

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



**KESİR DERECELİ PID DENETLEYİCİLERİN TASARIMINA  
YÖNELİK MATLAB ARAYÜZÜ GELİŞTİRİLMESİ**

**Kazim Dođuhan SELÇUK**

Yüksek Lisans Tezi

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Devreler ve Sistemler Bilim Dalı

TEMMUZ 2020

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

Elektrik Elektronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

**KESİR DERECELİ PID DENETLEYİCİLERİN TASARIMINA**  
**YÖNELİK MATLAB ARAYÜZÜ GELİŞTİRMESİ**

Tez Yazarı

**Kazim Dođuhan SELÇUK**

Danışman

Prof.Dr. Arif GÜLTEN

TEMMUZ 2020

ELAZIĞ

T.C.

**FIRAT ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

Elektrik Elektronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı: Kesir Dereceli PID Denetleyicilerin Tasarımına Yönelik Matlab Arayüzü Geliştirilmesi

Yazarı: Kazım Doğuhan SELÇUK

İlk Teslim Tarihi: 17.6.2020

Savunma Tarihi: 8.7.2020

**TEZ ONAYI**

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

İmza

Danışman: Prof.Dr.Arif GÜLTEN

Onayladım

Fırat Üniversitesi, Fen Fakültesi

Başkan: Dr.Öğr.Üyesi Mehmet ÜSTÜNDAĞ

Onayladım

Turgut Özal Üniversitesi

Üye: Doç.Dr.Vedat ÇELİK

Onayladım

Fırat Üniversitesi

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Soner ÖZGEN  
Enstitü Müdürü

## BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım “Kesir Dereceli PID Denetleyicilerin Tasarımına Yönelik Matlab Arayüzü Geliştirilmesi” Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

8/07/2020

**Kazim Doğuhan SELÇUK**



# Önsöz

---

Kontrol sistemleri gündelik hayattaki en basit problemenden endüstrideki en karmaşık problemlere kadar her yerde insan hayatıyla iç içedir. Karşılaşılan bu problemlerin çözümünde en önemli nokta bu problemlerin arzu edilen limitler ve sürelerin içerisinde çözülmesidir. Bu nedenle kontrol sistemleri alanı daha güzel çözümler üretebilmek adına araştırmacıların ilgisini çekmektedir. Temel olarak tasarım, simülasyon ve gerçekleştirme eksenini üzerinde konumlanan araştırmacılar kullanışlı tasarım ve simülasyon programlarına ihtiyaç duymaktadır. Bu tez kapsamında geliştirilen arayüz bu alandaki gelişmelere büyük katkı sağlayacaktır.

Bu çalışmamda her türlü yardımını ve desteklerini esirgemeyen kıymetli hocam Prof. Dr. Arif GÜLTEN ‘ sonsuz teşekkürlerimi sunarım.

Hayatım boyunca benden hiçbir desteğini esirgemeyen kıymetli aileme şükranlarımı sunar ve teşekkür ederim.

Mesleğime tutku ile bağlanmamı sağlayan Dr.Öğr. Üye. Fikret ATA teşekkür ve minnettarlığımı sunarım. Akademik birikimleriyle bana yol gösteren kıymetli hocalarım Prof. Dr.Yakup DEMİR , Prof. Dr. Mustafa TÜRK , Doç. Dr. Vedat ÇELİK ve Dr. Öğr .Üye. Barış KARAKAYA’ya teşekkürlerimi sunarım.

**Kazim Dođuhan SELÇUK**  
ELAZIĞ, 2020

# İÇİNDEKİLER

## Sayfa

|                                                                                     |           |
|-------------------------------------------------------------------------------------|-----------|
| ÖNSÖZ .....                                                                         | iv        |
| İÇİNDEKİLER .....                                                                   | v         |
| ÖZET .....                                                                          | vii       |
| ABSTRACT .....                                                                      | viii      |
| ŞEKİLLER LİSTESİ .....                                                              | ix        |
| TABLO LİSTESİ .....                                                                 | xi        |
| SEMBOLLER LİSTESİ .....                                                             | xii       |
| <b>1. GİRİŞ .....</b>                                                               | <b>1</b>  |
| 1.1. Literatür Taraması .....                                                       | 2         |
| <b>2. KONTROL ELEMANLARI VE KONTROL SİSTEMLERİ.....</b>                             | <b>4</b>  |
| 2.1. Açık-Kapalı (on-off) Denetleyici .....                                         | 5         |
| 2.2. Oransal Denetleyici (P).....                                                   | 5         |
| 2.3.Oransal + İntegral Denetleyici (PI).....                                        | 7         |
| 2.4.Oransal + Türev Denetletici (PD).....                                           | 8         |
| 2.5 Oransal + İntegral + Türev Denetleyici (PID) .....                              | 10        |
| 2.6. PID Parametrelerini Belirleme Yöntemleri .....                                 | 13        |
| 2.6.1. Klasik Yöntemler.....                                                        | 13        |
| 2.6.2 Sezgisel Yöntemler .....                                                      | 18        |
| <b>3. KESİR DERECELİ SİSTEMLER .....</b>                                            | <b>20</b> |
| 3.1. Kesir Dereceli Sistemleri Farklı Tanımları.....                                | 20        |
| 3.1.1. Riemann – Liouville Tanımı.....                                              | 21        |
| 3.1.2 Grünwald – Letnikov Tanımı .....                                              | 21        |
| 3.1.3. Caputo Tanımı .....                                                          | 22        |
| 3.2. Kesir Dereceli Sistemlerin Transfer Fonksiyonu.....                            | 22        |
| 3.3. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyicisi.....                        | 22        |
| <b>4. MATERYAL VE METOT.....</b>                                                    | <b>26</b> |
| <b>5. KESİR DERECELİ <math>PI^{\lambda}D^{\mu}</math> DENETLEYİCİ TASARIMI.....</b> | <b>28</b> |
| <b>6. MATLAB ARAYÜZ TASARIMI .....</b>                                              | <b>33</b> |
| 6.1 Kesir Dereceli Sistemlerin Modellenmesi .....                                   | 33        |
| 6.2 Kesir Dereceli Sistemlerin Geri Beslemeleri.....                                | 35        |
| 6.3 Kesir Dereceli Sistemlerin Simülasyonu .....                                    | 36        |
| 6.4 Kesir Dereceli Sistemlerin Frekans Domeninde Analizi .....                      | 38        |

|                                                                                                |           |
|------------------------------------------------------------------------------------------------|-----------|
| 6.5. Kesir Dereceli Kontrolörlerin Parametrelerinin Tespit Edilmesi .....                      | 42        |
| 6.5.1. Ziegler-Nichols Yöntemi .....                                                           | 43        |
| 6.5.2. Cohen-Coon Yöntemi .....                                                                | 47        |
| 6.5.3. Chien-Hrones-Reswich Yöntemi .....                                                      | 52        |
| 6.6. Arayüz Tasarımı İçin Gelştirilen Komut Dizileri .....                                     | 58        |
| 6.6.1. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Parametrelerinin Tespit Edilmesi ..... | 58        |
| 6.6.2. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Transfer Fonksiyonunun Tasarımı .....  | 60        |
| 6.6.3. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Simülasyonu .....                      | 61        |
| <b>7. MATLAB GUI TASARIMI .....</b>                                                            | <b>67</b> |
| <b>8. SONUÇLAR .....</b>                                                                       | <b>77</b> |
| KAYNAKLAR .....                                                                                | 78        |
| ÖZGEÇMİŞ .....                                                                                 |           |

## ÖZET

---

### Kesir Dereceli Denetleyicilerin Tasarımına Yönelik Matlab Arayüzü Geliştirilmesi

**Kazim Dođuhan SELÇUK**

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

Devreler ve Sistemler

Temmuz 2020, Sayfa:xii +78

---

Kontrol sistemleri günümüzde en basit sistemlerden en karmaşık mühendislik problemlerine kadar her noktada karşımıza çıkmaktadır. Gündelik hayatla bu kadar iç içe olması kontrol sistemlerini popüler bir araştırma alanı haline getirmiştir. Günümüzde kontrol sistemlerinin tasarlanması ve denetimi çeşitli yöntemlerle yapılmaktadır. Kontrol edilecek sistemlerin çeşitliliği birçok denetleyici türünün literatüre kazandırılmasını sağlamıştır. Bu denetleyici türlerinden en popüler olan PID denetleyicisidir. Sistemin analiz kriterlerini en optimum düzeyde iyileştirmesi onun bu alanın en popüler denetleyicisi olmasına neden olmuştur. Ancak son yıllarda bilgisayar teknolojisinin gelişmesiyle birlikte matematiksel hesaplama gücünün çok artması, Kesir dereceli sistemlerin daha doğru ve etkin bir şekilde incelenmesine ve araştırmacıların bu konuya yoğunlaşmalarına neden olmuştur. Bu ilgiyle beraber kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicilerin klasik PID denetleyicilerden daha etkin olduğu ortaya konulmuştur. Bu çalışmada kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyiciler ile klasik PID denetleyiciler arasındaki farklar incelenmiş, kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici parametre belirleme yöntemleri anlatılmış ve transfer fonksiyonu bilinen bir sistem üzerinde kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicisi ile klasik PID denetleyici performansları karşılaştırılmıştır. Kesir dereceli ve tam sayı dereceli sistemlerin tasarım ve analizi için Matlab programında bir arayüz tasarlanmıştır.

**Anahtar Kelimeler:** PID denetleyici , Kesir Dereceli Sistemler , Kesir Dereceli Denetleyici



## ABSTRACT

---

### Development of Matlab Interface For Desing Of Fractional Order PID Controllers

Kazim Dođuhan SELÇUK

Master's Thesis

FIRAT UNIVERSITY

Graduate School of Natural and Applied Sciences

Department of Electrical Electronics Engineering

Circuits and System

July 2020, Pages: xii + 78

---

Nowadays, the control systems come us across every point from the simplest systems to the most complex engineering problems . Being so intertwined with everyday life has made control systems a popular field of research. Nowadays, control systems are designed and controlled by various methods. The variety of systems tabe controlled has led to the introduction of many types of controllers into the literature. The most popular of these controller types is the PID controller. Optimizing the analysis criteria of the system at the optimum level has made it the most popular controller in this field. However, with the development of computer technology in recent years, mathematical computing power has increased much. Also, it has led to more accurate and effective examination of fractional order systems and to focus researchers on this subject. Therefore, it is seen that the fractional order PID is more efficient than conventional PID controllers. In this study, the differences between these tho PID controllers are examined and the fractional order PID controller parameter determination methods are decsribed with a system whose transfer function is known, these two PID's are compared. An interface is designed in the Matlab program for the design and analysis of fractional and integer order system.

**Keywords:** PID Controller , Fractional Order System , Fractional Order Controller

## ŞEKİLLER LİSTESİ

|                                                                                     | Sayfa |
|-------------------------------------------------------------------------------------|-------|
| Şekil 2.1. Açık Çevrim Kontrol Sistemi .....                                        | 4     |
| Şekil 2.2. Kapalı Çevrim Kontrol Sistemi .....                                      | 4     |
| Şekil 2.3. On – Off Denetleyici Grafiği.....                                        | 5     |
| Şekil 2.4. Oransal Denetleyici Grafiği.....                                         | 6     |
| Şekil 2.5. Oransal Denetleyici Devresi.....                                         | 6     |
| Şekil 2.6. PI Denetleyici Grafiği .....                                             | 7     |
| Şekil 2.7. PI denetleyici Devresi .....                                             | 8     |
| Şekil 2.8. PD Denetleyici Grafiği.....                                              | 9     |
| Şekil 2.9. PD Denetleyici Devresi .....                                             | 9     |
| Şekil 2.10. PID Denetleyici Çıktısı .....                                           | 10    |
| Şekil 2.11. PID Denetleyici Bloğu İç Yapısı.....                                    | 11    |
| Şekil 2.12. Denetleyici Performans Parametreleri.....                               | 11    |
| Şekil 2.13. PID Denetleyici Devresi.....                                            | 12    |
| Şekil 2.14. Birim Basamak Giriş Cevap Eğrisi Grafiği .....                          | 14    |
| Şekil 2.15. Titreşim Yöntemi Grafiği.....                                           | 15    |
| Şekil 3.1. Kapalı Çevrim $PI^{\lambda}D^{\mu}$ Denetleyici.....                     | 23    |
| Şekil 3.2. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Bloğu İç Yapısı .....   | 23    |
| Şekil 3.3. $PI^{\lambda}D^{\mu}$ Denetleyici Kontrol Bölgesi.....                   | 24    |
| Şekil 5.1. $G(s)$ Fonksiyonun Kontrolsüz Cevabı .....                               | 28    |
| Şekil 5.2. $G(s)$ Fonksiyonun Klasik PID Denetleyici Cevabı.....                    | 29    |
| Şekil 5.3. Matlab/Simülink Blog Diyagramı .....                                     | 30    |
| Şekil 5.4. $PI^{\lambda}D^{\mu}$ Denetleyici İle Klasik PID Denetleyici Cevabı..... | 30    |
| Şekil 5.5. Klasik PID Denetleyici Max Aşma Değeri .....                             | 31    |
| Şekil 5.6. $PI^{\lambda}D^{\mu}$ Denetleyici Max Aşma Değeri.....                   | 31    |
| Şekil 5.7. Klasik PID Denetleyici Oturma Süresi.....                                | 32    |
| Şekil 5.8. $PI^{\lambda}D^{\mu}$ Denetleyici Oturma Süresi .....                    | 32    |
| Şekil 6.1. İsim Kapalı Çevrim Simülasyon Çıktısı .....                              | 38    |
| Şekil 6.2. Bode Grafiği Çıktısı .....                                               | 40    |
| Şekil 6.3. Nichols Eğrisi Çıktısı .....                                             | 41    |
| Şekil 6.4. Nyquist Eğrisi Çıktısı .....                                             | 41    |
| Şekil 6.5. Ziegler-Nichols Birim Basamak Cevap Eğrisi .....                         | 42    |
| Şekil 6.6. Kapalı Çevrim Birim Basamak Cevabı (Ziegler Nichols).....                | 47    |
| Şekil 6.7. Birim Basamak Cevabı (Ziegler Nichols) .....                             | 47    |
| Şekil 6.8. Ziegler-Nichols Birim Basamak Cevap Eğrisi .....                         | 48    |
| Şekil 6.9. Kapalı Çevrim Birim Basamak Cevabı (Cohen-Coon) .....                    | 51    |

|                    |                                                                                            |    |
|--------------------|--------------------------------------------------------------------------------------------|----|
| <b>Şekil 6.10.</b> | Birim Basamak Çıktısı (Cohen-Coon) .....                                                   | 51 |
| <b>Şekil 6.11.</b> | Kapalı Çevrim Birim Basamak Cevabı (Referans Regülasyonu).....                             | 54 |
| <b>Şekil 6.12.</b> | Birim Basamak Çıktısı (Referans Regülasyonu).....                                          | 55 |
| <b>Şekil 6.13.</b> | Kapalı Çevrim Birim Basamak Cevabı (Bozucu Etkiyi Elimine Etmek).....                      | 57 |
| <b>Şekil 6.14.</b> | Birim Basamak Çıktısı (Bozucu Etkiyi Elimine Etmek).....                                   | 57 |
| <b>Şekil 6.15.</b> | Kapalı Çevrim Birim Basamak Cevabı .....                                                   | 62 |
| <b>Şekil 6.16.</b> | Klasik PID Denetleyici Cevabı .....                                                        | 63 |
| <b>Şekil 6.17.</b> | Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Cevabı .....                              | 63 |
| <b>Şekil 6.18.</b> | Klasik PID ile Kesir Dereceli $PI^{\lambda}D^{\mu}$ Karşılaştırılması.....                 | 64 |
| <b>Şekil 7.1.</b>  | Arayüz Ekranı .....                                                                        | 67 |
| <b>Şekil 7.2.</b>  | Trasnsfer Fonksiyonunun Tanımlanması (Kesir Dereceli).....                                 | 68 |
| <b>Şekil 7.3.</b>  | Trasnsfer Fonksiyonunun Tanımlanması (Tam Sayı Dereceli).....                              | 69 |
| <b>Şekil 7.4.</b>  | Transfer Fonksiyonun Gösterilmesi .....                                                    | 69 |
| <b>Şekil 7.5.</b>  | Birim Basamak Giriş İçin Sistemin Kapalı Çevrim Cevabı .....                               | 70 |
| <b>Şekil 7.6.</b>  | $(T/L) < 4$ Durumu İçin Ekrana Gelen Hata Uyarısı.....                                     | 70 |
| <b>Şekil 7.7.</b>  | Denetleyici Parametrelerinin Tespit Edilmesi .....                                         | 71 |
| <b>Şekil 7.8.</b>  | $PI^{\lambda}D^{\mu}$ Denetleyici Parametrelerinin Tespit Edilmesi .....                   | 71 |
| <b>Şekil 7.9.</b>  | Denetleyici Transfer Fonksiyonunun Gösterilmesi .....                                      | 72 |
| <b>Şekil 7.10.</b> | Klasik PID Denetleyici Cevabının Elde Edilmesi .....                                       | 72 |
| <b>Şekil 7.11.</b> | Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Cevabının Elde Edilmesi.....              | 73 |
| <b>Şekil 7.12.</b> | Klasik PID ile Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyicinin Karşılaştırılması ..... | 73 |
| <b>Şekil 7.13.</b> | Klasik PID için Bode Analizi .....                                                         | 74 |
| <b>Şekil 7.14.</b> | Kesir Dereceli $PI^{\lambda}D^{\mu}$ için Bode Analizi.....                                | 74 |
| <b>Şekil 7.15.</b> | Klasik PID için Nichols Analizi .....                                                      | 75 |
| <b>Şekil 7.16.</b> | Kesir Dereceli $PI^{\lambda}D^{\mu}$ için Nichols Analizi.....                             | 75 |
| <b>Şekil 7.17.</b> | Klasik PID için Nyquist Analizi.....                                                       | 76 |
| <b>Şekil 7.18.</b> | Kesir Dereceli $PI^{\lambda}D^{\mu}$ için Nyquist Analizi .....                            | 76 |

## TABLÖLER LİSTESİ

|                                                                                     | Sayfa |
|-------------------------------------------------------------------------------------|-------|
| <b>Tablo 2.1.</b> Denetleyici Türelerinin Sistem Üzerindeki Etkileri.....           | 12    |
| <b>Tablo 2.2.</b> Ziegler – Nichols Yöntemi (Açık Çevrim Yöntemi) .....             | 14    |
| <b>Tablo 2.3.</b> Ziegler – Nichols Yöntemi (Frekans Cevabı Yöntemi) .....          | 15    |
| <b>Tablo 2.4.</b> Cohen – Coon Yöntemi .....                                        | 17    |
| <b>Tablo 2.5.</b> Chien – Hrones – Reswiche Yöntemi (Referans Regülasyonu) .....    | 17    |
| <b>Tablo 2.6.</b> Chien – Hrones – Reswiche Yöntemi (Bozucu Etkiyi Durdurmak) ..... | 18    |
| <b>Tablo 6.1.</b> Komut Dizileri ve Kullanım Amaçları.....                          | 64    |

## SEMBOLLER LİSTESİ

---

|                                    |                                          |
|------------------------------------|------------------------------------------|
| <b>P</b>                           | : Oransal Denetleyici                    |
| <b>PI</b>                          | : Oransal + İntegral Denetleyici         |
| <b>PD</b>                          | : Oransal + Türev Denetleyici            |
| <b>PID</b>                         | : Oransal + İntegral + Türev Denetleyici |
| <b>PI<sup>λ</sup>D<sup>μ</sup></b> | : Kesir Dereceli PID denetleyici         |
| <b>λ</b>                           | : İntegral Katsayısı                     |
| <b>μ</b>                           | : Türev Katsayısı                        |
| <b>E(t)</b>                        | : Hata                                   |
| <b>K<sub>p</sub></b>               | : Oransal Kazanç                         |
| <b>K<sub>i</sub></b>               | : İntegral Kazanç                        |
| <b>K<sub>d</sub></b>               | : Türevsel Kazanç                        |
| <b>K<sub>u</sub></b>               | : Maximum Oransal Kazanç                 |
| <b>L</b>                           | : Zaman Gecikmesi                        |
| <b>m<sub>p</sub></b>               | : Maximum Aşma                           |
| <b>T</b>                           | : Zaman Sabiti                           |
| <b>T<sub>d</sub></b>               | : Türevsel Zaman Sabiti                  |
| <b>T<sub>i</sub></b>               | : İntegratör Zaman Sabiti                |
| <b>t<sub>s</sub></b>               | : Oturma Süresi                          |
| <b>P<sub>u</sub></b>               | : Sinüzoidal Dalga Periyodu              |
| <b>U(t)</b>                        | : Çıkış                                  |
| <b>ζ<sub>m</sub></b>               | : Sönüm Frekansı                         |
| <b>W<sub>cp</sub></b>              | : Kritik Frekans                         |
| <b>Ø<sub>m</sub></b>               | : Faz Payı                               |

# 1. GİRİŞ

Son yıllarda gerçekleşen teknolojik gelişmelerle birlikte kontrol sistemleri önemli bir çalışma alanı haline gelmiştir. Günlük yaşıntıdaki en basit sistemlerden endüstrideki en karmaşık problemlere kadar karşımıza çıkması bu ilginin artmasındaki en önemli etkindir.

Kontrol sistemleri en temelde açık çevrim kontrol ve kapalı çevrim kontrol olmak üzere iki gruba ayrılır. Açık çevrim kontrol sistemleri denetleyici ve denetlenen sistem olmak üzere iki birimden oluşur. Çıkış değerinin giriş değerine herhangi bir etkisinin olmaması sistemin dezavantajıdır. Kapalı çevrim kontrol sistemleri denetleyici, denetlenen ve geri besleme olmak üzere üç bileşenden meydana gelir. Giriş işareti, çıkış işareti veya çıkışın ürettiği başka bir referans işaret ile toplama veya çıkarma işlemi gerçekleştirilerek geri beslemenin elde edildiği bir kontrol sistemidir. Kapalı çevrim kontrol sisteminin en önemli avantajı olan bu geri besleme sistemi kararlılık gibi tasarımın en önemli bileşeni üzerinde çok önemli bir payı vardır.

Kontrol sistemlerinden en önemli beklenti giriş şartlarının değişmesi sonucunda çıkışın bu girişe en kısa sürede ve kararlı bir şekilde yanıt vermesidir. Sistemin tasarım kriterlerine uygun bir denetleyicinin seçimi çok önemlidir. Bu sebeple denetleyici seçimi yapılırken sistem derecesine ve kontrol kriterlerine uygun en ekonomik denetleyici seçilmelidir.

Kontrol sistemlerinde denetleyiciler en temelde klasik ve sezgisel denetleyiciler olarak ikiye ayrılırlar. Klasik denetleyiciler on-off kontrol, P, PI, PD, PID gibi türleri bulunmaktadır. Günümüzde genellikle klasik denetleyicilerden PID denetleyicisi kullanılmaktadır. Sezgisel denetleyiciler genetik algoritmalar, bulanık mantık, yapay sinir ağları, kayan kipli kontrol, anfis denetleyici gibi denetleyici türleri de bulunmaktadır.

Kontrol sistemlerinde denetlenen sistemin derecesine, istenilen çıkış değerinin seçimine göre sistem tasarımı değişmektedir. Değişen sisteme göre klasik ve sezgisel denetleyici tiplerinden biri yada birkaçı kullanılabilir.

Son yıllara kadar hesaplama imkanlarının yetersizliği nedeni ile kesir dereceli sistemler ve denetleyiciler üzerinde çok durulmamıştır. Bilgisayar teknolojisinin gelişmesi ile birlikte son 25-30 yılda kesir dereceli sistemler ve denetleyiciler üzerinde bir çok çalışma gerçekleştirilmiştir.

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicisi Podlupny'nin çalışmalarında sunulmuştur. Bu denetleyicide  $\lambda$  integralin derecesini  $\mu$  ise türevin derecesini göstermektedir. Bu denetleyici parametreleri reel sayı olduğu için PID denetleyicisinin tamsayı değerlerini de içinde barındırmaktadır. Bu nedenle kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici hem kesir dereceli sistemlere hem tam sayı dereceli sistemlere uygulanabilmekte ve daha iyi sonuçlar ortaya koymaktadır.

## 1.1.Literatür Taraması

Türev mertebeleri herhangi bir reel sayı olabilen sistemlere kesir dereceli sistemler denir. Klasik tam sayı dereceli matematiğin kesir dereceli matematiğe uyarlanması yeni değildir. Kesir dereceli matematiğin ilk kullanımı 1695’de Leibniz ile L’Hospital arasındaki mektuplaşmalarda görülür. Daha sonra kesirli matematik ile ilgili ilk çalışmaların 19.yy da Liouville, Riemann ve Holmgren tarafından yapıldığı görülür.[1]

Yakın geçmişte bazı araştırmalar (Bagley ve Torvik (1984); Bagley ve Calico (1991); Makroglou, Miller ve Skaler (1994)) kesir dereceli türev ve integraller içeren kesir dereceli [2] durum denklemlerini dikkate almışlardır.Bu yeni modeller, daha önce kullanılan tam sayı dereceli modellerden daha elverişlidir. Kesir tabanlı modellerin temellerini Caputo ve Mainardi (1971) ve Westerlund (1994) tarafından verilmiştir.Ancak kontrol sistemlerinin tasarımı ve uygulanması noktasında yapılan çalışmalar matematiksel metotların ve bilgisayar teknolojilerinin eksikliği neticesinde çok az yer tutmuştur. Oustaloup (1988), Axtell ve Michael (1990), Miller ve Skaar (1994), tarafından bazı başarılı teşebbüsler olmuştur[1]. Kesir dereceli sistemler ve kesir dereceli denetleyiciler 21.yy kadar kontrol problemlerinin çözümlerinde fazla uygulanmamıştır.[1] Bilgisayar teknolojisinin gelişmesiyle birlikte hesaplama gücü artmış ve kesir dereceli sistemler sistem modelinin analizinde ve denetleyici tasarımlarında kullanılmaya başlanmıştır.

Podlupny (1999) , yaptığı çalışmalarda PID denetleyicisindeki tamsayı dereceli türev ve integral ifadelerinin reel sayı dereceli modelleri üzerinde çalışmış ve  $PI^aD^b$  olarak ifade ettiğimizi kesir dereceli PID denetleyicisini oluşturmuştur.[3] Petras (2000) yaptığı çalışmalarda baskın kökler yöntemini temel alan kesir dereceli PD denetleyici geliştirmiştir.[4] Klasik PD ile kesir dereceli PD denetleyiciyi karşılaştırmıştır. Kesir dereceli PD ‘nin oturma süresinin daha hızlı olduğunu ve salınımın azaldığını görmüştür. Monje et al (2008) yaptığı çalışmalarda faz geçiş frekansı ve faz payı değerlerini esas alan bir kesir dereceli PID denetleyici tasarımı yapmıştır. Petras (2009) yaptığı çalışmalarda DC bir motorun kesir dereceli modelini esas almış ve Matlab/Simulink programında simülasyon gerçekleştirmiştir.[5] Luo et al (2010) yaptığı çalışmalarda kesir dereceli sistemler için kesir dereceli PI geliştirmiş , yaptığı bu çalışmada kesir dereceli PI denetleyicisinin klasik PID denetleyicisinden daha elverişli olduğunu deneysel olarak ispatlamıştır. Yeroğlu (2011) yaptığı çalışmalarda kesir dereceli PID denetleyicisindeki parametrelerin tespit edilmesi üzerine tasarım yöntemi sunmuştur.[6] Türk, Atan, Tuntaş (2013) yaptığı çalışmada kesir dereceli PID denetleyicisinin elektronik devre elemanlarıyla tasarlanabildiğini ve başarılı sonuç verdiğini göstermiştir.[7]

Kesir dereceli PID denetleyicilerinin simülasyonunun gerçekleştirilebilmesi için Matlab/Simulink programında kullanılabilecek toolboxlar geliştirilmiştir. Oustaloup (1991) yaptığı çalışmada CroneToolbox’ı geliştirmiştir. Bu toolbox kesir dereceli sistemlerin kontrolü

için geliştirilmiş ve daha sonra MATLAB ortamında kullanıma sunulmuştur. DuarteValerio (2000) İlk olarak 2000 yılında Portekizce bir toolbox geliştirmiş ve son hali 2005 yılında “Toolboxnintegerfor MATLAB v.2.3 “ adıyla yayınlanmıştır. Yeroğlu, Şenol, (2011) yaptığı çalışmada hem tam sayı dereceli hem de kesir dereceli kontrol sistemlerinin BodeNyquist ve Nichols grafiklerinin kolayca çizilebildiği bir arayüz geliştirmişlerdir. Tepljakov (2015) yaptığı çalışmada kesir dereceli sistemlerin kontrolü için MATLAB programında “ FOMCOM “ toolbox’ını geliştirmiştir.[8]

Bu tez çalışmasında ;

İkinci bölümde P, PI, PD ve klasik PID sistemler anlatılmıştır. Klasik PID sistemler için parametre belirleme yöntemlerinden bahsedilmiştir.

Üçüncü bölümde kesir dereceli sistemlerin ve denetleyicilerin tanımı matematiksel ifadeleri ve klasik PID denetleyiciler ile kesir dereceli PID denetleyiciler arasındaki farklar anlatılmıştır.

Dördüncü bölümde ise kesir dereceli PID denetleyicilerin  $K_p$ ,  $K_i$ ,  $K_d$ , parametrelerinin ve integral ile türevin derecelerinin ( $\lambda$  ve  $\mu$ ) tespit edilme yöntemleri verilmiştir.

Beşinci bölümde ise transfer fonksiyonu bilinen bir sistem için klasik PID denetleyici parametreleri ve kesir dereceli PID denetleyici parametreleri tespit edilmiş ve FOMCOM toolbox’ı kullanılarak MATLAB programında simülasyon gerçekleştirilerek sonuçlar karşılaştırılmıştır.

Altıncı bölümde kesir dereceli ve tam sayı dereceli sistemler için arayüz geliştirme aşamasında Matlab programı için gerekli alt programlar hazırlanmış ve bunların simülasyonları yapılmıştır.

