının beyzine gelen tetikleme sinyali kesildiğinde ise transistör yalıtıma geçer ve bobin üzerinde indüklenen ters enerji ile D diyotu iletme geçer. Diyotun kısa devre olması ile kondansatör üzerindeki gerilim L bobini ve R yükü tarafından çekilen akımlardan dolayı boşalmaya başlar. Şekil 2.3’ de sürekli iletim modunda çalışan gerilim düşüren dönüştürücü için bobin üzerinde düşen gerilim ve akıma ait dalga şekilleri görülmektedir. Bu modda bobin akımı kesintisiz olarak akmaktadır. t_{on} süresince anahtar bobin akımı iletmekte olup, diyot ters polarmalanır. Bu, Şekil 2.3 ’de görüldüğü gibi bobin üzerinde $V_L = V_S - V_{RL}$ gibi pozitif gerilim düşmesine neden olur. Anahtar kesim durumuna geçtiği durumda ise bobinde depolanan enerjiden dolayı bobin akımı akmaya devam eder ve akım devresini diyot üzerinden tamamlayarak $V_L = V_{RL}$ negatif gerilimin oluşmasına sebep olur. Sürekli iletim modunda dalga şekli bir periyottan diğerine tekrar etmek durumunda olduğu için bobin geriliminin bir periyottaki integrali sıfır olur. $T_S = t_{on} + t_{off}$ olduğundan;

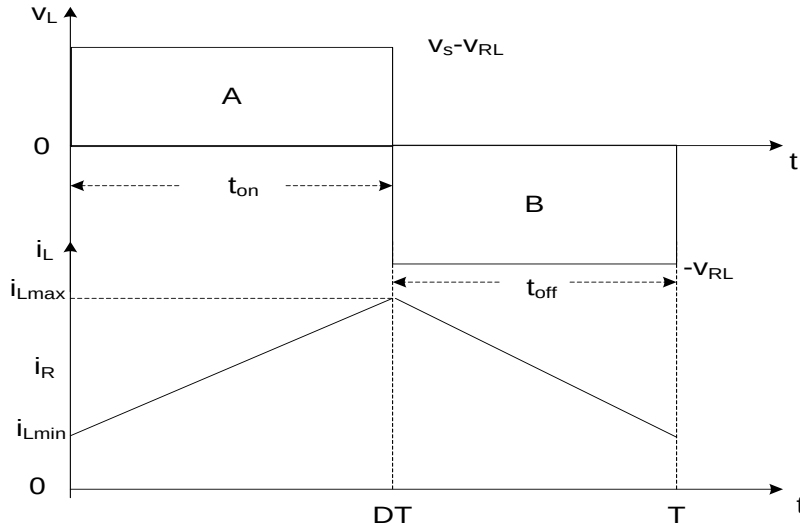
$$\int_0^{T_S} v_L . dt = \int_0^{t_{on}} v_L . dt + \int_{t_{on}}^{T_S} v_L . dt = 0 \quad (2.2)$$

Eş. 2.2’den Şekil 2.3 ’de gösterilen A ve B alanlarının birbirlerine eşit olduğu görülebilir.

$$(v_s - v_{RL}) \cdot t_{on} = v_{RL} \cdot (T_s - t_{on}) \Rightarrow \frac{v_{RL}}{v_s} = \frac{t_{on}}{T_s} = d \quad (2.3)$$

Eş. 2.3' den çıkış geriliminin anahtarın tetiklenme oranını ile doğrusal olarak değiştiği görülmektedir. Devre elemanları üzerindeki kayıplar ihmal edildiği durumda giriş gücü P_s çıkış gücü P_{RL} ya eşit olur. Bu eşitlikten yararlanarak elde edilen görev saykılı oranı D 'nin giriş ve çıkış akımları ile bağıntısını veren eşitlik aşağıda gösterilmektedir.

$$P_{RL} = P_s \Rightarrow V_s \cdot I_o = v_o \cdot i_o \Rightarrow \frac{i_R}{i_s} = \frac{v_s}{v_{RL}} = \frac{1}{d} \quad (2.4)$$

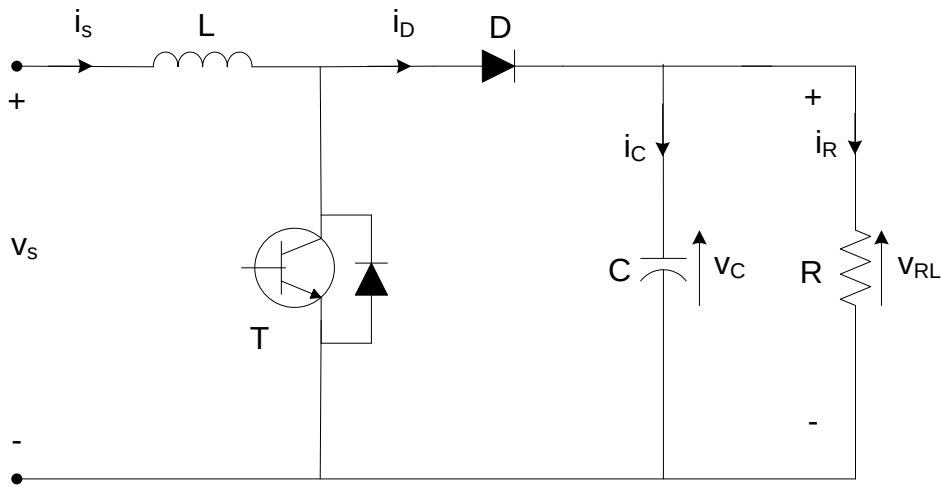


Şekil 2. 3. Gerilim düşüren dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri

Gerilim yükselten DA/DA dönüştürücüler

Düşük bir giriş geriliminden yüksek bir çıkış gerilimi elde etmeye olanak sağlayan izolesiz tip DA/DA dönüştürücü topolojisidir. Yapısı DA gerilim kaynağı, yükselme bobini, güç anahtarı, diyot, filtre kondansatörü ve saf dirençten oluşur. Şekil 2.4' de darbe genişlik modülasyonlu gerilim yükselten dönüştürücüye ait güç devresi

gösterilmektedir. Bu dönüştürücüler genel olarak çıkış gerilimi ayarlı DA güç kaynakları ile DA motorlarının enerji geri kazanımı frenlemesinde kullanılır. Transistörün beyzine herhangi bir tetikleme geldiği durumda transistör ilettime geçer ve diyot ters kutuplanır. Bu durumda giriş devresi çıkış devresinden ayrılır ve kaynak akımının tamamı bobin üzerinden geçer. Transistörün beyzine gelen tetikleme sinyali kesildiğinde ise transistör kesime gider. Bu durumda, çıkış hem kaynak hem de bobin üzerinde biriken enerji ile beslenmeye başlar [25].



Şekil 2. 4. Gerilim yükselten dönüştürücünün güç devresi

Sürekli iletim modunda bobin akımına ait dalga şekilleri Şekil 2.5’de göstermektedir. Kararlı durumda, bir zaman periyodunda bobin üzerindeki geriliminin zamana göre integrali sıfırdır.

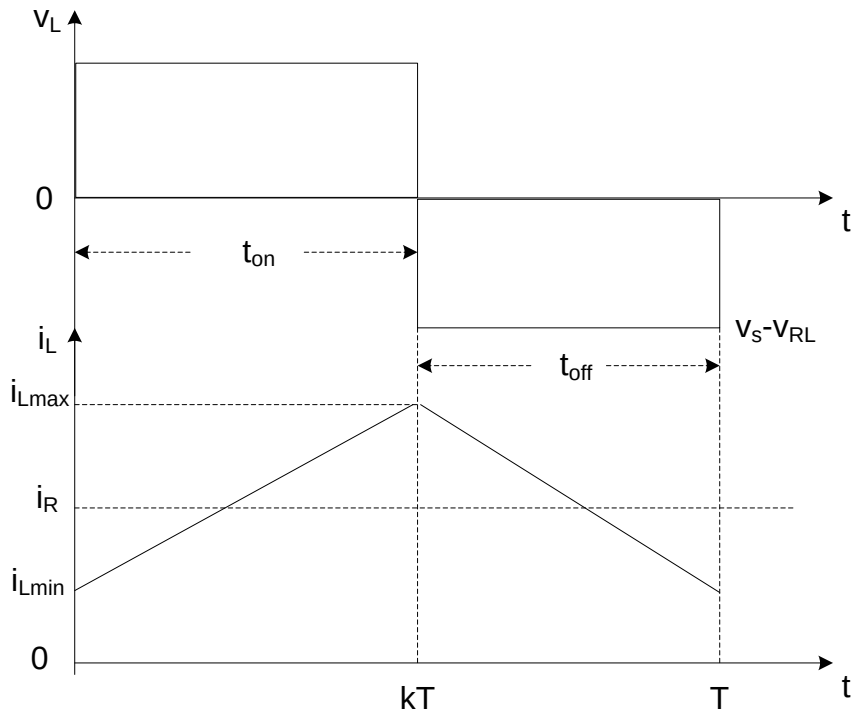
$$v_s \cdot t_{on} + (v_s - v_{RL})t_{off} = 0 \quad (2.5)$$

Her iki taraf T_s ile bölünerek eşitlik tekrar düzenlenirse çıkış ile görev süresi oranı arasındaki bağıntı aşağıdaki gibi elde edilir.

$$\frac{v_{RL}}{v_s} = \frac{T_s}{t_{off}} = \frac{1}{1-d} \quad (2.6)$$

Kayıplar ihmal edilirse $P_s = P_o$ olur ve görev saykılı oranı D 'nin giriş ve çıkış akımları arasındaki bağıntısı aşağıdaki gibi elde edilir.

$$v_s \cdot i_s = v_{RL} \cdot i_o \Rightarrow \frac{i_R}{i_s} = (1-d) \quad (2.7)$$



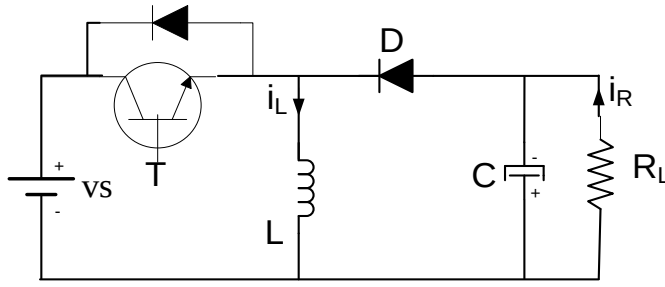
Şekil 2. 5. Gerilim yükselten dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri

Gerilim düşüren/yükselten DA/DA dönüştürücüler

Gerilim düşüren/yükselten (Buck/Boost) dönüştürücüler, DA güç kaynaklarının regülasyonunda, giriş geriliminin ortak terminaline göre çıkışın negatif kutuplu olması istenildiğinde ve çıkış geriliminin, giriş geriliminden daha az veya daha fazla olabileceği yerlerde yaygın bir biçimde kullanılır. Bu tip bir dönüştürücüde kararlı halde çıkış-giriş gerilim dönüşüm oranı; gerilim yükselten ve düşüren dönüştürücülerin çarpımına eşittir.

$$\frac{v_{RL}}{v_s} = d \cdot \frac{1}{1-d} \quad (2.8)$$

Şekil 2.6’ de gerilim düşüren/yükselten dönüştürücünün devre şeması verilmiştir. Transistörün beyzine her hangi bir tetikleme sinyali geldiğinde transistör ilettime geçer ve D diyotu ters polarmalandığından, tüm akım L bobini üzerinden akar. Transistör kesime gittiğinde bobin üzerinde depolanan akım D diyotu ilettime geçirir ve bu akım kondansatör, yük ve diyot üzerinden geçerek devresini tamamlar. Bu durum çıkış gerilimi ile giriş gerilimi arasında 180° faz farkı oluşmasına neden olur [24,26].



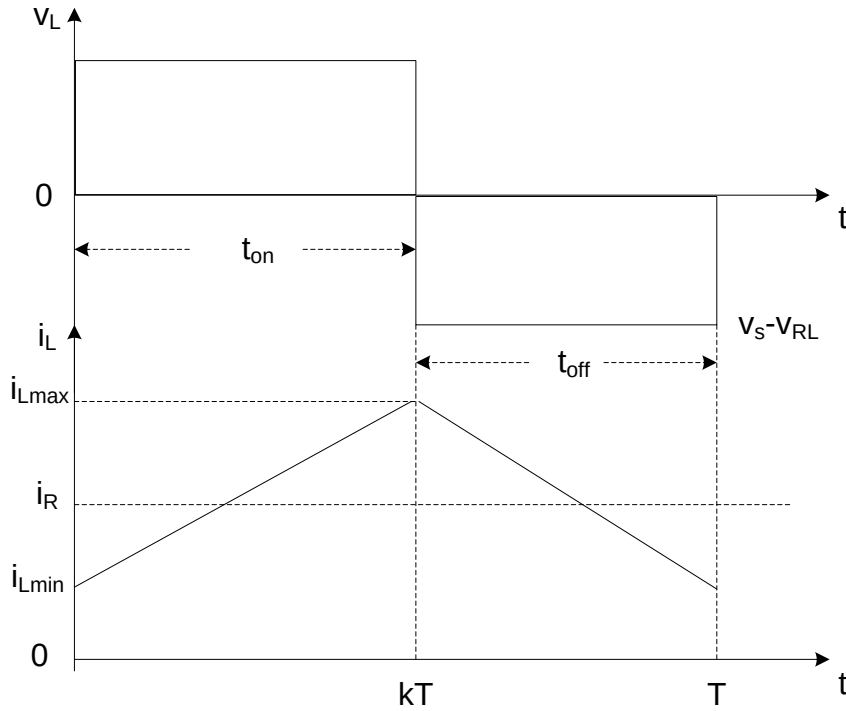
Şekil 2. 6. Gerilim düşüren/yükselten dönüştürücünün güç devresi

Şekil 2.7 bobin akımının sürekli hal durumu için dalga şekillerini göstermektedir. Bir zaman periyodu üzerinden bobin geriliminin toplamını sıfıra eşitleyerek bulunur.

$$v_s \cdot d \cdot T_s + (-v_{RL})(1-d)T_s = 0 \quad (2.9)$$

$$\frac{i_R}{i_s} = \frac{1-d}{d} (P_s = P_{RL}) \quad (2.10)$$

Eş. 2.8’ye göre çıkış geriliminin girişten daha az veya daha fazla olabileceğini ispatlamaktadır.

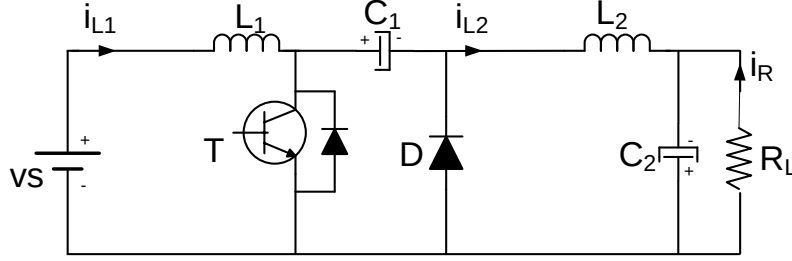


Şekil 2. 7. Gerilim düşüren/yükselten dönüştürücünün bobin üzerinde düşen akım ve gerilim dalga şekilleri

CÜK DA/DA dönüştürücüler

CÜK dönüştürücü, gerilim düşüren ve gerilim yükselten dönüştürücülerin bir birine kaskat bağlanmasından elde edilen bir DA/DA dönüştürücü topolojisidir. Güç devresi Şekil 2.8’de gösterilmiştir. Gerilim düşüren/yükselten dönüştürücü topolojisine benzer şekilde; çıkışında girişinden daha küçük ya da daha büyük bir negatif DA gerilim sağlar. Yukarıda bahsedilen topolojilerle CÜK dönüştürücüler arasındaki en önemli fark, onun girişinden çıkışına transfer edilen enerjinin şeklinde vuku bulur. CÜK dönüştürücüde bu transfer bir kondansatör üzerinden yerine getirilirken, diğer dönüştürücülerde bir bobin üzerinden gerçekleştirilir. Transistör yalıtımda iken D diyotu giriş bobini ve transfer kondansatörü üzerinden kaynağa bağlandığından ilettime geçer ve enerji transfer kondansatöründe depolanmaya başlar. Bu saykıl süresince bu kondansatör üzerinden geçen akım giriş bobin akımıdır. Transistör ilettime geçtiğinde transfer kondansatörü üzerinde depolanan enerji çıkış

bobini üzerinden yüke aktarılır. Bu saykıl süresince transfer kondansatörü üzerinden geçen akım çıkış bobin akımı olur [27].



Şekil 2. 8. CÜK dönüştürücünün güç devresi

Bobin akımları i_{L1} ve i_{L2} 'nin sürekli oldukları düşünüldüğünde kararlı durumdaki akım ve gerilim ifadeleri iki yolla bulunabilir. V_{C1} 'in sabit değeri için, bir periyot zaman dilimi üzerinden giriş ve çıkış bobinleri üzerinde düşen gerilimlerin toplamı sifıra eşitlenerek aşağıdaki denklemler bulunur.

$$L1: v_s \cdot d \cdot T_s + (v_s - v_{c1})(1-d) \cdot T_s = 0 \Rightarrow v_{c1} = \frac{1}{1-d} v_s \quad (2.11)$$

$$L2: (v_{c1} - v_{RL}) \cdot d \cdot T_s + (-v_{RL})(1-d) \cdot T_s = 0 \Rightarrow v_{c1} = \frac{1}{d} v_s \quad (2.12)$$

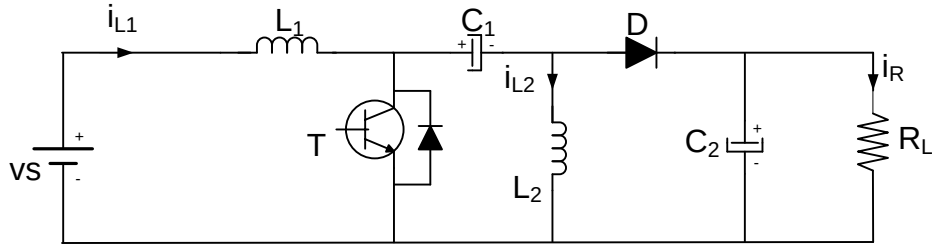
$$\frac{v_{RL}}{v_s} = \frac{d}{1-d} \quad (2.13)$$

$$P_s = P_{RL} \Rightarrow \frac{i_R}{i_s} = \frac{1-d}{d} \quad (2.14)$$

Eşitliklerden, ortalama giriş ve çıkış ilişkileri gerilim düşüren/yükselten dönüştürücü ile benzer olduğu görülebilir. Giriş ve çıkışı besleyen akımların oldukça serbest dalgacıklar şeklinde olması CÜK dönüştürücünün bir avantajıdır. Her iki bobin üzerinden geçen akım içindeki dalgacıkları eş zamanlı olarak ortadan kaldırmak mümkündür. Bu da daha düşük kapasiteli dış süzme üniteleriyle çalışabilmesini sağlar. Ancak transfer kondansatörünün büyük ölçüde dalgacık akımı taşıma kapasitesine sahip olması gerekir ki bu ise bir dezavantajdır.

SEPIC DA/DA dönüştürücüler

SEPIC dönüştürücü, girişine göre küçük, büyük ya da eşit büyüklükte pozitif regüleli çıkış gerilimi sağlayan bir DA/DA dönüştürücü topolojisidir. Şekil 2.9'de güç devresi görülen SEPIC dönüştürücü, geleneksel gerilim düşüren/yükselten dönüştürücüye benzer topolojidir. Ancak ilave bazı avantajları vardır. Bu avantajları; çıkışın evirilmemesi, giriş çıkış arasındaki izolasyon ve transistör yalıtımda iken çıkışın sıfıra düşmesi şeklinde sıralanabilir.



Şekil 2. 9. SEPIC dönüştürücünün güç devresi

Çıkış gerilimi ile giriş gerilimi arasındaki ilişki düşüren/yükselten çeviricide olduğu gibidir. Giriş gerilimi ile çıkış gerilimi arasındaki ilişki Eş. 2.15 ve Eş. 2.16'da verilmiştir.

$$d.v_s = (1-d).v_{RL} \quad (2.15)$$

$$\frac{v_{RL}}{v_s} = \frac{d}{1-d} \quad (2.16)$$

Bu devrenin çalışması da diğer dönüştürücü topolojilerinde olduğu gibi iki mod halinde açıklanabilir. Transistör iletimde iken giriş bobini kaynak üzerinden ve çıkış bobini transfer kapasitörü üzerinden şarj olur. Bu zaman aralığında çıkış kapasitörüne her hangi bir enerji transferi olmaz. Transistör kesime gittiğinde ise giriş bobini transfer kapasitörünü şarj eder ve aynı zamanda çıkış yükü için gerekli akımı sağlar. İlgili zaman aralığında çıkış bobini yüke bağlanmış olur. Transistörün

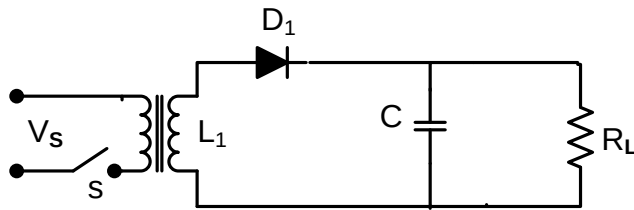
kesim anında çıkış kapasitörü akım palsine maruz kaldığı için çıkışta bir gürültü oluşur. Bu SEPIC dönüştürücülerin önemli bir dezavantajıdır [28].

