

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**BİRİNCİ DERECE DEN ÖLÜ ZAMANLI SİSTEMLER İÇİN
KESİRLİ DERECELİ Pİ KONTROLÖR TASARIMI**

YÜKSEK LİSANS TEZİ

Emre BALKIŞ

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

HAZİRAN 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**BİRİNCİ DERECEDEKİ ÖLÜ ZAMANLI SİSTEMLER İÇİN
KESİRLİ DERECELİ Pİ KONTROLÖR TASARIMI**

YÜKSEK LİSANS TEZİ

**Emre BALKIŞ
(504121136)**

**Kontrol ve Otomasyon Mühendisliği Anabilim Dalı
Kontrol ve Otomasyon Mühendisliği Programı**

Tez Danışmanı: Prof. Dr. M.Turan SÖYLEMEZ

HAZİRAN 2016

İTÜ, Fen Bilimleri Enstitüsü'nün 504121136 numaralı Yüksek Lisans Öğrencisi Emre BALKIŞ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “BİRİNCİ DERECEDEN ÖLÜ ZAMANLI SİSTEMLER İÇİN KESİRLİ DERECELİ Pİ KONTROLÖR TASARIMI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. M. Turan SÖYLEMEZ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Leyla Gören SÜMER**
İstanbul Teknik Üniversitesi

Prof. Dr. Galip CANSEVER
Yıldız Teknik Üniversitesi

Teslim Tarihi : **26 Nisan 2016**
Savunma Tarihi : **10 Haziran 2016**

ÖNSÖZ

Tez yazım aşamasında ilgi ve desteğini esirmeyen, bilgi ve tecrübesiyle bana yardımcı olan değerli hocalarım Prof. Dr. M. Turan SÖYLEMEZ'e, Araş. Gör. Uğur YILDIRIM'a ve her zaman bana destek olan aileme ve dostlarıma sonsuz teşekkür ederim.

Nisan 2016

Emre BALKIŞ

Elektronik ve Haberleşme Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER.....	vii
KISALTMALAR.....	ix
SEMBOLLER.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET	xvii
SUMMARY	xix
1. GİRİŞ.....	1
2. KESİR DERECELİ HESAPLAMALAR.....	3
2.1 Kesirli Türev-İntegral Tanımı	3
2.2 Kesirli Dereceli Hesaplamanın Özellikleri	4
3. BİRİNCİ DERECEDEKİ SİSTEMLER İÇİN KESİRLİ DERECELİ Pİ KONTROLÖR KATSAYILARININ KARARLILIK BÖLGESİNİN BELİRLENMESİ.....	5
4. BİRİNCİ DERECEDEKİ ÖLÜ ZAMANLI SİSTEMLER İÇİN KESİRLİ DERECELİ Pİ KONTROLÖR KATSAYILARININ KARARLILIK BÖLGESİNİN BELİRLENMESİ	13
4.1 Pade Yaklaşımı	13
4.2 Taylor Serisi	15
4.3 Crude Yaklaşımı	16
5. KESİRLİ DERECELİ SİSTEMLER İÇİN MATLAB FONKSİYONLARININ TANITIMI.....	23
5.1 KDTF için Nesne Tabanlı Programlama	23
5.1.1 KDTF sınıf oluşturma	23
5.1.2 Oluşturulan fonksiyonları programlama	24
5.1.3 KDTF modelleme gösterimi.....	25
5.2 KDTF Nesnelerinin Analizi	26
5.2.1 KDTF'un kararlılık analizi	26
5.2.2 KDTF'un zaman bölgesindeki analizi	26
5.2.3 KDTF'un frekans bölgesindeki analizi	28
6. KESİRLİ DERECELİ Pİ KONTROLÖRÜN KATSAYILARININ BELİRLENMESİ.....	29
6.1 Büyük Patlama- Büyük Çöküş (Big Bang- Big Crunch) Optimizasyon Metodu.....	29
7. SONUÇ VE ÖNERİLER	37
KAYNAKLAR.....	39
EKLER	41
ÖZGEÇMİŞ.....	47

KISALTMALAR

SGSÇ	: Sınırlı Giriş – Sınırlı Çıkış
GKS	: Gerçel Kök Sınırı
KKS	: Karmaşık Kök Sınırı
SKS	: Sonsuz Kök Sınırı
KÇTF	: Kapalı Çevrim Transfer fonksiyon
KDTF	: Kesirli Dereceli Transfer Fonksiyon
BBBC	: Big Bang- Big Crunch
BPBÇ	: Büyük Patlama Büyük Çöküş

SEMBOLLER

${}_aD_t^r$	: Türev-integral Operatörü
r	: Operatör Derecesi
$\Gamma(.)$	: Gamma Fonksiyonu
$R(s)$	: Referans Giriş Transfer Fonksiyonu
$C(s)$	: Kontrolör Transfer Fonksiyonu
$G(s)$	: Sistem Transfer Fonksiyonu
$G_D(s)$	: Ölü Zaman İçeren Sistemin Transfer Fonksiyonu
$Y(s)$	: Çıkış Transfer Fonksiyonu
$Y_D(s)$	: Ölü Zaman İçeren Sistemin Çıkış Fonksiyonu
k_p	: Oransal Kazanç
k_i	: İntegral Katsayısı
λ	: Kesirli İntegral Derecesi
$P_c(s)$	: Kapalı Çevrim Sistemin Karakteristik Denklemi
$P_{Dc}(s)$	: Ölü Zamanlı Kapalı Çevrim Sistemin Karakteristik Denklemi
PI	: Proportional Integral
PID	: Proportional-Integral-Derivative
PI^λ	: Kesirli Dereceli Proportional-Integral
ω	: Frekans
$e^{-L s}$	: Ölü Zaman
ζ	: Aşım Değeri

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 4.1 : $n=1$ ve $n=2$ için pade katsayıları	14
Çizelge 4.2 : Farklı dereceden Pade yaklaşım katsayıları ($L=1$)	21
Çizelge 6.1 : BPBÇ optimizasyon yönteminde ele alınan farklı kriterlere göre elde edilen sonuçlar	34
Çizelge 6.2 : BPBÇ optimizasyon yöntemi ile farklı L değerlerine göre elde edilen parametre değerleri	35

ŞEKİL LİSTESİ

Sayfa

Şekil 3.1 : Negatif geri beslemeli kontrol sistemi.....	5
Şekil 3.2 : Kesirli dereceli PI kontrolör için kararlılık bölgesi ($\lambda=0.5$)	10
Şekil.3.3 : Farklı λ değerleri için Kesirli dereceli PI kontrolör için kararlılık bölgeleri ($\lambda=0.1, 0.55, 1, 1.45, 1.9$)	11
Şekil.3.4 : Kesirli dereceli PI kontrolör için kararlılık bölgesinin sınırlarının 3 boyutlu gösterimi	11
Şekil 4.1 : $L=5$ Zaman gecikmesinin farklı dereceli pade yaklaşımı	15
Şekil 4.2 : Gerçek zaman gecikmesi ve Crude Yaklaşımı ($L=5$)	16
Şekil 4.3 : Birinci dereceden ölü zamanlı (2. dereceden pade yaklaşımı) sistemin kararlılık bölgesi ($\lambda=0.5$)	20
Şekil 4.4 : Birinci dereceden ölü zamanlı sistem için tasarlanacak kesirli dereceli PI kontrolör için kararlılık bölgesinin sınırlarının 3 boyutlu gösterimi ...	20
Şekil 4.5 : Farklı Pade Dereceleri için kontrolör katsayılarının kararlılık bölgesi ($L=1 \lambda=0.5$)	21
Şekil 4.6 : Farklı ölü zaman değerleri için kararlılık bölgesindeki değişim ($\lambda=0.5$)..	22
Şekil 5.1 : Basamak Yanıtı (zaman aralığı 0.01)	27
Şekil 5.2 : Basamak Yanıtı (zaman aralığı 0.01 ve 0.001)	27
Şekil 5.3 : Frekans Bölgesi Grafikleri (Bode Diagram Nyquist Diagram)	28
Şekil 6.1 : Yerleşme zamanı dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı.....	31
Şekil 6.2 : Aşım değeri dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı	32
Şekil 6.3 : Aşım değeri ve yerleşme zamanı aynı anda dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı	33
Şekil 6.4 : Farklı BPBÇ optimizasyon yöntemine göre basamak yanıtları	33
Şekil 6.5 : 6.3'deki birinci dereceden ölü zamanlı sistemin basamak yanıtı	36

BİRİNCİ DERECEDEN ÖLÜ ZAMANLI SİSTEMLER İÇİN KESİRLİ DERECELİ PI KONTROLÖR TASARIMI

ÖZET

Bu çalışmada ilk olarak kesirli dereceli hesaplamaların tarihçesinden ve kesirli türev-integral kavramlarından bahsedilmektedir. Kesirli dereceli PI kontrolörün katsayılarının (k_p , k_i) esneklikleri, klasik PI kontrolör ile benzerlik göstermesine rağmen integral derecesi (λ) farklıdır ve iki yapı arasındaki farklılık buradan kaynaklanmaktadır.

Kesirli dereceli kontrolör tasarlanırken dikkat edilmesi gereken en önemli nokta kontrolörün kapalı çevrim sistemi kararsız yapmamasıdır. Bu yüzden, öncelikle kontrolör katsayıları için sınırlı giriş – sınırlı çıkış (bounded input – bounded output) kararlılığından yararlanılarak katsayıların kararlılık bölgesini bulunur.

İncelenen sistem ölü zaman içerdiğinde kontrolör katsayılarının kararlılık bölgesini araştırmak biraz daha karmaşık hale gelmektedir. Bu yüzden, karmaşıklığı gidermek için ölü zamanın (e^{-Ls}) farklı yaklaşımları ele alınır. İncelenen yaklaşımlardan en uygunu pade yaklaşımıdır. Bu yaklaşım ile elde edilecek olan ölü zamanın derecesi istenildiği gibi ayarlanabilmektedir. Derece ne kadar artarsa ölü zamana o kadar yaklaşmaktadır. Bu yaklaşımın kullanılmasıyla birlikte ölü zaman ve sistem birbirine seri bağlanmış iki sistem gibi ele alınabilmektedir. Bu durumda ölü zaman içeren sistem için tasarlanan kontrolörün katsayılarının kararlılık bölgesi daha kolay bir şekilde bulunmaktadır.

Katsayıların kararlılık bölgesi bulunurken kesirli dereceli PI kontrolörün integral kısmının derecesi (λ) de önem kazanmaktadır. Katsayı parametrelerinin kararlılık bölgesi integral derecesinin değerine göre farklılık göstermektedir. Bu yüzden kontrolörün üç parametresiyle sınırlı olan kararlılık bölgesi bulunabilmektedir.

Kararlılık bölgesi içerisinde katsayıların ve integralin derecesinin farklı değerlerine göre sistemin performansı değişmektedir. Kesirli dereceli sistemlerin simulasyonu MATLAB üzerinde doğrudan yapılamamaktadır. Bu yüzden kesirli dereceli sistemlerin simulasyonunu yapabilmek için gerekli MATLAB araç kutuları araştırılmış olup bu çalışmada gerekli olan fonksiyonlar tanıtılacaktır. Ayrıca bu fonksiyonlar MATLAB üzerinde simulasyon yapılırken kolayca ihtiyaç duyulduğu yerde kullanılabilmektedir.

Bu aşamaya kadar yapılan araştırmalar sistemin istenilen performansı sağlayabilmesi için yol gösterici niteliktedir. Sistemin istenilen performansı sağlayabilmesi için gerekli olan parametreler Büyük Patlama Büyük Çöküş (Big Bang- Big Crunch) optimizasyon algoritmasıyla bulunacak ve elde edilen sonuçlar incelenecektir.

FRACTIONAL ORDER PI CONTROLLER DESIGN FOR FIRST ORDER SYSTEM WITH DEAD TIME

SUMMARY

In this study, history of the fractional order mathematics and concept of fractional derivative – integral is firstly mentioned. Actually, calculus is a branch of mathematics focused on limits, derivatives, integration etc. Conventional calculus deals with differentiation and integration of integer order. Generalization of conventional calculus to any order gave birth to fractional calculus. Since its birth the fractional calculus investigation was mostly mathematical for almost 300 years. In the field of fractional calculus, there were notable contributions from Leibnitz, Euler, Lagrange, Liouville, etc.

During the last decade, the use of fractional order calculus has gained popularity among many research areas. Its theoretical and practical interests are well established nowadays and, its applicability to science and engineering can be considered as an emerging new analytical approach. In this case, the number of applications where fractional calculus has been used rapidly grows. The main reason of the interest is that the new mathematical phenomena allows to describe real objects more accurately than the classical integer order calculus.

One of the areas which fractional calculus has been rediscovered by scientist and engineers is the area of control theory. The success of fractional order controllers is unquestionable with a lot of success due to emerging of effective methods in differentiation and integration of non-integer order equations.

Fractional order controllers have received a considerable attention in the last years both from academic and industrial point of view. In fact, in principle, fractional order calculus provides more flexibility in controller design with respect to the standard controller design. In spite of that flexibility of fractional order PI controller coefficient (k_p , k_i) is nearly similar to flexibility of the classical PI controller coefficient, the integral order (λ) flexibility is completely different from each other and the main difference between the controller structures results from integral order.

However, this is also implies that the tuning of controller can be more complex because the most important issue during designing the fractional order controller is not to make the closed loop system unstable. In this study, an approach to obtain of stabilizing fractional order PI controllers for first order dynamic systems in the parameter plane (k_p , k_i plane) is given. The problem of the determining all stabilizing values of the parameter of a fractional order PI controller is examined that enables the controller parameters to stay within a general stability region. Using this approach, a very fast and simple way of calculating the stabilizing values of fractional order PI controller is provided.

To describe the stability region, it is respectively found stability domain boundaries which are described by real root boundary, complex root boundary, infinite root

boundary. Thus, the complete set of stabilizing parameters of fractional order PI controller is obtained. The algorithm is has a simple, and so is practically useful in analysis and design of fractional order control systems.

When the system contains dead time, it is hard to find stability region of coefficient of fractional order controller because of that dead time is exponential function. Therefore, different approaches of dead time, which are Taylor series expansion, Pade approximation and Crude approximation, are considered in order to solve the complexity. Most suitable approximation of dead time is “Pade Approximation”. The extension of the achieved dead time with this approach can be adjusted as desired. The higher dead time approximation order is, the more likely it is approaches to real dead time. With the use of this approach, system and approximated dead time can be handled as the two systems connected in series. Then, the problem is converted to compute the parameters of the fractional order PI controller stabilizing the given system with dead time. In this case, it is easier to determine the stability region of fractional order controller coefficient for the system including the dead time as the system without dead time.

While the stability region of the fractional order controller coefficients is being determined, the integral order (λ) of the controller has very importance because the stability region is changeable according to the integral order value. Thus, the stability region is bounded with these three parameter (k_p , k_i , λ). Moreover, the dead time value has an importance for determining the stability region. The stability region changes according to the dead time value. While the dead time value increase, the stability region decreases.

Simulation of the fractional order systems is not easy as integer order system. If it is so, how to simulate fractional order systems? Numerical methods for simulating the fractional order system are briefly mentioned so that a beginner can get started quickly. Fractional order control systems are often modeled by fractional order differential equations. Then, A MATLAB class fractional order transfer function is designed, and based on it, a series of overload functions are provided which are useful in the evaluation of block diagram modeling. A mini toolbox is given. The stability test, time and frequency domain analysis of fractional order transfer function models are presented with the use of the toolbox. The required MATLAB functions is also introduced in order to simulate the fractional order systems in this study. the functions in the toolbox can be easily used with the other MATLAB functions. In this way, the fractional order MATLAB functions can be used for different fractional order systems analysis.

