İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

LİNEER SİSTEMLER İÇİN KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI: BİLGİSAYAR CEBRİ YAKLAŞIMI

YÜKSEK LİSANS TEZİ

Uğur YILDIRIM

Anabilim Dalı : Kontrol Mühendisliği

Programı : Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı : Doç. Dr. M. Turan SÖYLEMEZ

Haziran 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

LİNEER SİSTEMLER İÇİN KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI: BİLGİSAYAR CEBRİ YAKLAŞIMI

YÜKSEK LİSANS TEZİ
Uğur YILDIRIM
(504081144)

Tezin Enstitüye Verildiği Tarih : 07.05.2010
Tezin Savunulduğu Tarih : 15.06.2010

Tez Danışmanı : Doç. Dr. M. Turan SÖYLEMEZ (İTÜ)

Diğer Jüri Üyeleri : Prof. Dr. Leyla GÖREN SÜMER (İTÜ)

Prof. Dr. M. K. Külmiz ÇEVİK (İTÜ)

Haziran 2010

Anneme ve babama,

ÖNSÖZ

Yüksek lisans tezi boyunca bilgi ve deneyimleri ile bana her konuda destek olan değerli tez danışmanım Doç. Dr. Mehmet Turan SÖYLEMEZ'e teşekkürü bir borç bilirim.

Ayrıca bu bitirme projesini en iyi şekilde bitirmem için maddi ve manevi desteği esirgemeyen, bütün hayatım boyunca bana destek olan aileme en içten dileklerimle teşekkür ederim.

Mayıs 2010

Uğur YILDIRIM

Kontrol Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **v**
İÇİNDEKİLER ..... **vii**
KISALTMALAR ..... **ix**
ŞEKİL LİSTESİ ..... **xi**
ÖZET ..... **xiii**
SUMMARY ..... **xv**
1. GİRİŞ **1**
2. KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI **3**
2.1 Amaç 3
2.2 Tekil Frekanslar 3
2.3 Kararlılığı Sağlayan K_p Aralığının Bulunması 5
2.3.1 Teorem 2.1 (N. Bajcinca, 2006) 5
2.4 Seçilen Bir K_p Değeri İçin Kararlı K_i-K_d Bölgesinin Bulunması 5
2.5 Örnek 7
3. PARAMETRE BELİRSİZLİĞİ İÇEREN SİSTEMLER İÇİN DAYANIKLI KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI **11**
3.1 Amaç 11
3.2 Teorem 3.1 (Kharitonov Teoremi) 11
3.3 Parametrik Belirsizlik İçeren Sistemler için Köşe ve Kenar Sistemlerinin Bulunması 12
3.4 Örnek 12
4. KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASINA İLİŞKİN FONKSİYON KÜMESİNİN YAZILIMI **15**
4.1 Amaç 15
4.2 Yazılan Fonksiyonlar ve İşlevleri 15
4.2.1 ListSum fonksiyonu 15
4.2.2 UnstablePoleNumber fonksiyonu 16
4.2.3 IsPointInPolygon fonksiyonu 17
4.2.4 FindConvexHull fonksiyonu 18
4.2.5 ConvexPolygonIntersection fonksiyonu 18
4.2.6 FindStableConditions fonksiyonu 20
4.2.7 FindKpRange fonksiyonu 21
4.2.8 FindPID fonksiyonu 22
4.2.9 RegionIntersection fonksiyonu 24
4.2.10 FindRobustPID fonksiyonu 25
4.2.11 FindPID3D fonksiyonu 26
4.2.12 FindRobustPID3D fonksiyonu 27
4.2.13 GraphPID ve GraphPID3D fonksiyonları 28

4.2.14 ShowPI ve ShowPD fonksiyonları 29
4.3 Fonksiyonların Birbirleriyle Bağlantıları 30
4.4 Fonksiyon Kümesinin Doğruluk ve Performans Testleri 31
5. SONUÇLAR 37
6. KAYNAKLAR 39
EKLER 41

KISALTMALAR

PID **:** Proportional-integral-derivative
PI **:** Proportional-integral
PD **:** Proportional-derivative
$\mathbf{K_p}$ **:** Oransal kazanç
$\mathbf{K_i}$ **:** İntegral kazancı
$\mathbf{K_d}$ **:** Türev kazancı

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Örnek sistem için K_p - tekil frekans ilişkisi. 8
Şekil 2.2: Örnek sistem için K_i-K_d düzlemindeki bölgeler (K_p = -18). 8
Şekil 2.3: Örnek sistem için kararlı PID kontrolör bölgesi. 10
Şekil 3.1: Örnek sistem için K_p=1.05 değerindeki kenar sistemlerinin kararlı bölgeleri. 13
Şekil 4.1: ListSum fonksiyonunun akış diyagramı. 16
Şekil 4.2: UnstablePoleNumber fonksiyonunun akış diyagramı. 16
Şekil 4.3: IsPointInPolygon fonksiyonunun akış diyagramı. 17
Şekil 4.4: FindConvexHull fonksiyonunun akış diyagramı. 18
Şekil 4.5: ConvexPolygonIntersection fonksiyonunun akış diyagramı. 19
Şekil 4.6: FindStableConditions fonksiyonunun akış diyagramı. 20
Şekil 4.7: FindKpRange fonksiyonunun akış diyagramı. 21
Şekil 4.8: FindPID fonksiyonunun akış diyagramı. 23
Şekil 4.9: RegionIntersection fonksiyonunun akış diyagramı. 24
Şekil 4.10: FindRobustPID fonksiyonunun akış diyagramı. 25
Şekil 4.11: FindPID3D fonksiyonunun akış diyagramı. 26
Şekil 4.12: FindRobustPID3D fonksiyonunun akış diyagramı. 27
Şekil 4.13: GraphPID fonksiyonunun akış diyagramı. 28
Şekil 4.14: ShowPI fonksiyonunun akış diyagramı. 29
Şekil 4.15: Fonksiyonların birbirleriyle bağlantıları. 31
Şekil 4.16: Örnek sistem ve K_p=-18 değeri için K_i-K_d düzlemindeki kararlı bölge. . 32
Şekil 4.17: Örnek sistem için farklı K_p değerlerinde kararlı bölgeler. 32
Şekil 4.18: Örnek sistem için 3 boyutlu kararlı PID kontrolör bölgesi. 33
Şekil 4.19: Örnek sistem için 3 boyutlu kararlı PID kontrolör bölgesi. 34
Şekil 4.20: Örnek sistem için PI ve PD kontrolör bölgeleri. 34
Şekil 4.21: Örnek sistem için kenar sistemlerinin kararlı olduğu bölgeler. 35
Şekil 4.22: Örnek sistem için dayanıklı kararlı PID kontrolör bölgesi. 36

LİNEER SİSTEMLER İÇİN KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI: BİLGİSAYAR CEBRİ YAKLAŞIMI

ÖZET

Endüstriyel uygulamalarda çok kullanılan kontrolör çeşitlerinden birisi PID kontrolördür. PID kontrolörler yapısında oransal (proportional), türev (derivative) ve integral kazançları bulundurmaktadırlar. Yapılarındaki oransal kazanç ile o anki hataya göre bir kontrolör işareti üretilip sistemin kontrol edilmesi sağlanır. Sahip oldukları integratör ile geçmişte yapılan hataları da göz önünde bulundururlar. Ayrıca yine integratör sayesinde birim basamak girişler için orijinde sıfırı olmayan sistemlerin sürekli hal hatalarını sıfırlayabilmektedirler. PID kontrolör yapısındaki türev alıcı ile de gelecekte yapılacak hatalar öngörülerek sistemin daha iyi bir geçici hal yanıtına sahip olması sağlanabilir. Kontrolörün sistem kök eğrisine eklediklerine bakılacak olursa da orijinde bir kutba ve istenildiği gibi konumlandırılabilecek iki sıfıra sahip olduğu görülmektedir. Ayrıca kontrolör kazancı ile de kapalı çevrim kutuplarının kök eğrisi üzerinde istenilen konuma getirilmesi sağlanabilir. Tasarım kolaylığı ve dayanıklı yapısı ile PID kontrolörler, diğer kontrolör türlerine göre daha çok tercih edilmektedirler.

Sağladığı avantajlara rağmen verilen bir sistem için kararlılığı sağlayan tüm PID kontrolörlerin bulunması zor ve bilgisayar kullanılmadan çözülmesi uzun zaman alan bir problemdir. Literatürdeki yöntemlerde bu problemin çözümü için tekil frekanslardan yararlanılmıştır. Tekil frekanslarda sanal eksen üzerinden kutup geçişi olduğu için kapalı çevrim sistemin kararsız kutup sayısında değişimler meydana gelir. Bu değişimlerden yararlanarak öncelikle PID kontrolörün oransal kazancının (K_p) kararlılığı sağlayabileceği bir aralık bulunur. Daha sonra ise bu aralıktaki K_p kazançları için sistemin kararlı olduğu $K_i - K_d$ bölgesi bulunur. Bu bölgenin bulunmasında da yine tekil frekanslar kullanılır. $K_i - K_d$ düzleminde tekil frekanslar sınır doğruları oluşturur ve bu doğruların ortaya çıkardığı bölgelerden de kararlı bölge bulunur.

Bu tezde parametrik belirsizlik içermeyen ve içeren ölü zamansız sistemler için bu problemin çözümüyle ilgili literatürdeki bazı yöntemler aktarılacaktır. Daha sonra ise bu çözüm yöntemleri kullanılarak Mathematica programlama ortamında bir fonksiyon kümesinin yazılımı anlatılacaktır. Bu fonksiyon kümesinin içinde yöntemde kullanılan teoremleri, kararlı bölgenin elde edilmesi için yapılan geometrik işlemleri yapan ve tüm bu işlemlerin otomatik olarak tek bir komutla yapılmasını sağlayan fonksiyonlar bulunmaktadır. Ayrıca elde edilen sonuçların grafiksel olarak kullanıcıya gösterilmesini sağlayan fonksiyonlar da bulunmaktadır.

CALCULATION OF ALL STABILIZING PID CONTROLLERS FOR LINEAR SYSTEMS: COMPUTER ALGEBRA APPROACH

SUMMARY

One of the well-known and commonly used controllers in industrial applications are PID-type controllers. These controllers consist of proportional, derivative and integral gains. The proportional gain generates a control signs to by using the current value of the error. The integrator removes the steady state error completely for step references while the derivative of the PID controller can predict future errors and give a better time domain characteristics to the system. A PID controller adds one pole on the origin and two adjustable zeros to the root locus plot. Besides the position of the closed loop system poles can be changed by increasing or decreasing the controllers gain. Because of their numerous advantages including their easy design and robust structure PID controllers are more preferable among other controllers.

Despite being advantageous, finding all stabilizing PID controllers for a given system is a hard and time-consuming problem. In the literature, the methods for solving this problem are based on the usage of the singular frequencies. Unstable closed loop pole number changes at the singular frequencies because poles cross over the imaginary axis at singular frequencies. By using this property, a stabilizing proportional gain (K_p) interval can be found for a PID controller. Then, for the gridded K_p values in the interval, stabilizing $K_i - K_d$ region is again found by using singular frequencies. On the $K_i - K_d$ plane, boundary lines occur at the singular frequencies and these lines divide the plane into several regions. The stabilizing regions are found by calculating the unstable pole number for each of the regions that are on this plane.

In this thesis, some solution methods of this problem will be given for the certain and parametric uncertain delay-free systems. Then, by using these solution methods, a function set development with Mathematica will explained. This function set includes functions to apply the theorems and some geometric calculations, which are used in the solution methods. Besides, there are functions to find the solution for a given system automatically, along with the functions that are used to show the solution graphically to the user.

1. GİRİŞ

Günümüzde modern kontrolün yüksek dereceden kontrolörlere izin vermesine rağmen halen daha PID kontrolörler yaygın olarak kullanılmaktadır [1]. Bunun sebebi PID kontrolörlerin hem integral kazancı ile geçmişi, hem oransal kazancı ile içinde bulunduğu durumu, hem de türev kazancı ile geleceği öngörmesi ile referans ile çıkış arasındaki hatayı iyi bir şekilde bastırabilmesidir [2]. Performanslarının yanı sıra sağladıkları tasarım kolaylığıyla da PID kontrolörler çok farklı sistemler için kullanılabilmektedirler [3].

Öte yandan bir sistemin kararlı olmasını sağlayan tüm PID kontrolörlerin bulunması zor bir problemdir. Bu problemin çözümü için çeşitli çalışmalar yapılmıştır [4-7]. İncelenen bu tür sistemler için kararlı PID kontrolör bölgesinin bulunması problemi temel olarak iki alt probleme ayrılabilir. Bunlardan birincisi K_i ve K_d kazançlarından bağımsız olarak kararlı K_p aralıkları bulunması, ikincisi ise bulunan K_p aralıklarından seçilen bir değer için $K_i - K_d$ düzlemindeki kararlı bölgenin bulunmasıdır [8].

Belirsiz sistemler için de PID kontrol kullanılmaktadır. Yapılan çalışmalar [9-12] ile parametre belirsizliği içeren sistemler için kararlı PID kontrolör bölgesinin çeşitli yöntemlerle bulunabileceği gösterilmiştir.

Bu tezde literatürdeki yöntemler kullanılarak kararlılığı sağlayan tüm PID kontrolörlerin bulunmasına ilişkin bir fonksiyon kümesinin yazılımı hakkında bilgi verilecektir. Ayrıca, yazılan fonksiyonların sağladığı işlem kolaylıkları ve zamandan sağladığı kazançlar gösterilecektir.

2. KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI

2.1 Amaç

Transfer fonksiyonu

$$G(s) = \frac{N(s)}{D(s)} = \frac{a_m s^m + a_{m-1} s^{m-1} + \cdots a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + \cdots b_1 s + b_0} \quad \textbf{(2.1)}$$

şeklinde olan belirsizlik içermeyen ölü zamansız nedensel bir sistemi kararlı yapan transfer fonksiyonu

$$F(s) = K_p + \frac{K_i}{s} + K_d\, s \quad \textbf{(2.2)}$$

şeklinde olan PID kontrolör bölgesinin bulunması temel olarak iki farklı problemden oluşmaktadır. Bunlardan ilki sistemi kararlı yapabilecek olan K_p kazanç aralığının bulunmasıdır. Bu aralığın bulunması için sistemin hangi K_p değerinde kaç tekil frekansa sahip olduğu incelenir. İkinci problem ise seçilen bir K_p değeri için $K_i - K_d$ düzleminde kararlı bölgenin bulunmasıdır. Bu bölgenin bulunmasında da yine tekil frekanslardan yararlanılır. Her bir tekil frekansın $K_i - K_d$ düzleminde bir sınır doğrusu belirttiği için düzlemdeki kararlı bölge konveks çokgenler şeklinde olacaktır. Bu bölümde, belirtilen türdeki sistemler için kararlılığı sağlayan tüm PID kontrolörlerin bulunmasına ilişkin yöntemler aktarılacaktır.

2.2 Tekil Frekanslar

Kararlılığı sağlayan PID kontrolör bölgesi bulunurken tekil frekanslardan sıkça yararlanılır. Tekil frekanslar kapalı çevrim sistem kutuplarının sanal eksen üzerinden geçtikleri frekanslardır. Dolayısıyla tekil frekanslarda kapalı çevrim sistemin kararsız kutup sayısında değişim meydana gelir. Bu değişimden de yararlanılarak sistemin kararlı olduğu bölge bulunabilir.

