

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AKILLI YÖNTEMLERLE SAYISAL FİLTRE TASARIMI

YÜKSEK LİSANS TEZİ

Tuba TANYILDIZI
(091113103)

Anabilim Dalı: Elektrik ve Elektronik Mühendisliği
Programı: Devreler ve Sistemler

Danışman: Doç. Dr. Arif GÜLTEN

Tezin Enstitüye Verildiği Tarih: 31 Kasım 2012

ARALIK-2012

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AKILLI YÖNTEMLERLE SAYISAL FİLTRE TASARIMI

YÜKSEK LİSANS TEZİ

Tuba TANYILDIZI
(091113103)

Tezin Enstitüye Verildiği Tarih: 31 Kasım 2012

Tezin Savunulduğu Tarih:19 Aralık 2012

Tez Danışmanı: Doç. Dr. Arif GÜLTEN

Diğer Jüri Üyeleri: Doç. Dr. A. Bedri ÖZER

Yrd. Doç. Sencer ÜNAL

ARALIK-2012

ÖNSÖZ

Bu çalışmada bilgisiyle ve verdiği güven duygusu ile her türlü konuda yardımlarını benden esirgemeyen değerli danışmanım Sayın Doç. Dr Arif Gülten'e teşekkürlerimi sunarım.

Ayrıca tez çalışmam süresince manevi desteklerini benden esirgemeyen özellikle kız kardeşim Özge Tanyıldızı'na ve değerli aileme teşekkür ederim.

TUBA TANYILDIZI

Elazığ 2012

İÇİNDEKİLER

	Sayfa No
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	IX
SEMBOLLER LİSTESİ	X
KISALTMALAR LİSTESİ	XI
1. GİRİŞ.....	1
2. SAYISAL FİLTRE.....	4
2.1. Sayısal filtrelerin gerçekleştirilmesi.....	5
2.2. Sayısal FIR filtre	5
2.2.1. Sayısal FIR filtrenin karakteristik özellikleri	6
2.2.2. Sayısal FIR filtrenin avantajları	7
2.3. Sayısal IIR filtre	7
2.3.1. Butterworth IIR filtre.....	8
2.3.2. Chebyshev IIR filtre	8
2.3.3. Tip2 Chebyshev veya Ters Chebyshev IIR Filtre	9
2.3.4. Eliptik IIR filtre	10
2.3.5. Bessel IIR filtre	10
2.3.6. FIR ve IIR filtrelerin karşılaştırılması.....	10
2.3.7. Analog filtre ile sayısal filtre arasındaki farklar.....	11
3. YAPAY ZEKA	12
3.1. Yapay zeka 'nın kullanım alanları.....	12
3.2. Yapay zeka teknikleri.....	13
3.3. Yapay sinir ağları	13
3.4. Uzman sistemler	14
3.5. Çok katmanlı algılayıcılar ve öğrenme algoritmaları.....	14
3.6. Genetik algoritma	14

3.6.1.	Genetik algoritmanın araştırma teknikleri içerisindeki yeri.....	15
3.6.2.	Genetik algoritmaları diğer yöntemlerden ayıran en belirgin özellikleri	16
3.6.3.	Genetik algoritmaların uygulama alanları	16
3.6.3.1.	Mekanik öğrenme.....	16
3.6.3.2.	Eniyileme (Optimizasyon)	17
3.6.3.3.	Otomatik programlama ve bilgi sistemleri.....	17
3.6.3.4.	Ekonomik ve sosyal sistem modelleri	17
3.6.4.	Genetik algoritmanın aşamaları.....	18
3.6.5.	Genetik algoritma operatörleri	19
3.6.5.1.	Başlangıç popülasyonu.....	20
3.6.5.2.	Uygunluk fonksiyonu	20
3.6.5.3.	Seçim.....	20
3.6.5.4.	Çaprazlama.....	21
3.6.5.5.	Mutasyon.....	21
3.6.6.	Kromozomların şifrelenmesi.....	21
3.6.6.1.	İkili kodlama.....	22
3.6.6.2.	Permutasyon kodlama	22
3.6.6.3.	Değer kodlama	23
3.6.6.4.	Ağaç kodlama.....	23
3.6.7.	Çaprazlama ve Mutasyon	24
3.6.7.1.	İkili kodlanmış kromozomlarda çaprazlama	24
3.6.7.2.	İkili kodlanmış kromozomlarda mutasyon.....	25
3.6.7.3.	Permutasyon kodlanmış kromozomlarda çaprazlama ve mutasyon.....	25
3.6.7.4.	Değer kodlanmış kromozomlarda çaprazlama ve mutasyon.....	26
3.6.7.5.	Ağaç kodlanmış kromozomlarda çaprazlama ve mutasyon	26
3.6.8.	Genetik algoritmanın parametreleri.....	27
3.6.8.1.	Çaprazlama olasılığı.....	27
3.6.8.2.	Mutasyon olasılığı	27
3.6.8.3.	Diğer parametreler.....	28
3.6.9.	Genetik algortmada seçim.....	29
3.6.9.1.	Rulet tekeri seçimi.....	29
3.6.9.2.	Sıralama seçimi	30

3.6.9.3. Sabit durum seçimi.....	30
3.6.9.4. Turnuva seçimi	30
3.7. Parçacık sürü optimizasyonu.....	31
3.7.1. Sosyal ağ yapıları	32
3.7.1.1. Yıldız yapı	33
3.7.1.2. Halka yapı.....	33
3.7.2. PSO algoritma kodu	34
3.7.3. PSO parametre kontrolü	35
3.7.4. PSO'nun genetik algoritma ile karşılaştırılması.....	36
3.8. Yapay arı koloni algoritması (ABC)	36
3.8.1. Arılar	37
3.8.2. ABC algoritmasının özellikleri	38
3.8.3. ABC algoritmasının temel adımları	38
4 MATLAB'DA UYGULAMALAR.....	40
4.1. Filtrelerin katsayılarının bulunması.....	40
4.1.1. FIR filtrenin katsayılarının bulunması	40
4.1.2. Butterworth IIR filtrenin katsayılarının bulunması.....	45
4.1.3. Chebyshev IIR filtrenin katsayılarının bulunması.....	52
4.1.4. Eliptik IIR filtrenin katsayılarının bulunması	59
5. SONUÇLAR.....	66
KAYNAKLAR.....	67
ÖZGEÇMİŞ	71

ÖZET

Sayısal filtreler sayısal işaret işlemenin en temel elemanıdır. Temel olarak ikiye ayrılırlar. Bunlar; sonlu darbe cevaplı (FIR) filtre ve sonlu olmayan darbe cevaplı (IIR) filtrelerdir. Bu çalışmada Butterworth IIR, Chebyshev IIR, Eliptik IIR, Bessel IIR ve FIR filtre tasarımı için klasik yöntemler dışında farklı yöntemler kullanılmıştır. Çalışmada kullanılan yöntemler Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu ve yapay arı koloni algoritmasıdır. Filtre tasarımı için Matlab’da program ve arayüz oluşturulmuştur, elde edilen sonuçlar birbiriyle karşılaştırılmıştır.

Anahtar Kelimeler: Genetik algoritma, Parçacık sürü optimizasyonu, Yapay arı kolonisi, FIR filtre, IIR filtre, Butterworth filtre, Chebyshev II filtre, Eliptik filtre.

SUMMARY

Digital filters are the basic elements of the digital signal processing. These filters are divided about two parts of the finite impulse response (FIR) and infinite impulse response filters (IIR) filter. In this study, it was used that different methods apart from classical for Butterworth IIR, IIR Chebyshev, Elliptic IIR, IIR and FIR filter design. Used methods in study are the genetic algorithms (GA), particle swarm optimization (PSO) and artificial bee colony algorithm. Program and analysis in matlab was created for filter design and the obtained results are were compared

Keywords: Genetic algorithm, Particle swarm optimization, Artificial bee colony, FIR filter, IIR filter, Butterworth filter, Cheby2 filter, Elliptic

ŞEKİLLER LİSTESİ

	Sayfa No
Şekil 2.1. Filtreleme işlemleri blok diyagramları	5
Şekil 3.1. Araştırma teknikleri	16
Şekil 3.2. Genetik algoritmalarda evrimleşme döngüleri	19
Şekil 3.3. İkili kodlama için kromozom örneği	22
Şekil 3.4. Permutasyon kodlama için kromozom örneği	22
Şekil 3.5. Değer kodlama için kromozom örneği	23
Şekil 3.6. Ağaç kodlama için kromozom örneği.....	24
Şekil 3.7. Ağaç kodlamada çaprazlama operatörünün kullanımı	27
Şekil 3.8. Rulet seçimi	30
Şekil 3.9. Parçacığın hareketi.....	32
Şekil 3.10. Yıldız topolojisi (Tam bağlantılı graf).....	33
Şekil 3.11. Halka topolojisi.....	34
Şekil 3.12. PSO akış diyagramı	35
Şekil 4.1. FIR filtre	41
Şekil 4.2. n=3 FIR filtre için tahmin edilen ve gerçek değerler.....	42
Şekil 4.3. n=5 FIR filtre İçin tahmin edilen ve gerçek değerler	43
Şekil 4.4. GUI de n=7 için FIR filtre	44
Şekil 4.5. n=7 FIR filtre için tahmin edilen ve gerçek geğerler.....	45
Şekil 4.6. Butterworth IIR filtre	46
Şekil 4.7. n=5 Butterworth filtre için tahmin edilen ve gerçek değerler	48
Şekil 4.8. GUI de n=7 Butterworth filtre	49
Şekil 4.9. n=7 Butterworth filtre için tahmin edilen ve gerçek değerler	52
Şekil 4.10. Chebyshev IIR filtre	53
Şekil 4.11. n=5 Chebyshev filtre için tahmin edilen ve gerçek değerler	55
Şekil 4.12. GUI de n=7 için Chebyshev Filtre.....	56
Şekil 4.13. n=7 Chebyshev filtre İçin tahmin edilen ve gerçek değerler.....	60
Şekil 4.14. Eliptik IIR filtre	61
Şekil 4.15. n=5 Eliptik filtre için tahmin edilen ve gerçek değerler	62
Şekil 4.16. GUI de n=7 için eliptik filtre	63
Şekil 4.17. n=7 Eliptik filtre için tahmin edilen ve gerçek değerler	66

TABLÖLAR LİSTESİ

	Sayfa No
Tablo 3.1. Çeşitli GA parametrelerinin değerlendirilmesi.....	29
Tablo 4.1. $n=3$ için FIR Filtre Katsayıları.....	41
Tablo 4.2. $n=5$ için FIR Filtre Katsayıları.....	42
Tablo 4.3. $n=7$ için FIR Filtre Katsayıları.....	44
Tablo 4.4. $n=5$ için GUI de Butterworth IIR Filtrenin Katsayıları	46
Tablo 4.5. $n=5$ için Butterworth IIR Filtrenin Katsayıları	47
Tablo 4.6. $n=7$ için GUI de Butterworth IIR Filtrenin Katsayıları	50
Tablo 4.7. $n=7$ için Butterworth IIR Filtrenin Katsayıları	51
Tablo 4.8. $n=5$ için GUI de Chebyshev IIR filtrenin katsayıları	53
Tablo 4.9. $n=5$ için Chebyshev IIR filtrenin katsayıları	54
Tablo 4.10. $n=7$ için GUI de Chebyshev IIR Filtrenin Katsayıları.....	57
Tablo 4.11. $n=7$ için Chebyshev IIR Filtrenin Katsayıları.....	58
Tablo 4.12. $n=5$ için GUI de Eliptik IIR Filtrenin Katsayıları.....	60
Tablo 4.13. $n=5$ için Eliptik IIR Filtrenin Katsayıları	61
Tablo 4.14. $n=7$ için GUI de Eliptik IIR Filtrenin Katsayıları.....	63
Tablo 4.15. $n=7$ için Eliptik IIR Filtrenin Katsayıları	64

SEMBOLLER LİSTESİ

f_c	: Kesim frekansı
$y[k]$	: Filtrenin çıkışı
$x[k]$	: Filtrenin girişi
$H(z)$	: Filtrenin transfer fonksiyonu
N	: Filtrenin uzunluğu
G	: Filtrenin kazancı
ϵ	: Dalgacık parametresi
$T_n \left(\frac{\omega}{\omega_0} \right)$	: n. Dereceden Chebyshev çok terimli
P	: Popülasyon büyüklüğü
Ch	: Kromozomun uzunluğu
R	: Direnç
C	: Kondansatör
L	: İndüktans
$pbest$	: Parçacığın en iyi değeri
$gbest$	: Global en iyi değer
v_i	: Parçacığın hızı
x_i	: Parçacığın konumu
c_1	: Öğrenme faktörü
fi	: Uygunluk değeri
n	: Filtrenin derecesi
ω_n	: Filtrenin kesim frekansı
R_p	: Geçirme bandı dalgalanma direnci
R_s	: Durdurma bandı zayıflatma direnci
ω_p	: Geçirme bandı kenar frekansı
ω_s	: Durdurma bandı kenar frekansı

KISALTMALAR LİSTESİ

IIR	: Infinite Impulse response (Sonlu Olmayan Darbe Cevabı)
FIR	: Finite Impulse response (Sonlu Darbe Cevabı)
GA	: Genetik algoritma
PSO	: Parçacık sürü optimizasyonu (Particle Swarm Optimization)
ABC	: Yapay arı koloni algoritması (Artificial Bee Colony)
GUI	: Grafiksel ara birim (Graphical User Interface)
VLSI	: Çok geniş ölçekli entegre devre
CRPSO	: Craziness tabanlı parçacık sürü optimizasyonu
RGA	: Reel kodlu genetik algoritma
IPSO	: Geliştirilmiş parçacık sürü optimizasyonu
QPSO	: Kültürel sarçacık sürü optimizasyonu
FPGA	: Programlanabilir kapı dizileri
IABCLS	:Yerel arama ile artan ABC algoritması
FFT	: Hızlı Fourier dönüşümü
GSP	: Gezgin satıcı problemi

1. GİRİŞ

Bir işaretin frekans spektrumundaki belirli özelliklerini pekiştiren, işaretin frekans spektrumunu yeniden biçimlendiren veya işaretin frekans spektrumunu istenen belirli özelliklere göre değiştiren sistemlere genel olarak filtre denir. Filtrenin görevi istenilen sinyalleri geçirmek, istenilmeyen sinyallerin ve parazitlerin süzme işlemini gerçekleştirmektir. Filtre, bazı sinyallerin geçmesini, bazı sinyallerin de durdurulmasını sağlar. Filtrelerin kullanım alanları geniştir. Filtreler örneğin, gürültü gibi işaretlerdeki kirlenme etkenlerinin ayrıştırılması ve haberleşme kanalı gibi iletim ortamlarındaki bozucu etkilerinin giderilmesi için kullanılırlar. Aynı ortamda bulunan birden fazla işaretin ayrıştırılması için de filtreler kullanılırlar. Filtreler, işaretlerin frekans bileşenlerine ayrılması, işaretlerin demodüle edilmesi veya işaretlerin bant genişliğinin sınırlandırılması gibi alanlarda da kullanılırlar [1].

Sayısal filtre, ayrık zamanlı işaretleri filtrelemek için kullanılan sayısal bir sistemdir. Sayısal filtreler, işaretin örneklenmiş değerleri üzerinde nümerik hesapları yapabilmek için sayısal işlemcileri kullanır. Sayısal filtreler, yazılımsal ya da donanımsal olarak veya her ikisinin birleştirilmesiyle gerçekleştirilebilir. Yazılıma dayalı sayısal filtreler, düşük seviyeli diller kullanılarak genel amaçlı sayısal işaret işleme sistemleri yardımıyla tasarlanabilir. Yüksek seviyeli diller kullanılarak da iş istasyonları ile veya kişisel bilgisayar vasıtasıyla tasarlanabilirler. Donanımsal sayısal filtreler, özel geliştirilmiş çok geniş ölçekli entegre devre (VLSI) sistemleri kullanılarak tasarlanabilir. Donanımsal sayısal filtreler, yüksek hesap gücü ve hızlı işlem yetenekleri sayesinde gerçek zamanlı sayısal işaretleri işlemede yaygın olarak kullanılırlar [2].

Sayısal filtrelerde yapay zeka yöntemlerinin uygulanmasıyla ilgili çeşitli çalışmalar mevcuttur. Karaboğa çalışmasında genetik algoritma yardımıyla sayısal filtre katsayılarını yuvarlatmıştır. Genetik algoritma kullanarak değişik tip sayılar için FIR filtre katsayılarının kuantulanmasını gerçekleştirmiştir [3]. Batık çalışmasında filtre tasarımında (gerek IIR gerekse FIR filtrede) fonksiyonu oluşturan pay ve payda katsayılarını hesaplamıştır. Genetik algoritma ile katsayılar elde edilirken GA her çalıştırıldığında farklı sonuçlar elde edilmiş ve sonuçların birbirine çok yakın olduğu görülmüştür. Parametre değişikliğinden dolayı katsayıların hassasiyet oranlarının değişimi de incelenmiştir [4]. Çetinkaya çalışmasında genetik algoritma kullanılarak farklı

dereceden sonlu darbe cevaplı ve sonsuz darbe cevaplı alçak geçiren, yüksek geçiren ve bant geçiren filtrelerin tasarımları gerçekleştirilmiş ve elde edilen sonuçlar kendi içerisinde mukayese edilmiştir. Tasarlanmak istenen filtre yapılarının ideal genlik cevapları, kutup sıfır diyagramları, güç spektrumları ve katsayılar genetik algoritmanın çıkışından elde edilmiştir [2]. Karaboğa ve Çetinkaya GA kullanarak minimum fazlı sayısal FIR filtre tasarlanmıştır. Bant geçiren ve bant durduran filtreleri optimize etmek için Squared hata ortalaması fonksiyonunu kullanmıştır [5]. Zhang vd. dördüncü derecedeki Chebyshev filtreyi genetik algoritma yardımıyla modellemiştir. Bu model sayesinde daha büyük ölçekli devreler daha pratik yöntemlerle tasarlanmıştır [6]. Saha vd. ise sekizinci derece alçak geçiren IIR filtreye Craziness tabanlı parçacık sürü optimizasyonu (CRPSO) uygulanmıştır. CRPSO tekniği PSO tekniğinin gelişmiş versiyonudur. Reel kodlu genetik algoritma (RGA) ve PSO ile yapılan çalışmalar karşılaştırılmıştır. CRPSO algoritmasındaki benzer sonuçlar onaylanmıştır [7]. Mondal vd. geliştirilmiş parçacık sürü optimizasyonu (IPSO) kullanarak yüksek geçiren FIR filtre tasarlamıştır. Tasarım sürecinde filtre uzunluğu, band geçirme frekansı, bant durdurma frekansı, bant geçirme dalgalanma boyutu, bant durdurma dalgalanma boyutu belirlenmiştir. Yüksek geçiren FIR filtre tasarımı için gerçek kod genetik algoritma (RGA), parçacık sürü optimizasyonu (PSO) , geliştirilmiş parçacık sürü optimizasyonu (IPSO) gibi gelişmiş algoritmalar kullanılmıştır [8]. Zhao ve Gao FIR filterleri kültürel parçacık sürü optimizasyonu ile tasarlanmıştır. Bu modelde orijinal PSO'dan daha iyi bir yöntem olduğu gösterilmiştir. QPSO ile daha iyi bir yakınsama hızı ve daha iyi bir performans elde edilmiştir [9]. Krusienski ve Jenkins adaptif IIR filtreyi yapılarını tasarlarken parçacık sürü optimizasyonu kullanmıştır. PSO bilinmeyen parametrelerin uygun çözüm aramada GA ile benzer sonuçlar elde etmiştir. Her iki teknikte multimodal optimizasyon problemleri için çözüm üretmiştir [10]. Gao vd. adaptif IIR filtre, FPGA ve parçacık sürü optimizasyonu yardımıyla tasarlamıştır. Parçacık sürü optimizasyonu en uygun parametreyi tahmin etmiştir. FPGA uygulamaları kolaylaştırmıştır. Sentez ve fonksiyon sonuçları FPGA kullanarak PSO algoritması ile IIR filtrelerde uygulanmıştır [11]. Durmuş vd. IIR filtrenin parametre tanımlaması için yapay arı koloni algoritması kullanmıştır. Filtrenin bilinmeyen parametreleri bir vektör ile optimize edilmiştir. Yapay arı koloni algoritması diğer yöntemlere göre filtre parametreleri daha doğru tahmin etmiştir. IABCLS verimli bir şekilde kullanarak dijital IIR filtre tasarlanmıştır [12]. Vural ve Yıldırım alçak geçiren

Butterworth filtrelere yapay arı koloni algoritmasını uygulamıştır. Bu çalışmanın amacı E12 serisi ile uyumlu Butterworth alçak geçiren filtre tasarlanırken toplam tasarım hatasını en aza indirmektir. Butterworth filtrenin yüksek doğruluk ve maksimum düz yanıtı pasif bileşenleri kısa bir süre içinde ABC optimizasyon yöntemi yardımı ile seçilmiştir [13].

Yapılan tez çalışması 6 bölümden oluşmaktadır. Bölüm 2’de sayısal filtre, sayısal filtrenin özellikleri, sayısal FIR filtre, sayısal IIR filtre, Butterworth IIR filtre, Chebyshev IIR filtre, Eliptik IIR filtre, Bessel IIR filtre, FIR ve IIR filtrenin karşılaştırılması gibi konular incelenmiştir.

Bölüm 3’de yapay zeka, yapay zekanın amacı, yapay zekanın uygulama alanları, genetik algoritma, genetik algoritma aşamaları, genetik algoritma operatörleri, genetik algoritma parametreleri ayrıntılarıyla anlatılmıştır. Parçacık sürü optimizasyonu, PSO algoritma kodu, PSO parametre kontrolü, yapay arı koloni algoritması (ABC), ABC algoritmasının özellikleri, ABC algoritmasının temel adımlarından bahsedilmiştir.

Bölüm 4’de FIR filtre, Butterworth filtre, Chebyshev filtre, Eliptik filtrelerin katsayı değerleri GA, PSO, ABC ile bulunmuştur. Bulunan değerler tablo haline getirilmiştir.

Bölüm 5’de ise elde edilen sonuçlar karşılaştırılmıştır. Filtrelerde en iyi katsayı analizi yapan yöntemler tesbit edilmiştir.

