

MEKATRONİK SİSTEMLERDE KONTROLÇÜ TASARIMI

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

YÜKSEK LİSANS TEZİ

Mak. Müh. Kerem KIRCA

514981005

Tezin Enstitüye Verildiği Tarih : 22 Ocak 2001

Tezin Savunulduğu Tarih : 7 Şubat 2001

Tez Danışmanı : Doç. Dr. Levent GÜVENÇ

Diğer Jüri Üyeleri : Prof. Dr. Cemil ERKMAN

Yrd. Doç. Dr. Ata MUĞAN

[Handwritten signatures of the thesis advisor and jury members]

ŞUBAT 2001

104213

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	ix
ÖZET	x
SUMMARY	xi
1.GİRİŞ	1
2. MEKATRONİK UYGULAMALARDA KONTROLÇÜ GELİŞTİRME	3
2.1 Hızlı Kontrolcü Prototiplendirme	4
2.1.1 Hızlı Kontrolcü Prototiplendirme Sistemleri	4
2.1.2 dSPACE Prototiplendiricisi ile Hızlı Prototiplendirme	5
2.1.3 xPC Target Hızlı Prototiplendirme Sistemleri.....	9
2.2 Çevrim İçi Donanım İçeren Simülasyon.....	9
2.2.1 WINCON Çevrim İçi Donanım İçeren Simülasyon Sistemi	11
3. TERS SARKACIN SİMÜLASYON MODELİNİN OLUŞTURULMASI	14
3.1 Ters Sarkacın Matematik Modelinin Oluşturulması	15
3.2 Pol Ataması Tekniği İle Durum Geri Besleme Kontrolcü Tasarımı.....	17
3.2.1 Durum geri besleme kazanç matrisinin MATLAB® programı ile elde edilmesi ...	23
3.2.2 Durum Geri Besleme Kazancının Ackermann Formülü ile Elde Edilmesi	25
3.3 Ters Sarkaç Modelinin ve Kontrolçüsünün Gürbüzlük Analizi	26
3.4 Lineer Kuadratik Regülatör Tasarımı.....	30
3.5 Ters Sarkaç Sisteminin Matlab/Simulink® İle Modellenmesi	34
4. LİNEER OLMAYAN SİSTEMİN BULANIK MANTIK İLE KONTROLÜ	39
4.1 Bulanık Çıkarım Sistemi	39
4.2 Bulanık Sistem Simülasyonları	42
4.3 Üç Giriş Değişkenli Bulanık Çıkarım Sistemi.....	44
4.4 Üç Girişli Sistem İçin Bulanık Kontrolcü Simülasyonları.....	48
5. TERS SARKAÇ DENEY UYGULAMASI.....	51

6. SONUÇLAR VE ÖNERİLER.....	55
KAYNAKLAR.....	57
ÖZGEÇMİŞ	59



KISALTMALAR

BDM :Bilgisayar Destekli Mühendislik

ECU :Electronic Control Unit

DSP :Digital Signal Processing

CAN :Controller Area Network

RTI :Rea-Time Interface



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Bulanık Sistem Kural Tablosu.....	41
Tablo 4.2. Üç Girişli Sistem İçin Bulanık Kural Tablosu	46



ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1	Kontrolcü için Tüm Gelişim Sistemi Akış Şeması	6
Şekil 2.2	ControlDesk program paneli	8
Şekil 2.3	xPC Target Hızlı Prototiplendirme Sistemi	9
Şekil 2.4	Çevrim İçi Donanım ve Hızlı Prototiplendirme Sistemleri	11
Şekil 2.5	WinCon Çevrim İçi Donanım İçeren Simülasyon Sistemi	12
Şekil 3.1	Ters Sarkaç Sistemi	14
Şekil 3.2	Sarkaç Serbest Cisim Diyagramı	15
Şekil 3.3	Durum Geri Besleme Blok Diyagramı	20
Şekil 3.4	θ ve $\dot{\theta}$, Durum Geri Besleme Kontrolcüsü ile Kontrol	24
Şekil 3.5	Araba Konum ve Hızı, Durum Geri Besleme Kontrolcüsü ile Kontrol	25
Şekil 3.6	Ters Sarkaç Sistemi İçin L ve m Değişken Parametreleri ile Gürbüzlük Tasarımı	27
Şekil 3.7	Paradise Programı Parametre Özellikleri Penceresi	27
Şekil 3.8	L ve m Değişken Parametreleri için Paradise Programı Parametre Uzayı Penceresi	28
Şekil 3.9	Ters Sarkaç sistemi İçin K(1) ve K(3) Değişken Parametreleri ile Gürbüzlük Tasarımı	29
Şekil 3.10	Lineer Kuadratik Regülatör Sisteminin Basamak Girişe Cevabı	32
Şekil 3.11	Lineer Kuadratik Regülatör Sisteminin Basamak Girişe Cevabı	33
Şekil 3.12	Ters Sarkaç Alt Sistemi	34
Şekil 3.13	Ters Sarkaç Kontrol Sisteminin Simulink Modeli	34
Şekil 3.14	Sarkaç Kol Açısı ve Açısal Hızının Durum Geri Besleme Simülasyon Cevapları	35
Şekil 3.15	Araba Konumu ve Hızının Durum Geri Besleme Simülasyon Cevapları	35
Şekil 3.16	Sarkaç Kol Açısı ve Açısal Hızının Lineer Olmayan Sistemde Bozucu Etken Cevabı	37
Şekil 3.17	Araba Konumu ve Hızının Lineer Olmayan Sistemde Bozucu Etken Cevabı	37
Şekil 3.18	Sarkaç Kol Açısı ve Açısal Hızının LKR ile Lineer Olmayan Sistemde Bozucu Etken Cevabı	38
Şekil 3.19	Araba Konumu ve Hızının LKR ile Lineer Olmayan Sistemde Bozucu Etken Cevabı	38
Şekil 4.1	Bulanık Çıkarım Sistemi Blok Diyagramı	39
Şekil 4.2	Sarkaç Kolu Açısının Üyelik Fonksiyonları	40
Şekil 4.3	Sarkaç Kolu Açısal Hızının Üyelik Fonksiyonları	40
Şekil 4.4	Kuvvet Kontrol Büyüklüğünün Üyelik Fonksiyonları	41
Şekil 4.5	Kuvvet, Sarkaç Kolu Açısı ve Açısal Hızının 3-B Yüzey Diyagramı	42
Şekil 4.6	Bulanık Kuralların Görüntülenmesi	42
Şekil 4.7	Sarkaç Kolu İçin Bulanık Kontrolcü Simülasyon Modeli	43
Şekil 4.8	Bulanık Kontrolcü ile Sarkaç Kolu Açısının ve Açısal Hızının Cevapları	43

Şekil 4.9	Üç Girişli Bulanık Çıkarım Sistemi Blok Diyagramı	44
Şekil 4.10	Sarkaç Kolu Açısının Üç Girişli Sistem İçin Üyelik Fonksiyonları	45
Şekil 4.11	Sarkaç Kolu Açısıl Hızının Üç Girişli Sistem İçin Üyelik Fonksiyonları	45
Şekil 4.12	Araba Konumunun Üç Girişli sistem İçin Üyelik Fonksiyonları	45
Şekil 4.13	Kuvvet Kontrol Büyüklüğünün Üç Girişli sistem İçin Üyelik Fonksiyonları	45
Şekil 4.14	Üç Girişli Sistemin Sarkaç Kolu Açısı, Açısıl Hızı ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği.....	46
Şekil 4.15	Üç Girişli Sistemin Sarkaç Kolu Açısı, Araba Konumu ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği.....	47
Şekil 4.16	Üç Girişli Sistemin Sarkaç Kolu Açısıl Hızı, Araba Konumu ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği.....	47
Şekil 4.17	Üç Girişli Sistem için Bulanık Kuralların Görüntülenmesi	47
Şekil 4.18	Ters Sarkaç Sistemi İçin Üç Boyutlu Simülasyon Modeli	48
Şekil 4.19	Üç Girişli Bulanık Kontrolcü ile Sarkaç Kolu Açısının ve Açısıl Hızının Simülasyon Cevapları.....	49
Şekil 4.20	Üç Girişli Bulanık Kontrolcü ile Araba Konumu ve Hızının Simülasyon Cevapları.....	49
Şekil 4.21	Araba Konumu ve Hızının $t = 30$ (sn) için Bulanık Kontrolcü Sistem Cevabı.....	50
Şekil 5.1	IP-02 Ters Sarkaç Simulink Modeli.....	53
Şekil 5.2	Durum Geri Besleme Kazanç Çarpanları Blok Diyagramı	53
Şekil 5.3	Sarkaç Kolu Açısının Gerçek Zamanda Deney Cevabı.....	54

SEMBOL LİSTESİ

A	: Durum matrisi
B	: Kontrol giriş matrisi
C	: Çıkış matrisi
D	: Kontrol çıkış matrisi
E	: Bozucu etki matrisi
F	: Sarkaç koluna uygulanan bozucu etki kuvveti
G	: Sarkaç kolu ağırlık merkezi
g	: Yerçekimi ivmesi
I	: Sarkaç kolu atalet momenti
I_m	: Tahrik motorunun çektiği akım
J	: Performans endeksi
K	: Durum geri besleme kazanç matrisi
K_g	: Tahrik motoru dişli kutusundaki çevirme oranı
K_m	: Elektromotor kuvveti sabiti (Back EMF Constant)
L	: Sarkaç boyu
l	: Sarkaç yarı boyu
M	: Kontrol edilebilirlik matrisi
M	: Araba kütlesi
m	: Sarkaç kütlesi
Q	: Optimal kontrolcü parametresi
R	: Optimal kontrolcü parametresi
R_m	: Motor direnci
r	: Rayla temas halindeki pinyonun yarıçapı
s_{1,2}	: Karakteristik denklemin kökleri
T	: Geçiş matrisi
T_m	: Tahrik motounun ürettiği moment
t_s	: Yerleşme zamanı
θ	: Sarkaç kolu açısı
θ̇	: Sarkaç kolu açısal hızı
θ̈	: Sarkaç kolu açısal ivmesi
θ_d	: Sarkaç kolu açısı referans değeri
u	: Arabaya uygulanan kontrol kuvveti
V	: Motorun çektiği gerilim
W	: Ağırlık matrisi
ω_g	: Tahrik motoru çıkış açısal hızı
x	: Araba konumu
ẋ	: Araba hızı
ẍ	: Araba ivmesi
x_d	: Arabanın referans konumu
x_{1,2,3,4}	: Durum değişkenleri
x_G	: Sarkaç kolu ağırlık merkezi yatay koordinatı
y_G	: Sarkaç kolu ağırlık merkezi dikey koordinatı
ζ	: Sönümleme faktörü

ÖZET

Hızlı kontrolcü prototiplendirme ve çevrim içi donanım içeren simülasyonlar, tasarım ve bunu izleyen gerçekleştirme sürecini destekleyen önemli bilgisayar destekli mühendislik araçlarıdır. Bu tez çalışmasının kapsamında bu iki geliştirme sistemi tanıtılmıştır. Ayrıca bu sistemlerin basit olarak kullanıldığı ters sarkaç sisteminin modellenmesi ve kontrolü de uygulama örneği olarak sunulmuştur.

Ters sarkaç sistemi kararsız dinamik yapısı nedeniyle yeni geliştirilen birçok kontrol teknolojisinin uygulama alanı bulduğu bir modeldir. Bu çalışmada, lineer ve lineer olmayan matematik modelleri çıkarılan ters sarkaç sistemi için durum geri beslemesi ve lineer kuadratik optimal kontrol tasarımları yapılmıştır. Bu tasarımlarla elde edilen kontrolcüler için elde edilen simülasyon sonuçları da verilmiştir. Elde edilen durum geri besleme kontrolcülerinin parametre değişikliklerine dayanıklılığı da incelenmiştir. Ayrıca lineer olmayan sistem modeli için yine lineer olmayan bir kontrolcü olan bulanık mantık kontrolcüsü tasarlanmış ve performansı incelenmiştir.

Sonuç olarak, ters sarkaç sistemi için, hızlı kontrolcü prototiplendirme metotları kullanılarak bir kontrolcü modeli ortaya çıkarılmıştır. Bu model ve çeşitli kontrol yöntemleri kullanılarak, ters sarkaç sistemi için tasarlanmış kontrolcü performansı iyileştirilmeye çalışılmıştır.

CONTROLLER DESIGN IN MECHATRONIC SYSTEMS

SUMMARY

Rapid controller prototyping and hardware in the loop simulations are two important, computer aided engineering tools that support design and following implementation process. This thesis addresses the issue of these development systems. An inverted pendulum model and controller design using the rapid controller prototyping philosophy is taken as an application.

The inverted pendulum system with its unstable dynamic structure helps understanding many of the new developed control systems. In this thesis work state feedback and linear quadratic optimal controller designs took place for both linear and non-linear mathematical models of the inverted pendulum system. Graphical simulations of the obtained controller have been made. The results of these simulations are also given. The robustness of the designed state variable feedback controller to the varying parameters is studied. A non-linear fuzzy logic controller is designed and the performance of this controller is examined for non-linear model of the system.

As a result a controller model for the inverted pendulum system has been achieved by rapid controller prototyping methods. Using this model and various control methods, further optimization has been carried out.

1. GİRİŞ

Mekanik ve elektronik sistemlerinin birleştirilmesi ile yaratılan mekatronik teknolojisinin her dalında, karmaşık görevleri yerine getiren kontrolcü sistemlerine duyulan ihtiyaç hızla artmaktadır. Mekatronik teknolojisinin gelişimi için, tasarım ve bunu izleyen gerçekleştirme sürecini destekleyecek yüksek verimlilikli bilgisayar destekli mühendislik (BDM) araçlarına ihtiyaç vardır. Bu çalışma kapsamında hızlı kontrolcü prototiplendirme ve çevrim içi donanım simülasyonları olarak adlandırılan bu tip BDM araçları tanıtılacaktır.

Kontrol teknolojisinde artık standart uygulamalar haline dönüşmüş bu iki sistem teori ve pratik yaklaşımlar arasındaki engellerin aşılmasında birer gereklilik halini almışlardır. Bu çalışmanın kapsamında bu iki bilgisayar destekli mühendislik aracı tanıtılmıştır.

İkinci bölümde Hızlı Prototiplendirme Sistemleri hakkında bilgi verilirken bu alanda kullanılan MATLAB®, blok diyagram temelli grafiksel simülasyonlar için kullanılan Simulink®, gerçek zamanda kontrol uygulamaları için dSPACE ve tasarımcıya özel donanıma ihtiyaç duymadan daha hesaplı tasarımlar yapabilme imkanı sunabilen xPC gibi programlardan ve tüm gelişim sistemlerinden bahsedilmiştir. Hızlı prototiplendirme sistemleri tasarım aşaması ile deneysel gerçekleştirme arasında yumuşak ve hızlı bir geçiş imkanı sunmaktadır.

Çevrim içi donanım içeren simülasyonlar ikinci bölümün bir diğer alt başlığını oluşturmaktadır. Çevrim içi donanım içeren simülasyonlar, sanal ortamda yapılan sistem testlerinden sonra sadece ilgilenilen donanımın simülasyon ortamına eklenmesiyle prototip maliyetlerini azaltmaktadırlar. Çevrim içi donanım içeren simülasyon, kısa zamanda özellikle otomotiv araştırma ve geliştirme sürecinin ayrılmaz bir parçası haline gelmiştir. Örneğin çevrim içi donanım içeren simülasyonlar, elektronik kontrol birimlerinin (Electronic Control Unit) simüle edilmiş bir kapalı çevrim ortamında test edilmelerine olanak sağlamaktadırlar.

İkinci bölümde son olarak basit bir hızlı prototiplendirme sistemi ve çevrim içi simülasyon örneği olan WinCon'un ters sarkaç modeli tanıtılmıştır. Bu sistem aynı zamanda bir sonraki bölümde tasarlanacak ters sarkaç matematik modelinin parametrik ve fiziksel temelini oluşturmaktadır.

Üçüncü bölümde, hızlı kontrolcü prototiplendirme yöntemleri izlenerek bir ters sarkaç sisteminin lineer ve lineer olmayan matematik modellerinin oluşturulması, bu modeller için durum geri beslemesi ve lineer kuadratik optimal kontrol metotları kullanılarak kontrolcü tasarımı oluşturulması konu edilmiştir. MATLAB programları ve Simulink blok diyagramları yardımıyla ters sarkaç lineer modeli için tasarlanan kontrolcülere ait simülasyon sonuçları elde edilmiş ve incelenmiştir.

Daha sonra Simulink fonksiyon bloklarından faydalanılarak lineer olmayan sistem modeli de lineer sistem için tasarlanmış kontrolcülerle test edilebilmiştir. Elde edilen simülasyon sonuçları incelenmiştir. Ayrıca durum geri besleme kontrolcüsü ve ters sarkaç modelinin değişken parametreleri için gürbüzlük analizi de bu bölümde yapılmıştır.

Dördüncü bölümde ise tasarlanan lineer olmayan ters sarkaç matematik modelinin yine lineer olmayan fakat matematik modeli çıkarılamayacak kadar karmaşık sistemlerde modelden bağımsız gerçekleştirilebilen bulanık mantık kontrolcü sistemi oluşturulmuştur. Bu tasarım için MATLAB altında çalışan Fuzzy Logic Toolbox programından faydalanılırken, simülasyonlar için yine Simulink grafik temelli simülasyon programı kullanılmıştır.

Beşinci bölümde, üçüncü bölümde tasarlanıp simülasyonları gerçekleştirilen durum geri besleme kontrolcüsünün Wincon IP.02 ters sarkaç deney düzeneği ve Simulink ara yüz programı yardımı ile gerçek zamanda fiziksel sistem üzerinde test edilmiştir.

Altıncı ve son bölümde ise sonuçlar ve ileri zamanda yapılabilecek çalışmalar için öneriler yer almaktadır.

2. MEKATRONİK UYGULAMALARDA KONTROLCÜ GELİŞTİRME

Dijital elektronik teknolojisinde son on yıl içerisinde ulaşılan nokta 'Mekatronik' gibi yeni mühendislik disiplinlerinin doğmasına yol açmıştır. Mekanik ve elektronik sistemlerin birleştirilmesi ile yaratılan mekatronik teknolojinin her dalında, tasarlanan kontrolcülerin, karmaşık görevleri yerine getirdiği sistemlere ihtiyaç hızla artmaktadır.

Dijital elektroniğin getirdiği avantajların tam anlamı ile kullanılabilmesi için çeşitli mühendislik dallarının işbirliği gerekmektedir. Örneğin gelen talepler doğrultusunda, ihtiyaç duyulan yeni ürünlerin olabildiğince hızlı tasarımı ve geliştirilmesi makine, bilgisayar ve kontrol mühendislerinin beraber ve verimli çalışmalarına bağlıdır. Buna paralel olarak mekatronik teknolojinin gelişimi için, tasarım ve bunu takip eden gerçekleştirme sürecini destekleyen yüksek verimlilikte bilgisayar destekli mühendislik (BDM) araçlarına ihtiyaç vardır. Feng, Töngren ve Eriksson [1] bu araçların bazı örneklerini aşağıdaki gibi madde madde belirtilmiştir.

- Analiz, tasarım ve simülasyon araçları içeren Grafiksel Tasarım Uygulamaları
- Hızlı Prototiplendirme
- Gerçek Zamanda Simülasyon, Gerçek Zamanda Kontrol
- Çevrim İçi Donanım İçeren Simülasyon
- Dijital Sinyal Üretimi (DSP) tabanlı Üretim için Donanım Desteği, I/O ve İletişim Araçları, örnek: Controller Area Network (CAN)

Bu bağlamda, iki tane çok önemli geliştirme metodu artık standart uygulamalar haline dönüşmüştür. Bunlar Hızlı Kontrolcü Prototiplendirme (Rapid Controller Prototyping) ve Çevrim İçi Donanım Denemeleridir (Hardware in the Loop Test). Bu çalışmanın kapsamında da bu iki bilgisayar destekli mühendislik aracı tanıtılmıştır. Ayrıca bu sistemlerin basit olarak kullanıldığı bir ters sarkaç sisteminin modellenmesi de uygulama örneği olarak sunulmuştur.

2.1 Hızlı Kontrolcü Prototiplendirme

Yukarıda bahsedilen BDM araçları üzerinde yürütülen endüstriyel ve akademik araştırmalar sonucunda hızlı prototiplendirme sistemleri, yeni tasarım alternatiflerinin geliştirilmesinde ve teori ile pratik yaklaşımlar arasındaki engellerin aşılmasında bir gereklilik halini almıştır.

MATLAB/Simulink ve MATRIXx/SystemBuild gibi tasarım programları, kullanıcının geliştirmek istediği kontrolcüyü doğrudan blok diyagramlar içinde tasarlamasına imkan vermektedir. Bu blok diyagramlar, gerçek zaman kodlarını oluşturup, bu kodları değiştirilebilir prototip donanımında doğrudan uygulamaya sokabilmektedirler.

Kullanıcı, kapalı çevrimde, proses daha geliştirme aşamasındayken, yani düzeltme maliyetlerinin en az derecede olduğu bir ortamda, sistem parametrelerini çevrim-içi olarak değiştirebilir ve gerçek zamanda akış verilerini kaydedebilir. Ayrıca otomasyon tasarım planları üzerinden tekrar tekrar geçebilme şansına sahiptir. Bu avantaj üreticiyi, hataya oldukça açık elle programlamadan ve can sıkıcı biçimde tekrarlanan prototip yapımından kurtararak muazzam bir para ve zaman kazancı sağlamaktadır. Aynı zamanda kullanılan yazılım kalitesinde de önemli bir gelişim görülmektedir.

