

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİ  
İLE KESİR DERECELİ FİLTRE GERÇEKLEMESİ**

**ÖZLEM İMİK ŞİMŞEK**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**TEMMUZ 2018**

**T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİ  
İLE KESİR DERECELİ FİLTRE GERÇEKLEMESİ**

**ÖZLEM İMİK ŞİMŞEK**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ DONANIM ANABİLİM DALI**

**TEMMUZ 2018**

**Tezin Bařlıđı:** Parçacık Sürüsü Optimizasyon Yöntemi ile Kesir Dereceli Filtre Gerçeklemesi

**Tezi Hazırlayan:** Özlem İMİK ŞİMŞEK

**Sınav Tarihi:** 23.07.2018

Yukarıda adı gecen tez, jürimizce değerlendirilerek bilgisayar mühendisliđi donanım anabilim dalında yüksek lisans tezi olarak kabul edilmiştir.

### **Sınav Jüri Üyeleri**

**Tez Danışmanı:** **Dr. Öğr. Üyesi B.Baykant ALAGÖZ** .....  
İnönü Üniversitesi

**Dr. Öğr. Üyesi Faruk SERİN** .....  
Munzur Üniversitesi

**Doç. Dr. Celaleddin YEROĞLU** .....  
İnönü Üniversitesi

**Prof. Dr. Halil İbrahim ADIGÜZEL**  
Enstitü Müdürü

## ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduđum “PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİ İLE KESİR DERECELİ FİLTRE GERÇEKLEMESİ” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla dođrularım.

Özlem İMİK ŞİMŞEK



## ÖZET

Yüksek Lisans Tezi

### PARÇACIK SÜRÜSÜ OPTİMİZASYON YÖNTEMİ İLE KESİR DERECELİ FİLTRE GERÇEKLEMESİ

Özlem İMİK ŞİMŞEK

İnönü Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Donanım

Anabilim Dalı

70 + ix sayfa

2018

Danışman: Dr. Öğr. Üyesi Barış Baykant ALAGÖZ

Kesir dereceli sistemler sayısal sistemlerde yaklaşık ayırık eşdeğer filtreler ile gerçekleştirilebilmektedir. Bu tez çalışmasında, kesirli derece türev operatörünün gerçekleştirilmesi için literatürde önerilmiş olan temel yaklaşık eşdeğer ayırık filtre tasarım yöntemleri araştırılmış ve kesir derece sistemlerin transfer fonksiyonlarının sayısal sistemlerde gerçekleştirilebilmesi için sonlu impuls cevap (IIR) filtre yapıları kullanılmıştır. Parçacık sürüsü optimizasyonu (PSO) yöntemi ile bu eşdeğer IIR filtre yapıları tasarımları optimize edilmiştir. Bu amaçla, genlik ve faz cevabı yakınsamalarını iyileştirmek için ağırlıklandırılmış çoklu amaç fonksiyonu olarak ifade edilen bir maliyet fonksiyonu tanımlanmıştır. Bu maliyet fonksiyonu ayırık filtre yaklaşımlarının genlik veya faz cevaplarının iyileştirilebilmesini sağlamıştır. PSO algoritmasının kararlı yaklaşık filtre çözümleri üretmesini sağlamak için kararsız çözümleri temsil eden parçacıklara en yüksek maliyetler atanmıştır. Bu modifikasyon ile dinamik olarak amaç fonksiyonu biçimlendirilerek, parçacıkların kararlı filtre çözümü olan bölgelerde çözüm araması sağlanmıştır.

Bu yöntemlerin bilgisayar destekli hesaplanması ve kolay kullanımı için Matlab grafiksel kullanıcı arayüzü (GUI) tasarımları yapılmıştır. Bu GUI programları ile literatürde bulunan temel analitik yöntemler kullanılarak kesir dereceli filtrelerin yaklaşık eşdeğer ayırık filtre tasarımları yapılabilen, PSO optimizasyonu ile bu tasarımlar hedeflenen faz veya genlik cevabı performansına erişmesi için iyileştirilebilen, tasarlanan filtre çözümlerinin analizi için Bode grafikleri, birim basamak cevapları, kompleks düzlemde kutup yerleşimleri çizilmektedir. Bu GUI ile Matlab ortamında kesir dereceli sistemlerin yaklaşık eşdeğer ayırık filtre tasarımları kolaylaştırılmıştır.

**Anahtar Kelimeler :** Kesir dereceli sistemler, sayısal filtre tasarımı, PSO, genlik ve faz cevapları, Matlab

## ABSTRACT

MasterThesis

### IMPLEMENTATION OF FRACTIONAL ORDER FILTERS BY PARTICLE SWARM OPTIMIZATION METHOD

OZLEM IMIK SIMSEK

Inonu University  
Graduate School of Natural and Applied Science  
Department of Computer Engineering

70 + ix pages

2018

Supervisor: Dr. Baris Baykant ALAGOZ

Fractional order systems can be implemented in digital system by approximate equivalent discrete filters. In this thesis, fundamental approximate equivalent discrete filter design methods, which are proposed for the implementation of fractional order filters in the literature, are investigated and Infinite Impulse Response (IIR) filter structures are used for implementation of fractional order transfer function in digital systems. These IIR filter designs are optimized by using Particle Swarm Optimization (PSO) method. For this purpose, a cost function, which is expressed in the form of a weighed multi-objective function to improve magnitude and phase response approximations, is defined. This cost function can provide improvement of the magnitude and phase responses of discrete filter approximations. To allow PSO algorithm that yield stable approximate filter solutions, the particles that are representing instable filter solutions are assigned to maximum cost values. With this modification, the objective function is dynamically shaped and this allows searching of particles in the region of stable filter solutions.

For computer-aided computation and easy-use of these methods, Matlab Graphical User Interfaces (GUI) are designed. With this GUI programs, approximate equivalent discrete filter design of fractional order filters can be carried out by using fundamental analytical methods in the literature, this filter design can be improved to reach a targeted phase and magnitude response performance by means of PSO, for the analyses of designed filter solutions, Bode diagram, unit step responses, pole placements in complex plane are plotted. By means of this GUI, approximate equivalent discrete filter design of fractional order systems in Matlab is facilitated.

**Keywords:** Fractional order systems, digital filter design, PSO, amplitude and phase responses, Matlab

## TEŞEKKÜR

Tez konumun şekillenmesinde ve çalışmalarım boyunca gösterdiği her türlü ilgi, anlayış, destek ve katkıları için danışman hocam Sayın Dr. Öğr. Üyesi B. Baykant ALAGÖZ' e;

Her zaman destekçim olduğumu hissettiğim, çalışma disiplinini öğrendiğim Sayın Doç. Dr. Celaddin YEROĞLU' na;

Bölüm başkanımız Sayın Doç. Dr. Celaleddin YEROĞLU 'na ve tüm bölüm çalışanlarına;

Tüm hayatım boyunca olduğu gibi tez çalışmam sürecinde benden desteklerini esirgemeyen değerli Ailem' e ve özellikle Annem ve Babam' a,

Ayrıca her zaman yanımda olan ve tez yazım sürecinde desteğini esirgemeyen Eşim'e

TEŞEKKÜR EDERİM.

## İÇİNDEKİLER

ONUR SÖZÜ.....	i
ÖZET.....	ii
ABSTRACT.....	iii
TEŞEKKÜR.....	iv
İÇİNDEKİLER.....	v
SİMGELER VE KISALTMALAR.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	ix
1. <b>GİRİŞ</b> .....	1
2. <b>KURUMSAL TEMELLER</b> .....	4
2.1. Sistem Modellemede Kullanılan Temel Matematiksel Dönüşümler.....	4
2.1.1. Laplace Dönüşümü.....	4
2.1.2. Z Dönüşümü.....	5
2.1.3. Bazı Önemli Z –Dönüşüm Teoremleri.....	8
2.1.4. Sürekli Sistem Modellerinin Ayırıklaştırılması.....	9
2.2. Kesir Dereceli Türev.....	10
2.2.1. Kesir Dereceli Türev Operatörünün Temel Özellikleri .....	10
2.2.2. Kesir Dereceli Türev Tanımları.....	10
2.3. Kesir Dereceli Lineer Zamanla Değişmeyen (LZD) Sistem Modelleri ve Kararlılığı.....	12
2.4. Kesir Dereceli Sistemlerin Yaklaşık Eşdeğer Modellemesi ve Sayısal Gerçeklemesi.....	14
2.4.1. Kesir Dereceli Transfer Fonksiyonun Ayırıklaştırılması .....	14
2.4.2. Kesir Dereceli Türevin Ayırık Zaman Yaklaşık Eşdeğer Modellerin Elde Edilmesi.....	15
2.4.3. Dolaylı (Indirect) Ayırıklaştırma Yöntemleri.....	16
2.4.4. Doğrudan (Direct) Ayırıklaştırma Yöntemleri.....	25
2.5. Kesir Dereceli Transfer Fonksiyonun Ayırıklaştırılması İçin Geliştirilen Temel Yöntemlerin Matlab GUI İle Programlanması.....	34
2.5.1. Matlab GUI İle Analitik Yöntemler İle Yaklaşık Eşdeğer Filtre Tasarımı ve Tasarımların Karşılaştırılması ve Değerlendirilmesi.....	35
2.6. Matlab Sembolik Hesaplama İle Kesir Dereceli Transfer Fonksiyonun Ayırıklaştırılması.....	40
3. <b>MATERYAL VE YÖNTEM</b> .....	43
3.1. Parçacık Sürüsü Optimizasyonu (PSO) Hakkında Temel Bilgiler .....	43
3.2. Parçacık Sürüsü Optimizasyonunun Kesir Dereceli Transfer Filtre Fonksiyonlarının Ayırık Gerçeklemesine Uygulaması .....	49
3.3. Kararlı IIR Filtre Çözümlerini Elde Etmek İçin PSO 'da Yapılan Adaptasyonlar.....	51
4. <b>ARAŞTIRMA BULGULARI</b> .....	53
4.1. Kesir Dereceli Transfer Filtre Fonksiyonlarının Ayırıklaştırılması için Geliştirilen Toolbox.....	53
4.2. Analitik Yöntem Sonucu ile PSO Sonucunun Kıyaslaması .....	57
5. <b>TARTIŞMA VE SONUÇ</b> .....	61
5.1. Gelecek Çalışma Önerileri.....	63
6. <b>KAYNAKÇA</b> .....	64
EKLER.....	66
ÖZGEÇMİŞ.....	70

## SİMGELER VE KISALTMALAR

PSO	Parçacık Sürüsü Optimizasyonu
IIR	Sonsuz İmpuls Cevap Filtresi (Infinite Impulse Response)
LZD	Lineer Zamanla Değişmeyen
PID	Oransal-İntegral-Türev (Proportional-Integral-Derivative) Kontrolör
CFE	Sürekli Kesir Açılımı
GUI	Grafiksel Kullanıcı Ara yüzü
$s$	Sürekli zaman $s$ bölgesi kompleks değişken
$F(s)$	Sürekli zaman transfer fonksiyonu
$t$	Sürekli zaman bölgesi değişkeni
$f(t)$	Sürekli zaman bölgesi fonksiyonu
$L\{.\}$	Laplace dönüşümü
$L^{-1}\{.\}$	Ters Laplace dönüşümü
$s^\alpha$	Sürekli zaman $s$ bölgesinde $\alpha$ dereceli türev
$\Gamma(.)$	Gama fonksiyonu
$z$	Ayrık zaman $z$ bölgesi kompleks değişken
$Z\{.\}$	Z dönüşümü
${}_0D_t^\alpha$	$\alpha$ kesir dereceli türev operatörü
$\arg(.)$	Kompleks sayılar için açı operatörü (argüment)
$\omega$	Açısal frekans (rad/s)
$X_i[t]$	$i$ . parçacığın konum vektörü
$V_i[t]$	$i$ . parçacığın hız vektörü
$X_L[t]$	Lokal en iyi parçacık konumu
$X_G[t]$	Global en iyi parçacık konumu
$c_1$	Lokal öğrenme katsayısı
$c_2$	Global öğrenme katsayısı
$r_1$	Lokal öğrenme serbestliği için rastgele bir sayı
$r_2$	Global öğrenme serbestliği için rastgele bir sayı
$w$	Parçacığın ivme katsayısı
$\xi$	Yavaşlatma katsayısı

## ŞEKİLLER DİZİNİ

Şekil 2.1.	Örnekleme işlemi.....	6
Şekil 2.2.	LZD kesir dereceli sistemlerin kararlılık bölgesi [13].....	13
Şekil 2.3.	Oustaloup yöntemi seri filtre dizisi [8].....	17
Şekil 2.4.	Oustaloup yöntemi seri filtre grubunun oluşturduğu kutup ve sıfırların kesir derece türevin frekans cevabına yakınsaması [12].....	17
Şekil 2.5.	Oustaloup yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının genlik cevapları.....	19
Şekil 2.6.	Oustaloup yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının faz cevapları.....	19
Şekil 2.7.	Birim basamak cevabı.....	20
Şekil 2.8.	Sistem kutup ve sıfırlarının dağılımı.....	20
Şekil 2.9.	CFE yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının genlik cevapları.....	23
Şekil 2.10.	CFE yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının faz cevapları.....	23
Şekil 2.11.	Birim basamak cevabı.....	24
Şekil 2.12.	Sistem kutup ve sıfırlarının dağılımı.....	24
Şekil 2.13.	Özyinelemeli Tustin yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin genlik cevabı.....	27
Şekil 2.14.	Özyinelemeli Tustin yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin faz cevabı.....	28
Şekil 2.15.	Birim basamak cevabı.....	29
Şekil 2.16.	Sistem kutup ve sıfırlarının dağılımı.....	29
Şekil 2.17.	Özyinelemeli CFE yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin genlik cevabı.....	32
Şekil 2.18.	Özyinelemeli CFE yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin faz cevabı.....	32
Şekil 2.19.	Birim basamak cevabı.....	33
Şekil 2.20.	Sistem kutup ve sıfırlarının dağılımı.....	33
Şekil 2.21.	Analitik yöntemler ile kesir derece transfer fonksiyonu tasarımı için geliştirilen GUI modülü kullanıcı ara yüzü.....	34
Şekil 2.22.	Özyinelemeli Tustin yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin genlik cevapları ( $n=1,3,5,7,9$ ) .....	36
Şekil 2.23.	Özyinelemeli Tustin yöntemi uygulanarak elde edilmiş ayrık IIR filtrenin faz cevapları ( $n=1,3,5,7,9$ ).....	36
Şekil 2.24.	CFE ile dolaylı ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması ( $n=1,2,3,4$ ).....	37
Şekil 2.25.	CFE ile dolaylı ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin faz cevaplarının kıyaslaması ( $n=1,2,3,4$ ) .....	37
Şekil 2.26.	Özyinelemeli CFE doğrudan ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması ( $n = 1, 3, 5,7,9$ ) .....	38
Şekil 2.27.	Özyinelemeli CFE doğrudan ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin faz cevaplarının kıyaslaması ( $n=1,3,5,7,9$ ) .....	38
Şekil 2.28.	Oustaloup ile dolaylı ayrıklaştırmaya yöntemine ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması ( $n=1,3,5,7,9$ )....	39
Şekil 2.29.	Oustaloup ile dolaylı ayrıklaştırmaya yöntemine ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması ( $n=1,3,7,9$ ).....	39

Şekil 2.30.	Matlab sembolik programlama ile karmaşık kesir dereceli filtreleri gerçekleyen örnek kod.....	41
Şekil 3.1.	PSO algoritmasında global ve yerel en iyi parçacık konumuna göre parçacığın hız vektörüne etki eden terimler.....	45
Şekil 3.2.	PSO algoritmasının akış şeması.....	47
Şekil 3.3.	Parçacığın yerel en iyi ( $X_L[t]$ ) ve global en iyi ( $X_G[t]$ ) ile etkileşimi.....	48
Şekil 3.4.	Parçacığın yerinin yerel en iyi ( $X_L[t]$ ) ve global en iyi ( $X_G[t]$ ) göre güncellenmesi.....	48
Şekil 3.5.	Diğer parçacığın güncellenen yerel en iyi ( $X_L[t]$ ) ve global en iyi ( $X_G[t]$ ) ile etkileşimi.....	48
Şekil 3.6.	Kararsız filtre çözümleri için arama uzayında maliyet fonksiyonun modifiye edilmesi.....	52
Şekil 4.1.	PSO yöntemi ile optimizasyon sağlayan gelişmiş GUI programının kullanıcı ara yüzü.....	54
Şekil 4.2.	Maliyet fonksiyonun optimizasyon boyunca değişimi.....	55
Şekil 4.3.	PSO uygulanarak elde edilmiş ayrık IIR filtrenin genlik cevabı.....	56
Şekil 4.4.	PSO uygulanarak elde edilmiş ayrık IIR filtrenin faz cevabı.....	56
Şekil 4.5.	Birim basamak cevabı.....	57
Şekil 4.6.	Sistem kutup ve sıfırlarının dağılımı.....	57
Şekil 4.7.	Maliyet fonksiyonun optimizasyon boyunca değişimi.....	58
Şekil 4.8.	Özyinemeli Tustin ve PSO ile elde edilmiş ayrık IIR filtrenin genlik cevabı.....	59
Şekil 4.9.	Özyinemeli Tustin ve PSO ile elde edilmiş ayrık IIR filtrenin faz cevabı.....	60
Şekil 5.1.	Bu çalışmada uygulanan yaklaşım.....	62

## ÇİZELGELER DİZİNİ

Çizelge 2.1.	Bazı Ayrık Fonksiyonların $z$ dönüşümleri.....	7
Çizelge 2.2.	Bazı Temel Fonksiyonların $z$ dönüşümleri.....	7
Çizelge 2.3.	Bazı Temel Dönüşüm Fonksiyonları.....	9
Çizelge 2.4.	Genlik ve faz cevabı hatası.....	20
Çizelge 2.5.	Genlik ve faz cevabı hatası.....	23
Çizelge 2.6.	$s^\alpha$ seri açılım çözümlerine örnekler.....	25
Çizelge 2.7.	$A_n(z^{-1}, \alpha)$ fomülünün $n=1, \dots, 9$ [24].....	26
Çizelge 2.8.	Genlik ve faz cevabı hatası.....	28
Çizelge 2.9.	Farklı $r$ ifadeleri için $D_r(z)$ nin pay ve paydası [24].....	31
Çizelge 2.10.	Genlik ve faz cevabı hatası.....	33
Çizelge 2.11.	Genlik cevabı yakınsama hatalarının (Ortalama karesel hata) kıyaslamaları.....	40
Çizelge 2.12.	Faz cevabı yakınsama hatalarının (Ortalama karesel hata) kıyaslamaları.....	40
Çizelge 4.1.	Ortalama karesel genlik ve faz cevabı hatalarının kıyaslaması.....	60

## 1. GİRİŞ

Kesir dereceli türev üç yüzyılı aşan bir süredir bilinmesine rağmen kesirli matematik (fractional calculus) ve uygulamaları konusuna son yıllarda ilginin arttığı görülmüştür. Başta kontrol sistemleri alanında olmak üzere, nümerik analizlerde, sistem simülasyonlarında kesir dereceli sistem modelleri sıklıkla kullanılmaya başlanmıştır.

Kesir dereceli sistemlerin analizlerinde kesirli türev ve integral operatörlerinin çözümüne ihtiyaç duyulur. Kesir dereceli türev ve integral işleminin fiziksel yorumu tamsayı dereceli türev veya integral kadar kolay ifade edilemez. Örneğin, bir fonksiyonun birinci türevi fonksiyonun değişim hızını yada fonksiyonun eğimini ifade eder. Kesir dereceli türevlerin yorumu matematiksel tanımının uygulamadaki rolüne göre değişebilmektedir. Ancak, kesir dereceli türevin lokal bir operatör olmadığı, türev işleminin fonksiyonun geçmiş değerlerini de bağımlı olan bir operatör olduğu bilinmektedir. Bu durum uzun bellek (long memory) etkisi olarak adlandırılmıştır. Oysaki tamsayı dereceli türevler sadece hesaplama yapılan noktanın yakın civarındaki fonksiyon değerlerine ihtiyaç duyulur ve bu nedenle lokal bir operatörlerdir.

Kesirli matematik ifadesi yerine derecesi tamsayı olmayan türevleme ve integrasyon ifadesi kullanılabilir [1-5]. Bu tez çalışmasında kesir dereceli türev ve integral ifadesi tercih edilecektir. Kesir dereceli sistemler üzerinde yapılan çalışmalar kesir dereceli sistem modellerinin gerçek sistemleri temsil etmekte tamsayı dereceli modellere göre avantajlar sunduğunu göstermiştir [1-3]. Bu avantajların en önemlisi frekans cevabında görülür. Tamsayı dereceli sistem modellerinin genlik cevapları kutup ve sıfır frekanslarında 20 dB 'in katlarında ve faz cevapları  $\pi/2$  katlarında değişebilirken, kesir dereceli sistem modellerinde böyle bir sınırlama görülmez. Dolayısı ile kesir dereceli sistem modelleri frekans cevabı bu kurala uymayan sistemleri daha başarılı modelleyebilmektedir. Zaten kesir dereceli sistem modellemesi tamsayı dereceli sistem modellerini kapsadığı görülür.

Kesir dereceli türev operatörünün uzun bellek etkisi olarak adlandırılmış olan ve türevi alınan fonksiyonun lokal değerleri yanında fonksiyonun önceki değerlerine bağımlı olması durumu ayırık kesir dereceli türev hesaplamalarının işlem karmaşıklığını yükseltir. Bu karmaşıklık, zaman ilerledikçe fonksiyonun önceki değerlerinde tutulması ihtiyacından dolayı sürekli artar. Sonuçta hesaplama

sisteminin bellek ve işlemci gibi kaynakları yetersiz kalmaya başlar. Bu zorluk nedeni ile, kesir dereceli operatörler ve sistem modellerinin sayısal (ayrık) sistemlerde gerçekleştirmesini sağlayabilmek için çalışmalar yapılmıştır. Bu çalışmalar sonucunda, kesir dereceli operatörün hesaplanabilmesi veya sistem modellerinin ayrık sistemlerde gerçekleştirilmesi için yaklaşık eşdeğer sayısal filtreler geliştirilmiştir. Bunlar kesir dereceli sistem modellerinin yaklaşık eşdeğerleri olarak hesaplamalarda kullanılması ile kesir dereceli sistemlerin kabul edilebilir doğruluklarla gerçekleştirilmesi sağlanmaya çalışılmıştır. Bu alanda yapılmış olan çalışmalar kesir dereceli sistemlerin bilgisayar yada kontrol kartı gibi sayısal sistemlerde kabul edilebilir yaklaşıklıkla gerçekleştirilmesine imkan sağlamış, kesirli matematik veya kesir dereceli sistem modelleme ve analizinin diğer mühendislik ve bilim dallarında uygulanabilmesini kolaylaştırmıştır [1,6]. Kesir dereceli sistem uygulamalarının yaygınlaşmasında yaklaşık eşdeğer filtre gerçekleştirmelerinin sağladığı kolaylığın önemli bir rolü olmuştur.

Bu tez çalışmasında kesir dereceli sistemlerin yaklaşık ayrık filtre tasarımlarının meta sezgisel optimizasyon yöntemleri ile iyileştirilmesine dönük bir araştırma yapılmıştır. Bu amaçla, literatürde önerilmiş olan mevcut analitik yaklaşık eşdeğer filtre yapıları incelenmiş ve bu yapılar ile kesir derece transfer fonksiyonlarının IIR filtre formunda gerçekleştirilmesinde kullanılmıştır. Daha sonra, kesir dereceli sistemi temsil eden ayrık IIR filtre tasarımlarının pratik uygulama gereksinimlerine dönük olarak iyileştirilmesini sağlamak için IIR filtrenin genlik ve faz cevaplarının yakınsama performansının ayarlanabilmesine imkan sağlayabilen çoklu amaç fonksiyonuna sahip optimizasyon işlemi gerçekleştirilmiştir. Bu amaçla optimizasyonun maliyet fonksiyonu faz cevabı ve genlik cevabı hatalarının ağırlıklı ortalaması formunda ifade edilmiştir. Optimizasyon sonucunda elde edilen ayrık eşdeğer IIR filtrelerin kararlılığını sağlamak için kararsız IIR filtre çözümünü temsil eden parçacıklara çok yüksek maliyet değerleri atanarak dinamik bir maliyet fonksiyonu tanımlanmıştır. Böylece filtre çözümlerini temsil eden parçacıkların kararsız filtre çözümlerine yol açan bölgeyi tecrübe etmeleri durumunda anlık yüksek maliyet ataması ile kararsız çözüme yol açan bölgelerden uzaklaşmaları sağlanmıştır. Böylece parçacık sürüsünün nispeten düşük maliyetli olan kararlı filtre çözümlerine yol açan bölgelerde araştırma yapmaları zorlanarak çözümlerin kararlı olması sağlanabilmiştir.

Geliştirilen algoritmanın kolay kullanımı için Matlab 'da GUI uygulaması

geliştirildi. Bu GUI uygulaması yardımı ile literatürde önerilmiş olan temel kesir dereceli türev eşdeğerleri kullanılarak kesir dereceli sistem modellerinin yaklaşık eşdeğer IIR filtreleri analitik olarak tasarlanabilmektedir. Elde edilen yaklaşık eşdeğer IIR filtreleri tasarımlarının, uygulama amaç ve ihtiyaçlarına dönük olarak genlik cevabı yada faz cevabının iyileştirilmesi geliştirilen PSO algoritması ile gerçekleştirilmiştir. Geliştirilen GUI uygulaması, elde edilen iyileştirilmiş eşdeğer IIR filtre tasarımlarının faz ve genlik cevabı yakınsama performanslarını, birim basamak zaman bölgesi cevaplarını ve kompleks kök düzlemde kutuplarının dağılımı performans değerlendirmesi için göstermektedir. Böylece, temel analitik yöntemlerin uygulanabildiği, tasarımların PSO iyileştirilebildiği, sonuçların performans değerlendirmelerinin alınabildiği araştırmacı ve öğrencilerin kolaylıkla kullanabilecekleri bir Matlab GUI yazılımı hazırlanmıştır.

