

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMA İLE SAYISAL FİLTRE TASARIMI

Turgay KAYA

Tez Yöneticisi

Yrd. Doç. Dr. Melih Cevdet İNCE

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

ELAZIĞ, 2006

TEŞEKKÜR

Bu çalışmamda bana her türlü konuda yardımcı olan danışmanım Sayın Yrd. Doç. Dr. Melih Cevdet İNCE' ye teşekkürlerimi sunarım.

Turgay KAYA
ELAZIĞ – 2006

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMA İLE SAYISAL FİLTRE TASARIMI

Turgay KAYA

Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Ana Bilim Dalı

Bu tez, 14-08-2006 tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman: Yrd. Doç. Dr. Melih Cevdet İNCE

Üye: Prof. Dr. Yakup DEMİR

Üye: Yrd. Doç. Dr. Ali KARCI

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

İÇİNDEKİLER

ŞEKİLLER LİSTESİ.....	III
TABLOLAR LİSTESİ.....	V
SİMGELER LİSTESİ.....	VI
KISALTMALAR LİSTESİ.....	VII
ÖZET	VIII
ABSTRACT	IX
1. GİRİŞ	1
2. GENETİK ALGORİTMALAR.....	3
2.1. GA Operatörleri	8
2.2. Başlangıç Popülasyonu	8
2.3. Kromozomların Kodlanması (Encoding).....	9
2.3.1. İkili (Binary) Kodlama	9
2.3.1.1. Üçlü Kodlama.....	10
2.3.1.2. Karışık Kodlama.....	10
2.3.2. Permutasyon (Sıralı) Kodlama	11
2.3.3. Değer Kodlama.....	11
2.3.4. Ağaç Kodlama.....	12
2.4. Uygunluk Fonksiyonu ve Seçme	12
2.4.1. Rulet Tekerleği Seçimi	13
2.4.2. Sıralama Seçimi.....	13
2.4.3. Sabit Durum Seçimi	14
2.5. Çaprazlama (Crossover) ve Mutasyon (Mutation)	15
2.5.1. İkili Kodlamada Çaprazlama.....	15
2.5.1.1. Tek Noktalı Çaprazlama.....	15
2.5.1.2. İki Noktalı Çaprazlama	16
2.5.1.3. Tek biçimli (Uniform , Düzenli) Çaprazlama	16
2.5.1.4. Aritmetik Çaprazlama	16
2.5.2. İkili Kodlamada Mutasyon.....	16
2.5.3. Permutasyon Kodlamada Çaprazlama.....	17
2.5.4. Permutasyon Kodlamada Mutasyon.....	17
2.5.5. Değer Kodlamada Çaprazlama.....	17
2.5.6. Değer Kodlamada Mutasyon.....	18
2.5.7. Ağaç Kodlamada Çaprazlama	18
2.5.8. Ağaç Kodlamada Mutasyon	18
2.6. GA’larda Kontrol Parametreleri	19
2.6.1. Popülasyon Büyüklüğü.....	19
2.6.2. Çaprazlama Oranı.....	19
2.6.3. Mutasyon Oranı	19

3. SAYISAL FİLTRE TASARIMI	20
3.1. Sayısal Filtre Karakteristiklerinin Belirlenmesi.....	22
3.2. İdeal Sayısal Filtreler	23
3.2.1. İdeal Alçak ve Yüksek Geçiren Sayısal Filtreler.....	24
3.2.2. Fiziksel Gerçekleştirme	26
3.3. İdeal olmayan Sayısal Filtreler	28
3.4. Geçici Durum Davranışı	29
4. AYRIK ZAMANLI FIR FİLTRE TASARIMI.....	31
4.1. FIR Filtrelerin Avantajları ve Dezavantajları	32
4.2. Fourier Serisi Yöntemi	34
4.2.1. Pencere Fonksiyonu Kullanımı	37
4.3. Frekans Örnekleme Yöntemi	39
4.4. Filtre Tasarımının İyileştirilmesi	41
5. AYRIK ZAMANLI IIR FİLTRE TASARIMI.....	42
5.1. Değişmez İmpuls Cevabı Yöntemi ile IIR Filtre Tasarımı	43
5.2. Değiştirilmiş Değişmez-İmpuls Cevabı Yöntemi	45
5.3. Bilineer Dönüşüm Yöntemi	47
Bir sayısal IIR filtresi üç aşamadan meydana gelmektedir. Bunlar,	49
5.3.1. Bilineer Dönüşümün Özellikleri.....	49
5.4. Sarma Etkisi ve Önsarma.....	50
6. PROGRAMIN PROBLEME UYGULANMASI.....	54
6.1. Programla Çalışılan Örnekler	56
6.1.1. Örnek1	56
6.1.2. Örnek2	62
6.1.3. Örnek3	66
6.1.4. Örnek4	68
7. SONUÇ VE ÖNERİLER.....	71
KAYNAKLAR	72
ÖZGEÇMİŞ.....	74

ŞEKİLLER LİSTESİ

Şekil 2.1 GA çemberi.....	5
Şekil 2.2 Genetik Algoritmalar da evrimleşme döngüleri.....	5
Şekil 2. 3 Genetik Algoritma akış diyagramı.....	6
Şekil 2. 4 Uygunluğun lineer ölçeklenmesi	7
Şekil 2. 5 İkili kodlama için kromozom örneği.....	9
Şekil 2.6 Üçlü kodlama.....	10
Şekil 2.7 Karışık kodlama.....	10
Şekil 2.8 Permutasyon kodlama için kromozom örneği	11
Şekil2.9 Değer kodlama için kromozom örneği.....	11
Şekil 2.10 Ağaç kodlama için kromozom örneği.....	12
Şekil 2.11 Rulet seçimi	13
Şekil 2.12 Sıralamadan önceki durum (Uygunluk grafiği)	14
Şekil 2.13 Sıralamadan sonraki durum (düzenli sayıların grafiği).....	14
Şekil 2.14 İkili kodlamada tek noktalı çaprazlama	15
Şekil 2.15 İkili kodlamada iki noktalı çaprazlama.....	16
Şekil 2.16 İkili kodlamada tek biçimli (uniform) (düzenli) çaprazlama	16
Şekil 2.17 İkili kodlamada aritmetik çaprazlama.....	16
Şekil 2.18 İkili kodlamada mutasyon.....	17
Şekil 2.19 Permutasyon kodlamada çaprazlama.....	17
Şekil 2.20 Permutasyon kodlamada mutasyon.....	17
Şekil 2.21 Değer kodlamada mutasyon.....	18
Şekil 2.22 Ağaç kodlamada çaprazlama	18
Şekil 3. 1 Kesim frekansı $\Omega_c = \pi/2$ radyan olan sayısal filtrenin çeşitli biçimlerde frekans cevabının gösterilmesi.....	23
Şekil 3. 2 İdeal sayısal filtre frekans karakteristikleri; a) İdeal Alçak geçiren b) İdeal Yüksek geçiren c) İdeal Bant-geçiren d) İdeal Bant-durduran.....	24
Şekil 3. 3 İdeal alçak geçiren sayısal filtrenin genlik ve faz karakteristiği	25
Şekil 3. 4 İdeal alçak geçiren sayısal filtrenin impuls cevabı.....	27
Şekil 3. 5 Alçak geçiren filtrenin frekans cevabı	28
Şekil 3. 6 Bant geçiren filtrenin frekans cevabı	29
Şekil 3. 7 Sayısal filtrelerde geçici durum davranışı a) geçici cevapta aşma.....	30
Şekil 4. 1 Simetrik impuls cevabına sahip doğrusal fazlı FIR filtrenin tasarlanması.....	32
Şekil 4. 2 Sonsuz uzunluktaki ideal alçak geçiren sayısal filtrenin impuls cevabı	35
Şekil 4. 3 Sonlu uzunlukta (N=23) tasarlanan FIR filtrenin impuls cevabı	35
Şekil 4. 4 Fourier serisi yöntemi kullanılarak tasarlanan FIR filtrelerin genlik cevapları	36
Şekil 4. 5 FIR filtre tasarımında kullanılan pencere fonksiyonları	38
Şekil 4. 6 M=40 değeri için Hamming penceresi kullanılarak tasarlanan FIR filtrenin genlik cevabı	38
Şekil 4. 7 Hamming pencere fonksiyonu kullanılarak tasarlanan alçak geçiren FIR filtrenin.....	38
Şekil 4. 8 İdeal alçak geçiren filtreden eşit aralıklı örnek değerlerinin alınması (N=23).....	40
Şekil 4. 9 Şekil 4.8’ deki frekans örneklerine karşılık gelen filtrenin frekans cevabı.....	40
Şekil 4. 10 (T_1, T_2, T_3) değişken katsayılar a sahip olan ve frekans örnekleme yöntemi ile tasarlanan filtrenin frekans cevabı	41
Şekil 5. 1 Sürekli-zamanlı bir filtrenin ayırık-zamanlı eşdeğeri	43
Şekil 5. 2 Periyodik örnekleme nin gösterilmesi.....	44
Şekil 5. 3 Sistemdeki kutupların s-düzleminde n z-düzlemine dönüşümü.....	44
Şekil 5. 4 $H_a(\omega)$ ve $H_D(e^{j\omega T})$ frekans spektrumu ve meydana gelen örtüşme	45

Şekil 5. 5 s-düzleminden s' -düzlemine olan dönüşüm.....	47
Şekil 5. 6 Bilineer dönüşüm ile ω dan ω' ne dönüşüm ($T=1$ için).....	48
Şekil 5. 7 Bilineer dönüşüm uygulanarak s-düzleminden z-düzlemine geçiş.....	49
Şekil 5. 8 $H(j\omega)$ 'dan $H(e^{j\Omega})$, ya bilineer dönüşümün gösterilmesi	51
Şekil 5. 9 Genlik cevabı üzerindeki sarma etkisi	52
Şekil 6. 1 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk)ve Çözüm kromozomundaki katsayıların oluşturduğu genlik cevabı (kırmızı renk)	61
Şekil 6. 2 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve son generasyondaki 10. kromozomun (uygunluk değeri en kötü) katsayılarının oluşturduğu genlik cevabı (kırmızı renk)	61
Şekil 6. 3 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk)ve Çözüm kromozomundaki katsayıların oluşturduğu genlik cevabı (kırmızı renk)	66
Şekil 6. 4 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve son generasyondaki 10. kromozomun (uygunluk değeri en kötü) katsayılarının oluşturduğu genlik cevabı (kırmızı renk)	66
Şekil 6. 5 20 popülasyon, 50 generasyon sonrasında alçak geçiren filtre için en iyi kromozom (kırmızı renk) ve istenen filtre katsayıları için (siyah renk) genlik cevabı	68
Şekil 6. 6 20 popülasyon, 50 generasyon sonrasında yüksek geçiren filtre için en iyi kromozom (kırmızı renk) ve istenen filtre katsayıları için (siyah renk) genlik cevabı	70

TABLÖLER LİSTESİ

Tablo 6. 1 Başlangıç generasyonu	57
Tablo 6. 2 Çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri	57
Tablo 6. 3 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (2. generasyon)	58
Tablo 6. 4 İkinci generasyonda çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri	59
Tablo 6. 5 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (3. generasyon)	59
Tablo 6. 6 50. generasyon (Çözüm)	60
Tablo 6. 7 Başlangıç generasyonu	62
Tablo 6. 8 Çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri	63
Tablo 6. 9 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (2. generasyon)	63
Tablo 6. 10 İkinci generasyonda çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri	64
Tablo 6. 11 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (3. generasyon)	64
Tablo 6. 12 50. generasyon (Çözüm)	65
Tablo 6. 13 Son generasyon	67
Tablo 6. 14 Son generasyonla bulunan çözüm katsayıları	69

SİMGELER LİSTESİ

Hz	: Hertz
kHz	: Kiloherzt
dB	: Desibel
t	: Zaman
f	: Frekans
ω	: Analog frekans
Ω	: Sayısal frekans
Ω_p	: Geçirme bandı kenar frekansı
Ω_s	: Durdurma bandı kenar frekansı

KISALTMALAR LİSTESİ

GA	: Genetic Algorithm (Genetik Algoritma)
U	: Uygunluk
ÖU	: Ölçeklenmiş Uygunluk
KS	: Kopya Sayısı
FFT	: Fast Fourier Transform (Hızlı Fourier Dönüşümü)
FIR	: Finite Impulse Response (Sonlu İmpuls Cevabı)
IIR	: Infinite Impulse Response (Sonsuz İmpuls Cevabı)
LPF	: Lowpass Filter (Alçak Geçiren Filtre)
HPF	: Highpass Filter (Yüksek Geçiren Filtre)
BPF	: Bandpass Filter (Bant Geçiren Filtre)
BSF	: Bandstop Filter (Bant Durduran Filtre)

ÖZET

Yüksek Lisans Tezi

GENETİK ALGORİTMA İLE SAYISAL FİLTRE TASARIMI

Turgay KAYA

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

2006, Sayfa: 74

Bu tezde Genetik Algoritma (GA) kullanılarak sayısal filtre tasarımı yapılmıştır. Bunun için önce GA yöntemi ayrıntıları ile incelenmiştir. Algoritmada kullanılan çaprazlama, mutasyon operatörleri, başlangıç popülasyonu, amaç fonksiyonu ve uygunluk fonksiyonu açıklanmıştır.

Ayrıca sayısal filtre hakkında temel bilgiler verilerek, bilinen tasarım yöntemleri anlatılmıştır. Özellikle FIR (Finite Impulse Response) ve IIR (Infinite Impulse Response) filtre yapıları ve aralarındaki farklar vurgulanarak her bir filtre türünün tasarım yöntemleri açıklanmıştır.

Sayısal filtre tasarımında ve Genetik Algoritmada elde edilen bilgiler birleştirilerek GAFKAT(Genetik Algoritma ile Filtre Katsayılarının Hesaplanması) adı verilen bir yazılım üretilmiştir. MATLAB kullanılarak yazılan bu programla, istenilen FIR filtre özelliklerini sağlayan katsayılar bulunarak genlik cevabı çizdirilmiş, literatürde bilinen filtre cevaplarıyla karşılaştırılmış ve tatmin edici sonuçlar elde edilmiştir.

Anahtar Kelimeler: Genetik Algoritma, Sayısal Filtre, FIR Filtre, IIR Filtre

ABSTRACT

Masters Thesis

DIGITAL FILTER DESIGN WITH GENETIC ALGORITHM

Turgay KAYA

Firat University

Graduate School of Natural and Applied Sciences

Department of Electrical-Electronics Engineering

2006, Page: 74

In this thesis, design of digital filter was made using Genetic Algorithms (GA). At first GA methods were studied in detail and crossover and mutation operators, initial population, fitness function, object function used in algorithm were explained.

Furthermore, by using basic information about digital filters, well-known design methods were studied. Especially FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filter structures and differences between them were emphasized. Design methods of each filter types were explained.

A software, GAFKAT (**G**enetik **A**lgoritma ile **F**iltre **K**atsayılarının Hesaplanması), was produced by combining obtained knowledge about digital filter and genetic algorithms. Coefficient providing desirable FIR filter characteristic were found, amplitude responses were plotted and compared with well-known filter responses at literature and satisfying results were obtained in this thesis.

Key Words: Genetic Algorithm, Digital Filter, FIR Filter, IIR Filter

1. GİRİŞ

Sezgisel yaklaşım yöntemleri kullanılarak yapılan programlama yöntemlerine, genel olarak Evrimsel Hesaplama (evolutionary computing) yöntemleri denir. Bu programlama yöntemlerinden birisi olan Genetik Algoritmalar, (GA) Darwin'in evrim teorisi üzerine geliştirilmiştir [1]. Öyle ki, tabiatın var oluşundan günümüze kadar geçen sürede kendisini kusursuz bir şekilde koruyarak sürdürmesi, insanoğlunu bu muhteşem düzenden faydalanma fikrine itmiştir. Tabiat, ilk oluşumundan günümüze kadar kendisini belirli bir düzen içerisinde sürekli olarak yenileyerek devam etmiş, insanoğlu da bu olaydan esinlenerek problem çözmede farklı bir çözüm yolu geliştirmiştir. Bir GA programı genel olarak, en iyilerin hayatta kalma ilkesine dayanmaktadır [2]. Bu işlemler gerçekleştirilirken ilk olarak başlangıç popülasyonu oluşturulur [3, 4]. Daha sonra, bu popülasyon içerisindeki her bir kromozomun uygunluk değeri hesaplanır ve kromozomun kendisini bir sonraki nesle, kopyalama yöntemiyle aktararak kendi soyunu devam ettirerek optimum çözüm bulunur.

Evrimsel Hesaplama yöntemlerinden biri olan GA, bir çok alanda uygulanma imkanı bulmaktadır. Bu uygulama alanları içerisinde, genellikle bilinen hesaplama yöntemlerinin yetersiz olduğu veya bu yöntemlerin uygulanmalarının uzun vakit aldığı alanlar fazlaca görülmektedir. Bu yöntemlerden biriside filtre katsayılarının hesaplanmasıdır [3, 5, 6, 7, 8, 9]

Filtre, istenilen özellikleri sağlayacak yapının oluşturulması olarak tanımlanabilir. Sayısal işaret işleme ise, işaretin frekans spektrumu üzerinde istenilen işlemleri gerçekleştirme ve gerçekleştirilen işlemler neticesinde de istenilen özellikleri sağlayacak yapıyı oluşturmaktır. Böylece meydana getirilen bu yapıya, sayısal filtre adı verilmektedir [10, 11]. Sayısal filtreler, frekans cevaplarına göre sonlu impuls cevaplı filtreler (FIR Finite Impulse Response) ve sonsuz impuls cevaplı (IIR Infinite Impulse Response) filtreler olarak ikiye ayrılmaktadırlar.

Filtre tasarımında (gerek IIR gerekse de FIR filtrede) temel amaç, filtre transfer fonksiyonunu oluşturan pay ve payda katsayılarının hesaplanması işlemine dayanmaktadır [12, 13]. Bu katsayıların GA ile bulunması işlemi bu çalışmada amaçlanan hedeftir.

Yapılan tez çalışması 7 ana bölümden oluşmaktadır. Bu bölümlerden, Bölüm2' de GA hakkında genel bilgiler verilmiş, ayrıca GA' yı oluşturan kısımlar olan başlangıç popülasyonu, uygunluk fonksiyonu, çaprazlama işlemi ve mutasyon işlemleri ayrıntıları ile anlatılmıştır [1, 3, 5].

Bölüm3’ de Sayısal Filtre Tasarımı hakkında bilgiler, ideal filtrelerin genlik cevapları verilmiş ve fiziksel olarak neden gerçekleştirilemedikleri anlatılmış ve gerçekleştirilebilmeleri için nelere ihtiyaç olduğundan bahsedilmiştir [14, 15, 16].

Bölüm4’ de Sayısal filtre çeşitlerinden FIR sayısal filtreler hakkında bilgiler verilerek, bu tür filtrelerin hangi yöntemler kullanılarak gerçekleştirildikleri anlatılmıştır [10, 11, 17, 18].

Bölüm5’de Sayısal filtre çeşitlerinden diğeri olan IIR sayısal filtreler anlatılmıştır. Ayrıca hangi yöntemler kullanılarak hayata geçirildiklerinden ayrıntılarıyla bahsedilmiştir [10, 11, 17, 18].

Bölüm6’ da MATLAB’ da yazılan ve GAFKAT adı verilen program yardımıyla çeşitli örnekler çalıştırılmış ve elde edilen genlik cevapları ile MATLAB’ ın içerisinde yer alan FIR1 komutu ile bulunan katsayılardan elde edilen genlik cevapları karşılaştırılmıştır [19].

Son bölüm olan Bölüm7’ de elde edilen sonuçlar karşılaştırılmış, GAFKAT programının getirmiş olduğu kolaylıklardan bahsedilmiş ve gelecekte yapılabilecek çalışmaların neler olabileceğinden bahsedilmiştir.

2. GENETİK ALGORİTMALAR

Evrimsel hesaplamanın bir parçası olan GA, Darwin'in evrim teorisi modelinden esinlenerek meydana getirilmiştir. Evrimsel hesaplama, ilk olarak 1960'lı yıllarda I. Rechenberg'in "Evrimsel Stratejileri" adındaki çalışması yardımıyla gündeme gelmiştir. Evrimsel hesaplama genelde genetik algoritmaları, evrimsel stratejisini ve genetik programlamayı kapsamaktadır. Evrimsel hesaplama içerisinde yer alan, evrimsel stratejileri ve genetik algoritmalar arasındaki bazı farklılıklar vardır. İkisi arasındaki en önemli fark, genetik algoritmalar çaprazlama ve mutasyon işlemlerini kullanırken, evrimsel stratejisi sadece mutasyon işlemini kullanmaktadır. Ayrıca evrimsel stratejisi kullanıldığında problem içerisindeki verilerin kodlanmalarına gerek yoktur [1].

Genetik Algoritmaların günümüzde kullanılan biçimi, ilk olarak Michigan Üniversitesi'nden, psikolog ve bilgisayar bilimleri uzmanı J.H.Holland tarafından 1975 yılında ortaya konulmuştur [1]. Genetik Algoritmalar, en iyinin korunumu ve doğal seçim ilkesinin benzetim yoluyla bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir. Bilinen bir GA programında, her bir birey önceden belirlenmiş sınırlı uzunluğa sahip ve kromozom adı verilen vektörler yardımıyla modellenir. İlk olarak tamamıyla rasgele seçilen ve belirli bir sayıda kromozom alınarak başlangıç popülasyonu (toplum) elde edilir. Bu başlangıç popülasyonu elde edildikten sonra, kromozomlar yeni nesiller (generasyon) elde etmek için bazı değişikliklere uğrarlar. Rasgele elde edilen popülasyon içerisindeki vektörün (kromozom) iyiliği ölçülür. Yani kromozom önceden belirlenen amaç fonksiyonu içerisinde yazılarak kromozomun uygunluk değeri hesaplanır. Elde edilen uygunluk değerine göre bir sonraki neslin oluşumu için bazı kromozomlar yeniden üretilip, çaprazlanır veya mutasyon işlemlerine uğratılırlar.

Tekrar seçilme, çaprazlama veya mutasyon işlemlerine uğratılacak olan kromozomlar tamamen rasgele seçilirler. Bu işlemlerde, kromozomlar amaç fonksiyonunda yerine konularak bulunan yeniden seçilme ihtimaline göre, yüksek olan kromozom doğal olarak daha fazla çaprazlama işlemine girecek ve sonraki neslin içinde daha fazla bulunacaktır. Değeri az olan kromozom ise sonraki neslin içinde az bulunacak belki hiç bulunamayacak ve yok olacaktır.

Standart bir GA programı aşağıdaki adımlardan oluşmaktadır.

1. Olası çözümlerin oluşmasını sağlayacak ve kromozomlardan oluşan başlangıç popülasyonun oluşturulması:

Bu işlem için belirlen bir sayı yoktur. Yapılan programa göre popülasyon büyüklüğü değişmektedir. Ancak elde edilen popülasyonun sayısı programın sonuçlanma hızına etki etmektedir.

2. Popülasyon içerisinde yer alan her bir kromozomun ne kadar iyi olduğu hesaplanır:

Bu işlemde, kromozom önceden belirlenmiş amaç fonksiyonu içerisinde yerine yazılarak uygunluk değeri hesaplanır. Uygunluk değerinin bulunması işlemine evrimleşme (evaluation) bu amaçla kullanılan fonksiyona da uygunluk fonksiyonu adı verilmektedir. GA programında kullanılan ve problemde özel çalışan tek kısım, uygunluk fonksiyonudur. Özel olarak çalıştığı için de programın başarısını etkileyen önemli bir parametredir. Bu nedenle programın doğru bir şekilde çalışması, iyi bir uygunluk fonksiyonu ile yakından ilgilidir.

3. Rasgele seçilen kromozomlar kendi aralarında eşlenerek kopyalama ve değiştirme işlemleri uygulanır:

Bu adımda, kromozomlar uygunluk değerlerine göre birbirleriyle çaprazlama işlemine uğrarlar. Çaprazlama işlemi sonrasında yeni nesil toplum oluşturulur. Çaprazlama işlemi, kullanılan kromozomların uygunluk değerlerine göre yapılırken, seçim işlemi farklı yöntemlerle yapılmaktadır. GA'lar da en fazla kullanılan seçim yöntemleri arasında rulet tekerleği seçim yöntemi (roulette wheel selection) ve turnuva seçimi (tournament selection) gibi yöntemler verilebilir.