The research until this stage is guideline for achieving the desired system performance. The performance of the system is changeable according to different values of the controller parameters in the stability region. The required parameters to provide the desired performance of the system will be found with Big Bang- Big Crunch optimization algorithm. The optimization algorithm is one of the optimization methods nature present such as genetic algorithm and simulated annealing methods. All traditional evolutionary algorithms are heuristic population based search procedures that incorporate random variation and selection. This method consists of two parts which are big bang phase and big crunch phase. In the big bang phase, energy dissipation produces disorder and randomness is the main feature of this phase. In the big crunch phase randomly distributed particles are drawn into an order.

During the searching the required controller parameters to provide the desired system performance, firstly, it is created an initial population randomly from stability region. The creation of the initial population randomly is called the big bang phase. In this phase, the candidate solutions are spread all over the search space (stability region) in an uniform manner. In this phase, population size can be determined as desired. This size can be reduced or increased according to the convergence or number of iterations. Secondly, the big crunch is applied. The big crunch is a convergence operator that has many inputs but only one output which can be named as the center of mass, since the only output has been derived by calculating the center of mass. In this study, the center of mass is determined as a point which minimizes the settling time and overshoot value of the system. After the big crunch phase, the algorithm must create new members to be used as the big bang of next iteration step. This can be done in various ways, the simplest one being jumping to the first step and creating an initial population. The algorithm will have no difference than random search method by so doing since latter iterations will not use the knowledge gained from the previous one. After the second explosion, the center of mass is recalculated. These successive explosion and contraction steps are carried repeatedly until a stopping criterion has been met. The parameters to be supplied to normal random point generator are the center of mass of the previous step and standard deviation. The deviation term can be fixed, but decreasing its value along with the elapsed iterations produces.

The result to be found according to the presented methods will be simulated and examined. Then the simulation and performance analysis results of fractional order PI controller for first order system with dead time will be presented.

1.GİRİŞ

Son yıllarda, kesirli dereceli hesaplama çoğu mühendislik uygulamalarında popülerlik kazandı. Kesirli dereceli hesaplamanın bilime ve mühendisliğe uygulanabilirliği yeni bir analitik bakış açısı meydana getirdi.

Klasik kontrolörler, çoğu endüstriyel uygulamalarda en yaygın kontrolör olmasına rağmen, son yıllarda, kesirli dereceli dinamik sistemler ve kontrolörler kontrol alanında sürekli artan ilgiye sahiptirler.

Kesirli dereceli hesaplama, ilk olarak 1695'te Leibniz ve L'Hospital'ın yarım dereceli türevi bahsetmelerinde görülmektedir. 19 yy.'da, kesirli hesaplama ile ilgili ilk çalışmalar Liouville (1832), Holmgren (1864) ve Riemann (1953) tarafından yapılmıştır.[1] Kesirli dereceli hesaplamanın uygulama alanlarından biri olan kontrol alanında ilk çalışma Tustin tarafından katı nesnelerin konum kontrolü için yapılmıştır. Diğer bazı uygulamalar 1960 yılında Manabe [2] tarafından uygulanmıştır. Diğer taraftan, kesirli dereceli kontrol, kontrol mühendisliği ile gerekli matematik bilgisinin eksikliği ve bu zamandaki limitli hesaplama gücünden dolayı birleştirilememiştir. Son yirmi yılda, araştırmacılar kesirli dereceli diferansiyel denklemleri kullanarak çeşitli sistemleri tam sayı dereceli denklemlere kıyasla daha iyi modelleyebileceklerini buldular. Bu çalışmalarda Oustaloup [3] dinamik sistemlerin kontrolü için kesirli dereceli algoritmayı tanıştırdı. Sonrasında Podlubny [4], [5] PI ve PID kesirli dereceli kontrolörleri λ ve μ ile derecelendirerek genelleştirdi. Ayrıca, Podlubny, kesirli dereceli PI^λ ve $PI^\lambda D^\mu$ kontrolörlerinin klasik PI ve PID kontrolörlere kıyasla daha iyi yanıt verdiğini göstermiştir.

Kontrol mühendisliği, operasyon altındaki sistemi anlamak ile ilgilenir ve aynı zamanda sistem kısıtlamalarının varlığında en iyi sonucu elde etmeye çalışır. Bu yüzden, daha iyi ve daha basit kontrol algoritmaları geliştirmek asıl amacdır ve bu doğrultuda, basitliğinden ve sistemlere kolay bir şekilde uygulanabilirliğinden dolayı PI-PD-PID kontrolörleri kullanırlar. Bu kontrolörler kullanılırken sistemin performansını optimize etmek önemli konulardan biridir. Bu yüzden sistemin daha

iyi performans saęlaması için son zamanlarda kesirli dereceli kontrolör kullanılmaktadır. Bunun nedeni de, kesirli dereceli kontrolörlerin tam sayı dereceli kontrolörlere göre daha esnek yapıda olmasıdır. Ayrıca kontrolör tasarımıda önemli konulardan biri de kapalı çevrim sistemin kararlılığıdır. Kontrolör tasarlanırken bakılacak olan ilk nokta kararlılıktır. Kararlılık saęlandıktan sonra sistemin performansı ve dayanıklılığı göz önüne alınarak kontrolör parametreleri ayarlanır. Sistemin kararlılığı, dayanıklılığı ve performans kriteri için kesirli dereceli kontrolörler daha esnek yapıya sahip olduklarından dolayı tasarımcıya daha çok avantaj saęlamaktadır.

Kesirli dereceli sistemlerin araştırmalarından yararlanılarak ikinci bölümde kesirli dereceli hesaplamalardan bahsedilecektir. Ayrıca bu bölümde kesirli türev ve integral kavramları tanıtılacaktır.

2. KESİR DERECELİ HESAPLAMALAR

Kesir dereceli hesaplama düşüncesi, tam sayı dereceli hesaplamaların gelişmesinden beri yer almaktadır ve 1695'te Leibniz ve L'Hospital'ın yarım dereceli türev bahsetmesiyle ortaya çıkmaya başladı.

Kesirli dereceli hesaplama, türev ve integralin kesir dereceli temel operatöre ${}_aD_t^r$ genelleştirilmiş halidir. Sürekli türev-integral operatörü

$${}_aD_t^r = \begin{cases} \frac{d^r}{dt^r} & R(r) > 0 \\ 1 & R(r) = 0 \\ \int_a^t (d\tau)^{-r} & R(r) < 0 \end{cases}$$

şeklinde tanımlanır. r operatör derecesidir. Genellikle $r \in \mathbb{R}$ 'dir.

2.1 Kesirli Türev-İntegral Tanımı

Kesirli türev-integral için genellikle kullanılan iki tanım Grunwald-Letnikov ve Rieman-Liouville tanımlarıdır. Grunwald-Letnikov tanımı

$${}_aD_t^r f(t) = \lim_{h \rightarrow 0} h^{-r} \sum_{j=0}^{\lfloor \frac{t-a}{h} \rfloor} (-1)^j \binom{r}{j} f(t-jh) \quad (2.1)$$

ve Rieman-Liouville tanımı da

$${}_aD_t^r f(t) = \frac{1}{\Gamma(n-r)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{r-n+1}} d\tau \quad (n-r < r < n) \quad (2.2)$$

olarak ifade edilir ve $\Gamma(\cdot)$ gamma fonksiyonudur.

Mühendislik problemlerini çözerken rutin olarak Laplace transfer metodu kullanılır. (2.2)'de ifade edilen Rieman-Liouville'nin Laplace transferinin formülü

$$\int_0^\infty e^{-st} {}_0D_t^r f(t) dt = s^r F(s) - \sum_{k=0}^{n-1} s^k {}_0D_t^{r-k-1} f(t)|_{t=0} \quad (2.4)$$

$(n-r < r < n)$ ve $s \equiv j\omega$ olmak üzere (2.4)'teki gibi gösterilmektedir.

2.2 Kesirli Dereceli Hesaplamanın Özellikleri

Kesirli türev ve integralin temel özellikleri aşağıdaki gibi gösterilmektedir [6] .

1) $f(t)$ t 'ye bağlı analitik fonksiyon ise, fonksiyonun kesirli türevi ${}_0D_t^\alpha f(t)$, t ve α 'ya bağlıdır.

2) n tam sayı olmak üzere, $\alpha = n$ için, ${}_0D_t^\alpha f(t)$ işlemi klasik türev işlemi ile aynı sonucu verir.

3) $\alpha = 0$ için, ${}_0D_t^\alpha$ birim operatördür ve (2.5)'teki sonucu verir.

$${}_0D_t^\alpha f(t) = f(t) \quad (2.5)$$

4) Kesirli türev ve integral operatörü doğrusal operatördür.

$${}_0D_t^\alpha (a f(t) + b g(t)) = a {}_0D_t^\alpha f(t) + b {}_0D_t^\alpha g(t) \quad (2.6)$$

5) Kesirli türev-integral operatörünün birleşme özelliği bulunmaktadır.

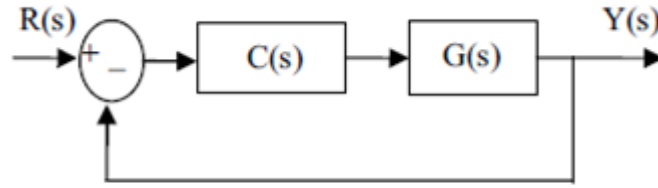
$${}_0D_t^\alpha {}_0D_t^\beta f(t) = {}_0D_t^\beta {}_0D_t^\alpha f(t) = {}_0D_t^{\alpha+\beta} f(t) \quad (2.7)$$

Ayrıca bu özellik tam sayı türev integral operatörü ile de birleşme özelliğine sahiptir.

$$\frac{d^n}{dt^n} ({}_0D_t^r f(t)) = {}_0D_t^r \left(\frac{d^n f(t)}{dt^n} \right) = {}_0D_t^{r+n} f(t) \quad (2.8)$$

3.BİRİNCİ DERECEDEKİ SİSTEMLER İÇİN KESİRLİ DERECELİ PI KONTROLÖR KATSAYILARININ KARARLILIK BÖLGESİNİN BELİRLENMESİ

Negatif geri beslemeli kontrol sistemi şekil 3.1'deki gibi gösterilebilir. R referans girişi, Y sistemin çıkışı, C(s) kontrolör ve G(s) sistemdir.



Şekil 3.1 : Negatif geri beslemeli kontrol sistemi

Kesirli dereceli PI tipindeki kontrolörün transfer fonksiyonu

$$C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s^\lambda} \quad (3.1)$$

olarak tanımlanır. Burada;

k_p oransal kazanç,

k_i integral katsayısı,

λ integral derecesidir. λ reel sayı olup (0,2) arasında herhangi bir reel dereceye sahip olabilir. Eğer $\lambda \geq 2$ olursa, kesirli dereceli PI tipi kontrolör daha yüksek dereceli yapılara dönüşür ve bu da PI kontrolörden tamamen farklıdır.

Eğer $\lambda=0$ alınırsa, kontrolör klasik P tipi kontrolöre dönüşür ya da $\lambda=1$ alınırsa, bu durumda da kontrolör klasik PI tipi kontrolöre dönüşür. Bu yapıdaki kontrolörler özel yapıdaki kesirli dereceli kontrolörlerdir.

Geri beslemeli sistemde yer alan sistemin transfer fonksiyonu (genel yapısı)

$$G(s) = \frac{N(s)}{D(s)} = \frac{b_n s^{\beta_n} + b_{n-1} s^{\beta_{n-1}} + \dots + b_1 s^{\beta_1} + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_1 s^{\alpha_1} + a_0 s^{\alpha_0}} = \frac{\sum_{i=0}^n b_i s^{\beta_i}}{\sum_{i=0}^n a_i s^{\alpha_i}} \quad (3.2)$$

$a_i, b_i \quad \beta_n > \dots > \beta_1 > \beta_0$ and $\alpha_n > \dots > \alpha_1 > \alpha_0$ tam sayılardır.

Şekil 3.1'deki kontrol sisteminin çıkışı $Y(s)$ 'dir.

$$Y(s) = \frac{C(s)G(s)}{1+C(s)G(s)} \quad (3.3)$$

Kapalı çevrim sistemin karakteristik denklemi (3.3)'ün paydasıdır ve “kesir dereceli karakteristik denklem” olarak adlandırılır. Bu denklem, (3.1), (3.2) ve (3.3)'teki denklemler kullanılarak elde edilir.

$$\begin{aligned} P_c(s) &= 1+C(s) G(s) \\ &= E(s) D(s) + U(s) N(s) \\ &= s^\lambda D(s) + (k_p s^\lambda + k_i) N(s) \\ &= \sum_{i=0}^n [a_i s^{(\lambda+\alpha_i)} + k_p b_i s^{(\lambda+\beta_i)} + k_i b_i s^{\beta_i}] \end{aligned} \quad (3.4)$$

Elde edilen kesir dereceli karakteristik denkleminin kararlılığının incelenmesi gerekmektedir. Burada kararlılığı incelerken Routh Hurwitz kararlılık kriterlerinden yararlanılamaz. Bunun nedeni de incelenecek olan denklemin tam sayı dereceli olmamasıdır. Bu yüzden kesirli dereceli karakteristik denklemin kararlılığı için sınırlı giriş – sınırlı çıkış (SGSÇ) (Bounded input – bounded output) kararlılığından yararlanılabilir. SGSÇ kararlılığını, kapalı çevrim sistem transfer fonksiyonunun karakteristik denkleminin hiçbir kökünün s - düzleminin sağ tarafında bulunmaması olarak tanımlanır.

(3.4)'te bulunan kesirli dereceli karakteristik denklem (3.5)'teki gibi genişletilmiş halde gösterebilir.

$$P(s) = p_k s^{q_k} + p_{k-1} s^{q_{k-1}} + \dots + p_1 s^{q_1} + p_0 s^{q_0} = \sum_{i=0}^k p_i s^{q_i} \quad (3.5)$$

p_i katsayı ve q_i de (3.5) 'teki denklemin kesirli dereceleridir ve p_i katsayıları sistemin pay ve paydasının derecesine bağlı olarak k_p ve k_i 'nin fonksiyonu olabilir. Bu yüzden $P(s)$, $P(s ; k_p, k_i, \lambda)$ olarak tanımlanabilir

SGSÇ kararlılığına göre, eksenleri k_p, k_i, λ olan P parametre uzayının alt uzayı olan S kararlılık bölgesindeki $\forall k_p, k_i, \lambda \in S$ noktasının oluşturduğu karakteristik denklemin bütün kökleri s -düzleminin sol yarı bölgesinde yer almalıdır. Parametre düzlemindeki kararlılık sınırlarını araştırırken aşağıda yer alan sınır kriterleri kullanılabilir.

- i) Gerçek Kök Sınırı (GKS) : S kararlılık bölgesinde, kapalı çevrim sistemin (3.5)'te yer alan karakteristik denklemindeki s yerine 0 yazılır. Daha sonra edilen denklem 0'a eşitlenir. $P(0) = 0$ denklemini sağlayan doğru da gerçek kök sınırı olarak tanımlanır.
- ii) Karmaşık Kök Sınırı (KKS) : S kararlılık bölgesindeki bu sınır, s yerine $\pm j\omega$ yazılarak elde edilen denklemin sıfıra eşitlenmesi ile bulunur. Burada yer alan ω değeri $(0, \infty)$ arasında değiştirilerek, eşitliğin gerçek ve sanal kısımları ayrı ayrı sıfıra eşitlenerek bir sınır doğrusu elde edilir.
- iii) Sonsuz Kök Sınırı (SKS) : S kararlılık bölgesinde sonsuz kök sınırı, s yerine ∞ yazılarak $P(\infty) = 0$ eşitliğini sağlayan doğru olarak belirlenir.

Yukarıda tanımlanan sınırları bulmak için, (3.4)'teki kesir dereceli karakteristik denkleme sırasıyla belirtilen tanımlamalar aşağıdaki gibi uygulanır;

- i) (3.4)'teki denklemde s yerine 0 yazılır ve denklem 0'a eşitlendiğinde;

$$k_i = 0 \quad (3.6)$$

olarak bulunur.

