

Nicemli Kontrol GiriŖi ile Ters Sarkaç Yukarı Kaldırma Benzetimi

Zuhair Shakor Mahmood

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliđi Anabilim Dalı

Eylül 2012

Inverted Pendulum Swing-up Simulation With Quantized Control Inputs

Zuhair Shakor Mahmood

MASTER OF SCIENCE THESIS

Department of Electrical and Electronics Engineering

September 2012

Nicemli Kontrol GiriŖi ile Ters Sarkaç Yukarı Kaldırma Benzetimi

Zuhair Shakor Mahmood

EskiŖehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliđi Uyarınca
Elektrik Elektronik Mühendisliđi Anabilim Dalı
Kontrol ve Kumanda Sistemleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Abdurrahman KARAMANCIOĐLU

Eylül 2012

ONAY

Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Zuhair Shakor Mahmood'un YÜKSEK LİSANS tezi olarak hazırladığı "Nicemli Kontrol Girişi ile Ters Sarkaç Yukarı Kaldırma Benzetimi" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Prof. Dr. Abdurrahman KARAMANCIOĞLU

İkinci Danışman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye: Prof. Dr. Abdurrahman KARAMANCIOĞLU

Üye: Prof. Dr. Hasan Hüseyin ERKAYA

Üye: Prof. Dr. Osman PARLAKTUNA

Üye: Yrd. Doç. Dr. Bünyamin TAMYÜREK

Üye: Yrd. Doç Dr. Ahmet YAZICI

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve..... sayılı kararıyla onaylanmıştır.

Prof.Dr.Nimetullah BURNAK
Enstitü Müdürü

Özet

Bu çalışmada, ters sarkaç sisteminin, yukarı kaldırma problemi için nicemlenmiş denetim girişleri kullanılarak çeşitli benzetimler yapılmıştır. Sistem denge konumunun yakın bir komşuluğuna erişince denetim doğrusal karesel denetim ile yürütülmüştür.

Sistem modelinin geçerliliğini belirlemek için sistemin doğrusal olmayan modeli kullanılarak çeşitli benzetimler yapılmış ve elde edilen matematiksel çözümlerin fiziksel beklentilerle örtüştüğü gözlenmiştir.

Yapılan benzetimlerde nicemlenmiş denetimin başarılı sonuçlar verdiği gözlenmiştir.

Anahtar Kelimeler: Ters sarkaç, yukarı kaldırma, nicemlenmiş denetim

Summary

In this study, various simulations were done using quantized control inputs in order to address the swing-up problem of inverted pendulum system. Control was conducted applying a linear quadratic control when the system reaches to a point at the neighborhood of its balance position.

Various simulations were conducted on a non-linear model of the system to determine the validity of the system model and the obtained mathematical solutions were in correlation with the physical expectations.

Quantized control was found out to be successful based on the results of the performed simulations.

Keywords: Inverted pendulum, swing-up, quantized control

Teşekkür

Yüksek Lisans öğrenimim boyunca engin bilgi birikimleri ile desteklerini benden esirgemeyen, değerli deneyimleri ile hayat seyrime farklı bakış açıları katan Sayın Prof. Dr. Abdurrahman Hocam'a şükranlarımı sunar, danışmanım olarak kendisi ile çalıştığım-dan dolayı büyük bir gurur duyduğumu belirtirim.

Bu günlere gelmemde en büyük paya sahip olan, bana maddi manevi her türlü desteği veren sevgili annem Edibe Kervancı'ya ve hep yanımda olup hayatıma renk katan kardeşlerim Sena, Suzan, Sündüs, Haydar ve Arzu Kazancı'ya minnetimi sunarım.

Eskişehir'e uyum sağlamamda ve derslerime adapte olabilmemde her türlü teknik ve manevi desteklerinden ötürü Nevin Canik, Alpaslan Yufka, Kemal Keskin, Burak Urazel, Sinem Bozkurt, Burak Kaleci, Yıldray Anagün'e teşekkürler ederim.

İyi ve kötü günlerimde hep yanımda olan manevi desteklerini esirgemeyen, her daim yüzümün gülmesine vesile olan dostlarım Cansu Küçük, Ethem Açıkbaş, Mehmet Bal ve Abdulselem Ağaoğlu'na en içten teşekkürlerim takdim ederim.

Son olarak da benimle aynı yollardan geçen aynı zorluklara katlanan arkadaşım ve meslektaşım Zaid Amer'e teşekkür eder hayatında başarılar dilerim.

İçindekiler

Özet	v
Summary	vi
Teşekkür	vii
Şekiller Listesi	x
Tablo Listesi	xii
Semboller Listesi	xiii
1 GİRİŞ	1
1.1 Sarkaç Tarihçesi	1
1.2 Sarkaç	1
1.3 Sarkaç Sisteminin Uygulama Alanları	4
2 TERS SARKAÇ	5
2.1 Ters Sarkaç	5
2.2 Ters Sarkaç Sisteminin Doğrusal Olmayan Matematiksel Modeli	6
2.3 Sistemin Dinamik Denklemlerinin Çıkarılması	7
2.4 Doğrusallaştırma ve Transfer Fonksiyonları	9
2.5 Ters Sarkaç Sisteminin Durum Uzayı Formülasyonu	10
3 SİSTEM MODELİ DOĞRULAMA DENEYLERİ	11
3.1 Kontrol Problemi Tanıtımı	11

3.2	Ters Sarkaç Modelinin Geçerliği	11
4	DENETLEYİCİ TASARIMI	19
4.1	Denetleyici Tasarımı Aşamaları	19
4.2	Enerji Kontrolü	19
4.3	Stabilizasyon Aşaması	20
4.4	Sistemin Doğrusal Modeli	23
4.5	Sonlu Sayıda Büyüklüklü Denetim Giriş Sinyalleri ile Simülasyon	25
5	İRDELEME VE SONUÇLAR	33
5.1	İrdeleme ve Sonuçlar	33
6	EKLER	37
A	Ters Sarkaçta Kullandığımız Denklemler TSdenklem.m	38
B	Ters Sarkaç ana Program Denklemleri Test etmek için TSana.m	39
C	Ters Sarkaç ana Program Denklemleri Test etmek için myequation.m	41
D	Ters Sarkaç <i>LQR</i> Yönteminin Sıfır Açıya Çekmek Testi mymain.m	42
E	Ters Sarkaçta Kullandığımız Denklem invpend.m	45
F	Ters Sarkacı Tek Kademeli Yukarı Kaldırma Programı Swing04.m	46
G	Ters Sarkacı İki Kademeli Yukarı Kaldırma Programı Swing05.m	55

Şekil Listesi

1.1	Basit Sarkaç	2
1.2	Basit Sarkaç Dinamik Denklemleri MATLAB Kodları: Sdenkem.m	2
1.3	Basit Sarkaç Dinamik Denklemleri MATLAB Kodları: Sana.m	3
1.4	Sarkaç Programı Grafik Çıktıları	4
2.1	Ters Sarkaç Sistemi	6
2.2	Ters Sarkaç Sistemine Etkiyen Kuvvetler	6
3.1	Başlangıç Davranışları Arayüzü	13
3.2	Ters Sarkaç Deney 1 Grafikleri	13
3.3	Ters Sarkaç Deney 2 Grafikleri	14
3.4	Ters Sarkaç Deney 3 Grafikleri	15
3.5	Ters Sarkaç Deney 4 Grafikleri	16
3.6	Ters Sarkaç Deney 5 Grafikleri	17
3.7	Ters Sarkaç Deney 6 Grafikleri	18
4.1	Ters Sarkaç Dinamik Denklemleri MATLAB Kodları: TSdenkem.m	23
4.2	Ters Sarkaç Dinamik Denklemleri MATLAB Kodları: TSana.m	24
4.3	Swing Programının Grafik Çıktıları ($A = \pm 2$)	25
4.4	Swing Programının Grafik Çıktıları ($A = \pm 3$)	26
4.5	Swing Programının Grafik Çıktıları ($A = \pm 4$)	26
4.6	Swing Programının Grafik Çıktıları ($A = \pm 5$)	27
4.7	Swing Programının Grafik Çıktıları ($A = \pm 7$)	27
4.8	Swing Programının Grafik Çıktıları ($A = \pm 8$)	28
4.9	Swing Programının Grafik Çıktıları ($A = \pm 9$)	28

4.10 İki Kademeli Yaklaşımında $A = \pm 10$ için Elde Edilen Grafikler	29
4.11 İki Kademeli Yaklaşımında $A = \pm 11$ için Elde Edilen Grafikler	30
4.12 İki Kademeli Yaklaşımında $A = \pm 15$ için Elde Edilen Grafikler	30
4.13 İki Kademeli Yaklaşımında $A = \pm 19$ için Elde Edilen Grafikler	31
4.14 İki Kademeli Yaklaşımında $A = \pm 20$ için Elde Edilen Grafikler	31
4.15 İki Kademeli Yaklaşım $A = \pm 25$ için Elde Edilen Grafikler	32
4.16 İki Kademeli Yaklaşım $A = \pm 30$ için Elde Edilen Grafikler	32

Tablo Listesi

3.1	Ters Sarkaç için Parametrelerin Tanımları ve Değerleri	12
3.2	Altı Adet Deneyin Koşulları	12
4.1	Swing-up Programındaki Parametrelerin Tanımları ve Değerleri	22
5.1	Tek Kademeli Yukarı Kaldırma için Deney Özetleri Tablosu	33
5.2	İki Kademeli Yukarı Kaldırma için Deney Özetleri Tablosu	34

Semboller Listesi

g	Yerçekimi İvmesi	2
l	Sarkaç Uzunluğu	2
θ	Sarkacın Düşey Eksenine ile Yapılan Açık	2
$\dot{\theta}$	Sarkacın Açısal Hızı	3
M	Arabanın Kütlesi	5
m	Sarkacın Kütlesi	5
b	Sürtünme Katsayısı	5
x	X Eksenine ve Arabanın Konumu	5
u	Arabaya Uygulanan Kuvvet	5
y	Y Eksenine	5
\dot{x}	Arabanın Hızı	6
L	Lagragian	7
KE	Kinetik Enerji	7
PE	Potansiyel Enerji	7
KE_m	m Sarkaç Kütlesine Ait Kinetik Enerji	7
V_c	Arabanın Hızı	7
ω	Sarkacın Hızı	7
r_c	Konum Vektörünün Türevi	7
KE_M	M Aracın Kütlesine Ait Kinetik Enerji	8
\ddot{x}	Arabanın İvmesini	9
J	Sarkacın Bağlantı Noktası Etrafındaki Eylemsizlik Momenti	20
E	Enerji	20
\dot{E}	E 'nin zamana göre türevi	20

BÖLÜM 1

GİRİŞ

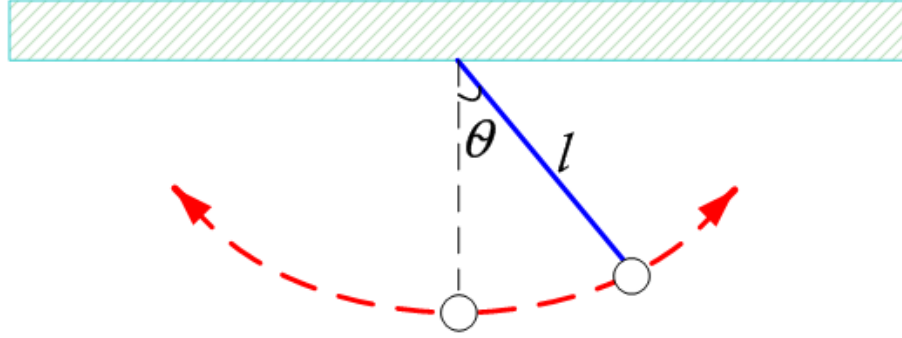
1.1 Sarkaç Tarihçesi

Sarkaç salınım hareketi üzerine ilk çalışma onuncu yüzyılda İbn-i Yunus tarafından yapılmış bu alanda yeni çalışmalara ışık tutmuştur [1] [3]. Sonrasında Galileo zamanı ölçmek için kullanmıştır. XVII. yüzyıl fizikçilerinden Huygens, Newton, Hooke; Galileo'nun gözlemleri ve çalışmaları ışığında sarkaç salınım hareketi konusunda çeşitli çalışmalar yapmıştır [1] [3].

Batıda bilimin gelişmesinde çok önemli rol oynayan sarkaç sistem; çarpışma konuları, kütlelerin kanunları, yerçekimine göre hızlanma, ekvator ve kutup bölgelerindeki yerçekiminin değişmesinin keşfiyle orantılı dünyanın şeklinin belirlenmesi gibi konularda adeta bir temel kaynak olarak görülmüştür [2] [3].

1.2 Sarkaç

Sarkaç; bir ipe veya çubuğa bağlı bir kütledir Şekil 1.1. Denge noktası dışında serbest bıraktığımızda bağlantı noktasından geçen düşey çizgi etrafında simetrik salınımlar yapar ve sistemde sürtünme varsa zamanla yavaşlayarak en aşağıdaki konumunda durur. Durduğu bu noktaya denge noktası denir. Bu nokta istisnai bir nokta olup bu noktada serbest bırakılması halinde kütle hareket etmez. Sarkaç sisteminin analizi ileride ele alacağımız ters sarkaç dinamiğini daha iyi anlamamızı sağlar.



Şekil 1.1: Basit Sarkaç

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin \theta = 0 \quad (1.2.1)$$

Sarkaç sistemini temsil eden diferansiyel denklem, aşağıda verilmektedir. Bu denklemde l sarkaç uzunluğunu, g yer çekimini, θ sarkacın düşey eksen ile yaptığı açığı göstermektedir [4].

Yukarıdaki denkleme ait MATLAB kodları aşağıda Program 1'de verilmektedir. (Program 1, iki dosyaya kayıtlıdır. Dosya isimleri *Sdenklem.m* ve *Sana.m* dir).

```
function xdot=Sdenklem(t,x)
g=9.81;L=0.5
xdot=[x(2);-g/L*sin(x(1))]
```

Şekil 1.2: Basit Sarkaç Dinamik Denklemleri MATLAB Kodları: Sdenklem.m

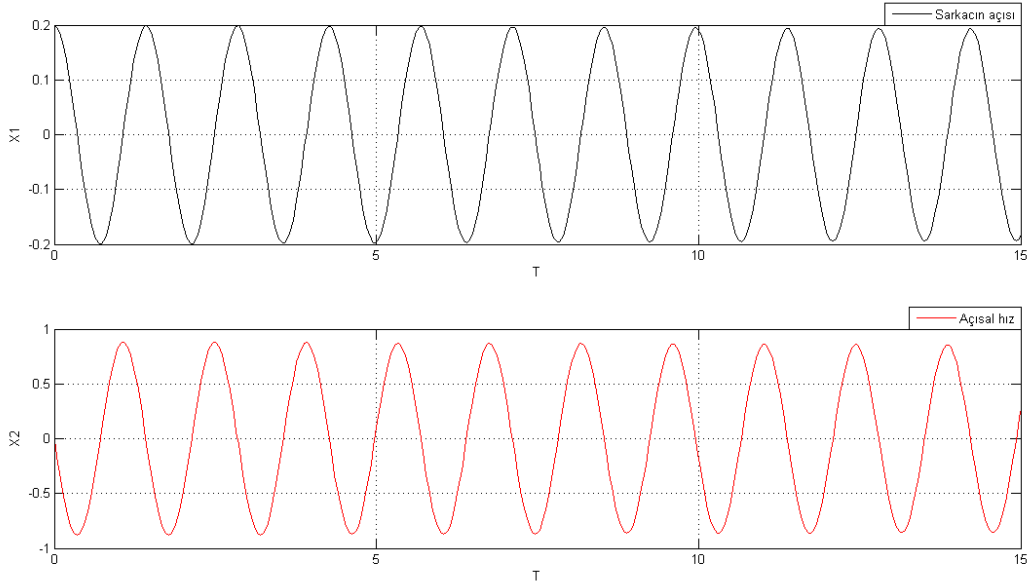
```

[t,x]=ode23('Sdenklem',[0,15],[0.2,0]);
subplot(2,1,1)
plot(t,x(:,1),'k')
grid
xlabel('T')
ylabel('X1')
legend('Sarkacın açısı')
subplot(2,1,2)
plot(t,x(:,2),'r')
grid
xlabel('T')
ylabel('X2')
legend('Açısal hız')

```

Şekil 1.3: Basit Sarkaç Dinamik Denklemleri MATLAB Kodları: Sana.m

Bu programın $(\theta, \dot{\theta}) = (0.2, 0)$ başlangıç durumundan itibaren $t = 0$ ile 15 sn arasında çalıştırılması sonucunda Şekil 1.4'deki grafikler elde edilir. (Bu grafiklerde $\theta := x_1$ ve $\dot{\theta} := x_2$ sırasıyla sarkaç açısını ve açısal hızını ifade etmektedir). Grafiklerde sarkaç açısı $+0.2$ ve -0.2 radyan arasında periyodik olarak değişmektedir ve bu grafikler sarkacın salınımının beklentilerimize uygun gerçekleştiğini göstermektedir.