Tez çalışmasının bölümleri şöyle özetlenebilir:

Bölüm 2 'de Kuramsal Temeller başlığı altında sistem ve filtre tasarımında temel matematiksel altyapıyı oluşturan Lablace dönüşümü ve Z dönüşümü kısaca değinilmiş ve kesir dereceli türev ve kesir dereceli sistem modelleme konusunda temel bilgiler verilmiştir. Devamında, kesir dereceli türev operatörünün ayırık zaman eşdeğer modellerinin elde edilmesi için literatürde önerilmiş olan temel analitik yöntemler anlatılmıştır. Bu yöntemler iki başlık altında incelenmiştir. Dolaylı ayırıklaştırma yöntemleri başlığı altında Oustasloup yakınsama yöntemi ve CFE yakınsama yöntemleri incelenmiştir. Doğrudan ayırıklaştırma yöntemleri başlığı altında özyinelemeli Tustin dönüşümü ve direk CFE ayırıklaştırma yöntemleri incelenmiştir. Bu bölümde, bu analitik yöntemleri temel alarak kesir dereceli sistemler için yaklaşık eşdeğer IIR filtre tasarımı gerçekleştiren GUI tanıtılmış ve bu yöntemler ile örnek filtre tasarımları sunulmuş ve karşılaştırılmıştır.

Bölüm 3'de geliştirilen PSO algoritması tanıtılmış ve bu algoritmanın Matlab ortamında kodlanma süreci ile ilgili detaylar verilmiştir.

Bölüm 4'te Araştırma Bulguları başlığı altında Bölüm 3 'de anlatılan PSO algoritmasını gerçekleştirmek için geliştirilen GUI tanıtılmış ve bu program ile elde edilen sonuçlar değerlendirilmiştir.

Bölüm 5'de Tartışma ve Sonuç başlığı altında elde edilen sonuçlar değerlendirilmiş, avantaj ve dezavantajlar tartışılmış ve gelecek çalışmalar için öneriler sunulmuştur. Bu tez çalışması için geliştirilen Matlab kodları Ekler bölümünde sunulmuştur.

## 2. KURAMSAL TEMELLER

### 2.1. Sistem Modellemede Kullanılan Temel Matematiksel Dönüşümler

#### 2.1.1. Laplace Dönüşümü

Bu bölümde, sürekli sistemlerin matematiksel modellemesinde sıklıkla kullanılan Laplace dönüşümü kısaca tanıtılacak ve kesirli dereceli türev ve integrallerin Laplace dönüşümü ele alınacaktır.

Laplace dönüşümü lineer diferansiyel denklemlerin çözümünde, lineer sistem analizi ve tasarımında yaygın olarak kullanım bulur. Lineer sabit katsayılı diferansiyel denklem sistemleri Laplace dönüşümü kullanıldığında denklem çözümü basit cebirsel denklem sistemine dönüşürler [7]. Bu cebirsel denklemler diferansiyel denklemlerin  $s$ -tanım bölgesinde elde edilir ve zaman tanım bölgesindeki çözüme ters Laplace dönüşümü uygulanarak ulaşılır [7].

Laplace dönüşümü temelde bir integral dönüşümdür ve integral dönüşümü,

$$T[f(t)] = F(s) = \int_a^b f(t)k(s,t)dt \quad (2.1)$$

şeklinde tanımlanır. Burada  $k(s,t)$  fonksiyonuna integral dönüşümünün çekirdeği (kernel) denir. Burada  $F(s)$  fonksiyonu  $f(t)$  fonksiyonunun  $T[f(t)]$  dönüşüm altındaki sonucudur. Dolayısı ile  $f(t)$  fonksiyonu  $F(s)$  'nin ters dönüşümüdür ve  $T^{-1}[F(s)]$  ile gösterilir. Çekirdeğin ve sınırların,  $k(s,t) = e^{-st}$ ,  $a = 0$ ,  $b = \infty$  olduğu durum için Laplace dönüşümü elde edilir. Eğer çekirdek ve sınırlar  $k(s,t) = e^{-i\omega t}$ ,  $a = -\infty$ ,  $b = \infty$  ile ifade edilirse Fourier dönüşümleri elde edilir.

Bu çerçevede,  $t > 0$  olmak üzere  $f(t)$  fonksiyonun Laplace dönüşümü,

$$F(s) = L\{f(t)\} = \int_0^{\infty} e^{-st} f(t)dt \quad (2.2)$$

olarak ifade edilir. Dinamik sistem analizi için türev operatörünün Laplace dönüşümü önemlidir. Kesir dereceli türevin Laplace dönüşümü, Denklem (2.3) ile ifade edilir.

$$L\left\{\frac{d^m f(t)}{dt^m}\right\} = s^m L\{f(t)\} - \sum_{k=0}^{n-1} s^k \left[\frac{d^{m-1-k} f(t)}{dt^{m-1-k}}\right]_{t=0} \quad (2.3)$$

Burada  $n$  bir tamsayı olup  $n - 1 < m < n$  şeklindedir. Burada türevlerin başlangıç

koşullarındaki değerler sıfır alınırsa daha basit bir forma kavuşur ve Denklem (2.4) 'deki gibi elde edilir.

$$L \left\{ \frac{d^m f(t)}{dt^m} \right\} = s^m L \{ f(t) \} \quad (2.4)$$

Bu ifade ters dönüşüm işlemlerinde kolaylık sağlamaktadır. Örneğin kesirli integral işlemi için ters Laplace dönüşümü alınırsa,

$$L \left\{ \frac{t^{m-1}}{\Gamma(m)} \right\} = \frac{1}{s^m} \quad (2.5)$$

elde edilir. Bunun ters Lablace dönüşümü,

$$L^{-1} \left\{ \frac{1}{s^m} \right\} = \frac{t^{m-1}}{\Gamma(m)} \quad (2.6)$$

şeklinde elde edilir.

Benzer şekilde elde edilen bu sonucu kullanarak frekans öteleme formülüne uygulanırsa Denklem (2.7) 'deki gibi bir sonuç elde edilir [8].

$$L^{-1} \left\{ \frac{1}{(s+a)^m} \right\} = \frac{t^{m-1} e^{-at}}{\Gamma(m)} \quad (2.7)$$

### 2.1.2. Z Dönüşümü

Lineer diferansiyel denklemler ile ifade edilen sürekli zaman lineer sistem modellerinin çözümünde Laplace dönüşümü yardımı ile analiz ve çözümler kolaylaştırılır. Benzer şekilde lineer fark denklemleri ile ifade edilen ayrık zaman sistemlerin analiz ve çözümünde z dönüşümü kolaylık sağlar. Z dönüşümü fark denklemlerinin cebirsel formda ifade edilmesini sağlar [9].

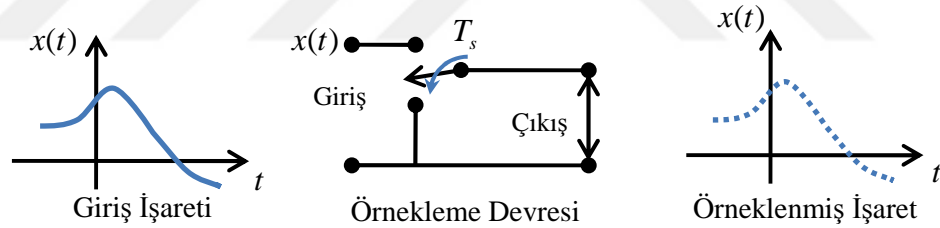
Tek yönlü ve çift yönlü olmak üzere Z dönüşümü iki şekilde ifade edilmiştir. Tek yönlü Z dönüşümü aşağıdaki gibi tanımlanır.

$$Z\{y(k)\} = Y(z) = \sum_{k=0}^{\infty} y(k)z^{-k} \quad (2.8)$$

Çift yönlü Z dönüşümü toplam sınırları  $-\infty$  ile  $+\infty$  aralığında tanımlanır. Bir sürekli zaman işaretinin bilgisayar ve mikro denetçi kartları gibi sayısal sistemlerde

işlenebilmesi için bu işaretin periyodik olarak örnekleme, diğer bir ifade ile belli bir periyot ile sinyal genliğinin ayırık değerler olarak ölçülmesi gerekmektedir. Bu periyodik olarak ayırık ölçümleme işlemine örnekleme denir ve yapılan ölçümlerin kümesi sürekli zaman işareti sayısal sistemlerde temsil edilen sayısal veri kümesini oluşturur. Nyquist örnekleme teoremi, sınırlı bir bant genişliğine sahip sürekli zaman sinyallerden yeterince çok örnek(ölçüm) alınması durumunda bu örneklerden kayıpsız olarak tekrar sürekli zaman sinyalinin elde edilebilme koşullarını ifade eder. Bu çalışma sayısal (ayırık) sistemleri konu aldığı için Nyquist örnekleme teorisine kısaca değinilmiştir.

Frekans spektrumu  $F_s$  frekansı ile sınırlı, bir sınırlı bant bir işaret,  $T_s \leq 1/2F_s$  periyotla örneklenirse bu işaret tam ve tek olarak yeniden elde edilebilir. Bu teorem, Shannon-Nyquist örnekleme teoremi olarak ifade edilir. Teorem temel olarak sınırlı bantlı bir  $x(t)$  analog işaretinin yeniden ve bozulmadan elde edilmesi için gereken örnekleme periyodu için bir sınır tanımlamaktadır. Örnekleme periyodu  $T_s > 1/2F_s$  durumunda örnekleme işlemi sonucunda  $x(t)$  tekrar elde edilemez.



Şekil 2.1. Örnekleme işlemi

Yukarıda belirtilen yöntemlere göre örneklenmiş bir sinyalin Z dönüşümü de aşağıdaki verildiği gibi ifade edilebilir.

$$Z\{x(kT_s)\} = X(z) = \sum_{k=0}^{\infty} x(kT_s)z^{-k} \quad (2.9)$$

Z dönüşümüne dikkat edilirse sonsuza giden bir seri olduğu görülür ve dolayısı ile burada serilerin yakınsaklığı konusu önem kazanır. Bazı durumlarda sonsuz seriler yakınsaktır ve z dönüşümü bu yakınsaklık durumları ile kolaylıkla gerçekleştirilir. Bu nedenle z dönüşümü bulunurken seri yakınsaklığı önem arz eder. Z dönüşümü için en önemli seri açılımı aşağıda verilen kuvvet serisi açılımı ve

yakınsaklık koşuludur.

$$\sum_{n=0}^{\infty} q^n = 1 + q + q^2 + q^3 + \dots + q^n = \frac{1}{1-q}, \quad |q| < 1 \quad (2.10)$$

Çizelge 2.1 ve Çizelge 2.2 ‘ de bazı ayrık ve sürekli temel fonksiyonlar için belirlenmiş z dönüşümleri verilmiştir.

Çizelge 2.1. Bazı ayrık fonksiyonların z dönüşümleri

Fonksiyon $f(k)$	Z Dönüşümü $F(z)$
$\delta(k)$	1
$u(k)$	$\frac{z}{z-1}$
$K$	$\frac{z}{(z-1)^2}$
$k^2$	$\frac{z(z+1)}{(z-1)^3}$
$(a)^k$	$\frac{z}{z-a}$
$\text{Sin}(wk)$	$\frac{z \text{Sin } w}{z^2 - 2z \text{Cos } w + 1}$
$\text{Cos}(wk)$	$\frac{z(z - \text{Cos } w)}{z^2 - 2z \text{Cos } wa + 1}$

Çizelge 2.2. Bazı temel fonksiyonların z dönüşümleri

Analog Fonksiyon $f(t)$	Örneklenmiş Fonksiyon $f(kT)$	z-Dönüşümü $F(z)$
$t$	$kT_s$	$T_s \frac{z}{(z-1)^2}$
$e^{-bt}$	$e^{-bkT_s}$	$\frac{z}{z - e^{-bT_s}}$
$\text{sin}(wt)$	$\text{sin}(w_k T_s)$	$\frac{z \text{Sin } w T_s}{z^2 - 2z \text{Cos } w T_s + 1}$
$\text{cos}(wt)$	$\text{cos}(w T_s)$	$\frac{z(Z - \text{Cos } w T_s)}{z^2 - 2z \text{Cos } w T_s + 1}$

### 2.1.3. Bazı Önemli Z Dönüşüm Teoremleri

(i) Toplama-Çıkarma ve Sabit Çarpanı:

$$Z\{ay_1(k) \pm by_2(k)\} = aY_1(z) \pm bY_2(z) \quad (2.11)$$

(ii) Geciktirme

- İki yönlü dönüşüm

$$Z\{y(k - n)\} = z^{-n}Y(z) \quad (2.12)$$

- Bir yönlü dönüşüm

$$Z\{y(k - n)\} = z^{-n}[Y(z) + \sum_{k=-1}^{-n} y(k)z^{-k}] \quad (2.13)$$

(iii) Öngörme

- İki yönlü dönüşüm

$$Z\{y(k + n)\} = z^n Y(z) \quad (2.14)$$

- Bir yönlü dönüşüm

$$Z\{y(k + n)\} = z^n [Y(z) + \sum_{k=0}^{n-1} y(k)z^{-k}] \quad (2.15)$$

(iv) İlk Değer Teoremi

$$Y(0) = \lim_{k \rightarrow 0} y(k) = \lim_{z \rightarrow \infty} Y(z) \quad (2.16)$$

(v) Son Değer Teoremi

$$Y(\infty) = \lim_{k \rightarrow \infty} y(k) = \lim_{z \rightarrow 1} (z - 1)Y(z) \quad (2.17)$$

(vi) Karmaşık Dönüşüm(üstel ile çarpma)

$$f(k) = a^k f_1(k) \Rightarrow F(z) = F_1\left(\frac{z}{a}\right) \quad (2.18)$$

(vii) Rampa ile Çarpma

$$f(k) = k^n f_1(k) \Rightarrow F(z) = \left(-z \frac{d}{dz}\right)^n F_1(z) \quad (2.19)$$

#### 2.1.4. Sürekli Sistem Modellerinin Ayrıklaştırılması

Sürekli sistem modellerinin sayısal sistemlerde gerçekleştirilmesi ve temsil edilebilmesi için bu sistem modellerinin ayrıklaştırılması gerekmektedir. Ayrıklaştırma işlemini,

$$z = e^{sT_s} \quad (2.20)$$

ifadesine bağlıdır. Burada  $s$  çekilirse,

$$s = \frac{1}{T_s} \ln z \quad (2.21)$$

elde edilir. Bu ifade sürekli zaman sistem modelinden ( $s$  tanım bölgesinden) ayrık zaman sistem modeline ( $z$  tanım bölgesinden) geçişi sağlar. Ancak, sistem modellerinin polinomsal ve rasyonel fonksiyonlar olması durumunda kolay gerçekleştirilebildiği için Denklem 2.21 ile ifade edilen logaritmik dönüşüm fonksiyonunun yaklaşık eşdeğerleri kullanılır. Bu yaklaşık eşdeğerler dönüşüm sonucunun da polinomsal ve rasyonel bir fonksiyon olmasını sağlar. Çizelge 2.3'de ayrıklaştırma işlemi için önerilmiş yaklaşık eşdeğer dönüşümler gösterilmiştir. Ayrık sistemler için kararlılık bölgesi  $z$  kompleks düzleminde birim çemberin içidir. Sürekli zaman sistem kararlılık bölgesi olan sol yarı düzlem, Tustin yöntemi ile doğrudan birim çember içine karşılık düşer. Bu nedenle, Tustin ayrıklaştırma yöntemi sistem kararlılık durumu bilgisini korur. Tabloda belirtilen diğer bazı temel yöntemler için bu söylenemez.

Çizelge 2.3. Bazı Temel Dönüşüm Fonksiyonları

Yöntem Adı	Dönüşüm Fonksiyonu
<i>Euler metodu(İleri Yönlü fark)</i>	$s \approx \frac{z - 1}{T_s}$
<i>Euler metodu(Geri Yönlü fark)</i>	$s \approx \frac{z - 1}{zT_s}$
<i>Tustin Yöntemi</i>	$s \approx \frac{2z - 1}{T_s z + 1}$

## 2.2. Kesir Dereceli Türev

### 2.2.1. Kesir Dereceli Türev Operatörünün Temel Özellikleri

Kesirli dereceli türev ve integralin özellikleri şunlardır [6]:

1-  $f(t)$  fonksiyonu  $t$ 'nin analitik bir fonksiyonu ise, onun kesirli türevi  ${}_0D_t^\alpha f(t)$   $t$  ve  $\alpha$  'nın analitik bir fonksiyonudur.

2- Burada  $n$  bir tam sayı ise,  $\alpha = n$  için,  ${}_0D_t^\alpha f(t)$  işlemi tamsayı dereceli  $n$  'nin klasik türevi ile aynı sonucu verir. Kesir dereceli türev ve integral operatörü en genel formda şöyle ifade edilmiştir [8].

$${}_0D_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \alpha > 0 \\ 1 & \alpha = 0 \\ \int_a^b (dt)^\alpha & \alpha < 0 \end{cases} \quad (2.22)$$

3-  $\alpha = 0$  için  ${}_0D_t^\alpha f(t)$  işlemi özdeşlik işlemidir. Diğer bir ifade ile  ${}_0D_t^0 f(t) = f(t)$  yazılır.

4- Kesirli türev ve kesirli integral lineer operatörlerdir. Diğer bir ifade ile  ${}_0D_t^\alpha (af(t) + bg(t)) = a{}_0D_t^\alpha f(t) + b{}_0D_t^\alpha g(t)$  olarak yazılır.

5-  ${}_0D_t^\alpha {}_0D_t^\beta f(t) = {}_0D_t^\beta {}_0D_t^\alpha f(t) = {}_0D_t^{\alpha+\beta} f(t)$  geçerlidir.

### 2.2.2. Kesir Dereceli Türev Tanımları

Kesir dereceli türev için literatürde çeşitli tanımlar verilmiştir. Yaygın olarak kullanılan iki tanım Grünwald- Letnikov (G-L) ve Riemann-Liouville (R-L) tanımlarıdır [8]. Bazı özel durumlar için bu tanımların birbirine özdeş oldukları gösterilmiştir. Bu iki tanım dışında, literatürde yaygın olarak kullanılan tanım Caputo tanımıdır. Bu bölümde bu tanımlar verilecektir.

(i) *Grünwald-Letnikov Tanımı* [8,10]: Bu tanım ilk olarak Anton Karl Grünwald tarafından 1867 de önerilmiş ve Aleksey Vasilievich Letnikov tarafından 1868 yılında geliştirilmiştir [8]. Grünwald-Letnikov türev tanımı  $p > 0$  olmak üzere [8];

$${}_a D_t^p f(t) = \lim_{h \rightarrow 0} h^{-p} \sum_{r=0}^n (-1)^r \binom{p}{r} f(t - rh) \quad (2.23)$$

gerekli düzenleme yapılırsa,

$${}_a D_t^p f(t) = \lim_{h \rightarrow 0} h^{-p} \sum_{r=0}^n (-1)^r \frac{\Gamma(p+1)}{\Gamma(r+1)\Gamma(p-r+1)} f(t - rh), \quad (2.24)$$

$\left[\frac{t-a}{h}\right]$  operatörü  $\frac{t-a}{h}$  işlemin tam kısmını ifade eder. Gama fonksiyonu ile ifade edilen  $\Gamma(\cdot)$  ifadesiyle verilir. Bu tanımın aşağıdaki şekilde ifade edilebileceği gösterilmiştir.

İntegral tanımı ise yine  $p > 0$  olmak üzere;

$${}_a D_t^{-p} f(t) = \lim_{h \rightarrow 0} h^p \sum_{r=0}^n \binom{p}{r} f(t - rh), \quad (2.25)$$

şeklinde verilmiştir.

(ii) *Riemann-Liouville Tanımı* [7,8,10]: Riemann-Liouville türev tanımı kesir dereceli türevi ifade etmek için kullanılan tanımlardan bir diğeridir. Burada  $p$  sıfırdan büyük herhangi bir sayı olmak üzere  $f(t)$  fonksiyonunun  $p$ . mertebeden türevi,

$${}_a D_t^p f(t) = \left(\frac{d}{dt}\right)^{m+1} \int_a^t (t - \tau)^{m-p} f(\tau) d\tau, \quad (m \leq p \leq m + 1) \quad (2.26)$$

ifadesi ile verilir [8].

(iii) *Caputo Tanımı* [7,8]: Bu kesir derece türev tanım İtalyan matematikçi Michele Caputo tarafından 1960'larda ortaya konmuş, özellikle Laplace dönüşümü alındığında daha kullanışlı başlangıç değer ifadeleri içerdiği için uygulamalı alanlarda ve mühendislikte tercih edilir [7,8]. Caputo tanımı,

$${}_a^c D^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (2.27)$$

ile ifade edilir. Burada,  $\Gamma(x) = \int_a^t e^{-t} t^{x-1} dt$  Gama fonksiyonudur,  $\alpha$  integral alt sınırıdır. Burada  $n - 1 \leq \alpha < n$  kesirli dereceli işleminin sırası ve  $n$  bir tamsayıdır [6].

$f(t)$  'nin Laplace transfer fonksiyonu  $F(s)$  ise, Riemann-Liouville ve Caputo tanımlarına göre kesirli dereceli operatörlerinin Laplace transfer fonksiyonları şu şekilde verilir:

$$L[{}_0D^\alpha f(t)] = s^\alpha F(s) - \sum_{k=0}^{n-1} s^k f^{(\alpha-k-1)}(t)|_{t=0} \quad (2.28)$$

$$L[{}_0^C D^\alpha f(t)] = s^\alpha F(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} f^{(k)}(t)|_{t=0} \quad (2.29)$$

$f(t)$  'nin bütün başlangıç durumları sıfır alınırsa  $f^{(k)}(0) = 0$ , bu ifadelerin sağ tarafındaki toplam terimleri sıfır olacağı için Laplace transfer fonksiyonları aynı forma kavuşur.

$$L[{}_0D^\alpha f(t)] = s^\alpha F(s) \quad (2.30)$$

Dolayısı ile,  $\alpha$  kesir dereceli türev operatörünün Laplace transfer fonksiyonu  $s^\alpha$  'dır. Çoğu zaman  $s^\alpha$  kesir dereceli türevin  $s$  tanım bölgesi ifadesi olarak kullanılır.

### 2.3. Kesir Dereceli Lineer Zamanla Değişmeyen (LZD) Sistem Modelleri ve Kararlılığı

Zaman bölgesinde, lineer kesir dereceli bir sistemi modelleyen kesirli dereceli diferansiyel denklem en genel formda Denklem 2.31 ' da verilmiştir;

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \dots + a_0 D^{\alpha_0} y(t) = b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \dots + b_0 D^{\beta_0} u(t) \quad (2.31)$$

Burada  $a_n > a_{n-1} > \dots > a_0$  ve  $\beta_m > \beta_{m-1} \dots > \beta_0$  gerçekte sayılardır ve kesir derece türevleri ifade ederler.  $a_k (k = 1, 2, \dots, n)$  ve  $b_k (k = 1, 2, \dots, m)$  gerçekte sayıları kesir dereceli sistemin sabit katsayılarıdır. Sistem analizi için genellikle Denklem (2.31) 'in Laplace dönüşümü alınır ve kesir dereceli sistemin transfer fonksiyonu elde edilir. Burada  $u(t)$  sistem girişi ve  $y(t)$  sistem çıkışı için başlangıç durumlarının sıfır olduğu varsayılır ve kesir dereceli sürekli sistemin transfer fonksiyonunun genel bir ifadesi için,

$$G(s) = \frac{N(s)}{D(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0}} \quad (2.32)$$

yazılabilir. Transfer fonksiyonu doğrusal olan kesirli dereceli sistemde  $s^\alpha$  gibi bir dizinin kesirli dereceli türev işlemi içerdiğini görülür. Bu rasyonel fonksiyonun payda polinomu sistemin karakteristik polinomu olarak adlandırılır ve sistemin kararlılık analizinde büyük öneme sahiptir. Yukarıdaki transfer fonksiyonu için karakteristik polinom,

$$D(s) = a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0} \quad (2.33)$$

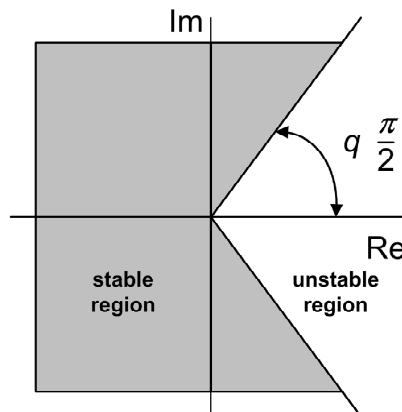
ile ifade edilir.

Bir karakteristik polinomun bütün köklerin reel kısmı negatif ise LZD (linear, zamanla değişmeyen) sistemin sol yarı düzlem (Left Half Plane) kararlıdır denir. Bu durum, karakteristik polinom köklerinin kararlılık bölgesi olan karmaşık düzlemin sol yarısında bulunduğunu ifade eder. Kesirli-dereceli LZD sistemler için kararlılık bölgesi; açısı  $\frac{\pi}{2}$  olan karmaşık (sanal) eksenin sol tarafı değildir. Kesir derece nedeni ile kararlılık bölgesi aşağıdaki koşulu sağlayan diğer bir ifade ile açısı  $\pm p \frac{\pi}{2}$  olan düzlem parçalarının sol tarafıdır (Şekil 2.2) [11].

Matignon kararlılık teoremi kesirli dereceli transfer fonksiyonu  $G(s) = \frac{N(s)}{D(s)}$  için kararlılık koşulunu,

$$|\arg(\sigma)| > q \frac{\pi}{2}, \forall \sigma \in C, P(\sigma) = 0, \quad (2.34)$$

ile ifade etmiştir [11]. Burada  $\sigma = s^q$  dur.



Şekil 2.2. LZD kesir dereceli sistemlerin kararlılık bölgesi [11]

## 2.4. Kesir Dereceli Sistemlerin Yaklaşık Eşdeğer Modellemesi ve Sayısal Gerçekleşmesi

Kesir dereceli sistemlerin modellerinin analiz ve simülasyonları daha kolay gerçekleştirilebilir ve hesaplanabilir nitelikte olan yaklaşık eşdeğerler modeller yardımı ile gerçekleştirilir. Bu durum örneğin irrasyonel bir sayının rasyonel yaklaşımını elde etmeye ve hesaplamalarda kullanmaya benzetilebilir. Bu konuda mühendislik hesaplamaların sık kullandığımız  $\pi = 3.1415926\dots$  sayısı örnek verilebilir. Hesaplamalarda pi sayısı ondalık sayı olarak ifade edildiğinde, noktadan sonra sonsuza giden ondalık kısım işlem maliyetini çok artırdığı için, pi sayının istediğimiz hassasiyette rasyonel bir yaklaşık değerini yuvarlama yaparak kullanabiliyoruz. Daha yakın bir örnek olarak, tamsayı dereceli türev yerine nümerik hesaplamalarda yaklaşık eşdeğeri olarak sonlu fark eşdeğeri kullanıyor olmasını verebiliriz.

