

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MATLAB VE SIMULINK KULLANARAK LQR VE
KUTUP YERLEŞİMİ METOTLARI İLE TEPE VİNCİ
KONTROLÜ**

Elektrik Müh. Kaan KANDEMİR

F.B.E. Elektrik Mühendisliği Anabilim Dalına Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yard.Doç.Dr. Şeref Naci ENGİN

İSTANBUL, 2006

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖNSÖZ.....	viii
ÖZET	ix
ABSTRACT	x
1 GİRİŞ.....	1
Motivasyon.....	1
1.2 Literatür Taraması	1
1.3 Amaç	2
1.4 Vinç Çeşitleri	3
2 TEPE VİNCİNİ OLUŞTURAN SARKAÇ VE ARABA SİSTEMİNİN MODELLENMESİ	5
2.1 Lagrangian Dinamikleri Analizi	5
2.2 Çalışma Noktası Yakınında Linearleştirme.....	9
2.3 Sistemin Transfer Fonksiyonu.....	10
2.4 Durum-Uzay (State-Space) Gösterimi.....	11
3 KONTROLEDİLEBİLİRLİK VE GÖZLENEBİLİRLİK KAVRAMLARI	13
3.1 Kontrol edilebilirlik (Controllability).....	13
3.2 Gözlenebilirlik (Observability).....	14
4 TİP 1 SERVO SİSTEMİN TANIMLANMASI	17
4.1 İntegratör İçeren Tip 1 Servo Sistemin Sürekli Zamanda Genel Gösterimi.....	17
4.2 İntegratör İçermeyen Tip 1 Servo Sistemin Genel Gösterimi	18
4.3 Ayrık Zamanlı İntegratör İçermeyen Type 1 Servo Sistemin Genel Gösterimi	20
5 DURUM GERİ BESLEME KONTROLÜ	24
5.1 Kutup Yerleşim Metodu (Pole placement).....	24
5.1.1 Yöntem1- Kontrol edilebilir Kanonik Forma Dönüşüme Dayalı Metot	26
5.1.2 Yöntem 2- Ackermann Formülü ile K Değeri Hesaplanması.....	27
5.2 Dinamik Ayrık Zamanlı Bir Sistemin Durum Yeniden Yapılandırılması-Gözlemci Dizaynı (Observer –Estimator Design)	29
5.2.1 Deterministik Prosesin Kapalı Çevrim Tahmini	30

5.2.2	Stabilite Kriteri	32
6	TEPE VİNCİ SİSTEMİNİN DURUM UZAY SİMULİNK MODELİ.....	34
6.1	Sistemin Açık Çevrim Cevabı	34
6.2	Ayrık Zamanlı (Discrete) Durum-Uzay Modeli.....	36
6.2.1	Ayrık Zamanda Açık Çevrim Cevabı	37
7	GERİBESLEME KONTROL SİSTEMİ TASARLANMASI.....	38
7.1	Amaç.....	38
7.2	Geribesleme Kontrol Sistemleri	38
7.3	Sistemin Luenberger Durum Gözlemcisi Tasarımı (Observer Design)	38
7.4	Kutup Yerleşim Metodu ile Sistemin Durum-Uzay Kontrolü.....	49
7.4.1	Sisteme İntegratör Eklenmesi	54
7.4.2	Sisteme Distürbans Eklenmesi	58
7.4.3	Değişken yüklerde Kutup Yerleşim Metodu Cevabı.....	61
7.5	LQR Yöntemi ile Sistemin Durum-Uzay Kontrolü.....	65
7.5.1	Q_{11} ve Q_{33} Ağırlıklarının Seçilmesi	67
7.5.2	Sisteme İntegratör Eklenmesi	71
7.5.2	Sisteme Distürbans Eklenmesi	73
7.5.3	Değişken yüklerde LQR Metodu Cevabı	75
8	SONUÇLAR.....	80
	KAYNAKLAR.....	81
	EKLER	83
Ek 1	Örnek kutup noktaları ve zaman cevapları	83
Ek 2	Program Kodları	84
	ÖZGEÇMİŞ.....	103

SİMGE LİSTESİ

x	Arabanın yatay pozisyonu
x_1	Sarkacın yatay pozisyonu
θ	Salınım açısı
y	Sarkacın dikey pozisyonu
l	Sarkaç ipi uzunluğu
M	Arabanın kütlesi
m	Sarkaç yükünün kütlesi
g	Yerçekimi ivmesi
L	Kinetik ve potansiyel enerjiden oluşan Lagrangian sistemi
T	Sistemin toplam kinetik enerjisi (E_K)
U	Sistemin toplam potansiyel enerjisi (E_P)
D	Sürtünmede kaybolan enerji
i	$x(t)$ ve $\theta(t)$ serbestlik dereceleri
r	Referans giriş
q_i	Genelleştirilmiş koordinat setleri ; $i=1,2,3,\dots,n$
F_i	q serbestlik derecesinde oluşan net kuvvet
\dot{x}	Arabanın hızı
\dot{x}_1	Sarkacın yatay hızı
$\dot{\theta}$	Salınım hızı
ξ	Sönüm oranı
n	Sistem derecesi
τ	Zaman sabiti

KISALTMALAR LİSTESİ

LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
IEEE	Institute of Electrical and Electronics Engineers

ŞEKİL LİSTESİ

Şekil 1.1 Rotary vinç	3
Şekil 1.2 Boom vinç	3
Şekil 1.3 Gantry vinç	4
Şekil 2.1 Tepe vinci sisteminin şematik gösterimi	5
Şekil 2.2 Kaskat halde transfer fonksiyonu gösterimi	10
Şekil 4.1 İntegratör içeren tip1 servo sistem	17
Şekil 4.2 İntegratör içermeyen tip1 servo sistem	18
Şekil 4.3 Durum gözlemcili integratör içermeyen tip1 servo sistem	20
Şekil 4.4 Ayrık zamanlı integratör içermeyen tip1 servo sistem.....	20
Şekil 4.5 Durum gözlemcili tip1 servo sistem genel gösterimi.....	23
Şekil 5.1 Geribesleme ile kutup yerleşim metodu.....	25
Şekil 5.2 Durum yeniden yapılandırıcısı (gözlemcisi) nin şematik gösterimi	30
Şekil 5.3 Sistemin durum yeniden yapılandırıcısı ile beraber genel gösterimi	31
Şekil 5.4 Durum yeniden yapılandırıcısı	31
Şekil 6.1 Vinç sisteminin Simulink modeli	34
Şekil 6.2 Sistemin basamak cevabı – x konumu.....	34
Şekil 6.3 Sistemin basamak cevabı – $teta$ açısı	35
Şekil 6.4 Sistemin başlangıç koşulu cevabı – x konumu.....	35
Şekil 6.5 Sistemin başlangıç koşulu cevabı – $teta$ açısı	36
Şekil 6.6 Ayrık zamanlı sistemin basamak cevabı – x konumu	37
Şekil 6.7 Ayrık zamanlı sistemin basamak cevabı – $teta$ açısı.....	37
Şekil 7.1 Araba ve sarkaç (vinç) sisteminin estimator tahmin edicili Simulink modeli	40
Şekil 7.2 Gözlemci dizaynı $F(t)$ girişi - $x(t)$ konum cevabı.....	47
Şekil 7.3 Yeniden yapılandırılmış konum sinyalleri	47
Şekil 7.4 Yeniden yapılandırılmış açı sinyalleri.....	48
Şekil 7.5 Gözlemcili sistemin geribesleme Simulink modeli.....	51
Şekil 7.6 Kutup yerleşim metodu basamak kuvvet girişi cevabı.....	52
Şekil 7.7 Kutup yerleşim metodu basamak kuvvet girişi cevabı- $x(t)$ konumu.....	53
Şekil 7.8 Kutup yerleşim metodu basamak kuvvet girişi cevabı- $teta$ açısı.....	53
Şekil 7.9 İntegratörlü sistemin genel gösterimi	54
Şekil 7.10 Terminator altsistemi.....	55
Şekil 7.11 İntegratör içeren gözlemcili geribesleme Simulink modeli	56
Şekil 7.12 İntegratör katmanı sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	57
Şekil 7.13 İntegratör katmanı sonrası $teta$ açısı cevabı	57
Şekil 7.14 İntegratör katmanı sonrası uygulanan kontrol kuvveti.....	58
Şekil 7.15 Distürbans eklenmiş Simulink modeli	59
Şekil 7.16 Distürbans sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	60
Şekil 7.17 Distürbans-kontrol sinyali cevabı.....	60
Şekil 7.18 Distürbans sonrası $teta$ açısı cevabı	61
Şekil 7.19 50 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	62
Şekil 7.20 50 kg sarkaç yükünde $teta$ açısı cevabı	62
Şekil 7.21 0,5 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	63
Şekil 7.22 0,5 kg sarkaç yükünde $teta$ açısı cevabı	63
Şekil 7.23 100 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	64
Şekil 7.24 100 kg sarkaç yükünde $teta$ açısı cevabı	64
Şekil 7.25 LQR metodu basamak kuvvet girişi cevabı	70
Şekil 7.26 LQR metodu basamak kuvvet girişi cevabı- $x(t)$ konumu	70

Şekil 7.27 LQR metodu basamak kuvvet girişi cevabı- $teta$ açısı.....	71
Şekil 7.28 LQR metodu integratör katmanı sonrası $x_c(t)$ - $x(t)$ konumu	72
Şekil 7.29 LQR metodu integratör katmanı sonrası $teta$ açısı.....	72
Şekil 7.30 LQR metodu integratör katmanı sonrası uygulanan kontrol kuvveti.....	73
Şekil 7.31 LQR kontrolü distürbans sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı.....	74
Şekil 7.32 LQR kontrolü distürbans-kontrol sinyali cevabı.....	74
Şekil 7.33 LQR kontrolü distürbans sonrası $teta$ açısı cevabı.....	75
Şekil 7.34 LQR metodu 50 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	76
Şekil 7.35 LQR metodu 50 kg sarkaç yükünde $teta$ açısı cevabı	76
Şekil 7.36 LQR metodu 0,5 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı	77
Şekil 7.37 LQR metodu 0,5 kg sarkaç yükünde $teta$ açısı cevabı	77
Şekil 7.38 LQR metodu 100 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı ...	78
Şekil 7.39 LQR metodu 100 kg sarkaç yükünde $teta$ açısı cevabı	78
Şekil 7.40 Örnek kutup noktaları ve zaman cevapları.....	83

ÖNSÖZ

Öncelikle, bu tez hazırlanma safhası boyunca yardımlarını esirgemeyen ve değerli zamanını ayırarak çalışmamda yol gösterici olan danışman hocam sayın Yrd.Doç. Dr. Şeref Naci ENGİN' e teşekkür etmeyi bir borç bilir; aileme sabır ve desteklerinden dolayı şükranlarımı sunarım.

ÖZET

Vinçler, günümüzde ağır cisimlerin bir yerden başka bir yere nakledilmesinde en çok kullanılan yapılar olup; genellikle operatör kontrolünde manuel hareket ettirilmektedirler. Bu da sistemde istenmeyen salınımlara yol açmakta hem cismin kontrolsüz şekilde taşıyıcı halat etrafında osilasyona girmesine yol açmakta; hem de etraftaki diğer cisimler; çalışanlar için hayati tehlike oluşturmaktadır.

Bu tez çalışmasında, farklı iki kontrol metodu kullanarak sisteme kontrolcü tasarlanmış ve bahsedilen vinç salınımlarını en aza indirerek pozisyon kontrolü yapılması amaçlanmıştır. Öncelikle en çok kullanılan vinç yapısı olan tepe vinci sistemi ele alınmış; 2 boyutta dijital sistem modellemesi yapılarak sistem karakteristikleri elde edilmiştir. Bulunan karakteristikler lineerleştirilerek üzerinde asıl çalışılacak olan lineerleştirilmiş sistem durum-uzay modelinde gösterilmiştir. Sistem modelleri Simulink yazılımı ile görsel forma sokulmuş; denklemlerin elde edilmesinde ise Matlab yazılımından faydalanılmıştır. Kontrol edilebilirlik ve gözlenebilirlik kavramları işlenmiş ayrıca servo sistemler ele alınarak sistemin teorisine katkı sağlanmıştır.

Sistemin tahmin edici-gözlemci dizaynı yapılarak tüm durum değişkenleri elde edilerek yapılan modele eklenmiştir. Kontrolcü olarak kutup yerleşim metodu ve LQR linear quadratic regulator yöntemleri seçilerek; sistem bu iki kontrolcü için teker teker ele alınarak sistem cevapları elde edilmiştir. Elde edilen cevaplar daha sonra dış distürbanslara karşı test edilerek sonuçları gözlenmiştir. Vinçlerde taşınacak cisimlerin ağırlıkları değişken olduğundan; son olarak bulunan katsayılar ile yapılan geribesleme kontrol sistemleri, değişken yüklerde denenerek sonuçları analiz edilmiştir. Yapılan simülasyonlar sonrasında elde edilen grafiklerden sonuçlar çıkartılarak; aynı sisteme uygulanan LQR ve kutup yerleşim metodlarının birbirlerine olan üstünlükleri ve zayıflıklarına dikkat çekilerek tez sonlandırılmıştır.

Matematiksel işlemler gerçekleştirilirken klasik yöntemler ile sistem çözümlenmiş; ayrıca Matlab komut satırları da açıklanarak teze eklenmiştir.

Anahtar kelimeler: Tepe vinci, kontrol teorisi, Matlab, Simulink, kutup yerleşimi, LQR, gözlemci dizaynı, lineer kontrol

ABSTRACT

Nowadays cranes are the mostly used tools for transporting heavy loads from one place to another. They are generally operated manually which causes unwanted oscillation movement of the load that causes danger to workers and surroundings.

In this thesis, using two different control methods, controllers designed for the system to give the desired position with minimum oscillation of the load. First we deal with widely used crane construction: the overhead crane. Modeled in 2 dof discrete state-space and system characteristics obtained. These characteristics linearized and the main system model, which is the linearized digital state-space model, is shown. By using Simulink software; the system model is transformed into visual form and the Matlab software is used for obtaining equations. Controllability, observability and servo system concepts are also represented.

By estimator-observer design of the system, all state variables obtained and added to system model. LQR linear quadratic regulator and pole placement methods are chosen as controllers and by taking into consideration of those two controller; the system responses obtained for each. With adding disturbances to the obtained system, the disturbance responses of the system observed.

Because of the varying mass of the loads, the feedback control systems obtained by feedback coefficients are tried on varying loads and plots analyzed. After all simulations done; conclusions made based on responses of the system to LQR and pole placement methods. Taking consideration the superiority and weakness of both controllers, the thesis is finalized.

Both classical approach and Matlab command lines are used to overcome the mathematical equations. All command lines that are used, added to the thesis.

Keywords: Overhead crane, control theory, Matlab, Simulink, pole placement, LQR, observer design, linear control

1. GİRİŞ

1.1 Motivasyon

Günümüzde sanayide ağır nesnelere bir yerden başka bir yere taşınması işlemlerinde en çok vinçlerden faydalanılmaktadır. Fakat yükün salınım hareketi genellikle operasyon verimi ve güvenliğini sınırlamaktadır. Yükün salınım açısını düşürmek için, vinç özel bir kontrol şemasına ihtiyaç duyar.

Bu problem popüler olarak vinç sistemlerinin salınım (anti-swing) kontrolü olarak bilinir ve kontrol mühendislerinin uzun zamandır ilgisini çekmektedir. Kontrolör daha önceden bir ön bilgi olmadan geniş bir aralıktaki yükleri taşıyabilecek kapasitede olmalıdır, bu da endüstriyel bir vinç için tipik bir senaryodur. Bu salınım açısını gidermek üzere öngördüğümüz tasarımda sistem modeli, klasik kontrol yöntemleri olan PD – PID Kontrolörlere göre daha sağlıklı sonuç sağlayan Kutup Yerleşimi (Pole Placement) Metodu ve LQR (Linear Quadratic Regulator) yöntemleri ile kontrol edilecektir.

Özellikle sanayide kullanılan vinçler, bir operatör tarafından manuel olarak kullanılmaktadırlar. Burada operatörün en küçük bir ani hareketi bile sistemde büyük salınımlar oluşturarak bir çok hataya neden olmaktadır.

1.2 Literatür Taraması

Bir tepe vinçinin (overhead crane) modellenmesi ve kontrolü uzun zamandır araştırması konusu olmaktadır. Birçok araştırmacı vinçlerin salınım kontrolü konusu üzerinde çalışmıştır. Onların çalışmaları genel olarak 2 grupta toplanabilir:

- Açık çevrim kontrol
- Kapalı Çevrim kontrol

Açık çevrim metotlar; alçak geçiren (low pass) filtre, simetrik ivmelenme profili (symetric acceleration profile) ve giriş şekillendirme (input shaping) metotları salınım hareketini harekete geçirmemeye çalışmaktadır. Uygulaması basittir fakat dış distürlanslardan kaynaklanan salınım hareketini bastıramazlar. Aynı zamanda geniş bir yük yelpazesinde tatmin edici performans verememektedirler.

Diğer taraftan salınım hareketinin kapalı çevrim metodu, var olan salınım hareketini elemine etmeyi sağlar ve eklenen sensörlerin yardımıyla distürlanslardan kurtulur.

Ho-Hoon Lee kök yer eğrisi temelli kapalı çevrim kontrolör geliştirmiştir. Aynı zamanda 3 boyutlu dinamik tepe vinci modelini 2 serbestlik dereceli salınım açısına dayanarak yaklaşık olarak elde etmiştir. Ama ayrık (decoupled) kontrol yasasını bulmak için dinamik eşitliğini lineerleştirmiştir. Yeni ayrık kontrol yasası değişen yüklerde çalışmamakta, yük hakkında ekstra bilgiye ihtiyaç duymaktadır.

Wen, LQR metoduna dayalı bir kapalı çevrim kontrolör tasarlamıştır. Bu yaklaşımda da aynı şekilde yük hakkında önceden bilgi sahibi olmak gerekmektedir.

Chi-Chen Cheng ve Cheng-Yi Chen geribesleme lineerleştirme tekniği kullanan kontrol yasası elde etmiştir. Bu metotta değişen yük konusu işlenmiştir.

J.A. Mendez self-tuning kontrolüne dayalı daha kompleks nöral network kontrolör hazırlamıştır. M. Mahfouf ve A. Benhidjeb fuzzy logic tabanlı kontrolör geliştirmişlerdir. Yukarıda bahsedilen tüm kontrolörler bir dereceli lineer tepe vinci modelleriyle sınırlıdır.

Ji Sup Yoon kapalı ve açık çevrim yaklaşımları kullanan salınımı giderici bir çok kontrolör dizayn etmiştir. Daha verimli ve robust kontrolör geliştirmek için, önceden programlanmış geribesleme kontrolör ve fuzzy kontrolör geliştirmiştir. Tüm kontrolörlerini 2 serbestlik dereceli lineer vinç sistemine uygulamıştır.

Genelde analizler sadece 2 serbestlik dereceli tepe vinçlerinde sınırlandırılmıştır. Aynı zamanda vinç kriterleri genellikle ihmal edilmiş ve sadece bir hız kaynağı olarak kontrol edilmiştir. Yapılan çalışmalardan görülebildiği gibi tüm sistemin kesin modelini gerektirmekte ve geniş bir yük yelpazesinde salınım kontrolünü tam olarak sağlayamamaktadır.

1.3 Amaç

Bu bahsedilen hataları önlemek üzere bir kontrolcü tasarlanması ve iki serbestlik derecesinde; salınım açısını minimize ederek pozisyon kontrolü sağlayacak operatörsüz çalışabilen bir tepe vinci sistemi simüle etmektir. Yapılan çalışma; sistem dinamiklerinin teorik analizlerini ve kontrol metotlarını içermektedir. Matlab yazılımı ve Simulink blokları yardımı ile sistemin simülasyonu elde edilerek sonuçların alınmasında kolaylık sağlanmış ayrıca görsel olarak sistemin daha kolay anlaşılmasına yardımcı olunmuştur.

1.4 Vinç Çeşitleri

Vinçler mekanik ve dinamik yapılarına göre genel olarak 3 çeşide ayrılmıştır.

- 1 eksen etrafında dönen vinç (Rotary crane)

İnşaat sektöründe en çok kullanılan vinç tipidir. Sabit bir nokta etrafında rotasyonel hareket yaparak bir konumdan istenilen başka bir konuma geçiş yapabilmektedir.



Şekil 1.1 Rotary crane (Al-Mousa 2000)

- Boom vinci

Gemi ve limanlarda kullanılan çok aksamlı vinç türüdür. Hem radyal hem de yatay hareket sağlayabilmektedir.



Şekil 1.2 Boom crane (Al-Mousa 2000)

- Tepe vinci (Gantry- Overhead crane)

X - Y - Z düzlemlerinde doğrusal olarak hareket ederek daha hassas konum kontrolü sağlayan bir

vinç türüdür. Kullanım sahası en geniş olan vinçtir. Açık ve kapalı alanlarda kullanıma uygundur. Tez çalışmasında üzerinde çalışılan vinç tipidir.



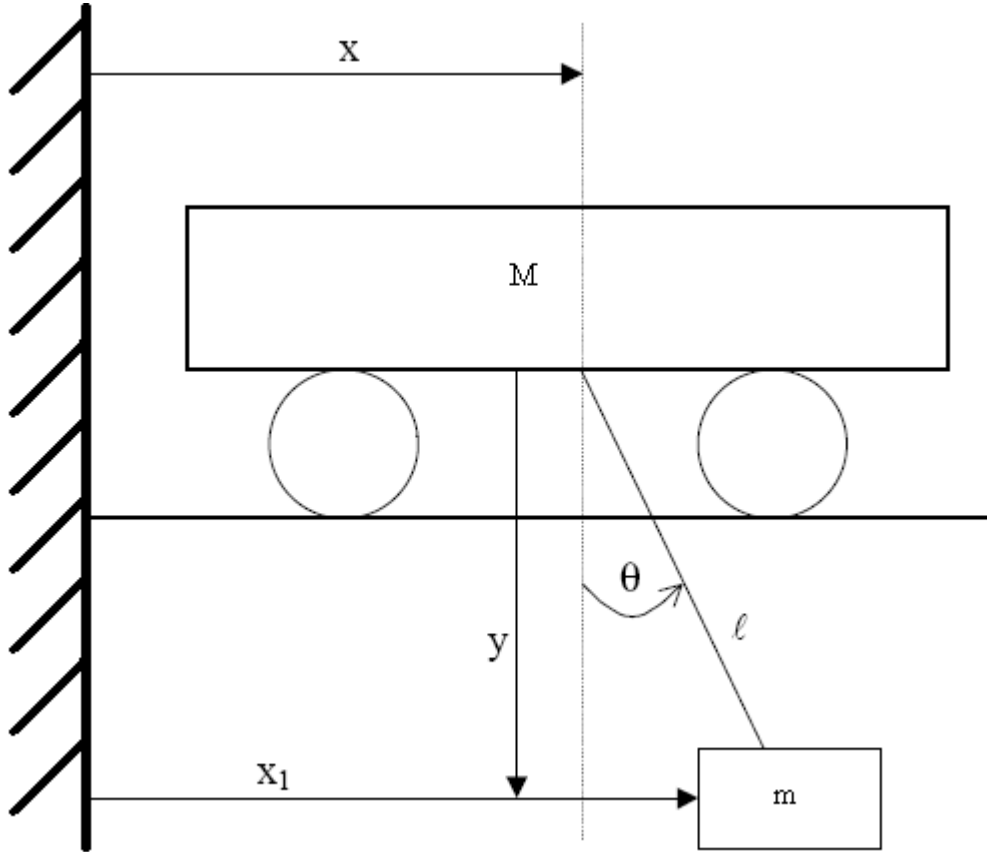
Şekil 1.3 Gantry crane (Al-Mousa 2000)

2. TEPE VİNCİNİ OLUŞTURAN SARKAÇ VE ARABA SİSTEMİNİN MODELLENMESİ

Bu bölümde vinci oluşturan araba ve kart sisteminin matematiksel modelinin elde edilmesi açıklanmıştır. Durum eşitlikleri, enerji denklemlerinden faydalanılarak Lagrangian yaklaşımı ile elde edilmiştir. Çalışma noktası etrafında lineerleştirilerek sistemin transfer fonksiyonu bulunmuştur.

2.1 Lagrangian Dinamikleri Analizi

İlk olarak genelleştirilmiş koordinatlar seçilmiş; enerji fonksiyonları ve de lagrangian denklemleri sonucu hareket denklemleri elde edilmiştir.



Şekil 2.1 Tepe vinci sisteminin şematik gösterimi

Şekil 2.1 de basit bir tepe vinci sisteminin genel diyagramı gösterilmiş olup analizler için bu model kullanılmıştır. Burada araba ağırlığı M , sarkaç ağırlığı m olarak adlandırılmış; kullanılan ipin ağırlıksız ve rijit olduğu kabul edilmiştir. Ayrıca sistem sürtünmesiz kabul edildiğinden sürtünme bileşenleri modelde gösterilmemiştir.

x : arabanın yatay pozisyonu (m)

x_I : sarkacın yatay pozisyonu (m)

y : Sarkacın dikey pozisyonu (m)

θ : salınım açısı (rd)

l : Sarkaç ipi uzunluğu (m)

İki serbestlik derecesine sahip bu sistemin modeli Lagrangian dinamikleri kullanarak elde edilmiştir. Lagrangian eşitlikleri hareket denklemlerini, sürtünmeli ve de sürtünmesi ihmal edilmiş sistemler için elde etmemizi sağlar. Sürtünmesiz yaklaşımda sistemin enerji kaybının olmadığı kabul edilir. Lagrangian eşitlikleri kullanan bir sistemin dinamiklerini elde etmek için gerekli matematiksel eşitlikler, aşağıdaki şekilde verilmiştir:

Kinetik ve potansiyel enerji denklemlerinden

$$L = T - U = E_k - E_p \quad (2.1)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = F_i \quad (2.2)$$

(Williams, James 1996)

L : Kinetik ve potansiyel enerjiden oluşan Lagrangian sistemi

T : Sistemin toplam kinetik enerjisi (E_K)

U : Sistemin toplam potansiyel enerjisi (E_P)

D : Sürtünmede kaybolan enerji

F_i : q serbestlik derecesinde oluşan net kuvvet

i : $x(t)$ ve $\theta(t)$ serbestlik dereceleri (number of degrees of freedom)

q_i : genelleştirilmiş koordinat setleri ; $i=1,2,3,\dots,n$

Lagrangian fonksiyonu oluşturmak için toplam kinetik enerjiden sistemin toplam potansiyel enerjisinin çıkartılması gerekmektedir.