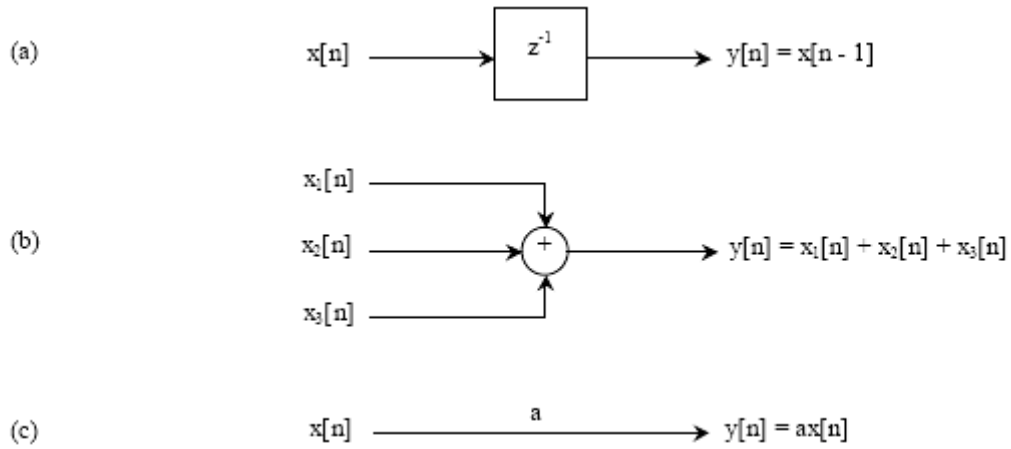
2. SAYISAL FİLTRE

Sayısal filtre, sayısallaşmış sinyaller üzerinde çalışan, giriş sinyalini istenen çıkış sinyaline dönüştüren yöntem ya da algoritmadır. Sayısal filtreler, ayrık zamanlı işaretlerin filtrelenmesi için kullanılan sayısal sistemlerdir. Sayısal filtrelerin temelini oluşturan prensipler zaman ve frekans bölgeleri arasındaki ilişkiye dayandırılmıştır. Bu ilişkiyle ilgili bilgi sahibi olmadan da sayısal filtreler tasarlanabilir. Ancak sayısal filtrelerle ilgili temel bilgi sahibi olmak bile süreci çok daha fazla anlaşılır hale getirecektir. Zaman ve frekans bölgeleri arasında dönüşüm yapmak için Fourier ve Laplace dönüşümleri kullanılır. Sayısal filtreler, zaman formunda iş gören yapılar olmalarına karşın, tasarımları frekans formunda gerçekleştirilir. Her şeyden önce sayısal filtrelerden beklenen, filtrelerin kararlı olması ve işaretin bozulmaya uğratılmadan işleminden geçirilmesi için filtre transfer fonksiyonu faz cevabının frekansa göre lineer olmasıdır. Sayısal filtrelerin hızlı bir şekilde fonksiyonlarını yapabilmeleri ve mikroişlemci uygulamalarında gerçek zamanda çalışabilmeleri için filtre katsayılarının tam sayı olması gerekmektedir [1].

Sayısal filtreler yazılım ve donanım olmak üzere iki biçimde olabilir. Yazılım yönteminde bir sayısal bilgisayar filtrenin benzetimini yapmakta kullanılır. Yazılım türü sayısal filtreler, 1940'larda ilk sayısal bilgisayarın ortaya çıkması ile kullanılmaya başlanmıştır. İlk sayısal bilgisayarlar, ayrık zamanlı işaretleri temsil eden sayı dizilerinin oluşturduğu fonksiyonlar için Newton, Stirling ve Everet gibi klasik nümerik analiz formülleri ile fark denklemlerinin çözümü, nümerik integral ve interpolasyon gibi işlemler için kullanılıyordu. Bu işlemler için yapılan programlar, ilk yazılım türü sayısal filtre programı olarak tanımlanabilir. Donanımda ise, sayısal filtrenin devre yapısı özel amaçlı bir aygıtı dönüştürülür. Sayısal filtreler bilgisayarlarda kullanılabilecek şekilde, yazılım paketleri ile gerçekleştirilebileceği gibi çarpıcı, kaydırıcı, toplayıcı vb. bunun gibi elemanları içeren donanım devreleri halinde de gerçekleştirilebilir. Özellikle FFT işlemciler, frekans sentezörü ve dalga analizörü gibi sayısal işaret işleyici filtreler günümüzde yaygın olarak kullanılmaktadır [14]. Entegre devre teknolojisindeki hızlı gelişmeler sayısal filtre elemanlarının daha küçük boyutlarda ve daha iyi performanslı halde üretilmelerini sağlamıştır. Yazılım ve donanım tekniklerinin bir arada kullanıldığı uygulamalarda daha az maliyet giderleri saptanmış ve daha verimli çalışmalar da gerçekleştirilmiştir [15].

2.1. Sayısal Filtrelerin Gerçekleştirilmesi

Sayısal bir filtrenin iç yapısı blok diyagramlar veya sinyal akış şemaları ile gösterilebilir. Bu çeşit bir gösterime sayısal filtrenin gerçekleştirilmesi denir. Birim gecikme ile Şekil 2.1.'de basit filtreleme işlemleri blok diyagramlar olarak gösterilmiştir. Birim gecikme Şekil 2.1.a.'da, toplama işlemleri Şekil 2.1.b'de, sinyallerin sabit katsayılarla çarpılma işlemleri Şekil 2.1.c.'de görülmektedir. Z-bölgesinde gecikme operatörü z^{-1} notasyonu ile gösterilir.[16].



Şekil 2.1. Filtreleme İşlemleri Blok Diyagramları [16].

2.2. Sayısal FIR Filtre

Sayısal FIR filtreler önce sayısallaştırılmış bir sinyali bir dizi farklı geciktirme ögesinden geçirip sonra da her bir geciktirme ögesinin çıkışını bir sayı ile çarpma yolu ile fonksiyon oluşturmuştur. Sonra da her bir saat periyodunda tüm çarpma fonksiyonları tarafından üretilen değerler bir çıkış oluşturacak şekilde toplanır. FIR filtreler, birkaç başarılı örnek alır ve sonra bu örneklerin ortalamasını alarak toplar. Zaman geçtikçe örnekler filtrelerden dalgalanarak geçer. Girişteki herhangi bir ani değişiklik çıkışı geçiş süresi boyunca etkiler. FIR filtre kararlıdır. Ancak çarpım faktörü kullanılan katsayıların kesilmesi ideal olmayan bir frekans yanıtına da yol açabilir.

Her iki şekilde de bir FIR filtrenin avantajı çıkışın doğrusal fazlı olmasıdır. Her bir giriş sinyali tüm geciktirici elemanlardan geçtiği için doğrusal fazlıdır. Bu anlamda yavaş değişen bir sinyal, hızlı değişen bir sinyalle aynı işlemlerden geçmektedir. Tüm frekanslar

eşit düzeyde geciktirilir. Yani grup geciktirmesi sabittir ve filtredeki geciktirici elemanların sayısıyla orantılıdır.

Bir FIR filtrenin tüm sinyallerinin maruz kaldığı gecikme aynıdır ve sinyal frekansına bağlı olmaması nedeniyle grup gecikmesi sabittir. Bu durum sayısal sinyallerle çalışan filtrelerde önem arz etmektedir. Çünkü dürtüler geniş bir frekans bandını içermektedir. Grup gecikmesinin sabit olmadığı ve bazı frekansların diğerlerine göre daha fazla geciktirildiği durumlarda dürtü, dalga formu üzerine bindirilmiş bu sinyal açısından istenmeyen bir durum oluşturur. Bu durum da bir bozulmadır. Diğer yandan temel konuşma iletim grup gecikmesi değişikliğinden çok fazla etkilenmez. Bunun için bu tür uygulamalar da IIR filtre çok daha uygundur

Bir FIR filtrenin kesim frekansı (f_c) veri örnekleme saat frekansı ile doğru orantılıdır. Kesim frekansı tek bir katsayı kümesi kullanılarak örnekleme saat frekansını ikiye katlamak suretiyle kullanılmıştır. Giriş sinyalinin uygun örnekleri ile $b_0, b_1, b_2, b_3 \dots$ katsayı grubunun çarpılarak ortaya çıkan terimlerin toplanmasıdır [17].

$$y[k] = b_0 \cdot x[k] + b_1 \cdot x[k-1] + b_2 \cdot x[k-2] \dots + b_k \cdot x[0] \quad (2.1)$$

Bu işlem başka bir ifadeyle, giriş sinyalinin, $b_0, b_1, b_2, b_3, \dots$ değerlerinden oluşan bir filtrenin (örneğin bir dürtü cevabı ile) konvolüsyon işlemine tabi tutulmasıdır. Konvolüsyon işlemi

$$y[k] = \sum_{n=-\infty}^{\infty} h[n] \cdot x[k-n] \quad (2.2)$$

ile verilir.

2.2.1. Sayısal FIR Filtrenin Karakteristik Özellikleri

Temel bir FIR filtre aşağıdaki iki denklemle karakterize edilir.

$$y[k] = \sum_{n=0}^{N-1} h[n] \cdot x[k-n] \quad (2.3)$$

$$H(z) = \sum_{n=0}^{N-1} h[n] \cdot z^{-n} \quad (2.4)$$

FIR filtrenin $H(z)$ transfer fonksiyonu Denklem 2.4'de görülmektedir. FIR filtrelerin transfer fonksiyonunda geri besleme ifadesinin olmadığı görülmektedir. Burada $h[n]$, $n=0,1,2,\dots,N-1$ filtrenin birim dürtü cevap katsayılarıdır. $H(z)$ filtrenin transfer fonksiyonudur ve N ise filtrenin uzunluğu olup filtre katsayılarının sayısına eşittir. FIR filtre keskin gecikme bandı karakteristiğini sağlamak için yüksek dereceli gerçekleştirmeye ihtiyaç duyar. Bu nedenle fazla keskin olmayan karakteristikler tercih edilmelidir [18].

2.2.2. Sayısal FIR Filtrelerin Avantajları

- FIR filtreler lineer fazlı ve önceden belirlenmiş genlik frekans karakteristiğini sağlayacak şekilde kolaylıkla tasarlanabilir. Buna ek olarak herhangi bir frekans karakteristiğini (hem faz hem de genlik) FIR filtre yaklaşık olarak sağlayabilir.
- FIR filtreler hem tekrarlı hem de tekrarlı olmayan şekilde gerçekleştirilebilir. Tekrarlı gerçekleştirmede tarak filtresi ve rezenatör bankası kullanılır. Tekrarlı olmayan gerçekleştirmede ise doğrudan konvolüsyon için hızlı konvolüsyon yöntemleri kullanılır. Tekrarlı olmayan şekilde gerçekleştirilen bir FIR filtre daima kararlıdır. Bu tür filtrelerin z -düzleminde sadece sıfırlar olup kutupları olmaz. Bu yüzden bu tür filtreler daima kararlıdır.
- IIR filtrelerin gerçekleştirilmesinde doğal olarak kuantalama ve yuvarlatma hataları ortaya çıkar. Ancak, FIR filtrelerin tekrarlı olmayan gerçekleştirilmesinde bu hatalar önemsizdir. Keskin kesim frekanslı IIR filtre tasarımında filtre katsayılarından gelen doğruluk problemi önemlidir. Ancak, aynı özellikte FIR filtre için katsayı hataları daha az önemlidir [19].

2.3. Sayısal IIR Filtre

Sonsuz dürtü cevaplı filtreler IIR filtreler olarak adlandırılır. IIR filtrelerin en temel özelliği, geribesleme katı içermeleridir. Bu nedenle bu filtreler, geribeslemeli filtreler olarak da bilinirler. IIR filtrelerinin darbe cevapları sonsuz sürelidir. Çünkü girişine bir darbe uygulandığında geribesleme katından dolayı sonsuz sayıda sıfırdan farklı çıkış elde edilir [2]. Genel bir geribeslemeli süzgeç yapısı gerçekleştirilirken, girişi o andaki ve geçmişteki değerleri pay polinomunun katsayıları olan b_k katsayıları ile çarpılır. Benzer

şekilde, çıkışın geçmişteki değerleri de payda polinomunun katsayıları olan a_k katsayıları ile çarpılır [20].

IIR filtre tasarımında en önemli parametre, tasarlanan sistemin kararlılığıdır. IIR filtrelerin kararlı olabilmesi için, sistemin bütün kutuplarının kompleks düzlemde birim dairenin içerisine yerleşmiş olması gerekir. Birim dairenin üzerine ya da dışına yerleşmiş olan kutuplar sistem kararsızlığına yol açarlar. Bir sistemin bütün kutuplarının birim dairenin içerisine yerleşmiş olması demek, o sistemin darbe cevabının mutlak değerce toplanabilir olması demektir [2].

$$y[k] + b_1 \cdot y[k-1] + b_2 \cdot y[k-2] + \dots + b_m \cdot y[k-m] = a_0 \cdot x[k] + a_1 \cdot x[k-1] + a_2 \cdot x[k-2] + \dots + a_n \cdot x[k-n] \quad (2.5)$$

IIR filtrenin transfer fonksiyonu:

$$H(z) = \frac{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}}{1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}} \quad (2.6)$$

olarak yazılabilir [19].

2.3.1. Butterworth IIR Filtre

Bu filtre için geçiş bandında ve durdurma bandında dalgalanma yoktur. Fonksiyon tekdüze olarak azalır. Butterworth filtrenin diğer filtrelerden farkı, derecesi artırıldığında durdurma bandındaki sert düşüşü dışında frekans genlik eğrisinde şeklini korumasıdır. Butterworth, Chebyshev ve Eliptik filtrelere göre daha geniş geçiş bölgesine sahip olduğundan, durdurma bandı özelliklerinin doğru olarak uygulanabilmesi için yüksek derecelere ihtiyaç duyar. Chebyshev ve Eliptik filtrelere göre daha doğrusal bir frekans tepkisine sahiptir [21].

$$|B(\omega)|^2 = \frac{1}{1 + (\omega / \omega_c)^{2N}} \quad (2.7)$$

2.3.2. Chebyshev IIR Filtre

Geçirme bandı Butterworth filtreye göre daha dardır ve geçirme bandında dalgacıklar vardır. Chebyshev filtre mevcut dalgacıklar dışında bu özelliği sayesinde ideal

filtreye daha yakındır. Eğer filtrede dalgacıklar geçirme bandında ise bu Chebyshev 1.tipi filtredir. Eğer dalgacıklar durdurma bandında ise Chebyshev 2. tip filtre olarak isimlendirilir. Eğer dalgacıklar her iki bantta ise bu durumda ortaya çıkan filtreye Eliptik filtre denir. Eliptik filtrelerin geçiş bandı iki çeşit filtreden de daha kısadır. Dolayısıyla ideale en yakın filtredir. Ancak tasarımı zordur. Chebyshev tip1 filtrenin tanımı:

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 \cdot T_n^2\left(\frac{\omega}{\omega_0}\right)}} \quad (2.8)$$

Burada " ε " dalgacık parametresidir ve geçirme bandında eş dalgacıkların karakteristiğini belirler. $T_n\left(\frac{\omega}{\omega_0}\right)$ ise n.dereceden Chbeyshev çok terimlisidir. Değişen frekansa göre 0 ve 1 arasında değişen bir tepki üretir. Bu değişkenlerle filtrenin kazancı geçirme bandından en fazla $G=1$ ile $G = 1/\sqrt{1 + \varepsilon^2}$ arasında değişir. Kesim frekansına gelindiğinde kazanç geçirme bandındaki en küçük değerindedir.

Chebyshev filtrenin derecesi, analog tasarımında kullanılan reaktif elemanların (L ve C) sayısına eşittir. Butterworth filtreden farklı olarak Chebyshev filtrenin kutupları s düzleminde elips şeklinde konumlanır. Filtrenin kararlılığı ve nedenselliği için imajer eksenin sol tarafındaki kutuplar seçilir [21].

2.3.3. Tip2 Chebyshev veya Ters Chebyshev IIR Filtre

Bu filtrenin kullanımı pek yaygın değildir. Çünkü geçiş bandı diğer tipine göre daha uzundur. Chebyshev tip 2 filtrede 1.tipin aksine geçirme bandındaki dalgacıklar durdurma bandındadır. Tekdüze azalma durdurma bandı yerine geçirme bandındadır.

Chbeyshev tip2 filtrenin tanımı:

$$G_n(\omega, \omega_0) = \frac{1}{\sqrt{1 + \frac{1}{\varepsilon^2 T_n^2(\omega_D / \omega)}}} \quad (2.9)$$

Durdurma bandında Chebyshev çokterimlisi 0 ile 1 arasında salınım yapar. Dolayısıyla

kazanç 0 ile $\frac{1}{\sqrt{1+\varepsilon^2}}$ arasında değişir [21] .

2.3.4. Eliptik IIR Filtre

Eliptik filtre, geçirme ve durdurma bandında sayıları birbirinden bağımsız olarak değiştirilebilen, eş dalgacıklar barındıran bir elektronik filtre çeşididir. Aynı dereceden başka bir filtreye göre geçiş bölgesi daha kısadır. Eliptik filtre, tanıtılan diğer filtrelerin bu dalgacıkların karakteristiklerinin değiştirilmesiyle de oluşabilir. Durdurma ve geçirme bandındaki dalgacıklar sıfıra yaklaştığında da ise Eliptik filtre Butterworth filtreye dönüşebilir [21]. Eliptik filtrenin aktarım işlevi:

$$G_n(\omega) = \frac{1}{\sqrt{1+\varepsilon^2 R_n^2(\xi, \omega/\omega_0)}} \quad \text{şeklindedir.} \quad (2.10)$$

2.3.5. Bessel IIR Filtre

Bessel filtreleri, bütün IIR filtrelerde var olan doğrusal olmayan faz bozulmasını azaltmak için kullanılır. Bessel filtrelerin tasarımı kolaydır. İstenen derecede tasarım yapılır. Genelde alçak geçiren filtre tasarımında kullanılır. Dönüşüm yaparak yüksek geçiren, bant geçiren, bant durduran filtreleri tasarlanabilir. Bessel filtrenin derecesi arttıkça ideale yaklaşır. Bessel filtrelerde hatayı minimize etmek için yüksek dereceli filtrelere ihtiyaç duyulur [17].

2.3.6. FIR ve IIR Filtrelerin Karşılaştırılması

- **Faz Cevabı:** FIR filtrelerin faz cevapları daima doğrusaldır. Bu nedenle, FIR filtreler giriş işaretini geciktirirler fakat fazını bozmazlar. Yani FIR filtreye uygulanan bir işarete hiçbir faz bozulması görülmez. IIR filtreler ise doğrusal olmayan faz yanıtına sahiptirler.
- **Kararlılık:** FIR filtreler sınırlı bir darbe cevabına sahip olduklarından daima kararlıdırlar. Bu yüzden, filtrenin kararsızlığa düşme tehlikesi olmaksızın, filtre katsayıları değiştirilerek sistem performansı iyileştirilebilir. IIR filtre ise her zaman kararlı değildir.
- **Frekans Seçiciliği:** Aynı dereceye sahip bir IIR filtrenin frekans seçiciliği FIR filtreye göre çok daha iyidir. Frekans seçiciliği, FIR filtrelerde ise daha fazla hesap

yükü ile sağlanırken, aynı frekans seçiciliği IIR filtrelerde çok daha az hesap yükü ve daha düşük filtre uzunluğu ile sağlanabilir. Ayrıca, aynı dereceye sahip bir IIR filtrenin frekans cevabının keskinliği FIR filtreye göre çok daha yüksektir.

- **Nümerik Hesaplamalar:** İstenilen bir filtre karakteristiğini gerçekleştirebilmek için, IIR filtreler çok daha az hafızaya ve nümerik hesaplamaya ihtiyaç duyarken, FIR filtreler daha yüksek hafıza ve kompleks hesaplamaları gerektirir [2].

2.3.7. Analog Filtre ile Sayısal Filtre Arasındaki Farklar

Sayısal filtreler analog filtrelerle mümkün olmayan karakteristiğe sahiptir yani doğrusal faz cevaplıdır. Analog filtrelerin tersine sayısal filtreler ortam değişikliğinden etkilenmezler (örneğin sıcaklık değişimi). Sayısal filtrelerin frekans cevabı eğer programlanabilir bir işlemci kullanılarak gerçekleştirilirse (yani adaptif filtrelerde) frekans cevabı otomatik olarak ayarlanır. Sayısal filtrelerde, hem filtrelenmiş hem de filtrelenmemiş veriler daha sonra kullanılmak üzere saklanabilir. Sayısal filtrelerin bir diğer avantajı üretimlerinde VLSI teknolojisinde aşırı bir ilerlemenin olması ve dolayısıyla çok küçük boyutlarda, düşük güç tüketiminde daha az ücretle temin edilebilmeleridir. Analog filtrelerle pratikte başarılabilen doğruluk sınırlıdır. Sayısal filtrelerle bu doğruluk sadece kullanılan kelime uzunluğu ile sınırlıdır. Sayısal filtreler çoğu biyomedikal uygulamalarda olmak üzere çok düşük frekanslarla kullanılabilirler. Ayrıca örnekleme frekansını değiştirmek suretiyle çok geniş frekans aralığında kullanılabilirler [22]. Sayısal filtrenin dezavantajı analog sinyallerin örneklenmesiyle oluşur. Örnekleme frekansı analog sinyalinin en yüksek frekansından iki kat daha az olması durumunda eksik örnekleme görülür. Sayısal filtrenin diğer dezavantajları ise enerjiyi geçirmemek, örnekleme etkileri, güç kaynağı gerektirmeleri ve frekans aralığı sınırlandırmaları olmasıdır [4].

3. YAPAY ZEKA

Yapay zeka, insan zekâsına özgü olan, algılama, öğrenme, çoğul kavramları bağlama, düşünme, fikir yürütme, sorun çözme, iletişim kurma, çıkarım yapma ve karar verme gibi yüksek bilişsel fonksiyonları yapması beklenen yapay bir işletim sistemidir.

Yapay zeka kavramının geçmişi modern bilgisayar bilimi kadar eskidir. Fikir babası, “Makineler düşünebilir mi?” sorusunu ortaya atarak Makine Zekâsını tartışmaya açan Alan Mathison Turing’dir. 1943 te II. Dünya Savaşı sırasında üretilen elektromekanik cihazlar sayesinde bilgisayar bilimi ve yapay zeka kavramları doğmuştur.