### 2.4.1. Kesir Dereceli Transfer Fonksiyonun Ayrıklaştırılması

Kesir dereceli sistem modellerinin ayrık analizi, simülasyonu ve sayısal gerçekleştirilmesi yüksek işlem maliyetleri getirebilmektedir. Örneğin, Grunwald-Letnikov tanımını incelediğimizde kesir dereceli türevin o anki değerlerinin kesir dereceli fonksiyonun sifıra kadar bütün eski değerlerine bağlıdır. Bu, her yeni hesaplamada bu değerlerin tekrar tekrar artan sayıda hesaba katılmasını gerektirir. Buna uzun bellek etkisi (long-memory effect) denir. Bu zaman ilerledikçe, her yeni kesir derece türev hesaplaması için işlem maliyetini sürekli artırır. Bu etki sonucunda kesir dereceli sistemlerin sayısal sistemlerde ideal gerçekleştirilmesi yüksek işlem maliyetleri getirmektedir. Bu nedenle, kesir dereceli sistemlerin sayısal sistemlerde gerçekleştirilmesi için tamsayı dereceli eşdeğer modelleri önerilmiştir. Çünkü tamsayı dereceli türevlerin nümerik analizi, hesaplaması yapılan noktanın yakın civarındaki fonksiyon değerlerine ihtiyaç duyar ve işlem maliyeti bu nedenle oldukça sınırlanır. Kontrol uygulamaları gibi birçok alanda, kesir dereceli sistem modellerinin temsili için tamsayı dereceli yaklaşık eşdeğer modeller kullanılmış ve bu amaçla yaklaşık eşdeğer model üreten yöntemler önerilmiştir. Oustaloup yaklaşım metodu ve CFE (continued fraction expansion - sürekli kesir açılımı) metodu bu alanda yaygın olarak kullanılan analitik yöntemlerdir. Bunlar dışında kesirli dereceli türev operatörünün tamsayı dereceli yaklaşık eşdeğer modellerini iyileştirmek için sezgisel optimizasyon

yöntemleri [12] kullanılmıştır.

Mühendislikte kesir dereceli matematiğin kullanımının yaygınlaşması ile birlikte, sayısal sistemlerde, nümerik analizde ve sistem simülasyonlarında kesir dereceli sistem modelleri sıklıkla kullanılmaya başlanmıştır. Bu nedenle, sistemin çalışma frekans bölgesinde kesir dereceli transfer fonksiyonuna genlik veya faz cevabı bakımından daha iyi yakınsayan ayrıklaştırma yöntemlerine ihtiyaç duyulmuştur. Geliştirilen ayrıklaştırma yöntemleri ile kesir dereceli transfer fonksiyonları, ayrık filtreler ile sayısal sistemlerde gerçekleştirilmiş, simülasyon ve nümerik analizler makul işlem maliyetleri ile kabul edilebilir doğruluklarda gerçekleştirilebilmiştir.

Kesir dereceli diferansiyel sistemler  $s$  tanım bölgesinde  $s^\alpha$  ve  $s^{-\alpha}$  ;  $\alpha \in R$  , gibi terimler veya bunların kombinasyonundan oluşturulan transfer fonksiyonları ile ifade edilmektedir. Kesir dereceli bir transfer fonksiyonunun belli bir frekans bandı aralığında davranışına yakınsayabilecek olan işlem maliyeti daha düşük ve kolay gerçekleştirilebilir olan tamsayı dereceli yaklaşık eşdeğer transfer fonksiyonu kullanılabilmesi için yaklaşım yöntemleri önerilmiştir [13]. Bazı yaklaşım yöntemleri frekans bölgesi davranışı daha iyi temsil edebilirken bazı yöntemler daha iyi zaman bölgesi cevabı temsil edebilmektedir. Bu nedenle en iyi yaklaşım metodunun hangisi olduğunu kesin olarak söyleyemeyiz ancak bir uygulama gereksinimlerine daha iyi cevap verebilen bir yöntem belirlenebilir [13].

#### **2.4.2. Kesir Dereceli Türevin Ayrık Zaman Yaklaşık Eşdeğer Modellerin Elde Edilmesi**

Bu bölümde kesir dereceli türevin ayrık zaman eşdeğer modellerinin elde edilmesi incelemiştir. Bu amaçla geliştirilen yöntemler iki grupta incelenir.

i. *Dolaylı (Indirect) Ayrıklaştırma Yöntemleri*: Kesir dereceli türev operatörünün ayrıklaştırılmasının aşamalı olarak gerçekleştirildiği yöntemlerdir. Dolaylı yöntem grubunda, kesir dereceli transfer fonksiyonun önce sürekli frekans bölgesinde tamsayı dereceli sürekli zaman bir eşdeğer modeli elde edildikten sonra bu tamsayı dereceli modelin ayrıklaştırıldığı iki aşamalı yöntemler ele alınacaktır. Bu çalışmada Oustasloup Yakınsama yöntemi ve CFE yakınsama yöntemi yapılabilen dolaylı ayrıklaştırma yöntemleri incelenecektir.

ii. *Doğrudan (Direk) Ayırıklaştırma Yöntemleri*: Kesir dereceli türev operatörünün doğrudan ayırık zaman filtre gerçekleştirmesini üreten yaklaşımlardır. Bu grupta özyinelemeli Tustin dönüşümü ve direk CFE ayırıklaştırma yöntemi incelenecektir.

### 2.4.3. Dolaylı (Indirect) Ayırıklaştırma Yöntemleri

Dolaylı ayırıklaştırma iki adımda yapılır; önce kesirli mertebeden türev veya integralin sürekli zamanda rasyonel fonksiyon ile ifade edilen tamsayı dereceli yaklaşık eşdeğer ifadeleri s-tanım bölgesinde elde edilir [8]. Sonra s-tanım bölgesinde elde edilen transfer fonksiyonundan temel ayırıklaştırma yöntemleri ile z-tanım bölgesine geçilir [8,9,14,15]. Bu tez çalışmasında s-tanım bölgesinde elde edilen tamsayı dereceli yaklaşık eşdeğer transfer fonksiyonlarının ayırıklaştırılması için Tustin yöntemi kullanılmıştır. Bunun için tamsayı dereceli eşdeğer transfer fonksiyonda görülen  $s$  yerine aşağıdaki verilen Tustin eşdeğeri kullanılır.

$$s \rightarrow \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} \quad (2.35)$$

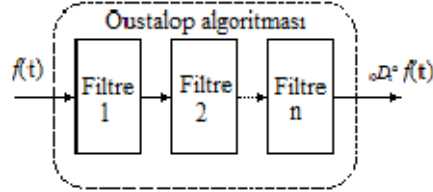
Bu işlem çalışmalarımızda Matlab programlama dilinin  $c2d(.)$  fonksiyonu yardımı ile gerçekleştirilmiştir.

Bu bölümde yaygın olarak kullanılan kesir derece türevin sürekli frekans bölgesi eşdeğerini elde etmek için önerilen iki yakınsama yöntemi özetlenmiştir:

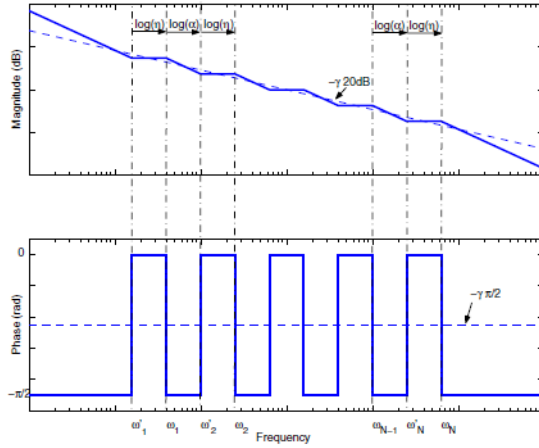
(i) *Oustasloup Yakınsama Yöntemi ile Kesir Dereceli Türevin Tamsayı Dereceli Eşdeğer Modeli*

Oustasloup yakınsama yönteminde, verilen frekans bandı küçük aralıklarla bölünür. Her aralıkta, kesir dereceli operatöre, kat kat bağlı birinci derece sürekli filtre serisi ile yakınsanır. Bütün frekans bandı boyunca yakınsama bu filtrelerin Şekil 2.3 'de görüldüğü gibi seri bağlanması ile sağlanır. Temelde kesir dereceli filtrenin frekans cevabına, her bir birinci derece filtrenin sağlamış olduğu sıfırların ( $\omega'_k$ ) ve kutupların ( $\omega_k$ ) frekansları ayarlanarak yakınsanır. Denklem (2.37) ile sıfırların yeri ve denklem (2.38) ile kutupların yerleri belirlenir. Her sıfır genlik cevabında 20 dB ve faz cevabında  $\frac{\pi}{2}$  artım sağlar. Her kutup ise genlik cevabında -20dB ve faz cevabında  $-\frac{\pi}{2}$  düşüm sağlar. Bu artım ve düşümler, sıfır ve kutupların

Şekil 2.4 'de görüldüğü gibi ardışık olarak peş peşe getirilmesi ile  $s^\gamma$  kesir dereceli türev operatörünün genlik cevabı olan  $\gamma$  20dB ve faz cevabı  $\gamma \frac{\pi}{2}$  'ye bir yakınsama sağlanır.



Şekil 2.3. Oustaloup yöntemi seri filtre dizisi [8]



Şekil 2.4. Oustaloup yöntemi seri filtre grubunun oluşturduğu kutup ve sıfırların kesir derece türevin frekans cevabına yakınsaması [16]

Yukarıda anlatıldığı gibi Oustaloup algoritmasının temelinde seçilen frekans bandı içinde kutup ve sıfır frekansları uygun olarak ayarlanmış tamsayı dereceli filtrelerin seri bağlanması ile  $s^\alpha$  'nın yaklaşık değerinin elde edilmesi yatar. Bu algoritma aşağıda özetlenmiştir [8];

Uygun frekans aralığı olarak  $\omega_b \cdot \omega_h = 1$  olacak şekilde  $(\omega_b, \omega_h)$  seçilirse, çok katlı filtre transfer fonksiyon modeli aşağıdaki gibi yazılabilir:

$$s^\gamma \cong G_j(s) = K \prod_{k=-N}^N \frac{s + \omega'_k}{s + \omega_k} \quad (2.36)$$

Burada,  $G_j(s)$  sıfırlarının frekansı

$$\omega'_k = \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k+N+(1-r)/2}{2N+1}} \quad (2.37)$$

ve kutuplarının frekansı,

$$\omega_k = \omega_b \left( \frac{\omega_h}{\omega_b} \right)^{\frac{k+N+(1-r)/2}{2N+1}} \quad (2.38)$$

Burada  $K = \omega_k^r$  ' dir [17].

**Örnek 2.1 :** Aşağıda verilen kesir dereceli alçak geçiren filtrelerinin transfer fonksiyonunu Oustaloup yöntemi ile dolaylı ayrıklaştırma uygulayarak ayrık IIR filtre formunda gerçekleyelim ve yakınsama performansını inceleyelim.

$$F(s) = \frac{1}{s^{0.9}+1} \quad (2.39)$$

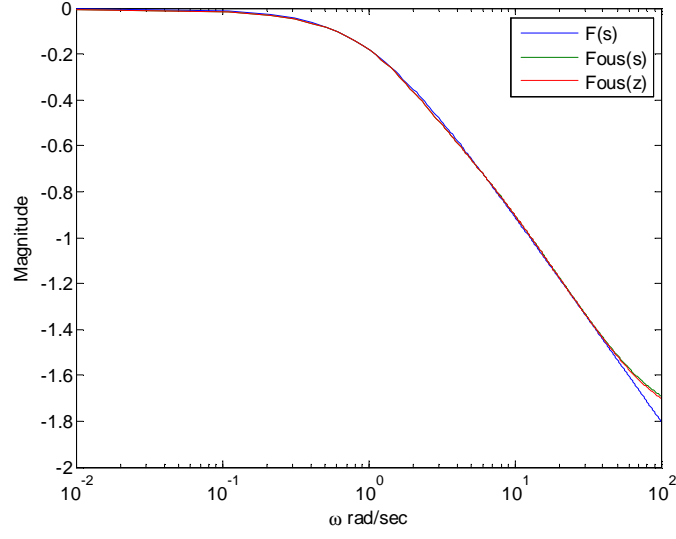
Bu örnekte tasarlamış olduğumuz Matlab GUI ile sonucu elde ettik, sonucu elde ederken örnekleme periyodu  $T=0,001$  sn, IIR filtre derecesi 3 ve Oustaloup'un frekans aralıkları 0,01-100 arasında belirlenerek elde edilmiş Oustaloup sürekli transfer fonksiyonu aşağıda verilmiştir.

$$F_{ous}(s) = \frac{s^3+89.94s^2+358s+63.1}{64.1s^3+448s^2+448s+64.1} \quad (2.40)$$

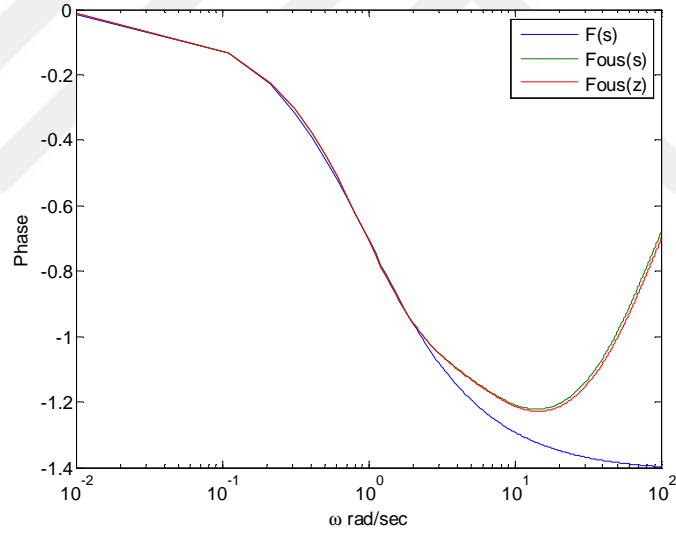
Bu fonksiyon Matlab  $c2d()$  fonksiyonu yardımı ile Tustin yöntemi uygulanarak ayrıklaştırıldığı zaman, aşağıdaki ayrık transfer fonksiyonu elde edilir.

$$F_{ous}(z) = \frac{0.0156z^3-0.0454z^2+0.04401z-0.01421}{z^3-2.993z^2+2.986z-0.993} \quad (2.41)$$

Bu yöntem sonucunda elde edilen ayrık IIR filtresinin genlik ve faz cevapları Şekil 2.5 ve Şekil 2.6 'da görülmektedir.



Şekil 2.5.Oustaloup yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının genlik cevapları

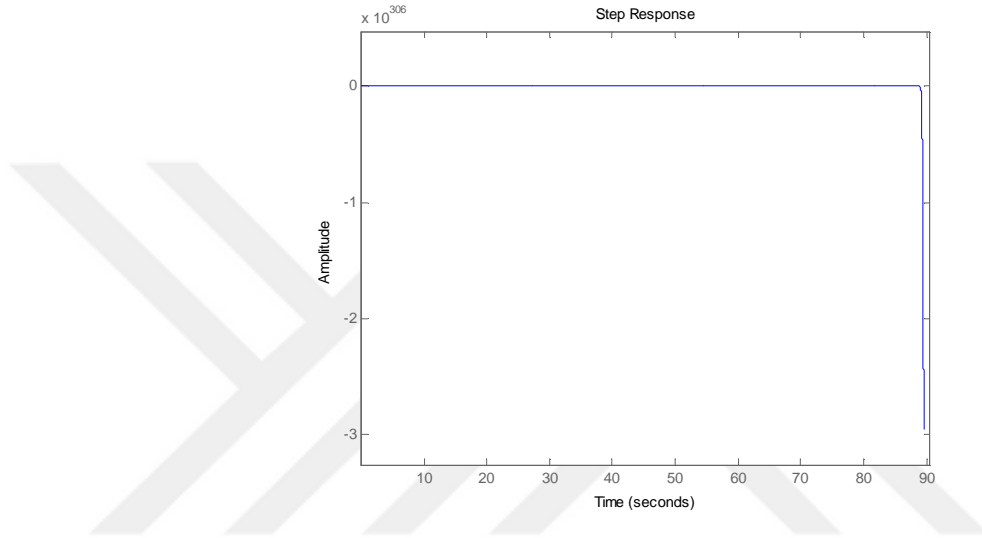


Şekil 2.6.Oustaloup yöntemi uygulanarak elde edilmiş transfer fonksiyonlarının faz cevapları

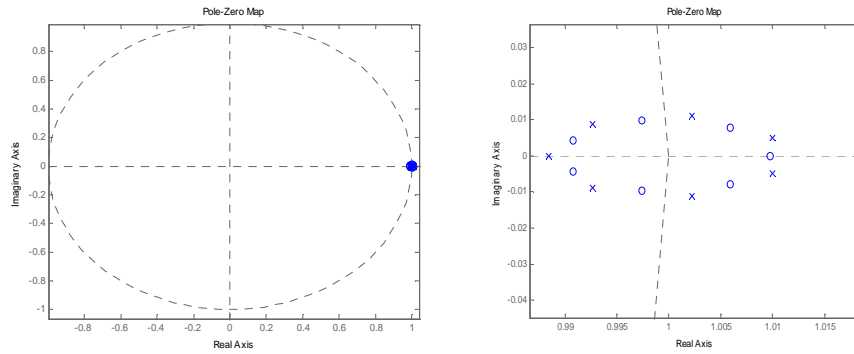
Çizelge 2.4 'de bu sistemin genlik ve faz cevap yakınsama hata değerleri verilmiştir.

Çizelge 2.4. Genlik ve faz cevabı hatası

<i>Frekans Cevabı</i>	<i>Ortalama Karesel Hata</i>
Faz Cevabı	0.1785
Genlik Cevabı	$7.5 \cdot 10^{-6}$



Şekil 2.7. Birim basamak cevabı



Şekil 2.8. Sistem kutup ve sıfırlarının dağılımı

Şekil 2.8 'de yöntemin kutup ve sıfır dağılım grafiği verilmiştir. Sağdaki grafik kutup ve sıfırların toplandığı bölgenin yakın görünümüdür. Grafik incelendiğinde bazı kutupların birim çember dışında kaldığı görülmüştür. Bu Oustaloup yöntemi ile yapılan dolaylı ayrıklaştırmanın kararsız filtre sonucu

üretebildiğini göstermektedir.

(b) *CFE Yakınsama Yöntemi ile Kesir Dereceli Türevin Tamsayı Dereceli Eşdeğer Modeli*

Kesir dereceli sistem transfer fonksiyonlarının yaklaşık eşdeğer modelinin elde edilmesi için önerilen bir diğer yöntem sürekli kesir açılımı (CFE) yöntemidir [18].

Kuvvet serilerine açılım gibi, bazı irrasyonel fonksiyonlarda sürekli kesir açılımı formunda ifade edilebilmektedirler. Sonsuza kadar giden bu kesir serisinin bir noktada kesilmesi gerekmektedir ve serinin sonlu bir elemanda kesilmesi sonucu yaklaşık eşdeğeri elde edilir. Kesir dereceli türevin yaklaşık eşdeğerini bulmak için  $(1+x)^\alpha$  'nın sürekli kesir açılımı şöyle ifade edilir [19],

$$(1+x)^\alpha = b_0 + \frac{a_1x}{b_1 + \frac{a_2x}{b_2 + \frac{a_3x}{b_3 + \dots}}} = \left\{ b_0; \frac{a_1x}{b_1}; \frac{a_2x}{b_2}; \frac{a_3x}{b_3}; \dots; \frac{a_ix}{b_i} \dots \right\} \quad (2.42)$$

$a_ix$  ve  $b_i$  rasyonel formda serinin kısmi pay ve payda terimlerini oluşturur. Bu kesir serisinin terimleri aşağıdaki formda da elde edilebilmiştir.

$$(1+x)^\alpha = \left\{ \frac{1}{1-}; \frac{\alpha x}{1+}; \frac{(1+\alpha)x}{2+}; \frac{(1-\alpha)x}{3+}; \frac{(2+\alpha)x}{2+}; \frac{(2-\alpha)x}{5+\dots} \right\} \quad (2.43)$$

Bu formülasyonda  $s^\alpha$  'nın hesaplanması için  $x = s - 1$  kullanılırsa, terimler rasyonel formda yazılıp düzenlenerek birinci, ikinci, üçüncü ve dördüncü dereceden tamsayı yaklaşık eşdeğerleri aşağıdaki gibi elde edilmiş [20].

1.dereceden Yaklaşım:

$$s^\alpha \cong \frac{(1+\alpha)s+(1-\alpha)}{(1-\alpha)s+(1+\alpha)} \quad (2.44)$$

2.dereceden Yaklaşım:

$$s^\alpha \cong \frac{(\alpha^2+3\alpha+2)s^2+(-2\alpha^2+8)s+(\alpha^2-3\alpha+2)}{(\alpha^2-3\alpha+2)s^2+(-2\alpha^2+8)s+(\alpha^2+3\alpha+2)} \quad (2.45)$$

3.dereceden Yaklaşım:

$$s^\alpha \cong \frac{(\alpha^3+6\alpha^2+11\alpha+6)s^3+(-3\alpha^3-6\alpha^2+27\alpha+54)s^2+(3\alpha^3-6\alpha^2-27\alpha+54)s+(-\alpha^3+6\alpha^2-11\alpha+6)}{(-\alpha^3+6\alpha^2-11\alpha+6)s^3+(3\alpha^3-6\alpha^2-27\alpha+54)s^2+(-3\alpha^3-6\alpha^2+27\alpha+54)s+(\alpha^3+6\alpha^2+11\alpha+6)} \quad (2.46)$$

4.dereceden Yaklaşım:

$$s^\alpha \cong \frac{(\alpha^4+10\alpha^3+35\alpha^2+50\alpha+24)s^4+(-4\alpha^4-20\alpha^3+40\alpha^2+320\alpha+384)s^3+(6\alpha^4-150\alpha^2+864)s^2+(-4\alpha^4+20\alpha^3+40\alpha^2-320\alpha+384)s+(\alpha^4+10\alpha^3+35\alpha^2+50\alpha+24)}{(\alpha^4-10\alpha^3+35\alpha^2-50\alpha+24)s^4+(-4\alpha^4+20\alpha^3+40\alpha^2-320\alpha+384)s^3+(6\alpha^4-150\alpha^2+864)s^2+(-4\alpha^4-20\alpha^3+40\alpha^2+320\alpha+384)s+(\alpha^4+10\alpha^3+35\alpha^2+50\alpha+24)} \quad (2.47)$$

CFE yöntemi basittir ancak düşük frekans bölgesinde sınırlı bir frekans bandında kesir dereceli operatöre yakınsayabilir.

**Örnek 2.2 :** Denklem (2.39) ile verilen sürekli zaman kesir dereceli transfer fonksiyonunu CFE yaklaşık eşdeğeri yardımı ile dolaylı ayrıklaştırma yöntemi ile IIR filtre formunda gerçekleyelim ve yakınsama performansını inceleyelim.

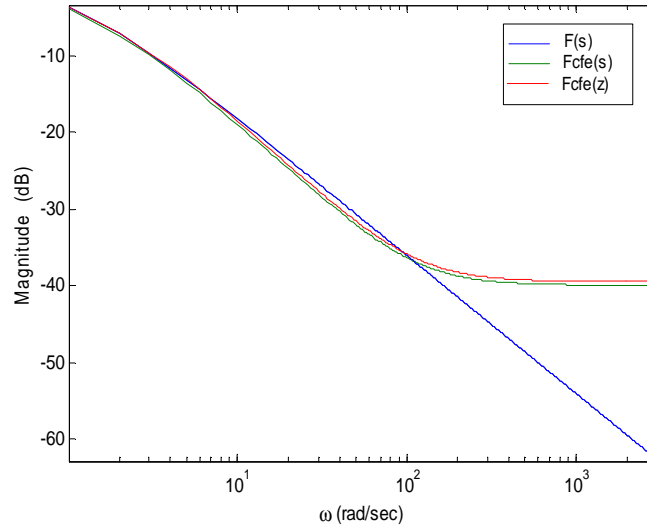
Bu örnekte tasarladığımız Matlab GUI programı  $T=0,001$  sn ve IIR filtre derecesi 3 alınarak kullanılmıştır. CFE yöntemi ile elde edilen sürekli zaman yaklaşık eşdeğer fonksiyon,

$$F_{CFE}(s) = \frac{21.49s^3+71.25s^2+27.03s+0.231}{0.231s^3+27.03s^2+71.25s+21.49} \quad (2.48)$$

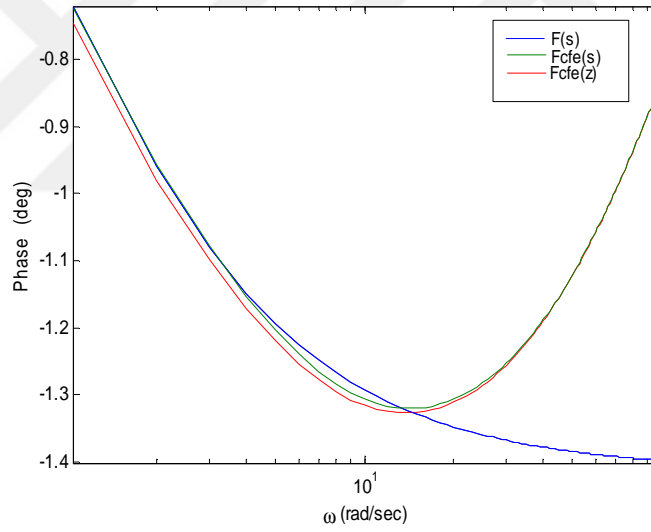
Bu fonksiyon Matlab `c2d()` fonksiyonu yardımı ile CFE yöntemi ayrıklaştırıldığı zaman, aşağıdaki ayrık transfer fonksiyonu elde edilir.

$$F_{CFE}(z) = \frac{z^3-2.889z^2+2.779z-0.8896}{94.03z^3-281.7z^2+281.7z-93.61} \quad (2.49)$$

Bu yöntem sonucunda elde edilen ayrık IIR filtresinin genlik ve faz cevapları Şekil 2.9 ve Şekil 2.10 'da görülmektedir. Çizelge 2.5 'de bu sistemin genlik ve faz cevap yakınsama hata değerleri verilmiştir.



