

**EVİRİCİLERİN HARMONİK ELİMİNASYONUNDA  
OPTİMUM ANAHTARLAMA AÇILARININ  
GENETİK ALGORİTMALAR İLE ELDE EDİLMESİ**

**YÜKSEK LİSANS TEZİ**

**Nezih KÜÇÜK**

**Anabilim Dalı : ELEKTRİK MÜHENDİSLİĞİ**

**Programı : ELEKTRİK MÜHENDİSLİĞİ**

**OCAK 2007**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**EVİRİCİLERİN HARMONİK ELİMİNASYONUNDA  
OPTİMUM ANAHTARLAMA AÇILARININ  
GENETİK ALGORİTMALAR İLE ELDE EDİLMESİ**

**YÜKSEK LİSANS TEZİ  
Nezihe KÜÇÜK  
504031026**

**Tezin Enstitüye Verildiği Tarih : 25 Aralık 2006  
Tezin Savunulduğu Tarih : 31 Ocak 2007**

**Tez Danışmanı : Prof. Dr. Emin TACER**

**Diğer Jüri Üyeleri : Prof. Dr. Mehmet Sait TÜRKÖZ**

**Yrd. Doç. Dr. Deniz YILDIRIM**

**OCAK 2007**

## **ÖNSÖZ**

Tez konumun belirlenmesi aşamasından sonlandırma aşamasına kadar bana her konuda yardımcı olan tez danışmanım Prof. Dr. Emin TACER'e sonsuz teşekkürlerimi sunarım.

**Aralık 2006**

**Nezihe KÜÇÜK**

## İÇİNDEKİLER

<b>KISALTMALAR</b>	<b>v</b>
<b>TABLO LİSTESİ</b>	<b>vi</b>
<b>ŞEKİL LİSTESİ</b>	<b>vii</b>
<b>SEMBOL LİSTESİ</b>	<b>viii</b>
<b>ÖZET</b>	<b>ix</b>
<b>SUMMARY</b>	<b>xi</b>
<b>1. GİRİŞ</b>	<b>1</b>
<b>2. GENETİK ALGORİTMALAR</b>	<b>3</b>
2.1. Giriş	3
2.2. Genetik Algoritmanın Gelişimi	4
2.2.1. Genetik Algoritmanın Üstünlükleri	5
2.3. Genetik Algoritma Tekniği	7
2.3.1. Kodlama	9
2.3.2. Seçim	10
2.3.2.1. Amaç Fonksiyonu	10
2.3.2.2. Uygunluk Değeri	10
2.3.2.3. Sıra Tabanlı Uygunluk Atama Yöntemi	11
2.3.2.4. Seçim Yöntemleri	11
2.3.3. Çaprazlama	13
2.3.3.1. Tek Noktalı Çaprazlama	13
2.3.3.2. Çok Noktalı Çaprazlama	14
2.3.3.3. Gerçel Sayı Çaprazlaması	15
2.3.4. Mutasyon	16
2.3.4.1. İkili Kodlamada Mutasyon	17
2.3.4.2. Gerçel Kodlamada Mutasyon	17
2.3.5. Çaprazlama Ve Mutasyon Olasılığı	18
2.3.6. Elitizm	19
2.3.7. Üreme	19
2.4. Sonuç	21
<b>3. EVİRİCİLERİN HARMONİK ELİMİNASYONU</b>	<b>22</b>
3.1. Giriş	22
3.2. Eviriciler	23
3.3. Evirici Devreleri	24
3.3.1. Yarım Köprü Evirici	25
3.3.2. Tam Köprü Evirici	26
3.4. Eviricilerin Tetiklenmesi	28
3.4.1. Kare Dalga Evirici	28

3.4.1.1. Devre Analizi	31
3.4.1.2. Fourier Serisi Analizi	33
3.4.1.3. Toplam Harmonik Dağılım(THD)	34
3.4.1.4. Harmoniklerin Elenmesi	36
3.4.1.5. Genlik ve Harmonik Kontrolü	36
3.4.2. Darbe Genişlik Modülasyonlu (PWM) Eviriciler	40
3.4.2.1. PWM Eviriciler İçin Fourier Analizi	44
3.5. Sonuç	47
<b>4. OPTİMUM AÇI PROBLEMİNİN GENETİK ALGORİTMALARA UYGULANMASI</b>	<b>48</b>
4.1. Giriş	48
4.2. Program Aşamaları	49
4.2.1. Kodlama	52
4.2.2. Başlangıç Nüfusu	53
4.2.3. Amaç Fonksiyonu	54
4.2.4. Uygunluk Değeri	57
4.2.5. Seçim	57
4.2.6. Çaprazlama	58
4.2.7. Mutasyon	59
4.2.8. Elitizm	60
4.2.9. Yeni Nüfusun Oluşturulması	60
4.3. Program Sonuçları	60
<b>4. SONUÇLAR VE TARTIŞMA</b>	<b>63</b>
<b>KAYNAKLAR</b>	<b>68</b>
<b>EKLER</b>	<b>71</b>
<b>ÖZGEÇMİŞ</b>	<b>77</b>

## KISALTMALAR

<b>AC</b>	: Alternatif Akım (Alternative Current)
<b>BJT</b>	: Bipolar Junction Transistor
<b>DNA</b>	: Deoxyribonucleic Acid
<b>DC</b>	: Doğru Akım (Direct Current)
<b>GTO</b>	: Gate Turn On Thyristor
<b>IGBT</b>	: Insulated Gate Bipolar Transistor
<b>MATLAB</b>	: Matrix Laboratory Program
<b>MOSFET</b>	: Metal Oxide Semiconductor Field Effect Transistor
<b>ORCAD</b>	: Computer Aided Design Simulation Program
<b>PF</b>	: Güç Faktörü (Power Factor)
<b>PWM</b>	: Darbe Genişlik Modülasyonu (Pulse Width Modulation)
<b>THD</b>	: Toplam Harmonik Dağılım (Total Harmonic Distortion)

## TABLO LİSTESİ

	<u>Sayfa No</u>
<b>Tablo 2.1</b> İkili kodlama .....	9
<b>Tablo 2.2</b> Gerçel sayı kodlaması .....	10
<b>Tablo 2.3</b> Sıra tabanlı uygunluk ataması .....	11
<b>Tablo 2.4</b> Rulet tekerleği seçilme olasılıkları.....	12
<b>Tablo 2.5</b> İkili Kodlu birey için mutasyon işlemi.....	17
<b>Tablo 2.6</b> Gerçel kodlu birey için mutasyon işlemi.....	18
<b>Tablo 4.1</b> 3 aç ve 3 harmonik için genetik algoritma ile bulunan değerler....	61
<b>Tablo 4.2</b> 5 aç ve 5 harmonik probleminin genetik algoritma ile çözümü.....	62

## ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Hesaba dayalı yöntemler için kolay, tek tepeli bir fonksiyon.....	6
Şekil 2.2 : Hangi tepe ikilemini gösteren, çok tepeli fonksiyon.....	7
Şekil 2.3 : Standart genetik algoritmaların akış diyagramı.....	8
Şekil 2.4 : Rulet tekerleği.....	12
Şekil 2.5 : Tek noktalı ikili kod çaprazlaması.....	14
Şekil 2.6 : Çok noktalı çaprazlama.....	15
Şekil 2.7 : Gerçel sayı çaprazlaması.....	16
Şekil 2.8 : Elitizm uygulaması.....	19
Şekil 3.1 : DC-AC evirici.....	23
Şekil 3.2 : Yarım köprü evirici devre şeması.....	25
Şekil 3.3 : Yarım köprü eviricinin Orcad uygulaması.....	26
Şekil 3.4 : Yarım köprü eviricinin Orcad simülasyonu.....	26
Şekil 3.5 : Tam köprü eviricinin devre şeması.....	27
Şekil 3.6 : Anahtarlama sırasında oluşan çıkış gerilimi ve iletimdeki anahtarlar.....	27
Şekil 3.7 : Tam köprü eviricinin yarı iletken elemanlarla devre şeması.....	28
Şekil 3.8 : Kare dalga tam köprü eviricinin Orcad simülasyonu.....	29
Şekil 3.9 : Omik yüklü tam köprü kare dalga eviricinin simülasyonu .....	30
Şekil 3.10 : Endüktif yüklü tam köprü kare dalga eviricinin simülasyonunda çıkış gerilimi ve akımı.....	30
Şekil 3.11 : Farklı frekanslardaki harmonikler.....	35
Şekil 3.12 : Darbe genişliği kontrolü, genlik kontrolü.....	37
Şekil 3.13 : Genlik ve harmonik kontrolü ile evirici çıkış gerilimi.....	37
Şekil 3.14 : Anahtarlama sırası.....	38
Şekil 3.15 : Orcad simülasyon sonuçları.....	39
Şekil 3.16 : İki kutuplu anahtarlamanın elde edilişi.....	41
Şekil 3.17 : Tek kutuplu anahtarlamanın elde edilişi.....	42
Şekil 3.18 : Anahtarlama frekansı 1kHz iken Orcad simülasyonu.....	43
Şekil 3.19 : Anahtarlama frekansı 2kHz iken Orcad simülasyonu.....	43
Şekil 3.20 : İki kutuplu anahtarlama açma noktaları.....	44
Şekil 3.21 : Tek kutuplu anahtarlama açma noktaları .....	45
Şekil 3.21 : Akış diyagramı.....	51



## SEMBOL LİSTESİ

$a_0$	: Fourier serisi DC bileşen parametresi
$a_n$	: Fourier serisi Sinüs bileşen katsayı parametresi
$b_n$	: Fourier serisi Cosinüs bileşen katsayı parametresi
$D$	: Diyot
$k$	: Çaprazlama Noktası
$L$	: Endüktans
$N$	: Nüfus içerisindeki birey sayısı
$P_i$	: Birey $i$ ' nin sıra numarası
$Q_n$	: Rölatif açısal fark
$r$	: Rastgele Sayı
$R_1, R_2$	: Ebeveyn parametreleri
$R$	: Direnç
$S$	: Statik Anahtar
$S_i$	: Birey $i$ ' nin seçilme olasılığı
$T$	: Periyot
$THD_v$	: Gerilimin toplam harmonik dağılım değeri
$U_i$	: Birey $i$ ' nin uygunluk değeri
$U_{max}$	: Bireylerin alabileceği maksimum uygunluk değeri
$V_n$	: Harmonik $n$ ' in maksimum değeri
$V_0$	: Doğru Bileşen
$V_1$	: Temel bileşenin maksimum değeri
$V_{eff}$	: Gerilimin etkin değeri
$V_{sin}$	: Sinüzoidal gerilim
$V_{üçgen}$	: Üçgen dalga
$\alpha$	: Anahtarlama açısı

# **EVİRİCİLERİN HARMONİK ELİMİNASYONUNDA OPTİMUM ANAHTARLAMA AÇILARININ GENETİK ALGORİTMALAR İLE ELDE EDİLMESİ**

## **ÖZET**

Anahtarlama güç kaynakları birçok elektronik sistemde kullanılmaktadır. Genel olarak çalışma prensipleri aynıdır şebeken yüksek frekansta çalışan elektronik bir anahtar ile enerji paketini alır ve çıkışa yansıtır. Anahtarlama elemanının kullanım frekansına ve oranına bağlı olarak çıkış gerilimi değiştirilir ve regülasyon sağlanır. Anahtarlama elemanın eviricilerde kullanılması durumunda sinüs olmayan harmonik bileşenler oluşmaktadır. Harmonikleri bastırmak için aktif veya pasif filtreler kullanılabilir. Ancak harmonikleri bastırmaktan ziyade harmoniklerin üretilmemesi daha anlamlıdır. Anahtarlama elemanın iletimde ve kesimde kalma süreleri değiştirilerek harmonik dağılım miktarı azaltılabilir. Uygun anahtarlama açısı tespiti optimizasyon algoritması kullanılarak belirlenebilir. Genetik algoritmalar; çok sağlam temellere dayanan ve insanlık tarihi boyunca kusursuz işleyen evrim teorisinden esinlenilerek oluştuğundan dolayı güçlü bir optimizasyon algoritmasıdır. Evirici çıkışındaki tam sinüs olmayan işarete meydana gelen harmonikleri bastırabilmek için, işaretin Fourier serisi açılımından elde edilen denklemlerdeki anahtarlama açılarının yerlerinin bir arama ve optimizasyon yöntemi olan Genetik Algoritmalar mantığı ile yazılan Matlab programı sayesinde bulunması amaçlanmıştır.

Öncelikle genetik algoritmaların tarihsel gelişiminden, çalışma mantığından ve çalışması sırasında kullanılan yöntemlerden bahsedilmiş bu yöntemlerin içinden hangisinin niçin seçildiği hakkında bilgi verilmiştir. Tezin üçüncü bölümünde ise eviricilerin çalışma mantığı, çeşitleri hakkında bilgi verilmiştir. Evirici çeşitlerinin matematiksel analizleri yapılmıştır. Yapılan bu matematiksel analizler sonucunda evirici çıkışında etkili olan anahtarlama açılarının optimizasyon bağıntısı elde

edilmiştir. Optimum anahtarlama açıları, birinci harmonik genliğini en büyük, bastırılacak harmonik genliklerini de en küçük yapan açılardır. Tezin dördüncü bölümünde elde edilen optimizasyon bağıntısını çözmek için Matlab’de genetik algoritma programının yazım aşamaları anlatılmıştır. Bu aşamadan sonra öncelikle üç anahtarlama açısı ile üç harmonik bileşenin eliminasyonu için optimizasyon bağıntısındaki açı değerleri bulunmuş ve klasik optimizasyon yöntemi ile bulunan sonuçlarla karşılaştırılmıştır. Bir sonraki aşamada ise beş anahtarlama açısı ve beş harmonik bileşenin eliminasyonu için açı değerleri bulunmuştur böylece Matlab programında yazılan optimizasyon programın çalıştığı görülmüş ve daha yüksek miktardaki harmonik değerleri için programın çözüm bulacağı ön görülmüştür.

Sonuçta elde edilen verilerin doğrulaması yapılmıştır ve klasik yöntemlerle elde edilemeyecek çözüm kümelerinin genetik algoritmalar ile rahatlıkla çözülebileceği tespit edilmiştir ve yöntemin başarısı belirtilmiştir.

## **DETERMINING OPTIMUM SWITCHING ANGLE AT HARMONIC DISTORTION OF INVERTERS BY GENETIC ALGORITHMS**

### **SUMMARY**

A lot of electronic system could use switched mode power supply. Generally operating principles are same, takes energy pocket from main by electronic switch and gives it to output at high frequency. Output voltage could change and be regulated owing to switching component operating frequency and duty cycle value. At the usage of switching component in inverters produces non sinusoidal harmonics component. Active passive filters could be used in order to eliminate harmonics. At first stage producing pure signal which has no harmonics is more valuable than eliminating harmonics at end of the stage in electronic system. Harmonic distortion value could be reduced by changing turn on and turn off time value of switching component. Suitable turn on and turn off time values could be determined by optimization algorithm. Genetic algorithms are powerful owing to construction on strong basics and inspired from evolution theorem which is working perfectly through human history. Determining optimum switching angle in equation which is obtained from fourier series extension at Matlab program by genetic algorithms in order to eliminate harmonics on non sinusoidal signal that is occurred at output of inverter is purposed.

First of all information about; historical development stages of genetic algorithms, operating principles, used methods , what types of methods are chosen and why are they chosen ,are given. At the third stage of thesis information about inverters operating principles and kind of inverters are given. Mathematical analysis of inverters is done. From this mathematical analysis, switching angle's optimization equation which is effective at output stages of inverters is determined. Optimum switching angles make primary harmonic amplitude to be biggest and also harmonic amplitude which will be eliminated to be smallest. At the fourth stage of thesis

explanation for genetic algorithm program development stage in Matlab in order to solve optimization equation. From this stage first of all three harmonic component and three switching angle equation is solved, results are compared by other result which is find by conventional optimization method. At the next stage angles for five switching angle and five harmonic components are finding. By the results confirmation on operation of Matlab program is done and foreseen getting result on bigger harmonic level and lot's of switching angle.

In conclusion, confirmation on all results is done and solving the optimization domain which is could not be get by conventional methods, by genetic algorithm easily is determined and success of genetic algorithm method becomes visible.

## 1. GİRİŞ

Güç elektroniği devrelerinin en büyük problemi olan harmonikleri önlemek bütün tasarımcıların en büyük problemidir. Eviriciler içinde geçerli olan harmonik problemi; çıkış geriliminin kare dalga formunda olmasından kaynaklanır. Çıkış geriliminin fourier serisine açılımı ile elde edilen denklemler bize hangi harmoniklerin hangi miktarda olacağını söylemektedir. Harmoniklerin elenmesi için filtre kullanılmadan önce; evirici devresindeki anahtarlama elemanlarının anahtarlama açıların en uygun olanlarının bulunması tercih edilir. Optimum anahtarlama açıları, birinci harmoniğin genliğini en büyük, bastırılacak harmoniklerin genliklerini de en büyük yapan açı değerleridir.

Bilgisayar teknolojisindeki gelişmeler sayesinde; optimizasyon problemleri bilgisayar ortamında çözümlenmektedir. Genel olarak evrimsel programlama adıyla adlandırılan yöntemlerin en önemli yere sahip olanı genetik programlamadır.

Yaşayan sistemlerdeki bilginin temel birimi olan genin yaşamımızdaki etkisi büyüktür. Genler fiziksel özelliklerimize, davranışlarımıza, kişiliğimize, sağlığımıza ve uzun ömürlülüğümüze hükmederler. Başlangıçta genler hakkında çok fazla bilgiye sahip değilken; bugün genlerin çoğunun hangi özelliklere katkıda bulduklarını ve nasıl bir başkalaşım geçirdiğini söyleyebiliyoruz. Bunun nedeni eskiden sadece gözlemleyebildiğimiz, fiziksel olarak değerlendirebildiğimiz evrimi, daha doğru bir biçimde moleküler seviyede takip edebiliyoruz. Elde ettiğimiz verileri genetik algoritmalar gibi diğer çalışmalarımızda kullanabiliyoruz. Son yıllardaki araştırmalar; genlerin geliştirilebilir ve kontrol edilebilir olduğunu ortaya çıkarmıştır.

Genetik algoritma klasik optimizasyon yöntemleri ile çözümü zor ya da imkansız olan problemlerin çözümünde kullanılmaktadır. Genel anlamda genetik algoritma deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleri olmak üzere iç uygulama alanında kullanılmaktadır.

Mühendislik problemlerinde optimizasyon amaçlı kullanılması büyük fayda sağlamıştır. Çünkü mühendislik problemlerinin çözümü genelde çok geniş bir çözüm kümesinin taranmasını gerektirmekte ve karmaşık denklemleri içermektedir. Optimum açılı problemi gibi mühendislik problemlerini çözmek için kullanılan klasik optimizasyon yöntemlerinden biri olan Lagrange yöntemi olasılığın çok olması ve denklemlerin karmaşıklığı nedeniyle güçlüklerle karşılaşmaktadır. Klasik optimizasyon yöntemlerinin sonuca ulaşabildiği durumlar ise çözüm kümesinin taranmasının zaman almasından dolayı uzun sürmektedir. Ancak genetik algoritma ile daha kısa bir sürede kabul edilebilir sonuçlar elde edilmektedir.

## **2. GENETİK ALGORİTMALAR**

### **2.1 Giriş**

Bu bölümde, genetik algoritmalar, gelişimleri ve diğer optimizasyon yöntemlerine göre farkları anlatılacaktır. Genetik algoritmalar bir arama ve optimizasyon yöntemidir.

Optimizasyon, herhangi bir problemin çözüm kümesi içindeki en iyi çözümün belirlenmesi işidir. Diğer bir deyişle; bir problemin optimizasyonu konulan tüm sınır denklemlerini sağlayan ve amaç fonksiyonunu minimize eden değişkenler kümesinin bulunmasını ifade eder. Geçmiş yıllarda yapılan optimizasyon çalışmalarında matematiksel yöntemler kullanılmasına karşın, bilgisayar teknolojisindeki gelişmeler sayesinde programlama ön plana çıkmıştır. Genetik algoritmalar, gelişmelerin en önemli örneğidir.



## 2.2 Genetik Algoritmanın Gelişimi

Genetik algoritmalar yapay zekanın gittikçe genişleyen bir kolu olan ve Darwin'in evrim teorisinden ilham alınarak geliştirilmiş olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadırlar [1]. Darwin'e göre, mevcut organizmalar arasında meydana gelen üreme, çaprazlama ve mutasyon işlemlerinin sonucunda yeni tür canlılar oluşmaktadır. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır [2]. Genetik algoritma, doğada var olan bir yarışma ortamında, ancak daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildiği, biyolojik olaylara benzetilerek geliştirilmiştir [3].

Evrimsel hesaplama ilk olarak 1960'larda Alman bilim adamı I. Rechenberg tarafından "Evrimsel Stratejileri" isimli kitabında tanıtılmıştır ve başlangıçta uçakların aerodinamik profilinin tasarımında denenmiştir. Evrimsel stratejileri fikriyle; ebeveyn ve ebeveynin başkalaşması ile ortaya çıkan yavru olmak üzere iki sonuç ortaya koymuştur. Onun fikri daha sonra başka araştırmacılarında ilgisini çekmiş ve evrimsel stratejilerinin gelişmesi sağlanmıştır. John Holland, evrimsel sürecinin, anlayamadığı çözüm yöntemlerinin bir bilgisayara öğretilebileceğini düşünmüştür. Genetik algoritma, 1970'lerde John Holland tarafından bu düşüncenin bir sonucu olarak bulunduğundan sonra, onun öğrencileri ve arkadaşları tarafından geliştirilmiştir [2]. 1975 yılında ilk olarak Holland, "Doğal ve Yapay Sistemlerin Uyumu" adlı kitapta genetik algoritmaların temel operatörleri çoğalma, çiftleşme ve mutasyonu ilk defa bir arada kullandı [3]. Michigan üniversitesindeki bu araştırmalar başlangıçta araştırma teknikleri olarak doğal seçim ve doğal genetiğin mekanizmaları üzerine kurulmuştur. Başlangıçta; iyi bilinmeyen genetik algoritma Goldberg'in 1989 yılında basılan "Araştırma, Optimizasyon ve Makine Öğretisinde Genetik Algoritmalar" kitabından sonra optimizasyon, adaptasyon ve öğrenme metodoloji alanlarında dikkate değer konuma gelmiştir [4].

Araştırmaların sonucunda genetik algoritmaların doğal seçim ve doğal kalıtım kurallarına dayandığı ortaya konmuştur. Doğal seçim çevre koşullarına uyum

sağlamış olan canlıların yaşamlarını sürdürmesi, uyum sağlamamış olanların ise elenmesi yani ölmesi anlamına gelir. Doğal kalıtım ise, seçilmiş olan bireylerin birbirleriyle çiftleşerek yeni yavrular meydana getirmesi olarak açıklanabilir.

Genetik algoritmalar, problemin çözümünü bulmak için doğadaki evrimin bilgisayar ortamındaki benzetimini kullanmaktadır. Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılmaktadır [5]. Ayrık ve zor optimizasyon problemleri için idealdir [4]. Çok genel anlamda genetik algoritmanın üç uygulama alanı bulunmaktadır. Bunlar deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleridir [2].

Genetik algoritma, çözüm ya da çözümler bulmak için taranması gereken çözüm uzayının çok büyük olduğu durumlarda, çözümü arama işlemi için en akılcı yöntemdir. Evrimin her sürecinde edinilen bilgi, sonraki kuşaklara aktarılarak taramanın daha uygun bölgelerde gezmesi sağlandığı gibi değişim işlemi yardımı ile yerel çözüm noktalarına sıkışıp kalma olasılığı da azalmaktadır [1].

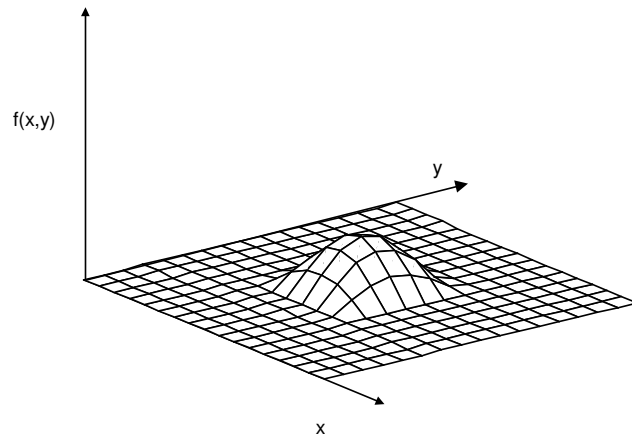
Genetik algoritma, mühendislik problemlerinde genellikle optimizasyon amaçlı olarak kullanılmaya başlanmıştır. Problemlerin çözümü çok geniş bir çözüm havzasının taranmasını gerektirdiği durumlarda, bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte, genetik algoritmayla ise kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir [2].