Modern bilgisayarın atası olan bu makineler ve programlama mantıkları aslında insan zekâsından ilham alınmıştır. Ancak sonraları, modern bilgisayarlarımız daha çok uzman sistemler diyebileceğimiz programlar ile gündelik hayatımızın sorunları çözmeye yönelik kullanım alanlarında daha çok yaygınlaşmıştır. 1970’li yıllarda büyük bilgisayar üreticileri olan Microsoft, Apple, Xerox, IBM gibi şirketler kişisel bilgisayar modeli ile bilgisayarı popüler hale getirdiler ve bilgisayarlar yaygınlaştı [23].

Yapay zekanın önceki dönem çalışmalarında insan psikolojisi göz ardı edilmiştir. Ama günümüzde bu yönelim beynin çalışmasına dair teorileri temel alan bağlantıcılığın gelişmesiyle değişmiştir. Bağlantıcılığın temelinde basit temel elemanlar arasında belirli bir biçimi olan bilgi parçalarının aktarılmasını da içeren ve öğrenmeyi modellemeye çalışan karmaşık fonksiyonların kullanılması vardır. Günümüzde yapay zeka alanında önemli gelişmeler sağlanmış olmakla beraber, araştırma düzeylerinin hâlâ kuluçka safhasında olduğu söylenebilir. Genel zekâ veya “güçlü yapay zeka” henüz ulaşılabilmiş bir hedef değildir, ancak yapay zeka araştırmacılarının uzun dönemli hedefidir [24].

3.1. Yapay Zekanın Kullanım Alanları

- Problem çözme, arama ve sezgisel programlama
- Bilgi çıkarsama
- Planlama
- Oyunlar
- Yapay yaşam
- Teorem ispatlama

- Doğal dil anlama ve çeviri
- Bilgi tabanlı sistemler
- Makine öğrenmesi
- Makine buluşları
- Robotik
- Şekil tanıma [25].

3.2. Yapay Zeka Teknikleri

- Makine Zekâsı (Sembolik yapay zeka)
- Yapay Sinirsel Ağları (Sibernetik yapay zeka)
- Doğal Dil İşleme (Dil ile düşünme)
- Konuşma Sentezi (Yapay Konuşma)
- Konuşma Anlama (Konuşma Analizi)
- Uzman Sistemler
- Örüntü Tanıma
- Bulanık Mantık
- Çok Katmanlı Algılayıcılar ve Öğrenme Algoritmaları
- Genetik Algoritmalar
- Genetik Programlama [23].

3.3. Yapay Sinir Ağları

Yapay sinir ağları örneklerle ilgili bilgiler toplamakta, genellemeler yapmakta ve daha sonra hiç görmediği örnekler ile karşılaşınca öğrendiği bilgileri kullanarak o örnekler hakkında karar verebilmektedir. Yapay sinir ağları bu öğrenebilme ve genelleme özellikleri sayesinde günümüzde birçok bilim alanında geniş uygulama olanağı bulmakta ve karmaşık problemleri başarı ile çözebilme yeteneğine sahip olmaktadır.

Sinir ağları insan beynindeki sinirlere benzer, bir araya getirilen sinirlerin değişik bağlantılı geometrisi ile birbirlerine bağlanması sonucu oluşmuştur. Sinir ağları paralel hesaplama tekniğini kullanan sistemlerdir. Programlama yerine doğrudan mevcut örnekler üzerinden eğitilerek sinir ağları üzerinde işlem yapılır. Bağımsız değişkenler ile bu değişkenlere ilişkin bağımlı değişkenler arasında ki matematiksel ilişkiyi öğrenebilen

sistemlerdir.

Bu sistemde kullanılan ileri beslemeli ağ mimarisinde sinirler katmanlar halinde yerleştirilir. İleri beslemeli sinir ağı en az üç katmandan oluşur. Bu katmanlı yapısından dolayı çok katmanlı algılayıcı olarak isimlendirilmektedir. İleri beslemeli sinir ağları geniş bir uygulama alanına sahiptir [26].

3.4. Uzman Sistemler

Yapay zekanın en önemli uygulama alanlarından biri uzman sistemleridir. Bu tip sistemler belli bir alanda uzman olan kişilerin uzmanlıklarına dayanarak çözüm arar. Bunu bir tür bilgisayarda düzenlenmiş danışma sistemi olarak düşünebiliriz. Uzman sistemlerin oluşturulmasında sırasıyla; tanımlama, kavramsallaştırma, formüle etme, test etme ve değerlendirme aşamaları uygulanır. Bir uzman sistem kural tabanı, veri tabanı, kural çözümleyici olmak üzere üç bölümden oluşur [26].

3.5. Çok Katmanlı Algılayıcılar ve Öğrenme Algoritmaları

Çok katmanlı algılayıcı modeli bir giriş, bir veya daha fazla ara ve çıkış katmanından oluşur. Bir katmandaki işlem elemanları diğer katmandaki işlem elemanlarına bağlıdır. Birçok öğretim algoritmasının bu ağı eğitmede kullanılabilir olması bu modelin yaygın kullanılmasının nedenidir.

Çok katmanlı algılayıcı ağlarındaki örnekler giriş katmanına uygulanır, ara katmanlarda işlenir ve çıkış katmanında çıkışlar elde edilir. Kullanılan eğitim algoritmasına göre ağı çıkışı ile arzu edilen çıkış arasındaki hata tekrar geriye doğru yayılarak minimuma düşünceye kadar ağırlıkları değiştirilir.

Anlaşılması kolay ve matematiksel olarak ispatlanabilir olmasından dolayı en çok tercih edilen öğrenme algoritmasıdır. Bu algoritma hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Tipik çok katmanlı geri yayılım ağı, daima bir giriş tabakası, bir çıkış tabakası ve en az bir gizli tabakaya sahiptir. Sinir ağı yapısı bulgularının daha başarılı bir şekilde sınıflandırılabilmesi için genetik algoritma ve sinir ağı yöntemi geliştirilmiştir [26].

3.6. Genetik Algoritma

Genetik algoritmalar (GA), doğadaki canlıların geçirdiği süreci örnek alır ve kendi

yaşamlarını korurken kötü nesillerin yok olması ilkesine dayanır. Matematiksel modellemenin yapılamadığı veya kesin çözümün olmadığı problemlerde genetik algorithmadan yararlanılır. Bu algoritma, anne ve baba bireyden (bir önceki nesil) doğan yeni bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesine dayanır. Yeni bireyler, anne ve babasından gelen iyi genleri bünyelerinde muhafaza edebileceği gibi kötü genleri de almış olabilir. Bu durumda da kötü genlere sahip bireyler varlıklarını sürdüremeyecektir [27].

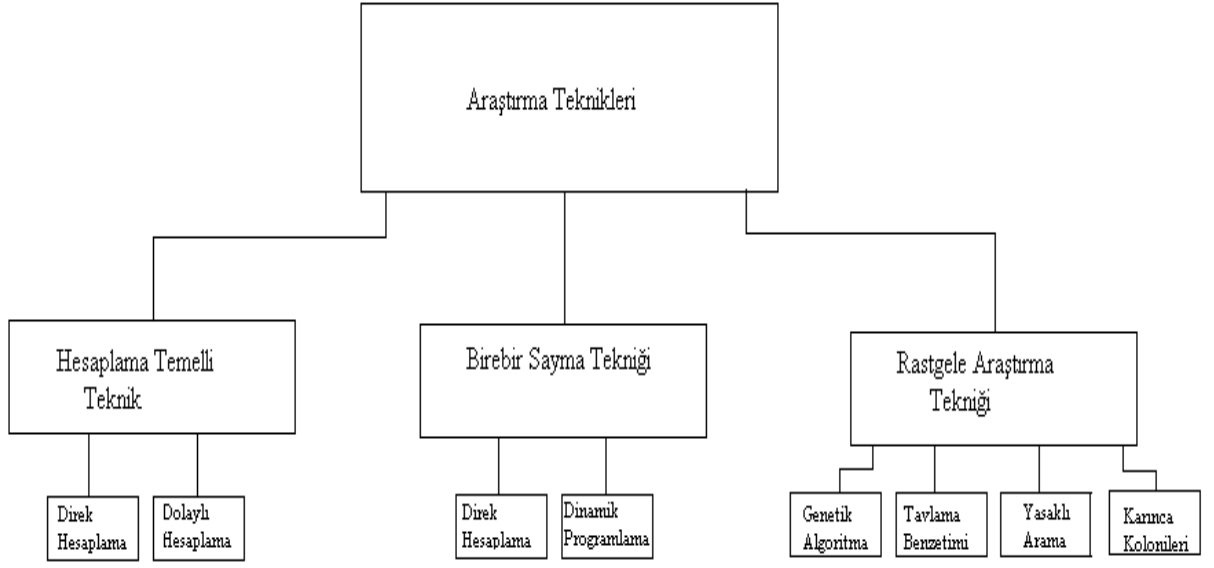
GA, arama ve eniyileme problemlerini çözmekte kullanılan uyarlanabilir bir yöntemdir [28].

GA yapay zekanın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmuştur. Herhangi bir problemin GA ile çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmıştır. Evrimsel hesaplama ilk olarak 1960’larda I.Rechenberg tarafından “Evrimsel stratejileri” isimli eserinden tanıtılmıştır. John Holland evrim sürecinin bir bilgisayar yardımıyla bilgisayara anlayamadığı çözüm yöntemlerinin öğretilabileceğini düşünmüştür. GA böylece John Holland (1975) tarafından bir düşüncenin sonucu olarak bulunmuştur. 1992 yılında John Koza GA’yı kullanarak çeşitli görevleri yerine getiren programlar getirmiştir. Bu metoda Genetik Programlama adını vermiştir [29].

GA uygunluk işlevini yerine getirebilmek için çaprazlama ve değiştirme gibi operatörler kullanır. GA’nın önemli özelliklerinden biri de bir grup üzerinden çözümü araması ve bu sayede çok sayıda çözümün içinden en iyiyi seçmesidir. Goldberg’in tanımına göre GA, rastlantısal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir. GA evrimsel yaklaşım ilkeleri ışığında rastlantısal araştırma yöntemlerini kullanarak kendi keline öğrenme ve karar verme sistemlerinin düzenlenmesini hedef alan bir araştırma tekniğidir [27].

3.6.1. Genetik Algoritmanın Araştırma Teknikleri İçerisindeki Yeri

Bir eniyileme probleminin çözümünde belirli matematiksel ifadeler veya kurallara dayanan algoritmalar kullanılır. Bazı algoritmalar gerçek çözümü bulmak için garanti çözüm sunmazlar. Bunun yerine en iyiye yakın çözümü garanti eden algoritmalar vardır. Bir problemin çözümü için farklı araştırma teknikleri kullanılabilir. Şekil 3.1.’de araştırma teknikleri gösterilmiştir [27].



Şekil 3.1. Araştırma Teknikleri [27]

3.6.2. Genetik Algoritmaları Diğer Yöntemlerden Ayıran En Belirgin Özellikleri

- GA uygulamasında kullanılan operatörler rastlantısal yöntemlere dayanmaktadır. Belirli ve kesin yöntemler kullanılmamıştır [27].
- Önceden bilinen çoklu çözümlerin özelliklerini bir araya getirerek, yeni çözümler üreten bir mekanizmaya sahiptir [30].
- GA aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalınmaz.

3.6.3 Genetik Algoritmaların Uygulama Alanları

- Mekanik Öğrenme
- Eniyileme (Optimizasyon)
- Otomatik Programlama ve Bilgi Sistemleri
- Ekonomik ve Sosyal Sistem Modelleri

3.6.3.1. Mekanik Öğrenme

Mekanik öğrenme; ilki gözlenmiş bir veri takımını anlamak ve yorumlamak, ikincisi de görülmemiş objelerin özelliklerini tahmin etmek olan iki temel amaç için model

kurmayı amaçlar. Parametrik istatistikten ziyade çok büyük veri takımlarının yönetimi üzerinde çalışır. Kullandığı metotların çoğu dağılımdan bağımsız metotlar olarak sınıflandırılabilir. Uygun model seçimi için işe problem hakkındaki varsayımlarla başlamaz. Onun yerine uygun model yapısını belirlemek için doğrudan mevcut veriden hareketle bir araç kutusu yaklaşımı kullanılır. Sınıflama sistemi, GA mekanik öğrenme alanının bir uygulamasıdır. Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir. Bu üretim sistemi, “eğer-sonra” kural yapısını kullanır. Bir üretim kuralı, “eğer” yapısından sonra belirtilen durum için, “sonra” yapısından sonra gelen faaliyetin gerçekleştirilmesini içerir. GA, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmıştır. GA ayrıca, sinir ağlarında ve proteinin yapısal analizinde kullanılmıştır [31].

3.6.3.2. Eniyileme (Optimizasyon)

GA araştırmaların önemli bir bölümü fonksiyon eniyilemesi ile ilgilidir. GA, geleneksel eniyileme tekniklerine göre zor süresiz gürültü içeren fonksiyonları çözmede daha etkindirler. GA'nın uyguladığı diğer bir eniyileme problemi ise, istenen amaçlara ulaşmak üzere, sınırlı kaynakların etkin tahsis edilmesiyle ilgili birleşik eniyileme problemleridir. Gezgin satıcı problemi, araç yön bulma, iş atölyesi çizelgeleme problemi, yerleşim tasarımı problemi, birleşim eniyileme problemlerine örnektir [27].

3.6.3.3. Otomatik Programlama ve Bilgi Sistemleri

Genetik algoritmalar kullanılarak dağıtılmış bilgisayar ağlarının tasarımı gerçekleştirilmiştir. Bu problem tipinde ağ güvenilirlik parametrelerini (çap, ortalama uzaklık ve bilgisayar ağ güvenilirlik gibi) optimize etmek için birden fazla amaç fonksiyonu kullanılmıştır. Genetik algoritmalar ile 100 düğüme kadar olan ağlar başarıyla tasarlanmıştır. Ağ tasarımında genetik algoritmaların kullanılması, tasarım sürelerinin ve maliyetlerinin azalmasında önemli bir katkı sağlamıştır. Özellikle, maksimum miktardaki verinin minimum iletişim hattıyla taşınmasında yüksek bir performans göstermiştir [32].

3.6.3.4. Ekonomik ve Sosyal Sistem Modelleri

Bir sistemi ölçen ve amprik olan değişkenler arasındaki matematiksel ilişkiyi

keşfetme problemi, ekonomide en önemli problemlerden biridir. Bu tip problemler, sembolik sistem tanımlama, kara kutu, veri madenciliği ve modelleme problemi olarak bilinir. Eğer keşfedilen model, sistemin durum değişkenlerinin gelecek değerlerini tahmin etme için kullanılacaksa problem öngörüleme problemi adını alır. Geleneksel doğrusal, kuadratik ve üstsel regresyon modellerinde sapma hataları minimize edilerek fonksiyonlara uygun sayısal katsayılar bulunur. Buradaki yaklaşım, model seçildikten sonra uygun sayısal katsayıların aranmasıdır. Gerçek problem ise verinin değerlendirilmesi için hangi tip modelin uygun olduğuna karar vermektir. Bilgisayarlar herhangi bir matematiksel problemi, formüller ve denklemler aracılığı ile açıklar. Genetik algoritmaların kullanıldığı genetik programlamayla bu tip problemlere tatmin edici çözümler daha kolay bulunmuştur.

Genetik algoritmalar yenilik sürecinin modellenmesi amacıyla da kullanılmıştır. Ayrıca genetik algoritmaların, fiyat verme stratejilerinin gelişim süreçlerini ve kazanç getiren pazarların ortaya çıkış süreçlerini modelleme alanlarında da kullanımları oldukça yaygındır. Genetik algoritmalar sosyal sistemlerin evrimsel yönlerini anlamak amacıyla kullanılmıştır. Bunlara örnek olarak işbirliğinin evrimi, iletişimin evrimi ve karıncalardaki iz takibi davranışının evrimi verilmektedir [31].

3.6.4. Genetik Algoritmanın Aşamaları

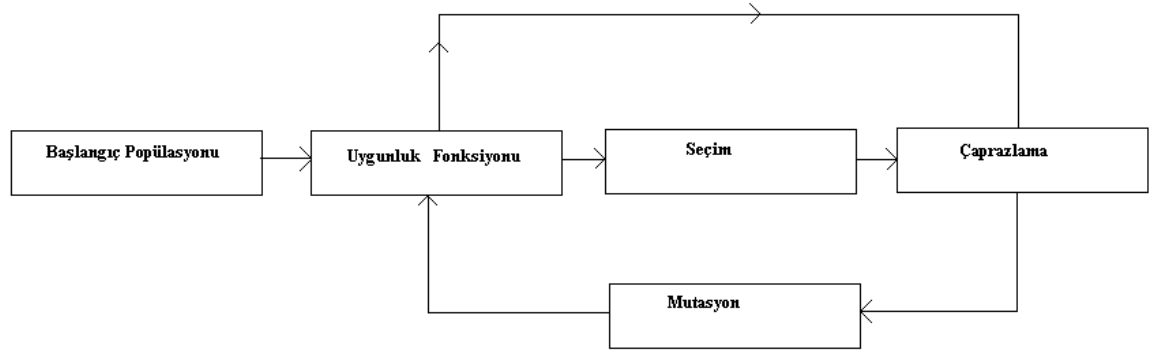
- **Başlangıç popülasyonu:** Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur (çözüm grubu, biyolojideki benzerliği nedeniyle, çözümlerin kodları da kromozom olarak adlandırılır). Toplumda bulunacak birey sayısı için bir standart yoktur. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Toplum bu işlemten sonra rastgele oluşturulur. Birey sayısı belirlendikten sonra probleme bağlı olarak kromozomların kodlanması gerekmektedir. Değişik kodlama biçimleri vardır.
- **Uygunluk Fonksiyonu:** Toplumdaki her kromozomun ne kadar iyi olduğu bilinir. Bu amaçla kullanılan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyon yardımıyla kromozomların uygunluklarının bulunmasına ise evrimleşme adı verilir. Uygunluk fonksiyonu genetik algoritmanın beynini oluşturmuştur. Genetik algortmada probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte ve sonra bu parametrelere göre hesaplamayı yaparak

kromozomların uygunluğunu bulmuştur. Çoğu zaman genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına bağlıdır [33].

- Seçim: İki bireyin uygunluk değerine göre turnuva, rulet tekerleği gibi seçme operatörlerinden problemin yapısına uygun olanın seçilmesi işlemidir [34].
- Çaprazlama: Uygunluk değeri iyi olan bireyler eşleştirilerek bu bireylerden yeni bireyler oluşturulmasına denir. Çaprazlama olasılığı gen çeşitliliğinin sağlanması açısından %60-%90 aralığında seçilir [27].
- Mutasyon: Mutasyon olma olasılığına göre seçilen herhangi bir bireyin kromozomlardaki bir bitin değiştirilme işlemine denir. Mutasyon olasılığı %0.1-%15 aralığında seçilir [27].
- Elitizm: Mevcut topluluktaki uygunluk değeri en iyi olan bireyin olduğu gibi yeni topluluk havuzuna aktarılması işlemidir.
- Yeni topluluk havuzu: Yeni oluşan bireylerin bir havuza alınması, eski bireylerin öldürülerek havuzdan atılması [34].
- Sonuç: Topluluktaki en iyi bireyin temsil ettiği değer istenilen sonucu veriyorsa algoritmanın sona erdirilmesidir. Genetik algoritma nerede duracağını bilmez, istenilen sonuç problemin yapısına göre belirlenmelidir. Bazı sınırlayıcı parametreler (Jenerasyon sayısı gibi) belirlenerek GA istenildiği zaman durdurulabilir. Topluluktaki en iyi bireyin temsil ettiği değer istenilen sonucu vermiyorsa döngü 2. adıma geri dönülerek yeni jenerasyona göre başlanır [34].

3.6.5. Genetik Algoritma Operatörleri

Genetik algoritmalarının tekrarlı karakter taşıdığı görülmüştür. Her tekrarlı üreteci olarak adlandırılır. Şekil 3.2.'de stokastik arama yöntemlerinden olan genetik algoritmaların tekrarlı yapısı gösterilmiştir [33].



Şekil 3.2. Genetik algoritmalarda evrimleşme döngüleri

3.6.5.1. Başlangıç Popülasyonu

Problemin en iyi çözümünü aramak için kullanılan rastgele olarak belirlenmiş başlangıç çözüm setidir. Problemin yapısına ve içeriğine göre değişen sayıda kromozom (başlangıç çözümü) belirlenebilir. Başlangıç popülasyonu olarak da isimlendirilebilecek, belirlediğimiz popülasyon büyüklüğü sayısınca kromozomun rastgele seçilmesiyle oluşturulan başlangıç çözümüdür. GA operatörleri kullanılmadan oluşturulmuş tek popülasyondur. Tamamen tesadüfî olarak seçilmiş olan bir popülasyonla çözüme başlamanın en iyi yol olduğu düşünülmüştür. Genetik süreç içerisinde başlangıç popülasyonu gelişigüzel üretildiğinden, her zaman çözüme ulaşmak yerine bazen yerel çözümde kalınırken, bazen de çözümden uzaklaşma durumu ile karşılaşılabilir. Çünkü üretilen popülasyon gelişigüzel bir yapıya sahiptir ve çözümden uzakta bir araya toplanmış kromozomlardan oluşabilir. Hatta, bütün kromozomları çözümden uzakta bulunabilir. Bu durumda çözüme ulaşmak zor olabilir ve fazla zaman alabilir. Bundan dolayı, bazı durumlarda popülasyondan bu dezavantajlı durumu gidermesi beklenir [35].

3.6.5.2. Uygunluk Fonksiyonu

GA tarafından bulunan çözümlerin uygunluk derecelerinin ölçülmesini sağlayan bir fonksiyondur. Uygunluk fonksiyonu, kromozomların ne kadar iyi bir çözüm verdiğini hesaplayarak gösteren, genetik algoritmanın en temel yapısıdır. Her problem için bir uygunluk fonksiyonun belirlenmesi gerekmektedir. Genetik algoritmada probleme özgü çalışan tek kısım uygunluk fonksiyonudur. Bu fonksiyon, problemin yapısına ve içeriğine göre değişmektedir. Uygunluk fonksiyonu, tasarımın veya parametre değerinin ne kadar iyi

olduğunu gösterir ki, bunun için mantıklı bir uygunluk fonksiyonun modelleme sırasında seçilmesi önemli bir aşamayı temsil eder [31].