Hareket Halindeki Sistemin Kinetik Enerji Modeli:

Arabanın Kinetik Enerjisi:

$$T_1 = \frac{1}{2}M\left(\frac{dx}{dt}\right)^2 = \frac{1}{2}M(\dot{x})^2 \quad (2.3)$$

Sarkaç Yükünün Kinetik Enerjisi:

$$T_2 = \frac{1}{2}m(\dot{x}_1^2 + \dot{y}^2) \quad (2.4)$$

Burada

$$x_1 = x + l \sin \theta, \quad y = l \cos \theta \quad (2.5)$$

$$\dot{x}_1 = \dot{x} + l\dot{\theta} \cos \theta, \quad \dot{y} = -l\dot{\theta} \sin \theta \quad (2.6)$$

araba + sarkaç kinetik enerjisi toplamı

$$E_K = T_1 + T_2 = \frac{1}{2}M(\dot{x})^2 + \frac{1}{2}m\left[(\dot{x} + l\dot{\theta} \cos \theta)^2 + (-l\dot{\theta} \sin \theta)^2\right] \quad (2.7)$$

Olarak bulunur. Buradan:

$$T = E_K = \frac{1}{2}(M + m)\dot{x}^2 + mxl\dot{\theta} \cos \theta + \frac{1}{2}m\left[(l\dot{\theta})^2 (\cos^2 \theta + \sin^2 \theta)\right] \quad (2.8)$$

elde edilir.

Sistemin Potansiyel Enerjisi:

Araba sadece yatay ekseninde hareket ettiği için, sistemin potansiyel enerjisi sadece sarkaç açısından kaynaklanmaktadır.

$$U = E_p = mgl - mgl \cos \theta \quad (2.9)$$

Lagrangian Denklemi:

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

$$L = T - U = \frac{1}{2}(M + m)\dot{x}^2 + mxl\dot{\theta}\cos\theta + \frac{1}{2}m\left[l\dot{\theta}\right]^2(\cos^2\theta + \sin^2\theta) - mgl(1 - \cos\theta) \quad (2.10)$$

Sistemde arabaya ve salınım dinamiklerine etki eden sadece bir adet dış kuvvet bulunmaktadır. Bu da sisteme yatay ekseninde hareket kazandıran F_x kuvvetidir. θ koordinat sistemine herhangi bir dış kuvvet direkt olarak etki etmemektedir.

Ayrıca simülasyon çalışmasında 2 serbestlik derecesi içinde sürtünme kayıpları göz ardı edildiğinden $D=0$ alınmıştır.

O halde (2.2) eşitliği aşağıdaki şekli alır:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F \quad (2.11)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = F_\theta = 0 \quad (2.12)$$

yazılabilir.

q(t)=θ(t) Serbestlik Derecesi için Lagrange Denklemi:

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2\dot{\theta} + m\dot{x}\cos\theta \quad (2.13)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = -ml^2\ddot{\theta} + m\ddot{x}\cos\theta - m\dot{x}\dot{\theta}\sin\theta \quad (2.14)$$

$$\frac{\partial L}{\partial \theta} = -m\dot{x}\dot{\theta}\sin\theta - mgl\sin\theta \quad (2.15)$$

Lagrange denkleminde yerine yerleştirildiğinde:

$$l\ddot{\theta} + \dot{x}\cos\theta + g\sin\theta = 0 \quad (2.16)$$

elde edilir.

q(t)= x(t) Serbestlik Derecesi için Lagrange Denklemi:

Karta uygulanan kuvvet $F_x = F(t)$;

$$\frac{\partial L}{\partial \dot{x}} = (M + m)\dot{x} + ml\dot{\theta}\cos\theta \quad (2.17)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = (M + m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml(\dot{\theta})^2\sin\theta \quad (2.18)$$

$$\frac{\partial L}{\partial x} = 0 \quad (2.19)$$

Lagrange denkleminde yerine yerleştirildiğinde:

$$(M + m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml(\dot{\theta})^2\sin\theta = F(t) \quad (2.20)$$

elde edilir.

2.2 Çalışma Noktası Yakınında Lineerleştirme

Elde edilen model non-lineer dir. Sistem üzerinde daha fazla çalışma yapmadan önce sistemin lineerleştirilmesi gerekmektedir.

Sarkacın denge noktası θ_0 etrafında küçük bir değişim gösterdiğini kabul edelim. Yeni değerine θ adını verelim:

$$\theta = \theta_0 + \varepsilon \quad (2.21)$$

$$\dot{\theta} = \dot{\varepsilon} \quad (2.22)$$

Taylor Serisi açılımından, θ 'nın herhangi bir fonksiyonunun birinci derece yaklaşımı

$$f(\theta) \approx f(\theta_0) + \varepsilon \left. \frac{df}{d\theta} \right|_{\theta_0} \quad (2.23)$$

olup, aynı zamanda daha büyük dereceli terimler ihmal edildiğinden

$$\dot{\varepsilon}^2 \approx 0 \quad (2.24)$$

$\dot{\varepsilon}^2 \approx 0$ $\theta=0$ için :

$$\begin{aligned} \cos\theta &\approx \cos(0) + \theta[-\sin(0)] = 1 \\ \sin\theta &\approx \sin(0) + \theta[\cos(0)] = \theta \end{aligned} \quad (2.25)$$

yazılabilir.

Sonuçları elde ettiğimiz Lagrange eşitliklerinde yerlerine koyarsak:

$$\left. \begin{aligned} (M+m)\ddot{x} + ml\ddot{\theta} &= F(t) \\ \ddot{x} + l\ddot{\theta} + g\theta &= 0 \end{aligned} \right\} \text{Lineer Diferansiyel Denklemleri elde edilir.} \quad (2.26)$$

Düzenlenirse:

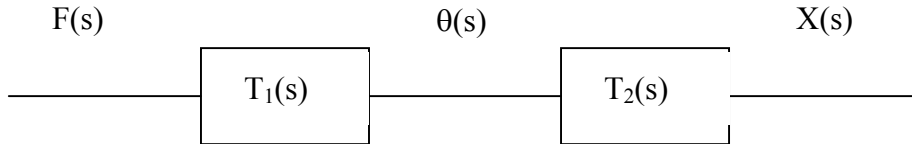
$$\left. \begin{aligned} \ddot{\theta}(t) &= -g \frac{(M+m)}{Ml} \theta(t) - \frac{1}{Ml} F(t) \\ \ddot{x}(t) &= g \frac{m}{M} \theta(t) + \frac{1}{M} F(t) \end{aligned} \right\} \text{bulunur.} \quad (2.27)$$

2.3 Sistemin Transfer Fonksiyonu

Diferansiyel denklemlerden, $\theta(t)$ nin $F(t)$ ile; $x(t)$ nin ise hem $\theta(t)$ hem de $F(t)$ ile bağlantılı olduğunu görüyoruz.

Bu prosesi kaskat bağlı iki transfer fonksiyonu ile göstermek mümkündür:

$$T_1 = \frac{\theta(s)}{F(s)} \quad \text{ve} \quad T_2 = \frac{X(s)}{\theta(s)} \quad \text{olarak aldığımızda} \quad (2.28)$$



Şekil 2.2 Kaskat halde transfer fonksiyonu gösterimi

Laplace Transformu alındığında;

$$s^2\theta(s) + g \frac{(M+m)}{Ml} \theta(s) = -\frac{1}{Ml} F(s) \quad s^2 X(s) = g \frac{m}{M} \theta(s) + \frac{1}{M} F(s) \quad (2.29)$$

Buradan

$$T_1(s) = \frac{\theta(s)}{F(s)} = \frac{-1}{Mls^2 + g(M+m)} \quad (2.30)$$

$$T_2(s) = \frac{X(s)}{\theta(s)} = -\frac{g + ls^2}{s^2} \quad (2.31)$$

elde edilir.

2.4 Durum-Uzay (State-Space) Gösterimi

Vinç sistemini durum uzay modeli şeklinde göstermemizin sebebi; sistemi matrisel forma sokmak istememizdir.

Durum uzay gösterim şekli

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad \mathbf{Y} = \mathbf{Cx} + \mathbf{Du} \quad (2.32)$$

cinsindedir.

Burada durum vektörü X için notasyonlarda değişiklik yapmak gerekmektedir. Şöyle ki:

$X_1 = x$, $X_2 = \dot{x}$, $X_3 = \theta$, $X_4 = \dot{\theta}$ olarak atarsak

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{pmatrix} \quad (2.33)$$

Elde ettiğimiz düzenlenmiş lineer diferansiyel denklemler kullanılarak:

$$\frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & gm/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -g(M+m)/Ml & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/Ml \end{bmatrix} F(t) \quad (2.34)$$

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F(t) \quad (2.35)$$

elde edilir.

A sistemin dinamiği matrisi, U giriş, Y çıkış, B , C ve D matrisleri yardımcı matrislerdir.

Simülasyon için $M=10$; $m=5$; $l=1$; ve de $g=10$ alınmış olup elde edilen matrisler

$$\dot{x} = Ax + Bu \text{ ve } y = Cx + Du$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -15 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0.1 \\ 0 \\ -0.1 \end{bmatrix} u \quad (2.36)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (2.37)$$

halini almıştır. Matlab ile bu sistem aşağıdaki komut satırı ile elde edilebilmektedir:

% Açık çevrim durum-uzay sürtünmesiz vinç sistemi

M = 10; m = 5

l = 1; g = 10

k1 = m*g/M

k2 = -(M+m)*g/(M*l)

k3 = 1/M

k4 = -1/(M*l)

% Durum-uzay matris gösterimi

A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0]

B = [0; k3; 0; k4]

C = [1 0 0 0; 0 0 1 0]

D = 0

%Kod sonu

3. KONTROLEDİLEBİLİRLİK VE GÖZLENEBİLİRLİK KAVRAMLARI

3.1 Kontrol edilebilirlik (controllability)

Tanım: Verilen bir $x(t_i)$ durum vektöründe; sistemin her bir durum değişkeni, eğer sınırlandırılmamış $u(t)$ kontrol değişkenleri ile, herhangi bir başlangıç konumundan istenen başka bir keyfi konuma sonlu örnekleme periyodu içinde transfer edebiliyorsa sistem tamamen KONTROLEDİLEBİLİRDİR denir.

Teorem:

$$\dot{x} = Ax(t) + Bu(t) ; \quad (3.1)$$

ve ayrık zamanda

$$x(k+1) = A_d x(k) + B_d u(k) \quad (3.2)$$

x : Durum vektörü (n vektör)

u : kontrol sinyali

A : $n \times n$ lik bir matris

B : $n \times 1$ lik bir matris

Olarak verilen n . Dereceden bir sistemde;

Kontrol edilebilirlik matrisinin rankı n e eşit ise sistem durum kontrol edilebilirdir.

$$x(nT) = A^n x(0) + \sum_{j=0}^{n-1} A^{n-j-1} B u(jT) \quad (3.3)$$

$$= A^n x(0) + A^{n-1} B u(0) + A^{n-2} B u(T) + \dots + B u((n-1)T) \quad (3.4)$$

Matrisel forma geçerse ;

$$x(nT) - A^n x(0) = \begin{bmatrix} B & AB & \dots & A^{n-1} B \end{bmatrix} \begin{bmatrix} u((n-1)T) \\ u((n-2)T) \\ \vdots \\ u(0) \end{bmatrix} \quad (3.5)$$

Kontrol edilebilirlik matrisi

$$M_{cont} = [B : AB : \dots : A^{n-1}B] \quad (3.6)$$

$$\text{ayrık zamanda } M_{contd} = [B_d : A_d B_d : \dots : A_d^{n-1} B_d] \quad (3.7)$$

dir.

$$\text{rank } [B : AB : \dots : A^{n-1}B] = n \quad ; \quad [B_d : A_d B_d : \dots : A_d^{n-1} B_d] = n \quad (3.8)$$

(Kuo 1999)

Bu denklemin sağlanması sistemin kontroledilebilmesi için yeter koşuldur.

matlab de kontroledilebilirlik denklemi;

ctrb(A,B); komut satırı girilerek elde edilebilmektedir.

3.2 Gözlenebilirlik (Observability)

Tanım: Bir dinamik sistemin çıkış ve kontrol vektörlerinin incelenerek; durum-konumunu sonlu sayıda örnekleme periyodu içinde belirlemekle ilgilidir.

Eğer $x(0)$ konumundaki bir sistemin konumunu; sonlu sayıda örnekleme periyodu içinde, çıkış ve kontrol vektörlerini gözlemleyerek elde edebiliyorsak sistem GÖZLEMLENEBİLİRDİR denir. Kısaca çıkış ve giriş ölçümlerinden durum değişkenleri hakkında bilgi edinmek olarak açıklanabilir.

Gözlenebilirlik ve kontroledilebilirlik kavramları R.E. KALMAN tarafından bulunmuştur.

Teorem:

aşağıdaki eşitliklerdeki gibi tanımlı n . dereceden bir sistemin

$$\dot{x} = Ax(t) + Bu(t) \quad \text{ve} \quad y = Cx(t) + Du(t) \quad ; \quad (3.9)$$

$$\text{ayrık zamanda} \quad x(k+1) = A_d x(k) + B_d u(k) \quad y(k) = C_d x(k) + D_d u(k) \quad (3.10)$$

gözlenebilirlik matrisinin rankı n e eşit ise sistem gözlenebilirdir.

$$x(kT) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-j-1} B u(jT) \quad (3.11)$$

$$y(kT) = CA^k x(0) + \sum_{j=0}^{k-1} CA^{k-j-1} Bu(jT) + Du(kT) \quad (3.12)$$

A, B, C, D matrisleri ve $u(kT)$ 'nin bilinmesinden dolayı son denklemin sağ tarafındaki 2. ve 3. terimler bilinmektedir. Çıkışı gözlemleyip $x(0)$ 'ı bulduktan sonra $x(kT)$ 'de bulunabilir.

($u(0), u(T), \dots, u(kT)$ bilindiğinden dolayı)

Ayrık zamanda;

Sonlu sayıda örnekleme periyodu içinde verilen T çıkışı, başlangıç durumu vektörü $x(0)$ 'ı belirlemeyi mümkün kılıyorsa sistem tam olarak gözlenebilir.

$$x(kT) = A^k x(0) \quad (3.13)$$

$$y(kT) = CA^k x(0) \quad (3.14)$$

Tam gözlemlenebilirlik; verilen $y(0), y(T), y(2T) \dots$ 'dan $x_1(0), x_2(0) \dots, x_n(0)$ 'ı elde etmek demektir.

$$y(0) = Cx(0)$$

$$y(T) = CA x(0) \quad (3.15)$$

⋮

$$y((n-1)T) = CA^{n-1} x(0)$$

Buradan $nm \times n$ 'lik matris $\begin{bmatrix} C \\ \dots \\ CA \\ \dots \\ \vdots \\ \dots \\ CA^{n-1} \end{bmatrix}$ 'in rankı n olmalıdır.

Bu matrisin konjüge tranzpozesi alınırsa $n \times nm$ 'lik matris ;

$[C' : A'C' : \dots : (A')^{n-1} C']$ 'dir ve rankı da n 'e eşit olmalıdır. Bu matrisi genel olarak gözlenebilirlik matrisi denir.

Gözlenebilirlik matrisi aşağıdaki genel formdaki biçimdedir:

$$N_{obs} = [C : CA : \dots : CA^{n-1}]^T \quad (3.16)$$

$$\text{Ayrık zamanda } N_{obsd} = [C_d : C_d A_d : \dots : C_d A_d^{n-1}]^T \quad (3.17)$$

(Kuo 1999)

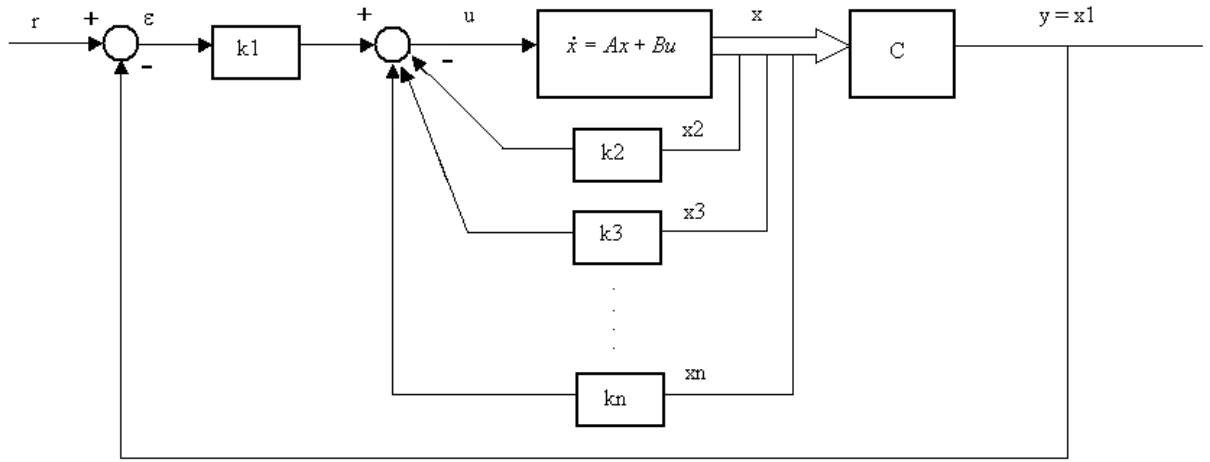
Matlab de gözlenebilirlik denklemi

obsv(A,B) komut satırı girilerek elde edilebilmektedir.

4. TİP 1 SERVO SİSTEMİN TANIMLANMASI

Tip 1 servo sistem dizayn edilmesinin temel amacı stabil ($n+1$). dereceden regülatör sistemi tasarlayarak hata vektörü $e(t)$ yi; verilen herhangi bir başlangıç koşulu $e(0)$ için, sifıra götürebilmektir. Aşağıda sürekli zaman ve de ayrık zamanlı sistemler incelenmiş ve eşitlikleri gösterilmiştir.

4.1 İntegratör İçeren Tip 1 Servo Sistemin Sürekli Zamanda Genel Gösterimi



Şekil 4.1 İntegratör içeren tip1 servo sistem

$$\dot{x} = Ax + Bu; \quad (4.1)$$

$$u = -\begin{bmatrix} 0 & k_2 & k_3 & \dots & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + k_1(r - x_1) \quad (4.2)$$

$$K = \begin{bmatrix} k_1 & k_2 & k_3 & \dots & k_n \end{bmatrix} \quad (4.3)$$

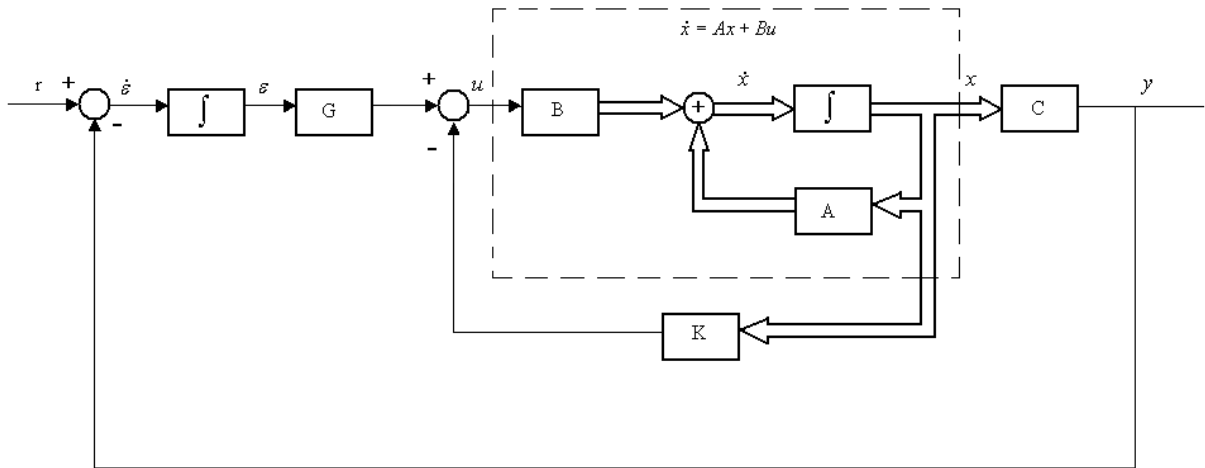
olduğundan,

$$u = -Kx + k_1 r \quad (4.4)$$

$$\dot{x} = (A - BK)x + Bk_1 r \quad (4.5)$$

olduğu görülmektedir.

4.2 İntegratör İçermeyen Tip 1 Servo Sistemin Genel Gösterimi (Tip 0 Plant)



Şekil 4.2 İntegratör içermeyen tip1 servo sistem

Sistemin kapalı çevrim eşitlikleri

$$\dot{X} = Ax + Bu \quad (4.6)$$

$$y = Cx \quad (4.7)$$

$$u = G\varepsilon - Kx \quad (4.8)$$

$$\dot{\varepsilon} = r - y = r - Cx \quad (4.9)$$

şeklinde olup, durum gösterimi

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (4.10)$$

halini alır; u ifadesi yerine yazılırsa

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A - BK & BG \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (4.11)$$

bulunur.

Asimptotik kararlı bir sistem dizayn ederek; $u(\infty)$, $X(\infty)$ ve $\varepsilon(\infty)$ un sabit değerlere yaklaştığı; kalıcı halde $\dot{\varepsilon}(\infty) = 0$ ve $y(\infty) = r$ olduğu ayrıca r referans girişinin birim basamak fonksiyon olduğu kabul edilmektedir.

$$\begin{bmatrix} \dot{x}(\infty) \\ \dot{\varepsilon}(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(\infty) \\ \varepsilon(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(\infty) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\infty) \quad (4.12)$$

$r(t) = r(\infty) = 1$ (birim basamak giriş)

$$\begin{bmatrix} \dot{x}(t) - \dot{x}(\infty) \\ \dot{\varepsilon}(t) - \dot{\varepsilon}(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) - x(\infty) \\ \varepsilon(t) - \varepsilon(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} [u(t) - u(\infty)] \quad (4.13)$$

$$x_e = x(t) - x(\infty) \quad (4.14)$$

$$\varepsilon_e = \varepsilon(t) - \varepsilon(\infty) \quad (4.15)$$

$$u_e = u(t) - u(\infty) \quad (4.16)$$

olarak tanımlanırsa;

$$\begin{bmatrix} \dot{x}_e \\ \dot{\varepsilon}_e \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_e \\ \varepsilon_e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e \quad (4.17)$$

$u_e(t) = G\varepsilon_e - Kx_e$ olduğu göz önüne alınır ve $(n+1)$. dereceden aşağıdaki gibi yeni bir hata vektörü tanımlanırsa,

$$e = \begin{bmatrix} x_e \\ \varepsilon_e \end{bmatrix} = (n+1)\text{vektör} \quad (4.18)$$

denklem:

$$\dot{e} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} e + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e = \hat{A}e + \hat{B}u_e \quad (4.19)$$

Buradan:

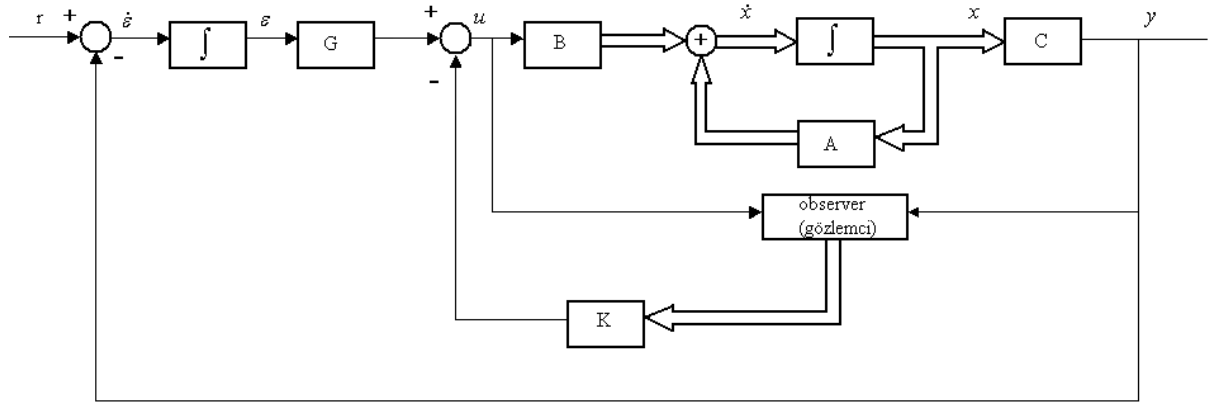
$$u_e = -\hat{K}e \quad \text{ve} \quad \hat{K} = [K \quad -G] \quad (4.20)$$

bulunmaktadır.

$\dot{e} = \hat{A}e + \hat{B}u_e$ ifadesi tamamen durum kontroledilebilir ise \hat{K} matrisi kutup yerleşimi; lqr metotları ile tayin edilebilir.