2.1.1 Hızlı Kontrolcü Prototiplendirme Sistemleri

Elektro–mekanik sistemlerin gün geçtikçe daha karmaşık yapılarda gelişmesi, iyi tasarlanmış, hassas kontrol sistemlerinin olabildiğince hızlı yapılandırılması ihtiyacını doğurmuştur. Günümüzde, Hızlı Kontrolcü Prototiplendirme Sistemleri (Rapid Controller Prototyping Systems) ile çeşitli kontrol problemleri çözümünde uğraşan mühendisler, herhangi bir programlama diline ihtiyaç duymadan her türlü tahrik elemanına, örneğin; servo motorlara, kumanda edebilecek kontrolcüler tasarlayabilmektedirler. Studt [2], kontrol sistemleri için hızlı tasarım araştırmalarını incelemiş, MathWorks Inc.™ tarafından üretilen MATLAB® ve SIMULINK® Simülasyon Sistemlerinin son birkaç yıldır bu tarz uygulamalarda sıkça kullanılmakta olduğunu belirtmiştir. Bu ürünlerin yeni versiyonları (SIMULINK ver.3.0, Stateflow ver.2.0 ve Real Time Workshop ver.3.0), kontrolcü tasarımı ve otomasyon çözümlmeleri için daha hızlı ve daha doğru simülasyonlar yapma imkanı sunarken,

büyük ölçekteki kontrol uygulamalarında, kod ve sistem genişletme çalışmalarında optimum desteği sağlamaktadırlar.

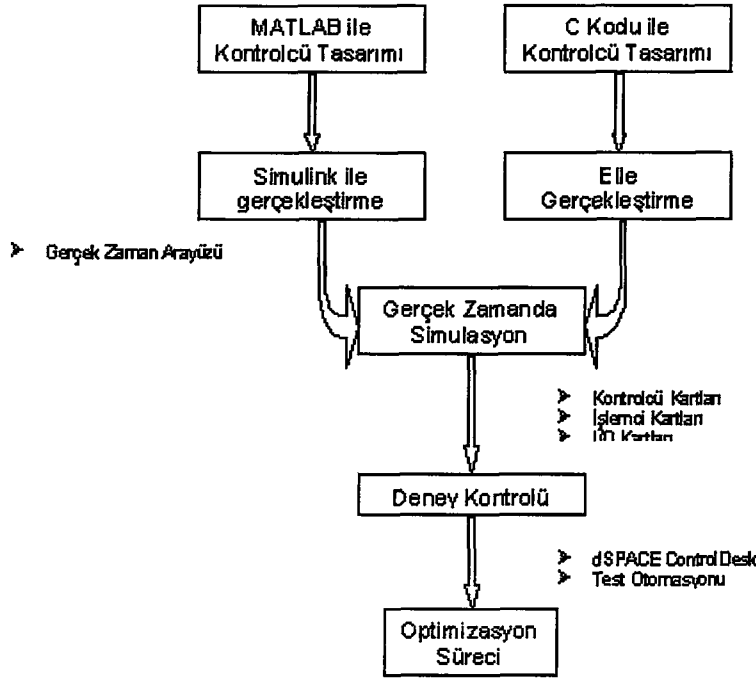
SIMULINK içinde kullanım kapasitesindeki geliştirme çalışmaları, tasarımın içeriğini anlamaya yardım eden model özelliklerinin gösterimi, diyagram yapıları arasında otomatik bağlantı oluşturma, MATLAB Control Design Toolbox ve SIMULINK arasında hızlı kontrolcü algoritma analizi için bütünleşik lineer analiz yapma yeteneği gibi konular içermektedir.

2.1.2 dSPACE Prototiplendiricisi ile Hızlı Prototiplendirme

dSPACE prototiplendiricisi, kullanıcıya gerçek sistemler üzerinde hiçbir programlama işine ihtiyaç duymadan, doğrudan SIMULINK® modelleri ile tasarladığı kontrolcüyü test etme imkanı sunan esnek bir geliştirme sistemidir. Tasarımcı, kişisel dSPACE prototiplendiricisini oluşturabileceği geniş bir yazılım/donanım yelpazesinden seçme şansına sahiptir.

Hızlı kontrolcü prototiplendirme kavramına uygun olarak dSPACE, kontrolcü tasarımı sürecinin bütün adımlarını içeren tam bir gelişim ortamı (Total Development Environment) sunmaktadır. MATLAB ve Simulink gibi tasarım ve analiz programları ile birleşerek kullanıcıya geliştirme sürecinin her aşamasında tasarımını test edip doğrulayabileceği tümleşik bir yazılım/donanım sistemi sağlamaktadır. Bu sistem aynı zamanda MATLAB ve Simulink dışında elle oluşturulmuş C kodu gibi diğer kaynakları da kullanabilmektedir.

Hanselmann [3], bilgisayar destekli kontrol sistemlerinin, kontrolcü tasarımı ve çevrim dışı simülasyonlarla kısıtlı kalmayarak gerçek zamanda yapılan hızlı kontrolcü prototiplendirmeyi de kapsadığını gözlemlemiştir. Sistem temel olarak; analiz, tasarım, optimizasyon ve çevrim dışı veri işlenmesinde MATLAB®, blok diyagram temelli simülasyonda SIMULINK®, tasarlanan blok diyagramlardan C kodu oluşturulmasında Real-Time Workshop™, bu kodun hedef işlemci sistemde çalıştırılması için Real-Time Interface (RTI) ve son olarak kapalı çevrim deneylerle iletişim ve etkileşimi sağlayan yine MATLAB tarafından otomatik hale getirilen ya da doğrudan kullanıcı tarafından oluşturulan bir grup yazılım kullanarak çalışmaktadır.



Şekil 2.1 Kontrolcü için Tüm Gelişim Sistemi Akış Şeması [4]

Tasarımcının Tüm Gelişim Ortamını kullanırken izleyeceği Şekil 2.1'de gösterilen minimum süreç Hanselmann [3], tarafından şu şekilde özetlenmiştir:

1. adım: Kontrol edilen sistemin teorik modelinin lineer ve/veya lineer olmayan denklemlerle çıkartılması. Daha sonra bu denklemlerin sistem dinamiğinin analizi için MATLAB m-file veya Simulink blok diyagram düzeninde uyarlanması.
2. adım: Sistem için bir ön kontrolcü tasarımı için Control System Toolbox, Non Linear Control Toolbox ve Robust Control Toolbox gibi çeşitli MATLAB toolbox 'ların kullanılması.
3. adım: Tasarlanan kontrol sisteminin çevrim dışı simülasyonları için Simulink'ten yararlanılarak MATLAB altında çalışan blok diyagram temelli grafiksel simülasyonların gerçekleştirilmesi.
4. adım: Tasarlanan sistemi gerçek zamanda testler için hazırlamak amacıyla Simulink blok diyagram kütüphanesinde tanımlı I/O (A/D çeviriciler, artımsal kodlayıcı ara yüzleri, vb.) bloklarının kullanılması.

5. adım: Hedef dijital sinyal prosesör sistemi (DSP) için C kodunun bir fare tıklaması sayesinde oluşturulması ve sistem için özelleştirilmiş I/O fonksiyonu ile gerçek zaman yazılımı içine yerleştirilmesi, derlenmesi ve son olarak hedef DSP sistemle bağlantı kurarak yazılımın aktarılması.
6. adım: Gerçek zamanda çalışan sistem ve kontrolcü arasında sanal bir panel sayesinde iletişimin kurulması. Böylelikle çevrim sırasında yazılım ve parametre ayarlamalarının yapılabilmesi.
7. adım: Tasarlanan kontrol tasarımı kabul edilebilecek seviyede çalışıyor olsa bile tasarımcı yine de sistemin iyileştirilmesi için yeni yollar aramalıdır. Kapalı çevrim sırasında elde edilen verilerin sistem çalışırken modele geri yüklenmesi söz konusu olabilir. Bu ihtiyaç yine dSPACE'in sağladığı TRACE programı sayesinde giderilebilir.
8. adım: Tekrar ilk adıma dönülerek gerçek zamanda gerçekleştirilen testler, sistem modelinin iyileştirilmesi için geri besleme şansı verir. Böylece tasarlanan kontrol sisteminin işlerliği veya dayanıklılığı hakkında fikir sahibi olunabilir.

Şekil 2.1'deki MATLAB ile kontrolcü tasarımı ve Simulink'ten gerçekleştirme bölümleri incelendiğinde, MATLAB/Simulink gibi modelleme sistemlerinin, kontrolcü prototiplendirmenin temellerini oluşturmaları dikkat çekmektedir. MATLAB, kullanıcıya sayısal hesaplamalardan, sistem analizi ve optimizasyonuna kadar geniş bir platform sunmaktadır. Simulink modelleme ve çevrim dışı simülasyonlar için geliştirilmiş, kullanımı kolay blok diyagramlardan oluşan interaktif bir programdır. Stateflow sayesinde Simulink içine durum diyagramları da entegre edilebilir. Aynı zamanda Real-Time Workshop ve Stateflow Coder programları tasarlanan blok diyagramlardan, bir sonraki aşama olan Gerçek Zaman Ara Yüzünün (Real Time Interface) temellerini oluşturacak C kodlarını üretebilmektedirler. Şekil 2.1'deki diğer bloklar dSPACE'in ilgili web sayfasında [5] aşağıdaki paragraflardaki gibi detaylı olarak anlatılmışlardır.

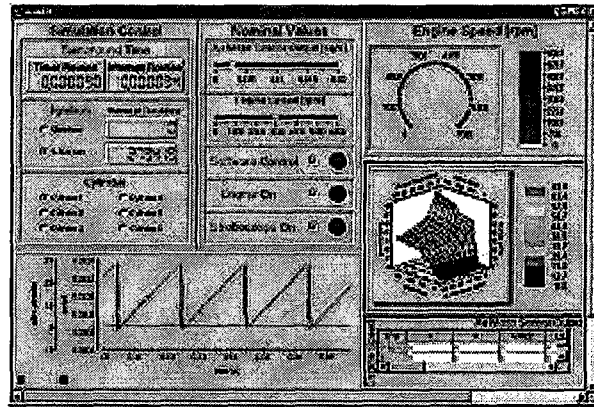
Hızlı tasarım yaklaşımları için gerçek zaman donanımı üzerindeki kontrol sisteminin hızlı ve hassas olarak yapılacak gerçekleştirilmesi esas alınmalıdır. Gerçek zaman ara yüzü ile Simulink modelleri, otomatik olarak dSPACE prototiplendirici donanımına, hiçbir programlama alt yapısına sahip olmadan yerleştirilebilir. Aynı zamanda C kodlu modellerin de sisteme entegrasyonu yapılabilmektedir.

Kontrol tasarımı sürecinde tamamen tasarım işine konsantre olunması gerekirken bu süre içinde donanım sınırlamaları ile mücadele etmek can sıkıcı olabilir. dSPACE donanımı hızlı kontrolcü prototiplendirme işini rahatlıkla yapabilecek kapasiteye sahiptir. Modüler dSPACE donanımı, gerçek zamanda karmaşık fonksiyonlardan oluşan sistemlerde dahi çevrim içi donanım denemeleri gerçekleştiren simülasyonları sonuçlandıracak güçtedir.

dSPACE sistemleri modüler olarak kurulabildikleri için, performans sınırlamaları da çok fazla değildir denebilir. TI - DSP, DEC – Alpha ya da Motorola tarafından üretilen PowerPC gibi işlemciler ve yüksek çözünürlükteki I/O kartları sayesinde sistem zorlu kontrol işlerinin üstesinden gelebilir.

Tasarlanan kontrol sistemi gerçek zaman donanımında çalışırken, kullanımı kolay PC yazılımları tasarımcıya deney sırasında sistemi kontrol etme şansını sağlamaktadırlar. Bu yazılımlarla sistem parametreleri değiştirilebilirken aynı zamanda değişkenlerin zamana göre akışı kapalı çevrim işlemlerinde bozucu etki yaratmadan izlenebilir, kaydedilebilir ve veri olarak saklanabilir. Ayrıca MATLAB programı için oluşturulan ara yüz ile test ve optimizasyon işlemleri otomatikleştirilebilmektedir.

Şekil 2.2.'de ara yüzü gösterilen ControlDesk, kullanıcının deney sırasında kontrol edilen sistemi sezgisel olarak yönetebildiği dSPACE sisteminin geniş kapsamlı deney programıdır. ControlDesk dSPACE prototiplendiricisi içine tümüyle entegre olmuş bir sistemdir.



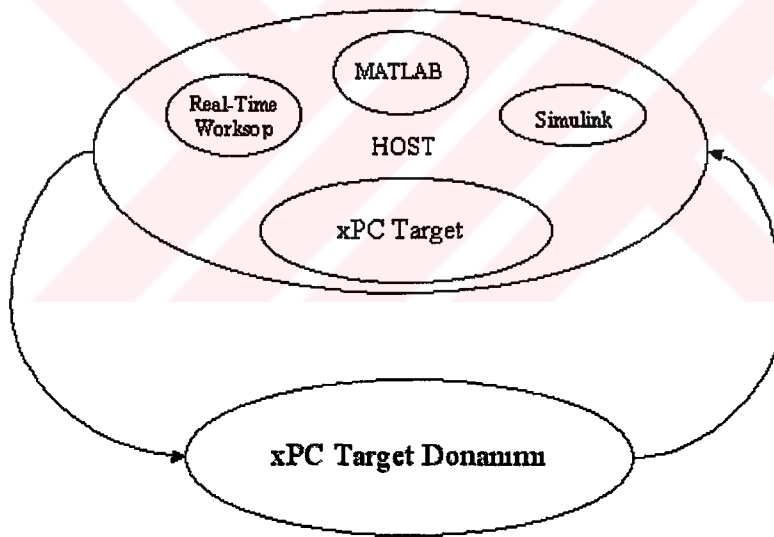
Şekil 2.2 ControlDesk program paneli

2.1.3 xPC Target Hızlı Prototiplendirme Sistemleri

Math Works Inc.TM 'ün geliştirdiği bu yeni program, kullanıcının, özel bir donanıma ihtiyaç duymadan standart PC donanımı ile gerçek zamanda çalışabilen sistem modelleri oluşturmaya imkan tanımaktadır. dSPACE[®] gibi tamamıyla profesyonel sistemlere göre bu özelliği sayesinde daha hesaplı olması ile dikkat çekmektedir.

İlgili web sayfasında yer alan [6] ve Şekil 2.3'te gösterilen xPC Target, tasarımcıya, oluşturduğu Simulink blok diyagramlarına I/O bloklarını ekleyip, Real-Time Workshop ile 'kod' oluşturma imkanı sağlarken, ayrıca bu kodu, xPC Target real-time kernel programını çalıştırarak hedef sistemi oluşturan ikinci bir PC 'ye aktarır. Bunun yanında kullanıcı üretim, veri toplama, kalibrasyon ve test uygulamalarında tasarladığı sistemleri gerçek zamanda PC donanımı üzerine yerleştirebilir.

xPC Target Intel 586/486/Pentium işlemcili her PC 'de gerçek zaman hedef sistemi olarak çalışma imkanı sunarken, tek bir karttan oluşan sistemleri de (single board computers) desteklemektedir.



Şekil 2.3 xPC Target Hızlı Prototiplendirme Sistemi

2.2 Çevrim İçi Donanım İçeren Simülasyon

Kontrol yazılımının, prototip içine yerleştirildikten sonra, elle ya da otomatik olarak belirli denemelerden geçirilmesi gerekmektedir. Geleneksel deneme sürecinde, kontrolcü prototipi genellikle kontrol uygulanacak sistem içine yerleştirilerek saha denemeleri sırasında dikkatlice test edilmektedir. Fakat bu yöntem, tahmin edileceği gibi uçaklar, fabrikalar ve akla gelebilecek farklı karmaşık sistemlerin geliştirme sürecinde oldukça sorunlu ve zahmetli olmaktadır. Geliştirme çalışmalarının, her

geçen gün artan zaman ve maliyet baskısı ve standartları yükselen güvenlik önlemleri, günümüz mühendislerini saha denemelerine geçmeden önce bir dizi ön test yapmaya zorlamaktadır.

Örneğin, otomobil üreticileri, yukarıda bahsedilen güvenlik standartları nedeni ile tasarım ve üretim sırasında yaptıkları test çalışmalarını olabildiğince geniş kapsamlı tutmak zorundadırlar. Hanselmann [7], geleneksel yöntemlerin; yapılan prototip araçların deneme sürüşleri yada gerçek şartların sağlandığı geniş test bantları ile, üretici firmaları ekonomik baskılar ve zaman faktörleriyle zorlayan, ayrıca tasarımcı için de yavaş gelişen ve hiç de esnek olmayan çalışmalar olduklarını belirtmiştir. Aynı zamanda, belirli şartların yapılan her testte aynı şekilde sağlanamamasının kontrolcü geliştirmede ek handikaplar doğurduğunu ifade etmektedir.

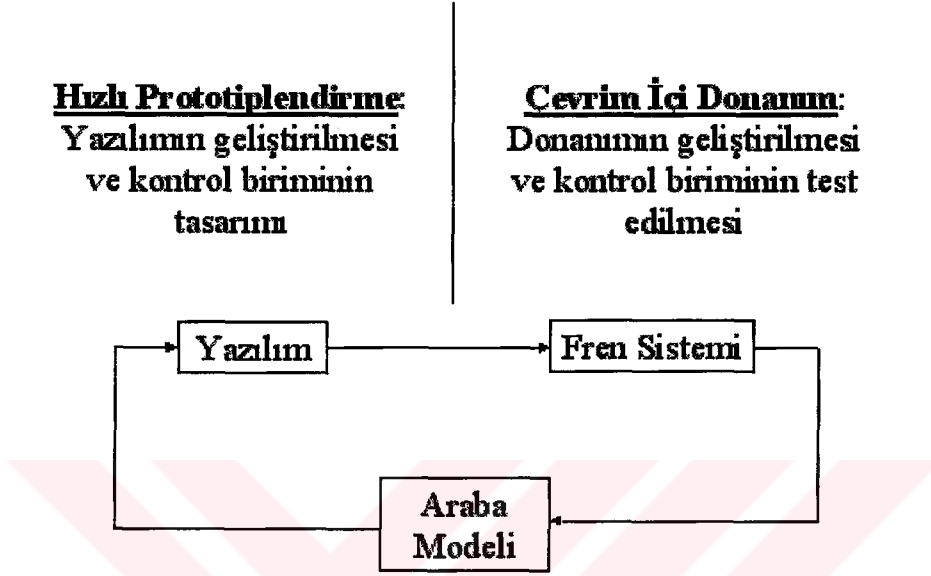
Çevrim içi donanım içeren simülasyonlar, sanal ortamda yapılacak sistem testlerinde en güvenilir ve düşük maliyetli yöntemlerden biridir. Yapılan denemeler sırasında hemen hemen her tipteki çevre koşulu matematik modeller haline dönüştürülerek kapalı çevrim içine yerleştirilebilmektedir. Zaman ve maliyet tasarrufunu artırabilmek adına kolayca çoğaltılabilen, yeniden üretilen veya otomatikleştirilebilen bir yöntem olan çevrim içi donanım içeren simülasyon, kısa zamanda geliştirme sürecinin ayrılmaz bir parçası halini almıştır.

Hanselmann [7], çevrim içi donanım içeren simülasyonların başlıca avantajlarını aşağıdaki gibi sıralamıştır.

- Kontrolcü geliştirme çalışmaları sırasında donanım içermeyen tasarım ve simülasyonlar için geliştirilmiş sistem modelleri, elektronik kontrol birimlerinin üretim testleri sırasında yeniden kullanılabilir.
- Sistem yazılımında yapılan her değişimin tasarım ile tutarlılığı incelenebilir.
- Kontrolcü tasarımı aşamasında kullanılan test yazılımı ve toplanan test verileri tekrar kolayca kullanılabilir.

Çevrim içi donanım simülasyonları tasarımcıya çeşitli imkanlar sunabilmektedir. Örneğin; Şekil 2.4'teki fren sistemi [8] gibi tüm otomobil sisteminin bir parçası ve genelde diğer tüm alt sistemler yeri geldiğinde kapalı çevrim içine, davranışlarını modelleyen gerçek zaman simülasyonları olarak girerken çevrimdeki tek gerçek parça elektronik kontrol devresinin kendi olabilir. Bazı durumlarda kontrol edilecek

parça yerine konacak matematik modelin hassas davranışından kaçınmak ya da bilhassa parçanın kendisini test etmek amacı ile çevrim içinde mekanik olarak yerini almaktadır. Tasarımcının isteğine göre, kimi durumlarda da kontrol biriminin kendisi çevrim dışı bırakılarak fren sistemi gibi kontrol edilecek sistemlerin simule edilmiş ortamdaki davranışları incelenebilmektedir.



Şekil 2.4 Çevrim İçi Donanım ve Hızlı Prototiplendirme Sistemleri

Tasarımcının yapacağı bu ayırım, aslında hızlı kontrolcü prototiplendirme ve çevrim içi donanım simülasyonları arasındaki temel farkı meydana getirmektedir. Bu iki sistem arasındaki bir diğer fark da Hanselmann [7], tarafından I/O yönlerindeki ters kurgu ve çevrim içi donanım simülasyonlarının, kontrolcü sistemlerin aksine gerçek zamanda çalışan modellere ihtiyaç duyması olarak açıklanmıştır.