3.6.5.3. Seçim

Bu operatör yeni topluluk içerisinde, uygunluğu yüksek olan bireylerin bulunmasını sağlamıştır. Burada bahsedilen uygunluk, Genetik algoritma ile bulunan problemin sonuç değerini göstermektedir. Bireylerin uygunluk değerlerini esas alarak seçim işlemi yapılmıştır. Seçim işlemi topluluk içerisinde uygunlukları düşük olan bireyleri eleyip, elenenler yerine uygunlukları yüksek olan bireylerin birkaç adet kopyasını yaparak tamamlamıştır [36].

3.6.5.4. Çaprazlama

Biyolojik terim olarak çaprazlama genel olarak, bireyler arasındaki benzer alt kromozomların değişimlerini ifade etmektedir. Genetik algoritmada çaprazlama, iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. İki ebeveyn arasında seçilmiş olan genlerin yerleri değiştirilerek çaprazlama işlemi tamamlanmıştır [34].

3.6.5.5. Mutasyon

Mutasyon, organizmada radyasyon ve başka sebeplerle değişiklik meydana gelmesinden esinlenerek geliştirilmiş bir operatördür. Çaprazlama neticesinde farklı çözümlere ulaşmak bazen zor olmaktadır. Mutasyon, yeni çözümleri aramanın kolaylaştırılması ve aramanın yönünü değiştirmek amacı ile bir kromozomun bir elemanın (gen) değiştirilmesi işlemidir. Bir problemin havuzu içinden kaç kromozomun mutasyona uğratılacağını mutasyon oranına göre karar verilmiştir. Kromozomlarda tesadüf olarak belirlenen bir noktada yapılan değişikliktir. Toplulukta çeşitlilik yaratabilmek, çaprazlama sonucunda kaybolabilen iyi özellikleri geri kazanabilmek ve en iyiye ulaşabilmek için bireylerdeki kodlar belli bir olasılık ile değişime (mutasyon) uğratılmıştır. GA'nın bu işleyişi önceden belirlenen bir durdurma koşulu sağlanıncaya kadar devam edilmiştir. GA, çeşitlendirmeye gitmede mutasyondan faydalanır. Mutasyonda genlerden biri rastlantısal olarak değiştirilir. Eğer ikili (0-1) yapıda bir metot tercih edilmişse bu durumda bu

durumda bir genin 0 ise 1, 1 ise 0 yapılması bir mutasyon olacaktır. Genellikle kullanılan mutasyon oranı, bir bireyin gen uzunluğuna bağlıdır. Örneğin 100 gen birimine sahip bir bireyin için oran 0.01'dir. Diğer deyişle rastlantısal olarak düşünüldüğünde her bir genin mutasyona uğrama olasılığı %1'dir. Algoritmanın durması için sağlanması gereken kriter; genellikle belli bir uygunluk değerinin yakalanması veya belirlenen sayıda döngünün sağlanması olarak verilmiştir [31].

3.6.6. Kromozomların Şifrelenmesi

- İkili Kodlama
- Permutasyon Kodlama
- Değer Kodlama
- Ağaç Kodlama

3.6.6.1. İkili Kodlama

İkili kodlama en çok kullanılan yöntemdir. Çünkü ilk genetik algoritma araştırmaları bu kodlama yöntemini kullanmıştır. İkili kodlama, her kromozom bit (0 veya 1) karakter dizilerinden oluşmuştur. İkili kodlama, fazla olasılıkta kromozom verir. Ancak bu yöntem çoğu problem için doğal bir kodlama değildir. Çaprazlama ve mutasyondan sonra düzeltme yapılmıştır [37].

Kromozom	A	100000000111100
Kromozom	B	111110001100101

Şekil 3.3. İkili kodlama için kromozom örneği

3.6.6.2. Permutasyon Kodlama

Bu kodlama, Gezgin Satıcı Problemi (GSP) ve iş sıralama problemleri gibi permutasyon problemlerinde kullanılır. Burada her kromozom numaralar dizisidir.

Kromozom	A	19761978
Kromozom	B	19801991

Şekil 3.4. Permutasyon kodlama için kromozom örneği

Bu problemde olan ikili düzende olan kodlama kullanılabilir. Örneğin, her onluk sayı için ikili düzendeki sayılar yazılır.

0	için	000
1	için	001
2	için	010
...	...	...
7	için	111 gibi.

Bu durumda 8 genden (8 şehirden) oluşan permutasyon kodlamalı kromozomlarımız 24 genden oluşacaktır ve her üç gen bir şehri ifade edecektir [33].

3.6.6.3. Değer Kodlama

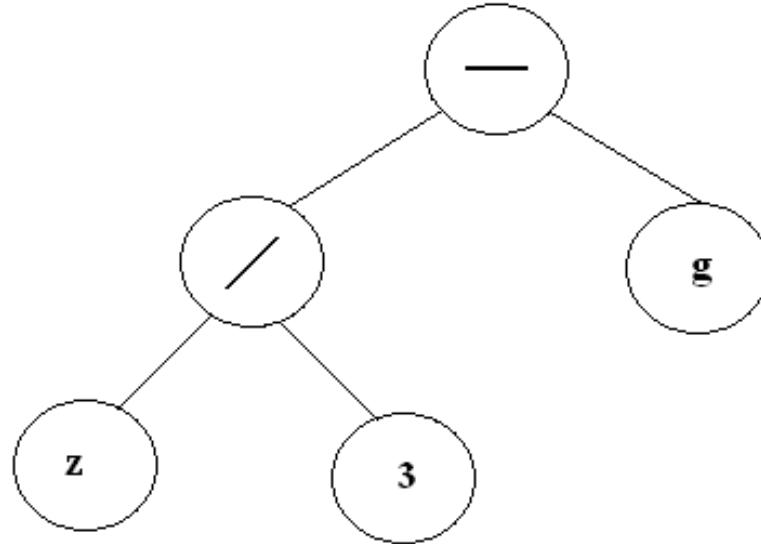
Gerçek sayılar gibi karmaşık değerlerin kullanıldığı problemlerle doğrudan değer kodlama kullanılabilir. İkili kodlamanın bu tip problemler için kullanılması, problemin zorlaşmasına neden olacaktır. Değer kodlamada, her kromozom bazı değerlere eşittir. Vereceğimiz değer, problemle ilgili herhangi bir şey olabilir. Bunlar sayı, karakter veya herhangi bir nesneden de oluşabilir [37].

Kromozom	A	1.959 1.944 1980 1976
Kromozom	B	TUBATANYILDIZI
Kromozom	C	(sağ), (sol),(geri),(ileri)

Şekil 3.5. Değer kodlama için kromozom örneği

3.6.6.4. Ağaç Kodlama

Ağaç kodlama genellikle genetik programlamada programlar ve ifadeler oluşturmak için kullanılır. Ağaç kodlamada her kromozom, adından da anlaşılacağı gibi nesneler ve nesneler arası işlemleri içeren bir ağaç yapısından oluşur.



Şekil 3.6. Ağaç kodlama için kromozom örneği

Ağaç kodlama, program geliştirmek için uygundur. Örneğin, LISP ve Prolog gibi programlama dillerinde ağaçtan yapısal olarak sık bir biçimde kullanılır. Ağaç kodlamada

çaprazlama ve mutasyon çok kolay bir şekilde uygulanabilir [33].

3.6.7. Çaprazlama ve Mutasyon

3.6.7.1. İkili Kodlanmış Kromozomlarda Çaprazlama

- Tek Noktalı Çaprazlama: İki kromozom için tek bir çaprazlama noktası seçilir. Kromozom-1'in ilk başından, çaprazlama noktasına kadar olan genler ve kromozom-2'nin çaprazlama noktasından, sonuna kadar olan genler alınır ve döl-1 oluşur. Kromozom-1'in çaprazlama noktasından sonuna kadar olan genler ve kromozom-2'nin ilk başından çaprazlama noktasına kadar olan genler alınır ve döl-2 alınır.

Kromozom-1: **110001**| 00100110000
Kromozom-2: 1101111| **11000011111**
Döl-1: **1100011**| **11000011111**
Döl-2: 1101111|00100110000

- Çift Noktalı Çaprazlama: İki kromozom için iki çaprazlama noktası seçilir. Kromozom-1'in ilk başından ilk çaprazlama noktasına kadar olan genler, kromozom-2'nin 1.çaprazlama noktasının sonundan ikinci çaprazlamanın başına kadar olan genler ve kromozom-1'in ikinci çaprazlama noktasından sonuna kadar olan genler alınır ve yeni döl oluşur.

Kromozom-1: **11**|0011|**11**
Kromozom-2: 11|**0111**|00
Yeni Döl : 11|0111|11

- Tek biçimli Çaprazlama: Yeni döl oluşturmak için rastgele çaprazlama noktaları seçilir ve genler çaprazlama noktasından itibaren alınır.

Kromozom-1: **1**|10|**011**|11
Kromozom-2: 0|**10**|000|**10**
Yeni Döl: 11001110

- Aritmetik Çaprazlama: Yeni bir döl oluşturmak için aritmetik olarak gen takası

yapılır.

$$11001011+11011111=11001001 \text{ VE işlemi}$$

3.6.7.2. İkili Kodlanmış Kromozomlarda Mutasyon

Mutasyon işlemi kromozomların genlerinin rastgele olarak değiştirme işlemidir. Değiştirme işlemi olarak 0 olan biti 1, 1 olan biti ise 0 olarak değiştirilebilir.

Orijinal döl: 1101111100011010

Mutasyona uğramış döl: 1101111000011110

3.6.7.3. Permutasyon Kodlanmış Kromozomlarda Çaprazlama Ve Mutasyon

Çaprazlama: Permutasyon kodlanmış iki kromozomun belli bir çaprazlama noktası seçilir. Kromozom-1'in ilk başından çaprazlama noktasına kadar olan kısmı, kromozom-2'in çaprazlama noktasından sonuna kadar olan kısmı alınır.

Kromozom-1: 12765|6789

Kromozom-2: 31051|1980

Yeni Döl: 127651980

Böylece yeni dölde aynı rakamlar oluşmuştur. Bu yüzden aynı olan rakamları olmayanlarla değiştirmemiz gerekir.

Yeni Döl1: 127653980

- **Mutasyon:** Kromozomda iki gen seçilir ve yerleri değiştirilir.

Kromozom-1: 24681357

Yeni Döl: 27681354

3.6.7.4. Değer Kodlanmış Kromozomlarda Çaprazlama Ve Mutasyon

- **Çaprazlama:** Kromozomların çaprazlama yöntemi kullanılırken ikili kodlamada kullanılan tek noktalı çaprazlama, çift noktalı çaprazlama, tek biçimli çaprazlama,

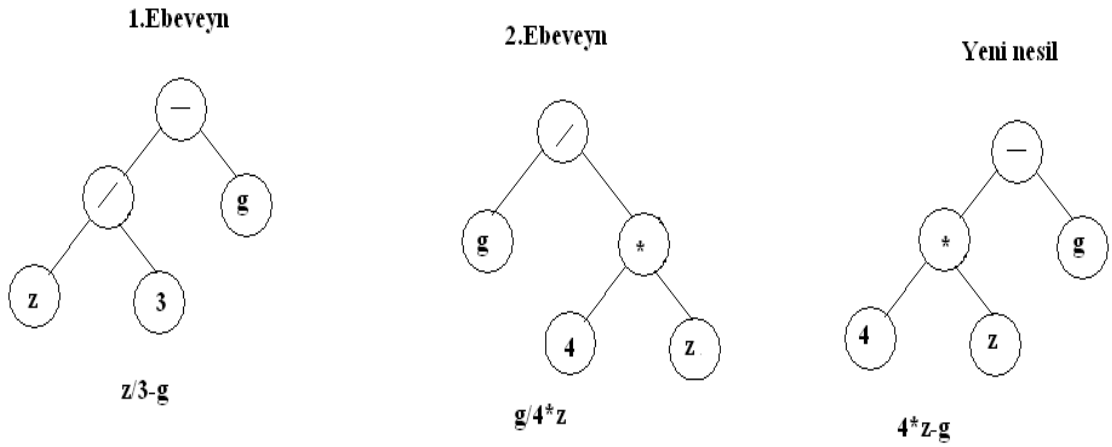
aritmetik çaprazlama yöntemleri kullanılır.

- **Mutasyon:** Kromozomun seçilen genden bir sayının eklenmesi veya çıkartılması ile yeni bir döl oluşur.

$$(1.29 \ 5.68 \ 2.86 \ 4.11 \ 5.55) = (1.29 \ 5.68 \ 2.86 \ 4.00 \ 5.44)$$

3.6.7.5. Ağaç Kodlanmış Kromozomlarda Çaprazlama Ve Mutasyon

- **Çaprazlama:** Her iki ebeveynden birer takas noktası seçilerek kromozom parçalanır. Bu noktaların altındaki kısımlar, aralarında değiştirilerek yeni bireyler oluşturulur [33].



Şekil 3.7. Ağaç kodlamada çaprazlama operatörünün kullanımı

- **Mutasyon:** Seçilen düğümlerdeki numaralar veya işlemler değiştirilir [33].

3.6.8. Genetik Algoritmanın Parametreleri

- Çaprazlama Olasılığı
- Mutasyon Olasılığı

- Diğer Parametreler

3.6.8.1. Çaprazlama Olasılığı

Çaprazlama olasılığı çaprazlamanın hangi sıklıkla olacağını belirtir. Çaprazlama olasılığı %100 ise çaprazlama yapıldıktan sonra oluşacak, yeni döl ile eski döl tamamen farklıdır. Çaprazlama olasılığı yüksek ise iyi bireylerin özelliklerini kaybetme olasılığı yüksektir. Ayrıca çaprazlama olasılığı yüksek olursa problem uzayı çok büyüyecektir. Çaprazlama olasılığı %0 ise çaprazlama yapıldıktan sonra bireyin özelliklerinin hiçbirini değişmemiş ve eskisiyle tamamen aynı olması anlamına gelir. Çaprazlama olasılığı düşük ise en iyi bireyi bulma işlemi zorlaşacaktır. Çaprazlamanın amacı iyi bireylerin kromozomlarını birleştirerek en iyi dölü bulmaktır. Çaprazlama olasılığı genellikle 0.75-0.90 arasındadır.

3.6.8.2. Mutasyon Olasılığı

Mutasyonun hangi sıklıkla olacağını belirtir. Mutasyonun %100 olması dölün bütün kromozomlarının değişmesidir. Mutasyonun %0 olması ise döldeki hiçbir kromozomun değişmemesi anlamına gelir. Mutasyon olasılığının yüksek olması probleme rastgelelik kazandırır ve yerel çözüme takılmayı engeller. Mutasyon olasılığının düşük olması da problem uzayının tamamen araştırılması engellenir. Mutasyon olasılığı da %0.001 'dir. Mutasyon döldeki her kromozoma da yapılabilir. Mutasyonun amacı problem uzayının çeşitliliğini artırmaktır ve yeni dölleri oluşturmaktır. Ayrıca problem uzayının yerel çözümlere takılmasını engeller.

3.6.8.3. Diğer Parametreler

Çaprazlama ve mutasyon oranlarının dışında diğer önemli parametrelerden biri popülasyon büyüklüğüdür. Popülasyon büyüklüğü, popülasyonda kaç kromozom olduğunu ifade eder. Popülasyon büyüklüğünün de optimum değerlerden uzak olması algoritmanın performansını etkilemiştir. Nesildeki birey sayısı az ise çözüm uzayı yeterince araştırılmadığından yerel optimum noktaya yakalanma olasılığı da düşüktür. Bu durumda çaprazlama işlemi için yalnız birkaç olasılık olduğundan problem alanının sadece küçük bir kısmı araştırılabilecektir. Bu sayı aşırı arttığında ise neslin evrimleşmesi oldukça uzun

zaman içerisinde gerçekleştirilebildiğinden algoritma yavaşlayacaktır. Araştırmalar popülasyon boyutu için, şifrelenmeye bağlı olarak, bir üst sınırın belirlenmesi gerektiğini göstermiştir. Bu değerin üstüne çıkıldığında ise problemlerin çözümünde bir iyileşme görülmeyecektir. Goldberg, ikili kodlamalı durumlarda (P) popülasyon büyüklüğü ile (Ch) kromozom uzunluğu arasında aşağıdaki bağıntıyı önermiştir.

$$P = 1.65 \cdot 2^{0.2Ch} \quad (3.1)$$

Yapay zekada kullanılan optimizasyon yöntemlerinden olan GA parametrelerinin kendilerinin de optimizasyona gerek duyduğu görülmüştür. Bu parametrelerle ilgili önerilen bazı değerler aşağıdaki tabloda verilmiştir.

Tablo 3.1. Çeşitli GA parametrelerinin değerlendirilmesi [33]

Parametreler	Negnevitsky	D.Jong	Michalewicz	Schaffer	Grefenstette
Popülasyon büyüklüğü	50	50-100	50-100	20-30	30
Çaprazlama Oranı	0,7	0,6	0,5-1,0	0,75-0,95	0,95
Mutasyon	0,001-0,01	0,001	0,001-0,01	0,005-0,01	0,01

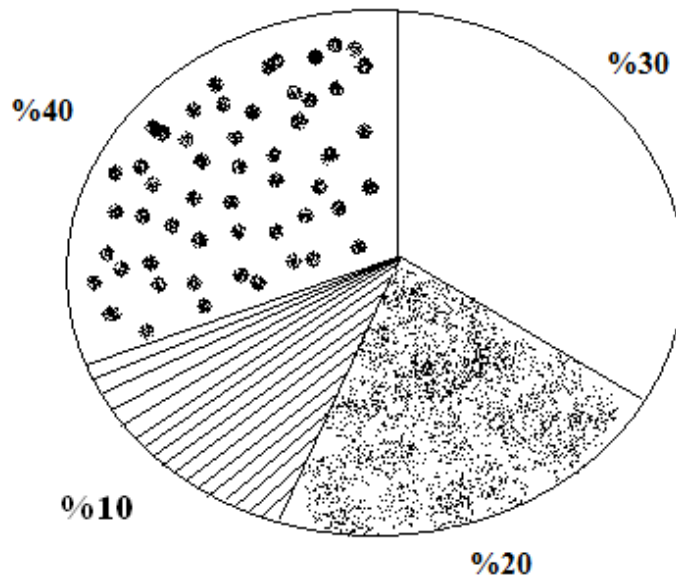
GA parametrelerinin ilgili farklı değerlendirilmelerin olduğu Tablo 3.1. 'de görülmüştür. Dolayısıyla problemlerin niteliğine uygun olarak kontrol parametrelerinin uyarlanması gerekmiştir. Diğer yandan operatörlerin de jenerasyona bağlı olarak adaptasyonu önemli etmenlerdendir [33].

3.6.9. Genetik Algoritmada Seçim

- Rulet Tekeri Seçimi
- Sıralama Seçimi
- Sabit Durum Seçimi
- Turnuva Seçimi

3.6.9.1. Rulet Tekeri Seçimi

Rulet tekeri seçimi uygunluk değerine göre yapılan seçme işlemine rulet tekeri seçimi denir. Rulet tekerleği, tekerleğe benzediği için rulet tekerleği denilmiştir. Bu yöntemde her bir kromozomun uygunluk değeri hesaplanır. Sonra hepsi toplanır ve her bir kromozomun uygunluk değerine bölünür. Böylece hepsinin seçilme oranları bulunmuş olur. Her birinin seçilme oranlarına göre rulet tekerleğine yerleştirilir. Rulet tekerleğine yerleştirilen bireylerin, seçilme oranlarına göre rulet tekerleğinde yer alacaktır. Rulet tekerleği çevirilir. Okun durduğu noktadaki birey (kromozom) seçilir. Rulet tekerleğinde seçilme oranı yüksek olan bireyin seçilme şansı yüksektir. Bu yöntemin avantajı uygunluk değeri yüksek olan bireyin yani en iyi bireyin seçilme olasılığının artmasıdır. Bu yöntemin dezavantajı ise yerel çözümlere takılmasıdır.



Şekil 3.8. Rulet seçimi

3.6.9.2. Sıralama Seçimi

Sıralı seçimi de, uygunluk değerine göre popülasyondaki bireyler sıralanır. Uygunluk değeri en küçük olan kromozom en kötü kromozomdur. Bu yöntemde bütün kromozomların seçilme şansı olacaktır. Fakat en iyi çözüme ulaşma zamanı uzayacaktır.

3.6.9.3. Sabit Durum Seçimi

Bu yöntemde göre ebeveynlerin seçimi için kromozomların büyük parçaları bir sonraki jenerasyona taşınmalıdır. GA da her nesilde yeni bir döl oluşturmak için birkaç kromozom seçilir (büyük uygunlukta iyi olanlar). Daha sonra az uygunlukta kötü olan bazı kromozomlar atılır ve yeni döl onların yerine getirilir. Popülasyonun geri kalanı değiştirilmeden yeni nesle aktarılır [33].

3.6.9.4. Turnuva Seçimi

Bu yöntemde iki tane kromozom (birey) karşılaştırılır ve uygunluk değeri iyi olan birey seçilir, diğer birey ise atılır. Bu işlem popülasyondaki bütün bireyler için yapılır. Böylece

bir yarışma gibi karşılaştırma işlemi bittikten sonra en iyi birey seçilmiş olur.

3.7. Parçacık Sürü Optimizasyonu

Parçacık Sürü optimizasyonu (PSO), kuş ve balık sürülerinin iki boyutlu hareketlerinden esinlenerek ilk olarak 1995 ve 1996 yıllarında Kennedy ve Eberhart tarafından geliştirilmiştir. PSO bireyler arasındaki sosyal bilgi paylaşımını esas alır. PSO’da arama işlemi genetik algoritmada olduğu gibi popülasyondaki bireyler tarafından ve belirlenen nesil sayısınca yapılır. Her bireye parçacık denir ve parçacıklardan oluşan popülasyona da sürü ismi verilir. Her bir parçacık kendi pozisyonunu, bir önceki tecrübesinden yararlanarak bir önceki sürüdeki en iyi pozisyona doğru ayarlar [38].