Transfer fonksiyonu (2.1)'deki gibi olan bir sistem ile transfer fonksiyonu (2.2)'deki gibi olan bir PID kontrolör birim geri besleme ile kapalı çevrime alınırsa, kapalı çevrim sistemin transfer fonksiyonu

$$T(s) = \frac{G(s)F(s)}{1 + G(s)F(s)} = \frac{\left(K_p\, s + K_i + K_d\, s^2\right) N(s)}{s\, D(s) + \left(K_p\, s + K_i + K_d\, s^2\right) N(s)} \qquad \textbf{(2.3)}$$

olarak bulunur. Kapalı çevrim sistem transfer fonksiyonunun paydasına bakılarak kapalı çevrim sistemin karakteristik polinomu

$$\delta(s) = s\, D(s) + \left(K_p\, s + K_i + K_d\, s^2\right) N(s) \qquad \textbf{(2.4)}$$

olarak elde edilir. Burada $N(s)$ yerine $A(s)$ ve $s\, D(s)$ yerine $B(s)$ yazılırsa

$$\delta(s) = B(s) + \left(K_p\, s + K_i + K_d\, s^2\right) A(s) \qquad \textbf{(2.5)}$$

elde edilir. Daha sonra ise s yerine $j\omega$ yazılırsa; R_A, R_B, I_A ve I_B sırasıyla $A(j\omega)$ ve $B(j\omega)$ polinomlarının gerçek ve sanal kısımlarını göstermek üzere

$$\delta(s) = (R_B + j\, I_B) + \left(K_p\, j\omega + K_i - K_d\, \omega^2\right)(R_A + j\, I_A) \qquad \textbf{(2.6)}$$

elde edilir. Bu ifade

$$\begin{bmatrix} R_A & -I_A \\ I_A & R_A \end{bmatrix} \left(\begin{bmatrix} 1 & -\omega^2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K_i \\ K_d \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\, K_p \end{bmatrix} \right) + \begin{bmatrix} R_B \\ I_B \end{bmatrix} = 0 \qquad \textbf{(2.7)}$$

şeklinde matris olarak gösterilip biraz daha düzenlenerek

$$\begin{bmatrix} 1 & -\omega^2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K_i \\ K_d \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\, K_p \end{bmatrix} + \frac{1}{{R_A}^2 + {I_A}^2} \begin{bmatrix} R_A R_B + I_A I_B \\ R_A I_B - I_A R_B \end{bmatrix} = 0 \qquad \textbf{(2.8)}$$

şeklinde yazılabilir. Buradan kapalı çevrim sistemin kutuplarının sanal kısımlarının sadece K_p kazancına bağlı olduğu görülür. Bu ifadeden K_p değeri çekilecek olursa

$$K_p = -\frac{R_A I_B - I_A R_B}{\omega({R_A}^2 + {I_A}^2)} \qquad \textbf{(2.9)}$$

ifadesi elde edilir. Bu ifade frekans değeri değiştirilerek hangi tekil frekansın hangi K_p kazancında oluştuğu bulunabilir. Böylelikle seçilen bir K_p değeri için kaç tane tekil frekans olduğu görülebilir. Bu tekil frekanslar hem kararlılığı sağlayan K_p aralığının belirlenmesinde hem de $K_i - K_d$ düzlemindeki sınır doğrularının elde edilmesinde kullanılır.

2.3 Kararlılığı Sağlayan K_p Aralığının Bulunması

Verilen sistem için K_p kazancı ile tekil frekanslar arasındaki bağıntı bulunduktan sonra aşağıdaki teoremden yararlanılarak kararlılığı sağlayabilecek K_p kazanç aralığı belirlenir.

2.3.1 Teorem 2.1 (N. Bajcinca, 2006)

Verilen sistemin sanal eksen üzerinde sıfırı olmadığı var sayımı altında, kapalı çevrim sistemin karakteristik polinomunun derecesi N, sistem pay polinomunun derecesi M, sistemin sağ yarı düzlem sıfır sayısı P ve sistemin tekil frekans sayısı Z olmak üzere

$$Z \geq \begin{cases} \dfrac{N-M+2P+2}{2}, N+M\ tekse \\ \dfrac{N-M+2P}{2}, N+M\ çiftse \end{cases} \quad \textbf{(2.10)}$$

koşulunu sağlayan tekil frekans sayısına sahip K_p değerleri için verilen sistemi kararlı yapan bir PID kontrolör bulunabilir.

2.4 Seçilen Bir K_p Değeri İçin Kararlı K_i-K_d Bölgesinin Bulunması

Sistemi kararlı yapabilecek K_p kazanç aralığı bulunduktan sonra bu aralıktan belli aralıklarla K_p değerleri seçilerek $K_i - K_d$ düzlemindeki kararlı bölge bulunur. Bu bölgenin bulunması için yine tekil frekanslardan yararlanılmaktadır. $K_i - K_d$ düzlemindeki kararlı bölgenin bulunması için öncelikle

$$v(s) = \delta(s)\, N(-s) \quad \textbf{(2.11)}$$

polinomu hesaplanır. Daha sonra bu polinom

$$v(s) = v_{even}(s^2, K_i, K_d) + s\, v_{odd}(s^2, K_p) \quad \textbf{(2.12)}$$

şeklinde tek ve çift kısımlarına ayrılır. Polinomun tek kısmı olan $v_{even}(s)$ polinomunda K_i ve K_d kazançlarının, $v_{odd}(s)$ polinomunda ise K_p kazancının kaldığı görülür. Bu polinomlarda s yerine $j\omega$ yazılarak

$$v_{odd}(-\omega^2, K_p) = 0 \quad \textbf{(2.13)}$$

ve

$$v_{even}(-\omega^2, K_i, K_d) = 0 \quad \textbf{(2.14)}$$

ifadeleri elde edilir. Seçilen bir K_p değeri için (2.13) ifadesinin kökleri bulunarak o K_p değerindeki tekil frekanslar elde edilir. Elde edilen bu tekil frekansların her birinde sanal eksen üzerinden kutup geçişi olur. Dolayısıyla tekil frekanslarda kapalı çevrim sistemin kararsız kutup sayısında bir değişme meydana gelir. v_{even} polinomu tekil frekanslarda incelenecek olursa $K_i - K_d$ düzlemine ilişkin sınır doğruları elde edilir. Tüm tekil frekanslara göre sınır doğruları çizilince $K_i - K_d$ düzlemi konveks çokgenler şeklinde bölgelere ayrılmış olur. Bu bölgelerin her birinde kapalı çevrim sistemin kararsız kutup sayısı eşittir. Kararsız kutup sayısının sıfır olduğu bölgelerin birleşimi, işlem yapılan K_p değeri için $K_i - K_d$ düzlemindeki kararlı bölgeyi oluşturur.

Sınır doğruları çizildikten sonra kararlı bölgenin bulunması için (S. P. Bhattacharyya ve diğ., 2009)'da verilen yöntem de kullanılabilir. Bu yöntem için öncelikle

$$\sigma(v) = n - m + 1 + 2z^+ \quad \textbf{(2.15)}$$

ifadesinin değeri bulunur. Burada n ve m incelenen sistemin sırasıyla payda ve pay polinomlarının derecelerini göstermektedir. z^+ ise incelenen sistemin sağ yarı düzlem sıfır sayısını belirtmektedir.

$\sigma(v)$ değeri hesaplandıktan sonra işlem yapılan K_p değeri için bulunan tekil frekanslar küçükten büyüğe doğru $0 < \omega_1 < \omega_2 < \cdots < \omega_{l-1}$ şeklinde sıralanır. Burada $\omega_0 = 0$ ve $\omega_l = \infty$ olarak alınır. Daha sonra ise

$$j = sgn[v_{odd}(-\omega^2, K_p)] \quad \textbf{(2.16)}$$

ifadesinin değeri hesaplanır. Hesaplanan j değerine göre $n + m$ çiftse

$$j(i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1} + (-1)^l i_l) = \sigma(v) \quad \textbf{(2.17)}$$

ifadesi, $n + m$ tekse de

$$j(i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1}) = \sigma(v) \quad \textbf{(2.18)}$$

ifadesi incelenir. Oluşturulan ifadede i değişkenleri yerine +1 veya -1 yazılarak olası tüm diziler oluşturulur ve toplamı $\sigma(v)$ değerine eşit olan diziler $K_i - K_d$ düzlemindeki kararlı bölgeleri gösterir. Dizideki her bir i değişkeni $K_i - K_d$ düzlemindeki sınır doğrularının hangi tarafının alınacağını gösterir. Böylelikle kararlı bir dizi için tüm sınır doğrularının istenilen tarafları alındığında oluşan konveks çokgen bölge kararlı bölgeyi oluşturur.

Bir K_p değeri için kararlı bölge bulunduktan sonra aynı işlemler seçilen diğer K_p değerleri için de tekrarlanarak üç boyutlu PID kontrolör parametrelerinin belirttiği kararlı bölge bulunmuş olur.

2.5 Örnek

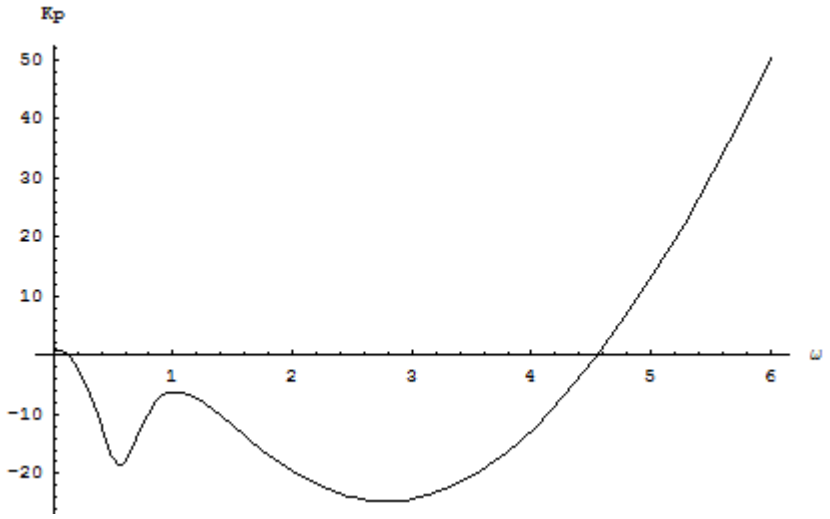
Belirsizlik içermeyen zaman gecikmesiz sistemleri kararlı yapan tüm PID kontrolörlerinin bulunmasına ilişkin örnek olarak transfer fonksiyonu

$$G(s) = \frac{s^3 - 2s^2 - s - 1}{s^6 + 2s^5 + 32s^4 + 26s^3 + 65s^2 - 8s + 1} \quad \textbf{(2.19)}$$

şekilde olan bir sistem incelenecek olursa (2.9) ifadesinden K_p ile tekil frekanslar arasındaki bağıntı bulunacak olursa

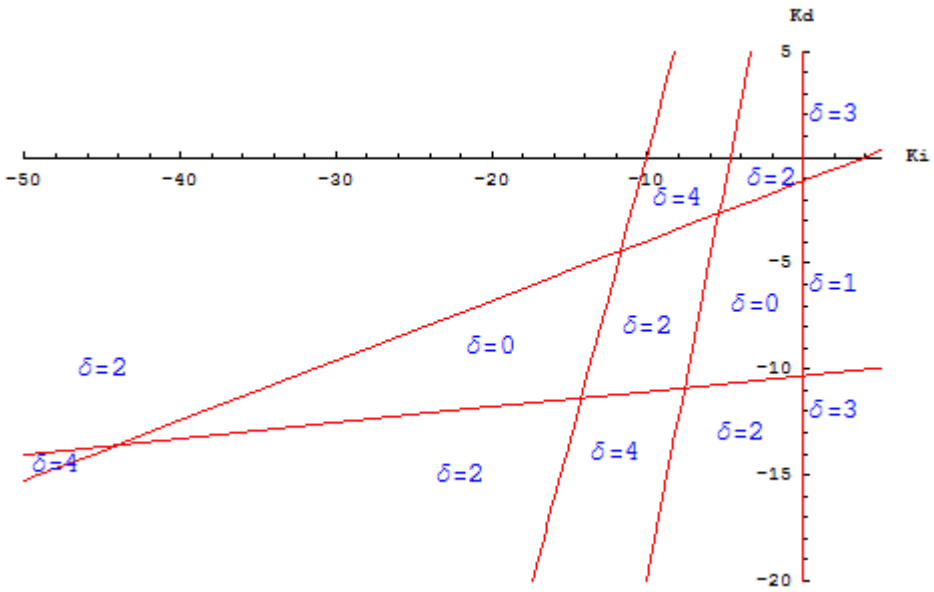
$$K_p = \frac{4\omega^8 - 89\omega^6 + 128\omega^4 - 75\omega^2 + 1}{\omega^6 + 6\omega^4 - 3\omega^2 + 1} \quad \textbf{(2.20)}$$

bulunur. Değişik ω değerleri için K_p hesaplanıp çizdirilirse Şekil 2.1'deki grafik elde edilir. Kapalı çevrim sistemin karakteristik polinomunun derecesi $N = 7$, verilen sistemin pay polinomunun derecesi $M = 3$ ve sistemin sağ yarı düzlem sıfır sayısı $P = 1$ olarak bulunduktan sonra (2.10) ifadesinden sistemi kararlı yapabilecek K_p kazançlarında en az 3 adet tekil frekansa sahip olması gerektiği görülür.



Şekil 2.1: Örnek sistem için K_p - tekil frekans ilişkisi.

En az 3 tekil frekansa sahip K_p aralığına bakıldığı zaman $-24{,}75 < K_p < 1$ aralığı bulunur. Bu aralık için $K_i - K_d$ düzleminde kararlı bir bölge bulunabilir. $K_p = -18$ değeri için (2.13) ifadesinden tekil frekanslar hesaplanacak olursa $\omega_0 = 0$, $\omega_1 = 0{,}5195$, $\omega_2 = 0{,}6055$, $\omega_3 = 1{,}8804$ ve $\omega_4 = 3{,}6848$ frekansları elde edilir. Bu frekanslar (2.14) ifadesinde yerin koyularak $K_i - K_d$ düzlemindeki sınır doğruları elde edilir. Sınır doğruları çizdirilip oluşan bölgelerdeki kararsız kutup sayısına bakılacak olursa Şekil 2.2'deki grafik elde edilir.



Şekil 2.2: Örnek sistem için K_i-K_d düzlemindeki bölgeler (K_p = -18).

Bu grafikte δ değişkenleri kapalı çevrim sistemin kararsız kutup sayısını göstermekte olup $\delta = 0$ olan bölgeleri kararlı bölgeleri ifade etmektedir. Diğer yöntem ile kararlı bölge bulunacak olursa (2.17) ifadesinden

$$(-1)\{i_0 - 2i_1 + 2i_2 - 2i_3 + 2i_4 - i_5\} = 6 \quad \textbf{(2.21)}$$

elde edilir. Bu eşitliği sağlayan diziler ise

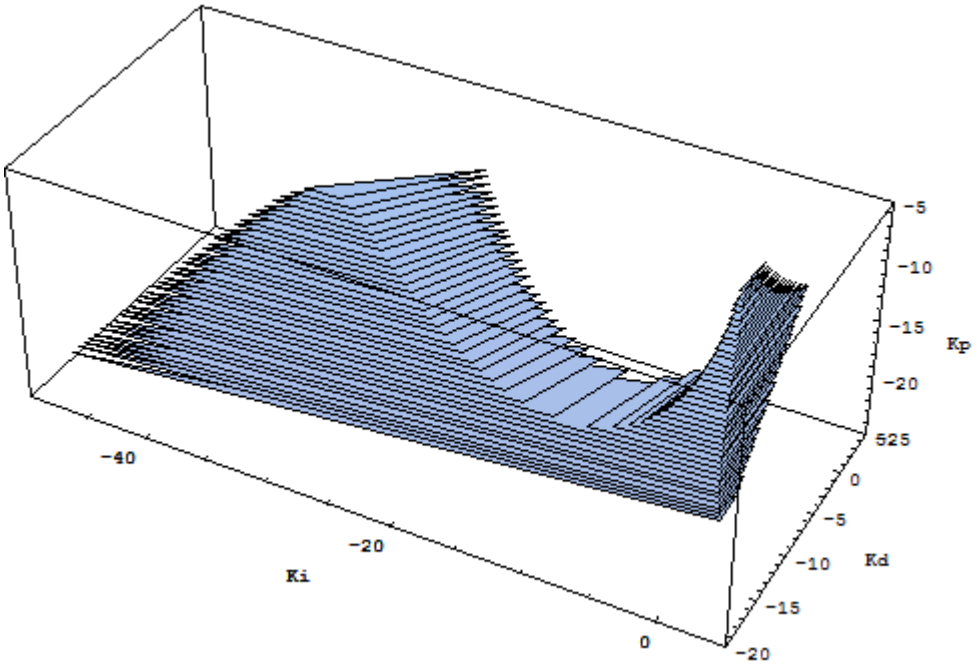
$$\begin{Bmatrix} I_1 = \{-1,-1,-1,1,-1,1\} \\ I_2 = \{-1,1,1,1,-1,1\} \\ I_3 = \{-1,1,-1,-1,-1,1\} \\ I_4 = \{-1,1,-1,1,1,1\} \\ I_5 = \{-1,1,-1,1,1,1\} \end{Bmatrix} \quad \textbf{(2.22)}$$

olarak bulunur. I_1 kümesi için şartlar sınır doğrularına uygulanacak olursa

$$\begin{Bmatrix} K_i < 0 \\ 3{,}03 - 0{,}17K_d + 0.65\ K_i < 0 \\ 7{,}62 - 0{,}28K_d + 0.76\ K_i < 0 \\ -428{,}8 - 387{,}57K_d + 109{,}61\ K_i > 0 \\ -500459 - 48465{,}1K_d + 3569{,}47\ K_i < 0 \end{Bmatrix} \quad \textbf{(2.23)}$$

bölgeleri bulunur. Bu bölgelerin kesişimi alınarak elde edilen bölge kararlı bölgedir. Aynı işlemler (2.22)'deki diğer şartlar için de yapılarak $K_p = -18$ değeri için kararlı bölge aynı şekilde bulunur.