2.2.2. İzoleli tip DA/DA dönüştürücüler

Elektriksel izolasyonun gerektiği durumlarda ve giriş çıkış gerilimleri arasındaki ilişkinin yüksek katsayı ile ifade edildiği durumlarda kullanılan dönüştürücü türleridir. Bu tür dönüştürücüler, devre yapısı itibarıyla manyetik kuplajlı olmasından dolayı izoleli dönüştürücüler olarak isimlendirilirler.

Flyback DA/DA dönüştürücüler

Flyback Dönüştürücüler, gerilim yükselten dönüştürücünün izole edilmiş hali olarak düşünülebilir. Yani transistörün iletim süresince kondansatörün şarj olması ve kesim süresince de ise deşarj olması temeline dayandırılan izoleli tip bir güç topolojisidir. Devrede bir transistör kullanılır. Flyback dönüştürücüler, bu transistörün anahtarlamasında üçgen dalga kullanılmasıyla kapasitesinin tepe akım değerine ulaşmamasından kaynaklı olarak uygulamada yaklaşık 100W üzerindeki güçlerde kullanılmazlar. Yüksek gerilim ve düşük güç gerektiren CRT (Catot Ray Tube) uygulamalarda yaygın olarak kullanılır. Ayrıca bu durum çıkış kondansatörü üzerinde ripıların oluşmasına da sebep olur. Bu ripıllar şarj deşarj akımları sırasında kondansatör üzerinde oluşan rampa gerilimleri dışında kondansatörün doğal halinde var olan seri iç direnç ve seri indüktansdan da kaynaklanır. Devrede yer alan kuple diyotu ve sargı kondansatörü çoklu çıkış uygulamaları için uygun tercih olmasını sağlar [29].

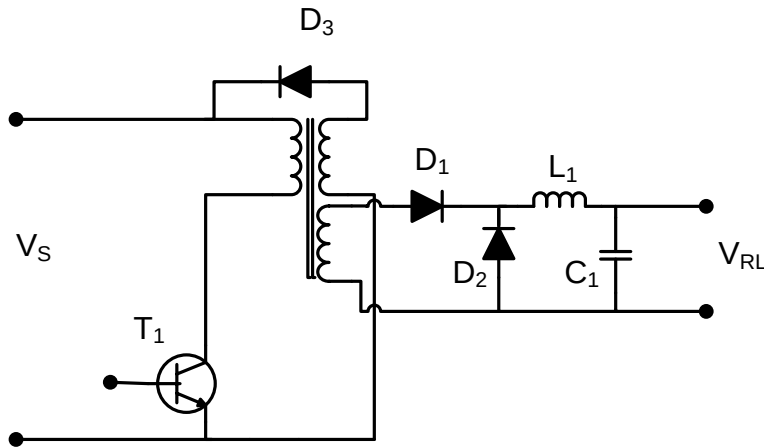


Şekil 2. 10. Flyback DA/DA dönüştürücü güç devresi

Şekil 2.10 'daki temel devredeki S anahtarı $D \times T$ boyunca kapalı ve $(1 - D) \times T$ boyunca açık olacak şekilde kullanılır. Anahtar kapalı iken i_1 , $(E/L_1)V_S$ doğru orantılı olarak artarken $i_2=0$ 'dır. Diyot V_C ve pozitif değerli e_2 'nin bileşke değeri ile ters polarite olmuş durumdadır. Anahtar kapalı iken ise i_1 sıfıra doğru giderken i_2 depolanan enerji ile artar ve $N_1=N_2$ olduğundan bu akım i_1 'in önceki değerine ulaşır. Bu sırada diyot iletme geçer.

Forward DA/DA dönüştürücüler

Çıkış gücü flyback dönüştürücüye göre iki kat kadar daha yüksek olan ve giriş gerilimi üzerinden bağlanan sargının transformatöre akım taşıması temeline göre çalışan bir güç dönüştürücüsü tipidir [30]. Şekil 2.11'da verilen devre şemasında görülen dönüştürücü tek transistörlüdür ve iletimi boyunca güç çıkış sargısına aktarılır. Elde edilmek istenen çıkış gücü çoklu transistör ve çoklu faz ilaveleri ile kW seviyelerine kadar arttırılabilir.

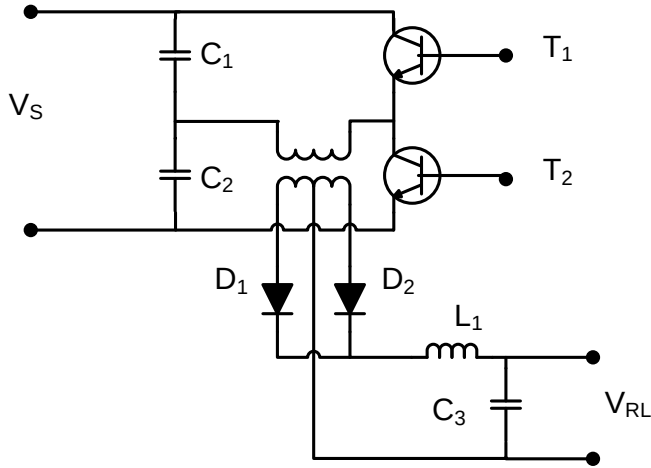


Şekil 2. 11. Forward dönüştürücü güç devresi

Yarım köprü DA/DA dönüştürücüler

Şekil 2.12' devre şeması görülen yaygın kullanılan bir dönüştürücü türüdür. Transistör V_S kaynağının negatif ve pozitif değerleri arasında anahtarlama işlevini görür. Düzeltilmiş bu temel kaynak doğrudan kapalı güç kaynakları için önemlidir. T_1

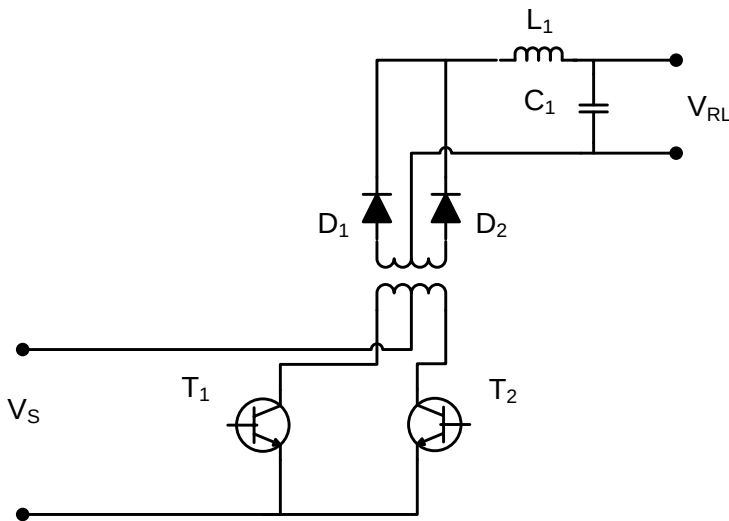
transformatörü C_1 ve C_2 kondansatörlerinin ortasına bağlanarak $V_s/2$ değerinin altında ve üstündeki durumlarda dengeleyici rolü üstlenir.



Şekil 2. 12. Yarım köprü dönüştürücü

Tam köprü DA/DA dönüştürücüler

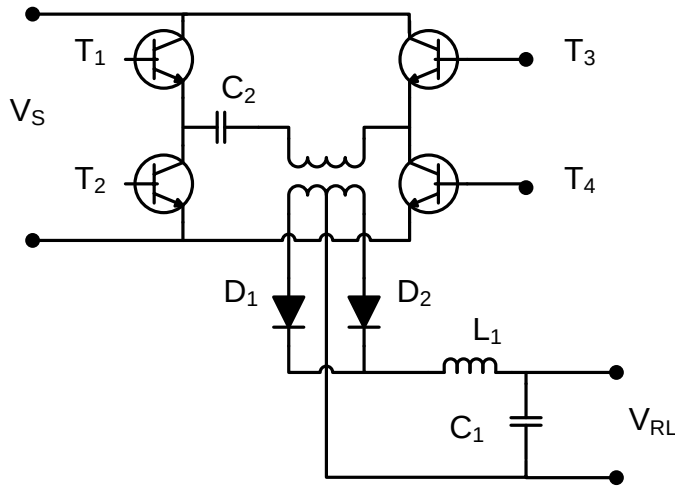
Şekil 2.13’de görüleceği gibi Yarım Köprü dönüştürücülere benzer bir yapısı vardır. Farklı olarak kolektör gerilimi $V_s/2$ yerine V_s ’ ye eşittir. Asimetrik anahtarlama zaman dengeleyici yoktur. Bir dengeleyici arzu edilirse DA polarma gerilimi için seri kondansatör C_2 ilave edilebilir [31].



Şekil 2. 13. Tam köprü dönüştürücü

Push-Pull DA/DA dönüştürücüler

Push-Pull dönüştürücülerin devre tasarımı Şekil 2.14’de gösterilmektedir. V_i 50V civarı ve daha düşük besleme ile 100V civarı kolektör gerilimine izin verir ki bu push-pull dönüştürücülerde en ideal çalışma şartıdır. Daha ekonomik ve basit yapıda olan ve 300V’un altında verimli çalışsan osilasyonlu tip bir dönüştürücü topolojisidir. Daha yüksek gerilimlerde transformatörün akı yoğunluğu nedeniyle asimetrik zamanlamada dengelemeyi sağlayamazlar [30].



Şekil 2. 14. Push-Pull dönüştürücü

2.2.3.DA/DA dönüştürücülerin denetimi

DA/DA dönüştürücülerin denetiminde açık döngülü geri beslemeli kontrol algoritmaları kullanılır. Ancak uygulamada geri beslemeli kontrol algoritmalarının daha çok tercih edildiği görülür. Çünkü geri beslemeli kontrol algoritmaları; giriş gerilimi, yük akımı ve devre bileşenlerinin sebep olduğu olumsuzluklara rağmen dönüştürücü performansı üzerinde daha iyi bir etkiye sahiptirler. Bu algoritmalar darbe genişlik modülasyonlu ve değişken frekanslı olmak üzere iki farklı kategoride ele alınırlar. Şimdiye kadar DA/DA dönüştürücülerden iyi bir performans elde etmek için çok sayıda kontrol algoritması önerilmiştir. Bunların en popüler olanlarını gerilim mod kontrol, ortalama akım mod kontrol, tepe akım mod kontrol, histeresiz

akım kontrol, vadi akım kontrol ve V^2 kontrol şeklinde sıralayabiliriz. Bu teknikler simülatörün geliştirilmesi başlığı altında ele alınmıştır.

3. SEZGİSEL TABANLI OPTİMİZASYON ALGORİTMALARI

Sezgisel Algoritmalar, genellikle durum uzayı çok büyük (ve ya sonsuz) olan problemleri çözmeyi hedefler. Bu zorluk ve karmaşıklıklardan dolayı (sınırlama, kural, strateji, yaklaşım kullandığından) bulunan çözümler genelde, “çözümlerden biri”, “çözümlerden en yaklaştığı”, “yeterince doğru olan çözüm” ve ya “kabul edilebilir çözüm” şeklinde isimlendirilir. Genel anlamda bir başlangıç popülasyonu üzerinde evrimsel hesaplama yaparak en kısa adımlama ile çözüme ulaşmayı hedefler. Burada sezgisel tabanlı algoritmalarından bazılarına; ne zaman ortaya çıktığına, gerçek hayatta nereden esinlenildiği ve algoritmadaki karşılığına, arama sürecinin nasıl işlediğine kısaca değinilecektir.

3.1. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu (PSO), 1995 yılında J.Kennedy ve R.C. Eberhart tarafından bilim dünyasına duyurulmuştur [32]. Doğada sürü halinde yaşayan balık ve kuş gibi canlıların sürü ile birlikte (sürü zekâsıyla) yaptıkları davranışların benzetiminden yola çıkan bir optimizasyon tekniğidir.

Sürü ile yaşayan canlılar, bireysel olarak elde etmeleri çok zor ve ya imkânsız olan özellikle yaşam şartlarını elde etme ve savunma ihtiyaçlarını sürü yardımıyla elde ederler. Sürüdeki her birey ulaştığı yiyecek, içecek, barınak gibi daha iyi yaşam imkânları ve ya sezdikleri tehlikeleri sürü ile paylaşır. Sürü zekâsı ile ulaşılan sonuçlar sürünün yeni hareketi belirler.

Algoritmada popülasyon sürüyü, parçacık ise sürüdeki bireyi temsil eder. PSO, önce çözümü arayan parçacıklardan bir sürü oluşturulur. Her parçacık ulaştığı en iyi çözümü sürü ile paylaşımıyla başlayan algoritma adım adım optimizasyonu gerçekleştirir. Burada her bir parçacık için sürüdeki bireylerde olduğu gibi pozisyonu, hareketi (yönü ve miktarı), sürü ile bilgi paylaşımı, önceki bilgisi ve bu bilgiler ışığında karar vermesi şeklinde temel özellikleri tanımlanır [33].

Algoritma adımları:

- 1.[Başla]: Başlangıç popülasyonunu oluştur
- 2.[Hesaplama]: Uygunluk fonksiyonu ile başlangıç popülasyonunu değerlendir.
- 3.[Pbest hesapla]: Her parçacık için en iyi değeri bul ve sakla.
- 4.[Gbest hesapla]: Sürü için en iyi değeri bul ve sakla.
5. [Sonlama]: İstenilen uygunluk değeri ve maksimum döngü kriterine bak.
- 6.[Hareket vektörünü güncelle]:
- 7.[Döngü]: Adım-2'ye dön.

Başlangıç popülasyonu, D boyutunda arama uzayında hareket eden, arama aralığı içinden rastlantısal olarak belirlenen N sayıda parçacıktan oluşturulur. Uygunluk fonksiyonu ise her sezgisel algortmada olduğu gibi en önemli parçadır ve probleme özgü olarak belirlenir. Çünkü bulunan çözümü değerlendirerek istenilen çözüme veya yeterli yakınlığa ulaşıp ulaşmadığımızı tespit eder ve arama sürecini sonlandırır. Pbest, arama sürecinde her bir parçacığın uygunluk fonksiyonu ile değerlendirilmesi ile elde edilen çözüme yakınlık değerlerinin en iyi olanıdır. Gbest ise Pbest'ler içindeki en iyi değerdir ve ulaşılan en iyi çözümdür. Gbest çözüm ya da kabul edilebilir çözüm aralığına ulaşırsa ve ya maksimum sayıda döngü gerçekleşirse arama süreci sonlanır. Hareket vektörleri güncellenirken Eş. 3.1 ve Eş. 3.2 kullanılır.

$$V_i^{k+1} = w^k \cdot V_i^k + c_1 r_1^k (Pbest_i^k - X_i^k) + c_2 r_2^k (Gbest^k - X_i^k) \quad (3.1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (3.2)$$

Burada V parçacığın hızını yani yeni pozisyonuna hareket vektörünü, X ise parçacığın konumunu gösterir. Eş.3.1. ile bulunan V_i^{k+1} değeri ile Eş.3.2. kullanılarak parçacık yeni konumuna taşınmış olur ve yeniden adım-2'ye gidilerek döngü tekrarlanır.

Eşitliklerde yer alan r_1, r_2 $[0,1]$ aralığında rastlantısal olarak her döngüde tekrar belirlenen stokastik doğasını etkilemek için kullanılan, c_1, c_2 $[0,1], [0,2]$ ve ya $c_1+c_2=4$ gibi arama başlangıcında probleme özgü olarak belirlenen hızlanma katsayılarıdır. Ayrıca yine probleme özgü olarak Eş.3.1.'de V_i^k 'e çarpım katsayısı olarak $[0,1]$ aralığında w değeri kullanılabilir ki atalet ağırlığıdır. Bir önceki hızın sonrakini ne kadar etkileyeceğini bir katsayıya bağlar. Normalde bu değer 1 alınır [34].

PSO az parametrelili ve kolay uygulanabilen verimli bir algoritmadır. Bu yüzden çok sayıda başarılı uygulaması vardır [35]. PSO için c_1, c_2 katsayılarının belirlenmesi, w kullanılıp kullanılmayacağı ve sürünün kaç parçacıktan oluşacağı uygulama öncesi açığa kavuşturulması gereken önemli noktalar. c_1, c_2 katsayıları parçacıkların yüksek veya düşük hızlı hareketlerine sebep olabilir. Her iki durumda da çözüme ulaşılsa bile yüksek sayıda iterasyon ve uzun çözüm zamanı demektir ki istenmeyen durumdur. Parçacık sayısı işlem yoğunluğunu artırır. Bu yüzden verimli çözüme ulaşılabilen en az sayıda parçacık kullanılmalıdır.

3.2. Yapay Arı Kolonisi Algoritması

Yapay Arı Kolonisi Algoritması (YAK) Karaboğa, D. tarafından geliştirilmiştir [36]. Sezgisel tabanlı bir yapay zekâ tekniğidir. Sürü zekâsı denilen balık, kuş kümeleri, karınca, termit ve arı kolonileri gibi hayvan ve böcek topluluklarının bireysel anlamda erişemeyeceği, çevrenin ve kaynakların verimli ve etkin kullanımı gibi sonuçlar ortaya koyan kolektif davranış sergilemelerinin benzetimine bir örnektir. Bu teknik, arıların sürü zekâsı ile yaptığı yiyecek arama davranışını modelleyerek pek çok problemde kullanılabilen yüksek performanslı bir algoritma sunmaktadır.

Arıların yiyecek arama davranışları incelendiğinde, kolonideki arılar arasında bir iş bölümü olduğu görülmektedir. Arılar nektar, polen ve bal elde etmek için gittikleri bölgeleri yiyecek kaynağı olarak kullanırlar. Bir yiyecek kaynağının değerini ise yuvaya yakınlığı, nektar yoğunluğu ve çıkarılmasının kolaylığı parametrelere bağlıdır. Bu kaynakları bulma ve kullanma sırasında iş bölümünü şu şekilde kabaca tanımlayabiliriz:

Görevi Belirli İşçi Arılar: İşçi arılar, daha önceden keşfedilen belli kaynaklara ait nektarın kovana getirilmesinden sorumludurlar. İşçi arılar aynı zamanda ziyaret ettikleri kaynağın kalitesi ve yeriyle ilgili bilgiyi kovanda bekleyen diğer arılarla paylaşırlar. Bundan sonra görevi belirli işçi arılara kısaca görevli arılar da denilir.

Görevi Belirsiz İşçi Arılar: Bu arılar nektarını toplayabilecekleri kaynak arayışı içerisindeyler. Görevi belirsiz iki çeşit işçi arı bulunmaktadır: içsel bir dürtüye veya bir dış etmene bağlı olarak rastgele kaynak arayışında olan kâşif arılar ve kovanda bekleyen ve görevli arıları izleyerek bu arılar tarafından paylaşılan bilgiyi kullanarak yeni bir kaynağa yönelen gözcü arılar. Kâşif arıların sayısının tüm koloniye oranı ortalama % 5–10 arasındadır.

Arılar arasında bilgi paylaşımı kolektif bilginin oluşumunda en önemli husustur. Bir kovan göz önüne alındığında, tüm kovanlarda ortak olan bazı parçalara ayırmak mümkündür ve bu parçalardan en önemlisi ise dans alanıdır. Yiyecek kaynağının kalitesi ve yeri ilgili bilgi paylaşımı dans alanında olmaktadır. Bir arı dans ederken diğer arılarda ona antenleri ile dokunurlar ve bulduğu kaynağın tadı ve kokusu ile ilgili de bilgi alırlar. Ziyaret ettikleri kaynağa daha fazla arı yönlendirebilmek için kovandaki çeşitli alanlarda bu dansı gerçekleştirir ve kaynağına geri döner.