PSO kuş veya balık sürüsünün davranışlarını benzetir. Bir alanda rastgele yiyecek arayan bir kuş grubunun olduğunu ve arama yapılan alanda yalnızca bir parça yiyecek olduğunu varsayalım. Kuşların hiçbiri yiyeceğin nerede olduğunu bilmesin. Fakat her bir iterasyon sonunda yiyeceğin ne kadar uzakta olduğunu bilsin. Bu durumda en iyi strateji nedir? En etkili olan yiyeceğe en yakın olan kuşu takip etmektir. PSO bu senaryoya göre çalışır ve optimizasyon problemlerini çözmek için kullanılır. PSO’da her bir çözüm arama uzayındaki bir kuştur ve bunlar bir “parçacık” olarak isimlendirilir. Tüm parçaların, optimize edilecek uygunluk fonksiyonu tarafından değerlendirilen bir uygunluk değeri ve uçuşlarını yönlendiren hız bilgileri vardır. Parçacıklar problem uzayında mevcut optimum parçacıkları takip eder. PSO bir grup rastgele üretilmiş çözümle başlatılır ve jenerasyonlar

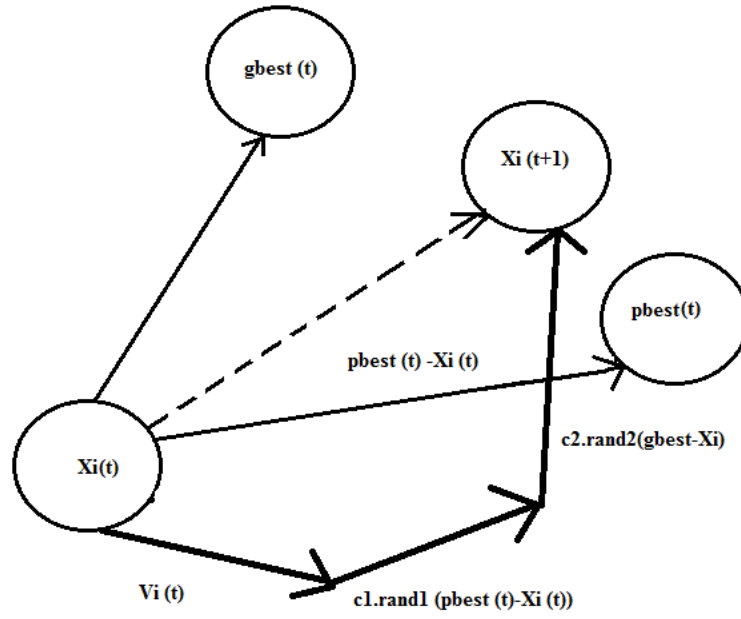
güncellenerek en uygun değer araştırılır. Her iterasyonda, her bir parçacık iki “en iyi” değere göre güncellenir. Bunlardan birincisi bir parçacığın o ana kadar bulduğu en iyi uygunluk değeridir. Ayrıca bu değer daha sonra kullanılmak üzere hafızada tutulur ve “pbest” yani parçacığın en iyi değeri olarak isimlendirilir. Diğer en iyi değer ise popülasyondaki herhangi bir parçacık tarafından o ana kadar elde edilmiş en iyi uygunluk değerine sahip çözümdür. Bu değer popülasyon için global en iyi değerdir ve “gbest” olarak isimlendirilir [39].

Her bir parçacığın konumu $x_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ olarak ifade edilir. Önceki en iyi uygunluk değerini veren i'ninci parçacığın pozisyonu $pbest_i = [p_{i1}, p_{i2}, \dots, p_{iD}]$ olarak ifade edilir. Gbest ise her iterasyonda tüm parçacıklar için tektir ve $gbest = [p_1, p_2, \dots, p_D]$ şeklinde gösterilir. i'ninci parçacığın hızı $v_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ olarak ifade edilir. İki en iyi değer bulunmasından sonra parçacık hızları ve konumları aşağıda verilmiştir.

$$v_i^{k+1} = v_i^k + c_1 \cdot rand_1^k \cdot (pbest_i^k - x_i^k) + c_2 \cdot rand_2^k \cdot (gbest^k - x_i^k) \quad (3.2)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (3.3)$$

Denklemden c_1 ve c_2 öğrenme faktörleridir. c_1 ve c_2 her parçacığın pbest ve gbest pozisyonlarına doğru çeken, hızlanma terimlerini ifade eden sabitleridir. c_1 , parçacığın kendi tecrübelerine göre hareket etmesini, c_2 ise süredeki diğer parçacıkların tecrübelerine göre hareket etmesini sağlar [40]. Bir parçacığın pozisyon güncelleme denklemine göre hareketi Şekil 3.9.'da verilmiştir.



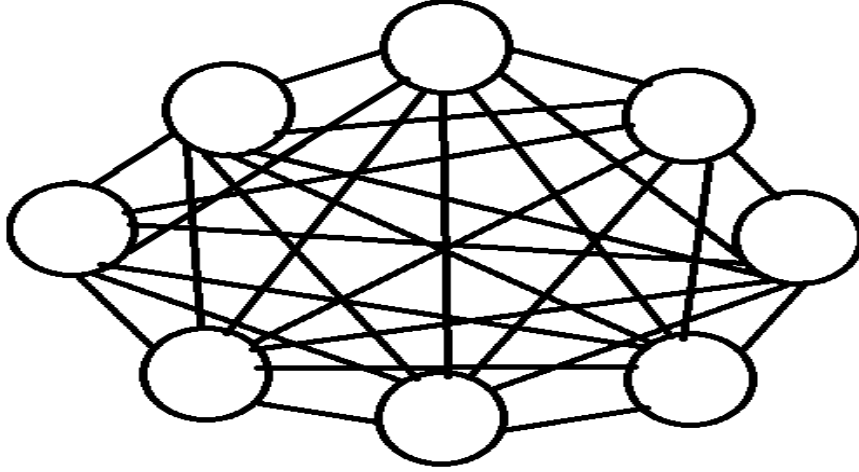
Şekil 3.9. Parçacığın hareketi

3.7.1. Sosyal Ağ Yapıları

PSO'yu yönlendiren özellik sosyal etkileşimdir. Sürü içerisindeki parçacıklar bilgiyi birbirlerinden öğrenirler ve elde edilen bilgiye dayalı olarak komşularına daha çok benzeyecek şekilde hareket ederler. PSO için sosyal yapı, çakışan komşulukların oluşturulması ile belirlenir ve bu komşuluk yapısı içerisindeki parçacıklar birbirini etkiler. PSO'nun performansı sosyal ağın yapısına oldukça bağlıdır. PSO için farklı sosyal ağ yapıları geliştirilmiştir. Örneğin yıldız, halka, teker, piramit, dört küme ve Von Neumann sosyal yapıları gibi. Kennedy, büyük komşuluğa sahip PSO'nun basit; küçük komşuluklu PSO'nun ise karmaşık problemler üzerinde daha iyi performans göstereceğini belirtmiştir [41].

3.7.1.1. Yıldız Yapı

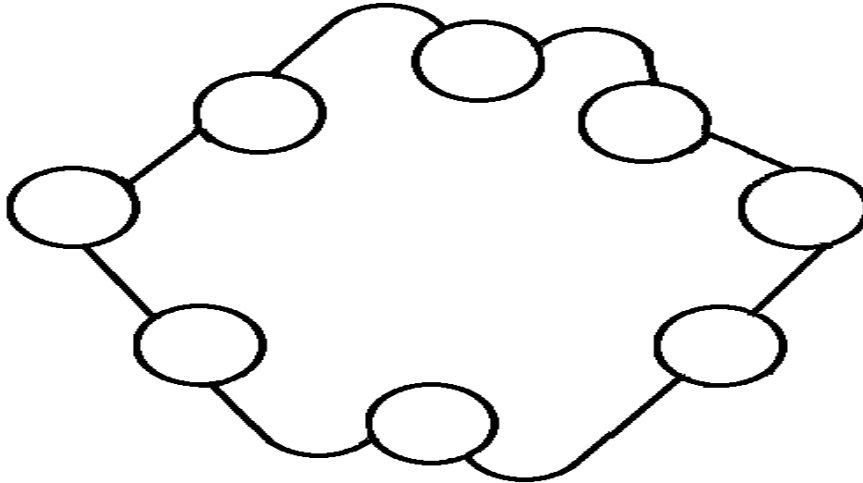
Yıldız yapı şekil 3.9 'da görüldüğü gibi tüm parçacıklar birbiriyle bağlantı halindedir. Yani, her parçacık tüm sürü tarafından bulunan en iyi çözüme doğru çekilir. PSO'nun ilk uygulaması yıldız ağ yapısını kullanmış olup bu yapı gbest-PSO'dur. Yıldız yapıda gbest-PSO, diğer ağ yapılarına göre çok daha hızlı yakınsayabilmekte ancak bölgesel minimuma takılma ihtimali de artmaktadır [42].



Şekil 3.10. Yıldız topolojisi (Tam bağlantılı graf)

3.7.1.2. Halka Yapı

Halka yapı şekil 3.10.'da görüldüğü gibi her parçacık hemen kendi civarındaki komşularla iletişim halindedir. Komşuluğun iki olduğu durumda her parçacık en yakındaki komşularla iletişim halindedir. Her parçacık komşuluğunda bulunan en iyi çözüme doğru hareket etmek suretiyle ona benzemeye çalışır. Bu davranış, halka yapının çok modlu problemlerde yıldız yapıdan daha iyi performans sergilenmesini sağlar. Bu PSO algoritması genellikle İbest-PSO olarak adlandırılmıştır.



Şekil 3.11. Halka Topolojisi

3.7.2. PSO Algoritma Kodu

Her parçacık için:

Parçacığı başlangıç için ayarla.

Bitir

Her parçacık için:

Uyum değerini hesapla.

Eğer

Uyum değeri o ana kadar bulunan en iyi değerden daha iyi ise bulunan değeri en iyi değer olarak ayarla.

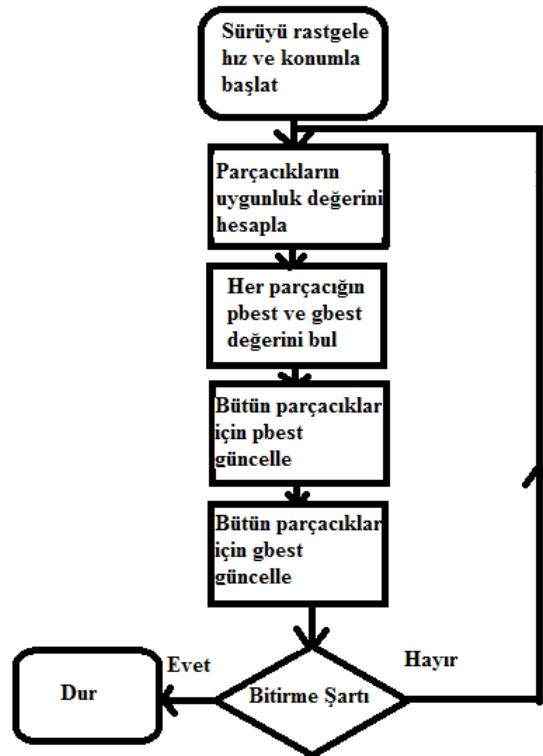
Bitir

Tüm parçacıklar arasında en iyi değere sahip olanı tüm parçacıkların global en iyisi seç.

Her parçacık için:

Hız ve konumunu denklem (3.2)'e göre ayarla. Güncellemeyi ise denklem (3.3)'e göre ayarla.

Bitir [41].



Şekil 3.12. PSO Akış Diyagramı

3.7.3. PSO Parametre Kontrolü

PSO da ayarlanması gereken çok fazla sayıda parametre yoktur.

Parçacık sayısı: 20 ile 40 arasındadır. Birçok problem için 10 parçacık kullanmak yeterlidir. Bazı zor veya özel problemlerde ise 100 veya 200 parçacık kullanılması gerekebilir.

Parçacık boyutu: Optimize edilecek probleme göre değişebilir.

Parçacık aralığı: Optimize edilecek probleme göre değişmekle birlikte farklı boyutlarda ve aralıklarda parçacıklar tanımlanabilir [40].

Pozisyon vektörü: X_i^k ile ifade edilir. Problem boyutu kadar eleman içerir. Pozisyon vektörü sürü içerisinde gösterimi $X_i^k = \{x_{i1}^k, x_{i2}^k, \dots, x_{i(par.boyutu)}^k\}$ şeklindedir. Burada yer alan x_{i1}^k , k. iterasyonda i.parçacığın pozisyon vektöründeki birinci elemanını gösterir.

Hız vektörü: Parçacığın bir sonraki konumunu belirleyen parametrelerinden bir olan hız vektörü V_i^k ile ifade edilir. Parçacığın hız vektörü sürüsü içerisinde, burada yer alan V_{i1}^k , k.iterasyonda i. parçacığın hız vektöründeki birinci elemanını gösterir [38].

Öğrenme Faktörleri: c_1 ve c_2 genellikle 2 olarak seçilir. Fakat farklı da seçilebilir. Genellikle c_1 , c_2 ye eşit ve $[0,4]$ aralığında seçilir [40].

Durma Koşulu: Bir durdurma kriteri seçilirken iki önemli noktaya dikkat edilmiştir. Bunlardan ilk durdurma şartı, PSO'nun erken yakınsamasına sebep olmamalıdır. Yoksa bu durum bölgesel optimum çözümlerin bulunmasına neden olacaktır. İkincisi ise işlem karmaşıklığına neden olmamalıdır. Aşağıda verilen şartlardan biri durdurma kriteri olarak kullanılabilir.

- Önceden belirlenen maksimum iterasyon sayısına erişilmişse
- Kabul edilebilir bir çözüm bulunmuşsa
- Belirli sayıdaki iterasyonlar boyunca ve herhangi bir gelişim gözlenmiyorsa
- Normalize edilmiş sürü yarıçapı sıfıra doğru yaklaşmışsa
- Amaç fonksiyonun eğimi sıfıra yaklaşmışsa sistem durdurulabilir [42].

3.7.4. PSO'nun Genetik Algoritma ile Karşılaştırılması

- PSO'da çaprazlama ve mutasyon gibi evrimsel operatörler yoktur. Genetik algoritmanın ise çatı noktası çaprazlama ve mutasyondur.
- PSO'da parçacık denilen potansiyel çözümler, mevcut en iyi çözümleri takip ederek problem uzayında gezinirler. Genetik algoritma da ise çaprazlama ve mutasyon operatöründen sonra uygunluk değerlerine göre en iyi çözüm bulunur.
- PSO'da genetik algoritma gibi uygunluk fonksiyonuna gerek duyar.
- Başlangıç popülasyonu her iki algortmada rastgele seçilir.
- PSO'nun gerçekleşmesi genetik algortmaya göre daha kolaydır.
- PSO'nun ayarlanması gereken çok az parametre vardır [39].

3.8. Yapay Arı Koloni Algoritması (ABC)

Yapay arı koloni algoritması, bal arısı sürülerinin kendilerine özgü zeki davranışlarını örnek alarak, arıların besin ararken kullandıkları yöntemlerden esinlenerek oluşturulmuş bir optimizasyon algoritmasıdır. Sürü zekâsına dayanan bu algoritma, doğada sürü halinde hareket eden arıların besin bulmada sergilemiş oldukları davranışları temel olarak optimizasyon problemlerini çözmek için kullanılmıştır.

Sürü zekası; karıncalar, kuşlar, arılar, balıklar gibi sosyal bir toplulukta yaşayan bireylerin davranışlarını örnek alarak, problemlere çözüm getirmeyi amaçlayan bir Yapay zeka tekniğidir. Karıncaların gitmiş oldukları yollara kimyasal madde bırakarak diğer karıncalara bilgi iletmeleri, kuş ve balık sürülerinin birlikte hareket ederken konum ve hızlarını ayarlayarak ilerlemeleri, arı kolonilerinin kovan etrafında yapmış olduğu salınım hareketiyle birbirlerini bilgi iletmeleri sürü zekasının ortaya çıkmasında temel alınan zeki davranışlardır [43].

Doğal bir arı kolonisinde yapılacak işler o iş için özelleşmiş arılar tarafından yapılır. Yani, yapılacak işlere göre arılar arasında bir iş bölümü vardır ve herhangi bir merkezi otorite olmadan bu iş dağılımını gerçekleştirdikleri için kendi kendine organize olabilmektedir. İş bölümü yapabilme ve kendi kendine organize olabilme sürü zekâsının iki önemli özelliğidir. Tereshko'nun reaktif difüzyon denklemine dayalı olarak önerdiği kollektif zekânın ortaya çıkmasını sağlayan minimal yiyecek arama modelinden

esinlenmiştir [44]. ABC algoritması araştırma esnasında 4 farklı seleksiyon işlemi kullanılmıştır.

- Diğer kaynaklara göre daha iyi kaynakların belirlenmesine yönelik çalışmaların yapıldığı küresel seleksiyon.
- Görevli ve gözcü arıların renk, şekil, koku gibi nektar kaynağının türünün belirlenmesi ve sonra görsel bilgiyi kullanarak bir bölgeye eriştikten sonra o bölgedeki bir kaynağa karar verilmesine vesile olan bölgesel olasılık tabanlı seleksiyon.
- İşçi ve gözcü arıların daha iyi olan kaynağı belirlemesi amacıyla kullandıkları aç gözlü seleksiyon işlemi.
- Kaşif arılar tarafından gerçekleştirilen rastgele seleksiyon işlemi.

Bütün bu seleksiyon metotların bir arada kullanılmasıyla ABC algoritması hem iyi bir küresel araştırma hem de bölgesel araştırma yapılabilmiştir [42].

3.8.1. Arılar

ABC algoritmasının temel aldığı minimal modelde üç çeşit yapay arımız bulunuyor. Bunlar işçi arılar, gözcü arılar ve kaşif arılar. İşçi arılar komşuluk prensibine dayanarak daha fazla nektarın olduğu besin kaynaklarını araştırırlar. Her bir besin kaynağında bir işçi arı vardır. Dolayısıyla işçi arı sayısı besin kaynağın sayısına eşittir. Gözcü arılar kovanda bekler. Arıların besin kaynakları ile bilgileri dansla paylaştıktan sonra nektarın fazla olduğu besin kaynağına yönelirler. Kaşif arılar yiyecek arama sürecinin başlangıcında rastgele dağılarak yiyecek aramaya başlarlar [45]. Arıların organize olabilmeleri aşağıdaki adımlara bağlıdır.

- **Pozitif Geri besleme:** Yiyecek kaynaklarındaki nektar miktarı gözcü arıların onları ziyaret sayısı ile değişir.
- **Negatif Geri besleme:** Bir yiyecek kaynağının araştırma sürecinin bitirilmesidir.
- **Dalgalanmalar:** Kâşifler, yeni yiyecek kaynaklarının keşfi için rastgele bir arama sürecini tamamlar.
- **Çoklu Etkileşim:** Arılar dans alanında yiyecek kaynağı pozisyonları hakkında bilgilerini paylaşırlar [46-47].

3.8.2. ABC Algoritmasının Özellikleri

- Basit ve esnek bir algoritmadır.
- Arıların gerçek yiyecek arama davranışlarına çok yakın olarak tasarlanmıştır.
- Sürü zekasına dayalıdır.
- Başlangıçta nümerik problemler için tasarlanmış ancak ilerleyen süreçte gerçek tasarım problemler için kullanılmıştır.
- Kontrol parametresi azdır [48].

3.8.3. ABC Algoritmasının Temel Adımları

- 1: Tüm Başlangıç değerlerinin atanması ve çözüm geliştirmeme sayaçların sıfırlanması (failure=0)
- 2: $f_i = f(x_i)$ fonksiyon değerlerinin ve bu değerlere karşılık gelen uygunluk (fitness) değerlerinin, fitness, hesaplanması
- 3: repeat
- 4: for $i=1$ to SN do
- 5: x_i çözümünün görevli arısı için yeni bir kaynak üret, uygunluk değerini hesapla.
- 6: v_i ve x_i arasında aç gözlü seleksiyon işlemi uygula ve daha iyi olanı seç.
- 7: x_i çözümü gelişmemişse sayacı bir artır (failure=failure+1). Gelişmişse sıfırla (failure=0).
- 8: end for.
- 9: Gözcü arıların seçim yaparken kullanacakları uygunluk değerine dayalı olasılık değerlerini, p_i , hesapla.
- 10: $t=0$, $i=1$
- 11: repeat
- 12: if random < p_i then
- 13: Gözcü arı için yeni bir kaynak, v_i üret.
- 14: v_i ve x_i arasında aç gözlü seleksiyon işlemi uygula ve daha iyi olanı seç
- 15: x_i çözümü gelişmemişse sayacı bir artır (failure=failure+1). Gelişmişse sıfırla (failure=0).

16: $t=t+1$
17: endi f
18: until $t=SN$
19: If $\max(\text{failure}) > \text{limit}$ then
20: x_i 'yi üretilen rastgele bir çözümle değiştir
21: end if
22: En iyi çözümü hafızada tut.
23: until Durma kriteri [48].

4. MATLAB'DA UYGULAMALAR

4.1. Filtrelerin Katsayılarının Bulunması

Bu çalışmada FIR, Butterworth IIR, Chebyshev IIR ve Eliptik IIR filtrenin katsayılarının farklı yöntemlerle bulunması amaçlanmıştır. Günümüzde en çok kullanılan yöntemlerden biri olan genetik algoritma bu çalışmada filtre katsayılarının bulunması için tercih edilmiştir. İstenen özelliklere göre filtre katsayıları hesaplanmıştır. Uygunluk değeri en iyi olan birey Rulet tekerleğine göre seçilmiştir. En iyi filtrenin katsayıları bulunmuş ve katsayıların gerçek değeri ile tahmin edilen değeri karşılaştırılmıştır.