Yedinci bölümde geliştirilen alt programlar ile kesir dereceli sistemler ve kesir dereceli  $PI^{\lambda}D^{\mu}$  denleyiciler için tasarım ve simülasyonun yapılabileceği bir arayüz tasarlanmıştır.



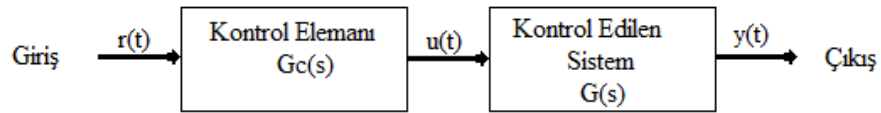
## 2. KONTROL ELEMANLARI VE KONTROL SİSTEMLERİ

Kontrol sistemlerinin temel çalışma prensibi kontrol edilen istemin çıkışından ölçülen çıkış değeri ile giriş değerinin karşılaştırılması sonucu hata ( $e(t)$ ) değeri elde edilmesi üzerinedir. Bu hata değeri denetleyici girişine uygulanır. Denetleyici bu hata sinyalini işleyerek kontrol edilen sistemin girişine uygular.

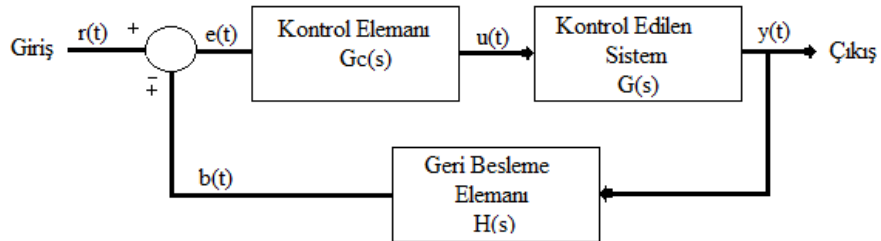
Kontrol sistemlerinin türüne derecesine ve kontrol kriterlerine göre çeşitli denetleyici tipleri geliştirilmiştir. Bu denetleyicilerden bazıları;

1. Açık-kapalı (on-off) denetleyici
2. Oransal denetleyici (P)
3. Oransal+integral denetleyici (PI)
4. Oransal+türev denetleyici (PD)
5. Oransal+türev+integral denetleyici (PID)

En temelde kontrol sistemleri açık çevrim ve kapalı çevrim kontrol sistemleri olarak ikiye ayrılırlar. Kontrol edilecek sistemin kontrol kriterleri açısından gerekliliklerine göre değişim gösterebilir. çoğunlukla kapalı çevrim sistemler kullanılmaktadır. Kapalı çevrim sistemlerin tercih edilmesindeki en önemli sebep sistemin geri beslemeye sahip olmasıyla çıkışın giriş sinyalini daha iyi takip edebilmesidir. Bu iki kontrol sisteminin farklılıkları blok diyagramlarına bakılarak daha iyi anlaşılabilir. Aşağıda Şekil 2.1 açık çevrim kontrol sistemlerinin blok diyagramı Şekil 2.2 ise kapalı çevrim kontrol sistemlerinin blok diyagramıdır.



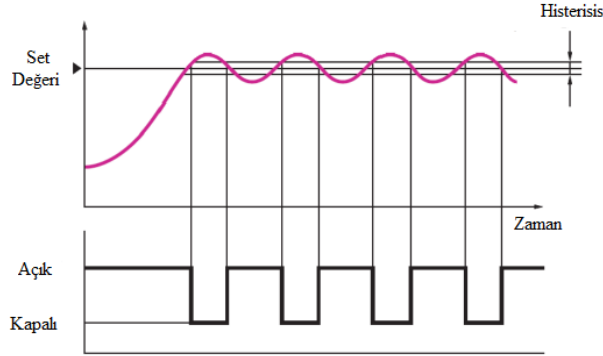
Şekil 2.1 Açık Çevrim Kontrol Sistemi



Şekil 2.2 Kapalı Çevrim Kontrol Sistemi

## 2.1. Açık-Kapalı (on-off) Denetleyici

Açık-kapalı (on-off) kontrol denetleyicinin açık veya kapalı olma prensibine dayanan bir kontrol türüdür. Denetleyiciye girilen referans değer ile çıkışta ölçülen değer karşılaştırılarak bir hata sinyali elde edilir. Bu hata sinyali referans değerinin altında veya üstünde olması durumuna göre denetleyici ya tamamen açıktır ya da tamamen kapalıdır. (Şekil 2.3 )



Şekil 2.3 On – Off Denetleyici Grafiği

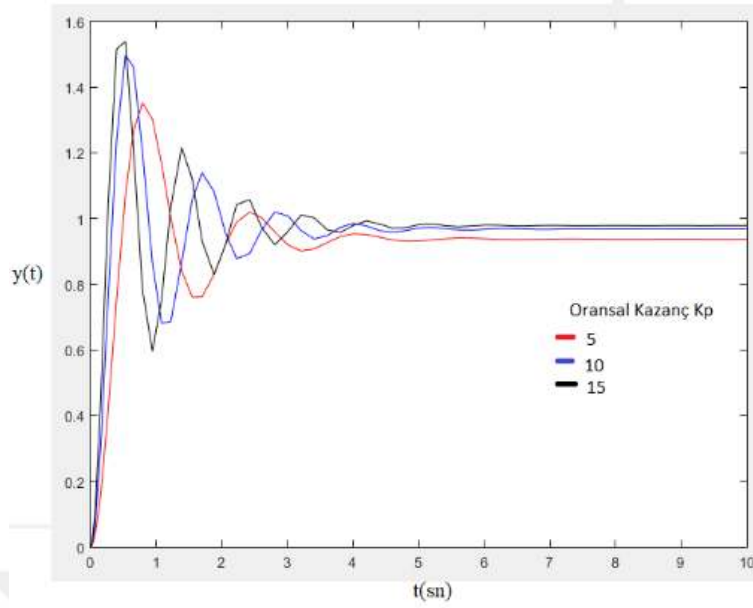
Açık-kapalı/on-off kontrol sisteminde kalıcı bir hata değeri oluşur. Bu hata değerinin minimuma indirilmesi için on-off işleminin çok kısa sürelerde gerçekleştirilmesi gerekir. Bu durum denetleyicinin çıkışını zorlamakta ve ömrünü kısaltmaktadır.

## 2.2. Oransal Denetleyici (P)

Oransal kontrol sistemin hata ( $e(t)$ ) değeri ve denetleyicinin ürettiği  $K_p$  katsayısı ile gerçekleştirilen bir kontrol sistemidir. Oransal kontrol sistemin kalıcı durum hatasını azaltır ve dinamik cevabını değiştirir.

$$u(t) = K_p \cdot e(t) \quad (2.1)$$

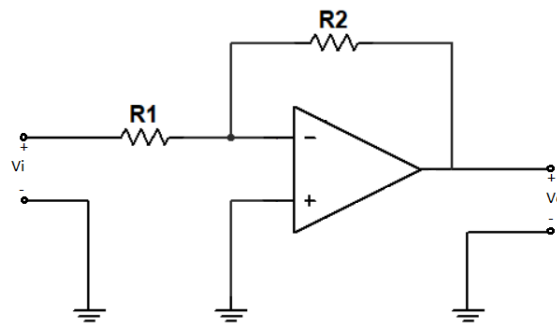
Denetleyici tarafından üretilen kontrol sinyalinin etkisi, hata değeri azaldıkça kontrol etkisi düşmektedir. Bu durum daha az salınımla referans değerinin yakalanması sağlar. Fakat hata küçüldükçe kontrol etkisi çok zayıfladığı için çıkış değeri referans değeri asla tam olarak yakalayamaz. (Şekil 2.4,kapalı çevrim birim basamak cevabı,Şekil2.2)



**Şekil 2.4** Oransal Denetleyici Grafiği

Yani oransal kontrol kalıcı durumda kararlı çalışmasına rağmen kalıcı durum hatası daima mevcuttur. Bu daimi olan kalıcı durum hatası  $K_p$  parametresinin artırılması ile minimize edilebilir. Ancak dikkat edilmesi gereken en önemli nokta  $K_p$  değeri çok büyükse sistem on-off kontrol kullanılan sisteme çok benzer. Bu durum sistem salınımlarını artırır ve kararsızlığa götürebilir. Bu nedenle oransal denetleyiciler birçok sistem için tek başına yeterli değildir.

Elektronik devre elemanlarıyla gerçekleştirilmiş oransal kontrole ait devre Şekil 2.5 deki gibidir. Devrenin matematiksel ifadesi Denklem 2.2 de verilmiştir



**Şekil 2.5** Oransal Denetleyici Devresi

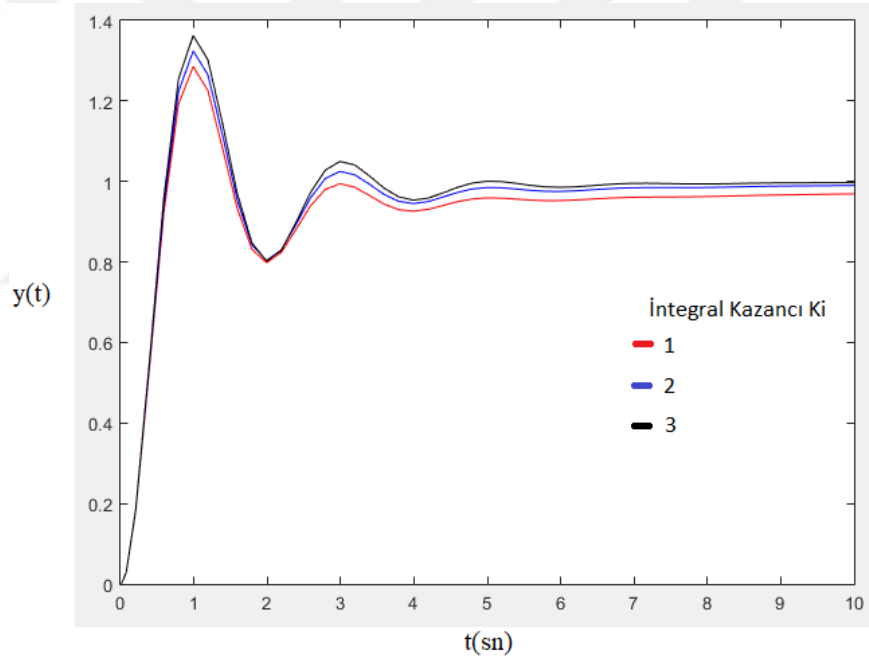
$$K_p = \frac{R_2}{R_1} \quad (2.2)$$

### 2.3.Oransal + İntegral Denetleyici (PI)

Oransal + integral kontrol, oransal denetleyici ile integral denetleyicisinin birleştirilmesi ile meydana gelir. Denetleyici iki parametreyi kullanarak sisteme müdahale eder. Bu parametreler oransal kontrolden gelen  $K_p$  ve integral kontrolden gelen  $K_i$  parametreleridir. (Denklemler 2.3)

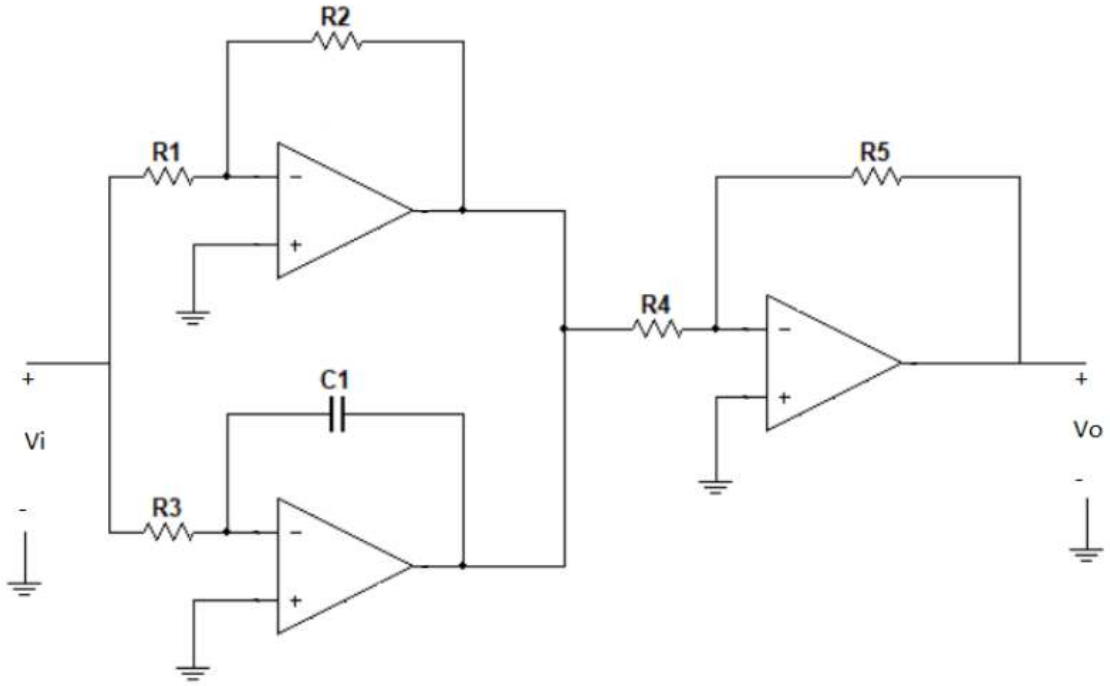
$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) \cdot dt \quad (2.3)$$

İntegral denetleyicinin görevi oransal denetleyicinin tek başına kullanılmasında giderilemeyen kalıcı durum hatasının ortadan kaldırılmasıdır. (Şekil 2.6, kapalı çevrim birim basamak cevabı, Şekil 2.2) PI denetleyici kalıcı durum hatasını ortadan kaldırırken aşma değerini ve oturma süresini arttırmaktadır. Bu nedenle integral kontrolden gelen  $K_i$  değeri çok yükseltirse sistem kararsızlığa gidebilir.



Şekil 2.6 PI Denetleyici Grafiği

Elektronik devre elemanlarıyla gerçekleştirilmiş PI kontrolöre ait devre aşağıdaki gibidir. (Şekil 2.7) Devrenin matematiksel ifadesi Denklem 2.4 ve Denklem 2.5'de verilmiştir



Şekil 2.7 PI denetleyici Devresi

$$K_p = \frac{R_2}{R_1} \quad (2.4)$$

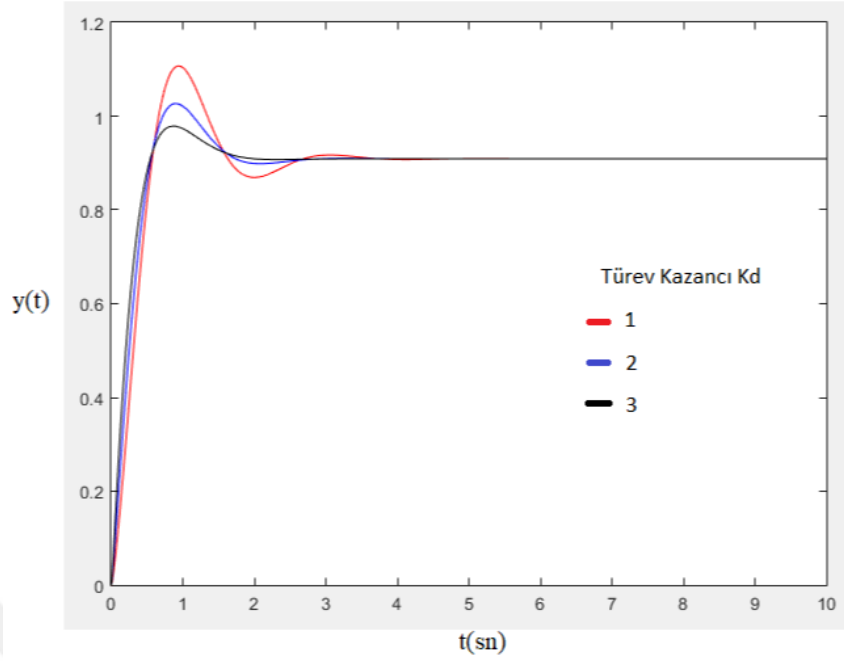
$$K_i = \frac{1}{R_3 C_1} \quad (2.5)$$

## 2.4.Oransal + Türev Denetletici (PD)

Oransal + Türev (PD) kontrol, denetlenen sistemi oransal kontrolden alınan  $K_p$  ve türev kontrolden alınan  $K_d$  değeri ile kontrol eder.(Denklem 2.6) Türev denetleyicisi hızlı ve kısa süreli değişiklikleri oluşan gürültüleri ve çıkış değişimlerini algılar.

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt} \quad (2.6)$$

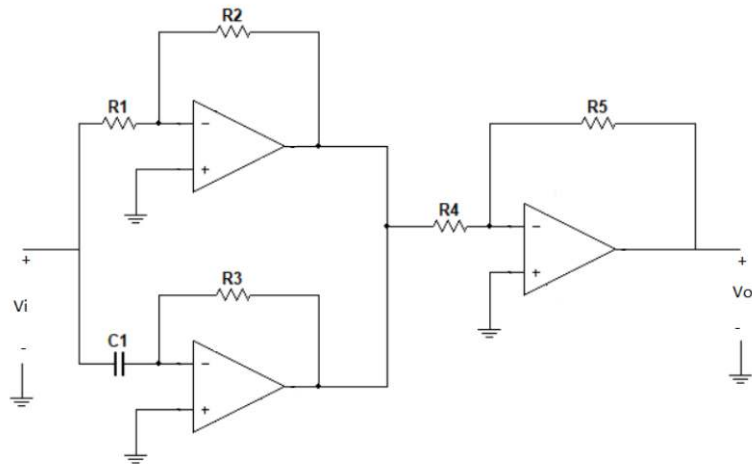
Türev kontrol sistemin çıkışında türev alma işlemi yapar. Türev işlevi sistemin bir sonraki çıkışta alacağı değere ilişkin bir veri üretir. Böylece sisteme bir öngörü kazandırılmış olur. Bu özelliği sayesinde çıkış değerinin aşımına uğramadan kontrol edilmesini sağlar.(Şekil 2.8, kapalı çevrim birim basamak cevabı, Şekil2.2) Türev denetleyicisinin bu yeteneği sistemin kalıcı durumda daha kararlı çalışmasına yol açar. Ancak sistemde hala tam olarak ortadan kaldırılamayan bir kalıcı durum hatası mevcuttur. Bunun sebebi bir sabitin türevi sıfır olduğundan türev denetleyicinin sabit hatalar üzerinde etkisinin olmamasıdır.



**Şekil 2.8** PD Denetleyici Grafiği

Türev denetleyicisi parametresi çok yüksek seçildiğinde sistemi kararsızlığa götürebilir. Bu sebeple  $K_d$  değeri belirlenirken dikkat edilmelidir. Pratikte türev denetleyiciler bozucu etkilere karşı çok hassas oldukları için diğer denetleyicilerle birlikte kullanılırlar.

Elektronik devre elemanlarıyla gerçekleştirilmiş PD kontrolöre ait devre aşağıdaki gibidir.(Şekil 2.9). Devrenin matematiksel ifadesi Denklem 2.7 ve Denklem 2.8’de verilmiştir



**Şekil 2.9** PD Denetleyici Devresi

$$K_p = \frac{R_2}{R_1} \quad (2.7)$$

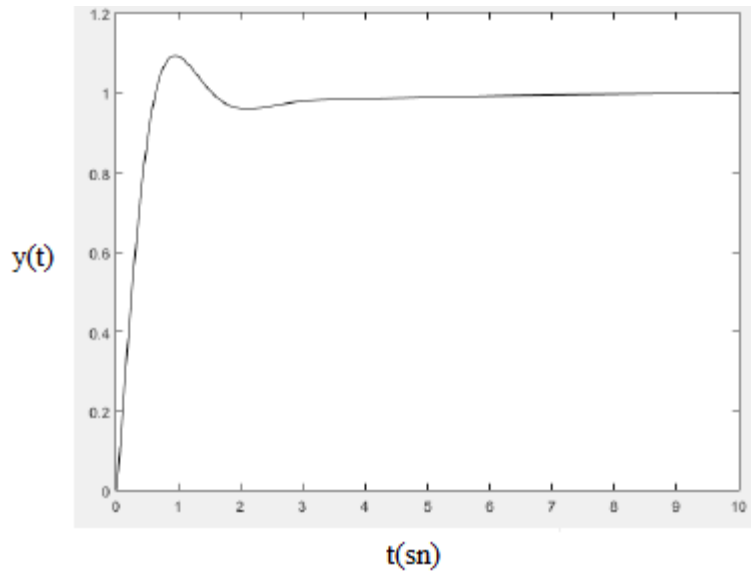
$$K_d = R_3 \cdot C_1 \quad (2.8)$$

## 2.5 Oransal + İntegral + Türev Denetleyici (PID)

Oransal + integral + türev (PID) kontrol denetlenen sistemi oransal kontrolden alınan  $K_p$ , integral kontrolden alınan  $K_i$  ve türev kontrolden alınan  $K_d$  değerleri ile kontrol eder. PI denetleyici ve PD denetleyicinin sistem kontrolüne olan katkılarının tek bir denetleyici üzerinde bir araya gelmesi PID denetleyicinin endüstride en yaygın kullanılan denetleyici olmasındaki en önemli etkidir.

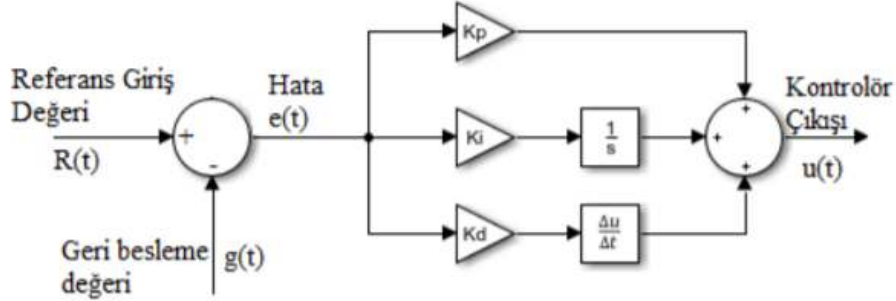
$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t) \cdot dt + K_d \frac{de(t)}{dt} \quad (2.9)$$

PID denetleyiciler kontrolün zor ve karmaşık olduğu sistemlerde tercih edilir. PI ve PD denetleyici denetlenen sistemin farklı noktalarını iyileştirmektedir. Ancak bunun yanında PI ve PD denetleyiciler sistem kontrolünde tek başlarına kullanıldıklarında sistemde bozucu etkiler de bulunmaktadır. Bu denetleyicilerin bir arada kullanılması kontrol edilecek sistemin tüm kontrol parametrelerini iyileştirmektedir. Ayrıca tek başlarına kullanımda ortaya çıkan bozucu etkiler de ortadan kalkmaktadır. PID denetleyicinin kontrol edilecek sisteme kontrol kalitesi açısından en önemli katkıları kalıcı durum hatasını ortadan kaldırmayı ve türev denetleyici sayesinde sistemin bir sonraki çıkışının tahmin edilebilirliğini aynı kontrol işlemi üzerinde gerçekleştirebilmesidir. (Şekil 2.10)( kapalı çevrim birim basamak cevabı,Şekil2.2)



Şekil 2.10 PID Denetleyici Grafiği

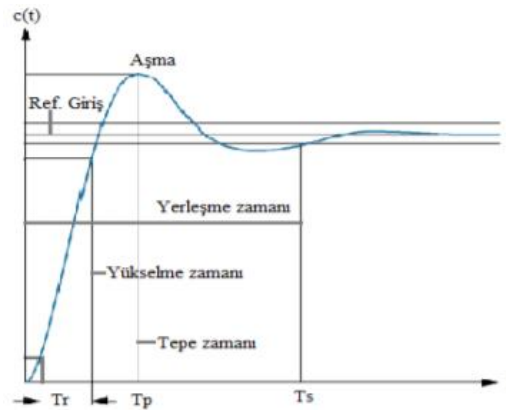
Elde edilen hata ( $e(t)$ ) oransal denetleyici, integral denetleyici ve türev denetleyiciden geçirilerek bu üç denetleyicinin çıkışında toplanır.(Şekil 2.11)



Şekil 2.11 PID Denetleyici Bloğu İç Yapısı

Herhangi bir denetleyicinin basamak cevabı için kontrol kalitesi ve performansının belirlenmesi için bazı parametreler bulunmaktadır.(Şekil 2.12)

1. Tepe Zamanı: Sistem cevabının tepe değerine ilk ulaştığı ana kadar geçen süreyi ifade eder.
2. Yerleşme Zamanı: Sistem cevabının giriş sinyalinin (%2'lik bant) yüzde 98'ine ulaştığı ana kadar geçen süre olarak tanımlanır.
3. Aşım: Sistem cevabının ulaştığı maximum nokta ile referans değeri arasındaki fark olarak tanımlanır.
4. Yükselme Zamanı: Sistem cevabının referans girişin yüzde 10'undan yüzde 90 olana kadar geçen süre olarak tanımlanır.
5. Kalıcı Durum Hatası: Sistem cevabı ile referans giriş arasında kalıcı durumda oluşan hatayı tanımlamak için kullanılır.



Şekil 2.12 Denetleyici Performans Parametreleri

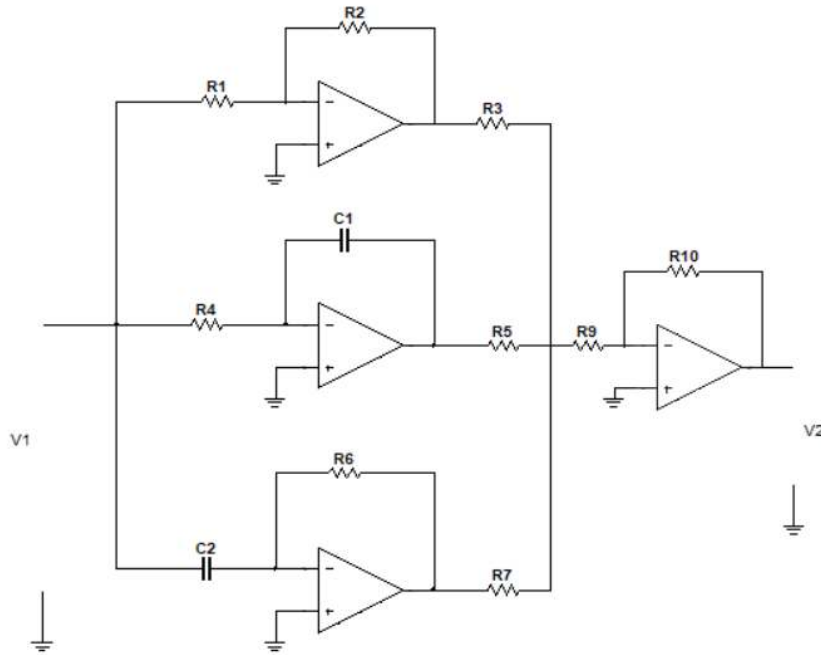


Denetleyici türlerinin denetlenecek sistem üzerindeki etkileri Tablo 2.1’ de gösterilmiştir. PID denetleyici tasarımı yapmadan önce tablonun incelenmesi denetleyici tasarım sürecine katkı sağlayacaktır.

**Tablo 2.1** Denetleyici Türelerinin Sistem Üzerindeki Etkileri

| Kapalı Çevrim Cevabı | Yükselme Zamanı      | Maximum Aşma | Yerleşme Zamanı      | Kalıcı Durum Hatası  |
|----------------------|----------------------|--------------|----------------------|----------------------|
| $K_p$                | Azalır               | Artar        | Küçük Oranda Değişir | Azalır               |
| $K_i$                | Azalır               | Artar        | Artar                | Ortadan Kalkar       |
| $K_d$                | Küçük Oranda Değişir | Azalır       | Azalır               | Küçük Oranda Değişir |

Elektronik devre elemanlarıyla gerçekleştirilmiş PID denetleyiciye ait devre aşağıdaki gibidir.(Şekil 2.13). Devrenin matematiksel ifadesi Denklem 2.10, Denklem 2.11, Denklem 2.12, Denklem 2.13 ve Denklem 2.14’de verilmiştir.



**Şekil 2.13** PID Denetleyici Devresi

$$K_p = \frac{R_2}{R_1} \quad (2.10)$$

$$K_i = \frac{1}{R_4 C_1} \quad (2.11)$$

$$K_d = R_6 \cdot C_2 \quad (2.12)$$

$$V_2 = K_p \cdot V_1 + K_i \int V_1 \cdot dt + K_d \frac{dV_1}{dt} \quad (2.13)$$

$$\frac{V_2}{V_1} = K_p + \frac{K_i}{s} + K_d s \quad (2.14)$$

PID denetleyicinin gerçek zamanlı tasarımı için oransal, integral ve türevsel devreler birleştirilmiştir.

## 2.6. PID Parametrelerini Belirleme Yöntemleri

Endüstriyel kontrol sistemlerinde en çok kullanılan denetleyici türü olan PID denetleyicinin parametrelerinin uygun bir şekilde belirlenmesi sistemden alınması beklenen performans açısından çok önemlidir. Bu sebeple tasarım aşamasında  $K_p$ ,  $K_i$ ,  $K_d$  parametrelerinin en uygun şekilde seçilmesi hedeflenmektedir.

PID denetleyici parametrelerinin belirlenmesi için klasik yöntemler ile birlikte günümüzde yapay zeka tabanlı sezgisel yöntemler de kullanılmaktadır.

### 2.6.1. Klasik Yöntemler

PID denetleyici parametrelerinin belirlenmesinde kullanılan klasik yöntemler Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswick, Åström-Hägglund olarak sıralanabilir [2].