2.2.1 WINCON Çevrim İçi Donanım İçeren Simülasyon Sistemi

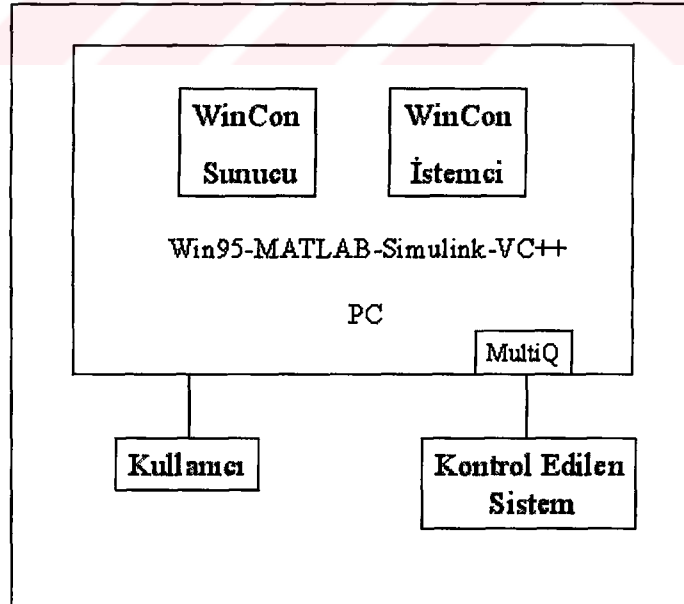
WinCon [9], MATLAB/Simulink blok diyagramları ile oluşturulan kontrolcü sistemi, bir PC üzerinde Real-Time Workshop kullanarak gerçek zamanda çalışan bir Windows 95 uygulamasıdır. Wincon yapısında bir sunucu ve istemci içermektedir. Her sunucu birden çok istemci ile, yine aynı şekilde her istemci de birkaç sunucu ile iletişim kurabilmektedir. Bir PC aynı zamanda hem sunucuyu hem de istemciyi aynı anda bünyesinde barındırabilmektedir. Bu da ikinci bir donanıma ihtiyaç duymadan bir kontrol çevrimi kurulmasına olanak vermektedir. Sunucu ve istemcinin uygulama alanları şu şekilde özetlenebilir.

Sunucu:

- Simulink blok diyagramlarının Real-time Workshop kullanarak C++ kodlarına dönüştürülmesi
- Bu kodların Visual C++ kullanılarak derlenmesi ve bağlantı oluşturulması
- Tasarlanan kodların istemci üzerinden çalıştırılması için aktarılması
- İstemci ile TCP/IP iletişimi gerçekleştirilmesi
- Simulink ve istemci arasındaki iletişimi kurarak gerçek zamanda parametre ve kazanç ayarlamalarına imkan tanınması
- Gerçek zamanda arzu edilen istemci üzerinden veri aktarımı, bu verilerin grafiksel hale dönüştürülmesi ve saklanması.

İstemci:

- Kontrolcü kodunun sunucudan alınması
- Bu kodun gerçek zamanda çalıştırılması
- Arzu edilen durumlarda sunucuya veri aktarımı



Şekil 2.5 WinCon Çevrim İçi Donanım İçeren Simülasyon Sistemi [9]

Şekil 2.5'teki sistem WinCon'un [9] en basit ve en çok uygulanan konfigürasyonu tek PC ile kurulan dolayısıyla bilgisayar ağı içermeyen düzendedir. Gerekli donanım ve yazılımı içeren bir PC ile aynı zamanda hem sunucuyu hem de istemciyi çalıştırarak gerçek zamanda kontrol, ayarlama ve gözlemlene yapılabilmektedir. WinCon bunun dışında çeşitli bağlantı biçimleri içeren farklı konfigürasyonlar da sunmaktadır.

- İki PC doğrudan bağlantı
- İki PC internet bağlantısı
- Bir sunucu çoklu istemci internet bağlantısı
- Çoklu sunucu ve çoklu istemci internet bağlantısı.

Wincon 'un sağladığı ve bundan sonraki çalışmanın parametrik temelini oluşturacak ters sarkaç sisteminin deney düzeneği, bir model araba ve arabanın ön tarafında bulunan ve serbest olarak salınabilen sarkaç kolundan oluşmaktadır. Arabanın üzerinde iki adet kodlayıcı bulunmaktadır. Bunlardan birincisi rayla temas eden dişli bir pinyon yardımı ile arabanın pozisyonunu ölçerken, diğeri sarkaç kolunun rotasyonel hareketini kodlamaktadır. Sistem bu kodların uygun birimlere dönüştürülmesi ve aktarılması işini, aynı firma tarafından üretilmiş MultiQ-3 kartı ile yapmaktadır. 8 adet analog giriş/çıkış ve 16 dijital giriş/çıkış kanalına sahip bu kartın yerleştirildiği PC, WinCon programı yardımı ile mekanik sistemin kontrolünü sağlamaktadır.

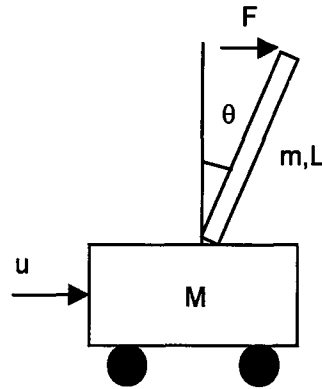
Tezin bundan sonraki bölümlerinde böyle bir ters sarkaç sistemi için kontrolcü tasarımı çalışmaları yer almaktadır. Bölüm 3'te kontrolcü tasarımı için durum geri beslemesi ve lineer kuadratik regülatör metotları kullanılmıştır. Dördüncü bölümde doğrusal olmayan bir kontrolcü olan bulanık mantık kontrolcüsü tasarlanarak Simulink'te oluşturulan ters sarkaç simülasyon modeli yardımı ile bir dizi simülasyon gerçekleştirilmiştir.

3. TERS SARKACIN SİMÜLASYON MODELİNİN OLUŞTURULMASI

Günümüzde, modern kontrol teorisi, pratik kontrol sistemleri oluşturacak özgün tasarım tekniği sağlayabilecek niteliktedir. Bunun yanında, her geçen gün, daha hızlı ve güçlü donanımla birlikte gelişen yazılım paketleri ile karmaşık kontrolcü tasarımlarında, hesaplama ve programlama yoğunluğu en aza indirilebilmektedir.

Bu bölümde, durum geri beslemesi ve lineer kuadratik optimal kontrol metotları kullanılarak ters sarkaç için kontrolcü tasarımı gerçekleştirilmeye çalışılacaktır. Ters sarkaç sisteminin kararlılık problemi [17], [18], [19] ve [20] gibi daha pek çok araştırmanın esas konusu olmuştur. Bu özel uygulama, motor sürücüsü ile kumanda edilen bir model araba üzerine monte edilen ters sarkaç kolunun, arabanın konumunu kontrol ederek, denge konumunda tutulmasıdır. Bu uygulamanın uzay araçları, füze ve roketlerdeki iticilerin kalkış sırasında yükseklik kontrolünün basit bir modelini oluşturduğu düşünülebilir. Yükseklik kontrolü probleminde hedef, iticilerin, değişen koşullarda istenilen konumu yada rotayı takip etmelerini sağlamaktır.

Ters sarkaç, kararsız denge dinamiğine sahip olduğu için uygun kontrol kuvveti uygulanmadığı sürece her an düşmesi kaçınılmaz sonuçtur. Bu uygulamada, ters sarkacın sadece sayfa düzleminde hareket edebildiği iki boyutlu bir sistemle ilgilenilecektir (Şekil 3.1). Kontrol kuvveti 'u' arabaya uygulanmaktadır. Sarkaç kolunun ağırlık merkezinin geometrik merkezle aynı yerde olduğunu kabul edilecektir.



Şekil 3.1 Ters Sarkaç Sistemi

$$m \frac{d^2}{dt^2} (x + l \sin\theta) = H + F \quad (3.2)$$

kolun dikey hareketi;

$$m \frac{d^2}{dt^2} (l \cos\theta) = V - mg \quad (3.3)$$

arabanın yatay hareketi ise;

$$M \frac{d^2 x}{dt^2} = u - H \quad (3.4)$$

şeklindedir.

(3.1), (3.2), (3.3), (3.4) denklemleri ters sarkaç sisteminin hareket denklemleridir. Denklemler, sinüs ve kosinüs'lü terimler içerdiğinden lineer değildirler. Eğer θ açısının küçük aralıklarda değiştiği kabul edilirse, $\cos\theta \approx 1$ ve $\sin\theta \approx \theta$ kullanılarak denklemler aşağıdaki şekilde lineerleştirilebilir.

$$I\ddot{\theta} = V l \theta - H l + F l \quad (3.5)$$

$$m(\ddot{x} + l\ddot{\theta}) = H + F \quad (3.6)$$

$$0 = V - mg \quad (3.7)$$

$$M\ddot{x} = u - H \quad (3.8)$$

(3.6) ve (3.8) denklemleri birbirine eklenirse

$$(M + m)\ddot{x} + m l \ddot{\theta} = u + F \quad (3.9)$$

elde edilir. (3.5) den (3.7) denkleminin çıkarılması ve H için (3.6) denkleminin kullanılmasıyla

$$I\ddot{\theta} = m l g \theta + (F - H) l$$

$$I\ddot{\theta} = m l g \theta + 2 F l - m \ddot{x} l - m l^2 \ddot{\theta}$$

veya

$$(I + m l^2) \ddot{\theta} + m l \ddot{x} = m g l \theta + 2 F l \quad (3.10)$$

elde edilir. (3.9) ve (3.10) denklemleri ile ters sarkaç sisteminin hareketi tanımlanabilmektedir.

3.2 Pol Ataması Tekniği İle Durum Geri Besleme Kontrolcü Tasarımı

Bu bölümde amaç, sarkaç kolunun her türlü dış etki karşısında dik konumda denge halinde tutabilecek bir kontrolcü oluşturabilmektir. Bunun için de herhangi bir dış etki ile salınıma geçen sarkaç kolu arabaya uygulanan 'u' kontrol kuvveti ile dik konuma getirilecektir. Aynı zamanda arabanın da başlangıç konumuna ($x = 0$), en kısa zamanda (%2'lik tolerans bandı ile yerleşme zamanı: $t_s = 2$ sn), kabul edilebilir bir sönümlleme ile (örneğin: kapalı çevrim karakteristik denklem kökleri $\zeta = 0.5$ sönümlleme faktörüne sahip olsun) gelebileceği bir kontrol sistemi tasarlanacaktır.

Sarkaç kolunun uygulama sonunda alacağı konum $\theta_d = 0^\circ$ olmalıdır. Bu uygulamada Ogata'nın [11] aktardığı tarzda, kontrolcü pol ataması tekniği ile durum geri besleme kontrol metodu kullanılarak tasarlanacaktır. Burada pol ataması için gerekli olan şart sistemin tamamen durum kontrol edilebilirliği (Complete State Controllable). Bu durumda ilk adım ters sarkaç sisteminin matematik modelinin çıkartılması olacaktır.

' θ ' sarkacın dik eksenle olan açısının küçük aralıklarla değiştiği kabul edilmişti. Bu durumda sistemin denklemleri hatırlanırsa:

$$(M + m) \ddot{x} + m l \ddot{\theta} = u + F \quad (3.11)$$

$$(I + m l^2) \ddot{\theta} + m l \ddot{x} = m g l \theta + 2 F l \quad (3.12)$$

Bu analizde sarkaç kolunun atalet momentini çok küçük bir değere sahip olduğu ve denkleme toplam artımı olarak girdiği için $I = 0$ olarak kabul edilmektedir. Bu durumda (3.12) denklemi;

$$m l^2 \ddot{\theta} + m l \ddot{x} = m g l \theta + 2 F l \quad (3.13)$$

haline dönüşür.

(3.11) ve (3.13) denklemleri şu şekilde düzenlenebilir:

$$M I \ddot{\theta} = (M + m)g\theta - u + \left(\frac{2M}{m} + 1\right)F \quad (3.14)$$

$$M \ddot{x} = u - m g \theta - F \quad (3.15)$$

(3.14) denklemi $\ddot{x}$ 'i (3.11) ve (3.13) denklemlerinden eleyerek elde edilirken, (3.15) de aynı şekilde $\ddot{\theta}$ elenerek bulunmuştur.

(3.14) denklemi ile sistemin araba kuvvet girişi ve bozucu etken girişi ile açısal konum arasındaki transfer fonksiyonları oluşturulacaktır:

$$\frac{\theta(s)}{-U(s)} = \frac{1}{MIs^2 - (M + m)g} \quad (3.16)$$

$$\frac{\theta(s)}{F(s)} = \frac{\left(\frac{2M}{m} + 1\right)}{MIs^2 - (M + m)g} \quad (3.17)$$

model parametreleri olarak verilen sayısal değerler yerine yerleştirilirse:

$$\frac{\theta(s)}{-U(s)} = \frac{1}{s^2 - 40.545} = \frac{1}{s^2 - (6.367)^2}$$

Ters sarkaç reel eksenin solunda ve sağında $s_{1,2} = \pm 6.367$ ile verilen köklere sahiptir. Bu durumda oluşturulan sistem $s_1 = +6.367$ polünden dolayı kararsız bir açık çevrime sahiptir. Şimdi sistemin durum uzay modelini oluşturmak için durum değişkenleri belirlenecektir:

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

$$x_3 = x$$

$$x_4 = \dot{x}$$

θ açısı sarkaç kolunun bağlantı noktası etrafında dönmesini belirtirken, x de arabanın konumunu ifade etmektedir. Bu iki büyüklük kolay izlenip, ölçülebilir olmaları nedeniyle sistemin çıkış değerleri olarak alınmışlardır.

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

Durum değişkenlerinin tanımı ve (3.14), (3.15) denklemlerinin yardımı ile aşağıdaki denklemler elde edilmiştir:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{M+m}{Ml}gx_1 - \frac{1}{Ml}u + \frac{\left(\frac{2M}{m}+1\right)}{Ml}F$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -\frac{m}{M}gx_1 + \frac{1}{M}u - \frac{F}{M}$$

Bu denklemleri matris formunda yazarak;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{M+m}{Ml}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{m}{M}g & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{\left(\frac{2M}{m}+1\right)}{Ml} \\ 0 \\ -\frac{1}{M} \end{bmatrix} F \quad (3.18)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (3.19)$$

elde edilir. (3.18) ve (3.19) denklemleriyle sistemin durum-uzay modeline ulaşılmıştır. Modele ait sayısal değerleri de yerlerine koyarak:

$$\dot{x} = Ax + Bu + EF$$

$$y = Cx$$

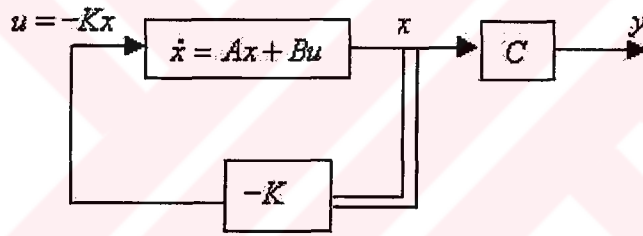
Buradan;

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 40.545 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -2.527 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -4.022 \\ 0 \\ 1.226 \end{bmatrix}, E = \begin{bmatrix} 0 \\ 35.248 \\ 0 \\ -1.226 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

olarak hesaplanmıştır.

Oluşturulan bu durum uzay modeline sahip sistemde şimdi durum geri-besleme (state variable feedback) tasarımı yapılacaktır.

Sistemin kontrol büyüklüğü: $u = -Kx$



Şekil 3.3 Durum Geri Besleme Blok Diyagramı

Bunun için öncelikle sistemin tamamen kontrol edilebilirliği (c.s.c) incelenmelidir. Eğer oluşturulacak M matrisinin rankı: $n = 4$ ise sistem kontrol edilebilir bir sistemdir.

$$M = [B \quad AB \quad A^2B \quad A^3B] = \begin{bmatrix} 0 & -4.022 & 0 & -162.733 \\ -4.022 & 0 & -162.733 & 0 \\ 0 & 1.227 & 0 & 10.168 \\ 1.227 & 0 & 10.168 & 0 \end{bmatrix} \quad (3.20)$$

$$\text{Rank}(M) = n = 4 \quad (\text{c.s.c})$$

Sistemin karakteristik denklemi çıkartılırsa:

$$|sI - A| = \begin{vmatrix} s & -1 & 0 & 0 \\ -40.545 & s & 0 & 0 \\ 0 & 0 & s & -1 \\ 2.527 & 0 & 0 & s \end{vmatrix}$$

$$= s^4 - 40.545s^2$$

$$= s^4 + a_1s^3 + a_2s^2 + a_3s + a_4 = 0$$

Bu durumda; $a_1 = 0$, $a_2 = -40.545$, $a_3 = 0$, $a_4 = 0$

Şimdi kapalı çevrim karakteristik denklem köklerini kompleks düzlemde tasarımın başında belirlenen şartlara uygun yerleştirilirse: Sistemin %2'lik yerleşme zamanı süresinde ($t_s = 2$ sn.) diferansiyel denklemi ikinci derece olan sisteme en uygun sönümlleme faktörü ile ($\zeta = 0.5$) istenen kapalı çevrim kökleri: $s = \mu_i$ ($i = 1,2,3,4$)

$$\mu_1 = -2 + j2\sqrt{3}, \quad \mu_2 = -2 - j2\sqrt{3}, \quad \mu_3 = -10, \quad \mu_4 = -10$$

Bu durumda μ_1, μ_2 dominant kapalı çevrim kökleridir ($\zeta = 0.5$ ve $\omega_n = 4$). Diğer iki kök dominant köklerden uzağa yerleştirilmiştir. Böylece sistem cevabına etkileri azaltılmıştır. Sistemin hızı ve sönümlemesi uygun hale getirilmiştir. Sonuçta istenen karakteristik denklem şu hale dönüşmüştür:

$$(s - \mu_1)(s - \mu_2)(s - \mu_3)(s - \mu_4) = (s + 2 - j2\sqrt{3})(s + 2 + j2\sqrt{3})(s + 10)(s + 10)$$

$$= (s^2 + 4s + 16)(s^2 + 20s + 100)$$

$$= s^4 + 24s^3 + 196s^2 + 720s + 1600$$

$$= s^4 + \alpha_1s^3 + \alpha_2s^2 + \alpha_3s + \alpha_4 = 0$$

Denklemin katsayıları:

$$\alpha_1 = 24, \quad \alpha_2 = 196, \quad \alpha_3 = 720, \quad \alpha_4 = 1600$$

Durum geri besleme kazancı olan K matrisi aşağıda tanımlanmıştır:

$$K = [\alpha_4 - a_4 \quad \alpha_3 - a_3 \quad \alpha_2 - a_2 \quad \alpha_1 - a_1] T^{-1}$$

Burada T matrisi : $T = MW$ olarak tanımlanmıştır.

$$W = \begin{bmatrix} a_3 & a_2 & a_1 & 1 \\ a_2 & a_1 & 1 & 0 \\ a_1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -40.545 & 0 & 1 \\ -40.545 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad (3.21)$$

İken, T matrisi ve tersi T^{-1} matrisi aşağıda hesaplanmıştır:

$$T = MW = \begin{bmatrix} 0 & -33.044 & 0 & 0.815 \\ -33.044 & 0 & 0.815 & 0 \\ 0 & 0.815 & 0 & 0 \\ 0.815 & 0 & 0 & 0 \end{bmatrix}$$

ve

$$T^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1.227 \\ 0 & 0 & 1.227 & 0 \\ 0 & 1.227 & 0 & 49.748 \\ 1.227 & 0 & 49.748 & 0 \end{bmatrix}$$

Bu durumda istenen geri besleme kazanç matrisi:

$$K = [1600 - 0 \quad 720 - 0 \quad 196 + 40.545 \quad 24 - 0] T^{-1}$$

$$= [1600 - 0 \quad 720 - 0 \quad 196 + 40.545 \quad 24 - 0] \begin{bmatrix} 0 & 0 & 0 & 1.227 \\ 0 & 0 & 1.227 & 0 \\ 0 & 1.227 & 0 & 49.748 \\ 1.227 & 0 & 49.748 & 0 \end{bmatrix}$$

$$K = [-71.141 \quad -11.530 \quad -40.543 \quad -18.224]$$

Kontrol sinyali 'u' ise;

$$u = -Kx = 71.141x_1 + 11.530x_2 + 40.543x_3 + 18.224x_4 \quad (3.22)$$

Oluşturulan bu sistemin bir regülatör sistem olduğunu göz önünde bulundurulmalıdır. Sarkaç kolunun istenen denge konumu $\theta_d = 0^\circ$ ve arabanın istenen konumu $x_d = 0$ olarak belirlenmiştir. Yani referans giriş değerleri her zaman sıfırdır. Yukarıda elde edilen durum geri besleme kazanç matrisi Ogata'nın [12] tasarlamış olduğu MATLAB® programı, sisteme göre düzenlenerek kolayca elde edilmiştir.