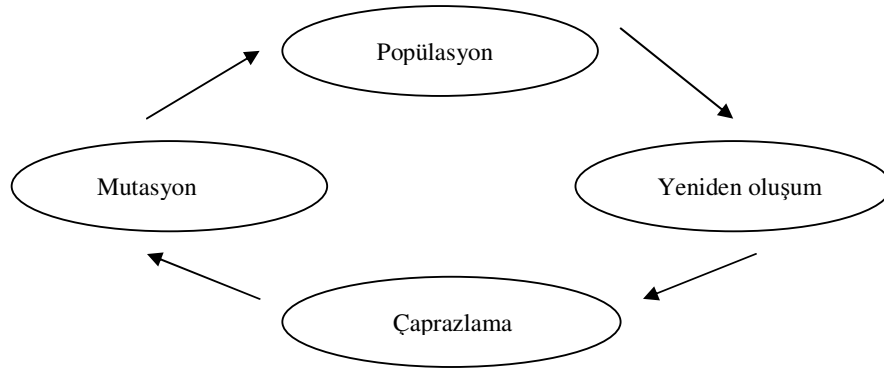
4. Çaprazlama neticesinde elde edilen yeni kromozomlara yer açmak için, var olan eski kromozomların bazıları (uygunluk değeri düşük olanlar) çıkartılarak ilk baştaki başlangıç popülasyon sayısının sabit kalması sağlanır.
5. Elde edilen yeni popülasyondaki her bir kromozomun uygunluk değeri, yeniden amaç fonksiyonunda yerine yazılarak elde edilir.
6. Rasgele üretilen iki sayının birbirine eşit olması durumunda aramanın daha geniş bir uzayda hesaplanmasını sağlamak için mutasyon işlemine başvurulur.
7. Bu işlemler, istenilen yakınsama belirlendiğinde veya önceden belirlenen generasyon sayısına ulaşıldığında sona erdirilir [2].

Yukarıdaki adımlardan oluşan GA, birçok yönden klasik optimizasyon yöntemlerinden ayrılmaktadır. GA'yı diğer metotlardan ayıran özellikler aşağıdaki gibidir:

- GA bir arama noktası değil, bir grup arama (adaylar) üzerine çalışarak sonuca ulaşmaktır. Böylelikle tek bir arama değil de, aramayı bir arama uzayında yaparak yerel değil de global yapmaktadır.
- GA, bireylerin (kromozomların) uygunluk değerlerine bakarken sadece amaç-uygunluk fonksiyonunu göz önüne alır. Böylelikle de türevsel ve diferansiyel denklemlerle işlem yükünden kurtulur.
- Yeni nesillerin oluşması için gerekli olan seçme ve birleştirme işlemlerinde, deterministik kurallar değil olasılık kurallarını kullanır.

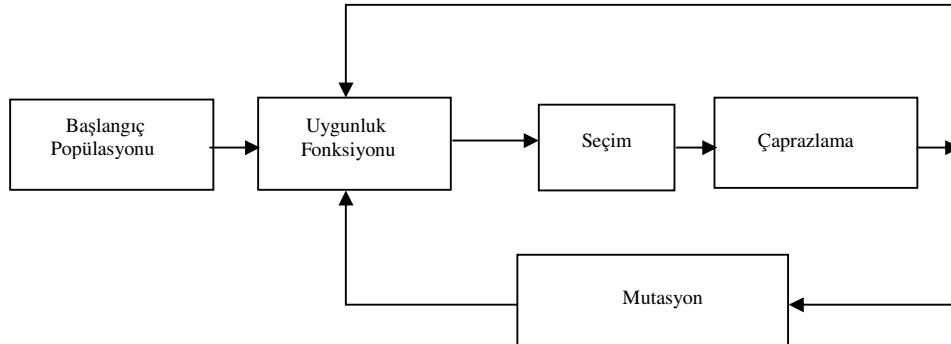
- Diğer metotlar doğrudan parametreler üzerinde işlemler yaparken, GA, optimize edilecek parametreleri kodlayarak parametreler üzerinde değil de bu kodlar üzerinde çalışmaktadır.
- GA, olasılık kurallarına göre çalıştığı için önceden ne kadar iyi çalıştığı belirlenemez.

Bu özellikleri ile diğer birçok programlama yöntemlerinden ayrılan GA, basit olarak Şekil 2.1’ de görülen GA çemberi yardımıyla modellenenabilir.

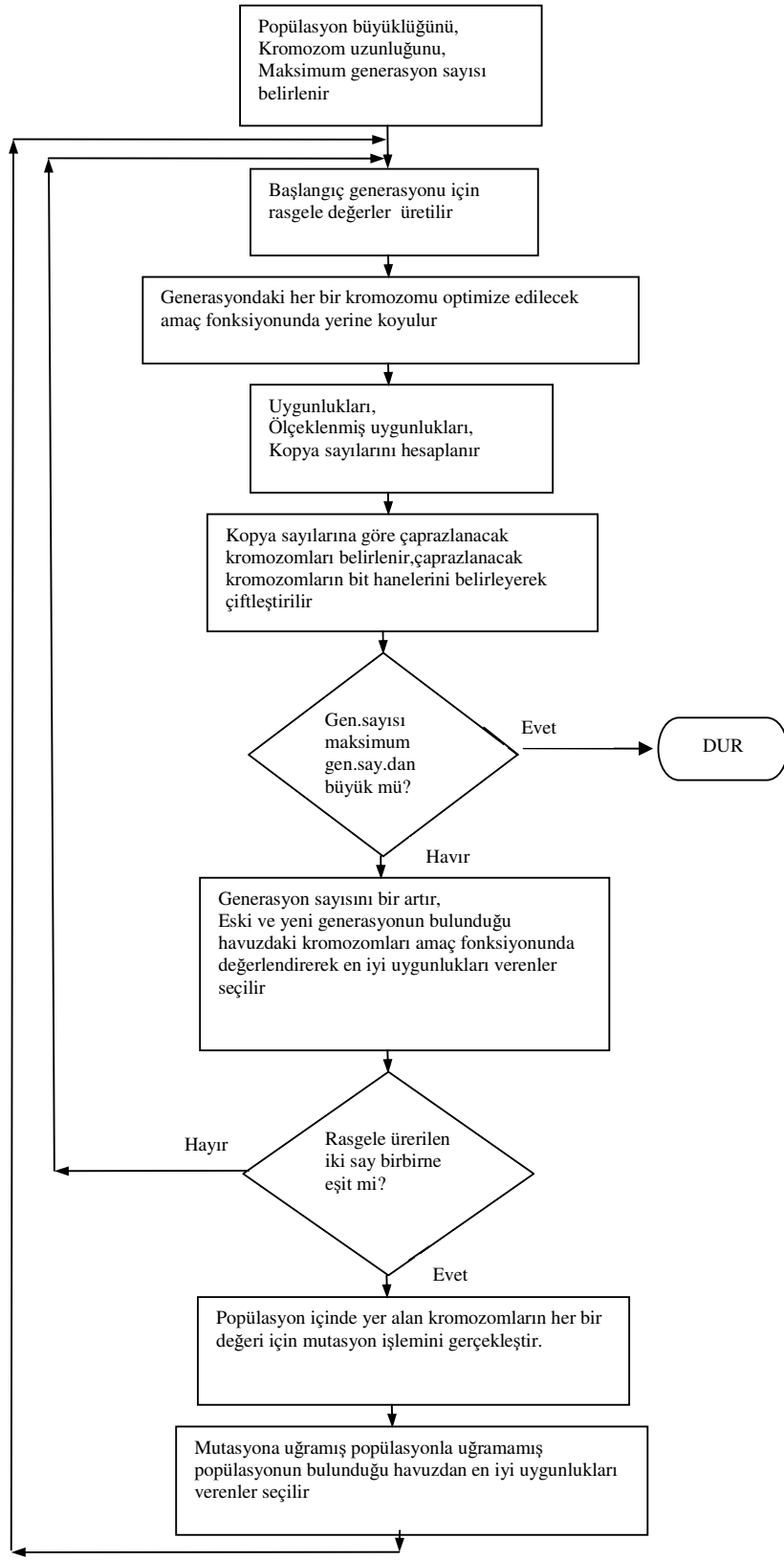


Şekil 2.1 GA çemberi

Şekil 2.1’ deki basit GA çemberi biraz daha geliştirilirse, GA ‘nın oluşumunu özetleyen Şekil 2.2’ deki blok diyagramı elde edilir. Şekil 2.3’ de de GA programına ait akış diyagramı görülmektedir.



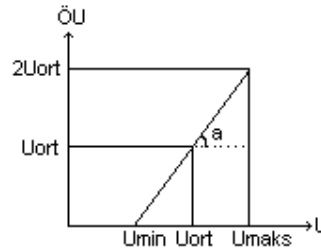
Şekil 2.2 Genetik Algoritmalar da evrimleşme döngüleri



Şekil 2. 3 Genetik Algoritma akış diyagramı

Şekil 2.3' deki akış diyagramından da görüleceği gibi, başlangıç olarak GA programının kaç popülasyondan oluşacağı belirlenir. Popülasyon içerisinde yer alan kromozomlar kullanıcı tarafından belirlenen sayı uzunluğu ile kodlanır. Ayrıca GA programının kaç generasyon ile yapılacağı, başka bir ifade ile programın kaç defa tekrar edileceğini belirleyen sayının kullanıcıdan girilmesi beklenir. Girilen bu değerler yardımıyla GA 'nın ilk aşaması olan ve tamamen rasgele üretilen başlangıç popülasyonu oluşturulur.

Popülasyon içerisinde yer alan kromozomların her biri, önceden belirlenen amaç veya uygunluk fonksiyonunda yerine yazılarak her biri için uygunluk değeri hesaplanır. Bu uygunluk değerlerine göre, kromozomların ölçeklenmiş uygunluk ve kopya sayıları belirlenir. Kromozomların, uygunluk ve ölçeklenmiş uygunluk değerlerinin bulunmasında değişik yöntemler vardır. Bu tezdeki GA da Şekil 2.4' deki yapıdan faydalanılmıştır.



Şekil 2. 4 Uygunluğun lineer ölçeklenmesi

$$\ddot{U} = \tan(a) \cdot (U - U_{ort}) + U_{ort} \quad (2.1)$$

$$\tan(a) = \frac{U_{ort}}{U_{maks} - U_{ort}} \quad (2.2)$$

Burada, U: her bir kromozomun uygunluk fonksiyonunda yerine yazılmasıyla hesaplanan uygunluk değeri, U_{max} : uygunluk değerleri içerisindeki en büyük uygunluk değeri, U_{ort} : ise uygunluk değerlerinin ortalamasını temsil etmektedir.

Yapılan işlemler neticesinde lineer ölçeklemenin yakınsamada olumlu sonuç verdiği gözlemlenmiştir.

$\ddot{U}_{top}$, ele alınan popülasyondaki kromozomların her biri için lineer ölçeklenmiş uygunlukların toplamı, $\ddot{U}_i$ ise ele alınan satırdaki değişkenler için ölçeklenmiş uygunluk olsun.

Bu değerlere göre herhangi bir satırın seçilme ihtimali; $\ddot{U}_i / \ddot{U}_{top}$ dır.

I' inci satırın kopya sayısı,

Popülasyon sayısı * $\ddot{U}_i / \ddot{U}_{top}$ olarak hesaplanır [2].

Popülasyon içerisindeki kromozomlar hesaplanan kopya sayılarına göre çaprazlama işlemine uğratılırlar. Böylece yeni bir generasyon elde edilmiş olur. Elde edilen bu generasyon, yeniden uygunluk fonksiyonunda yerlerine yazılarak yeni uygunluk değeri hesaplanır. Elde edilen bu değerler ile bir önceki generasyon arasında en iyi olan kromozomlar alınarak yeni bir popülasyon elde edilir. Bu popülasyon ile ilk baştaki işlemler takip ettirilerek döngü sürdürülür. Bu işlemler, istenilen yakınsama belirlendiğinde veya önceden belirlenen generasyon sayısına ulaşıldığında sona erdirilir.

Program bu şekilde çalışırken bazı durumlarda kromozomların mutasyona uğramaları gerekebilir. Mutasyon işlemi, programın başında kullanıcı tarafından belirlenen bir mutasyon oranı veya olasılığına bağlı olarak yapılır. Başlangıçta rasgele bir sayı üretilir ve daha sonra her yeni generasyonda başka bir rasgele sayı üretilip bu iki sayının birbirine eşit olması durumunda mutasyon yapılır. Böylece arama işlemi, bir noktaya sıkışıp kalmaktan kurtulur.

2.1. GA Operatörleri

Genel olarak bir GA programı; başlangıç popülasyonu, uygunluk fonksiyonu, seçim işlemi, çaprazlama ve mutasyon evrelerinden oluşmaktadır. Bu aşamaların her biri kendi başlarına özel veya rastlantısal olarak oluşmalarına rağmen, GA programının çalışmasını önemli derecede etkileyen parametrelerdir.

2.2. Başlangıç Popülasyonu

Algoritma, ilk olarak popülasyon (topluluk) olarak tanımlanan ve programın çalışmasını sağlayan kromozomlardan oluşmaktadır. Gerçek mühendislik problemlerinin çözümü için değil de araştırma yapmak amacıyla yazılan programlarda kullanılan GA'ların başlangıç popülasyonları genellikle rasgele üretilir. Fakat problemle ilgili başlangıçta bazı çözümler biliniyorsa bu durumda, popülasyon tamamen rasgele olarak değil de bir kısmı bilinen değerlerden oluşarak yapılmaktadır. Bu durum tercih edilir ve programın optimum çözümü bulmasında zaman yönünden tasarruf sağlanır [3].

Gerek rastlantısal gerekse bir kısmı bilinerek oluşturulan başlangıç popülasyonunun büyüklüğü, algoritmanın performansını etkileyen önemli bir etkidir. Popülasyon büyüklüğünün fazla seçilmesi, işlemlerin karmaşıklığı ve aramanın derinliğini arttıracaktır. Bu durum da çalışma zamanının uzamasına neden olacaktır. Tam tersi olarak popülasyonun az

sayıda seçilmesi ise kromozomların çeşitliliğinde azalmaya neden olacağından, çözümün hassasiyetini etkileyecektir. Ayrıca GA programında, generasyon sayısı ile popülasyon büyüklüğünün dengelenmesi gerekmektedir. Öyle ki, 100 kromozomlu bir popülasyon, çözümü 10 generasyonla bulabilir. Fakat 20 generasyonda çözümü bulan 20 kromozomlu bir başlangıç popülasyonuna sahip bir GA 'dan 4 kat daha uzun zaman alabilir. Bu durumda, başlangıç popülasyonu küçük olan bir GA, başlangıç popülasyonu büyük olan bir GA 'dan daha fazla generasyon sayısı ile çözümü bulmasına rağmen, daha kısa bir sürede optimum çözüme ulaşabilir sonucu çıkartılır [4].

GA ile problemin çözümünde, bireyleri temsil eden kromozomların kodlanması işlemi büyük önem taşımaktadır. Bu durum, GA 'nın etkinliğini ve çalışma hızını etkiler. Kromozomların kodlanmalarında problemin türüne göre farklı yöntemler kullanılmaktadır.

2.3. Kromozomların Kodlanması (Encoding)

Bir GA programının yazılması işleminde karşılaşılan ilk problemlerden biri kromozomların kodlanmasıdır. Kodlama işlemi problemin türüne göre değişmektedir. Bazen binary (ikili) kodlama en iyi çözüm bulmada kullanılırken, bazen de permutasyon veya değer kodlama yöntemleri optimum sonucu bulmada en uygun kodlama yöntemi olabilir.

2.3.1. İkili (Binary) Kodlama

İlk GA programlarında kullanılan bu yöntem, halen en çok kullanılan kodlama yöntemidir. Bu kodlama yönteminde, popülasyonu oluşturan kromozomların her biri 0 ve 1'ler ile ifade edilerek kodlanırlar. Sonuçta kodlanan kromozom bir sayıya karşılık gelmektedir [5,6].

Kromozom	A	1011100010110
Kromozom	B	010110100000

Şekil 2. 5 İkili kodlama için kromozom örneği

Bu kodlama yöntemiyle kodlanan kromozomlar, sadece ikili gösterimle değil ikinin kuvveti olarak gösterilip farklı bir yöntem olarak kullanılıp işlem kolaylığı getirilmektedir. Bu yöntemler üçlü ve karışık kodlama yöntemleridir.

2.3.1.1. Üçlü Kodlama

Bu kodlamada dizi içerisindeki sayılar 0, 1, veya -1 olabilirler. Bu tür kodlama yönteminde, üçlü dijitin her biri ikinin kuvveti ile ilgilidir. Örneğin, $k.2^m$ 'i göz önüne alalım. Eğer k pozitif ise (+1), dizi içerisindeki $|m|$ 'nin durumu 1 sayısı ile kodlanmaktadır. Öte yandan k negatif ise (-1), dizi içerisindeki $|m|$ 'nin durumu $\bar{1}$ sayısı ile kodlanır. Diğer durumlarda ise 0 ile kodlama yapılır [7,8].

Üçlü kodlanmış 0.421875 sayısını göz önüne alalım. Bu sayı aşağıdaki gibi kodlanmaktadır.

$$\text{Katsayı} = 0.421875 \text{ ise } 2^{-1} - 2^{-4} - 2^{-6}$$

0	1	0	0	$\bar{1}$	0	$\bar{1}$	0
---	---	---	---	-----------	---	-----------	---

LSB

Şekil 2.6 Üçlü kodlama

2.3.1.2. Karışık Kodlama

Bu kodlama yönteminde, her bir katsayı ikili dizi ve üçlü dijit ile alakalı olarak kodlanmıştır. İkinin kuvveti olan $k.2^{m1}$, $k.2^{m2}$, $k.2^{m3}$ sayılarını ele alalım. Karışık kodlama yönteminde, üçlü değeri temsil eden k_i ; 0, 1 veya -1 olabilir. Bu sayıları m_i 'nin ikili kodlanmış dizisi takip etmektedir [7].

Örnek olarak yine 0.421875 sayısını göz önüne alalım.

$$\text{Katsayı} = 0.421875 \text{ ise } 2^{-1} - 2^{-4} - 2^{-6} \text{ ile kodlanır.}$$

1	0	0	1	$\bar{1}$	1	0	0	$\bar{1}$	1	1	0
---	---	---	---	-----------	---	---	---	-----------	---	---	---

MSB LSB

Şekil 2.7 Karışık kodlama

Şekilden de görüleceği gibi, -2^{-4} sayısı $\bar{1}$ katsayısı ve [1 0 0] dizisi yardımıyla kodlanmıştır [7].

Ancak ikili kodlama yönteminde, küçük boyutlu problemler için çok büyük kromozom vektörleri meydana gelebilmektedir. Bu yüzden farklı kodlama yöntemleri problemin türüne göre kullanılmaktadır.

2.3.2. Permutasyon (Sıralı) Kodlama

Bu kodlama yöntemi, genel olarak gezgin satıcı ve şebeke tasarımları gibi sıra gözeten problemlerde yaygınca kullanılır.

Kromozom	A	3 5 1 2 7 6 0 4
Kromozom	B	0 1 5 6 2 3 4 7

Şekil 2.8 Permutasyon kodlama için kromozom örneği

Bu tür problemlerde yine permutasyon olarak düşünülen kodlama ikili diziye çevrilerek kodlanabilir. Yapılan bu değişiklikte, her sayı kaç bit (dijit) ile kodlanmışsa sonuçta elde edilen ikili dizi şeklindeki kromozomun dijit sayısı bir sayıyı temsil eder [5].

2.3.3. Değer Kodlama

Bu kodlama yöntemi, gerçel gibi kompleks sayıların da yer aldığı problemlerde kullanılır. Bu tür problemler için ikili kodlama oldukça zordur. Burada her kromozom, bazı değerler dizisidir ve bu değerler problemle ilişkilidir. Örneğin gerçel sayı, karakter veya kompleks sayılar olabilir. Bu tür kodlama, yapay sinir ağları için ağırlıkların hesaplanmasında yaygın olarak kullanılmaktadır [1].

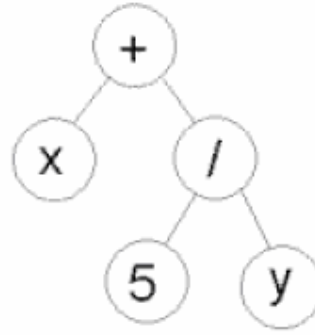
Kromozom	A	3.274 6.121 2.456 0.299 2.115
Kromozom	B	ABCJFJKHDERJFDLFFEGHK
Kromozom	C	(geri), (sağ), (ileri), (geri), (sol)

Şekil2.9 Değer kodlama için kromozom örneği

2.3.4. Ağaç Kodlama

Ağaç kodlama genellikle GA' da programlar ve ifadeler oluşturmak için kullanılır. Ağaç kodlama yapılarak oluşturulan kodlamada her kromozom, adından da anlaşılacağı gibi nesneler ve nesneler arası işlemleri içeren bir ağaç yapısından oluşur [1].

Kromozom A



$$(x + (5 / y))$$

Şekil 2.10 Ağaç kodlama için kromozom örneği

2.4. Uygunluk Fonksiyonu ve Seçme

Popülasyon içerisinde yer alan ve her bir bireyi temsil eden kromozomların ne kadar iyi olduklarını bulan fonksiyona uygunluk fonksiyonu (uyumluluk, fitness) fonksiyonu denir. Bu GA programında özel çalışan tek kısımdır. Uygunluk fonksiyonu kromozomların şifrelerini çözer (decoding) ve sonra da hesaplama yaparak bu kromozomların uygunluklarını hesaplar.

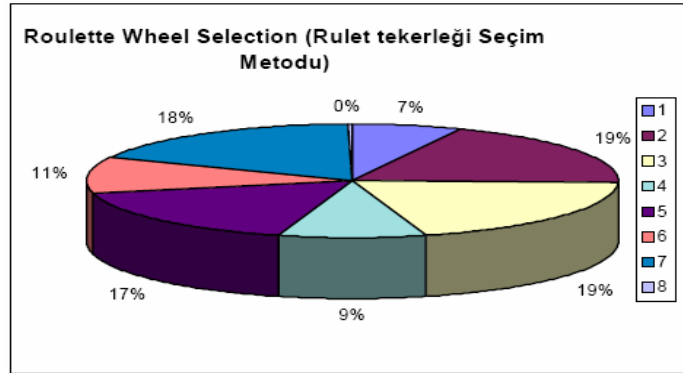
Elde edilen uygunluk değerlerine göre kromozomların yeni generasyon da olup olmayacaklarına karar verilir. Bütün popülasyonun (topluluğun) uygunluk değerleri belirlendiğinde sonuçlandırma kriterinin sağlanıp sağlanmadığına bakılır. Sonuçlandırma kriteri birkaç şekilde seçilebilmektedir. Bu seçeneklerden biri, istenilen nesil sayısına ulaşılması olabileceği gibi, tüm topluluğun uygunluk değerinin en iyi çözümün uyumluluk değerine oranının yeterli görülen bir değere ulaşması olabilir [9]. Eğer istenilen bu iki kriterden biri sağlanamazsa genetik operatörler olan seçme, çaprazlama ve kopyalama işlemine devam edilir.

Kromozomlar yeni ve daha iyi kromozomlar oluşturmak için çaprazlama işlemine popülasyon içerisinde seçilerek girerler. Burada problem, seçim işleminin nasıl olacağı ile

ilgilidir. Darwin' in evrim teorisinden faydalanarak oluşturulan yöntemde, en iyi bireyler çaprazlama işlemine girerek nesillerini devam ettirip hayatta kalırken, kötü olanlar ise çaprazlama işlemine alınmayarak hayatlarına son verilir. En iyi kromozomları seçmek içinde birçok yöntem vardır. Bu seçim yöntemlerinden bazıları; Rulet tekerleği seçimi (roulette wheel selection), Boltzman seçimi (Boltzman selection), turnuva seçimi (tournament selection), sıralama seçimi (rank selection) ve sabit durum seçimidir (steady state selection) [5].

2.4.1. Rulet Tekerleği Seçimi

Bu seçim işleminde, kromozomların seçilme olasılıkları her bir kromozomun sahip olduğu uygunluk değerine bağlıdır. Ancak bu yöntemde uygunluk değeri, en iyi olanın seçilmesini garanti etmez fakat seçilme ihtimalinin yüksek olduğu kesindir. Tüm bireylerin uygunluk değerleri bir tabloda tutulur. Daha sonra bu uygunluk değerlerinin her biri, toplam uygunluk değerine bölünerek her biri için % olarak seçilme ihtimali bulunmuş olur [5].

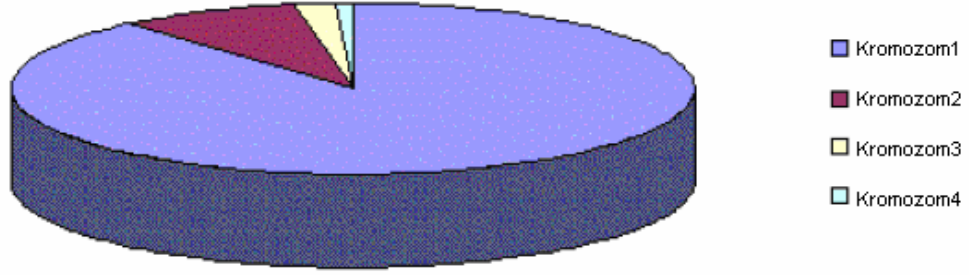


Şekil 2.11 Rulet seçimi

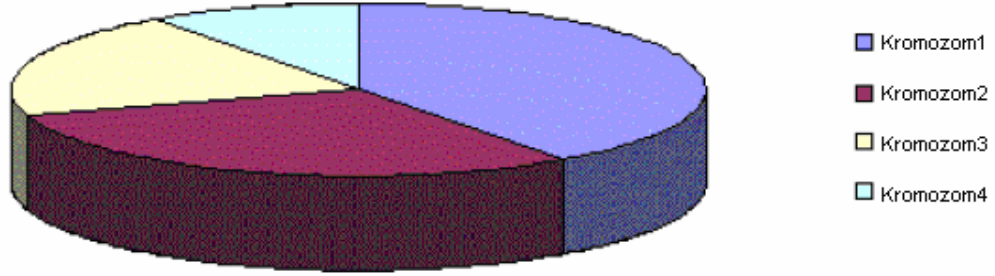
2.4.2. Sıralama Seçimi

Rulet tekerleği seçimi, uygunluklar çok farklıysa problemlere yol açacaktır. Örneğin, en iyi kromozomun uygunluğu, rulet tekerleğinin %90 'ında ise, diğer kromozomlar çok az seçilme şansına sahip olacaktır.