Kesin çizgileri olmamakla birlikte genelde kovandaki uçuş yapılan yere yakın bir yerde olan dans alanında yapılan danslar, taşınan bilgiye göre ve canlılığa göre değişmektedir. En önemlisi nektarın tatlılığıdır. Bu üstünlük dans eşik değerini belirler. Ancak bu uyarıcı yeterli değildir. Bunun yanında nektarın çıkarılmasının kolaylığı da bir etkidir. Kovandan olan uzaklık, çiçek nektarının kıvamı, besinin genel durumu, kalitedeki göreceli değişimler, hava koşulları ve günün hangi vaktinin olduğu dansı etkileyen diğer etmenlerdir. Dansın biçimi, hızı, tekrarlanma sayısı kaynak hakkında diğer arılara bilgi aktarımı sağlar. Görevi belirsiz olan bir işçi arının içsel ya da dışsal faktörlerle yiyecek arayan kâşif arı ve ya dans alanını izleyen gözcü arı rollerinden biri ile yiyecek kaynağına ulaşması ile başlar. Kaynağına ulaşması ile artık görevi belli işçi arı rolüne başlar. Kaynaktan alıp geldiği nektarı depolayıcı arılara aktarır. Bundan sonra üç durumdan biri ile görevine devam eder: ya gittiği

kaynağı bırakarak tekrar dans alanını gözleyen gözcü arı ve ya arama yapan kâşif arı olur ve ya geldiği kaynağa dönmeden önce dans eder ve kovandaki arkadaşlarını da aynı kaynağa yönlendirebilir ya da diğer arıları yönlendirmeden tekrar kaynağa dönebilir. Yiyecek kaynağı bulma ve kaynaktan yiyecek taşıma bu adımlarla ilerler [37].

Algoritmada, aranan yiyecek kaynakları aranan çözümü ya da çözüm kümesini temsil eder. Belirli ölçütler dâhilinde aday çözümlerden oluşan yiyecek kaynakları popülasyonu oluşturulur. Her bir yiyecek kaynağı yani çözüm adayına algoritmanın formülünün uygulanması ise arı görevlendirme işlemidir. Uygunluk fonksiyonun uygulanması ile yapılan değerlendirme ise polen miktarının hesaplanmasıdır. Yeni yiyeceklerle ulaşılması ve daha iyi ise koloniye bildirilmesi, çözümlerin uygunluk fonksiyonuna göre seçilmesidir.

Algoritma adımları:

1. [Başla]: Başlangıç popülasyonu için N (yiyecek kaynağı) sayısını belirle ($X_{1,2,3..N}$).
2. [Çözümün parametre sayısını belirle]: Her bir yiyecek kaynağı X_i (çözüm) için optimize edilmek istenen parametre sayısınca (D) boyutlandır.
3. [Kısıtlama belirle]: Her bir $X_{1,2,3..N}$ için rasgele değerlerin her bir parametresi için $X_{\min}^i$ ve $X_{\max}^i$ değerleri belirlenir.
4. [Yiyecek Kaynaklarını Üret]: ilk 3 adımda belirlenen kısıtlamalara göre rasgele yiyecek kaynaklarını Eş. 3.3 ile oluştur.

$$X_i^j = X_{\min}^j + rand(0,1) * (X_{\max}^j - X_{\min}^j) \quad (3.3)$$

- 5.[Kaynak Bırak Değeri belirle]: Her bir yiyecek kaynağı için bırakılıp bırakılmayacağını belirlemek amacıyla birer $failure_i$ belirlenir ve başlangıçta 0'dır.

6. [Uygunlukları belirle]: Her bir X_i için uygunluk fonksiyon ile uygunluk değeri hesaplanır.

7. [Kaynaklara görevli arı gönder]: Eş.3.4. ile her bir X_i için görevli arı gönder ve mevcut kaynağı iyileştir ve ya yeni kaynak bul.

$$V_{ij} = \begin{cases} X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) & R_j < MR \\ X_{ij} \text{ ve } failure_i + 1 & R_j \geq MR \end{cases} \quad (3.4)$$

$\phi_{ij} = [-1,1]$: arasında uniform rasgele sayı.

$R_j = [0,1]$: arasında uniform rasgele sayı.

$MR = [0,1]$: kontrol parametresi.

k : “i” den farklı bir indis.

8. [Uygunlukları belirle]: Her bir X_i için uygunluk fonksiyon ile uygunluk değeri hesaplanır.

9. [Kaynaklar arası seçim yap]: V_i ile X_i arasında Debb vb. bir metot ile seçim yapılır ve yeni X_i olarak belirlenir. Debb metodu kullanılırsa V_i ile X_i arasındaki seçimde; öncelikle kabul edilebilir çözüm bölgesinde olanı alır. Eğer her ikisi de kabul edilebilir bölgede ise küçük maliyetli olanı alır. Ya da ikisi de kabul edilebilir bölgede değilse sınır aşım değer küçük olanı alır ve seçimi gerçekleştir.

10. [Dans ile çözümü anlat]: X_i (nin uygunluğu / uygunluk değerlerinin toplamı)

11. [Gözcü arılara seçim yaptır]: Kaynakların uygunluk değerine göre (rulet-çember) şeklinde kaynaklar seçilir.

12. [Bırakılacak kaynakları belirle]: $failure_i$ değerine ulaşan kaynaklar bırakılır ve yerine Eş. 3.3 ile yeni kaynaklar popülasyona eklenir.

13. [Sonlama]: maksimum iterasyon sayısı ya da çözüme/kabul edilebilir çözüme ulaşıp ulaşılmadığını kontrol et.

14. [Döngü]: Adım-8'e dön, şeklinde adımlanabilir [37].

Farklı problemlerde algoritmanın performansını arttırmak için ek parametreler kullanılmaktadır. Ayarlanması gereken parametre sayısının azlığı (kaynak sayısı, maksimum iterasyon, kabul edilebilir çözüm aralığı) uygulamasını kolaylaştırdığından kullanımını hızla yaygınlaştırmaktadır [37].

3.3. Genetik Algoritma

Genetik Algoritma (GA), Holland'ın 1975 yılındaki çalışmasıyla bilim dünyasında duyulmaya başlayan Genetik Algoritmaların gelişim sürecinin en önemli adımı ise Goldberg'in 1989 yılındaki çalışmasıdır [38,39]. Bu çalışma ile birlikte birçok bilim alanında optimizasyon ve karar süreçlerinde kullanılmaya başlamıştır. 1973'deki Rechenberg'in evrimsel hesaplama ile ilgili çalışması ise bu alan için önemli bir esin kaynağı olmuştur [40].

Koza'nın 1992 yılında genetik algoritmayı kullanarak genetik programlamayı geliştirmesi, genetik algoritmanın uygulama alanına geçişine farklı bir boyut katmıştır [41]. Genetik algoritma günümüzde; Otomatik Programlama ve Bilgi Sistemleri (yonga seti tasarımı, ders-sınav programlarının yapılması, ağ tasarımı), mekanik öğrenme (gözlenmiş bir veri takımını anlamak ve yorumlamak ve diğer özelliklerini tahmin etmek içerikli modellemeler), finans (hisse senedi değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi portföy optimizasyonu, endeks izleme, işlem maliyetleri ve risk tercihler), pazarlama (tüketicilere ait verileri analiz etmek (veri madenciliği), çeşitli tüketici kalıpları çıkarmak ve bu kalıplara dayanarak pazarlama stratejileri uygulamak), atama, hücrel üretim, sistem güvenilirliği, taşıma, rota belirleme gibi pek çok alanda çözüm aracı olmaktadır.

Tabiatта zayıf bireylerin zamanla yok olması ve güçlü bireylerin seleksiyonla korunumu (seçilim) ilkesinin benzetimi biçimindeki sezgisel tabanlı bir yapay zekâ

teknikidir. Biyolojideki bu ilkedden esinlenerek ortaya çıktığı için parametre isimleri de biyolojide kullanılan terimlerden alınmıştır.

Bilindiği gibi canlılar hücrelerden oluşur. Her hücrede aynı kromozom kümeleri bulunur. Kromozomlar DNA dizileri olup, tüm organizmanın örneği olarak hizmet ederler. Bir kromozom gen adı verilen DNA bloklarından oluşur. Her gen ise canlının belirli bir özelliğinin (göz rengi vb.) kodlarını taşır. Tüm genetik malzeme kümesine (tüm kromozomlara) genom adı verilir. Genom üzerindeki belli gen kümelerine genotip adı verilir. Genotipler, doğumdan sonra gelişmeyle fenotiplere (canlının göz rengi, zekâ vb. fiziksel ve zihinsel özelliklerine) dönüşür. Canlıların üremesi sırasında, atalardan gelen genler yeni bir kromozom üretmek için çeşitli biçimlerde (çaprazlama) bir araya gelirler. Ortaya çıkan yeni canlı daha sonra mutasyona uğrayabilir. Mutasyon DNA elemanlarının değişmesidir. Yeni canlının başarısı ise hayatta kalma ile ölçülür.

Algoritmada;

Gen : kendi başına anlamlı bilgi taşıyan en küçük birim olarak tanımlanır. Kodlama ile elde edilen çözümün anlamlı parçasıdır.

Kromozom (Birey) : Birden fazla genin bir araya gelerek oluşturduğu diziye denir. Kromozomlar, alternatif aday çözümleri gösterirler.

Popülasyon : Kromozom topluluğudur yani geçerli alternatif çözümler kümesidir. Popülasyondaki birey sayısı (kromozom) genelde sabit tutulur ancak sayısı ile ilgili genel bir kural yoktur ve probleme özgü belirlenmelidir. Gereğinden fazla olması fazla iterasyon, işlem ve zaman kaybı anlamına gelir. Gereğinden az olması ise algoritmanın düzensiz sonuç üretmesine neden olur ve güvenilirliğini kaybeder.

Kodlama : Kodlama GA için çok önemli bir işlem basamağıdır. Probleme GA uygulanmadan önce, verinin uygun şekilde kodlanması gerekmektedir. Kurulan genetik modelin hızlı ve güvenilir çalışması için bu kodlamanın doğru seçilmesi gerekir. Örnek çeşitleri;

Binary Kodlama: Her kromozom ikili (binary) sayı sisteminde olduğu gibi { 0, 1 } ile ifade edilecek şekilde oluşturulur. Bu dizideki her bit, çözümün belli karakteristiğini temsil eder veya tüm dizi bir çözümün bir parçasını temsil eder. Kodlamada en sık kullanılan yöntemdir. Örnek:

Kromozom A	1	0	1	1	0	0	0	1
-------------------	---	---	---	---	---	---	---	---

Permütasyon Kodlama: Düzenleme problemlerinde kullanılır. Burada her kromozom, sayıları bir sırada temsil etmektedir. Permütasyon kodlama, gezgin satıcı ve çizelgeleme problemleri için kullanışlıdır. Örnek:

Kromozom A	1	7	3	5	6
Kromozom B	7	1	5	6	3
Kromozom C	6	3	1	7	5

Değer Kodlama: Tam olmayan sayılar gibi karmaşık ve kodlaması zor olan değerlerin kullanıldığı problemlerde, işlem kolaylığı için değerler doğrudan kod olarak kullanılabilir. Örnek:

Kromozom A	3,8154	4,8524	3,1425	3,9654	4,2154
Kromozom B	2,2542	3,5625	3,2155	3,8877	3,0000
Kromozom C	3,5000	3,5741	4,4423	5,0105	2,9875

Ağaç Kodlama: Çözüm arama sürecinde; ihtimal, gelişme, değişme varsa durum temsilleri açısından kullanılabilir. Kodlamada her kromozom, nesneler ve nesneler arası ilişkileri içeren bir ağaç yapısıdır.

İşleçler (operatör) : Çözüm arama sürecinde popülasyon ve kromozomlar üzerindeki yapılacak işlemlerdir. İşleçler iyileştirme ve ya sonlandırma amaçlıdır.

Çaprazlama (gen takası): İki kromozomun (çözümün) birbirleri arasında gen alışverişinde (gen takası) bulunup iki yeni kromozom oluşturmasıdır. İkili yöntemle kodlanmış değişkenlerin yaptıkları üreme faaliyeti, kromozomların çaprazlamasına benzemesi dolayısıyla böyle adlandırılmaktadır. Eğer kodlamada gerçek değerler

kullanılıyorsa, klasik çaprazlama yöntemi yerine daha farklı yöntemler kullanılabilir. Rastlantısal olarak iki kromozom seçilir ve karşılıklı çaprazlama işlemine tâbi tutulur. Çeşitleri:

Tek Noktalı: Random seçilen bir noktadan kromozomlar genlerini takas eder.



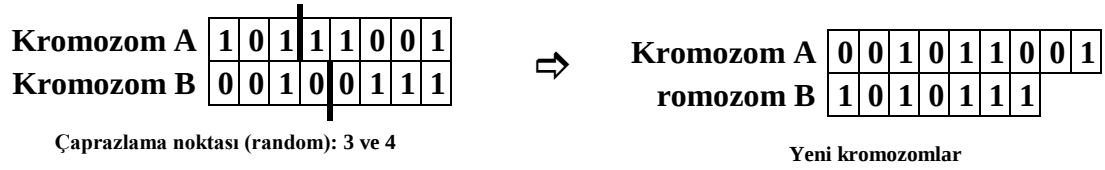
Çift Noktalı: Random seçilen iki noktadan kromozomlar genlerini takas eder.



Maskeleme: Random bir maske kromozom belirlenir. Maskedeki genlerin 0 olduğu yerlerdeki genler takas edilir, 1 ise aynen kalır.



Kes-Ekle: Random olarak her kromozom için belirlenen iki ayrı noktadan bölünerek gen takası gerçekleşir. Kromozomlardaki gen sayısını değiştiren bir operatördür olduğundan problemin durumuna göre sınırlı olarak kullanılabilir.



Mutasyon: Amaç mevcut kromozomların genlerinin bir ya da birkaçını değiştirerek yeni kromozomlar elde etmektir ve popülasyondaki çeşitliliğin azalmamasını sağlamaktır. Bir gene mutasyon uygulanıp uygulanmayacağı mutasyon ihtimalini “Mutasyon Oranı” yardımıyla random olarak belirlenir. Bu oran mutasyonun amacını aşmaması için genelde (0.1, 0.01 gibi) çok küçük bir değer olarak belirlenir. Bu nedenle mutasyon etkileri kromozomlarda az görülür. Mutasyon sırasında kromozomdaki gen sayısı değişmez. Mutasyon ve diğer işlemlerin uygunluk oranı en yüksek olan kromozoma uygulanmamasına “elitizm” denir. En iyi bireyi sonraki nesillere aktarılmasını sağlar. Rastgele belirlenerek, iki komşu gen, iki komşu olmayan gen, üç gen ve ya bir geni alıp sona yerleştirip kaydırma yaparak uygulanabilir. İki komşu gen için örnek:



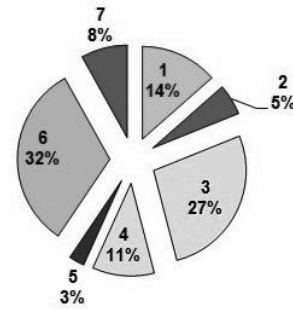
Tamir Operatörü: Probleme özgü olarak kullanılabilir. Çaprazlama ve mutasyon sonunda kromozomlarda: uygunluk değerini yüksek olan genlerin bozulması, gen sayısının artması azalması, anlamlı kromozomun gen yapısının bozulması gibi oluşabilecek istenmeyen durumların düzeltilmesi amacıyla probleme özgü tasarlanmış algoritmalar. Çözüm arama sürecinde emin olduğumuz kısıtlama durumlarının oluşmasını ve GA’nın çözümün olmadığı yerlerde gezinmesini engellemek amaçlıdır. Kromozom ve gen yapısının, iterasyon sırasında olumsuz etkilenmesini engeller ve GA’nın çözüme ulaşma performansını artırır.

Sonlama: Kabul edilebilir bir uygunluk seviyesi (0.95, 0.90, 0.75 gibi) bir kromozomda elde edilirse GA sonlanmalıdır. Kabul edilebilir seviyeye ulaşamıyorsa belirli bir iterasyon sonunda arama süreci sonlanmalıdır. Aksi halde sonsuz döngüler oluşabilir. Bazen de kabul edilebilir bir uygunluk bulunsa da belirli bir iterasyon kadar daha devam ettirilebilir. Yani iterasyona üst limit konulduğu gibi birde alt limit konulabilir.

Seçim (selection) : Uygunluk fonksiyonu ile yapılan hesaplamadan sonra popülasyon istenilen çözüme ulaşamamışsa yeni bir popülasyon oluşturulmalıdır. Yeni popülasyon mevcut popülasyonun en iyilerinden seçilerek elde edilir. Yeni popülasyonun seçilmesinde kaç ferden nasıl seçileceği seçim fonksiyonuyla sağlanır. Ebeveynler uygunluk değerlerine göre eşleşmek üzere seçilirler. Çeşitli yöntemler ile uygulanabilir:

Rulet-Çember Yöntemi: Tüm fertler birbirine bitişik bir şekilde düz bir çizgi üzerine dizilirler. Her bir ferde ilişkin bölümün uzunluğu, onun uygunluk değeri kadar olur. Toplam uygunluk maksimum olacak şekilde rasgele sayı üretilir ve rasgele sayı hangi bölüm içerisine gelirse, o bölümün ait olduğu fert seçilir. Ya da uygunluk değerlerine göre çemberde işgal edeceği yüzde belirlenerek kromozomlar çembere yerleştirilir. Çember 1–100 arası random sayı üretilerek çevrilir ve seçim yapılır. İşlem ulaşılabilecek popülasyonun gerekli adedi elde edilene kadar devam eder.

<u>Uygunluk değerleri</u>		
Kromozom 1	5	1 - 5
Kromozom 2	2	6 - 7
Kromozom 3	10	8 - 17
Kromozom 4	4	18-21
Kromozom 5	1	22
Kromozom 6	12	23-34
Kromozom 7	3	35-37
	37	



Şekil 3. 1. Rulet çember yöntemi

Rank (sıralama) Yöntemi: En iyi kromozomun uygunluk değeri çok yüksek ise, Rulet-Çember seçim yöntemi, sürekli yüksek olasılığa sahip kromozomun seçilmesi gibi hatalı bir yöne gidebilir. Bu nedenle, Rank seçim yöntemi uygulanabilir.

Popülasyon uygunluk değerine göre azdan çoğa doğru sıralanır. Sıralamadaki yeri kadar değerli (uygun) sayılarak rulet uygulanır. Bu durumda uygunluğun yüksekliği değil kaçınıcı en uygun değer olduğu dikkate alınmış olur. Uygunluklar arası oluşan aşırı farkların sürece etkisi azaltılır.

Turnuva Yöntemi: Kromozomların uygunluk değerlerine bakmadan belirlenen bir grup sayısı kadar (ikili, üçlü, dördü gibi) kromozom rasgele seçilir ve içlerinden en iyi uygunluk değerine sahip olan bir sonraki nesle aktarılır. Popülasyon sayısınca bu turnuva gruplamaları yapılarak seçim işlemi devam eder.

Sabit Durum Yöntemi: Kromozomların uygunluk değerlerine göre sıralayıp en iyi (üçü, dördü, beşi) olanlarının alınması mantığına dayanır.

Uygunluk Fonksiyonu: Bir popülasyondaki kromozomların (olası çözümlerin), aranan çözüm ya da istenilen sonuca uygunluğunu (yaşama başarısını) tespit etmek için kullanılan fonksiyondur. GA için en önemli parçalardan biridir. Hatta en önemlisidir denilebilir. Çünkü hazırlanan tüm algoritmanın sonucunu değerlendirmek için kullanılır. Tek Kriterli ve Çok Kriterli olmak üzere farklı türleri vardır.

Genetik Algoritma bu parametrelerin yardımıyla çözüm arama sürecini şu şekilde gerçekleştirir:

1. [Başla] n tane kromozomdan oluşan rasgele bir popülasyon üret (problem için muhtemel çözümler)
2. [Fitness] Popülasyondaki her x kromozomunun uygunluk değerini $f(x)$ uygunluk fonksiyonu yardımıyla hesapla.
3. [Yeni Popülasyon] Yeni popülasyon tamamlanıncaya kadar aşağıdaki adımları tekrarlayarak yeni bir popülasyona oluştur.
4. [Seçim] Yeni popülasyondan 2 ebeveyn kromozomu uygunluk değerlerine göre seç (uygunluk ne kadar iyiye seçilme şansı da o kadar yüksek olacak şekilde rastlantısal olarak seçim yapılır) ve sırasıyla;

5. [Crossover] Ebeveynlerden gelen bilgilerin (kuralların) çaprazlanarak yavruya aktarılması ve yeni bir olasılık oluşumunu sağla. (Eğer çaprazlama olmazsa yavrular ebeveynin birebir kopyası olur.)
6. [Mutasyon] Mutasyon ile yavru kromozomda genleri değiştir.
7. [Kabul] Yeni yavruyu bir popülasyona koy
8. [Yer Değiştir] Yeni popülasyonu algoritmanın bir sonraki koşuşu için kullan
9. [Test] Sonlanma koşulu sağlanırsa dur ve mevcut popülasyondaki en iyi çözümü döndür
10. [Loop] Adım-2 ye git

Görüleceği üzere Genetik Algoritma bir döngü içinde çözümü aramaktadır. Bu çözüm arayışının kısa sürmesi ve sağlıklı bir sonuca doğru ilerlemesi yani performansının yüksek olması için bazı etkenlerin bilinmesi gerekir:

1. Popülasyon büyüklüğü / Kromozom sayısı: Kromozom sayısını arttırmak çalışma zamanını arttırırken, azaltmak da kromozom çeşitliliğini yok eder.
2. Mutasyon Oranı: Kromozomlar birbirine benzemeye başladığında hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi genetik algoritmanın sıkıştığı yerden (tüm kromozomlar aynı platoda) kurtulmak için tek yoludur. Ancak yüksek bir değer vermek algoritmanın kararlı bir noktaya ulaşmasını engelleyecektir.
3. Kaç Noktalı Çaprazlama Yapılacağı: Normal olarak çaprazlama tek noktada gerçekleştirilmekle beraber yapılan araştırmalar bazı problemlerde çok noktalı çaprazlamanın çok yararlı olduğunu göstermiştir. Probleme özgü bu durumun tespit edilmesi gerekir.
4. Çaprazlamanın sonucu elde edilen bireylerin nasıl değerlendirileceği: Elde edilen iki bireyin birden kullanılıp kullanılamayacağı bazen önemli olmaktadır.