### **2.2.1 Genetik Algoritmanın Üstünlükleri**

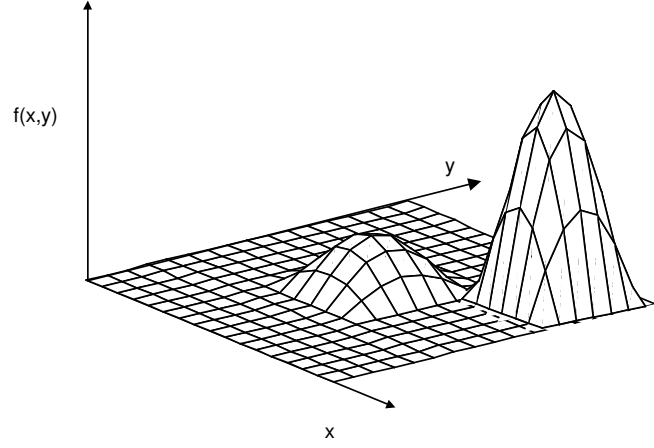
Literatürde yaygın olarak kullanılan üç tip optimizasyon yöntemi vardır: hesaba dayalı, sayıma dayalı ve rasgele yöntemler. Goldberg, rasgele yöntemlerden biri olan genetik algoritmanın, diğer optimizasyon yöntemlerinden ayrılmasını dört maddeye bağlamıştır [3].

1. Genetik algoritmalar; çözümler üzerinde değil, çözüm setlerini kodlayarak bunlar üzerinde çalışırlar. Örnek olarak, çözümler üzerinde çalışan sayıma dayalı yöntemlerde her defasında bir nokta olmak üzere arama uzayının her noktasında amaç fonksiyon değerine bakılacağı için büyük çözüm kümesine sahip fonksiyonlarda sayıma dayalı yöntemin verimsizliği ortaya çıkacaktır.

2. Genetik algoritmalar; tek bir çözümlü değil, çözüm popülasyonunu araştırırlar. Diğer optimizasyon yöntemleri çözümlü aramaya tek bir noktadan başladıkları için ilk buldukları yerel optimum noktasında aramayı bitirirler. O an bulunulan noktadan yola çıkarak, komşu noktalar arasından en iyisi bulunmaya çalışılır. Ancak, birden çok optimuma sahip fonksiyonlar için bu sonuç yeterli değildir. Genetik algoritmalar ise tüm yerel optimum noktalarını bularak bunları karşılaştırır ve fonksiyonun evrensel optimum noktasına ulaşır ( Şekil 2.1, Şekil 2.2).
3. Genetik algoritmalar; türev ya da diğer yardımcı bilgileri değil, uyumluluk fonksiyonunu kullanırlar. Gerçek dünyada yapılan birçok incelemede karşımıza sürekli olmayan, birden çok en uygun olan ve parazitlerin karıştığı fonksiyonlar çıkmaktadır. Bu nedenle bu fonksiyonların türevleri olmayabilir ve türevini alıp sifıra eşitleme esasına dayanan hesap tabanlı çözüm yöntemleri bu fonksiyonlar için yetersiz kalır.
4. Genetik algoritmalar, deterministik değil rastlantısal geçiş kurallarını kullanırlar. Rastlantısal geçiş kurallarına bağlı olarak çalıştığından model değiştirilmese bile her çalıştığında farklı çözümler elde edilir. Geleneksel yöntemler ise aynı değerler ile her çalıştığında aynı çözüme ulaşılır.



**Şekil 2.1:** Hesaba Dayalı Yöntemler için Kolay, Tek Tepeli Bir Fonksiyon



**Şekil 2.2:** Hangi Tepe İkilemini Gösteren, Çok Tepeli Fonksiyon

Ayrıca genetik algoritma operatörleri seçim, çaprazlama ve mutasyon ile diğer optimizasyon yöntemlerinden ayrılır. Topluluktaki daha uygun bireylerin seçilerek daha sonraki nesillere aktarılması ve uygun olmayanları elenmesi en iyi çözümü bulmak açısından genetik algoritmanın üstünlüğüdür. Genetik algoritmanın bu üstünlüklerinin yanında iki önemli zayıflığı bulunmaktadır: büyük çözüm kümeli problemlerde her bir çözüm aynı anda değerlendirileceğinden çözüme ulaşmanın zaman alması ve uygun olmayan çözüme yakınsama gerçekleşmesi olasılığı bulunmaktadır.

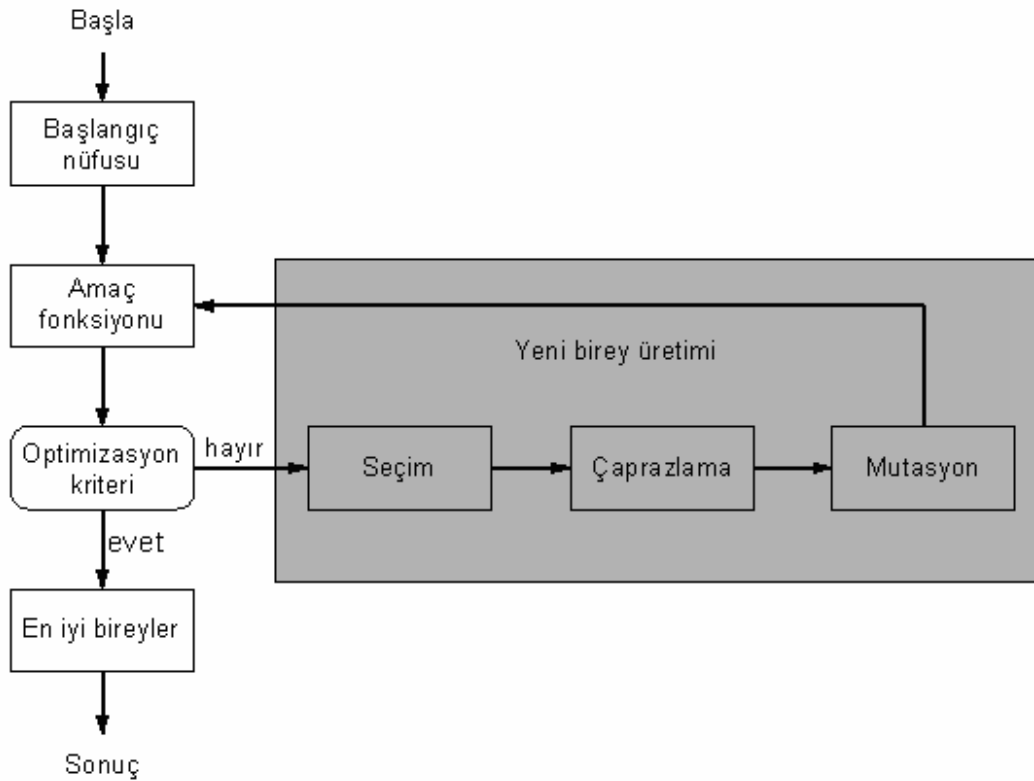
### 2.3 Genetik Algoritma Tekniği

Genetik algoritmalar ile problem çözümünde ilk adım, rasgele üretilen, büyüklüğü veya ne kadar birey içereceği probleme bağlı olan ve çözüm kümesini taramak için oluşturulan başlangıç nüfusunun belirlenmesidir. Nüfus denilen çözüm seti içerisinde kromozom (birey) adı verilen dizilerden oluşmaktadır. Nesil içindeki her kromozom problemin muhtemel çözümünü gösterir. Kromozomlar, parametre veya değişken bilgilerini temsil eden, genetik bilginin taşınmasını sağlayan gen adı verilen ikili ya da reel sayı düzenine göre kodlanmış sembol dizilerinden oluşmaktadırlar.

Yaşamımızda genlerin etkisi büyüktür. Genler fiziksel özelliklerimize, davranışlarımıza, kişiliğimize, sağlığımıza ve uzun ömürlülüğümüze hükmederler. Son yıllardaki araştırmalar genlerin geliştirilebilir ve kontrol edilebilir olduğunu ortaya çıkarmıştır. Genetik algoritmalarda da; yaşayan sistemlerde olduğu gibi

bilginin temel birimi gendir. Genelde, tek bir karakter ya da fiziksel özelliği etkileyen ya da bunlara karar veren kromozomların oranına gen denir. Genetik biliminde, kromozomlar DNA kısımlarından oluşur [5].

İkinci adımda, başlangıç nüfusunun bütün bireyleri “amaç fonksiyonu”nda (objective function) yerine konularak bireylerin “uygunluk değerleri” (fitness value) yani bir sonraki nesilde yaşama olasılıkları belirlenir. Çünkü uygunluk değerleri, nüfus içerisindeki bireylerin ne oranda çözüm sağladıklarını gösteren sayısal değerlerdir.



**Şekil 2.3:** Standart Genetik Algoritmaların Akış Diyagramı

Üçüncü adımda, nüfus içerisindeki bireylerin optimizasyon kriterini sağlayıp sağlamadıkları kontrol edilir. Optimizasyon kriteri, bireylerin amaç fonksiyon ve uygunluk değerleri için bir eşik değeri olabileceği gibi, algoritmayı belli bir nesil sayısında tutmak ya da topluluktaki en iyi bireyin belirli bir nesil sayısı boyunca değişmemesi olabilmektedir. Optimizasyon ölçütü sağlanmıyorsa, yeni nüfus üretimine geçilir. Yeni (çocuk) bireylerin üretiminde kullanılacak olan mevcut (ebeveyn) bireylerin seçiminde, uygunluk değerleri esas alınır. Çünkü uyumlu olanların daha iyi sonuçlar elde etmesi daha yüksek bir olasılıktır ve bu nesilden alınan sonuçlarla bir öncekinden daha iyi olacağı beklenen yeni bir nesil oluşturmak

amaçlanmıştır. Önceden belirtilen oranda ebeveyn çaprazlamaya girerek birbirleriyle gen alış verişinde bulunurlar ve ortaya bir sonraki toplumun bireylerin oluşturacak olan “çocuklar” çıkar. Yine önceden tanımlanmış mutasyon oranları nedeniyle ortaya çıkan çocuklarda başkalaşımalar meydana gelir. Çocuk bireylerin uygunluk değerleri hesaplanır ve ebeveyn bireylerle beraber yeni nüfus içerisine katılır. Yeni nüfus üretim döngüsü optimizasyon ölçütü sağlanıncaya kadar sürdürülür. Optimizasyon ölçütü sağlanıyor ise, nüfus içerisindeki uygunluk değeri en iyi olan birey problemin çözümünü oluşturmaktadır. Bu işlem problem çözümünün son adımıdır [2–4, 6].

### 2.3.1 Kodlama

Kodlama, parametre veya değişkenlerin bir semboller dizisi halinde art arda dizilerek bilgi dizisi veya birey oluşturma işidir. Genetik algoritmalarla problem çözümünde, nüfus oluşturulurken ilk yapılacak iş bireylerin kodlama biçiminin belirlenmesidir. Çünkü genetik algoritmalar değişken veya parametrelerin değerlerini değil kodlarını kullanarak çözüm ararlar. Kodlama doğrudan problemin içeriğine dayalıdır ve çözüm ile ilgili bilgi içerir. Kodlama için ikili ya da gerçel sayı kodlamaları en çok tercih edilen kodlama çeşitleridir [7].

İlk genetik algoritma çalışmalarında ikili kod kullanılmıştır. İkili kodlamada bireyler 0 ve 1 bitlerinin art arda gelerek oluşturduğu diziler olduğu için, basittirler ve bilgisayar tarafından daha kolay ve hızlı bir biçimde işlenebilirler. İkili kodlamanın uygun olmadığı durumlarda gerçel sayı kodlaması kullanılır. Gerçel sayı kodlamasında ise; gerçel sayılar art arda yazılarak bireyler oluşturulur. Gerçel sayı kodlamasının ikili kodlamaya göre avantajı, kodlama ve kod çözme işlemleri gerekmemesidir [3].

İkili kodlamada, her bir parametre veya değişken için N adet bit kullanılmaktadır. N sayısı her parametre için farklı olabilmektedir. Bireyler için, tabloda ikili kodlamaya ve gerçel sayı kodlamasına göre örnekler verilmiştir.

**Tablo 2.1:** İkili Kodlama

Birey 1	1 0 1 0 0 1 1 0 1 1 0 0
Birey 2	0 1 0 0 0 0 1 1 0 1 1 1

**Tablo 2.2:** Gerçel Sayı Kodlaması

Birey 1	0,483	0,322	-5,261	2,333	6,986	5,284
Birey 2	-1,258	0,111	0,007	5,269	7,568	-4,88

### 2.3.2 Seçim

Seçim, çaprazlama için, mevcut nüfus içerisindeki bireylerden hangilerinin ebeveyn olarak kullanılacağı belirlenmesidir. İyi çocuk bireyler üretmek için, uygun seçme yöntemleri kullanarak iyi bireyler seçmek gerekmektedir. Teoriye göre iyi olan bireyler yaşamını sürdürmeli ve bu bireylerden yeni bireyler oluşmalıdır. Seçme ölçütleri; uyumluluğu esas olarak birbiriyle uyumlu olan bireyleri seçer. Daha sonra çaprazlama ve mutasyon uygulanacak olan uyumluluğu yüksek olan bireylerden daha uyumlu yeni bireylerin ortaya çıkması daha yüksek olasılıktır. Bireylerin tamamı uyumluluğa göre seçilebilir veya bir kısmı rasgele seçilerek yeni kuşağa aktarılabilir. Seçim işlemi için gerekli olan bazı kavramlar aşağıda açıklanmıştır [8].

#### 2.3.2.1 Amaç Fonksiyonu

Amaç fonksiyonu, nüfus içerisindeki bireylerin genetik gelişimini sağlayan mekanizmanın temelini oluşturmaktadır. Genetik algoritma ile çözümü aranan problem arasındaki tek bağlantıdır. Birey bir giriş olarak alınır ve problemi tanımlayan fonksiyondan elde edilen çözümler amaç fonksiyonun sonuçlarına karşılık bir sayı üretir.

#### 2.3.2.2 Uygunluk Değeri

Genetik algoritmaların farklı problemlere uygulanmasıyla oluşan amaç fonksiyon değeri farklılıklarını ortadan kaldırmak için, amaç fonksiyonlar uygunluk değerlerine dönüştürülürler. Bireylere uygunluk değeri atanmasında, kendisinin ve nüfus içerisindeki diğer bireylerin amaç fonksiyon verileri kullanılır. Genetik algoritma ile problem çözümlerinde, yaygın olarak kullanılan ve tezdeki uygulamada da uygulanan uygunluk değeri atama yöntemi sıra tabanlı uygunluk atama yöntemidir.

### 2.3.2.3 Sıra Tabanlı Uygunluk Atama Yöntemi

Sıra tabanlı uygunluk atamasında, nüfus içerisindeki bireyler amaç fonksiyon değerlerine göre sıraya konulmaktadır. Her bir bireye, amaç fonksiyonu değerine bakılmaksızın, sıra numarasına göre uygunluk değeri atanmaktadır [9]. Sıra tabanlı uygunluk atamasında, bireylerin uygunluk değeri denklem 2.1 ile hesaplanmaktadır:

$$U_i = U_{\max} (N - P_i) / (N - 1) \quad (2.1)$$

Burada;  $U_i$  birey  $i$ 'nin uygunluk değerini,  $U_{\max}$  bireylerin alabileceği maksimum uygunluk değerini,  $N$  nüfus içerisindeki birey sayısını,  $P_i$  birey  $i$ 'nin sıra numarasını göstermektedir.

Sıra tabanlı uygunluk atamasında, sıralama işlemi yapılırken, en uygun bireyin sıra numarası 1'e, en kötü bireyin sıra numarası  $N$ 'e eşitlenir. Amaç fonksiyon değerleri sıraya konularak ve  $U_{\max} = 2$  alınarak, tabloda sıra tabanlı uygunluk atamasına bir örnek verilmektedir.

**Tablo 2.3:** Sıra Tabanlı Uygunluk Ataması

Birey sıra numarası ( $P_i$ )	1	2	3	4	5	6	7	8	9	10	11
Amaç fonk. değeri ( $f_i$ )	1	3	4	7	8	9	10	15	20	30	95
Uygunluk değeri ( $U_i$ )	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,6	0,4	0,2	0

### 2.3.2.4 Seçim Yöntemleri

Topluluk içindeki bireylerin seçilme yöntemlerinden bazıları aşağıda açıklanmıştır.

- ❖ Rulet çarkı seçimi: Rulet çarkı seçimi en yaygın kullanılan seçin yöntemidir. Bu yöntemde ebeveynlerin uygunluk (fitness) değerlerinin toplamı rasgele oluşturulan bir sayı dizisi ile çarpılır. Sırayla ebeveynlerin uygunluk değerleri toplanmaya başlanır. Oluşturulan dizideki elemandan büyük bir sayıya toplamda ulaşıldığında, son eklenen uygunluk değerinin ait olduğu ebeveyn seçilmiş olur. Diğer taraftan her ebeveynin uygunluk değeri, uygunluk değerlerinin toplamına



bölünerek [0,1] aralığında sayılar elde edilir. Bu sayılar sırayla birbirine eklenerek rasgele bir sayıya kadar ilerlenir. Rasgele sayıdan daha büyük bir sayı elde edildiğinde son eklenen sayının ait olduğu ebeveyn seçilir [1].

Seçilme olasılıkları denklem 2.2 ile belirlenmektedir.

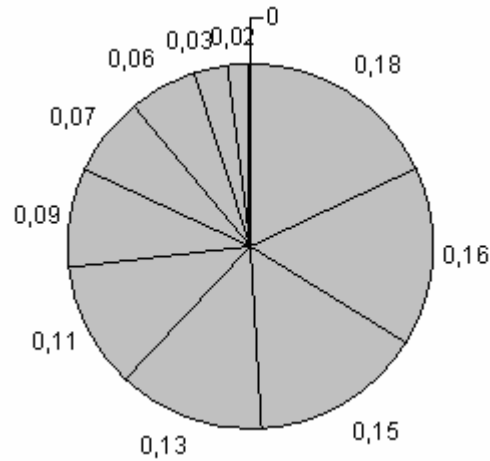
$$S_i = \frac{U_i}{\sum U} \quad (2.2)$$

Burada;  $S_i$  birey  $i$ 'nin seçilme olasılığını,  $U_i$  birey  $i$ 'nin uygunluk değerini,  $\sum U$  nüfus içerisindeki bireylerin toplam uygunluk değerlerini göstermektedir.

Tabloda, 11 bireyden oluşan, maksimum uygunluk değeri 2 olan ve uygunluk değerleri sıra tabanlı uygunluk atamasıyla belirlenmiş olan bir nüfus görülmektedir.

**Tablo 2.4:** Rulet Tekerleği Seçilme Olasılıkları

Birey sıra	1	2	3	4	5	6	7	8	9	10	11
Uygunluk değeri	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,6	0,4	0,2	0,0
Seçilme	0,18	0,16	0,15	0,13	0,11	0,09	0,07	0,06	0,03	0,02	0,0



**Şekil 2.4:** Rulet Tekerleği

- ❖ Sıra seçimi: Uygunluk değerleri birbirinden çok farklı olan durumlar rulet çarkı seçimi için problem yaratır. Örneğin, en iyi bireyin uygunluk değeri %90 ise çark döndürüldüğünde diğer bireyler çok az oranda seçilecek ve algoritmanın kısır döngüye girme ihtimali olacaktır. Sıra seçimi öncelikle topluluğu sıralar ve diğer bireylerin uygunluk değerlerini bu sıralamaya göre düzenler. Sıra seçiminde tüm bireylerin seçilme olasılığı vardır fakat en iyi birey diğer bireylerden çok fazla farklılık göstermediğinden bu yöntem çok yavaş bir çözümdür.
- ❖ Kararlı hal değişimi: Bu yöntem ebeveyn seçimi için uygun değildir. Kararlı hal seçiminde uygunluk derecesi yüksek olan bireyler belirlenerek yavru oluşturmak üzere çaprazlamaya girerken, uygunluk derecesi düşük olan bireyler atılarak erlerine yeni oluşan yavrular konulur.
- ❖ Turnuva seçimi: Bu yöntemde, bir grup birey rastlantısal olarak seçilir ve bu grup içerisindeki en iyi birey çiftleşmeye gönderilir. Gruptaki bireylerin sayısı, turnuva büyüklüğü olarak adlandırılır. Turnuva seçimi hızlı olması nedeni ile tercih edilir [1].

### 2.3.3 Çaprazlama

Kromozomların nasıl temsil edileceğine karar verildikten sonra çaprazlama yapılabilir. Evrim teorisinde, mayoz bölünmenin bir aşamasında kromozomlar arasında kırılmalar ve yeniden birleşmeler yoluyla genetik bilginin değiş tokuşu sağlanır. Genetik algoritmada da aynı şekilde; çaprazlama iki ebeveynin rasgele seçilmiş bir konumdan sonraki genlerinin yeni bireyler oluşturmak için değiş tokuş edilmesidir. Amaç seçimde olduğu gibi ebeveynlerdeki iyi genlerin çocuk bireylerde birleştirilerek daha iyi bireyler elde etmektir.

#### 2.3.3.1 Tek Noktalı Çaprazlama

Tek noktalı çaprazlama, ikili kodlanmış ebeveyn bireylerin çaprazlanmasında kullanılmaktadır. Ebeveyn bireyler çaprazlanırken, öncelikle bir çaprazlama noktası belirlenir. Çaprazlama noktası  $k$ ,  $[1,2,\dots,N-1]$  aralığında rasgele üretilen bir tamsayıdır. Burada  $N$ , ebeveyni temsil eden ikili koddaki bit sayısıdır [10]. Çaprazlama noktasının belirlenmesinden sonra, karşılıklı olarak ebeveynlerin bilgileri yer değiştirilir. Tek noktalı çaprazlama için aşağıda bir örnek verilmektedir.

Ebeveyn bireyler:

Ebeveyn-1            0 1 1 1 0 0 1 1 0 1 0

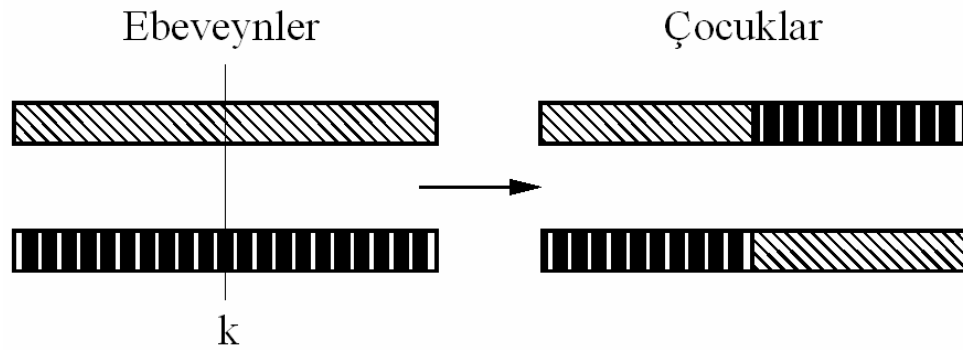
Ebeveyn-2            1 0 1 0 1 1 0 0 1 0 1

Çaprazlama noktası:        k=5

Çocuk Bireyler:

Çocuk-1                0 1 1 1 0 | 1 0 0 1 0 1

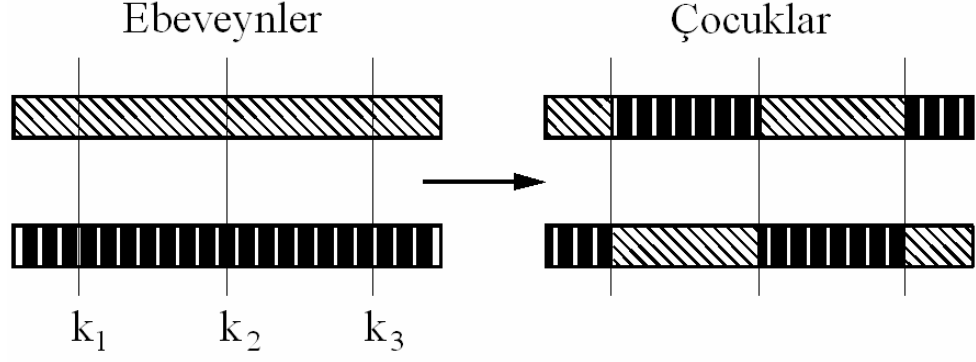
Çocuk-2                1 0 1 0 1 | 0 1 1 0 1 0



Şekil 2.5: Tek Noktalı İkili Kod Çaprazlaması

### 2.3.3.2 Çok Noktalı Çaprazlama

İkili kodlanmış ebeveynlerin çaprazlanmasında kullanılan bir diğer yöntem ise, çok noktalı çaprazlamadır. Bu yöntemde, çaprazlama noktası olarak birden fazla nokta üretilmektedir. Ancak çok noktalı çaprazlama ile ebeveynlerden oldukça farklı yavrular meydana gelir ve başlangıçta uygun olan bireylere yakınsama olasılığı azalır. Şekilde, üç noktalı çaprazlama görülmektedir [10]. Çok noktalı çaprazlama için aşağıda bir örnek verilmektedir.