Sıralama seçimi, önce popülasyonu sıralamakta ve ardından her kromozomun bu sıralamada uygunluğunu aramaktadır. En kötüsü 1 uygunlukta, ikinci kötüsü 2 uygunlukta olarak sıralanırsa, en iyisi ise N uygunlukta olacaktır (Burada N, popülasyondaki kromozom sayısıdır) [1].



Şekil 2.12 Sıralamadan önceki durum (Uygunluk grafiği)



Şekil 2.13 Sıralamadan sonraki durum (düzenli sayıların grafiği)

Yapılan bu seçim işleminden sonra bütün kromozomlara seçim şansı doğacaktır. Fakat bu yöntemin bir dezavantajı, en iyi kromozomun diğerlerinden fazla farkı olamayacağından çözüme ulaşma yavaş olacaktır.

2.4.3. Sabit Durum Seçimi

Bu yönteme göre, yeni nesil kromozomların (ebeveynler) oluşması için kromozomların büyük parçaları bir sonraki generasyona taşınmalıdır. GA, daha sonra şu şekilde çalışmaktadır. Yeni nesil kromozomları oluşturmak için var olan kromozomlardan uygunlukları en iyi olan bir kaç kromozom seçer. Daha sonra popülasyonda geri kalan ve uygunlukları kötü olan kromozomlar atılır ve yerlerine yeni kromozomlar getirilir. Böylelikle popülasyonun geri kalan kısmı yeni nesilde hayattadır. Kısacası; alt popülasyon oluşturulduktan sonra uygunluklar hesaplanır, en kötü kromozomlar yerlerini başlangıçtaki en iyi kromozomlara terk ederler.

Yapılan bu seçim yöntemleri içerisinde biri kullanılarak gelecek neslin oluşturulması işlemi, uygunluk değerleri kötü olan kromozomların yerlerine iyi olan kromozomlar alınarak gerçekleştirilir.

2.5. Çaprazlama (Crossover) ve Mutasyon (Mutation)

Genel olarak çaprazlama işlemi; kromozomların nasıl kodlanacağı belirlendikten sonra popülasyondaki kromozomlardan uygunluk değeri yüksek olanlar içerisinde rasgele seçilen iki birey arasında, yine rasgele seçilen bir noktaya göre karşılıklı gen alış-verişi olarak tanımlanır. Çaprazlama işlemi, uygunluk değeri iyi olan ve rasgele seçilen iki kromozomdan daha iyi uygunluk değerine sahip, yeni kromozomların elde edilmesi amacıyla yapılmaktadır. Yeni neslin uygunluk değerlerinin iyi çıkması ve problemin optimum çözüme ulaşabilmesi açısından çaprazlama işlemi son derece önem taşımaktadır. Bu aşamada, farklı kromozomlar alınarak çaprazlama işlemi yapıldığı için değişik özelliklerin test edilmesine ve hızlı olarak yeniden oluşmasına olanak sağlar. Bu işlem neticesinde ise, iki bireyin kopyası durumundaki kopyalanmış bireyler ile çeşitlilik sağlanmış ve en iyi çözüme yaklaşım sağlanmıştır.

GA programının sonuca ulaşmasında önemli olan başka bir etkende mutasyon işlemidir. Programın belirli bir alana yoğunlaşarak sonuca ulaşmayı engelleyen bazı durumlarda, problemin daha geniş bir alanda arama yapabilmesi için mutasyon işlemine başvurulur. Böylelikle popülasyondaki çözümlerin yerel optimuma düşmesi engellenmiş olur.

Kodlama yöntemine bağlı olarak, çaprazlama ve mutasyon işlemlerinin farklı şekilleri bulunabilmektedir.

2.5.1. İkili Kodlamada Çaprazlama

İkili kodlama yönteminde 4 çeşit çaprazlama gerçekleştirilebilmektedir.

2.5.1.1. Tek Noktalı Çaprazlama

Çaprazlama işlemine uğratılacak olan kromozom üzerinden herhangi bir nokta alınır. Bu noktaya olan bitler (genler) birinci kromozomdan, noktadan sonraki bitler ise ikinci kromozomdan alınarak çaprazlama işlemi tamamlanır.

$$11011011 \quad x \quad 11101010 = 11001010$$

Şekil 2.14 İkili kodlamada tek noktalı çaprazlama

2.5.1.2. İki Noktalı Çaprazlama

Bu tür çaprazlama işleminde iki tane kırılma noktası alınır. İlk noktaya kadar olan bitler birinci kromozomdan, iki nokta arasındaki bitler ikinci kromozomdan, kalanlar ise tekrardan birinci kromozomdan yeni bireye kopyalanırlar.

$$11011011 \times 11101010 = 11101011$$

Şekil 2.15 İkili kodlamada iki noktalı çaprazlama

2.5.1.3. Tek biçimli (Uniform , Düzenli) Çaprazlama

Bu çaprazlama yönteminde bitler rasgele şekilde, her iki kromozomdan yeni kromozoma kopyalanmaktadır.

$$11011011 \times 11101010 = 11011011$$

Şekil 2.16 İkili kodlamada tek biçimli (uniform) (düzenli) çaprazlama

2.5.1.4. Aritmetik Çaprazlama

Yeni nesilde yer alacak kromozomu elde edebilmek için değişik aritmetik işlemler uygulanır.

$$11011011 \times 11101010 = 11001010 \text{ (AND) VE işlemi}$$

Şekil 2.17 İkili kodlamada aritmetik çaprazlama

2.5.2. İkili Kodlamada Mutasyon

İkili sayı sisteminde kodlanmış bir bireyin aynı düzende mutasyon işleminde, rasgele seçilmiş bir veya birkaç bit, 0'dan 1'e veya 1'den 0'a değiştirilir.

$$(1\ 1\ 0\ 1\ 1\ 0\ 1\ 1) = > (1\ 0\ 0\ 1\ 1\ 0\ 1\ 1)$$

Şekil 2.18 İkili kodlamada mutasyon

2.5.3. Permutasyon Kodlamada Çaprazlama

Bu çaprazlama yönteminde ikili çaprazlamaya benzer bir mantık vardır. Bu yöntemde, rasgele bir çaprazlama yöntemi belirlenir. Bu noktaya kadar olan ifadeler birinci kromozomdan, kalan kısım ise ikinci kromozomdan alınarak yeni nesil kromozom oluşturulur.

$$(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9) + (4\ 5\ 3\ 6\ 8\ 9\ 7\ 2\ 1) = (1\ 2\ 3\ 4\ 5\ 9\ 7\ 2\ 1) = (123459768)$$

Şekil 2.19 Permutasyon kodlamada çaprazlama

2.5.4. Permutasyon Kodlamada Mutasyon

Bu mutasyon işleminde kromozom içerisinde iki gen seçilir ve seçilen bu iki gen kromozom içerisinde yer değiştirilir.

$$(1\ 2\ 3\ 4\ 5\ 6\ 8\ 9\ 7) = > (1\ 8\ 3\ 4\ 5\ 6\ 2\ 9\ 7)$$

Şekil 2.20 Permutasyon kodlamada mutasyon

2.5.5. Değer Kodlamada Çaprazlama

İkili kodlanmış kromozomlarda uygulanan tüm çaprazlama yöntemleri, bu kodlama yöntemi ile kodlanmış kromozomların çaprazlama işlemleri içinde uygulanabilir.

2.5.6. Değer Kodlamada Mutasyon

Bu kodlama yönteminde, seçilen değerlerden küçük bir sayı çıkarılarak veya eklenerek yeni bir kromozom elde edilir.

$$(1.29 \ 5.68 \ \mathbf{2.86} \ \mathbf{4.11} \ 5.55) \Rightarrow (1.29 \ 5.68 \ \mathbf{2.73} \ \mathbf{4.22} \ 5.55)$$

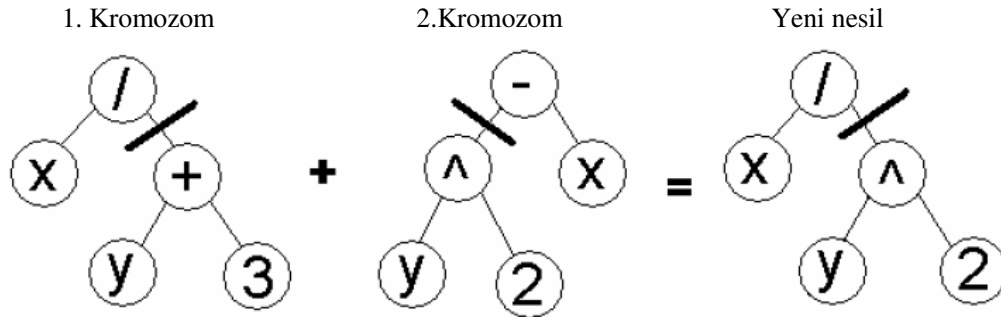
Şekil 2.21 Değer kodlamada mutasyon

2.5.7. Ağaç Kodlamada Çaprazlama

Çaprazlanacak kromozomlardan her ikisinden birer takas noktası seçilerek kromozom parçalanır. Bu noktaların atındaki kısımlar, aralarında değiştirilerek yeni kromozomlar elde edilir.

2.5.8. Ağaç Kodlamada Mutasyon

Bu tür kodlanmış kromozomda mutasyon işlemi uygulanırken, seçilen düğümlerdeki numaralar veya işlemler değiştirilir.



Şekil 2.22 Ağaç kodlamada çaprazlama

2.6. GA’larda Kontrol Parametreleri

GA’ların kontrol parametrelerinin değerlerinin seçimi de algoritmanın performansını etkileyen önemli etkenlerdendir. Basit bir GA’ın temel parametreleri şunlardır. Popülasyon büyüklüğü, maksimum generasyon sayısı, çaprazlama ve mutasyon oranlarıdır [3].

2.6.1. Popülasyon Büyüklüğü

GA programında seçilen popülasyon büyüklüğü, algoritmanın performansını iki şekilde etkilemektedir. Birincisi, popülasyon büyüklüğünün aşırı küçük seçilmesi algoritmanın, arama uzayının daralmasından dolayı belirli bir alt optimal noktaya doğru sürüklenmesine neden olacaktır. İkinci etkisi ise, popülasyonun aşırı büyük seçilmesi, GA programının bir generasyonu tamamlayabilmesi için gerekli olan zamanı büyültmekte, bu durumda özellikle gerçek zamanlı (real-time veya on-line) uygulamalarda büyük sorun oluşturmaktadır. Bu sebeplerden dolayı, popülasyon büyüklüğü seçilirken uygun bir değer ile algoritma başlatılmalıdır.

2.6.2. Çaprazlama Oranı

Çaprazlama oranı yeni nesli oluşturmak için var olan popülasyon içerisindeki kromozomlara uygulanacak çaprazlama operatörünün frekansını belirlemek amacıyla kullanılan bir parametredir. Bu oranın düşük olması, yeni nesle çok az sayıda yeni yapının girmesine neden olacaktır. Dolayısıyla üreme operatörü, algoritma içerisinde etkili operatör haline gelmekte ve araştırma işleminin sonuca ulaşma hızını düşürmektedir. Yüksek çaprazlama oranının kullanılmasında ise çaprazlama operatörü benzer veya daha iyi yapıları üretmeden kuvvetli olan yapılar çok hızlı olarak bozulduğundan algoritmanın performansını düşürmektedir.

2.6.3. Mutasyon Oranı

Mutasyon operasyonun oranı, GA programında iyi sonuç elde edebilmek için iyi kontrol edilmelidir. Bu oran çaprazlama oranında olduğu gibi, fazla yüksek veya düşük seçilmesine göre program üzerinde önemli etkiler oluşturmaktadır. Bunlardan yüksek mutasyon oranı, araştırmaya aşırı derecede rasgelelik katacağından araştırmayı çok hızlı ıraksayacaktır. Başka bir deyişle, popülasyonun gelişmesini değil kötüye gitmesine neden olacaktır. Eğer bu oran küçük tutulursa da, ıraksamayı aşırı düşüreceğinden arama uzayının tamamı taranmayacak ve sonuçta problem alt optimal çözüme doğru gidecektir.

3. SAYISAL FİLTRE TASARIMI

Fiziksel bir durum hakkında bilgi taşıyan, o durum hakkında bizlere bilgi veren, bir veya daha fazla değişkeni içeren fonksiyonlara işaret denir. İstenilen özelliklerde işaret üreten veya girişine uygulanan işarete göre arzu edilen özelliklerde çıkış üreten düzeneklere ise sistem adı verilir. İşaret işleme ise; işaretlerin bilgisayarlar ya da özel olarak üretilmiş sayısal donanımlarda, bir sayısal dizi olarak gösterilmesi ve bu donanımlardaki işaret dizisi üzerinde, çeşitli işlemler yapıldıktan sonra çıkışında istenilen bir bilginin bu diziden elde edilmesi işlemidir [10].

Sayısal işaret işlemenin amaçlarından biri, bir işaretin frekans spektrumu üzerinde belirli frekanslarda istenilen işlemleri yapan, yapılan bu işlemler neticesinde de istenilen özelliklerde sonuç alabilecek yapıyı oluşturmaktır. İşte oluşturulan bu yapıya sayısal filtre adı verilir. Sayısal filtreler yazılımla veya donanımla gerçekleştirilebilirler.

Sayısal filtreler impuls cevaplarına göre, FIR (sonlu impuls cevaplı) filtreler ve IIR (sonsuz impuls cevaplı) filtreler olarak ikiye ayrılmaktadırlar. Sayısal filtreler bir ayrık zamanlı sistem olarak incelenebilirler. Bu nedenle de ayrık zamanlı bir sistemin gerçekleştirilmesi sırasında izlenen adımlar, sayısal filtrenin hayata geçirilmesinde de uygulanabilir. Bir ayrık-zamanlı sistemin gerçekleştirilmesi üç ana safhadan meydana gelmektedir [11]. Bunlar;

1- Sistem özelliklerinin belirlenmesi:

Tasarımın ilk aşaması olan bu safhada sistemin yerine getirmesi gereken özellikler tanıtılır. Örneğin, üzerinde işlem yapılacak işaret için sistemin hangi frekans aralığını geçireceği, hangi frekans aralığını bastıracağı, ayrıca sistemin bu geçirme ve durdurma bantlarında ne kadarlık bir dalgalanmaya izin vereceği belirlenir.

2- Belirlenecek özellikleri yerine getirecek sistemin tasarımı:

Sistemden yerine getirilmesi istenen özellikler belirlendikten sonra, bu özellikleri sağlayacak sistemin tasarımı gerçekleştirilir. Ayrıca sistemin bu özellikleri yerine getirirken hangi tür sistem yapısına sonlu-impuls cevaplı (FIR) veya sonsuz-impuls cevaplı (IIR) ihtiyaç olduğuna karar verilir. Tasarımda kullanılan sistem yapısına karar verildikten sonra, yapısı gerçekleştirecek olan sistemin tasarım yöntemlerinden uygun olanı belirlenerek tasarım gerçekleştirilir. Ayrıca bu aşamada sistemin fiziksel olarak hayata geçirilebilmesi için gereken parametreler belirlenir.

3- Sistemin hayata geçirilmesi:

İstenen parametreler yerine getirildikten sonra sistem hayata geçirilir. Sistemin transfer fonksiyonu göz önüne alınarak, sistem sabit katsayılı FIR veya IIR olarak gerçekleştirilir.

Sayısal filtreler sahip oldukları diğer özelliklere göre de;

- 1- Doğrusal,
- 2- Doğrusal olmayan,
- 3- Zamanla-değişen,
- 4- Zamanla-değişmeyen sayısal filtreler

olarak sınıflandırılırlar [11].

Bir ayrık-zamanlı sistemde sistemin özelliklerini taşıyan transfer fonksiyonu, fark denklemlerinden, durum denklemlerinden ve devre yapısından bulunabilir.

Filtrenin giriş çıkış ilişkisi fark denklemi yardımıyla modellenabilir,

$$y(n) = \sum_{k=0}^M a(k)x(n-k) - \sum_{k=1}^N b(k)y(n-k) \quad (3.1)$$

Denlem (3.1)' den z -dönüşümü yardımıyla sayısal filtrenin transfer fonksiyonu $H(z)$ bulunabilir.

$$Y(z) = H(z)X(z) \quad (3.2)$$

Burada;

$$H(z) = \frac{\sum_{k=0}^M a(k)z^{-k}}{1 + \sum_{k=1}^N b(k)z^{-k}} \quad (3.3)$$

(3.2)' de verilen denklemin zaman domeni karşılığı konvolüsyon toplamıdır.

$$y(n) = \sum_{l=0}^{\infty} h(l)x(n-l) \quad (3.4)$$

Burada $h(n)$ sayısal filtrenin impals cevabıdır. Bir sayısal filtreden istenilen özellikler genellikle frekans domenin de verilir. Böyle bir durumda filtrenin frekans cevabını bulabilmek için, z değişkeni yerine $z = e^{j\omega}$ konularak istenilen sonuca ulaşılabilir.

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M a(k)e^{-j\Omega k}}{1 + \sum_{k=1}^N b(k)e^{-j\Omega k}} \quad (3.5)$$

Sayısal filtrenin frekans domeni karakteristiğini temsil eden frekans cevabı, $H(e^{j\Omega})$, Ω frekansının karmaşık değerli bir fonksiyonudur. $H(e^{j\Omega})$ 'nin ters Fourier dönüşümü alındığında elde edilen sonuç, filtrenin impuls cevabı olan $h(n)$ 'i vermektedir. Bu impuls cevabı da filtrenin zaman domeni özelliklerini temsil etmektedir. Fakat bir sayısal filtrenin özellikleri verilirken zaman domeni yerine frekans domeni özellikleri verilmesi tercih edilir. Bu amaçla bir sayısal filtrede temel amaç, filtrenin giriş ve çıkışı arasındaki fark denkleminde elde edilen transfer fonksiyonundaki (denklem 3.3) katsayılar olan $a(k)$, $k=0, 1, 2, \dots, M$ ve $b(k)$, $k=1, 2, \dots, N$ bulunmasıdır.

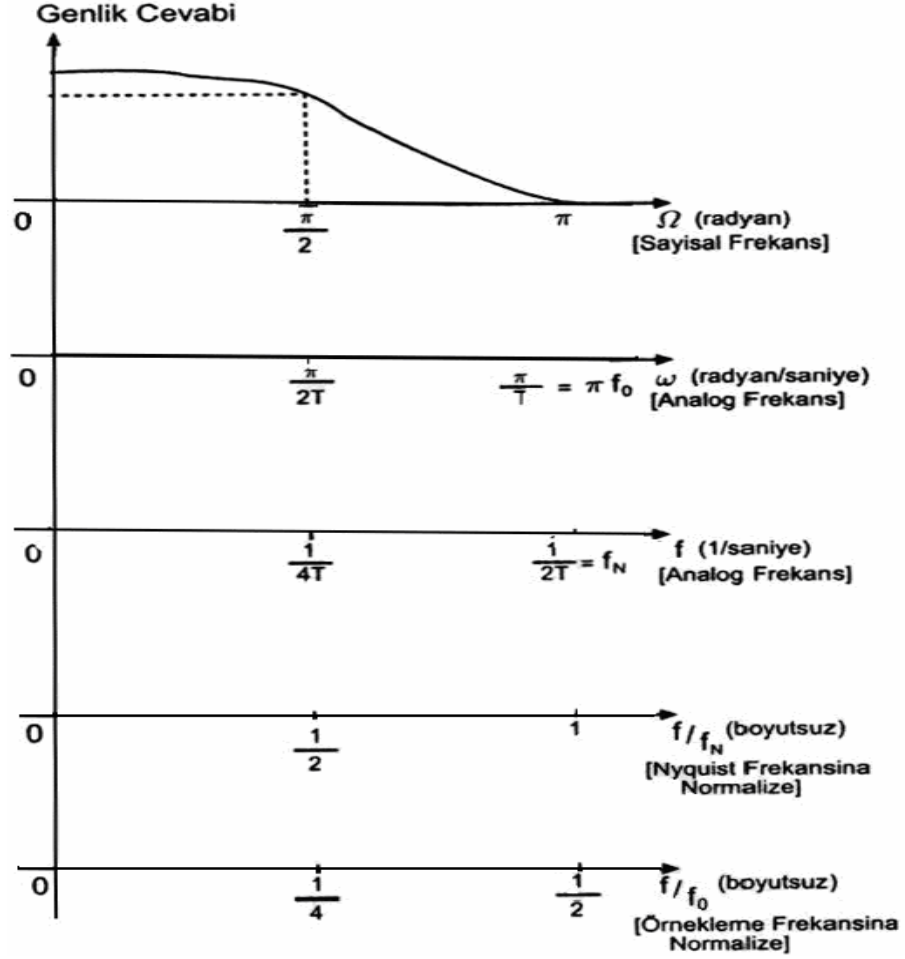
3.1. Sayısal Filtre Karakteristiklerinin Belirlenmesi

Sayısal filtre tasarımında, filtrenin frekans cevabı özelliklerinin belirlenmesinde, literatürde çeşitli yöntemler kullanılmaktadır [10, 11]. Bazı kaynaklarda filtrenin frekans cevabı özellikleri sayısal frekans cinsinden verilirken, bazı kaynaklarda filtre özellikleri analog frekans cinsinden verilmektedir. Bazen de frekans cevabının Nyquist frekansı veya örnekleme frekansına normalize edilerek tanımlandığı görülmektedir. Bütün bu gösterimler birbirine eşit olup, birbirlerine karıştırılmadan kullanılabilmeleri için, Şekil 3.1. dikkate alınmalıdır [10].

(3.5)'de verilen sayısal filtrenin frekans cevabı, $H(e^{j\Omega})$ 'nin periyodu 2π 'dir. Burada radyan cinsinden ifade edilen Ω , sayısal frekans olarak adlandırılır. Eğer filtreleme işlemi $\Delta t = T$ örnekleme aralığı olan bir işarete uygulanacak olursa, Ω sayısal frekansına karşılık gelen analog frekans ω (r/s) veya f (Hz) cinsinden ifade etmek mümkündür.

$$\begin{array}{ccccc} \text{Sayısal frekans} & = & \text{Analog frekans} & \times & \text{Örnekleme aralığı} \\ (\Omega) & & (\omega) & & (T) \end{array}$$

Nyquist frekansının $f_N = 1/2T$ olduğu hatırlandığında, filtre karakteristiği f/f_N olarak ifade edilebilir. Eğer frekans, Nyquist frekansına göre ifade edilirse, temel periyot $[-1, 1]$ arasındır [10]. Bazen de frekansın $f_0 = 1/T$ örnekleme frekansına oranı da (f/f_0) boyutsuz kullanılır. Bu gösterimde ise temel periyot $(-1/2, 1/2)$ aralığındadır. Analog bir filtre karakteristiği tanımlanırken kullanılan frekans aralığı $(-\infty, \infty)$ aralığında iken sayısal filtrelerde kullanılan frekans aralığı $(-f_N, f_N)$ dir. Literatürde sayısal frekans belirtmede farklı simgeler kullanılabilmektedir. Ω yerine λ veya Θ olabilir.