5. Durum kodlanmasının nasıl yapıldığı: Bir parametrenin doğrusal ya da logaritmik kodlanması genetik algoritmanın performansında önemli bir farka yol açabilir.

6. Başarı değerlendirmesinin nasıl yapıldığı: kötü seçilmiş bir uygunluk değerlendirme işlevi, çalışma zamanını uzatabileceği gibi çözüme de hiçbir zaman ulaşmamasına neden olabilir.

7. Ayarlanabilir fazla parametreye sahip olması probleme özgü çözümler sunmak adına önemli bir avantajdır [42]. Ancak her parametrenin dikkatli seçilmesi gerekmektedir.

3.4. Karınca Kolonisi Algoritması

Karınca Kolonisi Algoritması (KKA) ilk olarak Dorigo ve arkadaşları tarafından önerilmiş ve üzerinde çalışılmıştır [43,44]. Karınca kolonilerinin yiyecek arama ve yuvaya dönmeleri sırasında her zaman en kısa yolu bulmaları şeklindeki sürü davranışının benzetimini hedefleyen sezgisel bir algoritmadır.

Karıncalar koloni halinde yaşayan canlılardır. Görme yetenekleri zayıf olan karıncalar yiyecek ararken ve yuvaya dönerken her zaman en kısa yolu kullanırlar. Her hangi bir sebeple en kısa yol yok olur ya da daha kısa olan bulunursa artık bu yolu kullanırlar.

Araştırmalar karıncaların bu davranışı, bulunan yiyeceğin yolunu diğer karıncalara iletmek amacıyla yola bıraktıkları feromon (pheromone) isimli bir madde yardımıyla yaptıklarını göstermektedir. Bu maddenin miktarı (kokusu) ise yolun ve yiyeceğin uygunluğuna bağlı olarak artmaktadır ve diğer karıncalar tarafından tercih sebebi olmaktadır [45].

Algoritmada, yola bırakılan maddeler ve yoğunluklar çözüm uygunluğunu yollar ise çözüm adaylarını temsil eder. Gerçek durumda karıncaların görme, hafıza vb. özellikleri olmamasına rağmen algoritmadaki karınca kolonisi farklı özelliklere sahiptir. Örneğin: ayırık küme elemanlardan oluşur, önceki yolları hafızalarında

tutarlar, bulunan çözüm için iz bırakma işlemi farklı yapılarda olabilir, yerel en iyiyi arayabilirler. Bu farklılıklar algoritmanın kullanılabilirliğini ve performansını artırmayı hedefler [46].

Probleme özgü adımlama ve parametreleri olan KKA'nın klasik bir gezgin satıcı problemi için adımlaması şu şekilde oluşturulabilir [47]:

1. [Başla]: m karınca rastgele seçilen şehirlerden serbest bırakılır ve başlangıç feromonları hesapla.
2. [Seçim]: Uygunluk değerine göre (mesafeye - feromon) karıncanın bir sonraki gideceği şehri belirle.
3. [İzleri belirle]: Ulaşılan yeni şehir için tekrar fenomonu güncelle. Bir önceki durumu hafızada tut ve seçim yaparak diğer şehre ilerle. Fenomon azalması durumunda hafızadaki değerlerden yararlanarak başka bir hedef şehir seçerek ilerle.
4. [Döngü]: Tüm şehirlerden geçene kadar adım 2–3–4'ü tekrarla.
5. [Hesapla]: Tüm karıncaların uygunluklarını (toplam tur mesafeleri) karşılaştır. En kısıy en iyi çözüm olarak ata.
6. [Sonlama]: En iyi çözüm yeterli değilse karıncaları sıfırla ve adım-2'ye dön. En iyi çözüm yeterli ise ve ya maksimum iterasyon değerine ulaşılmışsa aramayı durdur.

3.5. Diferansiyel Gelişim Algoritması

Diferansiyel Gelişim Algoritması (DGA), Price ve Storn'un 1995 yılındaki çalışmaları ile duyurulmuştur. Genetik algoritmaya benzeyen yönleri ve sürekli veri öbeklerinde etkin arama sonuçları ile öne çıkan sezgisel bir optimizasyon tekniğidir [48].

Genetik Algoritmada olduğu gibi en iyinin seçilimi ilkesinden esinlenir ve benzer terimler benzer anlam ve işlevlerde kullanılmaktadır. Önemli bir fark ise kodlamada işleminde yöntem olarak daha çok “Değer kodlama” yani çözüme ait değerleri

değiştirmeden kullanmayı önerir. Price ve Storn değer kodlamanın zorluklarını azaltan ve performansını artıran bazı önermelerde bulunmuşlardır.

Algoritmanın adımları:

- 1.[Başla]: Popülasyon büyüklüğünü belirle (NP)
 - 2.[Atama yap]: Başlangıç sabit parametrelerinin atamasını yap. (Değişken sayısı (D), maksimum kuşak sayısı (G_{max}), Çaprazlama oranı (CR), Ölçekleme (F))
 - 3.[Popülasyonu oluştur]: Başlangıç popülasyonunu Eş.3.5. ile oluştur [49].
- $$\forall_i \leq NP \wedge \forall_j \leq D : x_{j,i,G=0} = x_j^{(1)} + rand_j[0,1](x_j^{(u)} - x_j^{(1)}) \quad (3.5)$$
- 4.[Çaprazlama]: Popülasyondaki tüm bireyler için Çaprazlama işlecini uygula.
 - 5.[Mutasyon]: Popülasyondaki bireylere mutasyon uygula.
 - 6.[Uygunluk Hesapla]: Uygunluk değer fonksiyonu ile popülasyon bireyleri için uygunluk değerlerini hesapla.
 - 7.[Seçim]: Bireyler arasında uygunluklara göre seçim yaparak yeni nesli oluştur.
 - 8.[Sonlama]: Maksimum iterasyon ve ya kabul edilebilir çözümü kontrol et.
 - 9.[Döngü]: Sonlama gerçekleşmemişse adım-4'e dön.

Burada popülasyondaki her bir birey optimize edilmek istenen parametre sayısı kadar uzunlukta bir vektördür. Çaprazlama (rekombinasyon) işlemi ise $[0,1]$ aralığındaki rasgele CR değeri ile kontrol edilir. Bu işlemin gerçekleşebilmesi için en az uygulana birey dışında üç vektöre daha ihtiyaç vardır ki bu da popülasyonun en az 4 ($NP \geq 4$) vektörden oluşması anlamında gelir. Mutasyon işlecisi ise GA da olduğu gibi çözüm adaylarının bazı parçalarının değişikliğe uğratılması anlamındadır ve farklı mutasyon yöntemleri önerilmiştir [50]. Örneğin rastgele seçilen birbirinden farklı üç vektörden ikisi arasındaki fark alınır ölçekleme oranı (F) ile çarpılım üçüncüye eklenir. İşleç çözüm adayı olan vektörler için uygulandığında mutasyona uğramış popülasyon elde

edilmiş olur. Mutasyon parametresi çözüm uzayı içinde vektörlerin hareketini ve farklı noktalardaki çözümleri yakalamayı amaçlayan bir işleçtir. Algoritmanın dinamiğini artırır. Ancak F oranı yüksek tutulursa algoritma stabilizesini kaybeder ve düzgün olmayan sonuçlar verir. Seçim işleminde ise karmaşık seçim algoritmaları yerine en yüksek uygunluklarının bir sonraki nesle aktarımı önerilmiştir [51]. Ulaşılan popülasyonun uygunluk fonksiyonu ile değerlendirilmesi ile kabul edilebilir çözüme ulaşılmışsa arama sonlandırılır. Eğer ulaşamıyorsa her optimizasyon algoritmasında olduğu gibi sonsuz döngülerden kurtulmak için maksimum iterasyon değerine göre de sonlama kontrolü yapılır.

4. DA/DA DÖNÜŞTÜRÜCÜLER İÇİN SEZGİSEL TABANLI PI, PD VE PID DENETLEYİCİLİ BİLGİSAYAR ARAYÜZÜ PROGRAMININ GERÇEKLEŞTİRİLMESİ

DA/DA dönüştürücülerin performansları, doğrusal olmayan karakteristikleri yüzünden uygulama alanına bağlı olarak değişir. Çoğu zaman PI, PID gibi geleneksel denetleyicilerle yerine getirilen kontrol işlevi tüm sistem performansı açısından arzu edilen performansı sağlamaz. Bir başka deyişle kazanç parametreleri çalışma şartlarına göre değişmediği için bu denetleyiciler sistem üzerinde düşük bir performans etkisi sahip olurlar. Literatürdeki çalışmalar genetik ve PSO gibi sezgisel algoritmaların doğrusal olmayan sistemlerin denetiminde kullanılan geleneksel denetleyici parametrelerini optimize etmede oldukça başarılı sonuçlar verdiğini göstermiştir. Bu çalışmalar ağırlıklı olarak elektrik motor sürücüsü olarak kullanılan güç dönüştürücülerinin denetimi üzerine yapılmıştır. Bununla birlikte farklı kontrol teknikleri ve farklı uygulama şartları altında DA/DA dönüştürücü topolojilerinin denetimi ve bunların performanslarının karşılaştırılması üzerine herhangi bir çalışma bulunmamaktadır. Bu amaçla hazırlanmış olan Toolbox; değişik yük şartları altında farklı kontrol tekniklerinin, geleneksel denetleyicilerin ve sezgisel tabanlı denetleyicilerin temel DA/DA dönüştürücülerin performansını incelleme ve analiz etmede önemli katkılar sağlar. Burada önerilen toolbox'ın tasarımında kullanılan bileşenler, geliştirilen toolbox'ın yapısı ve onunla yapılan analizler alt bölüm başlıkları halinde sunulacaktır.

4.1. Kullanılan Güç Dönüştürücüler

Bölüm 2' de anlatıldığı gibi geçmişten günümüze tek anahtarlı konfigürasyonlardan dört anahtarlı konfigürasyona kadar değişik tipte DA/DA dönüştürücü topolojileri önerilmiştir. Hangi tip devre konfigürasyonu kullanılırsa kullanılsın, bu dönüştürücülerin analiz ve denetiminde kullanılan teknik ve kontrol metotları birbirine oldukça benzer bir yapı ihtiva ederler. Bu yüzden geliştirilen toolbox' ta isimleri buck, boost, buck-boost, CÜK ve SEPIC olan 5 temel dönüştürücü topolojisi kullanılmıştır. Yazılım bileşeninde bu dönüştürücülerin dinamik davranışları, onları

tanımlayan matematiksel eşitlikler kullanılarak temsil edilir. Bölüm 2’ de ki devre şemalarından anlaşıldığı gibi her bir topoloji tek bir aktif anahtardan oluşmuş olup, bu topolojileri tanımlayan dinamik denklemler aktif anahtarın on-off durumuna göre oluşmuş olan iki devre şemasına Kirchoff kanunları uygulanarak elde edilmiştir. Toolbox’ da kullanılan dönüştürücüler için çıkarılmış eşitlikler aşağıda verilmiştir.

Buck Dönüştürücü;

$$\frac{di_L}{dt} = -\frac{1}{L} \cdot [r_L \cdot i_L + v_C - v_s \cdot sw] \quad (4.1)$$

$$\frac{dv_C}{dt} = \frac{1}{C} \cdot [i_L - i_R] \quad (4.2)$$

Boost dönüştürücü;

$$\frac{di_L}{dt} = \frac{1}{L} \cdot [v_s - r_L \cdot i_L - v_C \cdot (1 - sw)], \quad (4.3)$$

$$\frac{dv_C}{dt} = \frac{1}{C} \cdot [i_L \cdot (1 - sw) - i_R] \quad (4.4)$$

Buck-boost dönüştürücü;

$$\frac{di_L}{dt} = \frac{1}{L} \cdot [v_s \cdot sw - r_L \cdot i_L - v_C \cdot (1 - sw)] \quad (4.5)$$

$$\frac{dv_C}{dt} = \frac{1}{C} \cdot [i_L \cdot (1 - sw) - i_R] \quad (4.6)$$

CÜK dönüştürücü;

$$\frac{di_{L_1}}{dt} = \frac{1}{L_1} \cdot [v_s - r_{L_1} \cdot i_{L_1} - v_{C_1} \cdot (1 - sw)] \quad (4.7)$$

$$\frac{di_{L_2}}{dt} = \frac{1}{L_2} \cdot \left[\frac{1}{v_{C_1} \cdot sw - v_{C_2}} - r_{L_2} \cdot i_{L_2} \right] \quad (4.8)$$

$$\frac{dv_{C_1}}{dt} = \frac{1}{C_1} \cdot [i_{L_2} \cdot sw - i_{L_1} \cdot (1 - sw)] \quad (4.9)$$

$$\frac{dv_{C_2}}{dt} = \frac{1}{C_2} \cdot [i_{L_2} - i_R] \quad (4.10)$$

SEPIC Dönüştürücü;

$$\frac{di_{L_1}}{dt} = \frac{1}{L_1} \cdot [v_s - r_{L_1} \cdot i_{L_1} - (v_{C_1} + v_{C_2}) \cdot (1 - sw)] \quad (4.11)$$

$$\frac{di_{L_2}}{dt} = \frac{1}{L_2} \cdot [(v_{C_1} - r_{L_2} \cdot i_{L_2}) \cdot sw - (v_{C_2} + r_{L_2} \cdot i_{L_2}) \cdot (1 - sw)] \quad (4.12)$$

$$\frac{dv_{C_1}}{dt} = \frac{1}{C_1} \cdot [i_{L_2} \cdot sw - i_{L_1} \cdot (1 - sw)] \quad (4.13)$$

$$\frac{dv_{C_2}}{dt} = \frac{1}{C_2} \cdot [i_{L_1} - i_{L_2} \cdot (1 - sw) - i_R] \quad (4.14)$$

Burada sw darbe genişlik modülasyonu sinyallerine göre anahtarın on-off konumuna karar veren fonksiyondur. Buck, boost ve buck-boost dönüştürücü eşitliklerinde; IL bobin akımını, IO çıkış akımını ve VC filtre kondansatörü gerilimini temsil eder. CÜK ve SEPIC dönüştürücü eşitliklerinde; IL1 giriş bobin akımını, IL2 çıkış bobin akımını, VC1 transfer kondansatör gerilimini ve VC2 çıkış kondansatör gerilimini gösterir.

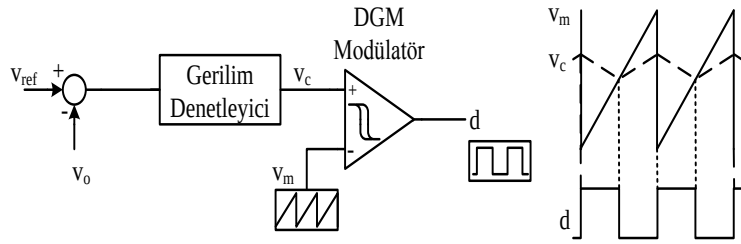
4.2. Kullanılan Denetim Teknikleri

Geliştirilen toolbox; DA/DA dönüştürücülerde kullanıcıya denetim açısından kapsamlı bir analiz olanağı sağlayacak şekilde tasarlanmıştır. Bu amaçla geniş bir literatür taraması yapılarak bu dönüştürücülerde yaygın olarak kullanılan denetim teknikleri ile son zamanlarda önerilmiş denetim teknikleri tespit edilmiştir. Bu tespite göre toolbox' da gerilim mod kontrol, tepe akım kontrol, ortalama akım kontrol, vadi akım kontrol, V^2 kontrol ve histerisiz kontrol teknikleri kullanılmıştır. Bu tekniklerin yapısı ve çalışması alt bölüm başlıklarında verilmiş olup, her bir tekniğe ait matematiksel model bu teknikler içerisinde kullanılan blokların zaman domeni

tabanlı eşitlikleri elde edilerek çıkarılmıştır. Bloklara ait denklemler bloğun ilgili teknik içerisindeki işlevine bağlı olarak elde edilmiştir.

4.2.1. Gerilim mod kontrol

Gerilim mod, en basit kontrol metotlarından biridir. Şekil 4.1’de temel şeması verilen teknik; gerilim döngüsü, DGM (Darbe Genlik Modülasyonu) karşılaştırıcısı ve testere dişi dalga form üretici olmak üzere üç birimden oluşur. Dönüştürücülerdeki güç transistörlerini tetiklemek için gerekli kontrol sinyali referans sinyal ile testere dişi dalga formunun karşılaştırılması ile elde edilir. Bu yapıda gerilim döngüsü dönüştürücülerin çıkış gerilim hatasını kullanarak DGM karşılaştırıcısına uygulanacak referans dalga formunu üretir.

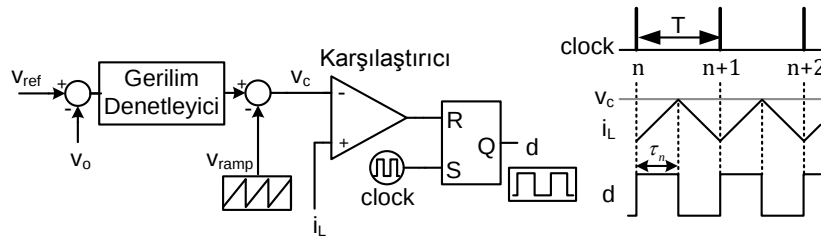


Şekil 4.1. Gerilim mod kontrol tekniği

Gerilim mod kontrol tekniğinin öne çıkan en önemli özelliği gerçek zamanlı uygulamalar için basit ve esnek bir yapısının olmasıdır. Ancak bu yapı bazı dezavantajları beraberinde getirir. Örneğin, hat gerilimi ile yük akımındaki değişimlere karşı diğer tekniklere göre daha yavaş cevap verir. Ayrıca kontrol döngüsüne bastırılması gereken iki kutup ilave eder.

4.2.2. Tepe akım kontrol

Tepe akım kontrol, basit olmasına rağmen güçlü bir kontrol yapısının olmasından dolayı DA/DA dönüştürücülerin denetiminde yaygın olarak kullanılır. Şeması ile tipik dalga formları Şekil 4.2’de gösterilmiştir.

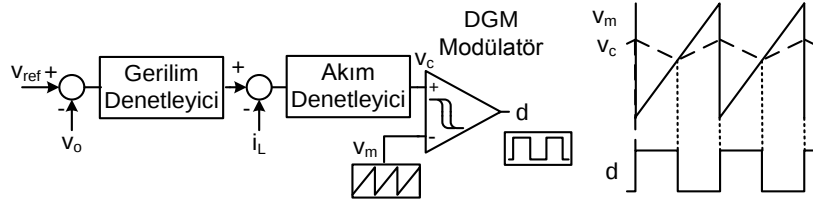


Şekil 4.2. Tepe akım kontrol tekniği

Bu teknikte, anahtarlama sinyallerinin üretimi için bobin akımının tepe değeri ile gerilim döngüsünden elde edilen referans akımının değeri karşılaştırılır. Şekilde verilen dalga formlarında gösterildiği gibi saat palsi üretici, anahtarlama periyodunun başlangıcında flip-flobu “set” eder. Bu da güç transistörünü ilettime geçirir. Bobin akımı referans akımın tepe değerine ulaşır ulaşmaz, akım karşılaştırıcısı filip-flobu “reset” eder ve güç transistörü bir sonraki periyodun başlangıcına kadar yalıtımda kalır. Bu çalışma şekli Tepe Akım tekniğinin temel mantığını oluşturur. Bu tekniğin uygulanmasında karşılaşılan en büyük problem, görev saykılı oranının %50’ den büyük olduğu durumlarda alt harmonik problemlerinin yaşanmasıdır.

4.2.3. Ortalama akım kontrol

Ortalama Akım Kontrol, yaygın kullanılan akım tabanlı kontrol tekniklerinden biridir. Tepe akım kontrol metodunda bobin akımının tepe değeri alınırken bu teknikte aynı değerın ortalaması alınır. Ortalama akım kontrol için şema ve tipik dalga formları Şekil 4.3.’de gösterilmiştir.