3.2.1 Durum geri besleme kazanç matrisinin MATLAB® programı ile elde edilmesi

Karakteristik denklemin köklerini kompleks düzlemde istenilen şekilde yerleştirilmesini sağlayan durum geri-besleme kazanç matrisi bir MATLAB® programı yardımı ile elde edilmeye çalışılacaktır:

```
%----Ters sarkaç kontrol sistemi tasarımı----

M = 0.815;
m = 0.21;
g = 9.81;
l = 0.305;

% A, B, C ve D matrisleri
A = [0 1 0 0;
      ((M+m)/(M*l))*g 0 0 0;
      0 0 0 1;
      -(m/M)*g 0 0 0];
B = [0; -1/(M*l);
      0; 1/M];
C = [1 0 0 0; 0 0 1 0];
D = [0;0];
eig(A)

% Sistemin Kontrol edilebilirliği
M=[B A*B A^2*B A^3*B]
rank(M)

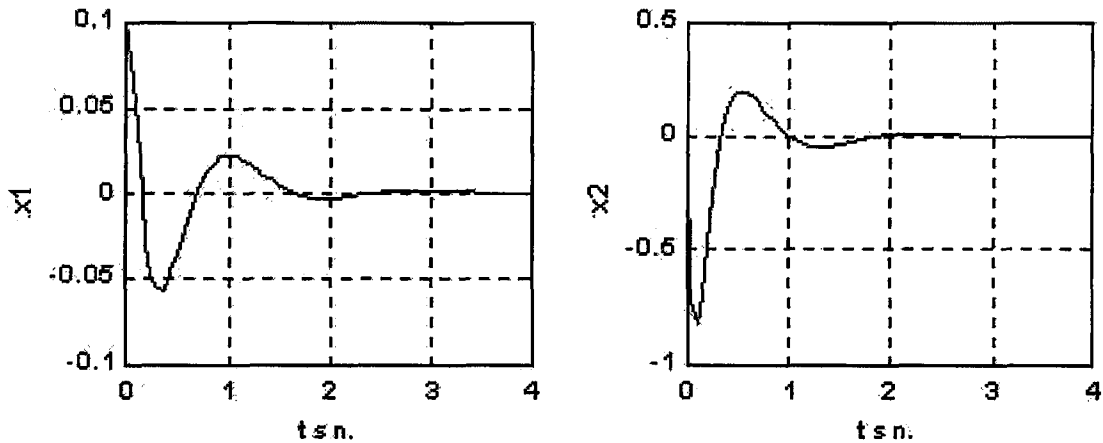
% Eğer M matrisinin rankı n = 4 ise sistem kontrol edilebilir
% pol ataması tekniğini kullanmak mümkün olacak
JA = poly(A)
a1 = JA(2);
a2 = JA(3);
a3 = JA(4);
a4 = JA(5);
W = [a3 a2 a1 1;a2 a1 1 0;a1 1 0 0;1 0 0 0];
T = M*W;
% İstenen karakteristik denkleme J matrisini tanımlayarak
% ulaşılabacaktır.
% ζ=0.5
% ts = 2 sec.
J = [-2+2*sqrt(3)*i 0 0 0;
      0 -2-2*sqrt(3)*i 0 0;
      0 0 -10 0;
      0 0 0 -10];
```

```

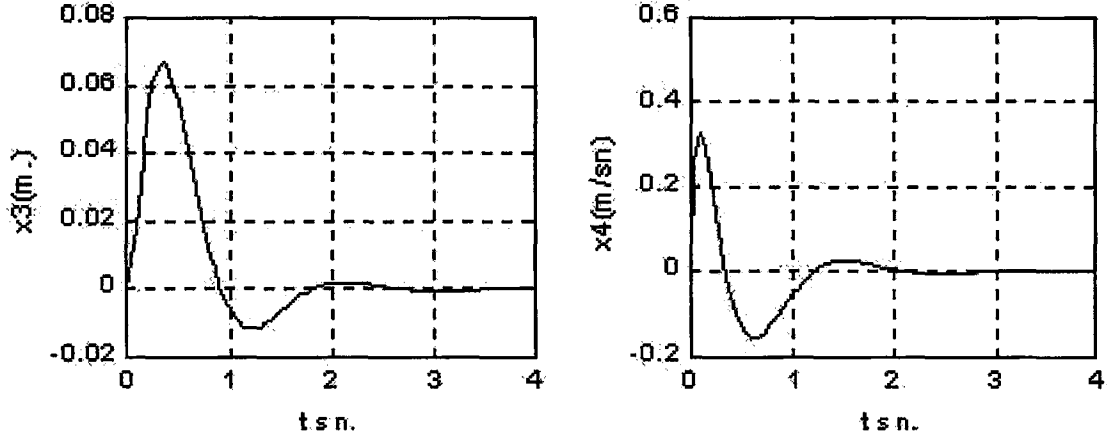
JJ = poly(J)
aa1 = JJ(2);
aa2 = JJ(3);
aa3 = JJ(4);
aa4 = JJ(5);
% Durum geri besleme kazanç matrisi K şu şekilde ifade
% edilebilir:
K=[aa4-a4 aa3-a3 aa2-a2 aa1-a1]*(inv(T))
%----İlk şartlara sistemin cevabı----
AA=A-B*K
% İlk şartlar: BB
BB=[0.1;0;0;0];
t=0:0.05:4;
[x,z,t] = step(AA,BB,AA,BB,1,t);
x1 = [1 0 0 0]*x';
x2 = [0 1 0 0]*x';
x3 = [0 0 1 0]*x';
x4 = [0 0 0 1]*x';

% Sistem cevabının eğrilerinin çizimi
subplot(2,2,1);
plot(t,x1);grid
title ('x1(teta) & t')
xlabel ('t sn.')
ylabel ('x1')
subplot(2,2,2);
plot(t,x2);grid
title ('x2(teta üssü) & t')
xlabel ('t sn.')
ylabel ('x2')
subplot(2,2,3);
plot(t,x3);grid
title ('x3(araba konumu) & t')
xlabel ('t sn.')
ylabel ('x3(m.)')
subplot(2,2,4);
plot(t,x4);grid
title ('x4(araba hızı) & t')
xlabel ('t sn.')
ylabel ('x4(m/sn)')

```



Şekil 3.4 θ ve $\dot{\theta}$, Durum Geri Besleme Kontrolcüsü ile Kontrol



Şekil 3.5 Araba Konum ve Hızı, Durum Geri Besleme Kontrolcüsü ile Kontrol

Sarkaç kolu için ilk şart olarak $\theta(0) = 0.1\text{rad}$ verilmiştir. Şekil 3.4 ve 3.5 de oluşturulan sistemin ilk şart girişine verdiği cevap grafiklerinde sistemin çıkışları olan sarkaç kolunun açı değişimi ve arabanın konumu referans değerlerine ($\theta_d = 0$ ve $x_d = 0$) istenilen %2'lik yerleşme zamanına yakın bir değere ($t_s \cong 3 \text{ sn.}$) ulaştıkları görülmektedir.

Sistem uygun geri besleme kazanç katsayısı olan K matrisi sayesinde kararsız konumdan kararlı hale gelmiştir. Sistemin referans girişleri her zaman sıfır olarak alındığı için programda gösterilmemiştir.

3.2.2 Durum Geri Besleme Kazancının Ackermann Formülü ile Elde Edilmesi

Bir önceki bölümde kullanılan pol ataması yaklaşımı $n = 2$ veya 3 için uygun bir çözüm yöntemidir ancak $n = 4, 5, 6, \dots$ olması halinde sonucun hesaplama süresinin artması söz konusudur. Bunun için durum geri besleme kazanç matrisini tanımlarken Ackermann formülünden yararlanılacaktır. Burada yapılması gereken $\Phi(A)$ matrisinin karakteristik denklemini 'polyvalm(poly(J),A)' matlab komutu ile elde etmektir. Şimdi oluşturulan MATLAB® programı bu şekilde düzenlenecektir:

İstenen karakteristik denklem J matrisinin hesaplanması ile bulunduktan sonra bu kez sistem denklemini yeniden düzenleyerek;

$$\dot{x} = (A - BK)x$$

$$\Phi(A) = A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I = 0$$

olarak ifade edilen Ackermann formülü ile $\Phi(A)$ matrisinin karakteristik denklemini hesaplayacak matlab kodu programa eklenir.

```
%----Ackermann formülü ile  $\Phi$  karakteristik denklemi----  
phi = polyvalm(poly(J),A);  
%--- Durum geri besleme matrisi K :  
K = [0 0 0 1]*(inv(M))*phi
```

Her iki yöntemle bulunan sistemin geri besleme kazanç matrisi **K** aynı değerdedir, çünkü sisteme uygulanan kontrolcü değişmemiştir.

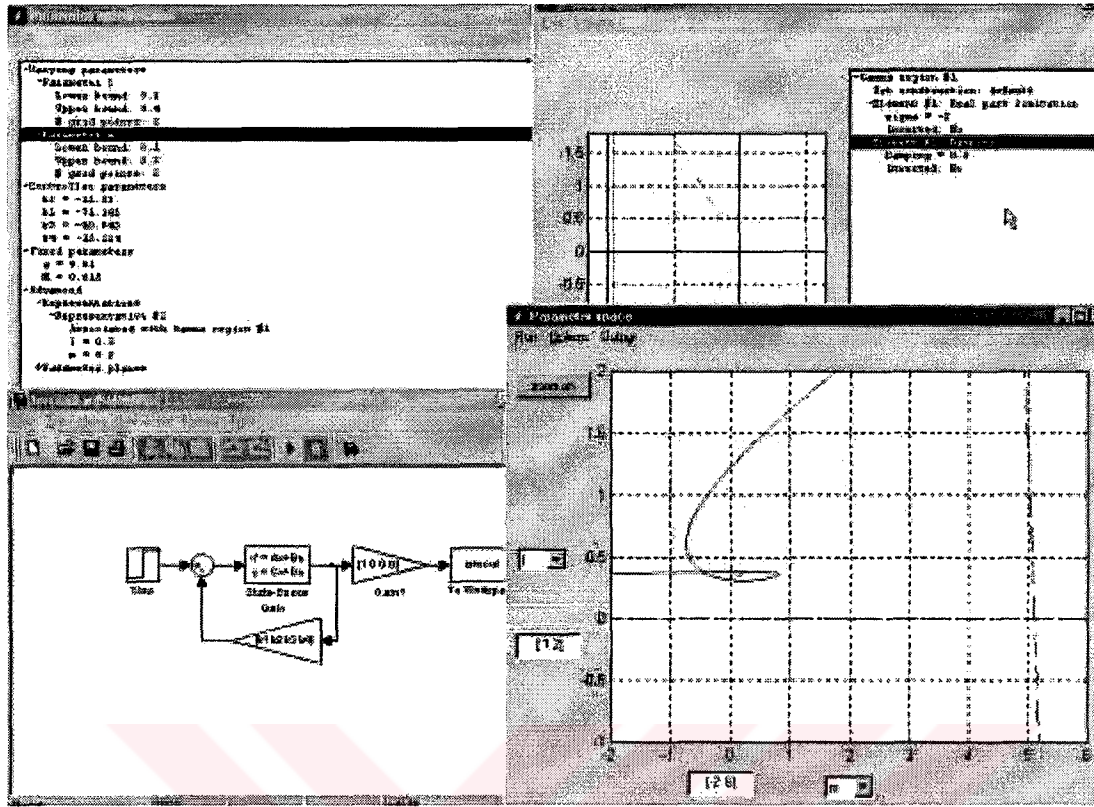
$$K = [-71.141 \quad -11.530 \quad -40.543 \quad -18.224]$$

3.3 Ters Sarkaç Modelinin ve Kontrolcüsünün Gürbüzlük Analizi

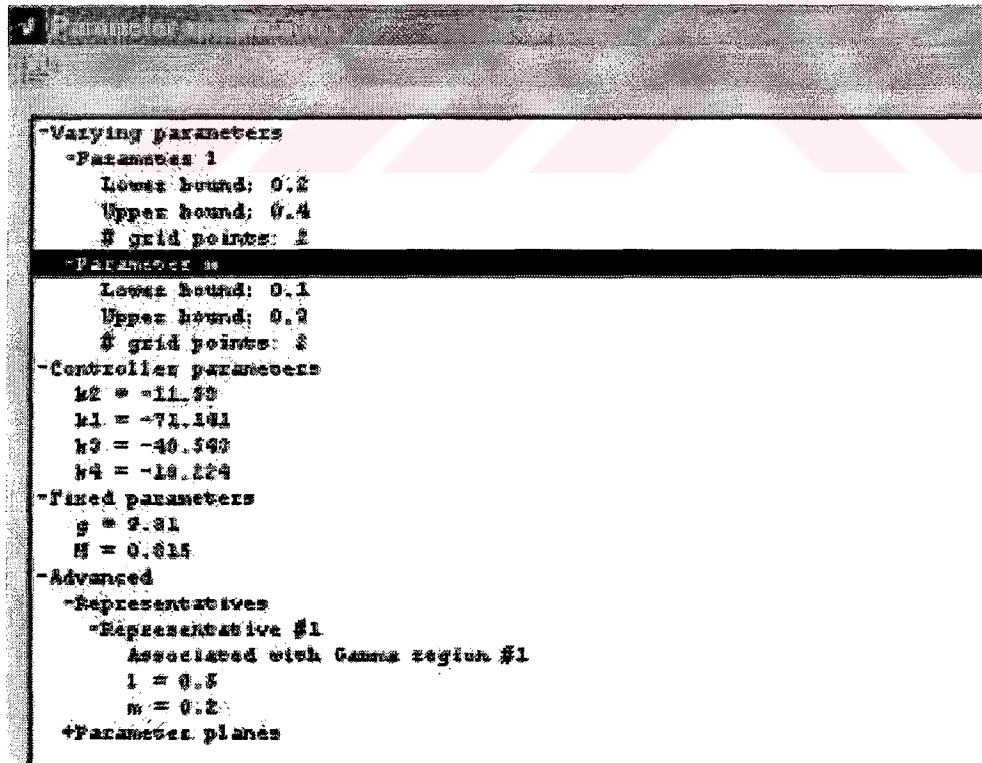
Geliştirilen kontrolcünün iyileştirilmesi adına sistemin gürbüzlüğü incelenebilir. Bu analizde temel olarak araba kütlesi (M) ve yerçekimi ivmesi (g) sabit tutulurken, sarkaç kol boyu (L) ve ağırlığı (m) değişken parametreler olarak seçilebilir. Sarkaç kolu kütlesi olan m sarkaç kol boyuyla doğru orantılıdır. Böylece tasarlanan kontrolcünün m ve L 'nin hangi değerleri için kararlı olduğu ve istenen performansı (seçilen sönümlleme değeri ve yerleşme zamanı) sağladığı çıkartılmış olacaktır.

Ayrıca sistem için tasarlanan kontrolcünün kazanç katsayıları değişken parametreler olarak seçilebilir. Böylece durum geri besleme kazanç matrisinin parametreleri olan $K(1)$, $K(2)$, $K(3)$ ve $K(4)$ 'ün hangi aralıklarda değişebileceği elde edilebilir.

Bu analiz için tasarımcıya parametre uzayında kararlı parametreler kümesini belirleyebilme imkanı veren parametre uzayı yaklaşımı kullanılmıştır. Bunun için yine MATLAB altında çalışan PARADISE isimli toolbox'tan yararlanılmıştır. Öncelikle tasarım sonucu elde edilen ters sarkaç sistemi Simulink blok diyagramları kullanılarak modellenmiştir. Şekil 3.6 da gösterilen blok diyagramda sistem bir durum uzay bloğu içerisine parametrik olarak yerleştirilmiştir. Bu blok dördüncü dereceden lineerize edilmiş ters sarkaç modelinden oluşmuştur ve durum uzay matrisleri (A , B , C ve D) bu bloktaki yerlerini almışlardır. Bu sistemin çıkışı, vektör halindeki durum geri besleme kazancı ($K = [K1 \ K2 \ K3 \ K4]$) ile çarpılarak geri beslenmiştir.



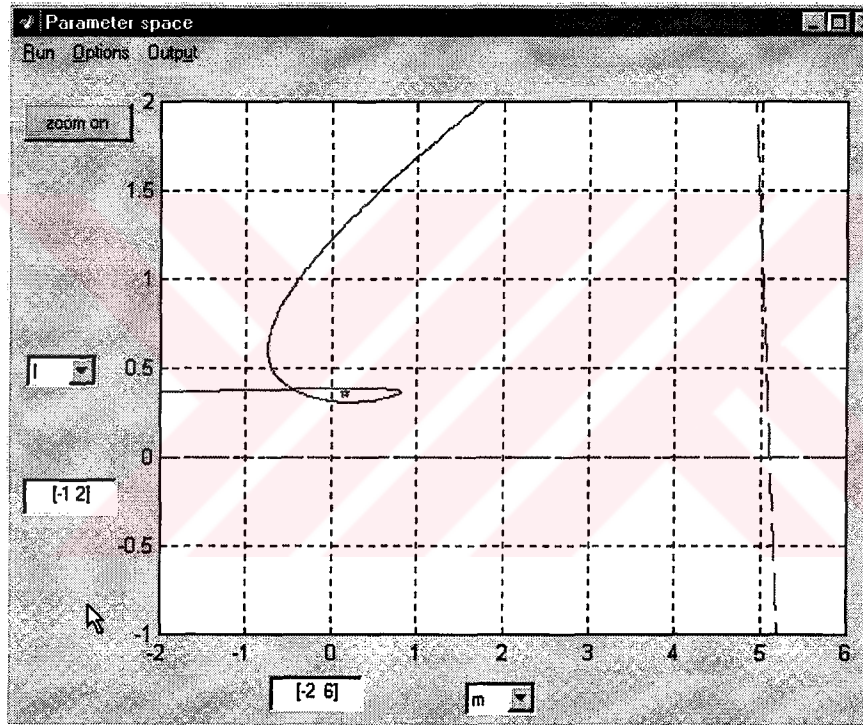
Şekil 3.6 Ters Sarkaç Sistemi İçin L ve m Değişken Parametreleri ile Gürbüzlük Tasarımı



Şekil 3.7 Paradise Programı Parametre Özellikleri Penceresi

Modele ait bütün parametreler bloğa yerleştirildikten sonra sistemin kontrol edilmek veya iyileştirilmek istenen değişken (m , L), sabit (M , g) ve kontrolcü ($K(1)$, $K(2)$, $K(3)$, $K(4)$) parametreleri saptanmıştır. Şekil 3.7'de oluşturulan koda değişken parametrelerin alt ve üst sınırları, $m = [0.2:0.4](\text{kg})$, $L = [0.1:0.2](\text{m})$ olarak belirlenmiştir.

Daha sonra, pratik uygulamalara uygun olan ve sistemin öz değerleri için yerleşme zamanı ve sönümlleme faktörü gibi kriterleri öz önüne alarak elde edilen Gamma (Γ) kararlılık bölgesi oluşturulmuştur. Son olarak Şekil 3.8'de gösterilen parametre uzayı penceresi sayesinde parametre düzleminde, tasarlanan ters sarkaç sistemi için sarkaç kolu kütlesi ve uzunluğu parametrelerinin kararlılık bölgesi elde edilmiştir.



Şekil 3.8 L ve m Değişken Parametreleri için Paradise Programı Parametre Uzayı Penceresi

Şekil 3.8'in hesaplarında karakteristik köklere ait gerçek kök sınırları ve sanal kök sınırları ayrı ayrı hesaplanmıştır. Paradise programı sayesinde bu sınırların oluşturduğu bölgelerden hangisi veya hangileri içinde sistemin Γ -kararlı olup olmadığı öğrenilebilmektedir. Tasarımcı açısından büyük kolaylık sağlayan bu fonksiyon seçenekler menüsündedir. Parametre düzlemi üzerinde bir fare tıklaması ile oluşan yeşil yıldız bu bölgenin Γ -kararlılığını göstermektedir. Şekil 3.8'deki Γ -kararlı bölge içinde negatif değerleri içeren kısım fiziksel olarak sistem tasarımı açısından bir şey ifade etmemektedir.

[illegible]

Şekil 3.9'un hesaplarında gerçek kök sınırları ve sanal kök sınırları ayrı ayrı hesaplanmıştır. Tasarlanan ters sarkaç sistemi K(1) için $[-80:-40]$ aralığında ve K(3) için $[-20:-40]$ aralığında kalan kapalı bölgede alacağı her değerde Γ -kararlılığını göstermektedir. Gürbüz kontrol sayesinde ters sarkaç modelinin sarkaç kolunun alacağı farklı uzunluk ve buna bağlı ağırlık değerleri için davranışı incelenebilirken aynı zamanda sisteme uygulanan kontrolcü kazançlarının alabileceği değerlerin kararlılık sınırları belirlenmiştir.

3.4 Lineer Kuadratik Regülatör Tasarımı

Çalışmadaki bir sonraki adımda, sistemin tam durum geri beslemeye sahip olduğu varsayılarak (sahip olunan dört durum değişkeninin de ölçülebildiği düşünülüp) oluşturulmuş, sistem için optimal kontrol yapma imkanı sağlayan temel esasları Anderson ve Moore'den [13] alınmış Lineer Kuadratik Regülatör tasarımı yapılmıştır. Sonuçta kuadratik performans indekslerine dayanan kararlı bir kontrol sistemi elde edilmek istenmektedir.

Kontrol edilmek istenen sistem basitçe tanımlanacak olursa;

$$\dot{x} = Ax + Bu \quad (3.22)$$

Burada; x = durum vektörü (n boyutlu vektör)

u = kontrol vektörü (r boyutlu vektör)

A = $n \times n$ sabit matrisi

B = $r \times n$ sabit matrisidir.

Bir kontrol sistemi tasarlanırken ilgilenilen nokta, performans endeksi minimize edilmiş bir ' $u(t)$ ' kontrol vektörünün seçilebilmesidir. Bu durumda aşağıdaki gibi tanımlanan, kuadratik bir performans endeksini asgari değerde tutan $u(t) = -Kx(t)$ şeklindeki lineer bir kontrolcüdür. Aşağıda $L(x,u)$ x ve u 'nun kuadratik bir fonksiyonudur.

$$J = \int_0^{\infty} L(x,u)dt$$

Buradan da anlaşılacağı gibi kuadratik performans endeksine dayalı optimal regülatör sistemlerinin tasarımı, K matrisinin elemanlarının gerekli optimizasyon probleminin çözümü sonucu belirlenmesiyle gerçekleştirilmektedir [14]. Bu tasarımı yaparken karşılaşılan bir zorluk da, Ricatti denklemlerinin çözümlenmesidir. Bunun için yine bir Matlab fonksiyonu olan 'lqr' komutundan yararlanılacaktır. Bu komutla sürekli zamanda Ricatti denkleminin çözümüne ulaşılırken, aranan optimal kazanç matrisi de elde edilecektir.(3.22)de tanımlanan sistem için performans endeksi

$$J = \int_0^{\infty} (x^* Q x + u^* R u) dt \quad (3.23)$$

şeklinde. Burada Q belirli-pozitif (yada kısmi belirli-pozitif) ve R de belirli-pozitif, reel simetrik matrislerdir. Ayrıca seçilen u(t) kontrol vektörünün büyüklüğü sınırsız olarak kabul edilmiştir. Buna göre Lyapunov' un ikinci metodu gereğince performans indeksi minimize edilmiş bir lineer kuadratik optimal kontrol sistemi karardır.