Şekil 1.4: Sarkaç Programı Grafik Çıktıları

1.3 Sarkaç Sisteminin Uygulama Alanları

Sarkaç sistemi geçmişten günümüze birçok alanda hayatımızı kolaylaştırılmıştır. İlk kullanım alanlarından biri duvar saatleridir. Zaman periyodu 2 sn. olarak verilen ve saniye sarkacı olarak da bilinen sarkacın her salınışında 2 sn. süre geçmekte ve sarkacın merkezine bağlı olduğu noktadaki dişliler aracılığıyla 60 saniyede bir derece yelkovan, yelkovanın 360 derecelik hareketinde de akrep bir saate karşılıklı gelecek şekilde hareket ederek zamanı ölçmeyi sağlamaktadır.

Dünyanın farklı bölgelerinde ve farklı noktalarında farklı yerçekimi değerleri olduğunun ölçümü de yine sarkaç sayesinde gözlenebilmektedir. Bunun yanı sıra hareketinin grafiğe dökülerek kullanıldığı sismometre gibi uygulamaları da mevcuttur.

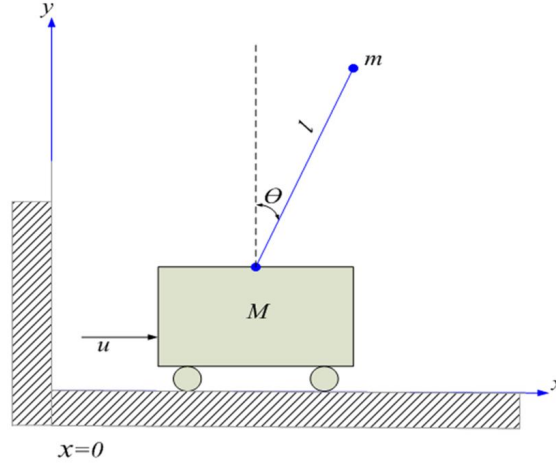
Sarkaç sistemlerin çift sarkaç ya da eş sarkaçlar kullanılarak belirli başlangıç şartlarında kaotik hareketlerin incelenmesi konusunda da çalışmalar bulunmaktadır. Aynı zamanda atalet yer gösterici sistemlerin tasarımında mutlaka kullanılması gereken Schuler ayarlama metodu da sarkaç mantığından hareketle oluşturulmuştur [5].

BÖLÜM 2

TERS SARKAÇ

2.1 Ters Sarkaç

Şimdi işi biraz daha zorlaştıralım; yerçekimi etkisi ile uzun süreli salınımlar sonrasında denge noktasında sabit hale gelen sarkacımızın yatay eksene göre simetriğini aldığımızda ters sarkacı elde ederiz. Şekil 2.1’de gösterildiği üzere ters sarkaç sistemi temel olarak hareket eden M kütlede bir araba ve üzerinde l uzunluğuna ve m kütleline sahip bir sarkaçtan oluşmaktadır. Arabanın üzerindeki sarkaca etki eden yerçekimi ivmesini g ile göstermekteyiz. Araba, hareket ettiği yönün ters yönde, sürtünme katsayısı b olan bir sürtünme kuvvetinin etkisi altındadır. Arabaya sadece x ekseninde kuvvet uygulanabilmektedir. Bu kuvvet u ile gösterilmekte olup denetim girişidir. Sarkacın ve arabanın hareketleri (x, y) düzlemi ile sınırlıdır. Modellemede basitlik için sarkacın kütlelerinin sarkaç çubuğunun üst uç noktasında yoğunlaştığı kabul edilmektedir.

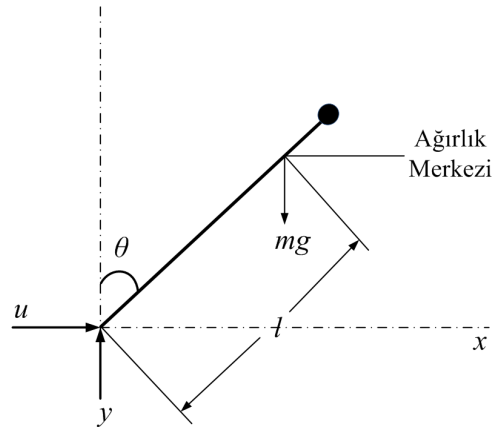


Şekil 2.1: Ters Sarkaç Sistemi

Sistem modelindeki x arabanın konumu ve θ sarkacın açısı olup kontrol edilen değişkenlerdir. Kontrol edilen değişken sayısı giriş sayısından fazla olduğu için ters sarkaç sistemi yetersiz uyarımlı sistem olarak sınıflandırılır. Sistemin durumları $x, \dot{x}, \theta,$ ve $\dot{\theta}$ olarak belirlenmiştir. Durumların daha ayrıntılı tanımlamaları için [5] ve [6]'ye müracaat edilebilir.

2.2 Ters Sarkaç Sisteminin Doğrusal Olmayan Matematiksel Modeli

Şekil 2.2'de sistem, araba ve sarkaç olmak üzere iki kısma bölünmüş, araba ve sarkaca etkiyen kuvvetler gösterilmiştir.



Şekil 2.2: Ters Sarkaç Sistemine Etkiyen Kuvvetler

Sarkacın dip kısmına yatay olarak etkiyen ve Şekil 2.2’de u ile gösterilen kuvvet, m sarkacın kütlesi, x arabanın konumu olmak üzere aşağıdaki şekilde ifade edilebilir.

$$u = m \frac{\partial^2}{\partial t^2} (x + l \sin \theta) \quad (2.2.1)$$

Ağırlık merkezinin ivmelenmesinden kaynaklanan bu kuvvetin düşey bileşeni ise

$$y = m \frac{\partial^2}{\partial t^2} (l \cos \theta) + mg \quad (2.2.2)$$

biçiminde yazılabilir[7].

2.3 Sistemin Dinamik Denklemlerinin Çıkarılması

İki serbestlik derecesine sahip bu sistemin modeli Lagrangian dinamikleri kullanarak elde edilmiştir. Aşağıda Lagrangian eşitliklerini kullanarak hareket denklemlerini, elde etmekteyiz. Sürtünme olmadığını varsayalım. Bu durumda enerji kaybının olmadığı kabul edilir. Lagrangian eşitlikleri kullanan bir sistemin dinamiklerini elde etmek için gerekli matematiksel eşitlikler, aşağıda verilmektedir.

Kinetik ve potansiyel enerji denklemleri arasındaki fark L olarak tanımlansın:

$$L = KE - PE \quad (2.3.1)$$

Sarkaç için kinetik enerji şu şekilde hesaplanır:

$$KE_m = \frac{1}{2} m V_c^2 + \frac{1}{2} l \omega^2 \quad (2.3.2)$$

burada ω sarkacın açısal hızı, V_c ise arabanın hızıdır ve r_c konum vektörünün türevi alınarak elde edilen yatay eksenindeki hızıdır [8].

$$r_c = (x - l \sin \theta) \hat{i} - l \cos \theta \hat{j} \quad (2.3.3)$$

$$V_c = \frac{dr_c}{dt} = (\dot{x} - l \cos \theta \dot{\theta}) \hat{i} + l \sin \theta \dot{\theta} \hat{j} \quad (2.3.4)$$

Açısal hız ise basitçe:

$$\omega = \dot{\theta} \quad (2.3.5)$$

(2.3.4) ve (2.3.5) denklemleri Denklem(2.3.2)'de yerine konulup düzenlenirse:

$$KE_m = \frac{1}{2}m(\dot{x}^2 - 2\dot{x}l \cos \theta \dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2}l\dot{\theta}^2 \quad (2.3.6)$$

Toplam kinetik enerjiyi bulmak için m sarkaç kütesine ait kinetik enerji ile M aracın kütesine ait kinetik enerjileri toplarsak

$$KE = KE_M + KE_m = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 - 2\dot{x}l \cos \theta \dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2}l\dot{\theta}^2 \quad (2.3.7)$$

ifadesini elde ederiz.

Araba yalnızca yatay ekseninde hareket ettiği için, sistemin potansiyel enerjisini sarkaç belirler. Potansiyel enerji

$$PE = -mgl \cos \theta \quad (2.3.8)$$

olarak ifade edilebilir.

Lagrangian eşitliğinde bulduğumuz enerji denklemlerini yerlerine yazarsak denklem aşağıdaki forma kavuşacaktır.

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 - 2\dot{x}l \cos \theta \dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2}l\dot{\theta}^2 + mgl \cos \theta \quad (2.3.9)$$

Sistemde arabaya ve salınım dinamiklerine etki eden sadece bir adet dış kuvvet bulunmaktadır. Bu da sisteme yatay ekseninde hareket kazandıran u kuvvetidir. θ koordinat sisteminde herhangi bir dış kuvvet sarkaca etki etmemektedir. Sistemin Lagrangian eşitliği aşağıdaki gibi yazılabilir:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = u \quad (2.3.10)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0 \quad (2.3.11)$$

İfade (2.3.9)'daki L'ye Lagrangian eşitliklerini uygularsak aşağıdaki Denklemleri elde ederiz [8] [9].

$$\begin{aligned} (M + m)\ddot{x} - ml\dot{\theta}^2 \sin \theta + ml\ddot{\theta} \cos \theta + b\dot{x} &= u \\ m\ddot{x} \cos \theta + ml\ddot{\theta} - mg \sin \theta &= 0 \end{aligned} \quad (2.3.12)$$

2.4 Doğrusallaştırma ve Transfer Fonksiyonları

Denklem (2.3.12) ile elde ettiğimiz sistemi tanımlayan denklemlerin doğrusal olmadığı görülmektedir ve bu sebeple analiz, benzetim ve kontrolör tasarımının yapılabilmesi için denklemleri doğrusallaştırılma gerekmektedir. Sistem iki denge noktasına sahiptir: $\theta = 0$ (sarkaç aşağı yönde ve sistem kararlı) ve $\theta = \pi$ (sarkaç yukarı yönde ve sistem kararsız). Doğrusal sistemlerin analizi ve bunlar için denetleyici tasarımı belirli bir olgunluğa erişmiş ve iyi anlaşılmış bir konu olduğundan sarkacı dikey konumunda dengeleme tasarımında doğrusal yaklaşığında faydalanılmaktadır. Ters sarkaç sisteminde kontrol amacı sarkacı dikey tutmak olarak belirlendiği durumu ele alalım. Bu durumda ters sarkaç sisteminde kontrol amacına uygun yörüngelerinin $(x_3, x_4) = (0, 0)$ çalışma noktası komşuluğuna ulaşması beklenmektedir. Yüksek dereceli terimler ihmal edilerek 1. dereceden Denklem (2.4.1) ile ifade edilen Taylor serisi açılımı Denklem (2.3.12)'ye uygulanırsa [8].

$$f(\theta) \cong f(\theta_0) + \left. \frac{df}{d\theta} \right|_{\theta=\theta_0} (\theta - \theta_0) \quad (2.4.1)$$

$\theta = 0$ noktası için:

$$\begin{aligned} (M + m)\ddot{x} + b\dot{x} - ml\ddot{\theta} &= u \\ -ml\ddot{x} + (ml^2 + l)\ddot{\theta} + mgl\theta &= 0 \end{aligned} \quad (2.4.2)$$

Transfer fonksiyonunu bulunabilmesi için doğrusallaştırılan Denklem (2.4.2) ve (2.4.4)'ün Laplace dönüşümü alınır ve l atalet momentinin sıfır olduğu varsayılarak gerekli düzenlemeler yapılırsa Denklem (2.4.3) ve (2.4.5) ile ifade edilen transfer fonksiyonları elde edilir.

$$\begin{aligned}\frac{X(s)}{F(s)} &= \frac{m l s^2 + s + m g l}{M m l^2 s^4 + \left[(M+m) + m l^2 b \right] s^3 + \left[(M+m) m g l + b \right] s^2 + m g l b s} \\ \frac{\theta(s)}{F(s)} &= \frac{m l s}{M m l^2 s^3 + \left[(M+m) + m l^2 b \right] s^2 + \left[(M+m) m g l + b \right] s + m g l b}\end{aligned}\quad (2.4.3)$$

$\theta = \pi$ noktası için:

$$\begin{aligned}(M+m)\ddot{x} + b\dot{x} - m l \ddot{\theta} &= u \\ m l \ddot{x} + (m l^2 + l)\ddot{\theta} - m g l \theta &= 0\end{aligned}\quad (2.4.4)$$

$$\begin{aligned}\frac{X(s)}{F(s)} &= \frac{m l s^2 + s - m g l}{M m l^2 s^4 + \left[(M+m) + m l^2 b \right] s^3 - \left[(M+m) m g l - b \right] s^2 - m g l b s} \\ \frac{\theta(s)}{F(s)} &= \frac{-m l s}{M m l^2 s^3 + \left[(M+m) + m l^2 b \right] s^2 - \left[(M+m) m g l - b \right] s - m g l b}\end{aligned}\quad (2.4.5)$$

2.5 Ters Sarkaç Sisteminin Durum Uzayı Formülasyonu

Denklem (2.3.12)'de elde ettiğimiz sistemi tanımlayan denklemlerin doğrusal olmadıkları görülmektedir, aynı zaman da bu denklemlerde ikinci mertebeden türevler olduğundan ve bu sebeple analiz, benzetme ve kontrolör tasarımının yapılabilmesi için denklemlerin durum uzayı formunda ifade edilmeleri gereklidir. $x_1 := x, x_2 := \dot{x}, x_3 := \theta, x_4 := \dot{\theta}$ olarak kabul edilirse, ters sarkaç sisteminin durum uzayı gösterimi şu şekilde olur.

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-b x_2 + m l (x_4)^2 \sin(x_3) - m g \sin(x_3) \cos(x_3) + u}{(M+m-m \cos^2(x_3))} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{b x_2 \cos(x_3) + (M+m) g \sin(x_3) - m l (x_4)^2 \cos(x_3) \sin(x_3) - u \cos(x_3)}{l (M+m-m \cos^2(x_3))}\end{aligned}\quad (2.5.1)$$

Denklem (2.5.1) *ode23.m* ile nümerik çözüm için uygun formattadır [12].