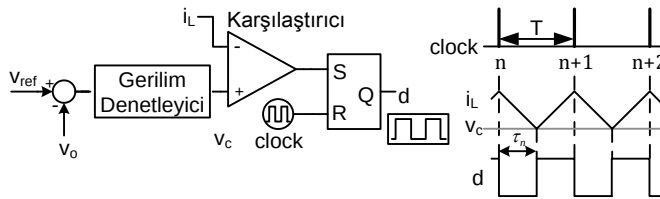


Şekil 4.3. Ortalama akım kontrol tekniği

Bu teknikte anahtarlama sinyallerini üretmek için gerilim döngüsü ve akım döngüsü olmak üzere iki döngülü bir kontrol yapısı kullanılır. Bu yapıda, gerilim döngüsü çıkış gerilim hatasını kullanarak akım referansına karar verirken, akım döngüsü de DGM karşılaştırıcısı için gerekli referans sinyalini üretir. Bu tekniğin avantajları olarak iyi gürültü süzmesi, uygulamada kolay akım sınırlaması ve mükemmel akım-gerilim regülasyonunu sayabiliriz.

4.2.4. Vadi akım kontrol

V^2 tekniğine benzeyen yeni bir akım tabanlı kontrol tekniğidir. Özellikle yük/giriş gerilimi değişimlerine hızlı cevap verebilmesi sebebiyle ilgi görmüş ve hızla gelişen bir uygulama alanı bulmuştur. Şekil 4.4'de devrenin çalışma prensibi ve çıkış dalga formları şematik olarak gösterilmiştir.



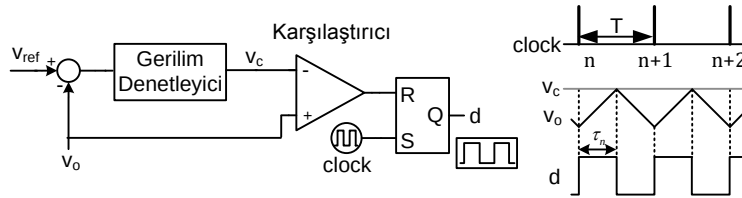
Şekil 4.4. Vadi akım kontrol tekniği

Çalışma prensibi kısaca şöyledir; her anahtarlama periyodunun başlangıcında saat pulsi üretici tarafından flip-flop resetlenir. Bobin akımı değeri voltaj döngüsü tarafından üretilen referans akım değerinin altına düşer düşmez, akım karşılaştırıcısı

flip-flobu set eder ve çıkışın aynı seviyede tutulmasını ve referans değeri ile aynı seviyede kalmasını sağlar.

4.2.5. V^2 akım kontrol

V^2 kontrol son yıllarda ortaya atılmış gerilim tabanlı bir kontrol tekniğidir. Bu teknik, özellikle hızlı yük değişimleri içeren uygulamalarda hızlı bir geçiş cevabına sahip olmasıyla öne çıkar. Ancak anahtarlama sinyallerini üretmesi çıkış kondansatörünün ripıl gerilimine bağlı olması, gerilim sıçramaları ve anahtarlama gürültülerinden etkilenmesine sebep olur ki bu da istenmeyen bir durumdur.



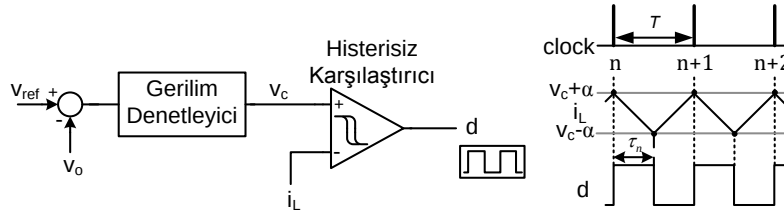
Şekil 4.5. V^2 kontrol tekniği

Şekil 4.5'te gösterildiği gibi, o akım mod kontrol tekniğine benzer şekilde iki geri besleme döngüsüne sahiptir. Bunlardan ilki çıkış gerilim hatasını kullanarak karşılaştırıcı için gerekli referansı üretirken, ikincisi referans sinyalle ölçülmüş çıkış gerilimini karşılaştırarak güç transistörünü sürmek için gerekli anahtarlama sinyallerini üretir.

4.2.6. Histerisiz kontrol

Histerisiz kontrol sabit frekans kontrol tekniklerine alternatif olarak önerilmiş bir değişken frekans kontrol tekniğidir. Son yıllarda hızlı bir dinamik cevaba sahip olması ve tasarımındaki kolaylıklardan dolayı bu tekniğe ilgi artmıştır. Histerisiz gerilim kontrol ve histerisiz akım kontrol olmak üzere iki tip kontrol algoritması vardır. Histerisiz gerilim kontrol elde edilebilen en basit kontrol tekniğidir. Çalışma prensibi, çıkış gerilimini alt ve üst olmak üzere iki gerilim band sınırında korumaya çalışmasına dayanır. Bu tekniğin tipik avantajları olarak basitlik, geri besleme için

kompanzasyona ihtiyaç duymaması ve yük geçişlerinde hızlı bir dinamik cevaba sahip olmasını sayabiliriz. Bununla birlikte gürültü hassasiyeti ve değişken frekans önemli zorlukları arasında yer alır. Histerisiz akım kontrol Şekil 4.6'de gösterildiği gibi gerilim kontrol ile benzer bir yapıya sahiptir.



Şekil 4.6. Histerisiz akım kontrol tekniği

İlgili teknikler arasındaki ana fark, histerisiz akım kontrolün anahtarlama sinyallerini üretmek için bobin akımını algılamaya ihtiyaç duymasıdır. Bu teknik aynı zamanda histerisiz band sınırının orta noktasına karar veren referans akımı elde etmek için bir gerilim döngüsü kullanır.

4.3. Kullanılan Sezgisel Tabanlı Denetleyiciler

Toolbox'da genetik, PSO ve YAK olmak üzere üç tip sezgisel algoritma kullanılmış olup bu algoritmalarla optimize edilen PID denetleyicinin tasarımının ilk işlem basamağı zaman domenindeki PID denetleyici eşitliğinin ayrık zaman domenine çevrilmesiyle başlar. PID denetleyiciye ait zaman domenindeki eşitlik Eş. 4.15'de verilmiştir.

$$u(t) = K_p \cdot e(t) + K_I \cdot \int_0^t e(t) \cdot dt + K_D \cdot \frac{d e(t)}{dt} \quad (4.15)$$

Burada K_p , K_i ve K_d PID denetleyicinin türev, integral ve oransal katsayıları ile ilgili kazanç parametreleridir. Burada $e(t)$, referans geriliminden dönüştürücünün çıkış geriliminin çıkarılmasıyla elde edilen hata değeridir ve kontrol değişkeni olarak

kullanılır. Eş. 4.15'e sırasıyla laplace dönüşümü, geri fark ve ters z-dönüşümü yöntemleri uygulanarak elde edilen ayrık zamanlı PID eşitliği Eş. 4.16'da verilmiştir.

$$u(k) = u(k-1) + K_0.e(k) + K_1.e(k-1) + K_2.e(k-2) \quad (4.16)$$

Burada $u(k)$ ve $u(k-1)$ sırasıyla k ve $k-1$ 'inci örnekleme zamanındaki ayrık zamanlı PID denetleyicinin çıkışlarını temsil eder. Burada K_0 , K_1 ve K_2 ise $\frac{K_d}{T}$, $\frac{-K_p - 2.K_d}{T}$ ve $K_p + K_i.T + \frac{K_d}{T}$ eşitlikleri kullanılarak hesaplanan denetleyici kazanç parametrelerini tanımlar.

Tasarımın ikinci aşamasında sezgisel tabanlı algoritmalarla K_p , K_i ve K_d parametrelerinin ayarlanması için gerekli işlem prosedürü yürürlüğe konur. Burada ilk olarak PSO tabanlı PID denetleyicinin tasarımı sunulacaktır. PID denetleyici K_p , K_i ve K_d olmak üzere üç ayarlanacak kazanç parametresine sahip olduğu için algoritma içerisinde üç boyutlu bir arama uzayı kullanılır. Bu uzayda her bir parçacık kazanç parametrelerini karakterize eden bir aday olup i 'inci parçacık için çözüm adayı Eş. 4.17'de verilmiştir.

$$K_i = (K_{i,0}, K_{i,1}, K_{i,2}) \quad (4.17)$$

Pratikte, 100 boyuta kadar olan problemlerin çözümünde 20–30 parçacıklı bir sürünün iyi sonuç verdiği bilinmektedir. Bu yüzden çalışmada oluşturulan sürüdeki parçacık sayısı denetleyici ve dönüştürücü türüne göre kullanıcının esnek seçimine olanak sağlayacak şekilde tasarlanmıştır. Bu algoritma optimal çözüme ulaşırken, aday parçacıkların bilgi paylaşımını esas alır. Her bir parçacık kendi pozisyonunu sürüdeki en iyi pozisyona doğru ayarlarken; bir önceki pozisyonu ve mevcut hız bileşeninden yararlanır. Parçacığa ait hız bileşeni ise, parçacığın kendi içerisindeki en iyi pozisyon (kişisel en iyi pozisyon), tüm parçacıklar içerisindeki en iyi pozisyon(global en iyi pozisyon) ve bir önceki hız bilgisinden yararlanılarak elde

edilir. Arama uzayında parçacıkların hız ve pozisyon bilgisi güncellenirken kullanılan matematiksel eşitlikler Eş. 3.1 ve Eş 3.2’de verilmiştir. Burada $V_{i,d}^k$ arama uzayının d’ninci boyutunda i’ ninci parçacığın k’ninci iterasyondaki hızını, $r_{1,2}^k$ 0–1 aralığında düzgün dağılımlı rasgele sayıları, $X_{i,d}^k$ arama uzayının d’ninci boyutunda i’ ninci parçacığın k’ninci iterasyondaki pozisyonunu, $Pbest_{i,d}^k$ arama uzayının d’ninci boyutunda i’ ninci parçacığın kişisel en iyi konum bilgisini ve $Gbest_d^k$ arama uzayının d’ninci boyutunda global en iyi konum bilgisini göstermektedir. $[c_1, c_2]$ her parçacığı kişisel en iyi konuma ve global en iyi konuma ayarlarken kullanılan stokastik hızlanma sabitleridir. c_1 parçacığın kendi tecrübelerine göre hareket etmesini, c_2 ise sürüdeki diğer parçacıkların tecrübelerine göre hareket etmesini sağlar. Bu sabitlerin değerinin düşük seçilmesi hedef konuma ulaşma süresini uzatırken, yüksek değerler seçilmesi beklenmedik hareketlerin oluşmasına ve hedef konuma yaklaşılamamasına sebep olur. Yapılan çalışmalar incelendiğinde c_1 ve c_2 için 0–4 aralığında belirleme yapıldığı görülmüştür. w^k , k’nci iterasyondaki atalet ağırlığı olup, bir önceki hızın mevcut hıza ne kadar etkisinin olacağını belirler. Bu değer iterasyon süresince aşağıdaki Eş. 4.17’de belirtilen fonksiyon kullanılarak güncellenir.

$$w^k = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \cdot iter \quad (4.17)$$

Burada $w_{\max}$ başlangıç ataletini, $w_{\min}$ son ataletini, $iter_{\max}$ maksimum iterasyonu ve $iter$ mevcut iterasyonu göstermektedir. Pratik çalışmalar iyi bir PSO performansı için atalet ağırlığının maksimum ve minimum değerlerinin sırası ile 0,9 ve 0,4 olması gerektiğini göstermektedir. Uygunluk değer fonksiyonu ise optimizasyonun başarısı için en önemli parametre olup bu çalışmada hata değerlerine göre hesaplama yapan bir fonksiyon kullanılmıştır (Eş. 4.18). Burada f_i ve α i’inci parçacık uygunluk değerini ve ağırlık faktörünü gösterir.

$$f_i^{k+1}(e_i) = \frac{1}{1 + \alpha \cdot \sum (e_i^k)^2} \quad (4.18)$$

PID denetleyicinin katsayılarının ayarlanmasında kullanılan ikinci algoritma genetik algoritmadır. Genetik algoritma uygulamanın ilk adımı problemin nasıl kodlanacağını belirlemektir. Bu çalışmada optimizasyonu hedeflenen denetleyici katsayıları K_p , K_i , K_d için aynı popülasyon içinde farklı gen boyutları kullanılmıştır ve gen üzerinde ondalık sayılar ile kodlanmıştır. Popülasyondaki kromozom sayısı ve her bir kromozomun kaç genden oluşacağı kullanıcı tercihiyle belirlenebilmektedir. Kromozomlar çözüm adaylarını, genlerin boyutları (3 boyut) K_p , K_i , K_d katsayılarını ve gendeki rakamlar ise çözümdeki sayının basamaklarını temsil eder. Gendeki rakamların hangi sayıyı temsil ettiği ise ondalık basamağının (virgölün) yerini gösteren ayrı bir dizi yardımıyla saklanır. Uygulamada ise 20–30 kromozomdan oluşan bir popülasyonun çoğu problemde yeterli çözüme ulaşabildiği görülmüştür.

Kullanıcı tarafından belirlenen aralıklar içinde rastgele oluşturulan sayılardan, rakamlar ve ondalık basamak yerini tutan diziye aktarılması ile Şekil 4.7’de gösterilen başlangıç popülasyonu oluşturulmuş olur.

		Gen 0	Gen 1	Gen 2	Gen 3	Gen ...	Gen ...	Gen ...	Gen m
Kromozom 0	Boyut 0 K_p	8	5	9	5	2	6	0	6
	Boyut 1 K_i	7	1	4	7	1	8	2	3
	Boyut 2 K_d	6	3	8	0	4	3	7	9
Kromozom 1	Boyut 0 K_p	:	:	:	:	:	:	:	:
	Boyut 1 K_i	:	:	:	:	:	:	:	:
	Boyut 2 K_d	:	:	:	:	:	:	:	:
Kromozom ...	Boyut 0 K_p	:	:	:	:	:	:	:	:
	Boyut 1 K_i	:	:	:	:	:	:	:	:
	Boyut 2 K_d	:	:	:	:	:	:	:	:
Kromozom N	Boyut 0 K_p	:	:	:	:	:	:	:	:
	Boyut 1 K_i	:	:	:	:	:	:	:	:
	Boyut 2 K_d	:	:	:	:	:	:	:	:

Şekil 4.7 Başlangıç popülasyonu

Uygunluk değeri fonksiyonu olarak PSO’da ifade edildiği gibi Eş. 4.18’den yararlanılmıştır. Her kromozomun uygunluk değeri bulunduktan sonra çaprazlama işlemine geçilir. Çaprazlama işleminde kromozomlar arası gen değiştirme işlemi yapılır. Çaprazlama oranı 0–1 arasında bir değerdir ve kullanıcı ara yüzünden seçilebilir. Pratik kullanımda 0,8–0,95 aralığında bir değerdir. Kromozom için gen değiştirme işlemi uygulanıp uygulanmayacağını belirler. Kullanıcı tarafından en iyi uygunluğa sahip kromozomların bir kısmı çaprazlama işleminden muaf tutulabilir. Ayrıca çaprazlama yaparken kromozomların Gen 0 (en yüksek basamağı)’dan başlayarak bazı basamakları da çaprazlama işleminden muaf tutulabilir. Bu işleme elitizm denir. Bu çalışmada kromozom üzerinde %20 genlerin üzerinde 1 basamak elitizm uygulanmıştır. Çaprazlama işlemi sonunda oluşan popülasyona mutasyon uygulanır. Kullanıcı tarafından belirlenen mutasyon oranı 0-1 arasına bir değerdir. Uygulamada bu oran 0,1–0,2 aralığındadır. Mutasyon sonunda yeni popülasyonun uygunluk değerleri yeniden hesaplanır. Uygunluk değerine göre rulet yöntemi ile seçim yapılarak yeni popülasyon oluşturulur. En iyi uygunluklar K_p , K_i ve K_d değerlerine atanır ve döngü içerisinde arama devam eder.

PID denetleyicinin katsayılarının ayarlanmasında kullanılan son algoritma ise Yapay Arı Kolonisi algoritmasıdır (YAK). Bu algortmada da PID denetleyicisinin K_p , K_i ve K_d katsayıları için kullanıcı tarafından belirlenen aralıkta ve belirlenen sayıda yiyecek kaynağının Eş. 3.3 yardımıyla ayrı dizi boyutu halinde oluşturulması ile optimizasyon işlemi başlar. Yiyecek kaynaklarının uygunluk değeri Eş. 4.18 kullanılarak hesaplanır. İşçi arıların yiyecek kaynağına gönderilmesi ile kaynakların güncellenmesi yapılır. Bu işlem için Eş. 3.4 kullanılır. Eş. 3.4’de X_{ij} j ’inci yiyecek kaynağının i ’nci boyutunun değerini gösterir. Tool ile yapılan uygulamalarda yiyecek kaynağı sayısı 30–40 arasında seçilmiştir. Boyut ise K_p , K_i ve K_d değerlerini temsil ettiği için 3’tür. X_{kj} ise aynı boyuttaki i ’den farklı k indisli kaynaktır. Yani aynı boyuttaki diğer bir yiyecek kaynağını temsil eder. Burada diğer bir yiyecek kaynağının değerine göre kaynağın kendisini güncellemesini sağlamak amacıyla kullanılmıştır. Güncelleme işlemi $[0,1]$ arasındaki MR parametresi ile kontrol edilir. Bu değer 1 seçilirse tüm kaynaklar her döngüde güncellenir. Kullanıcı bu parametreyi ara yüzden belirleyebilir. Güncelleme sonunda elde edilen yeni

kaynak öncekinden uygunluk değeri olarak daha kötü ise önceki konum değiştirilmez. Ancak kaynağa ait *failure* değeri bir arttırılır. Bu değer kaynağın kaç kez gelişmediğini sayar. Yine kullanıcı ara yüzünden belirlenen kaynak bırakma limiti ile bu değer belirlenebilir. Uygulamalarda *failure* 10 seçilmiştir. Yani 10 defa güncellemesi başarısız olan kaynak gelişmeyen kaynak kabul edilir ve yeni bir kaynak ile değiştirilir. ϕ_{ij} değeri ise $[-1,1]$ arasına rastgele bir değerdir.

Yiyecek kaynaklarının işçi ve gözcü arılar ile işletilmesi ve güncellenmesi ile elde edilen yeni kaynakların uygunluk değerleri hesaplanır. En iyi uygunluklar K_p , K_i ve K_d değerlerine atanarak optimizasyon döngüsüne devam edilir.

Anlatılan sezgisel algoritmalarda en iyiyi arama süreci belli bir sayıda döngü ile sınırlıdır. Yani istenilen uygunluk elde edilemese bile maksimum iterasyona ulaşıldığında arama sonlanır. Ancak tool ile yapılan uygulamalarda bu iterasyon değeri 1 olarak alınmış ve programın ana döngüsünün periyod örnekleme aralığında 1 defa sezgisel algoritma çalıştırılmıştır.

4.4. Yazılımın Geliştirilmesi

Yazılımın geliştirilmesinde Microsoft Visual Studio 2008 Visual C# .NET Framework 3.5 platformu kullanılmıştır. Bu platformun seçilmesinde yaygın kullanım ve önceki öğretiler etkili olmuştur. Veri tabanı ile ilgili ihtiyaçlar için MS Access 2010 tercih edilmiştir. Text dosyaya yazdırmak şeklinde ve daha kapsamlı bir veri tabanı yazılımı olan MSSQL üzerinde de denemeler yapılmıştır. Ancak çalışmamız için ya yetersiz kalmış ya da fazla işlem ve yer kullanımı sebepleriyle uygun görülmemiştir. MS Access'in önceki sürümleri ile de istenilen elde edilmişse de yaygınlaşmakta olan 2010 sürümü (.accdB) tercih edilmiştir.

Geliştirilen yazılım üç projeden oluşan bir Visual C# Windows Forms Application çözümü (solution) biçimindedir. Çözümde yer alan projelerden biri DA/DA dönüştürücülerine ait tüm durum denklemlerini içerecek DLL (Dynamic Link

Library) biçiminde tasarlanmıştır. Diğer proje (DLL) ise veri tabanı ile ilgili tüm işlemleri yürütmektedir. Üçüncü ve son proje ise ana projemiz olup içerisinde formlar, sınıflar vb. ile birlikte üç tane grafikler için geliştirilen UserControl içerir. UserControl Visual C# vb. dillerde kullanılan bir nesne türüdür. Bu nesneler geliştirildikten sonra derlendiğinde araç kutusu (toolbox) alanında görünür. Bu alandan diğer kontroller (buton, textbox vb.) gibi sürükleyip form üzerine bırakılarak kullanılabilir.

Grafik ile ilgili gereksinimler için bilinen bazı grafik çizim araçları (Zgraph, Chestysoft, DevExpressChart vb.) üzerinde çalışmalar yapılmıştır. Ancak ya kullanım zorluğu ya da yetersiz işlev ve görsellik gibi sebeplerle kullanılmamış ve bahsedilen UserControl tercih edilmiştir. Bu UserControl’lerin geliştirilmesinde Quinn-Curtis firmasına ait QCChart2D yazılımının ücretli DLL’i satın alınarak kullanılmıştır. Bu tercihte ilgili DLL’in yüksek veri sayısında çizim, yakınlaşma, yenileme vb. işlemlerde farklı donanımlara rağmen iyi bir performans göstermesi etkili olmuştur. Zira simülatörün çalışması sırasında durum denklemlerini içeren DLL tarafından her bir çıkış için (seçilen periyot örnekleme sayısı, frekans, simülasyon süresi gibi değerlere bağlı olmakla birlikte) ortalama en az 60.000–200.000 adet veri üreterek grafiğe aktarmaktadır. Farklı çalışma şartları seçildiğinde bu değer yükselmekte ve çoklu çizimlerde milyonu aşmaktadır. Pek sık ihtiyaç duyulmayan bu çalışma şartlarının zorlandığı durumlarda bile düşük performanslı da olsa DLL yanıt vermektedir.