Tüm bu işlemler bütün K_p değerleri için yapıldığında ise Şekil 2.3'de gösterilen üç boyutlu PID kontrolör bölgesi elde edilmiş olur.



Şekil 2.3: Örnek sistem için kararlı PID kontrolör bölgesi.

3. PARAMETRE BELİRSİZLİĞİ İÇEREN SİSTEMLER İÇİN DAYANIKLI KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASI

3.1 Amaç

Bu bölümde transfer fonksiyonu (2.1)'deki gibi olan bir sistemin $a_i \in [a_i^-, a_i^+]$ ve $b_i \in [b_i^-, b_i^+]$ şeklinde parametrik belirsizlik içerdiği durumlar için dayanıklı kararlı PID kontrolör bulunmasına ilişkin (J. Silva ve diğ., 2005)'deki yöntem aktarılacaktır. Yöntemde öncelikle Kharitonov Teoremi ile köşe sistemler elde edilmektedir. Daha sonra ise bu köşe sistemler ile yola çıkarak kenar sistemleri elde edilir ve tüm elde edilen bu sistemler için kararlı bölgeler bulunur. Bulunan tüm bölgelerin kesişimi alınarak verilen parametrik belirsizlik içeren sistemi dayanıklı kararlı yapan PID kontrolör bölgesi bulunmuş olur. Bunun için öncelikle Kharitonov Teoremi incelenmelidir.

3.2 Teorem 3.1 (Kharitonov Teoremi)

$$p(s) = p_0 + p_1 s + \cdots + p_n s^n \tag{3.1}$$

şekildeki olan ve katsayıları $p_i \in [x_i, y_i]$ şekilde değişen bir karakteristik polinom kümesinin kararlı olup olmadığını anlamak için

$$\begin{Bmatrix} K^1(s) = x_0 + x_1 s + y_2 s^2 + y_3 s^3 + x_4 s^4 + x_5 s^5 + y_6 s^6 + \cdots \\ K^2(s) = x_0 + y_1 s + y_2 s^2 + x_3 s^3 + x_4 s^4 + y_5 s^5 + y_6 s^6 + \cdots \\ K^3(s) = y_0 + x_1 s + x_2 s^2 + y_3 s^3 + y_4 s^4 + x_5 s^5 + x_6 s^6 + \cdots \\ K^4(s) = y_0 + y_1 s + x_2 s^2 + x_3 s^3 + y_4 s^4 + y_5 s^5 + x_6 s^6 + \cdots \end{Bmatrix} \tag{3.2}$$

Şeklinde tanımlanan 4 Kharitonov polinomunun kararlı olması yeterlidir. Oluşabilecek diğer polinomlar, ifade edilen bu 4 polinomun arasında olacağından Kharitonov polinomlarının dördünün de kararlı olması durumda tüm polinom ailesi kararlı olacaktır.

3.3 Parametrik Belirsizlik İçeren Sistemler için Köşe ve Kenar Sistemlerinin Bulunması

Bu tür sistemler için öncelikle pay ver payda polinomlarının ayrı ayrı Kharitonov polinomları oluşturulur. Pay polinomu $N(s)$, payda polinomu $D(s)$ olmak üzere Kharitonov polinomları $N^1(s)$, $N^2(s)$, $N^3(s)$, $N^4(s)$, $D^1(s)$, $D^2(s)$, $D^3(s)$ ve $D^4(s)$ elde edilir. Her bir pay ve payda Kharitonov poliomunun birbiriyle eşleşmesiyle 16 adet köşe sistemi oluşur. Verilen sistemin dayanıklı kararlı olduğu bölgeyi bulmak için bu 16 sisteme bakmak yeterli olmayabilir. Bu durumda

$$\begin{Bmatrix} NS^1(s,\lambda) = (1-\lambda)N^1(s) + \lambda\, N^2(s) \\ NS^2(s,\lambda) = (1-\lambda)N^1(s) + \lambda\, N^3(s) \\ NS^3(s,\lambda) = (1-\lambda)N^2(s) + \lambda\, N^4(s) \\ NS^4(s,\lambda) = (1-\lambda)N^3(s) + \lambda\, N^4(s) \end{Bmatrix} \quad \textbf{(3.3)}$$

Şekilde pay polinomu için kenar polinomları hesaplanır. Bu kenar polinomları payda polinomları ile birer birer eşlenerek 16 adet kenar sistemi elde edilir. Burada λ değişkeni 0 ile 1 arasında değer almakta olup iki köşe arasındaki kenar üzerinde bulunan sistemlerin elde edilmesinde kullanılmaktadır.

Öncelikle λ değişkeni 0 ile 1 arasında belli adımlarla değiştirilerek bir kenar üzerindeki sistemlerin kararlı bölgeleri bulunur. Daha sonra bu bölgelerin kesişimi alınarak o kenar üzerindeki dayanıklı kararlı bölge bulunur. Bu işlem tüm kenarlara uygulanarak elde edilen tüm bölgelerin kesişimi alındığında ise parametrik belirsizlik içeren sistemi dayanıklı kararlı yapan PID kontrolör bölgesi bulunmuş olur.

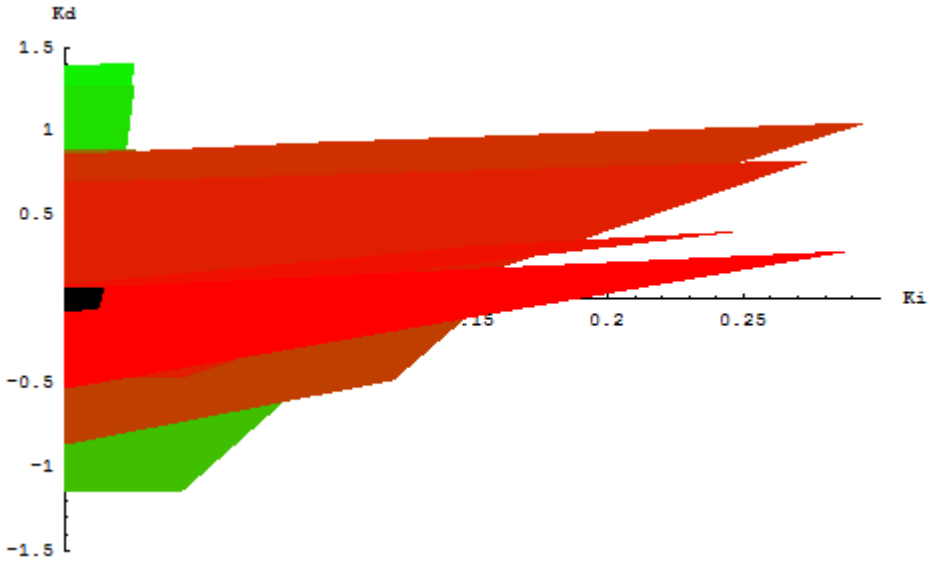
3.4 Örnek

Parametrik belirsizlik içeren sistemler için dayanıklı kararlılığı sağlayan PID kontrolörlerin bulunmasına ilişkin örnek olarak transfer fonksiyonu

$$G(s) = \frac{a_2\, s^2 + a_1\, s + a_0}{b_5\, s^5 + b_4\, s^4 + b_3\, s^3 + b_2\, s^2 + b_1\, s + b_0} \quad \textbf{(3.4)}$$

şeklinde verilen ve katsayıları $a_2 \in [1\ \ 1]$, $a_1 \in [-5\ \ -4]$, $a_0 \in [2\ \ 4]$, $b_5 \in [1\ \ 1]$, $b_4 \in [3\ \ 4]$, $b_3 \in [5\ \ 5]$, $b_2 \in [7\ \ 9]$, $b_1 \in [8\ \ 9]$, $b_0 \in [-2\ \ -1]$ aralıklarında değişmekte olsun.

Sistemin öncelikle seçilen bir K_p köşe ve kenar sistemleri hesaplanır. Kenar sistemleri için her kenar üzerinde belli sayıda sistem alınarak kararlı bölgeleri bulunur ve bu bölgelerin kesişimi alınarak o kenarın kararlı bölgesi bulunur. Daha sonra bu kenar bölgelerinin kesişimleri alınarak sistemi işlem yapılan K_p değerindeki dayanıklı kararlı bölgesi bulunmuş olur. Şekil 3.1'de $K_p = 1.05$ değeri için kenar sistemlerinin kararlı bölgeleri (yeşilden kırmızıya doğru giden renkler) ve bu bölgelerin kesişimi olan dayanıklı kararlı bölge (siyah renk) gösterilmiştir.



Şekil 3.1: Örnek sistem için K_p=1.05 değerindeki kenar sistemlerinin kararlı bölgeleri.

Bu işlem tüm K_p değerleri için uygulanarak sistemi dayanıklı kararlı yapan PID kontrolör bölgesi bulunabilir.

4. KARARLILIĞI SAĞLAYAN TÜM PID KONTROLÖRLERİN BULUNMASINA İLİŞKİN FONKSİYON KÜMESİNİN YAZILIMI

4.1 Amaç

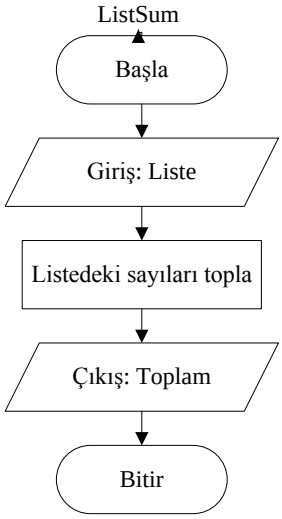
Bu bölümde, 2. ve 3. bölümde aktarılan yöntemler kullanılarak belirsizlik içeren ve içermeyen sistemler için kararlılığı sağlayan tüm PID kontrolör bölgesinin bulunmasına ilişkin Mathemtica programlama ortamı kullanılarak bir fonksiyon kümesinin geliştirilmesi anlatılacaktır. Bu küme içindeki fonksiyonların yaptıkları işler, çalışma şekilleri ve akış diyagramları verilip birbirleriyle olan bağlantıları açıklanacaktır. Son olarak da yazılan bu fonksiyonların doğruluğunu ve performansını incelemek için çeşitli örneklerle testler yapılacaktır.

4.2 Yazılan Fonksiyonlar ve İşlevleri

Kararlılığı sağlayan PID kontrolörlerin bulunmasına ilişkin yöntemlerde kullanılan benzer ve tekrarlanan işlemler için Mathematica kullanılarak çeşitli fonksiyonlar yazılmıştır. Bu fonksiyonlar arasında kararlılığı sağlayacak K_p aralığını bulan, $K_i - K_d$ düzleminde oluşan sınır doğrularını ve kararlı bölgeyi belirleyen, bu bölgelerin oluşturulmasını ve üzerinde çeşitli geometrik işlemlerin yapılmasını sağlayan, bulunan sonuçların grafiksel olarak gösterilmesini sağlayan fonksiyonlar bulunmaktadır.

4.2.1 ListSum fonksiyonu

ListSum fonksiyonu giriş olarak bir sayı dizisi alır ve bu dizi içindeki tüm sayıları toplayarak çıkış olarak toplam sonucu döndürür. Fonksiyonun akış diyagramı Şekil 4.1'de gösterilmiştir.



Şekil 4.1: ListSum fonksiyonunun akış diyagramı.

4.2.2 UnstablePoleNumber fonksiyonu

UnstablePoleNumber fonksiyonu giriş olarak bir karakteristik polinom alır. Bu fonksiyona kapalı çevrim bir sistemin karakteristik polinomu verildiğinde, bu polinomu sıfıra eşitleyerek köklerini bulur. Dolayısıyla kapalı çevrim sistemin kutuplarının yeri elde edilmiş olur. Fonksiyon bu kutupların da yerlerini inceleyerek, sağ yarı düzlemdeki kutup sayısını belirler. Böylece çıkış olarak, girilen karakteristik polinomun kararsız kutup sayısı elde edilir. Fonksiyonun akış diyagramı Şekil 4.2'de gösterilmiştir.

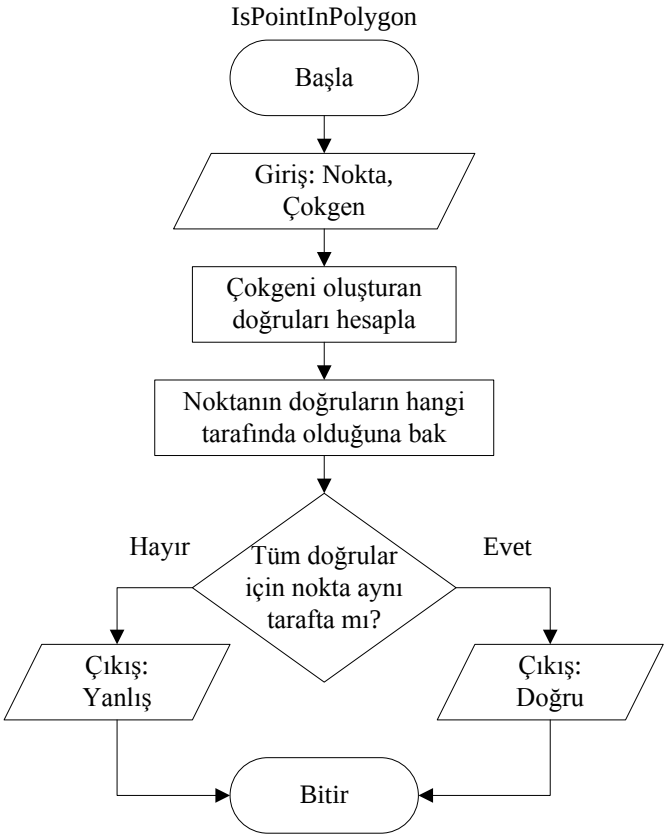


Şekil 4.2: UnstablePoleNumber fonksiyonunun akış diyagramı.

4.2.3 IsPointInPolygon fonksiyonu

IsPointInPolygon fonksiyonu giriş olarak bir nokta ve bir konveks çokgen alır. Nokta, iki boyuttaki koordinatlarını belirten bir listeden oluşur. Çokgen ise içinde birden çok nokta olan bir listeden oluşmaktadır. Ayrıca çokgenin konveks olması ve noktaların bu şartı sağlayacak şekilde listeye eklenmesi gerekmektedir.