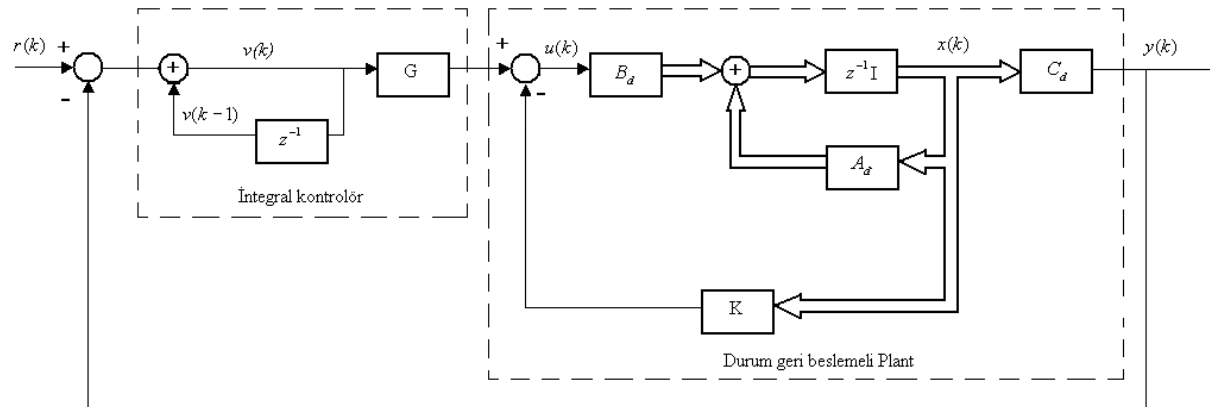
Eğer tüm durum değişkenleri direkt olarak ölçülemiyorsa (çoğunlukla uygulamalarda

ölçülememektedir); durum gözlemcisi (observer) kullanılması gereklidir.



Şekil 4.3 Durum gözlemcili integratör içermeyen tip1 servo sistem

4.3 Ayırık Zamanlı İntegratör İçermeyen Tip 1 Servo Sistemin Genel Gösterimi



Şekil 4.4 Ayırık zamanlı integratör içermeyen tip1 servo sistem

Sistemin kapalı çevrim eşitlikleri

$$x(k+1) = A_d x(k) + B_d u(k) \quad (4.21)$$

$$y(k) = C_d x(k) \quad (4.22)$$

$$v(k) = v(k-1) + r(k) - y(k); \quad (4.23)$$

integratör eşitliği

$$u(k) = G v(k) - K x(k) \quad (4.24)$$

şeklinde olup, tüm kapalı çevrim sistemin durum ve çıkış eşitliği

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} A_d - B_d K & B_d G \\ -C_d A_d + C_d B_d K & 1 - C_d B_d G \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1) \quad (4.25)$$

$$y(k) = \begin{bmatrix} C_d & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (4.26)$$

halini alır; basamak giriş için $r(k)=r$,

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} A_d - B_d K & B_d G \\ -C_d A_d + C_d B_d K & 1 - C_d B_d G \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \quad (4.27)$$

bulunur.

Basamak giriş için; k sonsuza yaklaştıkça; $u(k)$ $x(k)$ $v(k)$; $u(\infty)$, $X(\infty)$ ve $v(\infty)$ un sabit değerlere yaklaştığı; ayrıca r referans girişinin birim basamak fonksiyon olduğu kabul edilmektedir.

$$v(\infty) = v(\infty) + r(\infty) - y(\infty) \quad (4.28)$$

veya

$$y(\infty) = r(\infty) = r \quad (4.29)$$

Kalıcı hal hatası bulunmamaktadır. K sonsuzda iken denklem:

$$\begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} A_d - B_d K & B_d G \\ -C_d A_d + C_d B_d K & 1 - C_d B_d G \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \quad (4.30)$$

$$x_e(k) = x(k) - x(\infty) \quad (4.31)$$

$$v_e(k) = v(k) - v(\infty) \quad (4.32)$$

olarak tanımlanırsa;

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} A_d - B_d K & B_d G \\ -C_d A_d + C_d B_d K & 1 - C_d B_d G \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (4.33)$$

düzenlenirse,

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ -C_d A_d & 1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} B_d \\ -C_d B_d \end{bmatrix} \begin{bmatrix} -K & G \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (4.34)$$

$$w(k) = \begin{bmatrix} -K & G \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (4.35)$$

tanımlanırsa;

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} A_d & 0 \\ -C_d A_d & 1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} B_d \\ -C_d B_d \end{bmatrix} w(k) \quad (4.36)$$

elde edilir.

Bu iki denklemden aynı şekilde

$$\varepsilon(k+1) = \hat{A}_d \varepsilon(k) + \hat{B}_d w(k) \quad ; \quad w(k) = -\hat{K} \varepsilon(k) \quad (4.37)$$

bulunur.

Burada

$$\varepsilon = \begin{bmatrix} x_e(k) \\ \varepsilon_e(k) \end{bmatrix}, \quad \hat{A}_d = \begin{bmatrix} A_d & 0 \\ -C_d A_d & 1 \end{bmatrix}, \quad \hat{B}_d = \begin{bmatrix} B_d \\ -C_d B_d \end{bmatrix} \quad (4.38)$$

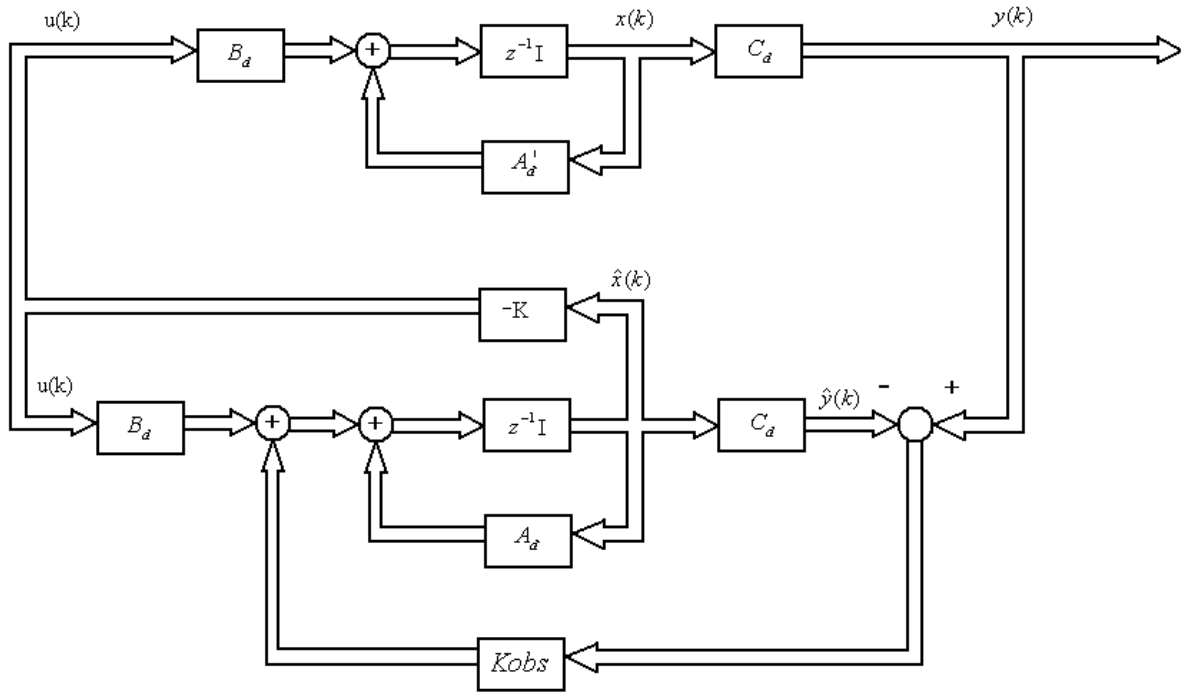
$$\text{ve } \hat{K} = \begin{bmatrix} K & -G \end{bmatrix} \quad (4.39)$$

bulunmaktadır.

\hat{K} matrisi daha sonra kutup yerleşimi metodu ile belir; Ackermann formülü kullanılarak K vektörü ve de G kazancı hesaplanır. (Sistemin tamamen durum kontrol edilebilir olması şarttır). (Ogata 1994)

Eğer tüm durum değişkenleri direkt olarak ölçülemiyorsa (çoğunlukla uygulamalarda ölçülememektedir); durum gözlemcisi (observer) kullanılması gereklidir.

Aşağıdaki şekilde ayrık zamanlı bir sistemde gözlemci (observer) dizaynı gösterilmiştir.



Şekil 4.5 Durum gözlemcili tip1 servo sistem genel gösterimi

5. DURUM GERİ BESLEME KONTROLÜ

5.1 Kutup Yerleşim Metodu (Pole Placement)

Tüm durum değişkenlerinin ölçülebilir ve de geribeslemeye uygun olduklarını kabul edelim. Eğer sistem tamamen durum kontroledilebilir ise; kapalı çevrim sistemin kutupları, istenilen herhangi keyfi pozisyona uygun geribesleme kazanç matrisi yardımı ile; durum geribesleme ile yerleştirilebilir. (Ogata 1995)

Bu teknik; istenilen kapalı çevrim kutuplarının, geçici hal cevabı ve/veya frekans cevabı (sönüm oranı, hız, bandgenişliği) ve de kalıcı hal gereksinimlerine göre elde edilmesi ile başlar.

Örnekleme periyodu seçilirken, sistemin büyük kontrol sinyallerine gereksinim duymamasına dikkat edilmelidir. Aksi halde sistemde doyum oluşabilir. Eğer sistemde doyum oluşur ise, sistem non-lineer hale gelir ve gösterilen dizayn metodu, sadece doğrusal zamanla değişmeyen (LTI) sistemlerde uygulanabildiğinden; kontrol sağlayamaz hale gelir.

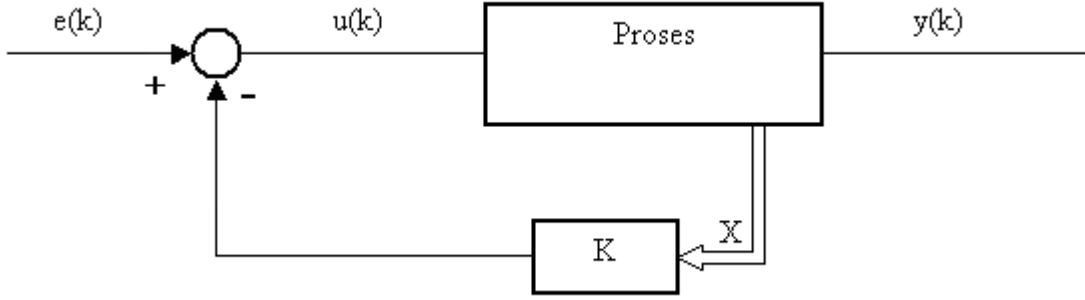
İstenilen kapalı çevrim kutup lokasyonları belirlendikten sonra, durum geribesleme için uygun kazanç matrisi seçilerek; sistem, istenilen noktalarda kapalı çevrim kutuplarına sahip olmasına zorlanabilir.

Bu yöntemde sistemin derecesi, kutup-sıfır sadeleştirilmesi olmayacak şekilde bir veya 2 derece artırılabilir.

Kapalı çevrim kutuplarını istenilen lokasyonlara yerleştirme problemine *kutup yerleşimi problemi* adı verilir.

Proses in dinamik davranışını, durum vektöründen hesaplanan bir geribesleme ile değiştirmek olarak tanımlanabilir.

Ayrık zamanlı bir proses için geribesleme ile kutup yerleşim metodu şekil 5.1 de gösterilmiştir.



Şekil 5.1 Geribesleme ile kutup yerleşim metodu

Proses veya sistemin A_d , B_d ve C_d matrisleri biliniyor.

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_d u(k) \quad y(k) = C_d x(k) \quad (5.1)$$

$$u(k) = e(k) - Kx(k) \quad (5.2)$$

buradan

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_d u(k) = (\mathbf{A}_d - \mathbf{K}B_d)x(k) + B_d e(k) \quad (5.3)$$

$$x(k+1) = [\mathbf{A}_d - \mathbf{K}B_d]x(k) + \mathbf{B}_d e(k) \quad (5.4)$$

$$y(k) = C_d x(k) \quad (5.5)$$

bulunur.

Prosesin dinamiklerini düzeltten durum geribeslemesi; karakteristik denklemdeki $\mathbf{A}_d - \mathbf{K}B_d$ matrisinin fonksiyonu haline sokulmuştur.

Stabilite ve geçici hal cevabı karakteristikleri, $\mathbf{A}_d - \mathbf{K}B_d$ matrisinin özdeğer (eigen values) yada diğer adıyla karakteristik denklemin kökleri tarafından belirlenir. Eğer K matrisi düzgün seçilirse; $x(0) \neq 0$ olmak koşulu ile, $\mathbf{A}_d - \mathbf{K}B_d$ matrisi asimptotik kararlı hale getirilebilir.

$\mathbf{A}_d - B_d K$ matrisinin özdeğerleri regülatör kutupları (regulator poles) adını alır.

Eğer regülatör kutupları, s düzleminin sol tarafına yerleştirilirse (asimptotik kararlı), t sonsuza gittiğinde $x(t)$ sifıra yaklaşır. (Ogata 2002)

K vektörünün elemanlarını hesaplamak için, düzeltilmiş sistemin α_i kutuplarını seçmek gereklidir.

$$\det[A_d - KB_d] = z^n + \alpha_1 z^{n-1} + \dots + \alpha_n \quad (5.6)$$

Bu hesaplamayı kolaylaştırmak için A_d ve B_d yi kontroledilebilir kanonik hale getirmemiz gereklidir. Kanonik halde matrisler aşağıdaki hali almaktadır:

$$\det[zI - A_d] = z^n - a_1 z^{n-1} + \dots + a_{n-1} z + a_n$$

$$A_{dc} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \quad B_{dc} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (5.7)$$

5.1.1 Yöntem 1 - Kontroledilebilir Kanonik Forma Dönüşüme Dayalı Metot

Aşağıda tanımlana şekilde bir T transformasyon matrisi tanımlanır.

$$T^{n-1} = A_d T^n + a_{n-1} B_d \quad (5.8)$$

$$T = M_{contd} W_c \quad (5.9)$$

$$T^n = B_d \quad (5.10)$$

Burada M_{contd} denklem(3.6) bahsedilen kontroledilebilirlik matrisidir. Ve W_c aşağıdaki şekilde verilmektedir. (Ogata 1994)

$$W_c = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} \quad (5.11)$$

T transformasyon matrisi bulunduktan sonra

$$A_{dc} = T^{-1} A_d T \quad (5.12)$$

$$B_{dc} = T^{-1} B_d \quad (5.13)$$

dönüşümlerini uygulanarak sistemin kanonik haldeki tanımı elde edilir.

$K = [k_n \quad k_{n-1} \quad \cdots \quad \cdots \quad k_1]$ olduğundan

$$[A_{dc} - B_{dc}K] = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -(a_n + k_n) & -(a_{n-1} + k_{n-1}) & \cdots & -(a_2 + k_2) & -(a_1 + k_1) \end{bmatrix} \quad (5.14)$$

karakteristik denklemi

$$\phi_c = \det[zI - (A_{dc} - B_{dc}K)] = z^n + (a_1 + k_1)z^{n-1} + \cdots + (a_{n-1} + k_{n-1})z + (a_1 + k_1) \quad (5.15)$$

elde edilir.

$$\alpha_i = a_i + k_i \quad (5.16)$$

olarak istenilen karakteristik denklemin katsayıları verilsin,

$$K_c = KT^{-1} = [k_n \quad k_{n-1} \quad \cdots \quad k_1]T^{-1} \quad (5.17)$$

Buradan

$$K_c = KT^{-1} = [\alpha_n - a_n \quad \alpha_{n-1} - a_{n-1} \quad \cdots \quad \alpha_1 - a_1]T^{-1} \quad (5.18)$$

elde edilir.

Yapılan simülasyonda yöntem1 den faydalanılmıştır.

5.1.2 Yöntem 2 - Ackermann Formülü ile K Değeri Hesaplanması

Elde edilmek istenen karakteristik denklem

$$|pI - A + BK| = p^n + \alpha_1 p^{n-1} + \cdots + \alpha_{n-1} p + \alpha_n = 0 \quad (5.19)$$

olarak verilsin.

$$\phi(A) = A^n + \alpha_1 A^{n-1} + \cdots + \alpha_{n-1} A + \alpha_n I \quad (5.20)$$

$$\hat{A} = A - BK \quad (5.21)$$

Cayley-Hamilton teoreminden

$$\phi(\hat{A}) = \hat{A}^n + \alpha_1 \hat{A}^{n-1} + \dots + \alpha_{n-1} \hat{A} + \alpha_n I = 0 \quad (5.22)$$

olduğu bilinmektedir.

Bu teorem Ackermann formülünü elde etmek için kullanılmaktadır. İşlem kolaylığı açısından $n=2$ olarak alınır (n 'in tüm diğer pozitif tamsayı değerleri için aşağıdaki işlem kolaylıkla uzatılabilir).

Buradan,

$$I = I \quad (5.23)$$

$$\hat{A} = A - BK \quad (5.24)$$

$$\hat{A}^2 = (A - BK)^2 = A^2 - ABK - BKA + (BK)^2 = A^2 - ABK - BK\hat{A} \quad (5.25)$$

Caylay-Hamilton formülünde (5.22) yerine yerleştirirken: ($\alpha_0 = I$)

$$\phi(\hat{A}) = \hat{A}^2 + \alpha_1 \hat{A} + \alpha_2 I = 0 \quad (5.26)$$

$$\phi(\hat{A}) = \alpha_2 I + \alpha_1 (A - BK) + A^2 - ABK - BK\hat{A} = 0 \quad (5.27)$$

$$\phi(\hat{A}) = (\alpha_2 I + \alpha_1 A + A^2) - ABK - BK\hat{A} = 0 \quad (5.28)$$

şekil(5.20) göz önünde bulundurularak:

$$\phi(\hat{A}) = \phi(A) - \alpha_1 (BK) - ABK - BK\hat{A} = 0 \quad (5.29)$$

$$\phi(A) = B(\alpha_1 K + K\hat{A}) + ABK \quad (5.30)$$

Matrisel formda

$$\phi(A) = \begin{bmatrix} B & AB \end{bmatrix} \begin{bmatrix} \alpha_1 K + K\hat{A} \\ K \end{bmatrix} \quad (5.31)$$

yazılabilir.

K dışarı alınır

$$K = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} B & AB \end{bmatrix}^{-1} \phi(A) \quad (5.32)$$

elde edilir.

$[B \ AB]$ matrisi kontroledilebilirlik matrisidir (M_{cont}). Yerine yazıldıktan sonra n . dereceden genelleştirilmiş Ackermann formülü:

$$K = [0 \ 0 \ \dots \ 0 \ 1] M_{cont}^{-1} \phi(A) \quad (5.33)$$

halini alır.

Matlab ile K hesaplanması Acker, Place, Polyvalm komutları ile gerçekleştirilebilmektedir.

5.2 Dinamik Ayırık Zamanlı Bir Sistemin Durum Yeniden Yapılandırılması-Gözlemci Dizaynı (Observer –Estimator Design)

Kutup yerleşim metodunda tüm durum değişkenlerinin geribeslemeye uygun olduğunu kabul etmiştik; fakat uygulamalarda tüm durum değişkenleri geribeslemeye uygun olmayabilir. O halde ölçülemeyen bu durum değişkenlerini tahmin etmemiz gerekmektedir.

Bu tahmin işlemine genel olarak *gözleme (observation)* denilmektedir.

Bir kontrol sistemindeki, çıkış ve kontrol değişkenlerinin ölçülmesine dayalı olarak gerçekleşen durum değişkeni tahminlerine *durum gözlemcisi (observer)* yada başka bir deyişle durum tahminedicisi (*estimator*) alt sistemleri denilmektedir.

Eğer bir proses gözlenebiliyor ise; A_d , B_d matrisleri bilgileri ile $u(t)$ ve $y(t)$ ölçümleri sayesinde durumu yeniden yapılandırabiliriz.

Bu koşulun sağlanması durum gözlemci dizaynı için gerek ve yeter koşuldur.

Bu yeniden yapılandırmanın avantajı;

- Bazı ölçülmesi zor, çok pahalı veyahut ölçülemeyen durum komponentlerinin elde edilmesine olanak sağlanması (çoğu sistemde sadece giriş ve çıkış değişkenleri ölçülebilmekte fakat tüm durum değişkenleri direkt olarak ölçülememektedir).
- Durum komponentlerini ölçmek için gerekli olan sensörlerin, çok güvenilir olmaması, veya gürültülü bir sinyal üretebilecekleridir.

Örnek vermek gerekirse; üzerinde çalışılan sistem 20. dereceden durum-uzay modeline sahip bir sistem ise, sisteme her biri bir durum değişkenini ölçen 20 adet sensör eklenmesi gerekmektedir. 20 adet sensör kullanmak hem masraflıdır hem de bu kadar sensörü yerleştirmek fiziksel zorluklara sebebiyet verebilmektedir. Bu ve benzeri zorlukların

üstesinden gelebilmek için tahminedicilerden yararlanılmaktadır.

Durum yeniden yapılandırıcısı veya gözlemcilerini 2 gruba ayırabiliriz:

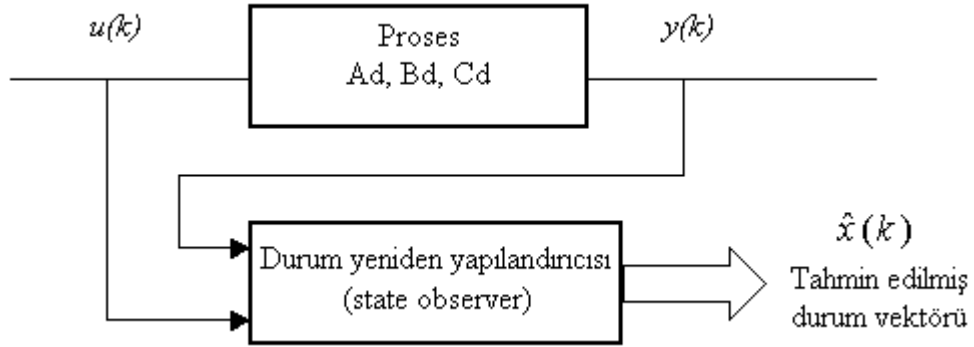
- deterministik proses gözlemcisi veya Luenberger gözlemcisi
- stochastic proses gözlemcisi veya Kalman tahminedicisi.

(Maybeck 1979)

5.2.1 Deterministik Prosesin Kapalı Çevrim Tahmini

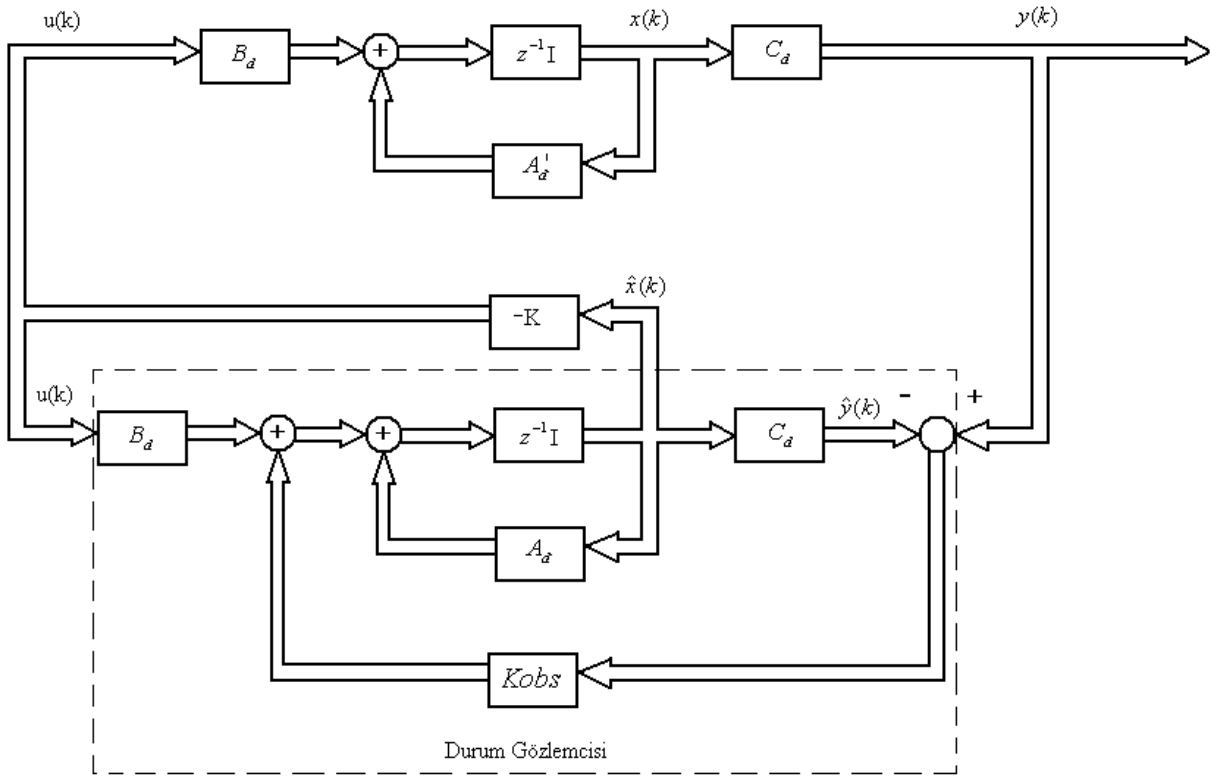
Proses veya sistemin A_d , B_d ve C_d matrisleri biliniyor.