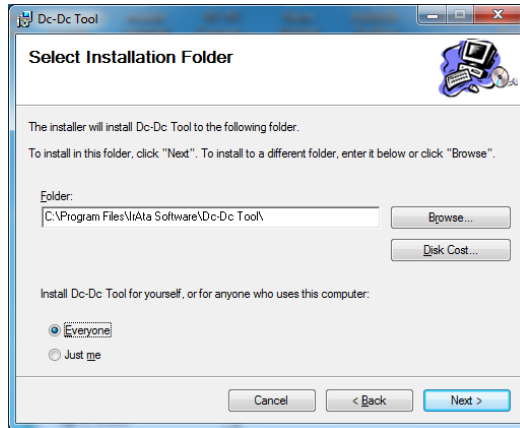
4.5. Gerçekleştirilen Bilgisayar Ara Yüzü Yazılımı

“DC/DC Tool” olarak isimlendirilen bilgisayar ara yüz yazılımı için Visual Studio 2008 platformundaki “Setup Project” türü ile .sln uzantılı platform çözüm dosyası ve tüm içerikleri derlenmiş ve kurulum dosyaları elde edilmiştir. Bu dosyalar Şekil 4.7’de gösterilmiştir. DC/DC Tool’un kurulabilmesi için en alt seviyede Windows Xp kurulabilen bir PC donanımı gereklidir. Kurulum yapılacak PC üzerinde işletim sistemi olarak Windows XP, Vista veya 7 sürümlerinden biri kurulu olmalıdır. Kullanılan grafik DLL’i 32 bit uyumlu olduğundan kurulu işletim sistemi 32 bit olmalıdır.

Ad	Değiştirme tarihi	Tür	Boyut
Dc-Dc Tool.msi	15.12.2012 15:42	Windows Installer Paketi	32.772 KB
setup.exe	15.12.2012 15:41	Uygulama	456 KB

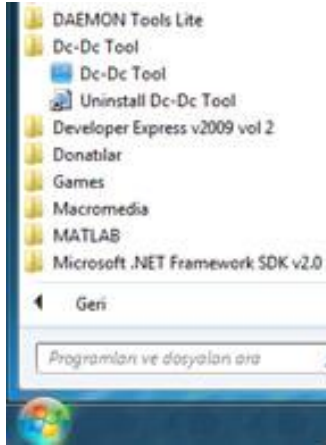
Şekil 4.8. Yazılımın setup dosyaları

Setup dosyası çift tıklandığında birkaç adımla tamamlanan kısa bir kurulum gerçekleşmektedir. Kurulum esnasındaki ekranlardan bir adımı Şekil 4.8’de görülmektedir. Kurulum başında .NET Framework 3.5 ve MS Access 2010 veri tabanı kullanımından kaynaklı olarak Access Database Engine ve Framework güncellemelerini internet üzerinden (gerekliyse) yapar. Daha sonra kurulum tamamlanır.



Şekil 4.9. Kurulum ekranı

Kurulum sonunda masaüstünde ve başlat menüsüne yazılımın kısayol ikonları yerleşir. Bu kısayollar Şekil 4.9 ve Şekil 4.10’da gösterilmiştir.

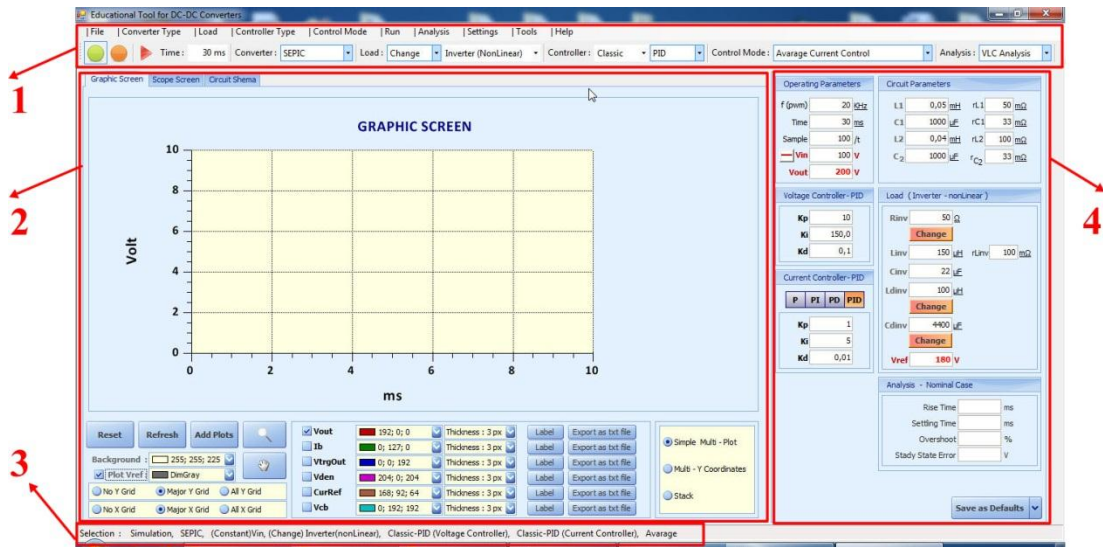


Şekil 4.10. Başlat menüsü kısayolu



Şekil 4.11. Masaüstü kısayolu

Bu kısayollardan biri tıklanarak program çalıştırılabilir. DC/DC Tool çalıştırıldığında gelen ilk ekran (açılış penceresi) Şekil 4.11'de gösterilmiştir.



Şekil 4.12. Yazılım çalıştığında gelen açılış penceresi

DC/DC Tool'un açılış penceresini gösteren Şekil 4.11'de rakamlarla belirtilen dört başlık üzerinde inceleyebiliriz; Menü ve Araç Şeridi(1), Grafik Ekranı ve Ayarları(2), Durum Çubuğu(3) ve Çalışma Parametreleri Girişleri(4).

Menü ve Araç Şeridi

Araç Şeridi (toolStrip), toolbox' ın çalışmasıyla ilgili temel seçimlerin yapıldığı bölümdür. DC/DC Tool ile beş ana dönüştürücü türünde analiz ve benzetim

çalışması yapılabilir. Bu dönüştürücüler Düşüren (Buck), Yükselten (Boost), Düşüren/Yükselten (Buck/Boost), CÜK ve Sepic dönüştürücülerdir. Bunlardan hangisi ile toolbox' ın çalışacağını seçmek için “Converter :” bölümündeki açılır seçim kutusu kullanılır.

“Load :” bölümünden ise iki seçim yapılabilir. Öncelikle devre üzerine bağlanan yük sabit/değişken seçimi yapılır. Burada değişkenlik çalışmanın belirli bir süresinde yükün belirli bir miktar artması ve ya azalması anlamındadır. Bu değişimle zaman/miktar ayarları “Çalışma Parametreleri Girişleri” bölümünden yapılır. Bu sayede yük üzerindeki değişimlerde dönüştürücünün nasıl bir performans gösterdiği analiz edilebilir. Her yük tipi için değişen devre elemanı farklı olarak ayarlanabilmektedir. Diğer seçim ise yükün tipidir. Yazılımda beş farklı yük tipi için analizler yapmak mümkündür. Bu yükler direnç(R), direnç-bobin(RL), direnç-kondansatör(RC), doğru akım motoru (DA-Motor) ve invertördür. Invertör türü yükte, invertörün kendisine bağlanan yüküne göre lineer veya non-lineer olarak ayrıca seçilebilir. Uygulamada dönüştürücülerin çıkışlarına bağlanan yükler genellikle bu beş türden biri ya da bunların eşdeğer devreleri olduğundan toolbox bu yük türleri eklenmiştir.

Bu şerit üzerindeki diğer seçim ise denetim tipidir. Araç şeridi üzerinde “Controller :” bölümünden bu seçim yapılabilir. Yine uygulamada sıkça kullanılan **P** (Propotional), **PI**(Proportional–Integral), **PD**(Proportional–Derivative), **PID**(Proportional-Integral–Derivative) denetleyici tipi seçimler arasında yer almaktadır. Burada DC/DC Tool simülatörünün önemli bir farkı anlamına gelen diğer bir seçim daha vardır ki bu da denetleyicinin türüdür. Açılır seçim kutu üzerinden denetleyiciyi ya klasik denetleyici ya da yapay zekâ teknikleri kullanan denetleyici biçiminde seçilebilir. Yapay zekâ tekniği olarak sezgisel algoritmalarından üç tanesi çalışmaya dâhil edilmiştir. Bu üç algoritma yenilik, yaygınlık, verimlilik-performans gibi değerlendirme başlıklarında incelendiğinde öne çıkması ile seçimde etkili olmuştur. Buradaki değerlendirmelerin bazılarına yapılan örnek benzetimlerle bir sonraki başlık altında değinilmiştir.

Farklı Kontrol Teknikleri kullanarak analiz yapmaya imkân vermesi DC/DC Tool'un pek çok çalışmadan farklı olmasını sağlamıştır. Yapılan çalışmalarda genel anlamda bir kontrol tekniği ve bu tekniğin daha iyilenmesi hedeflenmiştir. Simülatörde Gerilim Mod, Tepe Akım, Ortalama Akım, Vadi Akım, V^2 Akım ve Histerisiz Kontrol olmak üzere literatürde önemli yere sahip yedi tekniğe yer verilmiştir.



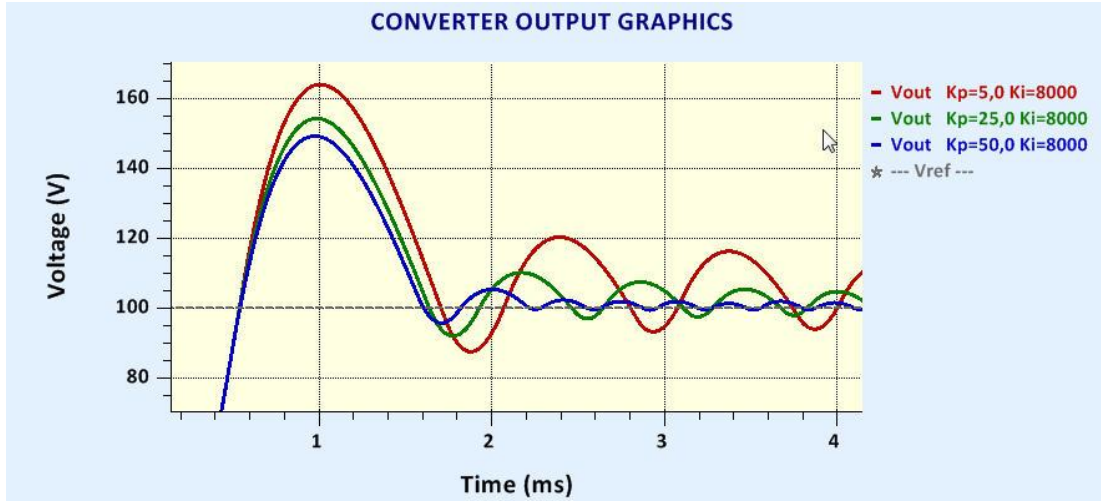
Butonu simülatörün çalışmasını sağlar. Kısayolu F5'tir.

Araç Şeridi üzerinden yapılan her türlü seçim ve ayarlama işlemi için ikinci bir yol olarak menülerdeki komutlar da kullanılabilir.

Grafik Ekranı ve Ayarları

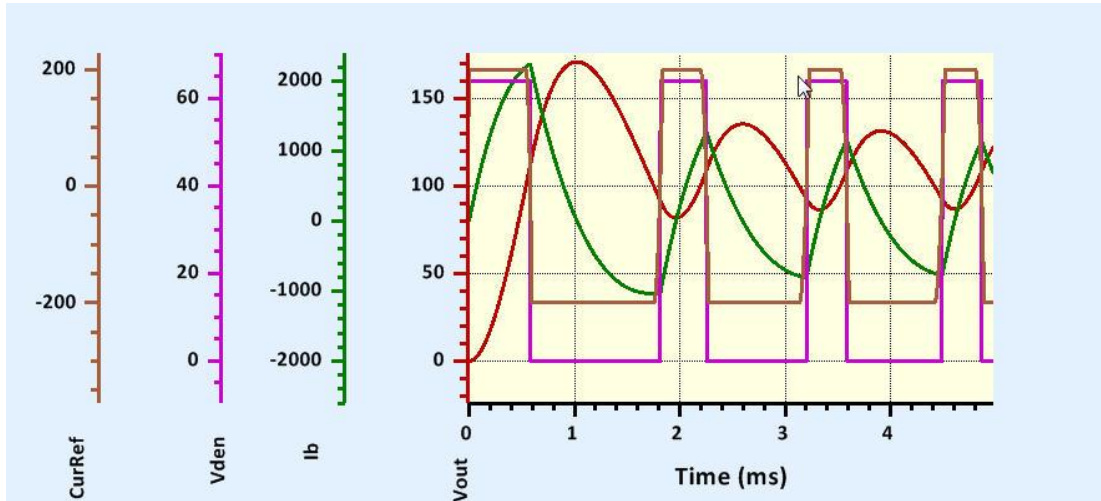
Grafik ekranında üç sekme (tabpages) vardır; Grafik (Graphic), Osilaskop (scope) ve devre (circuit). Grafik sekmesinde toolbox'ın çalıştırılması ile elde edilen verilerin görsel olarak analizi hedeflenmiştir. Bu analizleri yapmak için üç farklı tipte grafik kullanılmıştır.

Simple Multi-Plot: Bir çıkış verisini basitçe grafiğe dökmek için kullanılır. Bir veya daha fazla veriyi üst üste çizebilir. Şekil 4.12'de örnek bir grafik verilmiştir. Üst üste çizilen verilerin y-değerleri arasında anlamlı bir ortaklık varsa karşılaştırma yapmak için de uygundur. Üst üste çizim için Ayar ekranındaki "Add Plots" check butonu basılı olmalıdır.



Şekil 4.13. Simple Multi-Plot türü çizim örneği

Multi-Y Coordinates: Farklı değerlerdeki verileri karşılaştırmak veya aynı anda görmek için kullanımı uygundur. Örneğin aynı anda çıkış yük gerilimi ile bobin üzerinden geçen akımı birlikte görmek istenebilir. Bu iki değer hem yön hem büyüklük olarak farklıdır. Bu tip grafikte aynı anda birden çok Y eksenini bulduğundan birlikte görmek ve analiz yapmak mümkündür. Örnek bir çizim Şekil 4.13’de görülmektedir.



Şekil 4.14. Multi-Y Coordinates türü çizim örneği

Stack: Y eksenleri bir önceki grafik türünde yan yana iken bu türde üst üste yığılmıştır. Bir öncekine grafik türüne göre farklı açıdan bir karşılaştırma yapabildiği için eklenmiştir.

Osilaskop ekranı bu yazılıma gerçek zamanlı bir uygulama eklenmesi durumunda kullanmak üzere, seri porttan aldığı verileri bir osilaskop ekranı formatında göstermek için tasarlanmıştır. Mevcut şekliyle grafik ekranında görünen verileri osilaskop arayüzüyle çizmektedir.

Devre sekmesinde dönüştürücü ve yük tipine göre devrenin şematik gösterimleri verilmektedir.

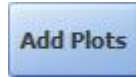
Ayarların bulunduğu alan ise grafik sekmelerinin hemen altındaki bölümdür. Grafiklerle ilgili ayarları ve gösterim seçeneklerini içerir. Bu bölümdeki ayarlar kısaca şöyledir:



: Grafik ekranındaki tüm çizimleri siler ve verileri temizler.



: Yapılan değişiklikleri (renk, kalınlık vb.) grafik ekranında görmek için ekranın “yenileme” butonudur. Bazı değişikliklerden sonra kullanılması gerekir.



: Bu seçim bir checkbox'tur. Basılı olması durumunda yapılan çizimler ekranda üst üste eklenir. Karşılaştırma ve analiz yapılıyorsa kullanılmalıdır. Basılı değilse her grafik çizilmeden önce ekrana “reset” işlemi yapılır.



: Bu seçim bir checkbox'tur. Basılı olması durumunda buton üzerinde çek işareti görünür ve “yakınlaştırma” işlevi aktif olur. Fare sol tuşla çizilecek dörtgen içindeki grafik alanı yaklaştırılır. Uzaklaşma için fare sağ tuşa basılır. “Tutma” aracı ile aynı anda aktif olmaz.



: Bu seçim bir checkbox'tur. Basılı olması durumunda üzerinde çek işareti görünür ve “Tutma” işlevi aktif olur. Bu işlev ile ekranda görülen grafik/ler fare sol ile tutularak hareket ettirilebilir. “Yakınlaştırma” aracı ile aynı anda aktif olmaz.



Seçilen hedef çıkış değerinin(Vout ya da Vref) grafik üzerinde gösterilip gösterilmeyeceğini belirler.

Export as txt file

: Her çıkışın karşısında bulunan bu butona basıldığında o çıkışa ait veriler bir txt dosyaya aktarılır.

Label

: Grafik üzerin görüntülenen bir veriye etiket vermek için kullanılır. Verilecek etiketin neler içereceği bu buton üzerinde fare sağ tuş yaparak seçilir.

☒ Vout

: Herhangi bir çıkış verisi için grafikte yer alıp almayacağını belirler. Bu alanda bir veri işaretli değilse grafikte görülmez. Her dönüştürücü ve kontrol tekniği için burada görüntülenecek çıkış değişkenleri menüden Settings-Scope/Plot Options-Plot Options seçeneği ile ulaşılabilecek penceredeki kontrol kutuları işaretlenerek belirlenebilir.

192; 0; 0

: Her çıkış verisi ve zemin rengi için farklı renkler kullanılabilir. Bu renk ayarı için aynı hizada bulunan renk seçim kutusu kullanılır. Renk seçimlerinin görülebilmesi için “yenileme” işlemi yapılır.

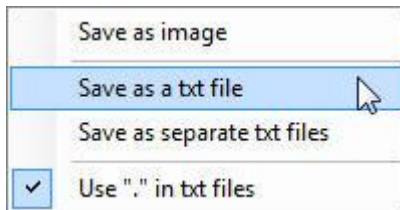
Thickness : 3 px

: Grafikteki bir verinin çiziminde kullanılan çizginin kalınlığını belirler. Kalınlık seçimleri “yenileme” işlemi gerektirir.

☐ No Y Grid ☒ Major Y Grid ☐ All Y Grid

: Hem X hem Y eksenini için kılavuz çizgi ayarı yapılmasını sağlar.

Şekil 4.14.’de ise grafik üzerinde sağ tuş yapıldığında açılan menü görülmektedir.



Şekil 4.15. Grafik üzerinde sağ tuş yapıldığında açılan menü.

Bu menüdeki “Save as image” ile çizilen grafiği Jpg, Bmp, Gif, Tiff, Png, Emf formatlarından biri ile resim olarak kaydetmemizi sağlar. “Save as txt file” seçeneği ile grafik ekran üzerinde çizili olan tüm veriler tek bir txt dosyaya yazdırılır. Bu seçimle kaydedilmiş bir txt dosya Şekil 4.15’de görülmektedir.

Time	Vout	Ib	VtrgOut
0	0.0	0.0	0
0.0005	0.00259004152892562	7.59619136363636	1.28
0.001	0.00517834684061702	11.3885692718574	2.56
0.0015	0.00862768255326446	15.1771289191172	3.84
0.002	0.0129371778342979	18.9618655133725	5.12
0.0025	0.0181059607670939	22.7427742760808	6.4
0.003	0.024133158354046	26.5198504421971	7.68
0.0035	0.0310178965196341	30.2930892601713	8.96
0.004	0.0387593001134934	34.062485991946	10.24
0.0045	0.0473564929134824	37.8280359129529	11.52
0.005	0.056808597628751	41.5897343121108	12.8
0.0055	0.0671147359028069	45.3475764918224	14.08
0.006	0.0782740283165825	49.1015577679715	15.36
0.0065	0.0902855943915001	52.8516734699204	16.64
0.007	0.103148552592537	56.5979189405066	17.92
0.0075	0.11686202033129	60.3402895360401	19.2
0.008	0.13142511396904	64.0787806263008	20.48
0.0085	0.146836948819811	67.8133875945348	21.76
0.009	0.163096639153439	71.5441058374521	23.04

Şekil 4.16. Verilerin aktarıldığı bir txt dosyası örneği

“Save as separate txt file” seçeneği ile ise grafik ekran üzerinde çizili olan tüm veriler ayrı ayrı txt dosyalara yazdırılır. Yani Şekil 4.15.’de görülen her bir veri grubu ayrı txt dosyalara yazdırılır. Bu menüde yer alan son seçenek “Use ‘.’ in txt files” ile txt dosyalara aktarım işlemlerinde ondalık ayracı olarak nokta veya virgöl kullanımı arasında tercih yapmamızı sağlar. Bu seçenek verileri başka amaçla farklı programlarda (Excel, MatLab vb.) analiz etmek için kullanmak istediğimizde ondalık ayracı sebebiyle yaşanacak problemleri önlemeyi hedeflemektedir.