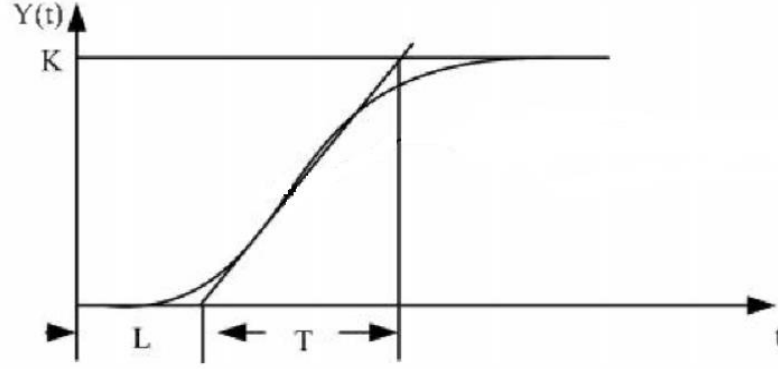
#### Ziegler-Nichols Yöntemi

PID denetleyici parametrelerinin bulunmasında kullanılan Ziegler-Nichols yöntemi geniş bir kullanım alanına sahiptir. Bu yöntemin kullanımı kolay olması bozucu etkiler karşı kararlı olması ve modellemeye ihtiyaç duymaması gibi avantajları bulunmaktadır. Ziegler-Nichols yöntemi birim basamak cevabı yöntemi ve frekans cevabı yöntemi olmak üzere ikiye ayrılır [2].

#### Açık Çevrim Yöntemi:

Literatürde açık çevrim yöntemi, birim basamak yöntemi, süreç eğrisi yöntemi gibi farklı adlandırmalarla geçmektedir.

PID denetleyici parametreleri kontrol edilecek sistemin açık çevrim durumunda birim basamak (K) cevabının elde edilip analiz edilmesiyle belirlenir. Yöntem oldukça hızlı ve kolaydır. Açık çevrim birim basamak cevabından elde edilmesi gereken veriler zaman gecikmesi (L) ve zaman sabiti (T) değerleridir.(Şekil 2.14)



Şekil 2.14 Birim Giriş Cevap Eğrisi Grafiği

Bu yöntemde Tablo 2.2’de verilen ampirik formüller ile hesaplanan PID denetleyici parametreleri daha sonra sistemin kapalı çevrimi için teste tabi tutulur. Yöntemin hedeflediği  $1/4$  genlikler oranı sağlandıysa en uygun parametre değeri bulunmuş demektir. Eğer genlikler oranı sağlanmazsa süreç tekrarlanır.

Tablo 2.2 Ziegler – Nichols Yöntemi ( Açık Çevrim Yöntemi )

| Denetleyici Tipi | $K_p$               | $T_i$           | $T_d$     |
|------------------|---------------------|-----------------|-----------|
| P                | $\frac{T}{L}$       | -               | -         |
| PI               | $0.9 * \frac{T}{L}$ | $\frac{L}{0.3}$ | -         |
| PID              | $1.2 * \frac{T}{L}$ | $2. L$          | $0.5 * L$ |

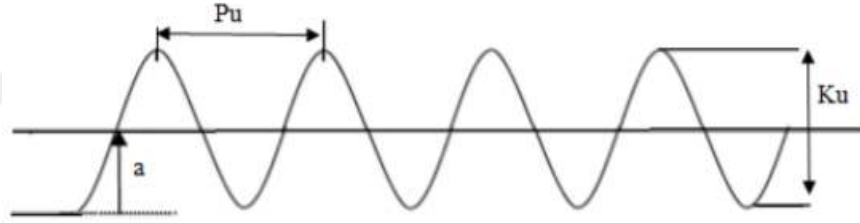
Parametreleri Ziegler-Nichols yöntemine göre belirlenen PID denetleyicisi  $L/T$  oranına çok duyarlıdır. Bu oran arttıkça Ziegler-Nichols yöntemine göre PID denetleyici parametrelerini

yakalamak zorlaşır. Burada  $L/T \geq 4$  olduğunda Ziegler-Nichols parametre belirleme yöntemi yetersiz kalmaktadır.

### Frekans Cevabı Yöntemi

Literatürde frekans cavabı yöntemi, salınım ve titreşim yöntemi, osilasyon yöntemi gibi farklı adlandırılmalarla geçmektedir.

Yöntemin temel prensibi integral ve türev katsayılarının devre dışı bırakılarak oransal katsayının sistemi osilasyona sokması sağlanır. Bu osilasyona sokma işlemi önce integratör zaman sabiti ( $T_i$ ) sonsuza ve türev zaman sabiti ( $T_d$ ) sıfıra çekilir. Böylece integral ve türevetkisi sıfır olur. Sistemde sadece oransal ( $K_p$ ) katsayı aktif olarak kalır. Daha sonra sisteme birim basamak girişi uygulanır ve oransal ( $K_p$ ) katsayı sürekli arttırılır. Sonuç olarak sistem salınıma girer. Oransal katsayının artırılması işlemi sistem kararsızlığa gitmeden hemen önceki değere kadar sürdürülür. Bu noktada elde edilen oransal ( $K_p$ ) katsayıya  $K_u$ , salınımın periyoduna ise  $P_u$  denir.(Şekil 2.15). Bu değerlere göre  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri Tablo 2.3'e göre belirlenir.



Şekil 2.15 Titreşim Yöntemi Grafiği

Tablo 2.3 Ziegler – Nichols Yöntemi ( Frekans Cevabı Yöntemi )

| Denetleyici Tipi | $K_p$        | $T_i$         | $T_d$         |
|------------------|--------------|---------------|---------------|
| P                | $0.5 * K_u$  | -             | -             |
| PI               | $0.45 * K_u$ | $0.825 * P_u$ | -             |
| PID              | $0.6 * K_u$  | $0.5 * P_u$   | $0.125 * P_u$ |

Frekans cevabı yönteminin avantajları uygulamanın basit olması ve tek bir parametre kontrolü ile elde edilen verilerin kullanılarak eksik parametre değerlerinin tespit edilmesi sayılabilir. Ancak dezavantajı ise test işleminin uzun sürmesi ve oransal kazancın sürekli artması sonucu artan bir salınımına sokulması sonucu sistemin zarar görebilme ihtimali sayılabilir.

Bazı süreç denetim sistemlerinde bu osilasyon işlemi çok sakıncalı olabilir. Bu durumda frekans cevabı yöntemi dışında bir yönteme ihtiyaç duyulmuştur. Frekans cevabı yönteminin bu dezavantajından dolayı yüksek ve uzun süreli osilasyonun kabul edilemez olduğu süreç denetim sistemlerinde bu yöntemin Harriott tarafından düzenlenmiş hali tercih edilmelidir.

Bu yöntemde de yine oransal ( $K_p$ ) katsayı kullanılmaktadır.  $K_p$  katsayısı küçük bir değere ayarlanır. Sistem cevap eğrisinde  $1/4$  genlik oranı elde edilinceye kadar  $K_p$  katsayısı arttırılır. Bu genlik oranındaki sinüzoidal dalganın periyodu  $P_u$ 'ya bağlı olarak integral ve türev zaman sabitleri Denklem 2.15 ve Denklem 2.16'ya göre hesaplanır.

$$T_i = P_u / 6 \quad (2.15)$$

$$T_d = P_u / 15 \quad (2.16)$$

Denetim organı üzerinde  $T_i$  ve  $T_d$  yukarıda bulunan formüllerle elde edildikten sonra cevap eğrisinde  $1/4$  genlik oranı elde edilinceye kadar test edilerek oransal kazanç yeniden hesaplanır.

### Cohen-Coon Yöntemi

Cohen-Coon Yöntemi PID denetleyici parametrelerinin tespit edilmesinde kullanılan bir başka yöntemdir. Ziegler-Nichols yöntemine benzer şekilde açık çevrim sistemin basamak cevabı kullanılmaktadır. Basamak cevabının sürekli durum değerinin %28.3'üne ulaşıldığı zaman  $t_1$  değeri ve %63.2 değerine ulaşıldığı zaman da  $t_2$  değeri tespit edilir. Elde edilen  $t_1$  ve  $t_2$  değerleri Denklem 2.17 ve Denklem 2.18 kullanılarak,

$$\zeta_m = \frac{3}{2}(t_2 - t_1) \quad (2.17)$$

$$d = t_2 - \zeta_m \quad (2.18)$$

tespit edilir. Sistem kazancı, kalıcı durum çıkışının sistem giriş genliğine bölümü Denklem 2.19 kullanılarak,

$$K_m = \frac{Y_{ss}(t)}{r(t)} \quad (2.19)$$

tespit edilir. Elde edilen bu veriler kullanılarak  $K_p$ ,  $K_i$ ,  $K_d$  değerleri Tablo 2.4'den belirlenir.

**Tablo 2.4** Cohen – Coon Yöntemi

| Denetleyici Tipi | $K_p$                                              | $T_i$                                       | $T_d$                                      |
|------------------|----------------------------------------------------|---------------------------------------------|--------------------------------------------|
| P                | $\frac{\zeta m}{Kmd} (1 + \frac{d}{3\zeta m})$     | -                                           | -                                          |
| PI               | $\frac{\zeta m}{Kmd} (0.9 + \frac{d}{12\zeta m})$  | $d \frac{30 + 3d/\zeta m}{9 + 20d/\zeta m}$ | -                                          |
| PD               | $\frac{\zeta m}{Kmd} (1.25 + \frac{d}{12\zeta m})$ | -                                           | $d \frac{6 - 2d/\zeta m}{22 + 3d/\zeta m}$ |
| PID              | $\frac{\zeta m}{Kmd} (1.333 + \frac{d}{4\zeta})$   | $d \frac{32 + 6d/\zeta m}{13 + 8d/\zeta m}$ | $d \frac{4}{11 + 2d/\zeta m}$              |

**Chien – Hrones – Reswrich Yöntemi**

Chien – Hrones – Reswrich yöntemi PID denetleyici parametrelerinin tespit edilmesinde kullanılan bir başka yöntemdir. Bu yöntem sistemin açık çevrim basamak cevabının elde edilmesi prensibine dayanır. Sistemin açık çevrim basamak cevabı kullanılarak L, T ve K değerleri elde edilir. Elde edilen bu değerler kullanılarak,

$$a = \frac{K.L}{T} \quad (2.20)$$

değeri hesaplanır. Daha sonra referans regülasyonu ve bozucu etkiyi durdurmak için gerekli denetleyici parametreleri Tablo 2.5 ve Tablo 2.6 kullanılarak tespit edilir.

**Tablo 2.5** Chien – Hrones – Reswrich Yöntemi (Referans Regülasyonu)

| Denetleyici Türü | $K_p$            | $T_i$     | $T_d$      |
|------------------|------------------|-----------|------------|
| <b>P</b>         | $\frac{0.3}{a}$  | -         | -          |
| <b>PI</b>        | $\frac{0.6}{a}$  | $4 * T$   | -          |
| <b>PID</b>       | $\frac{0.95}{a}$ | $2.4 * T$ | $0.42 * L$ |

**Tablo 2.6** Chien – Hrones – Reswiche Yöntemi (Bozucu Etkiyi Durdurmak)

| Denetleyici Türü | $K_p$            | $T_i$     | $T_d$     |
|------------------|------------------|-----------|-----------|
| <b>P</b>         | $\frac{0.3}{a}$  | -         | -         |
| <b>PI</b>        | $\frac{0.35}{a}$ | $1.2 * T$ | -         |
| <b>PID</b>       | $\frac{0.6}{a}$  | $T$       | $0.5 * L$ |

### Aström – Höggland Yöntemi

Aström – Höggland yöntemi PID denetleyici parametrelerinin tespit edilmesinde kullanılan bir başka yöntemdir. PID denetleyici parametrelerinin tespit edilmesi için sistemin kapalı çevrim cevabı kullanılmaktadır[2].

Aström – Höggland metodunda çizilen Nyquist eğrisi üstündeki bir noktanın öngörülen faz payı ve kazanç payına göre bir başka noktaya taşınması prensibine dayanır. Yöntemde, denetlenecek süreç röle geri besleme ile bir limit döngü osilasyonuna sokulur. Nyquist eğrisi üstündeki bir X noktası, osilasyona ve röleye has bilgilerden faydalanarak oluşturulur. Bu X noktasını, beklenen faz ve kazanç payına göre oluşturulan bir Y noktasına taşıyan PID denetleyicinin ayar parametreleri hesaplanır[2].

### 2.6.2 Sezgisel Yöntemler

Karmaşık kontrol sistemlerinde klasik yöntemler sıkıntılar yaşatmaktadır. İleri matematik kullanıma ihtiyacının doğduğu kontrol süreçlerinde parametre belirleme işlemi için klasik yöntemlerin yerini sezgisel yöntemler almaktadır.

Sezgisel yöntemler genetik algoritmalar, parçacık sürü optimizasyonu, farksal evrim algoritması gibi PID parametre belirlemede kullanılabilecek yöntemler barındırmaktadır.

### Genetik Algoritmalar

Genetik algoritma PID denetleyici parametrelerinin tespitinde gerçek kodlu ve ikili kodlu genetik algoritma yapılarından faydalanır[2].

PID denetleyicinin kontrol ettiđi sistemin birim basamak giriřine karřılık sũrekli zaman cevapları dikkate alınmaktadır[2].

Genetik algoritmanın ana kontrol parametreleri aprazlama oranı, popũlasyon bũyũklũđũ, maximum jenerasyon miktarı ve mutasyon oranıdır. Algoritmanın bařarısını direkt etkileyen faktũr popũlasyon bũyũklũđũ iin belirlenen deđerdir.

### **Farksal Evrim Algoritması**

Farksal evrim algoritması kullanılarak PID denetleyici parametrelerinin bulunması devamlı sistem girdileri olan problemlerde kullanılan bir algoritmadır. Kolay ve gũlũ popũlasyon tabanlı bir algoritmadır. Genetik algortmada kullanılan aprazlama, mutasyon ve seme operatũrlerinden yararlanılmaktadır[2].

Farksal evrim algoritmasının ũ avantajı vardır. Bu avantajlar az sayıda kontrol parametresinin belirlenmesi bařlangı deđerlerine bađlı olmadan gerek toplam minimumu hesaplama ve hız yakınsamasıdır. Farksal evrim algoritmasında binominal aprazlama uygulanarak belirli bir ۆlekleme faktũrũ alınıp ok sayıda iterasyon iřlemi yapılmaktadır. Bu iřlemler sonucunda  $K_p$ ,  $K_i$  ve  $K_d$  deđerleri elde edilmeye alıřılmaktadır[2].

### **Paracık Sũrũ Optimizasyonu**

Dođadaki sũrũ sistemlerinden esinlenerek elde edilen ok sayıda optimizasyon vardır. Paracık sũrũ optimizasyonu kullanılarak PID parametrelerinin belirlenmesi, bazı canlı tũrlerinin sũrũ olarak hareket etmesi, hayatı sũrdũrmesi ve yařamı idame ettirmesi iin yardımcı evre řartlarının incelenmesi gibi davranıřları esas almaktadır Paracık sũrũ optimizasyonu sistemdeki sorunlarla alakalı deđerlendirme fonksiyonunu en aza indirmede basit ve etkili bir metottur



### 3. KESİR DERECELİ SİSTEMLER

Kesir dereceli sistemler ilk olarak 1695 yılında Leibniz ile L'Hospital arasındaki mektuplaşmalarda karşımıza çıkmaktadır. L'Hospital'in Leibniz'e gönderdiği bir denklemin  $n$ . mertebeden türevi olan  $D^n f(x)/Dx^n$  için  $n=1/2$  olması durumunda ne olacağı sorusu ile kesir dereceli matematiğin ilk temelleri atılmıştır.

Kesir dereceli türev ve integral kavramları tam dereceli türev ve integral kavramlarından daha eski kavramlar olmasına rağmen tam dereceli türev ve integral kavramları kadar ilgi görmemiştir. Bunun en temel nedeni kısıtlı hesaplama imkanlarının olmasıdır.

Bu alana ilgi son yüz yılda kendini göstermeye başlamıştır. Fakat özellikle 21.yy bu alana olan ilgi oldukça artmıştır. Bilgisayar teknolojisini gelişmesiyle birlikte hesaplama gücünün artmasıyla birçok bilim insanı tarafından üzerine durulabilecek bir konu haline almıştır. Bu duyulan ilgi kesir dereceli sistemlerin mühendislik, ekonomi gibi alanlara uygulanabilir olduğunu ve gerçek sistemleri modellemede daha etkili olduğunu ortaya koymuştur. Böylece birçok bilim alanında kullanılan kesir dereceli sistemler kontrol sistemlerinde de kullanılmaya başlanmıştır. 1958 yılında Tuskin'in büyük objelerin pozisyon kontrolü çalışması ilk çalışmalardandır. Bu çalışmayı takip eden süreçte yapılanlar kesir dereceli sistemlerin çalışma sahasının büyük ölçüde kontrol uygulamalarına ve robotik alanına yönlendirmiştir.

Kesir dereceli sistemler için en genel tanım türev ve integral derecelerinin reel sayı olduğu tam sayı zorunluluğunun olmadığı diferansiyel denklemlerle tasarlanan sistemlere kesir dereceli sistemler denilmektedir. Bu durumdan anlaşılacağı gibi kesir dereceli sistemler aynı zamanda tam dereceli sistemleri de içlerinde barındırmaktadır. Bu durum gerçek sistemleri tanımlanmasında kesir dereceli sistemleri tam dereceli sistemleri de içlerinde barındırmalarından dolayı daha avantajlı hale getirmiştir.

Denklem (3.1) 'de kesir dereceli türev integratör elemanlarını genel gösterim biçimleri verilmiştir.

$${}_aD_t^a = \begin{cases} \frac{d^a}{dt^a} & Re(a) > 0 \\ 1 & Re(a) = 0 \\ \int_0^t (d\tau)^{-a} & Re(a) < 0 \end{cases} \quad (3.1)$$

#### 3.1. Kesir Dereceli Sistemleri Farklı Tanımları

Literatürde kesir dereceli sistemlerin hesaplanmasına yönelik farklı matematiksel yöntemler geliştirildiği görülmüştür. Bu yöntemlerin bir kısmı klasik tam sayı dereceli matematik hesaplamalarının geliştirilmesiyle oluşmuştur.

### 3.1.1. Riemann – Liouville Tanımı

Riemann – Liouville tanımına göre kesir dereceli sistemlerdeki integral ifadesi denklem (3.2) ‘de verilmiştir.

$${}_aD_t^a f(x) = \frac{1}{-a} \int_a^t (x - \tau)^{-a-1} f(\tau) d\tau, \quad a < 0 \quad (3.2)$$

$(n - 1) \leq a \leq n$  olması durumuyla denklem (3.2) ‘nin n. mertebeden türevi alınır

$${}_aD_t^a f(x) = \frac{1}{\Gamma(n-a)} \frac{d^n}{dx^n} \int_a^t (x - \tau)^{n-a-1} f(\tau) d\tau \quad n > 0 \quad (3.3)$$

elde edilir. Bu denklem ile  $f(x)$  fonksiyonun  $(n - a)$ . mertebeden kesir dereceli türevi ifade edilir.

### 3.1.2 Grünwald – Letnikov Tanımı

Grünwald – Letnikov tanımına göre  $a$  reel dereceli denklemi Denklem (3.4)’ de verilmiştir.

$${}_aD_t^a f(x) = \lim_{h \rightarrow 0} \frac{1}{h^a} \sum_{j=0}^{\frac{(t-a)}{h}} (-1)^j \binom{a}{j} f(x - jh) \quad (3.4)$$

Denklemde  $h$  adım sayısını göstermektedir. Bu denklemdeki toplam ifadesi denklem (3.5)

$$\sum_{j=0}^r (-1)^j \binom{a}{j} = \frac{1}{\Gamma(1-a)} \frac{\Gamma(r+1-a)}{\Gamma(r+1)} \quad (3.5)$$

olarak verilmektedir.

Denklem (3.5) ‘ de bulunan  $\Gamma(x)$  ile ifade edilen Euler – Gama fonksiyonu

$$x^a = x \cdot x \cdot \dots \cdot x, \quad a \text{ Tamsayı}$$
$$x^a = e^{a \ln x} \quad (3.6)$$

$$a! = 1.2.3 \dots (a - 1).a, \quad a \text{ Tamsayı}$$

$$a! = \Gamma(a + 1), \quad a \text{ Reelsayı}$$

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt \quad (3.7)$$

formülasyonu bulunur. Denklem (3.7) ‘deki  $x$  bir tam sayıdır. Denklem (3.8)’ de binom katsayısı

$$\binom{a}{j} = \frac{a.(a-1).(a-2) \dots (a-j+1)}{j!} \quad (3.8)$$

ile ifade edilir.

### 3.1.3. Caputo Tanımı

Caputo tanımı Denklem (3.9) ile ifade edilir. Bu tanımda  $n$  tam sayı,  $a$  bir reel sayıdır.

$${}_a D_t^a f(x) = \frac{1}{\Gamma(n-a)} \int_a^t \frac{f^n(\tau)}{(t-\tau)^{a+1-n}} d\tau, \quad (n-1 \leq a < n) \quad (3.9)$$

### 3.2. Kesir Dereceli Sistemlerin Transfer Fonksiyonu

Kesir dereceli denklemler kullanılarak kesir dereceli bir kontrol sisteminin tasarlanması Denklem (3.10) 'da olduğu gibi tanımlanır.

$$\begin{aligned} a_n D_t^{a_n} y(t) + a_{n-1} D_t^{a_{n-1}} y(t) + \dots + a_1 D_t^{a_1} y(t) + a_0 y(t) \\ = b_m D_t^{\beta_m} r(t) + b_{m-1} r(t) + \dots + b_1 D_t^{\beta_1} r(t) + b_0 r(t) \end{aligned} \quad (3.10)$$

Kontrol işlemi gerçekleştirilecek bir sistemin modellenmesinde dinamik davranışlar oldukça önemlidir. Bu nedenle sistem modelleme işlemi yapılırken sistemin dinamik olarak modellenmesi gerekmektedir. Bu amaçla dinamik davranışın incelenmesi için Laplace Dönüşümü alınarak transfer fonksiyonu elde edilir. Denklem (3.10)'a Laplace Dönüşümü uygulandığına aşağıdaki Denklem (3.11) elde edilir.

$$G(s) = \frac{\mathcal{L}[y(t)]}{\mathcal{L}[r(t)]} = \frac{b_1 s^{\beta_1} + b_2 s^{\beta_2} + \dots + b_m s^{\beta_m}}{a_1 s^{a_1} + a_2 s^{a_2} + \dots + a_n s^{a_n}} \quad (3.11)$$

Sisteme laplace dönüşümü uygulanmasının bir diğer avantajı laplace dönüşümünde  $s$  yerine  $j\omega$  yazarak frekans bölgesine geçiş sağlanabilir. Bu geçiş sayesinde Bode, Nyquist, Nichols gibi eğriler çizdirilerek frekans bölgesinde gerekli analizler gerçekleştirilebilir.

### 3.3. Kesir Dereceli $PI^\lambda D^\mu$ Denetleyicisi

Kesir dereceli  $PI^\lambda D^\mu$  denetleyiciler, kontrol sistemlerinde en çok kullanılan PID denetleyicilerin üzerinde revizyon yaparak elde edilmiş denetleyicilerdir. Kesir dereceli  $PI^\lambda D^\mu$  denetleyiciler dinamik sistemlerin kontrol edilmesinde daha iyi sonuçlar vermesi  $PI^\lambda D^\mu$  denetleyicisini popüler hale getirmiştir.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin klasik PID denetleyicisi ile arasındaki en büyük fark klasik PID denetleyicisinde bulunan integral ve türev derecelerinin reel sayı olmasıdır. Bu durum hem kesir dereceli sistemleri kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin daha dinamik ve daha etkin kontrol etmesini sağlamaktadır. Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin integral ve türev derecelerini reel sayı olması tam sayı dereceli olan klasik PID denetleyicisinin integral ve türev

derecelerini de içinde barındırmakta ve klasik PID denetleyicinin etkin olamadığı kontrol bölgelerinde etkinlik gösterebilmektedir.

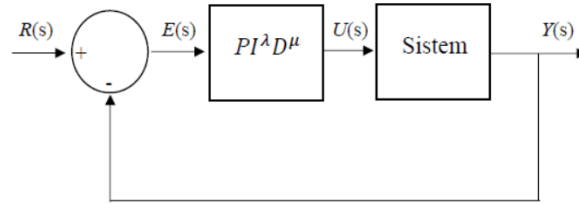
İlk olarak kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin bu modeli Pudlubny ortaya atmıştır. Daha sonra bu denetleyiciler üzerine F. Nonnrmacher ve W.G.Glöchle, I.Costa, C.A.Monje, B.M. Vinagre, Y.Q.Chen, V.Feliu, P.Lanusse, K.S.Miller ve B.ross gibi birçok bilim insanı çalışmıştır. Bu yapılan çalışmalar neticesinde kesir dereceli sistemlerin önemi artmış ve bu durum kontrol sistemleri üzerine çalışmalar gerçekleştiren bilim insanlarının ilgisini kesir dereceli  $PI^\lambda D^\mu$  denetleyicilere çekmiştir.

Pudlubny tarafından ortaya atılan kesir dereceli  $PI^\lambda D^\mu$  denetleyici klasik PID denetleyicisindeki  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerinin yanı sıra integral derecesi olan  $\lambda$  ve türev derecesi olan  $\mu$  parametrelerini de içermektedir.

Pudlubny tarafından geliştirilen  $PI^\lambda D^\mu$  denetleyicisinin s domeninde transfer fonksiyonu aşağıdaki gibidir.

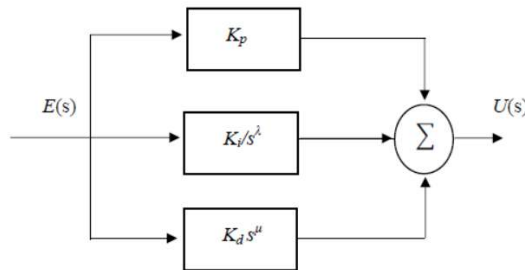
$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (3.12)$$

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisi kapalı çevrim bir sistemdeki blok diyagramı Şekil 3.1'deki gibidir.



Şekil 3.1 Kapalı Çevrim  $PI^\lambda D^\mu$  Denetleyici

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin blok diyagramı aşağıdaki gibidir.(Şekil 3.2)

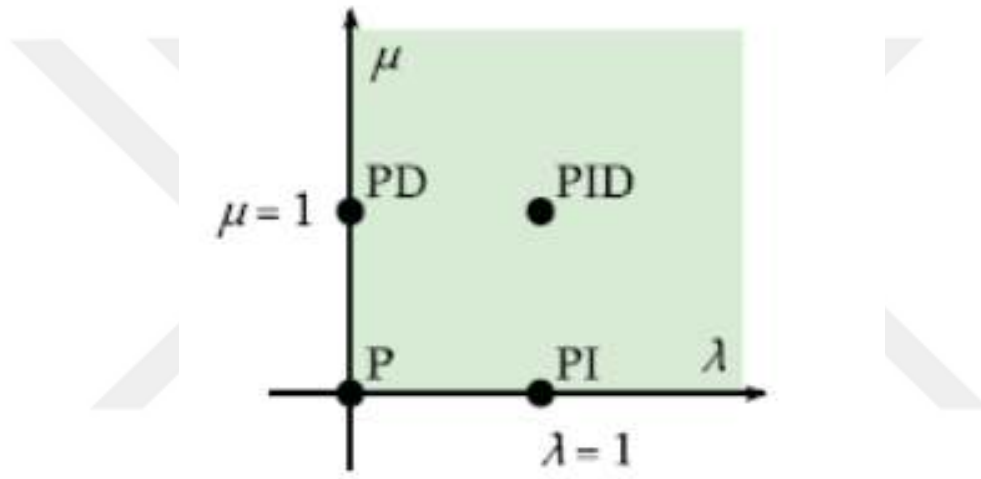


Şekil 3.2 Kesir Dereceli  $PI^\lambda D^\mu$  Denetleyici Bloğu İç Yapısı

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin beş tane ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) parametresinin bulunması kesir dereceli  $PI^\lambda D^\mu$  denetleyicisini klasik PID denetleyicisine karşı daha esnek ve kontrol alanı daha geniş bir hale getirmektedir. [9]

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin integral ve türev dereceleri reel sayı olabildiği için tam sayıları da içlerinde barındırır. Bu durum kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin istenildiği takdirde klasik bir PID denetleyicisi olabilmesinin de yolunu açar.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisi ile içinde barındırdığı klasik PID ve bileşenleri olan PI denetleyici ve PD denetleyici kontrol bölgeleri ve birbirlerine karşı esneklik özelliğinin karşılaştırılması açısından Şekil 3.3 önemlidir. [8]



Şekil 3.3  $PI^\lambda D^\mu$  Denetleyici Kontrol Bölgesi

Grafikten görüldüğü gibi kesir dereceli  $PI^\lambda D^\mu$  denetleyici klasik PID denetleyiciyi ve bileşenleri olan PI denetleyici ile PD denetleyici içinde barındırmaktadır. Ayrıca kesir dereceli  $PI^\lambda D^\mu$  denetleyici bu denetleyicilerden daha geniş ve daha esnek bir bölgede kontrol sağlamaktadır.

Gerek şartlar sağlandığında kesir dereceli  $PI^\lambda D^\mu$  denetleyicisi klasik PID denetleyicisine, PI denetleyicisine ve PD denetleyicisine uygun  $\lambda$  ve  $\mu$  değerleri seçilerek dönüştürülebilmektedir.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin  $\lambda$  ve  $\mu$  değerlerine göre oluşturulabilecek denetleyici türeleri ve gerek şartları aşağıdaki gibidir.

- Tüm denetleyici türelerinde  $\lambda$  ve  $\mu \geq 0$  olmalıdır.
- Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisi için  $\lambda$  ve  $\mu$  reel sayı olmalıdır.
- Klasik PID denetleyici için  $\lambda=1$  ve  $\mu=1$  olmalıdır.
- Kesir dereceli  $PI^\lambda$  denetleyici için  $\lambda$  reel sayı ve  $\mu=0$  olmalıdır.

- Kesir dereceli PD<sup>μ</sup> denetleyici için  $\lambda=0$  ve  $\mu$  reel sayı olmalıdır.
- Klasik PI denetleyici için  $\lambda=1$  ve  $\mu=0$  olmalıdır.
- Klasik PD denetleyici için  $\lambda=0$  ve  $\mu=1$  olmalıdır.
- Klasik P denetleyici için  $\lambda=0$  ve  $\mu=0$  olmalıdır.