**Şekil 2.6:** Çok Noktalı Çaprazlama

Ebeveyn bireyler:

Ebeveyn-1            0 1 1 1 0 0 1 1 0 1 0

Ebeveyn-2            1 0 1 0 1 1 0 0 1 0 1

Çaprazlama noktaları:        2   6   10

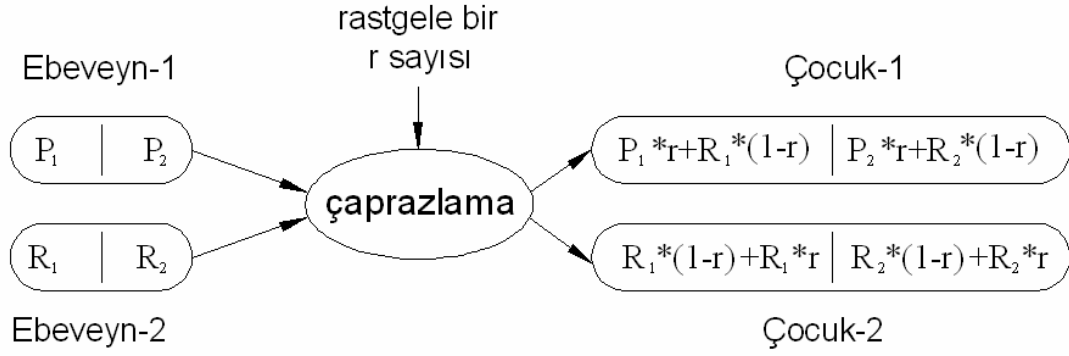
Çocuk Bireyler:

Çocuk-1                0 1 1 1 0 1 1 1 1 0 1 1

Çocuk-2                1 0 1 1 1 0 0 1 0 0 1 0 0

### 2.3.3.3 Gerçel Sayı Çaprazlaması

Gerçel sayı çaprazlamasında ebeveyn bireyler parametrelerine ayrılırlar ve her bir parametre etiketlenir. Örnek olarak, aşağıdaki gibi iki ebeveyn alınabilir ve bu iki ebeveynin parametreleri  $P_1$ ,  $P_2$ ,  $R_1$  ve  $R_2$  olarak etiketlenir. Örnekteki  $r$  sayısı 0 ile 1 aralığında rasgele üretilmektedir. Çaprazlamayla üretilen çocuk bireylerin parametre değerleri, etiketlenmiş ebeveyn parametre değerleri ile  $r$  rasgele sayısının şekilde görülen formüllerde kullanılmasıyla elde edilmektedir.



**Şekil 2.7:** Gerçel Sayı Çaprazlaması

Ebeveyn bireyler:

Ebeveyn-1	5,3	4,1
Ebeveyn-2	2,3	1,5
r rasgele sayısı		0,77

Çocuk Bireyler:

Çocuk-1	4,61	3,5
Çocuk-2	2,99	2,09

### 2.3.4 Mutasyon

Çaprazlama gerçekleştiikten sonra mutasyon gerçekleştirilir. Mutasyon oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek, sonuca daha hızlı ulaşmak amacıyla nüfusu oluşturan bireylerin çeşitliliğini arttırmak için ve nüfus içerisindeki bireylerin çoğunun yerel minimuma takılmasını engellemek için yapılır [11]. Evrim biliminde bir organizmanın gen veya kromozomlarında fiziksel veya kimyasal kökenli etmenlerin neden olduğu ani yapısal veya sayısal değişikliklerdir. Genetik algoritmada, mutasyon oluşan yeni bireylerin değişkenlerinin mutasyon oranı adı verilen düşük bir olasılıkla değişime uğratılmasıdır. Değişim biçimi, bireylerin kodlama biçimine bağlıdır. Gerçel kodlamada, bireyin mutasyona uğratılacak parametrelerine çok küçük bir sayı olan mutasyon olasılıkları eklenir veya çıkartılır. İkili kodlamada ise, bireyi oluşturan bitlerin tek tek mutasyon olasılıklarına göre tersi alınarak mutasyona uğratılır. Kromozomdaki gen sayısı değişmez, sabit kalır.

Mutasyon oranı deęişken sayısı ile ters orantılıdır. Deęişken sayısı arttıkça mutasyon oranı azalmaktadır. Mutasyon oranını belirleyen kesin bir koşul veya kriter yoktur. Farklı çalışmalarda, mutasyon oranı olarak 0.1 ile 0.001 arasında deęerler kullanılmıştır. Standart genetik algoritmalarda, mutasyon ikinci derecede rol oynamaktadır [3]. Mutasyonun en önemli görevi, çaprazlama sonucunda kaybedilen iyi özellikli bilgileri tekrar kazanabilmektir. Bazı araştırmacılar, mutasyon işleminin görevinin sadece çaprazlama kullanılarak gerçekleştirilebileceęi düşüncesiyle mutasyonu kullanmaya gerek olmadığını belirtmektedirler [12]. Bazı araştırmacılar ise mutasyon oranının çok yüksek, hatta çaprazlama oranından da fazla olması gerektiğini savunmaktadırlar.

#### 2.3.4.1 İkili Kodlamada Mutasyon

İkili kodlamada, mutasyona uğrıtılacak bitin tersi alınmaktadır. Bireyin hangi bitlerinin mutasyona uğrayacağı mutasyon oranı ile belirlenmektedir. Her bir bit için [0,1] aralığında rasgele bir sayı üretilir. Üretilen sayı mutasyon oranından küçük ise, tersi alınarak bit mutasyona uğrıtılır. Bu işleme tabloda bir örnek verilmektedir.

**Tablo 2.5:** İkili Kodlu Birey için Mutasyon İşlemi

Mutasyon oranı: 0,005								
Mutasyon olasılıkları	0,561	0,814	0,207	0,361	0,147	0,019	<b>0,003</b>	0,963
Mutasyondan önce	1	0	1	1	0	1	<b>1</b>	0
Mutasyondan sonra	1	0	1	1	0	1	<b>0</b>	0

#### 2.3.4.2 Gerçel Kodlamada Mutasyon

Mutasyona uğrıtılacak deęişkenin deęerine rasgele üretilen bir sayı eklenir veya çıkartılır. Bireyin hangi deęişkenin mutasyona uğrayacağı mutasyon oranı ile belirlenmektedir. Her bir deęişken için [0,1] aralığında rasgele bir sayı üretilir. Üretilen sayı mutasyon oranından küçük ise, sayı deęişkene eklenerek deęişken mutasyona uğrıtılır. Bu işleme tablo 2.6'da bir örnek verilmektedir.

**Tablo 2.6:** Gerçel Kodlu Birey için Mutasyon İşlemi

Mutasyon oranı: 0,005								
Mutasyon olasılıkları	0,561	0,814	0,207	0,361	0,147	0,019	<b>0,003</b>	0,963
Mutasyondan önce	4,384	2,756	0,547	-3,177	6,831	0,214	<b>0,351</b>	0,412
Mutasyondan sonra	4,384	2,756	0,547	-3,177	6,831	0,214	<b>0,354</b>	0,412

### 2.3.5 Çaprazlama Ve Mutasyon Olasılığı

Genetik algoritma tekniğinin çaprazlama olasılığı ve mutasyon olasılığı olmak üzere iki basit parametresi vardır. Çaprazlama olasılığı çaprazlamanın hangi sıklıkta yapılacağını belirtir. Eğer hiç çaprazlama yapılmaz ise (çaprazlama olasılığı %0) yeni bireyler eski bireylerin aynısı olur ama bu yeni kuşağın eskisiyle aynı olacağı anlamına gelmez. Eğer bu oran %100 olursa yeni bireyler tamamıyla çaprazlama ile elde edilir. Çaprazlama eski bireylerden iyi taraflar alınarak elde edilen yeni bireylerin daha iyi olması umudu ile yapılır.

Mutasyon olasılığı ise mutasyonun hangi sıklıkta yapılacağını belirtir. Mutasyon olmaz ise yeni birey çaprazlama veya kopyalama sonrasında olduğu gibi kalır. Eğer mutasyon olur ise yeni bireyin bir kısmı değiştirilmiş olur. Eğer bu oran %100 olursa kuşak içindeki bireyler tamamen değişir. %0 olursa hiç değişmeden kalır.

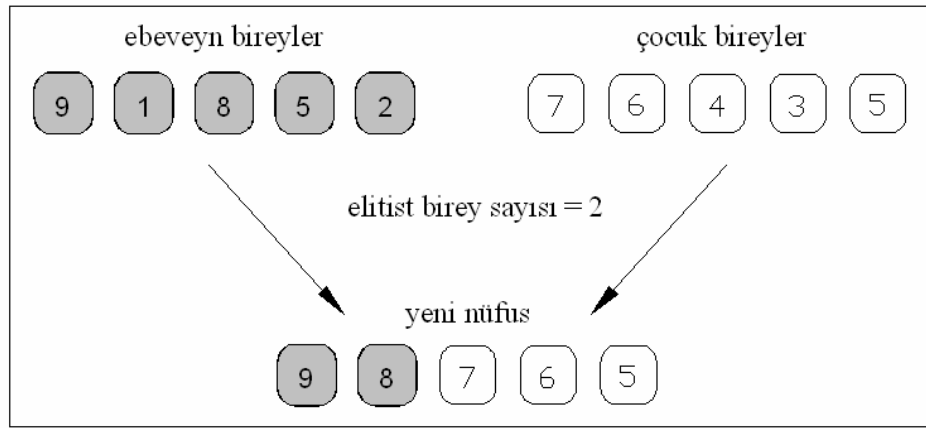
### 2.3.6 Elitizm

Seçim, çaprazlama ve mutasyon işlemleri sonrasında kuşakta bulunan en iyi uyumluluğa sahip birey sonraki kuşağa aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra oluşan yeni kuşaktaki en kötü birey bir önceki kuşağın en iyi (elit) bireyi ile değiştirilir. Elitizm sayesinde genetik algoritmanın performansı artar ve sonuca ulaşmak hızlanır. Elitizm, uygunluk değerleri en iyi olan belirli sayıdaki ebeveyn bireylerin, çaprazlama ve mutasyon işlemi ile iyi özellikleri bozulmadan bir sonraki kuşağa aktarılmalıdır.

Elitizmin uygulanmadığı standart genetik algoritmalarda, mutasyon işleminden sonra, çocuk bireyler yeni nüfusu oluşturmaktadır. Yeni nüfus oluşturulurken, çocuk

bireylerin sayısının ebeveyn bireylerin sayısına eşit olmasına dikkat edilir. Yani, ebeveynlerin tamamı nüfustan çıkartılıp, yerlerine çocuk bireyler konular. Böylece, her bir bireyin sadece bir kuşak yaşamasına izin verilmektedir. Ancak, oldukça basit olan bu yöntemde, iyi ebeveynler iyi çocuklar üretmeden nüfustan çıkartılabilmekte ve çözüm için iyi sonuç verebilecek faydalı bilgiler kaybolabilmektedir.

Elitizm uygulandığında, her bir kuşakta ebeveyn bireyler içerisinde, önceden belirlenmiş sayıda en iyi birey, en kötü çocuk bireylerle yer değiştirilerek yeni nüfusa katılmaktadır. Böylece, iyi bireylerin bir kuşak yerine, birkaç kuşak veya daha iyi bireyler üretilene kadar yaşaması sağlanmış olmaktadır. Elitizm ile bireylerin iyi özelliklerinin kaybolması engellenmektedir [13].



**Şekil 2.8:** Elitizm Uygulaması

Şekil 2.8’de, elitist birey sayısı iki olan yeni nüfus oluşturulmasına bir örnek gösterilmektedir. Örnekte, ebeveyn bireyler ile çaprazlama ve mutasyona uğratılmış çocuk bireylerin uygunluk değerleri verilmektedir. Çocuk bireylerin tümü yeni nüfusa aktarılmadan, elitizm uygulanarak, 3 ve 4 uygunluk değerlerine sahip kötü çocuklar yerine, 8 ve 9 uygunluk değerlerine sahip iyi ebeveynler yeni nüfusa katılmaktadır.

### 2.3.7 Üreme

Topluluğun yenilenmesinde, üremesinde iki temel yaklaşım bulunmaktadır. Nesilsel yaklaşımda her nesilde topluluk; seçim, çaprazlama ve mutasyon operatörleri tarafından oluşturulan yeni topluluk ile değiştirilir. Bu durumda, her nesildeki bireyler sadece kendi nesillerindeki bireylerle üreme yapabilir. Sürekli durum



yaklaşımında, oluşan toplulukta bireyler hemen yerlerini alır. Yeni topluluk bireyleri ve eski topluluk bireyleri aynı ortamda beraber bulunabilir. Genetik algoritmada genellikle nesilsel yaklaşım kullanılır. Tezde de nesilsel yaklaşım tercih edilmiştir [14].

Eski nüfusun bireylerinin seçim, çaprazlama ve mutasyon işlemlerinden geçirilmesinden sonra, oluşan yeni bireyler uygunluk değerlerine bağlı olarak yeni nüfusa eklenirler. Oluşturulan yeni bireyler nüfusa katılırken aşağıdaki yöntemlerden birisi kullanılabilir:

- ❖ Ebeveyn birey sayısı kadar çocuk birey üretilir ve çocuk bireyler ebeveyn bireylerin yerlerini alırlar. Böyle bir yöntemde, iyi ebeveynlerin iyi çocuklar üretmeden nüfustan çıkartılma riski vardır. Bu genetik evrim açısından bir dezavantaj oluşturmaktadır.
- ❖ Ebeveyn birey sayısından daha az sayıda çocuk birey üretilir. Çocuk birey sayısı kadar rasgele seçilen ebeveyn nüfustan çıkartılarak yerlerine çocuk bireyler konulur.
- ❖ Ebeveyn birey sayısından daha az sayıda çocuk birey üretilir. Elitizm uygulanarak, en kötü ebeveyn bireylerin yerlerini çocuk bireyler alırlar. İyi ebeveynler bir sonraki nüfusa aktarılmış olurlar.
- ❖ Ebeveyn birey sayısından daha fazla çocuk birey üretilir ve en iyi çocuk bireyler ebeveynlerin yerlerini alırlar.

Yeni nüfusun oluşturulmasında ana kriter, nüfus büyüklüğünün eski nüfus büyüklüğüne eşit olmasıdır. Yeni birey sayısı nüfus boyutu kadar olmak zorunda değildir, fakat nüfus boyutunun korunması esastır. Genetik evrim gereği, güçlü olan hayatta kalır mantığına dayanarak, iyi ebeveynlerin bir sonraki çözüm kümesinde yer almaları gereklidir. Bu nedenle, yukarıda da anlatıldığı üzere, yeni nesil içine bir önceki neslin elit bireyleri katılır.

## 2.4 Sonuç

Görüldüğü üzere genetik algoritmanın yapısı oldukça geneldir ve herhangi bir probleme uygulanabilir. Kromozomların tanımlanması genellikle ikili düzendeki sayılarla yapılır. Çaprazlama işlemi için kullanılan bireyler iyi bireylerden seçilir. Genetik algoritma kullanılarak bir problem çözülecekse algoritmanın ne zaman sonlanacağına kullanıcı karar vermektedir. Genetik algoritmanın belli bir sonlanma ölçütü yoktur. Sonucun yeterince iyi olması ve ya yakınsamanın sağlanması algoritmanın durması için ölçüt olarak kullanılabilir.

Bütün çözüm bulma yöntemleri gibi genetik algortmada da, her bir gen için tanımlı değer kümelerinin kartezyen çarpımından oluşan arama uzayında gezinilmektedir. Şüphesiz bu gezinmenin rasgele olmaması için çözümün ne olduğu, daha doğrusu neyin iyi neyin kötü çözüm olduğunun tamamen nicel bir tanımının yapılmasına gereksinim vardır. Bu tanım genetik algoritmanın özellikle seçim ve üreme işlemlerinde kullanılmak durumundadır. Örneğin havuzun en iyilerine bir kuşak daha yaşam hakkı tanımak, kötülerini (iyileşirler umuduyla) daha fazla değişime uğratmak uygulanan yöntemlerdendir. Kromozomun yapısının belirlenmesine koşut olarak kromozom üzerinde tanımlı bireyin çözüm olması bakımından iyiliğini sayısal olarak veren bir fonksiyon (uygunluk fonksiyonu) yazılması gerekmektedir.

### 3. EVİRİCİLERİN HARMONİK ELİMİNASYONU

#### 3.1 Giriş

Bu bölümde; öncelikle eviriciler, devre yapıları, çalışma prensipleri, harmonikler ve eviricilerin harmonik eliminasyonu ile darbe genişlik modülasyonu hakkında bilgi verilmiştir.

Eviriciler, (inverters) bir ve ya üç fazlı alternatif gerilim ile beslenecek yük için doğru gerilim kaynağından yüke enerji transferi yapan, frekansı ve gerilimi birbirinden bağımsız olarak ayarlanabilen devrelerdir.

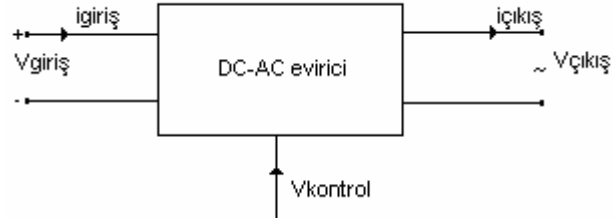
Evirici, anahtarlama elemanları ile endüktans, kapasite, trafolar ve dirençler gibi elemanlardan oluşur. Eviricilerin girişlerindeki doğru gerilim ya dc-dc dönüştürücü yardımı ile yükseltilerek ya da doğrudan, güç elektroniği devre elemanlarının uygun bir biçimde anahtarlama ile alternatif gerilime dönüştürülür. Ardından transformatör yardımı ile alternatif gerilim istenen çıkış gerilimi değerine ayarlanır. Ayrıca giriş ve çıkışta kullanılacak alçak geçiren filtreler giriş ve çıkıştaki işaretlerin içindeki yüksek frekanslı bileşenlerin azalmasını sağlar [15].

Eviricilerin çıkışlarının, doğru akım bileşeni olmayan sinüzoidal akım ve gerilim olması istenir. Ancak, eviricilerde çıkış akım ve gerilimi sinüzoidal olmayabilir. Küçük ve orta güçlü uygulamalar için kare dalga çıkış kabul edilebilirken; büyük güçlü uygulamalarda sinüzoidal çıkış istenir [16].

Eviricilerde sinüzoidal çıkış elde etmek için kullanılan yöntemlerden en önemlisi darbe genişlik modülasyonudur (PWM).

### 3.2 Eviriciler

Doğru gerilimi alternatif gerilime çeviren eviricilere ait prensip şema şekil 3.1’de görülmektedir. Eviriciler, alternatif besleme geriliminin olmadığı karavan, yat ve ya uzak alanlardaki ihtiyaçları karşılamak amacıyla kullanılmakla beraber [15], alternatif akım makinelerinin beslenmesinde, gerilim ve frekansı kontrol edilebilen güç kaynaklarında, kesintisiz güç kaynaklarında, frekans dönüştürücülerinde, endüksiyonla ısıtmada, ultrasonik dalga üretiminde, generatörlerde, aktif güç şebeke filtrelerinde kullanılırlar [17].



Şekil 3.1: DC-AC Evirici

Eviricilerin girişinde bulunan doğru akım kaynağı; doğrultucu ya da aküdür. Aküler aynı zamanda ac devre kesicilerin ve yük altında çalışabilen devre kesicilerin bobinlerine enerji vermek ve varsa doğru gerilim ile beslenebilen yükler için kullanılırlar. Özellikle enerji santrallerinde bu sistemler bulunduğundan; eviricilerin girişinde enerji santrallerinde akü kullanılır. Ayrıca alternatif besleme geriliminin bulunmadığı yerlerde de kaynak olarak akü kullanılır. Eviricilerin doğrultucu ile beslenmesine örnek olarak ise; kesintisiz güç kaynağı, rüzgar ve ya güneş panelleri uygulamaları verilebilir [15].

Küçük güçteki eviriciler (10kVA’ya kadar) genellikle tek fazlı olarak kullanılırken, orta ve büyük güçlü eviriciler üç fazlı olarak kullanılırlar.

Son yıllardaki güç elektroniği yarı iletken elemanları teknolojisindeki gelişmeler sayesinde, güç elektroniği dönüştürücülerinde kullanılan anahtarlama elemanlarının anahtarlama frekansı yükselmiştir. Yüksek hızdaki anahtarlama elemanları sayesinde, çıkış gerilimindeki harmonikler anahtarlama tekniği değiştirilerek de azaltılabilir [18].

Eviricilerin üreteceği dalga şekilleri ve frekansları kullanılan yarı iletken elemanların (tristör, BJT, IGBT, MOSFET) karakteristiklerine, iletim ve tıkama sürelerine bağlıdır.

PWM eviricilerde yüksek anahtarlama hızları gerektiğinden genellikle IGBT veya MOSFET kullanılır. Yüksek güçler için ise tristör veya GTO kullanılabilir, fakat bu durumda yarı periyottaki darbe sayısı sınırlı kalmaktadır. Ayrıca, eviricilerde anahtarlama elemanı olarak tristör tercih edilmemesinin bir diğer nedeni, tristörler için sürme sinyalinden sonra tristörün kesime gitmesi içinde belirli bir süre beklemek gereklidir [19]. Mosfet kullanılması durumunda ise üretilecek akım ve gerilim değerleri sınırlıdır. Ancak oldukça hızlı ve gerilim kontrollü olan bu elemanlar küçük güçlü ve yüksek frekanslı uygulamalarda kullanılır. Mosfetler birbirlerine seri ya da paralel bağlanarak daha yüksek güçlü eviricilerde kullanılabilirler. BJT'ler ise tristörlere göre hızlı mosfetlere göre yavaştır. İletim için büyük baz akımına ihtiyaç duyarken kesim için bu akımın kesilmesi yeterlidir.

Çıkış geriliminin frekansı ve genliği eviricideki elemanların anahtarlama durumu değiştirilerek ayarlandığından sistemin cevap verme süresi çok kısadır.

Eviriciler besleme kaynaklarına göre gerilim beslemeli ve akım beslemeli diye iki gruba ayrılır. Gerilim beslemeli eviriciler; omik ve endüktif yükleri beslemek ile sabit moment isteyen alternatif akım motor uygulamalarında idealdirler. Akım beslemeli eviriciler ise; büyük kapasitif yükleri beslemekte uygundur. Yük harmonik akımlara karşı yüksek empedans gösteriyorsa gerilim beslemeli eviriciler, yük harmonik akımlara karşı düşük empedans gösteriyorsa akım beslemeli eviriciler kullanılmalıdır [17].

Eviriciler tetiklenme tiplerine göre ise kare dalga ve PWM evirici olmak üzere iki ana gruba ayrılır.

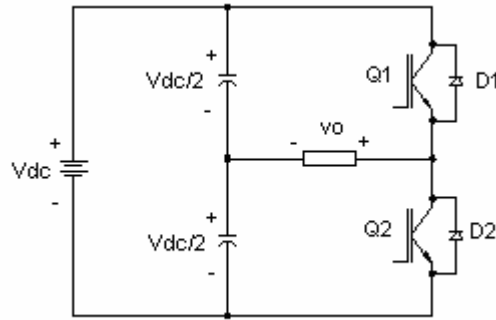
### **3.3 Evirici Devreleri**

Eviriciler devre yapılarına göre yarım köprü evirici, tam köprü evirici ve çıkış transformatörlü orta nokta bağlantılı evirici olmak üzere üçe ayrılır.