Durum Çubuğu

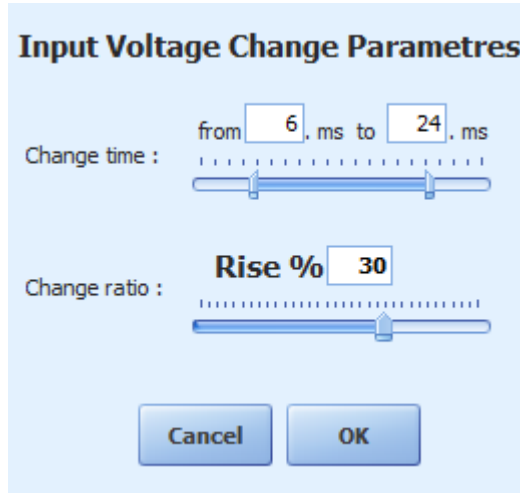
Çalışma penceresinin en altında yer alan durum çubuğu yaptığımız seçimleri takip edebilmemizi sağlar. Yani dönüştürücü, yük, denetleyici ve kontrol tekniği türlerinden hangisi ile simülasyonun çalışmakta olduğunu bir cümlecikle bilgilendirir. Bir işlem ya da süreç devam ederken de süreç çubuğu (progress bar) görünür.

Çalışma Parametreleri Girişleri

Birkaç panel içeren bu bölüm içinde çok sayıda seçim ve çalışma parametresi belirlenebilmektedir. Bu bölümler ve parametre ayarları:

Çalışma Parametreleri (Operating Parameters): Devrenin çalışma ve benzetim şartlarını belirleyen girişlerdir. Bu panelde V_{in} yani giriş gerilimi sabit/değişken

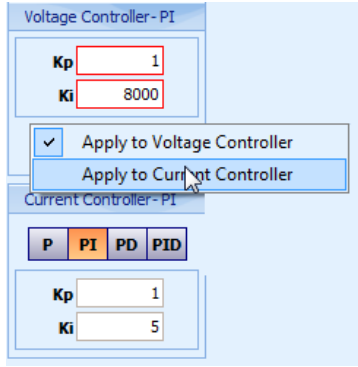
olarak ayarlanabilir. Bu işlem için V_{in} 'in hemen yanındaki checkbutona basılması yeterlidir. Butona basıldığında üzerindeki ikonun değiştiği görülecektir. Buton değişim ikonu varken üzerinde fare sağ tuş yapılırsa değişimle ilgili parametrelerin ayarlanabileceği Şekil 4.16.' de görülen pencere gelir.



Şekil 4.17. V_{in} için değişim parametrelerini ayarlama penceresi

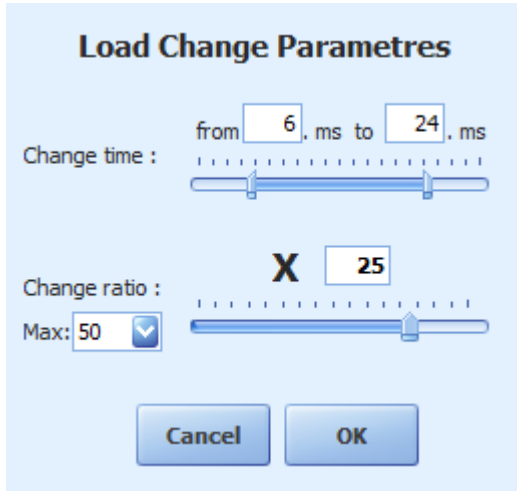
Devre Parametreleri (Circuit Parameters); Devre elemanlarının değerlerinin girişlerinin yapıldığı paneldir. Bazı devre elemanları için hemen yanındaki giriş alanlarına benzetim sonuçlarını gerçek uygulamaya yaklaştırmak için iç direnç değerleri girilebilir.

Denetleyici Parametreleri (Voltage/Current Controller Parametres): Seçime göre P, PI, PD veya PID denetleyicilerin katsayıları girişleri içindir. Kontrol tekniği olarak Ortalama Akım Kontrol seçilmişse ve denetleyiciyi bir yapay zekâ tekniği ile kontrol edilecek şekilde ayarlanırsa bu tekniğin hangi denetleyiciye (akım/gerilim denetleyici) uygulanacağı üzerinde sağ tuş yaparak belirlenebilir. (Bkz. Şekil 4.17.)



Şekil 4.18. Algoritmanın uygulanacağı denetleyici seçimi

Yük Parametreleri (Load Parameters): Seçilen yük tipine göre farklı giriş değerleri için kullanılan paneldir. Bu panelde yük tipi “değişken/change” seçilmişse değişimin uygulanabileceği parametrenin altında “Change” butonu açılır. Bu buton ile Şekil 4.18’deki pencereye ulaşılır. Bu pencerede ayrıntılı bir şekilde değişimin ne zaman ve ne kadar olacağını ayarlanır.



Şekil 4.19. Yük değişimi parametreleri için ayar penceresi

Birden fazla elemanına değişim uygulanabilen yük tiplerinde “Change” butonu üzerinde fare sağ tuş yaparak “No” yapılabilir. Bu durumda o elemana değişim uygulanmaz. Tek değişim elemanına sahip yük tiplerinde fare sağ tuş çalışmayacağı gibi çoklu değişim elemanlı yüklerde de tüm elemanlar için yük değişimi “No” yapılamaz. Bu değişikliğin toplu iptali anlamına geldiği için yük tipini “Change” yerine “Constant” ayarlamak daha doğru olur.

Analiz Paneli: Seçime bağlı olarak değişen iki farklı panel yer alır. Panellerde bulunan bu hesaplamalar dönüştürücünün performansını etmek için kullanılır. Şekil 4.19’da yer alan “Normal Durum” ekranında şu değerler hesaplanır:

Analysis - Nominal Case		
Rise Time	1,111	ms
Settling Time	3,261	ms
Overshoot	11,5	%
Stady State Error	5,715	V

Şekil 4.20. Normal durum analizi için performans parametreleri

Yükselme Zamanı (Rise Time): Çıkış geriliminin, referans geriliminin %10’dan %90’ına yükselmesi sırasında geçen zaman olarak ifade edilir.

Oturma Zamanı (Settling Time): Başlangıç anından itibaren, çıkış geriliminin referans sinyalinin $\pm\%5$ arasında bir değerde çok az bir değişimle hareket etmeye başladığı ilk ana kadar geçen süredir. Buradaki değişim miktarını hesaplarken bir öneri olarak yüzde standart sapma kullanılmıştır. Yüzde standart sapması ortalaması farklı olan grupların standart sapmalarının karşılaştırılmasında kullanılan bir istatistiksel yöntemdir. Burada da çıkış geriliminin referans sinyalinin $\pm\%5$ arasında bir değere ulaştıktan sonraki değişimler için her örnekleme aralığından itibaren kalan veri için değişim miktarı yüzde standart sapması kullanılarak hesaplanmıştır. Her hesaplamada kalan verinin ortalaması değiştiği için değişimin yeterince azalıp azalmadığını (oturmanın olup olmadığını) anlamak ve karşılaştırabilmek için yüzde standart sapması hesaplanmıştır. Yüzde standart sapmanın %5’in altında olması durumu oturmanın gerçekleşmesi olarak kabul edilmiş ve geçen süre oturma zamanı olarak belirlenmiştir. Bu yöntemle farklı çıkış grafikleri incelenerek anlık sonuçlar bulunmuştur. Gerçek zamanlı bir uygulamada da bu yöntem kullanılabilir. Önce gerçek zamanlı veriler belirli bir aralıkla (oturmanın gerçekleşmesi beklenen süre ile orantılı) kaydedilir. Kaydedilen veri üzerinde yukarıdaki hesaplamalar yapılır. Örneğin 5 ms civarında oturma bekleniyorsa bunu 6–8 katı süre kadar 30–40 ms gerçek zamanlı veriler kaydedilip bu veriler üzerinde metot uygulanabilir.

Aşım Miktarı (OverShoot): Çıkış gerilimi ile referans gerilimi arasında oluşan en büyük farkın referansa göre yüzdesi bulunarak hesaplanır.

Kalıcı Durum Hatası (Stady State Error): Oturma zamanından sonra çıkış sinyali ile referans arasında oluşan sürekli farktır. Yani çıkışın düzeltilemeyen miktarını gösterir. Oturma zamanında kullanılan yüzde standart sapma bu hesaplamada da kullanılmıştır. Çıkışın kalıcı hale geldiği durumdan sonra sinyalin ortalaması ile referansa arasındaki farka, ulaştığı en yüksek ve en düşük değer arasındaki farkın eklenmesi ile bulunmuştur. Kalıcı hale geline nokta ise yüzde standart sapmanın %0,02'nin altına indiği nokta olarak kabul edilmiştir.

Analysis - Voltage-Load Change		
Voltage Sag	14,80	%
Voltage Overshoot	7,95	%
Correction Time1	2,42	ms
Correction Time2	3,84	ms

Şekil 4.21. Giriş gerilimi veya yük değişimi analiz parametreleri

Giriş gerilimi veya yük değişimlerinde dönüştürücünün performansı farklı parametrelerle değerlendirilir. Bu parametreleri gösteren örnek panel görüntüsü Şekil 4.20.'de görünmektedir.

Voltaj Düşümü (Voltage sag): Giriş gerilimi/Yük değişimi anında çıkış geriliminde gözlenen düşüşün, dönüştürücünün tepkisiyle tekrar yükselmeye başladığı sırada ortaya çıkan gerilim farkının referansa yüzdesi hesaplanarak bulunur.

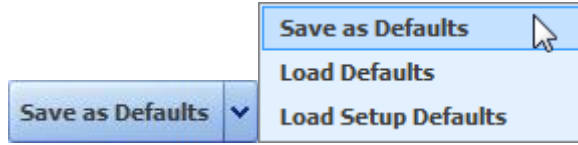
Voltaj Aşımı (Voltage Overshoot): Giriş gerilimi/Yük değişimi anında çıkış geriliminde gözlenen yükselişin, dönüştürücünün tepkisiyle tekrar düşmeye başladığı sırada ortaya çıkan gerilim farkının referansa yüzdesi hesaplanarak bulunur.

Düzeltilme Zamanı 1 (Correction Time 1): Giriş gerilimi/Yük değişimi anında çıkış geriliminde gözlenen düşüşle, dönüştürücünün tepkisiyle tekrar yükselerek önceki değere ulaşması sırasında geçen zamandır.

Düzeltilme Zamanı 2 (Correction Time 2): Giriş gerilimi/Yük değişimi anında çıkış geriliminde gözlenen yükselişle, dönüştürücünün tepkisiyle tekrar düşerek önceki değere ulaşması sırasında geçen zamandır.

Bu bölümde yer alan tüm paneller için geçerli olmak üzere altı çizili birim etiketleri tıklanarak birim değişikliği yapılabilen etiket anlamındadır. Tıklamaya devam ederek istenilen birime ayarlanabilir.

Sağ alt köşede yer alan dropdown buton seçenekleri ile dönüştürücülerin çalışma şartları ile elde edilen önemli analiz değerleri için veritabanında ön değerler (defaults) oluşturulabilir. (Bkz. Şekil 4.21.):



Şekil 4.22. Ön değer (defaults) kaydetme ve çağırma seçenekleri

Save as default: Simülatörün çalışması sırasında herhangi bir zamanda bu buton tıklanabilir. Tıklandığı an için tüm panellerdeki parametreler çalışma şartları olarak belirlenen seçim için bir ID ile ayrı ayrı veritabanına kaydedilir.

Load Defaults: Daha önceden kaydedilen bir seçime bağlı ön değerleri (defaults) geri çağırma ve parametre olarak belirleyebilmeyi sağlar. Çalışma sırasında esneklik olması için her farklı durum bir ID olarak belirlendiği için bir milyon civarında farklı ön değer ataması yapılabilir. Veritabanı bu ihtiyaca cevap verecek biçimde tasarlanmıştır.

Load Setup Defaults: Kurulum ayarı olarak bulunan ve ayrı bir veritabanında saklanan ön değerlerin (defaults) çağrılarak kullanımını sağlar.

4.6. Bilgisayar Ara Yüzü Yazılımı ile Yapılmış Benzetim Çalışmaları

Tekniklerin Karşılaştırılması

DC/DC Tool simülatöre gerilim mod, ortalama akım mod, tepe akım (rampalı/rampasız), histerisiz akım, vadi akım ve V^2 kontrol olmak üzere yedi farklı kontrol tekniği eklenmiştir. Bu teknikleri karşılaştırırken Şekil 4.22 ve Şekil 4.23’de gösterilen değerler sabit tutulmuştur.

Operating Parameters	
f (pwm)	20 kHz
Time	30 ms
Sample	100 /t
Vin	380 V
Vout	100,0 V

Şekil 4.23. Karşılaştırmada kullanılan çalışma şartları

Circuit Parameters	
L	0,05 mH
rL	100 mΩ
C	2200,0 μF
rC	33 mΩ

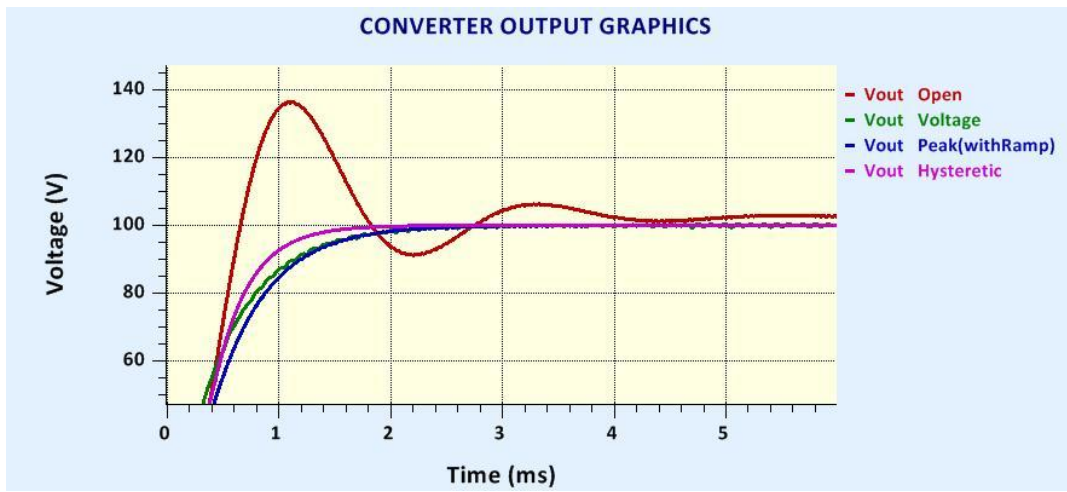
Şekil 4.24. Karşılaştırmada kullanılan devre elemanları

Karşılaştırma yükselten (buck) dönüştürücü üzerinde klasik PI kullanılarak yapılmıştır. Her denetim tekniğinin çalışma prensibi farklı olduğundan farklı K_p ve K_i katsayıları kullanılmıştır. Elde edilen performans değerleri Çizelge 4.1.’de verilmiştir.

Çizelge 4.1. Denetim tekniklerinin performans değerleri (sabit yük)

Denetim Tekniği Ve PI Katsayıları	Yük (R)	Yükselme Zamanı (ms)	Oturma Zamanı (ms)	Aşım Miktarı (%)	Kalıcı Durum Hatası (V)
Açık Devre	100 Ω	0,442	3,397	36,38	2,389
Voltaj Mod ($K_p= 50$ $K_i= 1000$)		1,080	1,558	0,42	0,698
Tepe Akım -with ramp ($K_p= 18$ $K_i= 300$)		1,096	1,598	0,14	0,094
Tepe Akım-without ramp ($K_p= 18$ $K_i= 300$)		1,069	1,564	0,14	0,204
Ortalama Akım ($K_p= 150$ $K_i= 8000$)		0,292	0,419	0,01	0,279
Vadi Akım ($K_p= 500$ $K_i= 30000$)		0,608	1,153	0,00	0,268
V^2 Akım ($K_p= 4500$ $K_i= 15000$)		0,816	1,253	0,00	0,286
Histerisiz Akım ($K_p= 100$ $K_i= 3000$)		0,808	1,103	0,00	0,248

Çizelge 4.1 incelendiğinde akım mod kontrol tekniklerinin genel performanslarının daha iyi olduğu görülmektedir. Bu tekniklerde görülen kalıcı durum hatasının çok düşük olması ve oturma zamanının çok kısa olması oldukça yüksek bir performans olarak değerlendirilir. Ayrıca aşım miktarı yok denilecek kadar az olması da önemlidir. Bu tekniklerden dördünün grafik olarak karşılaştırılması Şekil 4.24’da verilmiştir.

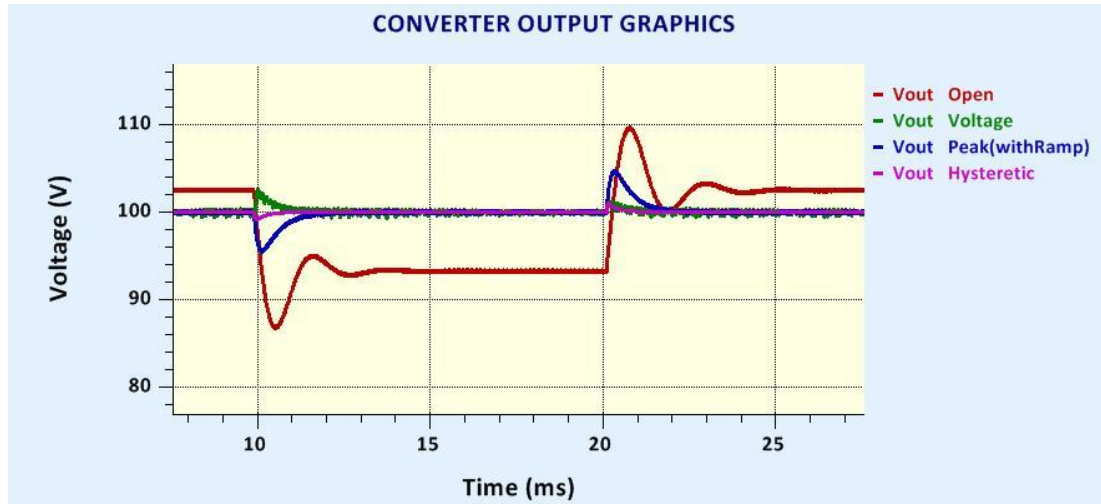


Şekil 4.25. Denetim tekniklerinin çıkış değerleri (sabit yük)

Bir başka karşılaştırma ise yük değişimi ile yapılmıştır. Tekniklerin 100 Ω direnç üzerinde simülasyonun belirli bir aralığında 100 kat düşürülmesi durumunda nasıl bir tepki verdiği incelenmiştir. Elde edilen değerleri gösteren Çizelge 4.2.'ye bakılırsa yine akım mod tekniklerinin daha iyi tepki verdiği görülür. Bu durum Şekil 4.25’de grafiksel olarak görülmektedir.

Çizelge 4.2. Denetim tekniklerinin performans değerleri (değişken yük)

Denetim Tekniği Ve PI Katsayıları	Yük (R)	Voltaj Düşümü (%)	Aşım Miktarı (%)	İlk Düzeltilme (ms)	Son Düzeltilme (ms)
Açık Devre	100 Ω (/100'e düşüm)	14,21	15,60	1,811	2,01
Voltaj Mod (Kp= 50 Ki= 1000)		2,70	1,76	1,098	0,898
Tepe Akım -with ramp (Kp= 18 Ki= 300)		4,96	4,71	0,962	0,972
Tepe Akım-without ramp (Kp= 18 Ki= 300)		5,01	4,76	1,042	1,023
Ortalama Akım (Kp= 150 Ki= 8000)		0,40	0,92	0,223	0,259
Vadi Akım (Kp= 500 Ki= 30000)		1,43	1,82	0,543	0,623
V ² Akım (Kp= 4500 Ki= 15000)		7,12	8,60	1,874	1,902
Histerisiz Akım (Kp= 100 Ki= 3000)		0,29	0,52	0,312	0,427



Şekil 4.26. Denetim tekniklerinin çıkış değerleri (sabit yük)

Burada karşılaştırma amaçlı olarak yalnızca PI katsayıları değiştirilmiş diğer şartlar sabit tutulmuştur. Ancak diğer şartlarda da değişikliğe gidilirse performanslarda ciddi değişiklikler olduğu görülmüştür. Yani tekniklerin performansı devre elemanlarının ve çalışma şartlarının değişmesiyle değişmektedir. Bu durum ise kullanım sahasına göre farklı tekniklerin biri birine göre daha iyi performans sergileyeceği anlamına gelir.