Şekil 2.9.CFE yöntemi uygulayarak elde edilmiş transfer fonksiyonlarının genlik cevapları

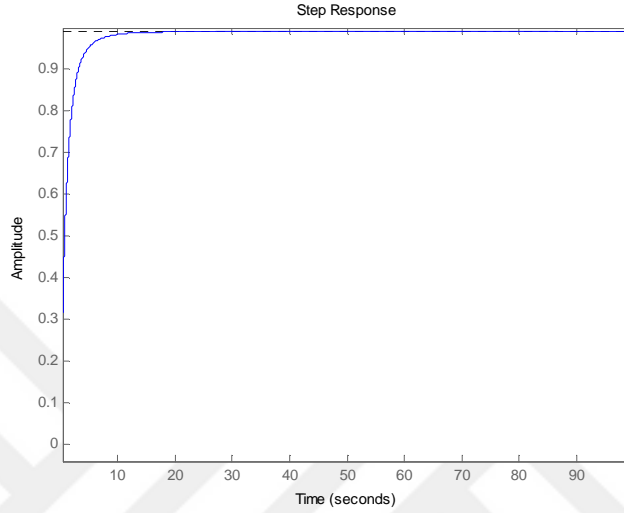


Şekil 2.10.CFE yöntemi uygulayarak elde edilmiş transfer fonksiyonlarının faz cevapları

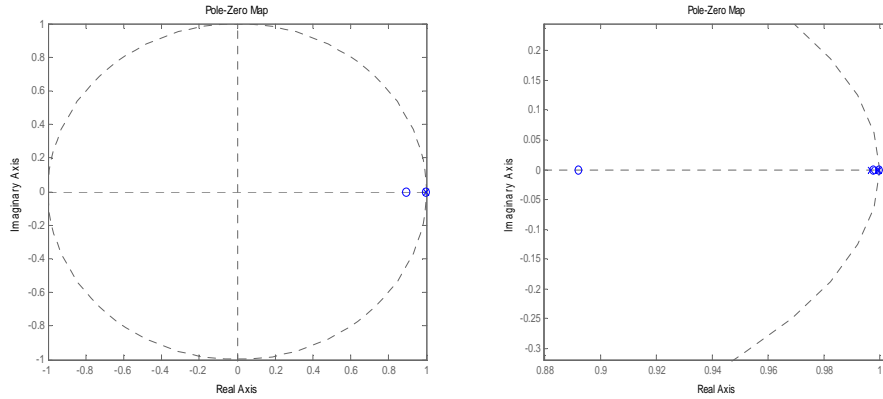
Çizelge 2.5. Genlik ve faz cevabı hatası

<i>Frekans Cevabı</i>	<i>Ortalama Karesel Hata</i>
Faz Cevabı	2.04102
Genlik Cevabı	0.246921

Şekil 2.11 'de birim basamak cevabı verilmiştir ve ayrık yaklaşık eşdeğer modelin kararlı olduğuna işaret etmektedir. Şekil 2.12 'de yöntemin kutup ve sıfır dağılım grafiği verilmiştir. Sağdaki grafik yakından görünümüdür. Kutupların birim çember içinde kalması filtre kararlılığını teyit etmiştir.



Şekil 2.11. Birim basamak cevabı



Şekil 2.12. Sistem kutup ve sıfırlarının dağılımı

Grafikte görüldüğü üzere düşük frekans bölgesindeki sürekli fonksiyona iyi yakınsama sağlanmış ve yüksek frekans bölgesine gidildikçe yakınsama performansı düşmüştür. Bu sonuç CFE yönteminin düşük frekans bölgesinde oldukça başarılı bir şekilde yakınsama sağladığını göstermektedir.

#### 2.4.4. Doğrudan (Direct) Ayırıklaştırma Yöntemleri

Doğrudan ayırıklaştırma yöntemleri, sürekli zamanda ifade edilen kesirli derece türev ve integralin doğrudan ayırık karşılıklarının elde edilmesini sağlar [7]. Kesirli dereceli  $s^\alpha$  'nin doğrudan ayırıklaştırılması için temel ayırıklaştırma yöntemlerinin seri açılımına dayalı çözümleri Çizelge 2.6 'da sunulmuştur [21]. Bu çözümler Taylor seri açılımından faydalanılarak ifade edilmiştir. Kesir dereceli sürekli zaman fonksiyonlarına yakınsama performansı düşüktür. Daha sonradan önerilen ve yakınsama performansı çok daha iyi olan doğrudan ayırıklaştırma özyinelemeli Tustin dönüşümü ve özyinelemeli CFE dönüşümleri detaylı incelenecektir.

Çizelge 2.6.  $s^\alpha$  seri açılım çözümlerine örnekler

Yöntem	Operatör	Seri Çözümü
Euler Yöntemi	$s \rightarrow \frac{1}{T} 1 - z^{-1}$	$s^\alpha \rightarrow \frac{1}{T^\alpha} \left( 1 - \alpha z^{-1} + \frac{\alpha(\alpha-1)}{2} z^{-2} + \dots \right)$
Tustin Yöntemi	$s \rightarrow \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$	$s^\alpha \rightarrow \left( \frac{2}{T} \right)^\alpha 1 - 2\alpha z^{-1} + 2\alpha^2 z^{-2} + \dots$

##### (a) Özyinelemeli Tustin Dönüşümü ile Doğrudan Ayırıklaştırma Yöntemi

Jeofizik veri işlemede petrol aramak için geliştirilen Muir özyinelemeli yöntemi Tustin dönüşümünde değerlendirilmiştir [22]. n-katmanlı düzlemsel tabakaların empedansına bağlı olarak gerçekleşen düzlemsel dalga yansıma cevaplarının hesaplanmasında Muir özyinelemesi kullanılmıştır. Bu yöntem Tustin dönüşüm fonksiyonun kesirli-dereceli kuvvetinin ardışık/özyinelemeli çözümünün elde edilmesinde kullanılmıştır [23-24].

$$s^\alpha \cong \left( \frac{2}{T} \right)^\alpha \left( \frac{1-z^{-1}}{1+z^{-1}} \right)^\alpha = \left( \frac{2}{T} \right)^\alpha \lim_{n \rightarrow \infty} \frac{A_n(z^{-1}, \alpha)}{A_n(z^{-1}, -\alpha)} \quad (2.50)$$

Burada,  $\alpha \in [-1, 1]$  olduğunu varsayılırsa özyinelemeli çözüm

$$A_0(z^{-1}, \alpha) = 1, A_n(z^{-1}, \alpha) = A_{n-1}(z^{-1}, \alpha) - c_n z^n A_{n-1}(z, \alpha) \quad (2.51)$$

ve

$$c_n = \begin{cases} \alpha/n & ; n \text{ tekse,} \\ 0 & ; n \text{ çiftse} \end{cases} \quad (2.52)$$

ile ifade edilir. Sonlu bir n derecesi için ayrık çözüm  $A_n(z^{-1}, \alpha)$  fonksiyonuna bağlı olarak,

$$s^\alpha \approx \left(\frac{2}{T}\right)^\alpha \frac{A_n(z^{-1}, \alpha)}{A_n(z^{-1}, -\alpha)} \quad (2.53)$$

Burada  $A_n(z^{-1}, \alpha)$  fonksiyonun hesaplaması  $n = 9$  'a kadar Çizelge 2.7 'de verilmiştir ve bu çizelgedeki çözümler Matlab GUI çalışmamızda kullanılmıştır. Kesir dereceli transfer fonksiyonunda bulunan  $s^\alpha$  terimleri yerine Çizelge 2.7 'den faydalanılarak Denklem (2.53) 'e göre elde edilen ayrık eşdeğerler terimlerin kullanılması ile ayrıklaştırma işlemi sağlanır.

Çizelge 2.7.  $A_n(z^{-1}, \alpha)$  formülünün  $n = 1, \dots, 9$  [24]

n	$A_n(z^{-1}, \alpha)$ fonksiyonu
0	1
1	$-\alpha z^{-1} + 1$
3	$-\frac{1}{3}\alpha z^{-3} + \frac{1}{3}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
5	$-\frac{1}{5}\alpha z^{-5} + \frac{1}{5}\alpha^2 z^{-4} - \left(\frac{1}{3}\alpha + \frac{1}{15}\alpha^3\right)z^{-3} + \frac{2}{5}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
7	$-\frac{1}{7}\alpha z^{-7} + \frac{1}{7}\alpha^2 z^{-6} - \left(\frac{1}{5}\alpha + \frac{2}{35}\alpha^3\right)z^{-5} + \left(\frac{26}{105}\alpha^2 + \frac{1}{105}\alpha^4\right)z^{-4} - \left(\frac{1}{3}\alpha + \frac{2}{21}\alpha^3\right)z^{-3} + \frac{3}{7}\alpha^2 z^{-2} - \alpha z^{-1} + 1$
9	$-\frac{1}{9}\alpha z^{-9} + \frac{1}{9}\alpha^2 z^{-8} - \left(\frac{1}{7}\alpha + \frac{1}{21}\alpha^3\right)z^{-7} + \left(\frac{34}{189}\alpha^2 + \frac{2}{189}\alpha^4\right)z^{-6} - \left(\frac{1}{5}\alpha + \frac{16}{189}\alpha^3 + \frac{1}{945}\alpha^5\right)z^{-5} + \left(\frac{17}{63}\alpha^2 + \frac{1}{63}\alpha^4\right)z^{-4} - \left(\frac{1}{3}\alpha + \frac{1}{9}\alpha^3\right)z^{-3} + \frac{4}{9}\alpha^2 z^{-2} - \alpha z^{-1} + 1$

Pratik uygulamalar açısından 9. dereceye kadar açılım yeterlidir. Daha yüksek dereceli dönüşümler işlem maliyetini çok artırırken yakınsama performansında bu artışla orantılı bir iyileşme sağlayamadıkları görülmüştür.

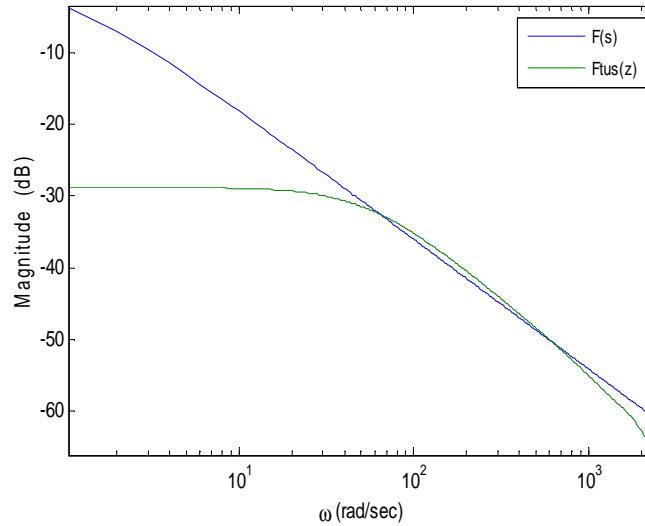
**Örnek 2.3** : Aşağıda verilen kesir dereceli alçak geçiren filtrelerin transfer fonksiyonunu özyinelemeli Tustin yöntemi ile ayrıklaştırılmalı ve elde edilen yaklaşık eşdeğer IIR filtrenin yakınsama performansını inceleyelim.

$$F(s) = \frac{1}{s^{0.9}+1} \quad (2.54)$$

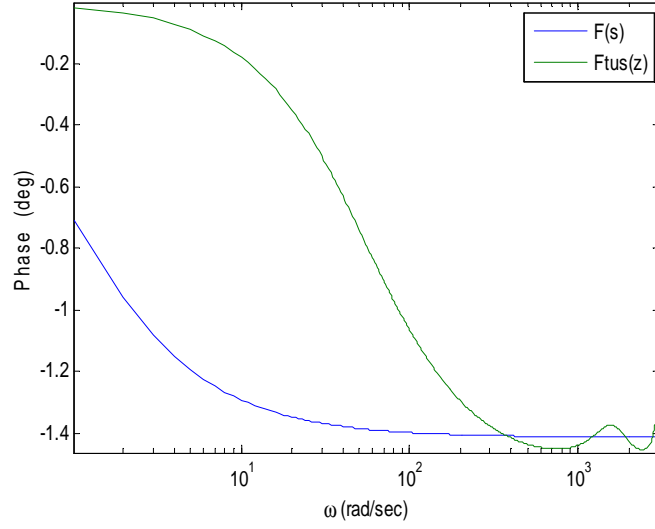
Tasarlanmış olan Matlab GUI programı ile  $T=0,001$  sn ve IIR filtre derecesi 3 kullanılarak elde edilen özyinelemeli Tustin ayrıklaştırma yönteminin sonucu

$$F_{tus}(z) = \frac{z^3+0.9z^2+0.27z+0.3}{936.2z^3-840.8z^2+252.8z-280.3} \quad (2.55)$$

elde edilmiştir.



Şekil 2.13.Özyinelemeli Tustin yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin genlik cevabı

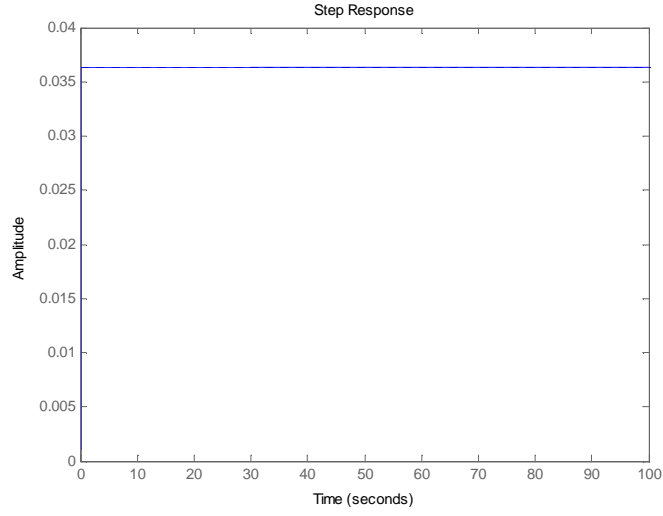


Şekil 2.14.Özyinelemeli Tustin yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin faz cevabı

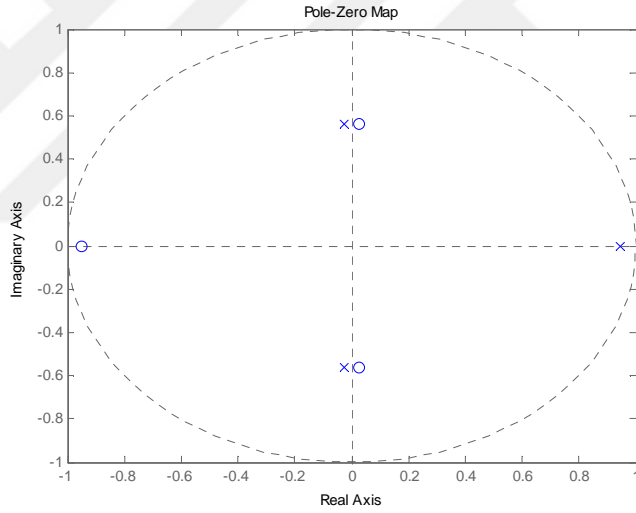
Bu yöntem sonucunda elde edilen ayrık IIR filtresinin genlik ve faz cevapları Şekil 2.13 ve Şekil 2.14 'de verilmiştir. Frekans cevabı yakınsaması için ortalama karesel hata değerleri Çizelge 2.8 'de verilmiştir. Şekil 2.15 birim basamak cevabı Şekil 2.16 'da sistem kutup ve sıfırlarının z düzleminde yerleşimi görülmektedir.

Çizelge 2.8. Genlik ve faz cevabı hatası

<i>Frekans Cevabı</i>	<i>Ortalama Karesel Hata</i>
Faz Cevabı	0.0406233
Genlik Cevabı	0.0005602



Şekil 2.15. Birim basamak cevabı



Şekil 2.16. Sistem kutup ve sıfırlarının dağılımı

Şekil 2.15 ve Şekil 2.16 incelendiği zaman elde edilen ayrık filtrenin kararlı olduğu görülür. Şekil 2.13 ve 2.14 'de görüldüğü üzere özyinelemeli Tustin yöntemi yüksek frekans bölgesinde daha iyi genlik ve faz cevabı yakınsaması göstermiştir. Düşük frekans bölgesinde yakınsama performansı nispeten düşüktür.

(b) Özyinelemeli CFE dönüşümü ile Doğrudan ayrıklaştırma

Sürekli kesir açılımı (CFE), özellikle düşük frekans bölgesinde oldukça iyi

yakınsama sağladığı görülmüştür [25,26]. Bu bölümde CFE yönteminin özyinelemeli formda doğrudan ayrıklaştırma için kullanımı incelenmiştir. Bu amaçla,  $s^\alpha \cong \left(\frac{2}{T}\right)^\alpha \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^\alpha$  ifadesi CFE yöntemi ile seri formunda ifade edilmiştir [24].

$$D_{\pm r}(z) = \frac{Y(z)}{F(z)} = \left(\frac{2}{T}\right)^{\pm r} CFE \left\{ \left(\frac{1-z^{-1}}{1+z^{-1}}\right)^{\pm r} \right\}_{p,q} = \left(\frac{2}{T}\right)^{\pm r} \frac{P_p(z^{-1})}{Q_q(z^{-1})} \quad (2.56)$$

Burada T örnekleme periyodu,  $CFE\{.\}$  sürekli kesir açılımı işlemini ifade eder.  $Y(z)$  ;  $y(nT)$  çıkış dizisinin z dönüşümüdür,  $F(z)$  ;  $f(nT)$  giriş dizisinin z dönüşümüdür,  $p$  ve  $q$  sırası ile  $P$  ve  $Q$  polinomlarının dereceleridir. Bu polinomlar  $CFE\{.\}$  seri açılımı sonucu gelir. CFE seri açılımı matematiksel olarak seri terimleri,

$$G(z) \cong a_0(z) + \frac{b_1(z)}{a_1(z) + \frac{b_2(z)}{a_2(z) + \frac{b_3(z)}{a_3(z) + \dots}}} \quad (2.57)$$

$$= \{a_0(z); \frac{b_1(z)}{a_1(z)+}; \frac{b_1(z)}{a_2(z)+}; \frac{b_3(z)}{a_3(z)+}; \dots\}$$

Gerekli çözümler yapıldığı zaman CFE seri açılımı aşağıdaki formda yazılmıştır [23].

$$D_r(z) = 1 + \frac{z^{-1}}{-\frac{11}{2r} + \frac{z^{-1}}{-2 + \frac{3r}{2r^2-1} + \frac{z^{-1}}{2 + \frac{5r^2-1}{2r(-4+r^2)} + \frac{z^{-1}}{-2+\dots}}} \quad (2.58)$$

Buna göre Denklem 2.56 'deki polinomlar Çizelge 2.9 'da sınırlı sayıda polinom derecesi için verilmiştir. Geliştirilen Matlab GUI programında bu çizelgedeki formüllerden faydalanılmıştır.

Çizelge 2.9.Farklı r ifadeleri için  $D_r(z)$  nin pay ve paydası [24]

p=q	$P_p(z^{-1})(k=1)$ ve $Q_q(z^{-1})(k=0)$
1	$(-1)^k z^{-1} r + 1$
3	$(-1)^k (r^3 - 4r) z^{-3} + (6r^2 - 9) z^{-2} + (-1)^k 15z^{-1} r + 15$
5	$(-1)^k (r^5 - 20r^3 + 64r) z^{-5} + (-195r^2 + 15r^4 + 225) z^{-4}$ $+ (-1)^k (105r^3 - 735r) z^{-3} + (420r^2 - 1050) z^{-2}$ $+ (-1)^k 945z^{-1} r - 945$
7	$(-1)^k (784r^3 + r^7 - 56r^5 - 2304r) z^{-7}$ $+ (10612r^2 - 1190r^4 - 11025 + 28r^6) z^{-6}$ $+ (-1)^k (53487r + 378r^5 - 11340r^3) z^{-5}$ $+ (99225 - 59850r^2 + 3150r^4) z^{-4}$ $+ (-1)^k (17325r^3 - 173250r) z^{-3}$ $+ (-218295 + 62370r^2) z^{-2} + (-1)^k 135135z^{-1} r$ $+ 135135$
9	$(-1)^k (-52480r^3 + 147456r + r^9 - 120r^7 + 4368r^5) z^{-9}$ $+ (45r^8 + 120330r^4 - 909765r^2 - 4410r^6$ $+ 893025) z^{-8}$ $+ (-1)^k (-5742495r - 76230r^5 + 145183r^3$ $+ 990r^7) z^{-7}$ $+ (-13097700 + 9514890r^2 - 796950r^4$ $+ 13860r^6) z^{-6} + (-1)^k (33648615r - 5405400r^3$ $+ 135135r^5) z^{-5} + (-23648625r^2 + 51081030$ $+ 945945r^4) z^{-4} + (-1)^k (-61486425r$ $+ 4729725r^3) z^{-3} + (16216200r^2 - 72972900) z^{-2}$ $+ (-1)^k 34459425z^{-1} r + 34459425$

**Örnek 2.4 :** Aşağıda verilen kesir dereceli alçak geçiren filtrelerin transfer fonksiyonunu özyinelemeli CFE yöntemi ile ayrıklaştıralım ve elde edilen yaklaşık eşdeğer IIR filtrenin yakınsama performansını inceleyelim.

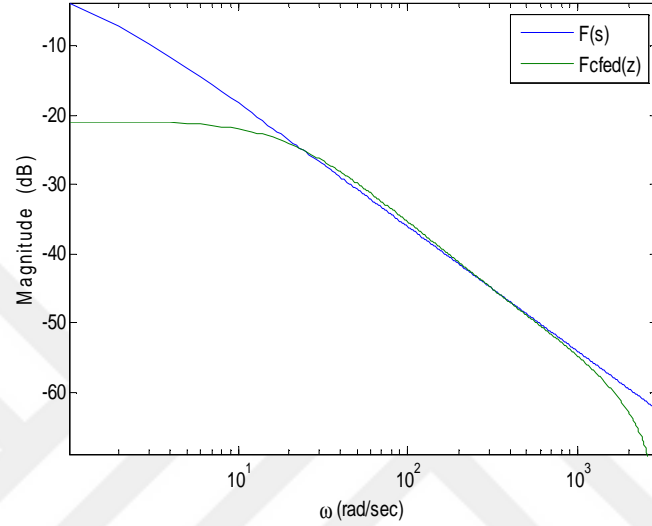
$$F(s) = \frac{1}{s^{0.9} + 1} \quad (2.59)$$

Tasarlanmış olan Matlab GUI programı ile  $T=0,001$  sn ve IIR filtre derecesi 3 kullanılarak elde edilen özyinelemeli CFE ayrıklaştırma yönteminin sonucu,

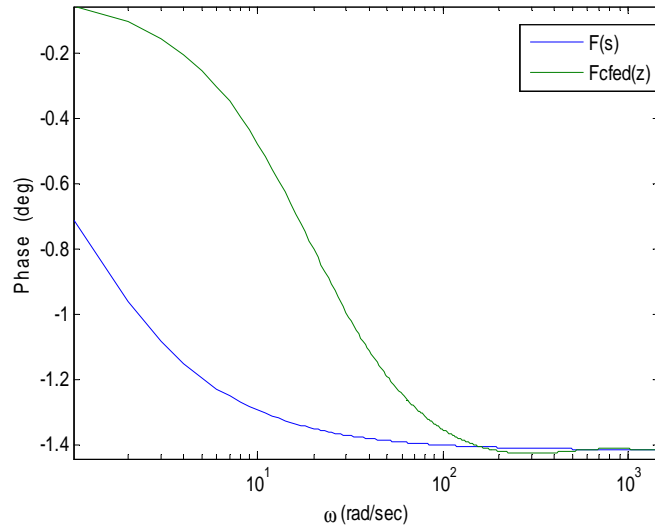
$$F_{cfed}(z) = \frac{15z^3 + 13.5z^2 - 4.14z - 2.871}{1.404e004z^3 - 1.261e004z^2 - 3876z - 2682} \quad (2.60)$$

Bu yöntem sonucunda elde edilen ayrık IIR filtresinin genlik ve faz cevapları

Şekil 2.17 ve Şekil 2.18 'de verilmiştir. Frekans cevabı ortalama karesel hata değerleri Çizelge 2.10 'da verilmiştir. Şekil 2.19 'da birim basamak cevabı ve Şekil 2.20 'de sistem kutup ve sıfırlarının yerleşimi görülmektedir. Bu yöntem kararlı filtre çözümleri verebilmiş ve oldukça geniş bir frekans bölgesinde kesir dereceli filtre cevabına yakınsama sağlayabilmiştir.