Q ve R parametrelerin doğru seçimi ile optimize edilmek istenilen sistemin girişi ve durum değişkenleri arasındaki dengenin kurulması sağlanacaktır. Parametre seçimi için en basit durumla işe başlanacaktır; R = 1 iken Q matrisi her iki çıkış da kullanılarak (sarkaç kolu açısı ve arabanın konumu), Q = C'*C olarak tanımlansın. LQR metodu genelde her iki çıkışı da kontrol etme imkanı sağlar, bu ters sarkaç sisteminde de bu mümkün olmaktadır. Sistemden istenen cevabı alabilmek için, Q matrisindeki sıfırdan farklı elemanların değiştirilmesi ile kontrolcü ayarlanabilir. Öncelikle Q matrisinin yapısı çıkartılmalıdır:

$$Q = C^*C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Elde edilen bu matrisin (1,1) elemanı sarkaç kolunun açısını, (3,3) elemanı da araba konumunu etkileyen ağırlık faktörleridir. Girişin ağırlık parametresi olan R matrisi 1 olarak kalırken, uygun kontrolü sağlayacak K geri besleme matrisine ulaşabilmek için Q matrisini oluşturan sıfırdan farklı elemanlarla denemeler yapılabilir. Bunun için şimdi sistemin cevabının grafiksel değişimini incelerken K matrisinin değerinin takip edilebileceği bir MATLAB programı [19] yazılacaktır.

```
% Ters sarkaç kontrol sistemi için Lineer Kuadratik Regülatör
tasarımı
M = 0.815;
m = 0.21;
g = 9.81;
l = 0.305;
% A, B, C ve D matrisleri
A = [0 1 0 0;
      ((M+m)/(M*l))*g 0 0 0;
      0 0 0 1;
      -(m/M)*g 0 0 0];
B = [0; -1/(M*l);
      0; 1/M];
C = [1 0 0 0; 0 0 1 0];
D = [0; 0];
```

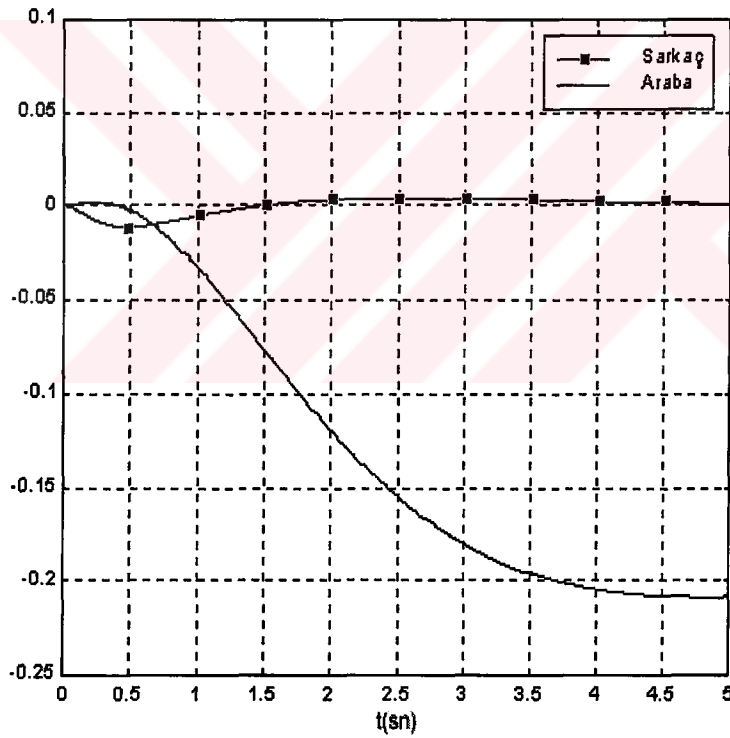
```

eig(A)
x = 1;
y = 1;
Q = [x 0 0 0;
      0 0 0 0;
      0 0 y 0;
      0 0 0 0];
R = 1;
K = lqr(A,B,Q,R)
Ac = [(A-B*K)];
Bc = [B];
Cc = [C];
Dc = [D];
T = 0:0.01:5;
U = 0.2*ones (size (T));
[Y, X] = lsim(Ac,Bc,Cc,Dc,U,T);
plot (T,Y)
legend ('Sarkaç', 'Araba')

```

Program işletilirse:

$K = [-25.1306 \ -4.0478 \ -1.0000 \ -1.7531]$ elde edilir.

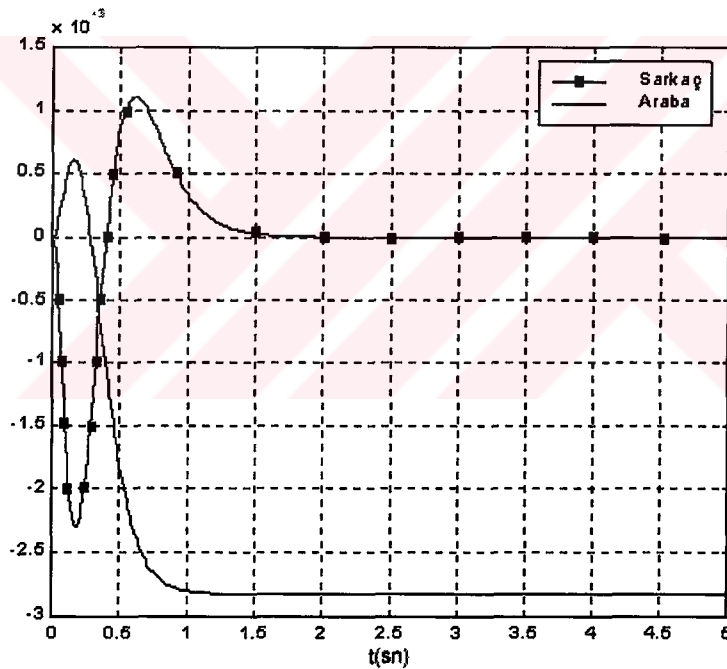


Şekil 3.10 Lineer Kuadratik Regülatör Sisteminin Basamak Giriş Cevabı ($x = 1$ ve $y = 1$)

Şekil 3.10'da sisteme u (arabaya uygulanan kuvvet) girişinden bir basamak giriş uygulanmıştır. Noktalı eğri, sarkaç kolunun açı değişimini 'radyan' olarak verirken, düz eğri ile araba konumunun 'metre' cinsinden değişimi izlenmektedir. Her iki eğride de bir miktar aşma vardır ve yerleşme zamanlarında mutlaka bir iyileştirme yapılmalıdır. Ayrıca arabanın istenilen konumdan oldukça uzakta denge konumuna geldiği gözlenmektedir. Matlab koduna geri dönülerek x ve y ağırlık faktörlerini değiştirerek daha düzgün bir cevap elde edilmeye çalışılacaktır.

y faktörünün değeri artırılarak yerleşme zamanlarında bir iyileşme sağlanırken, x faktörünün artması ile sistemin keskin hareketleri azaltılmakta ve sarkaç açısının sapmaları yumuşatılmaktadır.

$x = 100$ ve $y = 5000$ değerleri kullanılarak aşağıdaki K matrisi ve sistem davranışı elde edilmiştir. $K = [-100.6782 \ -17.2955 \ -70.7107 \ -36.1445]$

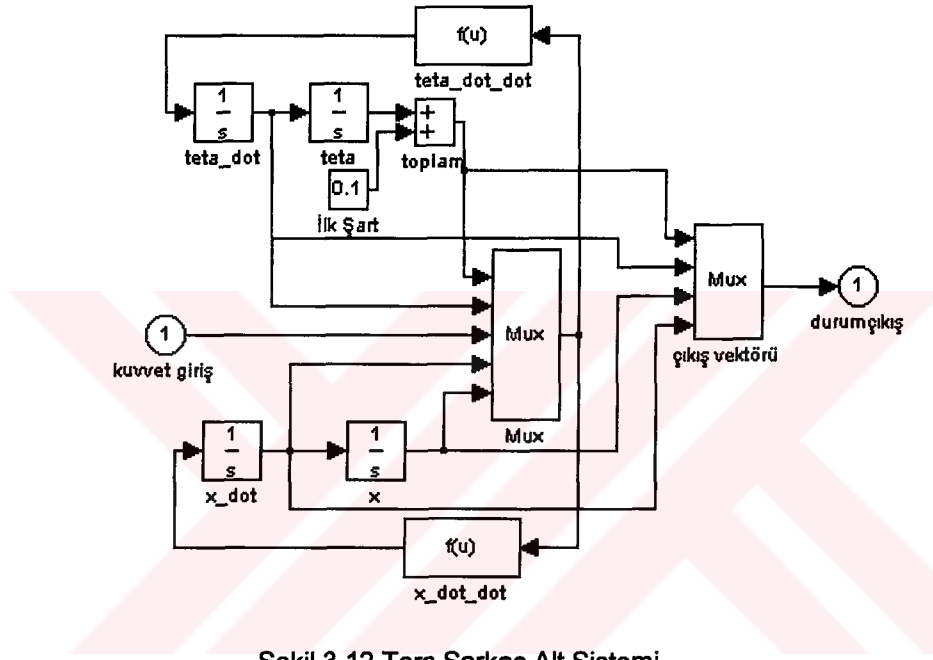


Şekil 3.11 Lineer Kuadratik Regülatör Sisteminin Basamak Girişe Cevabı ($x = 100$, $y = 5000$)

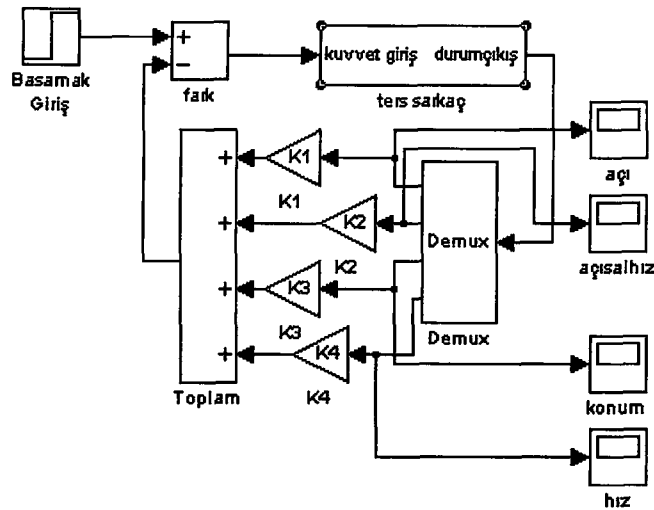
Sistem cevabını daha fazla iyileştirebilmek için x ve y değerleri biraz daha artırılabilir. Fakat sistem bu değerler altında istenen koşulları sağladığı için ağırlık faktörlerini olabildiğince minimum değerlerde tutmak, sistemin makul kontrol etkisi ile çalıştırılmasını sağlar. Şekil 3.11'de sistem, durum geri besleme kontrol sisteminde elde edilen 2 sn.lik yerleşme zamanından daha iyi bir performansla $t_s = 1.5$ sn.de denge konumuna ulaşmıştır. Fakat bu sonucu elde ederken sistemin sönümleme özelliği zayıflamıştır.

3.5 Ters Sarkaç Sisteminin Matlab/Simulink® İle Modellenmesi

Bu bölümde amaçlanan, MATLAB programının blok diyagram esaslı, grafik tabanlı Simulink® simülasyon ortamı yardımı ile sistemin, lineer ve lineer olmayan modellerinin, daha önce tasarlanan kontrolcünün denetimi altında olduğu bir simülasyon çalışmasını gerçekleştirmektir. Bu amaçla Dabney ve Harman'nın [15] oluşturduğu blok diyagram modeli örnek alınarak önce, kontrol edilecek sistemin Simulink® ortamında lineer ve lineer olmayan modelleri oluşturulacaktır. Simulink® ile elde edilen sistem ve kontrolcü blok diyagramları Şekil 3.12 ve 3.13 da verilmiştir.



Şekil 3.12 Ters Sarkaç Alt Sistemi

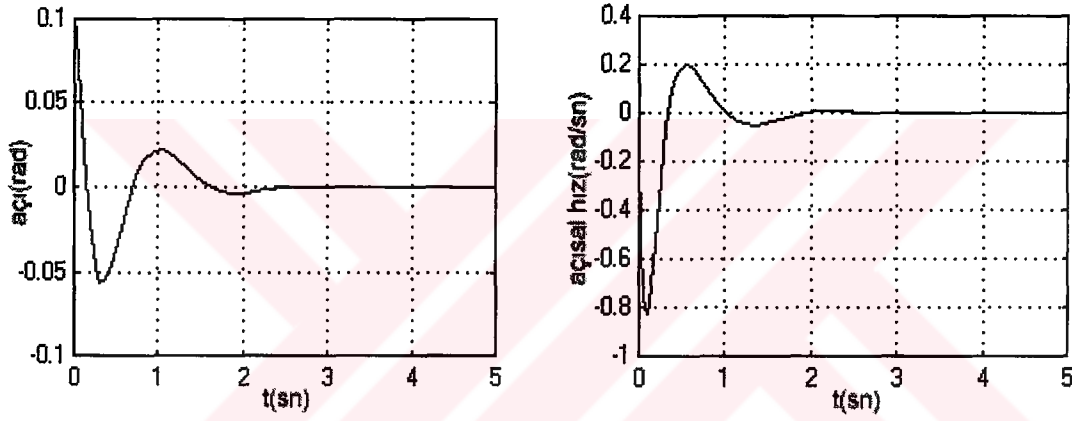


Şekil 3.13 Ters Sarkaç Kontrol Sisteminin Simulink Modeli

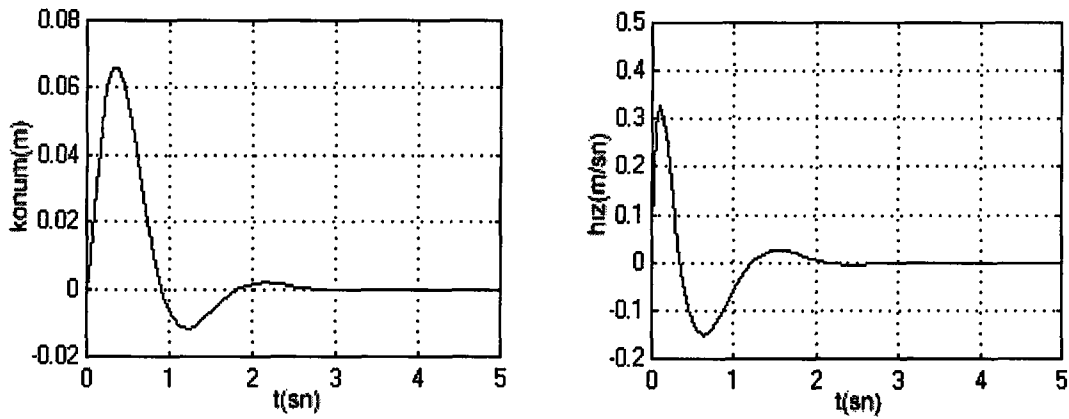
Kontrol uygulanmamış sisteme ait blok diyagramdaki teta_dot_dot ve x_dot_dot fonksiyon bloklarına önce lineerize edilmiş sisteme ait fonksiyonlar yerleştirilmiştir. Bu simülasyonlar sırasında bozucu etken kuvveti basamak girişi ile sağlandığından $F = 0$ olarak alınmıştır.

$$\ddot{\theta} = \frac{(M+m)g\theta + \left(\frac{2M}{m} + 1\right)F - u}{Ml} \quad (3.24)$$

$$\ddot{x} = \frac{u - F - mg\theta}{M} \quad (3.25)$$



Şekil 3.14 Sarkaç Kol Açısı ve Açısal Hızının Durum Geri Besleme Simülasyon Cevapları



Şekil 3.15 Araba Konumu ve Hızının Durum Geri Besleme Simülasyon Cevapları

Şekil 3.14 ve 3.15'deki simülasyon sonuçları, sabit bozucu etken karşısında çıkış değerlerinin $t_s = 3\text{sn}$ 'nin altına düşen yerleşme zamanı ile denge konumuna yerleştikleri görülmektedir. Simulink içinde fonksiyon bloklarından faydalanarak lineer olmayan sistemin modeli de oluşturulmuştur.

Bunun için ters sarkaç sistemine ait hareket denklemlerine geri dönecek olursa Newton'un ikinci yasası gereği tek bir serbest cisim olarak düşünülen araba ve sarkacın x yönündeki hareketi:

$$M \frac{d^2 x}{dt^2} + m \frac{d^2 x_G}{dt^2} = u \quad (3.26)$$

olarak yazılabilir. Buradan,

$$(M + m)\ddot{x} - ml(\sin \theta)\dot{\theta}^2 + ml(\cos \theta)\ddot{\theta} = u \quad (3.27)$$

Ayrıca sarkaç kolunun araba ile bağlantı noktası etrafındaki rotasyonel hareketi de:

$$mgl \sin \theta = m \frac{d^2 x_G}{dt^2} l \cos \theta - m \frac{d^2 y_G}{dt^2} l \sin \theta \quad (3.28)$$

şeklinde ifade edilebilir. Bu denklem de

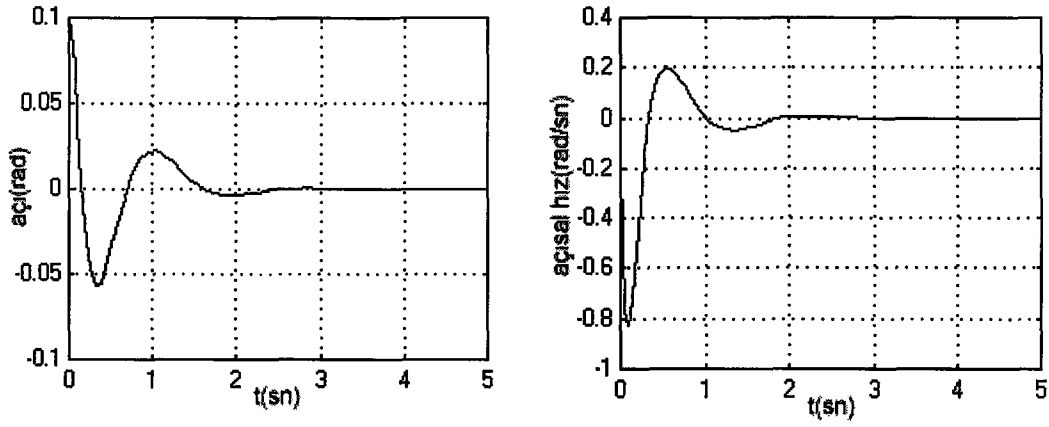
$$m\ddot{x} \cos \theta + ml\ddot{\theta} = mg \sin \theta \quad (3.29)$$

olarak sadeleştirilebilir. (3.27) ve (3.29) denklemleri kullanılarak Simulink fonksiyon bloklarına yerleştirilecek $\ddot{x}$ ve $\ddot{\theta}$ fonksiyonlarına ulaşılabilir:

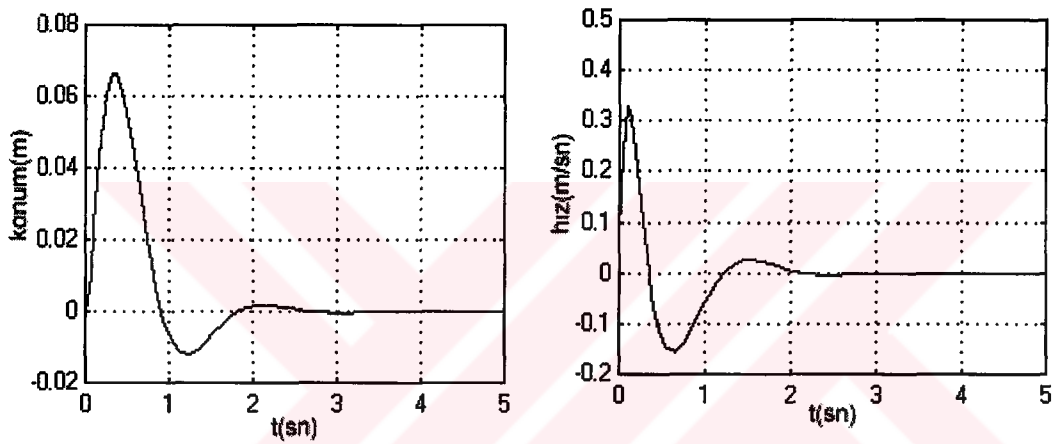
$$\ddot{\theta} = \frac{[(M + m)g \sin \theta - ml\dot{\theta}^2 \sin \theta \cos \theta - u \cos \theta]}{l(M + m \sin^2 \theta)} \quad (3.30)$$

$$\ddot{x} = \frac{(ml\dot{\theta}^2 \sin \theta) - (mg \sin \theta \cos \theta) + u}{(M + m \sin^2 \theta)} \quad (3.31)$$

lineerize edilmemiş denklemleri, simulink alt sistem blok diyagramındaki θ_{dot_dot} ve x_{dot_dot} fonksiyon bloklarına yerleştirildikten sonra; pol atamasıyla elde edilen kazanç katsayıları ile simülasyon gerçekleştirildiğinde elde edilen sonuçlar Şekil 3.16 ve 3.17 de gösterilmiştir.