## 4. MATERYAL VE METOD

Bu bölüm içeriğinde tez çalışmasının önemli noktası olan kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin  $\lambda$  ve  $\mu$  parametre değerlerinin bulunması için kullanılan method anlatılmıştır.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin transfer fonksiyonu Denklem (4.1)' de verilmiştir[10].

$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (4.1)$$

Yukarıdaki Denklem(4.1)' deki  $\lambda$  ve  $\mu$  değerleri  $\lambda = 1$  ve  $\mu = 1$  olarak belirlenirse denetleyici klasik PID denetleyicisi halini almaktadır.

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (4.2)$$

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinde belirlenmesi gereken beş adet parametre ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) mevcuttur. Oransal kazanç  $K_p$ , integral kazanç  $K_i$ , türev kazanç  $K_d$  Ziegler – Nichols metoduyla veya Matlab programından PIDTuner ara yüzü ile tespit edilebilir.

İntegral derecesi  $\lambda$  ve türev derecesi  $\mu$  değerlerinin doğru bir şekilde tespit edilmesi kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin klasik PID denetleyiciye göre avantajlarının ortaya çıkması açısından çok önemlidir. Bu sebeple bu seçimin uygun yöntemlerle ihtiyaca cevap verecek şekilde yapılması çok önemlidir.

Parametrelerin tespit edilmesinde literatürdeki çalışmalardan da görülebildiği üzere klasik PID parametre belirleme yöntemleri yol göstericidir. İntegral derecesi  $\lambda$  ve türev derecesi  $\mu$  değerlerinin sisteme en uygun değerler seçilebilmesinin en önemli ayağını klasik PID parametrelerinin doğru tespit edilmesi yatar. Bunun sebebi ise integral derecesi ile türev derecesinin belirlenmesinde klasik PID parametreleri olan  $K_p$ ,  $K_i$  ve  $K_d$  rol oynar.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyici parametrelerinin belirlenmesinde öncelikle klasik PID parametrelerinin tespitinde klasik yöntemlerden olan Ziegler – Nichols yöntemi ön plana çıkmıştır. Buradan elde edilen parametreler Aström – Höggland metodunda faz payına ulaşmak için elde edilen iki doğrusal olmayan denklemin nümerik olarak çözülmesiyle integral ve türev kazancı tespit edilebilmektedir.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyici parametrelerinin tespit edilmesinde kullanılacak yöntemde  $K_p$ ,  $K_i$  ve  $K_d$  değerlerinden faydalananak bulunması gereken başka parametreler de mevcuttur. Bu parametreleri Aström – Höggland yönteminden elde edilecek doğrusal olmayan denklemlerin nümerik olarak çözümü açısından büyük önemi vardır. Bu parametreler kritik frekans  $W_c$  ve kritik kazanç değeri  $K_c$  dir. Kritik frekans ve kritik kazanç değerleri  $K_p$ ,  $K_i$  ve  $K_p$  değerlerinden elde edilir. Bu bağıntılar Denklem (4.3), (4.4),(4.5) ve (4.6) da gösterilmiştir.

$$K_p = 0.6 * K_c \quad (4.3)$$

$$K_i = 0.54 * \frac{K_c}{P_u} \quad (4.4)$$

$$K_d = 0.075 * K_c * P_u \quad (4.5)$$

$$P_u = \frac{2 * \pi}{W_c} \quad (4.6)$$

G(s) sisteminde istenilen faz payı  $\phi_{pm}$  ile gösterilmiştir.  $\phi_{pm}$  ve  $\arg(G(jw_{cp})) = -180$  eşitliğini sağlayan Nyquist eğrisi üzerindeki kritik frekans değeri  $w_{cp}$  olması durumunda sistem kazanç payı Denklem (4.7)' deki gibidir.

$$G_m = \frac{1}{|G(jw_{cp})|} = K_c \quad (4.7)$$

$|G(jw_{cp}) C(jw_{cp})| = 1$  eşitliğinin ve  $\phi_{pm}$  değerinin sağlanması için Denklem (4.8)

$$C(jw_{cp}) = \frac{1}{|G(jw_{cp})|} e^{j\phi_{pm}} = K_c \cos(\phi_{pm}) + j K_c \sin(\phi_{pm}) \quad (4.8)$$

Denklem (4.7) ve Denklem (4.8)'den  $C(jw_{cp})$  çekilirse Denklem (4.9) oluşturulur.

$$\begin{aligned} C(jw_{cp}) &= K_p + K_i w_{cp}^{-\lambda} \cos\left(\frac{\pi}{2} * \lambda\right) + K_d w_{cp}^{\mu} \cos\left(\frac{\pi}{2} * \lambda\right) + j[-K_i w_{cp}^{-\lambda} \sin\left(\frac{\pi}{2} \lambda\right) \\ &+ K_d w_{cp}^{\mu} \sin\left(\frac{\pi}{2} \mu\right)] \end{aligned} \quad (4.9)$$

Denklem (4.9) da elde edilen denklemi düzenlediğimizde integral derecesini ve türev derecesi bulmak için kullanabileceğimiz doğrusal olmayan iki denklem elde edilmektedir. Bu denklemler Denklem (4.10) ve Denklem (4.11)'de gösterilmiştir.[11]

$$f_1(\lambda, \mu) = K_p + K_i w_{cp}^{-\lambda} \cos\left(\frac{\pi}{2} \lambda\right) + K_d w_{cp}^{\mu} \cos\left(\frac{\pi}{2} \mu\right) + K_c \left(\cos(\phi_{pm})\right) = 0 \quad (4.10)$$

$$f_2(\lambda, \mu) = -K_i w_{cp}^{-\lambda} \sin\left(\frac{\pi}{2} \lambda\right) + K_d w_{cp}^{\mu} \sin\left(\frac{\pi}{2} \mu\right) + K_c \left(\sin(\phi_{pm})\right) = 0 \quad (4.11)$$

Bu denklemlerin nümerik çözümleri için Matlab'dan “fsolve” komutu kullanılabilir.



## 5. KESİR DERECELİ $PI^\lambda D^\mu$ DENETLEYİCİ TASARIMI

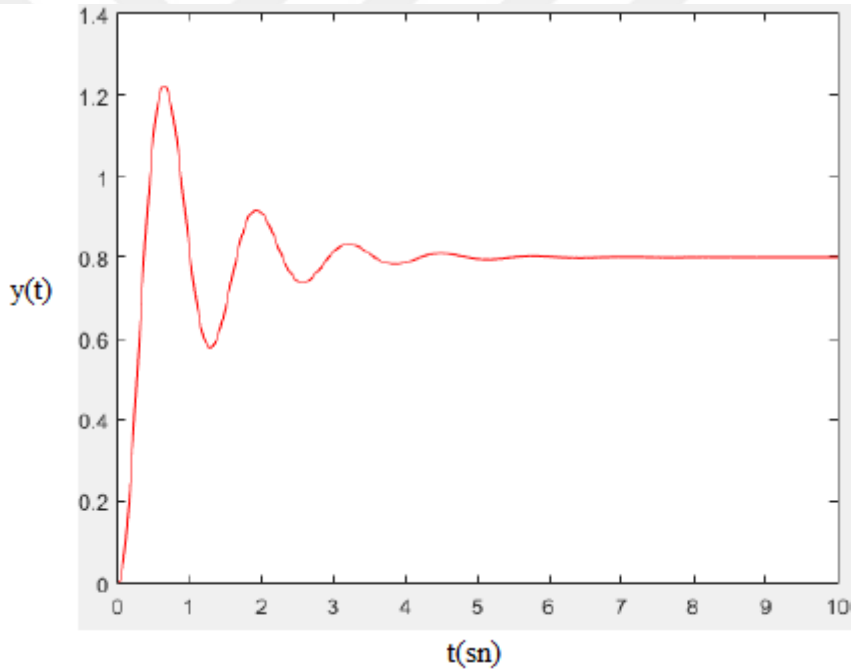
Arayüz tasarımında tespit edilecek  $\lambda$  ve  $\mu$  değerleri için bu bölümde kullanılacak olan Aström yönteminin doğru çalıştığının ortaya konulması için bu bölüm kapsamında transfer fonksiyonu bilinen bir sistem için kesir dereceli  $PI^\lambda D^\mu$  denetleyicisi tasarlanmış Matlab/Simulink programında FOMCON toolbox kullanılarak simülasyonu gerçekleştirilmiştir.

Kontrol edilecek sistem için seçilen transfer fonksiyonu aşağıdaki gibidir.

$$G(s) = \frac{20}{s^2 + 2s + 5} \quad (5.1)$$

$$H(s) = 1 \quad (5.2)$$

Simülasyon için birim geri beslemeli bir kapalı çevrim sistemde herhangi bir kontrol işlemine tabi tutulmadan sistemin birim basamak cevabı aşağıdaki gibidir.(Şekil 5.1)



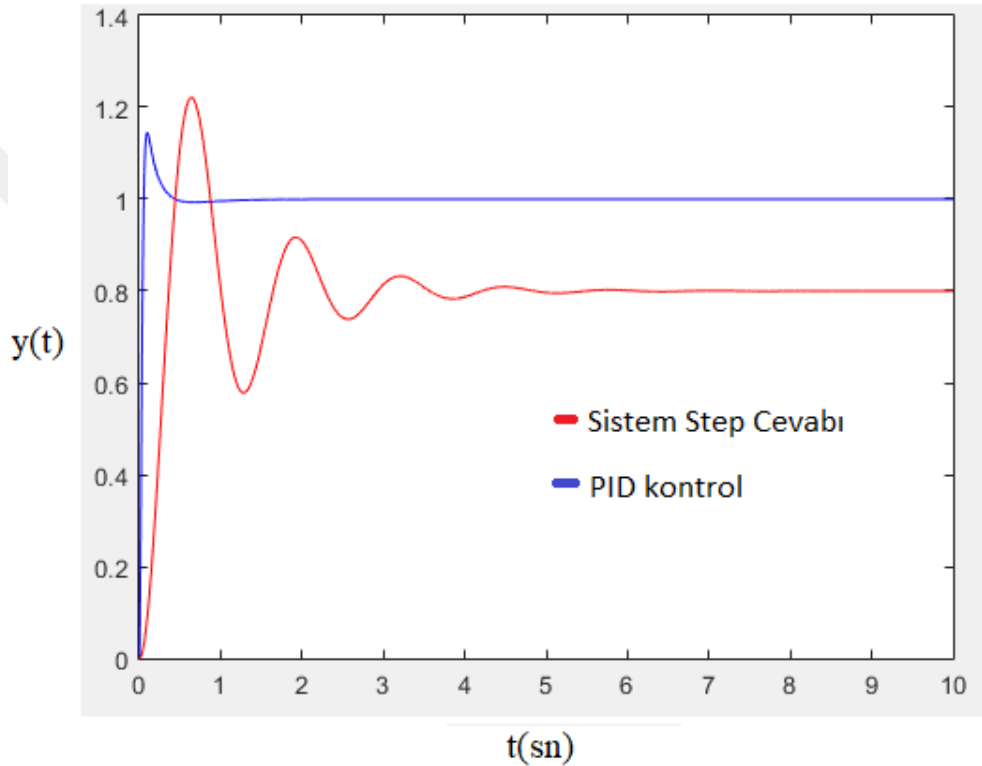
Şekil 5.1 G(s) Fonksiyonunun Kontrolsüz Cevabı

Sistem uygulanan girişi belirli bir kalıcı durum hatası ile (%20) takip etmektedir. Ayrıca sistemin geçici durumdaki salınımı çok yüksek olduğu gözlemlenmektedir. Bu sistem için tasarlanacak kesir dereceli  $PI^\lambda D^\mu$  denetleyicisinin klasik PID ile aynı olan  $K_p$ ,  $K_i$  ve  $K_d$  değerlerinin tespiti için Matlab programı kullanılmıştır. En optimum seviyedeki maximum aşma ve oturma zamanı göz önünde bulundurulmuştur.

Matlab programındaki PIDTuner ara yüzü kullanılarak sistem için en uygun parametrelerin tespiti sağlanmıştır. PIDTuner ara yüzünün tespit ettiği klasik PID denetleyici parametreleri aşağıda verilmiştir.

$$K_p = 12.84492, K_i = 23.77176, K_d = 1.54790$$

Tespit edilen klasik PID denetleyicisinin sistem kontrolüne etkisi aşağıda gösterilmiştir.(Şekil 5.2)



**Şekil 5.2** G(s) Fonksiyonunu Klasik PID Denetleyici Cevabı

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici parametrelerinde kullanacağımız integral derecesi ve türev derecelerinin tespiti için bu parametreler yol göstericidir.

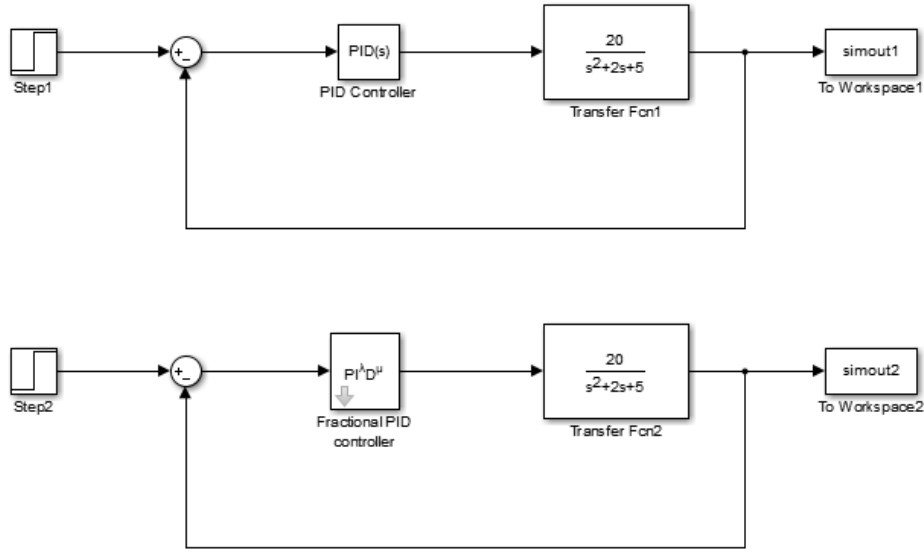
Aström – Höglund yönteminde bir önceki bölümde denklem (4.10) ve denklem (4.11) de elde ettiğimiz iki doğrusal olmayan denklemin çözümü için yukarıda tespit edilen  $K_p$ ,  $K_i$  ve  $K_d$  parametreleri kullanılacaktır. Denklem (4.10) ve denklem (4.11)'de kullanılacak diğer parametreler  $w_{cp}$  ve  $k_c$  parametrelerinin tespiti için denklem (4.3) denklem (4.5) ve denklem (4.6) kullanılmaktadır. Elde edilen parametreler denklem (4.10) ve denklem (4.11) deki iki denklemin çözümü için kullanılmaktadır. Bu denklemlerin çözümü için Matlab programının fsolve komutu kullanılmıştır. Bu işlemler sonucunda kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicisinin klasik PID den farklı olan  $\lambda$  ve  $\mu$  değerleri tespit edilir.

Örnek olarak aldığımız transfer fonksiyonu için tespit edilen  $\lambda$  ve  $\mu$  parametreleri aşağıdaki gibidir.

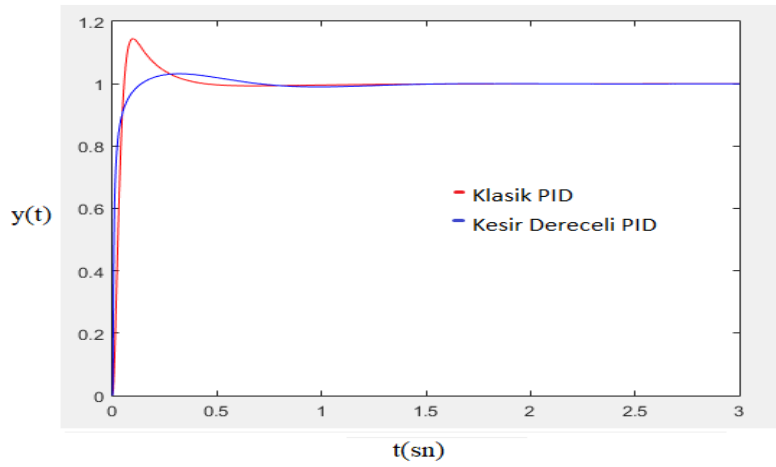
$$\lambda = 0.9553 \text{ ve } \mu = 1.2577$$

Bu parametreler ile tasarlanan kesir dereceli  $PI^\lambda D^\mu$  denetleyici parametreleri örnek transfer fonksiyonu için Matlab/Simulink’de model hazırlanmıştır.(Şekil 5.3)

Hazırlanan simülasyon kesir dereceli  $PI^\lambda D^\mu$  denetleyici ile klasik PID denetleyici karşılaştırılmıştır.



Şekil 5.3 Matlab/Simulink Blok Diyagramı



Şekil 5.4  $PI^\lambda D^\mu$  Denetleyici ile Klasik PID Denetleyici Cevapları

Şekil 5.4’de görüldüğü gibi kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici klasik PID denetleyicisine oranla özellikle maximum aşma ve oturma süresini büyük ölçüde düzeltmiştir.

İki denetleyici arasındaki maximum aşma değerleri Şekil 5.5 ve Şekil 5.6’ da gösterilmiştir.

| Time   | Data:1 |
|--------|--------|
| 0.0815 | 1.1313 |
| 0.0838 | 1.1352 |
| 0.0865 | 1.1387 |
| 0.0890 | 1.1411 |
| 0.0913 | 1.1427 |
| 0.0933 | 1.1436 |
| 0.0955 | 1.1442 |
| 0.0980 | 1.1443 |
| 0.1007 | 1.1440 |
| 0.1032 | 1.1433 |
| 0.1052 | 1.1425 |
| 0.1073 | 1.1415 |
| 0.1096 | 1.1401 |
| 0.1123 | 1.1383 |
| 0.1150 | 1.1364 |

Şekil 5.5 Klasik PID Denetleyici Max Aşma Değeri

| Time   | Data:1 |
|--------|--------|
| 0.3003 | 1.0318 |
| 0.3032 | 1.0318 |
| 0.3051 | 1.0319 |
| 0.3071 | 1.0319 |
| 0.3096 | 1.0319 |
| 0.3125 | 1.0320 |
| 0.3145 | 1.0320 |
| 0.3166 | 1.0320 |
| 0.3191 | 1.0320 |
| 0.3219 | 1.0320 |
| 0.3245 | 1.0319 |
| 0.3265 | 1.0319 |
| 0.3285 | 1.0319 |
| 0.3307 | 1.0319 |
| 0.3334 | 1.0318 |

Şekil 5.6  $PI^{\lambda}D^{\mu}$  Denetleyici Max Aşma Değeri

Şekil 5.5 ve Şekil 5.6’da görüldüğü gibi sistem klasik PID denetleyici ile kontrol edildiğinde maximum aşma  $M_p = 1.1443$ , kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici ile kontrol edildiğinde

ise  $M_p = 1.0320$  olmaktadır. Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici bu sistem için  $M_p$  değerini yaklaşık %14 den % 3 e düşürmektedir.

| Time   | Data:1 |
|--------|--------|
| 2.6964 | 0.9999 |
| 2.6991 | 0.9999 |
| 2.7019 | 0.9999 |
| 2.7042 | 0.9999 |
| 2.7061 | 0.9999 |
| 2.7081 | 0.9999 |
| 2.7106 | 0.9999 |
| 2.7135 | 1.0000 |
| 2.7155 | 1.0000 |
| 2.7176 | 1.0000 |
| 2.7201 | 1.0000 |
| 2.7229 | 1.0000 |
| 2.7254 | 1.0000 |
| 2.7275 | 1.0000 |
| 2.7294 | 1.0000 |

**Şekil 5.7** Klasik PID Denetleyici Oturma Süresi

| Time   | Data:1 |
|--------|--------|
| 1.6371 | 0.9999 |
| 1.6398 | 0.9999 |
| 1.6426 | 0.9999 |
| 1.6449 | 0.9999 |
| 1.6468 | 0.9999 |
| 1.6489 | 0.9999 |
| 1.6514 | 0.9999 |
| 1.6542 | 0.9999 |
| 1.6563 | 1.0000 |
| 1.6583 | 1.0000 |
| 1.6608 | 1.0000 |
| 1.6636 | 1.0000 |
| 1.6662 | 1.0000 |
| 1.6683 | 1.0000 |
| 1.6702 | 1.0000 |
| 1.6721 | 1.0000 |

**Şekil 5.8**  $PI^{\lambda}D^{\mu}$  Denetleyici Oturma Süresi

Şekil 5.7 ve Şekil 5.8’de görüldüğü gibi klasik PID denetleyici ile kontrol edildiğinde oturma süre  $t_s = 2.713s$  , kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici ile kontrol edildiğinde oturma süresi  $t_s = 1.6563s$  olmaktadır.

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyiciler kontrol ettikleri sistemleri özellikle geçici durumlarını klasik PID denetleyicilere göre daha iyi kontrol ettikleri görülmüştür.

## 6. MATLAB ARAYÜZ TASARIMI

Kesir dereceli  $P^{\lambda}D^{\mu}$  denetleyici tasarımı için kullanılan toolboxlar tasarım işlemi için tasarımcıya gerekli olan tüm program yetenekleri tek bir arayüz üzerinde bulunmamaktadır. Mevcut toolboxlar her transfer fonksiyonu için  $\lambda$  ve  $\mu$  değerlerini doğru tespit edememektedir. Bunun temel sebebi bu bölüm içerisinde de anlatıldığı gibi  $K_p$ ,  $K_i$  ve  $K_d$  parametrelerinin  $\lambda$  ve  $\mu$  belirleme yöntemine uyum sağlayamamasıdır. Burada geliştirilen arayüz ile tüm program yetenekleri tek bir arayüz üzerinden ulaşılabilir. Ayrıca  $\lambda$  ve  $\mu$  değerlerinin yanlış tespit edilememesi için transfer fonksiyonuna göre gerekli yönlendirmelerin yapıldığı bir arayüz tasarlanmıştır.

Kesir dereceli sistemler ve kesir dereceli denetleyiciler için hazırlanan arayüz Matlab programının GUI arayüz tasarım bölümünde geliştirilmiştir. Tasarlanan arayüzün istenilen tasarımları ortaya koyabilmesi için Matlab programında yazılan alt programlar kullanılmıştır.

### 6.1 Kesir Dereceli Sistemlerin Modellenmesi

Kesir dereceli matematik için Matlab programı araştırmacıların oldukça tercih ettiği bir programdır. Aynı şekilde kontrol sistemleri üzerine çalışan araştırmacılar içinde vazgeçilmez bir yeri olmuştur. Ancak Matlab programı kontrol sistemlerinde tam sayı dereceli kontrol işlemini yapacak toolboxlara sahip olmasına rağmen kesir dereceli kontrol işlemlerini uygulama yeteneğine sahip değildir. Bunun sebebi kontrol sistemlerinde kullanılan s domeni Matlab programı içinde sadece tam sayı dereceli olarak tasarlanabilmesidir. Kontrol sistemlerinde kullanılan s domeni günümüzde tasarlanan sistem modellerinin ve kontrolör modellerinin hemen hemen tamamında sistem analizi için çok ciddi avantajlara sahip olmasından dolayı kullanılmaktadır. Kesir dereceli bir sistem modelinin veya kontrolörün tasarım ve analizinde s domeni çok önemli bir yer tuttuğu için s domeninin kesir dereceli bir şekilde Matlab programına tanımlanması gerekmektedir. Bu amaçla öncelikle kesir dereceli sistemleri Matlab programına tanımlanabilmesi için **fotf.m** isimli bir **mfunction** tasarlanmıştır. Bu mfunction yaptığı işlem tasarımı ve analizi yapılacak kesir dereceli sistemi bir nesne olarak sisteme tanımlamasıdır. Matlab programında nesne tanımlaması yapmak ve bu nesneyi görüntülemek amacıyla fotf.m isimli mfunction dosyasının @fotf şeklinde tanımlanan bir dosyanın içine kaydedilmesi gerekmektedir. Bu tanımlanan nesnenin ise görüntülenebilmesi için Matlab programında tanımlanan display komutu modifiye edilerek yeni bir **display.m** yazılması gerekmektedir.[8] Kesir dereceli sistemlerin Matlab programına nesne olarak tanımlanması için kullanılacak fotf.m ve görüntüle için kullanılacak display.m komutları aşağıdaki gibidir.

## Kesir Dereceli Sistem Tanımı İçin Kullanılan Komut Dizisi

### fotf.m

```
function G=fotf(a,na,b,nb,T)
if nargin==0,G.a=[]; G.na=[];G.b=[]; G.nb=[];G.ioDelay=0;
G=class(G,'fotf');elseif isa(a,'fotf'),G=a;
elseif nargin==1 && isa(a,'double'),
G=fotf(1,0,a,0,0);
elseif nargin==1 && a=='s',
G=fotf(1,0,1,1,0);else
ii=find(abs(a)<eps); a(ii)=[]; na(ii)=[];
ii=find(abs(b)<eps); b(ii)=[]; nb(ii)=[];
if nargin==5,G.ioDelay=abs(T);else
G.ioDelay=0;end G.a=a;G.na=na; G.b=b;G.nb=nb;
G=class(G,'fotf');end end
```

Kesir dereceli sistem s domeninde transfer fonksiyonu olarak fotf.m mfunctionu kullanılarak tanımlanmıştır.

## Transfer Fonksiyonun Görüntülenmesi İçin Gereken Komut Dizisi

### display.m

```
function display(G)
strN=polydisp(G.b,G.nb);strD=polydisp(G.a,G.na);
nn=length(strN);nd=length(strD);
nm=max([nn,nd]);
disp([char(' '*ones(1,floor((nm-nn)/2))) strN]),
ss=[];T=G.ioDelay;
if T>0, ss=[' exp(-' num2str(T) 's)'];
end
disp([char('-'*ones(1,nm)), ss]);
disp([char(' '*ones(1,floor((nm-nd)/2))) strD])
function strP=polydisp(p,np)
P=''; [np,ii]=sort(np,'descend');p=p(ii);
for i=1:length(p),
P = [P,'+',num2str(p(i)),'s^{',num2str(np(i)),'}'];
end
P=P(2:end); P=strrep(P,'s^{0}','');
P=strrep(P,'+-',''); P=strrep(P,'^{1}','');
P=strrep(P,'+1s','+s');strP=strrep(P,'-1s','-s');
if length(strP)>=2 && strcmp(strP(1:2),'1s')
strP=strP(2:end);
end
end
end
```

Kesir dereceli sistem modellerin Matlab programına tanıtılması için örnek bir transfer fonksiyonu üzerinden aşağıdaki gibi gösterilmektedir.

$$G(s) = \frac{s^{1.2} + 1}{1.5s^{1.8} + 0.7s^{0.6} + 1} \quad (6.1)$$

transfer fonksiyonu için;

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
```

Matlab Command Window girdisinin çıktısı

```
s^{1.2}+1
```

```
-----
```

```
1.5s^{1.8}+0.7s^{0.6}+1
```

şeklinde kesir dereceli sistem görüntülenmektedir.

## 6.2 Kesir Dereceli Sistemlerin Geri Beslemeleri

Kontrol sistemlerin kapalı çevrim içerisinde geri besleme işlemlerini yapabilmeleri için Matlab programı tarafından geliştirilen komutlar bulunmaktadır. Ancak kesir dereceli olarak s domeninde tanımlanmış sistemlerin kapalı çevrim içerisinde geri besleme işlemini gerçekleştirebilmesi için bu komutlar çalışmamaktadır. Bu nedenle kesir dereceli sistemler için geri besleme işlemlerini gerçekleştirecek bir komut tasarlanması gerekmektedir. [8]

### Geri Besleme Komutu İçin Geliştirilen Komut Dizisi

#### simple.m

```
function G=simple(G1)
[a,n]=polyuniq(G1.a,G1.na);
G1.a=a;G1.na=n;na=G1.na;
[a,n]=polyuniq(G1.b,G1.nb);
G1.b=a;G1.nb=n;nb=G1.nb;
nn=min(na(end),nb(end));
nb=nb-nn;na=na-nn;
G=fotf(G1.a,na,G1.b,nb,G1.ioDelay);
function [a,an]=polyuniq(a,an)
[an,ii]=sort(an,'descend');
a=a(ii);ax=diff(an);key=1;
for i=1:length(ax)
if ax(i)==0,
a(key)=a(key)+a(key+1);a(key+1)=[];
an(key+1)=[];
else, key=key+1;
end
endendend
```



## Geri Besleme İçin Geliştirilen Komut Dizisi

### feedbackfotf.m

```
function G=feedbackfotf(F,H)
F=fotf(F);
H=fotf(H);
na=[];
nb=[];
if F.ioDelay==H.ioDelay
b=kron(F.b,H.a);
a=[kron(F.b,H.b), kron(F.a,H.a)];
for i=1:length(F.b),
nb=[nb F.nb(i)+H.nb];
na=[na, F.nb(i)+H.nb];
end
for i=1:length(F.a),
na=[na F.na(i)+H.na];
end
G=simple(fotf(a,na,b,nb,F.ioDelay));
end
end
```

Tanımlanan transfer fonksiyonunun kapalı çevrim birim geri beslemesi üzerine tasarlanan mfunctionların çalışması aşağıdaki gibidir.

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>>Gc=feedbackfotf(G,1)
      s^{1.2}+1
-----
1.5s^{1.8}+s^{1.2}+0.7s^{0.6}+2
```

Sisteme tanımlanan G transfer fonksiyonu feedbackfotf.m mfunctionu ile kapalı çevrim birim geri beslemesi gerçekleştirilmiştir. Kapalı çevrim birim geri beslemesi sonucu yine bir transfer fonksiyonu olarak görüntülenmesi sağlanmıştır.