- ii) Karmaşık kök sınırını belirlemek için (3.4)'teki denklemde s yerine $j\omega$ konularak elde edilen denklem 0'a eşitlenir.

$$\sum_{i=0}^n [a_i s^{(\lambda+\alpha_i)} + k_p b_i s^{(\lambda+\beta_i)} + k_i b_i s^{\beta_i}]$$

Yukarıda yer alan denklemde, s yerine $j\omega$ konulup 0'a eşitlendiğinde aşağıda yer alan eşitlik elde edilmiş olur.

$$P(j\omega; k_p, k_i, \lambda) = \sum_{i=0}^n [a_i (j\omega)^{(\lambda+\alpha_i)} + k_p b_i (j\omega)^{(\lambda+\beta_i)} + k_i b_i (j\omega)^{\beta_i}] = 0 \quad (3.7)$$

(3.7)'deki denklem karmaşık ve gerçek sayıya sahip olduğu için

$$R\{P(s)\} + Im\{P(s)\} = 0 \quad (3.8)$$

şeklinde ifade edilebilir.

Genel olarak $(\sigma + j\omega)^\zeta$ şeklindeki ifadeler (3.9)'daki gibi ifade edilebilir.

$$(\sigma + j\omega)^\zeta = (\sigma^2 + \omega^2)^{0.5\zeta} [\cos(\zeta \tan^{-1} \frac{\omega}{\sigma}) + j [\sin(\zeta \tan^{-1} \frac{\omega}{\sigma})]] \quad (3.9)$$

Burada σ karmaşık sayının gerçek kısmını, ω sanal kısmını ifade ederken, ζ karmaşık sayının kesir derecesini ifade etmektedir. Bu genel ifadede yer alan σ 'ya, (3.7)'deki denklemde 0 karşılık gelmektedir. (3.9)'daki denklemde yer alan σ 'nın yerine 0 yazıldığında

$$j^\zeta = \cos(\zeta \frac{\pi}{2}) + j [\sin(\zeta \frac{\pi}{2})] \quad (3.10)$$

denklemini elde edilir. (3.7)'de yer alan $P(j\omega ; k_p, k_i, \lambda)$ 'nin içine (3.10)'da yer alan j^ζ 'nin eşleneği yazılırsa aşağıda yer alan denklem elde edilir.

$$\begin{aligned} P(\omega ; k_p, k_i, \lambda) = & \sum_{i=0}^n [a_i(\omega)^{(\lambda+\alpha_i)} \{ \cos \left[(\lambda + \alpha_i) \frac{\pi}{2} \right] + j [\sin[(\lambda + \alpha_i) \frac{\pi}{2}]] \} + \dots \\ & \dots k_p b_i(\omega)^{(\lambda+\beta_i)} \{ \cos[(\lambda + \beta_i) \frac{\pi}{2}] + j [\sin[(\lambda + \beta_i) \frac{\pi}{2}]] \} + \dots \\ & \dots k_i b_i(\omega)^{\beta_i} \{ \cos \left[(\beta_i) \frac{\pi}{2} \right] + j [\sin[(\beta_i) \frac{\pi}{2}]] \}] = 0 \end{aligned} \quad (3.11)$$

(3.11)'deki denklemi (3.8)'deki gibi gerçel kısım ve sanal kısım olarak ifade etmek istenirse aşağıda yer alan denklemler bulunabilir.

$$\begin{aligned} R\{P(\omega, k_p, k_i, \lambda)\} = & \sum_{i=0}^n [a_i(\omega)^{(\lambda+\alpha_i)} \{ \cos \left[(\lambda + \alpha_i) \frac{\pi}{2} \right] \} + \dots \\ & \dots k_p b_i(\omega)^{(\lambda+\beta_i)} \{ \cos \left[(\lambda + \beta_i) \frac{\pi}{2} \right] \} + k_i b_i(\omega)^{\beta_i} \{ \cos \left[(\beta_i) \frac{\pi}{2} \right] \}] = 0 \end{aligned} \quad (3.12)$$

$$\begin{aligned} Im\{P(\omega, k_p, k_i, \lambda)\} = & \sum_{i=0}^n [a_i(\omega)^{(\lambda+\alpha_i)} \{ [\sin[(\lambda + \alpha_i) \frac{\pi}{2}]] \} + \dots \\ & \dots k_p b_i(\omega)^{(\lambda+\beta_i)} \{ [\sin[(\lambda + \beta_i) \frac{\pi}{2}]] \} + k_i b_i(\omega)^{\beta_i} \{ [\sin[(\beta_i) \frac{\pi}{2}]] \}] = 0 \end{aligned} \quad (3.13)$$

(3.12) ve (3.13)'teki denklemlerde yer alan terimleri aşağıdaki gibi ifade edilirse;

$$C_1(\omega) = \sum_{i=0}^n a_i(\omega)^{(\lambda+\alpha_i)} \cos \left[(\lambda + \alpha_i) \frac{\pi}{2} \right] \quad (3.14)$$

$$C_2(\omega) = \sum_{i=0}^n a_i(\omega)^{(\lambda+\alpha_i)} \sin \left[(\lambda + \alpha_i) \frac{\pi}{2} \right] \quad (3.15)$$

$$S_1(\omega) = \sum_{i=0}^n b_i(\omega)^{(\lambda+\beta_i)} \cos \left[(\lambda + \beta_i) \frac{\pi}{2} \right] \quad (3.16)$$

$$S_2(\omega) = \sum_{i=0}^n b_i(\omega)^{(\lambda+\beta_i)} \sin \left[(\lambda + \beta_i) \frac{\pi}{2} \right] \quad (3.17)$$

$$S_3(\omega) = \sum_{i=0}^n b_i(\omega)^{\beta_i} \cos \left[(\beta_i) \frac{\pi}{2} \right] \quad (3.18)$$

$$S_4(\omega) = \sum_{i=0}^n b_i(\omega)^{\beta_i} \sin \left[(\beta_i) \frac{\pi}{2} \right] \quad (3.19)$$

(3.14), (3.16) ve (3.18) denklemlerini kullanarak (3.12)'deki gerçel kısım denklemi

$$C_1(\omega) + k_p S_1(\omega) + k_i S_3(\omega) = 0 \quad (3.20)$$

şeklinde ifade edebilir. Aynı zamanda (3.15), (3.17) ve (3.19) denklemlerini kullanarak (3.13)'teki sanal kısım denklemi

$$C_2(\omega) + k_p S_2(\omega) + k_i S_4(\omega) = 0 \quad (3.21)$$

şeklinde ifade edilebilir.

Sonuç olarak, (3.20) ve (3.21)'deki 2 bilinmeyenli denklemler çözüldüğü zaman PI^λ tipi kontrolörün parametreleri aşağıdaki gibi bulunmuş olur.

$$k_p = \frac{C_2(\omega) S_3(\omega) - C_1(\omega) S_4(\omega)}{S_1(\omega) S_4(\omega) - S_2(\omega) S_3(\omega)} \quad (3.22)$$

$$k_i = \frac{C_1(\omega) S_2(\omega) - C_2(\omega) S_1(\omega)}{S_1(\omega) S_4(\omega) - S_2(\omega) S_3(\omega)} \quad (3.23)$$

k_p ve k_i , ω ve λ 'ya bağılı denklemlerdir. (3.22) ve (3.23) denklemlerinde ω 'yı 0'dan ∞ 'a kadar değiştirilerek elde edilen eğri (k_p , k_i) düzleminde karmaşık kök sınırını belirler.

iii) Sonsuz kök sınırı sadece (3.3)'teki kapalı çevrim transfer fonksiyonunun payı ve paydasının derecesi eşit olduğu zaman belirlenir. Bu durumda kararlılık bölgesi (3.7)'deki denklem 0'a eşitlenerek;

$$k_p = - \frac{a_n}{b_n} \quad (3.24)$$

şeklinde bulunur. (3.22) ve (3.23) denklemlerini de kullanarak, kararlılık bölgesi bulunur. Genellikle, kesirli dereceli kontrol sistemlerinde sistemin transfer fonksiyonunun paydasının derecesi payın derecesinden büyüktür ve bu durumlarda da geçerli sonsuz kök sınırı yoktur.

Karmaşık kök sınırı eğrisi (k_p , k_i) düzlemini kararlı ve kararsız bölgeler olmak üzere böler. Bu durumda her bölgeden rastgele test noktaları alınarak (3.7)'deki karakteristik denkleminde yerine konularak sistemin kararlı olup olmadığına bakılır. Eğer test noktasındaki karakteristik denklemin s-düzleminin sağ yarı tarafında kökü bulunmuyorsa, sistemin bu bölgede kararlı olduğu söylenir ve kararlılık sınıra kadar geçerlidir. Bu prosedür, λ 'nın (0,2) aralığındaki farklı değerleri için incelenip daha büyük kararlılık bölgesi bulunabilir.

Bazı durumlarda, birden fazla kararlılık ve kararsızlık bölgesi bulunabilir. Bu durumlar aşağıdaki gibi özetlenebilir.

i) $s^{\beta_0} = 1$ olması durumudur. KKS eğrisi GKS eğrisini bir ya da daha fazla noktada keser. Bu noktalarda, KKS eğrisinin frekans değeri (3.23)'teki denklemde k_i yerine 0 konularak bulunur.

ii) $\alpha_n = \beta_n$ olması durumudur. KKS eğrisi ve SKS doğrusu bir ya da daha fazla noktada kesişir.

iii) $s^{\beta_0} = 1$ ve $\alpha_n = \beta_n$ olması durumudur. Bu durum i) ve ii) durumlarının kombinasyonudur. KKS eğrisi, GKS eğrisi ve SKS doğrusuyla birçok noktada kesişir.

Bu kesişmelerin sonucu olarak çok fazla bölge elde edilebilir. Kararlılık bölgesi her bölgeden test noktaları alınarak kolaylıkla incelenebilir. Bazı durumlarda kararlılık bölgesi birbirinden kopuk farklı yerlerde olabilir.

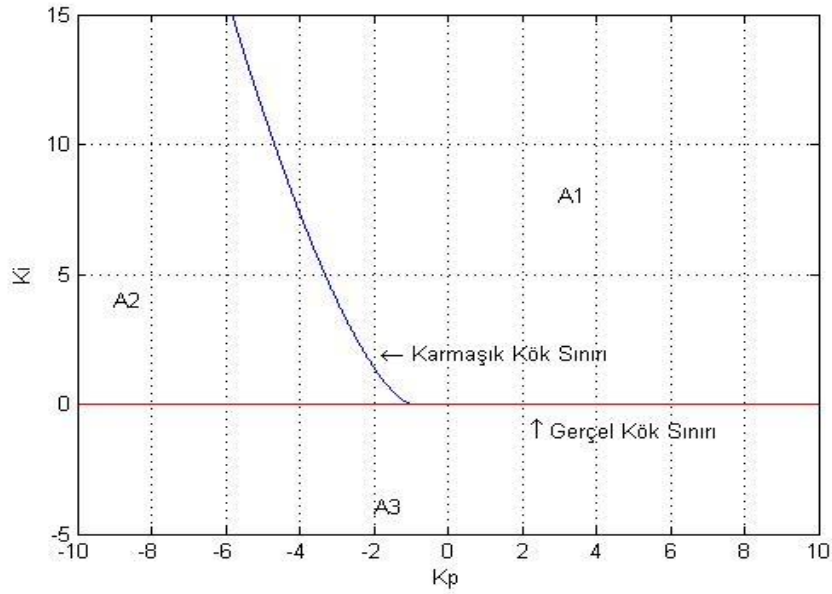
Örneğin aşağıda yer alan birinci dereceden sistem için tasarlanacak olan kesirli dereceli PI kontrolörün katsayılarını bulunsun.

$$G(s) = \frac{1}{s+1} \quad (3.25)$$

$\lambda=0.5$ olsun.

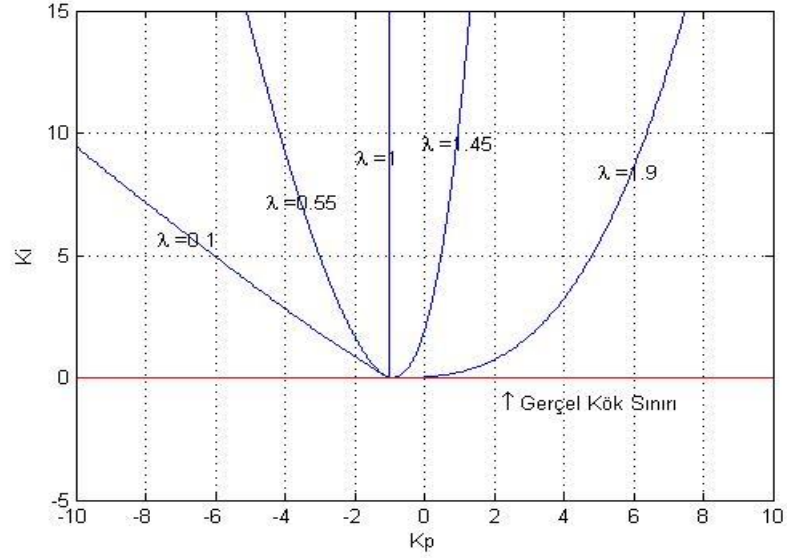
$$C(s) = k_p + k_i \frac{1}{s^\lambda} = k_p + k_i \frac{1}{s^{0.5}}$$

(3.22) ve (3.23) ‘teki denklemleri kullanarak, kapalı çevrim karakteristik denklemi kararlı yapan (k_p, k_i) noktalarını içeren kararlılık sınırları şekil 3.2 deki gibidir. Şekil3.2’ye bakıldığında sınırlar arasında kalan 3 bölge bulunmaktadır (A1, A2, A3). Bu bölgelerden rastgele test noktaları alınıp karakteristik denklem kökleri incelendiğinde A1 bölgesinin kararlı olduğu ve diğer bölgelerin kararsız olduğu bulunmaktadır. A1 bölgesindeki (k_p, k_i) noktaları kararlılık bölgesini oluşturmaktadır.



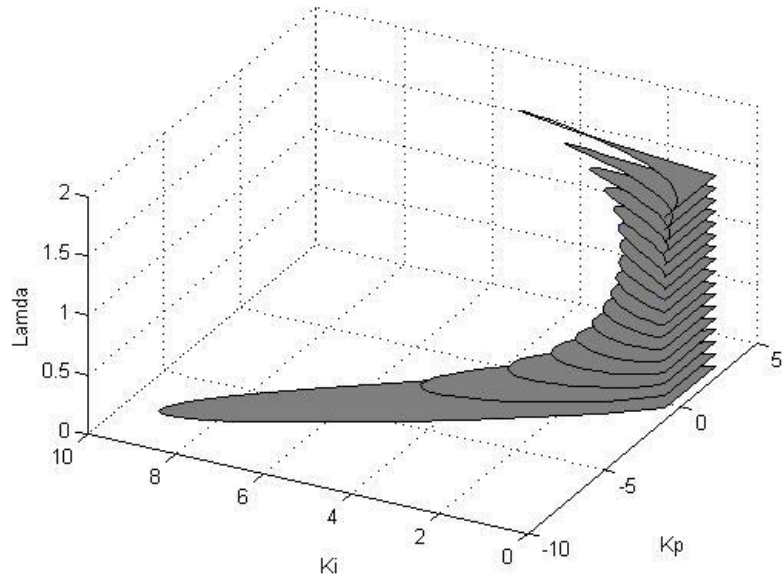
Şekil 3.2 Kesirli dereceli PI kontrolör için kararlılık bölgesi ($\lambda=0.5$)

Kesirli dereceli PI^λ kontrolör için $(0,2)$ arasındaki farklı λ değerleri için kararlılık sınırlarını belirlediğinde şekil 3.3’teki grafik elde edilmektedir. Şekilden de kolayca görüleceği gibi küçük λ değeri kendinden büyük λ değerlerinden daha büyük kararlılık bölgesine sahiptir.