Şekil 3. 1 Kesim frekansı $\Omega_c = \pi/2$ radyan olan sayısal filtrenin çeşitli biçimlerde frekans cevabının gösterilmesi

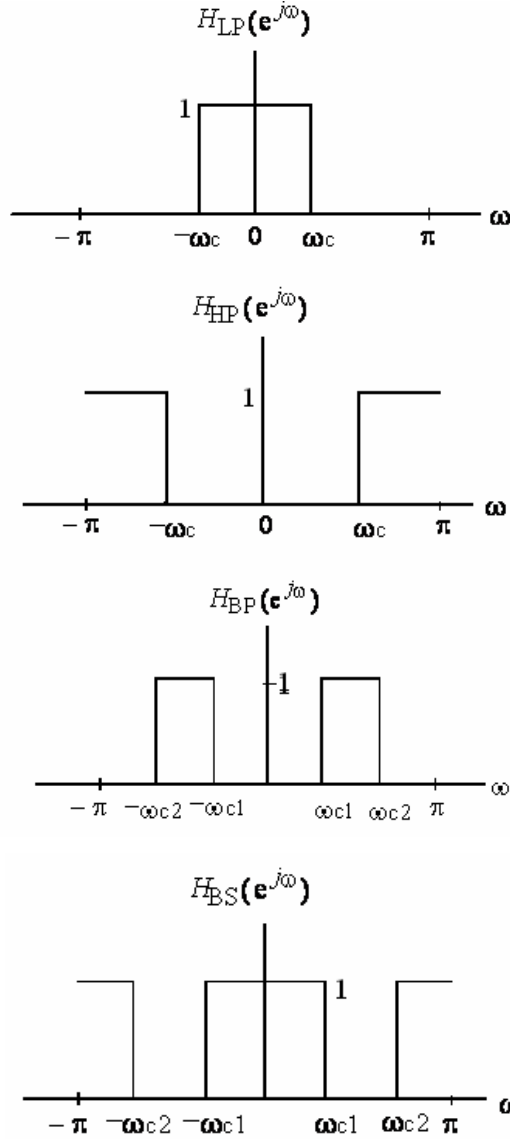
3.2. İdeal Sayısal Filtreler

Sayısal filtreler adlandırılırken, sahip oldukları frekans karakteristikleri dikkate alınır. Filtre karakteristiğinde, geçirme ve durdurma bantların da göstermiş olduğu davranışa göre filtreler [11],

1. Alçak geçiren filtreler (Low-pass filter, LPF)
2. Yüksek geçiren filtreler (High-pass filter, HPF)
3. Bant geçiren filtreler (Band-pass filter, BPF)
4. Bant durduran filtreler (Band-stop filter, BSF)

olarak isimlendirilirler. Bir ideal sayısal filtrenin frekans karakteristiğinin geçirme bandında, filtrenin girişine uygulanan işarete herhangi bir değişim uygulanmadan çıkışına aynen aktarılırken, durdurma bandında ise giriş işaretinin o frekans aralarındaki frekans bileşenleri

tamamen bastırılarak çıkışına aktarılır. Şekil-3.2’ de, bu türden ideal sayısal filtrelerin frekans karakteristikleri gösterilmiştir.



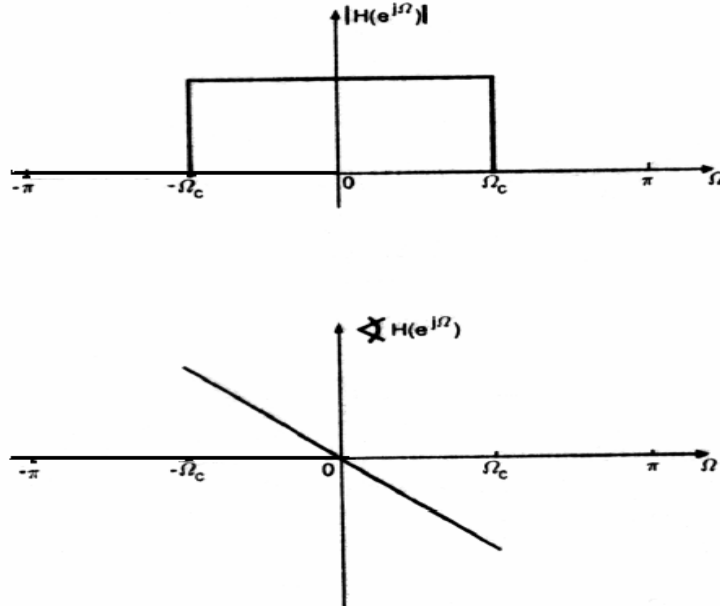
Şekil 3. 2 İdeal sayısal filtre frekans karakteristikleri; a) İdeal Alçak geçiren b) İdeal Yüksek geçiren c) İdeal Bant-geçiren d) İdeal Bant-durduran

3.2.1. İdeal Alçak ve Yüksek Geçiren Sayısal Filtreler

Genlik ve faz karakteristiği Şekil 3.3’deki gibi olan ideal alçak geçiren sayısal filtre göz önüne alınsın. Filtrenin transfer fonksiyonu [12];

$$H(e^{j\Omega}) = \begin{cases} 1.e^{-jn_0\Omega} & |\Omega| \leq \Omega_c \quad \text{için} \\ 0 & \Omega_c < |\Omega| \leq \pi \quad \text{için} \end{cases} \quad (3.6)$$

olarak verilir. Burada, n_0 pozitif bir sayıdır. Şekil 3.3' de $(0, \Omega_c)$ aralığı, filtrenin geçirme bandı olarak adlandırılır. Burada Ω_c , ideal alçak geçiren filtrenin geçirme bandı kesim frekansdır [10].



Şekil 3.3 İdeal alçak geçiren sayısal filtrenin genlik ve faz karakteristiği

İdeal alçak geçiren filtreler, karakteristiklerinden de anlaşılacağı gibi geçirme bandı dışında kalan tüm frekans bileşenlerini bastırırken, geçirme bandı içerisindeki bütün işaretleri ise hiçbir bozulmaya uğratmadan sadece belirli bir gecikme ilave ederek çıkışa aktarmaktadırlar. Tasarlanan bir alçak geçiren sayısal filtrenin frekans karakteristiğinde, geçirme bandı içerisinde genliğinin sabit kalması, faz cevabının ise ω 'nın doğrusal değişen bir fonksiyon olması istenir.

İdeal yüksek geçiren sayısal filtrede ise, kesim frekansını Ω_c olduğu ve filtre cevabının 2π ile periyodik olduğu durumda, filtrenin transfer fonksiyonu,

$$H_{yg}(e^{j\Omega}) = \begin{cases} 0 & \Omega < \Omega_c \\ 1 & \Omega_c < |\Omega| \leq \pi \end{cases} \quad (3.7)$$

şeklinde ifade edilir. İdeal yüksek geçiren sayısal filtrenin frekans cevabı, $H(e^{j\Omega}) = 1 - H_{ag}(e^{j\Omega})$ şeklinde alçak geçiren filtrenin frekans cevabı ile ilişkilendirildiğinde, ideal yüksek geçiren sayısal filtrenin frekans cevabı,

$$h_{yg}[n] = \delta[n] - h_{ag}[n] = \delta[n] - \frac{\sin \Omega_c n}{\pi n} \quad (3.8)$$

olarak elde edilir.

3.2.2. Fiziksel Gerçekleştirme

Denklem (3.6) daki gibi tanımlanan ideal alçak geçiren sayısal filtre fiziksel olarak gerçekleştirilebilir mi? Bir filtrenin fiziksel olarak gerçekleştirilebilmesi için filtrenin nedensel olması gerekmektedir [13]. Yani, filtrenin impuls cevabı

$$h(n) = 0, \quad n < 0 \quad \text{için} \quad (3.9)$$

şartını sağlaması gerekir. Filtrenin ters z-dönüşümü alındığında

$$h(n) = \frac{1}{2\pi j} \oint_C H(z) z^{n-1} dz \quad (3.10)$$

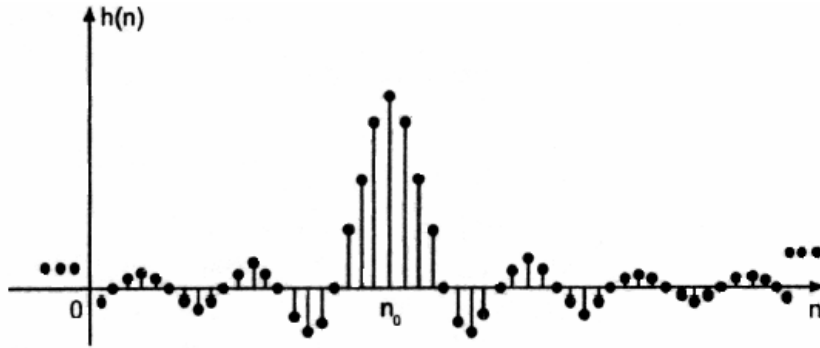
elde edilir. Burada C , kapalı konturu yakınsaklık bölgesi içerisindedir. Filtrenin fiziksel olarak gerçekleştirilebilmesi için kararlı olması gerektiğinden, yakınsaklık bölgesi $|z| = 1$ 'i kapsar. Bu nedenle C konturu, z -düzleminde birim daire olarak seçilebilir. $z = e^{j\Omega}$ değişimi uygulandığında (3.10) denklemi

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{jn\Omega} d\Omega \quad (3.11)$$

şeklinde yazılabilir. İdeal alçak geçiren sayısal filtrenin tanım denklemi olan (3.6), (3.11) de yerine yazıldığında

$$h(n) = \begin{cases} \frac{\sin(n - n_0)\Omega_c}{(n - n_0)\pi} & n \neq n_0 \text{ için} \\ \frac{\Omega_c}{\pi} & n = n_0 \text{ için} \end{cases} \quad (3.12)$$

elde edilir. Denklem (3.12) incelendiğinde filtrenin frekans cevabı olan $h(n)$ 'nin sağ taraflı bir dizi olmadığı görülür. Yani nedensellik şartı olan $n < 0$ için, $h(n) = 0$ 'ı sağlamamaktadır. Bu durumda filtre fiziksel olarak gerçekleştirilemez. Şekil 3.4' de, (3.12) denklemi ile verilen filtrenin impuls cevabı görülmektedir [13].



Şekil 3. 4 İdeal alçak geçiren sayısal filtrenin impuls cevabı

Şekil 3.4' den de görüldüğü gibi, filtrenin impuls cevabı $[-\infty, \infty]$ aralığında ∞ değer almaktadır. Dolayısıyla da filtre fiziksel olarak gerçekleştirilemez.

Böyle bir durumda filtrenin fiziksel olarak gerçekleştirilebilmesi için genlik ve faz karakteristikleri nasıl seçilmelidir?

Eğer bir filtrenin genlik ve faz karakteristikleri birbirinden bağımsız olarak seçilirlerse elde edilecek filtre fiziksel olarak gerçekleştirilemez. Sadece filtrenin genlik karakteristiği verilir de, faz karakteristiği serbest bırakılırsa istenilen özelliklere sahip filtreye mümkün olduğunca yaklaşılr. Ancak bu kez de istenilen özellikleri sağlayacak filtre çok yüksek dereceden olacaktır. Bu durum, tasarlanan filtrenin maliyetinin yüksek olmasına sebep olacaktır. Dolayısıyla pratikte tasarımı gerçekleştirilen filtrenin nedensel olmasından başka birde basit rasyonel katsayılı olması istenir.

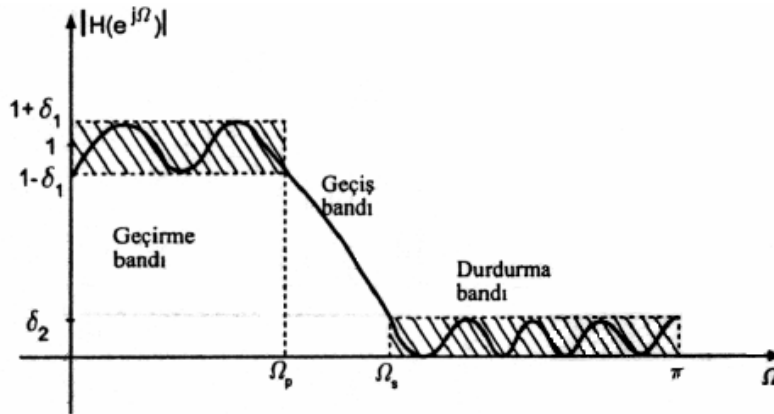
Filtrenin nedensellik, basitlik koşullarına ek olarak birde, kararlılık şartının eklenmesi gerekmektedir. Bu üç özelliğin de aynı anda gerçekleşmesi çok zordur. Bu nedenle pratik filtre tasarlarken filtrenin frekans karakteristiğinin tam olarak değil de, bazı kabul edilebilir toleranslarla birlikte verilmesi gerekir.

3.3. İdeal olmayan Sayısal Filtreler

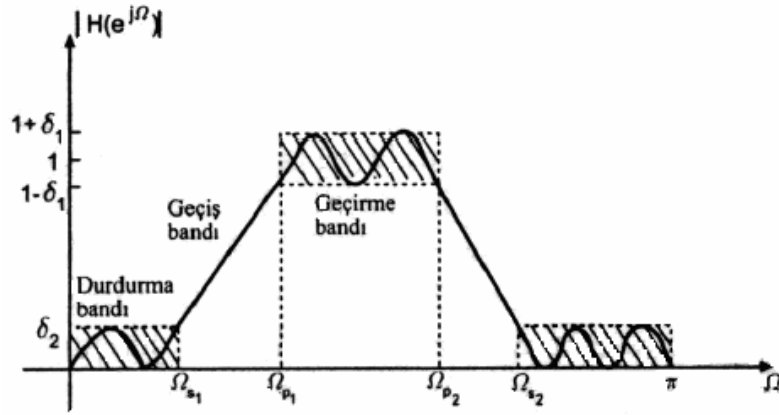
Pratik olarak tasarlanan filtreler, istenen özellikleri ancak yaklaşık olarak gerçekleştirirler. Ayrıca tasarlanan filtrelerin birçoğunda, ideal filtrelerin sahip olduğu frekans karakteristiğinin ki gibi olması istenmez. Bunun yerine, filtrenin geçirme ve durdurma bantların da bazı esneklik olması istenir ve geçirme bandı ile durdurma bantları arasında ideal filtrelerde olduğu gibi keskin bir kesim yerine daha kademeli bir geçişin olması istenir. Bu nedenle, tasarlanan filtrelerde geçirme bandı ile durdurma bantları arasına birde geçiş bandı ilave edilir. O halde alçak geçiren bir filtre karakteristiği, Şekil 3.5'deki gibi taralı alan içerisinde kalmalıdır. Bu şekilden de görüleceği gibi, filtrenin geçirme bandının 1 ' e göre $\pm \delta_1$ ' lik bir dalgalanma ile durdurma bandı içerisinde 0 ' a göre δ_2 ' lik bir dalgalanmaya izin verilir. Geçirme bandı içerisindeki birim genliğe göre olan bu dalgalanmaya geçirme bandı dalgalanması, durdurma bandındaki 0 ' a göre oluşan dalgalanmaya ise durdurma bandı dalgalanması adı verilir [11].

Ω_p ve Ω_s frekansları sırasıyla, alçak geçiren filtrenin geçirme bandı ve durdurma bandı kenar frekanslarıdır. Ω_p dan Ω_s e kadar olan mesafede ise geçirme bandı ile durdurma bantları arasındaki geçiş için bırakılmıştır. Filtrenin frekans karakteristiğinde bırakılan bu geçiş aralığına, geçiş bandı adı verilir.

Ayrıca $\Omega=0$ ve $\Omega=\pi$ frekansları dışındaki frekans aralığını geçiren ve bu frekanslar dışında kalan bütün frekansları bastıran filtreye ise bant-geçiren filtre adı verilmektedir.



Şekil 3. 5 Alçak geçiren filtrenin frekans cevabı



Şekil 3. 6 Bant geçiren filtrenin frekans cevabı

3.4. Geçici Durum Davranışı

Sayısal filtrelerin transfer fonksiyonu olan $H(z)$ ' nin genlik ve faz özellikleri, sürekli-durum cevabıyla alakalıdır. Sistemin sürekli-duruma ulaşana kadar ki durum cevabına ise sistemin geçici-durum cevabı adı verilir. Sistem tasarımında elbette ki geçirmiş olduğu geçici durum davranışı önemlidir. Şekil 3.7 de sistemin geçici durum davranışı gösterilmektedir.

Şekilden de görüleceği gibi sistemin sürekli-duruma ulaşması için geçirdiği geçici-durumun belirlenmesinde yükselme zamanı, yerleşme zamanı ve aşma gibi kavramların hesaplanması gerekmektedir [10]. Bir sistemin geçici-durum davranışında;

n_r ; sistemin yükselme zamanını

n_s ; sistemin yerleşme zamanını

y_{ss} ; sistemde ki aşma miktarına karşılık gelmektedir.

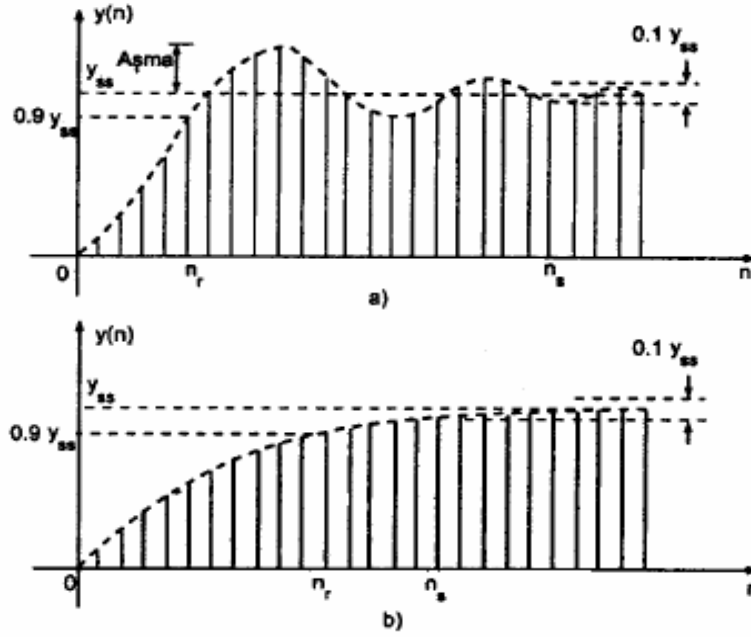
Eğer bir sistemin sürekli-durum cevabı;

$$y_{ss}(n) = \lim_{n \rightarrow \infty} y(n) \quad (3.13)$$

ise; sistemin yükselme zamanı;

$$y(n_r) \geq 0.9 y_{ss}(n) \quad (3.14)$$

şartını gerçekleştiren ilk n_r süresidir.



Şekil 3. 7 Sayısal filtrelerde geçici durum davranışı a) geçici cevapta aşma
b) aşmanın olmadığı geçici cevap

Sistemin yerleşme zamanı olan n_s ise;

$$|y(n) - y_{ss}| \leq 0.05 y_{ss} \quad (3.15)$$

şartını sağlayan ilk zamandır. İyi tasarlanmış bir sistemde küçük yerleşme zamanı, küçük yükselme zamanı ve küçük miktar da aşma olması istenir.

Bir sistemin tasarımı, hem sürekli-durum hem de geçici-durum davranışı birlikte düşünülerek gerçekleştirilir. Uygulamalarda filtrenin sadece istenilen genlik karakteristiği ve geçici-durum cevabı kontrol edilerek sonuca varılır. Eğer bu iki özellik arzu edilen değerlerde ise istenen tasarım gerçekleştirilmiştir. Aksi takdirde tasarım işlemi tekrarlanır.

4. AYRIK ZAMANLI FIR FİLTRE TASARIMI

Sonlu impals cevabına sahip olan filtreler FIR filtre adı verilmiştir. Bir FIR filtrenin transfer fonksiyonu, z^{-1} in fonksiyonu ve reel katsayılı olması gerekmektedir. Bu durumda filtrenin transfer fonksiyonu olan $H(z)$, N terim için,

$$H(z) = \sum_{n=0}^{N-1} h[n]z^{-n} \quad (4.1)$$

olur [12]. Buradan da anlaşılacağı gibi FIR filtreler de çıkış, girişin o andaki ve daha önceki değerlerine bağlı olup, çıkışın değerine bağlı değildir. Yani gerçekleştirilen filtre tekrarsız (özyinesiz) bir filtredir [10].

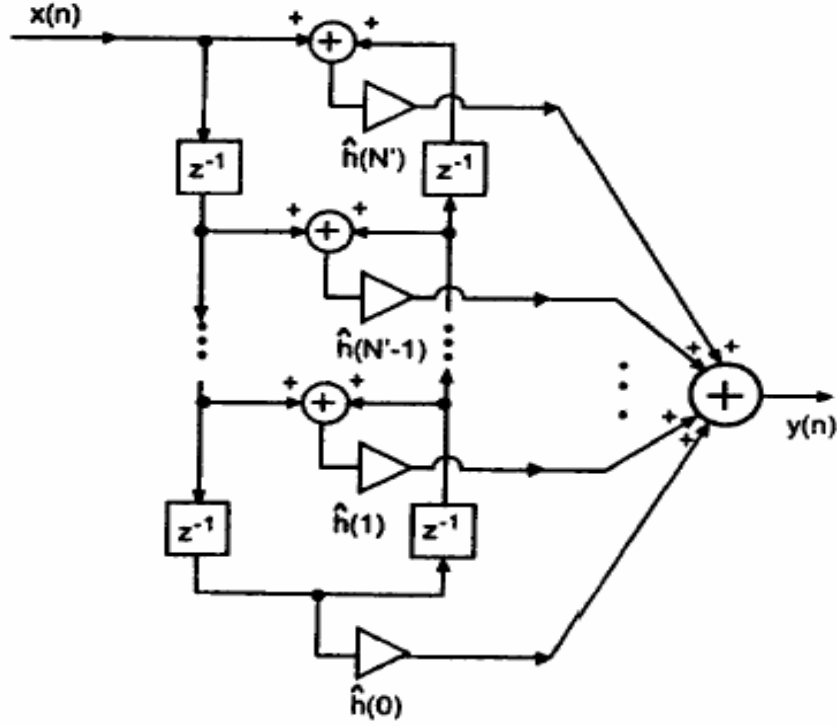
FIR filtrenin impals cevabının sonlu sayıda olduğu bilindiğine göre, bu değerlerin tek sayı olduğu varsayıp, (4.1) denkleminde $N' = (N-1)/2$ olarak tanımlansın. Eğer $h'(n) = h(n + N')$ nedensel olmayan filtrenin impals cevabı olarak tanımlandığında (4.1) denklemi

$$\begin{aligned} H(z) &= \sum_{n=0}^{2N'} h(n)z^{-n} \\ &= \sum_{k=-N'}^{N'} h(k + N')z^{-(k+N')} \\ &= z^{-N'} \sum_{k=-N'}^{N'} h'(k)z^{-k} = z^{-N'} H'(z) \end{aligned} \quad (4.2)$$

şeklinde yazılabilir. (4.2)' deki nedensel olmayan sayısal filtrenin transfer fonksiyonu $H'(z)$ ise

$$H'(z) = \sum_{k=-N'}^{N'} h'(k)z^{-k} \quad (4.3)$$

olarak tanımlanabilir. Şekil 4.1' de, bir FIR filtrenin gerçekleştirilmesinde kullanılan blok diyagramı verilmiştir.



Şekil 4. 1 Simetrik impuls cevabına sahip doğrusal fazlı FIR filtrenin tasarlanması

4.1. FIR Filtrelerin Avantajları ve Dezavantajları

Sayısal FIR filtrenin kullanılmasının nedenleri arasında sahip olduğu bazı avantajlar vardır. Bu avantajlar [10]:

1. Sayısal FIR filtreler, önceden belirlenmiş genlik ve faz cevabı özellikleri sağlayabilecek şekilde ve lineer fazlı olarak tasarlanabilirler.
2. FIR filtreler, hem tekrarlı hem de tekrarsız olarak gerçekleştirilebilirler. Tekrarlı gerçekleştirilen bir FIR filtre de, tarak (comb) ve rezonatör bankası kullanılır. Tekrarsız gerçekleştirmede ise doğrudan konvolüsyon ve Hızlı Fourier Dönüşüm (FFT, Fast Fourier Transform) yöntemlerinden birisi kullanılarak gerçekleştir.
3. Tekrarsız olarak gerçekleştirilen bir filtrenin transfer fonksiyonunda, sadece sıfırları olup kutupları olmadığından bu tür gerçekleştirilen filtreler daima kararlıdır.
4. Bir FIR filtrede, doğal olarak yuvarlatma ve kuvantalama hataları olacaktır. Ancak tekrarsız olarak gerçekleştirildiklerinden dolayı, yani herhangi bir andaki çıkış girişin

sadece o andaki değerine bağlı olup, çıkış geri besleme olarak tekrardan sisteme eklenmediğinden dolayı bu hatalarda önemsizdir.

5. Keskin kesim frekanslı filtre tasarımında FIR filtreler, katsayı hatalarına daha az duyarlıdır.

FIR filtrelerin sahip oldukları bu avantajların yanında, istenilen özellikleri sağlayan bir IIR filtreden daha yüksek derecede gerçekleştirilmeleri sahip oldukları dezavantajlarıdır.

Buda kullanılan eleman sayısını ve dolayısıyla da maliyetini arttırmaktadır.

FIR veya IIR filtrelerin her ikisi de, hem tekrarlı hem de tekrarsız olarak gerçekleştirilebilirler. Fakat genellikle FIR filtreler tekrarsız, IIR filtreler ise tekrarlı olarak gerçekleştirilirler. Tekrarlı ve tekrarsız gerçekleştirmenin tanımını yapmak gerekirse;

- a) Tekrarlı (Recursive – Özyineli) Gerçekleştirim:

Filtrenin çıkışı $y(n)$, tekrarlı gerçekleştirmede;

$$y(n) = f(y(n-1), y(n-2), \dots, x(n), x(n-1), x(n-2), \dots) \quad (4.4)$$

şeklinde yazılabilir. Bu denklem bizlere, filtre çıkışı $y(n)$ ' nin herhangi bir andaki değeri, geçmiş anlardaki filtre çıkışları ($y(n-1)$, $y(n-2)$, ...) , şimdiki ($x(n)$) ve daha önceki geçmiş filtre girişleri ($x(n-1)$, $x(n-2)$, ...) cinsinden ifade edilebilir.