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_d u(k) \quad y(k) = \mathbf{C}_d x(k) \quad (5.34)$$



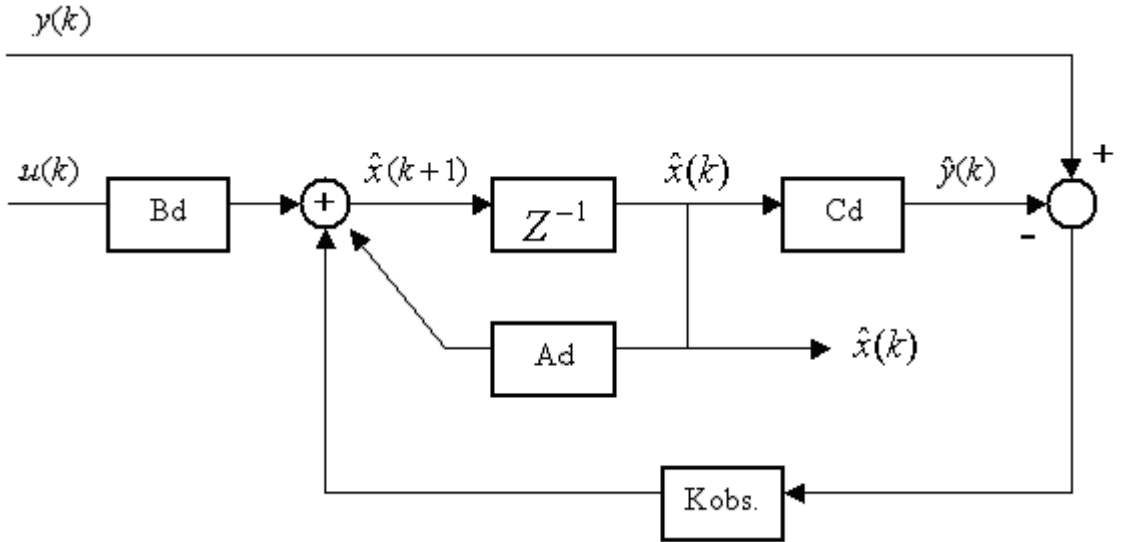
Şekil 5.2 Durum yeniden yapılandırıcısı (gözlemcisi) nin şematik gösterimi

Genel gösterim:



Şekil 5.3 Sistemin durum yeniden yapılandırıcısı ile beraber genel gösterimi

Sistemin daha kolay incelenebilmesi açısından grafik aşağıdaki forma sokulabilir.



Şekil 5.4 Durum yeniden yapılandırıcısı

$\hat{x}(k+1)$ i çıkış olarak kullandığımız taktirde, hem yeniden yapılandırıcı hem de tahmin edici olarak faydalanabiliriz.

Tahmin edici Eşitliği

$$\hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + K_{obs} (y(k) - \hat{y}(k)) \quad (5.35)$$

$$\hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + K_{obs} (y(k) - C_d \hat{x}(k)) \quad (5.36)$$

düzenlenirse,

$$\hat{x}(k+1) = (A_d - K_{obs} \cdot C_d) \hat{x}(k) + B_d u(k) + K_{obs} \cdot y(k) \quad (5.37)$$

elde edilir.

Tahmin $\hat{x}(k+1)$, $y(k)$ ölçümünden bir örnekleme periyodu önde olduğundan bu matrisle *tahmin edici gözlemci (prediction observer)* denir. $A_d - K_{obs} \cdot C_d$ matrisinin özdeğerlerine genel olarak *gözlemci kutupları (observer poles)* denilmektedir.

K_{obs} matrisi kazanç matrisidir.

Sistemi stabil kılarken aynı zamanda hızlı tahmin yapabilecek şekilde seçilmesine dikkat edilmelidir.

5.2.2 Stabilite Kriteri

$$e(k) = \hat{x}(k) - x(k) \quad (5.38)$$

hata vektörü olarak tanımlanırsa;

$$e(k+1) = (A_d - K_{obs} \cdot C_d) \hat{x}(k) + B_d u(k) + K_{obs} \cdot y(k) - A_d x(k) - B_d u(k) \quad (5.39)$$

$$e(k+1) = (A_d - K_{obs} \cdot C_d) (\hat{x}(k) - x(k)) \quad (5.40)$$

Buradan

$$e(k+1) = (A_d - K_{obs} \cdot C_d) e(k) \quad (5.41)$$

elde edilir.

Aşağıdaki karakteristik denklemin çözümü hata kutuplarını vermektedir.

$$\det[zI - A_d - K_{obs} \cdot C_d] = 0 \quad (5.42)$$

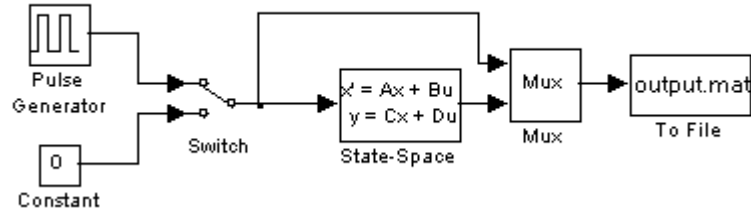
Hata sinyalinin dinamik davranışı $A_d - K_{obs} \cdot C_d$ matrisinin özdeğerleri ile elde edilmektedir.

Eğer $A_d - K_{obs} \cdot C_d$ matrisi kararlı ise; hata matrisi herhangi bir $e(0)$ hata başlangıç koşulunu sıfıra götürecektir. Buda demek oluyor ki $\hat{x}(0) = x(0)$ başlangıç koşulları ne olursa olsun $\hat{x}(k)$; $x(k)$ ile bir noktada birleşecektir.

Hata kutuplarını, orijine olabildiğince yakın yerleştirilmeye çalışılmalıdır.

K_{obs} sabit kazanç matrisinin seçimi hata dinamiklerini karakterize etmemizi sağlar; ancak bu gözlemci optimal değildir. Optimal gözlemci olabilmesi için $K_{obs} = K_{obs}(t)$ olması yani zamanla değişmesi gerekmektedir. Tez çalışmasında kullanılan optimal kontrolcü olan LQR metodunda P olarak tanımlanan kalıcı hal matrisi sabitlenene kadar iterasyona sokularak son stabil değeri bulup buradan K_{obs} kazancı elde edilmiştir.

6. TEPE VİNCİ SİSTEMİNİN DURUM-UZAY SİMULİNK MODELİ

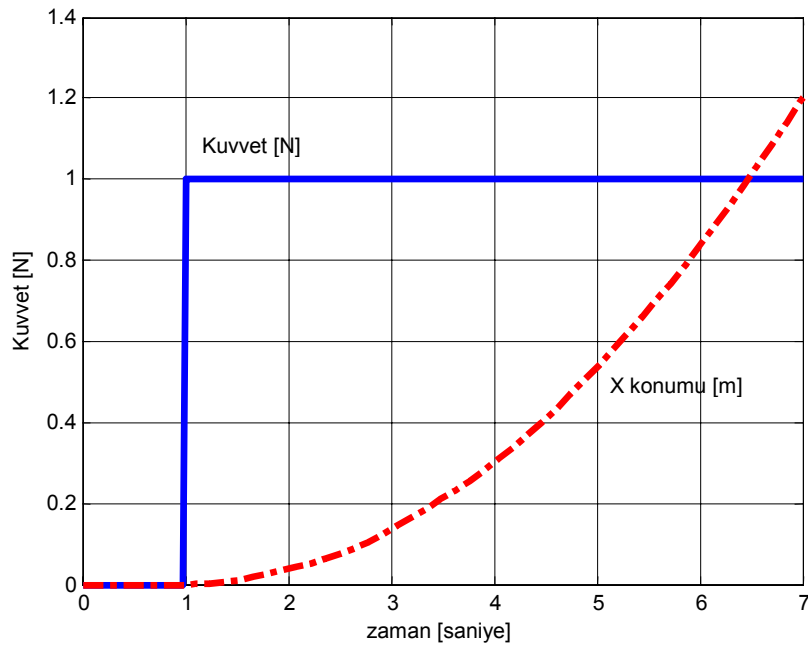


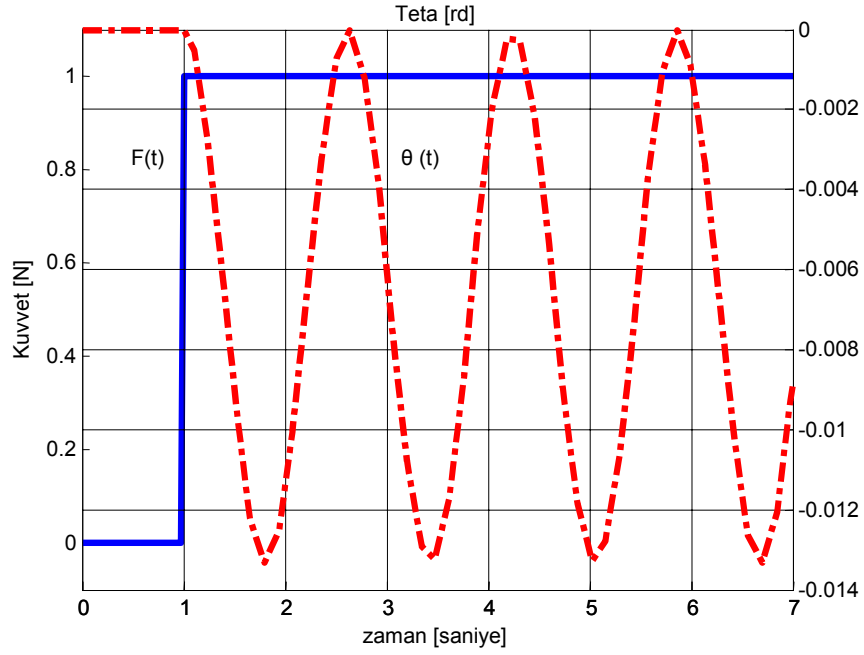
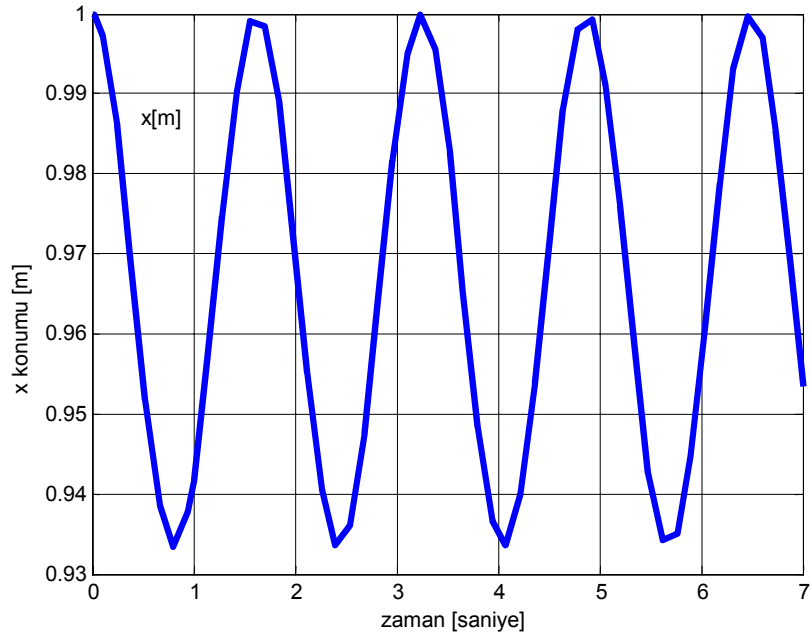
Şekil 6.1 Vinç sisteminin Simulink modeli

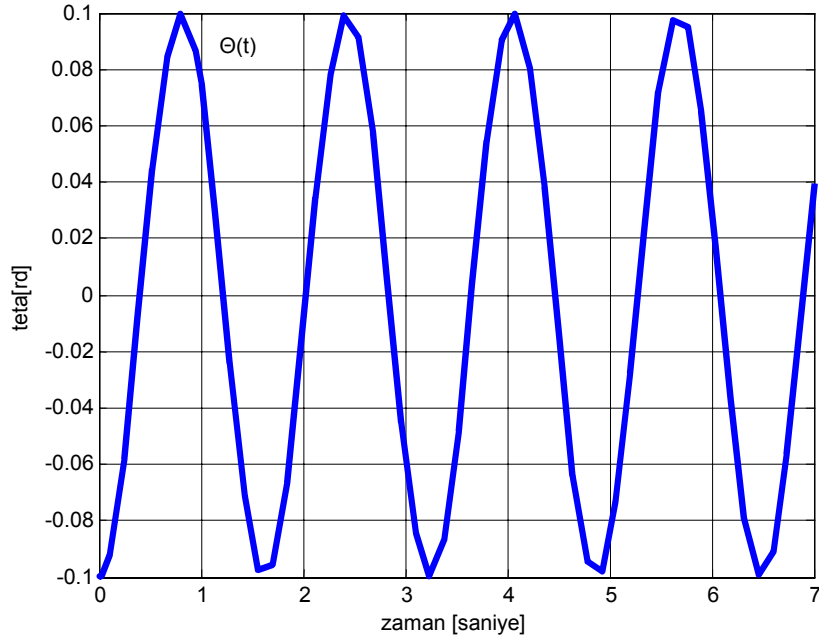
Yukarıda görülen Matlab-Simulink simülasyonunda; sistem, durum-uzay olarak modellenmiş ve sistemin açık çevrim cevabını alabilmek için basamak giriş uygulanmıştır. Durum-uzay modelinde gerekli olan A , B , C , D katsayıları m.file ile girilmiştir. output.mat dosyası ise çıktıların saklandığı bir dosya uzantısı olup; matlab de komut satırından okutularak çıktıları almamızda kolaylık sağlamaktadır.

6.1 Sistemin Açık Çevrim Cevabı

Şekil(6.2) ve şekil(6.3) sistemin basamak cevabını göstermektedir. Ayrıca sistemin karakteristik özelliklerinin daha iyi anlaşılabilmesi için; sisteme bir dış kuvvet etki etmeden sadece başlangıç koşulları ile oluşan davranışı incelenmiştir. Bu, x konumu için $1m$; sarkaç açısı için $-0.1rd$ değerleri girilerek sağlanmıştır. Şekil(6.4) ve şekil(6.5) görüldüğü gibi simülasyonda sürtünme kayıpları ihmal edildiğinden; arabanın yer değiştirmesinin devamlı olarak arttığı ve de sarkaç salınım hareketinde sönüm olmadığı gözlemlenmiştir.



Şekil 6.2 Sistemin basamak cevabı – x konumuŞekil 6.3 Sistemin basamak cevabı – $teta$ açısıŞekil 6.4 Sistemin başlangıç koşulu cevabı – x konumu



Şekil 6.5 Sistemin başlangıç koşulu cevabı – teta açısı

teta açısının -0.1 radyan başlangıç koşulunu alarak sarkacın osilasyona girmesine yol açtığını bununla beraber başlangıç koşulu 1m olan aracın x pozisyonunu da etkileyerek osilasyona soktuğu gözlemlenmiştir.

Sistemdeki osilasyonun $1,6$ saniyelik periyotlarla tekrarlandığı ölçülmüştür.

6.2 Ayırık Zamanlı (Discrete) Durum-Uzay Modeli

Sistemin zaman sabiti $\tau = 1,6\text{s}$ civarında olup; simülasyon için örnekleme periyodunu $0,1\text{ s}$ olarak alırsak $\pi/16$ halini almış olacaktır.

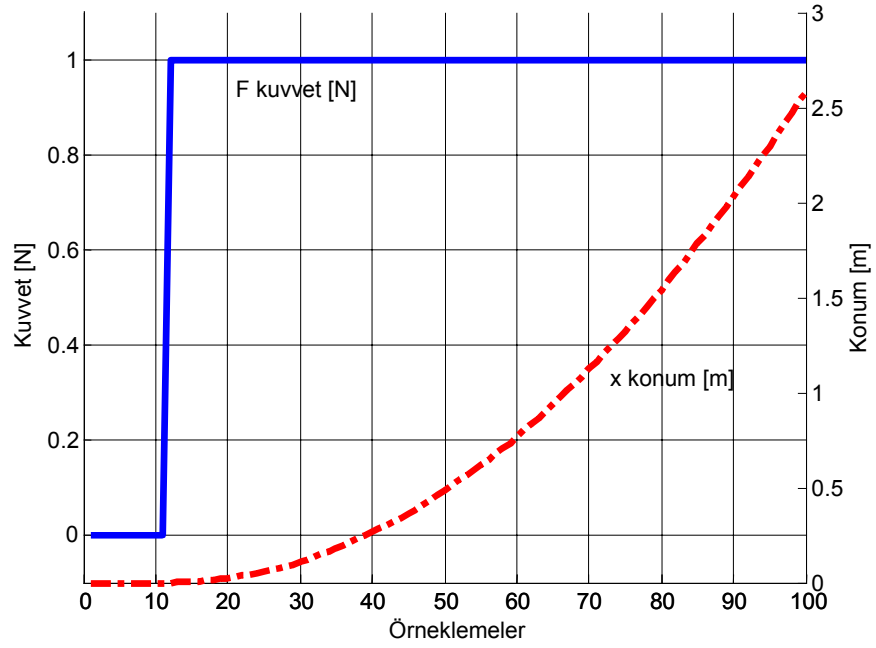
Önceden elde edilmiş olan A , B , C , D matrisleri Matlab deki `c2dt` komutu yardımı ile ayırık zamanlı hale getirilir. Bu komut sürekli zaman durum-uzay modeli ayırık zamana $f_s=10\text{Hz}$ ile çevirmektedir. Görüldüğü gibi sistemin karakteristiklerinde bir değişiklik olmamıştır.

Matlab kodu

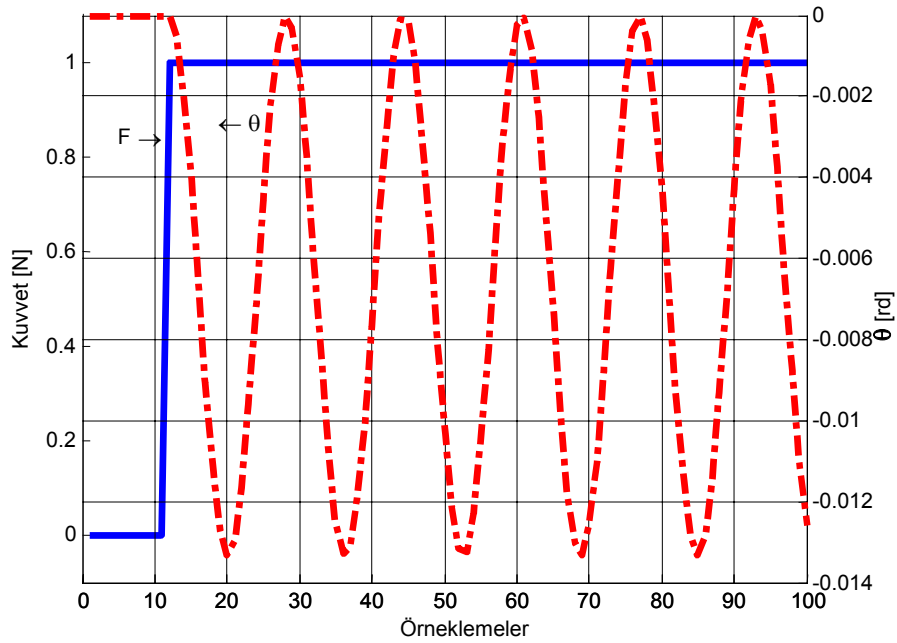
$T_e=0.1$

`[Ad,Bd,Cd,Dd]=c2dt(A,B,C,Te,0)`

6.2.1 Ayrık Zamanda Açık Çevrim Cevabı



Şekil 6.6 Ayrık zamanlı sistemin basamak cevabı – x konumu



Şekil 6.7 Ayrık zamanlı sistemin basamak cevabı – $teta$ açısı

7. GERİBESLEME KONTROL SİSTEMİ TASARLANMASI

7.1 Amaç

Arabayı istenilen x pozisyonuna getirmeye çalışırken, sarkacın minimum osilasyon yapmasını sağlamak ve bu osilasyonu olabildiğince hızlı şekilde sonlandırabilmektir. Bunun için önce karta uygulanan F kuvveti hesaplanmış; daha sonra sisteme giriş olarak konum bilgisi verilerek istenilen konuma gelmesi sağlanmıştır.

7.2 Geribesleme Kontrol Sistemleri

Bu bahsedilen amacı gerçekleştirmek üzere 2 adet kontrol metodu kullanılmıştır. İlk kontrolcü kutup yerleşim metoduna dayalı tam durum geribesleme kontrolüdür. Bu yöntemde uygun geribesleme kazanç matrislerinin seçilmesi ile sistemin kapalı çevrim kutupları istenen noktalara yerleştirilmiştir. İkinci kontrolcü LQR (Linear Quadratic Regulator) metodu olup indeks matrisinin minimize edilmesi ile elde edilen matris katsayılarının sisteme geribeslenmesi şeklinde uygulanmıştır.

7.3 Sistemin Luenberger Durum Gözlemcisi Tasarımı (Observer Design)

Öncelikle sistemin geribesleme kontrolünü elde edebilmek için sistemin tüm durum değişkenlerini bilmemiz gerekmektedir. Bu sebeple sistemin durum gözlemcisi tasarlanmıştır.

Ölçemediğimiz durum değişkenleri için bütün sistemin durum tahmin edicisinden faydalanarak durum vektörü elde edilmeye çalışılmıştır.

Bunun için gözlemlenebilirlik matrisinin rankına bakılması gerekmektedir. Rank, sistemin derecesine eşit ise sistem gözlenebilirdir ve giriş-çıkış değişkenleri incelenerek sistemin tüm durum değişkenleri tahmin edilebilir sistem yeniden yapılandırılabilir.

Sistemimizde:

Daha önceden bahsedildiği üzere gözlenebilirlik matrisi (3.17)

Ayrık zamanda $N_{obsd} = [C_d : C_d A_d : \dots : C_d A_d^{n-1}]^T$ şeklindedir.

Bu eşitlik matlab de:

% durum uzay denklemleri katsayıları

M=10; m=5; l=1; g=10;


```

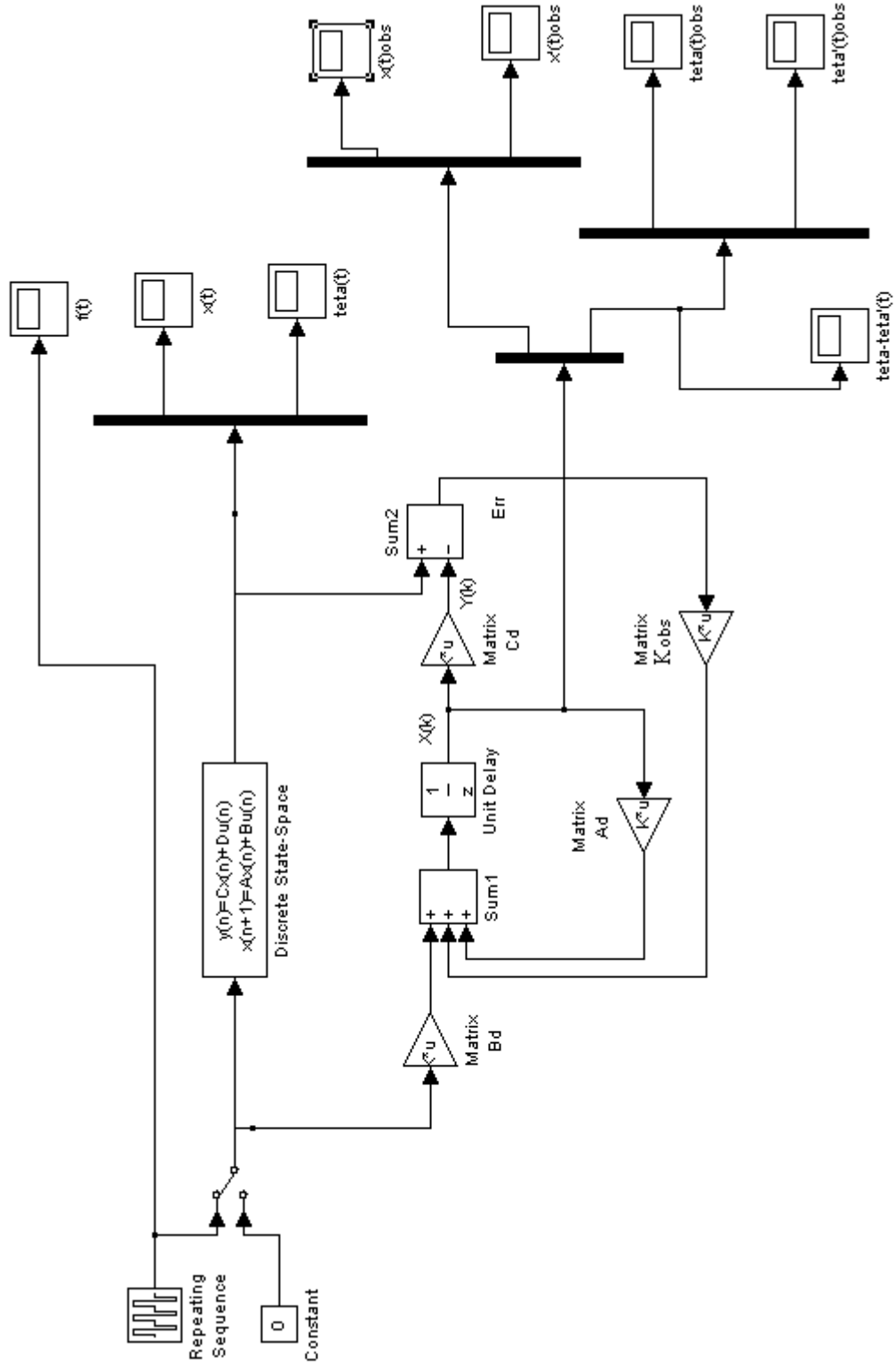
k1=m*g/M;
k2=-(M+m)*g/(M*I);
k3=1/M;
k4=-1/(M*I);
A=[0 1 0 0 ;0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0 ];
B=[0; k3; 0; k4];
C=[1 0 0 0; 0 0 1 0];
% ayrıık zamanlı durum uzay modeli
Te=0.1;
[Ad,Bd,Cd,Dd]=c2dt(A,B,C,Te,0);
% gözlenebilirlik (Observability) matrisi hesaplanması
Nobsd=obsv(Ad,Cd);
% çıktı alınması
printmat(Nobsd,'gözlenebilirlik matrisi');
% rank hesaplanması
disp(['rank= '(rank(Nobsd))]);
if rank(Nobsd)==4 disp('sistem gözlenebilirdir')
end;

```

$$\text{gözlenebilirlik matrisi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0.1 & 0.02469 & 0.00083 \\ 0 & 0 & 0.92593 & 0.09752 \\ 1 & 0.2 & 0.09510 & 0.00647 \\ 0 & 0 & 0.71470 & 0.18059 \\ 1 & 0.3 & 0.20080 & 0.02103 \\ 0 & 0 & 0.39760 & 0.23691 \end{bmatrix} \quad (7.1)$$

>> rank=4 sistem gözlenebilirdir.