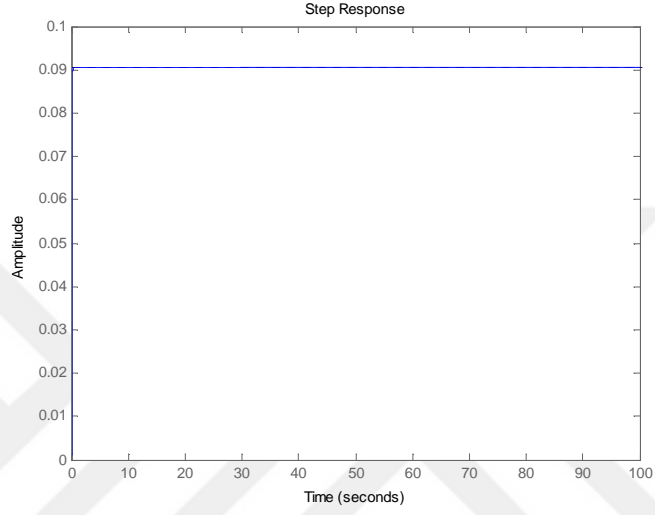
Şekil 2.17. Özyinelemeli CFE yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin genlik cevabı



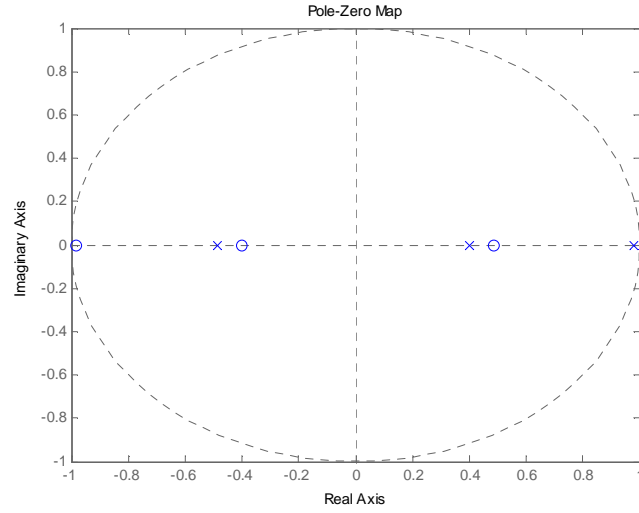
Şekil 2.18. Özyinelemeli CFE yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin faz cevabı

Çizelge 2.10. Genlik ve faz cevabı hatası

<i>Frekans Cevabı</i>	<i>Ortalama Karesel Hata</i>
Faz Cevabı	0.0124257
Genlik Cevabı	0.0004496



Şekil 2.19. Birim basamak cevabı

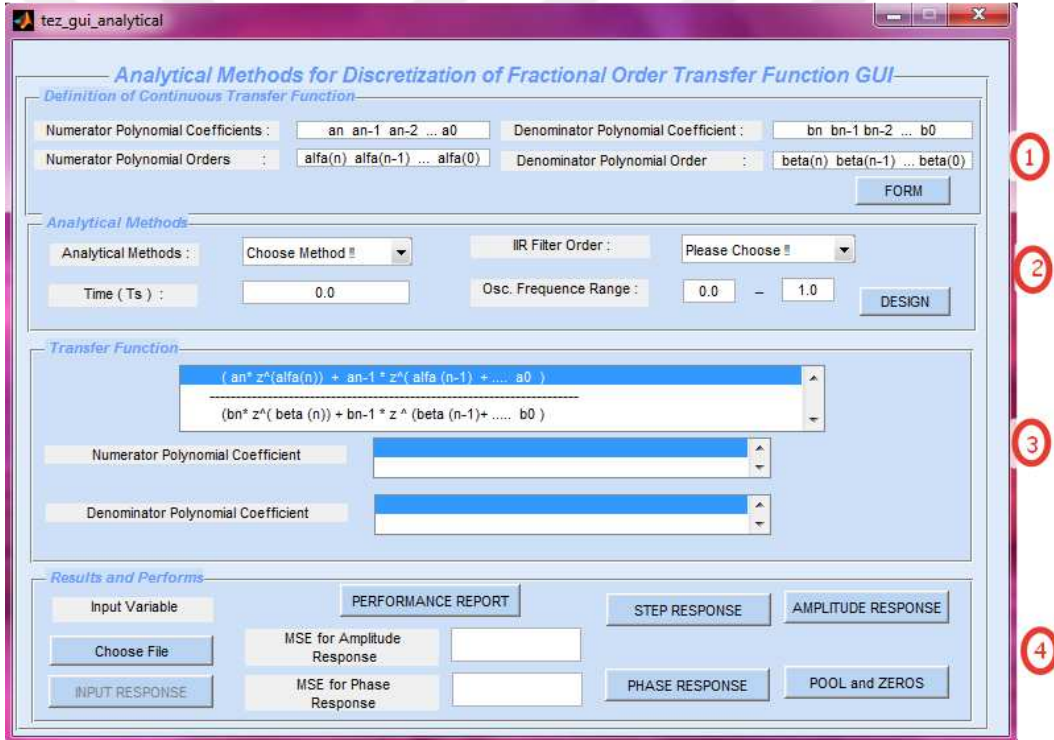


Şekil 2.20. Sistem kutup ve sıfırlarının dağılımı

## 2.5. Kesir Dereceli Transfer Fonksiyonunun Ayrıklaştırılması İçin Geliştirilen Temel Yöntemlerin Matlab GUI İle Programlanması

Bölüm 2.4.2 'de bahsi geçen kesir dereceli transfer fonksiyonlarını ayrıklaştırma yöntemleri için bir Matlab GUI tasarlanmıştır. Daha sonra bu program, PSO optimizasyon modülü eklenerek analitik yöntemlerin sonuçlarını uygulama gereksinimlerine göre optimize eden bir Matlab GUI programına dönüşmüştür. Bahsi geçen PSO destekli GUI programı gelecek bölümlerde tanıtılacaktır.

Şekil 2.21 'de literatürdeki kesir derece transfer fonksiyonlarının ayrıklaştırılması için önerilen temel analitik yöntemleri içeren Matlab GUI modülünün ekran çıktısı görülmektedir.



Şekil 2.21. Analitik yöntemler ile kesir derece transfer fonksiyonu tasarımı için geliştirilen GUI modülü kullanıcı ara yüzü

Matlab GUI modülü şöyle kullanılır:

Aşama 1: Kullanıcı sürekli zaman kesir dereceli filtre katsayı ve derecelerini ilk bölümde (Definition of Continuous Transfer Function paneli-1-) kısmında girer ve "FORM" butonuna basarak kesir dereceli transfer fonksiyonunu tanıtır.

Aşama 2: Analitik ayrıklaştırma yöntemi seçilir. Örnekleme frekansı girilir. Filtre derecesi belirlenir ve istenirse frekans aralığı ikinci kısımda (Analytical Methods paneli-2-) girilir. "DESIGN" tuşuna basılarak ayrık yaklaşık eşdeğer IIR filtre tasarlanır. Tasarlanan ayrık IIR filtre katsayıları üçüncü bölümde kullanıcıya (Transfer Function panel-3-) sunulur.

Aşama 3: Elde edilen ayrık filtrenin performans analizleri dördüncü bölümde (Results and Performs panel-4-) gerçekleştirilir. Burada, frekans cevapları olan genlik ve faz cevapları, birim basamak cevabı, kutup-sıfır yerleşim grafikleri çizilir. Ayrıklaştırılan sürekli zaman kesir derece genlik ve faz cevabına göre yakınsama hataları ortalama karesel hata (Mean Square Error) olarak hesaplanır.

### 2.5.1. Matlab GUI İle Analitik Yöntemler İle Yaklaşık Eşdeğer Filtre Tasarımı ve Tasarımların Karşılaştırılması ve Değerlendirilmesi

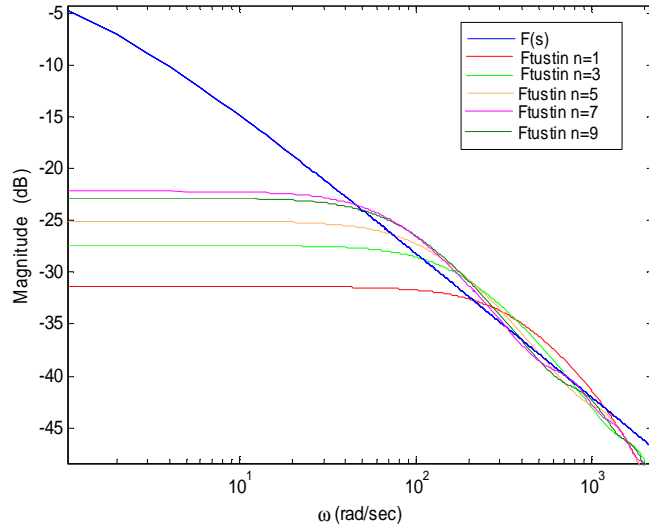
Bu bölümde GUI ara yüzü ile elde edilen ,

$$F(s) = \frac{1}{s^{0.7}+1} \quad (2.61)$$

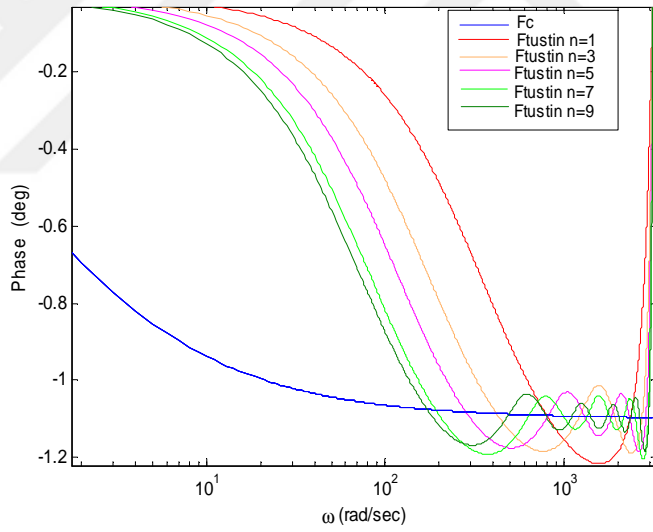
kesir dereceli filtre fonksiyona yakınsayan ayrık IIR filtre tasarımları farklı dereceler için gerçekleştirilmiştir. Bu gerçeklemeler için örnekleme frekansı  $T=0,001$  sn olarak alınmıştır.

(i) *Özyinelemeli Tustin doğrudan ayrıklaştırma yöntemine göre tasarım sonuçları:*

Şekil 2.22 'de Tustin Recursive yöntemi ile  $n = 1,3,5,7,9$  dereceleri için elde edilen yaklaşık IIR filtrelerin genlik cevapları verilmiştir. 5 dereceye kadar derece arttırmak yakınsamayı belirgin bir şekilde iyileştirmektedir. Ancak 7 ve 9. derece filtre genlik cevabı yakınsama performansları arasında kayda değer iyileşme görülmemektedir. Şekil 2.23 'de Tustin Recursive yöntemi ile elde edilen yaklaşık IIR filtrelerin faz cevapları karşılaştırılmıştır. Derece 5 'den sonra derece arttırmanın dikkate değer bir yakınsama performansı iyileşmesi getiremediği faz cevabında da görülmektedir. Genel anlamda yüksek frekans bölgesine doğru yakınsama performansının iyileştiği görülebilir. Bu nedenle özyinelemeli Tustin yönteminin düşük frekans uygulamalar için uygun olmadığı görülür.



Şekil 2.22.Özyinelemeli Tustin yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin genlik cevapları (  $n=1,3,5,7,9$  )

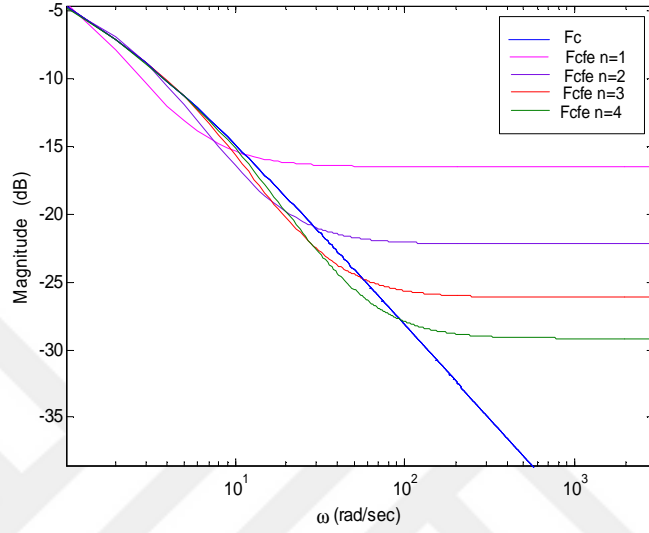


Şekil 2.23.Özyinelemeli Tustin yöntemi uygulayarak elde edilmiş ayrık IIR filtrenin faz cevapları (  $n=1,3,5,7,9$  )

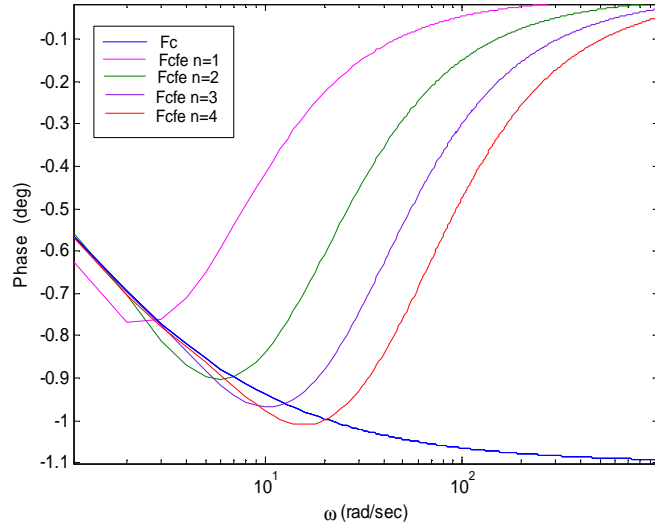
(ii) *CFE ile dolaylı ayrıklaştırma yöntemine göre tasarım sonuçları:*

Şekil 2.24 'de CFE yöntemini ile yapılan dolaylı ayrıklaştırma yöntemi ile elde edilen ayrık IIR filtre tasarımları için genlik cevapları verilmiştir. Burada filtre dereceleri  $n = 1,2,3,4$  seçilerek elde edilen yaklaşık IIR filtrelerin genlik cevapları karşılaştırılmıştır. Şekilde CFE ile dolaylı ayrıklaştırma sonucu edilen yaklaşık eşdeğer filtrelerin genelde düşük frekans bölgesinde iyi yakınsama sağladığı

görülmüştür. Bu nedenle CFE yönteminin kontrol uygulamaları gibi düşük frekans bölgesinde çalışan uygulamalar için kullanışlı bir ayırıklaştırma yöntemi olduğu görülebilir. Derece arttıkça daha geniş frekans bölgesinde yakınsama sağlanabildiği gözlemlenmiştir. Şekil 2.25 'de bu yöntem ile elde edilen ayırık IIR filtre tasarımları faz cevapları bu bulguları destekler niteliktedir.



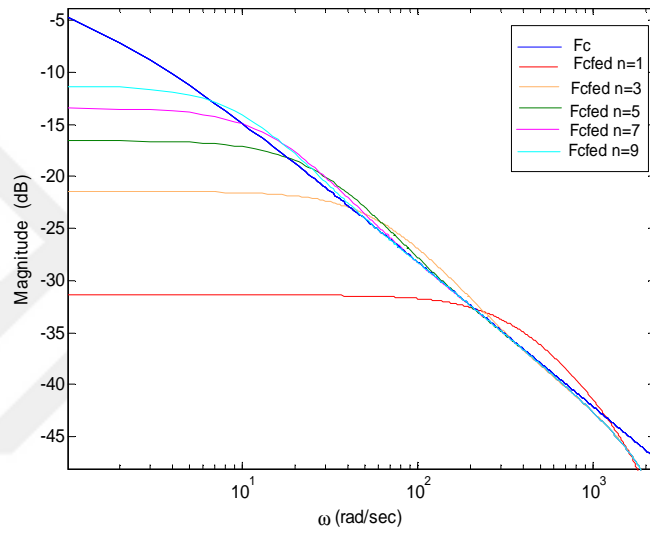
Şekil 2.24.CFE ile dolaylı ayırıklaştırma yöntemini ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması( n=1,2,3,4 )



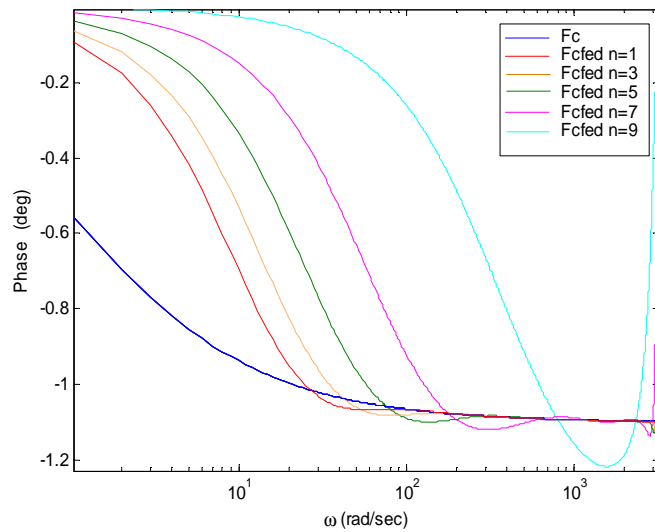
Şekil 2.25.CFE ile dolaylı ayırıklaştırma yöntemini ile elde edilen IIR filtrelerin faz cevaplarının kıyaslaması( n=1,2,3,4 )

(iii) Özyinelemeli CFE ile doğrudan ayrıklaştırma yöntemine göre tasarım sonuçları:

Şekil 2.26 'da CFE ile doğrudan ayrıklaştırma yöntemine göre elde edilen ayrık IIR filtre tasarımları için genlik cevapları verilmiştir. Bu ayrık eşdeğer IIR filtre tasarımları  $n = 1,3,5,7,9$  dereceleri için gerçekleştirilmiştir. Bu yöntem ile elde edilen filtrelerin orta ve yüksek frekansa doğru daha iyi yakınsama sağladığı görülmektedir. Derece arttıkça daha geniş frekans bölgesinde yakınsama sağlanabildiği gözlemlenmiştir. Şekil 2.27 'de bu yöntem ile elde edilen ayrık IIR filtre tasarımlarının faz cevapları bu bulguları desteklemektedir.



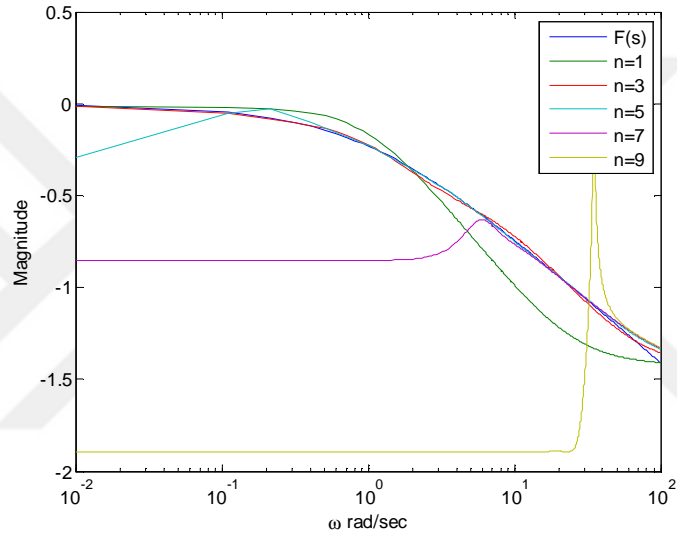
Şekil 2.26.Özyinelemeli CFE doğrudan ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması(  $n=1,3,5,7,9$  )



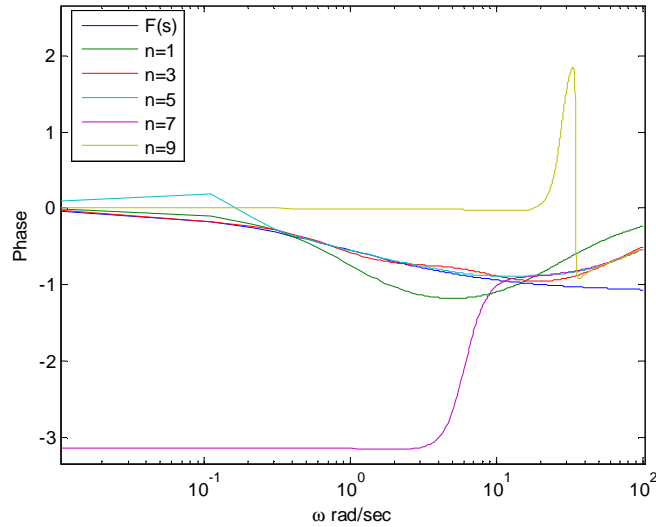
Şekil 2.27.Özyinelemeli CFE doğrudan ayrıklaştırma yöntemini ile elde edilen IIR filtrelerin faz cevaplarının kıyaslaması(  $n=1,3,5,7,9$  )

(iv) Oustaloup ile dolaylı ayrıklaştırma yöntemine göre tasarım sonuçları:

Şekil 2.28 'de Oustaloup dolaylı ayrıklaştırmaya yöntemi ile elde edilen ayrık IIR filtre tasarımları için genlik cevapları  $n = 1,3,5,7,9$  dereceleri için görülmektedir. Oustaloup yöntemi eşdeğer tamsayıli dereceli transfer fonksiyonlarının yakınsama performansı iyi olmasına rağmen ayrıklaştırma işlemi sonucu elde edilen IIR filtrelerin artan derecelerde yakınsama performansının düştüğü görülmüştür. Şekil 2.29 'da bu yöntem ile elde edilen ayrık IIR filtre tasarımları faz cevapları verilmiştir.



Şekil 2.28. Oustaloup ile dolaylı ayrıklaştırmaya yöntemi ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması (  $n=1,3,5,7,9$  )



Şekil 2.29. Oustaloup ile dolaylı ayrıklaştırmaya yöntemi ile elde edilen IIR filtrelerin genlik cevaplarının kıyaslaması (  $n=1,3,5,7,9$  )

Çizelge 2.11 ve Çizelge 2.12 'de genlik ve faz cevabı yakınsama hatalarının aynı filtre dereceleri için bir kıyaslaması sunulmuştur. Doğrudan ayırıklaştırma yöntemleri olan özyinelemeli CFE ve özyinelemeli Tustin yöntemlerinin genlik ve faz cevabı yakınsama performanslarının filtre derecesi arttıkça daha fazla iyileşebildiği görülmüştür. Fakat Oustaloup yöntemi ile dolaylı ayırıklaştırma yöntemlerinin yakınsama performanslarının düzensiz davranış gösterdiği görülmüştür. Filtre derecesi 3'e kadar iyi yakınsama sağlamıştır ancak artan derecelerde yakınsama performansının hızla kötüleştiği görülmüştür. CFE yöntemi ile dolaylı ayırıklaştırma yöntemi özellikle düşük frekans bölgesinde iyi yakınsama gösterebilmiştir.

Çizelge 2.11. Genlik cevabı yakınsama hatalarının (Ortalama karesel hata) kıyaslamaları

<i>IIR Filtre Derecesi(n)</i>	<i>Oustaloup ile Dolaylı Ayırıklaştırma</i>	<i>Özyinelemeli CFE</i>	<i>CFE ile Dolaylı Ayırıklaştırma</i>	<i>Özyinelemeli Tustin</i>
1	0.0015	0.0006760	0.0197206	0.0006760
3	0,0000224	0.0004819	0.0017111	0.0006079

Çizelge 2.12. Faz cevabı yakınsama hatalarının (Ortalama karesel hata) kıyaslamaları

<i>IIR Filtre Derecesi(n)</i>	<i>Oustaloup ile Dolaylı Ayırıklaştırma</i>	<i>Özyinelemeli CFE</i>	<i>CFE ile Dolaylı</i>	<i>Özyinelemeli Tustin</i>
1	0.3338	0.118881	1.17077	0.118881
3	0.1030	0.0182345	1.10208	0.0593001

## 2.6. Matlab Sembolik Hesaplama İle Kesir Dereceli Transfer Fonksiyonun Ayırıklaştırılması

Programlamada karşılaşılan en önemli güçlük  $s^\alpha$  yaklaşık eşdeğeri kullanılarak daha karmaşık rasyonel kesir dereceli transfer fonksiyonlarının yaklaşık eşdeğer filtrelerinin hesaplanması işlemidir. Güçlük, Matlab programının sembolik

hesaplama aracı kullanılarak aşılabilmektedir. Kesir dereceli transfer fonksiyonlarında her  $s^\alpha$  kesir dereceli terimler yerine doğrudan yaklaşık eşdeğer IIR filtresi  $F(z)$  kullanılarak, daha karmaşık transfer fonksiyonlarının Matlabta ifade edilebilmesi sağlanmıştır. Bu işlem şu örnekle açıklanabilir. Varsayalım,  $s^{\alpha_1} \cong F_1(z)$ ,  $s^{\alpha_2} \cong F_2(z)$ ,  $s^{\beta_1} \cong F_3(z)$  doğrudan CFE yöntemi ile elde edilmiş yaklaşık eşdeğer filtreler olsun. Biz daha karmaşık ifade olan  $G(s) = \frac{s^{\alpha_1+1}}{2s^{\alpha_3}+s^{\alpha_2}+1}$  transfer fonksiyonunu gerçeklemek isteyelim. Bu durumda sembolik programlama eşdeğer IIR filtre fonksiyonlarını doğrudan  $G(s)$  fonksiyonunda kullanarak eşdeğer transfer fonksiyonunu  $G(s) \cong \frac{F_1(z)+1}{2F_3(z)+F_2(z)+1}$  formunda yazabilmemize imkan sağlamıştır.

Bu işlem CFE için aşağıdaki kod yazımı ile kesir dereceli fonksiyon ne kadar çok kesir dereceli türev terimi  $s^\alpha$  içeriyor olursa olsun bu transfer fonksiyonun ayrık eşdeğer IIR filtre formunu Matlab'da oluşturulmasını sağlamıştır. Aşağıdaki şekilde CFE yöntemi ile elde edilen  $s^\alpha \cong F(z)$  terimlerinin transfer fonksiyonunun pay ve payda polinomlarında toplanması ve polinomların rasyonel formda yazılması ile IIR filtrenin elde edilmesi görülmektedir.

```
% CFEfunct() s^Alfa CFE eşdeğerini hesaplar

for i=1:num_npc
    fd1(i)=CFEfunct(num_order(i),n,Ts );
end
pay_pol=0;

for j=1:num_npc
    pay_pol=pay_pol+ num_coef(j).*fd1(j);
end

for i=1:dem_dpc
    fd2(i)=CFEfunct(dem_order(i),n,Ts );
end
payda_pol=0;

for j=1:dem_dpc
    payda_pol=payda_pol+ dem_coef(j).*fd2(j);
end

iir_filter=(pay_pol/payda_pol);
```

Şekil 2.30. Matlab sembolik programlama ile karmaşık kesir dereceli filtreleri gerçekleyen örnek kod

Şekil 2.30 'da verilen kod parçasında  $CFEfunct ( num\_order(i),n,Ts )$  fonksiyonu ile öz yinelemeli CFE yöntemi ile doğrudan tasarlanan yaklaşık eşdeğer filtre fonksiyonları  $fd1$  ve  $fd2$  'ler pay ve payda polinomunu oluşturmak üzere polinom katsayıları ile çarpılıp toplanmıştır. Sonuçta rasyonel kesirli dereceli transfer fonksiyonu  $iir\_filter=(pay\_pol/payda\_pol)$  ile elde edilmiştir. Bu yaklaşım diğer analitik yöntemler için kullanılmıştır.