BÖLÜM 3

SİSTEM MODELİ DOĞRULAMA DENEYLERİ

3.1 Kontrol Problemi Tanıtımı

Bu çalışmanın amacı, ters sarkacı aşağıdaki denge konumundan yukarıdaki denge konumuna getiren ve ardından bu konumun komşuluğunda tutan algoritmalar üretmektir. Bu amaca yönelik olarak ilk önce (2.5.1) numaralı denklem ile ters sarkaç dinamik modelinin doğruluğunu göstereceğiz. Bunu takiben ters sarkacın yukarı denge konumu komşuluğundaki konumlardan başlayan yörüngelerin bu denge konumuna getirilmesi (stabilizasyon) algoritmaları oluşturup simüle edeceğiz. En son olarak da aşağıdaki denge konumundaki sarkacı yukarıdaki denge konumuna getiren algoritmayı simüle edeceğiz. Bu algoritma sonuçlar kısmında değerlendirilmektedir.

3.2 Ters Sarkaç Modelinin Geçerliliği

Önceki bölümde (2.5.1) ile verilen ters sarkaç denklemini MATLAB’te uygulanan girişi ve başlangıç şartlarını değiştirerek altı temel deney yapacağız. Bu deneylerde, grafiklerde göreceğimiz bilgiler fiziksel gerçekler ile uyumalıdır. Bu durumda bu denklemlerin doğruluğu sağlanmış olacaktır. Deneylerde kullandığımız ters sarkaç parametre değerleri Tablo 3.1’de verilmektedir:

Tablo 3.1: Ters Sarkaç için Parametrelerin Tanımları ve Değerleri

Parametre	Sembol	Değer	Birim
Arabanın kütlesi	M	2	kg
Sarkacın kütlesi	m	0.25	kg
Sarkaç kolu uzunluğu	l	0.6	m
Sürtünme katsayısı	b	1.5	kg/sn
Yerçekimi ivmesi	g	9.81	m/sn^2

Bu deneylerin hepsinde dört durum değişkeninin grafiklerini elde ederek fiziksel beklentimizle karşılaştıracamız. Bu deneylere ait MATLAB program kodları şablonu Program 2a ve Program 2b ile verilmektedir. Program çalışmaya başladığında öncelikle başlangıç şartlarını ve uygulanan girişi aşağıdaki şekilde sormaktadır.

Tablo 3.2: Altı Adet Deneyin Koşulları

Deney Adı	Başlangıç şartları $X(0)$	Giriş u
Deney 1	(0 0 0.1 0)	0
Deney 2	(0 0 -0.1 0)	0
Deney 3	(0 0 0 0)	1
Deney 4	(0 0 0 0)	-1
Deney 5	(0 0 π 0)	1
Deney 6	(0 0 π 0)	-1

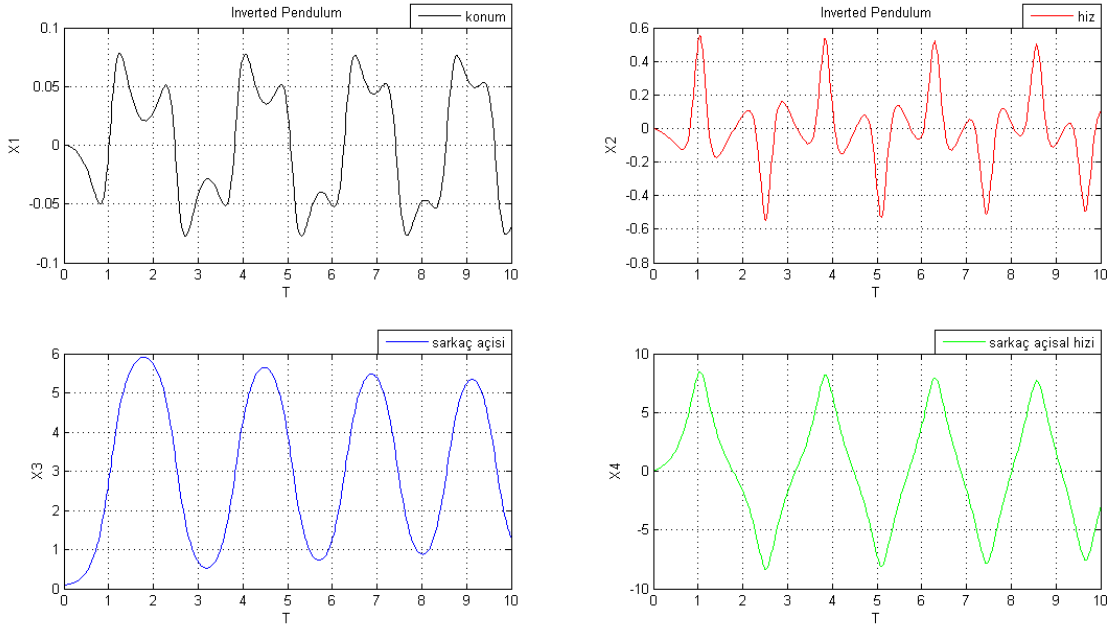
```

u degeriniz giriniz=0
x1 degeriniz giriniz=0
x2 degeriniz giriniz= 0
x3 degeriniz giriniz=0.1
x4 degeriniz giriniz=0
baslangiç zamani giriniz=0
bitis zamani giriniz=10

```

Şekil 3.1: Başlangıç Davranışları Arayüzü

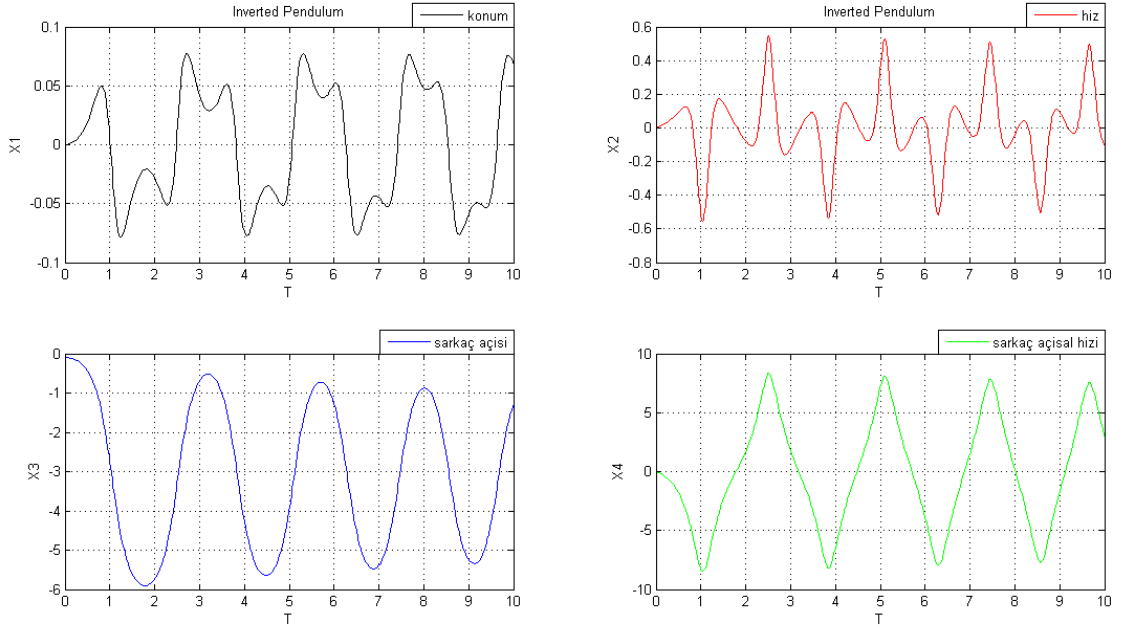
Deney 1'deki fiziksel beklentilerimiz: θ 'nın büyüyerek $2\pi - 0.1$ radyan açısı komşuluğuna ulaşması sonra da azalarak başlangıç açısı 0.1'e yaklaşması ve bu salınımların küçülerek devam etmesidir. Araç konumu da sarkaçtaki düşüştten etkilenerek uyumlu değişiklik göstermelidir. Deney 1'e ait program çıktıları Şekil 3.2'te verilmektedir. Değiştirdiğimiz başlangıç şartlarına göre girilen değerler EK A ve EK B'de belirtilmiştir.



Şekil 3.2: Ters Sarkaç Deney 1 Grafikleri

Şekilde de görüleceği üzere Deney 1 beklediğimiz fiziksel tepkileri vermektedir.

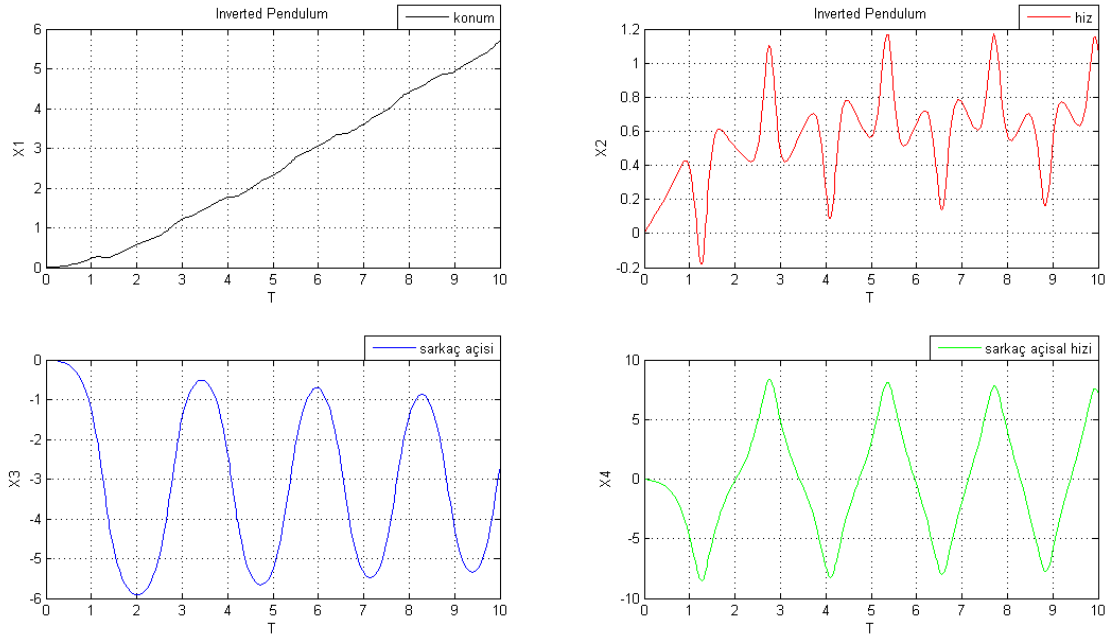
Deney 2'deki fiziksel beklentilerimiz: Deney 1 deki sarkaç açısal hareketinin ve araç konumunun ters yönlüsüdür. Deney 2'ye ait program çıktıları Şekil 3.3'te verilmektedir.



Şekil 3.3: Ters Sarkaç Deney 2 Grafikleri

Şekilde de görüleceği üzere Deney 2 beklediğimiz fiziksel tepkileri vermektedir.

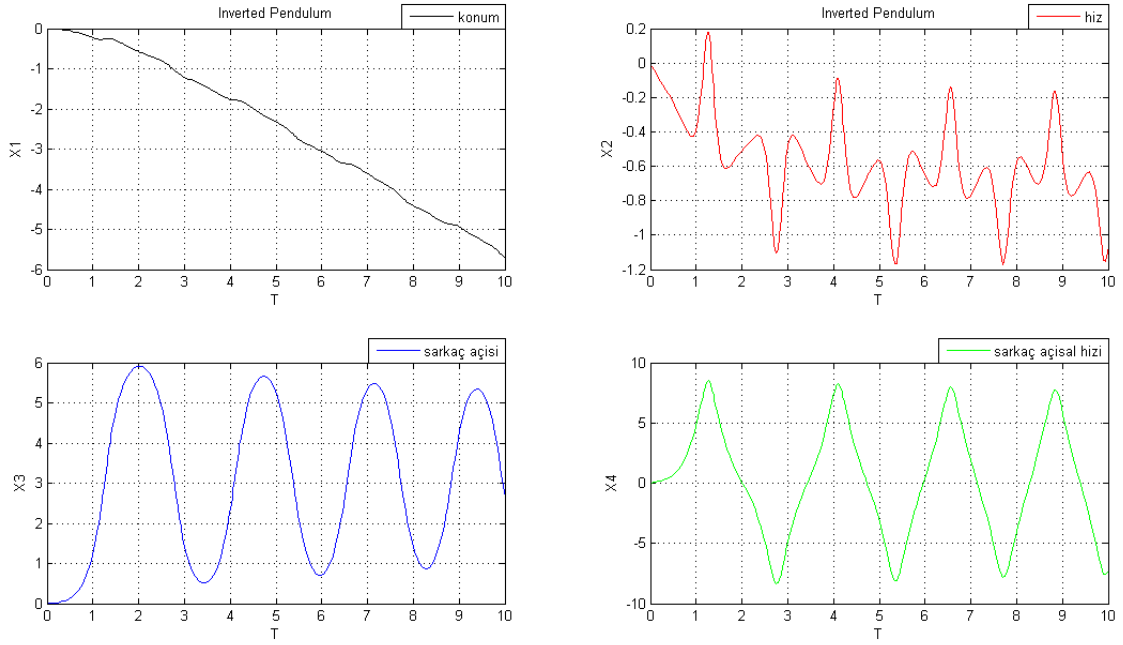
Deney 3'te fiziksel beklentimiz: aracın pozitif x yönünde hareket etmesidir. Bu esnada sarkacın saat yönünün tersinde harekete başlaması beklenir. Deney 3'e ait program çıktıları Şekil 3.4'te verilmektedir.



Şekil 3.4: Ters Sarkaç Deney 3 Grafikleri

Şekilde de görüleceği üzere Deney 3 beklediğimiz fiziksel tepkileri vermektedir.

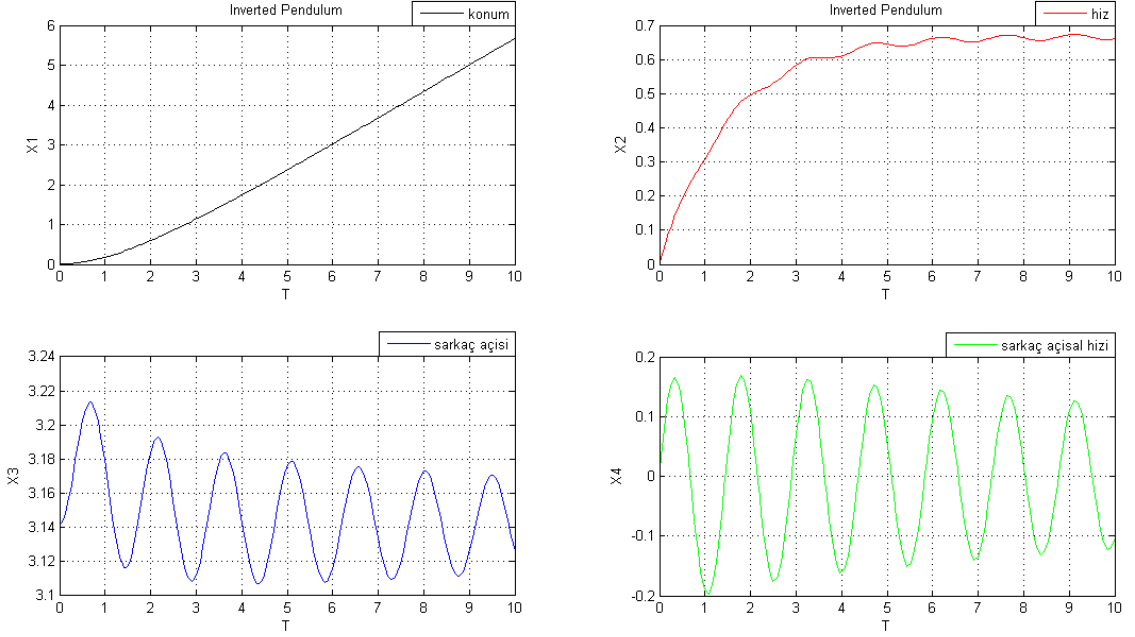
Deney 4'te fiziksel beklentimiz: Deney 3 teki hareketlerin ters yönlüleridir. Deney 4'e ait program çıktıları Şekil 3.5'de verilmektedir.