Yarım köprü evirici küçük güçlü uygulamalarda daha yaygındır. Tam köprü evirici ise, çıkışın kontrol edilebilmesi için PWM tekniğinin kullanılmasına izin verir. Yarım köprü ve tam köprü evirici devrelerinin çalışmasında birbirinden ayrılırken, diğer taraftan her ikisi içinde aynı formüller geçerlidir [19].

### 3.3.1 Yarım Köprü Evirici

Şekil 3.2'deki topolojiye sahip olan yarım köprü eviricide bulunan iki anahtar doğru akım kaynağı iki kondansatörle ikiye ayırır. Devredeki iki anahtara ters paralel yönde anahtarları korumak ve ters yöndeki enerji akışına izin vermek için diyot bulunmaktadır [19].



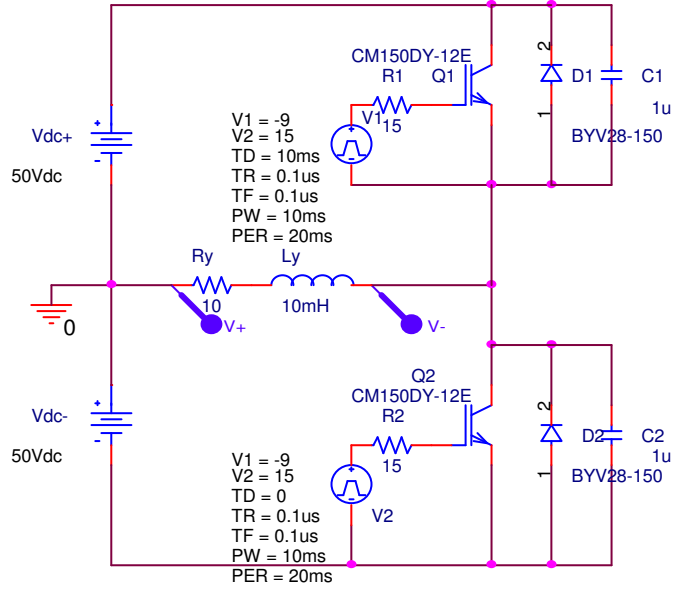
**Şekil 3.2:** Yarım Köprü Evirici Devre Şeması

Her bir kondansatörün üzerinde  $V_{dc} / 2$  gerilimi bulunmaktadır. Q1 anahtarı iletimde iken, yüke  $V_{dc} / 2$  gerilimi uygulanır. Yük akımı pozitif iken, akım Q1 anahtarı üzerinden akar ve yükü besler. Eğer yük akımı negatif ise, akım D1 diyotu üzerinden akar ve enerji doğru akım kaynağına doğrudur [20].

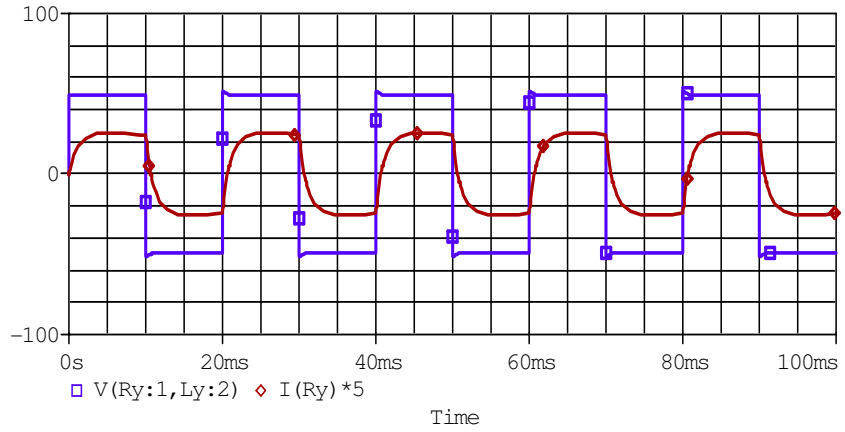
Benzer şekilde, eğer Q2 anahtarı iletimde ise,  $-V_{dc} / 2$  gerilimi yüke uygulanır. Yük akımı pozitif olduğu durumda akım D2 diyotu üzerinden kaynağa döner. Akım negatif olduğunda ise Q2 anahtarı üzerinden akan akım yükü besler.

Q1 ve Q2'nin aynı anda iletme girmemesi gerekir. Aksi halde; kaynak kısa devre edilmiş olur.

Şekil 3.3'de yarım köprü eviriciye ait Orcad uygulaması ve şekil 3.4'de yük akım ve gerilimi dalga şekilleri görülmektedir.



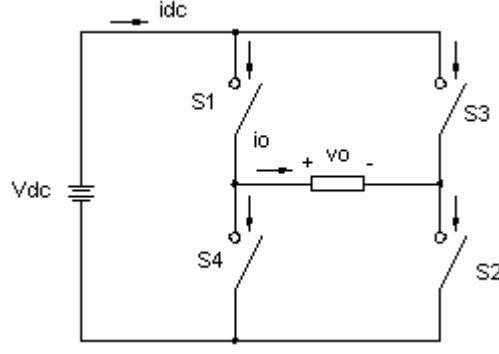
Şekil 3.3: Yarım Köprü Eviricinin Orcad Uygulaması



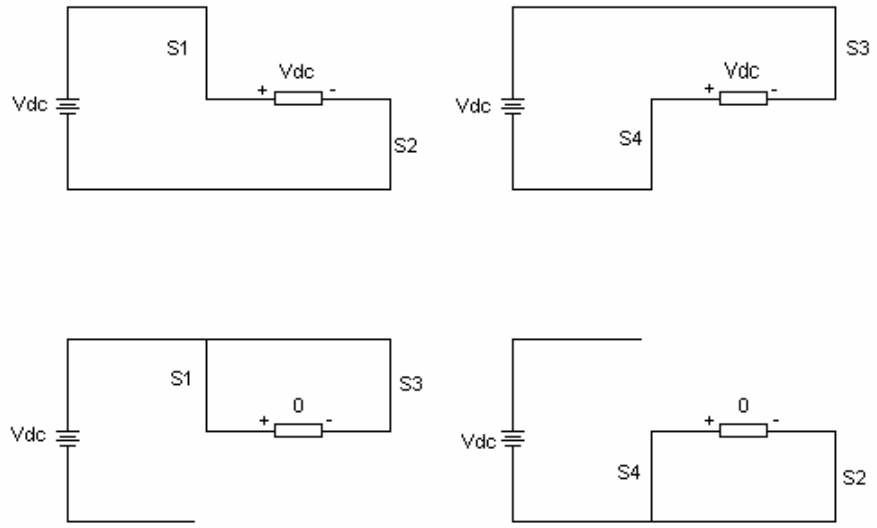
Şekil 3.4: Yarım Köprü Eviricinin Orcad Simülasyonu

### 3.3.2 Tam Köprü Evirici

Tam köprü eviriciye ait devre şeması şekil 3.5’de verilmiştir. Tam köprü eviricide sinüzoidal çıkış, doğru akım girişten anahtarları uygun aralıklarla açıp kapatılarak elde edilir. Çıkış gerilimi  $v_o$ , kapanan anahtarlara göre,  $+v_{dc}$ ,  $-v_{dc}$  ya da sıfır olabilir (Şekil 3.6) [17].



**Şekil 3.5:** Tam Köprü Eviricinin Devre Şeması

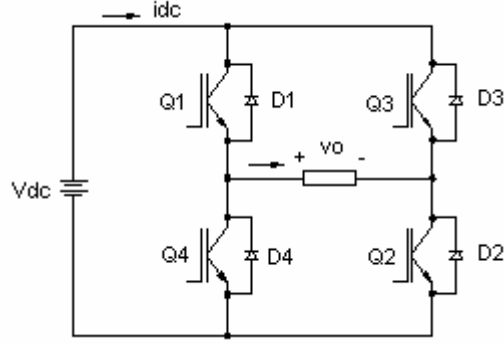


**Şekil 3.6:** Anahtarlama Sırasında Oluşan Çıkış Gerilimi ve İletimdeki Anahtarlar

S1 ve S4 veya S2 ve S3'ün aynı anda ilettime sokulması girişte bulunan doğru gerilim kaynağını kısa devre eder. Bu nedenle S1 ve S4 ile S2 ve S3 sırayla ilettime sokulup çıkartılır. Gerçekte eğer S4, S1 kesime götürülürken ilettime sokulursa, geçici rejimde S1 ve S4 kaynağı kısa devre ederler ve anahtarlama elemanlarında yüksek akımlar oluşur. Bunu engellemek için S1'in kesime götürülüp S4'ün ilettime sokulması arasında ölü zaman (dead time) adı verilen bir boşluk yaratılmalıdır. Bu durum S2 ve S3 içinde gerçekleştirilmelidir. Ölü zaman anahtarların tahmini en büyük kesime gitme (turn-off) süresinden büyük seçilmelidir [17].

Tam köprü eviricilerin IGBT ve ters diyot ile oluşturulduğu devre şeması ise şekil 3.7'de verilmiştir.





**Şekil 3.7:** Tam Köprü Eviricinin Yarı İletken Elemanlarla Devre Şeması

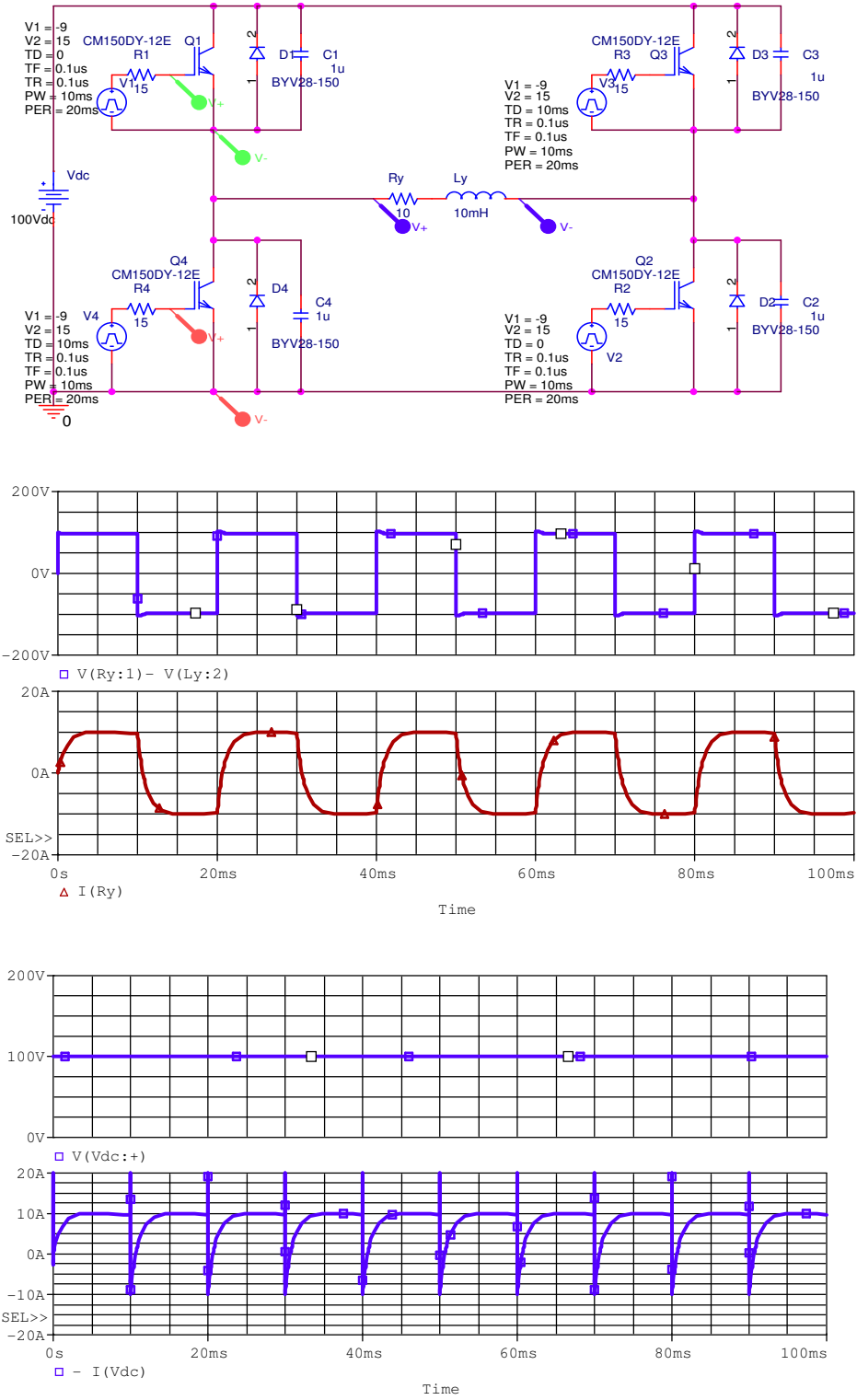
### 3.4 Eviricilerin Tetiklenmesi

Eviriciler tetiklenmelerine göre kare dalga evirici ve PWM evirici olmak üzere ikiye ayrılır.

#### 3.4.1 Kare Dalga Evirici

Tam köprü eviricilerde en basit anahtarlama tekniği ile kare dalga çıkış gerilimi elde edilir. S1 ve S2 kapatıldığında (Q1 ve Q2 iletme sokulduğunda) yüke  $+V_{dc}$ , S3 ve S4 kapatıldığında (Q3 ve Q4 iletme sokulduğunda) ise yüke  $-V_{dc}$  uygulanmış olur. Periyodik olarak bu anahtarlama tekrarlanırsa yük üzerinde  $+V_{dc}$  ve  $-V_{dc}$  arasında değişen bir gerilim elde edilir. Sinüzoidal olmayan bu gerilim otomatik kepenkler ve aydınlatma sistemleri gibi bazı uygulamalar için uygun olabilir [17].

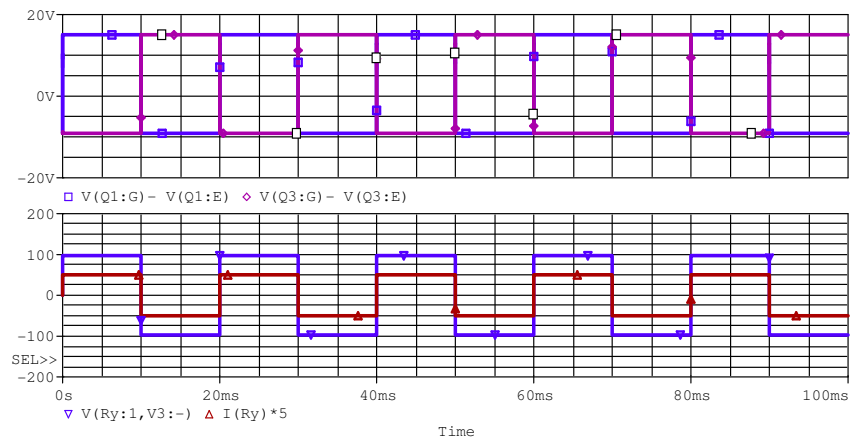
Tam köprü eviricinin kare dalga ile anahtarlandığı durumdaki Orcad simülasyonunun devre şeması ve simülasyon sonuçları şekil 3.8’de gösterilmiştir.



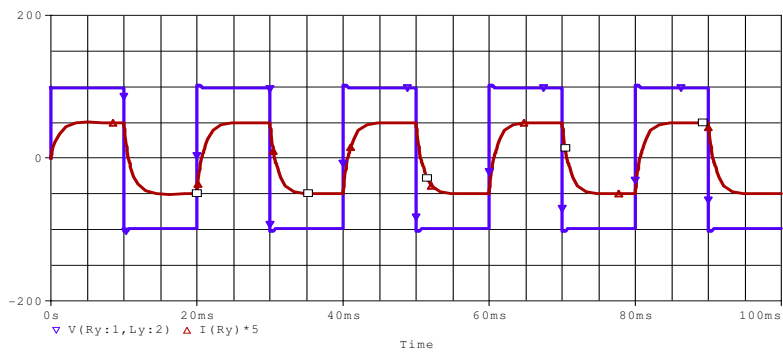
**Şekil 3.8:** Kare Dalga Tam Köprü Eviricinin Orcad Simülasyonu

Çıkış akımının dalga şekli yükün karakteristiğine bağlıdır. Omik yük için çıkış akımının dalga şekli çıkış gerilimi ile aynıdır (Şekil 3.9). Endüktif yükte ise; çıkış akımı çıkış gerilimine oranla daha fazla sinüzoidal karakteristikte olacaktır. Bunun

nedeni; endüktansın filtre etkisi göstermesidir (Şekil 3.10). Endüktif yükler için tam köprü eviricilerin tasarımında çift yönlü anahtarlama elemanları kullanılmasına dikkat edilmelidir. Simülasyonda da görüldüğü gibi tam köprü eviricide anahtarlama elemanları R-L yükte hem pozitif hem de negatif akım taşırlar. Elektronik devre elemanları genellikle tek yönlü akım taşıyabildikleri için, bu sorun her anahtarlama elemanına paralel olarak ters yönde diyot konularak çözülür. Anahtar üzerinden akacak olan negatif akım diyot üzerinden akar. Gerçekte güç elektroniğindeki yarı iletken modüller (IGBT, Mosfet) genellikle anahtarla birlikte ters diyota da sahiptir. Bu nedenle ikinci bir ters diyot kullanmaya gerek yoktur.



**Şekil 3.9:** Omik Yüklü Tam Köprü Kare Dalga Eviricinin Simülasyonu (Çıkış Gerilimi ve Akımı ile Anahtarlama İşaretleri)



**Şekil 3.10:** Endüktif Yüklü Tam Köprü Kare Dalga Eviricinin Simülasyonunda Çıkış Gerilimi ve Akımı

### 3.4.1.1 Devre Analizi

Kare dalga evirici için seri direnç ve endüktans (R-L) yükünde  $t = 0$ 'da S1 ve S2 iletme sokulursa yüke  $+V_{dc}$  değerinde gerilim uygulanır ve yük üzerinden ve S1, S2 anahtarları üzerinden akan akım artmaya başlar. Devreye ait diferansiyel denklem;

$$L \cdot \frac{di(t)}{dt} + R \cdot i(t) = V_{dc} \quad (3.1)$$

Çıkış akımı zorlanmış ve doğal çözümlerinin toplamıdır.

$$i_o(t) = i_f(t) + i_n(t) \quad (3.2)$$

$$= \frac{V_{dc}}{R} + A \cdot e^{-t/\tau} \quad 0 \leq t \leq \frac{T}{2} \quad (3.3)$$

A: ilk durumdan elde edilen sabittir. ( $t=0$ )

$$\tau = \frac{L}{R} \quad (3.4)$$

$t=T/2$ 'de, S1 ve S2 kesime götürülürken, S3 ve S4 iletme sokulur. Yük üzerindeki gerilim  $-V_{dc}$  olur. Akım ise;

$$i_o(t) = -\frac{V_{dc}}{R} + B \cdot e^{-(t-T/2)/\tau} \quad \frac{T}{2} \leq t \leq T \quad (3.5)$$

B: ilk koşuldanda elde edilen sabittir. ( $t=T/2$ )

Devre ilk defa enerji verildiğinde ve başlangıçta endüktans akımı sıfır iken, sistem kararlı hale geçmeden önce geçici bir durum oluşur. Sürekli halde çıkış akımı periyodik ve sifıra göre simetriktir. Denklem 3.3'de akım için ilk koşulu  $I_{min}$  ve denklem 3.5'de ise  $I_{max}$  olarak alalım.

$t = 0$  'da denklem 3.3:

$$i_0(0) = \frac{V_{dc}}{R} + A \cdot e^0 = I_{\min} \quad (3.6)$$

$$A = I_{\min} - \frac{V_{dc}}{R} \quad (3.7)$$

$t = T/2$  'de denklem 3.5:

$$i_0(T/2) = -\frac{V_{dc}}{R} + B \cdot e^0 = I_{\max} \quad (3.8)$$

$$B = I_{\max} + \frac{V_{dc}}{R} \quad (3.9)$$

Sonuç olarak sürekli halde akım dalga şekillerini şu şekilde tanımlayabiliriz.

$$i_0(t) = \begin{cases} \frac{V_{dc}}{R} + \left( I_{\min} - \frac{V_{dc}}{R} \right) \cdot e^{-t/\tau} & 0 \leq t \leq \frac{T}{2} \\ -\frac{V_{dc}}{R} + \left( I_{\max} + \frac{V_{dc}}{R} \right) e^{-(t-T/2)/\tau} & \frac{T}{2} \leq t \leq T \end{cases} \quad (3.10)$$

Bu denklemleri  $T/2$ 'de eşitlersek;

$$i(T/2) = I_{\max} = \frac{V_{dc}}{R} + \left( I_{\min} - \frac{V_{dc}}{R} \right) \cdot e^{-(T/2\tau)} \quad (3.11)$$

Şekil 3.10'da da görüldüğü gibi  $I_{\max} = -I_{\min}$  'dir. Buna göre;

$$I_{\max} = -I_{\min} = \frac{V_{dc}}{R} \left[ \frac{1 - e^{-T/2\tau}}{1 + e^{-T/2\tau}} \right] \quad (3.12)$$

### 3.4.1.2 Fourier Serisi Analizi

Periyodik bir fonksiyon olan  $f(t)$ 'nin Fourier serisine açılımı şu şekilde yapılabilir.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cdot \cos(nw_0t) + b_n \cdot \sin(nw_0t)] \quad (3.13)$$

$$a_0 = \frac{1}{T} \int_0^T f(t) \cdot dt = \frac{1}{2\pi} \int_0^{2\pi} f(\theta) d\theta \quad (3.14)$$

$$a_n = \frac{2}{T} \int_0^T f(t) \cdot \cos(nw_0t) dt = \frac{1}{\pi} \int_0^{2\pi} f(\theta) \cdot \cos(n\theta) d\theta \quad (3.15)$$

$$b_n = \frac{2}{T} \int_0^T f(t) \cdot \sin(nw_0t) dt = \frac{1}{\pi} \int_0^{2\pi} f(\theta) \cdot \sin(n\theta) d\theta \quad (3.16)$$

$f(t)$ 'de ki sinüs ve cosinüs terimleri düzenlenerek aşağıdaki gibi yazılabilir.

$$f(t) = a_0 + \sum_{n=1}^{\infty} C_n \cdot \cos(nw_0t + \theta_n) \quad (3.17)$$

$$C_n = \sqrt{a_n^2 + b_n^2} \quad (3.18)$$

$$\theta_n = \tan^{-1} \left( \frac{-b_n}{a_n} \right) \quad (3.19)$$

Eviricinin çıkış akım ve gerilimini Fourier serisinin terimlerine açmak; eviriciyi, yük akımını analiz etmenin ve çıkış gücünü hesaplamanın en kolay yoludur. Çıkışında doğru akım bileşeni bulunmayan bir eviricinin çıkış gerilimi ve akımı şu şekilde ifade edilebilir.

$$v_o(t) = \sum_{n=1}^{\infty} V_n \cdot \sin(n\omega_o t + \theta_n) \quad (3.20)$$

$$i_o(t) = \sum_{n=1}^{\infty} I_n \cdot \sin(n\omega_o t + \theta_n) \quad (3.21)$$

$$v_{o(t)} = V_0 + V_1 \cdot \sin(\omega t + \theta_1) + V_2 \cdot \sin(2\omega t + \theta_2) + \dots + V_n \cdot \sin(n\omega t + \theta_n) + \dots \quad (3.22)$$

Burada,

$V_0$  : Doğru bileşen

$V_1$  : Temel bileşenin maksimum değeri

.

.

$V_n$  : n. harmoniğin maksimum değeri

$\theta_n$  : Rölatif açısal fark

Fourier serisine açılacak olan dalga şeklinin ya da fonksiyonun varsa simetrisinden yararlanarak serinin terimlerini belirlemek basitleştirilebilir [19]. Eviricilerin çıkış dalga gerilimleri çeyrek dalga simetrisine sahip olduğu için basitleştirilebilir.