Şekil.3.3 Farklı λ değerleri için kesirli dereceli PI kontrolör için kararlılık bölgeleri ($\lambda=0.1, 0.55, 1, 1.45, 1.9$)

Şekil 3.3’ te elde edilen kararlılık bölgesi sınırları integral derecesi λ ’nın farklı değerleri için oluşturulmuştur. Şekil 3.4’te eksenleri k_p , k_i ve λ olmak üzere kararlılık bölgesi sınırlarının 3 boyutlu gösterimi yer almaktadır. 3 boyutlu şekle de bakarak λ ’nın daha küçük değerleri için kararlılık bölgesinin daha büyük olduğu kolaylıkla görülebilmektedir.



Şekil.3.4 Kesirli dereceli PI kontrolör için kararlılık bölgesinin sınırlarının 3 boyutlu gösterimi

4.BİRİNCİ DERECEDEN ÖLÜ ZAMANLI SİSTEMLER İÇİN KESİRLİ DERECELİ Pİ KONTROLÖR KATSAYILARININ KARARLILIK BÖLGESİNİN BELİRLENMESİ

Bu bölümde de şekil (3.1)'de yer alan negatif geri beslemeli kontrol sistemi ele alınacaktır. Bir önceki bölümden farklı olarak burada incelenecek olan sistem ölü zamanlı sistemdir ve genel olarak

$$G_D(s) = \frac{N(s)}{D(s)} e^{-Ls} = \frac{b_n s^{\beta_n} + b_{n-1} s^{\beta_{n-1}} + \dots + b_1 s^{\beta_1} + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_1 s^{\alpha_1} + a_0 s^{\alpha_0}} e^{-Ls} = \frac{\sum_{i=0}^n b_i s^{\beta_i}}{\sum_{i=0}^n a_i s^{\alpha_i}} e^{-Ls} \quad (4.1)$$

şeklinde gösterilebilir.

Aynı zamanda kontrolörümüz (3.1)'de yer alan kesirli dereceli PI^λ tipindeki kontrolördür. Şekil 3.1'de yer alan negatif geri beslemeli sistemin transfer fonksiyonu ölü zamanlı sistemler için

$$Y_D(s) = \frac{C(s)G_D(s)}{1+C(s)G_D(s)} \quad (4.2)$$

şeklinde gösterilebilir. Ölü zaman içeren kapalı çevrim sistemin karakteristik denklemi (2.2)'nin paydasıdır.

$$\begin{aligned} P_{Dc}(s) &= 1 + C(s) G_D(s) \\ &= E(s) D(s) + U(s) N(s) e^{-Ls} \\ &= s^\lambda D(s) + (k_p s^\lambda + k_i) N(s) e^{-Ls} \\ &= \sum_{i=0}^n [a_i s^{(\lambda+\alpha_i)} + (k_p b_i s^{(\lambda+\beta_i)} + k_i b_i s^{\beta_i}) e^{-Ls}] \quad (4.3) \end{aligned}$$

şeklinde ifade edilebilir. (4.3)'teki denklemin kararlı olabilmesi için köklerinin sağ yarı düzlemde bulunmaması gerekmektedir. Denklem kesirli dereceli olduğu için bir önceki bölümde yararlanılan sınırlı giriş – sınırlı çıkış (SGSÇ) (Bounded input – bounded output) kararlılığını burada da kullanabiliriz. Fakat (4.3)'teki karakteristik denklemi e^{-Ls} şeklinde üstel fonksiyon içerdiği için kararlılık analizi yapılması oldukça güçtür. Bu yüzden üstel fonksiyonu tam sayı dereceli polinoma dönüştürülmesi kararlılığın incelenmesi için kolaylık sağlayacaktır. Bunun için farklı yaklaşımlar kullanarak daha basit çözümler elde edebilir.

4.1 Pade Yaklaşımı

Zaman gecikmesinin transfer fonksiyonu

$$H(s) = e^{-L s} \quad (4.4)$$

L: zaman gecikmesi

olmak üzere (4.4)'teki gibi gösterilebilir. Bu transfer fonksiyonu irrasyonel fonksiyondur (payında ve paydasında s değişkenli polinom bulunmamaktadır). Bazı durumlarda özellikle ölü zaman içeren kontrol sistemlerinde rasyonel transfer fonksiyonu şeklinde gösterilmektedir. Buna en yakın olarak (4.5)'teki en genel yaklaşım pade yaklaşımıdır.

$$e^{-L s} \approx \frac{1 - k_1 s + k_2 s^2 + \dots + k_n s^n}{1 + k_1 s + k_2 s^2 + \dots + k_n s^n} \quad (4.5)$$

n yaklaşımın derecesini ifade etmektedir ve derece için farklı değerler seçilebilir. k_i katsayıları da n'nin fonksiyonlarıdır. Örnek olarak çizelge 4.1'de n=1 ve n=2 değerleri için k_i değerleri yer almaktadır.

Çizelge 4.1 n=1 ve n=2 için pade katsayıları.

n=1	n=2
$k_1 = L/2$	$k_1 = L/2 \quad k_2 = L^2/12$
diğer $k_i=0$	diğer $k_i=0$

Örneğin; zaman gecikmesi 5 olan 2. dereceden pade yaklaşımını incelensin.

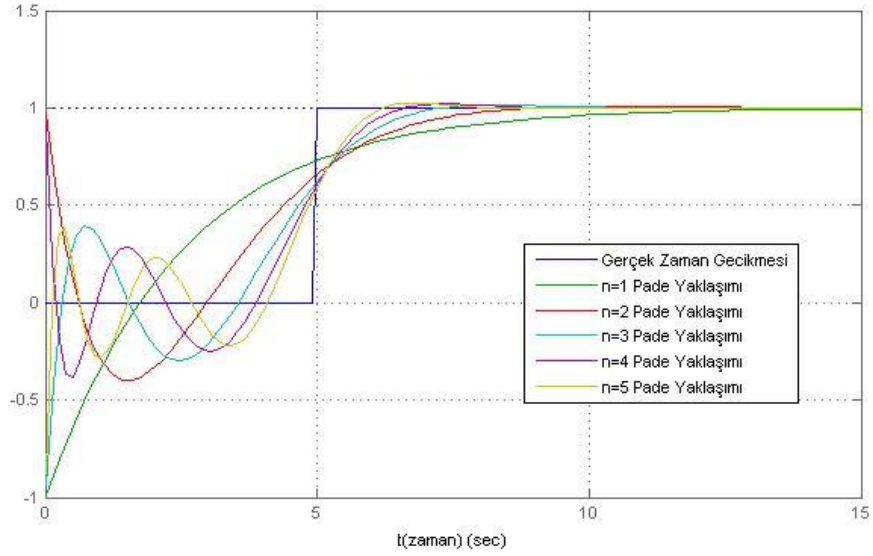
$$k_1 = 5/2 = 2.5$$

$$k_2 = 5^2/12 = 2.083$$

Pade yaklaşımının transfer fonksiyonunu

$$H_{pade}(s) = \frac{1 - 2.5 s + 2.083 s^2}{1 + 2.5 s + 2.083 s^2} \quad (4.6)$$

olarak bulunmaktadır.



Şekil 4.1 L=5 Zaman gecikmesinin farklı dereceli pade yaklaşımı

Şekil 4.1’de L=5 gerçek zaman gecikmesini ve bu zaman gecikmesinin farklı dereceli pade yaklaşımlarını gösterilmektedir. Burada görüldüğü gibi pade derecesi arttıkça gerçek zaman gecikmesine yaklaşmaktadır.

4.2 Taylor Serisi

En basit yaklaşımlardan biri olan Taylor seri açılımı, seri açılımı belirlenecek olan fonksiyonun terimlerinin tek bir noktadaki türev değerlerinden hesaplanan sonsuz toplamı şeklinde ifade edilebilir.

$$f(x) = f(a) + \frac{f'(a)}{1!} (x-a) + \frac{f''(a)}{2!} (x-a)^2 + \frac{f'''(a)}{3!} (x-a)^3 + \dots + \frac{f^n(a)}{n!} (x-a)^n \quad (4.7)$$

Taylor serisinin ilk 3 terimini aldığımızda aşağıdaki gibi sonuç elde ederiz.

$$f(s) = e^{-Ls}$$

$$f(s) = e^{-Ls}$$

$$f(0)=1$$

$$f'(s) = -Le^{-Ls}$$

$$f'(s) = -Le^{-Ls}$$

$$f'(0) = -L$$

$$f''(s) = L^2e^{-Ls}$$

$$f''(s) = L^2e^{-Ls}$$

$$f''(0) = L^2$$

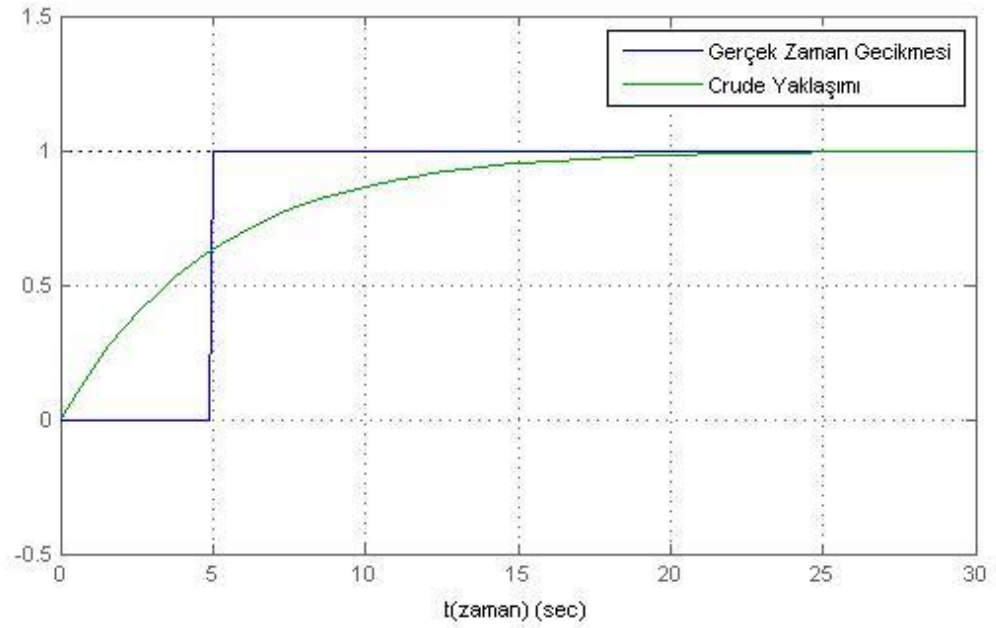
$$e^{-Ls} = 1 + \frac{-L}{1} s + \frac{-L^2}{2!} s^2$$

4.3 Crude Yaklaşımı

Zaman gecikmesi için farklı bir yaklaşım da Crude yaklaşımıdır [8] ve

$$e^{-Ls} = \frac{1}{1+Ls}$$

şeklinde gösterilebilir. Crude yaklaşımı ile gerçek zaman gecikmesini karşılaştırdığımızda elde edilen sonuç şekil 4.2’de gösterilmektedir.



Şekil 4.2 Gerçek zaman gecikmesi ve Crude Yaklaşımı (L=5)

Zaman gecikmesi için incelemiş olan yaklaşımlardan en uygunu pade yaklaşımıdır. Payının ve paydasının derecesinin birbirine eşit olması avantaj sağlarken aynı zamanda da istenilen derecenin seçilmesine olanak sağlamaktadır.

Zaman gecikmesi için seçilen pade yaklaşımı aşağıdaki gibi genelleştirebilir.

$$e^{-Ls} \approx \frac{1-k_1s+k_2s^2+\dots\pm k_ns^n}{1+k_1s+k_2s^2+\dots+k_ns^n} = \frac{\sum_{i=0}^n b_i s^{\beta_i}}{\sum_{i=0}^n a_i s^{\alpha_i}} = \frac{Np(s)}{Dp(s)} \quad (4.8)$$

(4.1)’deki ölü zamanlı sistemi (3.2) ve (4.8)’deki denklemleri kullanarak;

$$G_D(s) = \frac{N(s)}{D(s)} e^{-Ls} = \frac{N(s)}{D(s)} \frac{Np(s)}{Dp(s)} = \frac{\sum_i^n \sum_{j=0}^m b_i \cdot b_{pj} s^{(\beta_i + \beta_{pj})}}{\sum_i^n \sum_{j=0}^m a_i \cdot a_{pj} s^{(\alpha_i + \alpha_{pj})}} = \frac{\sum_{i=0}^n x_i s^{\mu_i}}{\sum_{i=0}^n y_i s^{\nu_i}} \quad (4.9)$$

şeklinde yazılabilir. Bu biçimde sade halde yazılması karmaşıklığın önüne geçecek ve elde etmiş olan sonuç hem sistemin değişkenleri hem de ölü zaman değişkenlerini tek formülde birleştirmiş olur. Buna ek olarak (4.3)'teki ölü zamanlı kapalı çevrim sistemin karakteristik denklemini aşağıdaki gibi yazabiliriz.

$$\begin{aligned}
P_D(s) &= 1 + C(s) G_D(s) \\
&= E(s) D_D(s) + U(s) N_D(s) \\
&= s^\lambda D_D(s) + (k_p s^\lambda + k_i) N_D(s) \\
&= \sum_{i=0}^n [y_i s^{(\lambda+\gamma_i)} + k_p x_i s^{(\lambda+\mu_i)} + k_i x_i s^{\mu_i}] \quad (4.10)
\end{aligned}$$

Kapalı çevrim sistemin karakteristik denkleminde üstel fonksiyon kalmadığı için (4.10)'da elde edilen denklemi kullanarak kolaylıkla kararlılık analizi yapılabilir.

Zaman gecikmesi içermeyen sistemler için birinci bölümde uygulanan kararlılık sınır kuralları ölü zaman içeren sistemlerin analizi için de sırasıyla aşağıdaki gibi uygulanabilir.

i) Gerçek Kök Sınırı (GKS): Birinci bölümde uygulanan sistemler için uygulanan kurallar zaman gecikmesi içeren sistemler için de geçerlidir. (4.10)'daki karakteristik denklemde s yerine 0 yazılır ve denklem 0'a eşitlenerek

$$k_i = 0 \quad (4.11)$$

sonucu bulunur.

ii) Karmaşık Kök Sınırı (KKS): Bu bölümde de zaman gecikmesi içeren sistemler için aynı kurallar geçerlidir. (4.10)'daki denklemde s yerine $j\omega$ yazılarak denklem 0'a eşitlenir.

$$P_D(j\omega; k_p, k_i, \lambda) = \sum_{i=0}^n [y_i (j\omega)^{(\lambda+\gamma_i)} + k_p x_i (j\omega)^{(\lambda+\mu_i)} + k_i x_i (j\omega)^{\mu_i}] = 0 \quad (4.12)$$

(4.12)'deki denklem gerçek ve karmaşık sayıya sahip olduğundan dolayı (3.8)'deki gibi ifade edilebilir.