Sonuçları elde edilir.



Şekil 7.1 Araba ve sarkaç (vinç) sisteminin (estimator) tahmin edicili Simulink modeli

Sistemin gözlemci dizaynı simulink modeli olarak yukarıda gösterilmiştir. Durum tahmin edicisinin davranışını düzeltecek olan kazanç matrisi K_{obs} olarak isimlendirilmiştir. Sistemin zaman cevabına göre, durum vektöründe çabuk yaklaşım yapabilecek şekilde seçilmelidir.

Bununla beraber, tahminedicinin stabilitesi sağlanmak zorundadır. Buda Z domenindeki $[A_d - K_{obs}.C_d]$ Matrisinin köklerinin birim çemberin içinde olmasını gerektirmektedir.

Ayrıca gözlemci; düzeltilmiş sisteme de aynı hızda cevap verebilmelidir. Buda özdeğerlerinin düzeltilmiş sistemden daha küçük olmasını gerektirmektedir. İleride anlatılacak olan kutup yerleşimi metodunda kutup noktaları seçilirken bu kritere dikkat edilerek seçim yapılması gerekmektedir.

Prosesin karakteristik denklemi

$$\phi = \det[A_d - B_d K] = z^n + a_1.z^{n-1} + \dots + a_{n-1}z + a_n \quad (7.2)$$

Varolan sistemde karakteristik denklem:

$$phi = \det[zI - A_d] = \phi = z^4 - 3.85z^3 + 5.7z^2 - 3.85z + 1 \quad (7.3)$$

şeklinde olup

Sistemin derecesi $n=4$ dür.

Bu denklem ve denklemin kökleri aşağıda verile Matlab komut satırları ile bulunmaktadır:

% prosesin karakteristik denkleminin elde edilmesi

phi=poly(Ad);

%karaktersitik denklemin kökleri

roots(phi);

kökleri aşağıda gösterilmektedir.

$$z1=0.9259 + 0.3777i \quad (7.4)$$

$$z2=0.9259 - 0.3777i \quad (7.5)$$

$$z3=1.0000 \quad (7.6)$$

$$z4=1.0000 \quad (7.7)$$

Durum tahmincisi (observer) için; sistemin doğal frekansı $\omega_n = 2\pi$ rd/s ve sönüm oranı $\zeta = 0.707$ (Matlab komut satırında *dr* değişkeni ile gösterilmiştir) olarak seçilerek 2. dereceden bir sistem oluşturulmuştur. Dijital sistem için örnekleme periyodu $T_e = 0.1$ saniye seçilmiştir.

$$\phi_r = z^2 + p_1 z + p_2 \quad (7.8)$$

s düzleminde sistemi yeniden ele alırsak:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad ; \quad \zeta = 0.707 \quad ; \quad \omega_c = 2\pi = 6.28 \quad (7.9)$$

(Vaccaro 1995)

Buradan karakteristik denklem:

$$s^2 + 2 * 0.707 * 6.28 + 6.28^2 = 0 \quad (7.10)$$

kökleri Matlabde; roots komutu ile:

$$s_1 = -4.4400 + 4.4413i \quad (7.11)$$

$$s_2 = -4.4400 - 4.4413i \quad (7.12)$$

olarak elde edilmiştir.

İstenen kapalı çevrim S domeni kutupları, Z domenine ZOH kutup düzenleyici formülü olan $z_i = e^{s_i T_e}$ kullanarak geçirdiğimiz taktirde;

$$z_1 = e^{0.1s_1} = 0.5792 + 0.2756i \quad (7.13)$$

$$z_2 = e^{0.1s_2} = 0.5792 - 0.2756i \quad (7.14)$$

Son olarak, bulunan z düzlemi kutupları $\phi_r(z)$ polinomunu elde etmek için kullanılırsa

$$\phi_r(z) = (z - z_1)(z - z_2) = z^2 - z(z_1 + z_2) + z_1 z_2 \quad (7.15)$$

$$p = \phi_r = z^2 - 1.1584z + 0.4112 \quad (7.16)$$

elde edilmiş olur.

Bu denklem ayrıca aşağıdaki p_1 ve p_2 denklemleri yerlerine yazılarak da elde edilebilir:

$$p_1 = -2e^{-\zeta\omega_n T_e} \cdot \cos(\omega_n T_e \sqrt{1-\zeta^2}) \quad (7.17)$$

$$p_2 = e^{-2\zeta\omega_n T_e} \quad (7.18)$$

Bu elde edilmek istenen tahmincinin karakteristik denklemi aşağıdaki Matlab komut satırları ile bulunabilmektedir:

%durum tahmin edicisinin karakteristik denklemi

%sönüm oranı

dr=sqrt(2)/2

% doğal frekans

wn=2*pi

%denklem

p1 = -2*exp(-dr*wn*Te)*cos(wn*Te*sqrt(1-dr^2));

p2 = exp(-2*dr*wn*Te);

p = [0 0 1 p1 p2];

Şimdi yapılması gereken; sistemin M_{contd} kontroledilebilirlik matrisini elde edilmesi ve rankının sistemin n değerine eşit olup olmadığının kontrol edilmesidir. Eğer rank eşit değilse sistem bu metot ile kontrol edilemez.

Matlab ile kontroledilebilirlik denklemi ve rankı:

%kontrol edilebilirlik matrisi

Mcontd=ctrb(Ad,Bd)

%rank hesaplanması

rank=rank(Mcontd)

printmat(Mcontd,'Kontroledilebilirlik matrisi =')

printmat(rank,'rank=')

if rank(Mcontd)==4 disp('sistem kontroledilebilirdir; keyfi kutup yerleşimi yapılabilir')

else disp('sistem kontrol edilemez')

end

komut satırları ile hesaplanır.

İşlemler sonucu

$$\text{Kontroledilebilirlik matrisi} = \begin{bmatrix} 0.0005 & 0.0015 & 0.0024 & 0.0032 \\ 0.0099 & 0.0094 & 0.0085 & 0.0074 \\ -0.0005 & -0.0014 & -0.0021 & -0.0025 \\ -0.0098 & -0.0083 & -0.0056 & -0.0021 \end{bmatrix} \quad (7.19)$$

$rank=4$

sistem kontrol edilebilirdir; keyfi kutup yerleşimi yapılabilir.

Bundan sonra yapılması gereken; bölüm(5.1.1) de bahsedilen T transformasyon matrisinden (5.9) faydalanarak sistemin yenide yapılandırılmasıdır.

$$T = M_{contd} \cdot W_c \quad (7.20)$$

olduğundan; ve de orijinal sistemin karakteristik denklemi olan (7.3) eşitliği ve (7.2) eşitliklerinden faydalanarak

$n=4$;

$$a_0 = 1; \quad a_{n-3} = a_1 = -3.85 \quad a_{n-2} = a_2 = 5.7 \quad a_{n-1} = a_3 = -3.85 \quad a_n = a_4 = 1 \quad (7.21)$$

katsayıları W_c (5.11) matrisinde yerine yazılırsa:

$$W_c = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}; \quad W_c = \begin{bmatrix} -3.85 & 5.7 & -3.85 & 1 \\ 5.7 & -3.85 & 1 & 0 \\ -3.85 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (7.22)$$

bulunur.

$$T = \begin{bmatrix} 0.0005 & 0.0015 & 0.0024 & 0.0032 \\ 0.0099 & 0.0094 & 0.0085 & 0.0074 \\ -0.0005 & -0.0014 & -0.0021 & -0.0025 \\ -0.0098 & -0.0083 & -0.0056 & -0.0021 \end{bmatrix} \begin{bmatrix} -3.85 & 5.7 & -3.85 & 1 \\ 5.7 & -3.85 & 1 & 0 \\ -3.85 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (7.23)$$

Matlab kullanarak transformasyon matrisinin hesabı aşağıdaki komut satırları ile yapılabilmektedir:

%T transformasyon matrisi

n = 4;

T(:,4) = Bd;

for i = 1:n-1,

 T(:,n-i) = Ad*T(:,n-i+1)+phi(n-i+1)*Bd;

end

Buradan T transformasyon matrisi

$$T = \begin{bmatrix} 0.0005 & -0.0004 & -0.0004 & 0.0005 \\ -0.0099 & 0.0288 & -0.0288 & 0.0099 \\ -0.0005 & 0.0005 & 0.0005 & -0.0005 \\ 0.0098 & -0.0293 & 0.0293 & -0.0098 \end{bmatrix} \quad (7.24)$$

olarak bulunmuştur.

$K_c = KT^{-1}$ ifadesini uygulama sokarak K_{obs} değeri hesaplanırsa:

$$K_{obsc} = K_{obs}.T^{-1} \quad (7.25)$$

Şimdi K_{obs} matrisini oluşturarak; nihai K_{obsc} değerini elde etmeye çalışmamız gerekiyor.

oluşturulan sistem: $p = \phi_r = z^2 - 1.1584z + 0.4112$ (7.16) dan

$$\alpha_4 = 0.4112; \quad \alpha_3 = -1.1584; \quad \alpha_2 = 1; \quad \alpha_1 = 0; \quad \alpha_0 = 0 \quad (7.26)$$

Buradan eşitlik(7.21) deki orijinal karakteristik denklemin katsayıları yardımı ile

$$K_{obsc} = K_{obs}.T^{-1} = [\alpha_n - a_n \quad \alpha_{n-1} - a_{n-1} \quad \cdots \quad \alpha_1 - a_1]T^{-1} \quad (7.27)$$

$$K_{obs} = [0.4112-1 \quad -1.1584+3.85 \quad 1-5.7 \quad 0+3.85] \quad (7.28)$$

$$K_{obs} = [-0.58 \quad 2.69 \quad -4.7 \quad 3.85] \quad (7.29)$$

$$K_{obs} = [-0.588 \quad 2.69 \quad -4.7 \quad 3.85]T^{-1} \quad (7.30)$$

K_{obs} :

$$= [-0.5888 \quad 2.6938 \quad -4.7037 \quad 3.8519] \begin{bmatrix} 0.0005 & -0.0004 & -0.0004 & 0.0005 \\ -0.0099 & 0.0288 & -0.0288 & 0.0099 \\ -0.0005 & 0.0005 & 0.0005 & -0.0005 \\ 0.0098 & -0.0293 & 0.0293 & -0.0098 \end{bmatrix}^{-1} \quad (7.31)$$

elde edilir.

Matlab kullanarak aşağıdaki komut satırı ile K_{obs} katsayısı :

```
%Tahmin edici kazancı
```

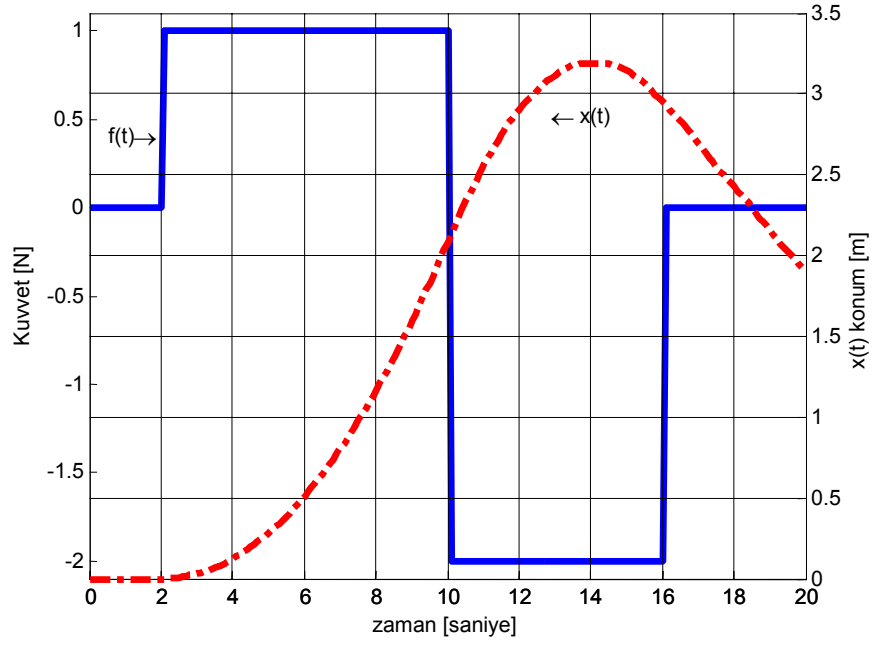
```
kobs=[p(1,5)-phi(1,5) p(1,4)-phi(1,4) p(1,3)-phi(1,3) p(1,2)-phi(1,2)];
```

```
kobsc = kobs*inv(T);
```

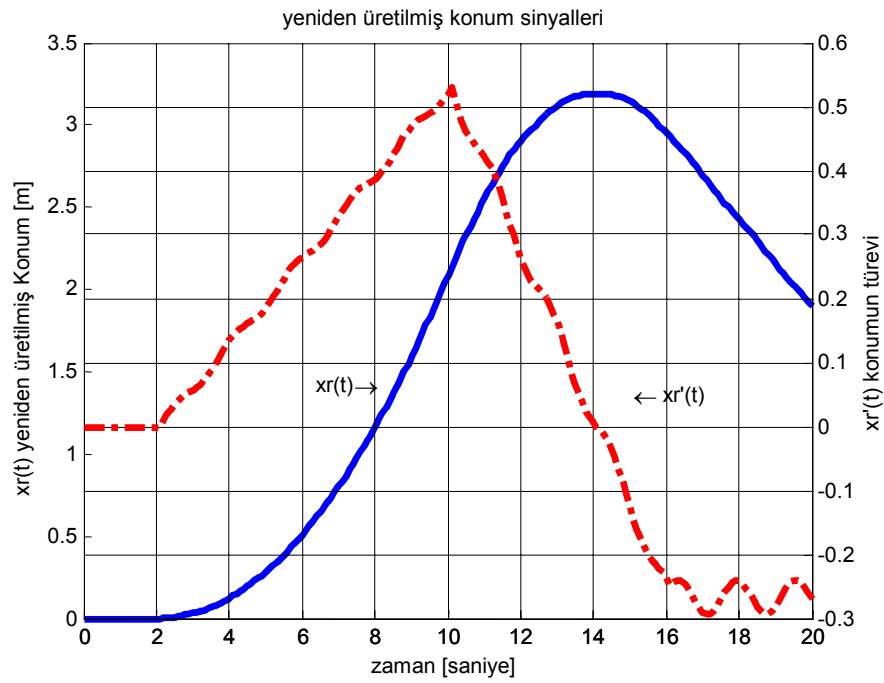
```
kobsc(2,:) = kobsc(1,:);
```

```
printmat(kobsc,'tahmin edici kazanc matrisi Kobsc')
```

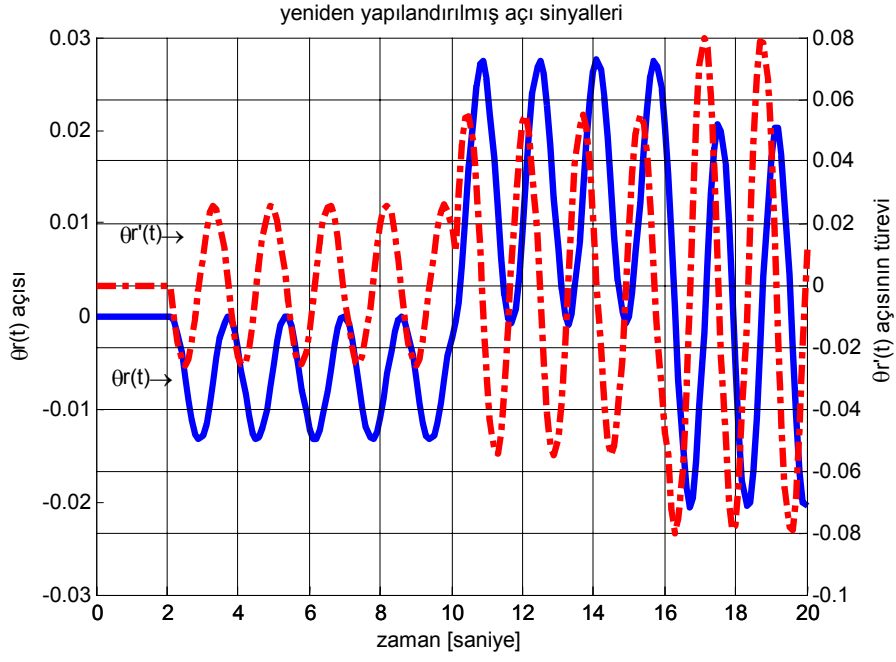
Simulink modelinde yukarıda elde ettiğimiz katsayıları yerine yazar ve de simülasyonu çalıştırır isek sistemin gözlemci yardımı ile açık çevrim cevabını elde etmiş oluruz. Aşağıdaki grafikler sistemin çıktılarını göstermektedir.



Şekil 7.2 Gözlemci dizaynı $F(t)$ girişi - $x(t)$ konum cevabı



Şekil 7.3 Yeniden yapılandırılmış konum sinyalleri



Şekil 7.4 Yeniden yapılandırılmış açılı sinyalleri

Sonuçlardan da görüldüğü gibi, $x(t)$ sinüzoidal fonksiyonu ve türev yani değişim grafiği; ayrıca açı ve açının türevi olan açı değişimleri, gözlemci metodu ile rahatlıkla elde edilebilmektedir.

Yapılan gözlemci tam durum gözlemci (full state observer) olduğu için sistemin tüm durum değişkenleri elde edilebilmektedir.

7.4 Kutup Yerleşim Metodu İle Sistemin Durum-Uzay Kontrolü

Düzeltilmiş sistemin dinamiği, $[A_d - K.B_d]$ matrisinin karakteristik eşitliğinin bir fonksiyonu olduğu bilindiğinden; düzeltilmiş sistemin α_i kutuplarını tayin etmemiz gereklidir.

Bunun için sistemde birçok kutup çiftleri denenmiş en makul sonucu veren kutup noktaları seçilerek sistem bu kutuplara göre tasarlanmıştır.

Varolan sistemin karakteristik denklemi eşitlik(7.3) de ve kökleri eşitlik (7.4), (7.5), (7.6), (7.7) elde edilmiş olup; bu köklerin

$$z_1=0.9259 + 0.3777i; z_2=0.9259 - 0.3777i; iz_3=1.0000; ve z_4=1.0000 \quad (7.32)$$

noktalarında oldukları biliniyor.

O halde birim çemberin içinde ve de sahip olunan kutuplardan daha küçük kutuplar seçilerek sistem düzeltilebilir. Düzeltilmiş sistemin karakteristik denklemini $(z - 0.6)^4$ olarak tayin edersek; 0.6 noktasında 4 adet kutbu olacak ve de stabilite koşulunu bozmayacaktır.

$$D_a=[0.6 \ 0.6 \ 0.6 \ 0.6]$$

poly(Da)

matlab komutları ile düzeltilmiş sistemin karakteristik denklemi aşağıdaki şekilde bulunmuştur. (Oluşmasını istediğimiz kontrol edilmiş sistemin karakteristik denklemi.)

$$\phi_c = z^4 - 2.4z^3 + 2.16z^2 - 0.864z + 0.1296 \quad (7.33)$$

Önce sistemin kontroledilebilirliğini test etmemiz gerekmektedir. Aşağıdaki Matlab komutları ile sistemin kontroledilebilirlik matrisi ve rankı ölçülmüştür:

$$Com=ctrb(Ad,Bd)$$

$$Rank=rank(Com)$$

Sistemin rankı n e eşit olduğundan koşul sağlanmıştır sistem kontroledilebilirdir. Kontrol için gerekli olan K geribesleme katsayılarının bulunma safhasına geçilebilir.

Beşinci bölümde bulunan (5.18) eşitliğinden $K_c = KT^{-1}$ faydalanarak sistemin geri dönüş kazanç değerleri hesaplanacaktır. Bunun için; eşitlik(7.3) deki orijinal sistemin karakteristik denkleminin katsayıları, eşitlik (7.33) de elde edilmiş olan düzeltilmiş sistemin katsayılarından çıkartılarak transformasyon matrisinin tersi ile çarpılmalıdır. Eşitlik (7.3) de

orijinal sistemin karakteristik denklemi

$$\text{phi} = \det[zI - A_d] = \phi = z^4 - 3.85z^3 + 5.7z^2 - 3.85z + 1 \text{ olduğundan} \quad (7.34)$$

$a_n = a_4 = 1$; $a_{n-1} = a_3 = -3.85$; $a_{n-2} = a_2 = 5.7$; $a_{n-3} = a_1 = -3.85$ $a_0 = 1$ değerleri eşitlik (7.21) den bilindiğinden;

oluşturulan sistemin karakteristik denklemi

$$\text{phic} = \phi_c = z^4 - 2.4z^3 + 2.16z^2 - 0.864z + 0.1296 \text{ olduğundan}$$

(7.2) prosesin karakteristik denkleminde faydalanılarak

$$\alpha_4 = 0.1296; \quad \alpha_3 = -0.864; \quad \alpha_2 = 2.16; \quad \alpha_1 = -2.4; \quad \alpha_0 = 1 \quad (7.35)$$

yazılabilir. Değerler eşitlik (7.36) da yerine yazıldığında:

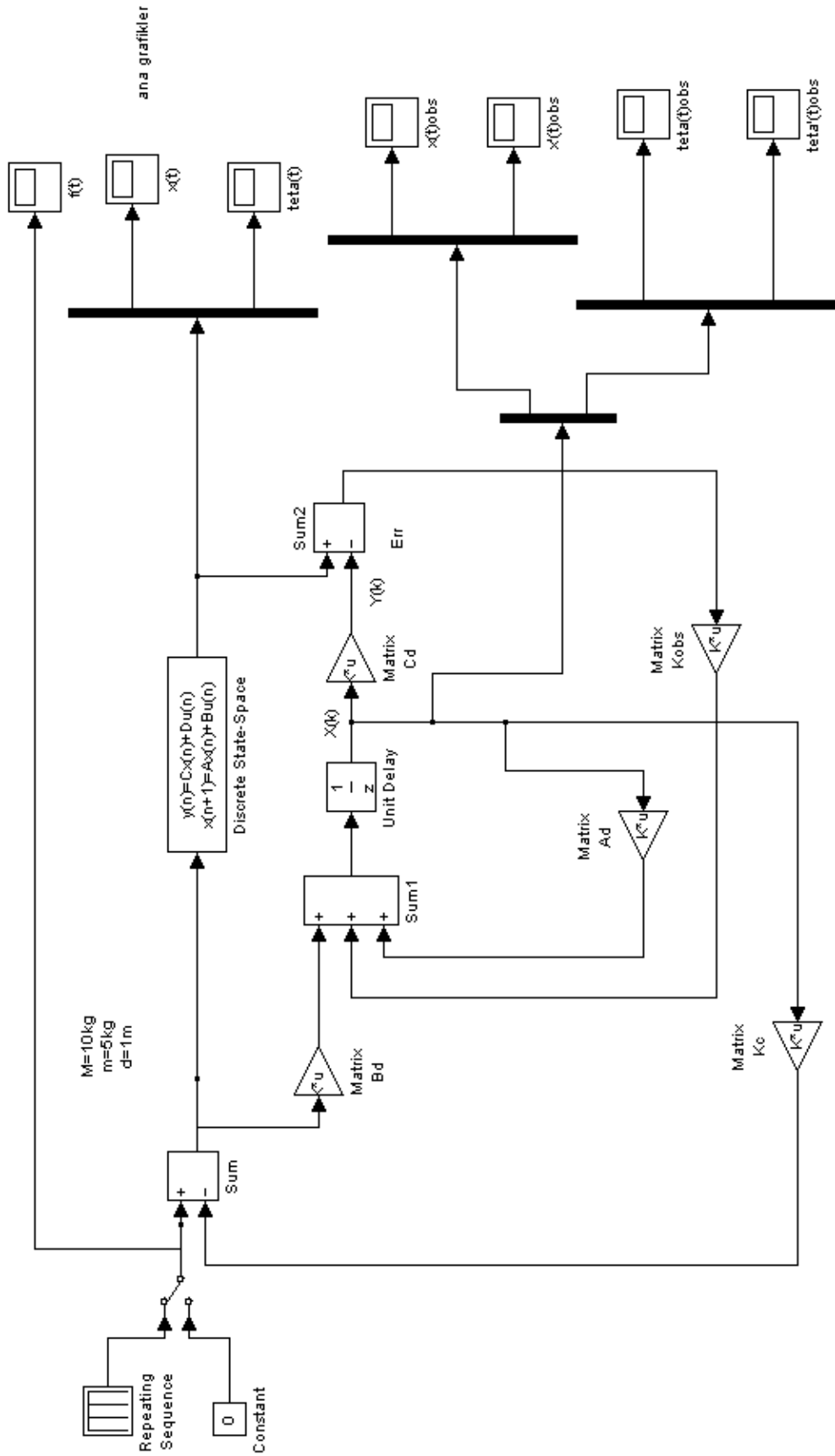
$$K_c = KT^{-1} = [\alpha_n - a_n \quad \alpha_{n-1} - a_{n-1} \quad \cdots \quad \alpha_1 - a_1]T^{-1} \quad (7.36)$$

$$K_c = KT^{-1} = [0.1296-1 \quad -0.864+3.85 \quad 2.16-5.7 \quad -2.4+3.85]T^{-1} \quad (7.37)$$

$$K_c = [-0.8704 \quad 2.9879 \quad -3.5437 \quad 1.4519] \begin{bmatrix} 0.0005 & -0.0004 & -0.0004 & 0.0005 \\ -0.0099 & 0.0288 & -0.0288 & 0.0099 \\ -0.0005 & 0.0005 & 0.0005 & -0.0005 \\ 0.0098 & -0.0293 & 0.0293 & -0.0098 \end{bmatrix}^{-1} \quad (7.38)$$

$$K_c = [259.2241 \quad 220.3405 \quad -327.3882 \quad 105.0104] \quad (7.39)$$

elde edilir.