- b) Tekrarsız (Nonrecursive – Özyinesiz) Gerçekleştirme:

Filtre çıkışı $y(n)$ ' nin herhangi bir andaki değeri, girişin sadece o andaki ve daha önceki değerlerine bağlı olduğu durumu gösterir. Bu durum

$$y(n) = f(x(n), x(n-1), x(n-2), \dots) \quad (4.5)$$

şeklinde yazılabilir.

FIR filtre tasarımında iki yöntem kullanılmaktadır. Bunlar Fourier serisi yöntemi ve Frekans örnekleme yöntemleridir. Bu yöntemlerin birbirlerine göre avantaj ve dezavantajları vardır. Bir sistemin tasarlanmasında kullanılacak yöntem, filtrenin istenilen karakteristiğine bakılarak seçilmektedir. Buna göre hangi tür tasarım tekniği kullanılacaksa, bu tekniğe uygun olan FIR filtre tasarım yöntemi tercih edilir.

4.2. Fourier Serisi Yöntemi

Bir sayısal filtrenin frekans cevabı $H(e^{j\omega})$, Ω sayısal frekansının periyodik bir fonksiyonu olup periyodu 2π dir. Periyodik olan bu frekans cevabı $H(e^{j\omega})$, Fourier serisi olarak yazılırsa,

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jn\Omega} \quad (4.6)$$

olur. Bu durumda istenen filtrenin frekans cevabına karşılık gelen impuls cevabı ise,

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{jn\Omega} d\Omega \quad (4.7)$$

ifadesinden bulunabilir. Denklem (4.6)' da $e^{j\Omega} = z$ yazıldığında frekans cevabına karşılık gelen filtrenin transfer fonksiyonu;

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \quad (4.8)$$

şeklinde bulunur. Bu sonuç, elde edilen filtrenin impuls cevabını $-\infty$ dan başlayıp ∞ ' a kadar devam ettiğini gösterir. Bu durumda da filtre fiziksel olarak gerçekleştirilemez. Bir filtrenin fiziksel olarak gerçekleştirilebilmesi için bu impuls cevabının sınırlı sayıda olması gerekmektedir. Yukarıda bulduğumuz impuls cevabının sınırlı sayıda olabilmesi için belirli bir gecikmeyle çarpılması gerekmektedir. Böylelikle sonsuz uzunluğa sahip olan sayısal filtre sonlu sayıda bir impuls cevabına sahip olacaktır. Denklem (4.8) de [10],

$$h(n) = 0, \quad |n| > N' = \frac{N-1}{2} \quad (4.9)$$

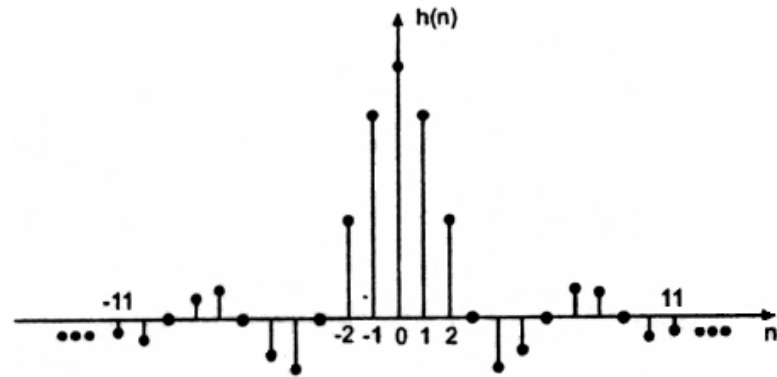
kabul edildiğinde;

$$H(z) = h(0) + \sum_{n=1}^{N'} [h(-n)z^n + h(n)z^{-n}] \quad (4.10)$$

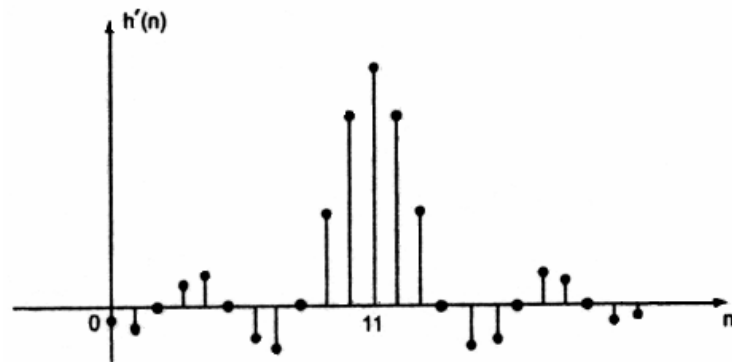
şeklinde bulunur. $H(z)$ nin $z^{-N'}$ ile çarpılmasıyla filtrede nedensellik şartı sağlanmış olur.

$$H'(z) = z^{-N'} H(z) \quad (4.11)$$

Bu sonuç sonlu gecikme eklenerek elde edilen sayısal filtrenin genlik cevabı olan $H'(z)$ 'nin, $H(z)$ ile aynı kaldığını fakat faz cevabının ise $N'T$ kadar kaydırıldığını gösterir.



Şekil 4. 2 Sonsuz uzunluktaki ideal alçak geçiren sayısal filtrenin impals cevabı



Şekil 4. 3 Sonlu uzunlukta (N=23) tasarlanan FIR filtrenin impals cevabı

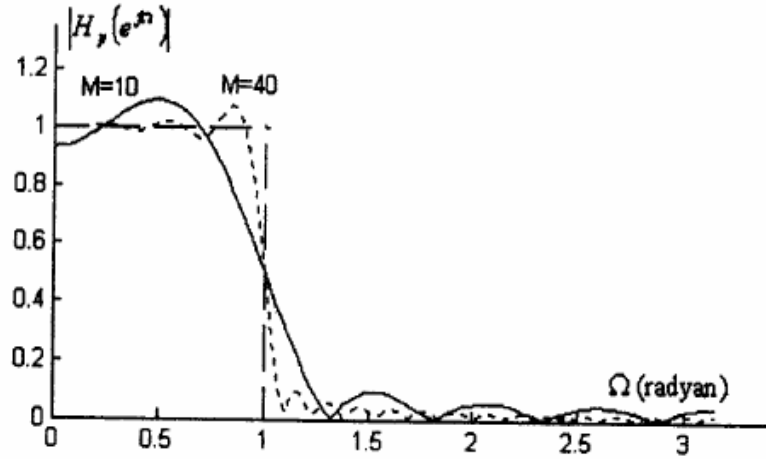
Pratikte tasarlanan filtrenin impuls cevabı belirli bir aralıkta tanımlanır ve tanımlandığı bu aralıkta ideal filtrenin impuls cevabına eşittir. Bu durumda pratikte tasarlanacak filtrenin impuls cevabı [11];

$$h_p[n] = \begin{cases} \frac{\sin \Omega_c n}{\pi n}, & -M/2 \leq n \leq M/2 \\ 0, & \text{aksidurumda} \end{cases} \quad (4.12)$$

şeklinde bulunur. Bulunan sonuçtan görülüyor ki, impuls cevabı sonlu sayıya sahiptir ve dolayısıyla bu filtre bir FIR filtredir. Pratikte tasarlanan filtre 2π ile periyodik olacak şekilde,

$$H_p(e^{j\Omega}) = \sum_{n=-M/2}^{M/2} h_p[n] e^{-j\Omega n} \quad (4.13)$$

olarak tanımlanır. Pratik olarak tasarlanan filtrenin frekans cevabı 2π ile periyodik olduğundan, $h_p[n]$ değerleri Fourier serisi katsayılarına karşılık gelmektedir. Dolayısıyla da bu yöntem Fourier serisi yöntemi adı verilmiştir. Şekil 4.4' de, $\Omega_c = 1$ ve farklı M değerleri için tasarlanmış pratik filtrelerin frekans cevapları gösterilmiştir.



Şekil 4. 4 Fourier serisi yöntemi kullanılarak tasarlanan FIR filtrelerin genlik cevapları

Görüldüğü gibi M değeri arttıkça, pratik filtrenin genlik cevabı ideal filtrenin genlik yanıtına yaklaşmaktadır.

4.2.1. Pencere Fonksiyonu Kullanımı

Pratikte tasarlanan filtrede kullanılan örnek değeri ne kadar fazla olursa, filtrenin genlik cevabı ideal filtrenin genlik cevabına o kadar yaklaşmaktadır. Pratik filtrenin frekans cevabı, ideal filtrenin impuls cevabından sınırlı sayıda örnek değeri almaktadır. Dolayısıyla tasarlanan filtrenin keskin kesim frekanslarında salınımlar oluşmaktadır. Meydana gelen bu salınımları ortadan kaldırmak için pencere fonksiyonu kullanılır [12].

Pencere fonksiyonu kullanılarak, pratik filtrenin impuls cevabı kırılıp sonlu sayılı bir impuls cevabı elde edilir. Bu ise bir $w(n)$ kırpm penceresi yardımıyla yapılabilir.

$$h_p[n] = h_{id}[n]w[n] \quad (4.14)$$

Kullanılan pencere fonksiyonun frekans cevabı

$$w[n] = \begin{cases} 1, & -M/2 \leq n \leq M/2 \\ 0, & \text{aksitakdirde} \end{cases} \quad (4.15)$$

şeklinde tanımlandığında, bir dikdörtgen biçimi aldığından bu pencere fonksiyonuna dikdörtgen penceresi adı verilmektedir [14].

Pencere fonksiyonu, genel olarak geçirdiği aralığı 1 ile çarpmakta, bastırdığı aralığı ise 0 ile çarparak pencerelemektedir. Kullanılan bu pencere fonksiyonu yardımıyla, ideal filtrenin keskin kesim frekanslarında meydana gelen Gibbs etkisi yok edilmektedir. FIR filtrelerin Fourier serisi yöntemiyle gerçekleştirilmelerinde çok sayıda pencere fonksiyonu kullanılmaktadır. Bunlar içerisinde en yaygın olarak kullanılan pencere fonksiyonları ve denklemleri aşağıda verilmiştir [14,15].

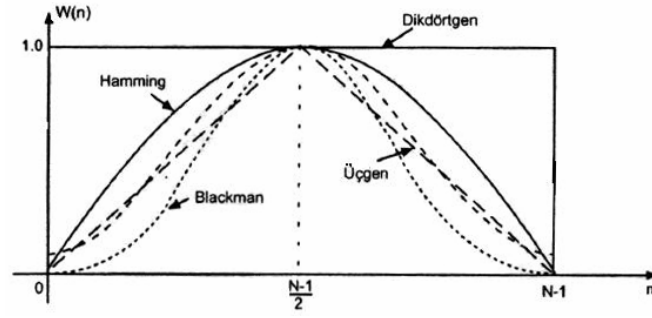
$$\text{Üçgen penceresi} \quad ; \quad w[n] = 1 - \left| \frac{2n}{M} \right| \quad (4.16)$$

$$\text{Blackman penceresi} \quad ; \quad w[n] = 0.42 + 0.5 \cos\left(\frac{2\pi n}{M}\right) + 0.08 \cos\left(\frac{4\pi n}{M}\right) \quad (4.17)$$

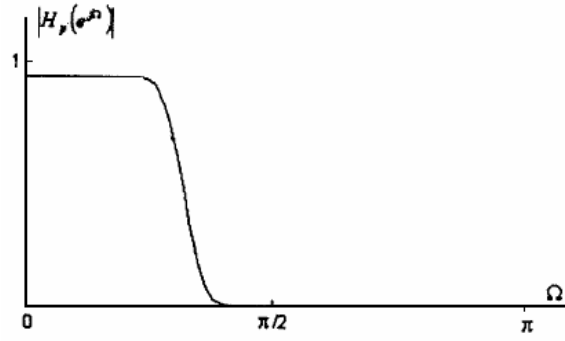
$$\text{Hamming penceresi} \quad ; \quad w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{M}\right) \quad (4.18)$$

$$\text{Hanning penceresi} \quad ; \quad w[n] = 0.5 + 0.5 \cos\left(\frac{2\pi n}{M}\right) \quad (4.19)$$

Şekil 4.5' de yaygın olarak kullanılan pencere fonksiyonlarının genlik cevapları görülmektedir [10].

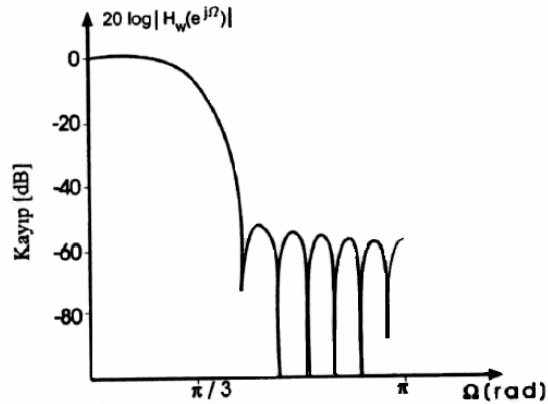


Şekil 4. 5 FIR filtre tasarımında kullanılan pencere fonksiyonları



Şekil 4. 6 M=40 değeri için Hamming penceresi kullanılarak tasarlanan FIR filtrenin genlik cevabı

Şekil 4.6' da Hamming penceresi kullanılarak elde edilen FIR sayısal filtrenin genlik cevabı görülmektedir. Şekilden de görüldüğü gibi pencere fonksiyonu kullanılmadan gerçekleştirilen sayısal filtrenin keskin kesim frekanslarında meydana gelen salınımlar, kullanılan pencere fonksiyonu yardımıyla ortadan kaldırılmıştır. Şekil 4.7' de ise frekans cevabı logaritmik olarak gösterilmiştir.



Şekil 4. 7 Hamming pencere fonksiyonu kullanılarak tasarlanan alçak geçiren FIR filtrenin (N=23 değeri için) genlik cevabı

4.3. Frekans Örnekleme Yöntemi

Sonlu sayıda impuls cevabına sahip olan bir filtrenin frekans spektrumu, ayrık Fourier dönüşümü (AFD) yardımıyla bulunabilmektedir. Bu işlemi yapmamızda ki amaç, filtrenin spesifikasyonu olan $H(e^{j\Omega})$ için, impuls cevabı olan $h(n)$ 'nin doğrudan bulunması işleminin zorluğundandır. Bu durumdan kurtulmak için, $h(n)$ 'i bulmaksızın doğrudan $H(e^{j\Omega})$ üzerinde işlem yapmak gerekir.

Bu anlamda frekans örnekleme aşağıda verilen iki temel aşamadan oluşmaktadır [10]. Bunlar:

1-Filtrenin sahip olduğu frekans cevabının eşit aralıklarla N adet örnek değerinin alınması.

2-Eşit aralıklarla alınan bu frekans değerlerinin ters AFD' sinin bulunması.

Yukarıda yazılan bu işlemlerin denklemler yardımıyla ifade edilmesinde ayrık Fourier dönüşümü, işaretin frekans cevabı

$$\Omega = \frac{2\pi}{N}k \quad k=0, 1, 2, \dots, N-1 \quad (4.20)$$

örnek değerleri alınarak N adet ayrık frekans değeri hesaplanmaktadır. Bir işaretin ayrık Fourier dönüşümü;

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)kn}, \quad k=0, 1, \dots, N-1 \quad (4.21)$$

bağıntısı yardımıyla hesaplanabilir. Bu denklemin ters ayrık Fourier dönüşümü ise

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j(2\pi/N)kn}, \quad n=0, 1, 2, \dots, N-1 \quad (4.22)$$

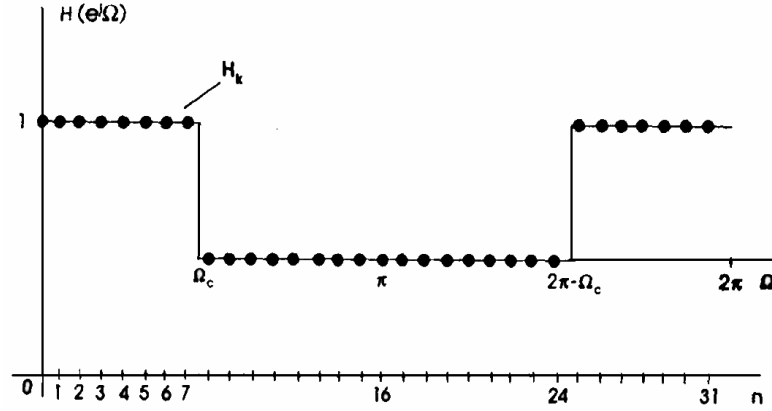
şeklinde bulunur.

Fourier serisi yönteminde yapılan işlemler kısaca özetlenecek olursa:

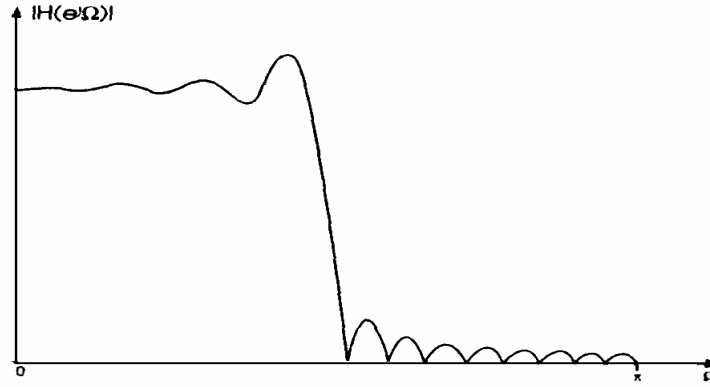
Filtre tasarımında öncelikle filtrenin frekans cevabı belirlenmektedir. Bu işlem sonrasında frekans cevabının 2π ' lik periyodunda, N adet örnek değeri alınmaktadır. Alınan bu frekans değerleri AFD katsayıları olarak kabul edilebilir [11]. Frekans cevabının örnek değerleri bulunduktan sonra, bu değerlerin ters AFD [16] alınarak istenen filtrenin N adet impuls cevabı

bulunmuş olur. Bu yöntemde de frekans cevabı üzerinden örnek değerleri alınıp ve değerler üzerinden işlemler yapıldığı için frekans örnekleme yöntemi adı verilir.

Şekil 4.8 ve 4.9’ da, frekans cevabı üzerinden eşit aralıklı örnek değerlerinin alınması ve bu değerlerden frekans cevabının elde edilmesi görülmektedir.



Şekil 4. 8 İdeal alçak geçiren filtreden eşit aralıklı örnek değerlerinin alınması ($N=23$)

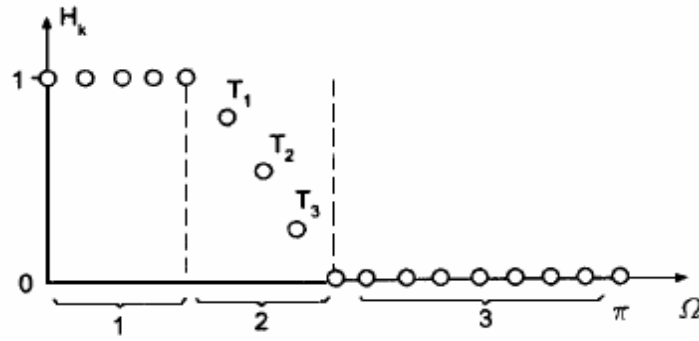


Şekil 4. 9 Şekil 4.8’ deki frekans örneklerine karşılık gelen filtrenin frekans cevabı

FIR filtre gerçekleştirilmesinde kullanılan Fourier serisi yöntemi ile frekans örnekleme yöntemi, benzer olmakla beraber farklılıklarda göstermektedirler. Fourier serisi yönteminde, filtre tasarımında tüm frekans değerleri için en uygun tasarımın gerçekleştirilmesi sağlanmaktadır. Bunun yanında frekans örnekleme yönteminde ise sadece örnek alınan frekans örneklerine eşit olması şartına göre filtre tasarlanmaktadır. Bu nedenlerden dolayı frekans örnekleme yöntemi yardımıyla tasarlanmış filtrelerdeki ortalama hata Fourier serisi yardımıyla tasarlanan filtrelerin ortalama hatalarından daha, fazla olmaktadır.

4.4. Filtre Tasarımının İyileştirilmesi

Filtre tasarımında yaklaşıklık yöntemlerinin kullanılmalarının sebebi, filtrelerin keskin kesim frekanslarında meydana gelen salınımlardan kaynaklanmaktadır. Alçak geçiren filtrenin frekans cevabı göz önüne alındığında, bu filtrede meydana gelen salınımlar geçiş bandının çok az olmasından kaynaklanmaktadır. Alçak geçiren filtrenin geçirme bandından alınan son örnek bire eşit iken, durdurma bandında alınan ilk örnek sıfıra eşittir. Frekans domenindeki süreksizlik uzun bir impuls cevabına karşılık gelmektedir. Bunun sonucunda ise, frekans örnekleme yöntemi kullanılarak bulunan filtrenin impuls cevabı zaman domenin de örtüşmeye uğrayacak ve bozulacaktır. Meydana gelen bu hatanın azaltılabilmesi için geçirme bandı ile durdurma bantları arasına yerleştirilen geçiş bandının geniş olması sağlanır. Böylelikle de geçiş bandından alınan örnek sayısı artmakta ve oluşan bozulma azalmaktadır [10].



Şekil 4. 10 (T_1, T_2, T_3) değişken katsayılara sahip olan ve frekans örnekleme yöntemi ile tasarlanan filtrenin frekans cevabı

5. AYRIK ZAMANLI IIR FİLTRE TASARIMI

Sonlu olmayan ve sonsuza kadar devam eden bir impuls cevabına sahip olan filtreler sonsuz impuls cevablı (Infinite Impulse Response, IIR) filtreler adı verilir. Bu türdeki bir filtrenin sabit katsayılı fark denklemi ile giriş/çıkış ilişkisi [12],

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \quad (5.1)$$

şeklinde gösterilir. Sistemin sahip olduğu transfer fonksiyonu ise,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (5.2)$$

olarak bulunur. Elde edilen bu ifade daha açık yazılırsa;

$$H(z) = \frac{a(0) + a(1)z^{-1} + a(2)z^{-2} + \dots + a(M)z^{-M}}{1 + b(1)z^{-1} + b(2)z^{-2} + \dots + b(N)z^{-N}} \quad (5.3)$$

elde edilir. Sayısal filtrenin özellikleri; filtrenin genlik, faz karakteristiği ve geçici durum davranışı bilgileridir. Fakat gerçek bir filtre tasarımında bu özelliklere ek olarak kararlılık, nedensellik ve basit rasyonel katsayılı olması gibi koşullardan dolayı, filtrenin yalnızca genlik karakteristiğine bakılır. Dolayısıyla da IIR filtre tasarımının da amaç, denklem (5.2) ile gösterildiği gibi a_k ve b_k katsayılarının bulunmasıdır [10].

IIR filtre tasarımında kullanılan yöntemler:

- Değişmez-impuls cevabı yöntemi
- Değiştirilmiş değişmez-impuls cevabı yöntemi
- Uygunlaştırılmış z-dönüşümü yöntemi
- Bilineer dönüşüm

IIR filtre tasarımında genellikle, istenilen özellikleri sağlayacak bir sürekli-zamanlı filtre tasarlandıktan sonra elde edilen bu filtrenin, ayrık-zamanlı bir filtreye dönüştürülmesi

yöntemi tercih edilmektedir. Bu yöntemin fazlaca kullanılmasının nedeni, sürekli-zamanlı IIR filtre yapımında çok fazla yöntemin bulunmasıdır. Ancak kullanılan bu yöntemde her zaman sürekli-zamanlı sistemden, ayrık-zamanlı sisteme dönüşümün basit olarak gerçekleşememesi bu yöntemin dezavantajıdır.