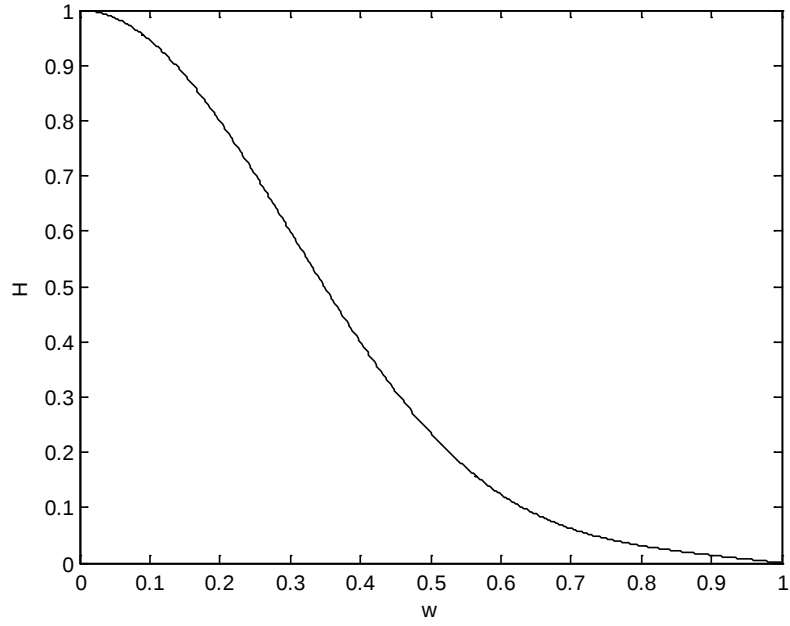
İkinci yöntem olarak parçacık sürü optimizasyonu kullanılmıştır. Parçacık sürü optimizasyonu ile ilk olarak başlangıç popülasyonu oluşturulmuştur. Popülasyonun uygunluk değerleri ve popülasyondaki elemanların katsayıları hesaplanmıştır. Parçacığın en iyi değeri olan pbest değeri ve popülasyonun en iyi değeri olan gbest değeri tespit edilmiştir. PSO algoritması ile yeni çözümler üretilmiştir. Uygunluk değerleri bulunmuştur. En iyi uygunluk değeri olan sisteme göre örnekleme yapılmıştır. Örneklenen değerlere göre katsayılar bulunmuştur.

Üçüncü yöntem olarak ABC algoritması kullanılmıştır. Başlangıç popülasyonu yapay arı koloni algoritmasına göre oluşturulmuştur. Başlangıç popülasyonun uygunluk değerleri ve filtre katsayıları bulunmuştur. Yapay arı koloni algoritmasına göre yeni çözümler üretilmiştir. Yeni çözümlere aç gözlü seleksiyon işlemi uygulanmıştır. Uygunluk değeri hesaplanmıştır. En iyi uygunluk değeri olan sistemde 30 noktada örnekleme yapılmış ve filtre katsayıları hesaplanmıştır.

4.1.1. FIR Filtrenin Katsayılarının Bulunması

FIR filtrenin fark denklemi Denklem 4.1'de verilmiştir. Filtrenin katsayıları $b_0, b_1, \dots, b_k$ reel sayılardır. Şekil 4.1'de $\omega_n = 0.1$ rad/sn, $n=5$ olan bir FIR filtrenin genlik karakteristiği görülmektedir.

$$y[k] = b_0 \cdot x[k] + b_1 \cdot x[k-1] + b_2 \cdot x[k-2] + \dots + b_k \cdot x[0] \quad (4.1)$$

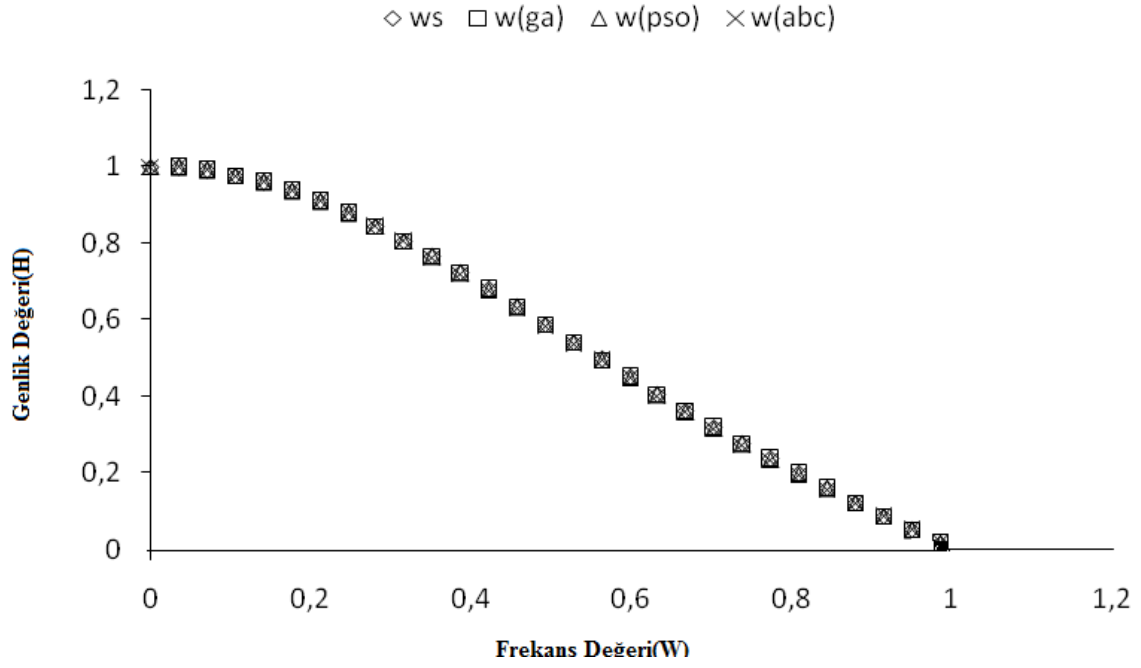


Şekil 4.1. FIR Filtre

Tablo 4.1. $n=3$ için FIR filtre katsayılar

	Gerçek değer	GA	PSO	ABC
b_1	0,8053	0,6172	0,7962	0,7863
b_2	-1,6882	-1,3179	-1,6697	-1,6498
b_3	1,6728	1,2979	1,6542	1,6342
b_4	-0,7898	-0,5971	-0,7806	-0,7706

Tablo 4.1’de FIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 3 olan FIR filtre için GA, PSO, ABC yardımıyla katsayı değerleri hesaplanmıştır. Tabloda da görüldüğü gibi paraçacık sürü optimizasyonu ile gerçek değerlere en yakın katsayılar bulunmuştur.



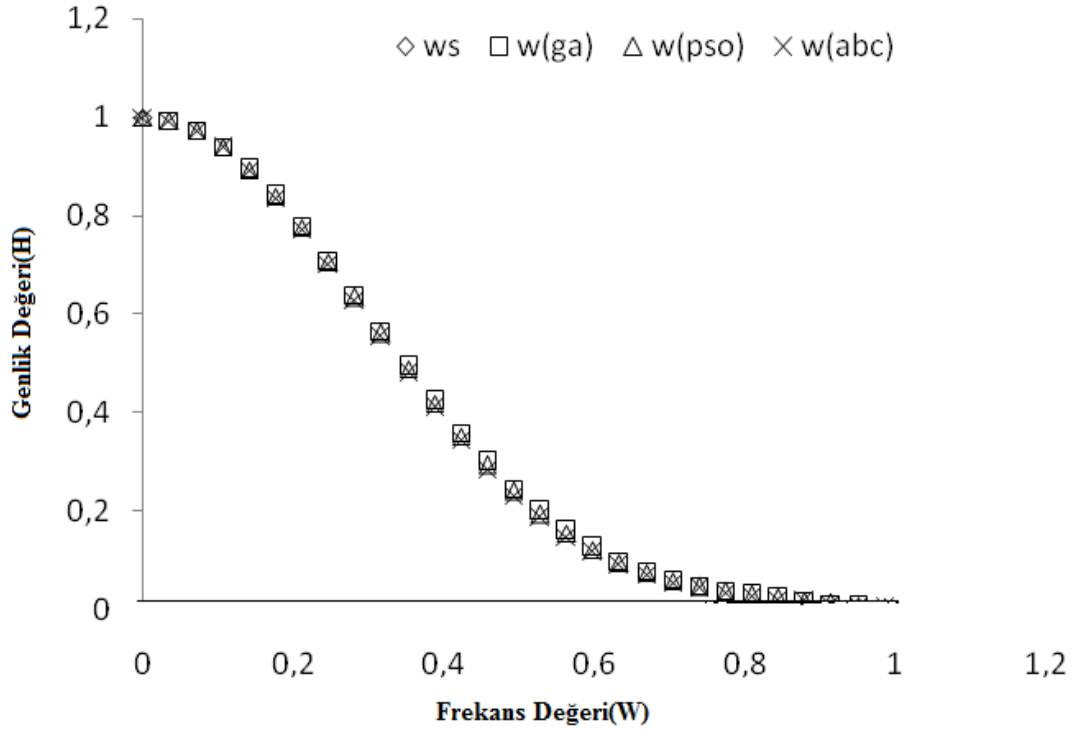
Şekil 4.2. n=3 FIR Filtre için Tahmin Edilen ve Gerçek Değerler

Aynı w_n değerine sahip derecesi $n=3$ olan bir alçak geçiren FIR filtre için katsayı değerleri hesaplanmıştır. Şekil 4.2.'de derecesi 3 olan bir FIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

Tablo 4.2. n=5 için FIR filtre katsayıları

	Gerçek değer	GA	PSO	ABC
b_1	0,1852	0,285	0,185	0,1847
b_2	-0,8855	-1,2751	-0,8824	-0,8788
b_3	1,5998	2,2816	1,5935	1,5864
b_4	-1,5895	-2,2752	-1,5831	-1,5759
b_5	0,8656	1,2627	0,8623	0,8585
b_6	-0,1756	-0,279	-0,1753	-0,1749

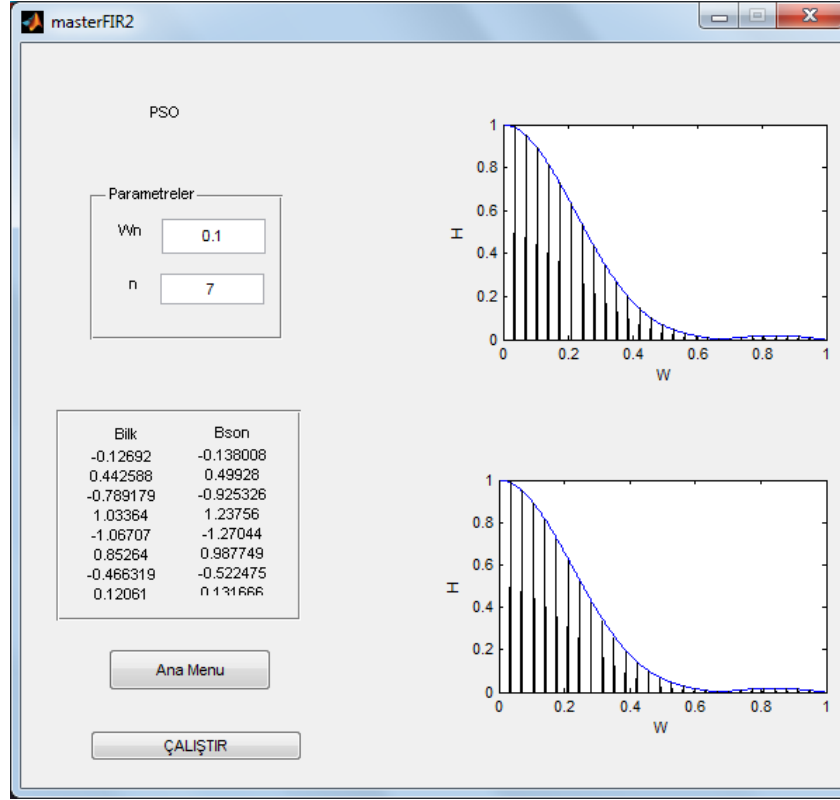
Tablo 4.2'de FIR filtrenin Matlab'da elde edilen katsayı değerleri verilmiştir. Derecesi 5 olan FIR filtre için GA, PSO, ABC yardımıyla katsayı değerleri hesaplanmıştır. Tabloda da görüldüğü gibi paraçacık sürü optimizasyonu FIR filtrenin derecesi artırıldığında daha iyi sonuçlar elde etmiştir.



Şekil 4.3. n=5 FIR Filtre için Tahmin Edilen ve Gerçek Değerler

Aynı w_n değerine sahip derecesi n=5 olan bir alçak geçiren FIR filtre için de katsayı değerleri hesaplanmıştır. Şekil 4.3.'de derecesi 5 olan bir FIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

Bu çalışmada katsayıların hesabını gerçekleştiren kullanıcıların girdileri daha kolay girmesini sağlamak amacıyla MATLAB ortamında bir kullanıcı arayüzü GUI hazırlanmıştır. Bu arayüzün bir örneği Şekil 4.4'de görülmektedir.



Şekil 4.4. GUI'de $n=7$ için FIR Filtre

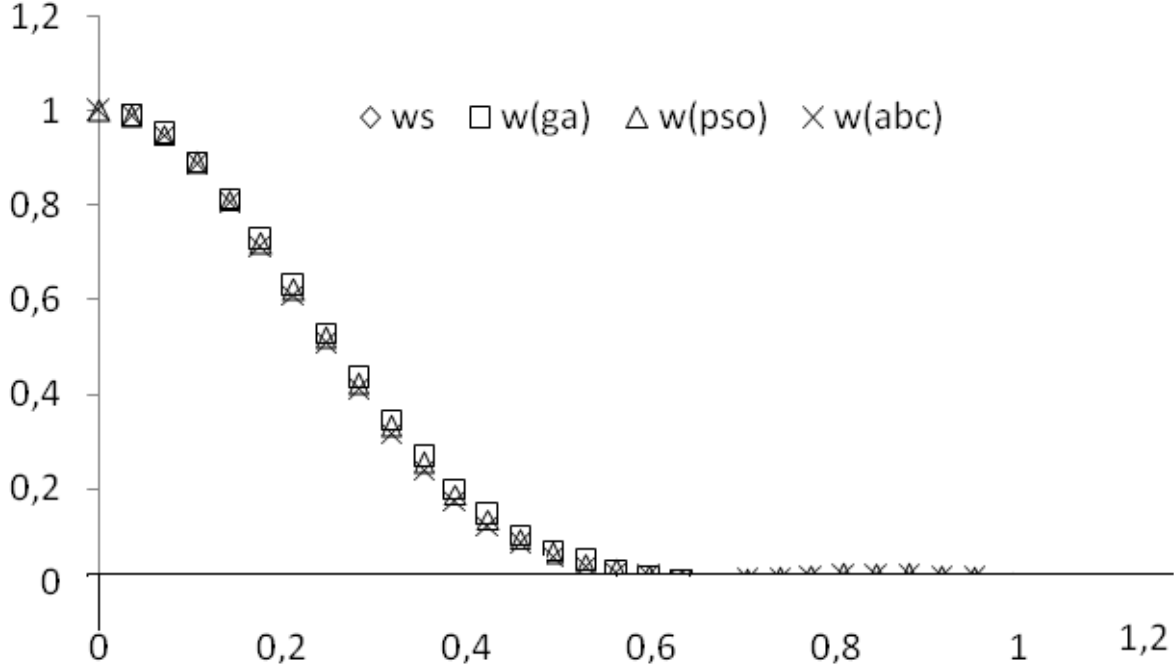
Şekil 4.4'de Derecesi 7 olan FIR filtrenin de PSO yöntemiyle bulunan katsayı değerleri verilmiştir. FIR filtrenin katsayıları GUI de GA, PSO, ABC ile hesaplanmıştır.

Tablo 4.3. $n=7$ için FIR filtre katsayıları

	Gerçek değer	GA	PSO	ABC
b_1	-0,1269	-0,1955	-0,138	-0,1466
b_2	0,4426	0,7012	0,4993	0,5436
b_3	-0,7892	-1,2382	-0,9253	-1,0324
b_4	1,0336	1,6017	1,2376	1,3982
b_5	-1,0671	-1,6531	-1,2704	1,4305
b_6	0,8526	1,3395	0,9877	1,0935
b_7	-0,4663	-0,7611	-0,5225	-0,566
b_8	0,1206	0,2055	0,1317	0,1401

Tablo 4.3'de FIR filtrenin Matlab'da elde edilen katsayı değerleri verilmiştir.

Derecesi 7 olan FIR filtre için GA, PSO, ABC yardımıyla katsayı değerleri hesaplanmıştır. FIR filtrenin derecesini en iyi bulan yöntem parçacık sürü optimizasyonudur.



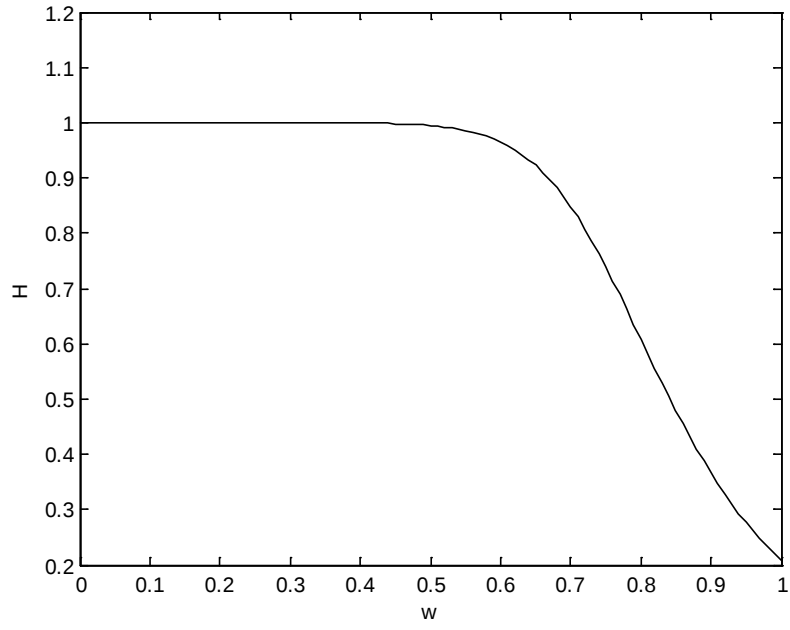
Şekil 4.5. n=7 FIR Filtre için Tahmin Edilen ve Gerçek Değerler

Aynı w_n değerine sahip derecesi $n=7$ olan bir alçak geçiren FIR filtre için katsayı değerleri hesaplanmıştır. Şekil 4.5.'de derecesi 7 olan bir FIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

4.1.2. Butterworth IIR Filtrenin Katsayılarının Bulunması

Butterworth IIR FIR filtrenin fark denklemi Denklem 4.2'de verilmiştir. Filtrenin katsayıları $b_0, b_1, \dots, b_k$ ve $a_0, a_1, \dots, a_k$ reel sayılardır. Şekil 4.6'da $n=5$, $W_p=0.1\text{rad/sn}$, $W_s=0.2\text{rad/sn}$, $R_p=0.001$, $R_s=0.5$ olan bir Butterworth IIR filtrenin genlik karakteristiği görülmektedir.

$$y[k] + b_1 \cdot y[k-1] + b_2 \cdot y[k-2] + \dots + b_m \cdot y[k-m] = a_0 \cdot x[k] + a_1 \cdot x[k-1] + a_2 \cdot x[k-2] + \dots + a_n \cdot x[k-n] \quad (4.2)$$



Şekil 4.6. Butterworth IIR Filtre

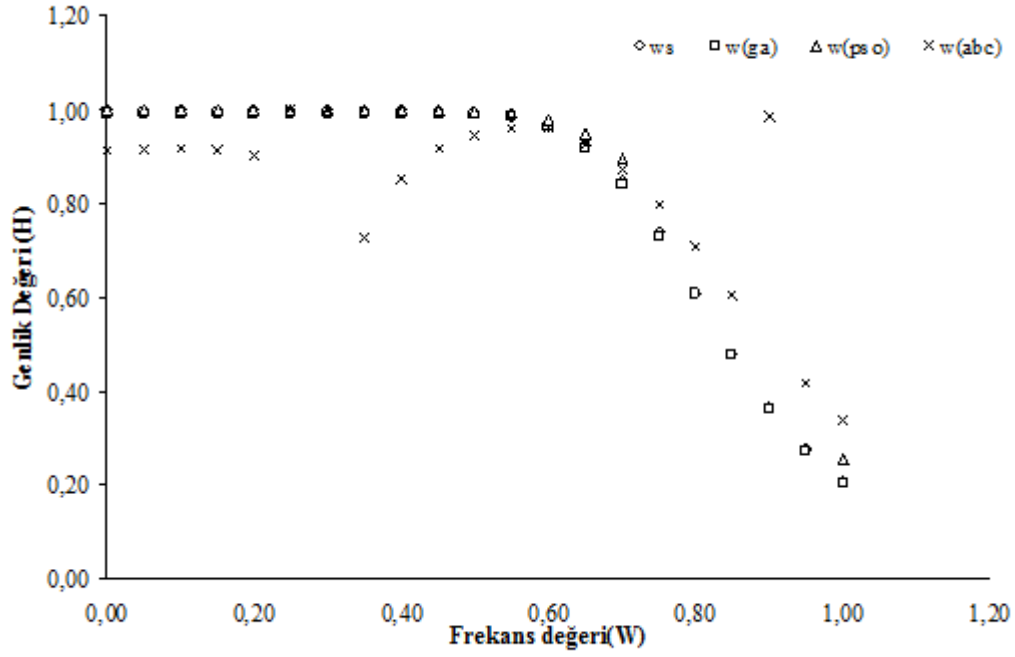
Tablo 4.4. n=5 için GUI de Butterworth IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-4,1	-4,04	-4,05	-3,89
a_3	7,2	7,19	7,24	6,85
a_4	-7,1	-7	-7,06	-6,68
a_5	3,74	3,72	3,75	3,61
a_6	-0,87	-0,87	-0,87	-0,88
b_1	0,11	0,11	0,1	0,075
b_2	-0,45	-0,45	-0,42	-0,29
b_3	0,79	0,79	0,75	0,51
b_4	-0,77	0,77	-0,73	0,5
b_5	0,41	0,41	0,39	0,27
b_6	-0,099	-0,099	-0,0937	-0,06

Tablo 4.5. n=5 için Butterworth IIR filtrenin katsayıları

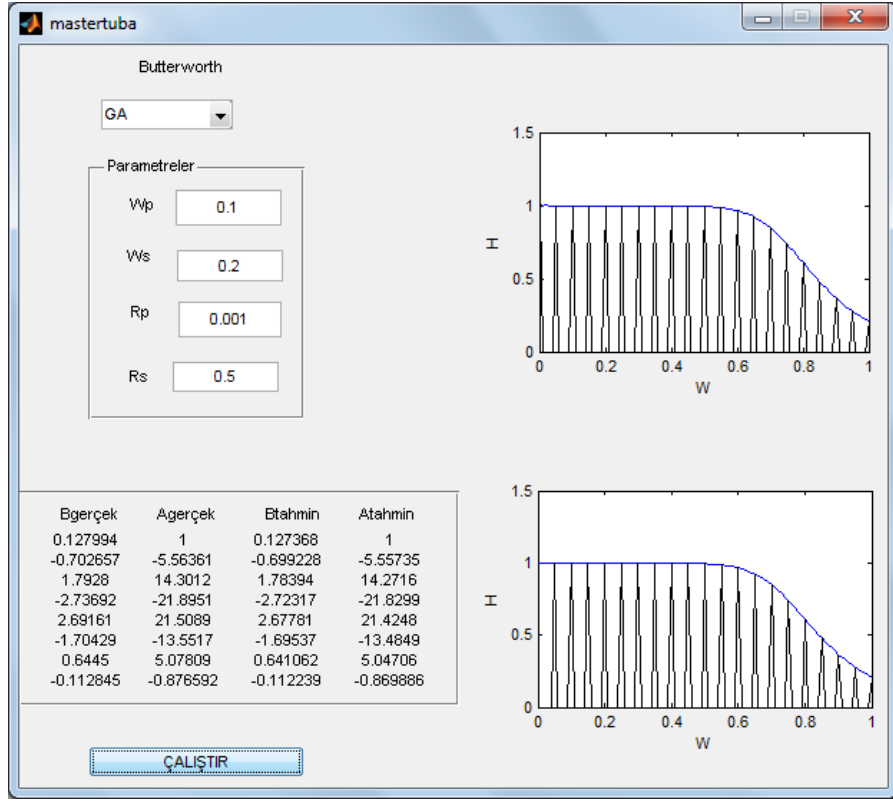
	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-4,0455	-4,0467	-4,0684	-3,9457
a_3	7,2151	7,2188	7,2258	6,9722
a_4	-7,0314	-7,0356	-6,9508	-6,819
a_5	3,7402	3,7424	3,6202	3,6855
a_6	-0,8736	-0,874	-0,8219	-0,8867
b_1	0,1133	0,1115	0,1021	0,1507
b_2	-0,4478	-0,4409	-0,4011	-0,589
b_3	0,7916	0,7795	0,7062	1,0367
b_4	-0,7729	-0,7611	-0,6885	-1,0139
b_5	0,4163	0,41	0,371	0,5499
b_6	-0,0996	-0,0981	-0,089	-0,1333

Tablo 4.4 ve Tablo 4.5’de Butterworth IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 5 olan Butterworth IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Tablolarda da görüldüğü gibi genetik algoritma en iyi performans gösteren algoritmadır.