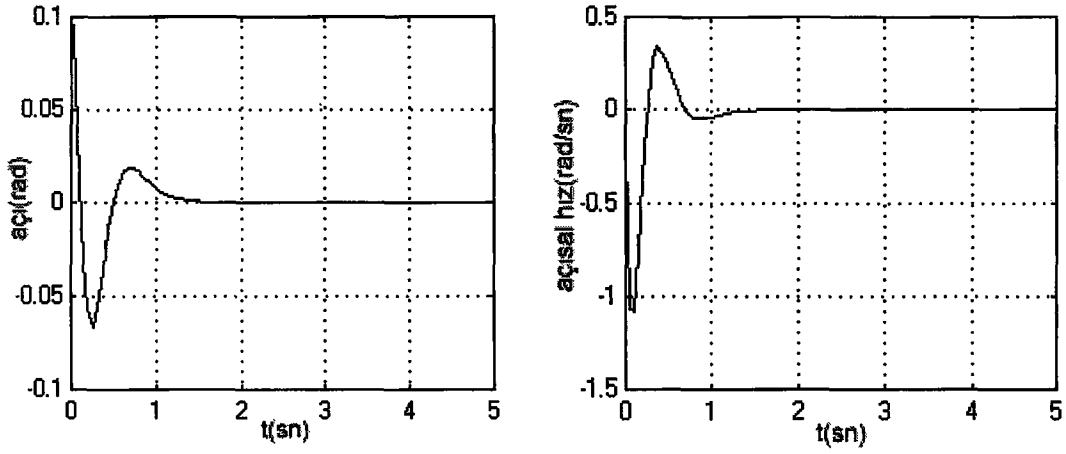
Şekil 3.16 Sarkaç Kol Açısı ve Açısal Hızının Lineer Olmayan Sistemde Bozucu Etken Cevabı



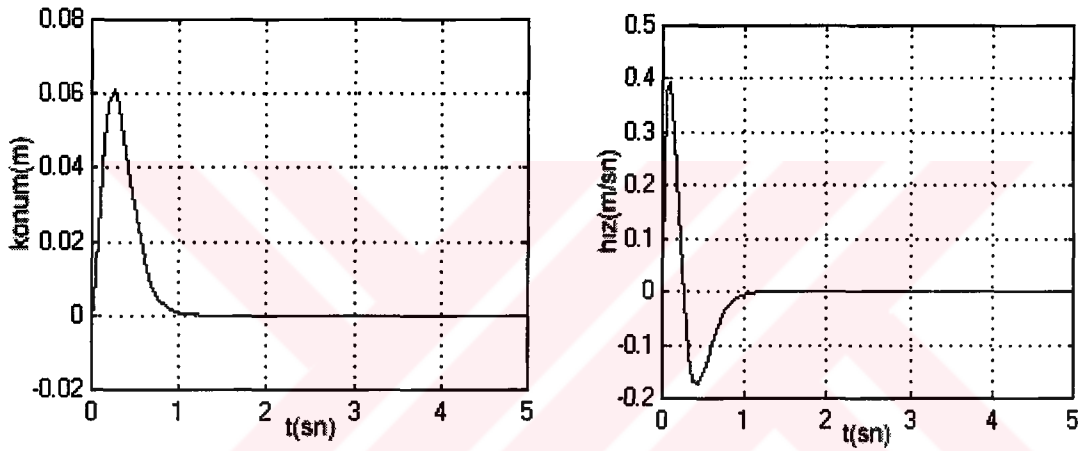
Şekil 3.17 Araba Konumu ve Hızının Lineer Olmayan Sistemde Bozucu Etken Cevabı

Şekil 3.16 ve 3.17'deki grafikler lineer sisteme ait Şekil 3.14 ve 3.15'deki grafiklerle karşılaştırıldığında, kullanılan basamak girişin değeri oldukça küçük olduğu için sistem performansında gözle görülür herhangi bir değişime rastlanmamıştır. Yerleşme zamanları ve aşma genlikleri hemen hemen aynı değerlerde kalmışlardır. Böylece lineer sistem için oluşturulan kontrolcü ile $\sin\theta$ ve $\cos\theta$ 'ü ifadelerle sahip lineerleştirilmemiş sistemin de küçük açı değişimleri için rahatlıkla kontrol edebildiği sonucuna ulaşılmıştır.

Ayrıca oluşturulan Simulink kontrolcü modeli ile Lineer Kuadratik Regülatöre ait kazanç katsayılarının da lineer olmayan sistemdeki performansları da incelenebilecektir. Bunun için Şekil 3.13 daki ters sarkaç kontrol sistemi blok diyagramındaki kazanç bloklarına, optimal kontrolcüye ait kazanç değerleri yerleştirilmiştir. Sisteme ait bozucu etken cevabı Şekil 3.18 ve 3.19'da gösterilmiştir.



Şekil 3.18 Sarkaç Kol Açısı ve Açısal Hızının LKR ile Lineer Olmayan Sistemde Bozucu Etken Cevabı



Şekil 3.19 Araba Konumu ve Hızının LKR ile Lineer Olmayan Sistemde Bozucu Etken Cevabı

Şekil 3.18 ve 3.19 incelendiklerinde, sarkaç kolu açısı ve araba konumu yerleşme zamanlarında $t_s \cong 1.5$ sn lik değerlerle denge konumuna ulaşırken arzu edilen %2lik t_s değerinin altına inmişlerdir. Aşma genliklerinde de iyileşmeler meydana geldiği gözlenmiştir, hatta araba konumu negatif yönde aşma yapmadan istenen denge konumuna ulaşmıştır. Sonuç olarak sistemin optimal kontrolcüye ait kazanç katsayıları ile daha hızlı ve kararlı bir performans sergilediği söylenebilir.

Bu bölümde MATLAB ve Simulink programları kullanılarak ters sarkaç sistemi için kontrolcü geliştirme çalışmaları yapılmıştır. Tasarlanan kontrolcünün küçük açı değişimleri altında lineer ve lineer olmayan sistem modelleri için yeterli performansı sergilediği gözlenmiştir. Bundan sonraki adımda, oluşturulan sistem modeli üzerinde farklı kontrol tekniklerinin uygulanması ve WinCon ters sarkaç deney düzeneğinde test edilmesi ile iyileştirme çalışmalarına devam edilecektir.

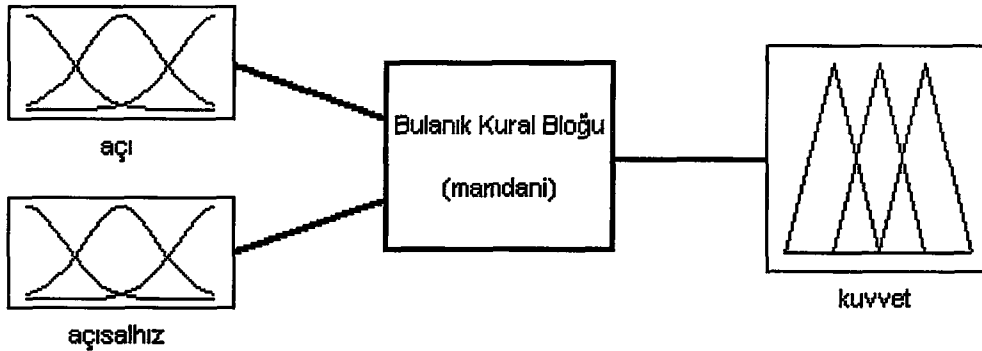
4. LİNEER OLMAYAN SİSTEMİN BULANIK MANTIK İLE KONTROLÜ

Son yıllarda bulanık sistemler ve kontrolcüler alanında hızlı gelişmeler gerçekleşmiştir. Bulanık kontrolcülerle tasarlanan pratiğe yönelik, tüketim ürünleri ve endüstriyel proses kontrolü gibi uygulamalarda elde edilen başarı bulanık kontrol sistemlerini geliştirmeyi amaçlayan araştırmaların sayısının artmasına neden olmuştur [16]. Bulanık kontrolcü tasarımı olağan uygulamalardan biraz daha fazla tasarım kararı alınmasını gerektirmektedir. Üyelik fonksiyonlarının seçimi kural tabanının oluşturulması, bulanıklaştırma ve saydamlaştırma konularında tasarımcının kararları büyük önem taşımaktadır.

Bu bölümde, kararsız denge dinamiğine sahip ters sarkaç sistemi için lineer olmayan bulanık bir kontrolcü tasarımı yapılmıştır. Bunun için MATLAB altında çalışan Fuzzy Logic Toolbox kontrolcü tasarım programından ve simülasyonlar için de blok diyagram temelli Simulink grafiksel simülasyon sisteminden yararlanılmıştır.

4.1 Bulanık Çıkarım Sistemi

Ters sarkaç için tasarlanan bulanık kontrol sistemi, öncelikle sarkacın kol açısı ve açısal hızının kontrolü amaçlanarak tasarlanmıştır. Bunun için araba konumu ve hızı dikkate alınmadan Mamdani yaklaşımı kullanılarak Şekil 4.1'deki kontrol bloğu oluşturulmuştur.



Şekil 4.1 Bulanık Çıkarım Sistemi Blok Diyagramı

Kontrol sisteminin giriş değişkenleri bulanıklaştırma prosesi yardımı ile bulanık değişkenler (açı ve açısal hız) haline getirilmiştir.

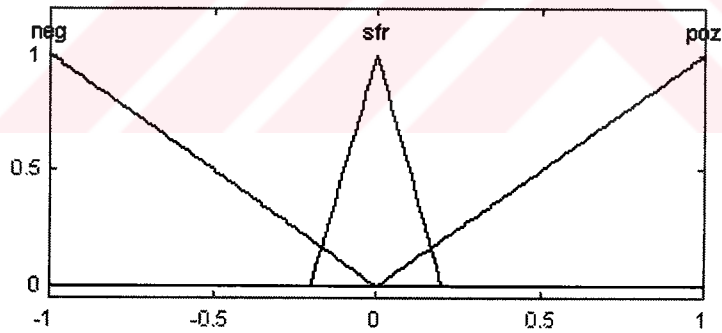
Şekil 4.1'de gösterilen blok diyagramda,

- aç: Sarkaç kolu aç değişimi
- açısal hız: Sarkaç kolu açısal hızı
- kuvvet: Ters sarkaç kontrol büyüklüğü

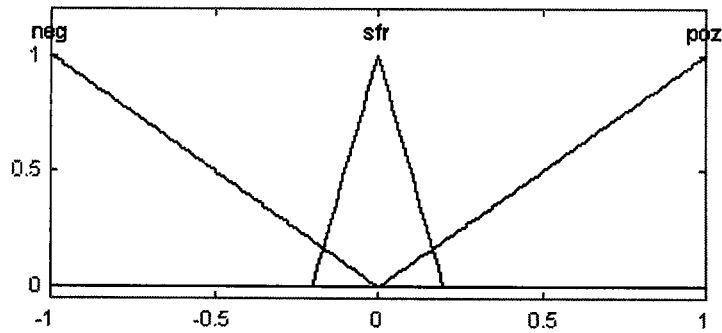
olarak tanımlanmaktadır.

Ayrıca her iki giriş değişkeni de, neg (negatif), sfr (sıfır) ve poz (pozitif) olarak üç linguistik terimle ifade edilmişlerdir. İki bulanık değişken ve her değişken için üç ayrı linguistik durum sonucunda $3^2 = 9$ olasılık söz konusu olacaktır. Bütün bu olasılıklar dokuz linguistik kural olarak kural bloğunda yer almıştır.

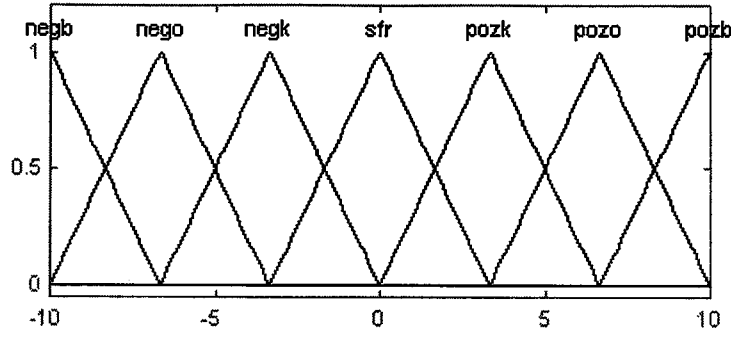
Optimal dengeleme kuvveti olan 'u' kontrol büyüklüğü, bulanık kontrol sisteminin kuvvet, çıkış değişkeni olarak düzenlenmiştir. Bu değişkene ait yedi linguistik terim ise: negb (negatif büyük), nego (negatif orta), negk (negatif küçük), sfr (sıfır) pozk (pozitif küçük), pozo (pozitif orta), pozb (pozitif büyük) olarak tanımlanmışlardır. Bulanık sistemin tasarımı ve simülasyonlar, MATLAB® v5.1, Fuzzy Logic Toolbox® v1.03 ve Simulink® v2.1 programları kullanılarak gerçekleştirilmiştir.



Şekil 4.2 Sarkaç Kolu Açısının Üyelik Fonksiyonları



Şekil 4.3 Sarkaç Kolu Açısal Hızının Üyelik Fonksiyonları



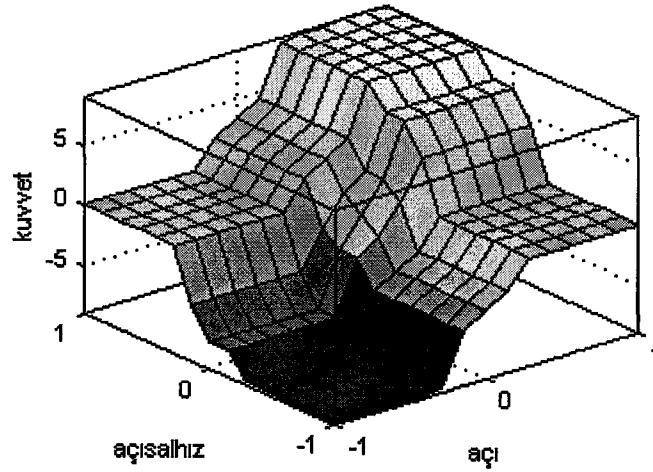
Şekil 4.4 Kuvvet Kontrol Büyüklüğünün Üyelik Fonksiyonları

Şekil 4.2, 4.3 ve 4.4'de gösterilen üyelik fonksiyonu grafiklerinde de görüldüğü gibi değişkenlerin tüm fonksiyonları, üçgen fonksiyonlar olarak seçilmişlerdir. Giriş ve çıkış değişkenleri arasındaki bulanık mantık ilişkisinin kurulmasında Mamdani yaklaşımına dayanan, 've' operatörü kullanılarak oluşturulan kural tabanı kullanılmıştır. Sistem giriş ve çıkış değişkenleri arasındaki bulanık mantıksal ilişkiyi ifade etmek için oluşturulan kural tablosu Tablo 4.1'de verilmiştir.

Tablo 4.1. Bulanık Sistem Kural Tablosu

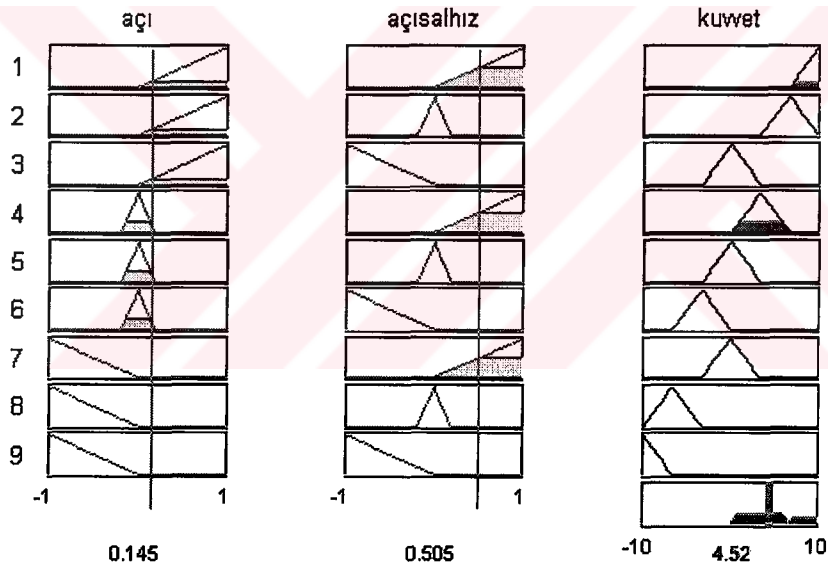
1. Eğer (açı = poz) ve (açısalhız = poz) ise (kuvvet = pozb)
2. Eğer (açı = poz) ve (açısalhız = sfr) ise (kuvvet = pozo)
3. Eğer (açı = poz) ve (açısalhız = neg) ise (kuvvet = sfr)
4. Eğer (açı = sfr) ve (açısalhız = poz) ise (kuvvet = pozk)
5. Eğer (açı = sfr) ve (açısalhız = sfr) ise (kuvvet = sfr)
6. Eğer (açı = sfr) ve (açısalhız = neg) ise (kuvvet = negk)
7. Eğer (açı = neg) ve (açısalhız = poz) ise (kuvvet = sfr)
8. Eğer (açı = neg) ve (açısalhız = sfr) ise (kuvvet = nego)
9. Eğer (açı = neg) ve (açısalhız = neg) ise (kuvvet = negb)

Kullanılan MATLAB/ Fuzzy Logic Toolbox programı yardımı ile sistemin girişleri ve çıkışı arasında, bu kurallarla oluşturulan, ilişkiyi üç boyutlu yüzeyler halinde görüntülemek mümkün olmaktadır. Üç boyutlu yüzey gösterimi, sistemin davranışı hakkında daha tasarım aşamasında iken bulanık değişkenler ve kontrol büyüklüğü arasındaki ilişkiyi görüntüleyebildiği için sistem davranışının öngörülmesini sağlamaktadır. Şekil 4.5'te gösterilen üç boyutlu grafik incelendiğinde kontrol büyüklüğü olan kuvvetin, sarkaç açısı ve açısal hızının değişimi ile sıfır denge konumuna göre kısmen simetrik bir davranış gösterdiği söylenebilir. Bu da kontrolcünün sarkaç kolunu istenilen konuma, yani $\theta_d = 0(\text{rad})$ değerine getirebileceğinin bir belirtisi olarak görülebilir.



Şekil 4.5 Kuvvet, Sarkaç Kolu Açısı ve Açısal Hızının 3-B Yüzey Diyagramı

Fuzzy Logic Toolbox'un kullanıcıya sunduğu bir diğer kolaylık oluşturulan kuralların kontrol büyüklüğüne ne şekilde etkiğini gösteren Şekil 4.6'daki kural görüntüleyici penceresidir.



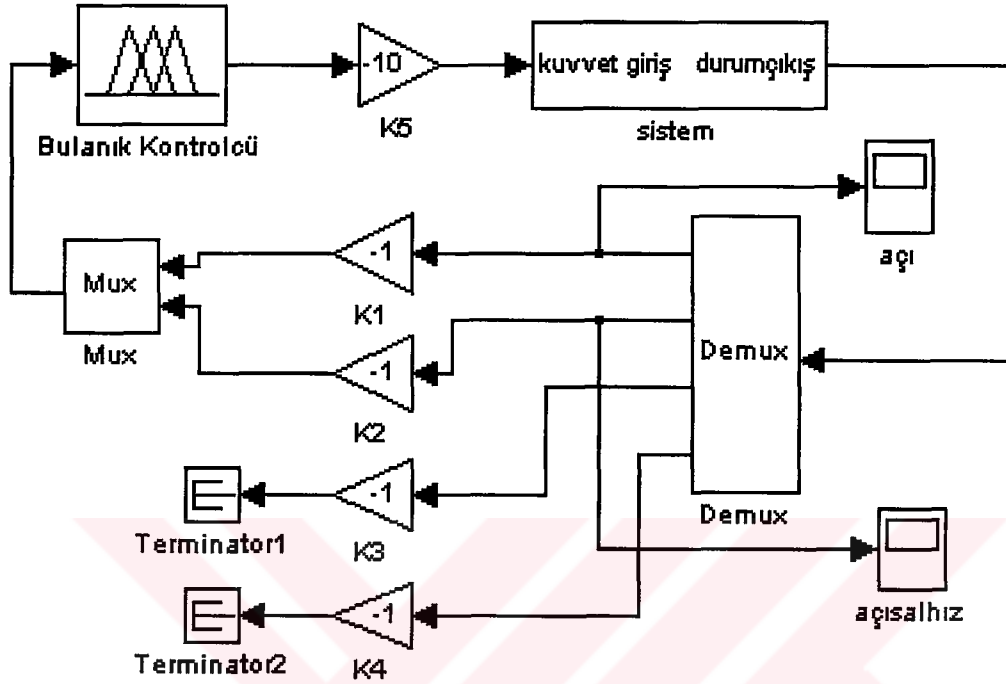
Şekil 4.6 Bulanık Kuralların Görüntülenmesi

4.2 Bulanık Sistem Simülasyonları

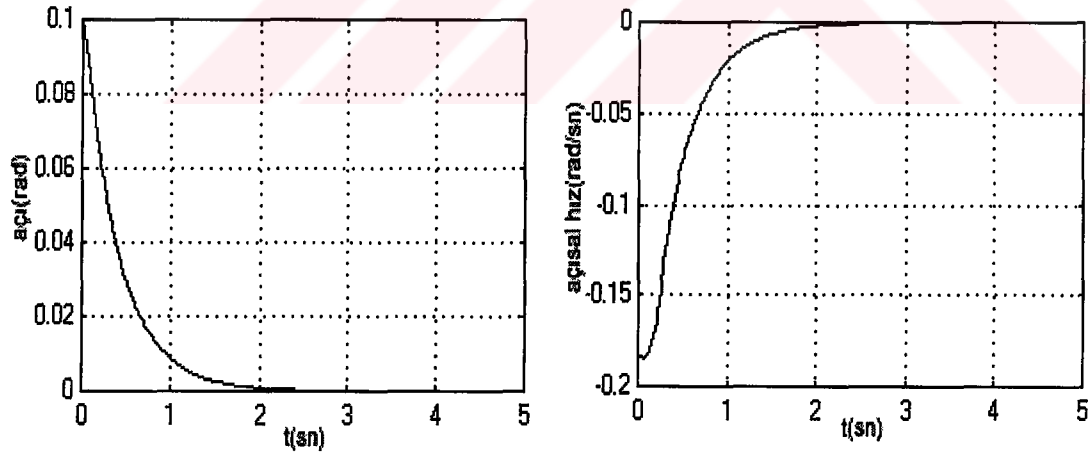
Tasarlanan bulanık mantık kontrol sistemini test etmek için sistem öncelikle bu sistemi Simulink blok diyagramı olarak tanıtılmıştır Bunun için kısa bir MATLAB kodundan yararlanılmıştır.

```
[xx, yy] = meshgrid(0:1);
input = [xx(:) yy(:)];
fismat1 = readfis('Bulanık Kural Bloğu');
out = evalfis(input, fismat1);
```

Bu işlemle elde edilen Bulanık Kontrolcü Bloğu önceden oluşturulan ters sarkaç simülasyon modeline Şekil 4.7'de gösterildiği gibi durum geri besleyerek yerleştirilmiştir.