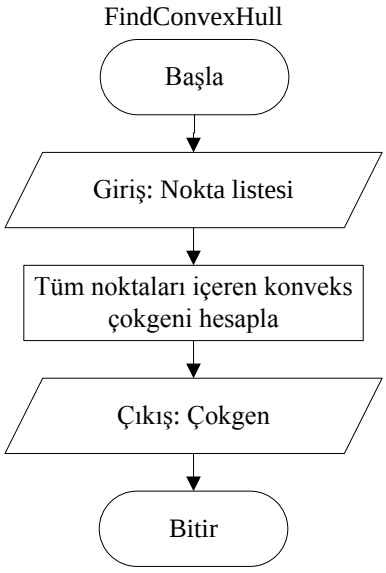
Fonksiyonun yaptığı ilk iş çokgeni oluşturan olası tüm ardışık iki noktayı seçerek bu noktalara göre doğru ifadeleri oluşturmaktır. Oluşan bu doğru ifadeleri, yönlü olarak çokgenin kenarlarını belirtmektedir. Girilen nokta, herhangi bir doğru ifadesinde yerine konulduğunda eğer sonuç sıfır çıkıyorsa, noktanın o kenarın belirttiği doğru üzerinde bulunduğu anlaşılır. Eğer sonuç sıfırdan büyük veya küçük çıkıyorsa da noktanın doğrunun yönüne göre sağda veya solda olduğu anlaşılır. Bu durum tüm kenarlar için uygulanır ve eğer nokta tüm kenarlar için ya hep sağda ya da hep solda ise noktanın çokgenin içinde olduğu anlaşılır. Eğer nokta çokgenin içindeyse fonksiyon "Doğru", değilse de "Yanlış" değerini döndürür. Fonksiyonun akış diyagramı Şekil 4.3'de gösterilmiştir.



Şekil 4.3: IsPointInPolygon fonksiyonunun akış diyagramı.

4.2.4 FindConvexHull fonksiyonu

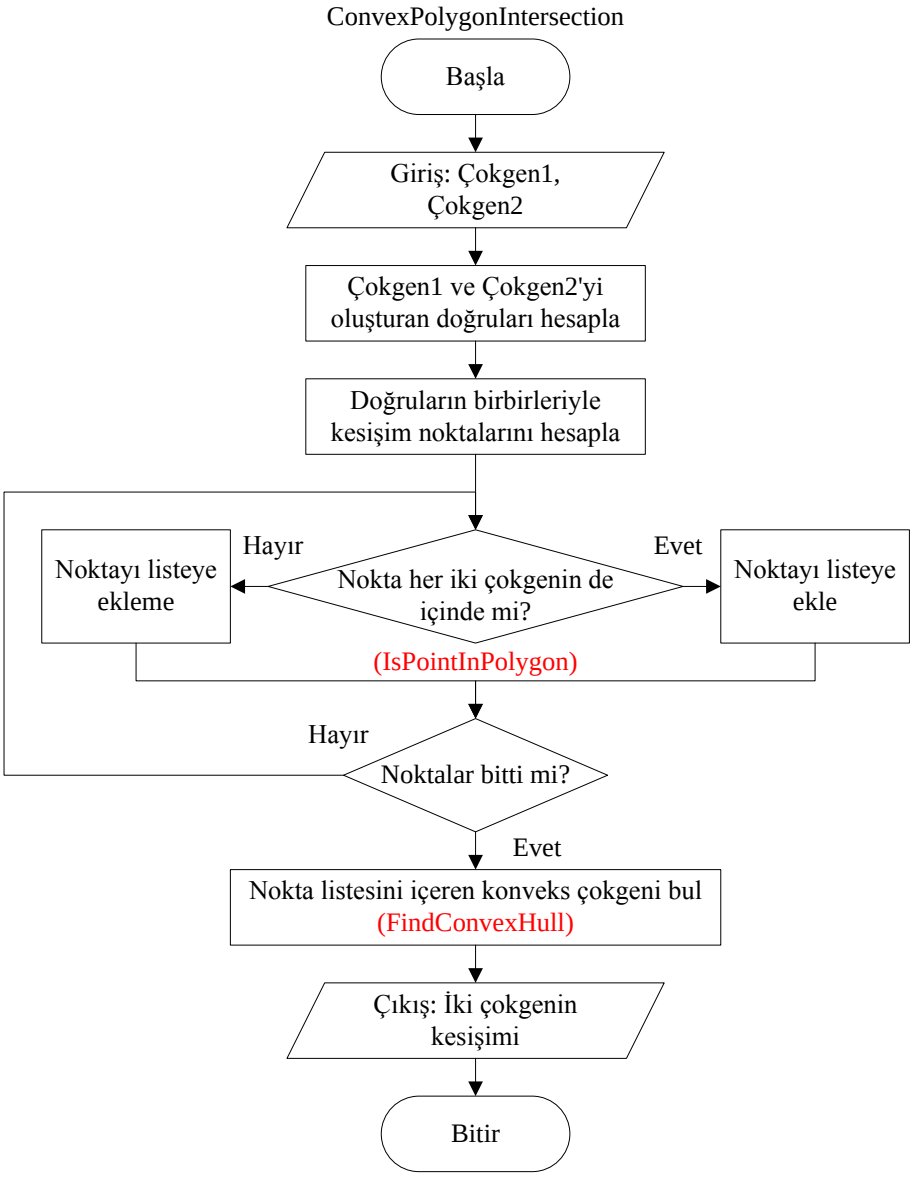
FindConvexHull fonksiyonu giriş olarak bir nokta listesi alır. Bu fonksiyon, Mathematica programının içinde bulunan ConvexHull fonksiyonunu kullanarak girilen nokta listesini kapsayan konveks çokgenin noktalarını, girilen nokta listesindeki sıra numaralarını elde eder. Daha sonra bu sıra numaralarında bulunan noktaları alarak konveks çokgenin çıkışına gönderir. Başka bir deyişle bu fonksiyonu ConvexHull fonksiyonunun istenilen bir şekilde çıkış vermesini sağlamaktadır. Fonksiyonun akış diyagramı Şekil 4.4'de gösterilmiştir.



Şekil 4.4: FindConvexHull fonksiyonunun akış diyagramı.

4.2.5 ConvexPolygonIntersection fonksiyonu

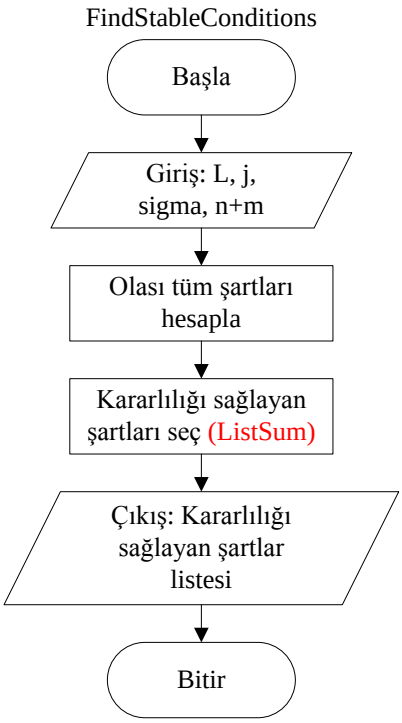
ConvexPolygonIntersection fonksiyonu giriş olarak iki adet dışbükey çokgen alır. Burada yine çokgenler nokta listeleri şeklinde girilir ve konveks olmalıdırlar. Fonksiyonda öncelikle iki çokgeni de oluşturan kenarların doğru denklemleri bulunur. Daha sonra oluşan tüm doğru denklemleri ikişerli çözülerek kesişim noktaları elde edilir. Tüm olası kombinasyonlarla bütün kesişim noktaları bulunduktan sonra tüm noktalar için IsPointInPolygon fonksiyonu ile iki çokgenin de içinde mi diye bakılır. İki çokgenin de içinde olma şartını sağlayan noktalar bir listede tutulur. Tüm noktalar incelendikten sonra oluşturulan liste FindConvexHull fonksiyonuna gönderilerek bir konveks çokgen elde edilir. Bu çokgen, iki çokgenin kesişimi olarak fonksiyon çıkışına verilir. Fonksiyonun akış diyagramı Şekil 4.5'de gösterilmiştir.



Şekil 4.5: ConvexPolygonIntersection fonksiyonunun akış diyagramı.

4.2.6 FindStableConditions fonksiyonu

FindStableConditions fonksiyonu Bölüm 2.4'de anlatılan yöntemi uygulamaktadır. Fonksiyon giriş olarak sistemin tekil frekans sayısı (L), $v_{odd}(\omega, {K_p}^*)$ polinomunun $\omega = 0^+$ için işaret değeri (j), $\sigma(v)$ ifadesinin değeri ve n ile m değerlerinin toplamını alır. Fonksiyon içinde öncelikle $n+m$ ve L değerine göre (2.17) veya (2.18) ifadesi elde edilir. Daha sonra bu ifadedeki i'ler yerine 1 ve -1 yazılarak olası tüm diziler elde edilir. ListSum fonksiyonu kullanılarak tüm bu dizilerin içindeki sayılar toplanarak sonucun $\sigma(v)$ ifadesinin değerine eşit olduğu görülen diziler bir listeye alınır. Bu diziler $K_i - K_d$ düzlemindeki sınır doğrularının hangi tarafının alınacağını göstermektedir ve bu düzlemdeki kararlı bölge koşulları olmak üzere fonksiyon çıkışına gönderilirler. Fonksiyonun akış diyagramı Şekil 4.6'da gösterilmiştir.



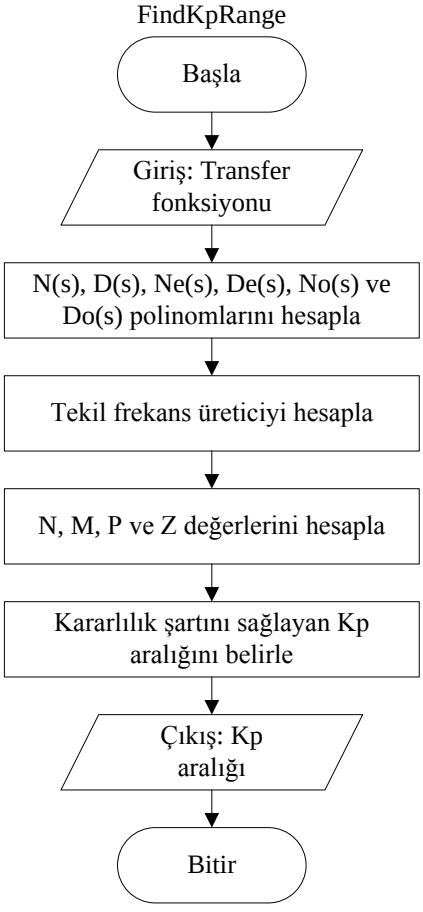
Şekil 4.6: FindStableConditions fonksiyonunun akış diyagramı.

4.2.7 FindKpRange fonksiyonu

FindKpRange fonksiyonunun amacı Bölüm 2.3'deki yöntemi kullanarak kararlılığı sağlayan PID kontrolörler için K_p kazanç aralığını belirlemektir. Fonksiyon giriş olarak bir sistem transfer fonksiyonu alır. Girilen sistem pay ve payda polinomlarını ($N(s)$ ve $D(s)$) ayrılır ve bu polinomların da tek ve çift kısımları ($N_o(s)$, $N_e(s)$, $D_o(s)$ ve $N_e(s)$) hesaplanır. Daha sonra ise (2.9) ifadesine göre tekil frekanslar ile K_p kazancı arasındaki bağıntı kurulur.

Bu işlemden sonra yöntemde kullanılan kapalı çevrim sistemin karakteristik polinomunun derecesi (N), verilen sistemin pay polinomunun derecesi (M) ve verilen sistemin sağ yarı düzlem sıfır sayısı (P) hesaplanır. Daha sonra ise yönteme göre sistemin kararlı olması için gereken en az tekil frekans sayısı (Z) hesaplanır.

Gerekli tüm değerler hesaplandıktan sonra hangi K_p değerleri için kaç tekil frekans oluştuğu bulunarak tekil frekans sayısının Z değerinden fazla olduğu aralık seçilerek fonksiyon çıktısı olarak verilir. Fonksiyonun akış diyagramı Şekil 4.7'de verilmiştir.



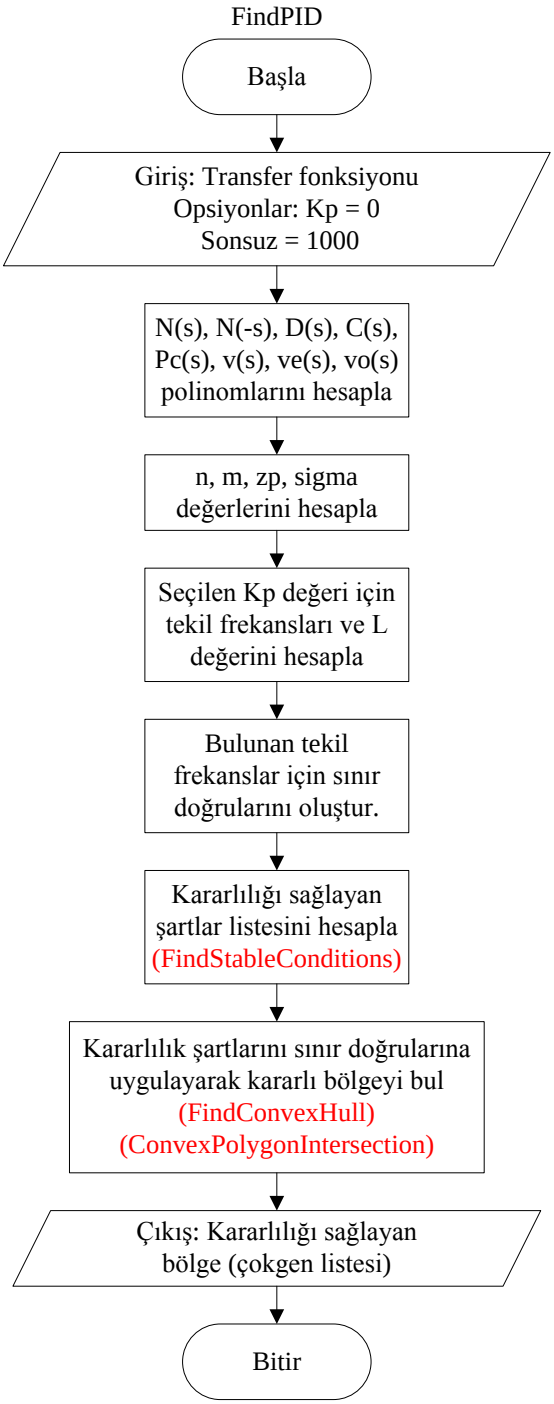
Şekil 4.7: FindKpRange fonksiyonunun akış diyagramı.

4.2.8 FindPID fonksiyonu

FindPID fonksiyonu giriş olarak bir sistem transfer fonksiyonu alır ve bu sistem için $K_p = 0$ durumunda $K_i - K_d$ düzlemindeki kararlı bölgeyi bulur. İsteğe bağlı olarak K_p değeri de kullanıcı tarafından girilebilir ve bu fonksiyon farklı K_p değerleri için kullanılarak verilen sistemi kararlı yapan tüm kararlı PID kontrolörler bulunabilir. Yine isteğe bağlı olarak açık bölgelerin ifade edilebilmesi için sonsuz ifadesinin ne kadar uzakta olduğu girilebilir. Eğer bir değer girilmezse bu değer 1000 olarak alınır.

Fonksiyon öncelikle verilen sistemin pay ve payda polinomlarını (($N(s)$ ve $D(s)$) hesaplar. Daha sonra girilen K_p değeri için PID kontrolörün sembolik transfer fonksiyonunu hesaplar. Sistem ile kontrolör kapalı çevrime alınarak kapalı çevrim sistemin transfer fonksiyonu ve karakteristik polinomu elde edilir. Bu işlemden sonra Bölüm 2.4'de aktarılan yöntem için gerekli olan $N(-s)$ ve $v(s)$ polinomları hesaplanır ve $v(s)$ polinomu $v_{odd}(s)$ ve $v_{even}(s)$ olmak üzere tek ve çift kısımlarına ayrılır. Daha sonra ise sistemin pay ve payda polinomları derecelerinden (n ile m) ve sistemin sağ yarı düzlem sıfır sayısından yararlanarak $\sigma(v)$ değeri bulunur. $v_{odd}(s)$ polinomundan yararlanarak j değeri ve tüm tekil frekansları hesaplanır. Tekil frekans sayısına bakılarak L değeri bulunur. Bulunan tekil frekanslar $v_{even}(s)$ polinomunda yerine konularak $K_i - K_d$ düzlemindeki sınır doğruları elde edilir.