Şekil 3.5: Ters Sarkaç Deney 4 Grafikleri

Şekilde de görüleceği üzere Deney 4'te fiziksel beklentilerimize ulaşılmaktadır.

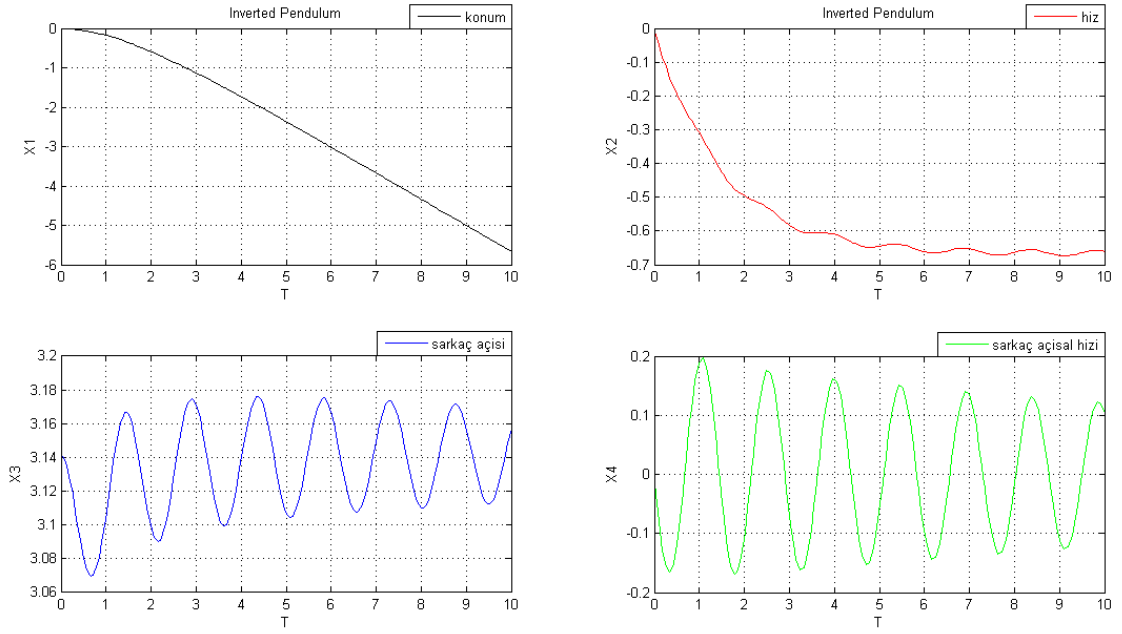
Deney 5'te fiziksel beklentimiz: Ters sarkacın uç noktasında arabanın hareket yönünün tersi yönünde ilk hareketi beklenilmektedir. Deney 5'e ait program çıktıları Şekil 3.6'de verilmektedir.



Şekil 3.6: Ters Sarkaç Deney 5 Grafikleri

Şekilde de görüleceği üzere Deney 5 de fiziksel beklentilerimize ulaşılmaktadır.

Deney 6'da fiziksel beklentimiz: Ters sarkacın uç noktası arabanın hareket yönünün tersi yönünde ilk hareketi beklenilmektedir. Deney 6'ya ait program çıktıları Şekil 3.7'da verilmektedir.



Şekil 3.7: Ters Sarkaç Deney 6 Grafikleri

Şekilde de görüleceği üzere Deney 6 da fiziksel beklentilerimize ulaşılmaktadır.

BÖLÜM 4

DENETLEYİCİ TASARIMI

4.1 Denetleyici Tasarımı Aşamaları

Ters sarkaç sistemi aşağı denge noktası $(\theta, \dot{\theta}) = (\pi, 0)$ ve yukarı denge noktası $(\theta, \dot{\theta}) = (0, 0)$ olmak üzere iki denge noktasına sahiptir. Bu sistem için kontrol amacımızı iki aşamada gerçekleştireceğiz. Birinci aşamada kullanacağımız kaldırma algoritması ile aşağı denge konumundaki sarkacı, yukarı denge konumunun yakınlıklarına getireceğiz. İkinci aşamada kullanacağımız stabilizasyon algoritması ile yukarı denge noktası yakınlığındaki sarkacı, tepe denge noktasına tam olarak getirmeye çalışacağız.

4.2 Enerji Kontrolü

Çoğu görev, sarkacın yer ve hız kontrolü yerine, enerji kontrolü tarafından tamamlanır. Örneğin, sarkacı kararsız dikey konuma getirmenin bir yolu, ona dikey konumdayken eşdeğer olacak enerjiyi vermektir. Ters sarkaç yukarı kaldırma (swing-up) problemi, sarkacı kararlı aşağı denge konumundan kararsız yukarı denge konumunun belirli bir komşuluğuna uygun denetim girdileri uygulanarak getirilmesi şeklinde tanımlanabilir. Uygun denetim girişlerini bulmak için fiziksel enerji prensiplerinden yararlanacağız. Herhangi bir denetim girişi uygulanmayan bir ters sarkacın enerjisi, verilen bir $(\theta, \dot{\theta})$ çifti için şu şekilde yazılabilir [9]:

$$E = K_E + P_E = \frac{1}{2}J\dot{\theta}^2 + mgl(\cos\theta - 1) \quad (4.2.1)$$

Burada K_E kinetik enerji, P_E potansiyel enerjiyi ve $J := ml^2$ sarkacın bağlantı noktası etrafındaki eylemsizlik momentini ifade etmektedir. Sarkacın enerji değişiminin ifadesi, Denklem (4.2.1) ile verilen ifadenin zamana göre türevi alınarak yazılabilir. Enerji kontrolünü gerçekleştirmek için denetim girişi tarafından enerjinin nasıl etkilendiğini anlamak gereklidir. Bu amaçla E 'nin zamana göre türevini hesaplırsak

$$\dot{E} = -ml\dot{x}_2x_4 \cos(x_3) \quad (4.2.2)$$

elde edilir [10]. Burada sistem dinamik denklemleri (2.5.1) kullanılırsa

$$\dot{E} = -mlx_4 \cos(x_3) \frac{-bx_2 + ml \sin(x_3)(x_4)^2 - mg \sin(x_3) \cos(x_3) + u}{(M + m - m \cos^2(x_3))} \quad (4.2.3)$$

elde edilir. Bu ifade enerjideki değişimi u denetim girişinin her fonksiyonu olarak ifade etmektedir. Biz, u denetim girişini, bu değişimi artan yönde yapacak şekilde belirleyeceğiz.

4.3 Stabilizasyon Aşaması

Stabilizasyon ters sarkacın, yukarı denge noktasının komşuluğundan denge noktasına ulaşması ve bu noktaya yeterince yakın olarak muhafaza edilmesi aşamasıdır. Bu aşamada ters sarkaç sisteminin denge noktası etrafındaki doğrusallaştırılmış modelinden yararlanılacaktır. Doğrusallaştırılmış modele göre tasarlanacak geri besleme katsayılarının belirleyeceği denetim girdileri bu amaca hizmet edecektir.

Ters sarkacın doğrusal olmayan durum uzayı modeli Denklem (2.5.1) ile verilmiştir:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{-bx_2 + ml(x_4)^2 \sin(x_3) - mgs \sin(x_3) \cos(x_3) + u}{(M+m-m \cos^2(x_3))} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{bx_2 \cos(x_3) + (M+m)g \sin(x_3) - ml(x_4)^2 \cos(x_3) - \sin(x_3) - u \cos(x_3)}{l(M+m-m \cos^2(x_3))}
\end{aligned} \tag{2.5.1}$$

Burada $x_1 := x, x_2 := \dot{x}, x_3 := \theta, x_4 := \dot{\theta}$ ve $X = [x_1, x_2, x_3, x_4]^T$ olarak tanımlanmıştır. Bu ifade daha kısa bir gösterimle

$$\dot{X} = f(X) + h(X) \tag{4.3.1}$$

şeklinde yazılabilir. Bu ifadeyi doğrusal doğrusal durum uzayı araçlarından yararlanabilmek için aşağıdaki şekilde doğrusal hale getirmek istemekteyiz:

$$\dot{x} = Ax + Bu \tag{4.3.2}$$

Denklemleri durum uzayı formunda yazabilmek için Denklem (4.4.3)'deki A ve B matrislerinin belirlenmesi gereklidir. Stabilizasyon problemi için ters sarkacın $x_3 = 0, x_4 = 0$ komşuluğundaki davranışları önemli olduğunu için bu nokta etrafında doğrusallaştırma yapacağız. Bu durumda

$$\begin{aligned}
A &= \left. \frac{\partial f}{\partial x} \right|_{(x_3, x_4)=(0,0)} \\
B &= \left. \frac{\partial f}{\partial u} \right|_{(x_3, x_4)=(0,0)}
\end{aligned} \tag{4.3.3}$$

olup doğrusallaştırılmış sistem modeli aşağıdaki gibi elde edilir:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{mg}{M}x_3 - \frac{b}{M}x_2 + \frac{1}{M}u \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{(M+m)g}{Ml}x_3 + \frac{b}{Ml}x_2 - \frac{1}{Ml}u
\end{aligned} \tag{4.3.4}$$

Yukarıdaki ifadeler matris nütasyonunda aşağıdaki gibi yazılır [13].

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-b}{M} & \frac{-mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{b}{Ml} & \frac{(M+m)g}{Ml} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{Ml} \end{bmatrix} u \tag{4.3.5}$$

Yukarıdaki denklemde kullandığımız ters sarkacın tipik değerleri aşağıdaki tabluda verilmektedir.

Tablo 4.1: Swing-up Programındaki Parametrelerin Tanımları ve Değerleri

Değişken Adı	Açıklama	Başlangıç Değeri
u	Giriş kuvveti	0
e	Enerji	0
d	Enerjinin türevi	0
T	Zaman aralığı	0.02
t_f	Toplam zaman	8

4.4 Sistemin Doğrusal Modeli

Sistemin doğrusal modeli bilgilerini aşağıda MATLAB programına aktarıp doğrusal karesel düzenleyici (Linear Quadratic Regulator; LQR) katsayılarını elde etmekteyiz.

```
function xdot=TSdenklem (t,x)
M=2;m=0.25;L=0.6;b=1.5;g=9.81;global u
y1=(-b*x(2)+m*l*sin(x(3))*(x(4))^2 -m*g*sin(x(3))*
cos(x(3))+u)/(M+m-m*(cos(x(3)))^2);
y2=((b*x(2)-u-m*l*sin(x(3))*(x(4))^2)*cos(x(3))
+(M+m)*g*sin(x(3)))/(1*(M+m-m*(cos(x(3)))^2));
xdot=[x(2);y1;x(4);y2]
```

Şekil 4.1: Ters Sarkaç Dinamik Denklemleri MATLAB Kodları: TSdenklem.m

```

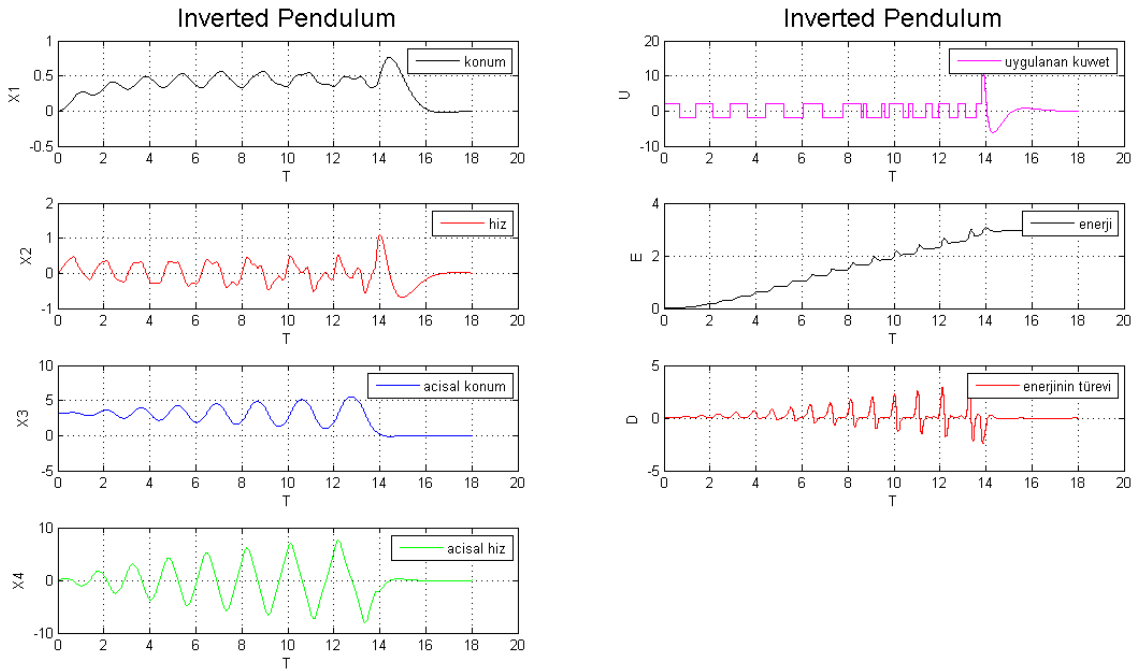
clear all,clc
global u
u=input ('u giriniz=')
x1=input ('x1 giriniz=')
x2=input ('x2 giriniz=')
x3=input ('x3 giriniz=')
x4=input ('x4 giriniz=')
t0=input ('baslangiç zamani giriniz=')
t1=input ('artis miktarı giriniz=')
t2=input ('bitis zamani giriniz=')
M=2;m=0.25;l=0.6;b=1.5;g=9.81;
A=[0 1 0 0;0 (-b)/M -m*g/M 0;0 0 0 1;0 (b)/(M*l)
(M*g+m*g)/(M*l) 0];
B=[0;1/M;0;-1/(M*l)];
Q=diag([1 1 1 1]);
R=0.1;
K=lqr(A,B,Q,R);
K=-K;
x0=[x1 x2 x3 x4];
for i=t0:t1:t2
u=K(1)*x0(1)+K(2)*x0(2)+K(3)*x0(3)+K(4)*x0(4);
[t,x]=ode23('TSdenklem',[i,i+t1],x0)