(4.12)'deki denklem içine (4.10)'da yer alan ifade yazılırsa,

$$\begin{aligned}
P_D(\omega; k_p, k_i, \lambda) &= \sum_{i=0}^n [y_i (\omega)^{(\lambda+\gamma_i)} \{ \cos \left[(\lambda + \gamma_i) \frac{\pi}{2} \right] + j [\sin \left[(\lambda + \gamma_i) \frac{\pi}{2} \right] \} + \dots \\
&\dots k_p x_i (\omega)^{(\lambda+\mu_i)} \{ \cos \left[(\lambda + \mu_i) \frac{\pi}{2} \right] + j [\sin \left[(\lambda + \mu_i) \frac{\pi}{2} \right] \} + \dots \\
&\dots k_i x_i (\omega)^{\mu_i} \{ \cos \left[(\mu_i) \frac{\pi}{2} \right] + j [\sin \left[(\mu_i) \frac{\pi}{2} \right] \}] = 0 \quad (4.13)
\end{aligned}$$

(3.8)'deki gerçel ve sanal kısımları olmak üzere (4.13)'teki denklemi de ayrı ayrı ifade edilebilir.

$$\begin{aligned} \operatorname{Re}\{P_D(\omega, k_p, k_i, \lambda)\} &= \sum_{i=0}^n [y_i(\omega)^{(\lambda+\gamma_i)} \{\cos[(\lambda + \gamma_i) \frac{\pi}{2}]\} + \dots \\ &\dots k_p x_i(\omega)^{(\lambda+\mu_i)} \{\cos[(\lambda + \mu_i) \frac{\pi}{2}]\} + k_i x_i(\omega)^{\mu_i} \{\cos[(\mu_i) \frac{\pi}{2}]\} = 0 \end{aligned} \quad (4.14)$$

$$\begin{aligned} \operatorname{Im}\{P_D(\omega, k_p, k_i, \lambda)\} &= \sum_{i=0}^n [y_i(\omega)^{(\lambda+\gamma_i)} \{\sin[(\lambda + \gamma_i) \frac{\pi}{2}]\} + \dots \\ &\dots k_p x_i(\omega)^{(\lambda+\mu_i)} \{\sin[(\lambda + \mu_i) \frac{\pi}{2}]\} + k_i x_i(\omega)^{\mu_i} \{\sin[(\mu_i) \frac{\pi}{2}]\} = 0 \end{aligned} \quad (4.15)$$

(4.14) ve (4.15)'teki denklemlerde yer alan terimleri aşağıdaki gibi ifade edilirse;

$$C_{1D}(\omega) = \sum_{i=0}^n y_i(\omega)^{(\lambda+\gamma_i)} \cos[(\lambda + \gamma_i) \frac{\pi}{2}] \quad (4.16)$$

$$C_{2D}(\omega) = \sum_{i=0}^n y_i(\omega)^{(\lambda+\gamma_i)} \sin[(\lambda + \gamma_i) \frac{\pi}{2}] \quad (4.17)$$

$$S_{1D}(\omega) = \sum_{i=0}^n x_i(\omega)^{(\lambda+\mu_i)} \cos[(\lambda + \mu_i) \frac{\pi}{2}] \quad (4.18)$$

$$S_{2D}(\omega) = \sum_{i=0}^n x_i(\omega)^{(\lambda+\mu_i)} \sin[(\lambda + \mu_i) \frac{\pi}{2}] \quad (4.19)$$

$$S_{3D}(\omega) = \sum_{i=0}^n x_i(\omega)^{\mu_i} \cos[(\mu_i) \frac{\pi}{2}] \quad (4.20)$$

$$S_{4D}(\omega) = \sum_{i=0}^n x_i(\omega)^{\mu_i} \sin[(\mu_i) \frac{\pi}{2}] \quad (4.21)$$

(4.16) , (4.18) ve (4.20) denklemlerini kullanarak (4.14)'deki gerçel kısım denklemi

$$C_{1D}(\omega) + k_p S_{1D}(\omega) + k_i S_{3D}(\omega) = 0 \quad (4.22)$$

şeklinde ifade edilebilir. Aynı zamanda (4.17) , (4.19) ve (4.21) denklemlerini kullanarak (4.15)'teki sanal kısım denklemi

$$C_{2D}(\omega) + k_p S_{2D}(\omega) + k_i S_{4D}(\omega) = 0 \quad (4.23)$$

şeklinde ifade edilebilir.

Sonuç olarak, (4.22) ve (4.23)'deki 2 bilinmeyenli denklemler çözüldüğünde ölü zamanlı sistemler için PI^λ tipi kontrolörün parametrelerini aşağıdaki gibi bulunmuş olur.

$$k_p = \frac{C_{2D}(\omega) S_{3D}(\omega) - C_{1D}(\omega) S_{4D}(\omega)}{S_{1D}(\omega) S_{4D}(\omega) - S_{2D}(\omega) S_{3D}(\omega)} \quad (4.24)$$

$$k_i = \frac{C_{1D}(\omega) S_{2D}(\omega) - C_{2D}(\omega) S_{1D}(\omega)}{S_{1D}(\omega) S_{4D}(\omega) - S_{2D}(\omega) S_{3D}(\omega)} \quad (4.25)$$

k_p ve k_i ω ve λ 'ya bağlı denklemlerdir. (4.24) ve (4.25) denklemlerinde ω 'yı 0'dan ∞ 'a kadar değiştirilerek elde edilen eğri (k_p , k_i) düzleminde, ölü zamanlı sistemler için tasarlanan PI^λ kontrolörün katsayılarının karmaşık kök sınırını belirler.

iii) Ölü zaman içeren sistemlerde de sonsuz kök sınırı sadece (4.2)'teki kapalı çevrim transfer fonksiyonunun payı ve paydasının derecesi birbirine eşit olduğu zaman belirlenir. Bu durumda kararlılık bölgesi (4.12)'deki denklem 0'a eşitlenerek;

$$k_p = - \frac{a_n}{b_n} \quad (4.24)$$

şeklinde bulunur. Ölü zaman içeren kapalı çevrim sistemlerin kararlılık bölgesi (4.24) ve (4.25) denklemlerini de kullanarak bulunur. Birinci bölümde bahsedildiği gibi, genellikle, kesirli dereceli kontrol sistemlerinde sistemin transfer fonksiyonunun paydasının derecesi payın derecesinden büyüktür ve bu durumlarda da geçerli sonsuz kök sınırı yoktur.

Gerçel kök sınırı ve karmaşık kök sınırı eğrileri (k_p , k_i) düzlemini kararlılık ve kararsız bölgeler olmak üzere böler. Hangi bölgelerin kararlı olduğunu belirlemek için her bölgeden test noktası olarak (4.12)'de elde edilen ve ölü zaman içeren karakteristik denklemde yerine konularak sistemin kararlı olup olmadığı incelenir. (4.12)'deki karakteristik denklemin s-düzleminin sağ yarı tarafında kökü bulunmuyorsa ölü zaman içeren kapalı çevrim sisteminin kararlı olduğu söylenir.

Örnek olarak, aşağıda yer alan birinci dereceden ölü zaman içeren ($L=1$) sistem için tasarlanacak olan kesirli dereceli PI kontrolörün katsayılarının kararlılık bölgesini bulunsun.

$$G_D(s) = \frac{1}{s+1} e^{-s} \quad (4.25)$$

Birinci dereceden sistem, ölü zaman içerdiği için üstel fonksiyon nedeniyle kararlılık analizi yapmak oldukça güçtür. Bu yüzden ölü zaman için 2. derece pade yaklaşımını kullanılsın. Ölü zamanı 1 alındığında, pade yaklaşımı

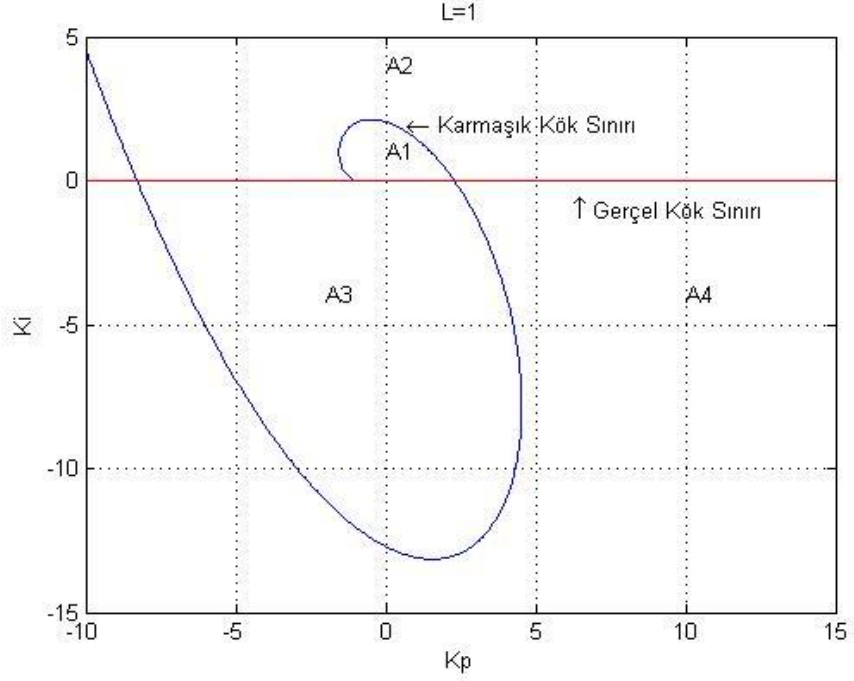
$$e^{-s} = \frac{s^2 - 1.2s + 0.48}{s^2 + 1.2s + 0.48} \quad (4.26)$$

şeklinde elde edilir.

Ayrıca kontrolör de aşağıdaki gibi olsun.

$$C(s) = k_p + k_i \frac{1}{s^\lambda} = k_p + k_i \frac{1}{s^{0.5}}$$

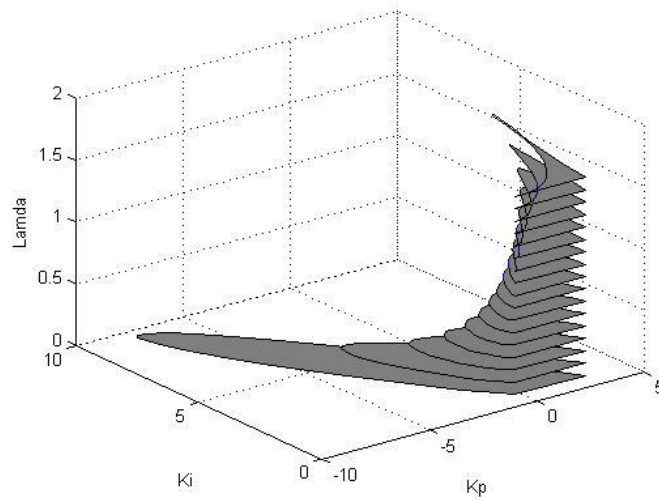
(4.24) ve (4.25) 'teki denklemleri kullanarak, ölü zaman içeren kapalı çevrim karakteristik denklemi kararlı yapan (k_p , k_i) noktalarını içeren kararlılık sınırları şekil 4.3 teki gibidir.



Şekil 4.3 Birinci dereceden ölü zamanlı (2. dereceden pade yaklaşımı) sistemin kararlılık bölgesi ($\lambda=0.5$)

Şekil 4.3 incelediğinde 4 farklı (A1, A2, A3, A4) bölge oluştuğu görülmektedir. Bu bölgelerden farklı test noktaları alınarak kararlılığı incelendiğinde sadece A1 bölgesinin kararlı olduğu görülmektedir.

Ayrıca, şekil 4.4'te eksenleri k_p , k_i ve λ olmak üzere kararlılık bölgesi sınırlarının 3 boyutlu gösterimi yer almaktadır. Ölü zaman içermeyen sistemlerde olduğu gibi ölü zaman içeren sistemlerde de λ 'nın değeri arttıkça kararlılık bölgesinin küçüldüğü gözlenmektedir.



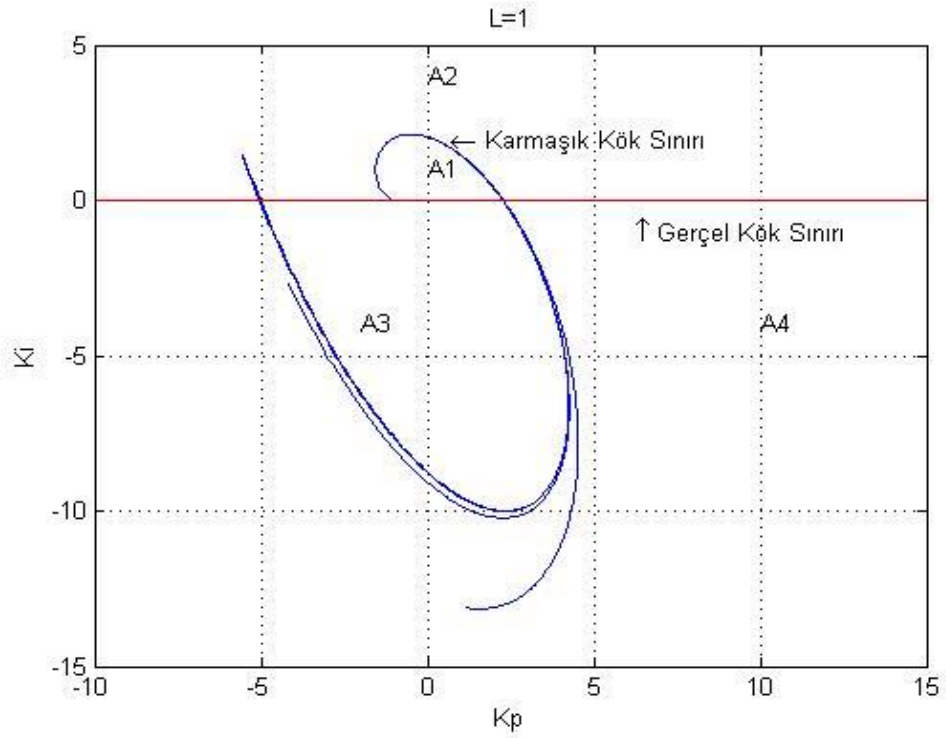
Şekil 4.4 Birinci dereceden ölü zamanlı sistem için tasarlanacak kesirli dereceli PI kontrolör için kararlılık bölgesinin sınırlarının 3 boyutlu gösterimi

Ölü zaman için farklı pade dereceleri ele alınırsa pade derecelerinin katsayıları çizelge 4.2'deki gibi bulunmaktadır.

Çizelge 4.2 Farklı dereceden Pade yaklaşım katsayıları ($L=1$)

Pade Derecesi	k1	k2	k3	k4	k5	k6
n=1	0.5	0	0	0	0	0
n=2	0.5	0,083333	0	0	0	0
n=3	0.5	0,1	0,008333	0	0	0
n=4	0.5	0,107143	0,011905	0,000595	0	0
n=5	0.5	0,111111	0,013889	0,000992	3,30688E-05	0

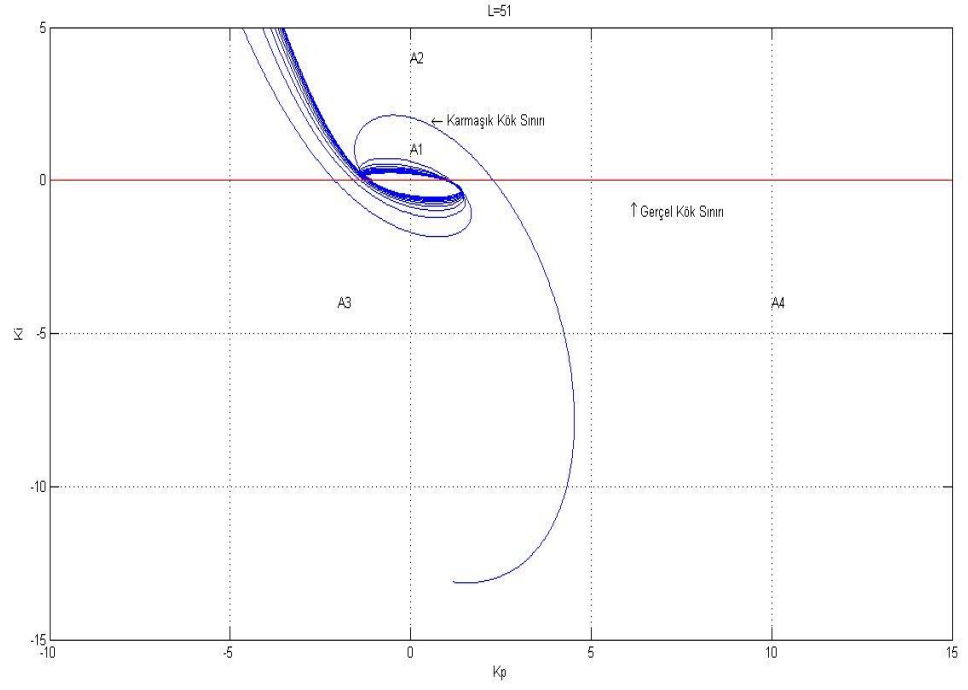
Ölü zaman için farklı Pade yaklaşımları ele alındığında, ölü zaman içeren kapalı çevrim karakteristik denklemi kararlı yapan (k_p , k_i) noktalarını içeren kararlılık sınırları şekil 4.5 teki gibi elde edilmektedir.