Sadece kesir derece türev operatörüne yakınsamak amacı ile geliştirilen  $s^\alpha \cong F(z)$  yaklaşık eşdeğerler filtreleri karmaşık transfer fonksiyonlarının frekans cevabı içinde iyi bir çözüm olmasını garanti etmemektedir. Transfer fonksiyonundaki türev terimi sayısı arttıkça yaklaşık türev eşdeğerlerinin ( $s^\alpha \cong F(z)$ ) taşıdığı yakınsama hataları transfer fonksiyonunda yığılmakta ve yakınsama performansını olumsuz etkileyebilmektedir. Bu nedenle bu çözümlerin tekrar bir ince ayar ile transfer fonksiyonu cevabına göre optimize edilmesi çözümleri iyileştirmektedir. Gelecek bölümlerde analitik yöntemlerin ürettiği yaklaşık eşdeğer IIR filtrenin transfer fonksiyonu cevabının PSO algoritması ile optimize edilmesi ve yakınsama performansının iyileştirilmesi çalışması anlatılacaktır.

### 3. MATERYAL VE YÖNTEM

#### 3.1. Parçacık Sürüsü Optimizasyonu (PSO) Hakkında Temel Bilgiler

PSO algoritması popülasyon temelli ve sürü zekasını taklit eden sezgisel bir optimizasyon tekniğidir [27]. 1995 yılında Kennedy ve ekibi tarafından geliştirilmiştir. Bu algoritma sürü hareketlerinden esinlenerek ortaya çıkan sezgisel bir optimizasyon tekniğidir. PSO algoritması balık ve kuş sürüleri gibi toplu hareket eden canlıların yiyecek arama faaliyeti sırasında ortaya çıkan sosyal davranışları esas alır. Birbirleri ve çevresi ile etkileşimi olan bireylerin sürü davranışları incelenerek geliştirilmiştir. Sürü davranışı sonucu ortaya çıkan zekaya sürü zekâsı denir. PSO algoritması sürü zekasını minimum noktaları arayan meta sezgisel optimizasyon işlemi için uygundur.

Bu yöntem, parçacık sürüsünde her bir parçacığın hareketini modeller ve parçacıkların en iyi çözümleri takip ederek arama uzayında içinde optimal sonuçlara varmasını hedefler. Örnek olarak, yiyecek arayan kuş sürülerine benzetilebilir. Burada her bir kuş parçacık olarak adlandırılacaktır ve her bir kuşun konumu (parçacık konumu) birer çözüme karşılık gelir. Bu parçacıkların temsil ettiği çözümler uygunluk fonksiyonu ile değerlendirilir. Her parçacığın konum bilgisi yanında, kuşların hareket prensiplerinde olduğu gibi hareketlerini yönlendiren bir hız bileşeni vardır. Parçacıkların arama uzayında konum değişimleri hız bilgilerine bağlı olarak gerçekleştirilir. PSO yönteminde başlangıçta parçacıklar arama uzayına rastgele dağılır. Bu durum yiyecek arayışı içerisinde olan kuşların, henüz keşfedilmemiş bölgeye rastgele dağılım yapmasına benzetilebilir. Sürüdeki kuşlar başlangıçta bağımsız hareket etme eğilimi gösterirken sürüdeki diğer kuşların hareketini izlerler ve bu etkileşim aralarında aynı zamanda zayıf bir koordinasyon oluşturur. Bu bir süre sonra yiyeceklerin bulunmaya başlaması ile sürü yiyecek olan alana doğru hareket meyili oluşturur. Bu durum bağımsız hareket güdüsünü azaltır ve koordine hareket güdüsünü güçlendirir. PSO algoritmasında parçacıklar arasında iyi çözüme yaklaşan parçacık oldukça diğer parçacıkların bu parçacığa doğru yönelimi artar ve parçacık sürüsü daha iyi çözüm olan bölgeye toplanmaya başlar. Bu hareket her parçacığın lokal iyi olan çözümü ve global iyi olan çözümü dikkate alarak hareketini belirlemesi ile sağlanır. Bu durum yiyeceğe yönelen kuşların hem yakın civardaki kuşların hareketini dikkate alması ( lokal iyi ) hem de sürünün genel

yönelimini (global iyi) dikkate almasına benzetilebilir. Bu yöntem kuşların diğer kuşların veya sürünün pozisyon bilgisini dikkate alması, hareket yönüne ve hızına karar vermesine benzetilebilir. Sürüdeki kuşlar arasında gerçekleşen bilgi paylaşımı yiyeceğe en yakın olan kuşun pozisyonunun sürünün hareketini belirleyen bilgi olmasını sağlar. Bu pozisyon bilgisine sahip olan diğer kuşlar artık bu bilgiyi kullanarak yiyeceğe ulaşırlar.

PSO, diğer optimizasyon yöntemlerine kıyasla ayarlanması gereken parametre sayısı daha az olması sebebiyle uygulanması kolay bir yöntemdir. PSO algoritmasında lokal en iyi ve global en iyi konumların tutulması, parçacık hareketlerinin bu iki konuma göre belirlenmesi ve belli bir rastgele serbestlik derecesi kazandıran rastgele sayı ile ölçekleme işlemi PSO'nun yerel optimumlara takılma olasılığını azaltır [27].

Parçacıklar başlangıçta rastgele bir pozisyonda ve hızda arama uzayına dağıtılır. Her bir iterasyonda parçacıkların uygunluk fonksiyonu hesaplanır. Lokal en iyi ve global en iyi parçacık belirlenir. Diğer parçacıklar bu en iyi lokal ve global parçacıklara göre hız ve konumlarını günceller. Böylece bütün iterasyon boyunca sürekli en iyi çözümler global en iyi ve lokal en iyi parçacık ile korunur ve daha iyisi aranır.

Çok boyutlu bir arama uzayında çözüm noktalarını temsil eden parçacıkların konumları  $X_i[t]$  ile ifade edelim. Arama uzayında parçacıkların yer değiştirmesi basit hareket denklemi ( $v = dx/dt$ ) ile modellenir. Uzayda bir  $i$ . parçacığın hızı  $V_i[t]$  ile gösterelim. PSO algoritması parçacıkların hızlarını şu denkleme göre günceller.

$$V_i[t + 1] = w \cdot V_i[t] + c_1 \cdot r_1 \cdot (X_L[t] - X_i[t]) + c_2 \cdot r_2 \cdot (X_G[t] - X_i[t]) \quad (3.1)$$

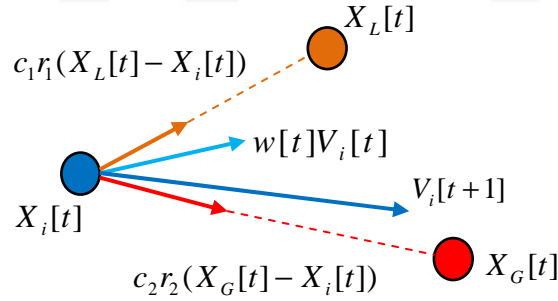
Parçacığın bu yeni hızına karşılık parçacığın yeni konumu,

$$X_i[t + 1] = X_i[t] + V_i[t + 1] \quad (3.2)$$

ile güncellenir.  $n$  boyutlu bir arama uzayı için her bir parçacığın konum ve hızı  $n$  boyutlu bir vektör ile temsil edilir.

$$X_i = [x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n] \text{ ve } V_i = [v_1 \ v_2 \ v_3 \ v_4 \ \dots \ v_n] \quad (3.3)$$

Denklem (3.1) incelendiği zaman lokal en iyi çözüm  $X_L[t]$  ve global en iyi çözüm  $X_G[t]$  bir parçacığın hızına etkilerinin  $c_1 \cdot rand. (X_L[t] - X_i[t])$  ve  $c_2 \cdot rand. (X_G[t] - X_i[t])$  terimleri ile olduğu görülür. Bu terimler vektörel konum hatasını ifade eder ve parçacığın yeni hız vektörü bu konum hatasına göre güncellenir. Burada  $c_1$  katsayısı lokal öğrenme katsayısıdır ve lokal en iyiye yönelimin derecesini ayarlar. Burada  $r_1$  ve  $r_2$  bir rastgele sayıdır ve bu yönetime bir rastgele serbestlik derecesi kazandırır. Bu rastgele serbestlik derecesi algoritmaya skolastik (olasılıksal) bir karakter kazandırır ve programın her çalışmasında yeni çözümler bulunmasına imkan sağlar. Burada  $c_2$  katsayısı global öğrenme katsayısıdır ve global en iyiye yönelimin derecesini ayarlar.  $w$  katsayısı parçacığın ivme katsayısıdır. İvme katsayısı parçacığın hızlanması veya yavaşlamasını sağlar.  $w[t + 1] = \xi w[t]$  ,  $0 < \xi \leq 1$  arasında sabit değerdir parçacıkların her yeni iterasyonda hızının azalmasını sağlar. İterasyonlar ilerledikçe artan bu yavaşlama, parçacıkları buldukları iyi çözümlere yerleşmelerine imkan sağlar. Şekil 3.1 parçacığın hızının lokal ve global en iyi ile etkileşimi sonucu güncellenmesinde etki eden bileşenleri göstermektedir.



Şekil 3.1. PSO algoritmasında global ve yerel en iyi parçacık konumuna göre parçacığın hız vektörüne etki eden terimler

Algoritma Adımları şöyle özetlenebilir:

Adım 1: Parçacık sürüsünün her elemanın başlangıç konum ( $X_i[t]$ ) ve hızını ( $V_i[t]$ ) rastgele belirle.

Adım 2: Her bir parçacığın uygunluk değerini (uygulamaya bağlı olarak maliyet veya hata fonksiyonu olabilir) hesapla ve uygunluk değerine göre lokal en iyi pozisyon ve global en iyi pozisyonu belirle.

Adım 3: Eğer istenildiği kadar küçük hataya ulaşılmışsa veya maksimum iterasyon

sayısına ulařılmış ise algoritmayı durdur. Ulařılmamıřsa, Denklem (3.1) ve Denklem (3.2) yardımı ile her bir parçacığın yeni konumunu belirle ve Adım 2'ye git.

Algoritma kodlanırken kullanıcı tarafından belirlenmesi gereken bazı parametre ve tanımlar:

*Parçacık sayısı*; parçacık sayısının artması arama bölgesinin daha iyi aranmasını saęlar. Normalde birçok problem için sayıyı 20-50 almak yeterli olabilmektedir. Bazı zor veya özel problemler için 100 veya 200 parçacık kullanılması gerekebilir.

*Parçacık boyutu*; Arama uzayının boyutudur.

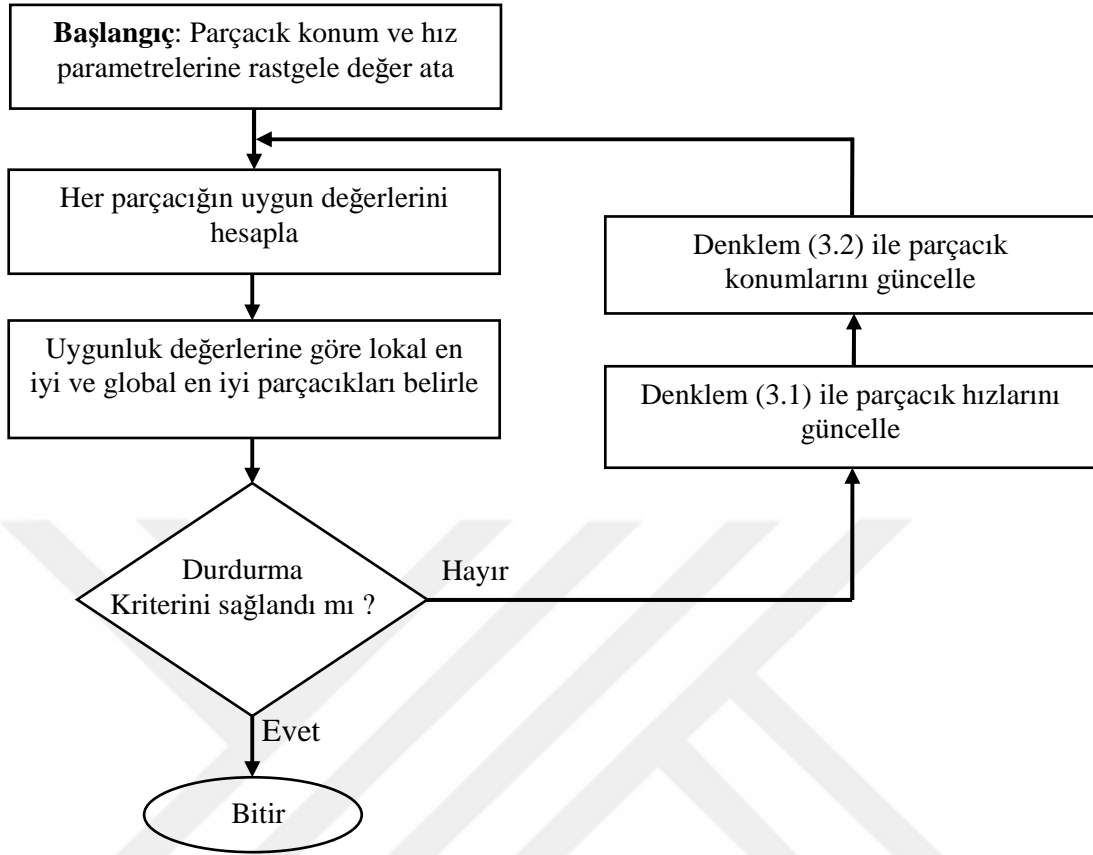
*Parçacık arama aralığı*; Arama uzayında parçacıkların hareket edebileceęi bölgeyi sınırlar.

*Vmax*: Bir iterasyonda, bir parçacıkta meydana gelecek maksimum deęişikliği (hız) belirler. Genellikle parçacık aralığına göre belirlenir. Örneęin  $X_1$  parçacığı (-10,10) aralığında ise  $V_{max}=20$  sınırlandırılabilir.

*Öęrenme Faktörleri*:  $c_1$  ve  $c_2$  genellikle 2 olarak seçilir. Fakat farklı da seçilebilir. Genellikle  $c_1, c_2$  'ye eşit ve  $[0, 4]$  aralığındadır.

*Durma Koşulu*: Maksimum iterasyon sayısına ulařıldığında veya deęer fonksiyonu istenilen seviyeye ulařtığında algoritma durdurulabilir.

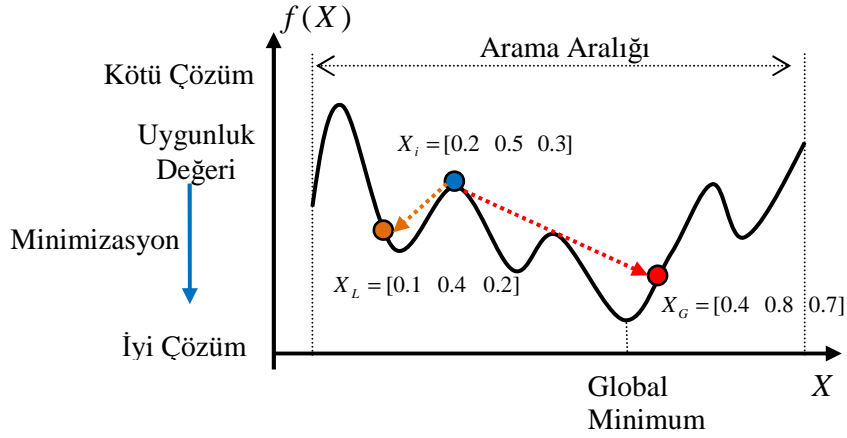
Çalıřmada kullanılan PSO algoritma akıř řeması Şekil (3.2) 'de sunulmuřtur.



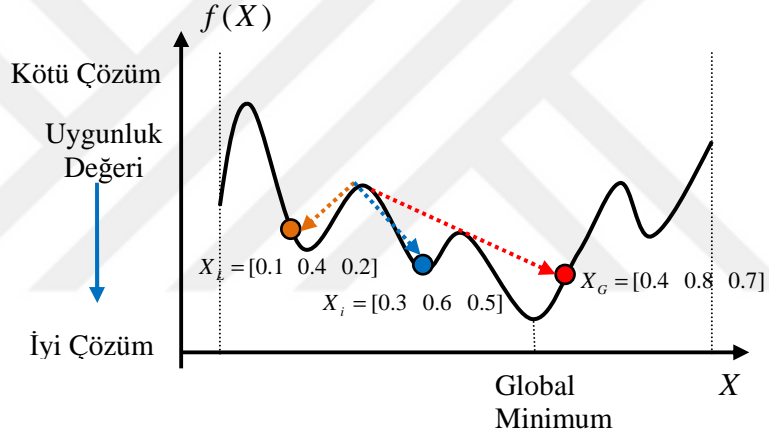
Şekil 3.2. PSO algoritmasının akış şeması

PSO algoritmasında parçacıkların hareketlerinin daha kolay anlaşılması için grafiksel açıklama aşağıda verilmiştir.

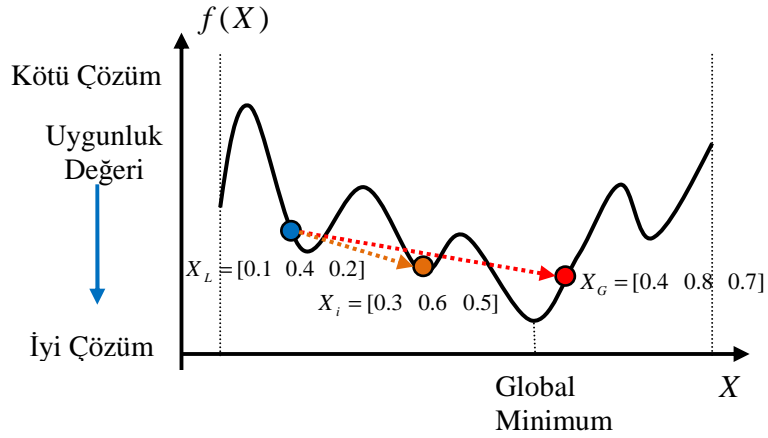
Şekil 3.3 'de lokal en iyi ( $X_L[t]$ ) ve global en iyi ( $X_G[t]$ ) parçacıkların belirlendiği bir üç boyutlu  $X$  arama uzayında  $f(X)$  uygunluk fonksiyonu verilmiş olsun.  $X_i$  parçacığının yeni konumu lokal ve global en iyi parçacıklar ile sağlanan etkileşim sonucunda Şekil 3.4 'de görüldüğü gibi güncellenebilir. Bu güncelleme sonucunda parçacık lokal en iyi parçacığa göre daha iyi uygunluk değerine sahip olduğu için yeni lokal en iyi parçacık bu parçacık olur. Bu durum Şekil 3.5 'de gösterilmiştir.



Şekil 3.3. Parçacığın lokal en iyi ( $X_L[t]$ ) ve yerel en iyi ( $X_G[t]$ ) ile etkileşimi



Şekil 3.4. Parçacığın yerinin lokal en iyi ( $X_L[t]$ ) ve yerel en iyi ( $X_G[t]$ ) göre güncellenmesi



Şekil 3.5. Diğer parçacığın güncellenen lokal en iyi ( $X_L[t]$ ) ve yerel en iyi ( $X_G[t]$ ) ile etkileşimi

### 3.2. Parçacık Sürüsü Optimizasyonunun Kesir Dereceli Transfer Filtre Fonksiyonlarının Ayrık Gerçeklemesine Uygulaması

Sürekli kesir dereceli transfer fonksiyonu en genel formda aşağıdaki gibi ifade edilir,

$$F_c(s) = \frac{\sum_{i=0}^k c_i s^{\beta_i}}{\sum_{i=0}^m d_i s^{\alpha_i}} \quad (3.4)$$

Burada  $c_i \in R$  ve  $d_i \in R$  kesir dereceli transfer fonksiyonun pay ve paydasının katsayılarıdır.  $F_c(s)$  'nin frekans cevabı ile eşleştirmek için IIR tipi ayrık filtre fonksiyonu kullandık.

Ayrık IIR filtrenin ( $F_d(z)$ ) fonksiyonu genel formda aşağıdaki gibi ifade edilebilir,

$$F_d(z) = \frac{\sum_{i=0}^l a_i z^i}{\sum_{i=0}^p b_i z^i} \quad (3.5)$$

Burada  $a = [a_1, a_2, \dots, a_l]$  ve  $b = [b_1, b_2, \dots, b_p]$  ayrık IIR filtre katsayılarıdır. Bu katsayılar PSO algoritmasında,

$$X_i = [b \ a] = [b_1, b_2, \dots, b_p, a_1, a_2, \dots, a_l] \quad (3.6)$$

vektörü ile temsil edilmiştir. Bu vektör aynı zamanda arama uzayında parçacık konumunu belirler. Sürüdeki parçacıklar arama uzayında konum değiştirdikçe her yeni konum yeni bir ayrık IIR filtre çözümünü ifade eder. Bu nedenle, parçacıklar aslında filtre katsayıları ile ifade edilen bir filtre tasarım uzayında dolaşırlar ve belirlenen amaç fonksiyonuna en uygun konumu ararlar. Bu konum amaç fonksiyonuna en uygun filtre tasarımını sunar.

$F_c(s)$  sürekli kesir dereceli transfer fonksiyonun ayrıklaştırılması için, hem genlik cevabı yakınsama performansı, hem de faz cevabı yakınsama performanslarının dikkate alınması gerekmektedir. Genlik cevabı yakınsama hatası karesel hata formunda,

$$E_g(\omega_i) = (|F_c(j\omega_i)| - |F_d(a, b, e^{j\omega T_s})|)^2 \quad (3.7)$$

ile ifade edilmiştir. Burada, sürekli kesir dereceli transfer fonksiyonun frekans cevabı  $F_c(j\omega) = F_c(s)|_{s=j\omega}$  ve ayrık IIR filtrenin frekans cevabı  $F_d(a, b, e^{j\omega_i T_s}) = F_d(a, b, z)|_{z=e^{j\omega_i T_s}}$  dir.  $E_g$  genlik cevabı hatası sıfır olması durumunda  $\omega_i$  frekansında kesir derece transfer fonksiyonu ile ayrık eşdeğer IIR filtre genlik cevaplarının aynı olması sağlanır. PSO algoritması istenen bir  $\omega_i$  frekansında karesel hatayı minimize eder bu frekansta iki filtrenin genlik cevaplarının birbirine yakınsamasını sağlar.

Faz cevabı yakınsama hatası karesel hata formunda,

$$E_f(\omega_i) = (\angle F_c(j\omega_i) - \angle F_d(j\omega_i))^2 \quad (3.8)$$

ile ifade edilmiştir.  $E_f$  genlik cevabı hatası sıfır olması durumunda  $\omega_i$  frekansında kesir derece transfer fonksiyonu ile ayrık eşdeğer IIR filtre faz cevaplarının aynı olması sağlanır. PSO algoritması istenen bir  $\omega_i$  frekansında karesel hatayı minimize eder bu frekansta iki filtrenin faz cevaplarının birbirine yakınsaması sağlar. Ancak uygulamada genlik cevabının yakınsama performansının iyileştirilmesinin, faz cevabının yakınsamasını performansını kötüleştirebildiği yada faz cevabının daha iyi yakınsamasının, genlik cevabının yakınsama performansını olumsuz etkilediği görülmüştür. Bu nedenle genlik veya faz cevabının yakınsama performanslarının ağırlıklı ortalaması ile hata fonksiyonu  $E = \lambda E_g + (1 - \lambda)E_f$  formunda  $\lambda \in [0,1]$  ağırlık katsayısı ile ağırlıklandırılmış formda ifade edilmiştir. Böylece,  $\lambda$  parametresine 0 ile 1 arasında değer verilerek PSO optimizasyonunda genlik cevabı yada faz cevabının iyileştirilmesi öncelikleşebilmiştir. Bu özellik uygulamanın gereksinimlerine göre yaklaşık eşdeğer filtrenin daha optimal tasarlanmasını sağlar. Örneğin, sayısal işaret işlemede genlik cevapları önemlidir. Bu nedenle işaret işleme uygulamalarında  $\lambda$  değeri bire yakın seçilerek genlik cevabının optimizasyonda daha ağırlıklı (öncelikli) yakınsaması sağlanmalıdır. Kontrol uygulamalarında faz cevabı kararlılık açısından öneme sahiptir ve kontrol uygulamaları için faz cevabı  $\lambda$  sıfıra yakın seçilerek optimizasyonda önceliği artırılmalı ve daha iyi faz yakınsama performansı elde edilmelidir.

Uygulamalar için bir çalışma frekans bölgesinde yakınsama gerekmektedir. Bu nedenle optimizasyonda  $E$  hata fonksiyonu tek bir  $\omega_i$  frekansı yerine bir frekans

bölgesi için ifade edilmelidir. Bunun için PSO algoritmasında istenen bir frekans bölgesinde  $\omega_{alt} < \omega_i < \omega_{ust}$  bütün frekans örnekleri için karesel hata toplamlarını minimize ederek iki filtrenin genlik cevaplarının  $\omega_{alt} < \omega_i < \omega_{ust}$  aralığında birbirine yakınsaması sağlanmıştır. Karesel hata fonksiyonu,  $\omega_{alt} < \omega_i < \omega_{ust}$  bölgesinde frekans örnekleme ile toplam formda ifade edilmiş ve bu fonksiyon maliyet(Cost) fonksiyonu olarak aşağıdaki gibi ifade edilmiştir.

$$f(a, b) = \sum_{\omega_i \in (\omega_{min}, \omega_{max})} \gamma_n(\omega_i) \left[ \frac{\lambda(|F_c(j\omega_i)| - |F_d(a, b, e^{j\omega, T_s})|)^2}{+(1 - \lambda)\eta(\angle F_c(j\omega_i) - \angle F_d(a, b, e^{j\omega, T_s}))^2} \right] \quad (3.9)$$

Burada  $\gamma$  frekans öncelik katsayısıdır ve hesaplama yapılan frekansın optimizasyonda ağırlığını ayarlamak için kullanılabilir. Bazı uygulamalar, örneğin kontrol uygulamaları gibi düşük frekans bölgesinde çalışırlar ve düşük frekans bölgesinde yakınsama performansının çok iyi olması istenir. Bazı uygulamalar, örneğin haberleşme gibi, yüksek frekans bölgesinde çalışırlar ve yüksek frekans bölgesinde yakınsama performansının daha iyi olması tercih edilir.  $\omega_{alt} < \omega_i < \omega_{ust}$  frekans bölgesinde optimizasyon algoritmasının da frekans ağırlıklandırması için  $\gamma(\omega_i)$  frekans öncelik katsayısı kullanılarak optimizasyonda frekans önceliklemesi sağlanmıştır.  $\eta$  katsayısı faz ve genlik cevabı normalizasyon katsayısıdır. Genliğin ve fazın en yüksek değerlerini birbirine eşitleyerek  $\lambda$  ağırlıklamasının daha sağlıklı yapılmasını sağlar. Bu normalizasyon katsayısı,

$$\eta = \frac{\max\{|F_c(j\omega_i)| - |F_d(a, b, e^{j\omega, T_s})|\}^2}{\max\{(\angle F_c(j\omega_i) - \angle F_d(a, b, e^{j\omega, T_s}))\}^2} \quad (3.10)$$

ile ifade edilir.