```

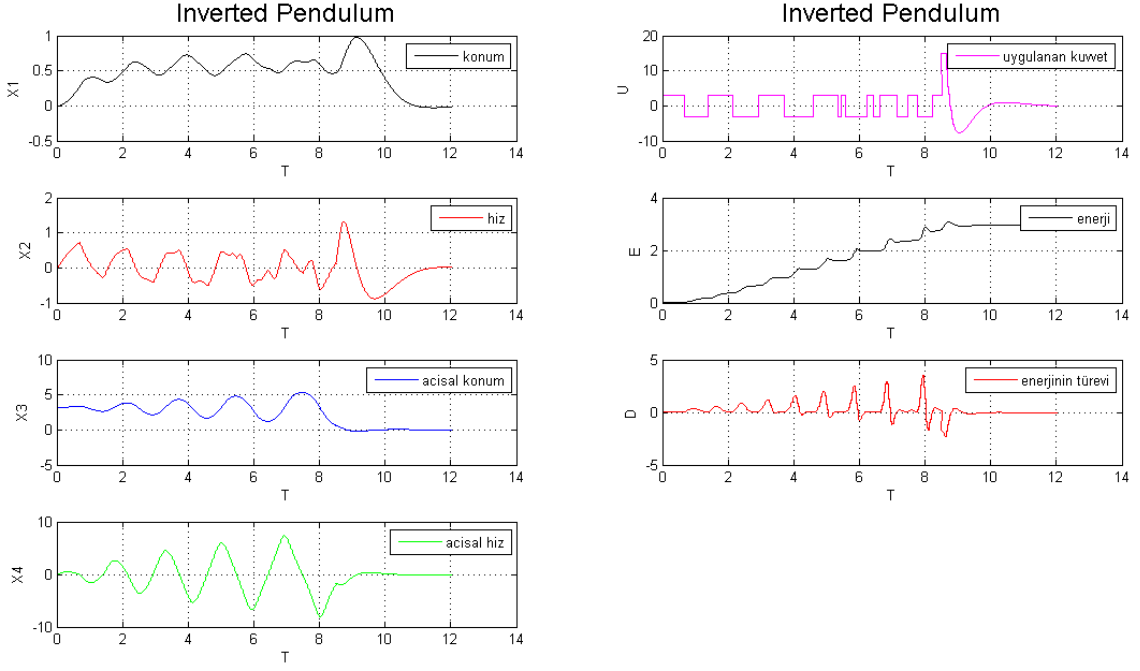
Şekil 4.2: Ters Sarkaç Dinamik Denklemleri MATLAB Kodları: TSana.m

4.5 Sonlu Sayıda Büyüklüklü Denetim Giriş Sinyalleri ile Simülasyon

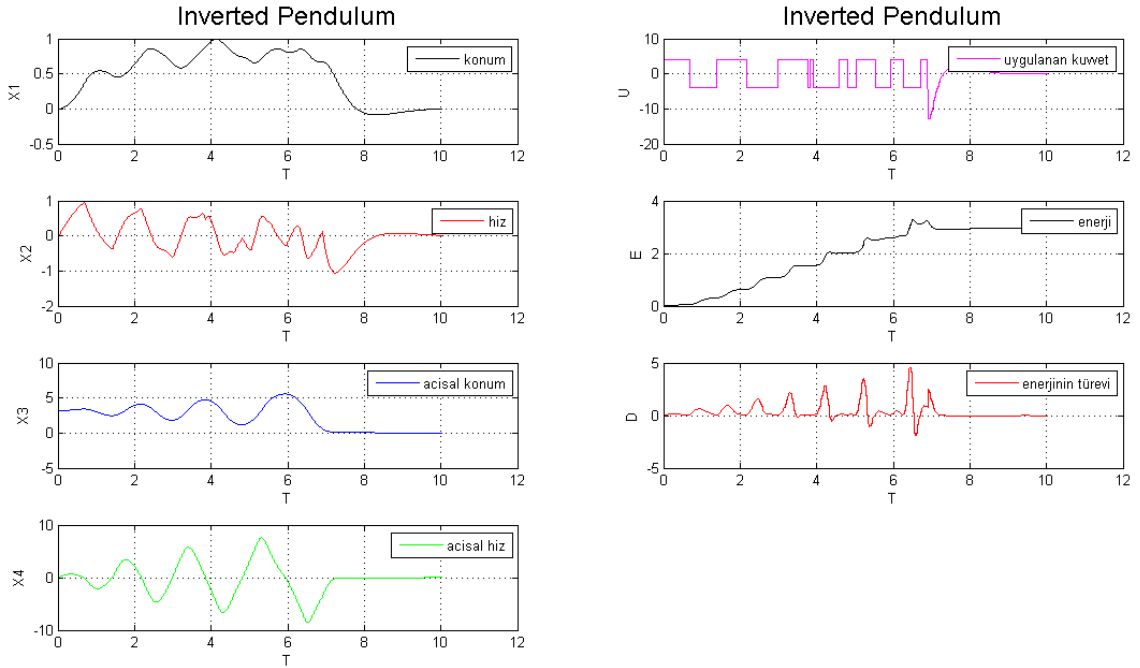
Enerji yaklaşımında yukarı kaldırma kısmında sadece iki farklı denetim girişi $u = +A$ ve $u = -A$ uygulama seçeneğimiz olan durum için programımız Ek E'dedir. A değeri olarak sırasıyla 2, 3, 4, 5, 7, 8, 9 uygulanmıştır. $\theta < 0.67$ radyan veya $\theta > 5.61$ radyan şartı sağlanınca yukarı kaldırma aşaması sonlandırılmış ve stabilizasyon kısmına geçilmiştir. Stabilizasyon kısmında ise LQR geribesleme katsayıları kullanılmıştır. Bu aşamada uygulanabilecek giriş kuvveti için üst ve alt sınırlar kullanılmıştır. Bu programın çalıştırılması sonucunda aşağıdaki grafikler elde edilmiştir. Grafiklerde istenen amaç $(x, \dot{x}, \theta, \dot{\theta}) = (0, 0, 0, 0)$ elde edilmektedir.



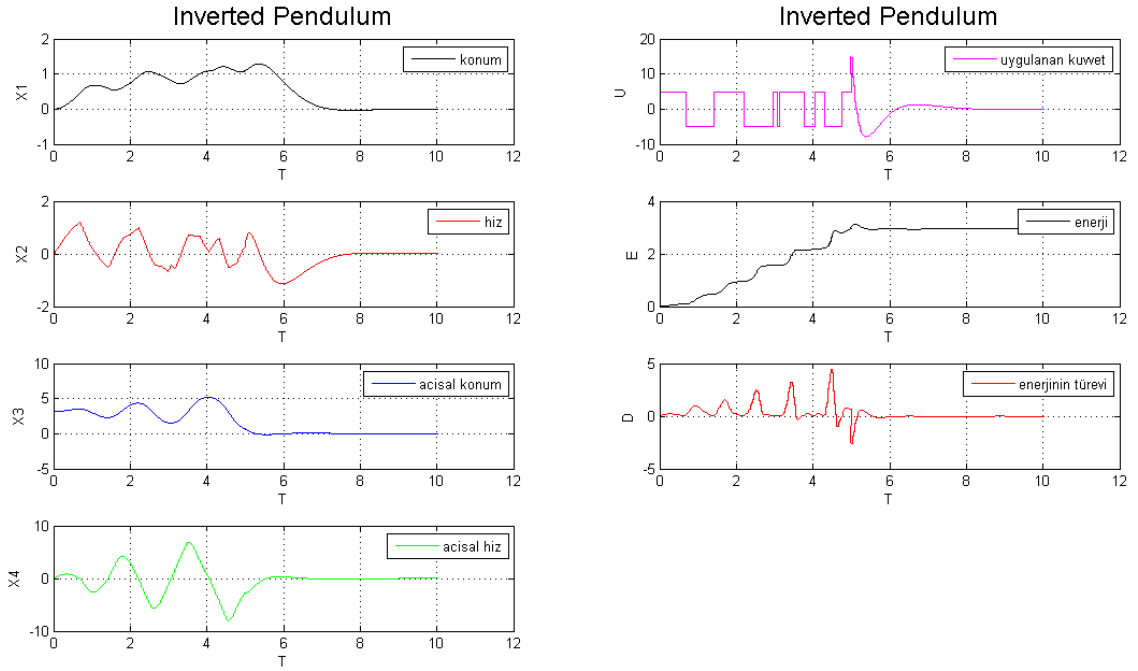
Şekil 4.3: Swing Programının Grafik Çıktıları ($A = \pm 2$)



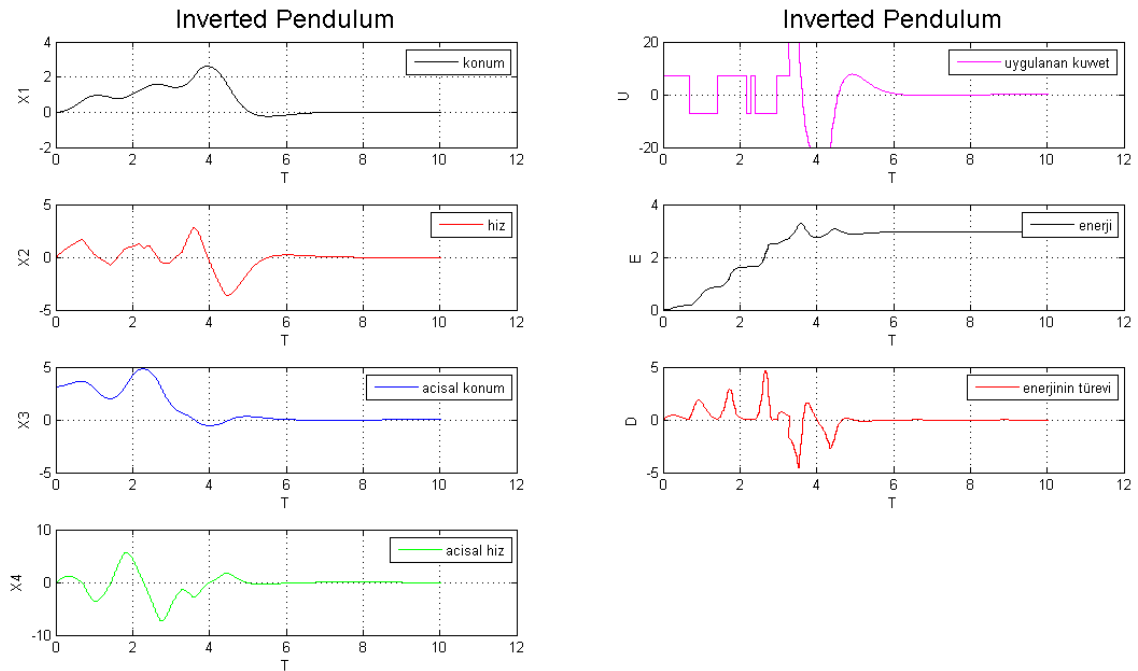
Şekil 4.4: Swing Programının Grafik Çıktıları ($A = \pm 3$)



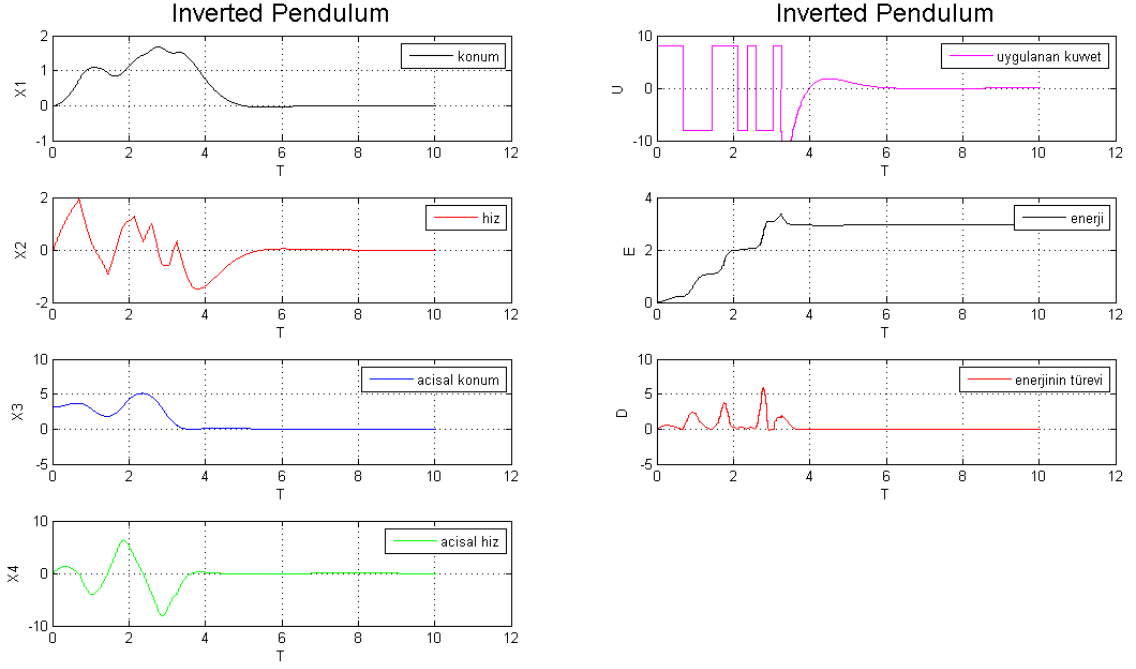
Şekil 4.5: Swing Programının Grafik Çıktıları ($A = \pm 4$)



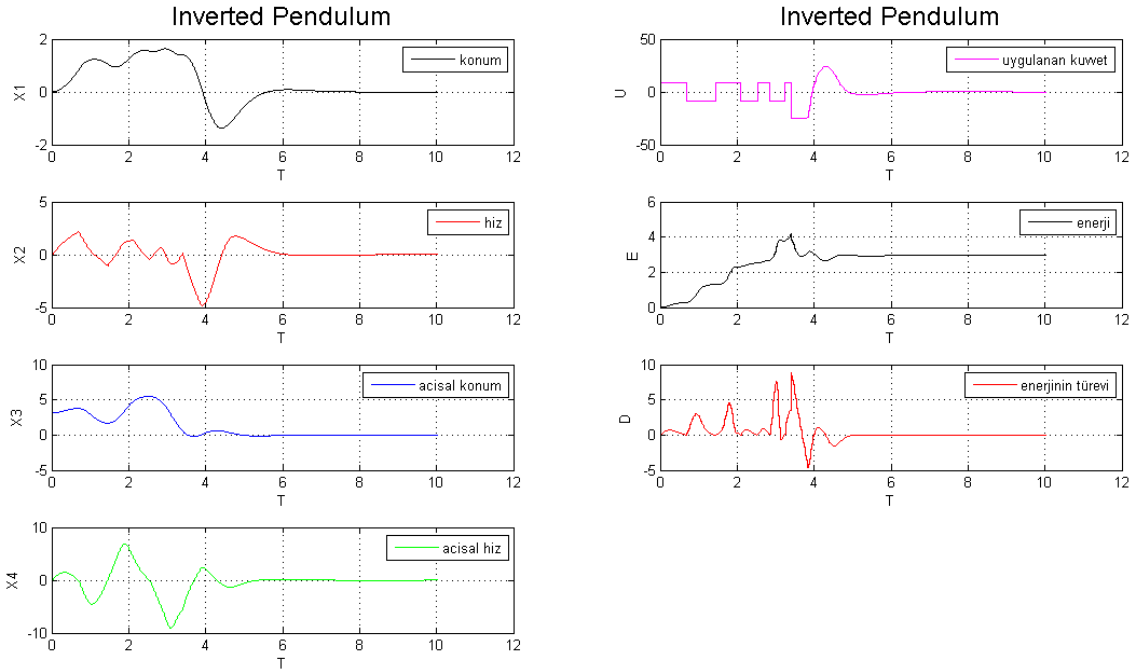
Şekil 4.6: Swing Programının Grafik Çıktıları ($A = \pm 5$)



Şekil 4.7: Swing Programının Grafik Çıktıları ($A = \pm 7$)