Şekil 4.5 Farklı Pade Dereceleri için kontrolör katsayılarının kararlılık bölgesi ($L=1$ $\lambda=0.5$)

Şekil 4.5 incelendiğinde A1 kararlılık bölgesinin sınırının değişmediği gözlenmektedir. Bu yüzden, kararlılık bölgesi araştırılırken Pade yaklaşımının derecesi istenildiği gibi seçilebilir. Kapalı çevrim sistemi daha karmaşık hale getirmemek için Pade yaklaşımının derecesini küçük seçmek faydalı olabilir.

Ayrıca farklı ölü zaman değerleri için kararlılık bölgesi incelendiğinde şekil 4.6'daki grafik elde edilmektedir.



Şekil 4.6 Farklı ölü zaman değerleri için kararlılık bölgesindeki değişim ($\lambda=0.5$)

Şekil 4.6'daki grafik incelendiğinde ölü zaman değerinin arttığında kararlılık bölgesinin daraldığı rahatlıkla görülebilmektedir.

5. KESİRLİ DERECELİ SİSTEMLER İÇİN MATLAB FONKSİYONLARININ TANITIMI

Kesirli dereceli sistemler genellikle kesirli dereceli denklemler tarafından modellenir ve (2.1)'deki denklemden yararlanılarak kesirli derece operatörünün nümerik olarak hesaplaması uygulanır.

Kesirli dereceli transfer fonksiyon (KDTF) modelinin gerekli parametreleri (3.2)'deki denklemden kolaylıkla görülebilir. Bu parametreler (5.1)'de gösterildiği gibi KDTF'nun katsayı ve derece vektörleridir.

$$\begin{aligned} n_b &= [\beta_n, \beta_{n-1}, \dots, \beta_0], \\ n_a &= [\alpha_n, \alpha_{n-1}, \dots, \alpha_0] \\ b &= [b_n, b_{n-1}, \dots, b_0], \\ a &= [a_n, a_{n-1}, \dots, a_0]. \end{aligned} \quad (5.1)$$

MATLAB'da kesirli dereceli sistemleri analiz yapabilmek için Y. Q. Chen [6] tarafından yazılan makalede anlatılan kesirli dereceli sistemlerin programlanması ve analizini içeren MATLAB kesirli derece fonksiyonları tanıtılacaktır. İlk olarak MATLAB'da KDTF sınıfı oluşturulur ve bu yapının üzerine fonksiyonlar yazılarak kararlılık testi, zaman ve frekans bölgesindeki analizler yapılabilir. Bu yapılar ve kullanımı sırasıyla bu bölümde gösterilecektir.

5.1 KDTF için Nesne Tabanlı Programlama

Bu bölümde MATLAB nesne tabanlı programlama teknikleri gösterilecek olup KDTF sınıfı yapısının örnekleri gösterilerek gerekli uygulamalar yapılacaktır. Ayrıca sınıf yapısına bağlı olarak KDTF blok yapılarının bağlantıları gösterilecek olup kesirli dereceli sistemler için geri beslemeli kontrol sistemlerinin yapısı modellenecektir.

5.1.1 KDTF sınıf oluşturma

MATLAB'da sınıf tanımlamak için, ilk olarak oluşturulacak dosya tanımlanmalıdır. Dosya tanımlandığında isminin başında “@” işaretinin olması gerekmektedir. Örneğin KDTF'ları için “@fotf” şeklinde dosya oluşturulabilir. Daha sonra bu

dosyanın içine sınıf yapısında oluşturulacak olan “.m” uzantılı MATLAB fonksiyonları eklenir. Bu dosyanın içine eklenmesi gerekli olan ilk fonksiyonlar “fotf.m” ve “display.m” dir. “fotf.m” KDTF oluşturmak için gerekli olan fonksiyondur ve “display.m” de oluşturulan KDTF’larını göstermek için gerekli olan fonksiyondur. Bu iki MATLAB fonksiyonu sırasıyla EK A’ da gösterilmiştir.

Örneğin 5.2’de yer alan KDTF’u tanımlanmak istenirse;

$$G(s) = \frac{-2s^{0.63} + 4}{2s^{3.501} + 3.8s^{2.42} + 2.6s^{1.798} + 2.5s^{1.31} + 1.5} \quad (5.2)$$

MATLAB’da aşağıda yer alan komutlar yazılarak tanımlanır.

$b = [-2, 4]; na = [3.501, 2.42, 1.798, 1.31, 0];$

$nb = [0.63, 0]; a = [2 \ 3.8 \ 2.6 \ 2.5 \ 1.5];$

$G = \text{fotf}(a, na, b, nb)$

ve KDTF nesnesi aşağıdaki gibi gösterilir.

$-2s^{0.63} + 4$

 $2s^{3.501} + 3.8s^{2.42} + 2.6s^{1.798} + 2.5s^{1.31} + 1.5$

5.1.2 Oluşturulan fonksiyonları programlama

Matematiksel gösterimde, iki sistem $G_1(s)$ ve $G_2(s)$ olmak üzere, birbirine seri biçimde bağlanmışsa, oluşacak olan yapı $G_1(s) G_2(s)$ şeklinde hesaplanır. Eğer paralel yapıda birbirine bağlanırsa oluşacak yeni yapı $G_1(s) + G_2(s)$ şeklinde hesaplanır.

KDTF nesneleri için, çarpma ve toplama işlemleri `mtimes.m` ve `plus.m` olmak üzere EK B’de sırasıyla gösterilmiş olup tanımlanmış olan KDTF’ları için uygulanabilir. Ayrıca bu fonksiyonlar da sınıf yapısında olduğundan dolayı “@fotf” dosyasının içinde yer almalıdır.

Ayrıca EK B’deki “unique.m” fonksiyonu KDTF’ları bir araya toplamak ve daha basit hale getirmek için kullanılabilir.

Yukarıda bahsedilmiş KDTF’ları için gerekli olan toplama (+) ve çarpma (*) fonksiyonları, yazılan programda uygun olan yerde çağrıldıklarında görevlerini yerine getirirler. Bunların haricinde işlemleri kolaylaştırmak için gerekli olan

geribesleme (“feedback()”), çıkarma (“minus.m”), fonksiyonun toplama işlemine göre tersi (“uminus.m”), fonksiyonun çarpma işlemine göre tersi (“inv.m”) gibi hazır fonksiyonlar bulunmaktadır ve EK C’de gösterilmektedir.

Ayrıca tam sayı dereceli sistemi, kesir dereceli sistem biçiminde ifade etmek isteniyorsa “@tf” klasörünün içinde EK C’deki “fotf.m” fonksiyonunu ekleyerek kolaylıkla dönüşüm yapılabilir.

Yukarıda bahsedilen fonksiyonlar, KDTF bloklarının seri, paralel ve geri besleme bağlantılarını MATLAB kontrol sistemi araç kutusunda (Control System Toolbox) kolaylıkla kullanılabilir.

5.1.3 KDTF modelleme gösterimi

Negatif geri beslemeli sistemlerde yer alan sistem ve kontrolörün aşağıdaki gibi olduğunu kabul edilsin.

$$G(s) = \frac{0.8 s^{1.2} + 2}{1.1 s^{1.8} + 0.8 s^{1.3} + 1.9 s^{0.5} + 0.4}$$

$$G_c(s) = \frac{1.2 s^{0.72} + 1.5 s^{0.33}}{3 s^{0.8}}$$

İlk olarak bu iki modeli sisteme tanımlamak gerekir ardından kapalı çevrim sistemin transfer fonksiyonu kolaylıkla bulunabilir.

```
G=fotf([1.1,0.8 1.9 0.4],...
[1.8 1.3 0.5 0],[0.8 2],[1.2 0]);
Gc=fotf([3],[0.8], [1.2 1.5],[0.72 0.33]);
H=fotf(1,0,1,0); GG=feedback(G*Gc,H)
```

Kapalı çevrim sistem transfer fonksiyonunu

$$G(s) = 0.96s^{1.92} + 1.2s^{1.53} + 2.4s^{0.72} + 3s^{0.33}$$

$$3.3s^{2.6} + 2.4s^{2.1} + 0.96s^{1.92} + 1.2s^{1.53} + 5.7s^{1.3}$$

$$+ 1.2s^{0.8} + 2.4s^{0.72} + 3s^{0.33}$$

şeklinde elde edilebilir. Burada görüldüğü gibi sistem ve kontrolör ne kadar karışık olursa olsun sistem kolaylıkla modellenabilir.

5.2 KDTF Nesnelerinin Analizi

Bu bölümde KDTF'in üç önemli analiz metodu ele alınacak ve MATLAB uygulamaları ile örneklendirilecektir.

5.2.1) KDTF'un kararlılık analizi

Önceki bölümlerde bahsedildiği gibi KÇTF'un kararlı olabilmesi için karakteristik denkleminin s düzleminin sağ yarı tarafında hiçbir kökünün bulunmaması gerekmektedir. Sıradaki MATLAB fonksiyonu da, verilen KDTF'nun kararlılığını test etmek için "isstable.m" fonksiyonu oluşturulmuştur. Bu fonksiyon EK D'de yer almaktadır.

Fonksiyon çalıştırıldıktan sonra kararlılık hakkında bilgi veren K parametresini oluşturmaktadır. Burada test edilen sistem kararlı ise 1 değil ise 0 yanıtını vermektedir. Burada K parametresinin değerine bakarak sistemin kararlı olup olmadığını kolayca anlaşılabilir.

Örneğin (5.2)'deki sistemi ele alınsın;

$$G(s) = \frac{-2s^{0.63} + 4}{2s^{3.501} + 3.8s^{2.42} + 2.6s^{1.798} + 2.5s^{1.31} + 1.5}$$

MATLAB'da aşağıda yer alan komutlar yazılarak sistemin kararlı olup olmadığı kolayca bulunabilir.

```
>> b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];  
nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];  
[K,q,err,apol]=isstable(G)
```

"isstable.m" fonksiyonu kullanılarak sistemin kararlı olup olmadığını kolaylıkla test edilmiş olur. K değerine bakıldığında değer 1 olduğu ve sistemin kararlı olduğunu görülebilmektedir.

5.2.2 KDTF'un zaman bölgesindeki analizi

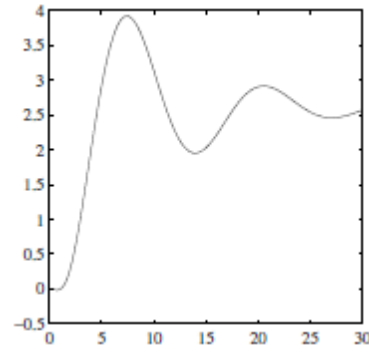
Kesirli dereceli sistemlerin zaman bölgesindeki yanıtı farklı metotlarla hesaplanabilir. Örneğin birim basamak ve impuls yanıtları Mittag Leffler fonksiyonları [4] kullanılarak bulunabilir. Fakat bu çözüm metodu zaman alıcı ve yavaştır.

Kesir dereceli sistemlerin zaman bölgesindeki yanıtı bulmak için Zhao ve Xue'nin [7]'deki çözüm metodunu kullanarak istenilen sonuç bulunabilir. Bu metot kullanılırken (2.1)'deki denkleme dayanmaktadır. Sistemin basamak yanıtını bulurken EK E'deki "step.m" ve "lsim.m" fonksiyonları kullanılmaktadır. Basamak yanıtını bulurken adım aralığı ne kadar küçük seçilirse sistemin yanıtı o kadar iyi olmaktadır.

Örneğin kararlı $G(s)$ sistemi ele alınsın.

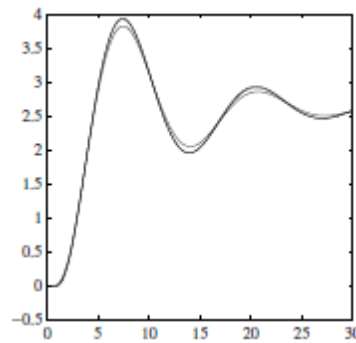
$$G(s) = \frac{-2s^{0.63} + 4}{2s^{3.501} + 3.8s^{2.42} + 2.6s^{1.798} + 2.5s^{1.31} + 1.5}$$

Aşağıda yer alan kodlar yazılarak sistemin basamak yanıtı bulunabilir. Bulunan sonuç şekil 5.1'de gösterilmiştir.



Şekil 5.1 Basamak Yanıtı (zaman aralığı 0.01)

Ayrıca basamak yanıtı küçüldüğü zaman elde edilen sonuç şekil 5.2'de gösterilmiştir.



Şekil 5.2 Basamak Yanıtı (zaman aralığı 0.01 ve 0.001)

5.2.3 KDTF'un frekans bölgesindeki analizi

Kesirli dereceli sistemlerin frekans bölgesindeki yanıtı, s yerine $j\omega$ yazılarak bulunur. Buna dayanarak, EK F'deki "bode", "Nyquist" VE "Nichols" fonksiyonları kullanılarak frekans bölgesindeki analiz kolayca yapılabilir.

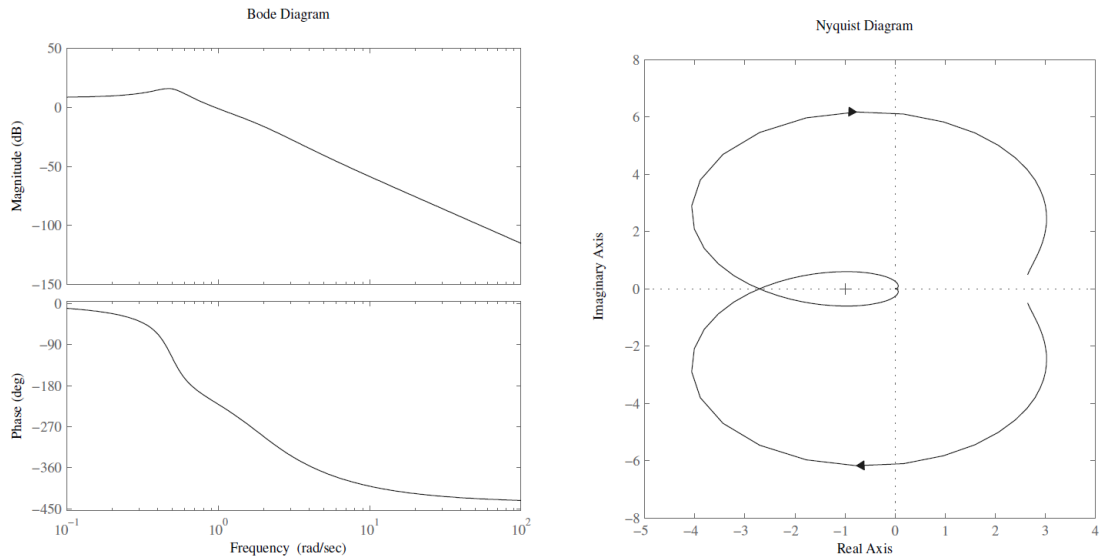
Örneğin (5.2)'deki sistem ele alındığında aşağıda yer alan komutlar yazılarak frekans yanıtları elde edilebilir.

```
b=[-2,4]; na=[3.501,2.42,1.798,1.31,0];
```

```
nb=[0.63,0]; a=[2 3.8 2.6 2.5 1.5];
```

```
G=fotf(a,na,b,nb); w=logspace(-1,2);
```

```
bode(G,w), figure; nyquist(G,w)
```



Şekil 5.3 Frekans Bölgesi Grafikleri (Bode Diagram Nyquist Diagram)

6. KESİRLİ DERECELİ PI KONTROLÖRÜN KATSAYILARININ BELİRLENMESİ

Bu bölüme kadar birinci dereceden ölü zamanlı sistemler için tasarlanacak olan kesirli dereceli PI kontrolörünün katsayılarının kararlılık bölgesi bulundu. Fakat bu bölgede yer alan parametrelerin hangi değeri için sistemin en iyi performansı sağlayacağı bilinmemektedir. Bu yüzden kararlılık bölgesindeki optimum noktayı belirleyebilmek için Büyük Patlama- Büyük Çöküş (Big Bang- Big Crunch) [9] optimizasyon metodu kullanılacaktır.