Kare dalga eviricilerde, Fourier serisi çıkış gerilimi ve akımının yarım dalga simetrisine sahip olmasından dolayı, çift harmonikler içermez. Buna göre;

$$v_o(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4 \cdot V_{dc}}{n \cdot \pi} (\sin(n \cdot \omega_o \cdot t)) \quad (3.23)$$

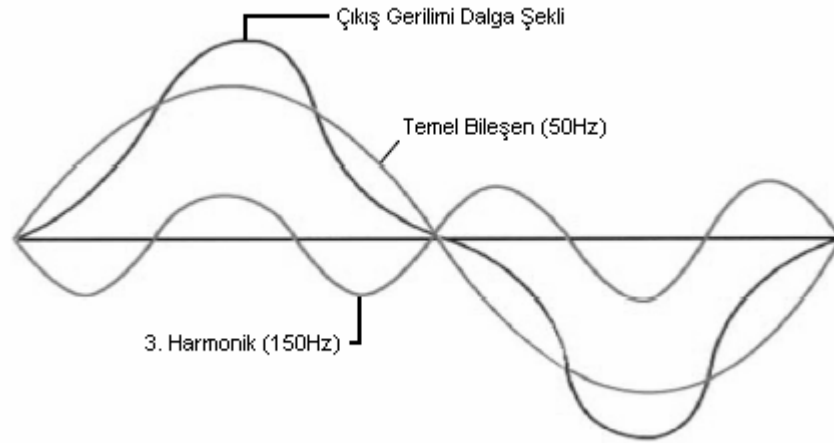
### 3.4.1.3 Toplam Harmonik Dağılımı (THD)

Evirici kullanılmadaki amaç; doğru gerilim kaynağı ile alternatif gerilim ile beslenmesi gereken yükleri beslemektir. İdeal olarak, eviricilerin çıkış akım ve

gerilim dalga şekilleri saf sinüzoidaldir. Elektronik ve diğer lineer olmayan yüklerden dolayı, akım ve gerilim dalga şekilleri bozulmaktadır. Sinüzoidal formda meydana gelen bu değişime harmonik adı verilir [21]. Güç elektroniği elemanları harmonik üretmelerinin yanı sıra harmoniklere karşı duyarlıdır, çalışmaları olumsuz yönde etkilenebilir. Bu nedenle; sistemin doğru bir şekilde çalışabilmesi için harmoniklerin kontrol edilmesi daha da önem kazanmaktadır.

Harmonikler temel frekansın (çıkış frekansının) tam sayı katlarında (1, 2, 3...) sinüs formunda oluşmaktadır (Şekil 3.11). Eviricinin çıkışındaki sinüs olmayan dalga şeklinin kalitesi ya da çıkış dalga şeklinin temel bileşene dalga şekli açısından yakınlığı, yani bozulmanın miktarı toplam harmonik dağılım (THD-total harmonic distortion) kavramı ile belirlenir. Toplam harmonik dağılım bozulmayı, temel gerilim ya da akım dalga şeklinin yüzdesi olarak açıklar [21]. Çıkışta doğru akım bileşeni olmadığını varsayarsak;

$$THD_V = \frac{\sqrt{\sum_{n=2}^{\infty} V_{n,rms}^2}}{V_{1,rms}} = \frac{\sqrt{V_{rms}^2 - V_{1,rms}^2}}{V_{1,rms}} \quad (3.24)$$



**Şekil 3.11:** Farklı Frekanslardaki Harmonikler

Akımın THD'si için formülde gerilimler yerine akımları yazmak yeterli olacaktır. Harmonik akımlarının akım yolundaki elemanlar üzerindeki etkisi akım THD'sini belirlerken, paralel bağlı elemanlar üzerindeki etki gerilim THD'sini gösterir. Güç



sistemlerinde gerilim THD'si bütün yükleri etkilediği için uygulamalarda sınırlamalar getirilir.

#### **3.4.1.4 Harmoniklerin Elenmesi**

Güç elektroniği devrelerini tasarlarken amaçlardan biri çıkış gerilimindeki harmonikleri azaltmaktır [22]. Akım harmonikleri gerilim harmoniklerine neden oldukları için elimine edilmek istenirler. Ancak asıl amaç gerilim harmoniklerinin azaltılmasıdır. Harmoniklerin azaltılmak istenmesinin başlıca nedenleri; mekanik rezonans sonucu istenmeyen gürültü, aşırı ısınma nedeni ile sistemdeki elemanların izolasyonlarının bozulması olarak ortaya çıkmaları ya da güç faktörünü (PF) azaltmalarıdır.

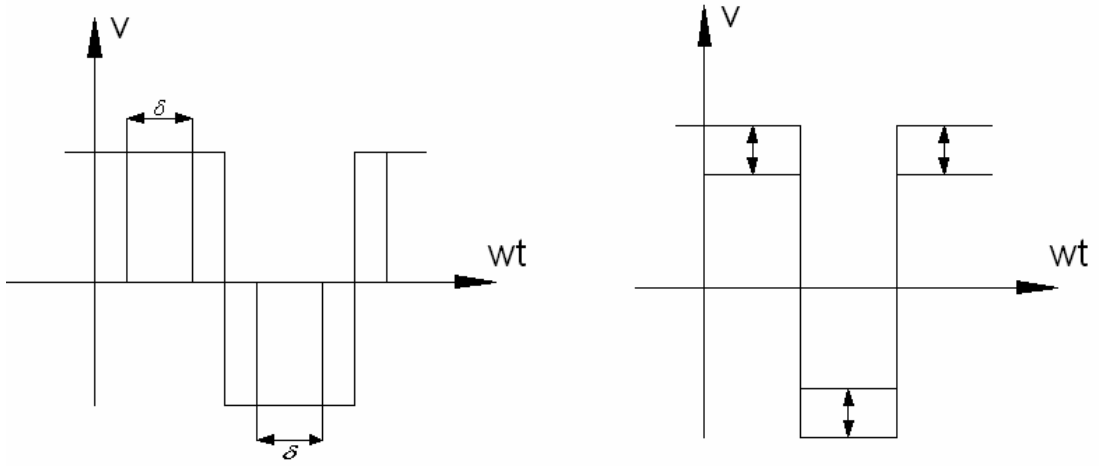
Eviricilerin çıkışlarının, doğru akım bileşeni olmayan sinüzoidal akım ve gerilim olması istenir. Ancak, eviricilerde çıkış akım ve gerilimi sinüzoidal olmayabilir. Küçük ve orta güçlü uygulamalar için kare dalga çıkış kabul edilebilirken; büyük güçlü uygulamalarda sinüzoidal çıkış istenir [16].

Çıkıştaki harmonikleri azaltmak için kullanılan tekniklerden bazıları; devrelerde kullanılan anahtarlama elemanlarını ve anahtarlama açılarını kontrol etmek, vektör kontrol ya da istenmeyen harmonikleri elimine etmek için uygun çıkış filtreleri tasarlamaktır.

Eviricilerin çıkışındaki sinüzoidal gerilimdeki harmonikleri azaltmak için kullanılan yollardan biri darbe genlik modülasyonu (PWM) ile anahtarlama açılarını kontrol etmektir. Bu yöntem en çok kullanılan ve ekonomik yoldur [23].

#### **3.4.1.5 Genlik Ve Harmonik Kontrolü**

Tam köprü eviricide kare dalga çıkış geriliminin genliği, kaynak geriliminin genliğinin değiştirilmesi ile kontrol edilebilir. Bunun için eviricilerin girişinde kontrollü doğrultucu kullanılabilir. Eviricinin çıkış genliği, anahtarların iletimde kalma süreleri ile de kontrol edilebilir (Şekil 3.12).

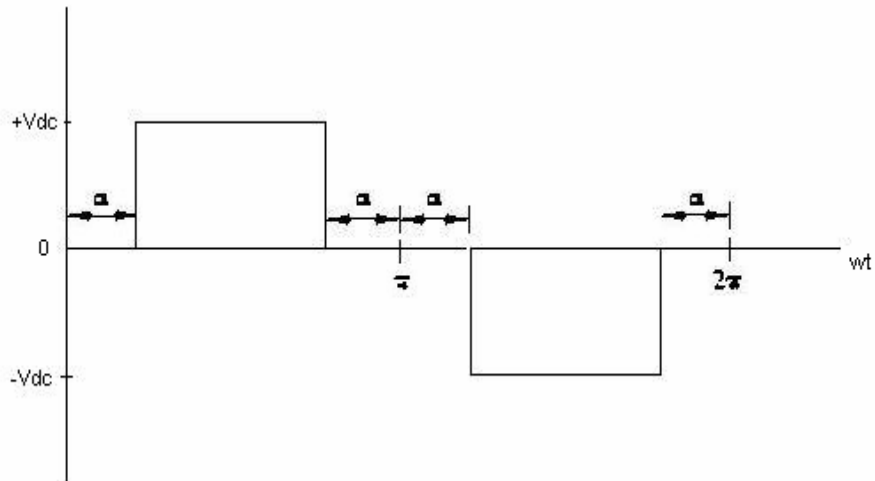


**Şekil 3.12:** Darbe Geniřlięi Kontrolü, Genlik Kontrolü

Şekil 3.13'deki gibi  $\alpha$  aralıęı deęiřtirilerek ıkıř geriliminin efektif deęeri kontrol edilebilir. Bu durum iin anahtarlama sırası ise Őekil 3.14'de gsterilmiřtir.

Şekil 3.13'deki gerilimin efektif deęeri;

$$V_{eff} = \sqrt{\frac{1}{\pi} \int_{\alpha}^{\pi-\alpha} V_{dc}^2 \cdot d(\omega t)} = V_{dc} \cdot \sqrt{1 - \frac{2\alpha}{\pi}} \quad (3.25)$$



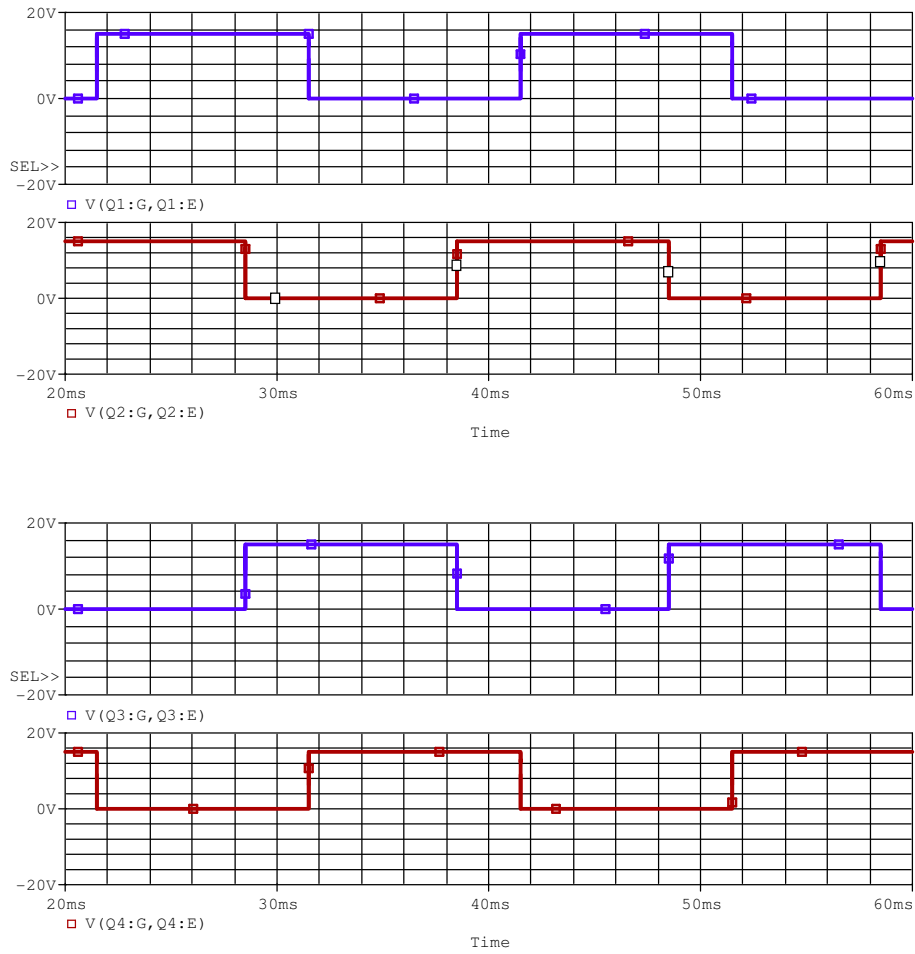
**Şekil 3.13:** Genlik ve Harmonik Kontrolü ile Evirici ıkıř Gerilimi

Şekil 3.13'deki dalga şeklinin Fourier serisine açılımı ise;

$$v_0(t) = \sum_{n,tek} V_n \cdot \sin(n\omega_0 t) \quad (3.26)$$

Yarım dalga simetrisinin avantajını kullanırsak, genlikler şöyle olur;

$$V_n = \frac{2}{\pi} \int_{\alpha}^{\pi-\alpha} V_{dc} \cdot \sin(n\omega_0 t) \cdot d(n\omega_0 t) = \left( \frac{4 \cdot V_{dc}}{n \cdot \pi} \right) \cos(n\alpha) \quad (3.27)$$



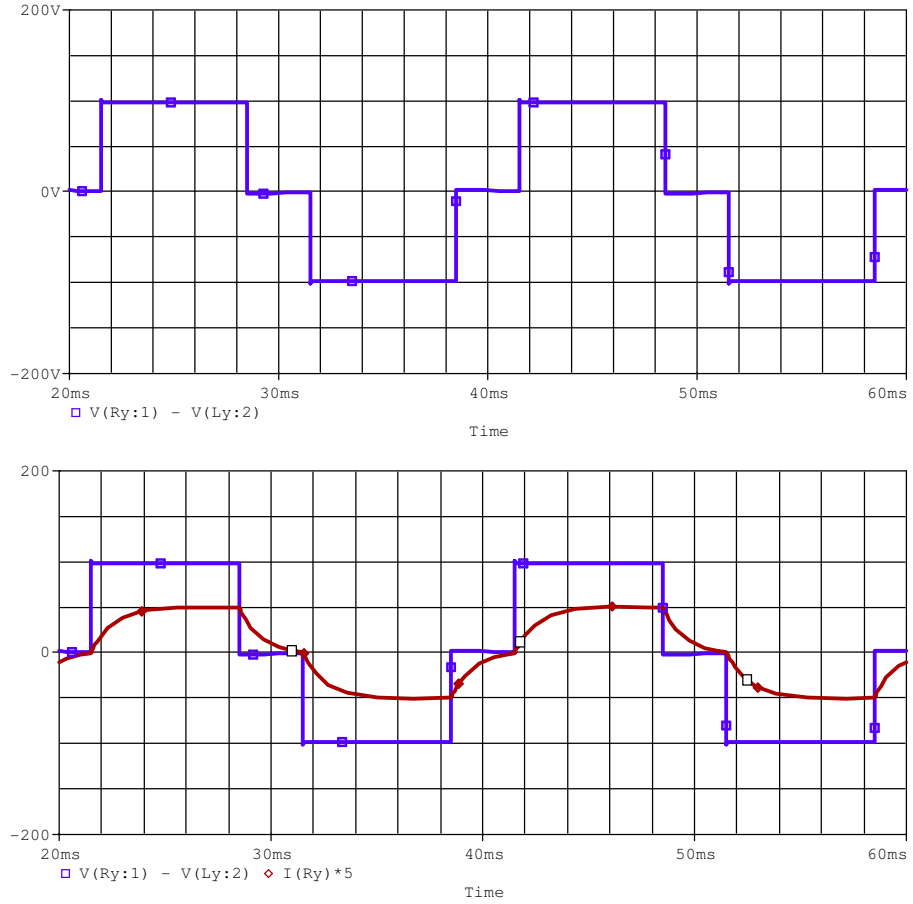
Şekil 3.14: Anahtarlama Sırası

Çıkışın her bir frekans için genliği  $\alpha$ 'nın bir fonksiyonudur. Genlik ve harmonik bileşenler  $\alpha$  uygun bir şekilde seçilerek kontrol edilebilir.

$$V_1 = \frac{4.V_{dc}}{\pi} \cos(\alpha) \quad (3.28)$$

Örneğin denklem 3.28'de n.ci harmonik  $\alpha = 90^\circ/n$  seçilerek elimine edilebilir.

Endüktif yüklü bir tam köprü evircinin şekil 3.14'deki gibi anahtarlama ile elde edilen çıkış gerilim ve akım dalga şekilleri şekil 3.15'de verilmiştir.



**Şekil 3.15:** Orcad Simülasyon Sonuçları

Genlik kontrolü ve harmonik bileşenlerden birinin elenmesi toplam harmonik dağılımının azalması ve ya harmoniklerin elenmesi açısından etkili bir yol değildir. Örneğin;  $\alpha = 30^\circ$  seçilerek üçüncü harmonik elimine edilebilir ve çıkış temel harmoniği denklem 3.29'daki gibi olur. Bunun dışında bir kontrol sağlanamaz. Bu

anahtarlama tekniđi ile hem genlik hem de harmonikleri kontrol etmek için; evirici girişindeki doğru akım kaynak geriliminin kontrol edilebilir olması gereklidir. Diğer taraftan; her bir yarı periyottaki darbe sayısını arttırmakta bir çözüm olabilir. Anahtarlama için her bir periyotta bulunan darbe sayısını artırarak darbe genişlik modülasyonlu (PWM) evirici topolojisi elde edilmektedir.

$$V_1 = \left( \frac{4 \cdot V_{dc}}{\pi} \right) \cos(30^\circ) = 1.1(V_{dc}) \quad (3.29)$$

### 3.4.2 Darbe Genişlik Modülasyonlu (PWM) Eviriciler

Darbe genişlik modülasyonlu (PWM) eviricilerde, anahtarlama elemanları her bir periyot boyunca anahtarlama için birden fazla darbe kullanılır. PWM eviricilerde kare dalga eviricilerden farklı olarak, bir faza ait gerilim üst ve alt kolun iletimde olmasına göre alternatif yarım periyot içinde çok kez doğru akım kaynağının geriliminin pozitif ve negatif değerlerini almaktadır. Bir periyottaki darbe sayısını arttırmaktaki amaç; çıkış geriliminin genliğini kontrol etmek, THD'nin azaltılması, temel bileşene yakın harmoniklerin azaltılması ve çıkış geriliminin genliğini ayarlamaktır.

PWM ile düşük dereceli harmonikleri içermeyen bir çıkış elde edilir. Fakat filtrelenmemiş PWM eviricinin çıkışı yüksek frekansta harmoniklere sahip olduğu için THD'si yüksektir. Ancak geriye kalan yüksek dereceli harmonikleri filtrelemek daha kolaydır. PWM evirici çıkışında filtre kullanılarak sinüzoidal gerilim elde edilebilir. PWM eviricideki anahtarlar kare dalgaya göre daha yüksek frekanslarda anahtarlanır. Bu nedenle PWM eviricilerin filtreleri daha basit, ucuz, hafif ve küçüktürler. PWM eviricinin dezavantajı ise kontrol devrelerinin karmaşık olması ve sürekli anahtarlardan dolayı kayıplarının artmasıdır [24].

PWM eviricilerin çıkışında harmoniklerden kaynaklanan bozulma; çıkış filtresinden, ölü zamanlardan, anahtarlama elemanları üzerindeki gerilim düşümlerinden veya doğru akım kaynak gerilimindeki değişimlerden kaynaklanır [25].

PWM eviricilerde her bir periyottaki darbeler birbirine eşit olabileceği gibi birbirinden farklıda olabilirler.

PWM eviricilerde anahtarlama için iki sinyal gereklidir. Bunlar referans ve taşıyıcı dalga olarak adlandırılır. Referans dalga kare, trapez veya sinüs dalgası olabilir. Referans dalga, çıkış geriliminin frekansını kontrol eder. PWM dalga şekli, yüksek frekanslı bir üçgen taşıyıcı dalgayı düşük frekanslı bir referans dalgası ile karşılaştırmak sureti ile elde edilebilir. Taşıyıcı dalga ile referans dalganın karşılaştırılması sonucunda, iki kutuplu anahtarlama ya da tek kutuplu anahtarlama olmak üzere iki çeşit PWM elde edilir.

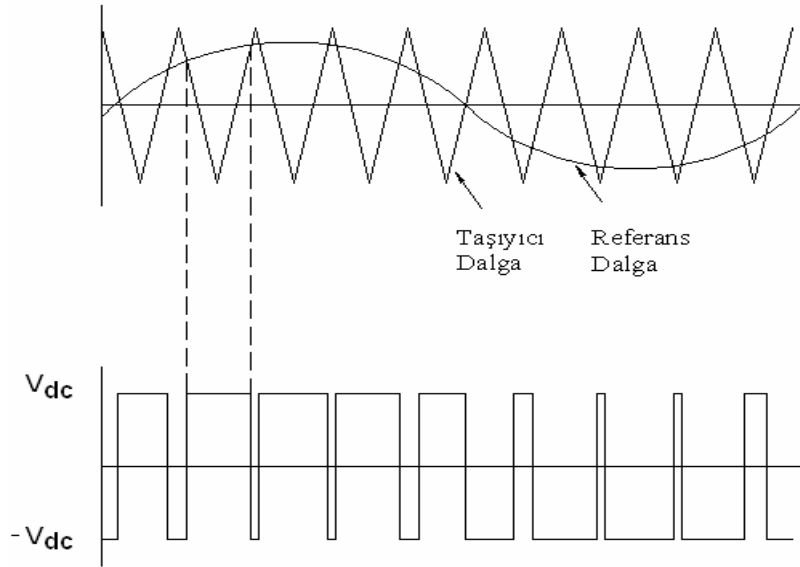
❖ İki kutuplu anahtarlama:

İki kutuplu anahtarlama (Şekil 3.16) sinüs referansının üçgen taşıyıcı dalgadan büyük olduğu yerlerde, çıkış  $+v_{dc}$ , küçük olduğu yerlerde ise çıkış  $-v_{dc}$  olur.

$$v_{\sin} > v_{\text{üçgen}} \Rightarrow v_o = +v_{dc}$$

$$v_{\sin} < v_{\text{üçgen}} \Rightarrow v_o = -v_{dc}$$

(3.30)



**Şekil 3.16:** İki Kutuplu Anahtarlamanın Elde Edilişi

PWM darbeleri hem pozitif hem de negatif değerler aldığı için iki kutuplu anahtarlama denmektedir [17].

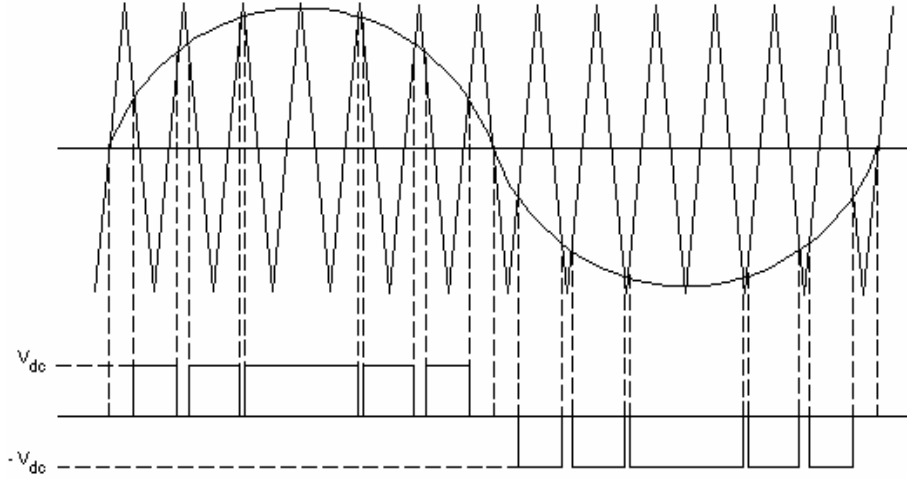
❖ Tek kutuplu anahtarlama:

Tek kutuplu anahtarlama (Şekil 3.17) üçgen taşıyıcı dalga pozitif ve negatif referans ile karşılaştırılır [26].

$$\begin{aligned} v_{\sin} > v_{\text{üçgen}} &\Rightarrow v_o = +v_{dc} \\ v_{\sin} < v_{\text{üçgen}} &\Rightarrow v_o = 0 \end{aligned} \quad (3.31)$$

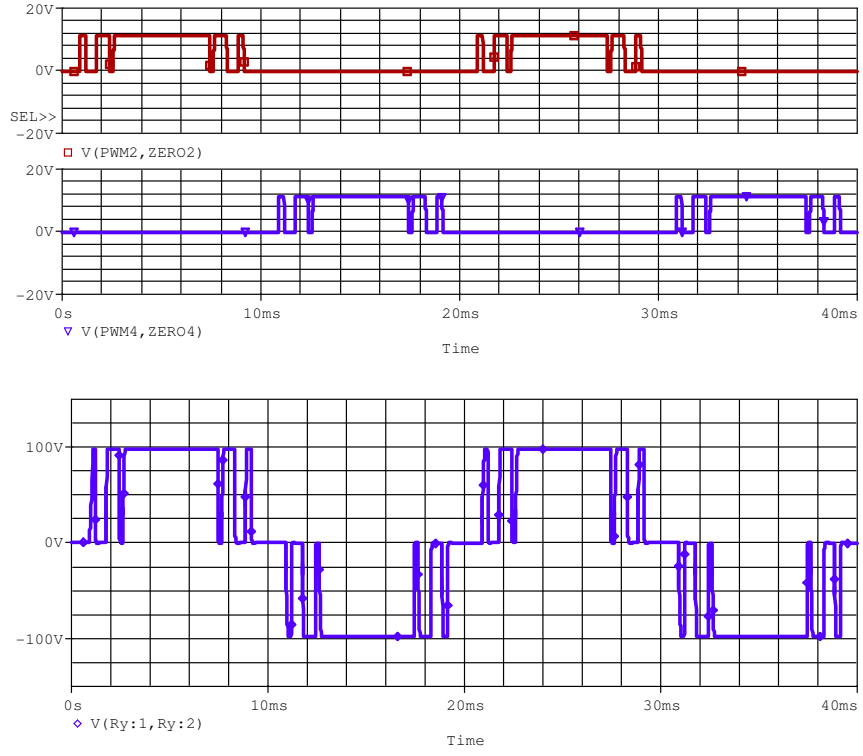
$$\begin{aligned} -v_{\sin} > v_{\text{üçgen}} &\Rightarrow v_o = +v_{dc} \\ -v_{\sin} < v_{\text{üçgen}} &\Rightarrow v_o = 0 \end{aligned} \quad (3.32)$$

Çıkış geriliminin genliği referans dalganın genliği değiştirilerek kontrol edilebilir [27].