## 6.3 Kesir Dereceli Sistemlerin Simülasyonu

Kesir dereceli sistemler için geliştirilecek arayüzün en önemli noktası sistem simülasyonudur. Matlab da tam sayı dereceli sistemlerin simülasyonları için kullanılan komut ve araçlar kesir dereceli sistem simülasyonları için kullanılamamaktadır. Tasarlanacak arayüz için

istenilen test sinyalinin tanımlanabildiği arzu edilen süre boyunca sisteme uygulanabildiği lsim.m isimli bir mfunction benzetimi gerçekleştirilmiştir.

Geliştirilen bu mfunction üç adet girdi barındırmaktadır. Bunlardan birincisi sistem ikincisi uygulanacak test sinyali üçüncüsü ise test sinyalinin uygulama süresidir. Burada sistem parameresini biraz daha izah edilmesi gerekmektedir. Arayüzün farklı kullanım noktalarında farklı amaçlarla kullanılacak olan lsim.m isimli mfunction komut dizisi giriş olarak bir transfer fonksiyonu almaktadır. Bu girilecek transfer fonksiyonu sistemin herhangi bir kapalı çevrim uygulanmamış hali, kapalı çevrim birim geri beslemeye uygulanmış hali veya kontrolör tasarımı sonucu elde edilmiş hali olabilmektedir.

Böylece sistemin tasarım ve analiz durumundaki her an için arzu edilen test sinyali ile test edilebilecek esnek bir yapı ortaya çıkmaktadır. Ayrıca tam sayı dereceli sistemler de aynı zamanda kesir dereceli sistemler grubuna dahil olduğundan bu simülasyon komut dizisi hem kesir dereceli sistemler için hemde tam sayı dereceli sistemler için kullanılabilir.

Kesir dereceli sistemlerin simülasyonu için geliştirilen lsim.m mfunction komut dizisi aşağıda gösterilmiştir.[8] Simülasyon sonucu Şekil 6.1’de gösterilmiştir.

### Kesir Dereceli Sistemlerin Simülasyonu İçin Geliştirilen Komut Dizisi

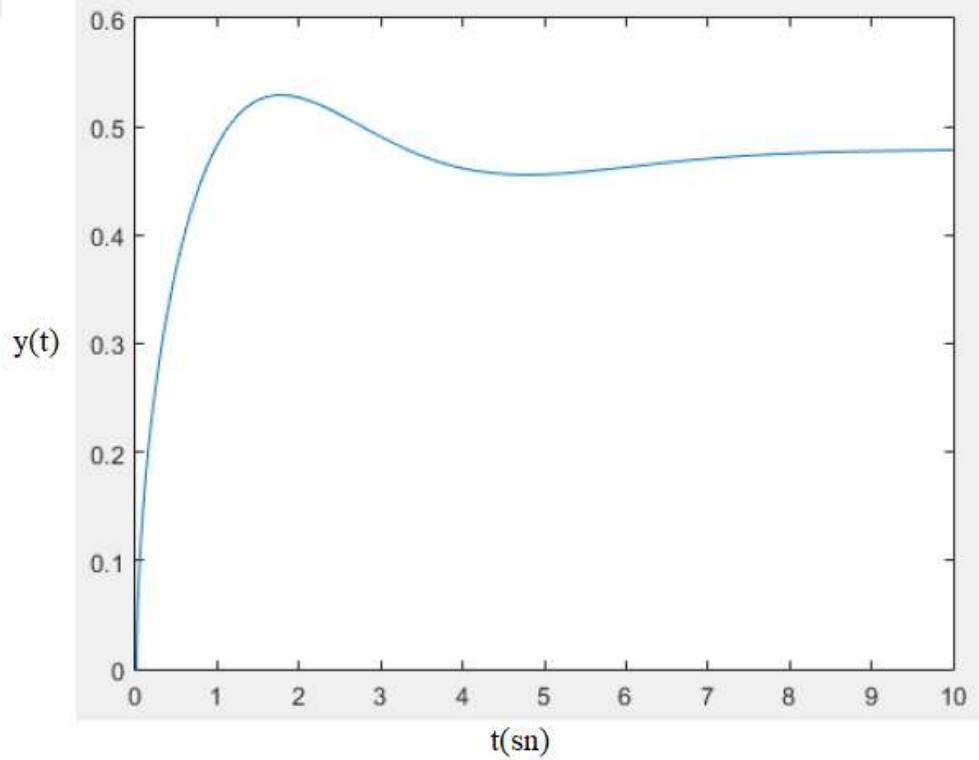
```
function y=lsim(G,u,t)
a=G.a;eta=G.na;b=G.b;
gamma=G.nb;
nA=length(a);h=t(2)-t(1);
D=sum(a./[h.^eta]);
W=[];nT=length(t);
vec=[eta gamma];
D1=b(:)./h.^gamma(:);
y1=zeros(nT,1);
W=ones(nT,length(vec));
for j=2:nT
W(j,:)=W(j-1,:).*(1-(vec+1)/(j-1));
endfor i=2:nT
A=(y1(i-1:-1:1))'*W(2:i,1:nA);
y1(i)=(u(i)-sum(A.*a./[h.^eta]))/D;
endy = zeros(nT,1);
for i=2:nT
y(i)=(W(1:i,nA+1:end)*D1)'*(y1(i:-1:1));
endii=find(t>G.ioDelay);
lz = zeros(ii(1)-1,1);
y = [lz; y(1:end-length(lz))];
plot(t,y)
end
```

Tanımlanan transfer fonksiyonu için geliştirilen lsim.m mfunction komut dizisi çalışması aşağıdaki gibidir.

```

>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>>Gc=feedbackfotf(G,1)
      s^{1.2}+1
-----
1.5s^{1.8}+s^{1.2}+0.7s^{0.6}+2
>> t=0:0.01:10;
>> u=ones(size(t));
>> y=lsim(Gc,u,t);

```



**Şekil 6.1** İsim Kapalı Çevrim Simülasyon Çıktısı

## 6.4 Kesir Dereceli Sistemlerin Frekans Domeninde Analizi

Kesir dereceli sistemleri analizi için kullanılan domein lerden biri ise frekans domenidir. Kontrol sistemleri üzerine çalışan araştırmacılar analizini yapacakları sistemin karakteristik özelliklerine veya analizini yapmak istedikleri verilere göre farklı domeinler kullanırlar. Tam sayı dereceli sistemlerin analizinde kullanılan frekans domeni kesir dereceli sistemlerin analizinde de

son derece önemlidir. Matlab programı içeriğinde tam sayı dereceli sistemlerin frekans domenindeki analizi için komut ve araçlar bulunmaktadır. Fakat bu durum kesir dereceli sistemlerin analizinde geçerli değildir. Geliştirilen arayüzde hedef hem s domeninde hemde frekans domeninde analiz gerçekleştirebilmektir. Bu sebeple hem tam sayı dereceli sistemlerde hemde kesir dereceli sistemlerde kullanılabilecek frekans domeninde analiz yapmamızı sağlayan bir komut dizisi oluşturulmuştur.

Frekans domeninde analiz için gerekli olan bode, nichols ve nyquist eğrileri için bode.m nichols.m ,nyquist. m mfunction komut dizileri yazılmıştır.

Bode, nichols, nyquist eğrilerinin kesir dereceli sistemlerde oluşturulabilmesi için freqresp.m komut dizisi kullanılmıştır.(A.Monje) . Simülasyon sonucu Şekil 6.2, Şekil 6.3 ve Şekil 6.4’ de gösterilmiştir.

### **Frekans Domeninde Analizi İçin Geliştirilen Komut Dizileri**

#### **freqresp.m ;**

```
function H1=freqresp(w,G)
a=G.a;na=G.na;b=G.b;
nb=G.nb;j=sqrt(-1);
for i=1:length(w)
P=b*(w(i).^nb. ');
Q=a*(w(i).^na. ');
H1(i)=P/Q;
endif G.ioDelay>0,
A=abs(H1);
B=angle(H1)-w1*G.ioDelay;
H1=A.*exp(j*B);
endend
```

#### **bode.m;**

```
function H=bode(G,w)
if nargin==1,w=logspace(-4,4);
endj=sqrt(-1);
H1=freqresp(j*w,G);
H1=frd(H1,w);if nargout==0,
bode(H1)elseH=H1;
endend
```

#### **nichols.m**

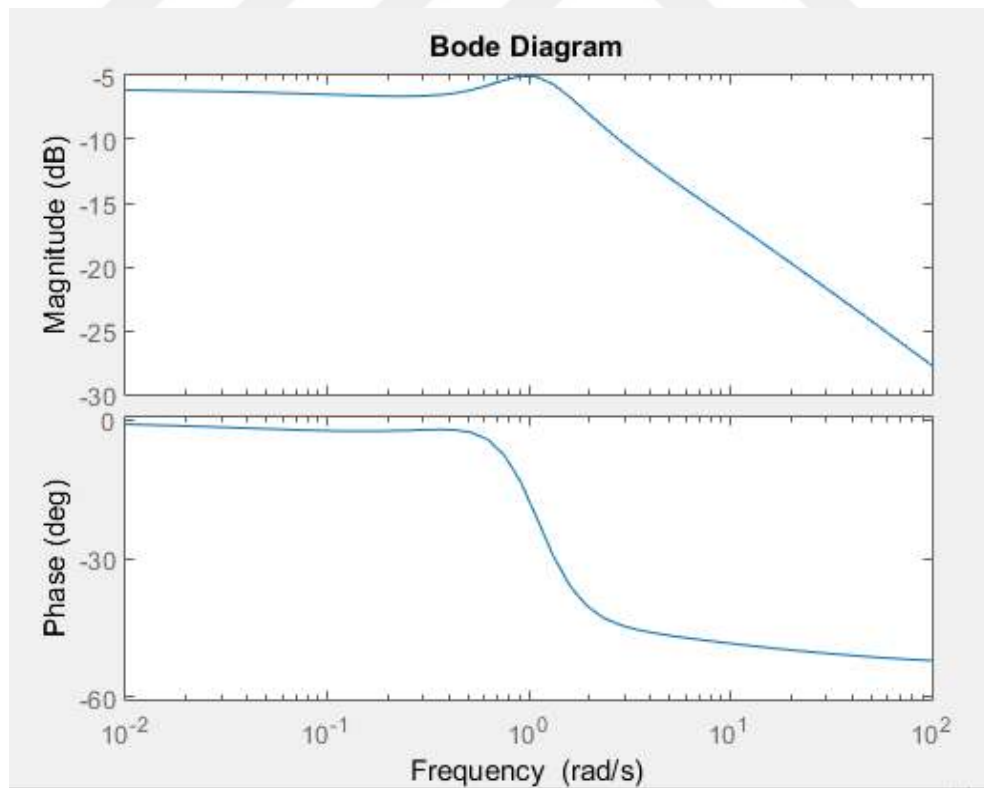
```
function nichols(G,w)
if nargin==1,w=logspace(-4,4);
endH=bode(G,w);
nichols(H);
```

### nyquist.m

```
function nyquist(G,w)
if nargin==1,w=logspace(-4,4);
endH=bode(G,w);
nyquist(H);end
```

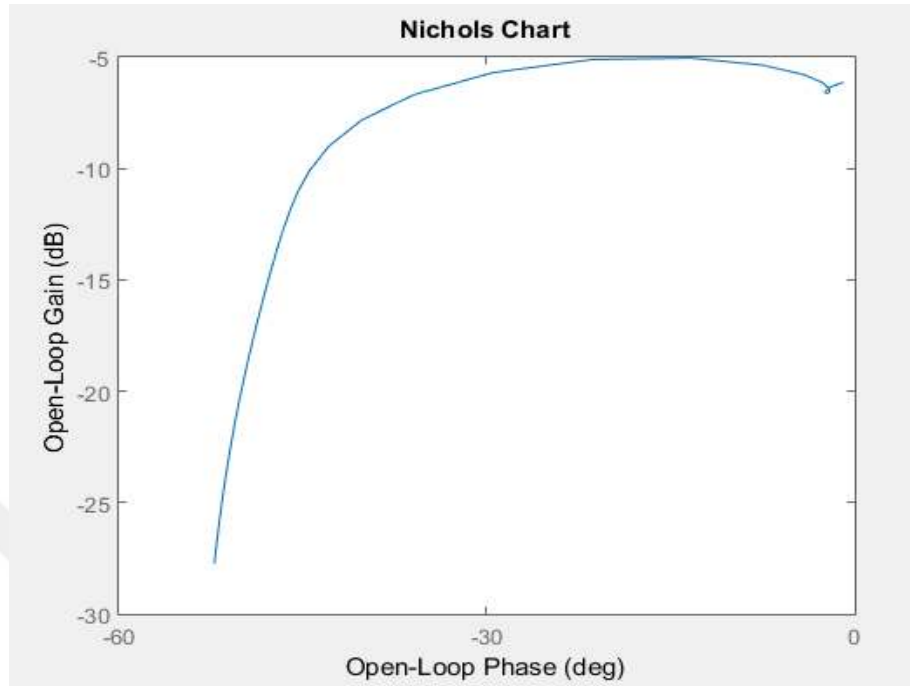
Tanımlanan transfer fonksiyonu için geliştirilen bode.m komut dizisinin çalışması aşağıdaki simülasyonla gösterilmiştir.

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>>Gc=feedbackfotf(G,1)
      s^{1.2}+1
-----
1.5s^{1.8}+s^{1.2}+0.7s^{0.6}+2
>> w=logspace(-2,2);
>> bode(Gc,w)
```



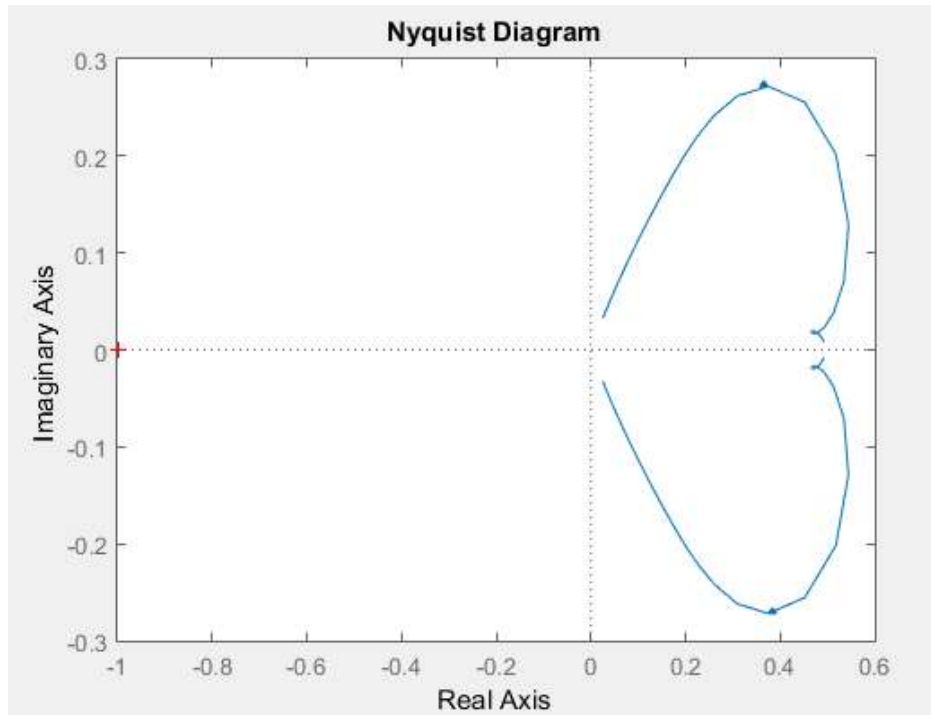
Şekil 6.2 Bode Grafiği Çıktısı

```
>>nichols (Gc,w)
```



Şekil 6.3 Nichols Eğrisi Çıktısı

```
>>nyquist (Gc,w)
```



Şekil 6.4 Nyquist Eğrisi Çıktısı

## 6.5. Kesir Dereceli Kontrolörlerin Parametrelerinin Tespit Edilmesi

Kesir dereceli sistemlerin kontrolörlerinin tasarlanması geliştirilen arayüz kapsamında tam sayı dereceli kontrolör tasarım modellerinin modifiye edilerek kesir dereceli tasarım modellerine dönüştürülmesiyle mümkün kılınmıştır.

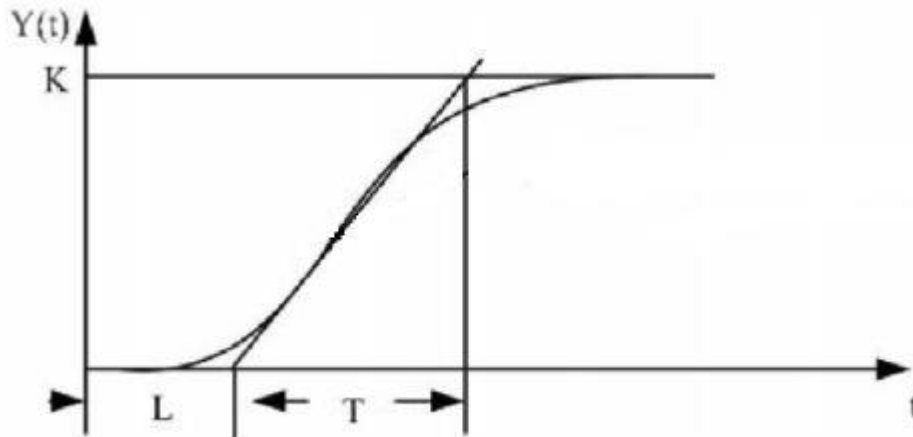
Kesir dereceli  $PI^\lambda D^\mu$  denetleyicilerin beş adet parametresi bulunmaktadır. Bunlar tam sayı dereceli PID denetleyicilerdeki ( $K_p$ ,  $K_i$ ,  $K_d$ ) parametreler ve bu parametrelere ek olarak integral katsayısı  $\lambda$  ve türev katsayısı  $\mu$  değerleridir. [14]

Kesir dereceli  $PI^\lambda D^\mu$  denetleyici tasarımı yapılırken  $K_p$ ,  $K_i$  ve  $K_d$  değerlerini tam sayı dereceli PID denetleyici parametrelerinin tespit yöntemlerini modifiye ederek tespit edilmiştir. Kesir dereceli  $PI^\lambda D^\mu$  denetleyici parametrelerinden  $\lambda$  ve  $\mu$  değerleri Aström'ün geliştirdiği 3. Bölümde anlatılan yöntemle bulunmuştur.

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin parametrelerinden  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri aşağıda sıralanan yöntemler kullanılarak tespit edilmiştir. Herbir yöntem için geliştirilen arayüz üzerinden tek bir buton ile parametrelerin tespit edilebilmesi sağlanmıştır.

- Ziegler-Nichols Açık Çevrim Yöntemi
- Cohen-Coon Yöntemi
- Chien-Reswich Referans Regülasyonu Yöntemi
- Chien-Reswich Bozucu Etkiyi Durdurma Yöntemi

Yukarıda gösterilen parametre belirleme yöntemleri incelendiğinde hepsinin temel parametre belirlemek için uyguladığı metodun aynı olduğu gözlemlenmiştir. Kontrol edilecek sisteme açık çevrim birim basamak girişi uygulanmış elde edilen cevap incelenerek gerekli değerler tespit edilmiştir.(Şekil 6.5)



Şekil 6.5 Ziegler-Nichols Birim Basamak Cevap Eğrisi

Uygulanan birim açık çevrim birim basamak cevabından elde edilen çıkış Şekil 5.5’de gösterilmiştir. Burada elde edilmek istenen değerler T ve L zamanlarıdır. Bu değerler çizilen bir teğet ile bulunmuş ve parametre belirleme yöntemlerinin tablolarında yerlerine konularak  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri tespit edilmiştir.

Bu nokta da açık çevrim birim basamak girişi uygulama işlemini kesir dereceli sistemlere uygulanabilir hale getirildiğinde bu parametre belirleme yöntemlerinin kesir dereceli sistemlerde uygulanabileceği gözlemlenmiştir.

Belirlenmek istenen T ve L zaman değerlerinin tespit edilebilmesi için açık çevrim birim basamak girişi kesir dereceli sistemlere uygulayıp çıkış elde edilmiş. Sonrasında bu elde edilen çıkışa teğet çizme işlemi uygulanarak T ve L tespit edilmiştir.

Yukarıda sıralanan parametre belirleme yöntemleri farklı sistemlere uygulanmıştır. Uygulama işleminde daha önceden literatürde ortaya konan bir sorun ile karşılaşmıştır. Bu yöntemler doğru ve kabul edilebilir limitler içinde çalışması için dikkat edilmesi gereken bir oran vardır. Bu oran T ve L değişkenlerine bağlıdır ve aşağıda gösterilmiştir. (Denklem 6.2)

$$a = \frac{T}{L} \quad (6.2)$$

Denklem 6.2’de gösterilen bu oran yöntemlerin sağlıklı çalışabilmesi için belirli bir değerin üstünde olması gerekmektedir. Denklemde gösterilen a katsayısı  $a \geq 4$  olması durumunda yöntemler doğru sonuç vermektedir. Bu oran dikkate alınarak arayüz tasarlanmıştır. Girilen kesir dereceli sistemin veya tam sayı dereceli sistemin açık çevrim birim basamak cevabı  $a < 4$  olması durumunda arayüz bu yöntemlerin kullanılabilirliğini ortadan kaldırmaktadır. Bu durumda kullanıcı Matlab pidTuner arayüzünde yönlendirilmekte veya parametreleri elle girmesi talep edilmektedir.

### 6.5.1. Ziegler-Nichols Yöntemi

Ziegler-Nichols yöntemi ampirik formüller kullanarak parametreleri belirleme yöntemleri içerisinde araştırmacılar tarafından en sık kullanılan yöntemdir. Parametrelerin belirlenmesi için öncelikle Şekil 6.5’de gösterildiği gibi T ve L zaman değerlerini bulmak gereklidir.[13] Bu işlem için geliştirilen mfunction aşağıda gösterilmiştir.

Yöntem için toplamda iki adet mfunction oluşturulmuştur. Bunlarda OpenLoop.m sadece tam sayı dereceli sistemler için ZieglerNicholsOpen.m kesir dereceli sistemler için geliştirilmiştir.



## Tam Sayı Dereceli Sistemlerin Tve L Değerleri İçin Geliştirilen Komut Dizisi

### OpenLoop.m

```
function [L,T,Ws] = OpenLoop( num,den )
s = tf('s');
sys = tf(num,den);
[y,t] = step(sys);
yp = diff(y);
ypp = diff(y,2);
t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
y_infl = interp1(t,y,t_infl,'linear');
h = mean(diff(t));
dy = gradient(y, h);
[~,idx] = max(dy);
b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
tv = [-b(2)/b(1); (1-b(2))/b(1)];
f = [tv ones(2,1)] * b;
L = tv(1);
T = tv(2) - tv(1);
Ws=sys end
```

## Kesir Dereceli Sistemlerin Tve L Değerleri İçin Geliştirilen Komut Dizisi

### ZieglerNicholsOpen.m

```
function [ L,T ] = ZieglerNicholsOpen( G1 )
t1=0:0.01:10;
u=ones(size(t1));
y=lsim(G1,u,t1);
t=t1(:);
yp = diff(y);
ypp = diff(y,2);
t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
y_infl = interp1(t,y,t_infl,'linear');
h = mean(diff(t));
dy = gradient(y, h);
[~,idx] = max(dy);
b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
tv = [-b(2)/b(1); (1-b(2))/b(1)];
f = [tv ones(2,1)] * b;
L = tv(1);
T = tv(2) - tv(1);
end
```

Buradan elde edilmek istenen T ve L değerleri örnek olarak seçilen sistem üzerinden aşağıda gösterilmiştir.

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>> [ L,T ] = ZieglerNicholsOpen( G )
L =0.0100      T =0.2962
```

Burada dikkat edilirse T ve L oranları yöntemin doğru çalışması için arzu edilen aralıktadır.

Elde edilen T ve L zaman değerleri bu yöntem için hazırlanan ampirik formüllerden oluşan tabloya götürülerek değerler tespit edilir. Bu işlemi otomatik olarak yapması için bir mfunction tasarlanmıştır. Bu mfunction içeriğine T ve L zaman değerleri gönderilerek  $K_p$  ,  $K_i$  ,  $K_d$  parametreleri elde edilmektedir. Bu komut dizisi içerisinde Tablo 2.2'deki ampirik formüller kullanılmaktadır.

### Kontrol Parametrelerinin Tespiti İçin Geliştirilen Komut Dizisi (Ziegler Nichols)

#### ZieglerNicholsOpenLoop.m

```
function [ Kp,Ki,Kd ] = ZieglerNicholsOpenLoop( L,T )
a = L/T;
Kp = 1.2/a;
Ti = 2*L;
Td = L/2;
Ki = Kp/Ti;
Kd = Kp*Td;
End
```

Parametrelerin tespit edilme işlemi örnek olarak verilen sistem üzerinden aşağıdaki gibi gösterilmiştir.

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>> [ L,T ] = ZieglerNicholsOpen( G )
>> [ Kp,Ki,Kd ] = ZieglerNicholsOpenLoop( L,T )
Kp =35.5410
Ki =1.7771e+03
Kd =0.1777
```

Tespit edilen parametreler ile sistemi kontrol edecek PID denetleyici tasarımı gerçekleştirmek için ZieglerNicholsOpenLoopPID.m isimli bir mfunction tasarlanmıştır. Tasarlanan komut dizisi aşağıdaki gibidir.

## Tespit Edilen Parametreler İle PID Denetleyici Tasarımı İçin Geliştirilen Komut Dizisi

### ZieglerNicholsOpenLoopPID.m

```
function G2 = ZieglerNicholsOpenLoopPID( Kp,Ki,Kd )
G2 = fotf([1 0],[1 0],[Kd,Kp,Ki],[2 1 0]);
end
```

Denetleyici tasarımı için tasarlanan mfunction komut dizisinin uygulanması ve denetleyici transfer fonksiyonu çıktısı aşağıdaki gibidir.

```
>> G2 = ZieglerNicholsOpenLoopPID( Kp,Ki,Kd )
0.17771s^{2}+35.541s+1777.0518
-----
s
```

Ziegler-Nichols açık çevrim yöntemi ile belirlenen  $K_p$  ,  $K_i$  ,  $K_d$  parametreleri kullanılarak kontrolör transfer fonksiyonu elde edilmiştir.

Kontrol edilecek sistemin kesir dereceli bir sistem olması Matlab programı içeriğinde bulunan kontrol sistemleri için geliştirilmiş simülasyon komutları çalışmamaktadır. Bu durumda bu bölümde anlatılan feedbackfotf.m ve lsim.m mfunctionları kullanılarak ZieglerNicholsOpenLoopFOTFsim.m isimli bir mfunction tasarlanmıştır. Tasarlanan bu komut dizisinde girdiler sistemin ve kontrolörün transfer fonksiyonlarıdır. Çıktılar sistem ve kontrolörün seri bağlı bir şekilde birim geri besleme ile kapalı çevrim transfer fonksiyonu ve birim basamak cevabıdır. Komut dizisi aşağıda gösterilmiştir.

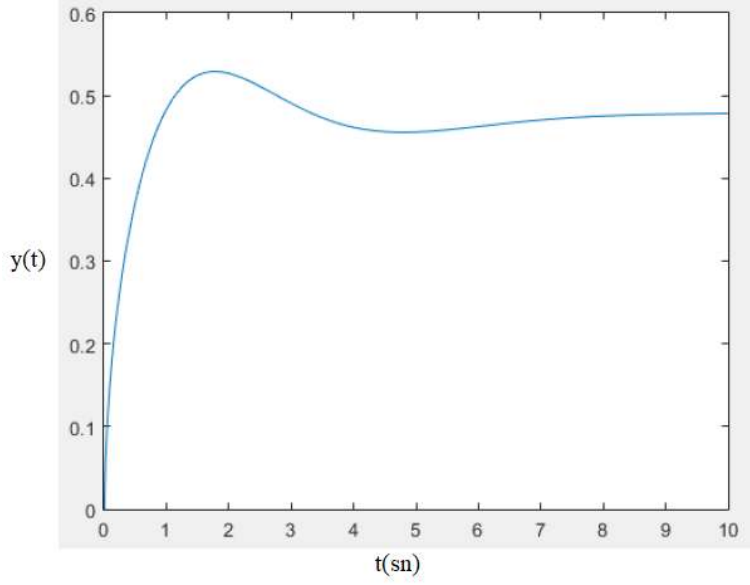
## Sistem Kapalı Çevrim Birim Basamak Çıktısı İçin Geliştirilen Komut Dizisi

### ZieglerNicholsOpenLoopFOTFsim.m

```
function [ Gc ] = ZieglerNicholsOpenLoopFOTFsim( G1,G2 )
sys = G1*G2;
Gc = feedbackfotf(sys,1);
t1=0:0.01:10;u=ones(size(t1));
y=lsim(Gc,u,t1);
plot(t1,y)
end
```

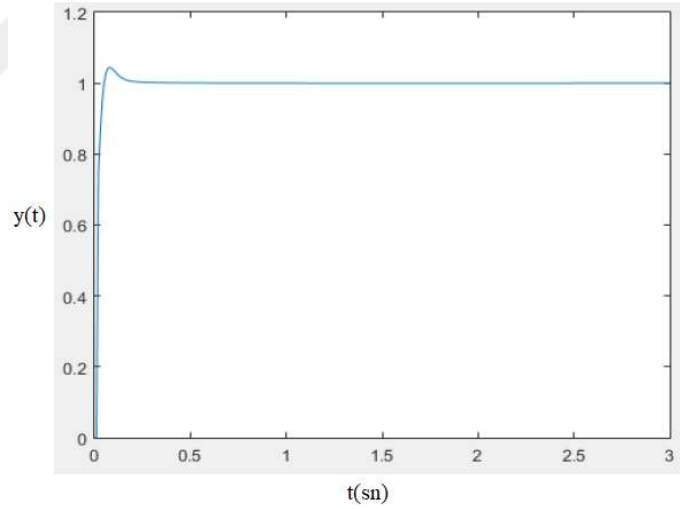
Simülasyon için tasarlanan komut dizisinin kullanım şekli yukarıda gösterilmiştir. Bununla birlikte sistemin kapalı çevrim birim basamak cevabı ile tasarlanan kontrolör ile kontrol edilmiş hali karşılaştırılmıştır.(Şekil 6.6, Şekil 6.7)

Örnek olarak seçilen kesir dereceli sisteminin birim basamak giriş için kapalı çevrim cevabı Şekil 6.6'da gösterilmiştir.



**Şekil 6.6** Kapalı Çevrim Sistemim Birim Basamak Cevabı (Ziegler Nichols)

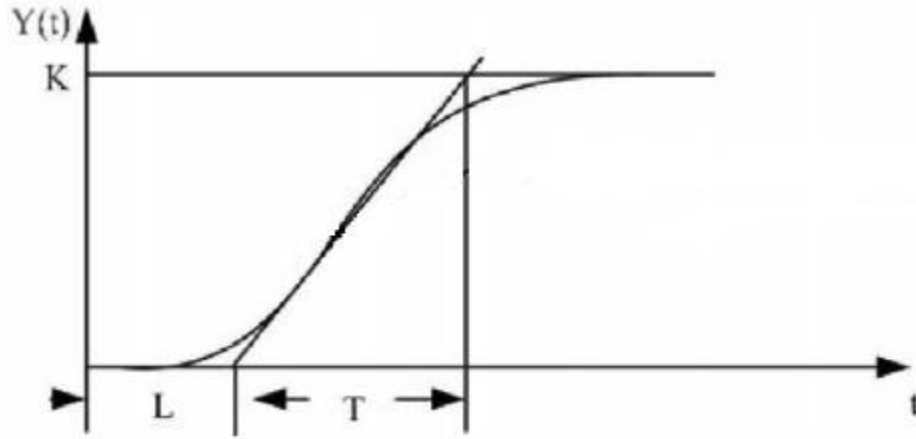
Örnek olarak seçilen kesir dereceli sistemin komut dizileri kullanılarak kontrolör ile kontrol edilmiş hali Şekil 6.7’da gösterilmiştir.



**Şekil 6.7** Birim Basamak Cevabı (Ziegler Nichols)

### 6.5.2 Cohen-Coon Yöntemi

Cohen-Coon yöntemi de aynı Ziegler-Nichols yöntemi gibi kontrol edilecek sisteme açık çevrim birim basamak girişi uygulamaktadır. Ziegler-Nichols ile benzer şekilde sistem cevabından elde edilen T ve L değerlerinin tespitine dayanmaktadır.[12] (Şekil 6.8)



**Şekil 6.8** Ziegler-Nichols Birim Basamak Cevap Eğrisi

Cohen-Coon yönteminde tespit edilmesi gereken T ve L değerleri için kesir dereceli sistemlerde kullanılabilecek bir mfunction tasarlanmıştır. Tasarlanan mfunction girdisi kontrol edilecek kesir dereceli sistem çıktıları ise T ve L zamanının değerleridir. Bu işlem için tasarlanan CohenCoon.m komut dizisi aşağıdaki gibidir.

#### **T ve L Değerleri İçin Geliştirilen Komut Dizisi (Cohen-Coon)**

##### **CohenCoon.m**

```
function [ L,T ] = CohenCoon( Gs )
t1=0:0.01:10;
u=ones(size(t1));
y=lsim(Gs,u,t1);
t=t1(:);
yp = diff(y);
ypp = diff(y,2);
t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
y_infl = interp1(t,y,t_infl,'linear');
h = mean(diff(t));
dy = gradient(y, h);
[~,idx] = max(dy);
b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
tv = [-b(2)/b(1); (1-b(2))/b(1)];
f = [tv ones(2,1)] * b;
L = tv(1);
T = tv(2) - tv(1);
end
```

CohenCoon.m komut dizisinin örnek sistem üzerinden T ve L değerlerinin bulunması işlemi aşağıda gösterilmiştir.