Şekil 7.5 Gözlemcili sistemin geribesleme Simulink modeli

Yukarıdaki işlemler; aşağıda verilen Matlab komut satırları kullanılarak da elde edilebilir:

%durum geribesleme katsayılarının bulunması

n = 4; % sistemin derecesi

phi = poly(Ad);

phic = poly(Da)

for i = 1:n

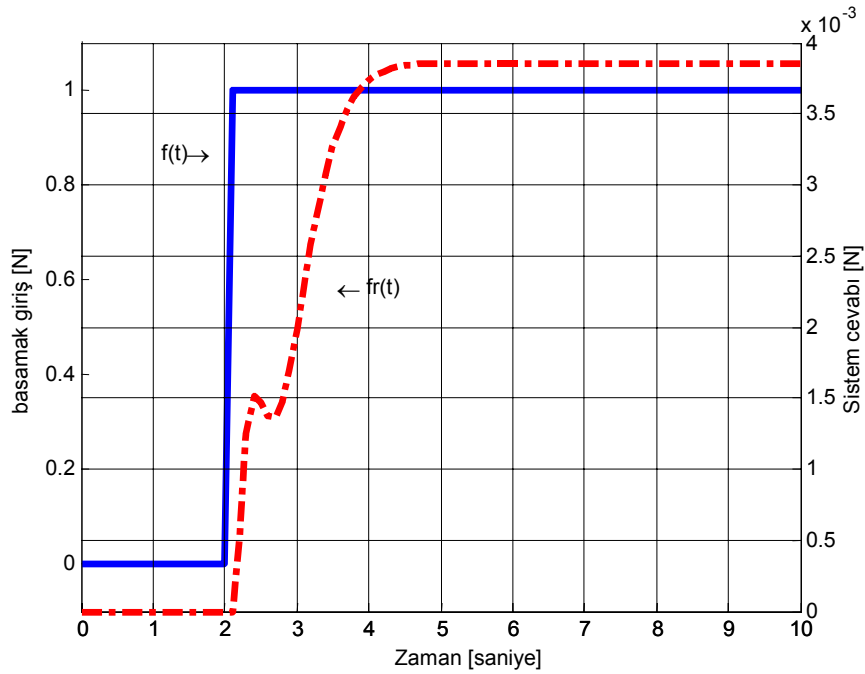
K(1,n-i+1) = phic(1,i+1)-phi(1,i+1);

end

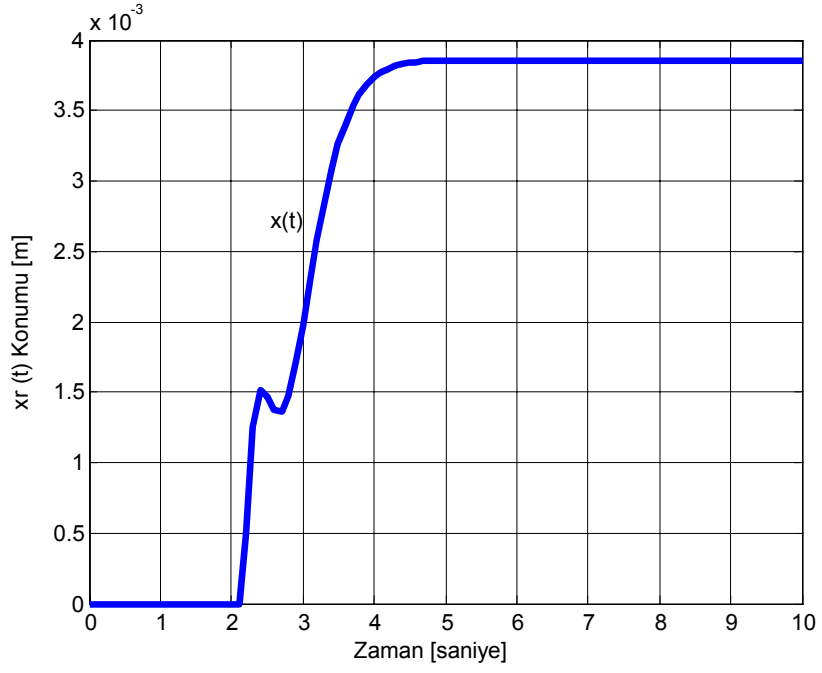
Kc = K*inv(T);

printmat(Kc, durum geribesleme katsayısı Kc matrisi')

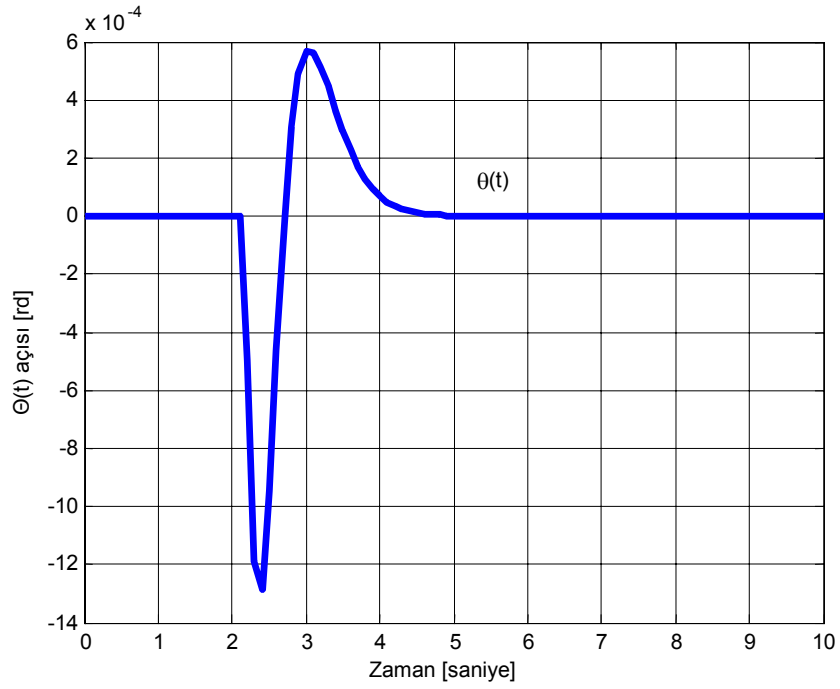
Simulink modelinde yukarıda elde ettiğimiz katsayıları yerine yazar ve de simülasyonu çalıştırır isek sistemin kapalı çevrim cevabını elde etmiş oluruz.



Şekil 7.6 Kutup yerleşim metodu basamak kuvvet girişi cevabı



Şekil 7.7 Kutup yerleşim metodu basamak kuvvet girişi cevabı- $x(t)$ konumu

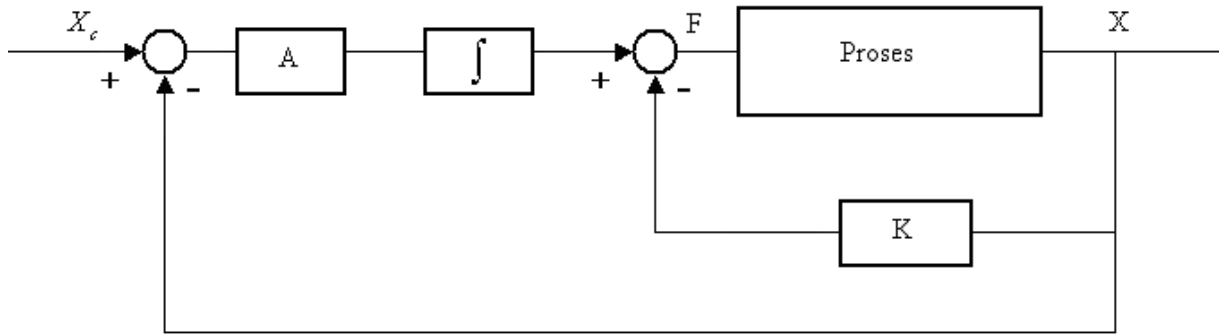


Şekil 7.8 Kutup yerleşim metodu basamak kuvvet girişi cevabı- $teta$ açısı

Yukarıda elde edilen grafikler kutup yerleşim metodu ile bulunan K_c katsayısının, gözlemci dizaynı içeren sisteme geribeslenmesi ile elde edilmiş olup sistem 3 saniye içinde verilen basamak girişe cevap verebilmektedir. Salınım açısı en çok 0.012 radyan olup 0.68 dereceye tekabül etmektedir. Sistemin salınım kontrolü sağlanmış; fakat araba pozisyonunda kalıcı hal hatası olduğu fark edilmiştir. Bunun için sisteme bir integratör katmanı ve kazanç değeri eklenerek x pozisyonundaki kalıcı hal hatası giderilmeye çalışılacaktır.

7.4.1 Sisteme İntegratör Eklenmesi

Kontrol edilen sistemin dinamiklerinde memnun olmak ile birlikte; kalıcı hal hatasını minimuma götürmeyi hedefliyoruz. O yüzden x pozisyonunu kontrol etmek için integratör katmanı sisteme eklenmiştir.



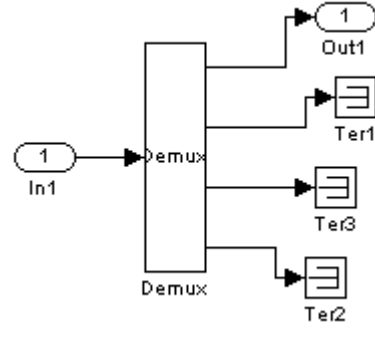
Şekil 7.9 İntegratörlü sistemin genel gösterimi

x konumunu kontrol edebilmek için durum vektörlerindeki ilk terimi geribeslemek gerekmektedir. Bu yüzden vektör içindeki diğer terimler terminate bloğu ile engellenmiştir.

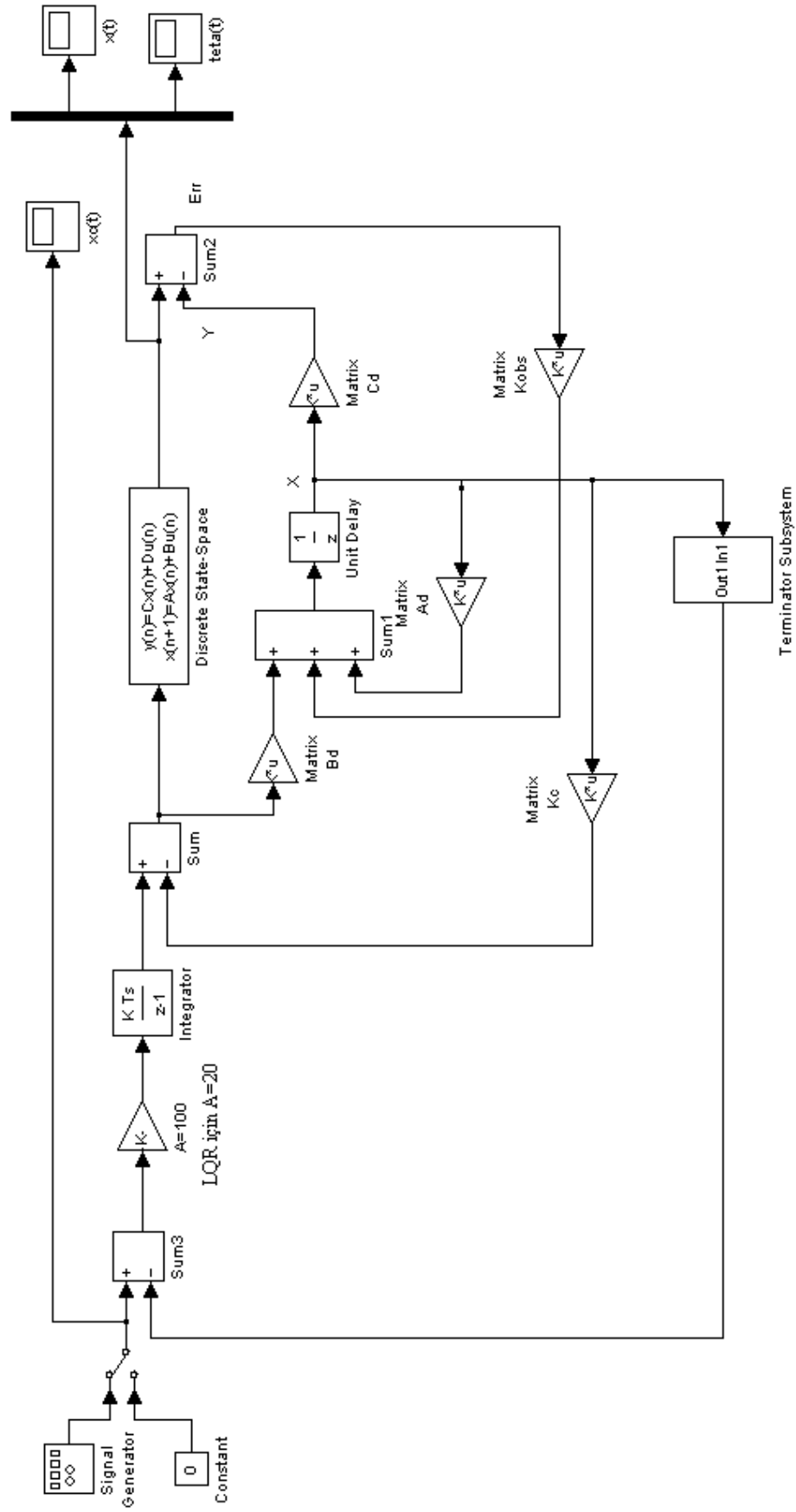
A katsayısı kutup yerleşimi metodu için 100 olarak seçilmiş olup üst aşımı (overshoot) engellemek için kullanılmıştır.

Şimdi giriş olarak kuvvet yerine pozisyon girerek pozisyon değerini kontrol edelim ve kalıcı hal hatasını sıfırlayalım:

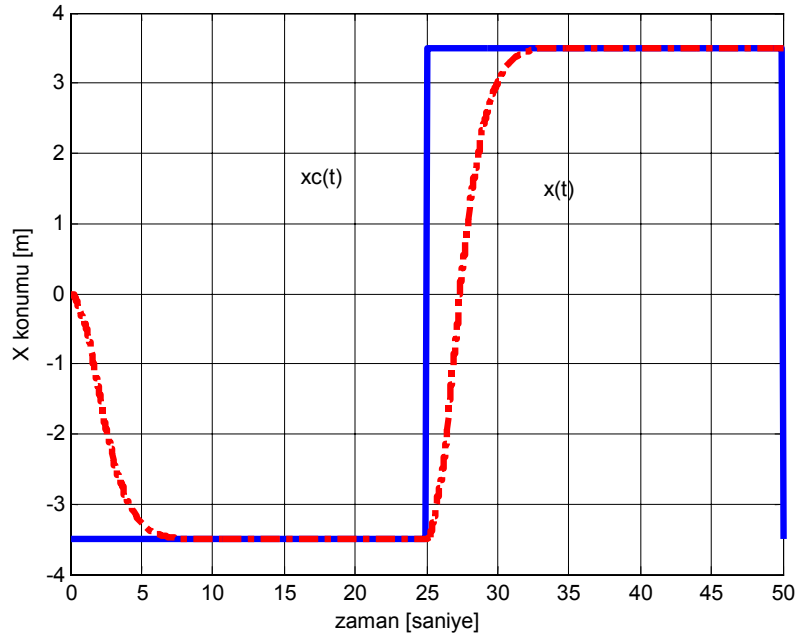
Terminator Altsistemi



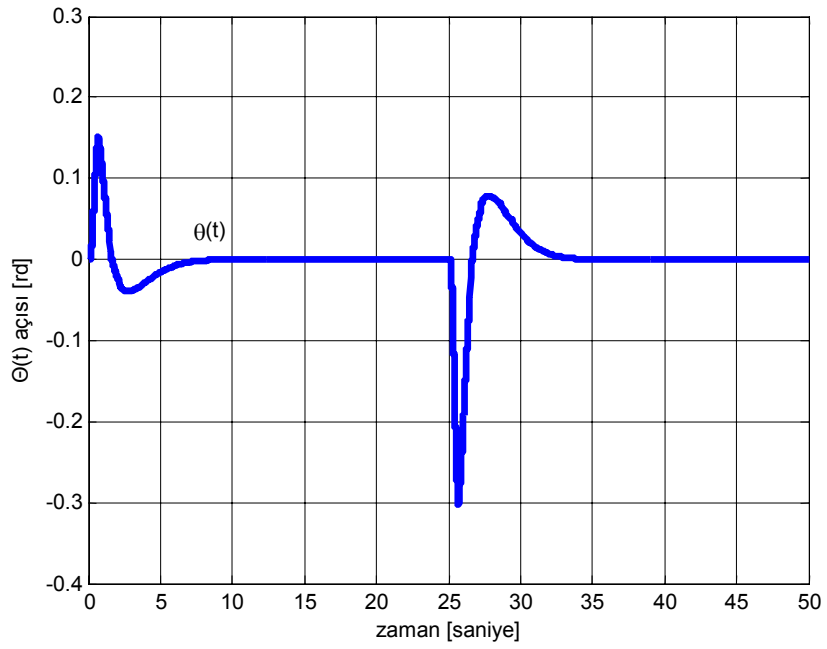
Şekil 7.10 Terminator Altsistemi



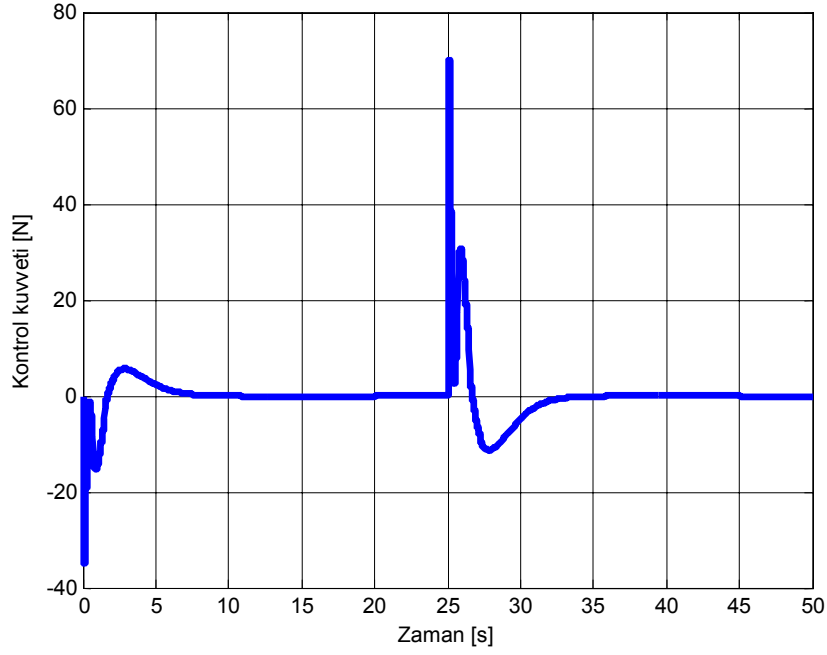
Şekil 7.11 İntegratör içeren gözlemcili geribesleme Simulink modeli



Şekil 7.12 İntegratör katmanı sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.13 İntegratör katmanı sonrası $teta$ açısı cevabı



Şekil 7.14 İntegratör katmanı sonrası uygulanan kontrol kuvveti

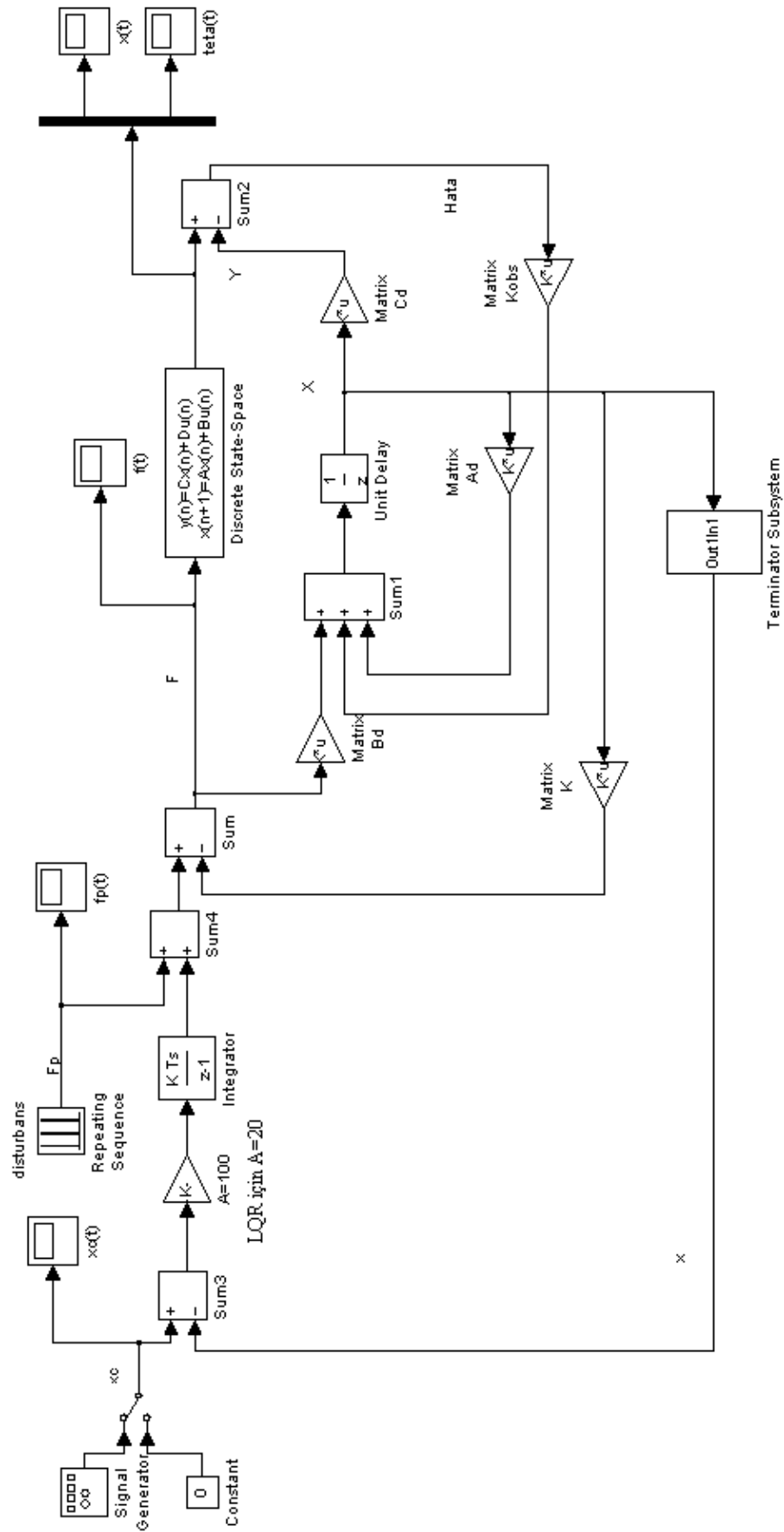
İnegratörlü simülasyon çalıştırıldığında yukarıdaki grafikler elde edilmektedir. Görülüyor ki araba istenilen pozisyona üst aşım yapmadan gidebilmektedir. Pozisyonadaki kalıcı hal hatası giderilmiştir.

Teta açısının değişimi ise maksimum 0.3 radyan olup arabanın yön değiştirdiği anda oluşmaktadır. Buda değer yaklaşık 17 derece olarak hesaplanmıştır.

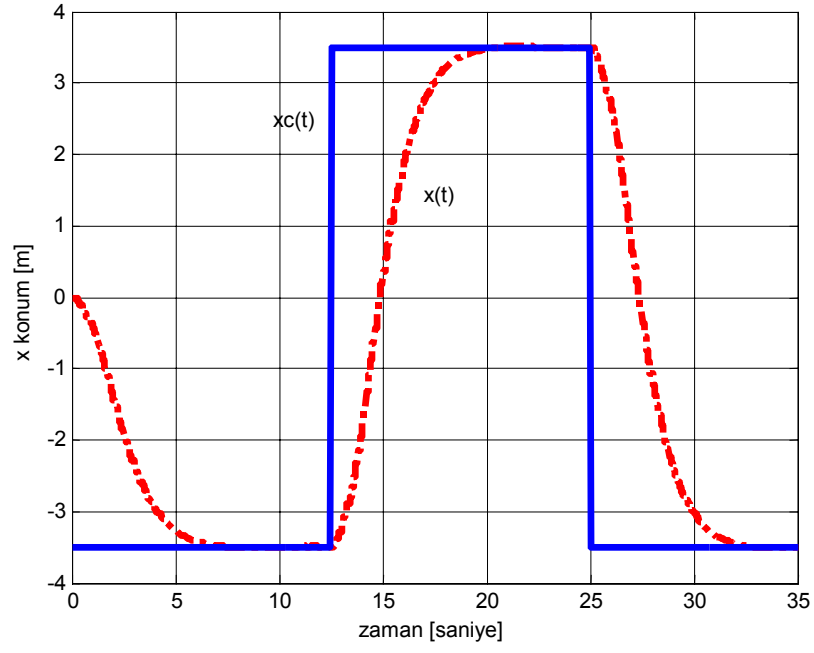
7.4.2 Sisteme Distürbans Eklenmesi

Sisteme Distürbans eklenerek sistemin dış kuvvetlere olan etkisi simülasyonda gözlenebilir.