Şekil 4.7. n=5 Butterworth Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi n=5 olan bir alçak geçiren Butterworth IIR filtre için katsayı değerleri hesaplanmıştır. Şekil 4.7.'de derecesi 5 olan bir Butterworth IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.



Şekil 4.8. GUI’de n=7 Butterworth Filtre

Şekil 4.8 ‘de Derecesi 7 olan Butterworth IIR filtrenin GA yöntemiyle bulunan pay ve payda katsayı değerleri verilmiştir. Butterworth IIR filtrenin katsayıları GUI de GA, PSO, ABC ile hesaplanmıştır. Wp, Ws, Rp, Rs değerleri ise kullanıcının tercihiine göre seçilebilir.

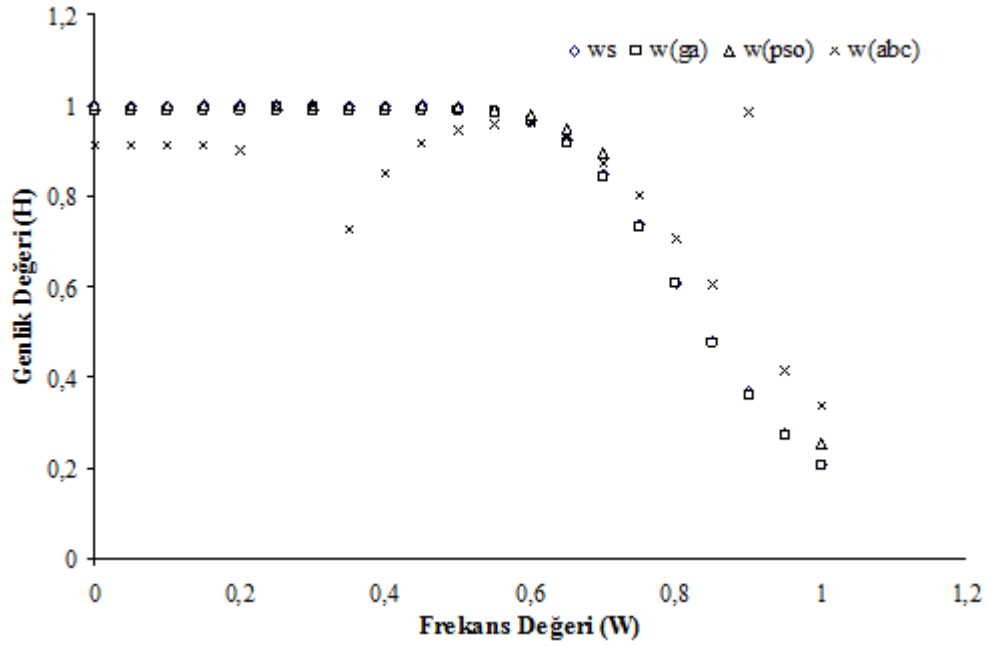
Tablo 4.6. n=7 için GUI de Butterworth IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,56361	-5,55735	-5,5757	-5,56889
a_3	14,3012	14,2716	14,3507	14,3304
a_4	-21,8951	-21,8299	-21,9836	-21,9648
a_5	21,5089	21,4248	21,5949	21,6007
a_6	-13,5517	-13,4849	-13,5972	-13,6223
a_7	5,07809	5,04706	5,0885	5,10813
a_8	-0,876592	-0,869886	-0,8765	-0,882112
b_1	0,127994	0,127368	0,121	0,128865
b_2	-0,702657	-0,699228	0,6649	-0,707873
b_3	1,7928	1,78394	1,6974	1,80734
b_4	-2,73692	-2,72317	-2,592	-2,76112
b_5	2,69161	2,67781	2,5492	2,71741
b_6	1,70429	1,69537	1,6137	1,72185
b_7	0,6445	0,641062	0,61	0,651577
b_8	-0,112845	-0,112239	-0,1067	-0,114152

Tablo 4.7. n=7 için Butterworth IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,5636	-5,5651	-5,554	-5,4936
a_3	14,3012	14,3074	14,2523	14,0142
a_4	-21,8951	-21,9058	-21,8426	-21,3717
a_5	21,5089	21,5182	21,5695	20,9757
a_6	-13,5517	-13,5552	-13,721	-13,2408
a_7	5,0781	5,078	5,2085	4,9858
a_8	-0,8766	-0,8762	-0,9117	-0,868
b_1	0,128	0,1262	0,0673	0,1666
b_2	-0,7027	-0,693	-0,3656	-0,9101
b_3	1,7928	1,7681	0,9265	2,3153
b_4	-2,7369	-2,6993	-1,4079	-3,529
b_5	2,6916	2,6545	1,3805	3,4687
b_6	-1,7043	-1,6807	-0,8729	-2,1972
b_7	0,6445	0,6355	0,3303	0,8321
b_8	-0,1128	-0,113	-0,058	-0,1461

Tablo 4.6 ve Tablo 4.7’da Butterworth IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 7 olan Butterworth IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Butterworth IIR filtrenin katsayılarını en iyi bulan yöntem GA’dır. Butterworth IIR filtrenin derecesi artırıldığında GA gerçek değere daha yakın değerler bulmuştur.

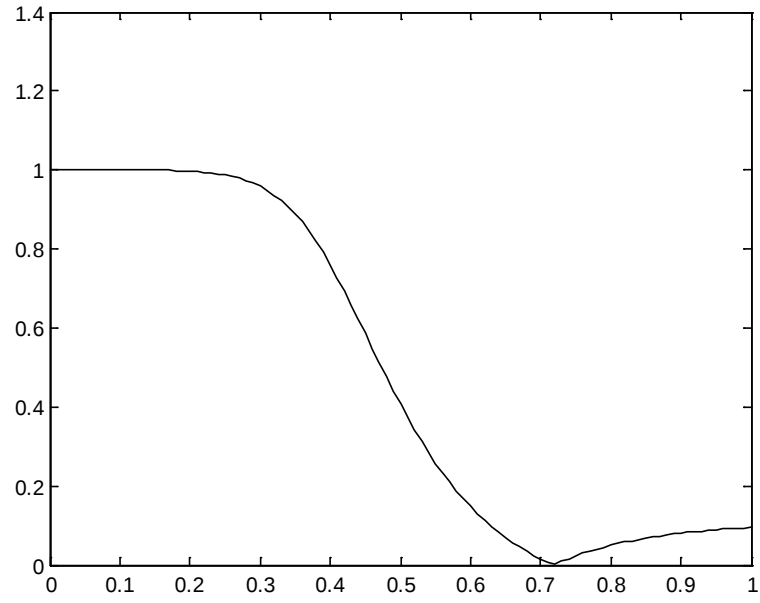


Şekil 4.9. n=7 Butterworth Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi n=7 olan bir alçak geçiren Butterworth IIR filtre için de katsayı değerleri hesaplanmıştır. Şekil 4.9.'de derecesi 7 olan bir Butterworth IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

4.1.3. Chebyshev IIR Filtrenin Katsayılarının Bulunması

Şekil 4.10'da Chebyshev IIR filtrenin n=5, $W_p=0.1\text{rad/sn}$, $W_s=0.2\text{rad/sn}$, $R_p=0.001$, $R_s=0.5$ değerlerine göre genlik karakteristiği çizilmiştir.



Şekil 4.10. Chebyshev IIR Filtre

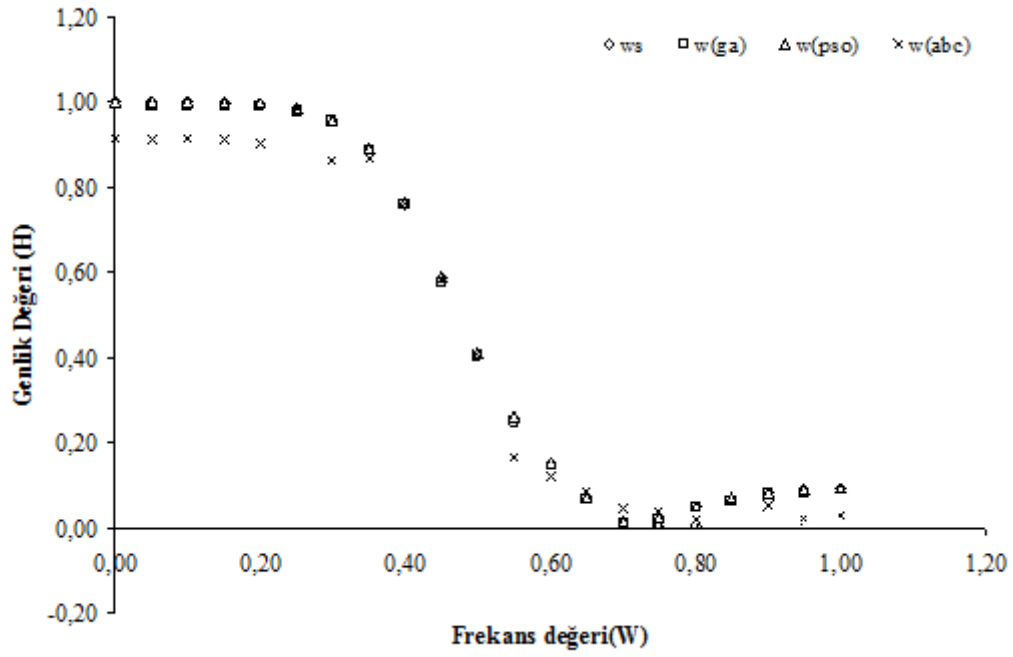
Tablo 4.8. n=5 için GUI de Chebyshev IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-4,12	-4,1	-4,11	-4,05
a_3	7,35	7,31	7,33	7,19
a_4	-7,14	-7,1	-7,16	-7,05
a_5	3,79	3,77	3,82	3,81
a_6	-0,88	-0,87	-0,88	-0,9
b_1	0,03	0,029	0,029	0,024
b_2	-0,13	-0,12	-0,12	-0,099
b_3	0,22	0,21	0,21	0,17
b_4	-0,22	-0,2	-0,2	-0,17
b_5	0,11	0,1	0,11	0,091
b_6	-0,03	-0,02	-0,026	-0,021

Tablo 4.9. n=5 için Chebyshev IIR filtrenin katsayıları

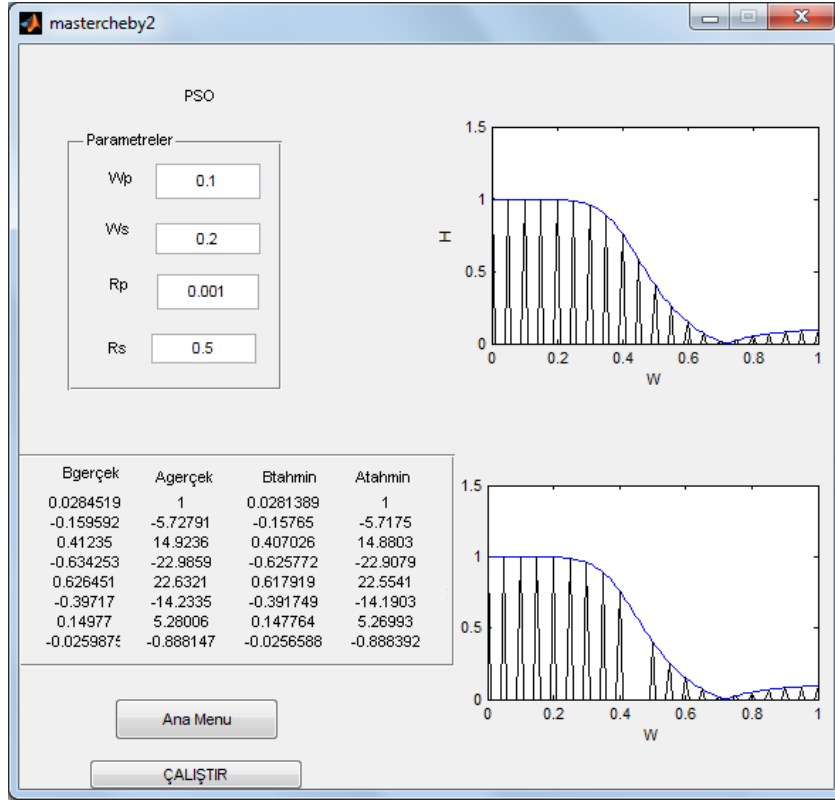
	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	4,12	-4,191	-4,1167	-4,1215
a_3	7,3439	7,538	7,3459	7,3425
a_4	-7,1427	-7,3003	-7,1434	-7,1055
a_5	3,7939	3,8144	3,7927	3,7449
a_6	-0,8776	-0,8594	-0,8768	-0,8586
b_1	0,031	0,0075	0,0313	0,0296
b_2	-0,1256	-0,0317	-0,1267	-0,1194
b_3	0,2225	0,0564	0,2244	0,2112
b_4	-0,2157	-0,0536	-0,2175	-0,2046
b_5	0,1142	0,0271	0,1152	0,1084
b_6	-0,0263	-0,0057	-0,0266	-0,025

Tablo 4.8 ve Tablo 4.9’da Chebyshev IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 5 olan Chebyshev IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Chebyshev IIR filtrenin katsayılarını en iyi bulan yöntem PSO’dur



Şekil 4.11. n=5 Chebyshev Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi n=5 olan bir alçak geçiren Chebyshev IIR filtre için katsayı değerleri hesaplanmıştır. Şekil 4.11.'de derecesi 5 olan bir Chebyshev IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.



Şekil 4.12. GUI’de $n=7$ için Chebyshev Filtre

Şekil 4.12 ‘de Derecesi 7 olan Chebyshev IIR filtrenin PSO yöntemiyle bulunan pay ve payda katsayı değerleri verilmiştir. Chebyshev IIR filtrenin katsayıları GUI de GA, PSO, ABC ile hesaplanmıştır. W_p , W_s , R_p , R_s değerleri ise kullanıcının tercihiine göre seçilebilir.

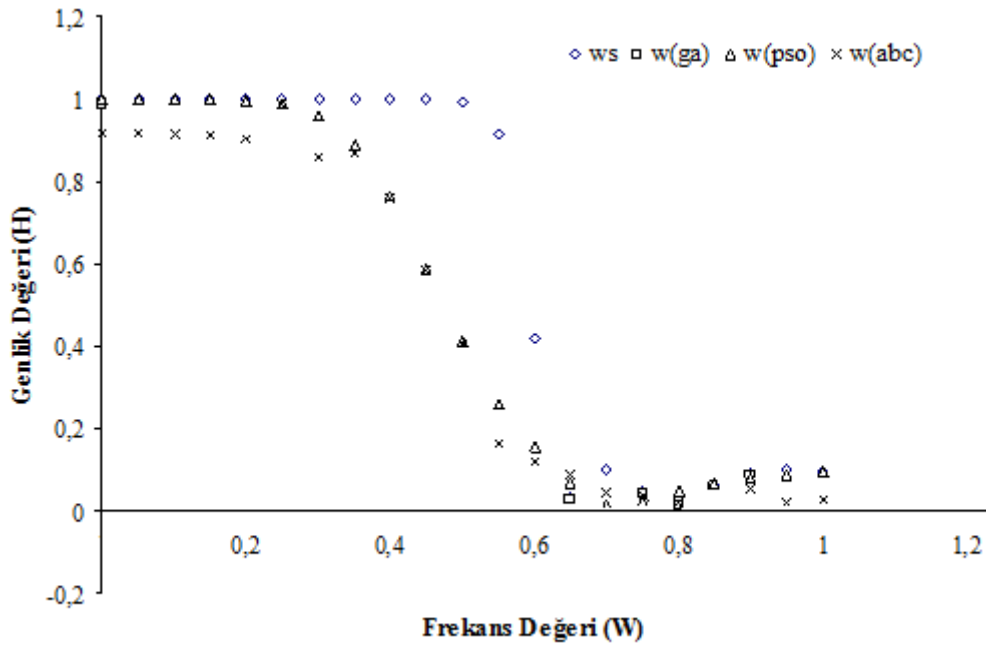
Tablo 4.10. n=7 için GUI de Chebyshev IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,72791	-5,72239	-5,7175	-5,44645
a_3	14,9236	14,8902	14,8803	13,7732
a_4	-22,9859	-22,8998	-22,9079	-20,8882
a_5	22,6321	22,5084	22,5541	20,4619
a_6	-14,2335	-14,1257	-14,1903	-12,9361
a_7	5,28006	5,22517	5,26993	4,89158
a_8	-0,888147	-0,875362	-0,888392	-0,855441
b_1	0,0284519	0,029644	0,0281389	0,0239588
b_2	-0,159592	-0,165754	-0,15765	-0,130684
b_3	0,41235	0,427483	0,407026	0,331758
b_4	-0,634253	-0,656936	-0,625772	-0,505145
b_5	0,626451	0,648709	0,617919	0,496635
b_6	-0,39717	-0,411454	-0,391749	-0,315075
b_7	0,14977	0,155334	0,147764	0,119638
b_8	-0,02598	-0,027005	-0,025658	-0,21061

Tablo 4.11. n=7 için Chebyshev IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,7279	-5,8384	-5,7277	-5,8494
a_3	14,9236	15,4208	14,9236	15,4647
a_4	-22,9859	-23,9476	-22,9873	-24,0129
a_5	22,6321	23,6462	22,6353	23,6845
a_6	-14,2335	-14,8367	-14,237	-14,8351
a_7	5,2801	5,4638	5,2822	5,4534
a_8	-0,8881	-0,9078	-0,8887	-0,905
b_1	0,0285	0,0027	0,0285	0,0095
b_2	-0,1596	-0,0161	-0,1601	-0,0552
b_3	0,4123	0,043	0,4136	0,1454
b_4	-0,6343	-0,067	-0,6362	-0,2252
b_5	0,6265	0,0662	0,6284	0,2216
b_6	-0,3972	-0,0414	-0,3984	-0,1385
b_7	0,1498	0,0151	0,1502	0,0508
b_8	-0,026	-0,0025	-0,0261	-0,0085

Tablo 4.10 ve Tablo 4.11’de Chebyshev IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 7 olan Chebyshev IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Filtrenin derecesi artırıldığında ise daha optimum sonuçlar elde edilmiştir.

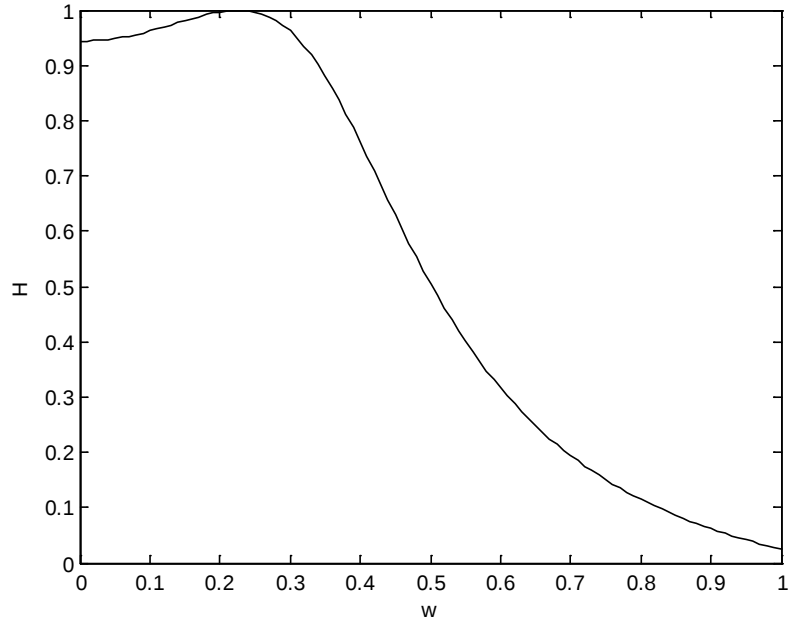


Şekil 4.13. n=7 Chebyshev Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi n=7 olan bir alçak geçiren Chebyshev IIR filtre için katsayı değerleri hesaplanmıştır. Şekil 4.13.'de derecesi 7 olan bir Chebyshev IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

4.1.4. Eliptik IIR Filtrenin Katsayılarının Bulunması

Şekil 4.14'de Eliptik IIR filtrenin n=5, $W_p=0.1\text{rad/sn}$, $W_s=0.2\text{rad/sn}$, $R_p=0.001$, $R_s=0.1$ değerlerine göre genlik karakteristiği verilmiştir.