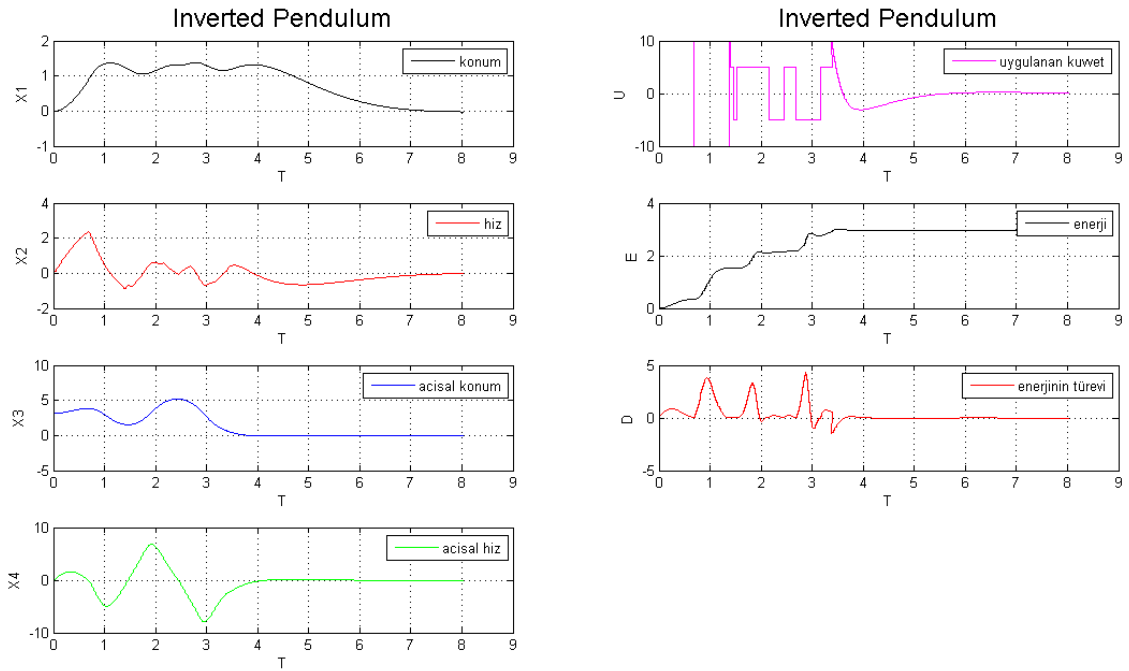


Şekil 4.8: Swing Programının Grafik Çıktıları ($A = \pm 8$)

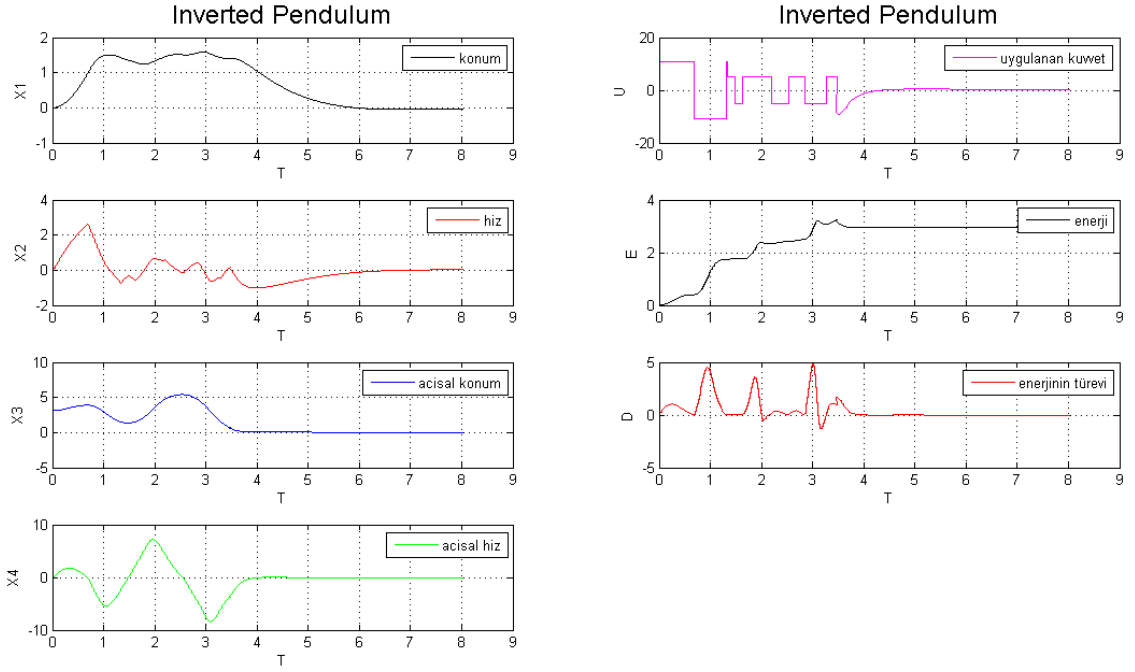


Şekil 4.9: Swing Programının Grafik Çıktıları ($A = \pm 9$)

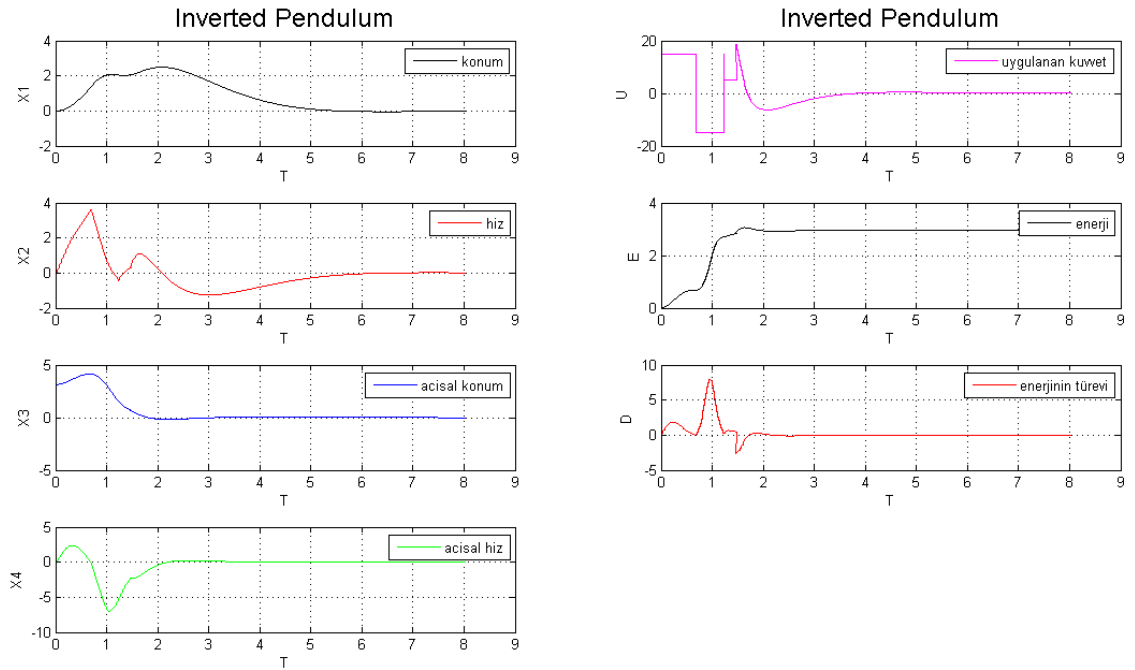
$A = 10$ ve daha büyük değerlerde yukardaki yaklaşım başarılı olmamaktadır, bunun sebebi stabilizasyon aşamasına sarkacın çok hızlı girmesi ve LQR geribeslemenin yetersiz kalmasıdır. Buna çözüm olarak iki kademeli yaklaşımı aşağıda denemekteyiz. Yeni yaklaşımında birinci kaldırma aşamasında $u = A$ ve $u = -A$ kullanılmakta olup $\theta < 1.28$ veya $\theta > 5.00$ olunca sonlanmakta ve ikinci kaldırma aşamasında $u = A/2$ ve $u = -A/2$ kullanılmaktadır. İkinci kaldırma aşaması $\theta > 0.67$ veya $\theta < 5.61$ olunca sonlanmakta ve stabilizasyon aşamasına geçilmektedir. İlgili grafikler aşağıdadır.



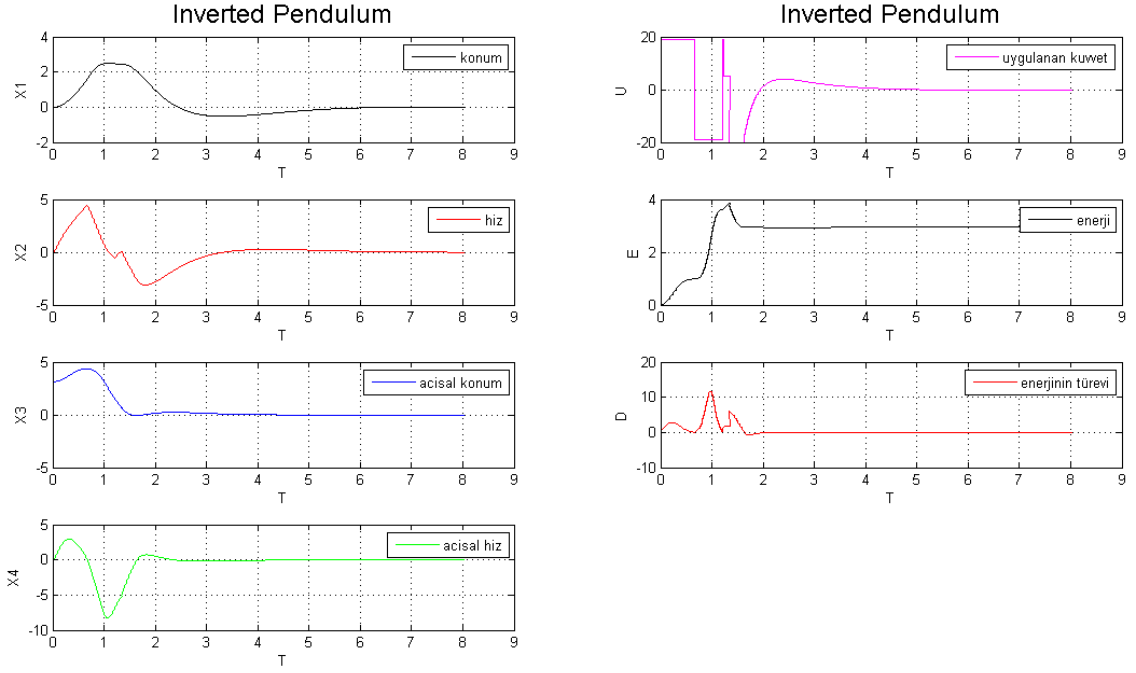
Şekil 4.10: İki Kademeli Yaklaşımında $A = \pm 10$ için Elde Edilen Grafikler



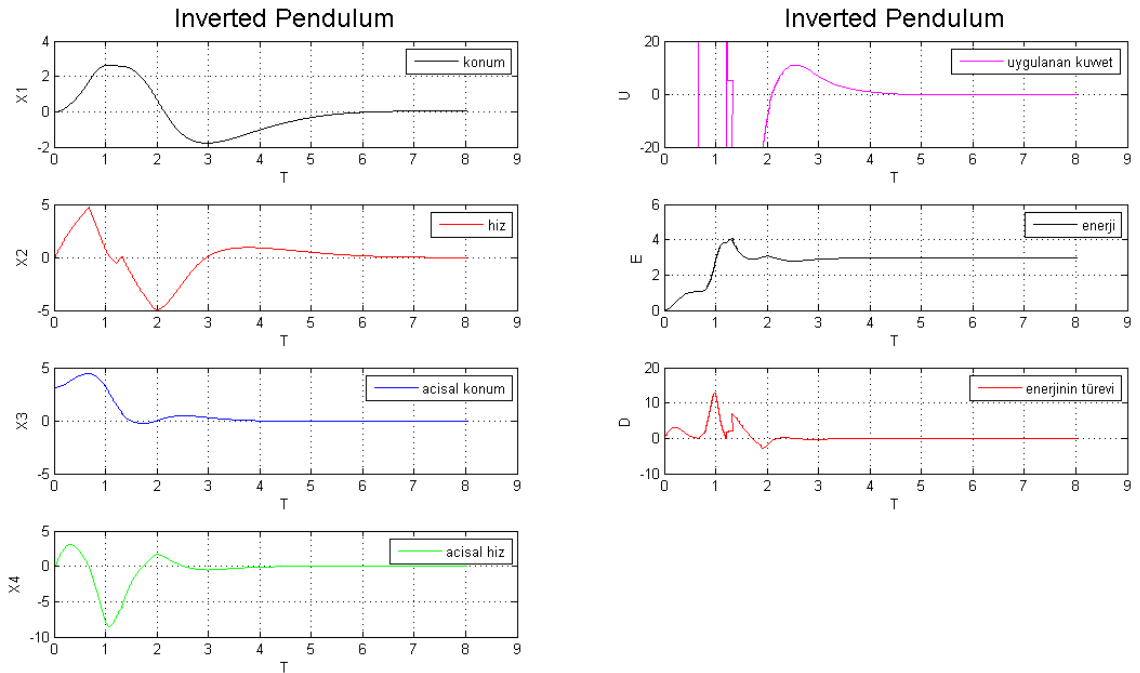
Şekil 4.11: İki Kademeli Yaklaşımda $A = \pm 11$ için Elde Edilen Grafikler



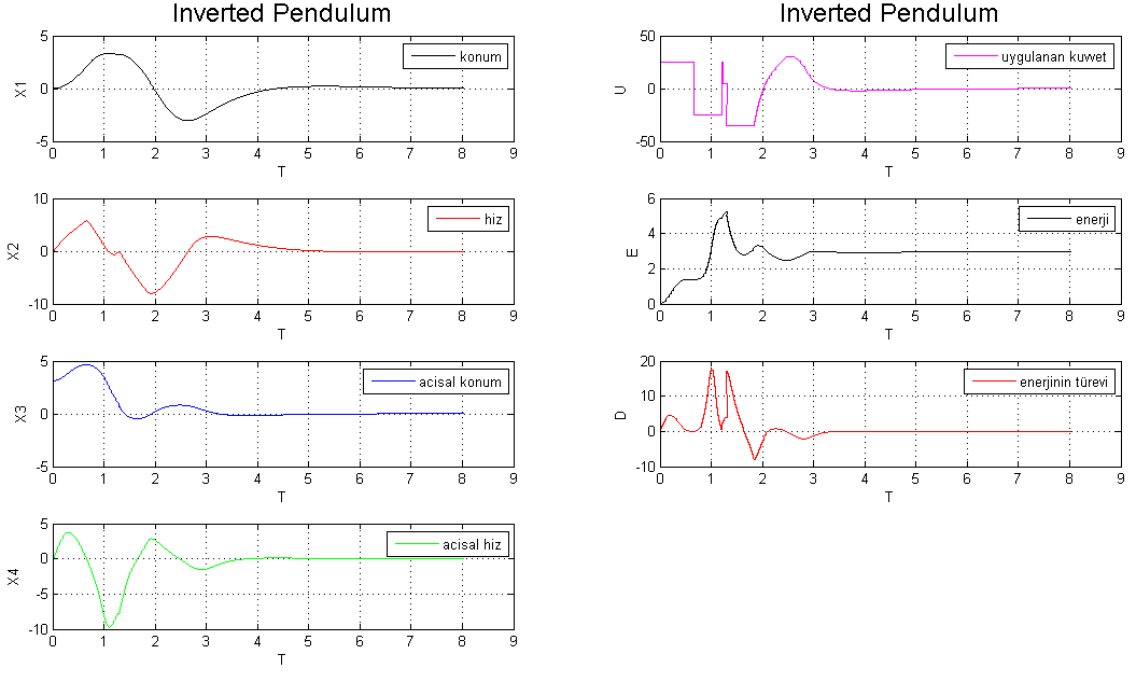
Şekil 4.12: İki Kademeli Yaklaşımda $A = \pm 15$ için Elde Edilen Grafikler