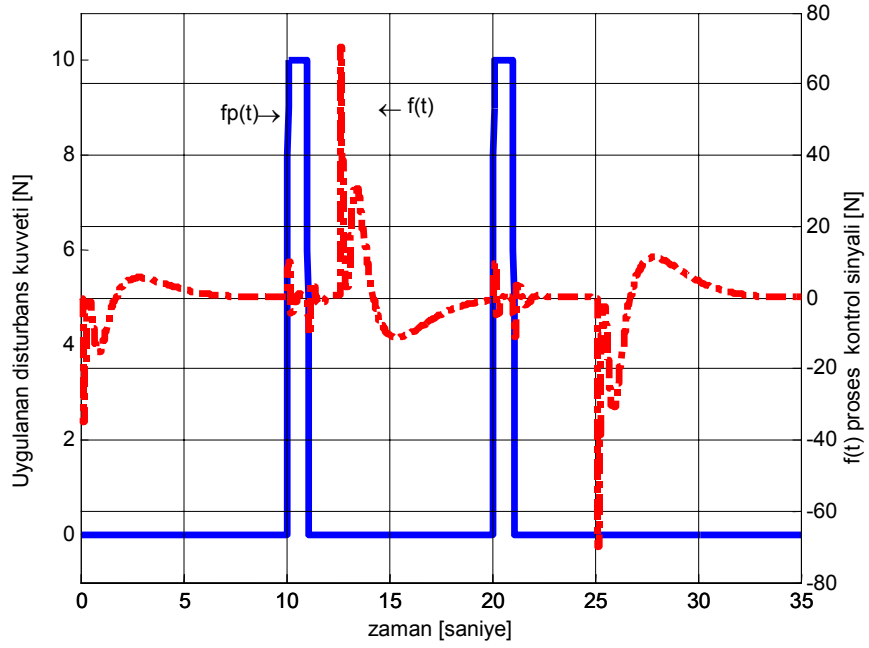
Sisteme belirli aralıklarda birer saniyelik 10N gücünde distürbans ilave edilirse sistem şekil(7.15) deki hali alacaktır. Şekil(7.16), (7.17) ve (7.18) de sistemin bu distürbansa karşı verdiği cevaplar elde edilmiş olup; görüldüğü gibi etkiyen dış kuvvet araba pozisyonunda belirgin bir etki yapmaz iken sarkaç salınımında küçük oynamalara yol açmaktadır; fakat maksimum salınım açısında bir artma gözlenmemiştir.



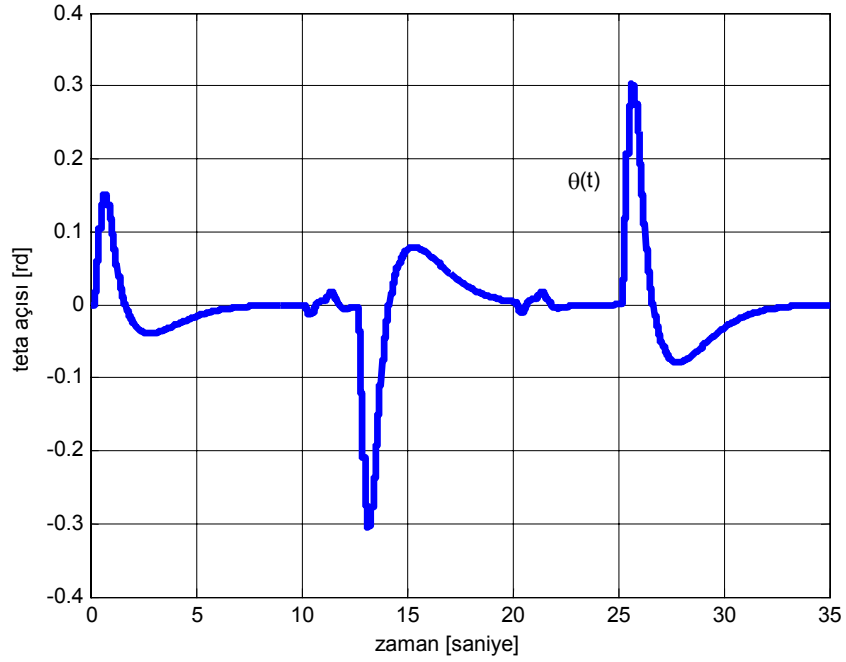
Şekil 7.15 Distürbans eklenmiş Simulink modeli



Şekil 7.16 Distürbans sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.17 Distürbans-kontrol sinyali cevabı

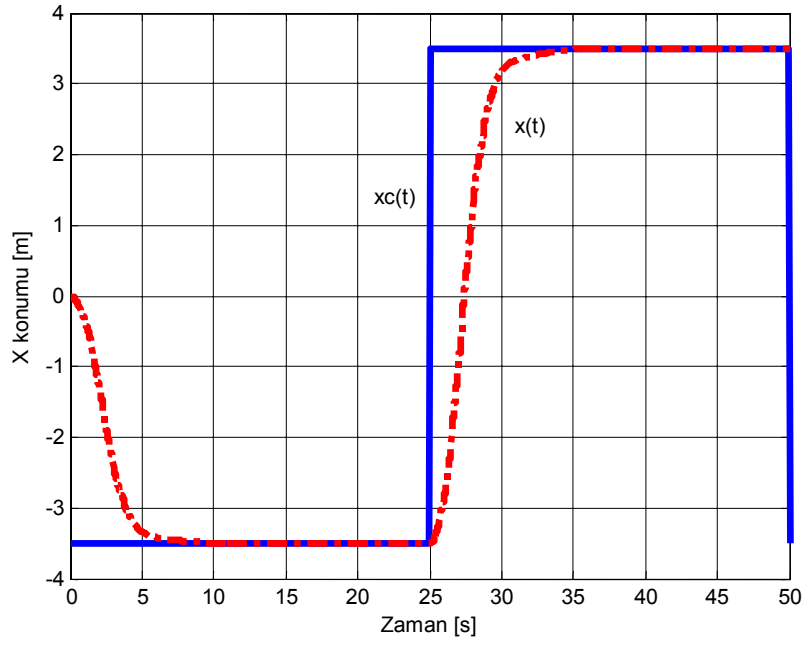


Şekil 7.18 Distürbans sonrası *teta* açısı cevabı

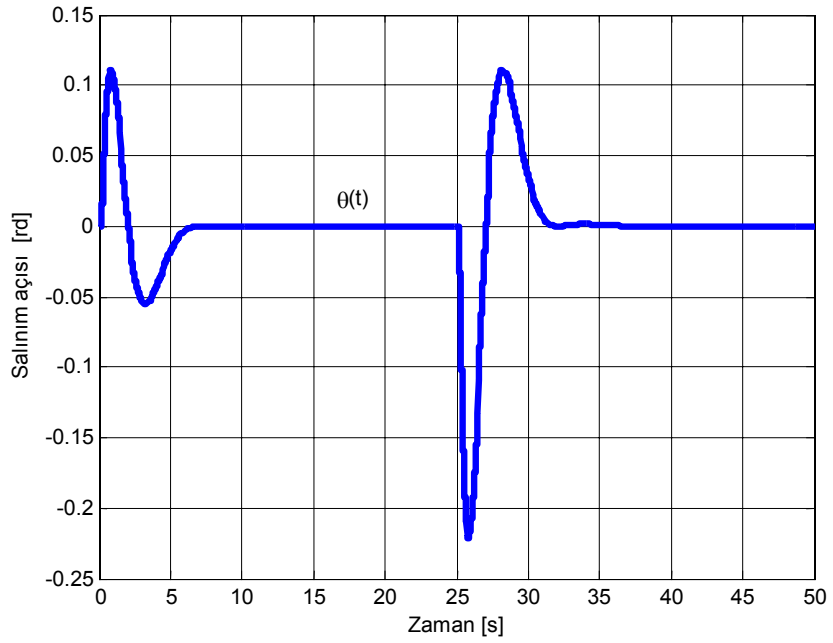
7.4.3 Değişken Yüklerde Kutup Yerleşim Metodu Cevabı

Eşitlik (7.39) da elde edilen K_c kutup yerleşimi metodu geri dönüş kazanç değeri, sistemde 5 kg lik sarkaç yükü olduğu kabulü üzerine bulunmuş bir değerdir. Fakat vinç sistemlerinde yükün çok farklı değerler aldığı bilinmektedir. Bu sebeple aynı K_c ve A integratör kazanç katsayıları kullanılarak sistemin değişken yüklerde cevabı incelenmiştir. Sisteme sırası ile 50; 0.5 ve 100 kg lik ağırlıklar koyulmuştur. Her bir yeni ağırlık için sistemin durum-uzay modelinde yeni A_d , B_d , C_d ve D_d katsayıları teker teker hesaplanarak Şekil (7.11) deki Simulink sistemde denenmiştir.

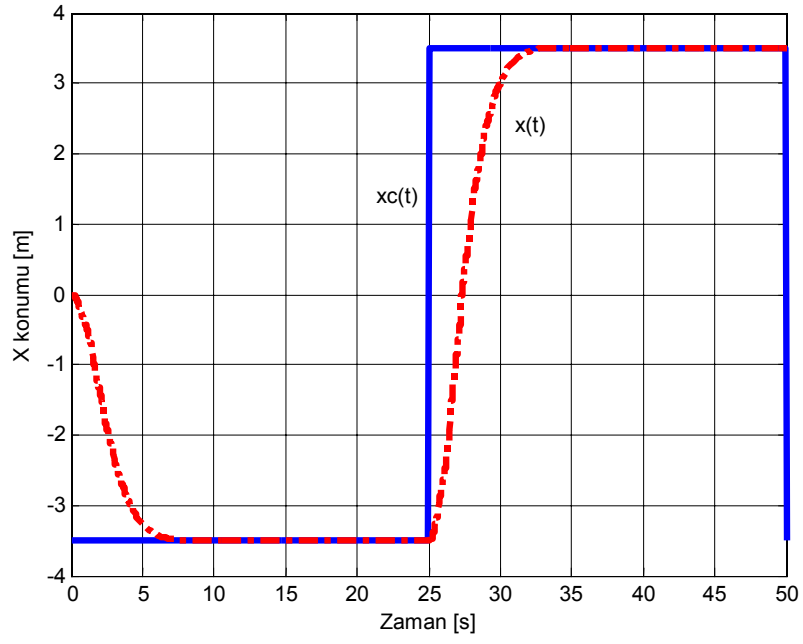
Aşağıdaki grafiklerde görüldüğü gibi; kutup yerleşim metodu ile hem 0.5 hem de 50 kg yüklerde pozisyon ve açı kontrolü istenen değerler arasında kalmaktadır. Sönüm zamanının az da olsa arttığı gözlemlenmiştir. 100 kg yükü için ise pozisyon hataları başlamakta buna rağmen pozisyonda kalıcı hal hatası görülmemekte; ayrıca salınım açısında bozulma gözlemlenmemiştir.



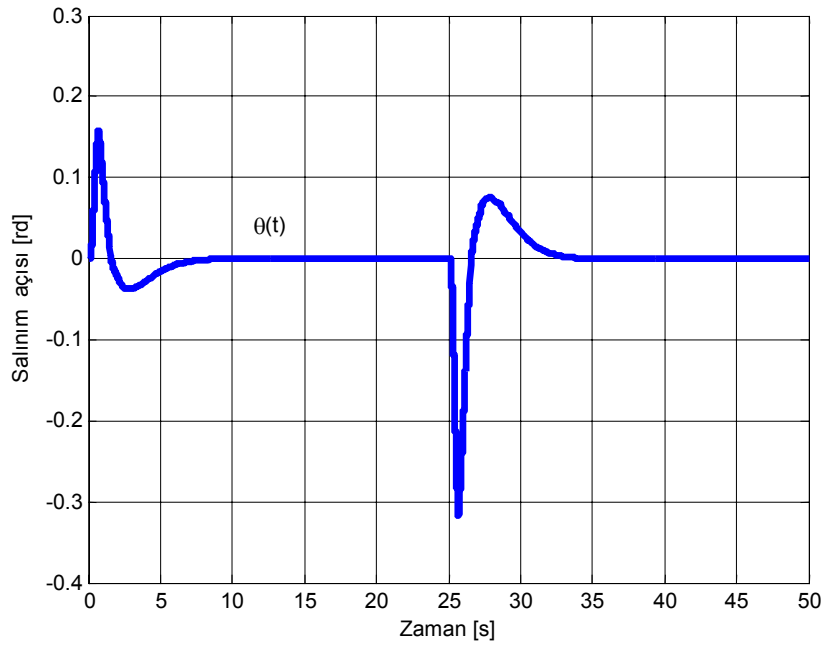
Şekil 7.19 50 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



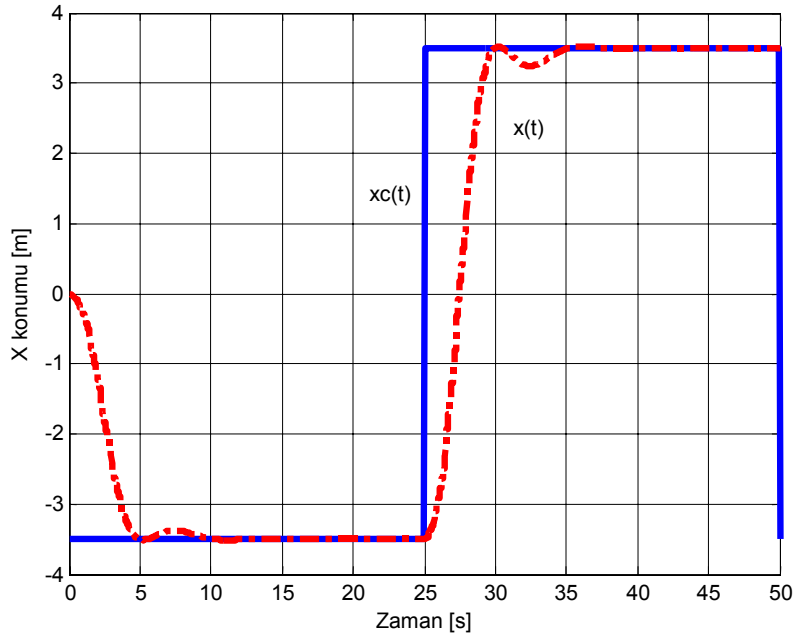
Şekil 7.20 50 kg sarkaç yükünde $teta$ açısı cevabı



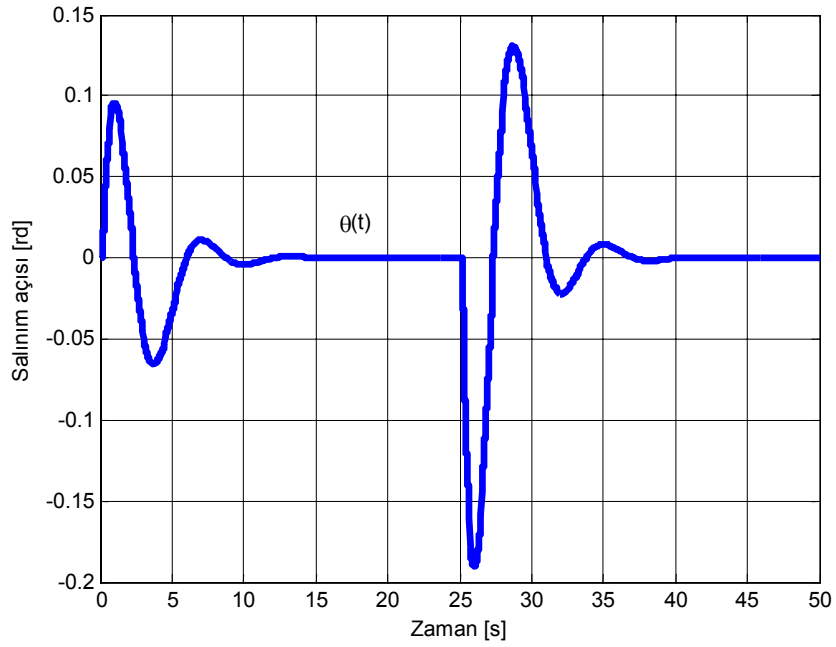
Şekil 7.21 0,5 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.22 0,5 kg sarkaç yükünde $teta$ açısı cevabı



Şekil 7.23 100 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.24 100 kg sarkaç yükünde $teta$ açısı cevabı

7.5 LQR Yöntemi ile Sistemin Durum-Uzay Kontrolü

Bir önceki bölümde, kutup yerleşim problemi; sistemi istediğimiz kapalı çevrim kutup noktalarına yerleştirebilmek için, durum geribesleme kazanç matrislerinin işleme sokulması ile çözülmüştü. Bölüm (5.1) de bahsedildiği gibi; istenen bu kutup noktalarını belirlemede birkaç metot olsa bile, gerçekte fiziksel sistemde bir çok farklı kutup noktaları belirlenerek bu iterasyonlar sonucu en güzel cevabın seçilmesine dayanmaktadır. Diğer bir deyişle kazanç değerlerini direkt olarak birer birer değiştirmek yerine (bu sistem kapalı çevrim sistemin stabilite koşulu garantilememektedir); istenen kutup noktaları değiştirilerek istenen cevaba ulaşılmaya çalışılmaktadır. Özetle kutup yerleşim dizaynı bir çok kutup noktasının seçilerek kontrol sisteminde denenmesidir.

Bu bölümde lineer kvadratik optimal regülatör kullanarak sistemin geribesleme katsayıları elde edilerek kapalı çevrim cevabı bulunacaktır. Bunun için bir performans indeks fonksiyonundan J faydalanılmaktadır. Bu fonksiyon kontrol sisteminin fiziksel cevaplarını inceleyerek performans ölçümü yapmaktadır. Sistemin istenilen davranışı ile gerçek davranışı arasındaki hatayı ölçer. Bu hata genellikle durum değişkenlerinin ve sistemin girişlerinin fonksiyonu şeklindedir.

Performans indeksi belirlendikten sonra yapılması gereken matematiksel işlemler (Calculus) ile girişe bağlı olarak bu indeks değerini minimize etmektir. Bunu giriş kontrol vektörünü indeks matrisini minimum yapacak değerde seçerek yapmaktayız. Bu giriş değerleri optimaldir. Eğer performans indeksi değişecek olursa optimal girişlerde beraberinde değişecektir. Optimal kontrol dizaynı kontrol sisteminin davranışı kabul edilebilir değerlere ulaşana kadar performans indeksinin değiştirilmesine dayanmaktadır. Kuadratik optimal kontrol şeması kullanmanın bir avantajı sistem dizaynının çok özel durumlar hariç stabil olmasıdır. (Ogata 1994)

Performans indeks fonksiyonu ayrık zamanda aşağıdaki şekilde gösterilmektedir:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x(k)^T Qx(k) + u(k)^T Ru(k)] \quad (7.43)$$

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x'(k)Qx(k) + u'(k)Ru(k)] \quad (7.44)$$

Bu indeks matrisi Lagrange formuna sokulur ise sonuç olarak K değeri aşağıdaki şekilde elde edilmektedir. (Ogata 1994)

$$K(k) = R^{-1}B'_d(A'_d)^{-1}[P(k) - Q] \quad (7.45)$$

Tanımlı $P(k)$ değeri kalıcı hal matrisi olup *Ricatti Denklemi* olarak adlandırılır.

$$P(k) = Q + A'_d P(k+1)A_d - A'_d P(k+1)B_d [R + B'_d P(k+1)B_d]^{-1} B'_d P(k+1)A_d \quad (7.46)$$

Ayrık zamanda Matlab ile bu K değeri dlqr fonksiyonu ile çözülebilmektedir. Bu komut P kalıcı hal matrisini iterasyonlar sonucu son değerine ulaşmasını sağlar. P matrisi son değere ulaştıktan sonra K matrisinde yerine konarak optimal geribesleme kazanç değerinin bulunmasını sağlamaktadır.

Buradaki Q ve R pozitif olarak tanımlanmış Hermitian veya gerçek simetrik matrislerdir. R matrisi Q ya göre büyük seçilirse optimal regülatör durum vektörünü sıfıra küçük bir kontrol eforu sarf ederek getirecektir. Fakat sistemin oturma zamanı artmaktadır. Öte yandan R matrisi Q ya göre küçük seçilirse, daha büyük bir kontrol eforu sarf edilerek durum vektörünü hızlı şekilde sıfıra götürerek oturma zamanını küçültecektir.

Q ve R değerlerini dengelenerek sistemin cevabının hızı ve kontrol eforu belirlenmektedir.

Q ve R matrislerindeki ağırlıkların $C'.C$ formundan elde edilebilmesi bir çözüm yaklaşımı olup belirlenmesinde temel rol oynayan özel bir metot bulunmamaktadır. (Keller 2004). Q ve R matrisleri ne kadar büyütülürse hata o kadar azalmakta ve sistem cevabı hızlanmaktadır. Fakat kontrol eforunun büyük değerlere ulaşmamasına dikkat edilmelidir.

Yapılan Çalışmada önce $R=1$ alınmış

$C'.C$ den:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.47)$$

elde edilmiştir.

Q_{11} değeri X pozisyonunun ağırlığını etkilemekte olup Q_{33} değeri teta salınım açısının ağırlığını etkilediği görülmüştür.

Sisteme Q_{11} ve Q_{33} ün farklı değerleri için sistemin basamak cevabı incelenmiş ve istenen cevaba uygun en küçük geribesleme katsayısı bulunmaya çalışılmıştır.

7.5.1 Q_{11} ve Q_{33} Ağırlıklarının Seçilmesi

Bunun için bir döngü tasarlanarak, bu Q_{11} ve Q_{33} e farklı değerler atanmış ve sistemin cevabı incelenmiş ve sonuçlar bir Excel dosyasına gönderilerek en makul cevap içlerinden ayıklanmıştır. Sistem cevabında incelenen değerler üst aşım ve sönüm oranı değerleridir. Sistemin kontroledilebilirliği ve gözlenebilirliği daha önceden ele alındığından tekrar bu kriterler incelenmemiştir. Sistemin kontroledilebilir ve gözlenebilir olduğu bölüm (7.3) den öğrenilebilir.

Bu işlem Matlab kullanarak:

```
%LQR bulunması
```

```
T=0:0.01:5;
```

```
U=ones(size(T));
```

```
Data=zeros(60,4); %60*4 lük sıfır matris tanımlansın( bu matrisin boyutları işlem sırasında %değişeceğiinden ilk başta sadece tanıtmak için verilmiştir).
```

```
%kx ve kteta teker teker deneniyor.
```

```
x=0
```

```
for i=5000:200:25000; %1, sütun
```

```
for j=5000:500:25000; %2. sütun
```

```
x=x+1
```

```
Data(x,1)=x;
```

```
Data(x,2)=i;
```

```
Data(x,3)=j;
```

```
%
```

```
kteta=j;
```

```
kx=i;
```

```
Q=[kx 0 0 0;0 0 0 0;0 0 kteta 0;0 0 0 1];
```

```
R=1;
```

```

klqr=dlqr(Ad,Bd,Q,R);

Ac=[Ad-Bd*klqr];    % klqr Geribesleme sonrası oluşan yeni A matrisi

% LQR simülasyonu; basamak giriş

[y1,x1]=dlsim(Ac,Bd,Cd,Dd,U);

%üst aşımın bulunması

maks=max(y1);

maksx=maks(1,1);

maksteta=maks(1,2);

Mpx=(maksx-y1(500,1));

Mpteta=(maksteta-y1(500,2)) ;

Data(x,4)=Mpx;

Data(x,5)=Mpteta;

% sönüm oranı

settling_percent = 0.0002;

chan1x = y1(:,1);

maxtimex = max(find(abs((chan1x - chan1x(end)) / chan1x(end) > settling_percent)));

settling_time_x = 0.01*(maxtimex + 1);

Data(x,6)=settling_time_x;  % x in değeri

chan1teta = y1(:,2);

maxtimeteta = max(find(abs((chan1teta - chan1teta(end)) / chan1teta(end)
>settling_percent)));

settling_time_teta = 0.01*(maxtimeteta + 1);

Data(x,7)=settling_time_teta;  % teta nın değeri

% klqr kazanç değeri

```

```

Data(x,9)=klqr(1,1);
Data(x,10)=klqr(1,2);
Data(x,11)=klqr(1,3);
Data(x,12)=klqr(1,4);

end

end

Data

xlswrite(Data,'ağırlıklar ve sistem cevapları')

% Kod sonu

```

Matlabde $dlqr$ komutu indeks matrisini minimize ederek geribesleme K_{lqr} katsayılarının bulunmasını sağlamaktadır. Bu katsayılar daha sonra simülasyonda yerine yazılarak sistem çalıştırılmış ve sonuçta Q ve R değerleri şu şekilde seçilmiştir:

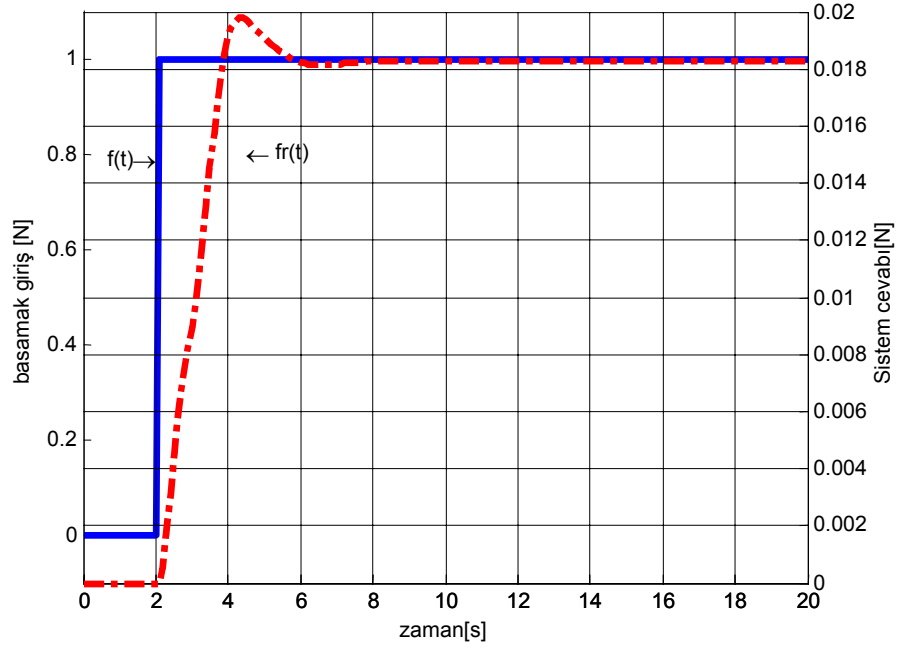
$$R=1; \quad Q = \begin{bmatrix} 5200 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 14500 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.48)$$

Bu ağırlıklardan elde edilen geribesleme katsayıları:

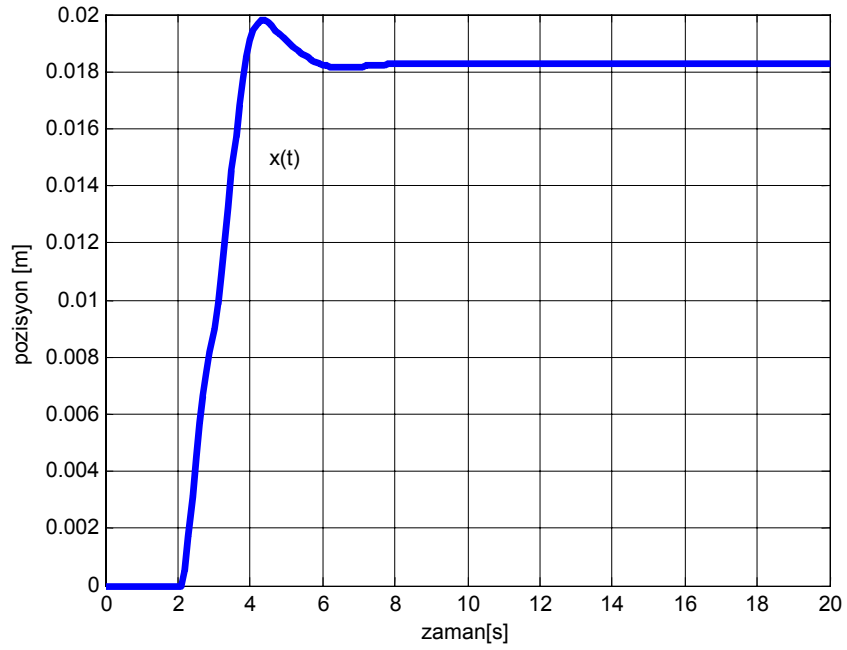
$$K_{lqr} = [54.7832 \quad 46.6893 \quad -48.9565 \quad -1.1554] \quad (7.49)$$

dir.