Şekil 4.14. Eliptik IIR Filtre

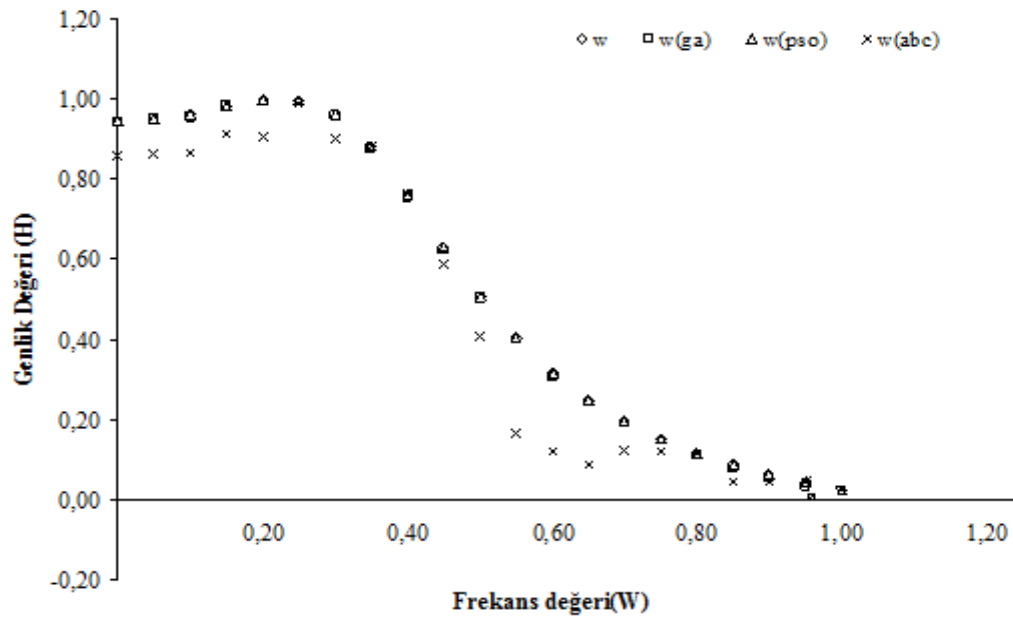
Tablo 4.12. n=5 için GUI de Eliptik IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a1	1	1	1	1
a2	-4,12	-4,11	-4,12	-4,05
a3	7,39	7,36	7,39	7,2
a4	-7,28	-7,25	-7,28	-7,06
a5	3,94	3,94	3,94	3,81
a6	-0,927	-0,922	-0,93	-0,9
b1	0,027	0,026	0,028	0,028
b2	-0,111	-0,107	-0,11	-0,11
b3	0,199	0,192	0,2	0,197
b4	-0,196	-0,189	-0,199	-0,194
b5	0,11	0,1	0,11	0,1
b6	-0,025	-0,024	-0,026	-0,025

Tablo 4.13. n=5 için Eliptik IIR filtre katsayıları

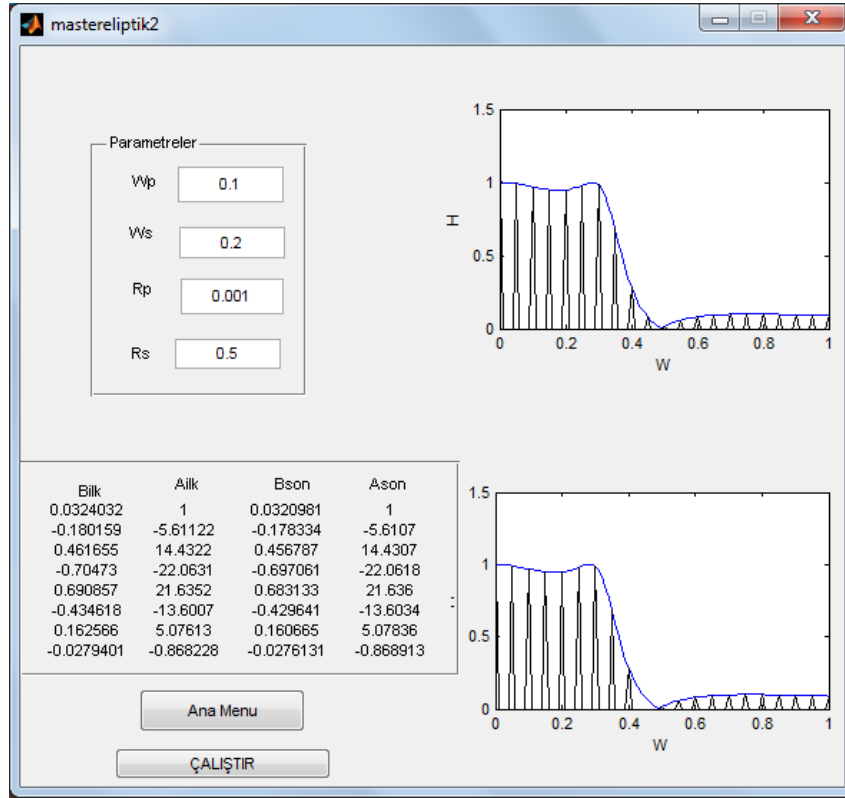
	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-4,3116	-4,2992	-4,3116	-4,2319
a_3	7,9102	7,856	7,9102	7,6909
a_4	-7,714	-7,6226	-7,714	-7,5176
a_5	3,9988	3,9322	3,9988	3,9556
a_6	-0,8819	-0,8653	-0,8819	-0,8955
b_1	0,0124	0,0107	0,0124	0,0146
b_2	-0,0532	-0,0449	-0,0532	-0,0613
b_3	0,0973	0,0808	0,0973	0,1106
b_4	-0,0949	-0,0781	-0,0949	-0,1078
b_5	0,0495	0,0404	0,0495	0,0569
b_6	-0,011	-0,0089	-0,011	-0,0129

Tablo 4.12 ve Tablo 4.13’da Eliptik IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 5 olan Eliptik IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Chebyshev IIR filtrenin katsayılarını en iyi bulan yöntem PSO’dur.



Şekil 4.15. n=5 Eliptik Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi $n=5$ olan bir alçak geçiren Eliptik IIR filtre için de katsayı değerleri hesaplanmıştır. Şekil 4.15.'de derecesi 5 olan bir Eliptik IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.



Şekil 4.16. GUI'de $n=7$ için Eliptik Filtre

Şekil 4.16 'da Derecesi 7 olan Eliptik IIR filtrenin PSO yöntemiyle bulunan pay ve payda katsayı değerleri verilmiştir. Eliptik IIR filtrenin katsayıları GUI de GA, PSO, ABC ile hesaplanmıştır. Wp, Ws, Rp, Rs değerleri ise kullanıcının tercihi göre seçilebilir.

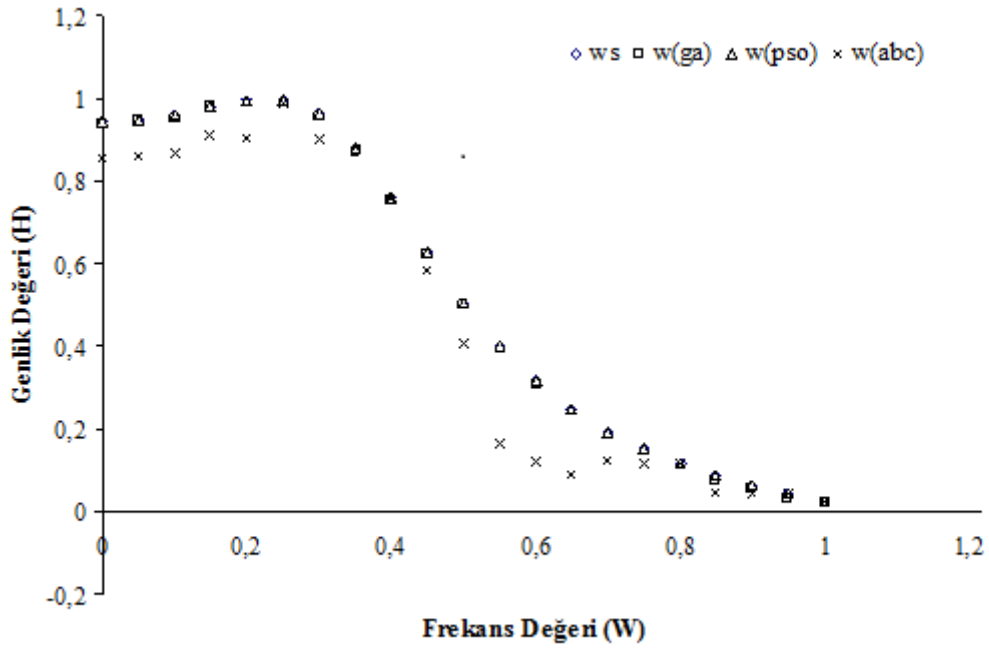
Tablo 4.14. n=7 için GUI de Eliptik filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,61122	-5,60076	-5,6107	-5,54415
a_3	14,4322	14,382	14,4307	14,18
a_4	-22,0631	-21,9543	-22,0618	-21,6777
a_5	21,6352	21,4986	21,636	21,3513
a_6	-13,6007	-13,4961	-13,6034	-13,5428
a_7	5,07613	5,02984	5,07836	5,12736
a_8	-0,868228	-0,85898	-0,868913	-0,893943
b_1	0,0324032	0,0316184	0,0320981	0,0245148
b_2	-0,180159	-0,175428	-0,178334	-0,135991
b_3	0,461655	0,448754	0,456787	0,348098
b_4	-0,70473	-0,683848	-0,697061	-0,531899
b_5	0,690857	0,669136	0,683133	0,522803
b_6	-0,434618	-0,420061	-0,429641	-0,330353
b_7	0,162566	0,156714	0,160665	0,124371
b_8	-0,02794	-0,026848	-0,027613	0,021539

Tablo 4.15. n=7 için Eliptik IIR filtrenin katsayıları

	Gerçek değer	GA	PSO	ABC
a_1	1	1	1	1
a_2	-5,8432	-5,8591	-5,8432	-5,8394
a_3	15,4586	15,5344	15,4586	15,411
a_4	-23,9806	-24,172	-23,9806	-23,8311
a_5	23,5466	23,8503	23,5466	23,3198
a_6	-14,6312	-14,9189	-14,6312	-14,4332
a_7	5,3276	5,4733	5,3276	5,2274
a_8	-0,8775	-0,9077	-0,8775	-0,854
b_1	0,016	0,0099	0,016	0,016
b_2	-0,0926	-0,0561	-0,0926	-0,0921
b_3	0,2432	0,1467	0,2432	0,2409
b_4	-0,3763	-0,227	-0,3763	-0,3717
b_5	0,3705	0,2246	0,3705	0,3651
b_6	-0,232	-0,1419	-0,232	-0,2282
b_7	0,0856	0,0532	0,0856	0,0841
b_8	-0,0144	-0,0091	-0,0144	-0,0141

Tablo 4.14 ve Tablo 4.15’de Eliptik IIR filtrenin Matlab’da elde edilen katsayı değerleri verilmiştir. Derecesi 7 olan Eliptik IIR filtrenin pay ve payda katsayı değerleri GA, PSO, ABC yardımıyla hesaplanmıştır. Eliptik IIR filtrenin katsayılarını PSO en iyi bulan algoritmadır.



Şekil 4.17. n=7 Eliptik Filtre İçin Tahmin Edilen ve Gerçek Değerler

Aynı parametrelere sahip derecesi n=7 olan bir alçak geçiren Eliptik IIR filtre için de katsayı değerleri hesaplanmıştır. Şekil 4.17.'de derecesi 7 olan bir Eliptik IIR filtrenin genlik karakteristiğinin değişimi kullanılan yöntemler için ayrı ayrı verilmiştir.

5. SONUÇLAR

Sayısal filtre, sayısallaşmış sinyaller üzerinde çalışan, giriş sinyalini istenen çıkış sinyaline dönüştüren yöntem ya da algoritmadır. Sayısal filtre yüksek güvenilirlik özelliğine sahiptir. Sayısal filtreler dürtü cevaplarına göre FIR ve IIR olmak üzere ikiye ayrılır. Filtre tasarlanırken ideale yakın filtre tasarlanması hedeflenir. İdeal filtrelerde pay ve payda katsayısını bulmak oldukça zordur. Akıllı yöntemler sayesinde hesaplama karmaşıklığı ortadan kalkacak ve hata olasılığı azalacaktır. Yüksek dereceli filtre tasarlanırken filtrenin katsayılarını elle bulmak oldukça karmaşıktır. Filtrelerin katsayıları Matlab ortamında hem daha kısa sürede hem de daha kolay bulunur.

Bu tez çalışmasında Butterworth IIR, Chebyshev IIR, Eliptik IIR, FIR filtrelerin katsayı değerleri genetik algoritma, parçacık sürü optimizasyonu ve yapay arı koloni algoritması ile bulunmuştur. Bu amaçla Matlab'da program ve kullanıcı arayüzü oluşturulmuştur. Elde edilen sonuçlar Matlab'ın kendi fonksiyonları ile elde edilen sonuçlarla kıyaslanmıştır.

Yapılan çalışma sonucunda parçacık sürü optimizasyonu FIR filtrenin katsayılarını en iyi bulan algoritma olduğu görülmüştür. Yapay arı koloni algoritması da FIR filtreler için iyi sonuçlar elde etmiştir. Butterworth IIR filtre için katsayıları en iyi yaklaşıklıkla bulan yöntem genetik algoritmadır. Butterworth IIR filtrenin derecesi artırıldığında genetik algoritma iyi sonuçlar vermiştir. Chebyshev IIR filtren için en iyi tahmin yapan yöntem parçacık sürü optimizasyonudur. Eliptik IIR filtre için kullandığımız akıllı yöntemlerin hepsinde iyi performans gözlemlenmiştir.

KAYNAKLAR

- [1] **Ertürk S.**, Sayısal İşaret İşleme, Birsen Yayınevi No: 975-511-309-6, İstanbul, 293s.
- [2] **Çetinkaya M. B.**, Genetik Algoritma Kullanarak Optimum Genlik Cevaplı Ve Minimum Fazlı Sayısal Süzgeçlerin Tasarımı, Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri, Temmuz 2004.
- [3] **Karaboğa N.**, Sayısal Filtre Katsayılarının Genetik Algoritma Kullanarak Yuvarlatılması, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri, Aralık 1994.
- [4] **Batık Z.**, Sayısal Filtre Tasarım Yöntemleri Ve Performans Analizleri, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri, Haziran 2011.
- [5] **Karaboğa N. ve Çetinkaya B.**, Optimal Design of Minimum Phase Digital FIR Filters By Using Genetic algorithm.
- [6] **Zhang vd.**, An Online Evolvable Chebyshev Filter Based on Immune Genetic Algorithm”,2008.
- [7] **Saha vd.**, IIR Filter design with Craziness based Particle Swarm Optimization Technique, 2011.
- [8] **Mondal vd.** Linear Phase High Pass FIR Filter Design using Improved Particle Swarm Optimization, 2011
- [9] **Zhao Z. ve Gao H.**, FIR Digital Filters Based on Cultural Particle Swarm Optimization, 2009.
- [10] **Krusienski D. J. and Jenkins W.K.**, Particle Swarm Optimization for Adaptive IIR Filter Structures.
- [11] **Gao vd.**, FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm.
- [12] **Durmuş B. Vd.**, IIR Filter Design using Incremental Artificial Bee Colony with Powell’s CDS, 2012.
- [13] **Vural T. ve Yıldırım T.**, Artifical Bee Colony Based Butterworth Filter Optimization, 2010.
- [14] **Lam H.Y-F.**, Analog and Digital Filter:Design and Realization, Prentice Hall, New Jersey, 1979.
- [15] **Lynn, P.A. and Fuerst, W.**, Introductory Digital Signal Processing with Comouter

- Applications, Wiley, East Kilbride, 1992.
- [16] **Kara A.**, DSP Tabanlı Sayısal Filtre Tasarımı Ve Uygulaması, Yüksek Lisans Tezi,Süleyman Demirel Üniversitesi Fen Bilimleri, 2008.
- [17] **S.Winder.**, Analog ve Dijital Filtre Tasarımı, Bileşim Yayınevi No:11, 4.Levent/İstanbul
- [18] **Thede L.**, Analog and Dijital Filter Design Using C, Prentice Hall, New Jersey,1996.
- [19] **Kayran A. H. ve Ekşioğlu E. M.**, Bilgisayar Uygulamalarıyla Sayısal İşaret İşleme, Birsan Yayınevi No: 975-511-İstanbul, 347s,2004.
- [20] **Mitra, S.K.**, Digital Signal Processing, Mc Graw-Hill Companies, 2001.
- [21] **Ağaoğlu A. E.**, Sayısal Süzgeç Tasarımı Ve Uygulamaları, Yüksek Lisans Tezi, Başkent Üniversitesi Fen Bilimleri, 2008.
- [22] **Ifeachor, E.C. and Jervis,B.W.**, Digital Signal Processing, Addison- Wesley, New York, 1993.
- [23] **URL1**,http://tr.wikipedia.org/wiki/Yapay_zek%C3%A2#Geli.C5.9Fim_s.C3.BCreci Yapay zeka.30 Temmuz 2011.
- [24] **URL-2**, <http://www.yapayzeka.org/modules/wiwimod/index.php?page=AI&back=WiwiHome> Yapay zekanın Gelişimi.25.10.09.
- [25] **Benzer İ.A.**, Yapay Zeka Uygulamalarında Kullanılan Arama Algoritmalarının Kıyaslanması, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri, Mayıs
- [26] **Serhatlıoğlu S.ve Hardalaç F.** ,Yapay Zeka Teknikleri ve Radyolojiye Uygulanması, **Fırat Tıp Dergisi** 2009, Cilt14, Sayı1, Sayfa(lar) 001-006.
- [27] **Elmas Ç.**,Yapay Zeka Uygulamaları, Seçkin Yayıncılık No:19\B-21 06410 Sıhhiye-Ankara
- [28] **Beasley D., Bull, D.R., Martin, R.R.**, An Overview of Genetic Algorithms: Part 1, Fundamentals.University Computing, Vol.15(2), pp. 58-69, UK,1993a.
- [29] **Uçan O. N., Onur O., Albora A. M.**, Nobel Yayın Dağıtım No:8 Çağaloğlu – İstanbul
- [30] **Özçakar N.**, Genetik Algoritmalar, **İ.Ü. İşletme Dergisi**, C:27, S:1/Nisan 1998 s. 69-82
- [31] **URL-3**, <http://www.sayisalyontemler.com/?q=node/83>. Genetik Algoritmaların

Uygulama Alanları. 14 Mart 2011.

- [32] **Davis L. and Coombs S. (1987)**, Genetic Algorithms And Communication Link Speed Design: Theoretical Considerations, *Grefenstette*, s. 252-256.
- [33] **Nabiyev V.V.**, Yapay Zeka, Seçkin Yayıncılık No:2106410 Sıhhiye-Ankara
- [34] **Kahraman M. ve Özdağlar D.**, Su Dağıtım Sistemlerinin Genetik Algoritma ile Optimizasyonu, DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi, Cilt:6 Sayı:3 sh.1-18 Ekim 2004
- [35] **Alataş B. ve Arslan A.**, Birlikte Kurallarının Madenciliği İçin Genetik Algoritma ve Bulanık Küme Tabanlı Yeni Bir Yaklaşım, **F.Ü. Fen ve Mühendislik Bilimleri Dergisi**, Cilt:17 No:1, 2005, s. 45.
- [36] **Turgut P. ve Arslan A.**, Sürekli Bir Krişte Maksimum Momentlerin Genetik Algoritmalar İle Belirlenmesi, DEÜ Mühendislik Fakültesi Fen ve Mühendislik Dergisi, Cilt:3 Sayı:3 sh.1-9 Ekim 2001
- [37] **URL-4**, <http://www.yapay-zeka.org/modules/wiwimod/index.php?page=GA>. Genetik Algoritmalar. 11 Nisan 2011.
- [38] **Şevkli M. ve Yenisey M. M.**, 2006. Atölye Tipi Çizelgeleme Problemi için Parçacık Sürü Optimizasyonu Yöntemi, *İTÜ Dergisi Mühendislik Serisi*, **5**, 58-68.
- [39] **Sağ T.**, Parçacık Sürü Optimizasyonu, 20 Haziran 2009.
- [40] **Tamer S. ve Karakuzu C.**, 2006. Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri, ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu, Bursa, 6-10 Aralık, s. 302-306.
- [41] **Ordukaya M.**, Asenkron Parçacık Sürü Optimizasyonu , Ankara Üniversitesi, 11-13 Haziran 2007,s. 1-4
- [42] **Karaboğa D.**, Yapay Zeka Optimizasyon Algoritmaları, Nobel yayın dağıtımı, No:74 Kızılay-Ankara
- [43] **Tokmak M.**, Yapay Arı Koloni Algoritması İle Ders Çizelgeleme Probleminin Çözümü, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri, 2011.
- [44] **Tereshko V.**, (2000) Reaction-Diffusion Model of a Honeybee Colony's Foraging Behaviour, PPSN VI: Proceedings of the 6th International Conference on Parallel Solving from Nature, 807-816, 3-540-41056-2, Springer-Verlag,

London, UK.

- [45] **Doğan A. ve Alçı M.**, Optimum Güç Akışının Yapay Arı Kolonisi İle Sağlanması, Erciyes Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü.
- [46] **Karaboğa D. ve Baştürk B.**, Performance of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing, 2008.
- [47] **Karaboğa D.**, An Idea On Honey Bee Swarm For Numerical Optimization, Technical Report-TR06, 2005.
- [48] **Akay B.**, Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi Algoritmasının Performans Analizi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri, Kasım 2009.

ÖZGEÇMİŞ

31.05.1980 yılında Elazığ'da doğdu. İlk, orta, lise eğitimini Elazığ'da tamamladı. 1998 yılında Fırat Üniversitesi Mühendislik Fakültesi Elektrik Elektronik Mühendisliği bölümünü kazandı. 2003 yılında Habaş şirketinde kontrol mühendisi olarak çalıştı. 2005 yılında Muş Alpaslan Üniversitesinde öğretim görevlisi olarak çalıştı. 2009 yılında Batman Üniversitesi Elektronik Haberleşme bölümünde öğretim görevlisi olarak çalıştı. Bu görevine halen devam etmektedir. 2009 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Bölümünde yüksek lisans eğitimine başladı.