5.1. Değişmez İmpuls Cevabı Yöntemi ile IIR Filtre Tasarımı

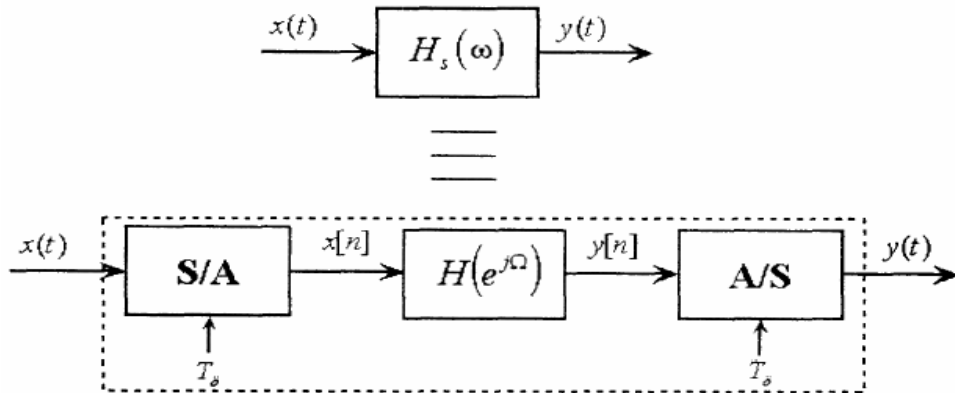
Analog alçak geçiren filtrenin transfer fonksiyonu $H_a(s)$ ile gösterilirse, filtrenin impuls cevabı transfer fonksiyonun ters Laplace dönüşümü [17] alınarak bulunabilir.

$$h_a(t) = \mathcal{E}^{-1}[H_a(s)] \quad (5.4)$$

Elde edilen bu impuls cevabı, $t=nT$ aralıklarıyla örneklendiğinde elde edilen $h_a(nT)$ tasarlandığında elde edilmek istenen filtrenin impuls cevabı bulunmuş olacaktır. Bu durumda istenen sayısal filtrenin transfer fonksiyonu;

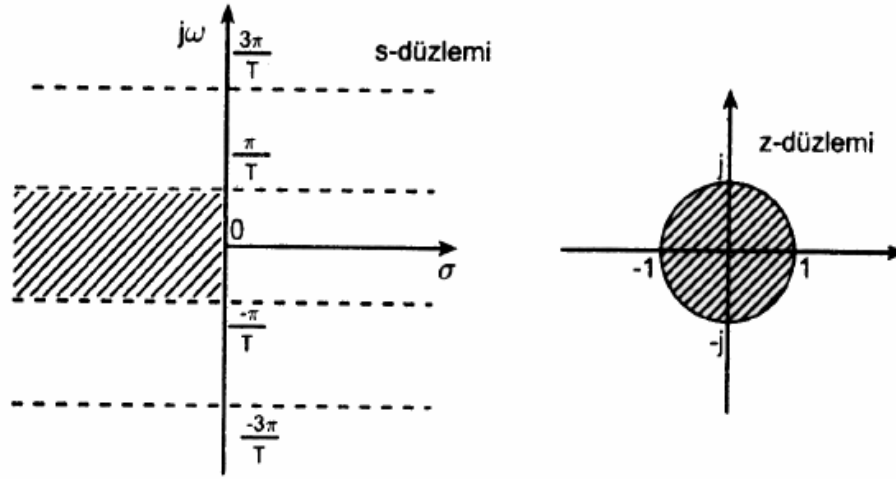
$$H_D(z) = Z[h_a(nT)] \quad (5.5)$$

olarak bulunur. Elde edilen bu sayısal filtrenin impuls cevabı ile analog filtrenin impuls cevapları birbirine eşit olduğundan, yönteme değişmez impuls cevabı yöntemi adı verilmiştir. Şekil 5.1’ de, sürekli zamanlı bir sistemden ayrık zamanlı sisteme nasıl geçildiği gösterilmektedir [12].

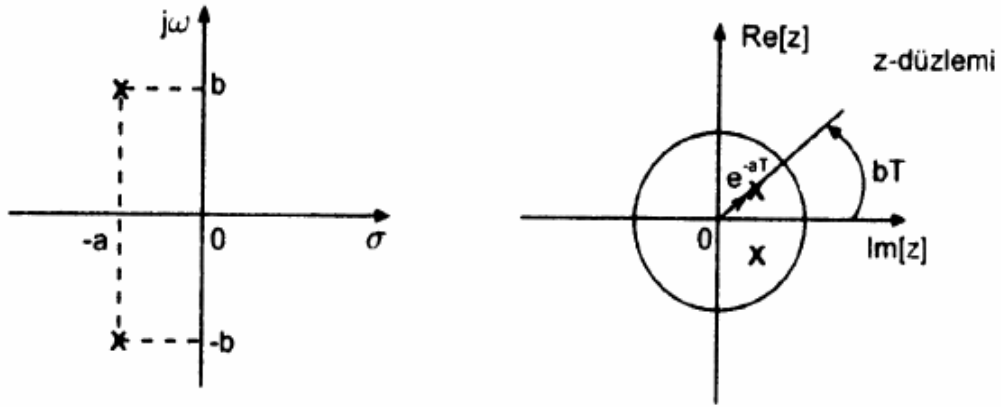


Şekil 5. 1 Sürekli-zamanlı bir filtrenin ayrık-zamanlı eşdeğeri

Şekilden de görüldüğü gibi, sürekli zamanlı bir sistemin girişine uygulanan $x(t)$ işareti, öncelikle T_0 örnekleme aralıklarıyla sürekli-zamanlı işaretin ayırık-zamanlı eşdeğeri elde edilir [17]. Elde edilen bu ayırık-zamanlı işaretin transfer fonksiyonu bulunur ve çıkış işareti olan $y[n]$ elde edilir. Bu sisteme ilk başlangıçtaki S/A (sürekli zamandan ayırık-zamana dönüşüm) dönüşümün tersi olan A/S (Ayırık-zamandan sürekli zamana dönüşüm) dönüşümü uygulanarak işaretin sürekli-zamandaki çıkış işareti karşılığı olan $y(t)$ elde edilir.



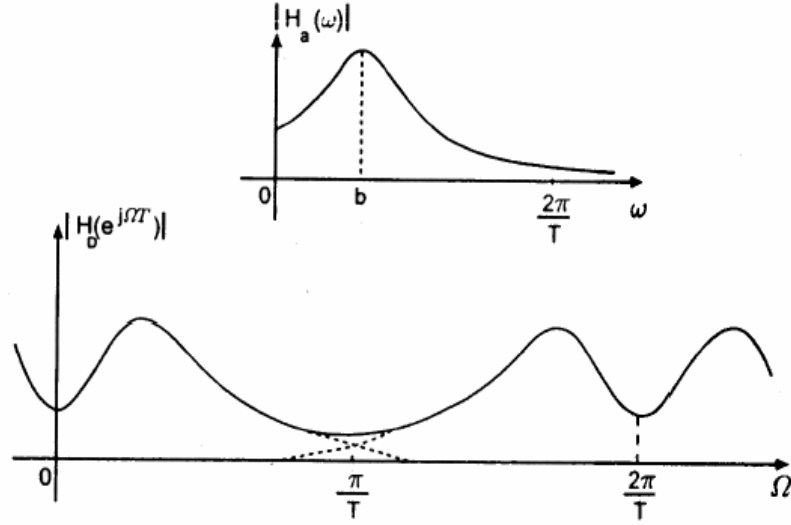
Şekil 5. 2 Periyodik örneklemenin gösterilmesi



Şekil 5. 3 Sistemdeki kutupların s-düzleminde z-düzlemine dönüşümü

Değişmez-impuls cevabı yönteminde, s-düzlemindeki s_k kutupları z-düzlemindeki $z_k = e^{s_k T}$ kutuplarına karşılık gelmektedir. Eğer sistemdeki s_k kutupları sol yarı s-düzlemindeyseler

$z_k = e^{s_k T}$ da birim dairenin içerisinde olacaktır. Ancak bu s-düzleminde, z-düzlemine olan geçiş bire-bir (one to one) değil de çoktan-bire (many to one) olan bir dönüşümdür [10]. Yani, s-düzlemindeki birçok nokta z-düzlemindeki aynı noktaya karşılık gelebilmektedir. Bunun neticesinde ise örtüşmeler meydana gelmektedir. Şekil 5.4' de bu durum gösterilmiştir.



Şekil 5.4 $H_a(\omega)$ ve $H_D(e^{j\Omega T})$ frekans spektrumu ve meydana gelen örtüşme

Oluşan örtüşmelerden dolayı da uygulanan dönüşüm bazı filtre türleri için iyi sonuç verirken (alçak geçiren filtre), bazı filtre türlerin yaklaşımda tamamen başarısız kalmaktadır (yüksek geçiren ve band-durduran filtreler).

5.2. Değiştirilmiş Değişmez-İmpuls Cevabı Yöntemi

Şekil 5.4' den de görüldüğü gibi meydana gelen örtüşmeden dolayı değişmez-impuls cevabı yöntemi sadece kutupları olan (all pole) filtreler için uygulanmaktadır. Ancak bu yöntemin değiştirilmiş çeşidi olan yöntem, sınırlı sayıda sıfırları olan filtreler için de uygulanma imkanı vermektedir. (5.6)'daki transfer fonksiyonu ele alınırsa,

$$H_a(s) = H \frac{N(s)}{D(s)} = H \frac{\prod_{k=1}^M (s - n_k)}{\prod_{k=1}^M (s - d_k)} \quad (5.6)$$

Burada H sabit ve $M < N$ olursa $H_a(s)$, kutupları olan iki filtre transfer fonksiyonun oranı şeklinde düşünülür. Bu durumda;

$$H_{a1}(s) = \frac{1}{D(s)} = \frac{1}{\prod_{k=1}^N (s - d_k)} \quad (5.7)$$

$$H_{a2}(s) = \frac{1}{N(s)} = \frac{1}{\prod_{k=1}^M (s - n_i)} \quad (5.8)$$

şeklinde tanımlandığında analog filtrenin transfer fonksiyonu;

$$H_a(s) = H \frac{H_{a1}(s)}{H_{a2}(s)} \quad (5.9)$$

biçiminde yazılır. $H_{a1}(s)$ ve $H_{a2}(s)$ analog fonksiyonlarına karşılık gelen z -domeni transfer fonksiyonları ise;

$$H_{D1}(z) = \sum_{k=1}^N \frac{A_k z}{z - e^{d_k T}} = \frac{N_1(s)}{D_1(s)} \quad (5.10)$$

$$H_{D2}(z) = \sum_{i=1}^M \frac{A_k z}{z - e^{n_i T}} = \frac{N_2(s)}{D_2(s)} \quad (5.11)$$

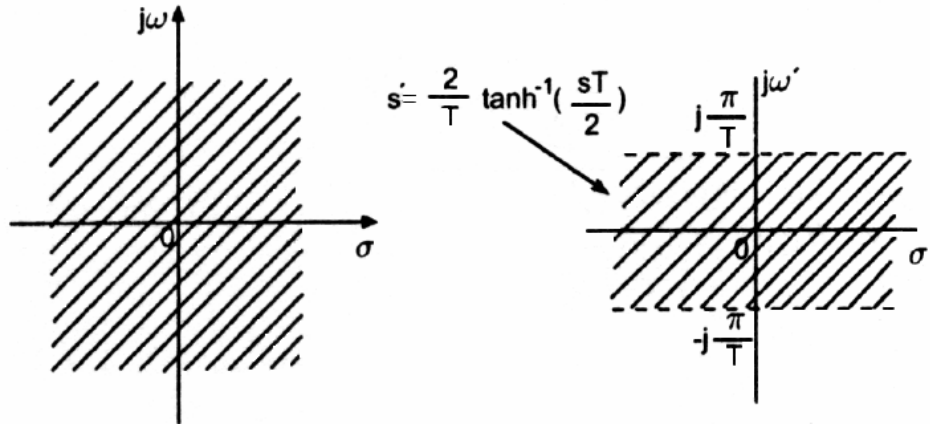
şeklinde olur. Bu durumda (5.10) ve (5.11) denklemlerinden,

$$H_D(z) = H \frac{H_{D1}(z)}{H_{D2}(z)} = H \frac{N_1(z)D_2(z)}{N_2(z)D_1(z)} \quad (5.12)$$

olarak yazılır. (5.12)' de pay ve payda kısımlarına ayrı ayrı değişmez-impuls cevabı yöntemi uygulanarak genel sonuç elde edilir. Bu yöntemin dezavantajı, transfer fonksiyonunun derecesinin artmasıdır.

5.3. Bilineer Dönüşüm Yöntemi

IIR filtre tasarım yöntemlerinden birisi olan değişmez-impuls cevabı yönteminde, sürekli zamanı temsil eden s -düzleminde ayrık zamanı temsil eden z -düzlemine geçiş, birden bire değil de çoktan bire dönüşümdür [12]. Bu durumda da şekil 5.4' de örtüşme meydana gelecektir. Bu sorunun ortadan kalkması için s -düzleminde, z -düzlemine geçişin birebir olması gerekmektedir. Bunu sağlamak içinde $-\infty < \omega < \infty$ aralığının $-\pi < \Omega < \pi$ aralığına karşılık getirilmesi gerekir. Bunun için, s -düzleminin öncelikle yeni bir ara düzlem olan s' -düzlemine dönüştürülmesi gerekir. Bu işlemden sonra s' den z -düzlemine dönüşüm sağlanır [18].



Şekil 5.5 s -düzleminde s' -düzlemine olan dönüşüm

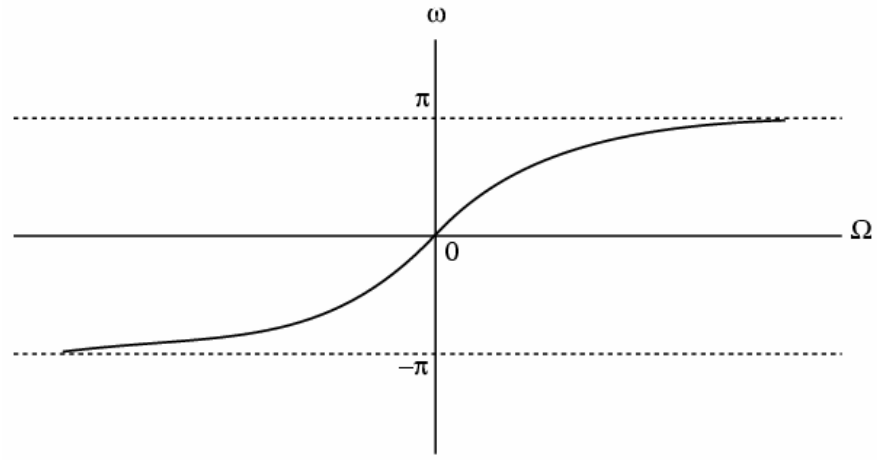
s -düzleminde s' -düzlemine geçiş için,

$$s' = \frac{2}{T} \tanh^{-1} \left[\frac{sT}{2} \right] \quad (5.13)$$

kullanılır. Bu dönüşümün $j\omega$ eksenindeki etkisi denklem (5.13)' de $s=j\omega$ ve $s'=j\omega'$ konularak görülebilir. Bu durumda,

$$\omega' = \frac{2}{T} \tan^{-1} \left(\frac{\omega T}{2} \right) \quad (5.14)$$

elde edilir. Şekil 5.6' da $j\omega$ ekseninin $(-\pi/T) \leq \omega' \leq (\pi/T)$ aralığına sıkışması gösterilmiştir. Bu durumda da dönüşüm birebirdir [10].



Şekil 5. 6 Bilineer dönüşüm ile ω dan ω' ' ne dönüşüm ($T=1$ için)

Bulunmaya çalışılan $s \rightarrow z$ dönüşüm denklemi (5.13)' ün tersinden

$$s = \frac{2}{T} \tanh\left(\frac{s'T}{2}\right) \quad (5.15)$$

ve $z = e^{s'T}$, nin tersinden

$$s' = \frac{1}{T} \ln z \quad (5.16)$$

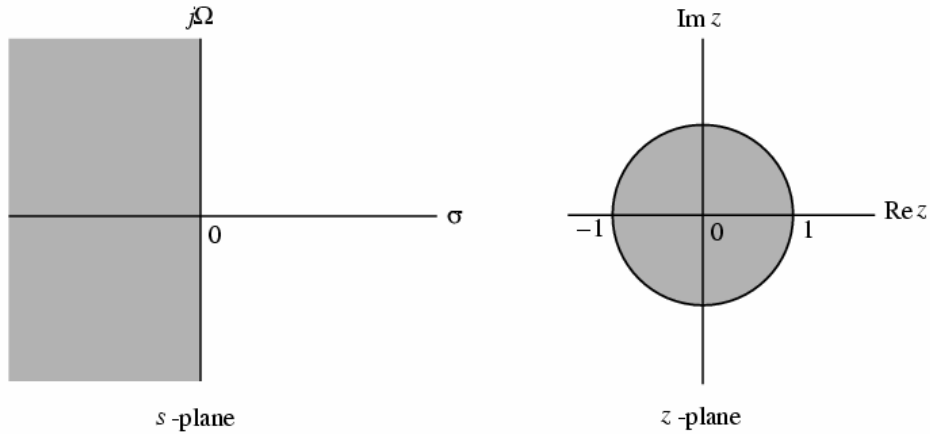
elde edilir. En genel formül ile de bilinear dönüşüm

$$s = \left(\frac{2}{T}\right) \frac{1 - z^{-1}}{1 + z^{-1}} \quad (5.17)$$

ile bulunur.

$$H(z) = H_a(s) \Big|_{s=\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}} \quad (5.18)$$

Denklem (5.18), analog filtre transfer fonksiyonundan sayısal filtre transfer fonksiyonunun tasarlanabileceğini gösterir [18].



Şekil 5. 7 Bilineer dönüşüm uygulanarak s-düzleminde z-düzlemine geçiş

Bir sayısal IIR filtresi üç aşamadan meydana gelmektedir. Bunlar,

1. Sayısal filtrenin özelliklerini temsil eden $H(z)$ ' ye ters bilineer dönüşüm uygulanarak $H_a(s)$ ' in özellikleri geliştirilir.
2. $H_a(s)$ ' nin tasarımı gerçekleştirilir.
3. $H_a(s)$ ' e bilineer dönüşüm uygulanarak $H(z)$ elde edilir.

5.3.1. Bilineer Dönüşümün Özellikleri

Bu dönüşümün sahip olduğu özellikler [10],

1. Sağ yarı s-düzlem bölgesi, z-düzleminde $|z|=1$ birim daire dışarısındaki noktalara karşılık gelmektedir.
2. s-düzleminde $j\omega$ eksenindeki noktalar, z-düzleminde $|z|=1$ birim dairesi üzerine karşılık gelmektedir.
3. Sol yarı s-düzlem bölgesi, z-düzleminde $|z|=1$ birim dairesi içindeki noktalara karşılık gelmektedir.

Yukarıdaki özelliklere bakılarak şu sonuçlar çıkarılabilir,

- İkinci özellikten dolayı, analog filtrenin geçirme ve durdurma bandı özellikleri aynen korunacaktır.
- Üçüncü özellikten dolayı, kararlı bir analog filtrenin transfer fonksiyonu yine kararlı bir sayısal filtre transfer fonksiyonu verecektir.

Dönüşümün sahip olduğu bu avantajların yanın da en önemli dezavantajı, impuls cevabının korunmamasıdır.

Bu dönüşüm yöntemi farklı tip analog filtreye uygulayıp, aynı tip sayısal filtre elde etmekte kullanıldığında iki tür yaklaşım kullanılarak gerçekleştirilir.

İlk yaklaşımda;

- Tasarlanmak istenen sayısal filtre $H_D(z)$ 'nin, aynı tipteki analog $H_D(s)$ 'nin frekans özelliklerine ulaşabilmesi için sayısal frekans değerleri belirlenir.
- $H_D(s)$ 'nin frekans özellikleri, prototip analog alçak geçiren filtre $H_{LP}(s)$ 'ye dönüştürülür.
- Analog alçak geçiren filtre tasarlanır.
- $H_{LP}(s)$ 'ye ters frekans dönüşümü uygulanarak, $H_D(s)$ elde edilir.
- $H_D(s)$ 'e bilineer dönüşüm uygulanarak, istenen özelliklerdeki $H_D(z)$ elde edilir.

İkinci yaklaşım;

- Tasarlanmak istenen sayısal filtre $H_D(z)$ 'nin, aynı tipteki analog $H_D(s)$ 'nin frekans özelliklerine ulaşabilmek için sayısal frekans değerleri belirlenir.
- $H_D(s)$ 'nin frekans özellikleri, prototip analog alçak geçiren filtre $H_{LP}(s)$ 'ye dönüştürülür.
- Analog alçak geçiren filtre tasarlanır.
- $H_{LP}(s)$ 'e bilineer dönüşüm uygulanarak, IIR sayısal filtre transfer fonksiyonu olan $H_{LP}(z)$ elde edilir.
- $H_{LP}(z)$, *spektral dönüşüm* yaklaşımı uygulanarak, istenen özelliklerdeki sayısal filtre transfer fonksiyonu $H_D(z)$ elde edilir.

Ancak IIR sayısal filtre tasarımında kullanılan bu bilineer dönüşüm yöntemi uygulandığında da, yüksek frekanslarda analog filtrenin frekans cevabında bazı bozulmalar meydana gelmektedir.

5.4. Sarma Etkisi ve Önsarma

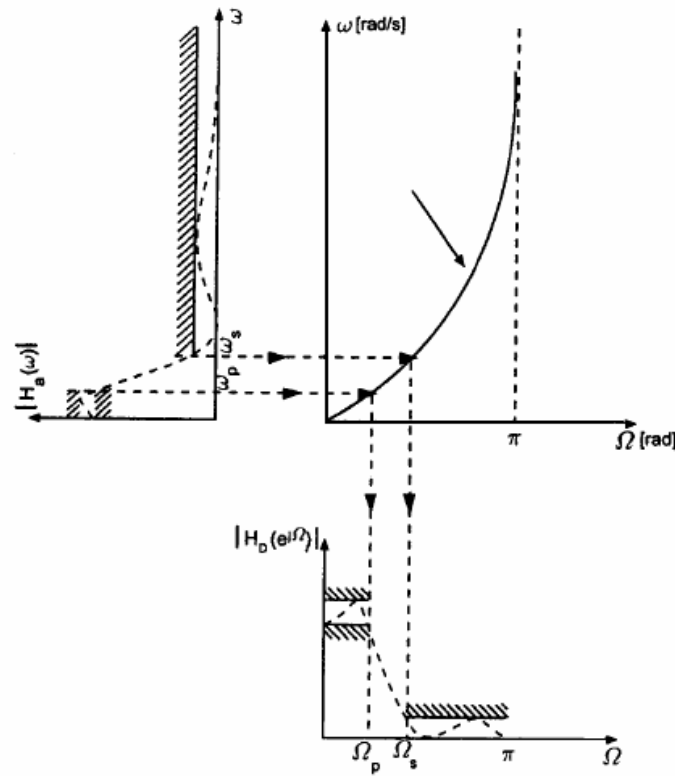
ω ve Ω sırasıyla analog ve sayısal filtrenin frekans değişkenlerini gösterirse, denklem (5.18)' de bilineer dönüşümün bu iki frekans arasında getirdiği ilişki [10],

$$\omega = \frac{2}{T} \tan\left(\frac{\Omega}{2}\right) \quad (5.19)$$

olmaktadır. $\Omega < 0.3\pi$ için,

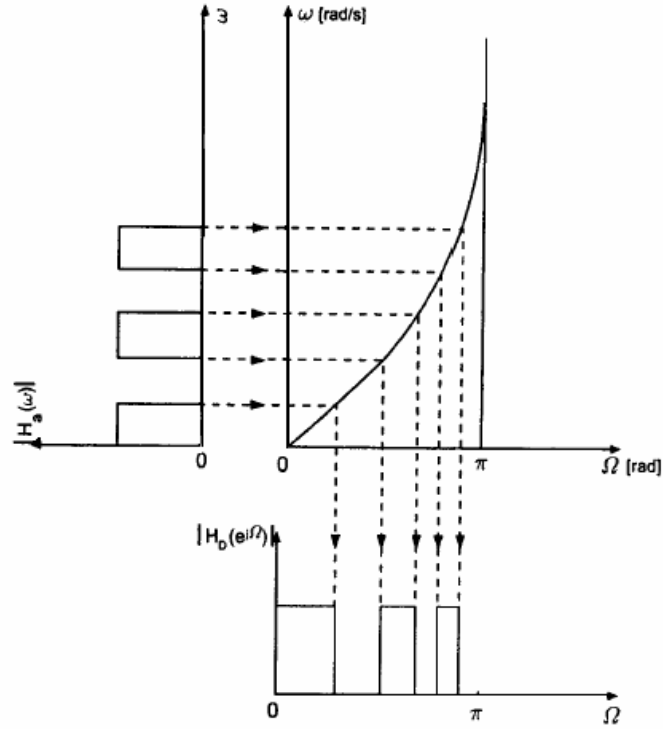
$$\omega \approx \Omega / T \quad (5.20)$$

yazılabilir. Bu sonuç bize, alçak frekanslarda ($\Omega < 0.3\pi$) sayısal filtrenin genlik cevabının, analog filtrenin genlik cevabı ile yaklaşık olarak aynı olduğunu gösterir. Ancak frekans değeri arttıkça, sayısal filtrenin genlik cevabı analog filtrenin genlik cevabından ayrılmakta yani değişim lineer olmaktan çıkmaktadır. Meydana gelen bu bozulmaya sarma (wrapping) etkisi adı verilmektedir. Bu durum Şekil 5.8’ de gösterilmiştir.



Şekil 5.8 $H(j\omega)$ 'dan $H(e^{j\Omega})$ 'ya bilineer dönüşümün gösterilmesi

Sarma etkisinin meydana getirdiği bozulmayı daha iyi anlayabilmek için, eşit geçirme bantı aralıklarına sahip analog filtrenin frekans cevabı göz önüne alınsın. Böyle bir durumda, bilineer dönüşüm uygulanarak elde edilen sayısal filtrenin frekans cevabı değişik olacaktır. Sayısal filtrenin yüksek frekanslarda, frekans cevabında geçirme bantlarının merkez frekansları ve bant genişlikleri azalacaktır. Bunun nedeni de değişimin lineer olarak değişmemesidir. Şekil 5.9’ da bu durum görülmektedir.