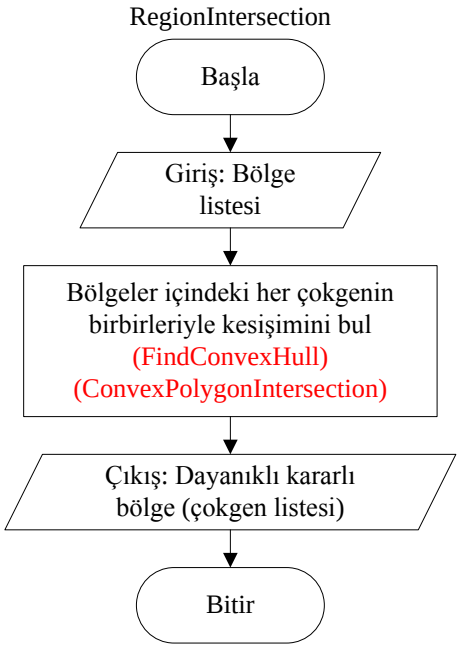
Sınır doğruları elde edildikten sonra FindStableConditions fonksiyonuna bulunan L, j, $\sigma(v)$, n ve m değerleri girilerek sınır doğrularının hangi taraflarının alınması sonucu kararlı bölge oluşabileceği elde edilir. Daha sonra bu şartlar sınır doğrularına uygulanarak yarı düzlemler oluşturulur. ConvexPolygonIntersection fonksiyonu ile bu yarı düzlemlerin kesişimi alınır ve bu kesişim sonucu oluşan çokgen bir listede tutulur. FindStableConditions fonksiyonundan gelen tüm şartlar için bu işlemler yapılarak oluşan çokgenler aynı listeye gönderilir. Fonksiyon çıktı olarak da çokgenlerin tutulduğu bu listeyi verir. Böylece girilen bir sistem ve seçilen bir K_p değeri için $K_i - K_d$ düzlemindeki kararlı bölge bulunmuş olur. Fonksiyonun akış diyagramı Şekil 4.8'de gösterilmiştir.



Şekil 4.8: FindPID fonksiyonunun akış diyagramı.

4.2.9 RegionIntersection fonksiyonu

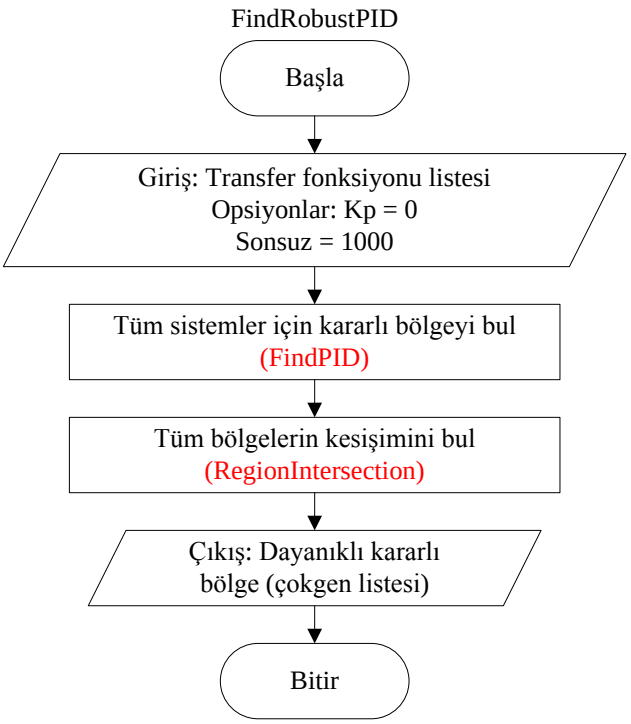
RegionIntersection fonksiyonu giriş olarak bir bölge listesi alır. Bu bölgelerin hepsi FindPID fonksiyonunun çıkışı gibi bir veya birden fazla çokgenden oluşabilir. Fonksiyon, öncelikle listedeki ilk iki bölgeyi alır ve içindeki çokgenlerin kesişimini ConvexPolygonIntersection fonksiyonu ile bulur. Eğer listede daha fazla bölge varsa bu bölgeleri de tek tek alarak önceki hesaplanan kesişim bölgesi ile kesişimini alır ve böylece tüm bölgeler bittiğinde girilen bölge listesinin tüm elemanlarının kesişimi alınmış olur. Bu işlem sistemlerin belirsizlik içermesi durumunda dayanıklı kararlı bölgeyi bulmak için kullanılacaktır. Fonksiyon çıktı olarak yine bir çokgen listesi verir. Fonksiyonun akış diyagramı Şekil 4.9'da gösterilmiştir.



Şekil 4.9: RegionIntersection fonksiyonunun akış diyagramı.

4.2.10 FindRobustPID fonksiyonu

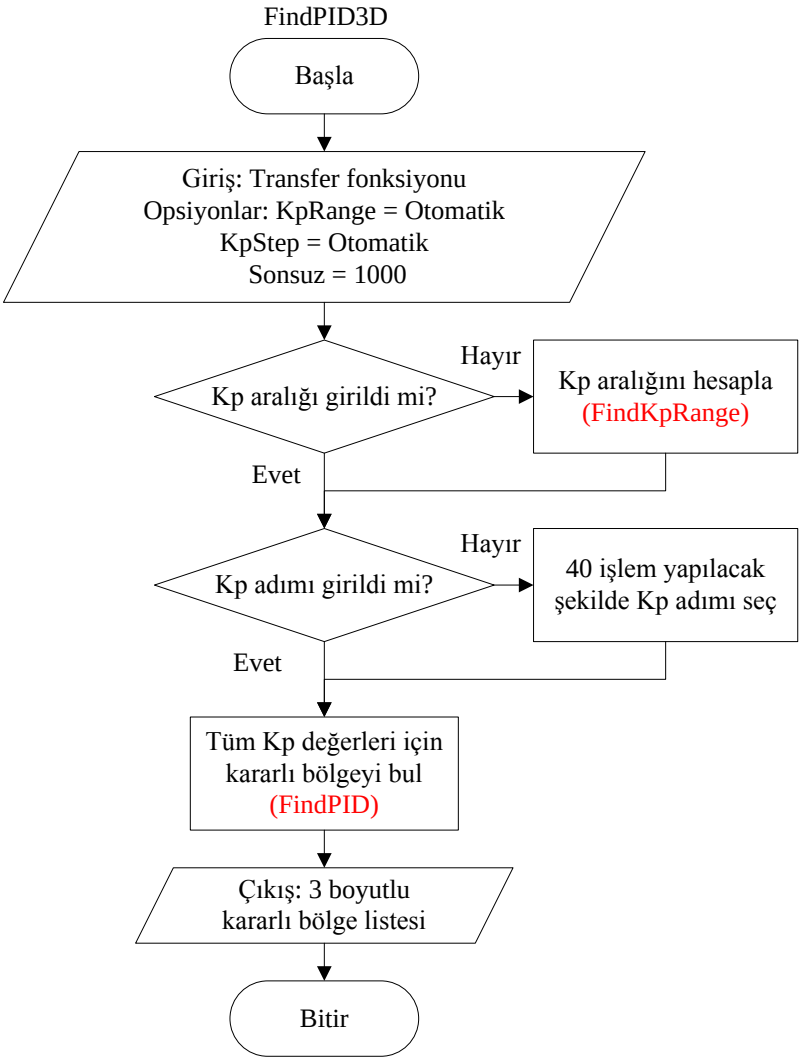
FindRobustPID fonksiyonu giriş olarak transfer fonksiyonlarından oluşan bir liste alır. FindPID fonksiyonundaki gibi K_p ve sonsuz değerleri isteğe bağlı olarak kullanıcı tarafından girilebilir. Fonksiyon, FindPID fonksiyonunu kullanarak girilen listedeki sistemler için K_p kazancını istenilen şekilde seçip $K_i - K_d$ düzlemindeki kararlı bölgeyi bulur. Daha sonra bu bölgelerin kesişimini RegionIntersection fonksiyonu ile hesaplayarak, girilen tüm sistemler için dayanıklı kararlı bölgeyi verir. Bu fonksiyona parametrik belirsizlik içeren bir sistemin Kharitonov teoremi ile elde edilen köşe veya kenar sistemlerinin girilmesiyle istenen K_p sistemin dayanıklı kararlı olduğu $K_i - K_d$ bölgesi bulunabilir. Fonksiyonun akış diyagramı Şekil 4.10'da verilmiştir.



Şekil 4.10: FindRobustPID fonksiyonunun akış diyagramı.

4.2.11 FindPID3D fonksiyonu

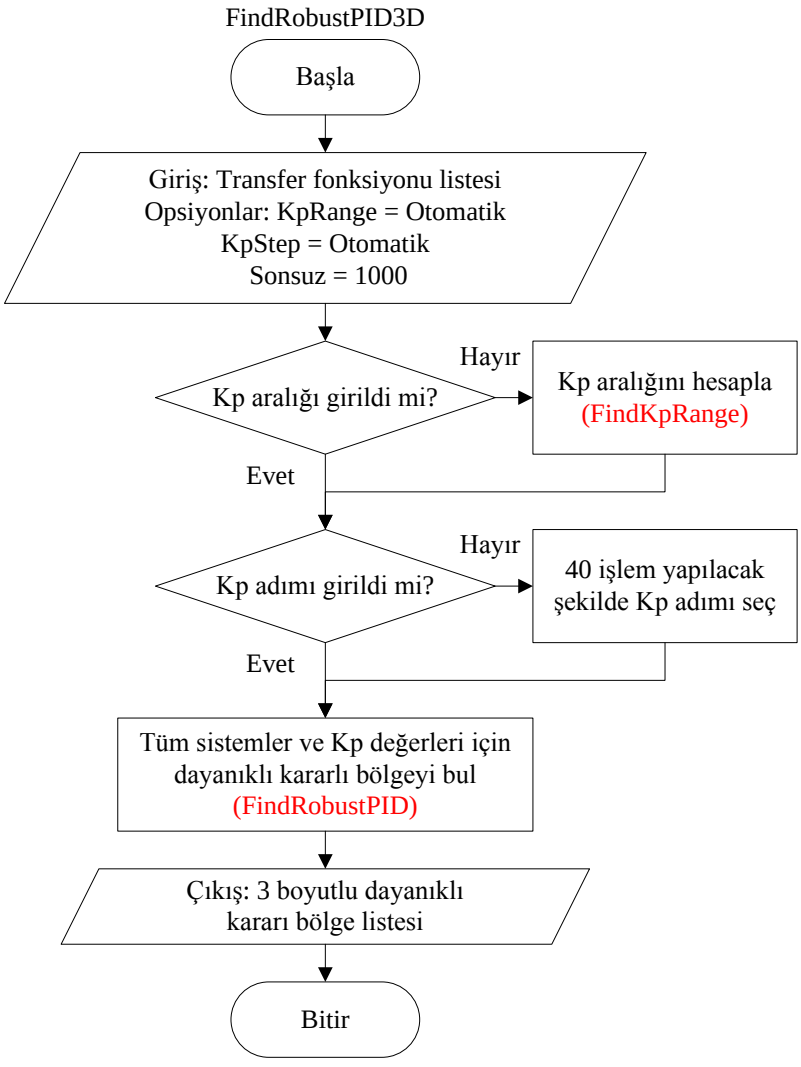
FindPID3D fonksiyonu, FindPID fonksiyonunu kullanarak girilen bir sistem için bir K_p aralığında sistemi kararlı yapan PID katsayılarının oluşturduğu bölgeyi bulur. K_p aralığı kullanıcı tarafından da girilebileceği gibi otomatik olarak da hesaplanabilir. Eğer kullanıcı bir K_p aralığı girmezse Bölüm 2.3'de aktarılan yönteme göre verilen sistemi kararlı yapabilecek K_p aralığı FindKpRange fonksiyonu ile hesaplanır. Ayrıca kullanıcı tarafından K_p aralığında kaç birim adımla gidilmesi istendiği de girilebilir. Eğer bunun için de bir değer girilmezse işlem yapılacak olan K_p aralığından eşit aralıkta 40 değer alınarak bu değerler için $K_i - K_d$ düzlemindeki kararlı bölgeler hesaplanır. Böylece tüm K_p değerleri için kararlı bölgeler bulunur ve bu bölgeler 3 boyutlu bir bölge listesi şekilde fonksiyon çıktısı olarak verilir. Fonksiyonun akış diyagramı Şekil 4.11'de gösterilmiştir.



Şekil 4.11: FindPID3D fonksiyonunun akış diyagramı.

4.2.12 FindRobustPID3D fonksiyonu

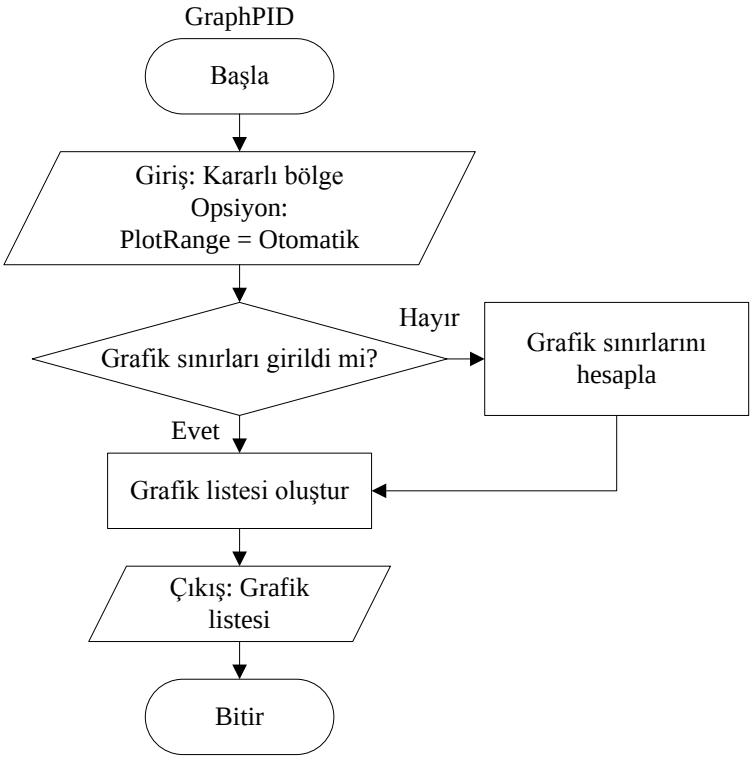
FindRobustPID3D fonksiyonu, FindPID3D fonksiyonunun yaptığı işi FindRobustPID fonksiyonunu kullanarak parametre belirsizliği içeren sistemler için yapmaktadır. Fonksiyonu giriş olarak bir transfer fonksiyonu listesi alır. Yine isteğe bağlı olarak K_p aralığı ve K_p adımı girilebilir. Eğer bu değerler girilmezse yine yanı şekilde otomatik olarak hesaplanır. Fonksiyon çıkış olarak girilen transfer fonksiyonu listesi için 3 boyutlu dayanıklı kararlı bölge listesini verir. Fonksiyonun akış diyagramı Şekil 4.12'de verilmiştir.



Şekil 4.12: FindRobustPID3D fonksiyonunun akış diyagramı.