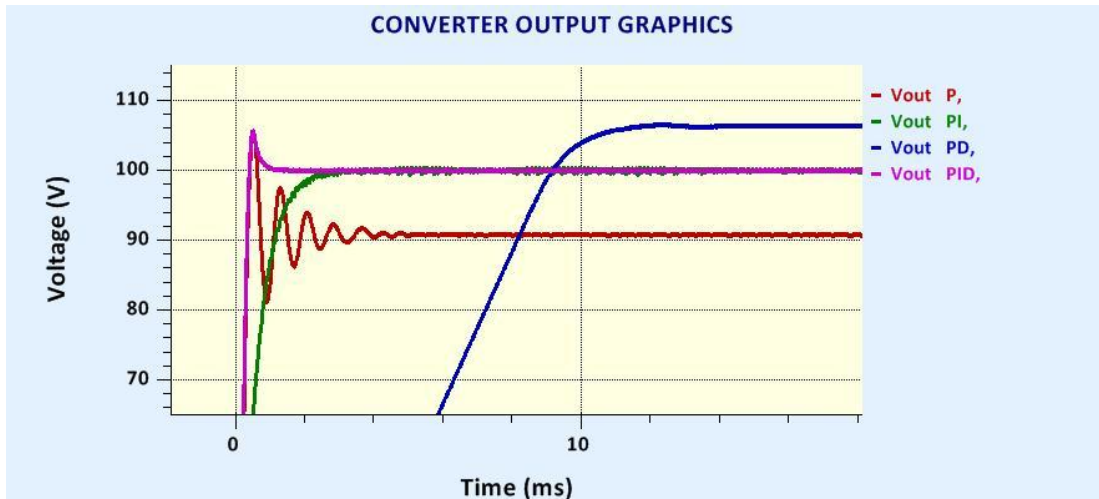
Denetleyicilerin Karşılaştırılması

Tekniklerin karşılaştırılması sırasında kullanılan aynı çalışma şartları altında düşüren (buck) dönüştürücünün voltaj kontrol mod ile P, PI, PD ve PID klasik denetleyicilerin nasıl bir performans gösterdiği incelenmiştir. Çizelge 4.3.'de bu değerler çizelge halinde verilmiştir.

Çizelge 4.3. Denetleyicilere göre performans değerleri

Denetleyici Katsayıları (Voltaj Mod)		Yük (R)	Yükselme Zamanı (ms)	Oturma Zamanı (ms)	Aşım Miktarı (%)	Kalıcı Durum Hatası (V)
P	$K_p=1,05$	100 Ω	0,27	1,748	5,33	9,444
PI	$K_p=50$ $K_i=1000$		1,08	1,558	0,42	0,698
PD	$K_p=1$ $K_d=0,099$		6,885	9,655	6,59	6,514
PID	$K_p=50$ $K_i=2670$ $K_d=0,095$		0,257	0,385	0,65	0,172

Çizelge 4.3. incelendiğinde PI ve PID denetleyicilerin iyi bir performans gösterdiği görülmektedir. Nispeten daha kötü olsa da uygulamanın şartlarına bağlı olarak P denetleyicinin performansı da uygulama için yeterlidir. Performansların grafiksel karşılaştırılması Şekil 4.26'da yapılmıştır.



Şekil 4. 27. Denetleyicilere göre çıkışların grafiksel gösterimi

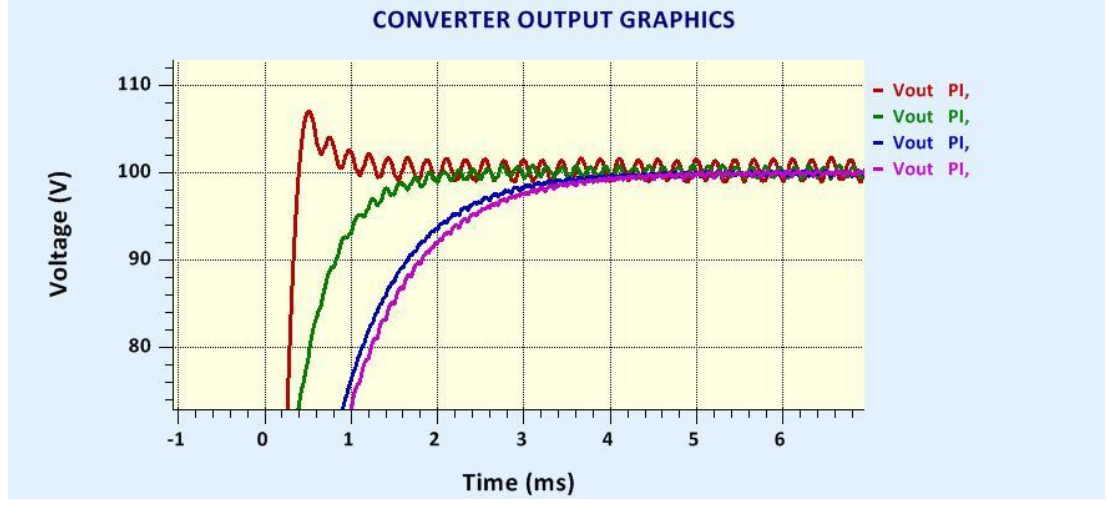
Sezgisel Algoritmaların Karşılaştırılması

Sezgisel algoritmaların karşılaştırılması sırasında denetleyici karşılaştırmasında kullanılan aynı parametreler kullanılmıştır. Denetleyici tipi olarak PI seçilmiştir. Elde edilen performans değerleri Çizelge 4.4’de gösterilmiştir.

Çizelge 4.4. Sezgisel algoritmalarla göre performans değerleri

Sezgisel Algoritmalar (Voltaj Mod, PI)	Yük (R)	Yükselme Zamanı (ms)	Oturma Zamanı (ms)	Aşım Miktarı (%)	Kalıcı Durum Hatası (V)
Klasik	100 Ω	1,08	1,558	0,42	0,698
PSO (Particle Swarm Optimization)		0,602	0,821	0,01	0,287
GA (Genetic Algorithm)		0,707	1,021	0,03	0,455
YAK (Artificial Bee Colony)		0,595	0,859	0,01	0,220

Çizelge 4.4’de görüldüğü gibi sezgisel algoritma kullanılmadan klasik bir PI ile elde edilen değerler kötü bir performansı gösteriyor. Sezgisel algoritma kullanıldığında performansın yükseldiği görülmektedir.(Bkz. Şekil 4.27.) Özellikle kalıcı durum hatası ve aşım miktarı çok iyidir. Oturma zamanı ise kayda değer derecede iyileşmektedir. Sezgisel algoritmaları birbiri ile karşılaştırdığımızda GA’nın çok az bir farkla geride kaldığı ve PSO ile YAK’ın öne çıktığı görülmektedir. Ancak algoritmanın ayarlanabilen parametreleri ile bu değerler birbirine göre daha iyi konuma getirilebilmektedir. Dolayısı ile seçim yapılırken hedeflenen uygulamanın çalışma şartları dikkate alınmalıdır.



Şekil 4.28. Sezgisel algoritmalar ile elde edilen çıkışların grafiksel gösterimi

5. SONUÇ VE ÖNERİLER

Bu çalışmada DA/DA dönüştürücülerde güç dönüştürücüleri ve onların denetiminde kullanılan kontrol metotları ile ilgili kapsamlı analizleri yapmaya olanak sağlayan yeni bir DA/DA dönüştürücü tool' u geliştirilmiştir. DC/DC Tool adı verilen bu yazılım; düşüren, yükselten, düşüren/yükselten, CÜK ve Sepic dönüştürücüler için farklı çalışma şartları altında kontrol teknikleri, geleneksel denetleyiciler ve sezgisel denetleyicilerin nominal durum ve giriş/yük değişim analizlerine izin veren bir yapı ihtiva eder. Tool içerisinde gerilim mod kontrol, tepe akım kontrol, ortalama akım kontrol, vadi akım kontrol, V^2 kontrol ve histerisiz kontrol olmak üzere 6 farklı kontrol tekniği kullanılmış olup, bu tekniklerin içerisinde kullanılan geleneksel P, PI, PD veya PID denetleyicilerin ve PSO, GA, YAK algoritmaları tabanlı P, PI, PD ve PID denetleyicilerin sistem performansı üzerindeki etkileri grafiksel ve nümerik olarak tasarlanmış olan grafik ara yüzü üzerinden izlenebilir.

Geliştirilen tool gerçek zamanlı uygulamaların geliştirmesinde yapılacak araştırma-geliştirme çalışmalarında kullanılabilir. Ayrıca özel olarak tasarlanmış grafik arayüzü ve devre şemaları ile elde edilen görsellik yoluyla ilgili eğitim çalışmalarına da önemli katkılar sağlayabilir. Bu çalışmadaki yazılım geliştirilmeye uygun bir şekilde yapılandırılmıştır. Örneğin farklı dönüştürücüler veya denetim teknikleri durum denklemlerini elde ettikten sonra kolaylıkla eklenebilir. Aynı şekilde algoritmalar ayrı bir şekilde tasarlandığı için tool' a başka bir sezgisel algoritma da eklenebilir. Bu durum algoritmalarda farklı bir uygunluk fonksiyonu denemeyi/kullanmayı da kolaylaştırmaktadır.

Çalışmanın simülatör olarak ulaştığı kapsam bu yönlerde geliştirilebileceği gibi etkileşimli olarak çalışabileceği gerçek zamanlı uygulama devreleri eklenmesiyle farklı bir boyut kazandırılabilir. Bu şekilde eğitim amaçlı bir deney seti elde edilebilir. Devre çizimlerine eklenecek animasyonlar eğitim amaçlı kullanımına önemli katkılar sağlayabilir.

KAYNAKLAR

1. Christiansen, D., Alexander, Ch.K.,. Standard Handbook Of Electronic Engineering. s.l. : **Mcgraw-Hill**, 2005. Fifth Edition. ISBN 0-07-138421-9.
2. Mammano, R.,. Switching Power Supply Topology Voltage Mode vs. Current Mode. **Unitrode Design Note**. 1994.
3. Sun, J., Choi, B.,. Averaged modeling and stability prediction for peak-current control. in Proc. **IEEE Power Electron. Spec. Conf.** Haziran 2005. 2764–2770.
4. Holland, B.,. Modeling, Analysis and Compensation of the Current-Mode Converter. **Proceedings of Powercon** . 1984. Cilt 11, 1-2.
5. Dixon, L. H.,. Average current mode control of switching power supplies. **Unitrode Power Supply Design Seminar**. Merrimack NH : Unitrode Corp., 1988. Cilt 5, 1-14.
6. Abu-Qahouq, J., Mao, H., Batarseh, I.,. Multiphase Voltage–Mode Hysteretic Controlled Dc–Dc Converter With Novel Current Sharing. **Ieee Trans. Power Electron**. 2004. Cilt 19 (6), 1397–1407.
7. Castilla, M., García de Vicuña, L., Guerrero, J.M., Matas, J., Miret, J.,. Design of voltage-mode hysteretic controllers for synchronous buck converters supplying microprocessor loads. **Proc. Inst. Elect. Eng., Elect. Power Appl.** 2005. Cilt 152, 5 s1171–1178.
8. Zhou, G., Xu, J., Jin, Y., Sha, J.,. Stability Analysis of V2 Controlled Buck Converter Operating in CCM and DCM. **IEEE Int. Symp. On Communications, Circuits and Systems**. 2010. s. 551-555.
9. Zhou, G., Bao, B., Xu, J., Jin, Y.,. Dynamical analysis and experimental verification of valley current controlled buck converter. **Chin. Phys. B** Vol.19. 2010. Cilt 5 s.1-7.
10. Chenhu, Y., Jianru, W., Hailiang, Z.,. Study on the Dynamical Characteristic for Current-Feedback BUCK Converter. **IEEE Int. Conf. On Control, Automation and Systems Engineering**. 2011. Cilt s 1-4.
11. Er, M.J., Sun, Y.L.,. Hybrid Fuzzy Proportional-Integral Plus Conventional Derivative Control of Linear and Nonlinear Systems. **IEEE Trans. Ind. Electron**. 2001. Cilt 48(6) s. 1109–1117 .
12. Li, W.,. Design of a Hybrid Fuzzy Logic Proportional Plus Conventional Integral-derivative Controller. **IEEE Trans. on Fuzzy Systems**. 1998. Cilt 6(4) s.449–463.

13. Atacak I. Design of a hybrid type-2 fuzzy logic/proportional-integral controller for single-phase three-wire inverter system. **Scientific Research and Essays** Vol. . 2011. Cilt 6(23), s. 5049-5064.
14. Jan, R.M., Tseng, C.S., Liu, R.J.,. Robust PID control design for permanent magnet synchronous motor: a genetic approach. **Electr. Power Syst. Res.** 2008. Cilt 78 s. 1161–1168.
15. Elmas, C., Yigit, T.,. Genetic algorithm based on-line tuning of a PI controller for a switched reluctance motor drive. **Elect. Power Compon.Syst.** 2007. Cilt 35 s. 675-69.
16. Gadoue, S.M., Giaouris, D., Finch, J.W.,. Tuning of PI Speed Controller in DTC of Induction Motor Based on Genetic Algorithms and Fuzzy Logic Schemes. D. Manolakis & A. Gogoussis **5th Int. Conf. on Tech. & Automation.** Thessaloniki : s.n., 2005 s. 85-90.
17. Lin, C.L., Jan, H.Y., Shieh, N.C.,. GA-Based Multiobjective PID Control for a Linear Brushless DC Motor.**IEEEUASME Trans. Mechatronics.** 2003.Cilt8 s.56-65.
18. Nasri, M.,, Nezamabadi, H., Maghfoori, M.,. A PSO-based optimum design of PID controller for a linear brushless DC motor. **Proc World Acad. Sci. Eng. Technol.** 2007. Cilt 20 s.211-215.
19. Moghaddas, M., Dastranj, M., Changizi, N., Rouhani, M.,. PID control of DC motor using particle swarm optimization (PSO) algorithm. **The J. of Math. and Comp. Science.** 2010. Cilt 1 s. 386-391.
20. Thangaraj, R., Chelliah, T.R., Pant, M., Abraham, A., Grosan, C.,. Optimal gain tuning of PI speed controller in induction motor drives using particle swarm optimization. **Logic Journal of IGPL.** 2010. Cilt 19 s.343-356.
21. Mohammed, A.J.,. A Particle Swarm Optimization (PSO) Based Optimum of Tuning PID Controller for a Separately Excited DC Motor (SEDM). **Eng. & Tech. Journal** . 2011. Cilt 29 s.3321-3333.
22. Güvenç, U., Altun, B.E., Duman, S.,. Optimal power flow using genetic algorithm based on similarity. **Ener Educ Sci Tech-A** . 2012. Cilt 29 s.1–10.
23. Demir, H., Ozturk, A., Kuru, L., Kuru, E.,. Weight optimization of a dry type core form transformer by using particle swarm optimization algorithm. **Ener Educ Sci Tech-A** . 2012. Cilt 29 s.1063-1072.

24. Mohan, N., Undeland, T. M., Robbins, W. P., (Çevirenler: Tuncay, N., Gökaşan, M., Bogosyan, S.). Güç Elektronikliği Çeviriciler, Uygulamalar Ve Tasarım. s.l. : **Literatür**, 2003 s.173-198, 323-342, 360-365, 725-751.
25. Pressman, A. I.,. Switching Power Supply Desing. **McGraw-Hill Inc.**, New. 1991 s.4-7, 9-35, 53-141, 267-317.
26. Rashid, M. H.,. Power Electronics Handbook. **Academic Pres.** San Diego : s.n., 2001 s.211-223, 487-494.
27. Saurabh, K.,. Analysis, Design and Modeling of DC/DC Converter using Simulink. Master's Thesis **Oklahoma State University**. 2004.
28. Jeff, F.,. Designing of DC/DC converters based on SEPIC topology. Analog Applications Journal, Power management, **Texas Instruments Incorporated**. 2008. Cilt 4 s. 18-23.
29. Kaya, A. Güç Elektoniğinde Konvertör Kontrol Devreleri . Yüksek Lisans Tezi . Ankara : **Gazi Üniversitesi**, 1997.
30. Yüksek, A. Anahtarlama Güç Kaynağı İçin Bulanık Mantık Denetleyicisi Tasarımı. Yüksek Lisans Tezi. Ankara : **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, 2002. Cilt S.34.
31. Işık, H. Gerilim ve Akım Sınırlama Özellikli Anahtarlama Modlu Güç Kaynağının Tasarımı ve Gerçekleştirilmesi. Yüksek Lisans Tezi. Ankara : **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, 1998.
32. Kennedy, J.C. ve Eberhart, R. Particle Swarm Optimization,. **Proc. IEEE Conf. Neural Networks**. Perth, Australia : s.n., 1995. 1942-1948.
33. Kennedy, J., Eberhart, R. C., Shi, Y. Swarm Intelligence. s.l. : **Morgan Kaufmann Publishers**, 2001.
34. Eberhart, E. ve Shi,Y.,. Tracking and optimizing dynamic systems with particle swarms. **In Proceedings of the IEEE Congress on Evolutionary Computation**. Piscataway, NJ : s.n., 2001. S 94-100.
35. Gencer, C. ve Bal, Ö.,. Parçacık Sürü Optimizasyonu Algoritması İle U-Tipi Hat Dengeleme. **KHO Bilim Dergisi**. 2010. Cilt 20, 1 Sayfa:200.
36. Karaboğa, D. An idea based on honey bee swarm for numerical optimization. **Technical Report TR06**. Kayseri : Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

37. Akay, B. Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi. Bilgisayar Mühendisliği Anabilim Dalı Doktora Tezi. Kayseri : **Erciyes Üniversitesi Fen Bilimleri Enstitüsü**, 2009. Cilt Sayfa 53.
38. Holland, J.H. Adaptation in Natural and Artificial Systems. s.l. : **University of Michigan**, 1975.
39. Goldberg, D.A. Genetic Algorithms in Search, Optimisation and Machine Learning. s.l. : The University of Alabama, **Addison-Wesley Publishing Company.**, 1989.
40. Rechenberg, I. Evolutionstrategie (Evolution Strategy). **Frommann-Holzboog**. Stuttgart : s.n., 1973.
41. Koza, J.R. Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge : **MIT Press**, 1992.
42. Grefenstette, J. J. Optimization of Control Parameters for G.A. **IEEE Transactions on Systems, Man and Cybernetics**. 1986. Cilt 16(1):122-128.
43. Dorigo M., Maniezzo, V. ve Colorni A. Positive Feedback As A Search Strategy. **Technical Report N.91-016**. s.l. : Politecnico di Milano, 1991.
44. Dorigo M. Optimization, Learning and Natural Algorithms. PhD thesis. s.l. : Dipartimento di Elettrotecnica, **Politecnico di Milano, IT**, 1992.
45. Deneubourg J.L., Aron S., Goss S., ve Pasteels J.M. The Self-organizing Exploratory Pattern of the Argentine Ant. **Journal of Insect Behavior**. 1990. Cilt 3:159-168.
46. Dorigo M., Caro G. ve Gambardella L. Ant Algorithms for Discrete Optimization. **Artificial Life**, . 1999. Cilt 3, 137-172.
47. Düzakın, E., Demircioğlu, M. Araç Rotalama Problemleri ve Çözüm Yöntemleri. **Çukurova Üniversitesi İİBF Dergisi**. 2009. Cilt 13. Sayı:1. ss.68-87.
48. Storn, R., Price, K. Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Berkeley : **TechnicaReport TR-95-012**, International Computer Science Institute, 1995.
49. Aslantaş, V., Oz, A. Diferansiyel Gelişim Algoritması İle Tekil Değer Ayırışımına Dayalı Resim Damgalama. **ISC Turkey**. 2007. 220.

50. Mayer, D.G., Kinghorn, B.P., ve Archer A.A. Differential Evolution – An Easy and Efficient Evolutionary Algorithm for Model Optimisation. **Agricultural Systems**. 2005. 83, 315-328.
51. Sun, J., Zhang,Q. ve Tsang, P.K.E. DE/EDA: A New Evolutionary Algorithm for Global Optimization. **Information Sciences**. 2005. 169, 249-262.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : IRMAK, Ahmet
 Uyuğu : T.C.
 Doğum Tarihi ve Yeri : 06.07.1974 Ankara
 Medeni Hali : Evli
 Telefon : 0 (312) 3420719
 Faks : 0 (312) 3414101
 E-posta : ahmetirmak@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Marmara Üniversitesi/ Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bilgisayar Ana Bilim Dalı	1997
Lise	Yenimahalle Teknik Lisesi	1991

İş Deneyimi

Yıl	Yer	Görev
2001 – 2011	Milli Eğitim Bakanlığı	Bilgisayar Öğretmeni
2011 – Halen	Milli Eğitim Bakanlığı	Müdür Yardımcısı

Yabancı Dil

İngilizce

Hobiler

Bilişim Teknolojileri