```

>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>> [ L,T ] = CohenCoon( G )
L =0.0100
T =0.2962

```

Elde edilen T ve L zaman değerleri bu yöntem için hazırlanan ampirik formüllerden oluşan tabloya götürülerek değerler tespit edilir. Bu işemin otomatik olarak gerçekleştirilmesi için bir mfunction tasarlanmıştır. CohenCoonPIDparameter.m isimli bu mfunction komut dizisinin girdileri T ve L zaman değerleri çıktılar ise  $K_p$ ,  $K_i$ ,  $K_d$  parametreleridir. Tasarlanan komut dizisi aşağıdadır.

### Kontrol Parametrelerinin Tespiti İçin Geliştirilen Komut Dizisi (Cohen-Coon)

#### CohenCoonPIDparameter.m

```

function [ Kp,Ki,Kd ] = CohenCoonPIDparameter( L,T )
Kp = (T/L)*(0.75 + (L/(4*T)));
Ti = L * (((32*T)*(6*L))/((13*T)*(8*L)));
Td = ((4*L*T)/((11*T)+(2*L)));
Ki = Kp/Ti;
Kd = Kp*Td;
End

```

Parametrelerin tespit edilme işlemi örnek olarak verilen sistem üzerinden aşağıdaki gibi gösterilmiştir.

```

>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>> [ L,T ] = CohenCoon( G )
>> [ Kp,Ki,Kd ] = CohenCoonPIDparameter( L,T )
Kp =22.4631
Ki =1.2168e+03
Kd =0.0812

```

Tespit edilen  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri kullanılarak sistemi kontrol edecek PID denetleyici tasarımı için CohenCoonPIDcontroller.m isimli mfunction komut dizisi tasarlanmıştır. Bu komut

dizisinin girdileri  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri çıktı ise kontrolör transfer fonksiyonudur. Tasarlanan komut dizisi aşağıdadır.

### Tespit Edilen Parametreler İle PID Denetleyici Tasarımı İçin Geliştirilen Komut Dizisi

#### CohenCoonPIDcontroller.m

```
function Gc = CohenCoonPIDcontroller( Kp,Ki,Kd )
Gc = fotf([1 0],[1 0],[Kd,Kp,Ki],[2 1 0]);
End
```

Denetleyici tasarımı için tasarlanan mfunction komut dizisinin uygulanması ve denetleyici transfer fonksiyonu çıktısı aşağıdaki gibidir.

```
>> Gc = CohenCoonPIDcontroller( Kp,Ki,Kd )
0.081186s^{2}+22.4631s+1216.7538
-----
s
```

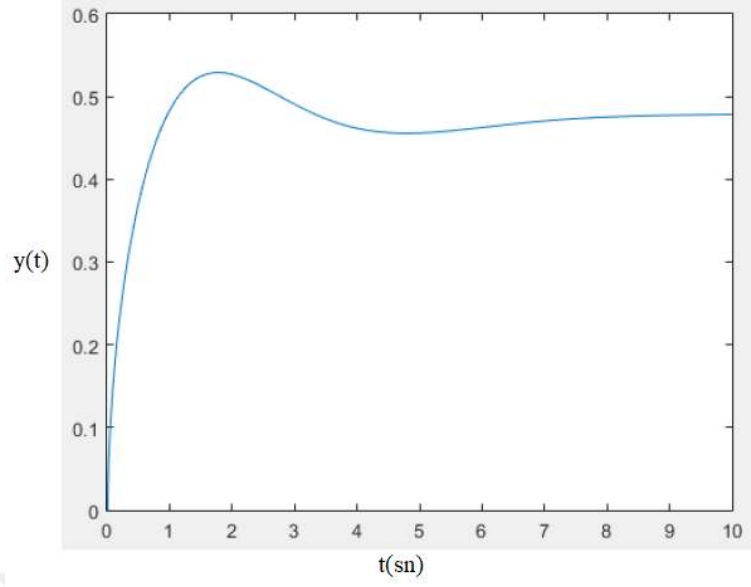
Kontrol edilcek sistemin kesir dereceli olmasından dolayı kontrol sistemleri için geliştirilmiş Matlab komutlarının yetersiz kaldığından Ziegler-Nichols yöntemini anlatırken bahsetmiştir. Yine benzer şekilde Cohen-Coon yönteminin çıktılarının simüle edilmesi için CohenCoonPIDsim.m isimli bir mfunction tasarımı kullanılacaktır. Komut dizisi girdisi sistemin ve kontrolörün transfer fonksiyonlarıdır. Çıktılar sistem ve kontrolörün seri bağlı bir şekilde birim geri besleme ile kapalı çevrim transfer fonksiyonu ve birim basamak cevabıdır.

### Sistem Kapalı Çevrim Birim Basamak Çıktısı İçin Geliştirilen Komut Dizisi

#### CohenCoonPIDsim.m

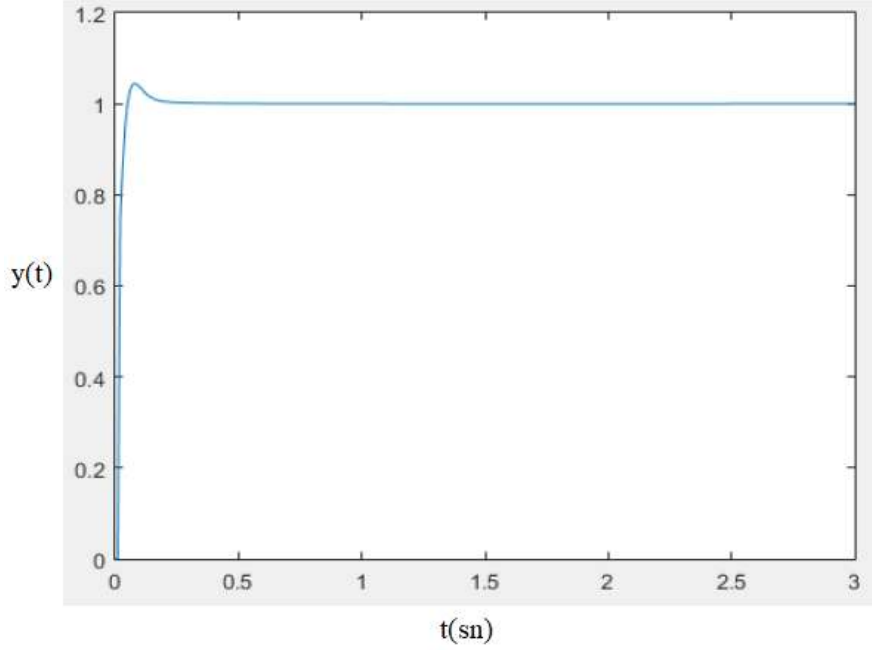
```
function [ Gout ] = CohenCoonPIDsim( Gs,Gc )
sys = Gc*Gs;
Gout = feedbackfotf(sys,1);
t1=0:0.01:10;
u=ones(size(t1));
y=lsim(Gout,u,t1);
plot(t1,y)
end
```

Örnek olarak seçilen kesir dereceli sisteminin birim basamak giriş için kapalı çevrim çıkışı Şekil 6.9'da gösterilmiştir



**Şekil 6.9** Kapalı Çevrim Sistemin Birim Basamak Cevabı (Cohen-Coon)

Örnek olarak seçilen kesir dereceli sistemin komut dizileri kullanılarak kontrolör ile kontrol edilmiş hali Şekil 6.10’da gösterilmiştir.



**Şekil 6.10** Birim Basamak Cevabı (Cohen-Coon)



### 6.5.3.Chien-Hrones-Reswich Yöntemi

Chien-Hrones-Reswich yöntemi ampirik formüllerin amaçlarının farklılaşmasına dayanarak ikiye ayrılır.

Bu ayrım dışında tespit edilmesi gereken analiz parametreleri tamamen aynıdır. Bu yöntemin Matlab üzerinden kesir dereceli sistemlere ve kontrolörlere uygulanması Ziegler-Nichols ve Cohen-Coon yöntemleriyle büyük benzerlik gösterir. Bunun nedeni tespit edilmesi gereken parametreler diğer iki yöntemdeki gibi bu yöntemde de T ve L değerlerinin tespit edilmesi ve arzu edilen aralıkta çalışmasıdır. Bu sebeple bu başlık altında anlatılacak T ve L zaman değerlerinin tespiti sadece bu yöntemin ilk başlığı olan referans regülasyonu başlığı altında anlatılacaktır. Aynı komut dizisi ve çalışma mantığı diğer alt başlık ile tamamen aynıdır.

Bu yöntemin tasarımını aşağıdaki iki başlık içeriğinde incelenmiştir. Bu başlıklar;

- Referans Regülasyonu
- Bozucu Etkiyi Durdurmak

#### Referans Regülasyonu

Referans regülasyonu yönteminde Ziegler-Nichols ve Cohen Coon yöntemlerinde olduğu gibi açık çevrim sistemin birim basamak cevabı kullanılarak T ve L zaman değerleri tespit edilir. Bu amaçla tasarlanan mfunction komut dizisi aşağıdadır.

#### Tve L Değerleri İçin Geliştirilen Komut Dizisi (Chien-Hrones-Reswich)

##### referanceRegulation.m

```
function [ L,T ] = referenceRegulation( Gs )
t1=0:0.01:10;
u=ones(size(t1));
y=lsim(Gs,u,t1);
t=t1(:);
yp = diff(y);
ypp = diff(y,2);
t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0);
y_infl = interp1(t,y,t_infl,'linear');
h = mean(diff(t));
dy = gradient(y, h);
[~,idx] = max(dy);
b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);
tv = [-b(2)/b(1); (1-b(2))/b(1)];
f = [tv ones(2,1)] * b;
L = tv(1);
T = tv(2) - tv(1); end
```

Örnek kesir dereceli sistem için verilen komut dizisinin çalışması aşağıdaki gibidir.

```
>>G=fotf([1.5,0.7,1],[1.8,0.6,0],[1,1],[1.2,0])
      s^{1.2}+1
-----
1.5s^{1.8}+0.7s^{0.6}+1
>> [ L,T ] = referenceRegulation( G )
L =0.0100   T =0.2962
```

Elde edilen T ve L zaman değerleri bu yöntem için hazırlanan ampirik formüllerden oluşan Tablo 2.4'e götürülerek değerler tespit edilir. Bu işlemin gerçekleştirilmesi için de bir mfunction tasarlanmıştır. İlgili Matlab fonksiyonuna ait komut dizisi aşağıdaki gibidir.

### Kontrol Parametrelerinin Tespiti İçin Geliştirilen Komut Dizisi (Cohen-Coon)

#### referenceRegulationParameter.m

```
function [ Kp,Ki,Kd ] = referenceRegulationParameter( L,T )
a = L/T;
Kp = (0.95/a);
Ti = (2.4*T);
Td = (0.42*L);
Ki = Kp/Ti;
Kd = Kp*Td;
end
```

Örnek sistem üzerinde tasarlanan mfunction komut dizisinin çalışması aşağıda gösterilmiştir.

```
>> [ Kp,Ki,Kd ] = referenceRegulationParameter( L,T )
Kp = 28.1367
Ki = 39.5833
Kd = 0.1182
```

Elde edilen  $K_p$ ,  $K_i$ ,  $K_d$  parametrelerine göre kontrolör transfer fonksiyonu için tasarlanan mfunction aşağıdaki gibidir.

### Tespit Edilen Parametreler İle PID Denetleyici Tasarımı İçin Geliştirilen Komut Dizisi

#### referenceRegulationPIDcontroller.m

```
function Gc = referenceRegulationPIDcontroller( Kp,Ki,Kd )
Gc = fotf([1 0],[1 0],[Kd,Kp,Ki],[2 1 0]); end
```

Örnek kesir dereceli sistem üzerinde referenceRegulationPIDcontroller.m komut dizisinin çalışması aşağıdaki gibidir.

```
>> Gc = referenceRegulationPIDcontroller( Kp,Ki,Kd )  
0.11817s^{2}+28.1367s+39.5833  
-----  
s
```

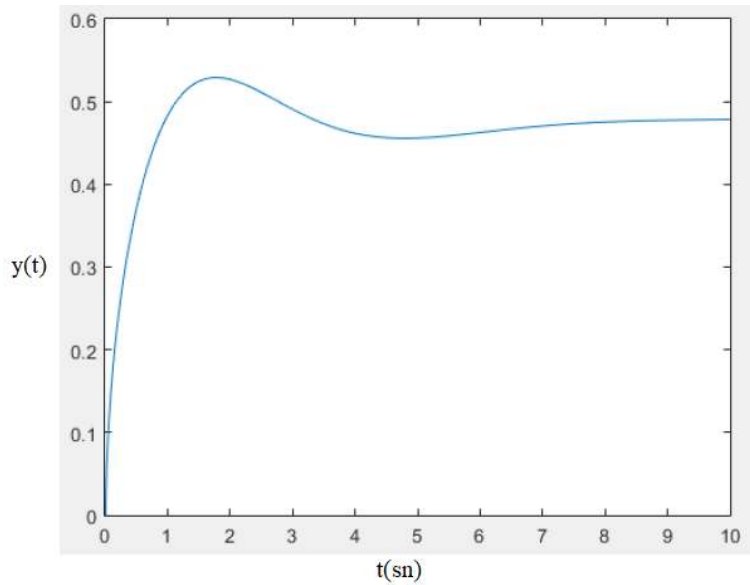
Kontrol edilecek sistemin kesir dereceli sistem olmasından dolayı simülasyon için tasarlanan mfunction komut dizisi aşağıda gösterilmiştir.

### Sistem Kapalı Çevrim Birim Basamak Çıktısı İçin Geliştirilen Komut Dizisi

#### referenceRegulationPIDsim.m

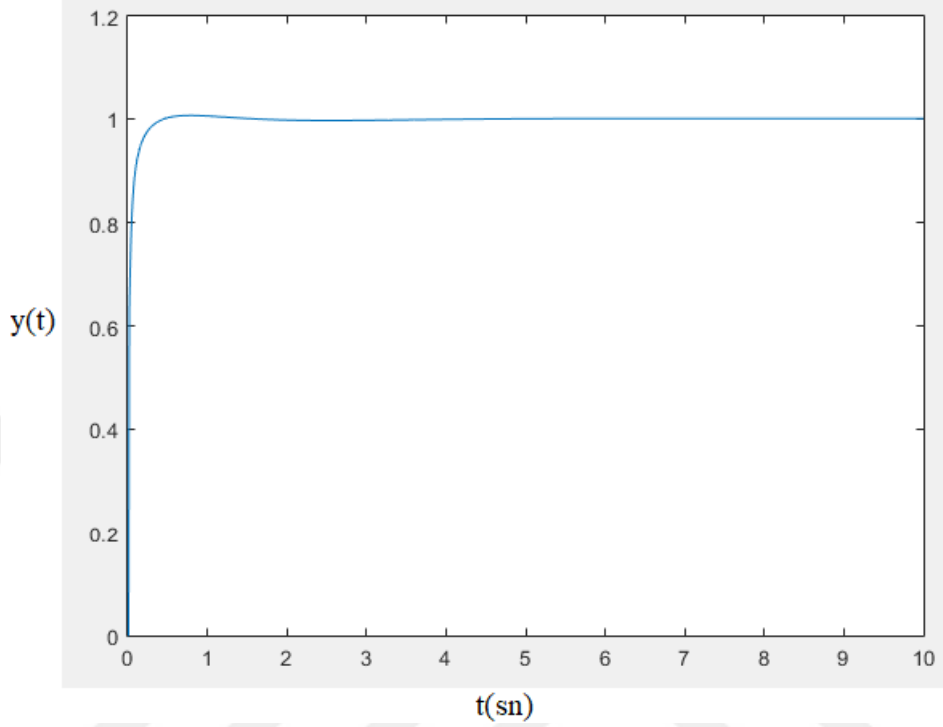
```
function [ Gout ] = referenceRegulationPIDsim( Gs,Gc )  
sys = Gc*Gs;  
Gout = feedbackfotf(sys,1);  
t1=0:0.01:10;  
u=ones(size(t1));  
y=lsim(Gout,u,t1);  
plot(t1,y) end
```

Örnek olarak seçilen kesir dereceli kapalı çevrim sistemin birim basamak cevabı Şekil 6.11'de gösterilmiştir.



Şekil 6.11 Kapalı Çevrim Sistemin Birim Basamak Cevabı (Referans Regülasyonu)

Örnek olarak seçilen kesir dereceli sistemin komut dizileri kullanılarak PID kontrolör ile kontrol edilmiş hali Şekil 6.12’de gösterilmiştir.



Şekil 6.12 Birim Basamak Cevabı (Referans Regülasyonu)

### Bozucu Etkiyi Durdurmak

Bozucu etkiyi durdurmak yönteminin referans regülasyonu yöntemin yönteminden farklılaştığı tek nokta  $K_p$  ,  $K_i$  ,  $K_d$  parametrelerinin tespit edilmesi noktasında farklı amaçla tasarlanmış ampirik formüller içeren bir tablo kullanmasıdır.

Bu yöntem anlatılırken T ve L zaman değerlerinin bulunması noktasından değil  $K_p$  ,  $K_i$  ,  $K_d$  parametrelerinin bulunmasından başlanmıştır. Örnek sistem için  $K_p$  ,  $K_i$  ,  $K_d$  parametrelerin tespit edilmesi amacıyla tasarlanan mfunction komut dizisi aşağıdadır.

### Kontrol Parametrelerinin Tespiti İçin Geliştirilen Komut Dizisi (Bozucu Etkiyi Durdurma)

#### disruptiveEffect.m

```
function [ Kp,Ki,Kd ] = disruptiveEffectParameter( L,T )  
a = L/T;  
Kp = (0.6/a);  
Ti = T;  
Td = (0.5*L); Ki = Kp/Ti; Kd = Kp*Td; End
```

Örnek sistem üzerinde tasarlanan mfunction komut dizisinin çalışması aşağıda gösterilmiştir.

```
>> [ Kp,Ki,Kd ] = disruptiveEffectParameter( L,T )
Kp = 17.7705
Ki = 60.0000
Kd = 0.0889
```

Parametreleri tespit edilen kontrolörün transfer fonksiyonun elde edilmesi amacıyla tasarlanan mfunction komut dizisi aşağıdadır.

### **Tespit Edilen Parametreler İle PID Denetleyici Tasarımı İçin Geliştirilen Komut Dizisi**

#### **disruptiveEffectPIDcontroller.m**

```
function Gc = disruptiveEffectPIDcontroller( Kp,Ki,Kd )
Gc = fotf([1 0],[1 0],[Kd,Kp,Ki],[2 1 0]);
End
```

Örnek kesir dereceli sistem üzerinde referenceRegulationPIDcontroller.m komut dizisinin çalışması aşağıdaki gibidir.

```
>> Gc = disruptiveEffectPIDcontroller( Kp,Ki,Kd )
0.088853s^{2}+17.7705s+60
-----
s
```

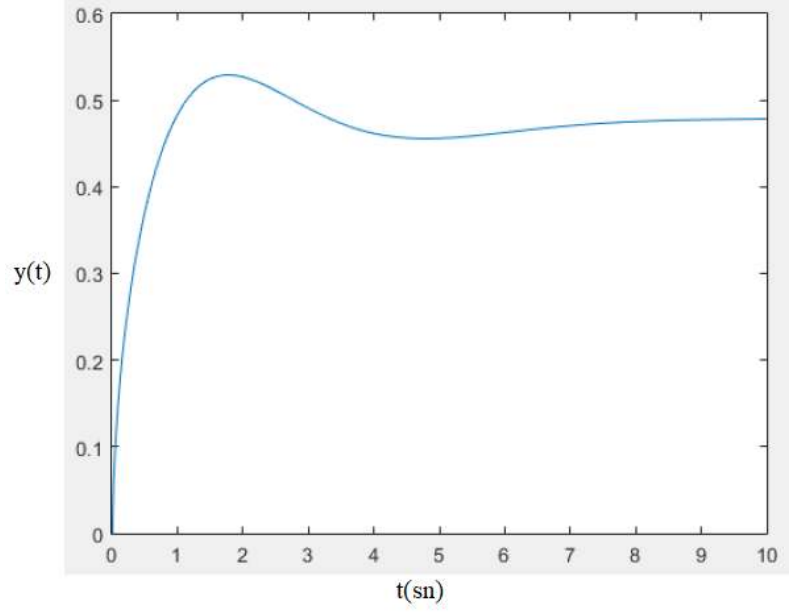
Kontrol edilecek sistemin kesir dereceli sistem olmasından dolayı simülasyon için tasarlanan mfunction komut dizisi aşağıda gösterilmiştir.

### **Sistem Kapalı Çevrim Birim Basamak Çıktısı İçin Geliştirilen Komut Dizisi**

#### **disruptiveEffectPIDsim.m**

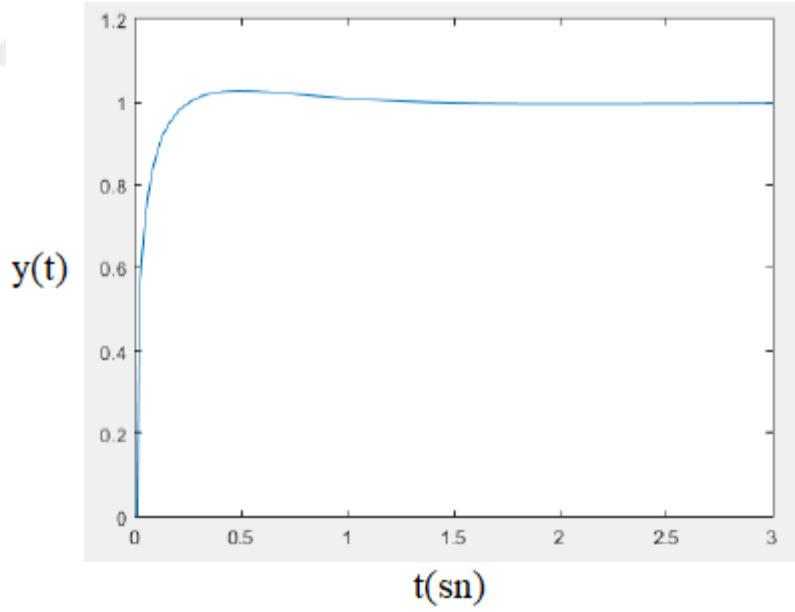
```
function [ Gout ] = disruptiveEffectPIDsim( Gs,Gc )
sys = Gc*Gs;
Gout = feedbackfotf(sys,1);
t1=0:0.01:10;
u=ones(size(t1));
y=lsim(Gout,u,t1);
plot(t1,y)
end
```

Örnek olarak seçilen kesir dereceli sisteminin birim basamak giriş için kapalı çevrim çıkışı Şekil 6.13’de gösterilmiştir.



**Şekil 6.13** Kapalı Çevrim Sistemin Birim Basamak Cevabı (Bozucu Etkiyi Elimine Etmek)

Örnek olarak seçilen kesir dereceli sistemin komut dizileri kullanılarak kontrolör ile kontrol edilmiş hali Şekil 6.14’de gösterilmiştir



**Şekil 6.14** Birim Basamak Cevabı (Bozucu Etkiyi Elimine Etmek)

## 6.6. Arayüz Tasarımı İçin Geliştirilen Komut Dizileri

Kesir dereceli ve tam sayı dereceli sistemlerin kontrolü için kesir dereceli  $PI^\lambda D^\mu$  denetleyici tasarımı bu tezin ana unsurunu oluşturmaktadır. Önceki bölümlerde kesir dereceli sistemlerin Matlab'a tanımlanması işlemi anlatılmıştır. Daha sonra tanımlanan kesir dereceli sistemin klasik PID için denetleyici parametrelerinin tespit yöntemleri üzerinde durulmuştur. Klasik PID denetleyiciden literatürde görüldüğü üzere kontrol kriterlerine daha uygun kesir dereceli  $PI^\lambda D^\mu$  denetleyici tasarlanması gerekmektedir. Kesir dereceli  $PI^\lambda D^\mu$  denetleyicilerin tasarlanması işleminin otomatik olarak gerçekleştirilmesi için bu tez kapsamında bir dizi mfunction komut dizisi oluşturulmuştur.

Böylece geliştirilen arayüz hem tam sayı dereceli sistemler hem de kesir dereceli sistemler için klasik ve kesir dereceli denetleyici tasarımını mümkün kılmıştır. Tasarımı gerçekleştirilen sistemin kontrol kriterlerinin simüle edilmesi için birim basamak cevabının, Bode, Nichols ve Nyquist eğrilerinin gözlemlenmesi sağlanmıştır. [16]

### 6.6.1. Kesir Dereceli $PI^\lambda D^\mu$ Denetleyici Parametrelerinin Tespit Edilmesi

Kesir dereceli denetleyicilerin beş adet parametresi bulunmaktadır. Bunlar  $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$  ve  $\mu$  parametreleridir. Bu parametreden  $K_p$ ,  $K_i$ ,  $K_d$  parametreleri klasik PID parametreleri ile aynıdır. Bu parametreler çeşitli yöntemler ile bulunabilmektedir. Kesir dereceli sistemler için bu tez kapsamında arayüzde yer alan Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswch yöntemleri ile tespit edilebilir hale getirilmiştir. Tam sayı dereceli sistemler için ise bu yöntemlere ek olarak Matlab programının pidTuner isimli arayüzü kullanabilmektedir. [15]

Belirtilen beş parametre içinden  $\lambda$  ve  $\mu$  parametrelerinin bulunması için Matlab programı içerisinde bir komut bulunmamaktadır. Bu parametrelerin tespiti için bölüm 4'de gösterilen Aström yönteminden elde edilen iki denklem kullanılmıştır. Bu işlemlerin otomatik olarak gerçekleştirilmesi ve arayüze adapte edilebilmesi için bir mfunction komut dizisi oluşturulmuştur.

Aström'ün  $\lambda$  ve  $\mu$  parametrelerinin tespit edilmesi için kullanılan denklemler aşağıda Denklem 6.3 ve 6.4 de ifade edilmektedir.

$$f_1(\lambda, \mu) = K_p + K_i w_{cp}^{-\lambda} \cos\left(\frac{\pi}{2} \lambda\right) + K_d w_{cp}^{\mu} \cos\left(\frac{\pi}{2} \mu\right) + K_c \left(\cos(\phi_{pm})\right) = 0 \quad (6.3)$$

$$f_2(\lambda, \mu) = -K_i w_{cp}^{-\lambda} \sin\left(\frac{\pi}{2} \lambda\right) + K_d w_{cp}^{\mu} \sin\left(\frac{\pi}{2} \mu\right) + K_c \left(\sin(\phi_{pm})\right) = 0 \quad (6.4)$$

$\lambda$  ve  $\mu$  değerlerinin hesabı için aşağıdaki komut dizisi oluşturulmuştur.

## $\lambda$ ve $\mu$ Parametrelerinin Tespit Edilmesi İçin Geliştirilen Komut Dizisi

### lamda\_nu.m

```
function [lamda,nu] = lamda_nu( Kp,Ki,Kd )
Kc = Kp/0.6;
Pu = (Kd/(0.075*Kc));
wc = ((2*pi)/Pu);
x0 = [1 1];
x = fsolve(@Astrom,x0);
lamda = x(1);
nu = x(2);
function F = Astrom(x)
F=[Kp+Ki*((wc)^x(1))*cos(pi/2*x(1))+Kd*((wc)^x(2))*cos(pi/2*
x(2))-Kc*cos(pi/4.5);
-Ki*((Kc)^-
x(1))*sin(pi/2*x(1))+Kd*((wc)^x(2))*sin(pi/2*x(2))-
Kc*sin(pi/4.5)];
end
end
```

Tasarlanan lamda\_nu.m mfunction komut dizisinin örnek sistem üzerinden çalışma şekli aşağıdaki gibidir.