### 3.3. Kararlı IIR Filtre Çözümlerini Elde Etmek İçin PSO da Yapılan Adaptasyonlar

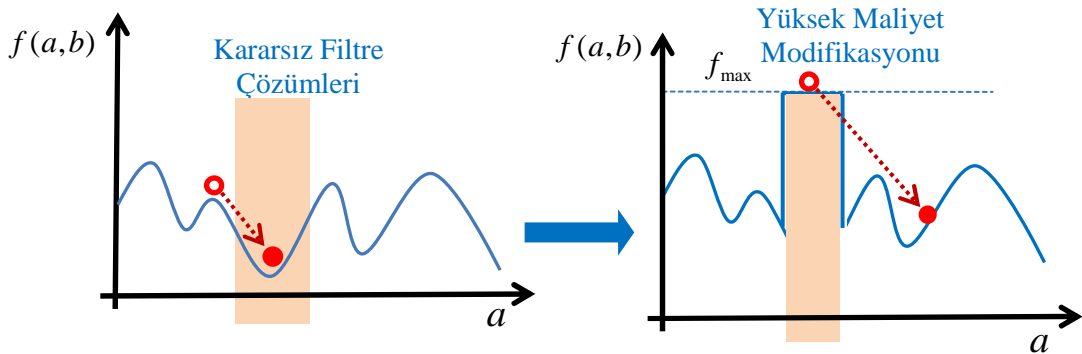
Sadece maliyet fonksiyonu olan Denklem (3.9) 'u minimize eden çözümler her zaman kararlı ayırık IIR filtre çözümleri vermemektedir. Bunun temel nedenlerinden birisi genlik faz cevapları iyi yakınsama gösteren bazı IIR filtrelerin kutuplarının birim çember kararlılık bölgesi dışında olabilmesidir. Bunun sonucu olarak genlik ve faz cevapları kesir derece filtre genlik ve faz cevabına iyi bir yakınsama gösterse de

zaman bölgesi cevapları kararsızdır. Kararsız filtre tasarımları pratikte kullanışsızdır. Bu nedenle PSO optimizasyonun kararlı filtre tasarım bölgelerinde optimizasyon yürütmesi ve kararlı optimal IIR filtre çözümleri vermesi sağlanmalıdır. Bu amaçla PSO algoritması şöyle modifiye edilmiştir:

Filtre tasarım uzayında gezinen parçacıkların kararsız bölgeden uzaklaştırılması için amaç fonksiyonu aşağıdaki fonksiyon ile güncellenmiştir.

$$f(a, b) = \begin{cases} f(a, b) & \text{Kararlı filtre} \\ f_{max} & \text{Kararsız filtre} \end{cases} \quad (3.11)$$

Buna göre  $X_i = [b \ a]$  ile temsil edilen filtre kararlı ise bu parçacık için maliyet olarak PSO algoritmasına Denklem (3.9) ile hesaplanan  $f(a, b)$  değeri döner. Eğer  $X_i = [b \ a]$  ile temsil edilen filtre kararsız ise maliyet olarak  $f_{max}$  ile belirtilen çok yüksek bir maliyet döner. Böylece, kararsız parçacık çözümlerine yüksek maliyet atanarak arama uzayı uygunluk değerleri dinamik olarak modifiye edilir. Parçacıklar yüksek maliyete sahip bu kararsız noktalardan uzaklaşma eğilimi gösterirler, aramalarını daha düşük maliyet bölgeleri olan kararlı filtre çözüm bölgelerinde sürdürürler. Bu durum parçacıkların bulunduğu çözümlerin kararlı çözümler olmasını sağlar. Şekil 3.6 'da kararsız IIR filtre çözümü veren parçacıklara  $f_{max}$  ile belirlenen yüksek maliyet değerleri atanması sonucu parçacıkların düşük maliyetli bölgelere hareket ederek kararsız IIR çözümlerini temsil eden bölgeden uzaklaşması temsili olarak gösterilmiştir.



Şekil 3.6. Kararsız filtre çözümleri için arama uzayında maliyet fonksiyonunun modifiye edilmesi

## 4. ARAŞTIRMA BULGULARI

### 4.1. Kesir Dereceli Transfer Filtre Fonksiyonlarının Ayırıklaştırılması için Geliştirilen Toolbox

Bu tez çalışmasında kesir dereceli transfer fonksiyonlarını ayırıklaştırmak için iki Matlab GUI programı tasarlanmıştır.

(i) Analitik yöntemler için geliştirilmiş GUI modülü: Literatürde bulunan temel analitik ayırıklaştırma yöntemleri yardımı ile kesir dereceli transfer fonksiyonların gerçekleştirilmesi için ayırık yaklaşık eşdeğer IIR filtre tasarımını gerçekleştiren GUI modülüdür. Bu program Bölüm 2 'de tanıtılmış ve birçok örnek verilmiştir.

(ii) PSO yöntemi ile optimizasyon uygulayan gelişmiş GUI programı: Başlangıç çözümü olarak analitik yöntem çözümlerinden faydalanan ve kullanıcının genlik ve faz cevabı öncelikleri ve çalışma frekans bölgesi tercihleri doğrultusunda bu çözümleri PSO ile optimize edebilen programdır. Bu bölümde bu program ve uygulama örnekleri anlatılacaktır.

Şekil 4.1 'de PSO yöntemi ile optimizasyon sağlayan gelişmiş Matlab GUI 'nin ekran çıktısı görülmektedir.

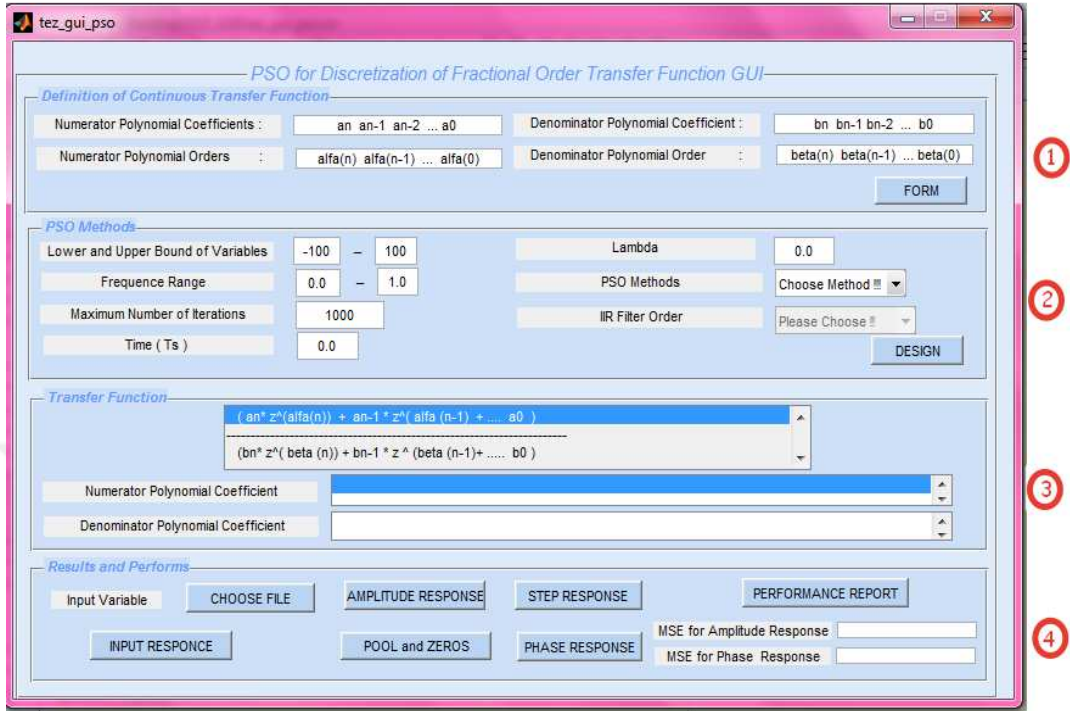
Bu GUI programının kullanım aşamaları şöyle özetlenebilir:

Aşama 1: Kullanıcı sürekli zaman kesir dereceli transfer fonksiyonu katsayı ve derecelerini ilk bölümde (Definition of Continuous Transfer Function paneli) kısmında girer ve "FORM" butonuna basarak bu transfer fonksiyonunu tanıtır.

Aşama 2: İkinci panelde (PSO Methods paneli) PSO optimizasyonu için kullanıcının tercih edebileceği parametreler girilir. Optimize edilecek IIR filtre katsayılarının değişim aralığı, filtrenin çalışma frekans aralığı, maksimum iterasyon sayısı,  $\lambda$  ağırlık katsayısı, örnekleme periyodu, istenen IIR filtre derecesi girilir. "DESIGN" tuşuna basılarak optimizasyon başlatılır ve optimize edilen ayırık yaklaşık eşdeğer filtre katsayıları üçüncü bölümde kullanıcıya (Transfer Function panel) sunulur.

Aşama 3: Elde edilen ayırık filtrenin performans analizleri dördüncü bölümde (Results and Performs panel) gerçekleştirilir. Burada, frekans cevapları olan genlik ve faz cevapları, birim basamak cevabı, kutup-sıfır yerleşim grafikleri çizilir.

Ayrıklaştırılan sürekli zaman kesir dereceli genlik ve faz cevabına göre yakınsama hataları ortalama karesel hata (Mean Square Error) olarak hesaplanır.



Şekil 4.1.PSO yöntemi ile optimizasyon sağlayan gelişmiş GUI programının kullanıcı ara yüzü

Bu GUI program yardımı ile bir örnek tasarım aşağıda sunulmuştur.

**Örnek 4.1 :** Aşağıda verilen kesir dereceli alçak geçiren filtre transfer fonksiyonun geliştirilen GUI programı yardımı ile ayrıklaştırılm ve elde edilen yaklaşık eşdeğer IIR filtrenin yakınsama performansını inceleyelim.

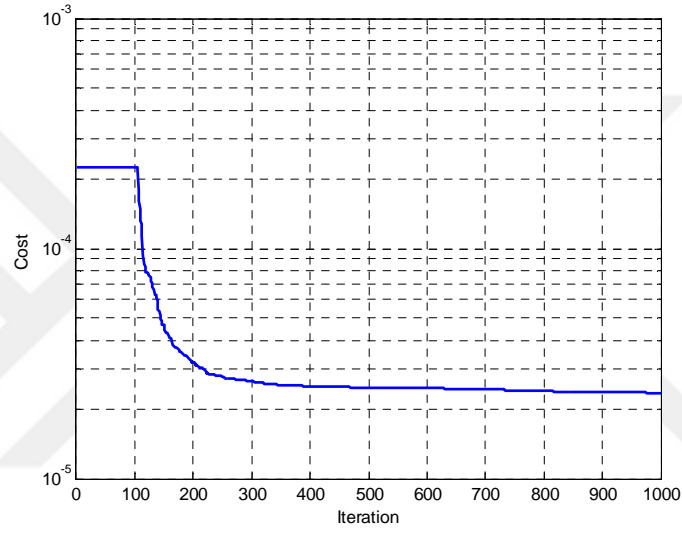
$$F(s) = \frac{1}{s^{0.7} + 1} \quad (4.1)$$

Bu örnekte PSO algoritmasında kullanılan PSO parametreleri şunlardır. Bireysel öğrenme katsayıları  $c_1 = 1.5$  ve  $c_2 = 2.0$ , ivme katsayısı  $w = 1$  ve  $\xi = 0.99$  alınmıştır. Filtre tasarım parametreleri şöyle belirlenmiştir, PSO Metot panelinde filtre katsayılarının değişim aralığı -100 ve 100 verilmiştir. Çalışma frekans aralığı 0.1 ile 1000 aralığında seçilmiştir. Maksimum iterasyon sayısı 1000 ve örnekleme periyodu 0.001 girilmiştir.  $\lambda = 0.6$  olarak ayarlanmıştır. Başlangıç çözümü olarak öz yinelenmeli Tustin analitik çözümü seçilmiştir. Bu çözüm PSO

optimizasyonu ile girilen kullanıcı tercihlerine göre optimize edilmiş ve sonuçta elde edilen kararlı optimum IIR filtre tasarımı:

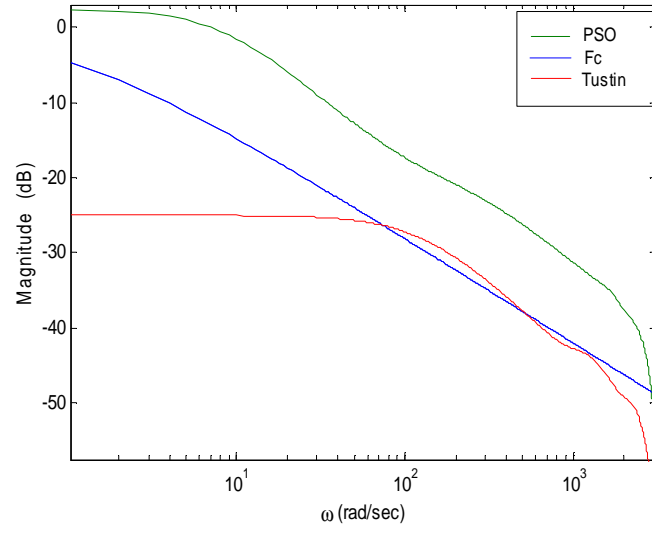
$$F_{PSO\_Tustin}(z) = \frac{1.789z^5 + 0.5011z^4 - 0.1931z^3 - 0.4499z^2 - 1.187z + 0.1347}{89.63z^5 - 99.97z^4 + 44.21z^3 - 71.95z^2 + 19.21z + 19.26} \quad (4.2)$$

Optimizasyon süresince maliyet fonksiyonu  $f(a, b)$  'nin değişimi Şekil 4.2 'de gösterilmiştir. Şekilde görüldüğü üzere algoritma maliyet fonksiyonu minimize ederek yaklaşık 500 iterasyonda optimum filtre tasarımına yakınsamıştır.

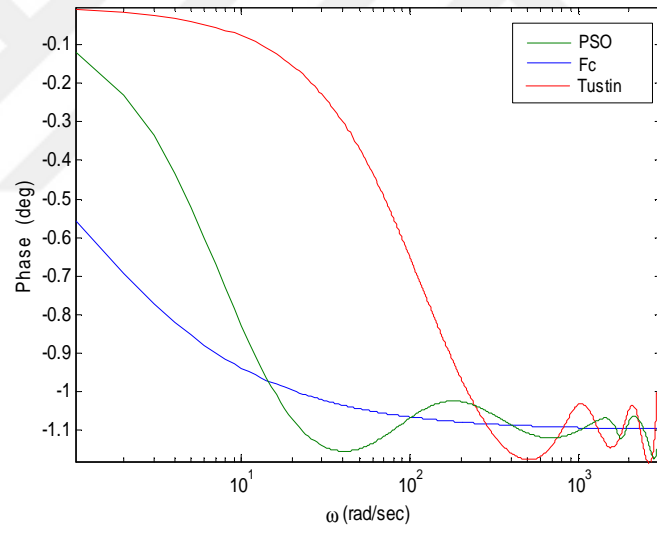


Şekil 4.2.Maliyet fonksiyonun optimizasyon boyunca değişimi

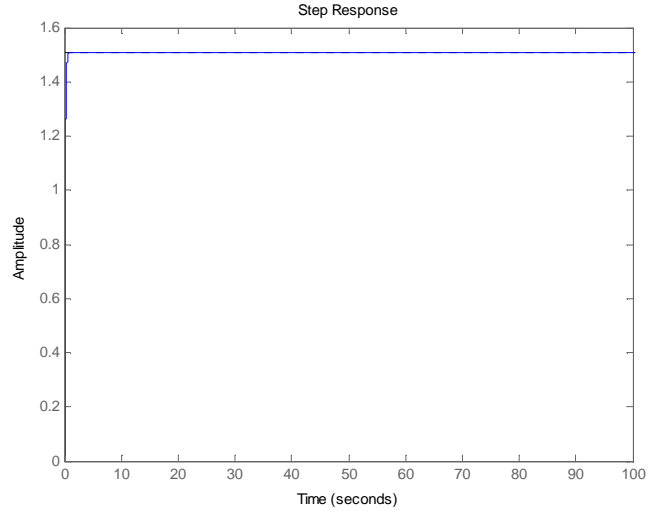
Bu yöntem sonucunda elde edilen ayrık IIR filtresinin genlik ve faz cevapları Şekil 4.3 ve Şekil 4.4 'te verilmiştir. Şekil 4.5 'de birim basamak cevabı ve Şekil 4.6 'da sistem kutup ve sıfırlarının yerleşimi görülmektedir. Elde edilen sonuçlar filtrenin kararlı ve geniş bir frekans bölgesinde makul bir faz ve genlik cevabı sağlayabildiğini göstermiştir. Frekans cevabı ortalama karesel hata değerleri Çizelge 4.1 'de verilmiştir.



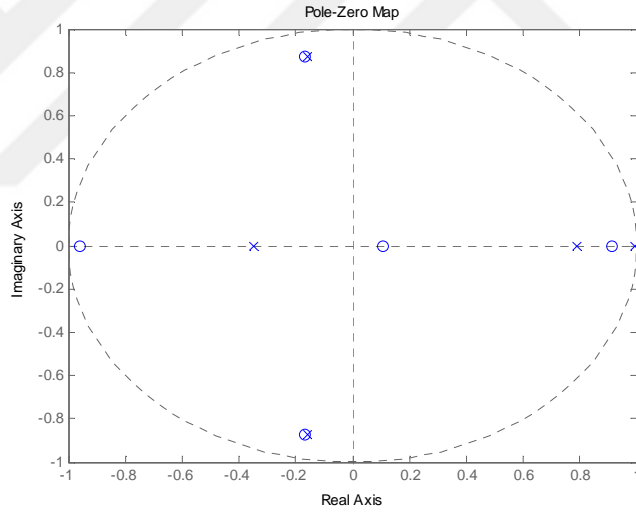
Şekil 4.3.PSO uygulayarak elde edilmiş ayrık IIR filtrenin genlik cevabı



Şekil 4.4.PSO uygulayarak elde edilmiş ayrık IIR filtrenin faz cevabı



Şekil 4.5. Birim basamak cevabı



Şekil 4.6. Sistem kutup ve sıfırlarının dağılımı

#### 4. 2. Analitik Yöntem Sonucu ile PSO Sonucunun Kıyaslaması

Bu bölümde analitik yöntem sonucu elde edilen ayrık yaklaşık eşdeğer IIR filtre ve PSO ile optimize edilen ayrık yaklaşık eşdeğer IIR filtrenin karşılaştırmalı sonuçları sunulacaktır. Bu amaçla nispeten biraz daha karmaşık olan bir kesir derece transfer fonksiyonu için aşağıdaki örnekte yaklaşık eşdeğer IIR filtre tasarımı yapılmıştır.

**Örnek 4.2:** Aşağıda verilen sürekli zaman kesir dereceli transfer fonksiyonu özyinelemeli Tustin yöntemi ile ayrıklaştıralım. Bu tasarımı PSO ile optimize edelim ve sonuçları kıyaslayalım.

$$F(s) = \frac{1}{0.5s^{0.9} + 0.8s^{0.8} + 1} \quad (4.3)$$

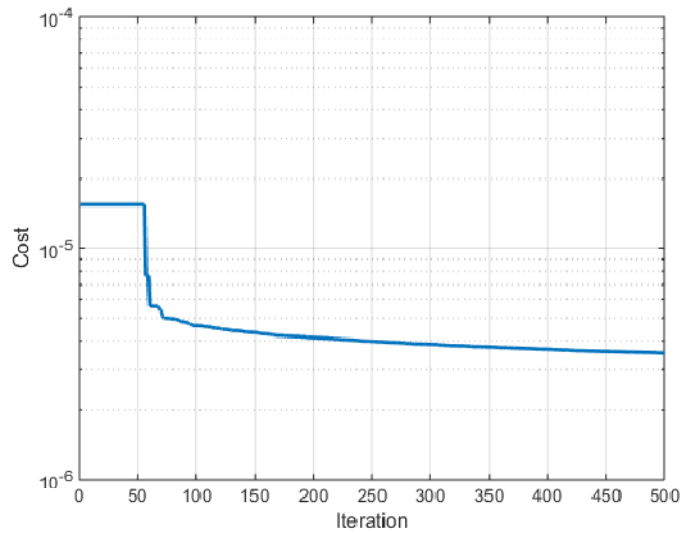
Özyinelemeli Tustin yöntemi ile elde edilen transfer fonksiyonu;

$$F_{Tustin}(z) = \frac{z^6 + 1.1z^5 + 0.4633z^4 + 0.4327z^3 + 0.1236z^2 + 0.022z + 0.02}{62.18z^6 - 38.49z^5 - 6.785z^4 - 10.39z^3 - 6.998z^2 + 0.8139z - 1.204} \quad (4.4)$$

PSO algoritması parametreleri bir önceki örnekte kullanılan parametrelerin aynısıdır. PSO ile optimizasyon sonucunda,

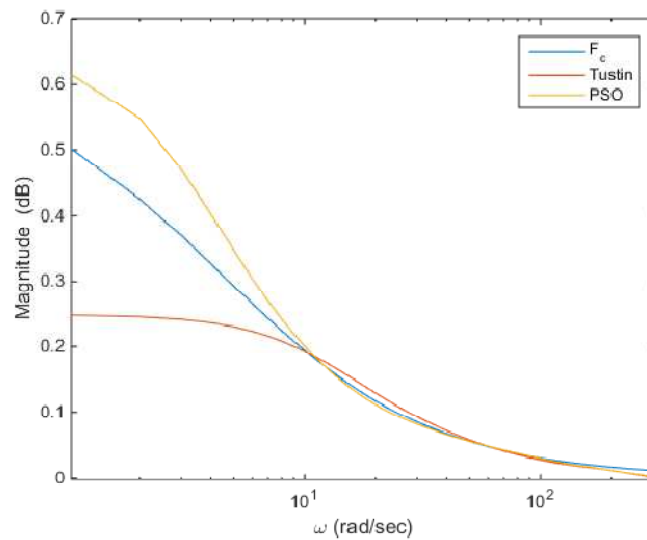
$$F_d(z) = \frac{1.071z^6 + 1.189z^5 + 0.4809z^4 + 0.4305z^3 + 0.1238z^2 + 0.08375z + 0.02727}{62.82z^6 - 37.44z^5 + 3.649z^4 - 9.242z^3 - 7.641z^2 + 0.6607z - 5.208} \quad (4.5)$$

IIR filtre tasarımı elde edilmiştir. Optimizasyon süresince maliyet fonksiyonun değişimi Şekil 4.7 'de gösterilmiştir. Şekilde görüldüğü üzere algoritma maliyet fonksiyonunu minimize ederek yaklaşık 500 iterasyonda optimum filtre tasarımına yakınsamıştır.

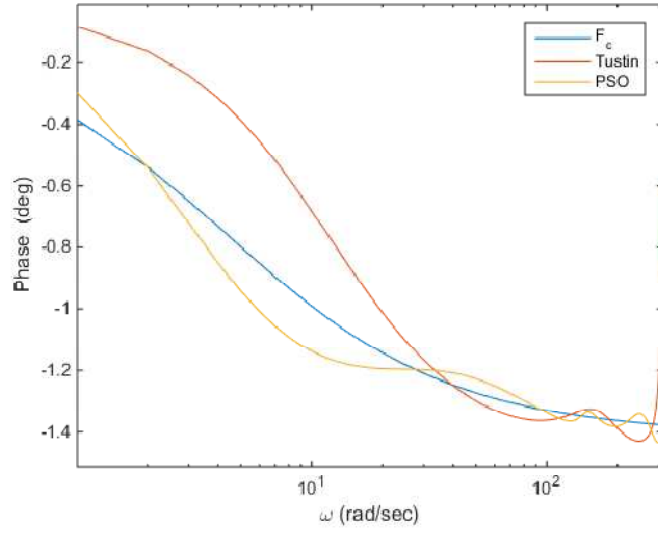


Şekil 4.7. Maliyet fonksiyonun optimizasyon boyunca değişimi

Elde edilen filtrelerin genlik ve faz cevapları karşılaştırmaları Şekil 4.8 ve Şekil 4.9 'da verilmiştir. Şekiller de görüldüğü üzere PSO optimizasyonu, özyinelemeli Tustin yöntemi ile elde edilen filtrenin genlik ve faz cevabı yakınsama performanslarını gözle görülür düzeyde iyileştirmiştir. Bu iyileşmeyi sağlayan temel nedenlerden biri de, özyineleme Tustin yönteminin sadece kesir derece türev terimlerine yakınsama sağlamak üzere geliştirilmesidir. Bu türev terimlerinin yaklaşık eşdeğerleri daha karmaşık bir kesir dereceli transfer fonksiyonlarında ayırklaştırmak için kullandıklarında transfer fonksiyonu frekans cevabına yakınsama performansı düşebilmektedir. Oysaki, PSO doğrudan bu karmaşık kesir dereceli transfer fonksiyonun genlik ve faz cevabına yakınsadığı için daha tutarlı sonuç verebilmektedir. Analitik yöntemlerin başarısı kesir dereceli transfer fonksiyonun karmaşıklığı arttıkça yakınsama hatalarının birikmesi sonucu düşmektedir. Çünkü, sadece tek bir türev terimine yakınsamak üzere tasarlanan analitik yöntemler kompleks transfer fonksiyonları için optimal çözüm sunmazlar. Genlik cevapları kıyaslandığında PSO çözümünün genelde kesir dereceli transfer fonksiyonu genlik cevabına daha yakın olduğu görülmektedir. Ancak, 10 rad/s frekansından sonra kesir dereceli transfer fonksiyonu genlik cevabına oldukça yakınsadığı görülür. Çizelge 4.1 'de genlik ve faz cevaplarının ortalama karesel hataları kıyaslamaktadır. Genlik ve faz cevabı yakınsama performansında sağlanan iyileşme açıkça görülmektedir.