Şekil 4.7 Sarkaç Kolu İçin Bulanık Kontrolcü Simülasyon Modeli



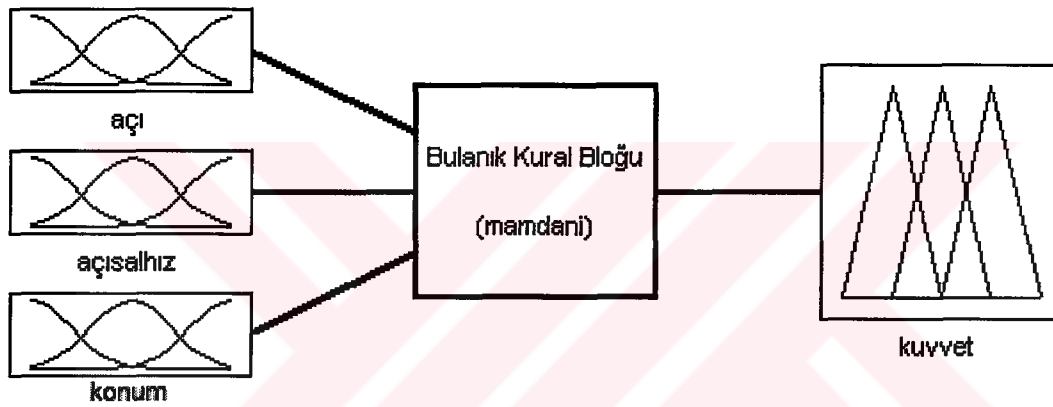
Şekil 4.8 Bulanık Kontrolcü ile Sarkaç Kolu Açısının ve Açısal Hızının Cevapları

Şekil 4.8'deki grafikler incelendiğinde bulanık mantıkla oluşturulan kontrolcünün, başarılı bir performansla kontrol edilen lineer olmayan sistemi, arzu edilen %2'lik yerleşme zamanı olan $t_s = 2$ sn.ye yaklaştırabildiği gözlenmiştir. Fakat sarkaç kolu $\theta_0 = 0^\circ$ de denge konumuna getirilirken araba konumu için herhangi bir kontrol uygulanmamıştır. Araba konumunu ve hızını da dikkate alarak daha etkili bir kontrol sistemi tasarlamak için sarkaç kolu açısı ve açısal hızını istenen değerlere getirirken

aynı zamanda arabanın konumunu dengeleyecek bir kontrol büyüklüğü oluşturulmalıdır. Burada karşılaşılan en büyük sorun sadece araba üzerinden uygulanan kontrol kuvveti, sarkaç açısı ve araba konumunun aynı anda negatif veya pozitif değerler aldığı uç noktalarda her iki değişkeni de aynı anda dengeye getirecek etkiyi oluşturmamasıdır. Bu handicap üyelik fonksiyonları ve bulanık kural tabanının değiştirilmesi, ayrıca araba konumunun da bulanık giriş değişkeni olarak sisteme eklenmesi ile aşılmaya çalışılmıştır.

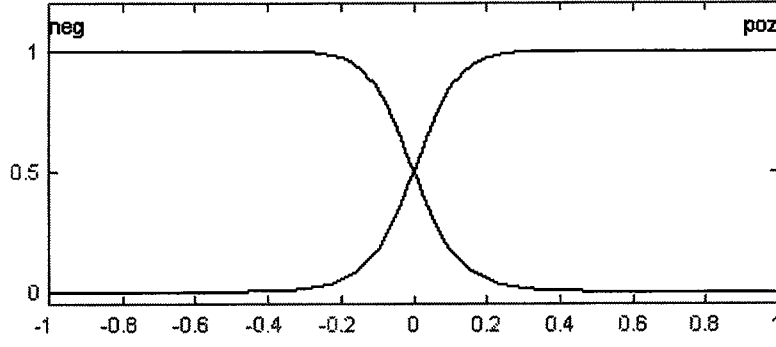
4.3 Üç Giriş Değişkenli Bulanık Çıkarım Sistemi

Bu aşamada, Şekil 4.9'da görülen araba konumunun sisteme giriş değişkeni olarak eklendiği yine Mamdani yaklaşımı ile oluşturulan bir bulanık kontrolcü tasarlanmıştır.

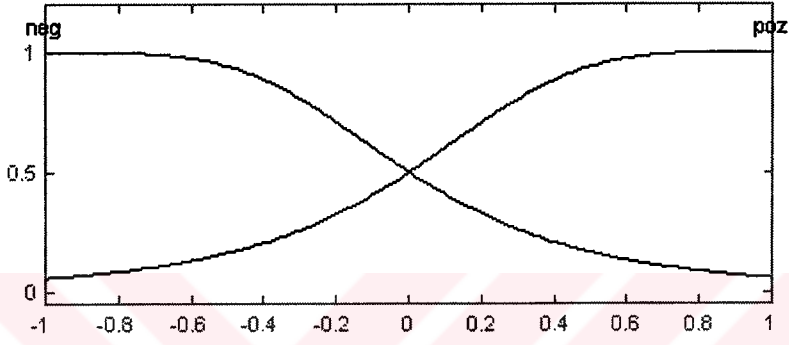


Şekil 4.9 Üç Girişli Bulanık Çıkarım Sistemi Blok Diyagramı

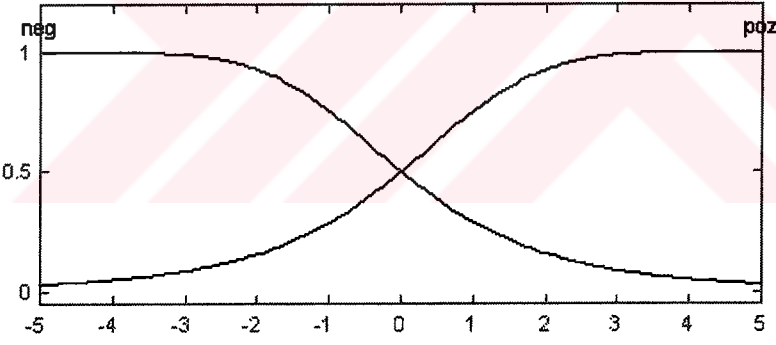
Şekil 4.9'da gösterilen blok diyagramda Şekil 4.1'dekinden farklı olarak eklenen konum giriş değişkeni arabanın konumunu ifade eden bulanık değişkendir. Kural sayısının makul bir değerde tutulması amacı ile her üç giriş değişkeni, Şekil 4.10, 4.11 ve 4.12'de görüldüğü gibi neg (negatif) ve poz (pozitif) olarak iki linguistik terimle ifade edilmişlerdir. Kontrol değişkeni kuvvet için Şekil 4.13'de gösterilen, negB (negatif büyük), negK (negatif küçük), pozK (pozitif küçük) ve pozB (pozitif büyük) şeklinde sadece dört linguistik terime yer verilmiştir.



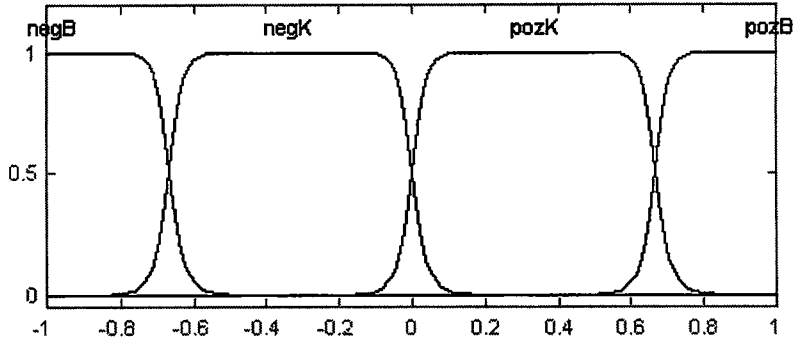
Şekil 4.10 Sarkaç Kolu Açısının Üç Girişli Sistem İçin Üyelik Fonksiyonları



Şekil 4.11 Sarkaç Kolu Açılmal Hızının Üç Girişli sistem İçin Üyelik Fonksiyonları



Şekil 4.12 Araba Konumunun Üç Girişli sistem İçin Üyelik Fonksiyonları



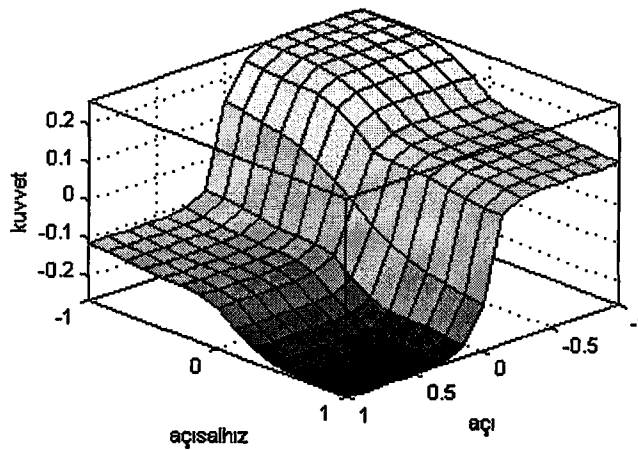
Şekil 4.13 Kuvvet Kontrol Büyüklüğünün Üç Girişli sistem İçin Üyelik Fonksiyonları

Bu tasarımda üyelik fonksiyonları bir öncekinden farklı olarak gauss çan eğrisi 'gbell' fonksiyonu seçilmiştir. Bunun nedeni bulanık kurallar arasında daha yumuşak bir geçişin sağlanmasıdır. Burada yine bir önceki tasarımdan farklı olarak her olasılık için ayrı bir kural yazmak yerine sarkaç kolu için $2^2 = 4$ ve araba için $2^1 = 2$ olmak üzere toplam altı kural oluşturulmuştur. Sistem giriş ve çıkış değişkenleri arasındaki bulanık mantıksal ilişkiyi ifade etmek için oluşturulan kural tablosu Tablo 4.2'de verilmiştir.

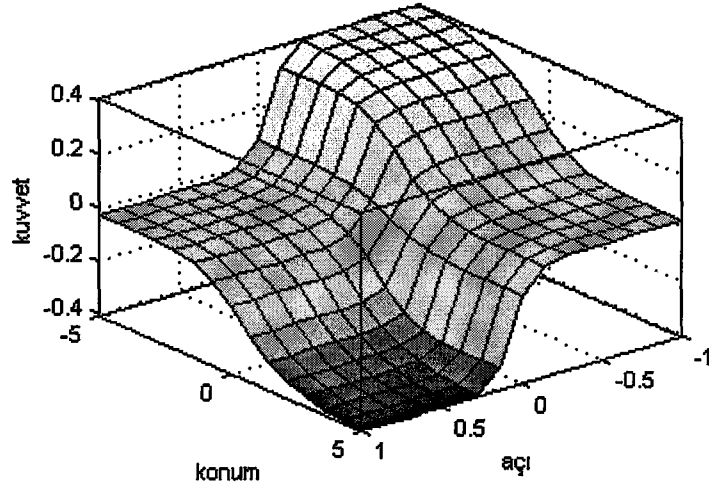
Tablo 4.2 Üç Girişli Sistem İçin Bulanık Kural Tablosu

1. Eğer (açı = neg) ve (açısalhız = neg) ise (kuvvet = pozB)
2. Eğer (açı = neg) ve (açısalhız = poz) ise (kuvvet = pozK)
3. Eğer (açı = poz) ve (açısalhız = neg) ise (kuvvet = negK)
4. Eğer (açı = poz) ve (açısalhız = poz) ise (kuvvet = negB)
5. Eğer (konum = neg) ise (kuvvet = pozK)
6. Eğer (konum = poz) ise (kuvvet = negK)

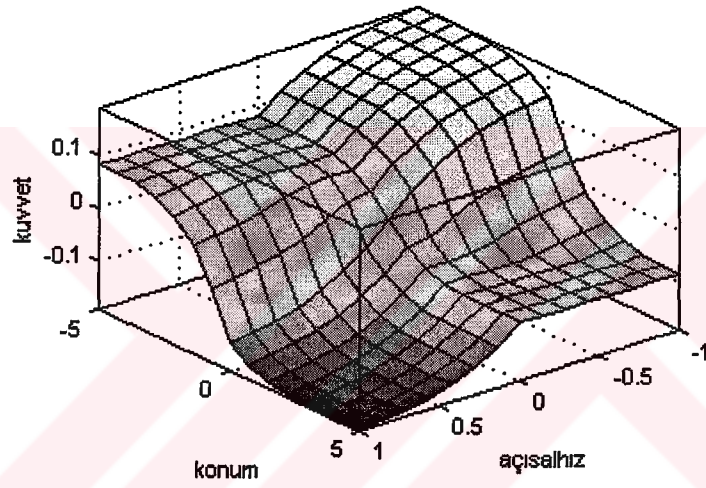
Fuzzy Logic Toolbox yardımı ile oluşturulan sistemin üç boyutlu yüzey grafikleri incelenirse Şekil 4.14'de kontrol büyüklüğü olan kuvvetin sarkaç kolu açı ve açısal hızına göre değişimi Şekil 4.5'de görülen ilk tasarıma ait grafiğe göre daha yumuşak bir geçişle oluşmaktadır. Bu yumuşak geçiş kullanılan 'gbell' fonksiyonu sayesinde meydana gelmiştir. Şekil 4.15 ve 4.16'daki grafiklerde de yine kullanılan üyelik fonksiyonlarının kurallar arası geçişi yumuşatan etkisi gözlenmektedir. Ayrıca araba konum ve açı arasındaki ilişkinin, açısal hız ile açı değişimi arasındaki ilişki ile benzer bir yapı göstermesi sistemin kararlılığı açısından önemli olabilir.



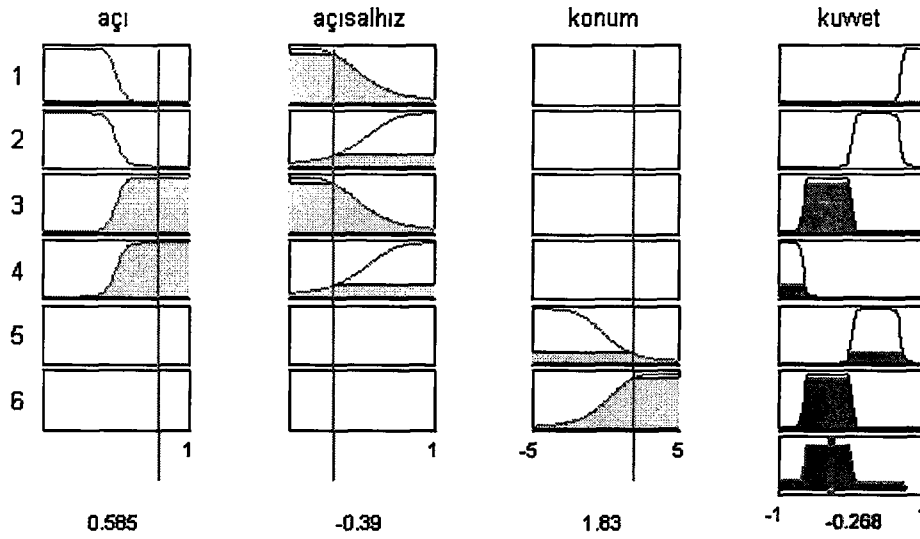
Şekil 4.14 Üç Girişli Sistemin Sarkaç Kolu Açısı, Açısal Hızı ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği



Şekil 4.15 Üç Girişli Sistemin Sarkaç Kolu Açısı, Araba Konumu ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği



Şekil 4.16 Üç Girişli Sistemin Sarkaç Kolu Açısal Hızı, Araba Konumu ve Kontrol Büyüklüğünün 3-B Yüzey Grafiği



Şekil 4.17 Üç Girişli Sistem için Bulanık Kuralların Görüntülenmesi

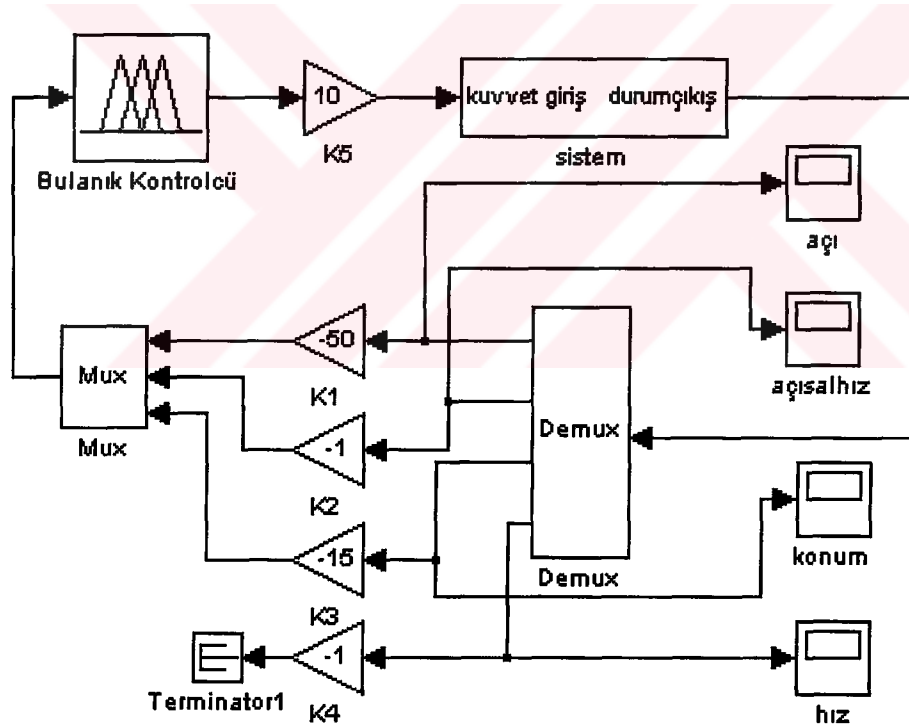
Kuralların kontrol büyüklüğünü nasıl etkilediğini inceleyebilmek için Şekil 4.17 'deki kural görüntüleme penceresi kontrol edildiğinde bulanık sistemin oluşan durum için uygun büyüklükte bir kontrol değeri oluşturduğu gözlenmektedir.

4.4 Üç Girişli Sistem İçin Bulanık Kontrolcü Simülasyonları

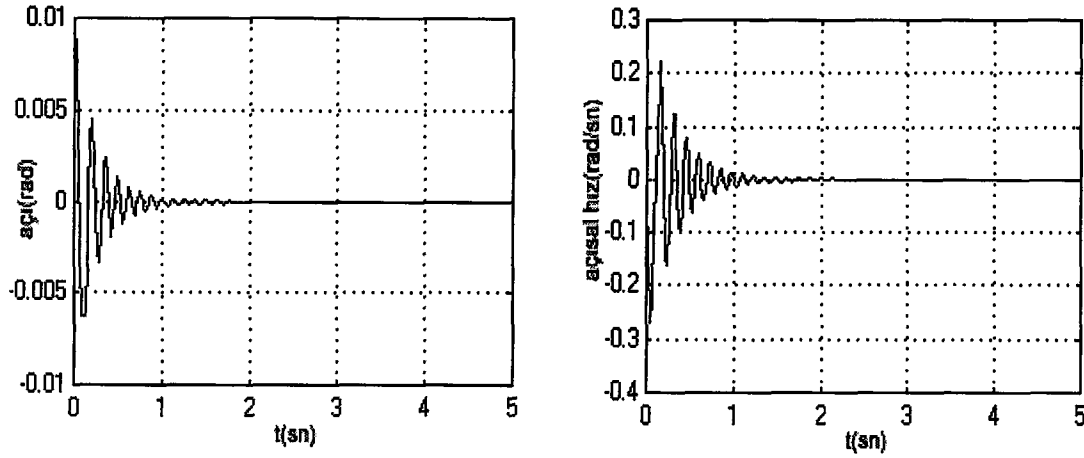
Tasarlanan bulanık kontrolcü sistemin Simulink'te bir blok diyagrama dönüştürülmesi için yazılan kodda üçüncü giriş olan araba konumu için aşağıdaki değişiklik yapılmıştır.

```
[xx, yy, zz] = meshgrid(0:1);
input = [xx(:) yy(:) zz(:)];
fismat1 = readfis('Bulanık Kural Bloğu');
out = evalfis(input, fismat1);
```

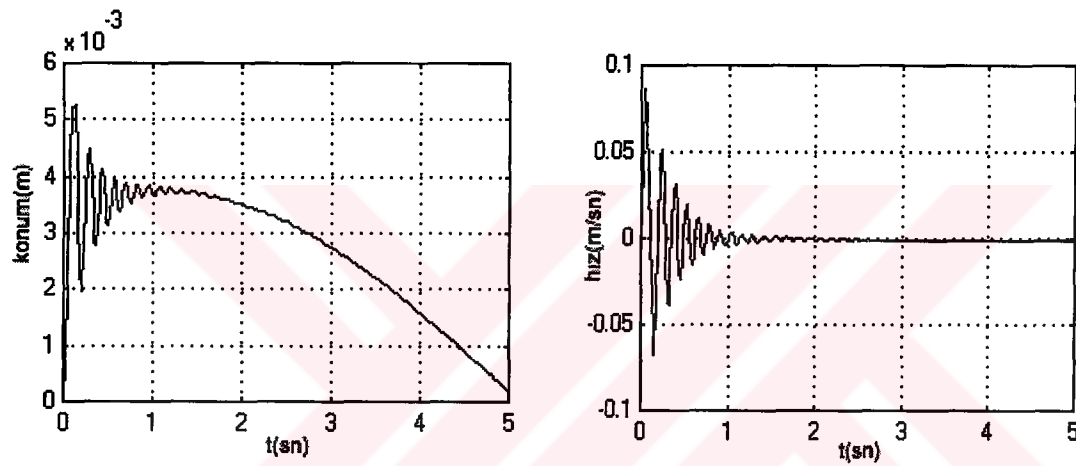
Daha önceki tasarımda oluşturulan Şekil 4.7'deki simülasyon modeli üzerinde üçüncü giriş olan araba konumu için gerekli değişikli yapıldığında Şekil 4.18'deki Simulink modeli elde edilmiştir.