```
>> G1=fotf([1 20 12],[2.5,1.1,0],12,0)

12
-----
s^{2.5}+20s^{1.1}+12
>> [ L,T ] = ZieglerNicholsOpen( G1 )
L = 0.1408
T = 1.6581
>> [ Kp,Ki,Kd ] = ZieglerNicholsOpenLoop( L,T )
Kp = 14.1286
Ki = 50.1614
Kd = 0.9949
>> [lamda,nu] = lamda_nu( Kp,Ki,Kd )
lamda = 0.9886
nu = 1.2076
```

$\lambda$  ve  $\mu$  parametreleri tespit edilirken bu örnek için Ziegler-Nichols yöntemi kullanılmıştır. Önceki bölümde anlatıldığı gibi T ve L değerlerinin belirtilen oranın üstünde olması  $\lambda$  ve  $\mu$  değerlerinin sağlıklı seçiminde çok önemlidir. Bu oran belirtilen limitin altında kalması durumunda sistemin sağlıklı çalışmaması söz konusudur. T ve L değerlerinin oranının limitin altında kalması durumunda alternatif parametre bulma yöntemlerine yönlendirme gerçekleştirilmelidir. Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicisinin sağlıklı çalışması en öncelikli olarak



$K_p$  ,  $K_i$  ,  $K_d$  parametrelerinin doğru seçimine bağlıdır. Aksi durumda sistem arzu edilen kontrol kriterleri için çalışmaz. T ve L oranının fazla düşmesi durumunda sistemin salınımına girme riski bulunmaktadır.

### 6.6.2. Kesir Dereceli $PI^\lambda D^\mu$ Denetleyici Transfer Fonksiyonunun Tasarımı

Parametreleri tespit edilen kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin simülasyon işlemlerinin gerçekleştirilmesi için sistem transfer fonksiyonun hazırlanması gerekmektedir. Klasik PID denetleyiciden farklı olarak  $\lambda$  ve  $\mu$  değerlerinin olması kesir dereceli  $PI^\lambda D^\mu$  denetleyicileri klasik PID denetleyicilerden Denklem 6.5 ve Denklem 6.6'daki gibi farklılaştırmıştır. Bu farklılık Matlab programında denetleyici transfer fonksiyonları için hazırlanmış komutları kullanılamaz hale getirmektedir.

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (6.5)$$

$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (6.6)$$

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin transfer fonksiyonu için tasarlanan mfunction komut dizisi aşağıdadır.

### Parametreleri Tespit Edilen Kesir Dereceli PID Denetleyici İçin Geliştirilen Komut Dizisi

#### foPID.m

```
function [ Gc ] = foPID( Kp,Ki,Kd,lamda,nu )
top = lamda+nu;
Gc = fotf([1 0],[lamda,0],[Kd,Kp,Ki],[top,lamda,0]);
end
```

Tasarlanan foPID.m mfunction komut dizisinin örnek sistem üzerinden çalışma şekli aşağıdaki gibidir.

```
Kp =14.1286
Ki =50.1614
Kd =0.9949
lamda =0.9886
nu =1.2076
>> [ Gc ] = foPID( Kp,Ki,Kd,lamda,nu )
0.99488s^{2.1962}+14.1286s^{0.98856}+50.1614
-----
s^{0.98856}
```

### 6.6.3. Kesir Dereceli $PI^{\lambda}D^{\mu}$ Denetleyici Simülasyonu

Tasarım işlemin tamamlanan kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicinin sistem üzerinden simülasyon işleminin gerçekleştirilmesi için yeni bir mfunction komut dizisi yazılmıştır.

Bu noktada dikkat edilirse kontrol edilecek sistem kesir dereceli bir sistem ve denetleyici olarak tasarlanan sistemde kesir dereceli bir sistemdir. Bu nedenle Matlab programının kontrol sistemleri kütüphanesindeki komut dizileri çalışmamaktadır. Sistemin simülasyon yapılabilmesi için tasarlanan komut dizisi önceki bölümlerde tasarım işlemleri anlatılan komut dizileri kullanılarak gerçekleştirilmiştir.

#### Tasarımı Tamamlanan Sistemin Simülasyonu İçin Geliştirilen Komut Dizisi

##### **fotf\_sim.m**

```
function [ G ] = fotf_sim( Gc,Gs,Kp,Ki,Kd,M,x )
if M==1
GG = Gc*Gs;
G = feedbackfotf(GG,1);
t2=0:0.01:x;
u=ones(size(t2));
y=lsim(G,u,t2);
plot(t2,y) end
if M==2
GG = Gc*Gs;
G = feedbackfotf(GG,1);
t2=0:0.01:x;
u=ones(size(t2));
y=lsim(G,u,t2);
GM = fotf([1 0],[1,0],[Kd,Kp,Ki],[2 1 0]);
GGM = GM*Gs;
G = feedbackfotf(GGM,1);
t2=0:0.01:x;
u=ones(size(t2));
y1=lsim(G,u,t2);
plot(t2,y,t2,y1) end
if M == 3
G = feedbackfotf(Gs,1);
t2=0:0.01:10;
u=ones(size(t2));
y=lsim(G,u,t2);
plot(t2,y) end
if M == 4
GM = fotf([1 0],[1,0],[Kd,Kp,Ki],[2 1 0]);
GGM = GM*Gs;
G = feedbackfotf(GGM,1);
t2=0:0.01:x;
u=ones(size(t2));
y1=lsim(G,u,t2);
plot(t2,y1) end end
```

Tasarlanan komut dizisinin girdileri sistem transfer fonksiyonu, kontrolör transfer fonksiyonu,  $K_p$ ,  $K_i$ ,  $K_d$ ,  $M$  ve  $x$  (simülasyon süresi) değerleridir. Burada  $M$  değeri arzı edilen simülasyon çıkış bilgisini göstermektedir.

**M değeri ;**

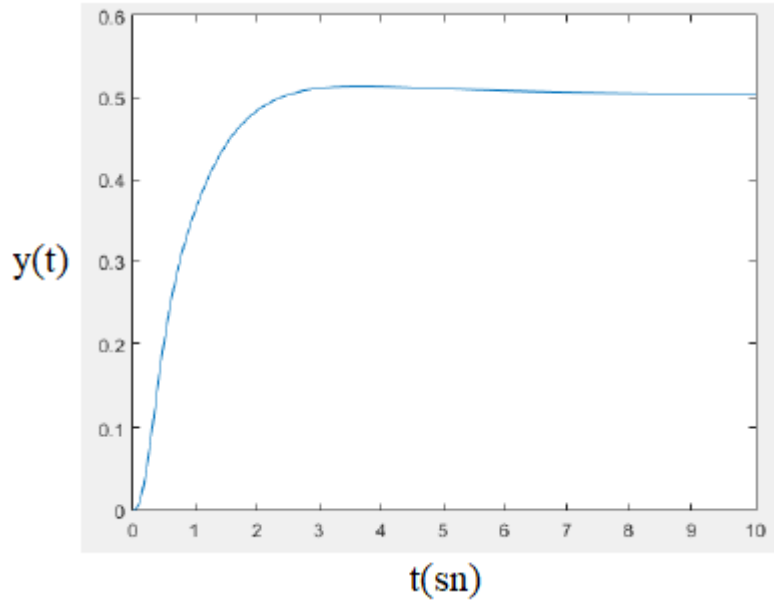
- 1 için; kesir dereceli  $PI^\lambda D^\mu$  denetleyici kullanılarak yapılan kontrol işleminin birim basamak giriş çıktısı.
- 2 için; kesir dereceli  $PI^\lambda D^\mu$  denetleyici ve klasik PID denetleyici kullanılarak yapılan kontrol sisteminin birim basamak giriş için karşılaştırılması.
- 3 için; sistemin birim basamak kapalı çevrim cevabı
- 4 için; klasik PID denetleyici kullanılarak yapılan kontrol işleminin birim basamak giriş çıktısı

Diğer bir parametre olan  $x$  değeri ise tüm bu simülasyon işlemi için arzu edilen simülasyon süresidir.

Tasarlanan fotf\_sim.m mfunction komut dizisinin örnek sistem üzerinden çalışma şekli aşağıdaki gibidir.

Sistemin birim basamak kapalı çevrim cevabı için  $M = 3$ ,  $x = 10$  sn. (Şekil 6.15)

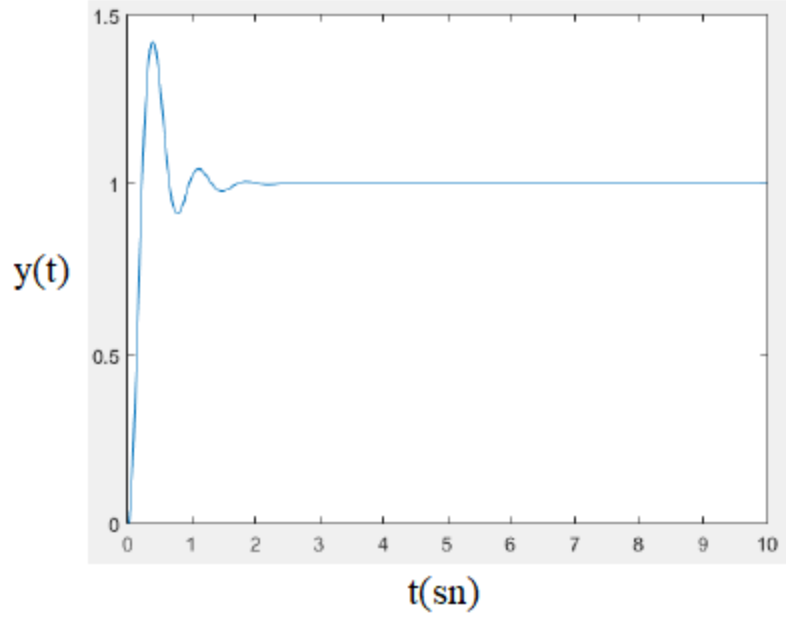
```
>> [ G ] = fotf_sim( Gc,Gs,Kp,Ki,Kd,3,10 )
```



**Şekil 6.15** Kapalı Çevrim Sistemim Birim Basamak Cevabı

Klasik PID denetleyici kullanılarak yapılan kontrol işleminin birim basamak giriş çıktısı için  $M = 4$ ,  $x = 10$  sn. (Şekil 6.16)

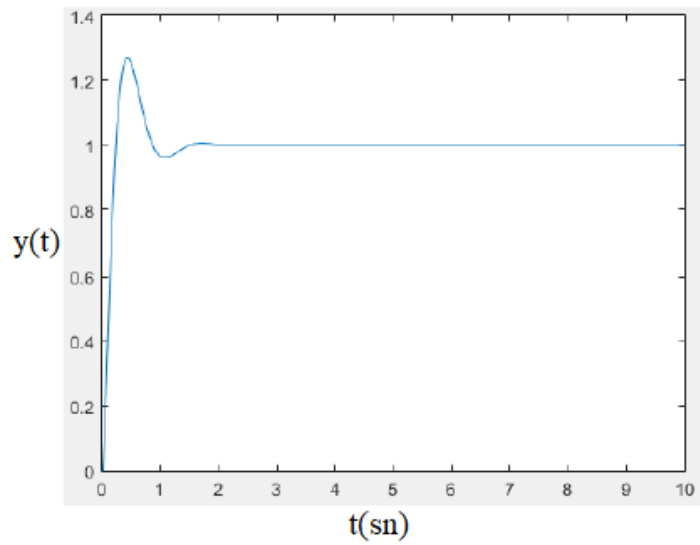
```
>> [ G ] = fotf_sim( Gc,Gs,Kp,Ki,Kd,4,10
```



Şekil 6.16 Klasik PID Denetleyici Cevabı

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici kullanılarak yapılan kontrol işleminin birim basamak giriş çıktısı için  $M = 1$ ,  $x = 10$  sn. (Şekil 6.17)

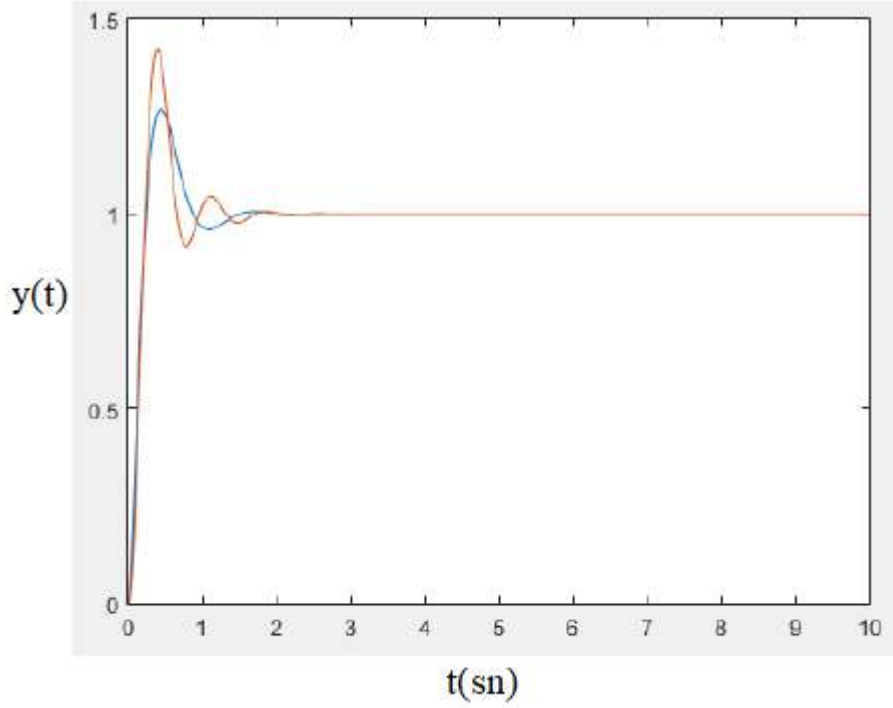
```
>> [ G ] = fotf_sim( Gc,Gs,Kp,Ki,Kd,1,10 )
```



Şekil 6.17 Kesir Dereceli  $PI^{\lambda}D^{\mu}$  Denetleyici Cevabı

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici ve klasik PID denetleyici kullanılarak yapılan kontrol sisteminin birim basamak giriş için karşılaştırılması için  $M = 2$  ,  $x = 10$  sn. (Şekil 6.18)

```
>> [ G ] = fotf_sim( Gc,Gs,Kp,Ki,Kd,2,10
```



Şekil 6.18 Klasik PID ile Kesir Dereceli  $PI^{\lambda}D^{\mu}$  Karşılaştırılması

Arayüz tasarımı için geliştirilen komut dizilerinin isimleri ve kullanım amaçları Tablo 6.1 de listelenmiştir.

Tablo 6.1 Komut Dizileri ve Kullanım Amaçları

| mfunction      | Kullanım Amaçları                                  |
|----------------|----------------------------------------------------|
| fotf.m         | Kesir dereceli sistemin Matlab programına tanıtımı |
| display.m      | Tanımlanan kesir dereceli sistemin gösterilmesi    |
| feedbackfotf.m | Sistemin geri beslemesi                            |
| lsim           | Sistem simülasyonu                                 |
| freqresp.m     | Frekans analizi                                    |

**Tablo 6.1** Komut Dizileri ve Kullanım Amaçları(devamı1)

|                                           |                                                         |
|-------------------------------------------|---------------------------------------------------------|
| <b>bode.m</b>                             | Bode eğrisi                                             |
| <b>nichols.m</b>                          | Nichols eğrisi                                          |
| <b>nyquist.m</b>                          | Nyquist eğrisi                                          |
| <b>Open Loop.m</b>                        | L ve T değerlerinin tespit edilmesi (tam sayı dereceli) |
| <b>ZieglerNicholsOpen.m</b>               | L ve T değerlerinin tespiti (kesir dereceli)            |
| <b>ZieglerNicholsOpenLoop.m</b>           | Yöntem ampirik formüllerin uygulanması                  |
| <b>ZieglerNicholsOpenLoopPID.m</b>        | Yönteme ait PID kontrolörün tasarımı                    |
| <b>ZieglerNicholsOpenLoopFOTFsim.m</b>    | Yöntem simülasyonu                                      |
| <b>CohenCoon.m</b>                        | L ve T değerlerinin tespiti (kesir dereceli)            |
| <b>CohenCoonPIDparameter.m</b>            | Yöntem ampirik formüllerin uygulanması                  |
| <b>CohenCoonPIDcontroller.m</b>           | Yönteme ait PID kontrolörün tasarımı                    |
| <b>CohenCoonPIDsim.m</b>                  | Yöntem simülasyonu                                      |
| <b>referenceRegulation.m</b>              | L ve T değerlerinin tespiti (kesir dereceli)            |
| <b>referenceRegulationParameter.m</b>     | Yöntem ampirik formüllerin uygulanması                  |
| <b>referenceRegulationPIDcontroller.m</b> | Yönteme ait PID kontrolörün tasarımı                    |
| <b>referenceRegulationPIDsim.m</b>        | Yöntem simülasyonu                                      |
| <b>disruptiveEffect.m</b>                 | Yöntem ampirik formüllerin uygulanması                  |

**Tablo 6.1** Komut Dizileri ve Kullanım Amaçları(devamı2)

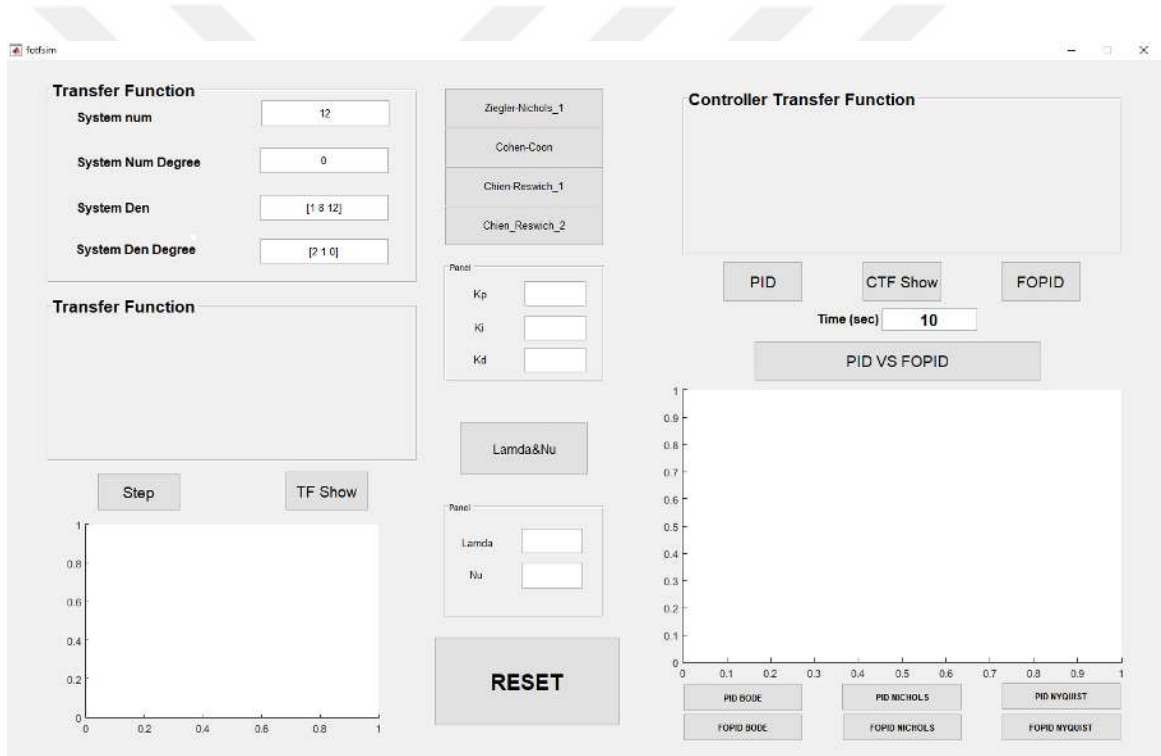
|                                        |                                                 |
|----------------------------------------|-------------------------------------------------|
| <b>disruptiveEffectPIDcontroller.m</b> | Yönteme ait PID kontrolörün tasarımı            |
| <b>disruptiveEffectPIDsim.m</b>        | Yöntem simülasyonu                              |
| <b>Lamda_nu.m</b>                      | $\lambda$ ve $\mu$ değerlerinin tespit edilmesi |
| <b>foPID.m</b>                         | Kesir dereceli denetleyicinin tasarlanması      |
| <b>fotf_sim.m</b>                      | Tasarımı tamamlanan sistemin simülasyonu        |

## 7. MATLAB GUI TASARIMI

Kesir dereceli sistem ve kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici için tasarlanan komut dizilerinin bir arayüz üzerinden kullanım işleminin gerçekleştirilmesi için Matlab arayüzü oluşturulmuştur. Bu amaçla MatlabGui aracı kullanılmıştır. Bu bölümde ise tasarlanan arayüzün tanıtımı yapılmıştır. Geliştirilen arayüzün ismi **fotfsim**arayüzüdür. Matlab üzerinden arayüzün çağırılması aşağıdadır.(Şekil 7.1)