Şekil 4.8.Özyinlemeli Tustin ve PSO ile elde edilmiş ayrık IIR filtrenin genlik cevabı



Şekil 4.9.Özyinemeli Tustin ve PSO ile elde edilmiş ayrık IIR filtrenin faz cevabı

Çizelge 4.1. Ortalama karesel genlik ve faz cevabı hatalarının kıyaslaması

<i>Frekans Cevabı</i>	<i>Özyinelemeli Tustin</i>	<i>PSO optimizasyonu</i>
Faz Cevabı	0.0286	0.0081
Genlik Cevabı	0.0018	5.7756e-04

## 5. TARTIŞMA VE SONUÇ

Bu çalışmada rasyonel formda ifade edilen kesir dereceli transfer fonksiyonlarının sayısal gerçekleştirilmesi için ayrık yaklaşık eşdeğer IIR filtre tasarımı gerçekleştirilmiştir. Kesir dereceli sistemlerin sayısal gerçekleştirilmesi ve kesir derece sistem modellerinin simülasyonu, çalışma süreleri ile sürekli artan yüksek işlem maliyetleri getirmektedir. Bunun temel nedeni, kesir dereceli türevin "uzun bellek" (long memory) etkisi olarak anılan, türev hesaplamasının fonksiyonun türev alınan noktanın yakın civarındaki lokal değerleri yanında fonksiyonun başlangıç anına kadar olan bütün değerlerine de bağımlı olmasıdır. Bunun sonucunda her yeni nokta için yapılacak kesir dereceli türev hesaplamasında önceki bütün noktalarında hesaba katılması gerekmektedir. Bu durum, çalışma süresi ile işlem yükünün sürekli artmasına yol açmakta ve sınırlı olan hesaplama kaynaklarını (bellek ve işlem gücü) tüketmeye başlamaktadır. Bu nedenle kesir dereceli sistemlerin sayısal sistemlerde ideal (minimum hata ile) gerçekleştirilmesi çok yüksek işlem maliyetlerine yol açmaktadır.

Bu sorunun çözümü için en etkin yol gerçekleştirilmek istenen kesir dereceli transfer fonksiyonunun yaklaşık eşdeğer modellerinden yararlanılması olarak görülmüştür. Bu amaçla frekans bölgesinde kesir dereceli türev operatörlerine ( $s^\alpha$ ) yakınsayan ayrık filtre tasarım yöntemleri geliştirilmiştir. Bu tür analitik yöntemler genelde seri açılım ve yakınsama tekniklerine dayanmakta ve sadece kesir dereceli türev operatörünün yaklaşık eşdeğer gerçekleştirilmesi için geliştirilmektedir. Bu nedenle, rasyonel formda pay ve paydasında birçok kesir derece türev  $s^{\alpha_i}$  barındıran daha karmaşık formdaki kesir dereceli transfer fonksiyonlarının bu kesir dereceli türev operatörünün yaklaşık eşdeğeri kullanılarak gerçekleştirilmesinde genlik ve faz cevabına yakınsama performansları düşebilmektedir. Bu tez çalışmasında, temel analitik yöntemlerin ürettiği yaklaşık IIR filtre çözümü ile başlayıp, bu çözümleri kullanıcının faz ve genlik cevapları yakınsama önceliği ve çalışma frekans bölgesine göre optimize eden bir yaklaşım geliştirilmiştir. Bu yaklaşımın uygulama açısından çalışma frekansı bölgesinde istenen özellikte yakınsama sağlayabilen yaklaşık IIR filtre çözümlerini elde etmekte faydalı olacağı görülmüştür [14]. Bunun için PSO algoritması adapte edilmiş ve elde edilen teorik birikim Matlab GUI programı ile bu alanda çalışan araştırmacılar için kolay kullanılabilir forma kavuşturulmuştur. Şekil 5.1 'de uygulanan yaklaşık eşdeğer filtre tasarım yaklaşımının aşamaları akış şeması ile

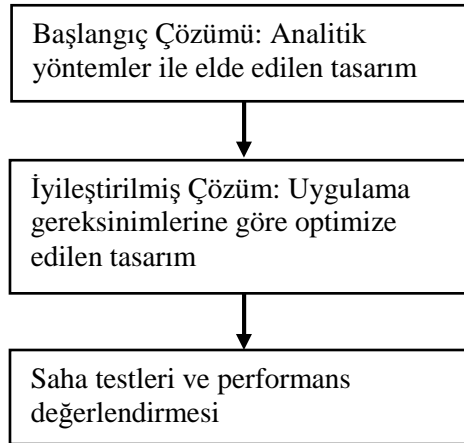
özetlenmiştir. Geliştirilen yöntemin temel özellikleri şöyle özetlenebilir.

(i) Temel analitik yöntemler ile elde edilen kesir dereceli türevin ayrık yaklaşık eşdeğer filtreleri kullanılarak rasyonel formda kesir dereceli transfer fonksiyonların gerçekleştirilmesi sağlanmış ve bu çözüm Matlab GUI modülüne dönüştürülmüştür.

(ii) Pratik uygulamaların gereklerine ve spesifikasyonlarına cevap verebilmesi için analitik yöntem ile elde edilen sonuçlar PSO optimizasyonu ile iyileştirilebilmiştir. Örneğin işaret işleme uygulaması için yüksek frekans bölgesi ve genlik cevapları önemli iken kontrol uygulamaları için düşük frekans bölgesi ve faz cevapları önemlidir. Bu gereksinimlere dönük olarak yaklaşık eşdeğer filtrelerin geliştirilen Matlab GUI yardımı ile iyileştirilebileceği görülmüştür.

(iii) Bu uygulama için PSO algoritması adapte edilmiştir. Parçacık sürüsünün filtre pay ve payda polinom katsayıları ile teşkil edilen filtre tasarım uzayında arama yapması sağlanmıştır. Başlangıç parçacıkları analitik yöntem çözümü ve bunun yakın civarında rasgele üretilmiştir. Böylece, optimizasyona analitik çözümlerden başlanmıştır.

(iv) Kararlı filtre çözümleri elde etmek için parçacıklar kararlı filtre çözümü veren bölgeye yönlendirilmiştir. Bu işlem kararsız filtre çözümü veren parçacıkların maliyet değerinin çok yüksek değere modifiye edilmesi ile sağlanmıştır. Böylece parçacıkların kararsız filtre çözümüne yol açan bölgelerden uzaklaşmaları sağlanmıştır. Bu işlem optimizasyon süresince filtre kararlılık durumuna göre biçimlenen dinamik bir maliyet fonksiyonu ile parçacık sürülerinin yayılımı kontrol edilebilmiştir.



Şekil 5.1. Bu çalışmada uygulanan yaklaşım

## 5.1. Gelecek Çalışma Önerileri

Bu tez çalışmasında teorik araştırma, çalışma yapılmış ve yazılım araçları (GUI) geliştirilmesine odaklanılmıştır. Belirli bir saha uygulamalarında performans değerlendirilmesi veya deneysel çalışma yapılmamıştır. Geliştirilen yöntem ile elde edilen çözümlerin kontrol, işaret işleme gibi alanlarda uygulanması ve spesifik bir kaç uygulama probleminde performansının değerlendirilmesi faydalı olacaktır.

Ayrıca, PSO dışında yeni nesil metasezgisel yöntemlerin uygulanması optimizasyon performansını artırabilir ve yakınsama performansı daha yüksek filtre tasarımları sağlayabilir. Gelecek çalışmalarda yöntemin yeni nesil metasezgisel yöntemlere uygulanması çalışılabilir.

## 6. KAYNAKÇA

- [1] R. Caponetto, G. Dongola, L. Fortuna ve I. Petras, *Fractional Order Systems: Modeling and Control Applications*, World Scientific, Singapore, 2010, 1-20.
- [2] A.Tustin, J.T. Allanson, J.M. Layton, R.J. Jakeways, *The Design of Systems for Automatic Control of the Position of Massive Objects*, **Proceedings of the IEEE - Part C: Monographs**, 105:1 (1958) 1-57.
- [3] R.L Bagley ve R. A. Calico, *Fractional Order State Equations for the Control of Viscoelastic Damped Structures*, **J. Guidance, Control and Dynamics**, 14:2 (1991) 304-311.
- [4] A. Makroglou, R.K. Miller ve S. Skaar, *Computational Results for a Feedback Control for a Rotating Viscoelastic Beam*, **J. Guidance, Control and Dynamics**, 17:1 (1994) 84-90.
- [5] T.F Nonnenmacher ve W.G. Glöckle, *A Fractional Model for Mechanical Stress Relaxation*, **Philosophical Magazine Lett**, 64:2 (1991) 89-93.
- [6] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.
- [7] D. Gül, İTÜ Polen, İTÜ Akademik Açık Arşiv, (2018) <http://polen.itu.edu.tr/> (Online erişim 05 Mart, 2017).
- [8] İlhan Mutlu, *Kesirli Mertebeden Kontrolörler Ve Uygulamaları* Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, 2010.
- [9] F.Ü.Teknolojileri Fakültesi EEM M. Gökbulut, Sayısal Kontrol Sistemleri. <http://ee.tek.firat.edu.tr/sites/ee.tek.firat.edu.tr/files/Say%C4%B1sal%20Kontrol.pdf> (Online erişim 03 Mart, 2017).
- [10] Y. Chen, I.Petras, D. Xue, *Fractional order control-a tutorial*, in American Control Conference, ACC'09, IEEE, (2009), pp.1397-1411.
- [11] D. Matignon, *Generalized Fractional Differential and Difference Equations, in Stability Properties and Modelling Issues. Proc. of Math. Theory of Networks and Systems Symposium, Padova, Italy, (1998) pp.145-158.*
- [12] J.A Tenreiro Machado, A.M Galhano, A.M.Oliveira, et al, *Optimal approximation of fractional derivatives through discrete-time fractions using genetic algorithms. Commun, Nonlinear Sci. Numer. Simul*, 15:3 (2010) 482-490.
- [13] I.Podlubny, I.Petras, B. M.Vinagre, P. O'leary ve L.Dorcak, *Analogue realizations of fractional order controllers*, **Nonlinear Dynamics**, 29 (2002) 281-296.
- [14] A. Ates, G. Kavuran, B. B. Alagoz ve C. Yeroglu, in : Improvement of IIR filter discretization for fractional order filter by discrete stochastic optimization. In Telecommunications and Signal Processing (TSP), 39th International Conference. IEEE, Vienna, (2016), pp. 583-586.
- [15] A. Ates, B. B.Alagoz, G. Kavuran, ve C. Yeroglu, Implementation of fractional order filters discretized by modified Fractional Order Darwinian Particle Swarm Optimization. **Measurement**, 107 (2017) 153-164.
- [16] R. Alti, M. Aoun, J. Sabatier ve A. Oustaloup, *Tutorial on system identification using fractional differentiation models*, **IFAC Proceedings**, 39:1 (2006) 606-611.
- [17] A. Özel, N. Tan, *DC-DC Boost Konvertörün PID ve Kesirli Dereceli PID ile Simulink/Matlab Ortamında Kontrolü*, TOK 2014 Bildiri Kitabı, Koacaeli, 11-13 Eylül, (2014), pp.170-175.
- [18] B.T. Krishna ve K. V. V. S.Reddy, *Active and passive realization of fractance device of order 1/2*, Active and Passive Electronic Components, 2008.
- [19] Z. Gao, X. Liao, *Rational approximation for fractional-order system by particle swarm optimization*, **Nonlinear Dyn** 67 (2012) 1387-1395.

- [20] M. M. Özyetkin, N. Tan, *Kesirli Dereceli Sistemlerin Tamsayı Dereceli Yaklaşımı*, SUI 2010 – IEEE 18. Sinyal İşleme ve İletişim Uygulamalar Kurultayı, Diyarbakır, 22-24 Nisan, (2010), pp.949-952.
- [21] R. S. Barbosa, J.A. T.Machado, I. M. Ferreira, Pole Zero Approximations of Digital Fractional-Order Integrators and Differentiators Using Signal Modeling Techniques, 16th IFAC World Congress, Prague, Czech Republic, July 4–8, (2005) , pp. 350-371.
- [22] J. F. Claerbout, *Fundamentals of Geophysical Data Processing with applications to petroleum prospecting*, Blackwell Scientific Publications, California, 1976.
- [23] Y.Q. Chen ve K.L. Moore, *Discretization schemes for fractional- order differentiators and integrators*, **IEEE Trans. On Circuits and Systems - I: Fundamental Theory and Applications**, 49:3 (2002) 363–367.
- [24] B.M. Vinagre, I.Podlubny, A. Hernandez ve V.Feliu, *Some approximation of fractional fractional order operators used in control theory and applications*, **J. Fractional Calculus Appl. Anal.**, 3:3 (2001) 47-66.
- [25] W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1992.
- [26] A.I. Khinchin, A.Ya. Khinchin, H. Eagle, *Continued Fractions*, Dover (Inc Scripta Technica,Translator), New York, 1997.
- [27] Mehmet Korkmaz, *Kesirli Dereceden PID Denetleyicilerin tasarımı Uygulaması ve Karşılaştırılması*, Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği Anabilimdalı, Selçuk Üniversitesi, Konya, 2013.

## EKLER

Bu bölümde analitik yöntemler ile ayrıklaştırma için yazılmış kodlar verilmiştir.

### Tustin\_tf.m

```
function [ tustin_res ] = Tustin_tf( Alfa,n,Ts )

if(n==1)
    num01=[(2/Ts)^Alfa ((2/Ts)^Alfa*(-Alfa))];
    dem01=[1 (Alfa)];
    tustin_res = tf(num01,dem01,Ts,'variable','z^-1');
elseif(n==3)

    num03=[          ((2/Ts).^Alfa)          ((2/Ts).^Alfa.*(-Alfa))
((2/Ts).^Alfa.*(1/3).*Alfa.^2) ((2/Ts).^Alfa.*(-1/3).*Alfa)];
    dem03=[(1) ((Alfa)) ((1/3).*Alfa.^2) ((-1/3).*(-Alfa))];
    tustin_res = tf(num03,dem03,Ts,'variable','z^-1')
elseif(n==5)
    num05=[(2/Ts)^Alfa ((2/Ts)^Alfa*(-Alfa)) ((2/Ts)^Alfa*((2/5)*(Alfa^2)))
...
          ((2/Ts)^Alfa*(((-1/3)*Alfa)+((-1/15)*(Alfa^3))))
((2/Ts)^Alfa*((1/5)*(Alfa^2))) ((2/Ts)^Alfa*((-1/5)*Alfa))];
    dem05=[(1) (Alfa) ((2/5)*(Alfa^2)) (((1/3)*Alfa)+((1/15)*(Alfa^3)))
((1/5)*(Alfa^2)) ((1/5)*Alfa)];
    tustin_res = tf(num05,dem05,Ts,'variable','z^-1');

elseif(n==7)
    num07=[(2/Ts)^Alfa          ((2/Ts)^Alfa*(-Alfa))
((2/Ts)^Alfa*((2/5)*(Alfa^2)))...
          ((2/Ts)^Alfa*((-1/3)*Alfa+((-2/21)*(Alfa^3))))
((2/Ts)^Alfa*((26/105)*(Alfa^2))+((1/105)*(Alfa^4)))...
          ((2/Ts)^Alfa*(((-1/5)*(Alfa))+((-2/35)*(Alfa^3))))
((2/Ts)^Alfa*((1/7)*(Alfa^2))) ((2/Ts)^Alfa*((-1/7)*(Alfa)))]];
    dem07=[(1) (Alfa) ((2/5)*(Alfa^2)) (((1/3)*Alfa+((2/21)*(Alfa^3))))...
          (((26/105)*(Alfa^2))+((1/105)*(Alfa^4))))...
          (((1/5)*(Alfa))+((2/35)*(Alfa^3))))          (((1/7)*(Alfa^2)))
          (((1/7)*(Alfa)))]];
    tustin_res = tf(num07,dem07,Ts,'variable','z^-1');

elseif(n==9)
```

```

num09=[(2/Ts)^Alfa ((2/Ts)^Alfa*(-Alfa))
((2/Ts)^Alfa*((4/9)*(Alfa^2)))...
((2/Ts)^Alfa*((-1/3)*Alfa+((-1/9)*(Alfa^3))))
((2/Ts)^Alfa*((17/63)*(Alfa^2))+((1/63)*(Alfa^4)))...
((2/Ts)^Alfa*((-1/5)*(Alfa))+((-16/189)*(Alfa^3))+((-
1/945)*(Alfa^5))) ...
((2/Ts)^Alfa*((34/189)*(Alfa^2))+((2/189)*(Alfa^4)))
((2/Ts)^Alfa*((-1/7)*(Alfa))+((-1/21)*(Alfa^3)))...
((2/Ts)^Alfa*((1/9)*(Alfa^2)) ((2/Ts)^Alfa*((-1/9)*(Alfa)))];
dem09=[(1) (Alfa) (((4/9)*(Alfa^2)))...
(((1/3)*Alfa+((1/9)*(Alfa^3))))
(((17/63)*(Alfa^2))+((1/63)*(Alfa^4)))...
((((1/5)*(Alfa))+((16/189)*(Alfa^3))+((1/945)*(Alfa^5)))) ...
((((34/189)*(Alfa^2))+((2/189)*(Alfa^4))))
((((1/7)*(Alfa))+((1/21)*(Alfa^3))))...
(((1/9)*(Alfa^2)) ((1/9)*(Alfa)))]];
tustin_res = tf(num09,dem09,Ts,'variable','z^-1');
else
disp('N değerlerini sadece 1, 3, 5, 7,9 seçebilirsiniz!!!')
end

```

## Oustafod.m

```

function G=oustafod(r,N,wb,wh,Ts)

mu=wh/wb;
k=-N:N;
w_kp=(mu).^((k+N+0.5-0.5*r)/(2*N+1))*wb;
w_k=(mu).^((k+N+0.5+0.5*r)/(2*N+1))*wb;
K=wh^r;
G1=tf(zpk(-w_kp',-w_k',K));
G=c2d(G1,Ts);
end

```

## Cfefunct\_direct.m

```

function [dr_CFE ] = CFefunct_direct( r,n,Ts )

if (r==0)
cfe_fc=tf(1,1,Ts,'variable','z^-1');

```

```

else
    z=(2/Ts)^r ;
if (n==1)
    P_CFE1=[(2/Ts)^r ((2/Ts)^r*(-1)*r)];
    Q_CFE1=[1 r];
    cfe_fc= tf(P_CFE1,Q_CFE1,Ts,'variable','z^-1');

elseif (n==3)

    P_CFE3= z* [ 15 (-1)*15*r (6*r^2-9) (-1)*(r^3-4*r)];
    Q_CFE3=[ 15 (15*r) (6*r^2-9) (r^3-4*r)];
    cfe_fc= tf(P_CFE3,Q_CFE3,Ts,'variable','z^-1')
elseif (n==5)
    P_CFE5= z*[945 ((-1)*945*r) ((420*r^2-1050)) ((-1)*(105*r^3-735*r)) ((-
195*r^2+15*r^4+225)) ((-1)*(r^5-20*r^3+64*r))];
    Q_CFE5=[945 (945*r) ((420*r^2-1050)) ((105*r^3-735*r)) ((-
195*r^2+15*r^4+225)) ((r^5-20*r^3+64*r))];
    cfe_fc= tf(P_CFE5,Q_CFE5,Ts,'variable','z^-1');

elseif (n==7)
    P_CFE7= z*[135135 ((-1)*135135*r) ((62370*r^2-218295)) ...
((-1)*(17325*r^3-173250*r)) ((99225-59850*r^2+3150*r^4)) ...
((-1)*(378*r^5-11340*r^3+53487*r)) ((10612*r^2-1190*r^4-
11025+28*r^6))...
((-1)*(784*r^3+r^7-56*r^5-2304*r))];
    Q_CFE7=[135135 (135135*r) ((62370*r^2-218295)) ...
((17325*r^3-173250*r)) ((99225-59850*r^2+3150*r^4)) ...
((378*r^5-11340*r^3+53487*r)) ((10612*r^2-1190*r^4-
11025+28*r^6))...
((784*r^3+r^7-56*r^5-2304*r))];
    cfe_fc= tf(P_CFE7,Q_CFE7,Ts,'variable','z^-1');

elseif (n==9)
    P_CFE9= z*[34459425 ((-1)*34459425*r) ((16216200*r^2-72972900)) ...
((-1)*(4729725*r^3-61486425*r)) ((51081030-23648625*r^2+945945*r^4))
...
((-1)*(135135*r^5-5405400*r^3+33648615*r)) ((9514890*r^2-
796950*r^4-13097700+13860*r^6))...
((-1)*(1451835*r^3+990*r^7-76230*r^5-5742495*r)) ((45*r^8-
909765*r^2+120330*r^4+893025-4410*r^6))...
((-1)*(r^9-52480*r^3-120*r^7+4368*r^5+147456*r))];
    Q_CFE9=[34459425 (34459425*r) ((16216200*r^2-72972900)) ...
((4729725*r^3-61486425*r)) ((51081030-23648625*r^2+945945*r^4)) ...

```

```

        ((135135*r^5-5405400*r^3+33648615*r))      ((9514890*r^2-796950*r^4-
13097700+13860*r^6))...
        ((1451835*r^3+990*r^7-76230*r^5-5742495*r))      ((45*r^8-
909765*r^2+120330*r^4+893025-4410*r^6))...
        ((r^9-52480*r^3-120*r^7+4368*r^5+147456*r))];
    cfe_fc= tf(P_CFE9,Q_CFE9,Ts,'variable','z^-1');
end
end
%dr_CFE = d2d(cfe_fc,Ts); % Continues function is discretized
dr_CFE=cfe_fc

end

```

### Cfefunct\_indirect.m

```

function [tf_CFE ] = CFefunct(Alfa,n,Ts )

if (n==1)
    num_CFE1=[(1+Alfa) (1-Alfa)];
    dem_CFE1=[(1-Alfa) (1+Alfa)];
    tfc=tf(num_CFE1,dem_CFE1);
elseif (n==2)
    num_CFE2=[(Alfa^2+3*Alfa+2) (-2*Alfa^2+8) (Alfa^2-3*Alfa+2)];
    dem_CFE2=[(Alfa^2-3*Alfa+2) (-2*Alfa^2+8) (Alfa^2+3*Alfa+2)];
    tfc=tf(num_CFE2,dem_CFE2);
elseif (n==3)
    num_CFE3=[(Alfa^3+6*Alfa^2+11*Alfa+6)      (-3*Alfa^3-6*Alfa^2+27*Alfa+54)
(3*Alfa^3-6*Alfa^2-27*Alfa+54) (-Alfa^3+6*Alfa^2-11*Alfa+6)];
    dem_CFE3=[(-Alfa^3+6*Alfa^2-11*Alfa+6) (3*Alfa^3-6*Alfa^2-27*Alfa+54) (-
3*Alfa^3-6*Alfa^2+27*Alfa+54) (Alfa^3+6*Alfa^2+11*Alfa+6)];
    tfc=tf(num_CFE3,dem_CFE3);
elseif (n==4)
    num_CFE4=[(Alfa^4+10*Alfa^3+35*Alfa^2+50*Alfa+24)      (-4*Alfa^4-
20*Alfa^3+40*Alfa^2+320*Alfa+384)      (6*Alfa^4-150*Alfa^2+864)      (-
4*Alfa^4+20*Alfa^3+40*Alfa^2-320*Alfa+384)      (Alfa^4-10*Alfa^3+35*Alfa^2-
50*Alfa+24)];
    dem_CFE4=[(Alfa^4-10*Alfa^3+35*Alfa^2-50*Alfa+24)      (-
4*Alfa^4+20*Alfa^3+40*Alfa^2-320*Alfa+384)      (6*Alfa^4-150*Alfa^2+864)      (-
4*Alfa^4-20*Alfa^3+40*Alfa^2+320*Alfa+384)
(Alfa^4+10*Alfa^3+35*Alfa^2+50*Alfa+24)];
    tfc=tf(num_CFE4,dem_CFE4);
end
tf_CFE =c2d(tfc,Ts); % Continues function is discretized
end

```

## ÖZGEÇMİŞ

<b>KİŞİSEL BİLGİLER</b>		
<b>Adı</b>	Özlem	
<b>Soyadı</b>	İMİK ŞİMŞEK	
<b>E-mail</b>	imikozlem@gmail.com	
<b>EĞİTİM VE ÖĞRETİM</b>		
<b>ADI</b>	<b>KURUM</b>	<b>DÖNEM</b>
<b>Yüksek Lisans</b>	İnönü Ün. Fen Bilimleri Enstitüsü Bilgisayar Müh. A.B.D, Malatya	2016- 2018
<b>Lisans</b>	İnönü Ün. Mühendislik Fak. Bilgisayar Mühendisliği, Malatya	2012 – 2016
<b>Hazırlık</b>	İnönü Ün. Yabancı Diller Y.O İngilizce Hazırlık. Malatya	2011 – 2012
<b>AKADEMİK ÇALIŞMALAR</b>		
<b>Yüksek Lisans Tezi</b>		
Parçacık Sürüsü Optimizasyon Yöntemi ile Kesir Dereceli Filtre Gerçeklemesi, İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Malatya, 2018		
<b>Yayınlar</b>		
Ozlem Imik, Baris Baykant Alagoz, Discretization of Fractional Order Transfer Functions by Weighted Multi-Objective Particle Swarm Optimization Method, International Artificial Intelligence And Data Processing Symposium, 2017.		
Ozlem Imik, Baris Baykant Alagoz, Celaledin Yeroglu, Hafiz Alisoy, Discretization of Fractional Order Transfer Function by Particle Swarm Optimization Method, International Workshop on Mathematical Methods in Engineering (MME-2017), Ankara, 27-29 April, 2017		
Ozlem Imik, Baris Baykant Alagoz, Abdullah Ates, Celaledin Yeroglu, Fractional Order Filter Discretization by Particle Swarm Optimization Method, International Workshop on Mathematical Methods in Engineering (MME-2017), Ankara, 27-29 April, 2017		
Ozlem Imik, Baris Baykant Alagoz, Abdullah Ates, Celaledin Yeroglu, Fractional Order Filter Discretization by Particle Swarm Optimization Method, In Mathematical Methods in Engineering, Springer International Publishing AG, part of Springer Nature (The book will be published in 2019).		