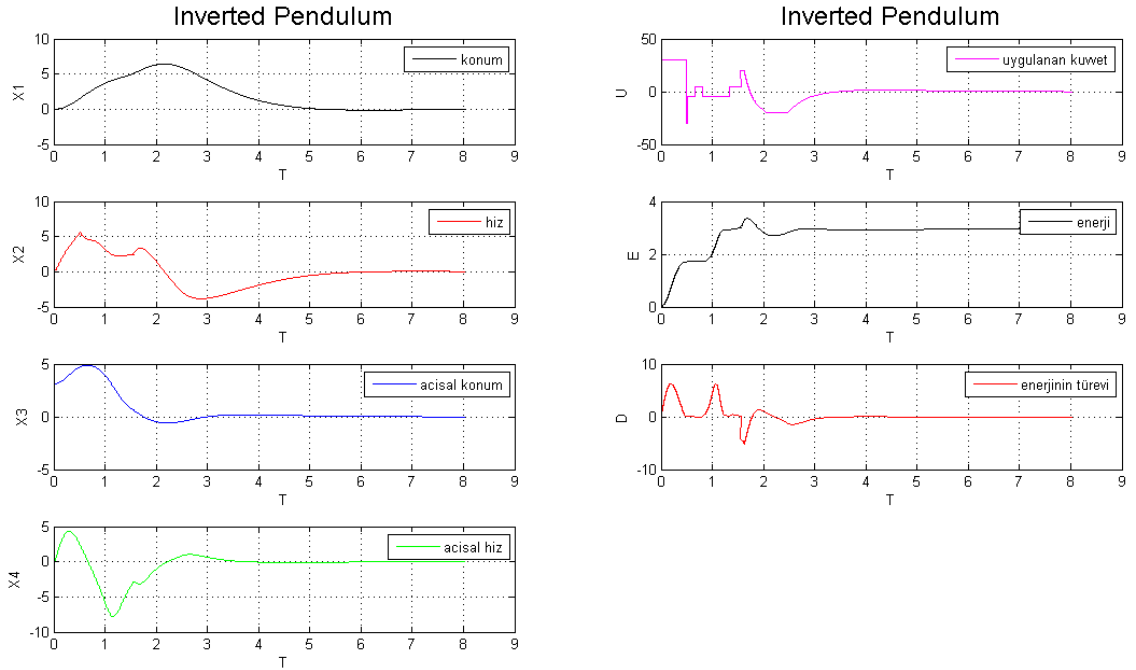
Şekil 4.13: İki Kademeli Yaklaşımda $A = \pm 19$ için Elde Edilen Grafikler



Şekil 4.14: İki Kademeli Yaklaşımda $A = \pm 20$ için Elde Edilen Grafikler



Şekil 4.15: İki Kademeli Yaklaşım $A = \pm 25$ için Elde Edilen Erafikler



Şekil 4.16: İki Kademeli Yaklaşım $A = \pm 30$ için Elde Edilen Grafikler

BÖLÜM 5

İRDELEME VE SONUÇLAR

5.1 İrdeleme ve Sonuçlar

Sonlu sayıda büyüklüklü denetim girişi ($u = +A$ veya $u = -A$) ile yukarı kaldırma ve sonra da LQR geribeslesi ile denge konumuna getirme işlemleri sonunda aşağıdaki tablo ile sonuçları özetlemek mümkündür.

Tablo 5.1: Tek Kademeli Yukarı Kaldırma için Deney Özetleri Tablosu

Deney No	A (Nt.)	Stabilizasyon aşamasına geçiş zamanı (sn.)	Durulma zamanı (sn.)	Başarı durumu
1	2	13.84	13.84	Başarılı
2	3	8.50	8.50	Başarılı
3	4	6.90	6.90	Başarılı
4	5	5.00	5.00	Başarılı
5	7	3.28	3.28	Başarılı
6	8	3.24	3.24	Başarılı
7	9	3.40	3.40	Başarılı

Tablo 5.1’de belirtilen deneylerde LQR katsayıları $K = [10.0000 \quad 13.2131 \quad 87.7737 \quad 21.1143]$ olarak elde edilmiştir. Stabilizasyon aşamasında giriş kuvveti için üst ve alt limitler sırasıyla $+10$ ve -10 olarak seçilmiştir.

Tablo 5.2: İki Kademeli Yukarı Kaldırma için Deney Özetleri Tablosu

Deney No	A (Nt.)	1. Kademeyi tamamlama zamanı (sn.)	Stabilizasyon aşamasına geçiş zamanı (sn.)	Durulma zamanı (sn.)	Başarı durumu
1	10	1.38	1.40	3.38	Başarılı
2	11	1.32	1.34	3.46	Başarılı
3	15	1.22	1.24	1.46	Başarılı
4	19	1.20	1.22	1.34	Başarılı
5	20	1.20	1.22	1.32	Başarılı
6	25	1.20	1.22	1.30	Başarılı
7	30	0.50	0.52	1.42	Başarılı

Tablo 5.2’de belirtilen deneylerde LQR katsayıları $K = [3.1623 \quad 7.1810 \quad 67.0914 \quad 16.0037]$ olarak elde edilmiştir. Stabilizasyon aşamasında giriş kuvveti için üst alt ve üst limitler sırasıyla $+10$ ve -10 olarak seçilmiştir.

Bu tabloda da görüldüğü üzere sınırlı sayıda denetim girişi kullanılması önemli bir kısıt olup, bu kısıt altında dahi kaldırma ve dengeleme aşamaları başarılı olarak sonuçlandırılmaktadır.

Kaynaklar

- [1] Artman, S. S., 1998, "The Simple Pendulum Is Not So Simple", Society for Industrial and Applied Mathematics, vol. 40, No. 4, pp. 927-930.
- [2] Matthews, M. R., Gauld, C., and Stinner, A., 2004, "The Pendulum: It's Place in Science, Culture and Pedagogy", Science & Education, 13: 261-277.
- [3] Küçükler, A., 2007, Asılı Sarkaç Sisteminde Konum Kontrolü, Sakarya Üniversitesi, Yüksek Lisans Tezi.
- [4] Boyce, W., E., Diprima, R., C., 1992, Elementary Differential Equations and Boundary Value Problems, John Wiley & Sons Ins.
- [5] Ogata, K., 2002, Modern Control Engineering, Fourth Edition, Pearson Education International.
- [6] Yazıcı, A., 2005, Doğrusal Olmayan Proglamlama Yönteminin Sistem Denetiminde Kullanımı, Osmangazi Üniversitesi, Doktora tezi. 118s.
- [7] Altın, R., 2005, Ters Sarkaç Sisteminin Denetimi, Hacettepe Üniversitesi, Yüksek Lisans Tezi,
- [8] Kiriz, S., Bingül, Z., Ousu, C., 2007, "Ters Sarkaç Probleminin PID ve Tam Durum Geribesleme Yöntemleri ile Kontrolü", TOK'07 bildiriler S. 49-54.
- [9] Kahvecioğlu, S., Karamancıoğlu, A., ve Yazıcı, A., 2008, "Enerji Kontrolüne Dayalı Doğrusal Olmayan Model Öngörümlü Denetim ile Ters Sarkacın Yukarı Kaldırması", 466-470 ELECO.

- [10] Yazıcı, A., Karamancıođlu, A., 2008, "A nonlinear programming approach for the swing-up control problem", Eng & Arch. Fac. Eskişehir Osmangazi University, Vol XXI, No: 2.
- [11] Stimac, A. K., 1999, "Standup and Stabilization of the Inverted Pendulum", Massachusetts Institute of Technology.
- [12] Yazıcı, A., Karamancıođlu, A., 2009, "Ters Sarkaç Sisteminin Kontrol Eđitiminde Test Aracı Olarak Kullanılması". EEBB Mühendislikleri Eđitimi 4. Ulusal Sempozyumu.
- [13] Yazıcı, A., 2000, Kayma Kipli Yaklaşımlarının Ters Sarkaç Sistemine Uygulanması, Omangazi Üniversitesi, yüksek lisans tezi.
- [14] Kandemir, K., 2006, MATLAB ve Simulink Kullanarak LQR ve Kutup Yerleşimi Metotları ile Tepe Vinci Kontrolü, Yıldız Teknik Üniversitesi, yüksek lisans tezi.
- [15] Ablay, G., Uçar, A., 2009, " Belirsizlik İçeren ve Doğrusal Olamayan Robot Kollarının Gürbüz Denetimi", Anadolu Üniversitesi Bilim ve Teknoloji Dergisi, sayı No:2:367-382
- [16] Adams, R., A., 2009, Calculus A Complete Course, 6th, Pearson Academic.

BÖLÜM 6

EKLER

EK A

Ters Sarkaçta Kullandığımız Denklemler TSdenklem.m

```
function xdot=TSdenklem (t,x)
global u
M=2;m=0.25;l=0.6;b=1.5;g=9.81;
y1=(-b*x(2)+m*l*sin(x(3))*(x(4))^2-m*g*...
sin(x(3))*cos(x(3))+u)/(M+m-m*(cos(x(3)))^2);
y2=((b*x(2)-u-m*l*sin(x(3))*(x(4))^2)*cos(x(3))+(M+m)*
g*sin(x(3)))/(l*(M+m-m*(cos(x(3)))^2));
xdot=[x(2);y1;x(4);y2]
```

EK B

Ters Sarkaç ana Program Denklemleri Test etmek için TSana.m

```
clear all,clc
global u
u=input ('u giriniz=')
x1=input ('x1 giriniz=');
x2=input ('x2 giriniz=');
x3=input ('x3 giriniz=');
x4=input ('x4 giriniz=');
x0=[x1 x2 x3 x4];
t0=input ('baslangiç zamani giriniz=');
t1=input ('bitis zamani giriniz=');
time=[t0 t1];
[t,x]=ode23('TSdenklem',time,x0);
subplot(2,2,1)
plot(t,x(:,1),'k')
grid
title('Inverted Pendulum','fontsize',10)
xlabel('T')
ylabel('X1')
legend('konum')
subplot(2,2,2)
plot(t,x(:,2),'r')
grid
```

```
title('Inverted Pendulum','fontsize',10)
xlabel('T')
ylabel('X2')
legend('hiz')
subplot(2,2,3)
plot(t,x(:,3),'b')
grid
xlabel('T')
ylabel('X3')
legend('sarkaç açisi')
subplot(2,2,4)
plot(t,x(:,4),'g')
grid
xlabel('T')
ylabel('X4')
legend('sarkaç açisal hizi')
```

EK C

Ters Sarkaç ana Program Denklemleri Test etmek için myequation.m

```
function xdot=myequation (t,x)
M=2;m=0.25;L=0.6;b=1.5;g=9.81;global u
xdot=[x(2);(-b*x(2)+m*L*sin(x(3))*(x(4))^2-m*g*sin(x(3))*
cos(x(3))+u)/(M+m-m*(cos(x(3)))^2);x(4);
((b*x(2)-u-m*L*sin(x(3))*(x(4))^2)*cos(x(3))+(M+m)*
g*sin(x(3)))/(L*(M+m-m*(cos(x(3)))^2))]
```

EK D

Ters Sarkaç *LQR* Yönteminin Sıfır Açığa Çekmek Testi mymain.m

```
clear all,clc
global u
u=input ('u giriniz=')
x1=input ('x1 giriniz=')
x2=input ('x2 giriniz=')
x3=input ('x3 giriniz=')
x4=input ('x4 giriniz=')
t0=input ('baslangiç zamani giriniz=')
t1=input ('artis miktarı giriniz=')
t2=input ('bitis zamani giriniz=')
M=2;m=0.25;l=0.6;b=1.5;g=9.81;
A=[0 1 0 0;0 (-b)/M -m*g/M 0;0 0 0 1;0 (b)/(M*l) ...
    (M*g+m*g)/(M*l) 0];
B=[0;1/M;0;-1/(M*l)];
Q=diag([1 1 1 1]);
R=0.1;
K=lqr(A,B,Q,R);
K=-K;
x0=[x1 x2 x3 x4];
for i=t0:t1:t2
    u=K(1)*x0(1)+K(2)*x0(2)+K(3)*x0(3)+K(4)*x0(4)
    [t,x]=ode23('TSdenklem',[i,i+t1],x0)
```

```
subplot(3,2,1)
plot(t,x(:,1),'k')
grid on
hold on
title('Inverted Pendulum','fontsize',10)
xlabel('T')
ylabel('X1')
legend('konum')
subplot(3,2,2)
plot(t,x(:,2),'r')
grid on
hold on
title('Inverted Pendulum','fontsize',10)
xlabel('T')
ylabel('X2')
legend('hiz')
subplot(3,2,3)
plot(t,x(:,3),'b')
grid on
hold on
xlabel('T')
ylabel('X3')
legend('sarkaç açisi')
subplot(3,2,4)
plot(t,x(:,4),'g')
grid on
hold on
xlabel('T')
ylabel('X4')
legend('sarkaç açisal hizi')
```



```
subplot(3,2,5)
plot(t,u,'m')
grid on
hold on
xlabel('T')
ylabel('u')
legend('uygulanan kuvvet')
[rowx,colx]=size(x);
x0(1,:)=x(rowx,:);
end
```

EK E

Ters Sarkaçta Kullandığımız Denklem invpend.m

```
%Bölüm-2: Invpend fonksiyonu
function xdot=invpend(t,x);
global u;global e;global d;

M=2;
m=0.25;
L=0.6;
b=1.5;
g=9.81;
xdot=[x(2);
      (-b*x(2)+m*L*sin(x(3))*(x(4))^2-m*g*sin(x(3))*...
      *cos(x(3))+u)/(M+m-m*cos(x(3))*cos(x(3)));
      x(4);
      ((b*x(2)-u-m*L*sin(x(3))*(x(4))^2)*cos(x(3))+(M+m)*
      g*sin(x(3)))/(L*(M+m-m*cos(x(3))*cos(x(3))))];
Program invpend.m
```