Şekil 5. 9 Genlik cevabı üzerindeki sarma etkisi

Eğer sadece filtrenin genlik cevabı ile ilgilenilecekse, meydana gelen sarma etkisi önsarma (prewarping) yöntemi uygulanarak giderilebilir. Öreğin, geçirme ve durdurma bant kenarları sırası ile Ω_p ve Ω_s olan bir sayısal filtre tasarlanmak isteniyorsa, bu değerlere aşağıdaki gibi önsarma uygulanabilir,

$$\omega_p = \frac{2}{T} \tan\left(\frac{\Omega_p}{2}\right) \quad (5.21)$$

$$\omega_s = \frac{2}{T} \tan\left(\frac{\Omega_s}{2}\right) \quad (5.22)$$

Kullanılan önsarma yönteminin amacı, bilineer dönüşüm sonucu istenilen Ω_p ve Ω_s sayısal kesim frekanslarını verecek olan analog filtrenin, uygun ω_p ve ω_s analog kesim frekanslarını elde etmektir. Bu amaçla,

$$\Omega'_p = 2 \tan^{-1} \left(\frac{\omega_p T}{2} \right) \quad (5.23)$$

$$\Omega'_s = 2 \tan^{-1} \left(\frac{\omega_s T}{2} \right) \quad (5.24)$$

şeklinde olur. Bu koşullar altında,

$$\Omega_p = \Omega'_p \quad (5.25)$$

$$\Omega_s = \Omega'_s \quad (5.26)$$

elde edilir. Böylece önsarma, sarma etkisini hesaba katarak, tasarlanan sayısal filtrenin istenen karakteristiklere uymasını sağlar. Bilineer dönüşümde yer alan $\frac{2}{T}$ katsayısı, tasarımda önemli değildir. Bu katsayı tasarımı basitleştirecek şekilde bir sabit sayı seçilebilir. Ancak seçilen bu sabit tasarım bitirilinceye kadar, aynı değerde tutulması gerekmektedir.

6. PROGRAMIN PROBLEME UYGULANMASI

Bu çalışmada gerçekleştirilen ve GAFKAT olarak isimlendirilen GA programındaki amaç, ilk başta rasgele üretilen katsayılardan en uygun olanların bulunarak istenilen özellikleri sağlayacak filtrenin genlik cevabını elde etmektir. Burada istenen genlik cevabı ile bulunan katsayıların oluşturduğu genlik cevapları arasındaki farkın minimum olması istenip, yeni durum bir minimizasyon problemidir. Problemde,

Amaç fonksiyonu= İstenin genlik cevabı değerleri – Bulunan genlik cevabı değerleri olarak tanımlanır. Dolayısıyla amaç fonksiyonu değeri ne kadar küçük olursa, istenilen çözüme o kadar yaklaşılmış olur. Program içerisinde kullanılan uygunluk fonksiyonu, amaç fonksiyonuna bağlı çalışmaktadır. Bu durumda,

Uygunluk fonksiyonu= | Olabilecek maksimum değer – Amaç fonksiyonu | olarak tanımlanabilir. Burada olabilecek maksimum değer; ele alınan karşılaştırma nokta sayısına göre, amaç fonksiyonunda kullanılmak istenen ve bulunan normalize genlik cevapları arasındaki en olumsuz durumdaki fark 1 olabileceğinden, karşılaştırma noktası sayısının belirlediği bir değerdir. Böylece aranan çözüme yaklaştıkça, uygunluk fonksiyonunun değerinin artması sağlanmış olur.

GAFKAT başlangıçta rasgele üretilen popülasyon içerisindeki katsayıların sahip oldukları genlik cevabı değerlerine göre uygunluk değerlerini hesaplamakta ve en iyi değer veren popülasyona doğru değer üretmekte, sonuçta da en uygun sonucu bulmaktadır.

Genetik algoritmaya dayalı olan ve MATLAB programının kullanılarak yazılan programın [19] çalışması aşağıdaki adımlardan oluşmaktadır.

1. Program ilk olarak kullanıcıdan;

Popülasyon büyüklüğünü,

Maksimum generasyon sayısını,

Filtre derecesini ,türünü, kritik frekans değerlerini ve örnekleme frekansını

istemektedir. İstenenlerin bu şekilde düzenlenmesi, MATLAB'ın hazır komutlarıyla hesaplanan filtrelerle karşılaştırmayı kolaylaştırmıştır.

2. Kullanıcının girmiş olduğu popülasyon büyüklüğüne ve filtre derecesine göre, tamamen rasgele sayılardan oluşan başlangıç popülasyonu oluşturulmaktadır. Popülasyon oluşturulurken, filtre katsayılarının sahip olduğu simetri göz önüne alınır. Sonuçta elde edilen popülasyon katlanarak, genlik cevabının oluşmasını sağlayacak olan filtre katsayılarına geri dönülmektedir.

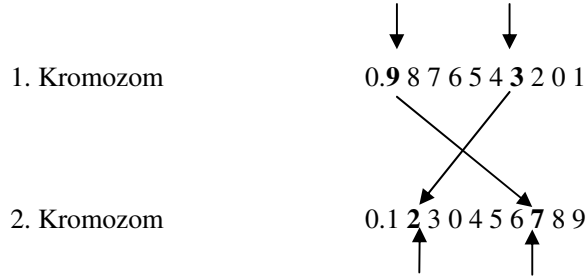
Program FIR sayısal filtre katsayılarını hesapladığından, elde edilen katsayılar sürekli 0 ile 1 arasında kalmaktadırlar. Dolayısıyla da bizim başlangıçta oluşturduğumuz popülasyondaki

katsayıların herhangi bir sayı ile çarpılmasına gerek yoktur. Ayrıca bu adımda, ileriki adımlarda (mutasyon işlemi için) kullanılmak üzere rasgele bir sayı üretilir.

3. Oluşturulan bu popülasyondaki her kromozom, önceden belirlenen uygunluk fonksiyonunda yerine yazılarak her biri için uygunluk, ölçeklenmiş uygunluk ve kopya sayıları belirlenir.

Elde edilen bu değerlere göre her bir kromozomun bir sonraki nesilde yer alıp alamayacakları belirlenir. Öyle ki, kopya sayısı 1 ve 1'den büyük olanlar, kendi benzerlerini kopya sayılarına göre diğer kromozomlar ile çaprazlayarak yeni nesle kopyalamakta, kopya sayısı 0 olan kromozomlar ise çaprazlama işlemine giremeyip yeni nesilde yer alamamaktadırlar.

4. Yazılan bu genetik algoritma programında hesaplanan kopya sayılarına göre kromozomların birbirleriyle çaprazlama işlemleri, bilinen yöntemlerdeki gibi sayıların ikili sayı sistemine dönüştürülmesiyle değil, doğrudan üretilen sayılar üzerinden değer çaprazlama yöntemi kullanılarak yapılmıştır. Bu çaprazlama işlemi yapılırken de katsayıyı temsil eden sayılardan her birinden iki sayı değeri, rasgele belirlenerek çaprazlamaya uğratılacak diğer kromozomdan rasgele seçilen iki sayı değeri ile yer değiştirilerek yapılmaktadır.



yapılan çaprazlama işlemi neticesinde ise,

1. Kromozom 0.7 8 7 6 5 4 2 2 0 1

2. Kromozom 0.1 3 3 0 4 5 6 9 8 9

şeklinde olmaktadır.

5. Kullanıcının girmiş olduğu generasyon sayısına ulaşılmamışsa, generasyon sayısı bir artırılır ve çaprazlama sonrasında elde edilen popülasyondaki her bir kromozom için yeniden uygunluk değerleri hesaplanır. Çaprazlama sonrasında elde edilen kromozomların genlik cevabı değeri normalize edilir ve normalize genlik cevabından bu cevabı sağlayan yeni katsayılarla geri dönülür.

6. Çaprazlama sonrasında elde edilen kromozomların uygunluk değerleri ile çaprazlamaya uğramamış kromozomlar yeni bir popülasyon oluştururlar. Yeni popülasyon ve eski popülasyonun birlikte oluşturduğu bu havuz içerisinde, uygunluk değerleri en iyi olanlar seçilerek yeni başlangıç popülasyonu oluşturulmaktadır.

7. Mutasyon oranına bağlı olarak bu adımda mutasyon işlemi yapılabilir.

Mutasyon, GAFKAT programının arama uzayında aramanın belirli bir noktaya sıkışmasını önlemek için kullanılan bir yöntemdir. Mutasyon yapılıp yapılmayacağına, rasgele bir sayı üretilip bu sayının önceden üretilen sayıya eşit olup olmadığına bakılarak karar verilir. Eğer üretilen sayılar birbirine eşitse mutasyon işlemi yapılır, değilse yapılmaz.

8. Kullanıcının başlangıçta girmiş olduğu generasyon sayısına ulaşıldığında, program en son elde edilen popülasyon içerisindeki kromozomlardan uygunluk değerlerine göre en iyi olandan başlayarak, popülasyondaki tüm kromozomların genlik cevaplarını hesaplayıp çizmektedir. Aksi halde program başa döner.

6.1. Programla Çalışılan Örnekler

Aşağıda verilen örnekler GAFKAT adı verilen programda çalıştırılarak sonuçlar elde edilmiş ve MATLAB’ da FIR komutu kullanılarak bulunan katsayılardan oluşan filtre genlik cevapları ile karşılaştırılmıştır.

Verilen örneklerde elde edilen genlik cevabı şekillerindeki frekans eksenini sayısal frekanstır.

6.1.1. Örnek1

MATLAB programı FIR1 komutunu kullanarak filtre katsayılarını hesaplarken, derece sayısından bir fazla katsayı hesaplamaktadır. GAFKAT programında da girilen filtre derecesinden bir fazla katsayı hesaplanıp program çıktısı olarak verilmektedir.

Filtre derecesi; 4

Popülasyon sayısı; 10

Generasyon sayısı; 50

Örnekleme frekansı; 20000

Filtre türü olarak AG (Alçak Geçiren) filtre

Geçirme bandı kesim frekansı; 3000 Hz

Durdurma bandı kesim frekansı; 4000 Hz

Tablo 6. 1 Başlangıç generasyonu

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
1. Krom.	-0.4219	-0.5974	0.1805	366.8547	707.7554	1
2. Krom.	0.4018	-0.0722	-0.0165	304.8923	588.2141	1
3. Krom.	-0.0084	-0.4019	0.5179	205.8648	397.1651	1
4. Krom.	-0.1686	-0.6072	0.7971	202.5224	390.7168	1
5. Krom.	0.5036	-0.7661	0.6611	212.7850	410.5159	1
6. Krom.	-0.3376	-0.5879	0.9679	214.4638	413.7548	1
7. Krom.	-0.0998	-0.2352	-0.9977	456.5558	880.8114	1
8. Krom.	-0.7433	-0.0809	0.0527	336.2396	648.6911	1
9. Krom.	0.8234	-0.8383	-0.0066	293.3358	565.9187	1
10.Krom.	0.8694	0.7699	0.1410	413.4352	797.6209	1

Tablo 6.1’de başlangıç generasyonunu oluşturan kromozomların her biri için uygunluk, ölçeklenmiş uygunluk ve kopya sayısı hesaplanır. Elde edilen kopya sayısına (bu örnekte kopya sayısı tesadüfi olarak hep 1 çıkmıştır) göre ilgili kromozom çaprazlama işlemine girecektir.

Tablo 6. 2 Çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri

Krom.	Filtre Katsayıları			Uygunluk
	1.Katsayı	2.Katsayı	3.Katsayı	
1. Krom.	-0.4260	-0.5934	0.0805	385.7111
2. Krom.	0.9628	0.6121	0.9689	388.9457
3. Krom.	0.4218	-0.0322	-0.0185	308.4145
4. Krom.	0.6234	-0.8333	0.1934	268.5431
5. Krom.	-0.0484	-0.4019	0.5978	200.3884
6. Krom.	-0.0998	0.5648	0.2022	388.9233
7. Krom.	-0.1696	-0.6002	0.4971	235.8747
8. Krom.	0.3567	-0.0509	0.0527	320.2461
9. Krom.	0.5036	-0.7621	0.4610	230.9100
10.Krom.	0.8194	0.7629	0.5411	417.9931

Çaprazlama işlemi sonrasında elde edilen kromozomlar ve uygunluk değerleri Tablo 6.2’ de verilmiştir. Çaprazlama sırasında kromozom sayısı sürekli olarak sabit kalacaktır. Yani, her kromozomun sahip olduğu kopya sayılarının toplamı popülasyon sayısına eşit olacaktır.

Başlangıçta elde edilen popülasyon ile çaprazlama sonrasında elde edilen toplam popülasyonlar içerisinde, uygunluk değerlerine bakılarak en iyi olan kromozomlar bir sonraki generasyonun başlangıç popülasyonunu oluşturmaktadırlar. Yeni generasyonun başlangıç popülasyonunu oluşturan kromozomlar ve onlara ait uygunluk, ölçeklenmiş uygunluk ve kopya sayıları Tablo 6.3’ de verilmiştir.

Tablo 6. 3 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (2. generasyon)

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
					1.0e+003 *	
1. Krom.	0.0998	-0.2352	-0.9977	456.5558	2.2081	1
2. Krom.	0.8194	0.7629	0.5411	417.9931	2.0216	1
3. Krom.	0.8694	0.7699	0.1410	413.4352	1.9996	1
4. Krom.	0.9628	0.6121	0.9689	388.9457	1.8811	1
5. Krom.	-0.0998	0.5648	0.2022	388.9233	1.8810	1
6. Krom.	-0.4260	-0.5934	0.0805	385.7111	1.8655	1
7. Krom.	-0.4219	-0.5974	0.1805	366.8547	1.7743	1
8. Krom.	-0.7433	-0.0809	0.0527	336.2396	1.6262	1
9. Krom.	0.3567	-0.0509	0.0527	320.2461	1.5489	1
10.Krom.	0.4218	-0.0322	-0.0185	308.4145	1.4917	1

Programda mutasyon işlemine önceden anlatıldığı gibi, rasgele üretilen sayıya, yeniden rasgele üretilen sayının eşit olması durumunda başvurulacaktır. Bu iki sayının birbirine eşit olmamasından, normal olarak program çaprazlama işlemine devam ederek yeni nesilleri oluşturmaktadır.

Tablo 6.1 ve Tablo 6.2 ‘den elde edilen, ikinci generasyondaki kromozomlar için hesaplanan kopya sayılarına göre ilgili kromozomlar kendi aralarında çaprazlandıklarında oluşan yeni popülasyon ise Tablo 6.4’ de verilmiştir.

Tablo 6. 4 İkinci generasyonda çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri

Krom.	Filtre Katsayıları			Uygunluk
	1.Katsayı	2.Katsayı	3.Katsayı	
1. Krom.	-0.0948	-0.2312	-0.9957	456.1441
2. Krom.	-0.4220	-0.5964	0.0815	384.8890
3. Krom.	0.8144	0.2627	0.5911	380.8917
4. Krom.	0.8644	0.9699	0.1420	411.9174
5. Krom.	0.4628	0.4121	0.9789	386.9136
6. Krom.	0.0567	0.2491	0.1526	417.1697
7. Krom.	-0.0918	0.5448	0.2042	392.9722
8. Krom.	-0.7733	-0.0309	0.4527	316.2884
9. Krom.	-0.4219	0.5026	0.1895	325.9021
10.Krom.	0.4278	-0.0362	-0.0165	308.6665

Çaprazlama sonrasında elde edilen Tablo 6.4 ‘deki kromozomlar ile Tablo 6.3 ‘deki kromozomlar arasında uygunluk değerleri en iyi olan kromozomlar alınarak 3. generasyonda kullanılacak olan yeni nesil popülasyon elde edilir. Bu durum Tablo 6.5’ de görülmektedir.

Tablo 6. 5 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (3. generasyon)

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
1. Krom.	-0.0998	-0.2352	-0.9977	456.5558	4.3398	1
2. Krom.	-0.0948	-0.2312	-0.9957	456.1441	4.3359	1
3. Krom.	0.8194	0.7629	0.5411	417.9931	3.9732	1
4. Krom.	0.0567	0.2491	0.1526	417.1697	3.9654	1
5. Krom.	0.8694	0.7699	0.1410	413.4352	3.9299	1
6. Krom.	0.8644	0.9699	0.1420	411.9174	3.9155	1
7. Krom.	-0.0918	0.5448	0.2042	392.9722	3.7354	1
8. Krom.	0.9628	0.6121	0.9689	388.9457	3.6971	1
9. Krom.	-0.0998	0.5648	0.2022	388.9233	3.6969	1
10.Krom.	0.4628	0.4121	0.9789	386.9136	3.6778	1

Tüm tablolardan da görüldüğü gibi çaprazlama sonrasında oluşan kromozomların uygunluk değerleri, bir önceki generasyondan daha iyi bir değer almakta ve dolayısıyla da program optimum sonuca ulaşmaya devam etmektedir. Mutasyon işleminin program içerisinde oluşması 1/1000 ihtimaline sahip olduğundan, program boyunca mutasyon işlemi gerçekleşmemiştir.

Program başlangıçta kullanıcının girmiş olduğu generasyon sayısı kadar tekrarlanarak sonuçta optimum sonucu bulmaktadır. Bu durumda elde edilen son popülasyon, FIR filtre tasarımında katsayıların simetri göstermesinden dolayı katlanarak filtre derecesinden bir fazla katsayı elde edilmiştir. Bu durum Tablo 6.6' da gösterilmiştir.

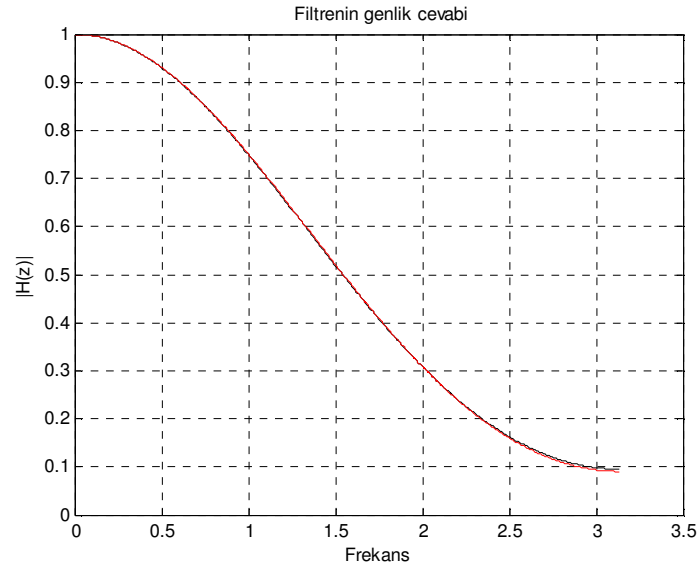
Tablo 6. 6 50. generasyon (Çözüm)

Krom.	Filtre Katsayıları					Çözüm
	1.Katsayı	2.Katsayı	3.Katsayı	4.Katsayı	5.Katsayı	
1. Krom.	0.0301	0.4292	0.9818	0.4292	0.0301	*****
2. Krom.	0.0301	0.4295	0.9808	0.4295	0.0301	
3. Krom.	0.0300	0.4345	0.9988	0.4345	0.0300	
4. Krom.	0.0300	0.4355	0.9918	0.4355	0.0300	
5. Krom.	0.0300	0.4343	0.9991	0.4343	0.0300	
6. Krom.	0.0300	0.4295	0.9888	0.4295	0.0300	
7. Krom.	0.0310	0.4343	0.9958	0.4343	0.0310	
8. Krom.	0.0301	0.4295	0.9898	0.4295	0.0301	
9. Krom.	0.0310	0.4349	0.9918	0.4349	0.0310	
10.Krom.	0.0310	0.4350	0.9918	0.4350	0.0310	

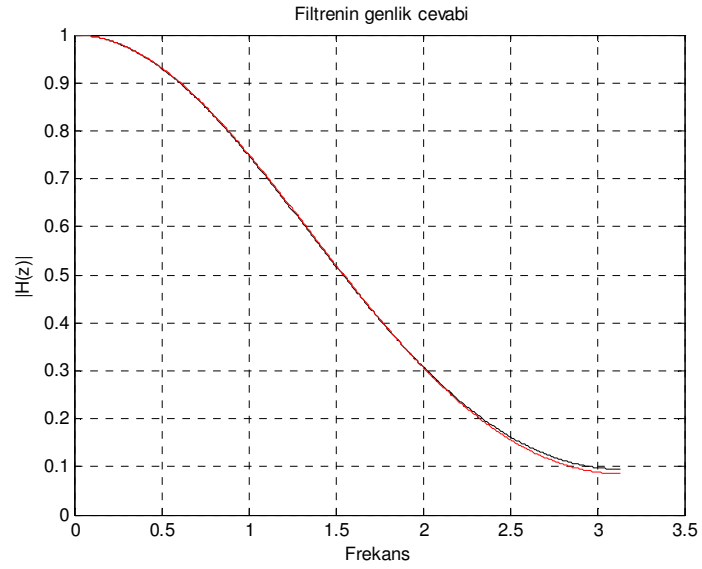
Bulunan son generasyon içerisinde uygunluk değeri, en iyi olan kromozoma ait katsayılar filtreden istenen genlik cevabını vermektedir. Ancak son generasyondaki bütün kromozomlar aslında çözüm olabilmektedir. Kullanıcının girmiş olduğu filtre özelliklerini sağlayacak olan filtrenin genlik cevabını oluşturan katsayılar ise şunlardır.

0.0152 0.2263 0.5171 0.2263 0.0152

Bu katsayılar ile programın bulduğu katsayılar birebir aynı değildir. Fakat Şekil 6.1 ve 6.2' de görülen genlik cevaplarına baktığımızda, bulunan değerlerin sağladığı genlik cevabı hemen hemen aynı çıkmaktadır.



Şekil 6. 1 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk)ve Çözüm kromozomundaki katsayıların oluşturduğu genlik cevabı (kırmızı renk)



Şekil 6. 2 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve son generasyondaki 10. kromozomun (uygunluk değeri en kötü) katsayılarının oluşturduğu genlik cevabı (kırmızı renk)

6.1.2. Örnek2

Filtre derecesi; 4

Popülasyon sayısı; 10

Generasyon sayısı; 50

Örnekleme frekansı; 20000

Filtre türü olarak YG (Yüksek Geçiren) filtre

Geçirme bandı kesim frekansı; 4000 Hz

Durdurma bandı kesim frekansı; 3000 Hz

İstenilen özellikleri sağlayacak filtrenin genlik cevabını oluşturan katsayıları bulmak için GAFKAT programı çalıştırıldığında ilk olarak rasgele üretilen başlangıç popülasyonu oluşturulur. Bu popülasyonda yer alan kromozomların her biri için uygunluk, ölçeklenmiş uygunluk ve kopya sayıları hesaplanır.

Tablo 6. 7 Başlangıç generasyonu

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
					1.0e+003 *	
1. Krom.	0.5222	-0.3977	-0.0867	378.1185	1.7782	1
2. Krom.	0.8235	0.4719	-0.5222	319.0563	1.5004	1
3. Krom.	0.9528	-0.2843	0.6393	340.3062	1.6004	1
4. Krom.	-0.7588	-0.7261	-0.3486	235.8862	1.1093	1
5. Krom.	0.4454	0.5801	0.4953	207.0530	0.9737	1
6. Krom.	0.6463	0.1794	-0.7582	339.7846	1.5979	1
7. Krom.	-0.4839	0.8738	-0.4462	343.2153	1.6140	1
8. Krom.	-0.6435	0.1955	0.9691	323.9453	1.5234	1
9. Krom.	-0.0265	-0.5786	-0.1091	302.2365	1.4213	1
10.Krom.	0.6503	-0.5327	0.8535	328.5318	1.5450	1

Tablo 6.7' den de görüldüğü gibi kromozomlar için hesaplanan kopya sayılarına göre, kromozomlar kendi aralarında çaprazlanarak yeni kromozomlar elde edilir. Burada oluşturulan popülasyonlar ilk başta rasgele üretildiğinden ve kopya sayılarına göre çaprazlamaya uğrayacak kromozomlar rasgele seçildiklerinden, çok farklı filtre katsayıları oluşmakta ve bu durumda sonuca ulaşmakta yardımcı olmaktadır.

Tablo 6. 8 Çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri

Krom.	Filtre Katsayıları			Uygunluk
	1.Katsayı	2.Katsayı	3.Katsayı	
1. Krom.	0.5222	-0.3957	-0.0367	375.3624
2. Krom.	-0.7888	0.1739	-0.3386	353.8907
3. Krom.	0.8035	0.3719	-0.5287	322.7179
4. Krom.	0.6433	0.1775	0.7418	260.5876
5. Krom.	0.2516	-0.2823	0.6593	351.7924
6. Krom.	0.7161	0.8738	-0.4473	345.0803
7. Krom.	0.4854	0.5610	0.4973	210.7181
8. Krom.	-0.6456	0.1055	0.9601	321.0263
9. Krom.	-0.0295	-0.5586	0.4909	411.2148
10.Krom.	0.0503	0.8673	0.8595	232.0343

Tablo 6.7 ve 6.8’ de yer alan kromozomların uygunluk değerleri içerisinde en iyi olanlar alınarak ikinci generasyon için popülasyon oluşturulur.