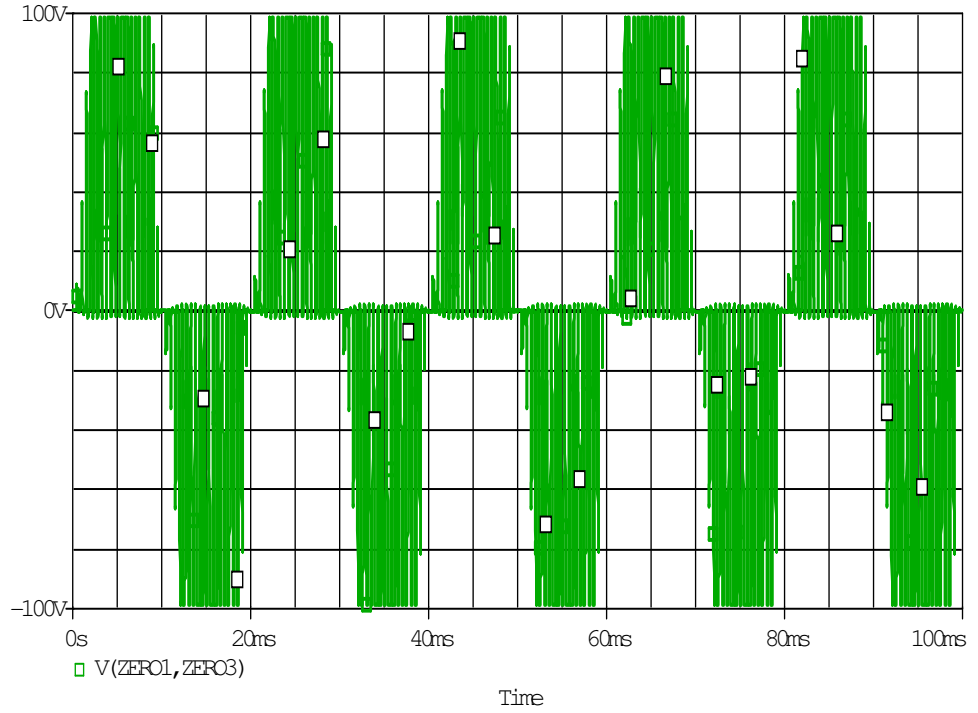
Şekil 3.17: Tek Kutuplu Anahtarlamanın Elde Edilişi

Şekil 3.18'de tek kutuplu anahtarlama ile tam köprü PWM evircinin anahtarlama frekansı 1 kHz iken ORCAD benzetim sonuçları görülmektedir.



**Şekil 3.18:** Anahtarlama Frekansı 1khz iken Orcad Simülasyonu

Şekil 3.19'de tek kutuplu anahtarlama ile tam köprü PWM evircinin anahtarlama frekansı 2 kHz iken ORCAD simülasyon sonuçları görülmektedir.

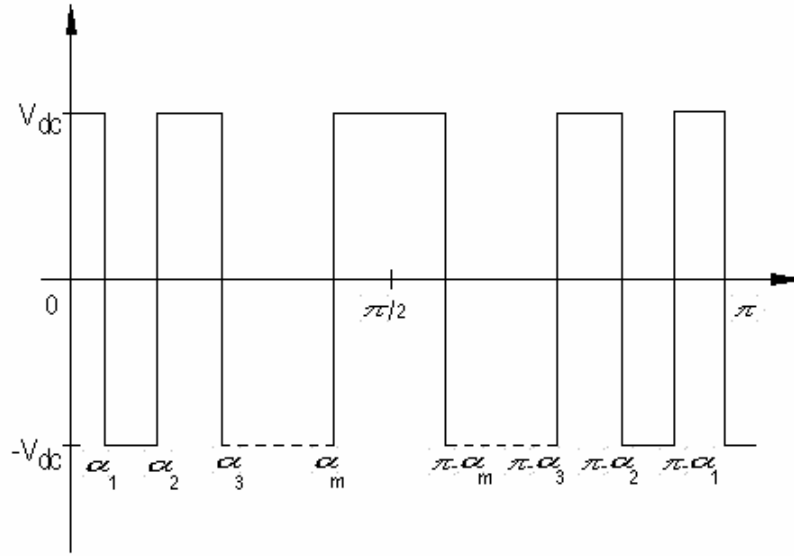


**Şekil 3.19:** Anahtarlama Frekansı 2khz iken Orcad Simülasyonu



Güç elektroniği yarı iletken elemanlarındaki teknolojinin hızla gelişmesi nedeni ile son yıllarda güç elektroniği devrelerindeki anahtarlama frekansları büyümektedir [18]. Taşıyıcı dalganın frekansını arttırmak harmoniklerin oluştuğu frekansında artmasıdır. Yüksek anahtarlama frekansının dezavantajı; anahtarlama elemanlarındaki yüksek kayıpların oluşmasıdır.

### 3.4.2.1 Pwm Eviriciler İçin Fourier Analizi



**Şekil 3.20:** İki Kutuplu Anahtarlama Açılımları

İki kutuplu anahtarlama ile kontrol edilen PWM eviricinin şekilde verilen çıkış dalga gerilimini Fourier serisine açılımı çeyrek dalga simetrisinden dolayı;

$$a_0 = 0$$

$$a_n = 0 \quad n = 2, 4, 6, \dots$$

$$b_n = 0$$

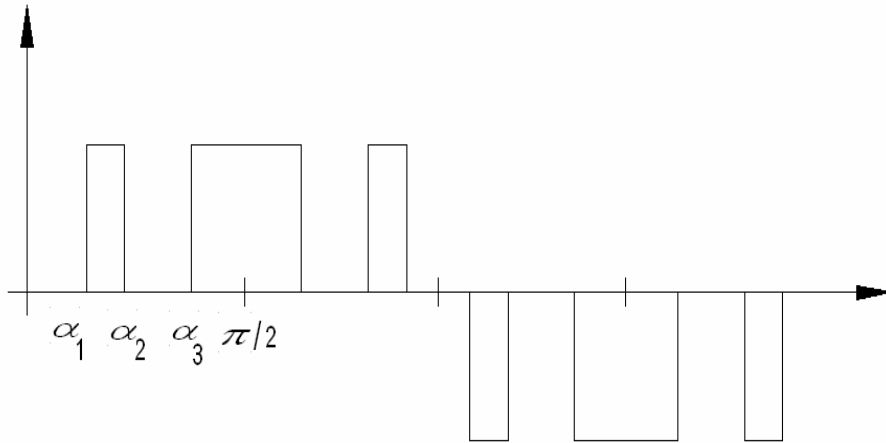
$$a_n = \frac{4}{\pi} \int_0^{\pi/2} V_{dc} \cdot \sin(nwt) dt$$

$$a_n = \frac{4.V_{dc}}{\pi} \left[ \int_0^{\alpha_1} \sin(nwt) dwt - \int_{\alpha_1}^{\alpha_2} \sin(nwt) dwt + \int_{\alpha_2}^{\alpha_3} \sin(nwt) dwt \right. \\ \left. - \int_{\alpha_3}^{\alpha_4} \sin(nwt) dwt + \int_{\alpha_4}^{\pi/2} \sin(nwt) dwt \right] \quad (3.33)$$

$$a_n = \frac{4.V_{dc}}{n.\pi} [1 - 2.\cos(n\alpha_1) + 2.\cos(n\alpha_2) - 2.\cos(n\alpha_3) + 2.\cos(n\alpha_4)] \quad (3.34)$$

Denklemi genelleştirirsek;

$$a_n = \frac{4.V_{dc}}{n.\pi} [1 - 2.\cos(n\alpha_1) + 2.\cos(n\alpha_2) - 2.\cos(n\alpha_3) + \dots \mp 2.\cos(n\alpha_m)] \quad (3.35)$$



**Şekil 3.21:** Tek Kutuplu Anahtarlama Açılı Noktaları

Tek kutuplu anahtarlama ile kontrol edilen PWM eviricinin şekilde verilen çıkış dalga gerilimini Fourier serisine açılımı çeyrek dalga simetrisinden dolayı;

$$a_0 = 0$$

$$a_n = 0 \quad n = 2, 4, 6, \dots$$

$$b_n = 0$$

$$a_n = \frac{4}{\pi} \int_0^{\pi/2} V_{dc} \cdot \sin(nwt) dwt = \frac{4 \cdot V_{dc}}{\pi} \left[ \int_{\alpha_1}^{\alpha_2} \sin(nwt) dwt + \int_{\alpha_3}^{\pi/2} \sin(nwt) dwt \right] \quad (3.36)$$

$$a_n = \frac{4 \cdot V_{dc}}{n \cdot \pi} [\cos(n\alpha_1) - \cos(n\alpha_2) + \cos(n\alpha_3)] \quad (3.37)$$

Tek kutuplu anahtarlamaya ait bağıntı denklem 3.37 deki gibi elde edilir.

### 3.5 Sonuç

Bu bölümde eviriciler tanıtılmış, devre yapıları, çalışma prensipleri ve kullanıldıkları yerler hakkında bilgi verilmiştir. Kare dalga evirici ve darbe genişlik modülasyonlu eviriciler karşılaştırmalı olarak anlatılmıştır ve Orcad simülasyon sonuçları verilmiştir. Görüldüğü gibi, darbe genişlik modülasyonlu eviriciler, çoğu uygulama için çıkışın sinüse daha yakın ve harmoniklerin etkisinin daha kolay azaltılabilmesi açısından yüksek öneme sahiptirler. Ayrıca harmonikler ve harmoniklerin etkisi üzerinde durulmuştur. Fourier serisi analizi ve toplam harmonik dağılım hakkında bilgi verilmiştir. Kare dalga evirici ve darbe genişlik modülasyonlu evirici ayrı ayrı tetiklenme durumlarına göre Fourier serisine açılmıştır ve denklemler ayrıntılı olarak görülmektedir. Son bölümde genetik algoritmalara uygulanacak olan darbe genişlik modülasyonlu eviricinin iki kutuplu anahtarlama durumuna ait denklemler çıkarılmıştır.

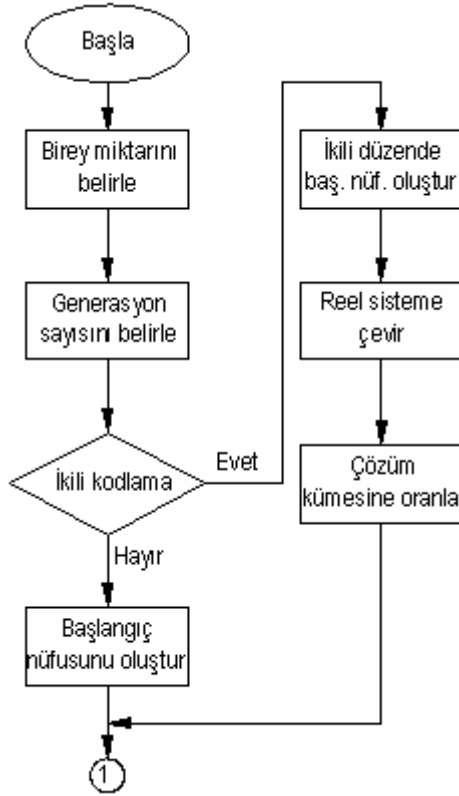
## **4. OPTİMUM AÇI PROBLEMİNİN GENETİK ALGORİTMALARA UYGULANMASI**

### **4.1 Giriş**

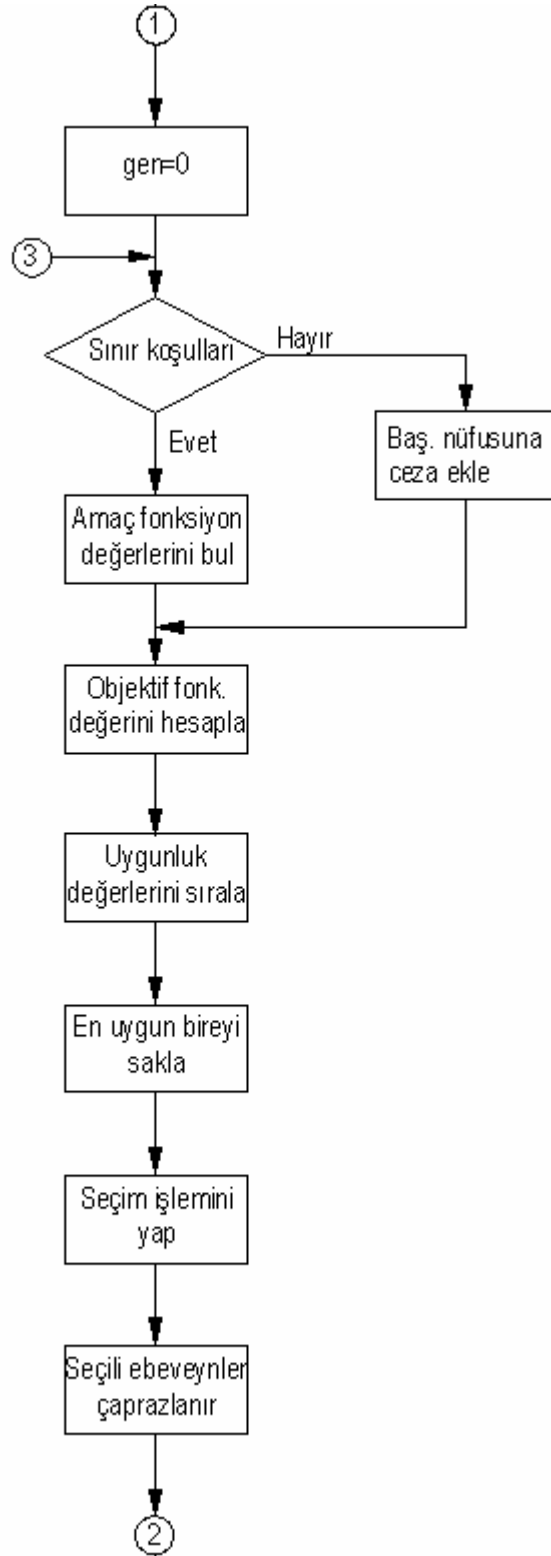
3. bölümde eviricilerin harmonikleride içerecek şekilde matematiksel modelleri verilmişti. Bu bölümde ise; 2. bölümde açıklanan genetik algoritma temelleri de göz önüne alınarak tez konusu olan eviricilerin optimum açı problemi için Matlab’de bir yazılım geliştirilmiştir. Geliştirilen programda genetik algoritmanın gerektirdiği seçim, çaprazlama, mutasyon ve ceza alt programları oluşturulmuştur. Geliştirilen Matlab programının koşturulmasında birey sayısı, generasyon sayısı ve mutasyon oranı gibi parametrelerin etkinlikleri incelenerek eviricilerin harmonik eliminasyonu için gerekli olan optimum açıların belirlenmesine çalışılmıştır. Sonuçta elde edilen veriler klasik optimizasyon yönteminden elde edilen veriler ile karşılaştırılarak değerlendirmeleri yapılmıştır.

## 4.2 Program Aşamaları

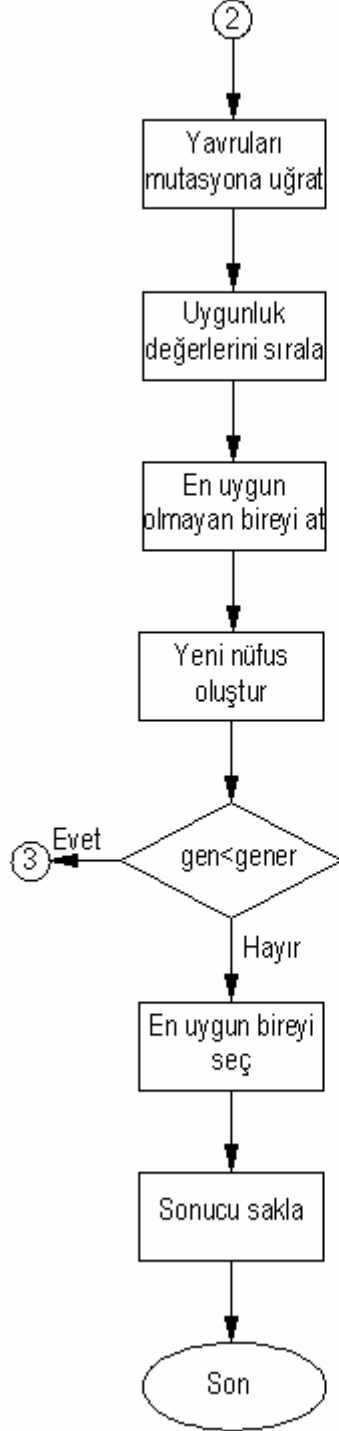
İki kutuplu anahtarlama yapısına sahip eviricilerin anahtarlama açısı noktaları Şekil 3.20’de belirtilmiştir. Bu anahtarlama açılarındaki çalışana sisteme ait çıkış geriliminin Fourier serisi açılımı denklem 3.32’de belirtilmiştir. Tez konusu olan eviricilerin harmonik eliminasyonunda optimum anahtarlama açılarının genetik algoritma ile elde edilmesinde bu denklem esas alınmıştır. Bu denklemin Matlab programında çözülmesi için akış diyagramı oluşturulmuştur. Akış diyagramına bağlı kalınarak Matlab’de gerekli olan ana ve alt programlar yazılmıştır. Matlab’de yazılan optimum açısı probleminin bütün aşamalarını sırası ile gösteren akış diyagramı Şekil 4.1’de verilmiştir. Akış diyagramında belirtilen ve programda kullanılan sınır koşulları  $0 < \alpha_1 < \alpha_2 < \alpha_3 < \pi/2$  olarak belirlenmiştir.



Şekil 4.1.a: Akış Diyagramı



**Şekil 4.1.b:** Akış Diyagramı



**Şekil 4.1.c:** Akış Diyagramı

Birey miktarı ve generasyon sayısı programın ilk koşturulması sırasında belirlenir. Bu aşamadan sonra kodlama türü seçilir. İkili kodlama seçilmiş ise ikili düzende başlangıç nüfusu oluşturulup bu nüfus reel sisteme çevrilir ve sınır koşulları kontrol



edilir. İkili kodlama seçilmemiş ise başlangıç nüfusu oluşturulur ve sınır koşulları kontrol edilir.

Sınır koşulları sağlanıyor ise amaç fonksiyonu, objektif fonksiyon, uygunluk değerinin sıralanması, en uygun bireyin saklanması, seçim işleminin yapılması, ebeveynlerin çaprazlanması, yavruların mutasyona uğratılması, en uygun değerın sıralanması, uygun olmayan bireyin atılması ve yeni nüfusun oluşturulması işlemleri yapılıp, generasyon sayısı bir arttırılır. Optimizasyon kriteri sağlanana kadar tekrar sınır koşulları testi aşamasına dönülür. Sınır koşulları sağlanmıyor ise başlangıç nüfusuna ceza eklenip diğer işlemler sırayla uygulanır. Generasyon sayısına ulaşıldığı durumda en uygun birey seçilir ve sonuç saklanarak program sonlandırılır. Programa ait kodlama, başlangıç nüfusu, amaç fonksiyonu, uygunluk değeri, seçim, çaprazlama, mutasyon, elitizm ve yeni nüfusun oluşturulması aşamaları ayrıntılı bir şekilde anlatılacaktır.

#### **4.2.1 Kodlama**

Program koşturulmadan önce başlangıç nüfus miktarı, generasyon sayısı ve mutasyon oranının belirlenmesinin ardından problem için belirlenecek çözüm kümesinin de yazılması gerekmektedir. Matlab’de yazılan programda gerçel sayı kodlaması tercih edilmiştir. Bunun nedeni, açılış değerlerini önce ikili düzende belirledikten sonra gerçel sayılara çevirip elde edilen değerleri objektif fonksiyona uygulamak ve tekrar ikili düzene dönüştürmenin getirdiği güçlülüdür. Ayrıca noktalı açılış değerlerinin ikili düzene dönüştürülmesi aşamasında zorluklarla karşılaşmaktadır. Örneğin; 10 bitlik ikilik düzende bireyler seçilir ise 1024 farklı olasılıklı bireyler oluşturulabilecektir. Oluşturulan bireylerin rasyonel sayı sisteminde seçilme aralığı 0–1.57 olduğundan dolayı ikilik sistemde bir bitlik değişim 0.00153 değerine denk gelecektir. Bu durum rasyonel sayı sistemine göre hassasiyeti azaltmaktadır. Bu tür nedenlerden dolayı rasyonel sayı sisteminde birey oluşturulması tercih edilmiştir.

Kodlamanın ikili sistemde yapılması durumunda başlangıç nüfusunun oluşturulması için yazılması gereken kodlar aşağıda verilmiştir.

```
birey=[10 10];           %gen uzunlukları  
y=sum(birey);           %birey uzunluğu
```

```

x=50; %birey sayısı
cift=x/2; %seçilecek çift sayısı
generasa=300; %maksimum generasyon
mutasyon_rate=0.001; %mutasyon oranı.
%Tablolar

minmax=[0 1.578;
        0 1.578];

%BAŞLANGIÇ NÜFUSU

[Y Z]=size(birey);
k=0;
for i=1:Z,
    t=birey(i);
    for j=1:t,
        k=k+1;
        mask(k,i)=power(2,t-j);
    end
end

nufus_bin= round(rand(x,y)); %rasgele üretilen ikili nüfus

```

Yukarıdaki programda birey sayısı 50, generasyon sayısı 300 ve 10 bitlik ikili sistemde kodlama yapılmış olup belirtilen sınırlar içerisinde başlangıç nüfusu oluşturulmuştur.

#### 4.2.2 Başlangıç Nüfusu

Başlangıç nüfusu; genetik algoritmaların çözüm için ihtiyaç duyduğu başlangıç çözüm kümesini ifade etmektedir. Başlangıç nüfus miktarı program koşturulmadan önce “nu\_birey” isimli değişkenin değerine bağlı olarak oluşturulmaktadır. Başlangıç nüfus miktarı sonuca ulaşılmasındaki önemli etkenlerden biri olmasından dolayı, uygun değer seçilmesi önemlidir. Nüfusun çok büyük oluşturulması çözümü yavaşlatacağı için tercih edilmezken, en uygun açığı problemi gibi geniş çözüm kümeli problemlerde küçük nüfus çözümün bulunmasını engellemektedir. Denemelerin ilk aşamasında; “nu\_birey” değeri 50 seçilmiştir. Daha sonra bu değer artırılarak (1000) harmonik değerlerin sıfıra daha yakın olduğu görülüp, birey sayısının yüksek

olmasının sonuca pozitif yönde etki yaptığına karar verilmiştir. Birey sayısı daha da arttırılarak 5000 ve 10000 denenmiştir. Ancak bu durumlarda birey sayısı artmasına rağmen sonucun daha iyi olmadığı ve program süresinin çok daha uzun olduğu görülmüştür. 1000 birey ile Pentium IV-3.0 işlemciye ve 512 MByte RAM belleğe sahip kişisel bir bilgisayarda Matlab programı çalıştırılmıştır, nüfustaki tüm bireylerin program tarafından değerlendirilmesi 3 saat sürmüştür. 10000 bireylik bir programda ise aynı koşullarda 23 saat 20 dakika sürmektedir.