6.1 Büyük Patlama- Büyük Çöküş (Big Bang- Big Crunch) Optimizasyon Metodu

Optimizasyon problemleri, amaç fonksiyonunu minimize eden parametreleri bulmaya çalışır ve bunun için verilen bölgede mümkün olan en doğru kararı vermeye çalışır. Bu çalışmada da Büyük Patlama- Büyük Çöküş (Big Bang- Big Crunch) optimizasyon metodundan yararlanılacaktır. Bu metod, Genetik Algoritma (GA) metoduna çok benzemektedir. GA’da olduğu gibi ilk olarak verilen sınırlar arasında rastgele bir popülasyon oluşturulur. Bu ilk popülasyonun oluşturulması büyük patlama (big bang) olarak adlandırılır. Bu kısımda aday olan çözümler, arama alanında muntazam bir şekilde dağılırlar. Bu kısmı, büyük çöküş (big crunch) olarak adlandırılan bölüm izlemektedir. Büyük çöküş kısmı yakınsama operatörüdür ve birçok girişe sahip olup bir tane sonuç bulur ve bu sonuç “kütle merkezi” olarak adlandırılır. Bu sonuç uygun olan optimum değerdir. Kütle merkezi x^c olarak gösterilir ve aşağıdaki formül ile hesaplanır.

$$x^c = \frac{\sum_{i=1}^N \frac{1}{f^i} x^i}{\sum_{i=1}^N \frac{1}{f^i}} \quad (6.1)$$

N: popülasyon boyutu

x^i : arama alanındaki seçilen noktalardan biri

f^i : x^i noktasında uygunluk fonksiyonunun değeri

Büyük çöküşün ardından bir sonraki denemede yeni üyelerin oluşturulması gerekmektedir. Yeni üyeler, kütle merkezinin çevresinde fonksiyon değerini azaltacak şekilde rastgele seçilir. Bu seçim rastgele sayının kütle merkezine eklenmesi ya da çıkartılması ile gerçekleşir.

$$x^{\text{yeni}} = x^c + \ell r / k \quad (6.2)$$

ℓ : parametrenin üst limiti

r : normal rastgele sayı

k : yineleme adımı

Yeni üyelerin yer aldığı popülasyonda aday çözümlerin uygunluk fonksiyon değerleri hesaplanır ve en iyi çözüm kütle merkezi olur. Bu metot optimum değeri bulana kadar devam eder.

Büyük patlama- büyük çöküş optimizasyon metodu adım adım aşağıdaki gibi özetlenebilir.

Adım 1 : Sınırların içerisinde yer alan N tane adayın yer aldığı ilk popülasyon oluşturulur.

Adım 2 : Popülasyonda yer alan aday çözümlerin uygunluk fonksiyon değeri hesaplanır.

Adım 3 : Kütle merkezi bulunur (Kütle merkezi olarak uygunluk fonksiyonunu minimum yapan nokta seçilir).

Adım 4 : Kütle merkezinin çevresinde fonksiyon değerini azaltacak şekilde rastgele yeni popülasyon üyeleri seçilir. Bu seçim rastgele sayının kütle merkezine eklenmesi ya da çıkartılması ile gerçekleşir.

Adım 5 : Optimum değer bulunana kadar Adım 2'ye geçilir ve tekrarlanır.

Örneğin birinci dereceden ölü zamanlı aşağıdaki sistem için kesirli dereceli PI^λ tipi kontrolörün katsayıları ve integral derecesi, büyük patlama - büyük çöküş optimizasyon metodu ile bulunsun.

$$G_D(s) = \frac{1}{s+1} e^{-s}$$

$$C(s) = k_p + k_i \frac{1}{s^\lambda}$$

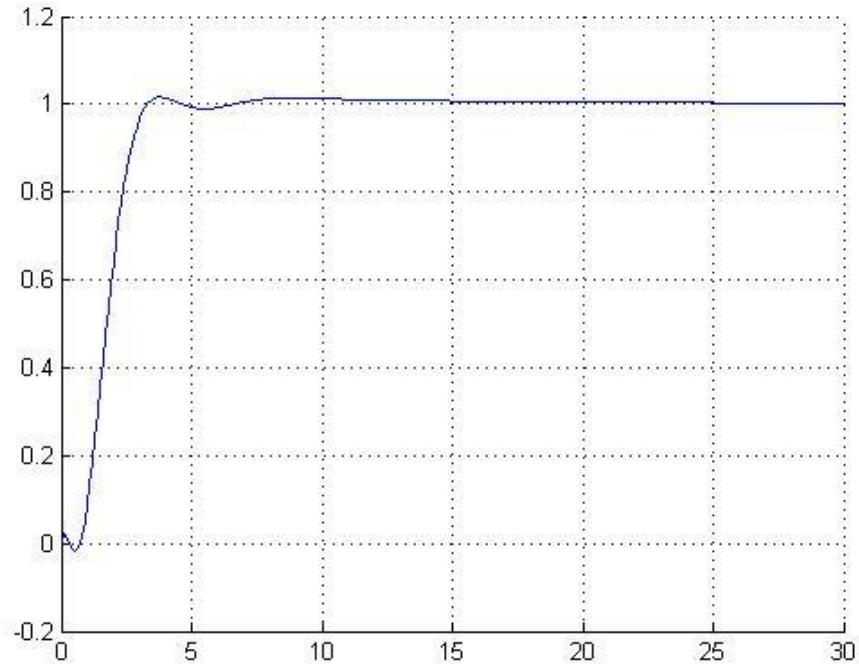
PI^λ tipi kontrolörün katsayılarını ve integral derecesini bulunurken dikkat edilecek olan kriterler kapalı çevrim sistemin yerleşme zamanı ve aşım değeridir. Bu değerleri minimum yapan kontrolör katsayıları ve integral derecesi, elde edilen optimum değer olacaktır.

İlk olarak büyük patlama - büyük çöküş optimizasyon metodu, sadece yerleşme zamanını minimum yapan senaryo üzerine kurulduğunda aşağıda yer alan parametre değerleri bulunmaktadır.

$$k_p = 0,7680$$

$$k_i = 0,5265$$

$$\lambda = 1,0450$$



Şekil 6.1 Yerleşme zamanı dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı

Elde edilen parametreler ile kapalı çevrim sistemin şekil 6.1’deki basamak yanıtına bakıldığında yerleşme zamanının 3 sn olduğu ve aşım değerinin 1,62 olduğu gözlenmektedir.

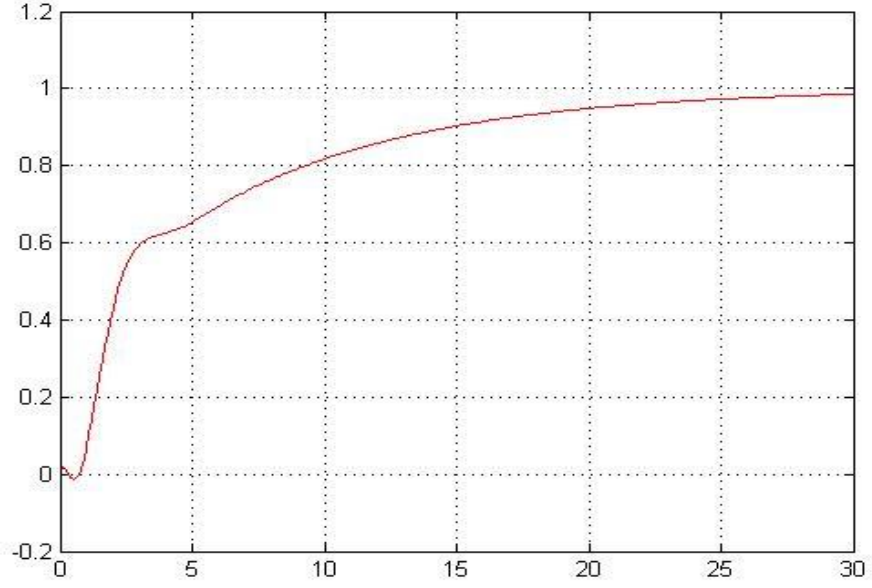
İkinci alternatif olarak, büyük patlama- büyük çöküş optimizasyon metodu, sadece aşım değerini minimum yapan senaryo üzerine kurulabilir. Bunu düşünerek, BPBÇ optimizasyon metodu uygulandığında kontrolör parametreleri aşağıdaki gibi bulunmaktadır.

$$k_p = 0,6265$$

$$k_i = 0,1777$$

$$\lambda = 1,0000$$

Elde edilen parametrelere göre sistemin basamak yanıtı şekil 6.2’deki gibi elde edilmektedir.



Şekil 6.2 Aşım değeri dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı

Şekil 6.2'deki basamak yanıtı incelendiğinde, kapalı çevrim sistemin aşım yapmadığı gözlemlenmektedir. Fakat sistem aşım yapmamasına rağmen yerleşme zamanı ilk senaryoda elde edilen yerleşme zamanından çok büyüktür. Bu yüzden, bulunan sonuç optimum sonuç değildir.

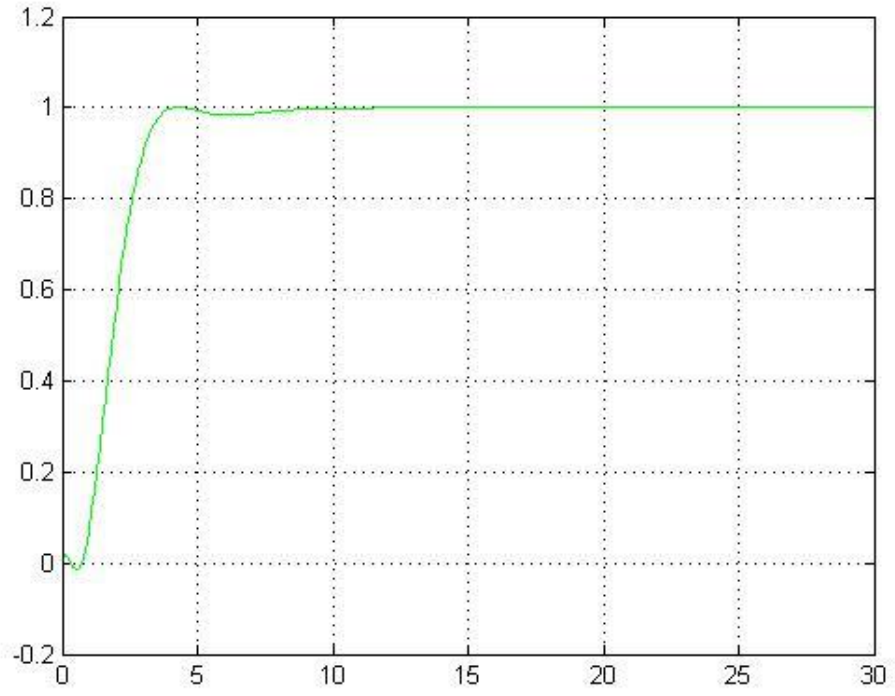
Bir diğer alternatif olarak, büyük patlama - büyük çöküş optimizasyon metodu, hem aşım değeri hem de yerleşme zamanını aynı anda minimum yapan senaryo üzerine kurulabilir. Bunu düşünerek, BPBÇ optimizasyon uygulandığında kontrolör parametreleri aşağıdaki gibi bulunmaktadır.

$$k_p = 0,6346$$

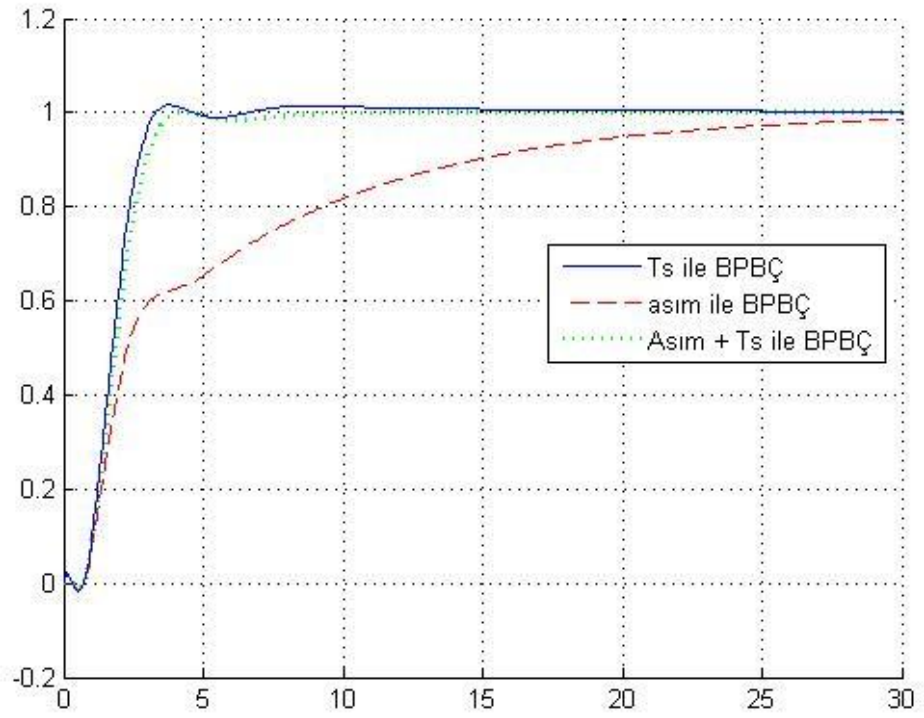
$$k_i = 0,5129$$

$$\lambda = 1,0000$$

Elde edilen parametrelere göre sistemin basamak yanıtı şekil 6.3'deki gibi elde edilmektedir. Şekil 6.3'deki basamak yanıtı incelendiğinde yerleşme zamanı 3,50 sn ve aşım değeri 0,07 olarak bulunmaktadır.



Şekil 6.3 Aşım değeri ve yerleşme zamanı aynı anda dikkate alınarak BPBÇ optimizasyon yöntemi ile elde edilen kontrolör parametrelerine göre sistemin basamak yanıtı



Şekil 6.4 Farklı BPBÇ optimizasyon yöntemine göre basamak yanıtları

Elde edilen 3 farklı sonuç, şekil 6.4'te gösterilmektedir. Ayrıca elde edilen sonuçlar çizelge 6.1'de gösterilmektedir.

Çizelge 6.1 BPBÇ optimizasyon yönteminde ele alınan farklı kriterlere göre elde edilen sonuçlar

Büyük Patlama Büyük Çöküş Metodunda ele alınan kriter	kp	ki	λ	Aşım Değeri (ζ)	Yerleşme Zamanı (Ts)
Ts	0,7680	0,5265	1,0450	1,62%	3,00 sn
ζ	0,6265	0,1777	1,0000	0,00%	27,40 sn
Ts+ ζ	0,6346	0,5129	1,0000	0,07%	3,50 sn

Bu sonuçlar kıyaslandığında optimum değerin; aşım ve yerleşme zamanının birlikte ele alınarak optimum değerin arandığı BPBÇ optimizasyonun daha iyi sonuç verdiği gözlenmektedir. Bu sonuçta, sistemin yerleşme zamanı küçük ve sistem neredeyse aşım yapmamaktadır. Ayrıca kontrolörün parametrelerine bakıldığında, bulunan kesir dereceli PI^λ kontrolörün integral derecesi 1 bulunmaktadır. Bu da bulunan kesir dereceli kontrolörün klasik PI kontrolör ile aynı yapıda olduğu anlamına gelmektedir. 1 sn ölü zamanlı sistemler için en ideal kontrolörün klasik PI tipi kontrolör olduğu söylenebilir.