```
>> fotfsim
```

Geliştirilen arayüzün genel görüntüsü Şekil 7.1 de verilmiştir.



Şekil 7.1 Arayüz Ekranı

Arayüz üzerinde önceki bölümlerde tasarımları anlatılan komut dizileri kullanılarak gerçekleştirilen özellikler vardır. Bunlar ;

- Tam sayı dereceli sistemlerin transfer fonksiyonlarının tanıtılması
- Kesir dereceli sistemlerin transfer fonksiyonlarının tanıtılması
- Tanımlanan sistem transfer fonksiyonlarının gösterilmesi
- Sistem transfer fonksiyonlarının birim basamak giriş için kapalı çevrim cevabının grafiksel olarak gösterilmesi



- Tez kapsamında kesir dereceli sistemler için modifiye edilen Ziegler-Nichols, Cohen-Coon, Chien Reswrich yöntemleri kullanılarak denetleyici parametrelerinin tespit edilmesi
- Kesir dereceli  $PI^\lambda D^\mu$  denetleyici parametrelerinden olan  $\lambda$  ve  $\mu$  parametrelerinin Aström yöntemi kullanılarak tespit edilmesi
- Tasarlanan kontrolörün transfer fonksiyonun gösterilmesi
- Arayüze tanımlanan sistem için tasarlanan klasik PID denetleyici simülasyonu
- Arayüze tanımlanan sistem için tasarlanan kesir dereceli  $PI^\lambda D^\mu$  denetleyici simülasyonu
- Klasik PID denetleyici ile kesir dereceli  $PI^\lambda D^\mu$  denetleyiciden elde edilen çıkışların karşılaştırılması
- Simülasyon işleminin ne kadar süre için gerçekleştirileceğinin belirlenmesi
- Sistem için tasarlanan klasik PID denetleyicinin Bode, Nyquist, Nichols eğrilerinin simüle edilmesi
- Sistem için tasarlanan kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin Bode, Nyquist, Nichols eğrilerinin simüle edilmesi

Yukarıda arayüze kazandırılan özelliklerin kullanımı detaylı bir şekilde aşağıda anlatılmıştır.

Üzerinde işlem yapılacak olan tam sayı dereceli veya kesir dereceli sistemin arayüze tanıtılması için transfer fonksiyonuna ilişkin genel gösterim Denklem 7.1’de görülmektedir.

Örneğin ;  $G(s) = \frac{12}{s^{2.5} + 20s^{1.1} + 12}$  için transfer fonksiyonunun arayüze girişi Şekil 7.2’deki gibidir

$$G(s) = \frac{a_n s^{bn} + a_{n-1} s^{bn-1} + a_{(n-2)} s^{b(n-2)} + \dots + a_1 s^{b1}}{c_n s^{dn} + c_{(n-1)} s^{d(n-1)} + c_{(n-2)} s^{d(n-2)} + \dots + c_1 s^{d1}} \quad (7.1)$$

Kesir dereceli sistem için Şekil 7.2’de gösterilmiştir

| Transfer Function |             |
|-------------------|-------------|
| System num        | 12          |
| System Num Degree | 0           |
| System Den        | [1 20 12]   |
| System Den Degree | [2.5 1.1 0] |

Şekil 7.2 Transfer Fonksiyonunun Tanımlanması (Kesir Dereceli)

Tam sayı dereceli sistem için Şekil 7.3’de gösterilmiştir

| Transfer Function |          |
|-------------------|----------|
| System num        | 12       |
| System Num Degree | 0        |
| System Den        | [1 8 12] |
| System Den Degree | [2 1 0]  |

Şekil 7.3 Trasnfer Fonksiyonunun Tanımlanması (Tam Sayı Dereceli)

Tanımlanan sistemlerin transfer fonkiyonlarının gösterilmesi için “TFShow” butonu tanımlanmıştır. Şekil 7.4’de gösterilmiştir

| Transfer Function |             |
|-------------------|-------------|
| System num        | 12          |
| System Num Degree | 0           |
| System Den        | [1 20 12]   |
| System Den Degree | [2.5 1.1 0] |

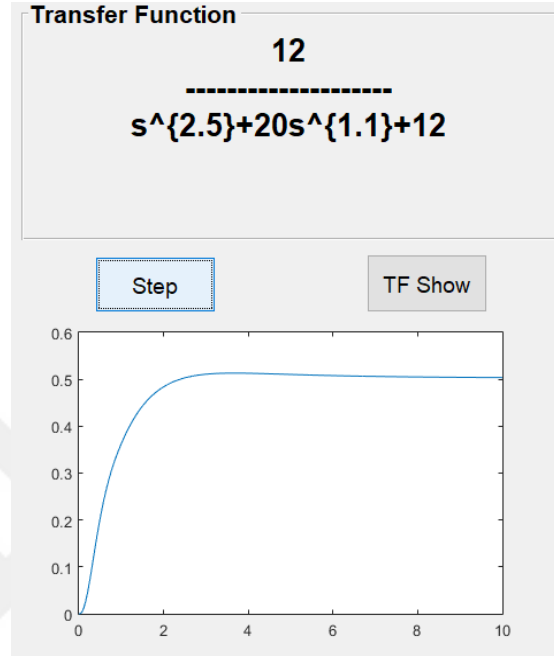
| Transfer Function                 |  |
|-----------------------------------|--|
| $\frac{12}{s^{2.5}+20s^{1.1}+12}$ |  |

Step

TF Show

Şekil 7.4 Transfer Fonksiyonunun Gösterilmesi

Transfer fonksiyonunun gösterim işleminin tamamlanmasından sonra kullanıcı insiyatifiyle kesir dereceli kapalı çevrim sistemin birim basamak cevabı “step” butonu kullanılarak Şekil 7.5’deki gibi görülebilir.



Şekil 7.5 Birim Basamak Giriş İçin Sistemin Kapalı Çevrim Cevabı

Transfer fonksiyonu tanımlanan sistem için uygun kontrolörün  $K_p$ ,  $K_i$ ,  $K_d$  parametrelerinin tespit edilmesi için kesir dereceli sistemlere göre değişiklik yapılan modifiye edilen yöntemlerden birinin seçim işleminin gerçekleştirilmesi gerekmektedir. Bu noktada önceki bölümde anlatılan  $T$  ve  $L$  değerlerinin oranının belirlenen alt limit içerisinde olmaması durumunda kullanıcı ekranına aşağıdaki uyarı mesajı gelmektedir. Şekil 7.6’da gösterilmiştir. Bununla birlikte  $K_p$ ,  $K_i$ ,  $K_d$  parametrelerinin belirlenmesinde kullanılan yöntemleri çalıştıran butonlar kullanılamaz hale gelmektedir. Bu durumda  $K_p$ ,  $K_i$ ,  $K_d$  kontrol parametrelerinin kullanıcı tarafından elle girilmesi gerekmektedir.



Şekil 7.6  $(T/L) < 4$  Durumu İçin Ekrana Gelen Hata Uyarısı

Belirlenen kriterlere uygun T ve L değerleri için sistemin çalışması durumunda modifiye edilen yöntemler için tanımlanan butonlar aktif hale gelerek denetleyicinin  $K_p$  ,  $K_i$  ,  $K_d$  parametreleri Şekil 7.7’de görüldüğü gibi tespit edilebilir.

| Panel |          |
|-------|----------|
| $K_p$ | 14.1286  |
| $K_i$ | 50.1614  |
| $K_d$ | 0.994883 |

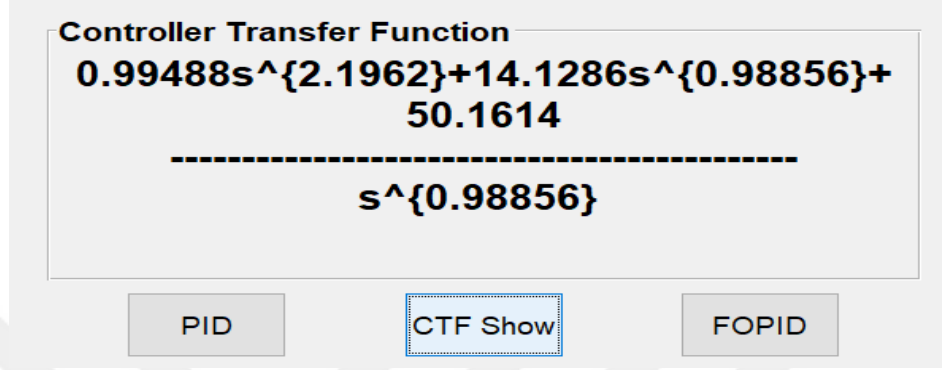
**Şekil 7.7** Denetleyici Parametrelerinin Tespit Edilmesi

Burada örnek olarak Ziegler-Nichols yöntemi gösterilmiştir.  $K_p$  ,  $K_i$  ,  $K_d$  parametreleri tespit edilen sistemin klasik PID denetleyici tasarımı tamamlanmıştır. Kesir dereceli  $PI^\lambda D^\mu$  denetleyici tasarımında gerekli olan  $\lambda$  ve  $\mu$  parametrelerinin bulunabilmesi için “Lamda&nu” butonu tanımlanmıştır. Şekil 7.8’de gösterilmiştir.

| Panel |          |
|-------|----------|
| Lamda | 0.988564 |
| Nu    | 1.20765  |

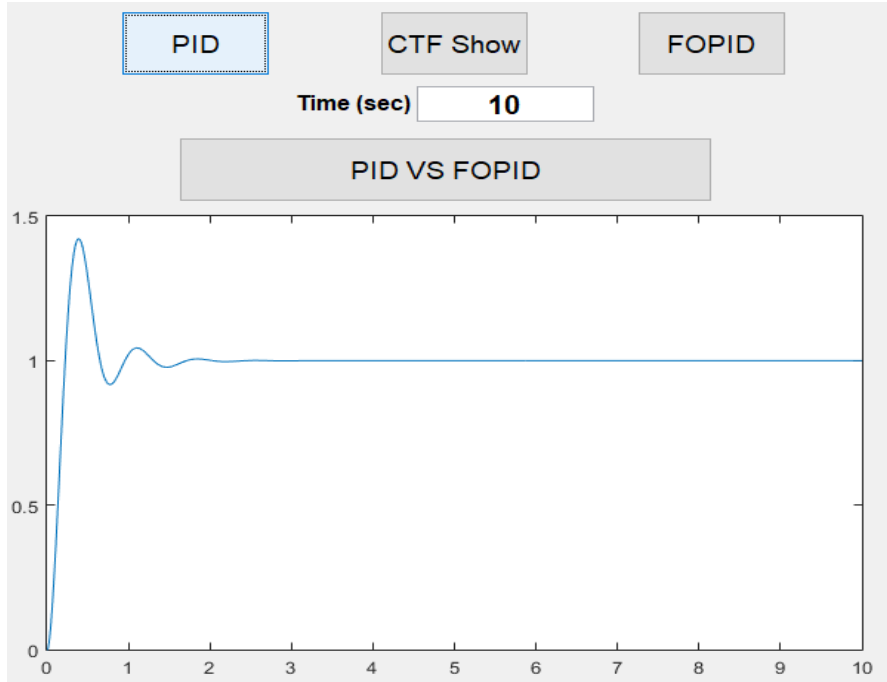
**Şekil 7.8**  $PI^\lambda D^\mu$  Denetleyici Parametrelerinin Tespit Edilmesi

Kesir dereceli  $PI^\lambda D^\mu$  denetleyicinin tasarımı için gerekli beş parametrenin (  $K_p$  ,  $K_i$  ,  $K_d$  ,  $\lambda$  ,  $\mu$  ) tespit edilmesiyle kesir dereceli denetleyicinin transfer fonksiyonu elde edilmiş olur. Elde edilen bu transfer fonksiyonunun gösterimi için “CTF Show” butonu tanımlanmıştır. Şekil 7.9’da örnek bir kullanım verilmiştir.



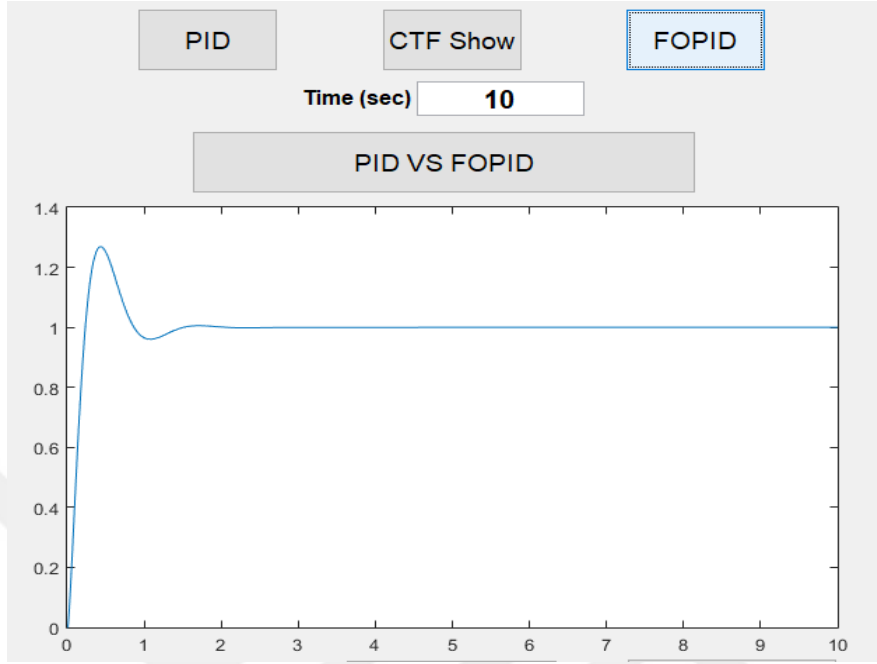
Şekil 7.9 Denetleyici Transfer Fonksiyonunun Gösterilmesi

Böylece kesir dereceli veya tam sayı dereceli sistem için arayüze tanımlama ve denetleyicinin tasarım işlemi sonlandırılmıştır. Denetleyicili sistem için birim basamak cevabı Şekil 7.10’da görülmektedir. Buradaki birim basamak cevabı tam dereceli PID denetleyicinin kullanıldığı durum içindir.



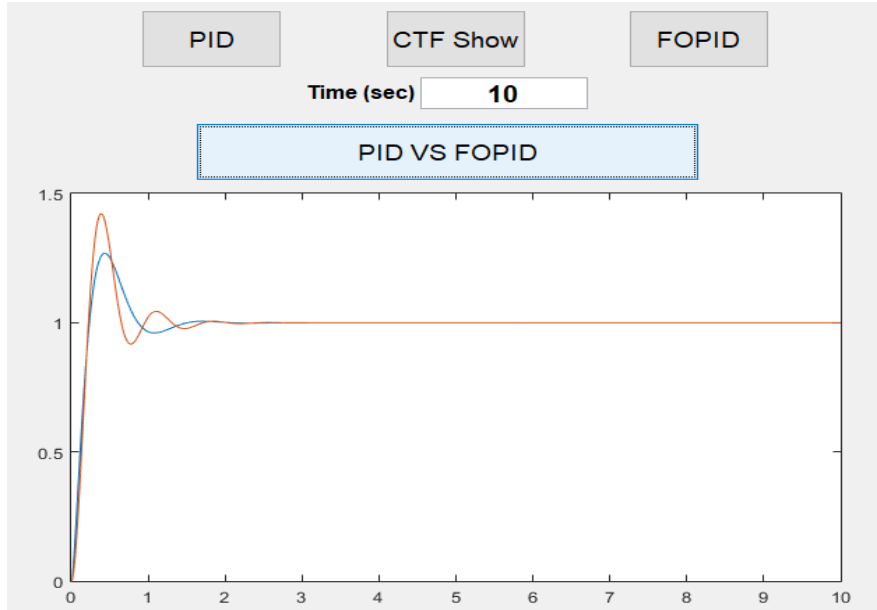
Şekil 7.10 Klasik PID Denetleyici Cevabının Elde Edilmesi

Kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyici için birim basamak cevabı ise Şekil 7.11’de verilmiştir.



Şekil 7.11 Kesir Dereceli  $PI^{\lambda}D^{\mu}$  Denetleyici Cevabının Elde Edilmesi

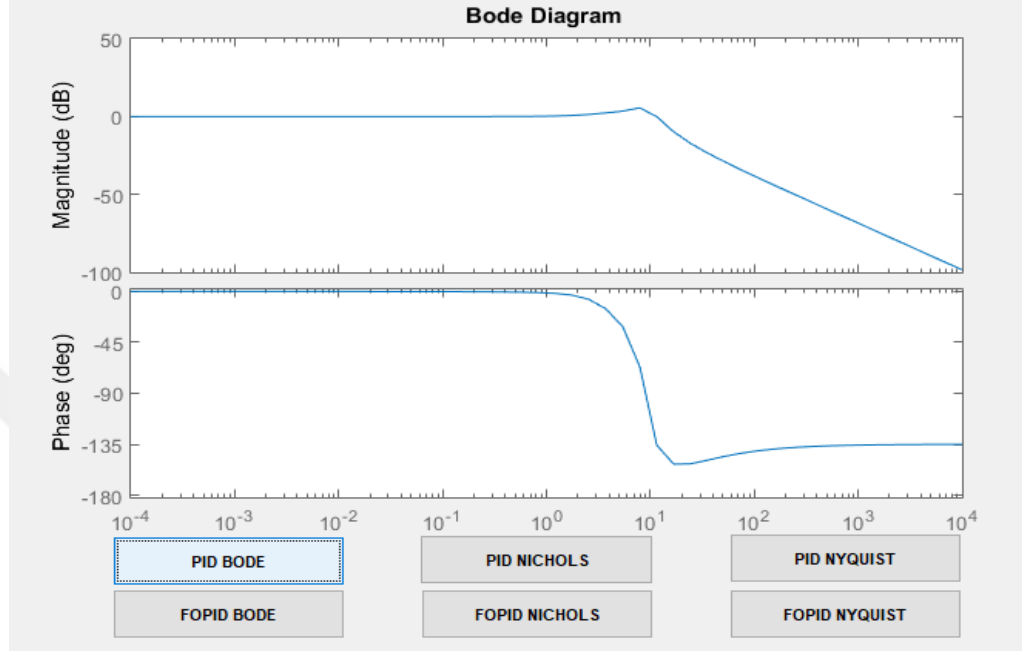
Klasik PID ile kesir dereceli  $PI^{\lambda}D^{\mu}$  denetleyicili bir sistemin birim basamak cevaplarının tek bir grafik üzerinden karşılaştırılma işlemi için “PID VS FOPID” butonu tanımlanmıştır. Şekil 7.12’de gösterilmiştir



Şekil 7.12 Klasik PID ile Kesit Dereceli  $PI^{\lambda}D^{\mu}$  Denetleyicinin Karşılaştırılması

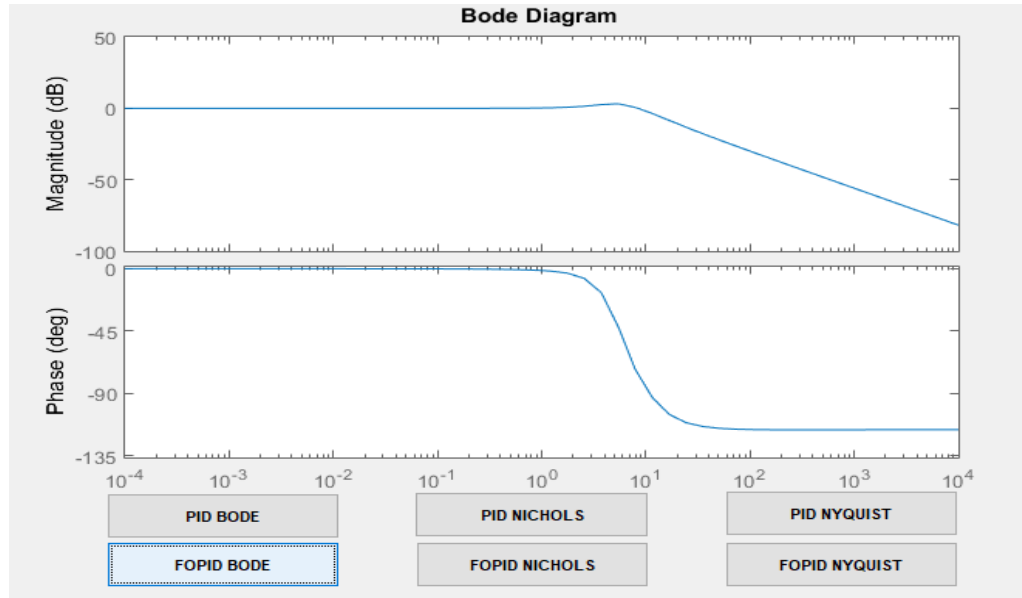
Sistem analizinde kullanılan Bode, Nyquist, Nichols eğrileri için arayüz üzerinde gerekli buton tanımlamaları yapılmıştır.

Klasik PID için bode eğrisi Şekil 7.13’de gösterilmiştir



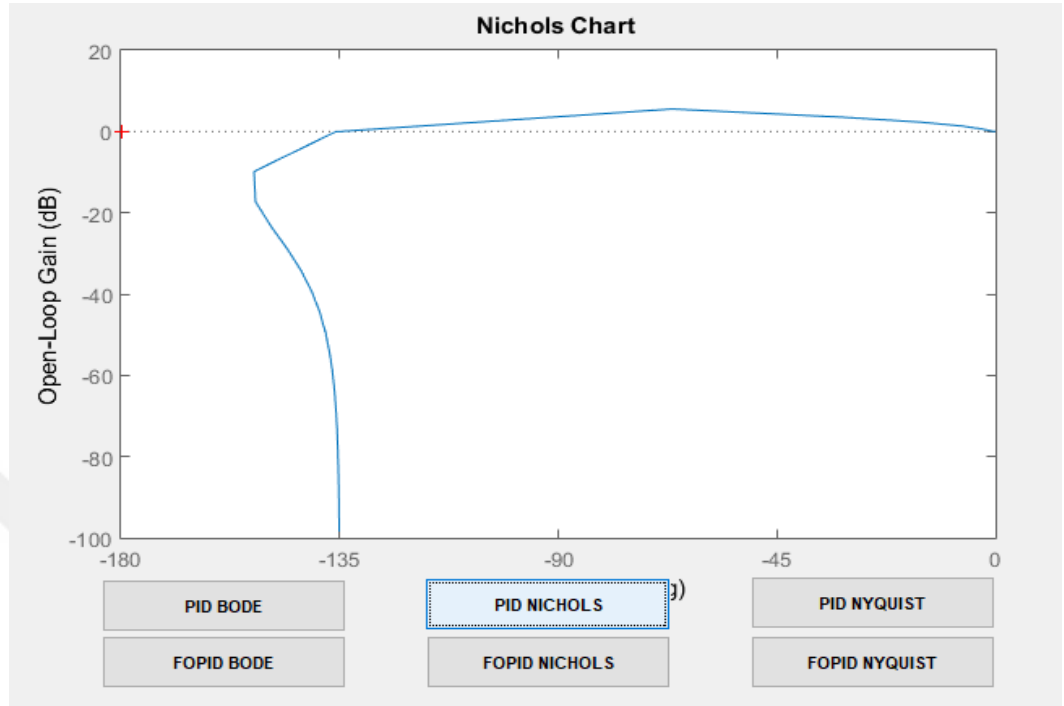
Şekil 7.13 Klasik PID için Bode Analizi

Kesir dereceli  $PI^\lambda D^\mu$  için bode eğrisi Şekil 7.14’de gösterilmiştir



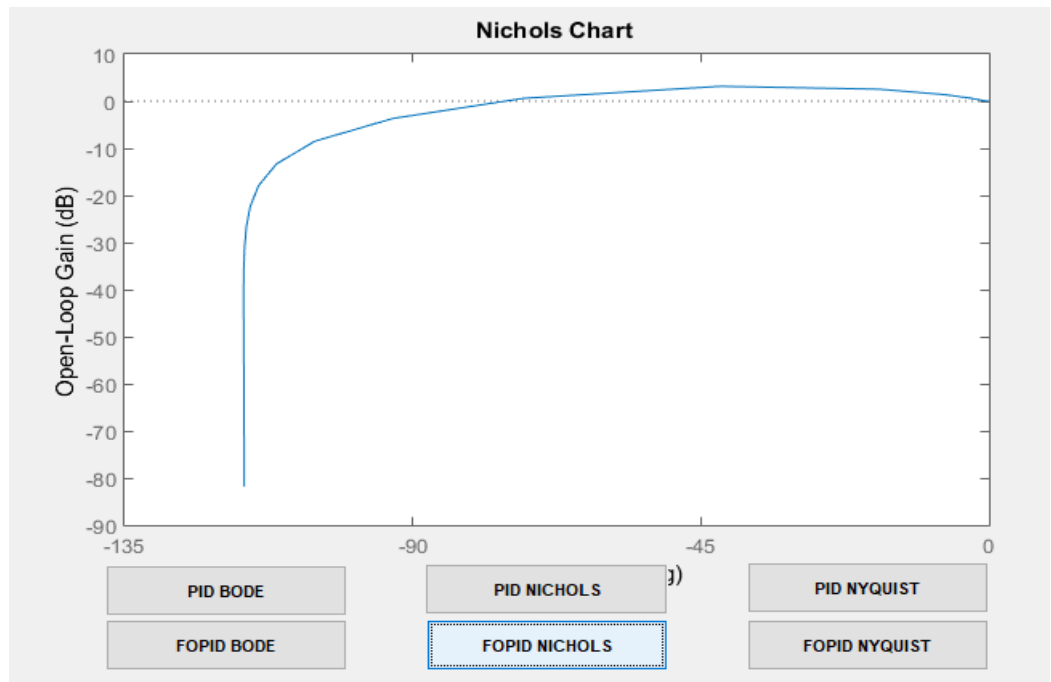
Şekil 7.14 Kesir Dereceli  $PI^\lambda D^\mu$  için Bode Analizi

Klasik PID için Nichols eğrisi Şekil 7.15’de gösterilmiştir



Şekil 7.15 Klasik PID için Nichols Analizi

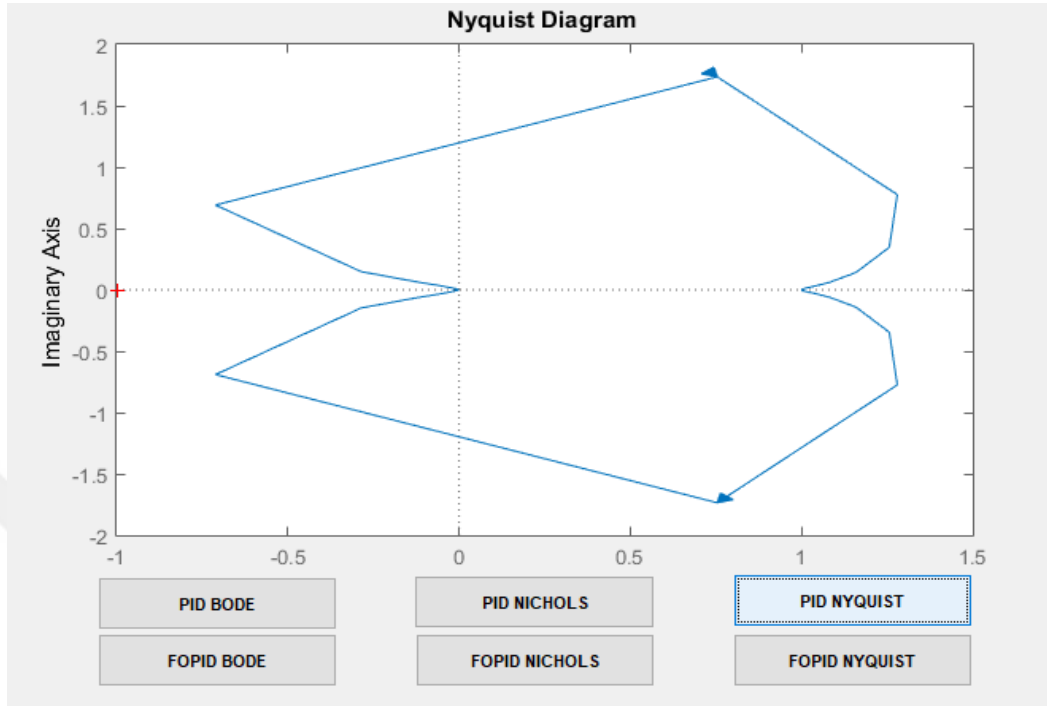
Kesir dereceli  $PI^{\lambda}D^{\mu}$  için Nichols eğrisi Şekil 7.16’de gösterilmiştir



Şekil 7.16 Kesir Dereceli  $PI^{\lambda}D^{\mu}$  için Nichols Analizi

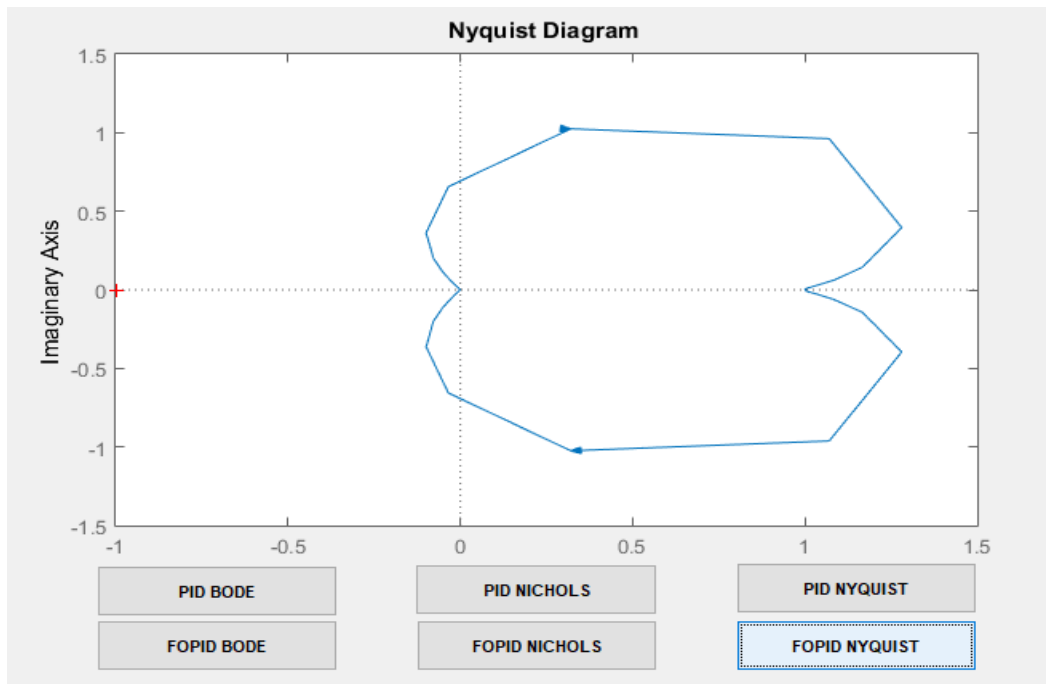


Klasik PID için Nyquist eğrisi Şekil 7.17’de gösterilmiştir



Şekil 7.17 Klasik PID için Nyquist Analizi

Kesir dereceli  $PI^{\lambda}D^{\mu}$  için Nyquist eğrisi Şekil 7.18’de gösterilmiştir



Şekil 7.18 Kesir Dereceli  $PI^{\lambda}D^{\mu}$  için Nyquist Analizi

## 8. SONUÇLAR

Bu tez çalışması kapsamında tam sayı dereceli ve kesir dereceli sistemler için tam sayı dereceli ve kesir dereceli denetleyici tasarımı kullanılabilecek bir arayüz geliştirilmiştir.

Öncelikle kontrol sistemlerinde kullanılan klasik denetleyici türleri hakkında bilgiler verilmiştir. Bunların birbirlerine olan üstünlükleri ortaya konulmuştur. Klasik denetleyici türleri içerisinde temel kontrol kriterleri açısından en verimli denetleyici olan PID denetleyicinin parametrelerinin hesaplama yöntemleri olumlu ve olumsuz yönleriyle açıklanmıştır. Bilgisayar teknolojileri ve hesaplama kapasitesinin yükselmesiyle popüleritesi artan kesir dereceli sistemlerin literatürdeki tanımlarına yer verilmiştir. Kesir dereceli PID denetleyiciler kavramı üzerinde durulmuştur. Klasik PID denetleyici ile kesir dereceli PID denetleyici farklılıkları üzerinde durulmuş, bu farklılıkların kontrol sistemleri açısından önemi ayrıntılı olarak anlatılmıştır. Daha sonra kesir dereceli PID denetleyiciler ile klasik PID denetleyici kıyaslanmıştır. Bu kıyaslama transfer fonksiyonu bilinen bir sistem üzerinde birim basamak giriş için gerçekleştirilmiştir. Kesir dereceli PID denetleyicinin klasik PID denetleyiciye göre temel kontrol kriterleri üzerinde daha verimli olduğu ortaya konulmuştur. Kesir dereceli PID denetleyicilerinin tasarım ve simülasyonu için Matlab programında kullanılabilecek komut dizileri tasarımları yapılmıştır. Bu tasarlanan komut dizileri kesir dereceli sistemlerinin modellerinin Matlab programında kullanılabilir olması ve bu modellere uygun kesir dereceli PID denetleyicilerin tasarlanabilmesini amaçlanmıştır. Elde edilen simülasyon sonuçları ve program çıktılarıyla amaca ulaşıldığı ortaya konulmuştur. Tasarlanan komut dizilerinin tek bir alan üzerinden araştırmacıların kolaylıkla tasarım ve simülasyon işlemlerini yapabileceği şekilde bir arayüz tasarımı Matlab Gui arayüzü kullanılarak gerçekleştirilmiştir.

Bu tez çalışması kapsamında geliştirilen arayüz sayesinde kesir dereceli ve tam sayı dereceli modeller için kesir dereceli denetleyici tasarımı tek bir arayüz üzerinde kolaylıkla yapılabilmektedir. Bu durum kesir dereceli denetleyici tasarımı üzerine çalışan araştırmacılar için çok büyük bir zaman tasarrufu sağlamaktadır.

Gelecekteki çalışmalarda kesir dereceli PID parametre belirleme yöntemlerinde yapay zeka yöntemleri üzerinde durulacaktır. Yapay zeka ile gerçekleştirilecek parametre belirleme sistemleri parametrelerin belirlenmesinde daha optimum sonuçlar ortaya koyacağı düşünülmektedir. Bu çalışma kapsamında geliştirilen arayüz transfer fonksiyonu bilinen bir sistem için tasarım ve simülasyon imkanı sunmaktadır. Yapılması planlanan bir başka çalışma ise arayüze transfer fonksiyonu bilinmeyen sadece blok diyagramları tespit edilen sistemlerin tanımlanabilmesi, tasarım ve simülasyonlarının gerçekleştirilebilmesi olacaktır.

## KAYNAKLAR

- [1] Göktürk, E., (2009). Kesirli PID Tasarım Yöntemi ve Klasik PID İle Karşılaştırmalar, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- [2] Karabalık, F, (2019). Kesir Dereceli Denetleyicilerin Gerçek Zamanlı Tasarımı ve Uygulanması Atatürk Üniversitesi Fen Bilimleri Enstitüsü , Erzurum
- [3] Podlubny, I., (1999). Fractional-order systems and  $PID\mu$ controllers. IEEE Transactions on Automatic Control, 44, 208-214.
- [4] Petras, I., (2000). The fractional-order controllers: Methods for their synthesis and application. arXiv, Kosice.
- [5] Petras, I., (2009). Fractional – Order Feedback Control Of A DC Motor. Journal OfElectrical Engineering, 60 (3), 117–128.
- [6] Yeroğlu, C., (2011). Kesir Dereceli Kontrol Sistemlerinin Frekans Cevaplarının Hesaplanması ve Tasarımı . Doktora, İnönü Üniversitesi Fen Bilimleri Enstitüsü, Malatya.
- [7] Ö, Atan, M, Türk, ve R, Tuntaş,(2013) “Kesir Dereceli PID Denetleyicisinin ElektronikDevre Tasarımı” - Otomatik Kontrol Ulusal Toplantısı, malatya, Türkiye, s: 2013.
- [8] A. Tepljakov, E. Petlenkov and J. Belikov, (2011) FOMCON Fractional-order modeling and control toolbox for MATLAB, 2011 Proceedings of the 18th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES), Gliwice, Poland , pp. 684-689.
- [9] Çelik, V., (2010) Kesir Dereceli Yapıların Doğrusal Olmayan Davranışlar Üzerine Etkileri, Doktora, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
- [10] Kuo, B. C., (1987). Automatic Control Systems.
- [11] Åström, K. J., Hägglund, T. (1995). PID Controllers - Theory, Design, and Tuning. ISA, 2. Baskı, Durham.
- [12] Cohen, G., Coon, G. A. (1953). Theoretical consideration of retarded control. Transactions of ASME, 75 (1), 827–834.
- [13] Çakmak, C., (2011). Ziegler-Nichols PID Kontrolör Parametrelerini Bulanık Tabanlı İyileştirme Yöntemi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- [14] Şenberber, H., (2018). Zeki Optimizasyon Teknikleri ile Sistemleri Kesirli Dereceli Modelleme ve Denetleyici Tasarımı. Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.
- [15] Zhao, C., Xue, D., Chen, Y., (2005). A Fractional Order PID Tuning Algorithm for A Class of Fractional Order Plants. IEEE, 216-221.
- [16] Concepción A. Monje, YangQuan Chen, Blas M. Vinagre, Dingyü Xue, Vicente Feliu(2010). "Fractionalorder Systems and Controls" , Springer Science and Business Media LLC

# ÖZGEÇMİŞ

Kazim Dođuhan SELÇUK

## KİŞİSEL BİLGİLER

**Doğum Yeri** : ELAZIĞ  
**Doğum Yılı** : 1993  
**Uyruđu** : T.C.  
**Adres** :Fırat Üniversitesi Mühendislik Fakóltesi Lojmanları M-3/4  
**E-posta** : doguhansalcuk@gmail.com

## EĞİTİM BİLGİLERİ

**Lisans** : Fırat Üniversitesi, Mühendislik Fakóltesi Fakóltesi, Elektrik Elektronik Mühendisliđi Bölümü, 2012-2017  
**Lise** : Mehmet Kolođlu Anadolu Lisesi, Elazığ,2007-2011