Kodlamanın reel sistemde yapılması durumunda başlangıç nüfusunun oluşturulması için yazılması gereken kodlar aşağıda verilmiştir.

```
nu_birey=1000; %birey sayısı
nu_cift=nu_birey/2; %seçilecek çift sayısı
gener=5000; %maksimum generasyon
mutasyon_rate=1; %mutasyon oranı.
%Tablolar

minmax=[0 1.5708;
        0 1.5708;
        0 1.5708];
[x y]=size(minmax);
n=[1 3 5]; %başlangıç nüfusu

nufus= (rand(nu_birey,x))*1.5708;
```

Yukarıdaki programda birey sayısı 1000, generasyon sayısı 5000 ve reel sayı sisteminde kodlama yapılmış olup belirtilen sınırlar içerisinde başlangıç nüfusu oluşturulmuştur.

### 4.2.3 Amaç Fonksiyonu

Programdaki hedef çözümün belirtildiği kısımdır. Bu nedenle, etkin sonuç alabilmek için bu fonksiyonun düzgün bir şekilde kodlanması önem taşımaktadır. Optimum açılış probleminde amaç fonksiyonu karmaşık bir yapıya sahip olduğundan ve programda birden fazla yerde çağırıldığından program yapısının sadeleşmesi için f\_obj\_func adı altındaki bir alt programa yazılmıştır. Bölüm 3’de tespit edilen denklem 3.32’de ki  $a_n$  denklemi amaç fonksiyonu olarak yazılır. Amaç fonksiyon içerisinde başlangıç

nüfusu içerisindeki çözüm kümesinin  $0 < \alpha_1 < \alpha_2 < \alpha_3 < \frac{\pi}{2}$  koşuluna uyup uymadığına bakılarak ceza katsayı değerinin objektif fonksiyona eklenip eklenmeyeceğine öncelikli olarak karar verilmektedir. Bu sayede her yeni nüfus oluşturma aşamasında elde edilen açıların çözüm kümesinin dışında kalma olasılığı ortadan kaldırılmış olur.  $a_n$  denklemindeki her bir harmonik değerleri için bağıl hata miktarları hesaplanıp mutlak değerleri toplanarak sistemin hedef noktasından toplam uzaklık miktarı genetik algoritmanın diğer bölümlerinde kullanılmak üzere bu alt programdan geri döndürüldü. Amaç fonksiyonundaki alt bileşenlerin hesaplanması ilk önce döngü içerisinde yapılmıştır. Fakat programın hantallaştığı görülüp bu alt işlemler ayırık satırlar halinde yazılmıştır. Amaç fonksiyonunda değişkenlerin seçim aralığı  $0 - \frac{\pi}{4}$  arasında seçilip denemeler yapılmıştır. Bu aralıkta çözüm kümesi küçüldüğü için çözüme ulaşmanın daha kolay olacağı düşünülmüştür. Ancak 1. harmoniği en büyük, elenecek harmonikleri ise sıfır yapacak çözümlerin olasılığı azalmıştır. Örneğin; 3 harmonik bileşenin hesaplanması için yazılan ayırık denklemler denklem 4.1-3'de verilmiştir. Burada her bir yarı periyottaki açı sayısına göre m değeri yazılabilir.

$$V_1 = \frac{4.V_{dc}}{\pi} [1 - 2.\cos(\alpha_1) + 2.\cos(\alpha_2) - 2.\cos(\alpha_3) + \dots \mp 2.\cos(\alpha_m)] = 1 \quad (4.1)$$

$$V_3 = [1 - 2.\cos(3.\alpha_1) + 2.\cos(3.\alpha_2) - 2.\cos(3.\alpha_3) + \dots \mp 2.\cos(3.\alpha_m)] = 0 \quad (4.2)$$

$$V_5 = [1 - 2.\cos(5.\alpha_1) + 2.\cos(5.\alpha_2) - 2.\cos(5.\alpha_3) + \dots \mp 2.\cos(5.\alpha_m)] = 0 \quad (4.3)$$

Örneğin; 5 harmonik bileşenin hesaplanması için yazılan ayırık denklemler denklem 4.4-8'de verilmiştir.

$$V_1 = \frac{4.V_{dc}}{\pi} [1 - 2.\cos(\alpha_1) + 2.\cos(\alpha_2) - 2.\cos(\alpha_3) + \dots \mp 2.\cos(\alpha_m)] = 1 \quad (4.4)$$

$$V_3 = [1 - 2.\cos(3.\alpha_1) + 2.\cos(3.\alpha_2) - 2.\cos(3.\alpha_3) + \dots \mp 2.\cos(3.\alpha_m)] = 0 \quad (4.5)$$

$$V_5 = [1 - 2.\cos(5.\alpha_1) + 2.\cos(5.\alpha_2) - 2.\cos(5.\alpha_3) + \dots \mp 2.\cos(5.\alpha_m)] = 0 \quad (4.6)$$

$$V_7 = [1 - 2.\cos(7.\alpha_1) + 2.\cos(7.\alpha_2) - 2.\cos(7.\alpha_3) + \dots \mp 2.\cos(7.\alpha_m)] = 0 \quad (4.7)$$

$$V_9 = [1 - 2 \cdot \cos(9 \cdot \alpha_1) + 2 \cdot \cos(9 \cdot \alpha_2) - 2 \cdot \cos(9 \cdot \alpha_3) + \dots \mp 2 \cdot \cos(9 \cdot \alpha_m)] = 0 \quad (4.8)$$

Objektif fonksiyon=(1-1.harmoniğin mutlak değeri)+elenmesi istenen diğer harmoniklerin mutlak değerlerinin toplamını sıfıra en yakın yapacak sayıdır. Bu bağlamda objektif fonksiyon 3 harmonik için programda denklem 4.9'daki gibi yazılmıştır.

$$OBJ1(i,1)=abs(1-obj(i,1))+abs(obj(i,2))+abs(obj(i,3)); \quad (4.9)$$

Birinci harmonik bileşeni için uygunluk değeri hesabında bu harmonik değerinin bir sayısına eşit olması istendiğinden dolayı ayrık denklemden elde edilen sonuç 1 sayısından çıkartılıp uzaklık miktarı hesaplanır. Diğer harmonik bileşenler için uygunluk değeri hesabında bu harmonik değerlerinin sıfıra eşit olması istendiğinden dolayı ayrık denklemden elde edilen sonuçların doğrudan mutlak değeri hesaplanıp hata miktarı belirlenir.

Kodlamanın reel sistemde yapılması durumunda amaç fonksiyonunun oluşturulması ve objektif fonksiyonun oluşturulması için yazılması gereken kodlar aşağıda verilmiştir.

```
function [OBJ]=f_obj_func(nufus,nu_birey,n)
for i=1:nu_birey,
    kosul(i,1)=1000;
    alfa=nufus(i,:);
    obj(i,1)=1.274*(1-(2*cos(n(1,1).*alfa(1,1)))+(2*cos(n(1,1).*alfa(1,2)))-
(2*cos(n(1,1).*alfa(1,3))));
    obj(i,2)=1-(2*cos(n(1,2).*alfa(1,1)))+(2*cos(n(1,2).*alfa(1,2)))-
(2*cos(n(1,2).*alfa(1,3))));
    obj(i,3)=1-(2*cos(n(1,3).*alfa(1,1)))+(2*cos(n(1,3).*alfa(1,2)))-
(2*cos(n(1,3).*alfa(1,3))));
    OBJ1(i,1)=abs(1-obj(i,1))+ abs(obj(i,2))+abs(obj(i,3));
    if alfa(1,3)<1.5709
        if alfa(1,2)<alfa(1,3)
            if alfa(1,1)<alfa(1,2)
                if alfa(1,1)>0
                    kosul(i,1)=0;
```

```
end
end
end
end
OBJ(i,1)=OBJ1(i,1)+kosul(i,1);
end;
```

#### 4.2.4 Uygunluk Değeri

Objektif fonksiyondan elde edilen toplam hata miktarlarına bağlı olarak her bir bireyin çözüm için ne kadar iyi olduğuna karar verilir. Karar verme işlemi sıra tabanlı uygunluk atama yöntemine göre yapılır. Hata miktarları küçükten büyüğe doğru sıralanır. En küçük değer en uygun çözümü temsil etmektedir. Optimum açılı probleminde en küçük hata miktarı birinci harmoniği bir, elenecek diğer harmonikleri ise sıfır yapacak açılı değerleri ile elde edilir. Örneğin; amaç fonksiyonunda elde edilen sonuçların 2, 5, 0.5, 4, 0.2, 0.8 olduğunu varsayarsak uygunluk değeri hesabı alt programı sonucunda 0.2, 0.5, 0.8, 2, 4, 5 olarak sıralanmaktadır. Burada en uygun sonuç 0.2 olarak görülmektedir ve bu nüfusta yeni nesle aktarılacak en elit birey olarak seçilmektedir.

Kodlamanın reel sistemde yapılması durumunda uygunluk değerinin bulunması için yazılması gereken kodlar aşağıda verilmiştir.

```
function [fitness]=f_fitness(obj_func)
[nu_birey x]=size(obj_func);
[Y Z]=sort(obj_func);
[K L]=sort(Y);
fitness(Z,1)=(nu_birey+1-L)./nu_birey;
```

#### 4.2.5 Seçim

Amaç fonksiyon ve uygunluk değerleri alt programlarında elde edilen sonuçlara bağlı olarak rulet çarkı seçim yöntemi kullanılarak bir sonraki neslin oluşturulması için gerekli işlemler yapılmaktadır. Uygunluk değeri belirleme aşamasında en yüksek uygunluğa sahip olan bireyin seçilme olasılığı rulet çarkı seçim yönteminde yüksektir. En uygun bireyin seçim çarkında yüksek paya sahip olması bu bireyin bir

sonraki nesil için seçilme olasılığını yükseltecektir. Bu yaklaşım tarzı genetik algoritmalarındaki iyi ebeveynlerden iyi çocuklar oluşur temeline dayanmaktadır.

Kodlamanın reel sistemde yapılması durumunda ebeveynlerin seçim işlemi için yazılması gereken kodlar aşağıda verilmiştir.

```
function [secim]=f_secim(fitness)
[nu_birey x]=size(fitness);
rullet=sum(fitness)*rand(nu_birey,1);
for j=1:nu_birey,
dilim=0;i=0;
while dilim<rullet(j,1),
i=i+1;
dilim=dilim+fitness(i,1);
end,
secim(j,1)=i;
end,
```

#### 4.2.6 Çaprazlama

Yeni bireylerin oluşturulması için eski iki birey ebeveyn olarak belirlenir. Ardından ebeveynler iki eşit parçaya bölünür. Birinci ebeveynin birinci parçası rasgele bir r sayısı ile ikinci ebeveynin ikinci parçası (1-r) sayısı ile çarpılarak ve elde edilen parçacıklar yan yana yazılarak yeni çocuk birey üretilir. Aynı işlem birinci ebeveynin ikinci parçası ve ikinci ebeveynin birinci parçasına da uygulanır. Sonuç olarak ilk nesille aynı birey sayısına sahip yeni çocuk bireyler üretilir. Çaprazlama için kullanılacak ebeveynler rasgele belirlenir. Rasgele belirlenme işlemi, yeni nesillerin oluşmasını sağlayıp program içerisinde yeni çözümler oluşturmaktadır.

Kodlamanın reel sistemde yapılması durumunda ebeveynlerin çaprazlanması için yazılması gereken kodlar aşağıda verilmiştir.

```
function [cr_nufus]=f_cross(ebeveyn1,ebeveyn2)
[nu_cift x]=size(ebeveyn1);
nokta=ceil(rand(nu_cift,1)*(x-1));
cocuk1=zeros(nu_cift,x);
cocuk2=zeros(nu_cift,x);
```

```

for k=1:nu_cift,
r=rand(1,1);
cocuk1(k,1:nokta)=(ebeveyn1(k,1:nokta)*r)+(ebeveyn2(k,1:nokta)*(1-r));
cocuk2(k,1:nokta)=(ebeveyn1(k,1:nokta)*(1-r)+(ebeveyn2(k,1:nokta)*r);
cocuk1(k,nokta+1:x)=(ebeveyn1(k,nokta+1:x)*r)+(ebeveyn2(k,nokta+1:x)*(1-r));
cocuk2(k,nokta+1:x)=(ebeveyn1(k,nokta+1:x)*(1-r)+(ebeveyn2(k,nokta+1:x)*r);
cr_nufus=cat(1,cocuk1,cocuk2);
end;

```

#### 4.2.7 Mutasyon

Nesillerin çaprazlama sırasında bireylerin belirli oranlarda değişime uğraması sağlanarak daha uygun çözüm kümelerinin bulunması için bireyler mutasyona uğrattılır. Bireylerin mutasyona uğrama olasılığı hedef çözüme ulaşma aşamasında önemli bir ölçüttür. Mutasyon başlangıçta iyi bir nüfusun oluşturulamaması durumunda nüfusun çeşitliliğini arttırmak için kullanılır. Bu nedenle “mutasyon\_rate” adlı değişken değeri program koşturulmadan önce belirlenir. Örneğin; çok tepeli fonksiyonda en yüksek tepeyi bulma olayında değişkenlerin bir noktada toplanmasını önlemek için değişkenler arasında çeşitlilik sağlanmalıdır. Bu işlemin gerçekleşmesi için, bireyler belirli oranlarda değişime uğrattılır. İkili sistemde bireylerin değişime uğrattılması en düşük ağırlıktaki bitin değişimiyle olmaktadır. Denemelerde öncelikle değişim oranı 0.001 olarak seçilmiştir. Bu işlem sonucunda değişkenlerin bir noktada toplandığı görülmüştür. Programın daha etkin çalışabilmesi için değişim oranı 1’e kadar arttırılarak çeşitliliğin sağlandığı gözlenmiştir. Değişim oranı 1 yapılarak her yeni nüfus oluşturulmasında mutasyon gerçekleştirilerek çeşitlilik arttırılmıştır. Her nüfusun en elit bireyi doğrudan yeni nüfusa alındığı için mutasyonun her nüfusa uygulanmasının çözüme olumsuz bir etkisi olmamaktadır.

Kodlamanın reel sistemde yapılması durumunda mutasyon aşaması için yazılması gereken kodlar aşağıda verilmiştir.

```

function [mu_nufus]=f_mutasyon(cr_nufus,mutasyon_rate)
[nu_birey x]=size(cr_nufus);
mo=(rand(nu_birey,x)-0.5)*2;
mo_a=zeros(nu_birey,x);
for i=1:nu_birey,

```



```
for j=1:x,  
if mo(i,j)<mutasyon_rate  
mo_a(i,j)=1;  
end,  
end;  
end;  
mu_nufus=cr_nufus+(mo.*mo_a);
```

#### **4.2.8 Elitizm**

Bireyler uygunluk değerlerine göre sıralandıktan sonra en uygun birey ya da bireyler optimizasyon aşamalarına katılarak çözümden uzaklaşmalarını önlemek amacıyla sonraki nüfusa doğrudan alınırlar.

Kodlamanın reel sistemde yapılması durumunda mutasyon aşaması için yazılması gereken kodlar aşağıda verilmiştir.

```
[s1 s2]=max(fitness);  
elitist(1,:)=nufus(s2,:);
```

#### **4.2.9 Yeni Nüfusun Oluşturulması**

Seçim, çaprazlama ve mutasyon işlemlerine tabi tutulan bireyler sonucunda yeni çocuk bireylerden oluşan eski nüfus ile aynı miktarda yeni nüfus elde edilir. Yeni nüfus oluşturma işlemi optimizasyon kriteri gerçekleşene kadar devam eder. Optimum açı problemi için uygun optimizasyon kriteri başlangıçta belirlenen “gener” adlı değişkenle ya da objektif fonksiyonun belli bir değerden küçük olması olarak belirlenebilir.

İlk generasyonun oluşturulması durumunda elde edilen nüfus  $nufus=mu\_nufus$  eşitliği ile yeni nüfus olarak atanır. Başlangıçta belirlenen generasyon sayısına ulaşılan kadar her defasında generasyon miktarı bir artırılarak yeni nüfus oluşturma işlemi bu şekilde yapılır.

#### **4.3 Program Sonuçları**

Üç açı ve üç harmonik için başlangıç birey sayısı 200, oluşturulacak yeni nesil sayısı 2000 ve mutasyon oranı 1 seçildiğinde ve program 10 kez çalıştırıldığında tablo

4.1’de ki değerler elde edilmiştir. Tablo 4.1’de ki ilk üç sütun sırası ile birinci, üçüncü ve beşinci harmonik bileşenlerinin genliklerini ifade etmektedir. Tablodaki dört, beş ve altıncı sütunlar ise anahtarlama için kullanılması gereken açıların radyan cinsinden değerlerini göstermektedir. Yedinci ve son sütun ise o satırdaki değerlere karşılık gelen toplam harmonik dağılımını göstermektedir.

**Tablo 4.1:** 3 Açılı ve 3 Harmonik için Genetik Algoritma ile Bulunan Değerler

1. Harmonik	3. Harmonik	5. Harmonik	1. Açılı	2. Açılı	3. Açılı	THD
1,0261	0,0143	-0,0032	0,4288	0,6054	1,5607	0,0143
1,0164	-0,0065	-0,0396	0,4265	0,6080	1,5593	0,0395
1,0566	0,0058	0,0259	0,4212	0,5917	1,5681	0,0251
1,0224	-0,0787	0,0031	0,4251	0,6131	1,5651	0,0770
1,0408	-0,0068	-0,0081	0,4220	0,5978	1,5650	0,0102
0,969	0,0443	0,0001	0,4478	0,6338	1,5467	0,0457
1,0218	0,0115	0,0011	0,4306	0,6083	1,5599	0,0113
1,0068	0,0333	-0,0413	0,4317	0,6108	1,5550	0,0527
0,9825	0,0255	-0,0617	0,4361	0,6218	1,5499	0,0680
0,9554	-0,0109	0,0001	0,4470	0,6426	1,5469	0,0114

Klasik optimizasyon yöntemi ile elde edilen sonuçlara göre birinci, üçüncü ve beşinci harmonik bileşenlerin genlikleri şöyledir:

1. harmonik 1.0952

3. harmonik 0.2389

5. harmonik -0.0323

Elde edilen genlik değerleri için anahtarlamanın 0.4023 radyan, 0.5397 radyan ve 1.5629 radyandaki açılı değerlerinde yapılması gerekmektedir. Klasik optimizasyon sonucunda bulunan değerlere göre toplam harmonik dağılım 0.22 olarak hesaplanır.

Tablo 4.1’de de görüldüğü gibi genetik algoritma ile elde edilen sonuçlar klasik optimizasyon ile toplam harmonik dağılım değerlerine göre karşılaştırıldığında genetik algoritma ile elde edilen sonuçların %100 başarılı olduğu görülmektedir. Ayrıca klasik optimizasyon yönteminde elde edilen 3. harmonik değeri genetik algoritma ile elde edilen değerlerden yüksek olmasının yanı sıra reel bir sistem içinde yüksek bir değerdir.

Beş açı ve beş harmonik için başlangıç birey sayısı 500, oluşturulacak yeni nesil sayısı 100000 ve mutasyon oranı 1 seçildiğinde ve program koşturulduğunda tablo 4.2'nin ilk satırındaki değerler elde edilir. Oluşturulacak yeni nesil sayısı 500000 olarak artırıldığında ise tablo 4.2'nin ikinci ve üçüncü satırındaki değerler elde edilmiştir. Nesil sayısının çözümde bir iyileşme sağladığı görülmektedir.

**Tablo 4.2:** 5 Açı 5 Harmonik Probleminin Genetik Algoritma ile Çözümü

HARMONİKLER					
1.	3.	5.	7.	9.	THD
1,3890	0,0715	-0,4192	0,0256	0,0366	0,307842
1,4227	0,1772	-0,2751	-0,0680	-0,1557	0,259161
1,0584	0,0104	0,2625	-0,0599	-0,1511	0,291878
AÇILAR					
1.	2.	3.	4.	5.	
0,0151	0,2078	0,4889	0,5839	1,5686	
0,1332	0,2654	0,5395	0,6041	1,5702	
0,2977	0,4683	0,8828	0,9900	1,5469	

Tablo 4.2'deki harmonikler başlığı altındaki beş sütun sırası ile birinci, üçüncü, beşinci, yedinci ve dokuzuncu harmonik bileşenlerinin genliklerini ifade etmektedir. Tablodaki açılar başlığı altındaki beş sütun ise anahtarlama için kullanılması gereken açıların radyan cinsinden değerlerini sırası ile göstermektedir. Son sütun ise o satırdaki değerlere karşılık gelen toplam harmonik dağılımını göstermektedir.

Tablo 4.2'de görüldüğü gibi; 5 açı ve 5 harmonik problemini genetik algoritma temellerine dayanarak Matlab'de yazılan programdan toplam harmonik değeri ele alınarak değerlendirme yapılırsa iyi bir sonuç elde edilememiştir. Toplam harmonik değerleri oldukça yüksektir. Ancak problem herhangi bir harmoniği elimine etmek için uygun olabilir. Tablo 4.2'nin birinci satırında 5. harmonik dışındaki harmonikler sifıra yakın çıkmıştır, ikinci satırda yedinci harmonik sifıra yakın çıkmıştır ve son satırda 3. ve 7. harmonikler sifıra yakın çıkmıştır. Diğer bir deyişle bu harmoniklerin elimine edilmiş oldukları söylenebilir.

## 5. SONUÇLAR VE TARTIŞMA

Tezde genetik algoritmalar ve eviriciler hakkında ayrıntılı bilgi verilmiştir. Bütün güç elektroniği devrelerinin tasarımındaki problemlerden biri olan harmonikler üzerinde durulmuştur ve eviricilerin harmonik analizi anlatılmıştır. İki kutuplu anahtarlamaya sahip darbe genişlik modülasyonlu eviricinin fourier analizi denklemleri çıkarılarak genetik algoritmalar için kullanılacak amaç fonksiyonu elde edilmiştir.

Son bölümde ise genetik algoritma mantığı ile Matlab'de yazılan programın detayları anlatılmıştır.

Matlab'de yazılan programın koşturulması sırasında program parametreleri ile ilgili bazı değişimler yapılmıştır. Öncelikle program üç açılı problemi için koşturulmuştur. Başlangıç denemelerinde birey sayısı 50, mutasyon oranı 0.001, generasyon sayısı 200 olarak belirlenmiştir. Bu koşullarda programın 10 kez çalıştırılması sonucunda elde edilen açılı değerlerine göre 3. ve 5. harmonik değerlerinin birden bile büyük çıktığı görülmüştür. Bu durumda öncelikle generasyon sayısı artırılarak 2000'e çıkarılmıştır. Bunun sonucunda objektif fonksiyon değeri küçülmesine rağmen harmonik değerleri hala büyük olduğundan programdaki birey sayısı artırılmıştır. Birey sayısı 100'e çıkarılıp program 10 kez çalıştırıldığı zaman 3. ve 5. harmonik bileşen değerleri %30 oranında 0.2'nin altına indiği görülmüştür. Bu bağlamda programın iyileştiği görülüp birey sayısı 200'e çıkarılıp program tekrar 10 kez çalıştırılmıştır. Sonuçta elde edilen 3. ve 5. harmonik değerleri %100 oranında 0.2'nin altına inmiştir. Bu noktadan itibaren program parametrelerinin uygun olduğu kabul edilmiştir.