4.2.13 GraphPID ve GraphPID3D fonksiyonları

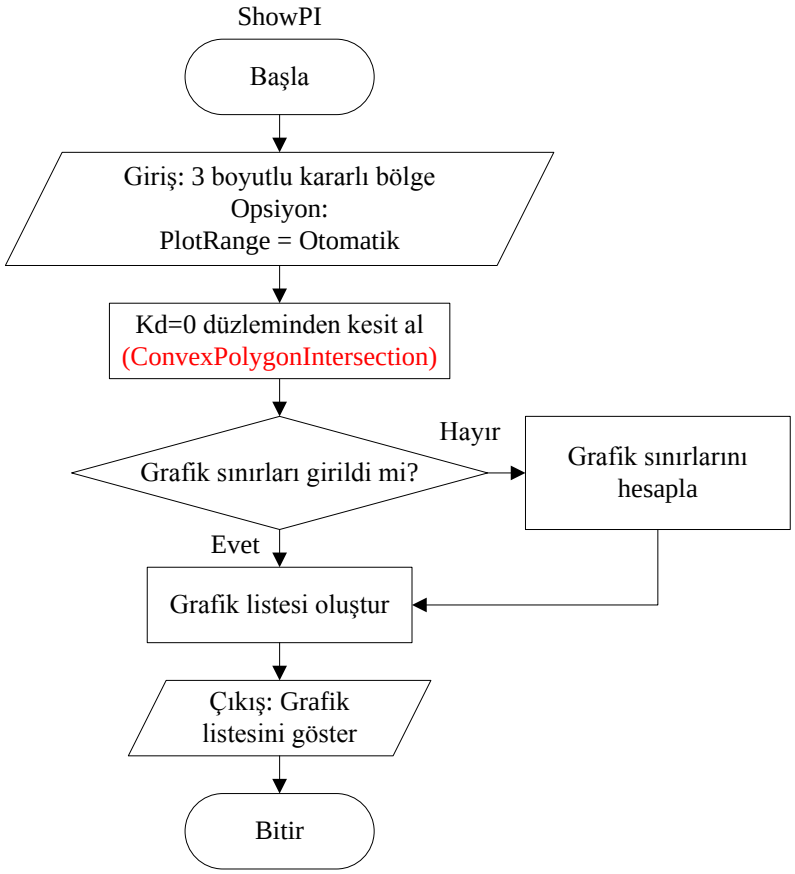
FindPID, FindRobustPID, FindPID3D ve FindRobustPID3D fonksiyonlarının çıktıları birbirleriyle veya başka işlemlerde kullanabilmek için liste şeklinde verilmektedir. Eğer bu fonksiyonların çıktıları grafiksel olarak görülmek isteniyorsa FindPID ve FindRobustPID fonksiyonları için GraphPID, FindPID3D ve FindRobustPID3D fonksiyonları için de GraphPID3D fonksiyonu kullanılarak bu listeler grafik elemanlarına dönüştürülebilir. Daha sonra bu grafik elemanları da Mathematica programının kendi fonksiyonu olan Show fonksiyonu ile gösterilebilir. GraphPID ve GraphPID3D fonksyionlarının yaptığı temel işlev, giriş olarak aldıkları listenin içindeki çokgenlerin grafiksel hale getirilmesidir. Fonksiyona isteğe bağlı olarak grafik sınırları girilebilir. Eğer bir sınır girilmezse sınırlar otomatik olarak hesaplanır. GraphPID fonksiyonunun akış diyagramı Şekil 4.13'de gösterilmiştir. GraphPID3D fonksiyonu da aynı mantıkla çalışmaktadır.



Şekil 4.13: GraphPID fonksiyonunun akış diyagramı.

4.2.14 ShowPI ve ShowPD fonksiyonları

FindPID3D ve FindRobustPID3D fonksiyonlarından elde edilen kararlılığı sağlayan 3 boyutlu PID bölgesinden istenilirse 2 boyutlu PI ve PD kontrolör bölgesinin incelenmesi için ShowPI ve ShowPD fonksiyonları yazılmıştır. ShowPI fonksiyonu 3 boyutlu PID bölgesini $K_d = 0$ düzleminden keserek elde ettiği bölgeyi grafiksel olarak gösterir. Böylece kararlılığı sağlayan PI kontrolör bölgesi bulunmuş olur. Aynı şekilde ShowPD fonksiyonu da 3 boyutlu PID bölgesini $K_i = 0$ bölgesinden keserek kararlı PD kontrolör bölgesini gösterir. ShowPI fonksiyonunun akış diyagramı Şekil 4.14'de gösterilmiştir. ShowPD fonksiyonu da aynı mantıkla çalışmaktadır.

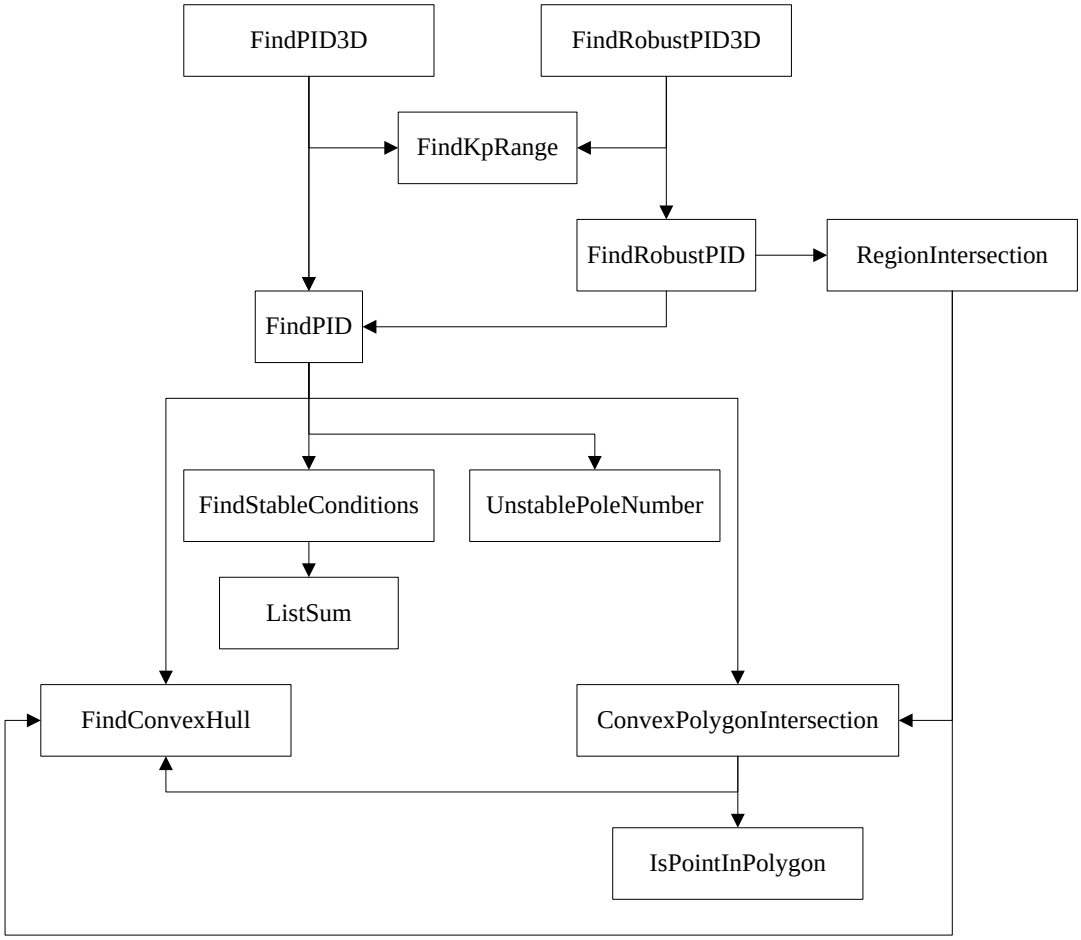


Şekil 4.14: ShowPI fonksiyonunun akış diyagramı.

4.3 Fonksiyonların Birbirleriyle Bağlantıları

Yazılan fonksiyon kümesindeki fonksiyonların birçoğunun birbirleriyle bağlantıları vardır. Bu bağlantılar Şekil 4.15'te gösterilmiştir. UnstablePoleNumber, ListSum, IsPointInPolygon, FindConvexHull ve FindKpRange fonksiyonları başka fonksiyonları kullanmadan tek başlarına çalışabilmektedirler. FindStableConditions fonksiyonu bir sayı dizisinin toplamını bulmak için ListSum fonksiyonunu kullanmaktadır. ConvexPolygonIntersection fonksiyonu iki çokgenin kesişimini alırken bulduğu noktaların çokgenlerin içinde olup olmadığına bakmak için IsPointInPolygon fonksiyonunu ve seçilen noktaları kapsayan konveks çokgeni bulmak için de FindConvexHull fonksiyonunu kullanmaktadır. FindPID fonksiyonu ise sınır doğrularının kararlılığı sağlayan taraflarını seçmek için FindStableConditions fonksiyonunu, kararlı bölgeyi kontrol etmek için UnstablePoleNumber fonksiyonunu ve kararlı bölgeyi oluşturmak için de FindConvexHull ile ConvexPolygonIntersection fonksiyonlarını kullanmaktadır. RegionIntersection fonksiyonu çokgenlerden oluşan bölgelerin kesişimini bulmak için ConvexPolygonIntersection ve FindConvexHull fonksiyonlarından yararlanır. FindRobustPID fonksiyonu sistemleri kararlı yapan PID kontrolör bölgesini bulmak için FindPID fonksiyonunu ve tüm sistemler için bulunan bölgelerin kesişimini almak için de RegionIntersection fonksiyonunu kullanır. FindPID3D fonksiyonu işlem yapacağı K_p aralığını bulmak için FindKpRange fonksiyonunu ve bu aralıktan seçilen K_p değerleri için kararlı bölgeleri bulmak için de FindPID fonksiyonunu kullanır. Aynı şekilde FindRobustPID3D fonksiyonu da FindKpRange ve FindRobustPID fonksiyonlarını kullanır.

Bu fonksiyonlar dışında FindPID, FindRobustPID, FindPID3D ve FindRobustPID3D fonksiyonlarının çıktılarını daha görsel bir şekilde ifade etmek için GraphPID, GraphPID3D, ShowPI ve ShowPD fonksiyonları kullanılır. FindPID, FindRobustPID, FindPID3D ve FindRobustPID3D fonksiyonlarının çıktıları bu fonksiyonlara giriş olarak uygulanır.



Şekil 4.15: Fonksiyonların birbirleriyle bağlantıları.

4.4 Fonksiyon Kümesinin Doğruluk ve Performans Testleri

Bu bölümde yazılan fonksiyon kümesinin kullanımına ilişkin örnekler verilerek yapılan örnekler için fonksiyonları doğruluğu ve performansı incelecektir. Örnekler için kullanılan kodlar EK A.1'de verilmiştir.

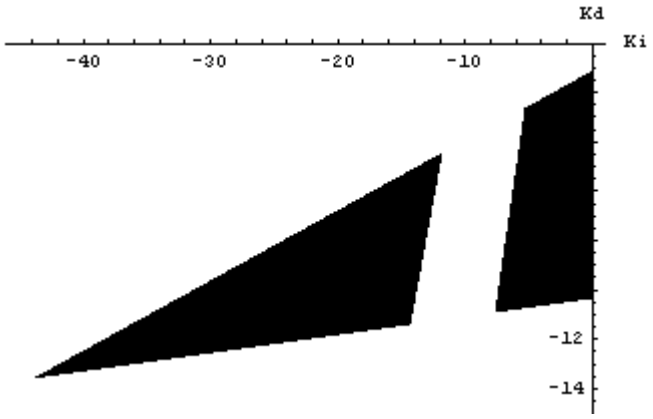
Transfer fonksiyonu

$$G(s) = \frac{s^3 - 2s^2 - s - 1}{s^6 + 2s^5 + 32s^4 + 26s^3 + 65s^2 - 8s + 1} \quad \textbf{(4.1)}$$

şekilde olan bir sistem için yazılan fonksiyonlar kullanılarak elde edilen sonuçlar aşağıda verilmiştir.

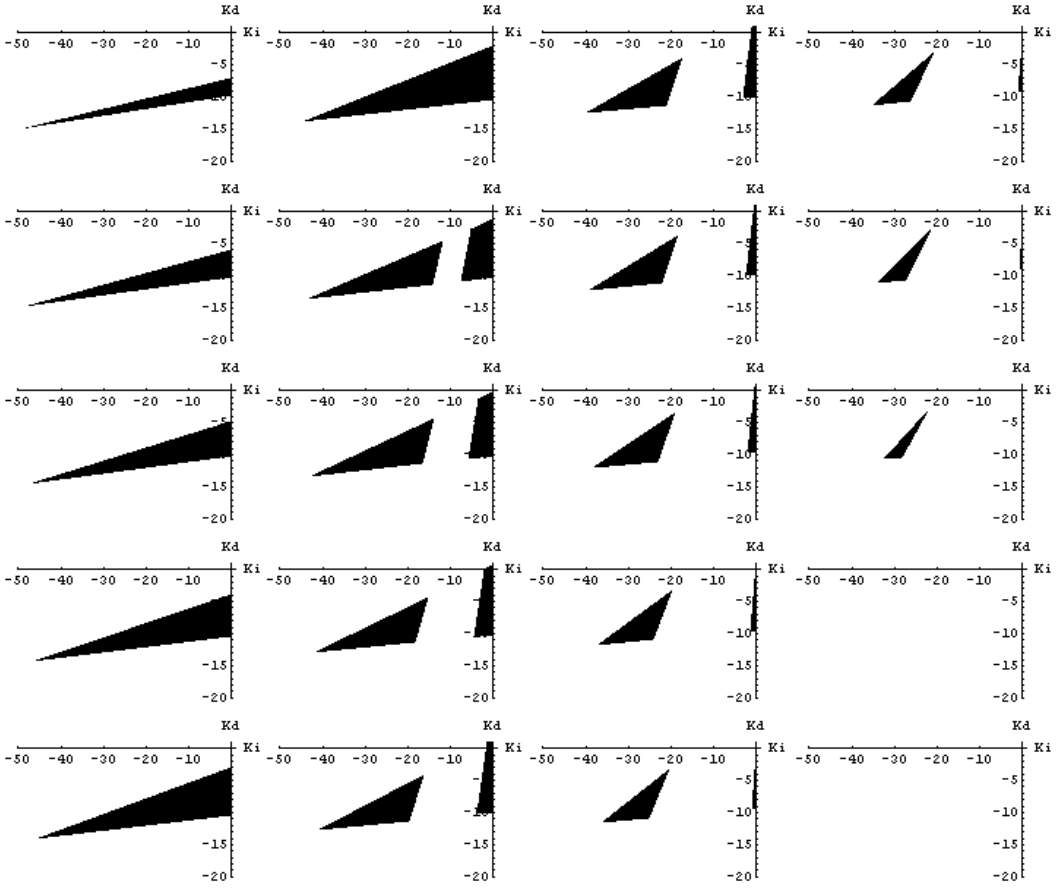
FindKpRange fonksiyonu ile sistemi kararlı yapabilecek K_p aralığı $[-24{,}7513 \;\; 1]$ olarak bulunmuştur. Bu fonksiyonun işletilmesi 0,484 saniye sürmüştür.

Bu aralıktan seçilen $K_p = -18$ değeri için FindPID fonksiyonu ile kararlı bölge 0,453 saniyede bulunmuştur. Bulunan bu bölgenin GraphPID fonksiyonu ile gösterilmesi 0,015 saniye sürmüş olup elde edilen sonuç Şekil 4.16'da gösterilmiştir.