Tablo 6. 9 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (2. generasyon)

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
					1.0e+003 *	
1. Krom.	-0.0295	-0.5586	0.4909	411.2148	2.6923	1
2. Krom.	0.5222	-0.3977	-0.0867	378.1185	2.4756	1
3. Krom.	0.5222	-0.3957	-0.0367	375.3624	2.4576	1
4. Krom.	-0.7888	0.1739	-0.3386	353.8907	2.3170	1
5. Krom.	0.2516	-0.2823	0.6593	351.7924	2.3033	1
6. Krom.	0.7161	0.8738	-0.4473	345.0803	2.2593	1
7. Krom.	-0.4839	0.8738	-0.4462	343.2153	2.2471	1
8. Krom.	0.9528	-0.2843	0.6393	340.3062	2.2281	1
9. Krom.	0.6463	0.1794	-0.7582	339.7846	2.2247	1
10.Krom.	0.6503	-0.5327	0.8535	328.5318	2.1510	1

Elde edilen yeni generasyon da sadece çaprazlama işlemi yapılmış, mutasyon işlemi için gerekli olan şart sağlanmadığından mutasyon işlemi yapılmamıştır. 2. generasyon için elde edilen popülasyonun kopya sayılarına göre çaprazlama yapıldığında,

Tablo 6. 10 İkinci generasyonda çaprazlama sonrası oluşan kromozomlar ve uygunluk değerleri

Krom.	Filtre Katsayıları			Uygunluk
	1.Katsayı	2.Katsayı	3.Katsayı	
1. Krom.	-0.0225	0.6414	0.2909	281.8266
2. Krom.	0.9621	-0.2555	0.6393	339.3942
3. Krom.	0.5292	0.7024	-0.0267	280.2967
4. Krom.	0.1161	0.8728	0.4532	253.7659
5. Krom.	0.5422	-0.3902	0.5633	332.9753
6. Krom.	0.6903	-0.5378	0.8539	329.9594
7. Krom.	-0.7088	0.1779	-0.3356	351.4043
8. Krom.	0.7121	0.8234	0.6527	213.7439
9. Krom.	0.0515	-0.2222	0.5593	414.9962
10.Krom.	0.6663	0.1693	0.4418	279.0840

Tablo 6. 11 Çaprazlama sonrasında elde edilen değerler ile başlangıç generasyonundaki değerlerin karşılaştırılıp en uygun olanların seçilmesi (3. generasyon)

Krom.	Filtre Katsayıları			Uygunluk	Ölç. Uyg.	Kop.
	1.Katsayı	2.Katsayı	3.Katsayı			
					1.0e+003 *	
1. Krom.	-0.0295	-0.5586	0.4909	411.2148	2.6923	1
2. Krom.	0.5222	-0.3977	-0.0867	378.1185	2.4756	1
3. Krom.	0.5222	-0.3957	-0.0367	375.3624	2.4576	1
4. Krom.	-0.7888	0.1739	-0.3386	353.8907	2.3170	1
5. Krom.	0.2516	-0.2823	0.6593	351.7924	2.3033	1
6. Krom.	0.7161	0.8738	-0.4473	345.0803	2.2593	1
7. Krom.	-0.4839	0.8738	-0.4462	343.2153	2.2471	1
8. Krom.	0.9528	-0.2843	0.6393	340.3062	2.2281	1
9. Krom.	0.6463	0.1794	-0.7582	339.7846	2.2247	1
10.Krom.	0.6503	-0.5327	0.8535	328.5318	2.1510	1

Elde edilen bu popülasyon ile bir önceki değerler içinden uygunluk değerleri en iyi olanlar alınarak üçüncü generasyon için tablo elde edilir. Bu durum Tablo 6.11’ de verilmiştir. Bu adımlar başlangıçta girilen generasyon sayısı için tekrar edildiğinde son olarak elde edilen çözüm popülasyonu ise Tablo 6.12’ de gösterilmiştir.,

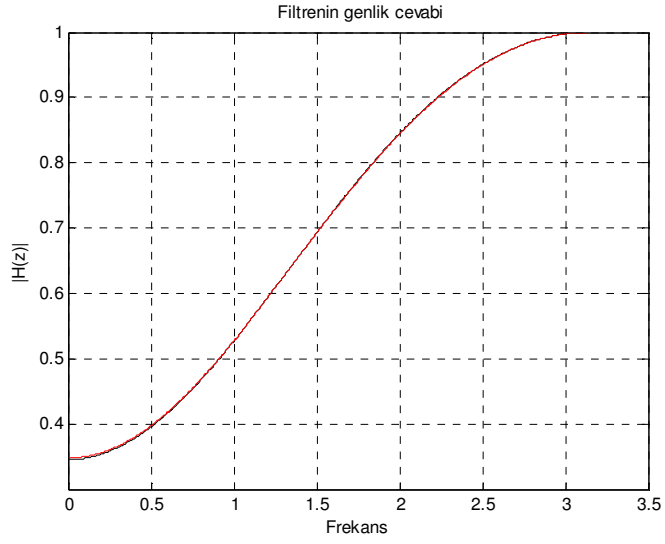
Tablo 6. 12 50. generasyon (Çözüm)

Krom.	Filtre Katsayıları					Çözüm
	1.Katsayı	2.Katsayı	3.Katsayı	4.Katsayı	5.Katsayı	
1. Krom.	-0.0125	-0.1796	0.7599	-0.1796	-0.0125	*****
2. Krom.	-0.0125	-0.1797	0.7560	-0.1797	-0.0125	
3. Krom.	-0.0135	-0.1777	0.7499	-0.1777	-0.0135	
4. Krom.	-0.0139	-0.1776	0.7699	-0.1776	-0.0139	
5. Krom.	-0.0105	-0.1877	0.7799	-0.1877	-0.0105	
6. Krom.	-0.0155	-0.1947	0.7928	-0.1947	-0.0155	
7. Krom.	-0.0129	-0.1773	0.7758	-0.1773	-0.0129	
8. Krom.	-0.0155	-0.1970	0.7958	-0.1970	-0.0155	
9. Krom.	-0.0129	-0.1763	0.7758	-0.1763	-0.0129	
10.Krom.	-0.0135	-0.1776	0.7799	-0.1776	-0.0135	

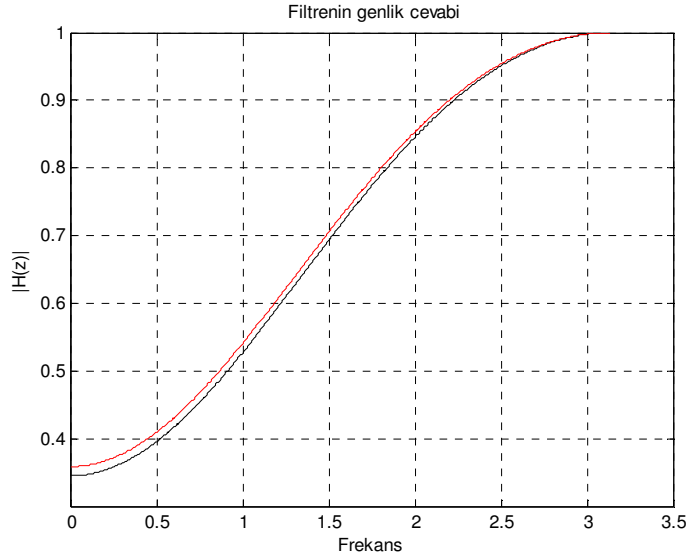
İstenilen özellikleri sağlayacak filtrenin impuls cevabı katsayıları;

-0.0110 -0.1637 0.6947 -0.1637 -0.0110

dur. Şekil 6.3 ve Şekil 6.4 ‘de istenilen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve GAFKAT programının bulduğu katsayılarla oluşan genlik cevabı (kırmızı renk) verilmiştir.



Şekil 6. 3 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve Çözüm kromozomundaki katsayıların oluşturduğu genlik cevabı (kırmızı renk)



Şekil 6. 4 İstenen özellikleri sağlayacak filtre genlik cevabı (siyah renk) ve son generasyondaki 10. kromozomun (uygunluk değeri en kötü) katsayılarının oluşturduğu genlik cevabı (kırmızı renk)

6.1.3. Örnek3

Filtre derecesi; 9

Popülasyon sayısı; 20

Generasyon sayısı; 50

Örnekleme frekansı; 20000

Filtre türü olarak AG (Alçak Geçiren)filtre
Geçirme bandı kesim frekansı; 3000 Hz
Durdurma bandı kesim frekansı; 4000 Hz

GAFKAT programı, istenilen özellikleri sağlayacak katsayıların bulunması için kullanıcının girmiş olduğu değerlere göre çalıştırıldığında, son (çözüm) generasyonda bulunan popülasyon Tablo 6.13’ de verilmiştir.

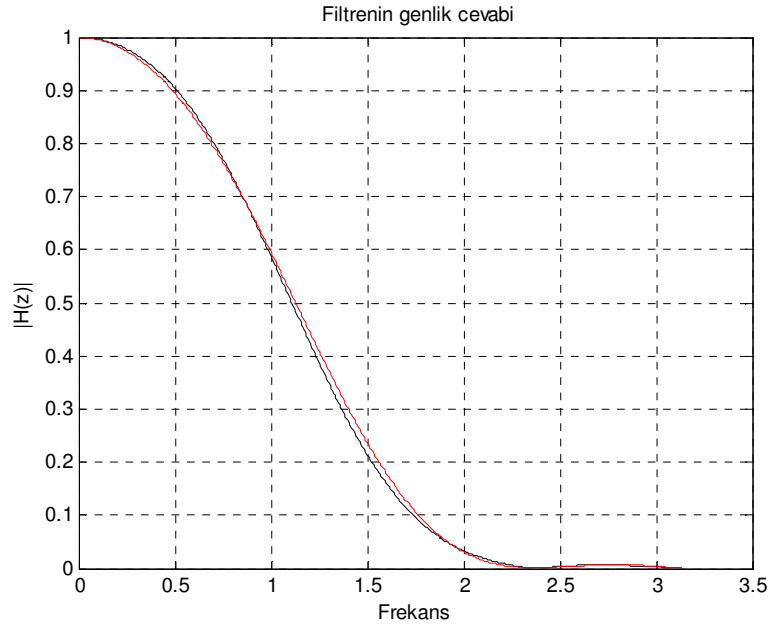
Tablo 6. 13 Son generasyon

Krm	Filtre Katsayıları										Çöz.
	1.Kat	2.Kat	3.Kat	4.Kat	5.Kat	6.Kat	7.Kat	8.Kat	9.Kat	10.Kat	
1.K.	0.0000	-0.0288	0.0467	0.3777	0.7757	0.7757	0.3777	0.0467	-0.0288	0.0000	****
2.K.	0.0001	-0.0228	0.0576	0.4299	0.8701	0.8701	0.4299	0.0576	-0.0228	0.0001	
3.K.	0.0010	-0.0389	0.0491	0.3447	0.7058	0.7058	0.3447	0.0491	-0.0389	0.0010	
4.K.	0.0010	-0.0218	0.0397	0.3877	0.7757	0.7757	0.3877	0.0397	-0.0218	0.0010	
5.K.	0.0005	-0.0328	0.0406	0.3347	0.7008	0.7008	0.3347	0.0406	-0.0328	0.0005	
6.K.	0.0004	-0.0098	0.0452	0.3507	0.7007	0.7007	0.3507	0.0452	-0.0098	0.0004	
7.K.	0.0001	-0.0208	0.0376	0.4249	0.8001	0.8001	0.4249	0.0376	-0.0208	0.0001	
8.K.	0.0010	-0.0118	0.0367	0.3377	0.7057	0.7057	0.3377	0.0367	-0.0118	0.0010	
9.K.	0.0011	-0.0088	0.0417	0.3477	0.7058	0.7058	0.3477	0.0417	-0.0088	0.0011	
10.K.	0.0011	-0.0188	0.0617	0.3447	0.7088	0.7088	0.3447	0.0617	-0.0188	0.0011	
11.K.	0.0000	-0.0288	0.0697	0.3497	0.7088	0.7088	0.3497	0.0697	-0.0288	0.0000	
12.K.	0.0046	-0.0299	0.0606	0.3645	0.7778	0.7778	0.3645	0.0606	-0.0299	0.0046	
13.K.	0.0046	-0.0098	0.0406	0.3608	0.7078	0.7078	0.3608	0.0406	-0.0098	0.0046	
14.K.	0.0110	-0.0200	0.0637	0.3857	0.7907	0.7907	0.3857	0.0637	-0.0200	0.0110	
15.K.	0.0000	-0.0288	0.0497	0.3877	0.7057	0.7057	0.3877	0.0497	-0.0288	0.0000	
16.K.	0.0011	-0.0082	0.0317	0.3547	0.7008	0.7008	0.3547	0.0317	-0.0082	0.0011	
17.K.	0.0010	-0.0200	0.0697	0.3757	0.7900	0.7900	0.3757	0.0697	-0.0200	0.0010	
18.K.	0.0076	-0.0299	0.0607	0.3445	0.7377	0.7377	0.3445	0.0607	-0.0299	0.0076	
19.K.	0.0045	-0.0048	0.0417	0.3377	0.7048	0.7048	0.3377	0.0417	-0.0048	0.0045	
20.K.	0.0006	-0.0209	0.0666	0.3675	0.7878	0.7878	0.3675	0.0666	-0.0209	0.0006	

İstenilen özellikleri sağlayacak filtre katsayıları ise;

-0.0056 -0.0113 0.0228 0.1655 0.3286 0.3286 0.1655 0.0228 -0.0113 -0.0056

şeklindedir. Program sonrası için genlik cevapları Şekil 6.5’ de görülmektedir.



Şekil 6. 5 20 popülasyon, 50 generasyon sonrasında alçak geçiren filtre için en iyi kromozom (kırmızı renk) ve istenen filtre katsayıları için (siyah renk) genlik cevabı

6.1.4. Örnek4

Filtre derecesi; 6

Popülasyon sayısı; 20

Generasyon sayısı; 50

Örnekleme frekansı; 20000

Filtre türü olarak YG (Yüksek Geçiren) filtre

Geçirme bandı kesim frekansı; 4000 Hz

Durdurma bandı kesim frekansı; 3000 Hz

Program, istenilen özellikleri sağlayacak katsayıların bulunması için kullanıcının girmiş olduğu değerlere göre çalıştırıldığında, son (çözüm) generasyonda bulunan popülasyon aşağıdaki tabloda verilmiştir. İstenilen özellikleri sağlayacak filtre katsayıları ise;

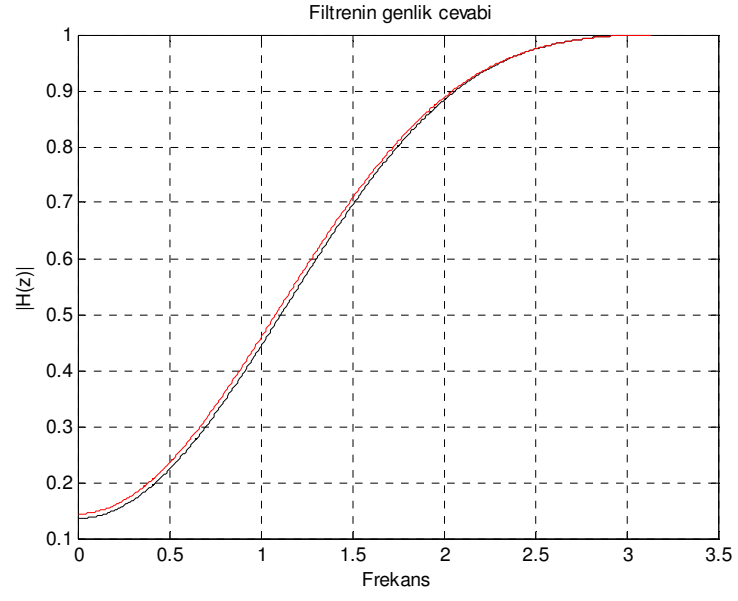
0.0013 -0.0397 -0.2175 0.6472 -0.2175 -0.0397 0.0013

şeklindedir.

Tablo 6. 14 Son generasyonla bulunan çözüm katsayıları

Krom.	Filtre Katsayıları							Çöz.
	1.Kat	2.Kat	3.Kat	4.Kat	5.Kat	6.Kat	7.Kat	
1. K.	-0.0000	-0.0095	-0.0490	0.1500	-0.0490	-0.0095	-0.0000	*****
2. K.	-0.0010	-0.0090	-0.0408	0.1170	-0.0408	-0.0090	-0.0010	
3. K.	-0.0010	-0.0090	-0.0409	0.1170	-0.0409	-0.0090	-0.0010	
4. K.	-0.0011	-0.0146	-0.0860	0.3001	-0.0860	-0.0146	-0.0011	
5. K.	-0.0000	-0.0090	-0.0440	0.1100	-0.0440	-0.0090	-0.0000	
6. K.	-0.0001	-0.0050	-0.0991	0.3130	-0.0991	-0.0050	-0.0001	
7. K.	-0.0000	-0.0050	-0.0890	0.3000	-0.0890	-0.0050	-0.0000	
8. K.	-0.0001	-0.0050	-0.0921	0.3130	-0.0921	-0.0050	-0.0001	
9. K.	-0.0000	-0.0095	-0.0801	0.3103	-0.0801	-0.0095	-0.0000	
10.K.	-0.0010	-0.0105	-0.0800	0.3101	-0.0800	-0.0105	-0.0010	
11. K.	-0.0080	-0.0095	-0.0490	0.1500	-0.0490	-0.0095	-0.0080	
12. K.	-0.0000	-0.0095	-0.0890	0.3500	-0.0890	-0.0095	-0.0000	
13. K.	-0.0000	-0.0100	-0.0880	0.3511	-0.0880	-0.0100	-0.0000	
14. K.	-0.0001	-0.0045	-0.0808	0.3107	-0.0808	-0.0045	-0.0001	
15. K.	-0.0000	-0.0090	-0.0880	0.3580	-0.0880	-0.0090	-0.0000	
16. K.	-0.0060	-0.0315	-0.0878	0.3011	-0.0878	-0.0315	-0.0060	
17. K.	-0.0080	-0.0311	-0.0850	0.3001	-0.0850	-0.0311	-0.0080	
18. K.	-0.0001	-0.0000	-0.0981	0.3030	-0.0981	-0.0000	-0.0001	
19. K.	-0.0001	-0.0005	-0.0888	0.3137	-0.0888	-0.0005	-0.0001	
20. K.	-0.0001	-0.0000	-0.0900	0.3080	-0.0900	-0.0000	-0.0001	

Program yardımıyla bulunan katsayılar ve olması gereken filtre katsayılar ile elde edilen genlik cevapları aşağıdaki şekilde verilmiştir.



Şekil 6. 6 20 popülasyon, 50 generasyon sonrasında yüksek geçiren filtre için en iyi kromozom (kırmızı renk) ve istenen filtre katsayıları için (siyah renk) genlik cevabı

7. SONUÇ VE ÖNERİLER

Bu çalışmada, sayısal filtre katsayılarını genetik algoritmaya bağlı olarak hesaplayan ve GAFKAT(Genetik Algoritma ile Filtre Katsayılarının Hesaplanması) adı verilen bir program MATLAB ortamında yazılmıştır. GAFKAT programının performansı örnekler ile incelenmiştir. Bu tezde yapılan işlemler maddeler halinde özetlenecek olursa;

- GAFKAT yardımıyla bulunan filtre katsayıları ile MATLAB kullanılarak hesaplanan filtre katsayıları için çizdirilen genlik cevapları karşılaştırıldı ve her ikisinin de yaklaşık aynı olduğu gözlemlendi. GAFKAT ile FIR filtrenin tasarımında, analitik tasarımında ki pencere fonksiyonu kullanımının getirdiği ilave hesaplama güçlüğü ortadan kaldırılmıştır.
- GAFKAT aynı problem için tekrar tekrar çalıştırıldığında, istenen genlik cevabını sağlayan çözümün tek olmadığı görüldü. Öyle ki olması gereken katsayılardan bazıları (-) iken, programın bulduğu katsayıların hepsinin (+) olması durumunda dahi genlik cevabının çok yakın olduğu, dolayısıyla da yazılan programın sonuca ulaşmada başarılı olduğu gözlemlendi.
- GAFKAT' da, GA' nın her bir aşaması (mutasyon, çaprazlama, uygunluk, ölçeklenmiş uygunluk ve kopya sayılarının hesabı) ayrı m-function'lar halinde yazılmış, böylece program sadece tek bir iş için değil, kullanıcının hesaplamak istediği farklı örnekler içinde kullanılabilecek modüler yapıya ulaşmıştır.
- Programda ayrıca tek bir filtre türü için değil, kullanıcının seçtiği dört farklı filtre türü içinde (Alçak geçiren, Yüksek geçiren, Bant-geçiren, Bant-durduran) çalışma imkanı sağlamıştır.

Bu tezde yapılanlar ileriki çalışmalara temel oluşturmakla beraber, kullanıcının isteğine bağlı olarak filtrenin genlik cevabındaki geçirme, geçiş ve durdurma bantlarında ki katsayı değerlerinin etkilerini arttırabilir veya azaltabilir.

Ayrıca program, sayısal filtre çeşitlerinden IIR filtre türü içinde hesaplama yapabilir hale getirilerek, kullanıcıya sorgu yöntemiyle ne tür filtre yapmak istediğini sorup filtre katsayılarını hesaplayabilir şeklinde paket program haline getirilebilir.

KAYNAKLAR

1. Nabiye V.V. 2005, *Yapay Zeka*, Seçkin Yayıncılık No: 975 347 985 9, Ankara, 764s
2. İnce M. C. 1996, *Mikroişlemci Algoritma ile Optimizasyon İçin Bir Fortran Bilgisayar Programı*, 4. Bilgisayar-Haberleşme Sempozyumu 107-110
3. Karaboğa D. 2004, *Yapay Zeka Optimizasyon Algoritmaları*, Atlas Yayın Dağıtım No:975 6574 37 2, İstanbul, 199s
4. Gürsu B. 2002, *Genetik Algoritmalar İle DC-AC Çeviricilerde Harmonik Eliminasyonu*, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, 99s.
5. Bakır M.A. 2003, *Tamsayılı Programlama Teori, Modeller ve Algoritmalar*, Nobel Yayın Dağıtım, No:975 591 572 9, Ankara, 630s
6. Raslan T., Horrocks D.H., 1995, *The Design of Analog and Digital Filters using Genetic Algorithms*, IEE, 2/1-2/5
7. Lee A., Ahmed M., Jullien G.A., Miller W.C., Lashkar R.S., 1998, *Digital Filtre Design Using Genetic Algorithm*, IEEE, 34-38
8. Suckley D., 1991, *Genetic Algorithm in the Design of FIR Filtres*, IEE Proceedings, 234-238
9. Cemes R., Ait-boudaoud D., 1993, *Genetic Approach to Design of Multiplierless FIR Filters*, IEE, 2090-2091
10. Kayran A. H., Ekşioğlu E. M. 2004, *Bilgisayar Uygulamalarıyla Sayısal İşaret İşleme*, Birsan Yayınevi No: 975-511-362-2, İstanbul, 347s
11. Ertürk S., *Sayısal İşaret İşleme*, Birsan Yayınevi No: 975-511-309-6, İstanbul, 293s
12. Schlichthale D., 2000, *Digital Filters Basics and Design*, Springer, No: 3-540-66841-1, Germany, 361p
13. Kayran A. H., 1990, *Sayısal İşaret İşleme*, İstanbul Teknik Üniversitesi Yayınları, No: 1409, İstanbul, 356s
14. Monson H. Hayes., 1999, *Digital Signal Processing*, No: 0-07-027389-8
15. Winder S., *Analog ve Dijital Filtre Tasarımı*, Bileşim Yayınları, No:975-6410-72-8, İstanbul, 363s
16. Hsu P. H., 2001, *Sinyaller ve Sistemler*, Nobel Yayınları, No:975-591-250-9, Ankara, 470s
17. Oppenheim A. V., Willsky A. S., Young I. T., 1983, *Signals and Systems*, No: 0-13-809731-3, United States of America, 796p

18. Oppenheim A. V., Schaffer R. W., 1975, *Digital Signal Processing*, No: 0-13-214107-8, 585p
19. Uzunoğlu M., Kızıl A., Onar Ç. Ö. 2003, *Her Yönü İle Matlab*, No: 975-6392-07-X, İstanbul,

ÖZGEÇMİŞ

Adı Soyadı: Turgay KAYA

Doğum Tarihi: 01 Ağustos 1982

Öğrenim Durumu:

Derece	Bölüm/ Üniversite	Yıl
Lisans	Elektrik-Elektronik Mühendisliği Bölümü, Fırat Üniversitesi	1999-2003
Yüksek Lisans	Elektrik Elektronik Mühendisliği Bölümü, Devreler ve Sistemler Anabilim Dalı, Fırat Üniversitesi	2004-2006

Yüksek Lisans Tez Başlığı ve Tez Danışmanı (Bitirme aşamasında):

Genetik Algoritma ile Sayısal Filtre Tasarımı
Yrd. Doç. Dr. Melih Cevdet İNCE

Akademik Çalışma Alanları:

- Sayısal Filtre
- Analog Filtre
- Sayısal İşaret İşleme

Görevler:

Görev Unvanı	Görev Yeri	Yıl
Arş.Gör.	Fırat Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü	Mayıs 2004-...