Program 5 açılı ve 5 harmonikli denklem çözümü için amaç fonksiyonu düzenlenmiş, objektif fonksiyon değiştirilmiş ve sınırlar gibi diğer parametreler düzenlenmiştir. Parametrelerin ilk değerleri: birey sayısı 50, mutasyon oranı 0.1 ve generasyon sayısı 5000'dir. Bu koşullarda program 10 kez çalıştırıldığında harmonik

değerleri bütün sonuçlarda 1'den bile büyük çıkmıştır. Programın iyileştirilmesi amacıyla birey sayısı sırası ile 500, 1000, 2000, 5000 ve 10000 yapılmıştır. Çalışmalar sonucunda 3, 5 ve 7. harmonik değerlerinin 5000 ve 10000 birey sayısında 1'den küçük çıktığı görülmüştür. Ardından generasyon sayısı 20000 yapılmıştır. Program çalıştırıldığında objektif fonksiyon değerinin 4'ün altına inmediği görülmüştür. Buna bağlı olarak programın tıkanıp karar verilep bireylerin çeşitliliğini sağlamak için mutasyon oranı artırılarak 0.8 yapılmıştır. Bu durumda objektif fonksiyon değeri 4'ün altına inmiştir ve bireylerin çeşitliliğinin programda olumlu etki yaptığına karar verilmiştir. Bu yaklaşıma bağlı olarak mutasyon oranı 1 yapıp programın her yeni nüfus oluşturma aşamasında bireylerin mutasyona uğraması sağlanmıştır. Bu iyileşmeden sonra generasyon sayısı sırası ile 50000 ve 100000 yapıp program üç kez çalıştırılmıştır. Elde edilen sonuçlara göre harmonik bileşen değerleri 0.4'ün altına inmiştir. Bütün bu çalışmaların sonucunda genetik algoritmalar ile her zaman en uygun açılı problemi için uygun sonuçlar bulunamadığı görülmüştür. Çözümün bulunması gereken harmonik sayısı arttığı zaman zorlaşmasının nedeni öncelikle çözüm kümesinin çok büyük ve açılıların sıralı olması gerektiği gibi bir durumun söz konusu olmasıdır. çözüm kümesinin genişliğinden dolayı yavaş bir yöntem olduğu söylenebilir. Ancak diğer optimizasyon yöntemleri ile optimum üç açılı bulmanın bile zaman aldığı ve bazen sonuca tam olarak ulaşamadığı düşünülürse genetik algoritmalar çok önemli bir üstünlük sağlamış olur.

Program oluşturulurken yapılan denemelerde program yapısının doğru bir şekilde genetik algoritma mantığına göre geliştirilmesinin yanı sıra objektif fonksiyonun uygun bir biçimde tanımlanması, başlangıç nüfus miktarı, mutasyon oranı ve optimizasyon kriteri gibi kavramların belirlenmesinin önemi ortaya çıkmıştır.

Başlangıç nüfus miktarının yüksek sayıda seçilmesinin çeşitliliği artırması sebebiyle olumlu etki sağladığı görülmüştür. Başlangıç nüfus miktarını çok büyük seçmek ise çözümden uzaklaşma gibi bir sorun ortaya çıkarmaktadır. Denemelerden başlangıç nüfusu konusunda elde edilen sonuç; başlangıç nüfusundaki birey sayısının bilinmeyen sayısı arttığı durumda (5 açılı gibi) 500-600 olması daha az açılı değerleri için problem çözülecek ise de 200 olmasının uygun olduğudur.

Mutasyon oranının belirlenmesinde; bölüm 2’de de anlatıldığı gibi çok büyük seçilmesi her yeni nesil oluşturulması aşamasında mutasyon gerçekleşeceği ve neslin bir öncekinden çok daha farklı olacağı anlamına gelmektedir. Ancak optimum açılı probleminde her yeni nesil oluşturma aşamasında mutasyona girmesi tercih edilmiştir. Mutasyon oranının büyük seçilmesi ile yeni nesil oluştururken çeşitlilik sağlanması ön planda tutulmuştur. Her nüfusun en üstün bireyi saklandığı için mutasyon oranının büyük seçilmesi bir olumsuzluk getirmemiştir.

Generasyon sayısı ise objektif fonksiyonun küçülmesi için tanınan bir zamandır. Büyük tutulması sayesinde her yeni nüfus oluşturulmasında objektif fonksiyonun küçülme olasılığı vardır. Ancak belli bir noktadan sonra sınır koşullarının çözümü zorlaştırması nedeniyle objektif fonksiyonun küçülmesi zorlaşmaktadır. Bu nedenle programın yukarıdaki örneklerde belirlenen generasyon değerlerinde sonlandırılması uygun görülmüştür.

Programın nasıl ve ne zaman sonlanacağını gösteren optimizasyon kriteri optimum açılı problemi için belirli sayıda yeni nesil oluşturulması ya da belirlenen bir sayıdan daha küçük değerlerde objektif fonksiyon değerini bulmak olabilir. Beş açılı ve beş harmonikten büyük değerler için yapılan denemelerde objektif fonksiyonun küçük değerlere ulaşmadığı ya da çok uzun süre aldığı görüldüğünden, programı sonlandırmak için “gener” adlı değişken ile yeni nesil sayısı belirlenmiştir.

Bölüm 4’de ki sonuçlardan da görüldüğü üzere üç açılı ve üç harmonik probleminde çok kısa sürede klasik optimizasyondan çok daha kabul edilebilir sonuçlar ortaya çıkarmıştır. Bu karşılaştırma toplam harmonik değerlerine göre yapılarak başarı kanıtlanmıştır.

Beş açılı beş harmonik durumunda ise çözüme ulaşılması hem zaman almış hem de sonuç olumsuz olmuştur. Aynı problemin elle çözümünün zorluğu düşünüldüğünde, genetik algoritmanın çözüme ulaşması kabul edilebilir. Ancak değerlerin uygun olmadığına dikkat edilmelidir. Toplam harmonik dağılım değeri 5 açılı 5 harmonik problemi için yüksek çıkmıştır.

Optimum açılı probleminde genetik algoritmalar ile istenilen harmoniklerin elenmesinin klasik optimizasyon yöntemlerinden üstün olarak mümkün olduğu sadece 3 açılı gibi bilinmeyen az olduğu durumlarda söz konusu olabilir. Daha fazla

bilinmeyen içeren optimum açılı probleminin çözümünün sonuca ulaşabilmesi zaman almaktadır. Algoritmanın verimli bir şekilde çalışabilmesi için probleme uygun yapılandırılması gerekmektedir. Aksi takdirde; verim düşecektir. Ayrıca bu algoritma kullanılarak güç kaynaklarının performans artırımı için gerekli anahtarlama şekilleri oluşturabilir (örneğin güç faktörü düzeltimi).

Tablolarda kullanılan örnek sayısı 3 açılı ve 3 harmonik problemin 10 iken 5 açılı 5 harmonik problemde üçtür. Bunun en önemli nedeni 5 açılı ve 5 harmonik probleminin çözümünün yaklaşık 23 saat sürmesidir.

Pratik dünyada optimum açılı probleminden elde edilen sonuçların kullanılabilmesi için anahtarlama frekansının kilo hertz mertebesinde olması gerekmektedir. Eviricilerin harmonik eliminasyonunda optimum anahtarlama açılarının genetik algoritmalar ile elde edilmesi durumunda da açılı sayısının frekansın kilo hertz mertebesinde olması için fazla olması gerekmektedir. Günümüzdeki bir çok anahtarlama güç kaynağının çalışma frekansları istenen performansa bağlı olarak 20 ile 100kHz arasında değişmektedir. Örneğin 20 kHz anahtarlama frekansına sahip bir eviricide çıkış frekansını 50 Hz olması durumunda anahtarlama açılarının genetik algoritmalar ile hesaplanmasını istersek eğer  $0 - \frac{\pi}{4}$  arasındaki periyoda

hesaplanması gereken açılı değeri :  $\frac{T_{cikis\_frekansı}}{T_{anahtarlama\_frekansı} * 4}$  kadar olacaktır.

Eğer periyot değerlerini yerine koyarsak her bir periyottaki darbe sayısı 100 olacaktır. 5 açılı bir denkleme çözüme ulaşmak için gerekli olan birey sayısı ve generasyon sayısı değerlerini göz önünde bulundurursak: Denklemin 100 açılı hesabında çözümü için çok daha yüksek oranlarda programın koşturulmasına gerek duyulacaktır. 5 açılı hesabında programın Pentium IV 3.0 ve 512 MByte'lık kişisel bir bilgisayarda bir kez koşturulması 23 saat sürmüştür. Bu yaklaşım tarzına bağlı olarak 100 açılı hesabı için gereken süre çok daha uzun olacaktır. Programın 20 kHz'lik anahtarlama frekansını için 100 açılı hesapladığımı kabul edelim. Bu kez elde edilecek açılı değerlerinin pratik dünyada uygulanması ile ilgili bir sorun karşımıza çıkacaktır. Matlab programı noktadan itibaren dört basamaklık bir hassasiyetle açılı değerlerini hesaplayabilmektedir. Bu açılı değerlerinde herhangi bir anahtarlama elemanını sürmeye kalkıştığımızda ise bazı güçlükler ile karşılaşırız. Güçlüklerden en önemlisi; anahtarlama elemanlarının iletme ve kesime gitme anlarında yaşanan gecikmeler

nedeni ile istenen açı değerlerinde sistemi kontrol etmemizi engellemesidir. Ayrıca bu açı değerlerinin pratik dünyada üretimi için çok yüksek hızda ve çözünürlükte darbe genişlik modülasyonu yapabilen sayısal işaret işleyicileri kullanılmalıdır. Sayısal işaret işleyicilerinde 100 açılı darbe genişlik modülasyonu için bakış tabloları oluşturulmaktadır. Bu tablolara elde ettiğimiz açı değerlerinin programdaki karşılıkları yazılarak modülasyon sağlanabilir. Sistemin pratik dünyada uygulanabilmesi ile ilgili bir diğer sorun ise  $0 - \frac{\pi}{4}$  aralığında açı sayısı yükseldikçe aralıklarının küçülmesi ve sistemin parçalı doğrusal bir hale gelmesi söz konusudur bu noktadan itibaren bir parçayı basit bir şekilde 100 e bölme yöntemi ile de bu açı değerlerine ulaşılabilir. Günümüzdeki birçok sayısal işaret işleyicileri de bu yaklaşım tarzına bağlı olarak oluşturulan açı değerlerinde darbe genişlik modülasyonu yapmaktadırlar.



## KAYNAKLAR

- [1] **Sapmaz, M. E.**, 2004. Generatör Birim Katkı Sorununun Genetik Algoritmalar ile Çözülmesi, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [2] **Kurt, M. ve Semetay, C.**, . Genetik Algoritma ve Uygulama Alanları, *Yüksek Lisans Tezi*, M.Ü. Teknik Eğitim Fakültesi Makine Bölümü.
- [3] **Goldberg, D. E.**, 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, pp. 1-23, Addison-Wesley Publishing Company, USA.
- [4] **Sakawa, M.**, 2002. Genetic Algorithms and Fuzzy Multiobjective Optimization, pp. 11-27, Kluwer Academic Publishers, Boston.
- [5] **Man, K. F. , Tang, K. S. and Kwang, S. ,** 1999. Genetic Algorithms: Concepts and Designs, Springer, London.
- [6] **Güçlü, A.**, 2004. Yüksek Gerilim Tekniğinde Genetik Algoritma Uygulamaları, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [7] **Angelov, P. P. and Wright, J. A.**, 2000. A centre of-gravity-based recombination operator for GA, pp. 259-264, IEEE Conf. on Industrial Control, Modeling and Instrumentation, IECON-2000, Nagoya, Japan.
- [8] **Bäck, T. and Hoffmeister, F.**, 1991. Genetic Algorithms, *Extended Selection Mechanisms in Genetic Algorithms Proceedings of the 4th ICGA*, pp 92-99.
- [9] **Hutter, M.**, 2002. Fitness uniform selection to preserve genetic diversity, In Fogel, D. et al., editors, *Proceedings of the Congress on Evolutionary Computation*, pages 783-788, Honolulu, USA. IEEE Pres.
- [10] **Jiri, K. and Jiri, L.**, 2001. A New Genetic Operator Maintaining Population Diversity, In Dubois Daniel M., editor, *Computing Anticipatory Systems*, pages 338-348, American Institute of Physics, Liege, Belgium.

- [11] **Spears, W. M.**, 1992. Crossover or Mutation?, *In Proceedings of Foundations of Genetic Algorithms Workshop*, pp:221,237.
- [12] **Rosca, J. and Ballard, D.**, 1995. Causality in Genetic Programming In Eshelman, L., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp: 256-263, Pittsburgh, PA, USA.
- [13] **Thierens, D.**, 1997. Selection schemes, elitist recombination and selection intensity, *Proceedings of the Seventh International Conference on Genetic Algorithms*, pp: 152-159, San Fransisco, U.S.A.: Morgan Kaufmann.
- [14] **Artaç, T.**, 2003. Genetik Algoritma ile Dağıtım Şebekelerinin Optimum Tasarımı, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [15] [www.jaycar.com.au](http://www.jaycar.com.au), (Erişim tarihi: Aralık 2006).
- [16] **Bina, M. T. and Hamill, D. C.**, 1999. Optimizing a Discrete Switching Pattern Using Two Simulated Annealing Algorithms, University of Surrey.
- [17] **Hart, D. W.**, 1997. Introduction to Power Electronics, Upper Saddle River, N, J., Prentice Hall.
- [18] **Rao, A. and Lipo, T. A.**, 1999. A Modified Single Phase Inverter Topology with Active Common Mode Voltage Cancellation, A.L.Julian,United Technologies Research Center, University of Wisconsin-Madison, IEEE.
- [19] [www.kettering.edu/~bguru/index.html](http://www.kettering.edu/~bguru/index.html), (Erişim tarihi: Ağustos 2006).
- [20] [www.powerdesigers.com/InfoWeb/design\\_center/articles/Inverters/inverters](http://www.powerdesigers.com/InfoWeb/design_center/articles/Inverters/inverters), (Erişim tarihi: Ağustos 2006).
- [21] [www.pge.com/docs/pdfs/biz/power\\_quality\\_notes/harmonics.pdf](http://www.pge.com/docs/pdfs/biz/power_quality_notes/harmonics.pdf) (Erişim tarihi: Eylül 2006)
- [22] **Chiasson, J. , Tolbert, M. and McKenzie, K. and Du, Z.**, 2003. A Complete Solution to the Harmonic Elimination Problem, ECE Department, The University of Tennessee, Knoxville, IEEE.

- [23] **Hasmukh, S. , Patel, R. and Hoft, G.**,1974. Generalized Techniques of Harmonic Elimination and Voltage Control in Thyristor Inverters: Part II-Voltage Control Techniques, pp:666-673, IEEE Transaction on Industry Application Vol IA-10 No:5.
- [24] **Czarkowski, D. , Chudnovsky D. V. and Chudnovsky, G. V.**,2002. Solving the Optimal PWM Problem for Single Phase Inverters, pp:465-474, IEEE.
- [25] **Oliva, A. , Chicchiarini, H. and Mandolesi, P.**, 2005. Reduction of Total Harmonic Distortion in Power Inverters, pp:89, Latin American Applied Research, Instituto de Investigaciones en Ing. Electrica, Universidad Nacional del Sur, Argentina.
- [26] **Mohan, N. , Undeland, T. M. and Robbins, W. P.**, 1995. Power Electronics: Converters, pp:210-220, Applications and Design, John Wiley & Sons, Inc.
- [27] **Krein, P.T. , Geng, X. and Balog, R.**,2002. High-Frequency Link Inverter Based on Multiple-Carrier PWM, Grainger Center for Electric Machinery and Electromechanisms, Department of Electrical and Computer Engineering, University of Illinois, Urbana.

## EK A

### OPTİMUM AÇI PROBLEMİNİN 5 AÇI 5 HARMONİK MATLAB PROGRAMI

#### EK A.1

#### OBJEKTİF FONKSİYON ALT PROGRAMI

```
function [OBJ]=f_obj_func9(nufus,nu_birey,n)
for i=1:nu_birey,
    kosul(i,1)=1000;
    alfa=nufus(i,:);

    obj(i,1)=1-(2*cos(n(1,1).*alfa(1,1)))+(2*cos(n(1,1).*alfa(1,2)))-
(2*cos(n(1,1).*alfa(1,3)))+(2*cos(n(1,1).*alfa(1,4)))-(2*cos(n(1,1).*alfa(1,5)));

    obj(i,2)=1-(2*cos(n(1,2).*alfa(1,1)))+(2*cos(n(1,2).*alfa(1,2)))-
(2*cos(n(1,2).*alfa(1,3)))+(2*cos(n(1,2).*alfa(1,4)))-(2*cos(n(1,2).*alfa(1,5)));

    obj(i,3)=1-(2*cos(n(1,3).*alfa(1,1)))+(2*cos(n(1,3).*alfa(1,2)))-
(2*cos(n(1,3).*alfa(1,3)))+(2*cos(n(1,3).*alfa(1,4)))-(2*cos(n(1,3).*alfa(1,5)));

    obj(i,4)=1-(2*cos(n(1,4).*alfa(1,1)))+(2*cos(n(1,4).*alfa(1,2)))-
(2*cos(n(1,4).*alfa(1,3)))+(2*cos(n(1,4).*alfa(1,4)))-(2*cos(n(1,4).*alfa(1,5)));

    obj(i,5)=1-(2*cos(n(1,5).*alfa(1,1)))+(2*cos(n(1,5).*alfa(1,2)))-
(2*cos(n(1,5).*alfa(1,3)))+(2*cos(n(1,5).*alfa(1,4)))-(2*cos(n(1,5).*alfa(1,5)));

    OBJ1(i,1)=abs(1-obj(i,1))+ abs(obj(i,2))+abs(obj(i,3))+abs(obj(i,4))+abs(obj(i,5));

    if alfa(1,5)<1.5709
        if alfa(1,4)<alfa(1,5)
            if alfa(1,3)<alfa(1,4)
                if alfa(1,2)<alfa(1,3)
                    if alfa(1,1)<alfa(1,2)
```

```

        if alfa(1,1)>0
            kosul(i,1)=0;
            end
        end
    end
end
end

OBJ(i,1)=OBJ1(i,1)+kosul(i,1);
end;

```

## **EK A.2**

### **ANA PROGRAM**

```

clear all;
clc;
nu_birey=50; %birey sayısı
nu_cift=nu_birey/2; %seçilecek çift sayısı
gener=5000; %maximum generation
mutasyon_rate=0.1; %mutasyon oranı.

minmax=[0 1.5708;
        0 1.5708;
        0 1.5708;
        0 1.5708];

[x y]=size(minmax);

n=[1 3 5 7 9];

%initial population
nufus= (rand(nu_birey,x))*1.5708;

%objektif function
[obj_func]=f_obj_func9(nufus,nu_birey,n);

```

```

%fitness function
[fitness]=f_fitness(obj_func);
for gen=1:gener,

    %elitist
    [s1 s2]=max(fitness);
    elitist(1,:)=nufus(s2,:);

    %seçim_roulet wheel
    [secim]=f_secim(fitness);

    %mating pool
    ebeveyn1(1:nu_cift,:)=nufus(secim(1:nu_cift,1),:);
    ebeveyn2(1:nu_cift,:)=nufus(secim(nu_cift+1:2*nu_cift,1),:);

    %crossover
    [cr_nufus]=f_cross(ebeveyn1,ebeveyn2);

    %mutasyon
    [mu_nufus]=f_mutasyon(cr_nufus,mutasyon_rate);

    %yeni nüfus
    nufus=mu_nufus;
    %objektif function
    [obj_func]=f_obj_func9(nufus,nu_birey,n);
    %fitness function
    [fitness]=f_fitness(obj_func);

    %elitism
    [s1 s2]=min(fitness);
    nufus(s2,:)=elitist(1,:);
    %objektif function
    [obj_func]=f_obj_func9(nufus,nu_birey,n);
    %fitness function
    [fitness]=f_fitness(obj_func);

    %SONUÇLAR

```

```

min(obj_func)
sakla(gen,1)=min(obj_func);
end

plot(sakla);
[a,b]=min(obj_func);
minimum_obj_func=a
alfa=nufus(b,:)
THD1=1-(2*cos(1.*alfa(1,1)))+(2*cos(1.*alfa(1,2)))-
(2*cos(1.*alfa(1,3)))+(2*cos(1.*alfa(1,4)))-(2*cos(1.*alfa(1,5)))

THD2=1-(2*cos(3.*alfa(1,1)))+(2*cos(3.*alfa(1,2)))-
(2*cos(3.*alfa(1,3)))+(2*cos(3.*alfa(1,4)))-(2*cos(3.*alfa(1,5)))

THD3=1-(2*cos(5.*alfa(1,1)))+(2*cos(5.*alfa(1,2)))-
(2*cos(5.*alfa(1,3)))+(2*cos(5.*alfa(1,4)))-(2*cos(5.*alfa(1,5)))

THD4=1-(2*cos(7.*alfa(1,1)))+(2*cos(7.*alfa(1,2)))-
(2*cos(7.*alfa(1,3)))+(2*cos(7.*alfa(1,4)))-(2*cos(7.*alfa(1,5)))

THD5=1-(2*cos(9.*alfa(1,1)))+(2*cos(9.*alfa(1,2)))-
(2*cos(9.*alfa(1,3)))+(2*cos(9.*alfa(1,4)))-(2*cos(9.*alfa(1,5)))

```

### **EK A.3**

#### **SEÇİM ALT PROGRAMI**

```

function [secim]=f_secim(fitness)
[nu_birey x]=size(fitness);
rullet=sum(fitness)*rand(nu_birey,1);
for j=1:nu_birey,
    dilim=0;i=0;
    while dilim<rullet(j,1),
        i=i+1;
        dilim=dilim+fitness(i,1);
    end,
    secim(j,1)=i;

```

end,

#### **EK A.4**

##### **ÇAPRAZLAMA ALT PROGRAMI**

```
function [cr_nufus]=f_cross(ebeveyn1,ebeveyn2)
    [nu_cift x]=size(ebeveyn1);
    nokta=ceil(rand(nu_cift,1)*(x-1));
    cocuk1=zeros(nu_cift,x);
    cocuk2=zeros(nu_cift,x);

    for k=1:nu_cift,
        r=rand(1,1);
        cocuk1(k,1:nokta)=(ebeveyn1(k,1:nokta)*r)+(ebeveyn2(k,1:nokta)*(1-r));
        cocuk2(k,1:nokta)=(ebeveyn1(k,1:nokta)*(1-r)+(ebeveyn2(k,1:nokta)*r);
        cocuk1(k,nokta+1:x)=(ebeveyn1(k,nokta+1:x)*r)+(ebeveyn2(k,nokta+1:x)*(1-r));
        cocuk2(k,nokta+1:x)=(ebeveyn1(k,nokta+1:x)*(1-r)+(ebeveyn2(k,nokta+1:x)*r);
        cr_nufus=cat(1,cocuk1,cocuk2);
    end;
```

#### **EK A.5**

##### **MUTASYON ALT PROGRAMI**

```
function [mu_nufus]=f_mutasyon(cr_nufus,mutasyon_rate)
    [nu_birey x]=size(cr_nufus);
    mo=(rand(nu_birey,x)-0.5)*2;
    mo_a=zeros(nu_birey,x);
    for i=1:nu_birey,
        for j=1:x,
            if mo(i,j)<mutasyon_rate
                mo_a(i,j)=1;
            end,
        end;
    end;
    mu_nufus=cr_nufus+(mo.*mo_a);
```



## **EK A.6**

### **CEZA ALT PROGRAMI**

```
function [PENALTY]=f_ceza(nufus,minmax,pen_ks)
[nu_birey x]=size(nufus);
PENALTY=zeros(nu_birey,1);
for i=1:nu_birey,
    for j=1:x,
        if (nufus(i,j)<minmax(j,1))|(nufus(i,j)>minmax(j,2))
            PENALTY(i,1)=PENALTY(i,1)+pen_ks;
        end;
    end;
end;
```

## **EK A.7**

### **UYGUNLUK ALT PROGRAMI**

```
function [fitness]=f_fitness(obj_func)
[nu_birey x]=size(obj_func);
[Y Z]=sort(obj_func);
[K L]=sort(Y);
fitness(Z,1)=(nu_birey+1-L)./nu_birey;
```

## **ÖZGEÇMİŞ**

21.09.1980 İstanbul doğumluyum. 1997 yılında Medine Tayfur Sökmen Lisesi'nden mezun oldum. 2003 yılında İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi Elektrik Mühendisliği Bölümünden mezun oldum. 2003 yılında Fen Bilimleri Enstitüsü Elektrik Mühendisliği Programına kabul edildim. 2003–2006 yılları arasında EKA Elektronik Kontrol Aletleri A.Ş.'de Araştırma ve Geliştirme Bölümünde Proje Sorumlusu olarak görev aldım. Halen özel bir firmada proje sorumlusu olarak görev almaktayım.