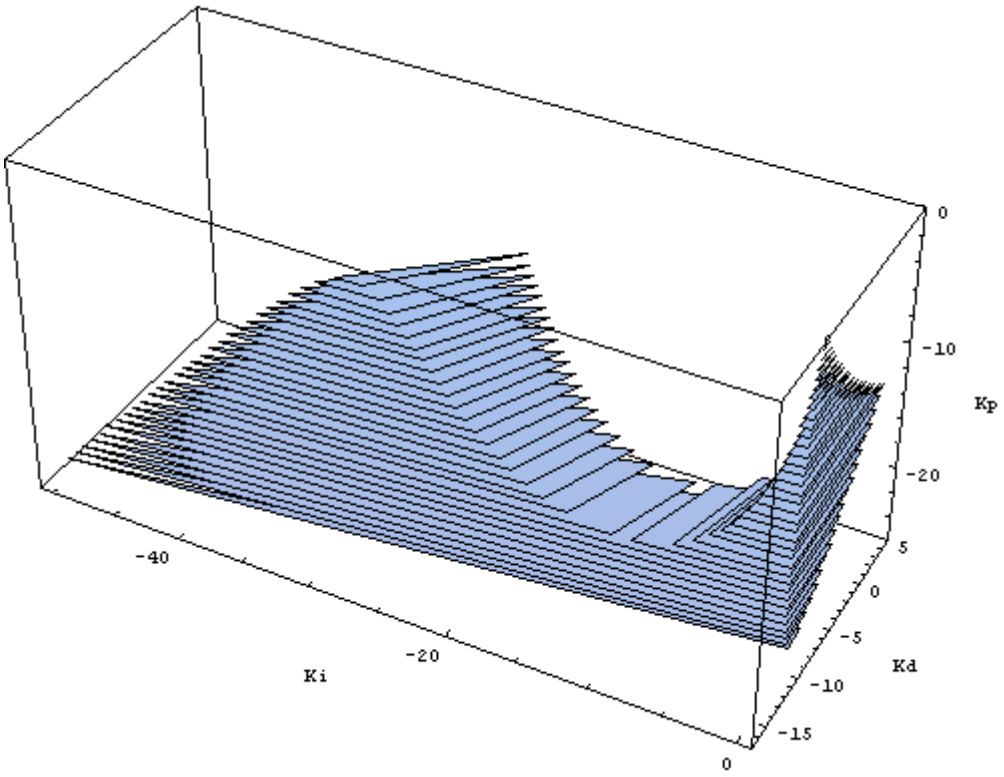
Şekil 4.16: Örnek sistem ve K_p=-18 değeri için K_i-K_d düzlemindeki kararlı bölge.

Bu işlemden sonra bulunan K_p aralığından 20 farklı örnek alınarak FindPID fonksiyonu ile kararlı bölgeler hesaplanmıştır. Bu işlem toplam 5,39 saniye sürmüştür ve elde edilen sonuçlar Şekil 4.17'de verilmiştir.



Şekil 4.17: Örnek sistem için farklı K_p değerlerinde kararlı bölgeler.

Son olarak tüm bu işlemelerin hepsini otomatik olarak yapan FindPID3D fonksiyonu kullanılmıştır ve sonuç GraphPID3D fonksiyonu ile gösterilmiştir. işlem toplam olarak 10,016 saniye sürmüştür ve işlemin sonucu Şekil 4.18'de gösterilmiştir.



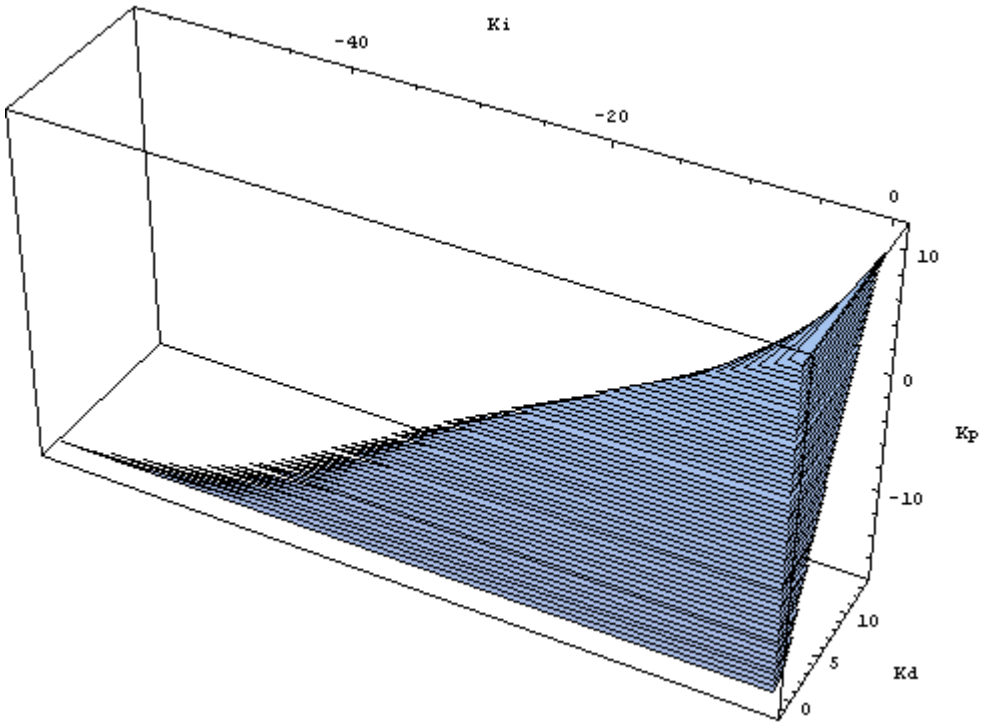
Şekil 4.18: Örnek sistem için 3 boyutlu kararlı PID kontrolör bölgesi.

Bulunan bu bölgeden rastgele noktalar seçilip kapalı çevrim sistemin karakteristik polinomunda yerine konulduğunda kapalı çevrim sistem kutuplarının hep kararlı bölgede olduğu görülmüştür.

İkinci bir örnek için ise transfer fonksiyonu

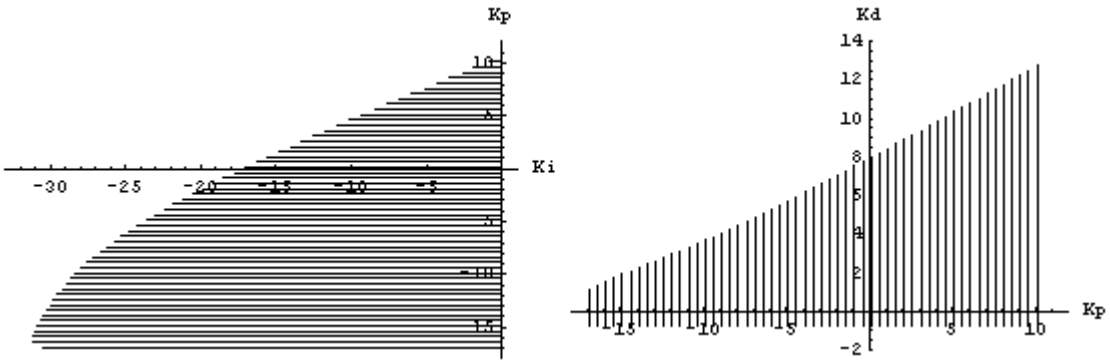
$$G(s) = \frac{s^3 + 4s^2 + 13s - 50}{s^4 + 21s^3 + 161s^2 + 511s + 530} \qquad \textbf{(4.2)}$$

şekilde bir sistem için FindPID3D fonksiyonu kullanılmıştır ve 4,282 saniyede Şekil 4.19'da gösterilen sonuç elde edilmiştir.



Şekil 4.19: Örnek sistem için 3 boyutlu kararlı PID kontrolör bölgesi.

Elde edilen bu PID kontrolör bölgesinden PI ve PD kontrolör bölgesine geçmek için de ShowPI ve ShowPD fonksiyonları kullanılabilir. Fonksiyonların işletilme süreleri sırasıyla 0.422 ve 0.437 saniye sürmüştür ve çıktıları Şekil 4.20'de gösterilmiştir.



Şekil 4.20: Örnek sistem için PI ve PD kontrolör bölgeleri.

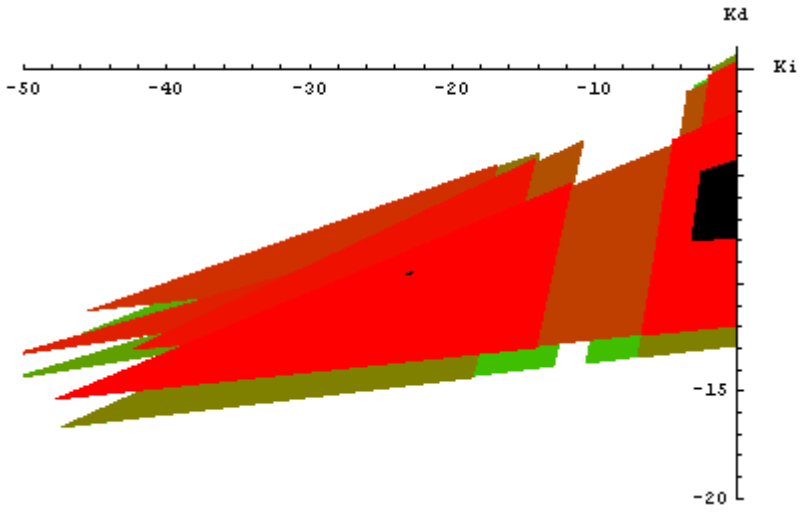
Yine bu sistem için de bulunan kararlı bölgelerden noktalar seçilip tüm noktalar için sistemin kararlı olduğu görülmüştür.

Son olarak parametrik belirsizlik içeren bir sistem için fonksiyonlar denenmiştir. Transfer fonksiyonu

$$G(s) = \frac{a_3\, s^3 + a_2\, s^2 + a_1\, s + a_0}{b_6\, s^6 + b_5\, s^5 + b_4\, s^4 + b_3\, s^3 + b_2\, s^2 + b_1\, s + b_0} \qquad \textbf{(4.3)}$$

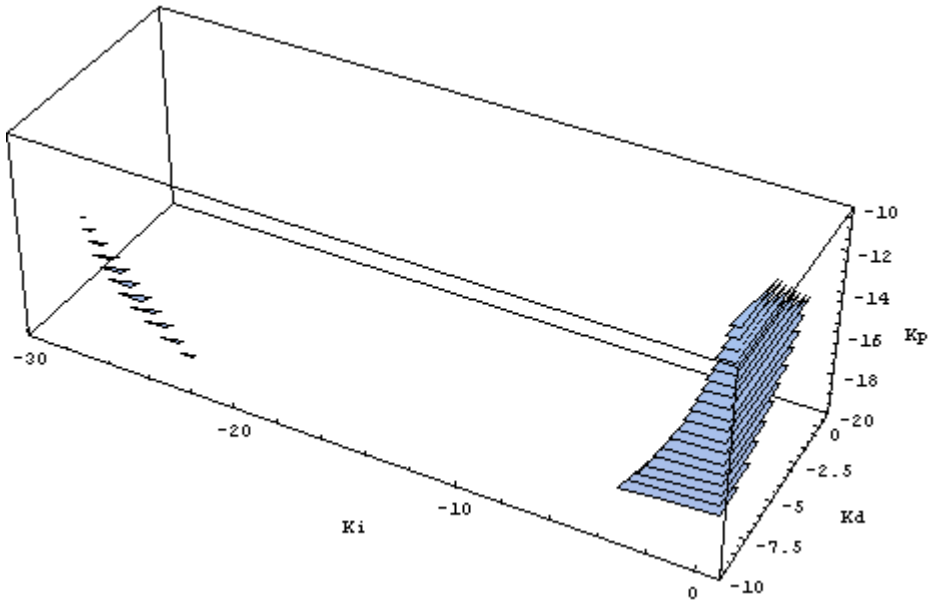
şeklinde olan ve katsayıları $a_3 \in [1\ \ 1]$, $a_2 \in [-2{,}4\ \ -2]$, $a_1 \in [-0{,}9\ \ -1]$, $a_0 \in [-1{,}1\ \ -1]$, $b_6 \in [1\ \ 1]$, $b_5 \in [2\ \ 2{,}5]$, $b_4 \in [30\ \ 34]$, $b_3 \in [25\ \ 27]$, $b_2 \in [62\ \ 68]$, $b_1 \in [-10\ \ -8]$, $b_0 \in [1\ \ 1]$ aralıklarında değişen parametrik belirsizliği olan bir sistem için Bölüm 3.3'teki yöntem uygulanmıştır. Öncelikle Kharitanov teoremi kullanılarak 16 adet köşe sistemi bulunmuştur. Daha sonra ise bu köşe sistemlerinden yola çıkarak 16 adet kenar sistemi bulunmuştur. Her bir kenar üzerinde 10 farklı örnek alınarak o kenar üzerindeki dayanıklı kararlı bölge bulunmuştur. Bu işlem tüm kenarlara uygulandıktan sonra tüm kenarlar için dayanıklı kararlı bölgelerin kesişimi alınarak belirsiz sistemi dayanıklı kararlı yapan PID kontrolör bölgesi bulunmuştur.

$K_p = -18$ değeri için kenar sistemlerinin kararlı bölgesinin FindPID fonksiyonu ile bulunması 6,015 saniye, bulunan bölgelerin RegionIntersection fonksiyonu ile kesişimi alınması 0,844 saniye ve elde edilin bölgelerin GraphPID fonksiyonu ile gösterilmesi 0,032 saniye sürmüştür. Elde edilen sonuç Şekil 4.21'de gösterilmiştir. Burada yeşilden kırmızıya doğru giden renkler kenar sistemlerin kararlı olduğu bölgeleri, siyah renk ise bu bölgelerin kesişimini göstermektedir.



Şekil 4.21: Örnek sistem için kenar sistemlerinin kararlı olduğu bölgeler.

Bu işlemlerin hepsi birden FindRobustPID fonksiyonu ile de yapılabilmektedir. Bu fonksiyonun kullanılması durumunda 6,849 saniye sonunda yine aynı sonuç alındığı görülmüştür. Son olarak tüm K_p değerleri için dayanıklı kararlı bölge FindRobustPID3D fonksiyonu ile hesaplanmıştır ve 182,969 saniye sonunda Şekil 4.22'deki sonuç elde edilmiştir.



Şekil 4.22: Örnek sistem için dayanıklı kararlı PID kontrolör bölgesi.

Bu işlemin bu kadar uzun sürmesinin nedeni FindPID fonksiyonunun birçok kez kullanılmasıdır. FindPID fonksiyonu ortalama olarak 0,5 saniyede sonuç verebilmektedir. Parametre belirsizliği olan bir sistem için 20 farklı K_p değeri için işlem yapılacaksa ve her K_p değeri için 16 kenar sistemi için FindPID fonksiyonu kullanılacaksa sonucun elde edilme süresinin çarpımsal olarak artacağı görülmektedir. Ayrıca kenar sistemler üzerinde seçilen örnek sayısının da sonucu elde etme süresine etki etmektedir.

Bulunan sonuç için bölgelerden çeşitli noktalar seçilerek belirsiz sisteme uygulanmıştır ve her kenar sistem için bu noktaların kararlı olduğu görülmüştür.

5. SONUÇLAR

Yapılan bu çalışma sonuncunda verilen ölü zamansız nedensel bir sistemi kararlı yapabilecek tüm PID kontrolörlerin bulunabilmesi için kullanılabilecek bir fonksiyon kümesi ortaya çıkmıştır. Elle hesaplanması çok uzun sürecek olan bu işlem, çok basit bir şekilde ilgili fonksiyonun içine bir sistem transfer fonksiyonu girilerek hızlıca yapılabilmektedir. Belirsizlik içermeyen bir sistemin kararlı olduğu PID kontrolör bölgesinin bulunması otomatik ayarlarda ortalama olarak 0,5 saniye sürmektedir. Bu süre, K_p aralığının ve adımının seçimine göre artıp azalabilmektedir.

Aynı şekilde parametrik belirsizliği olan sistemler için de literatürdeki herhangi bir yöntem ile incelenmesi gereken transfer fonksiyonları bulunduktan sonra, fonksiyon kümesindeki ilgili fonksiyon kullanılarak bu transfer fonksiyonlarının hepsinin birden kararlı olduğu dayanıklı PID kontrolör bölgesi bulunabilir. Bu işlem yapılırken incelenecek transfer fonksiyonlarının her biri için kararlı bölge bulunacağından incelenen transfer fonksiyonlarının miktarına bağlı olarak sonuç bulma süresi değişecektir.

Son olarak çeşitli örnekler için fonksiyonlar denenmiş ve elde edilen sonuçlardan farklı noktalar seçilip kapalı çevrim sistemin kutuplarının yerlerine bakılarak hepsinin sol yarı düzlemde olduğu görülmüştür.