EK F

Ters Sarkacı Tek Kademeli Yukarı Kaldırma Programı Swing04.m

```
%% yeni uygulama
clear all; clc
global u; global e; global d;
input = 2;
u=0; e=0; d=0; M=2; m=0.25; L=0.6; b=1.5; g=9.81;...
T=0.02; tf = 18; kirpma=15;
i5 = 0;
% programı hızlandırmak için gereken değişken tanıtımı
x0=[0 0 pi 0];
xx=[0 0 0 0];
tt=[0];
u_matrix=[0];
e_matrix=[0];
d_matrix=[0];
% programın hızlandırması için gereken tanıtımlar tamamlandı

for i=0:T:tf

    [t,x]=ode23('invpend',[i,i+T],x0);
    [r c]=size(x);
    if (x(r,3)>0 && x(r,3)<=(pi/4) && x(r,4)>0)
        u=-input;
```

```

elseif (x(r,3)>0 && x(r,3)<=(pi/4) && x(r,4)<0)
    u=input;

elseif (x(r,3)> (pi/4) && x(r,3)<=(2*pi/4) &&...
    x(r,4)>0) u=-input;
elseif (x(r,3)> (pi/4) && x(r,3)<=(2*pi/4) &&...
    x(r,4)<0) u=input;

elseif (x(r,3)>(2*pi/4) && x(r,3)<=(3*pi/4) &&...
    x(r,4)>0) u=input;
elseif (x(r,3)>(2*pi/4) && x(r,3)<=(3*pi/4) &&...
    x(r,4)<0) u=-input;

elseif (x(r,3)>(3*pi/4) && x(r,3)<=(4*pi/4) &&...
    x(r,4)>0) u=input;
elseif (x(r,3)>(3*pi/4) && x(r,3)<=(4*pi/4) &&...
    x(r,4)<0) u=-input;

elseif (x(r,3)>(4*pi/4) && x(r,3)<=(5*pi/4) &&...
    x(r,4)>0) u=input;
elseif (x(r,3)>(4*pi/4) && x(r,3)<=(5*pi/4) &&...
    x(r,4)<0) u=-input;

elseif (x(r,3)>(5*pi/4) && x(r,3)<=(6*pi/4) &&...
    x(r,4)>0) u=input;
elseif (x(r,3)>(5*pi/4) && x(r,3)<=(6*pi/4) &&...
    x(r,4)<0) u=-input;

elseif (x(r,3)>(6*pi/4) && x(r,3)<=(7*pi/4) &&...
    x(r,4)>0) u=-input;

```

```

elseif (x(r,3)>(6*pi/4) && x(r,3)<=(7*pi/4) &&...
    x(r,4)<0) u=input;

elseif (x(r,3)>(7*pi/4) && x(r,3)<=(8*pi/4) &&...
    x(r,4)>0) u=-input;
elseif (x(r,3)>(7*pi/4) && x(r,3)< (8*pi/4) &&...
    x(r,4)<0) u=input;
end

e=(0.5)*m*(L*x(r,4))^2+m*g*L*(cos(x(r,3))+1);
d=(-m*L*x(r,4)*cos(x(r,3))*(-b*x(r,2)+m*L*sin...
(x(r,3))*x(r,4))^2-m*g*sin(x(r,3))*cos(x(r,3))+u)...
/(M+m-m*(cos(x(r,3)))^2));

[rowx, colx]=size(x);
x0(1,:)= x(rowx,:);

c=xx;
xx=[c;x];
tc=tt;
tt=[tc;t];
sss=size(t);

for j=1:sss(1)
    c=u_matrix;
    u_matrix=[c;u];
end
for j=1:sss(1)
    c=e_matrix;
    e_matrix=[c;e];

```

```

end
for j=1:sss(1)
    c=d_matrix;
    d_matrix=[c;d];
end
yeni = abs(x(r,3));

% Sarkac 0 veya 2 pi noktasına yakınlaştığında...
swing-up kısmından çık.
if ((5.61<yeni && yeni<6.28) || (0<yeni && yeni<0.67))
    i5=i;
end
if (i5 > 0)
    break;
end
end
end
%%% ----- Store u, e, d, x to plot -----
ttindex=size(tt);
ttindex_x=ttindex(1);
tt3=tt(2:ttindex_x);

xxindex=size(xx);
xxindex_x=xxindex(1);
xx3=xx(2:xxindex_x,1:4);

uindex=size(u_matrix);
uindex_x=uindex(1);
uu3=u_matrix(2:uindex_x);

eindex=size(e_matrix);

```

```

eindex_x=eindex(1);
ee3=e_matrix(2:eindex_x);

dindex=size(d_matrix);
dindex_x=dindex(1);
dd3=d_matrix(2:dindex_x);

%%%*****

%%% Definitions for LQR equations
T=0.02;
% tf=12;
A=[0 1 0 0 ;0 (-b)/M -m*g/M 0;0 0 0 1;0 b/(M*L)...
   (M*g+m*g)/(M*L) 0];
B=[0;1/M;0;-1/(M*L)];
Q=diag([100 1 1 1]);
R=1;
K=lqr(A,B,Q,R);
K=-K;

%%%*****

for i=i5:T:tf
    u=K(1)*x(r,1)+K(2)*x(r,2)+K(3)*x(r,3)+...
    K(4)*x(r,4);
    if u>kirpma
        u=kirpma;
    elseif u<-kirpma
        u=-kirpma;
    end
    [t,x]=ode23('invpnd',[i,i+T],x0);
    [r c]=size(x);

```

```

e=(0.5)*m*(L*x(r,4))^2+m*g*L*(cos(x(r,3))+1);
d=(-m*L*x(r,4)*cos(x(r,3))*(-b*x(r,2)+m*L*...
sin(x(r,3))*(x(r,4))^2...
-m*g*sin(x(r,3))*cos(x(r,3))-u)/...
(M+m-m*(cos(x(r,3)))^2));
[rowx,colx]=size(x);
x0(1,:)=x(rowx,:);

c=xx;
xx=[c;x];
tc=tt;
tt=[tc;t];
sss=size(t);
for j=1:sss(1)
    c=u_matrix;
    u_matrix=[c;u];
end
for j=1:sss(1)
    c=e_matrix;
    e_matrix=[c;e];
end
for j=1:sss(1)
    c=d_matrix;
    d_matrix=[c;d];
end

end

%%% Store u, e, d, x to plot -----
ttindex=size(tt);
ttindex_x=ttindex(1);

```



```

tt3=tt(2:ttindex_x);

xxindex=size(xx);
xxindex_x=xxindex(1);
xx3=xx(2:xxindex_x,1:4);

uindex=size(u_matrix);
uindex_x=uindex(1);
uu3=u_matrix(2:uindex_x);

eindex=size(e_matrix);
eindex_x=eindex(1);
ee3=e_matrix(2:eindex_x);

dindex=size(d_matrix);
dindex_x=dindex(1);
dd3=d_matrix(2:dindex_x);

%%*****

%%----- PLOTS -----
subplot(4,2,1)
plot(tt3,xx3(:,1),'k')
grid on
hold on
title('Inverted Pendulum','fontsize',18)
xlabel('T')
ylabel('X1')
legend('konum')

subplot(4,2,3)

```

```
plot(tt3,xx3(:,2),'r')
grid on
hold on
xlabel('T')
ylabel('X2')
legend('hiz')

subplot(4,2,5)
plot(tt3,xx3(:,3),'b')
grid on
hold on
xlabel('T')
ylabel('X3')
legend('acisal konum')

subplot(4,2,7)
plot(tt3,xx3(:,4),'g')
grid on
hold on
xlabel('T')
ylabel('X4')
legend('acisal hiz')

subplot(4,2,2)
plot(tt3,uu3,'m')
title('Inverted Pendulum','fontsize',18)
grid on
hold on
xlabel('T')
ylabel('U')
```

```
legend('uygulanan kuvvet')
```

```
subplot (4,2,4)
```

```
plot(tt3,ee3,'k')
```

```
grid on
```

```
hold on
```

```
xlabel('T')
```

```
ylabel('E')
```

```
legend('enerji')
```

```
subplot(4,2,6)
```

```
plot(tt3,dd3,'r')
```

```
grid on
```

```
hold on
```

```
xlabel('T')
```

```
ylabel ('D')
```

```
legend('enerjinin türevi')
```

```
%%%*****
```

EK G

Ters Sarkacı İki Kademeli Yukarı Kaldırma Programı Swing05.m

```
clear all; clc
global u; global e; global d;
input = 30;
flag = 1;
angle1=4.71;
angle2=1.57;
angle3=5.58;
angle4=0.7;
u=0; e=0; d=0; M=2; m=0.25; L=0.6; b=1.5;
g=9.81; T=0.02; tf=8;
kirpma=50;
LqrStart = 0;
% programı hızlandırmak için gereken değişken tanıtımı
x0=[0 0 pi 0];
xx=[0 0 0 0];
tt=[0];
u_matrix=[0];
e_matrix=[0];
d_matrix=[0];
% programın hızlandırması için gereken tanıtımlar
tamamlandı
```

```

for i=0:T:tf

    [t,x]=ode23('invpend',[i,i+T],x0);
    [r c]=size(x);
    if      (x(r,3)>0 && x(r,3)<=(pi/4) &&
        x(r,4)>0) u=-input;
    elseif (x(r,3)>0 && x(r,3)<=(pi/4) &&
        x(r,4)<0) u=input;

    elseif (x(r,3)> (pi/4) && x(r,3)<=(2*pi/4) &&
        x(r,4)>0) u=-input;
    elseif (x(r,3)> (pi/4) && x(r,3)<=(2*pi/4) &&
        x(r,4)<0) u=input;

    elseif (x(r,3)>(2*pi/4) && x(r,3)<=(3*pi/4) &&
        x(r,4)>0) u=input;
    elseif (x(r,3)>(2*pi/4) && x(r,3)<=(3*pi/4) &&
        x(r,4)<0) u=-input;

    elseif (x(r,3)>(3*pi/4) && x(r,3)<=(4*pi/4) &&
        x(r,4)>0) u=input;
    elseif (x(r,3)>(3*pi/4) && x(r,3)<=(4*pi/4) &&
        x(r,4)<0) u=-input;

    elseif (x(r,3)>(4*pi/4) && x(r,3)<=(5*pi/4) &&
        x(r,4)>0) u=input;
    elseif (x(r,3)>(4*pi/4) && x(r,3)<=(5*pi/4) &&
        x(r,4)<0) u=-input;

```

```

elseif (x(r,3)>(5*pi/4) && x(r,3)<=(6*pi/4) &&
x(r,4)>0) u=input;
elseif (x(r,3)>(5*pi/4) && x(r,3)<=(6*pi/4) &&
x(r,4)<0) u=-input;

elseif (x(r,3)>(6*pi/4) && x(r,3)<=(7*pi/4) &&
x(r,4)>0) u=-input;
elseif (x(r,3)>(6*pi/4) && x(r,3)<=(7*pi/4) &&
x(r,4)<0) u=input;

elseif (x(r,3)>(7*pi/4) && x(r,3)<=(8*pi/4) &&
x(r,4)>0) u=-input;
elseif (x(r,3)>(7*pi/4) && x(r,3)<(8*pi/4) &&
x(r,4)<0) u=input;
end

e=(0.5)*m*(L*x(r,4))^2+m*g*L*(cos(x(r,3))+1);
d=(-m*L*x(r,4)*cos(x(r,3))*(-b*x(r,2)+m*L*...
sin(x(r,3))*(x(r,4))^2;
-m*g*sin(x(r,3))*cos(x(r,3))+u)/...
(M+m-m*(cos(x(r,3)))^2));

[rowx, colx]=size(x);
x0(1,:)= x(rowx,:);

c=xx;
xx=[c;x];
tc=tt;
tt=[tc;t];
sss=size(t);

```

```
for j=1:sss(1)
    c=u_matrix;
    u_matrix=[c;u];
end
for j=1:sss(1)
    c=e_matrix;
    e_matrix=[c;e];
end
for j=1:sss(1)
    c=d_matrix;
    d_matrix=[c;d];
end
yeni = abs(x(r,3));

if ((angle1 < yeni) || (yeni < angle2)) && flag ==1)
    flag = 0;
    input = 15;
    i
end

%Sarkac 0 veya 2 pi noktasına yakınlaştığında...
swing-up kısmından
if ((angle3 < yeni && yeni < 6.25) ||...
(0 < yeni && yeni < angle4)
    LqrStart = i;
end
if (LqrStart > 0)
    LqrStart
    break;
```

```

        end
    end
    end
    %% ----- Store u, e, d, x to plot -----
    ttindex=size(tt);
    ttindex_x=ttindex(1);
    tt3=tt(2:ttindex_x);

    xxindex=size(xx);
    xxindex_x=xxindex(1);
    xx3=xx(2:xxindex_x,1:4);

    uindex=size(u_matrix);
    uindex_x=uindex(1);
    uu3=u_matrix(2:uindex_x);

    eindex=size(e_matrix);
    eindex_x=eindex(1);
    ee3=e_matrix(2:eindex_x);

    dindex=size(d_matrix);
    dindex_x=dindex(1);
    dd3=d_matrix(2:dindex_x);

    %%*****
    %% Definitions for LQR equations
    T=0.02;
    % tf=10;
    A=[0 1 0 0 ;0 (-b)/M -m*g/M 0;0 0 0 1;0 b/(M*L)...
      (M*g+m*g)/(M*L) 0];
    B=[0;1/M;0;-1/(M*L)];
    Q=diag([100 1 1 1]);%([10 1 1 1]);

```



```

R=1;
K=lqr(A,B,Q,R);
K=-K;
%%%*****

for i=LqrStart:T:tf
    u=K(1)*x(r,1)+K(2)*x(r,2)+K(3)*x(r,3)+K(4)*x(r,4);
    if      u>kirpma
        u=kirpma;
    elseif u<-kirpma
        u=-kirpma;
    end
    [t,x]=ode23('invpend',[i,i+T],x0);
    [r c]=size(x);
    e=(0.5)*m*(L*x(r,4))^2+m*g*L*(cos(x(r,3))+1);
    d=(-m*L*x(r,4)*cos(x(r,3))*(-b*x(r,2)+m*L*...
    sin(x(r,3))*(x(r,4))^2;
    -m*g*sin(x(r,3))*cos(x(r,3))-u)/...
    (M+m-m*(cos(x(r,3)))^2));
    [rowx,colx]=size(x);
    x0(1,:)=x(rowx,:);

    c=xx;
    xx=[c;x];
    tc=tt;
    tt=[tc;t];
    sss=size(t);
    for j=1:sss(1)
        c=u_matrix;
        u_matrix=[c;u];
    end
end

```

```

end
for j=1:sss(1)
    c=e_matrix;
    e_matrix=[c;e];
end
for j=1:sss(1)
    c=d_matrix;
    d_matrix=[c;d];
end

end

%%% Store u, e, d, x to plot -----
ttindex=size(tt);
ttindex_x=ttindex(1);
tt3=tt(2:ttindex_x);

xxindex=size(xx);
xxindex_x=xxindex(1);
xx3=xx(2:xxindex_x,1:4);

uindex=size(u_matrix);
uindex_x=uindex(1);
uu3=u_matrix(2:uindex_x);

eindex=size(e_matrix);
eindex_x=eindex(1);
ee3=e_matrix(2:eindex_x);

dindex=size(d_matrix);

```

```

dindex_x=dindex(1);
dd3=d_matrix(2:dindex_x);
%%*****

%%----- PLOTS -----
subplot(4,2,1)
plot(tt3,xx3(:,1),'k')
grid on
hold on
title('Inverted Pendulum','fontsize',18)
xlabel('T')
ylabel('X1')
legend('konum')
subplot(4,2,3)
plot(tt3,xx3(:,2),'r')
grid on
hold on
xlabel('T')
ylabel('X2')
legend('hiz')
subplot(4,2,5)
plot(tt3,xx3(:,3),'b')
grid on
hold on
xlabel('T')
ylabel('X3')
legend('acisal konum')
subplot(4,2,7)
plot(tt3,xx3(:,4),'g')
grid on

```

```

hold on
xlabel('T')
ylabel('X4')
legend('acisal hiz')
subplot(4,2,2)
plot(tt3,uu3,'m')
title('Inverted Pendulum','fontsize',18)
grid on
hold on
xlabel('T')
ylabel('U')
legend('uygulanan kuvvet')
subplot(4,2,4)
plot(tt3,ee3,'k')
grid on
hold on
xlabel('T')
ylabel('E')
legend('enerji')
subplot(4,2,6)
plot(tt3,dd3,'r')
grid on
hold on
xlabel('T')
ylabel('D')
legend('enerjinin türevi')
%%%*****

```