Şekil 4.18 Ters Sarkaç Sistemi İçin Üç Boyutlu Simülasyon Modeli

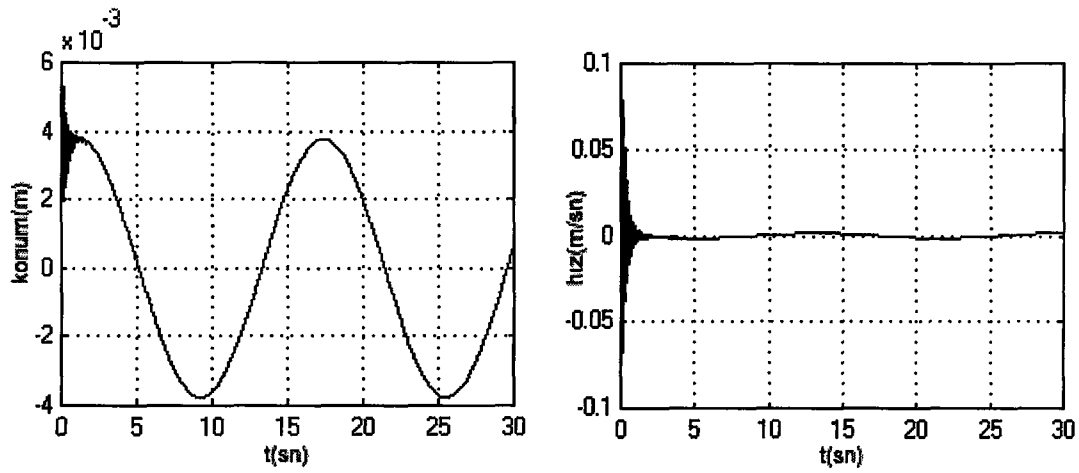


Şekil 4.19 Üç Girişli Bulanık Kontrolcü ile Sarkaç Kolu Açısının ve Açısal Hızının Simülasyon Cevapları



Şekil 4.20 Üç Girişli Bulanık Kontrolcü ile Araba Konumu ve Hızının Simülasyon Cevapları

Şekil 4.19 ve 4.20'deki grafikler incelendiğinde oluşturulan bulanık sistem, sarkaç kolu açısı ve açısal hızı için kabul edilebilir bir performans sağladığı görülmektedir. Sistem, sarkaç kolu için uyguladığı kontrolü araba konumu ve hızı için tam olarak gösteremese bile bu değişkenler, istenen referans değerleri ($x_d = 0$ ve $V_{xd} = 0$) etrafında, tolerans sınırı içerisinde sayılabilecek bir genlikle (araba konumu için bu değer $\approx 4 \times 10^{-3}$ 'dür) sinüzoidal bir davranış sergilemişlerdir. Araba konumunun ve hızının bu davranışlarının izlenebilmesi için simülasyonların süresi Şekil 4.21'de görüldüğü gibi $t = 30$ (sn) olarak tutulmuştur.



Şekil 4.21 Araba Konumu ve Hızının $t = 30$ (sn) için Bulanık Kontrolcü Sistem Cevabı

Tasarlanan bulanık kontrolcü sistemi linguistik kurallara dayalı lineer olmayan bir kontrolcüdür. Bu sistemin geliştirilebilmesi için bulanık PD, bulanık PI gibi farklı bileşik yapılarda sistemler oluşturulabilir. Aynı zamanda kullanılan kazançları ve parametreleri kendisinin ayarlayabildiği (self tuning) bulanık kontrol sistemleri oluşturulabilir.

5. TERS SARKAÇ DENEY UYGULAMASI

Quanser Consulting firmasının ürünü olan ters sarkaç deney düzeneği,

- dişli bir ray,
- bu ray üzerinde hareket edebilen ve bir doğru akım motoru ile tahrik edilen araba,
- arabaya monte edilebilen bir sarkaç
- motora akım ileten UPM-1503 üniversal güç modülünden oluşmaktadır.

IP-02 olarak tanımlanan sistem, bir doğru akım motoru tarafından tahrik edilen ve üzerinde iki adet kodlayıcı bulunan bir arabaya sahiptir. Kodlayıcılardan biri rayla temas halinde bulunan dişli bir pinyon yardımı ile arabanın pozisyonunu ölçerken diğer kodlayıcı da arabanın ön tarafında bulunan ve serbest olarak salınabilecek şekilde tespit edilmiş olan sarkaç kolu açısını ölçmektedir.

Sistemin denge konumuna gelebilmesi için uygulanacak kontrol büyüklüğü tasarlanan lineer modelde arabaya etkiyen bir u kuvvetine dayanmaktadır. Gerçek sistemde bu kontrol büyüklüğü elektriksel gerilim ifadesi olan voltajdır. Bu ifadeyi elde edebilmek için kullanılan tahrik motorunun bağıntılarından yararlanılacaktır.

$$V = I_m R_m + K_m K_g \omega_g = I_m R_m + K_m K_g \frac{\dot{x}}{r} \quad (5.1)$$

Burada;

V (volt) :Motora uygulanan gerilim

I_m (A) :Motorun çektiği akım

$K_m \left(\frac{V}{\text{rad/s}} \right)$:Elektromotor kuvveti sabiti (Back EMF Consant)

K_g :Motor dişli kutusundaki çevirme oranı

ω_g (rad/sn) :Motor çıkış açısal hızı

x (m/sn) :Araba hızı

r (m) :Rayla temas halindeki pinyonun yarıçapı olarak verilmiştir.

Motorun ürettiği moment (5.2) denklemi ile hesaplanmıştır.

$$T_m = K_m K_g I_m \quad (5.2)$$

Hesaplanan bu momentin arabaya ilettiği kuvvet ise (5.3)'de gösterilmiştir.

$$u = \frac{T_m}{r} \quad (5.3)$$

(5.1), (5.2) ve (5.3) kullanılarak kontrol kuvvetinin motora uygulanan gerilim cinsinden ifadesi (5.4) denklemindeki gibi çıkarılmıştır.

$$u = \frac{K_m K_g}{R_m r} V - \frac{K_m^2 K_g^2}{R_m r^2} \dot{x} \quad (5.4)$$

Üçüncü bölümde oluşturulan MATLAB kodu, elde edilen bu kuvvet denklemine göre düzenlenmiştir.

```
M = 0.522;
m = 0.231;
g = 9.81;
l = 0.305;
R = 2.6;
Kg = 3.7;
Km = 0.00767;
r = 0.00635;
kV = Km*Kg/R/r;
kxd = Km^2*Kg^2/R/r/r;

A = [0 1 0 0;
      ((M+m)/(M*l))*g 0 0 kxd/M/l;
      0 0 0 1;
      -(m/M)*g 0 0 -kxd/M];
B = [0; -kV/(M*l); 0; kV/M];
C = [1 0 0 0; 0 0 1 0];
D = [0; 0];

A(:,5)=0*ones(size(5,1));
A(5,:)=0 0 1 0 0;
B=[B;0];
eig(A)
P=[-2+2*j -2-2*j -30 -30.01 -0.1] '
[K,PREC,MESSAGE] = place(A,B,P)
```

```

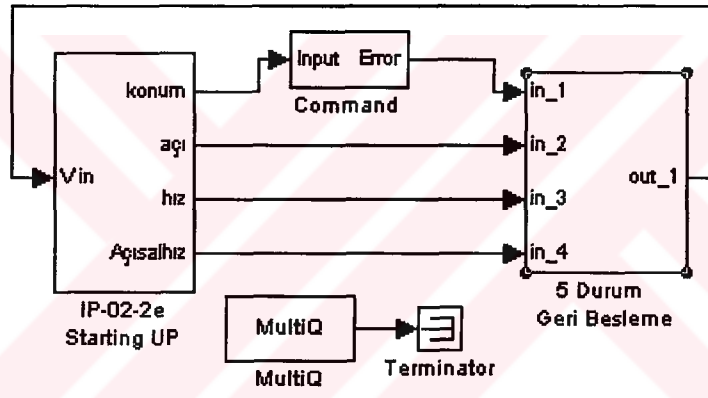
k(1)=K(3)/100;
k(2)=K(1)*pi/180;
k(3)=K(4)/100;
k(4)=K(2)*pi/180;
k(5)=K(5)/100;
k

```

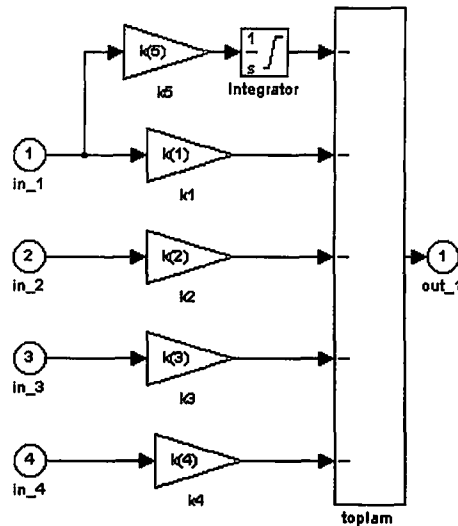
$K = [-133.1697 \ -18.0862 \ -71.8565 \ -44.2991 \ -6.8003]$ ve

$k = [-0.7186 \ -2.3242 \ -0.4430 \ -0.3157 \ -0.0680]$ olarak hesaplanmıştır.

Bu kodda, üçüncü bölümde oluşturulandan farklı olarak durum geri besleme kontrolcüsünü oluşturmak için MATLAB Control Toolbox'taki 'place' komutundan yararlanılmıştır. Ayrıca dominant olmayan kapalı çevrim kökleri de imajiner ekseninin biraz daha uzağına yerleştirilirken, elde edilen durum geri besleme kazanç parametreleri gerçek zamanda test için Wincon tarafından oluşturulan Şekil 5.1' de gösterilen Simulink modeline göre düzenlenmiştir.



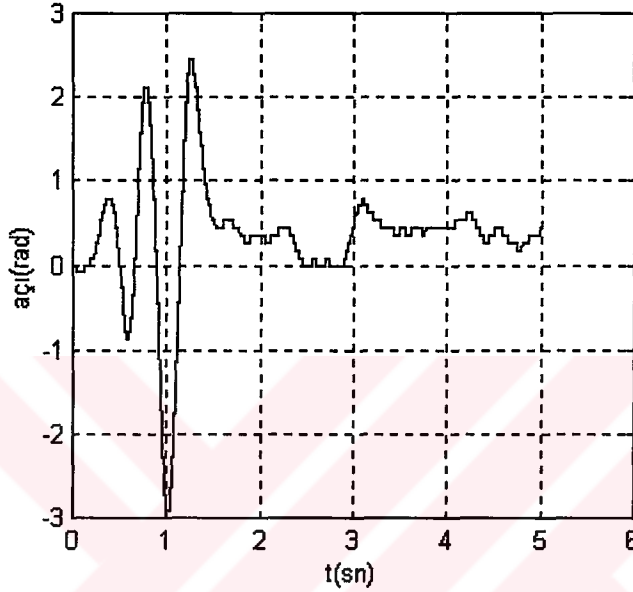
Şekil 5.1 IP-02 Ters Sarkaç Simulink Modeli



Şekil 5.2 Durum Geri Besleme Kazanç Çarpanları Blok Diyagramı

Şekil 5.2'de gösterilen durum geri besleme blok diyagramında görüldüğü gibi araba konumu giriş değeri, üçüncü bölümdeki tasarımdan farklı olarak, sistem sürekli rejim hatalarının azaltılması amacı ile $k(5)$ integratör kazancı ile çarpılmıştır.

Tasarlanan model, Simulink içindeki Wincon menüsünden 'Build' komutu ile ters sarkaç sistemi için gerçek zaman kontrolcüsüne dönüştürülmüştür. Daha sonra Wincon sunucu penceresinden kontrolcü sistem çalıştırılarak sarkaç kolu açısının davranışının grafiği Şekil 5.3'deki gibi elde edilmiştir.



Şekil 5.3 Sarkaç Kolu Açısının Gerçek Zamanda Deney Cevabı

Kontrol sistemi çalıştırıldıktan sonra birinci saniye içerisinde sarkaç koluna dışarıdan küçük bir açısal artım verilmiştir. Sistem kendisini 3 saniye içerisinde toparlamış, sarkaç kolu açısı ve araba konumu istenen denge konumlarına gelmişlerdir. Fakat sistemin uyguladığı kontrol kuvveti ani ve sert bir davranış göstermiştir. Bu handikap kontrolcü kazançlarında yapılacak ayarlamalarla düzeltilebilir.

Sonuç olarak ters sarkaç sistemi için tasarlanan durum geri besleme kontrolcüsü yapılan birkaç düzenleme ile gerçek sistem için yapılandırılarak başarılı sayılabilecek bir performans sergilemiştir.

6. SONUÇLAR VE ÖNERİLER

Bu yüksek lisans çalışmasında, mekatronik teknolojisine yönelik uygulamalarda kontrolcü geliştirme sistemleri incelenmiştir. Çalışma kapsamında hızlı prototiplendirme sistemleri ve çevrim içi donanım içeren simülasyonlar tanıtılmıştır. Bu sistemlere ait yazılımlar ve uygulama alanları için örnekler verilmiştir. Aslında geliştirme ve üretim prosesinin birer parçası olan bu iki bilgisayar destekli mühendislik aracı, tasarımcının amacına göre yapacağı, kontrol yazılımı geliştirilmesi yada kontrol edilecek mekanik birimin tasarlanan yazılımla test edilmesi tercihi ile birbirinden ayrılmaktadır.

Kararsız bir denge dinamiğine sahip ters sarkaç sisteminin matematik modelinin çıkarılması ve bu modele ait kontrolcü geliştirilmesi, hızlı kontrolcü prototiplendirme uygulaması olarak çalışmada yer almıştır. Bu bölümde sistemin matematik modeli belirli varsayımlar ve kabuller doğrultusunda WinCon ters sarkaç deney düzeneğinin parametreleri kullanılarak lineer ve non-lineer olarak çıkarılmıştır. Bu modele ait durum geri besleme ve lineer kuadratik optimal kontrolcü tasarımları gerçekleştirilmiş, MATLAB/Simulink programları yardımı ile çevrim dışı simülasyonlar gerçekleştirilmiş ve sistem cevabı grafiksel olarak çıkarılmıştır.

Ayrıca geliştirilen kontrolcünün kararlılık sınırlarının belirlenmesi için sistemin gürbüzlüğü incelenmiştir. Bu analizde temel olarak sarkaç kol boyu (L) ve buna bağlı ağırlığı (m) değişken parametreler olarak alınmıştır. Böylece tasarlanan kontrolcünün m ve L 'nin hangi değerleri için kararlı olduğu ve istenen performansı (seçilen sönümlleme değeri ve yerleşme zamanı) sağladığı çıkartılmıştır. Bunun yanında durum geri besleme kontrolcüsünün sarkaç kol açısı ve araba konumunun kazanç değerleri olan $K(1)$ ve $K(2)$ parametreleri için de parametre uzayı oluşturulmuş ve ortaya çıkan bölgeler için Γ -kararlılığı incelenmiştir.

Çalışmada bir sonraki adım lineer olmayan ters sarkaç modeli için yine lineer olmayan bir kontrol sistemi olan bulanık mantık kontrolcüsü tasarlanmıştır. Sadece sarkaç kolunun açı ve açısal hız değişiminin dikkate alındığı tasarımda oldukça başarılı olan sistem, araba konumu ve hızı için kısmen daha düşük bir performans

sergilemiştir. Bu handikabın aşılması için bulanık PI ve bulanık PD tarzında bileşik yapılardaki sistem tasarımları denenebilir.

Son olarak, geliştirilen durum geri besleme kontrolcüsünün, WinCon programı yardımı ile IP-02 ters sarkaç deney düzeneği üzerinde gerçek zamanda testlerinin yapılması olmuştur. Böylece Tüm Gelişim Ortamının (Total Development Environment) bir diğer kolu olan çevrim içi donanım içeren simülasyon çalışması da yapılmıştır.

Bundan sonra yapılacak çalışmalarda geliştirilen bu kontrolcüler iyileştirilebilir veya aynı fiziksel model üzerinde farklı kontrol stratejileri denenebilir. Sonuç olarak bu çalışma ile herhangi bir fiziksel model için hızlı kontrolcü prototiplendirme ve çevrim içi donanım içeren simülasyon metotları kullanılarak kontrolcü tasarımı yaklaşımı aktarılmaya çalışılmıştır.



KAYNAKLAR

- [1] **Feng, H. ,Törngren, M. and Eriksson, B.,** 1997. Experiences Using DSPACE Rapid Prototyping Tools for Real-Time Control Applications, *Proceedings of the DSP Scandinavia '97 Technical Conference*, SkyCity Stockholm, İsveç, Haziran, 3-4, 1997.
- [2] **Studt, T.,** 1998. Fast Designs for Control Systems, Research & Development Software & Systems, The MathWorks®, Inc Barrington, USA, 40 1-7.
- [3] **Hanselmann, H.,** 1996. Automotive Control: From Concept to Experiment to Product, *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, Dearborn, Michigan, Eylül, 15-18, 1996.
- [4] **Anonim,** 1999. Solutions for Control, www.cranesoftware.com, Cranes Software International Ltd., India.
- [5] **Anonim,** 2000. The Complete System for Controller Prototyping-dSPACE Prototyper, www.dspace.de, dSPACE GmbH, Germany, Ağustos, 9, 2000
- [6] **Anonim,** 1999. xPC Target 1.0-Real Time Rapid Prototyping and Development on PC Hardware, www.mathworks.com, The MathWorks®, Inc. USA, Kasım, 1999.
- [7] **Hanselmann, H.,** 1996. HardWare in the Loop Simulation Testing and its Integration into a CACSD Toolset, *Proceeding of the IEEE International Symposium on Computer Aided Control System Design*, Dearborn, Michigan, Eylül, 15-18, 1996.
- [8] **Güvenç, L.,** 2001. Kişisel Görüşme.
- [9] **Anonim,** 1997. Wincon 3.0.2a Real-time Digital Signal Processing and Control, Quanser Consulting, Inc.
- [10] **Anonim,** 1997. Advanced Teaching Systems-IP-02 Self Erecting Linear Motion Inverted Pendulum, Quanser Consulting, Inc.
- [11] **Ogata, K.,** 1997. Modern Control Engineering,.Third Edition, Prentice Hall, Inc.
- [12] **Ogata, K,** 1994. Designing Linear Control Systems with MATLAB®, Prentice Hall, Inc.
- [13] **Anderson, B.D.O., Moore, J.B.,** 1989. Optimal Control-Linear Quadratic Methods, Prentice Hall, International, Inc.
- [14] **Sarioğlu, M., K.,** 1985. Optimal Control Theory, *I.T.Ü Department of Electrical and Electronic Engineering*, İstanbul.

- [15] **Dabney, J. B., Harman, T.L.**, 1998. The Student Edition of SIMULINK®-Dynamic System Simulation for MATLAB, Prentice Hall, Inc.
- [16] **Wang, L.X.**, 1997. A Course in Fuzzy Systems and Control, Prentice Hall, Inc.
- [17] **Jantzen, J.**, 1998. Design of Fuzzy Controllers, *Technical University of Denmark, Department of Automation, Danimarka, Ağustos, 19, 1998.*
- [18] **Jantzen, J.**, 1998. Analysis of a Pendulum Problem, *Technical University of Denmark, Department of Automation, Danimarka, Ağustos, 19, 1998*
- [19] **Anonim.**, 1997. Control Tutorials for MATLAB-Modeling an Inverted Pendulum and State-Space Design for the Inverted Pendulum, *The University of Michigan, Carnegie Mellon, Aralık, 8, 1997.*
- [20] **Medrano-Cedra, G.,A.**, 1999. Robust Computer Control of an Inverted Pendulum, *IEEE Transactions on Control Systems*, Haziran, 1999 58-67
- [21] **Funabashi, H., Takeda, Y., Kenjo, Y., Kurai, K.**, 2000. Control of an Inverted Pendulum for Maintaining the State of Inversion Based on the Reference Trajectory, *Transactions on ISME*, Ser. C, Vol.66, No.643, pp.363-373, 2000



ÖZGEÇMİŞ

Kerem KIRCA, 1974 yılında Ankara'da doğdu. İlk öğrenimini 1985 yılında Konya, Akşehir Yıldırım İlkokulu'nda tamamladı. Aynı yıl İzmir Türk Kolejinde ve ardından İstanbul Atatürk Fen Lisesinde eğitime devam etti. 1993 yılında İstanbul Teknik Üniversitesi Makina Fakültesi'nde Makina Mühendisliği Lisans eğitime başlayan KIRCA, bölümünü 1998 yılında bitirdi. Savunma Teknolojileri Bilişim Programı'ndaki Yüksek Lisans eğitime 1998 yılında başladı.