6. KAYNAKLAR

[1] **Schlegel, M., Mertl, J., Čech M.,** 2003. Generalized robustness regions for PID controllers, *Process Control.*

[2] **Åström, K. J., Hägglund, T.,** 2001. The future of PID control, *Control Engineering Practice,* **9,** 1163-1175.

[3] **Shafiei, Z., Shenton A. T.,** 1997. Frequency-domain design of PID controllers for stable and unstable systems with time delay, *Automatica,* **33,** 2223-2232.

[4] **Ackermann, J., Kaesbauer D.,** 2001. Design of robust PID controllers, *European Control Conference,* Porto.

[5] **Keel, L. H., Bhattacharyya, S. P.,** 2005, PID controller synthesis free of analytical models, *IFAC.*

[6] **Bajcinca, N.,** 2006. Design of robust PID controllers using decoupling at singular frequencies, *Automatica,* **42,** 1943-1949.

[7] **Tan, N.,** 2009. Computation of stabilizing PI-PD controllers, *International Journal of Control, Automation, and Systems,* **7(2),** 175-184.

[8] **Söylemez, M. T., Munro, N., Baki, H.,** 2002. Fast calculation of stabilizing PID controllers, *Automatica,* **39,** 121-126.

[9] **Huang, Y. J., Wang Y. J.,** 2000. Robust PID tuning strategy for uncertain plants based on the Kharitonov theorem, *ISA Transactions,* **39,** 419-431.

[10] **Ho., M. T., Datta, A., Bhattacharyya S. P.,** 2001. Robust and non-fragile PID controller design, *International Journal of Robust and Nonlinear Control,* **11,** 681-708.

[11] **Silva, J., Datta, A., Bhattacharyya, S. P.,** 2005. PID Controllers for Time-Delay Systems, pp 57-67,199-204, Birkhäuser, Boston.

[12] **Bhattacharyya, S. P., Datta, A., Kell, L. H.,** 2009. Linear Control Theory, Structure, Robustness and Optimization, pp 33-39, CRC Press.

EKLER

EK A: Örnekler

EK A

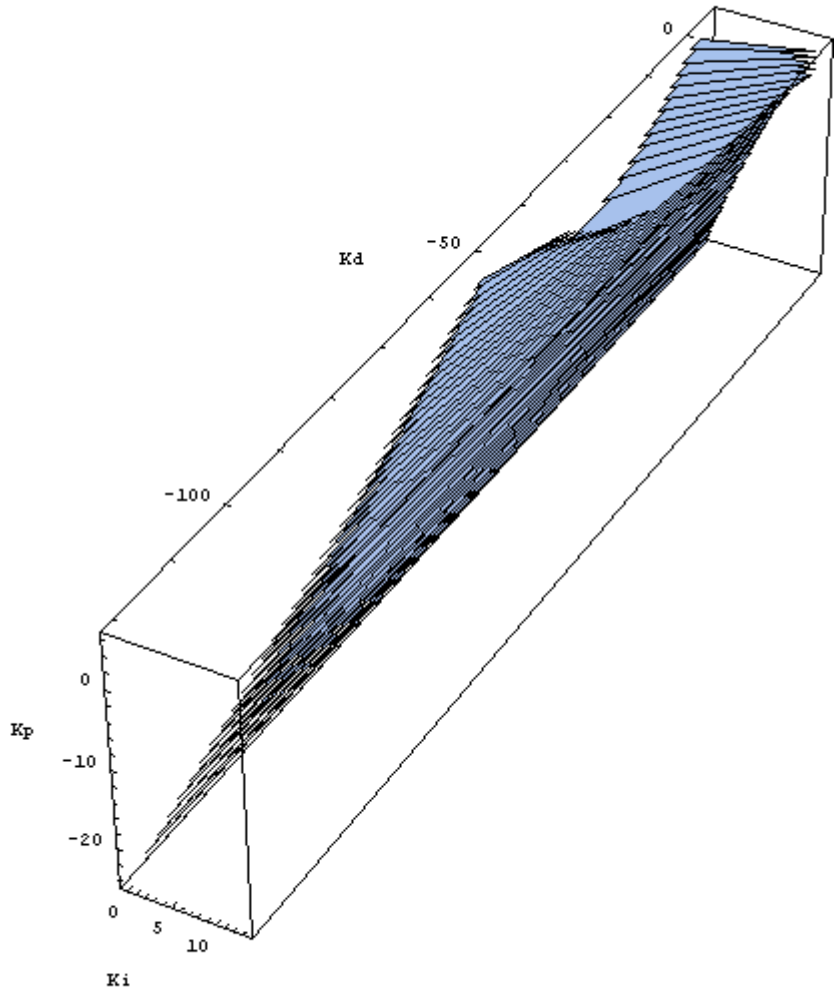
$$Gs = \frac{-0.5\,s^4 - 7\,s^3 - 2\,s + 1}{s^6 + 11\,s^5 + 46\,s^4 + 95\,s^3 + 109\,s^2 + 74\,s + 24};$$

```
FindKpRange[Gs]
```

```
{-24., 6.15651}
```

```
volume = FindPID3D[Gs];
```

```
Show[GraphPID3D[volume]]
```




```
- Graphics3D -
```

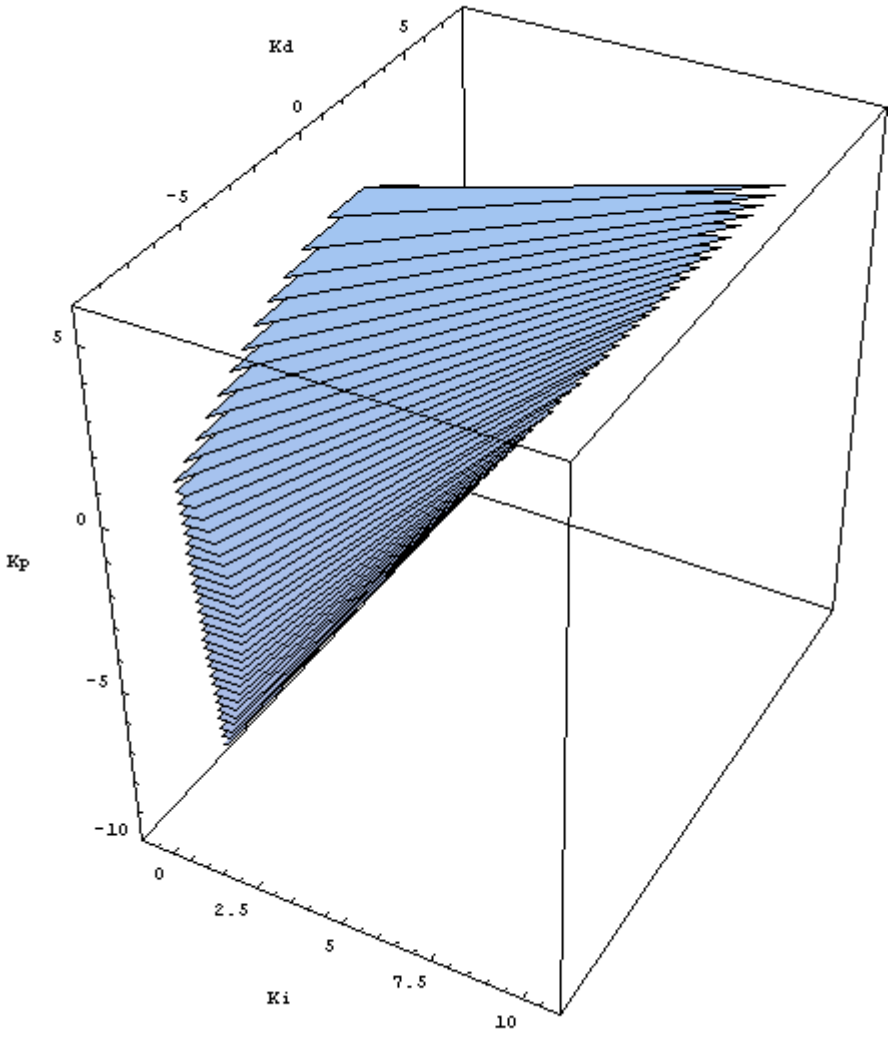
$$Gs = \frac{s^3 - 4s^2 + s + 2}{s^5 + 8s^4 + 32s^3 + 46s^2 + 46s + 17};$$

```
FindKpRange[Gs]
```

```
{-8.5, 4.23337}
```

```
volume = FindPID3D[Gs];
```

```
Show[GraphPID3D[volume]]
```




```
- Graphics3D -
```

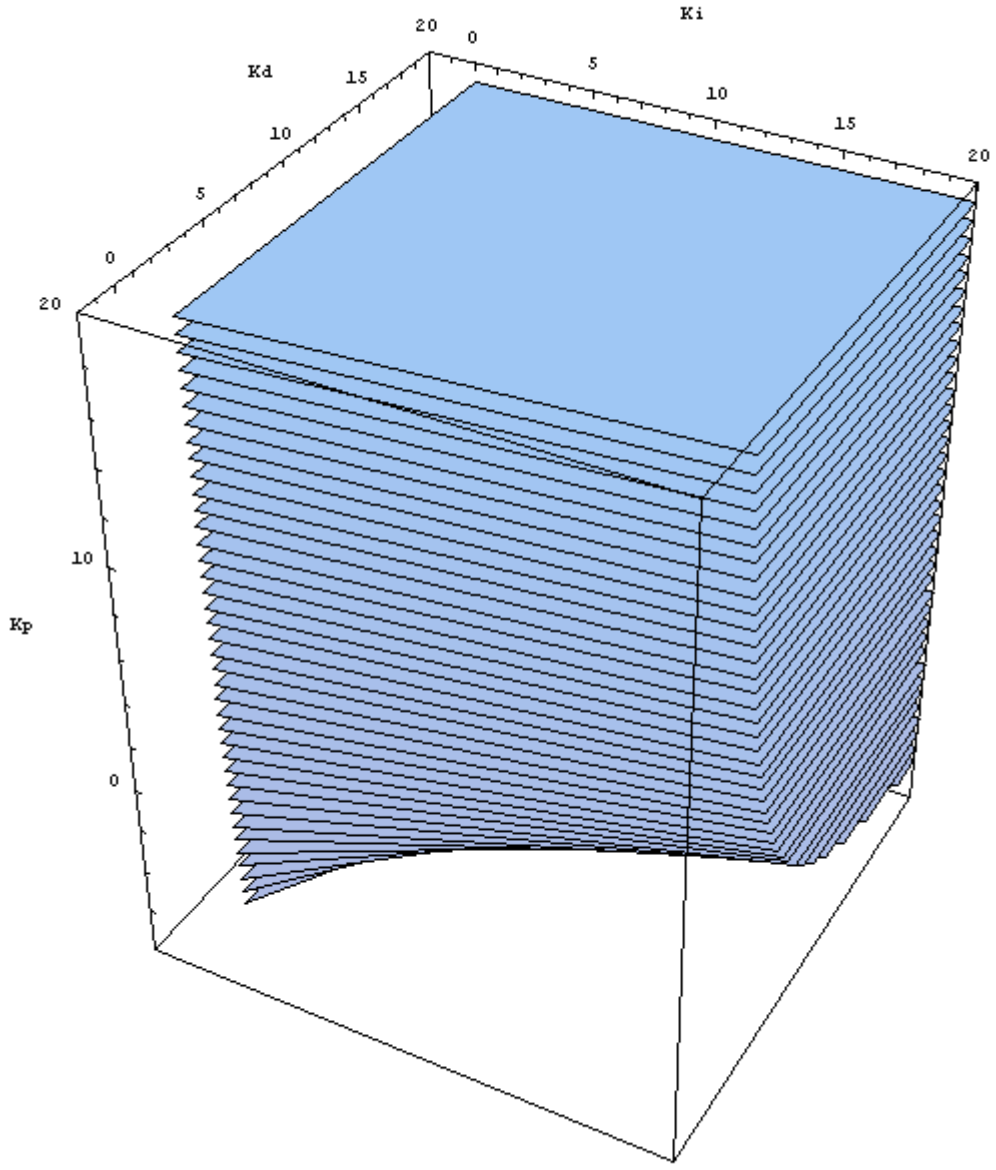
$$Gs = \frac{s+2}{2s^3+4s^2+6s+18};$$

```
FindKpRange[Gs]
```

```
{-9., 19994.}
```

```
volume = FindPID3D[Gs, KpRange → {-8, 20}];
```

```
Show[GraphPID3D[volume, PlotRange → {{-2, 20}, {-2, 20}, {-8, 20}}]]
```




```
- Graphics3D -
```

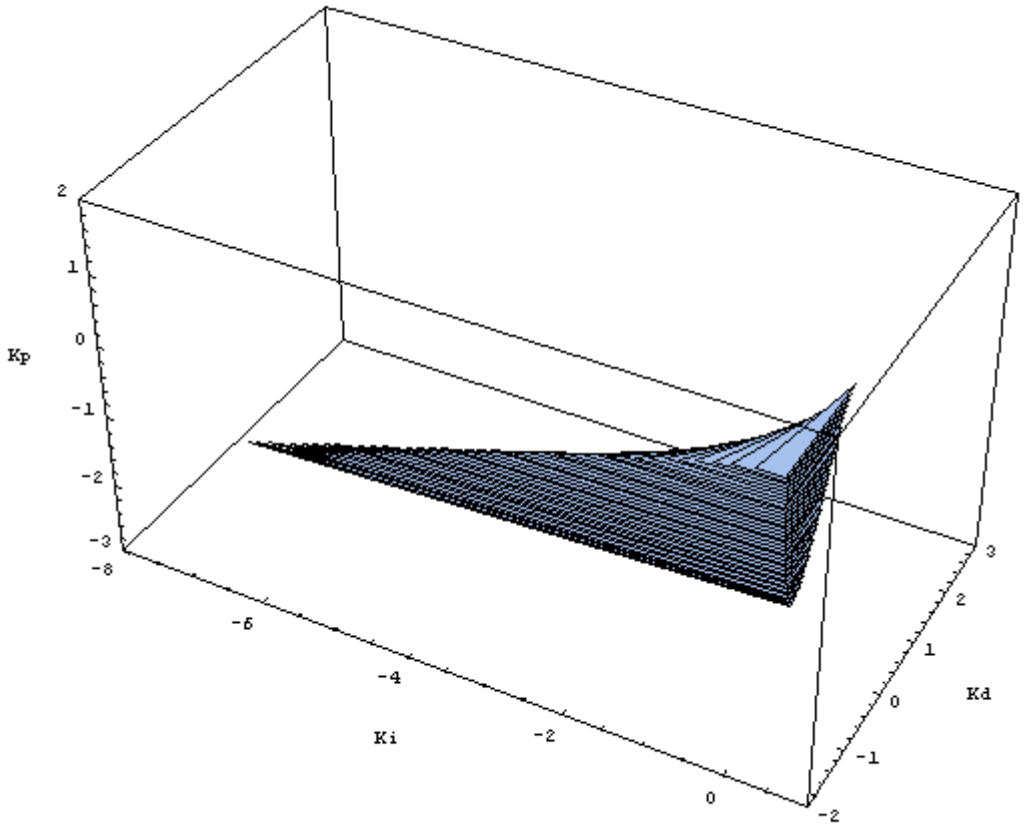
$$Gs = \frac{11.2\,(s+0.5)\,(s-3)}{(s+0.2)\,(s+3.5)\,(s+3.6)\,(s+2)};$$

```
FindKpRange[Gs]
```

```
{-1.79884, 0.308033}
```

```
volume = FindPID3D[Gs];
```

```
Show[GraphPID3D[volume]]
```




```
- Graphics3D -
```

ÖZGEÇMİŞ



Ad Soyad	**:** Uğur Yıldırım
Doğum Yeri ve Tarihi	**:** 03.08.1987
Adres	**:** Sefaköy İnönü Mah. Şehitler Sk. No:24 D:2 KÜÇÜKÇEKMECE/İSTANBUL
Lisans Üniversite	**:** İstanbul Teknik Üniversitesi