Aynı sistem ele alındığında farklı L değerleri için BPBÇ optimizasyon metodu uygulandığında elde edilen parametre değerleri çizelge 6.2'de gösterilmektedir.

Çizelge 6.2 BPBÇ optimizasyon yöntemi ile farklı L değerlerine göre elde edilen parametre değerleri

L	k_p	k_i	λ	Aşım Değeri (%)
1	0,3977	0,2663	1,0148	0,28%
2	0,2747	0,1822	0,9910	0,00%
3	0,3126	0,1864	1,0003	0,10%
4	0,4514	0,1452	1,0231	0,72%
5	0,0500	0,0790	0,9948	0,00%
6	0,2995	0,0987	1,0194	0,90%
7	0,1390	0,0837	0,9807	1,89%
8	0,0500	0,0554	0,9894	0,00%
9	0,2698	0,0717	0,9963	0,25%
10	0,3222	0,0649	1,0080	0,36%
11	0,5210	0,0427	1,0457	1,09%
12	0,3802	0,0493	1,0336	1,31%
13	0,9970	0,0498	1,0174	2,26%
14	0,2597	0,0447	1,0187	2,13%
15	0,1300	0,0372	0,9970	1,08%

Elde edilen sonuçlar incelendiğinde artan ölü zaman değeri nedeniyle kararlılık bölgesinin küçüldüğünden dolayı k_p ve k_i parametrelerinin küçüldüğü gözlemlenmektedir. Aynı zamanda elde edilen integral derecesinin klasik PI tipi integral derecesine çok yakın olduğu gözlenmektedir.

İkinci örnek olarak birinci dereceden ölü zamanlı aşağıdaki sistem için kesirli dereceli PI^λ tipi kontrolörün katsayıları ve integral derecesi, büyük patlama - büyük çöküş optimizasyon metodu ile bulunsun.

$$G_D(s) = \frac{1,1875}{2,81s+1} e^{-1,63s} \quad (6.3)$$

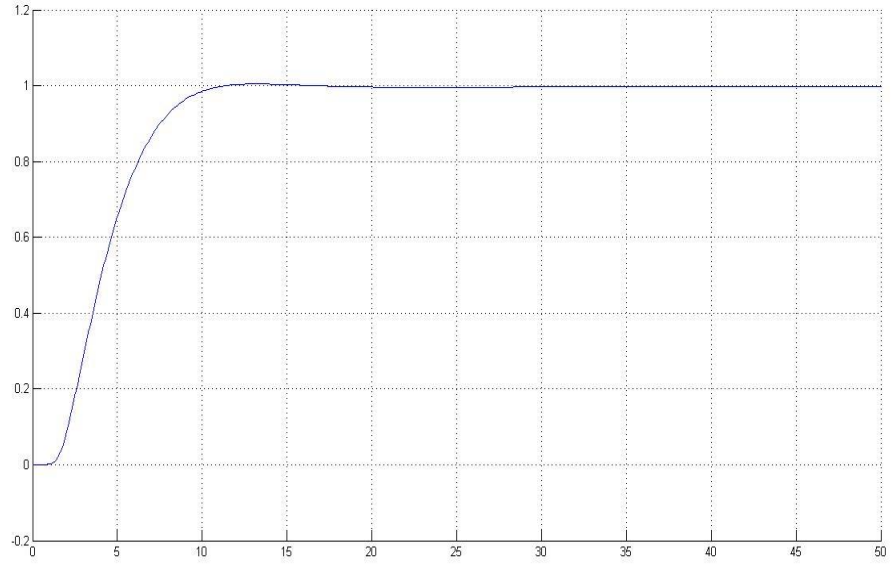
$$C(s) = k_p + k_i \frac{1}{s^\lambda}$$

Kontrolör parametrelerini belirlemek için BPBÇ optimizasyon yöntemi çalıştırıldığında aşağıda yer alan parametreler bulunmaktadır.

$$k_p=0,48267$$

$$k_i=0.200$$

$$\lambda=0,9847$$



Şekil 6.5 6.3'deki birinci dereceden ölü zamanlı sistemin basamak yanıtı

(6.3)'deki birinci dereceden sistem için elde edilen kesirli dereceli PI^λ kontrolörün integral derecesi 1'e yakındır. İncelenen her iki sistemi de kontrol etmek için elde edilen kesirli dereceli kontrolörün integral derecesi 1'e yakındır.

7. SONUÇ VE ÖNERİLER

Kesirli dereceli hesaplama ilk ortaya atıldığı yıllarda yeni bir kavram olması nedeniyle gerekli olan matematik bilgisinin eksikliğinden ve hesaplamadaki sınırlı olanaklardan dolayı çok gelişmemiştir. Son yıllarda hesaplama yapılabilecek olanaklar oluştu ve böylece kesirli dereceli denklemler kolaylıkla hesaplanmaya başladı. Buna paralel olarak kesirli dereceli hesaplamalar kontrol alanında da araştırılmaya ve uygulanmaya başladı.

Klasik kontrolörler endüstride çok yaygın olarak kullanılmasına rağmen sistemin performansını arttırmak amacıyla kesirli dereceli kontrolörlere olan ilgi gittikçe atmaktadır. Genellikle kontrolör tasarımında kullanılan kesir dereceli kontrolörler PI^λ ve $PI^\lambda D^\mu$ tipi kontrolörlerdir.

Bu çalışmada ilk olarak kesirli dereceli hesaplamaların tarihçesinden ve kesirli türev-integral kavramlarından bahsedilmektedir. Sonrasında, kontrolör tasarlanırken dikkat edilecek olan unsurlar ele alınmaktadır. Bunlardan en önemlisi kapalı çevrim sistemin kararlı olmasıdır. Bu durumda kapalı çevrim sistemi kararlı yapan kesirli dereceli kontrolörün parametrelerinin kararlılık bölgesi bulunmaktadır.

Sistem ölü zaman içerdiğinde kararlılık bölgesini kolaylıkla bulabilmek amacıyla ölü zaman yaklaşımları incelenmiş olup Pade yaklaşımının uygun olduğu belirlenmiştir. Bunun nedeni de Pade yaklaşımı ölü zamana en yakın olan yaklaşımdır ve istenilen derece ile ölü zaman yaklaşımı elde edilebilmektedir.

Ölü zaman içeren sistemler için tasarlanacak olan kesirli dereceli PI^λ kontrolörün parametrelerine (k_p , k_i , λ) bağlı olan 3 boyutlu kararlılık bölgesi elde edilmiştir. Elde edilen kararlılık bölgesi ölü zamanın yaklaşımı olan Pade yaklaşımının farklı derecelerine göre incelenmiş olup yaklaşım derecesi değiştirildiğinde kararlılık bölgesinin neredeyse değişmediği gözlenmiştir. Kararlılık bölgesi farklı ölü zaman (L) değerleri için incelendiğinde kararlılık bölgesinin küçüldüğü gözlenmiştir.

Kesirli dereceli sistemlerin performansı MATLAB / Simulink üzerinde doğrudan analiz edilememesi nedeniyle bu çalışmada gerekli olan MATLAB araç kutusu d-

tanıtılmıştır. Tanıtılan MATLAB araç kutularıyla sistemin zaman bölgesinde analizi yapılmıştır.

Bu kısma kadar yapılan araştırmalar kontrolörün parametrelerinin hangi aralıkta olmasını belirlemektedir. Sistemin istenilen performansı sağlayabilmesi için gerekli olan parametreleri bulabilmek amacıyla Büyük Patlama Büyük Çöküş (Big Bang-Big Brunch) optimizasyon algoritması kullanılmıştır. Bu optimizasyon algoritmasıyla ilk olarak kararlılık bölgesinde rastgele değerler seçilmektedir ve bu değerlerin içinden, optimum sistem performansı sağlayan değer kütle merkezi olarak seçilir. Ardından, kütle merkezinin etrafında sistem performansını iyileştirecek olan farklı parametre değerleri aranır ve bulunan sonuç kontrolör parametreleridir.

Büyük Patlama Büyük Çöküş (Big Bang- Big Brunch) optimizasyon algoritması ile elde edilen kontrolör parametreleri incelendiğinde, integral derecesinin 1'e yakın olduğu gözlenmektedir. Aynı zamanda birinci dereceden farklı sistemler için kontrolör parametreleri araştırıldığında da integral derecesinin 1'e çok yakın değere sahip olduğu gözlenmiştir. Bu durumda birinci dereceden ölü zamanlı sistemler için neredeyse en iyi performansın sağlayan kontrolörün klasik PI tipi kontrolör olduğu söylenebilir.

Ayrıca integral derecesi 1'den küçük olduğu durumlarda sistemin sürekli hal hatasına sahip olduğu gözlemlenmiştir. Bu yüzden birinci dereceden sistemler için tasarlanacak olan kesirli dereceli PI^λ kontrolörde integral derecesinin 1'den küçük olmaması gerekmektedir.

KAYNAKLAR

- [1] **Hamamci, S.E.** (2008).“Stabilization using fractional-order PI and PID controllers”. Nonlinear Dyn. 51, 329-343.
- [2] **Manabe, S.** (1960). The non-integer integral and its application to control systems. J. of IEE of Japan 80(860), 589-597 [In Japanese]. Shorter English version in: ETJ of Japan, 6(3/4), 83-87 (1961).
- [3] **Oustaloup, A.** (1990). CRONE control-principle, synthesis, performances with nonlinearities and robustnessinput immunity dilemma. Lecture Notes in Control and Information Sciences 144, 767-777.
- [4] **Podlubny, I.** (1999). Fractional Differential Equations. Academic Press, San Diego.
- [5] **Podlubny, I.** (1999) Fractional-order systems and PI_D_-controllers. IEEE Transaction on Automatic Control 44(1), 208–214.
- [6] **Y. Q. Chen.** (2009).” Fractional Order Control - A Tutorial” in 2009 American Control Conference Hyatt Regency Riverfront, St. Louis, MO, USA June 10-12.
- [7] **C. N. Zhao and D. Xue.** (2008). Closed-form solutions to fractional-order linear differential equations. Frantiers of Electrical and Electronic Engineering in China, 3(2):214–217.
- [8] **Rakhi Gopinath.** (2013).” Design of Fractional Order Controllers for First Order Plus Time Delay Systems” International Journal of Computer Applications (0975 – 8887)- International Conference on Innovations In Intelligent Instrumentation, Optimization And Signal Processing “ICIIOSP-2013”
- [9] **Erol O. K., Eksin I.** (2006)“ A New Optimization Method: Big Bang–Big Crunch” Advances in Engineering Software 37 (2006) 106–111

EKLER

EK A

fotf Fonksiyonu

```
function G=fotf(a,na,b,nb) %fotf.m
if nargin==0,
G.a=[]; G.na=[]; G.b=[];
G.nb=[]; G=class(G,'fotf');
elseif isa(a,'fotf'), G=a;
elseif nargin==1 & isa(a,'double'),
G=fotf(1,0,a,0);
else,
ii=find(abs(a)<eps); a(ii)=[];
na(ii)=[]; ii=find(abs(b)<eps);
b(ii)=[]; nb(ii)=[]; G.a=a;
G.na=na; G.b=b; G.nb=nb;
G=class(G,'fotf');
end
```

display Fonksiyonu

```
function display(G) %display.m
strN=polydisp(G.b,G.nb);
strD=polydisp(G.a,G.na);
nn=length(strN); nd=length(strD);
nm=max([nn,nd]); n1=(nm-nn)/2;
disp([char(' '*ones(1,floor(-n1))) strN])
disp(char('-'*ones(1,nm)))
disp([char(' '*ones(1,floor(n1))) 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 & strP(1:2)=='1s',
strP=strP(2:end);
end
```

EK B

çarpma fonksiyonu

```
function G=mtimes(G1,G2) % mtimes.m
G2=fotf(G2); a=kron(G1.a,G2.a);
b=kron(G1.b,G2.b); na=[]; nb=[];
for i=1:length(G1.na),
na=[na,G1.na(i)+G2.na];
end
for i=1:length(G1.nb),
nb=[nb,G1.nb(i)+G2.nb];
end
G=unique(fotf(a,na,b,nb));
```

toplama foksiyonu

```
function G=plus(G1,G2) % plus.m
a=kron(G1.a,G2.a); na=[]; nb=[];
b=[kron(G1.a,G2.b),kron(G1.b,G2.a)];
for i=1:length(G1.a),
na=[na G1.na(i)+G2.na];
nb=[nb, G1.na(i)+G2.nb];
end
for i=1:length(G1.b),
nb=[nb G1.nb(i)+G2.na];
end
G=unique(fotf(a,na,b,nb));
```

Unique fonksiyonu

```
function G=unique(G) % common unique.m
[a,n]=polyuniq(G.a,G.na); G.a=a; G.na=n;
[a,n]=polyuniq(G.b,G.nb); G.b=a; G.nb=n;
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
end
```

EK C

Geribesleme fonksiyonu

```
function G=feedback(F,H) % feedback.m
H=fotf(H); b=kron(F.b,H.a); na=[]; nb=[];
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=unique(fotf(a,na,b,nb));
```

fonksiyonun toplama işlemine göre tersi alan fonksiyon

```
function G=uminus(G1) % uminus.m
G=fotf(G1.a,G1.na,-G1.b,G1.nb);
```

Çıkarma fonksiyonu

```
function G=minus(G1,G2) % minus.m
G=G1+(-G2);
```

fonksiyonun çarpma işlemine göre tersi alan fonksiyon

```
function G=inv(G1) % inv.m
G=fotf(G1.b,G1.nb,G1.a,G1.na);
```

tam sayı dereceli sistemi, kesir dereceli sistem biçiminde ifade eden fonksiyon

```
function G1=fotf(G)
[n,d]=tfdata(G,'v');
i1=find(abs(n)<eps); i2=find(abs(d)<eps);
if length(i1)>0 & i1(1)==1, n=n(i1(1)+1:end); end
if length(i2)>0 & i2(1)==1, d=d(i2(1)+1:end); end
G1=fotf(d,length(d)-1:-1:0,n,length(n)-1:-1:0);
```

EK D

Kararlılık testi fonksiyonu

```
function [K,q,err,apol]=isstable(G) %isstable.m
a=G.na; al=fix(100*a); n=gcd(al(1),al(2));
for i=3:length(al), n=gcd(n,al(i)); end
q=n/100, a=fix(al/n), b=G.a;
c(a+1)=b; c=c(end:-1:1);
p=roots(c); p=p(abs(p)>eps);
err=norm(polyval(c,p));
plot(real(p),imag(p),'x',0,0,'o')
apol=min(abs(angle(p))); K=apol>q*pi/2;
xm=xlim; xm(1)=0; line(xm,q*pi/2*xm)
```

EK E

Step fonksiyonu

```
function y=step(G,t)
u=ones(size(t)); y=lsim(G,u,t);
if nargout==0, plot(t,y); end
```

lsim fonksiyonu

```
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));
end
for i=2:nT, A=[y1(i-1:-1:1)]'*W(2:i,1:nA);
y1(i)=(u(i)-sum(A.*a./[h.^eta]))/D;
end
for i=2:nT,
y(i)=(W(1:i,nA+1:end)*D1)'*[y1(i:-1:1)];
end
```

EK F

Step fonksiyonu

```
function H=bode(G,w)
a=G.a; na=G.na; b=G.b; nb=G.nb; j=sqrt(-1);
if nargin==1, w=logspace(-4,4); end
for i=1:length(w)
P=b*((j*w(i)).^nb. ');
Q=a*((j*w(i)).^na. '); H1(i)=P/Q;
End

H1=frd(H1,w);
if nargin==0, bode(H1); else, H=H1; end
```

nyquist Fonksiyonu

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

Nichols fonksiyonu

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


ÖZGEÇMİŞ

Ad Soyad : Emre BALKIŞ

Doğru Yeri ve Tarihi : Bursa, 13.05.1989

Adres : Osmangazi / Bursa

Lisans : İzmir Yüksek Teknoloji Enstitüsü