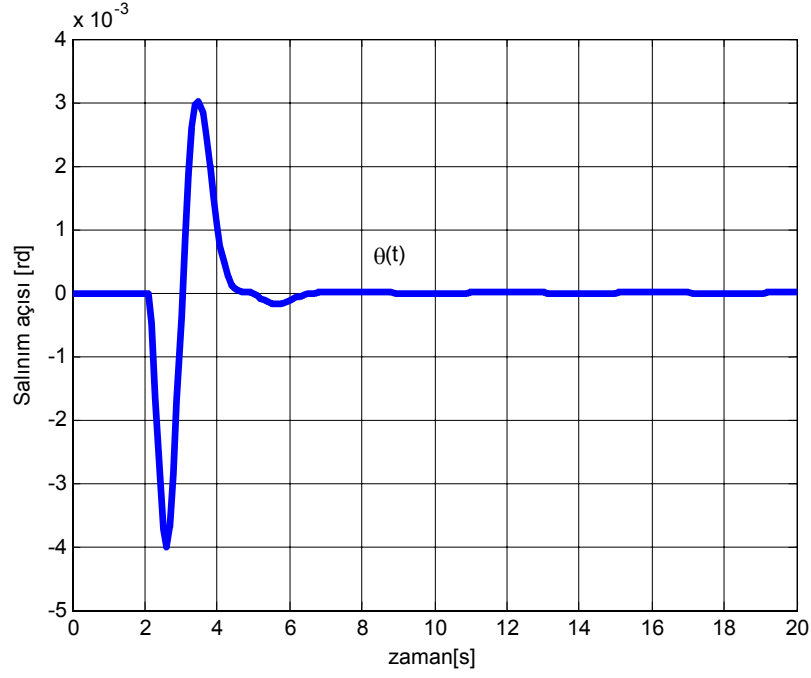
Bu katsayı, sistemde kutup yerleşimi için hazırlanmış olan gözlemcili Simulink dosyalarında (şekil 7.5) yerine yerleştirildiğinde sistemin çıktıları aşağıdaki gibi bulunmaktadır:



Şekil 7.25 LQR metodu basamak kuvvet girişi cevabı



Şekil 7.26 LQR metodu basamak kuvvet girişi cevabı- $x(t)$ konumu

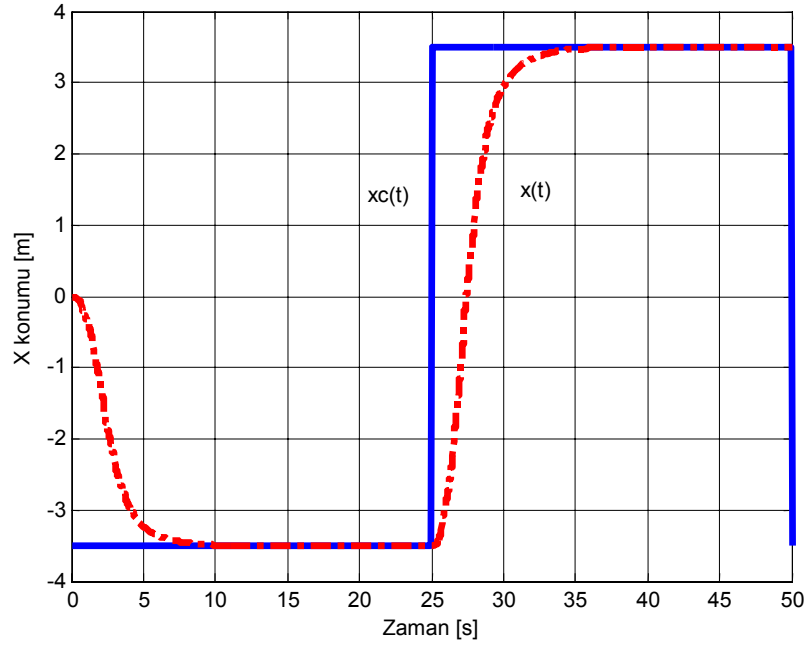


Şekil 7.27 LQR metodu basamak kuvvet girişi cevabı-*teta* açısı

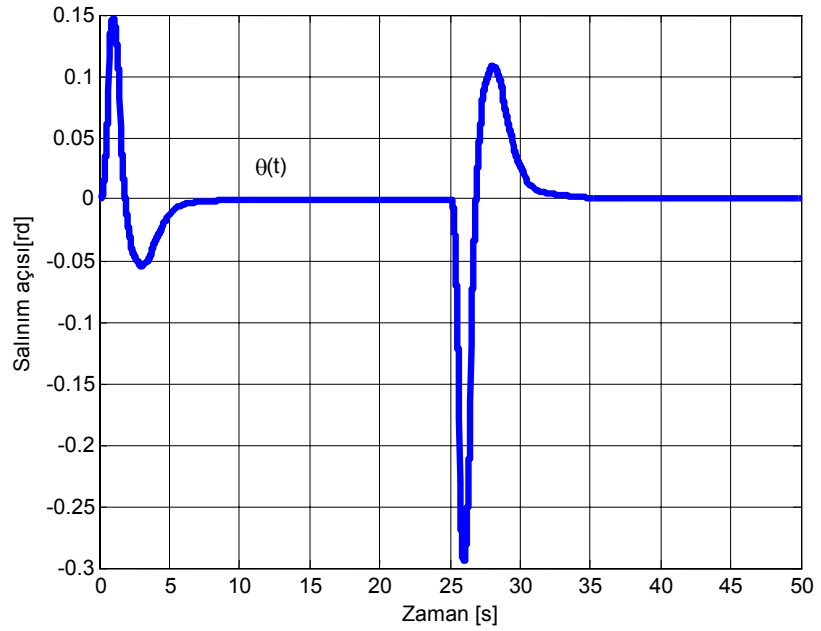
Grafiklerden görüldüğü gibi salınım açısı istenen değer aralığında olup pozisyonda kalıcı hal hatası mevcuttur. Pozisyon cevabında 0.002 gibi bir üst aşım olduğu görülse de integratör bloğu ekleneceğinden bu değer kabuledilebilir sınırın içinde kalmaktadır.

7.5.2 Sisteme İntegratör Eklenmesi

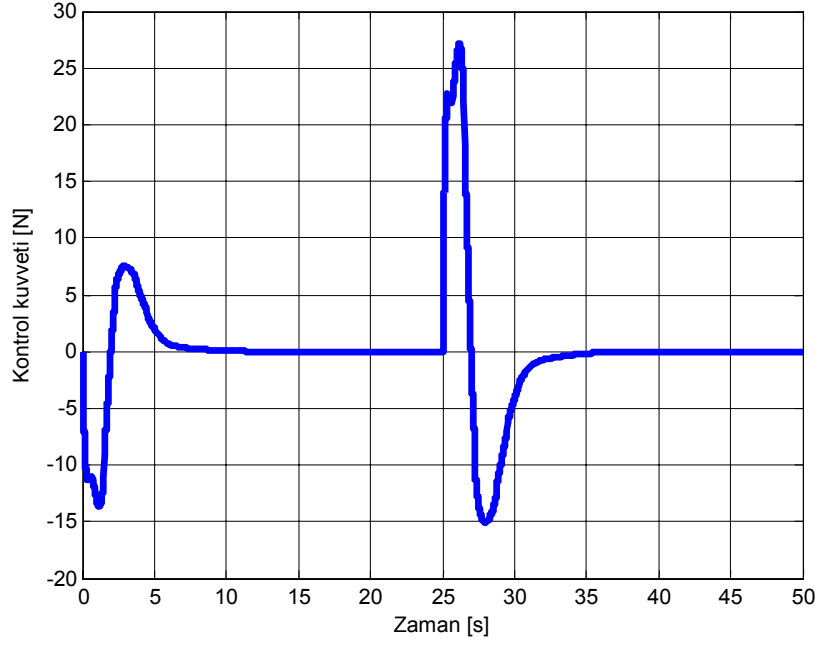
Sistemde kalıcı hal hatası olduğu görülüyor. Bunu diremek için integratör bloğu içeren simülasyonda bu değerler yerleştirilir ise (şekil 7.11) ve A değeri 20 olarak seçilirse çıktılar aşağıdaki şekilde bulunmaktadır. Görüldüğü gibi grafikler, şekil (7.12);(7.13) de bulunan kutup yerleşim metodu sistem cevaplarının hemen hemen aynısı olup; geribesleme kazanç değerlerinde kutup yerleşimine göre büyük oranda azalma gözlemlenmiştir.



Şekil 7.28 LQR metodu integratör katmanı sonrası $x_c(t)-x(t)$ konumu



Şekil 7.29 LQR metodu integratör katmanı sonrası $teta$ açısı

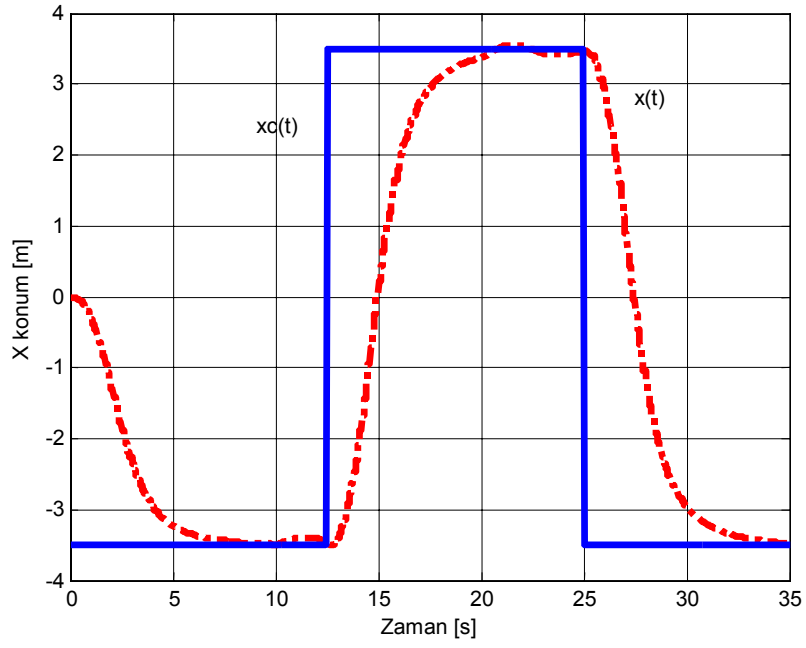


Şekil 7.30 LQR metodu integratör katmanı sonrası uygulanan kontrol kuvveti

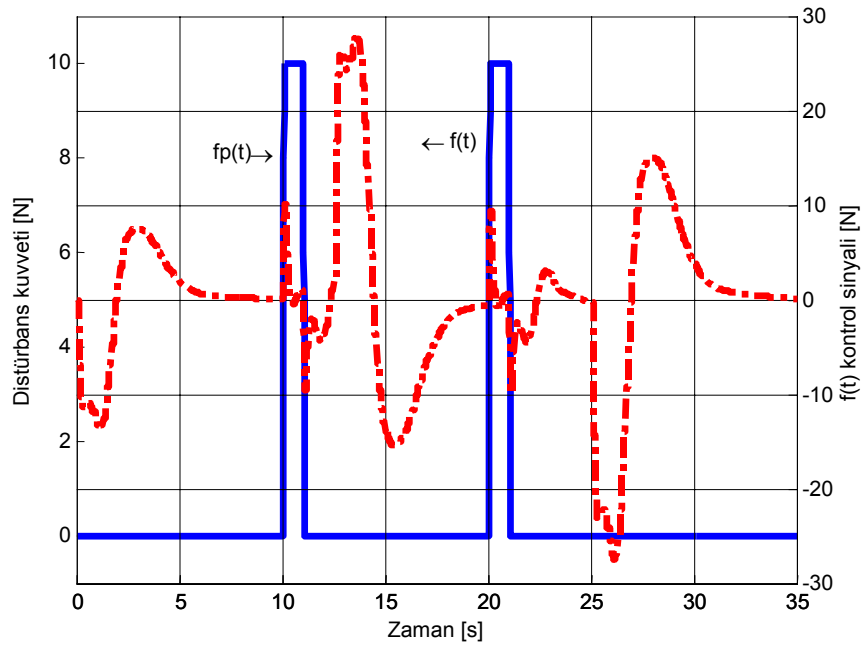
7.5.3 Sisteme Distürbans Eklenmesi

Sisteme Distürbans eklenerek sistemin dış kuvvetlere olan etkisi simülasyonda gözlenmiştir.

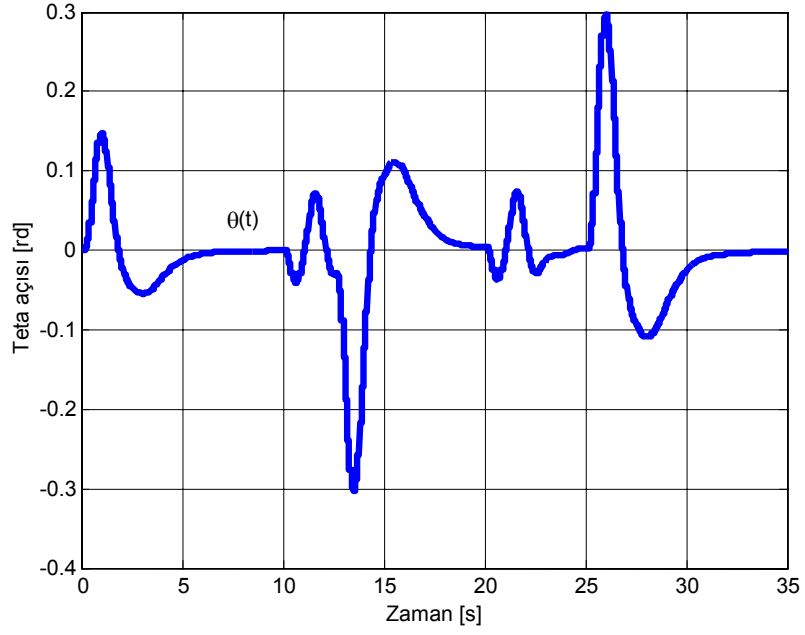
Sisteme belirli aralıklarda birer saniyelik 10N gücünde distürbans ilave edilirse sistem şekil(7.15) deki hali alacaktır. Şekil(7.31), (7.32) ve (7.33) de sistemin bu distürbansa karşı verdiği cevaplar elde edilmiş olup; görüldüğü gibi etkiyen dış kuvvet araba pozisyonunda küçük üst aşım sorunları yaratmakta; sarkaç salınımında ise maksimum salınımlara etki etmemekle birlikte sistemin salınım kontrolünü belirgin yönde etkilediği görülmüştür. Kutup yerleşim metoduna göre daha kötü sonuçlar almamızın sebebi olarak sistemin kendi içindeki kontrol kuvveti ile girilen distürbans değerinin birbirine daha yakın olması gösterilebilir.



Şekil 7.31 LQR kontrolü distürbans sonrası $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



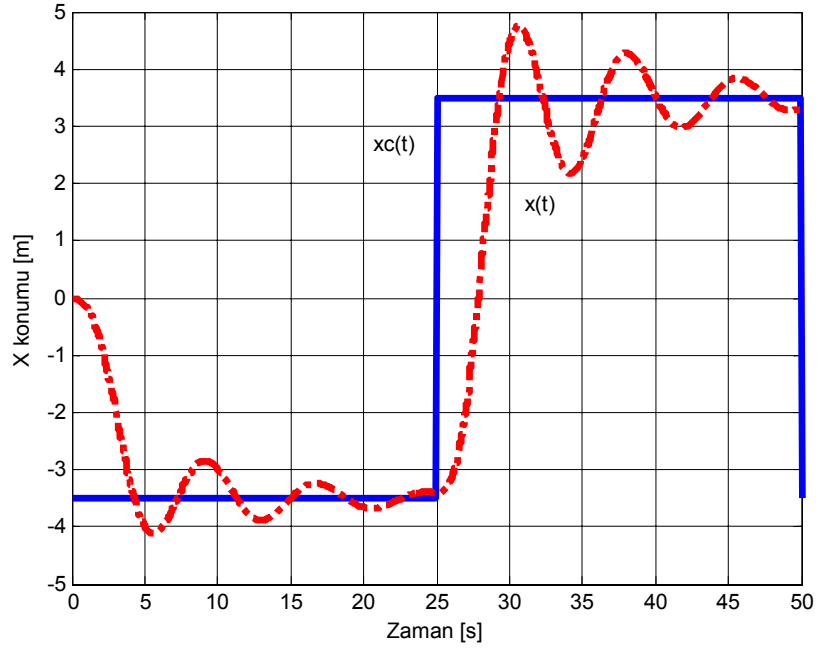
Şekil 7.32 LQR kontrolü distürbans-kontrol sinyali cevabı



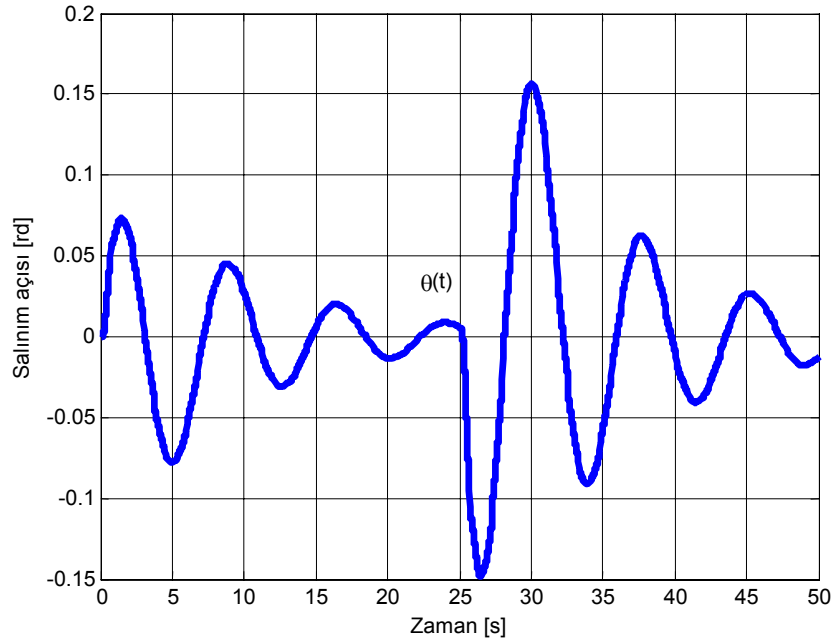
Şekil 7.33 LQR kontrolü distürbans sonrası *teta* açısı cevabı

7.5.4 Değişken Yüklerde Kutup Yerleşim Metodu Cevabı

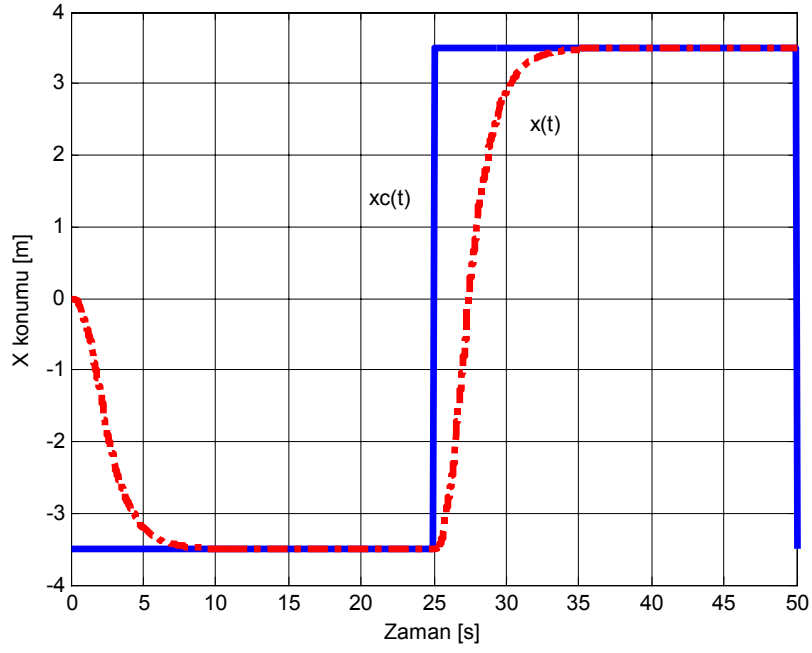
Eşitlik (7.49) da elde edilen K_{lqr} LQR metodu geri dönüş kazanç değeri, sistemde 5 kg lik sarkaç yükü olduğu kabulü üzerine bulunmuş bir değerdir. Fakat vinç sistemlerinde yükün çok farklı değerler aldığı bilinmektedir. Bu sebeple aynı K_{lqr} ve A integratör kazanç katsayıları kullanılarak sistemin değişken yüklerde cevabı incelenmiştir. Sisteme sırası ile 50; 0.5 ve 100 kg lik ağırlıklar koyulmuştur. Her bir yeni ağırlık için sistemin durum-uzay modelinde yeni A_d , B_d , C_d ve D_d katsayıları teker teker hesaplanarak Şekil (7.11) deki Simulink sistemde denenmiştir.



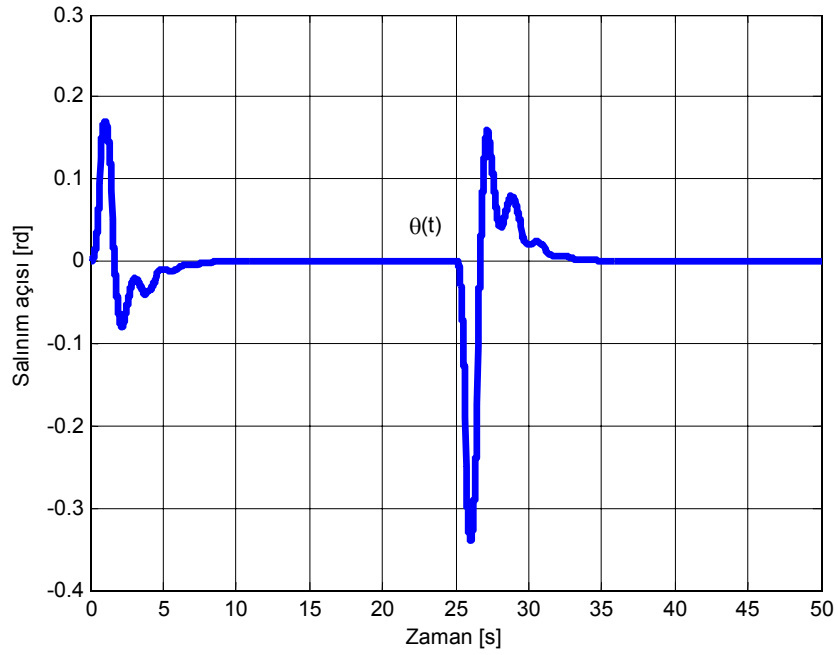
Şekil 7.34 LQR metodu 50 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



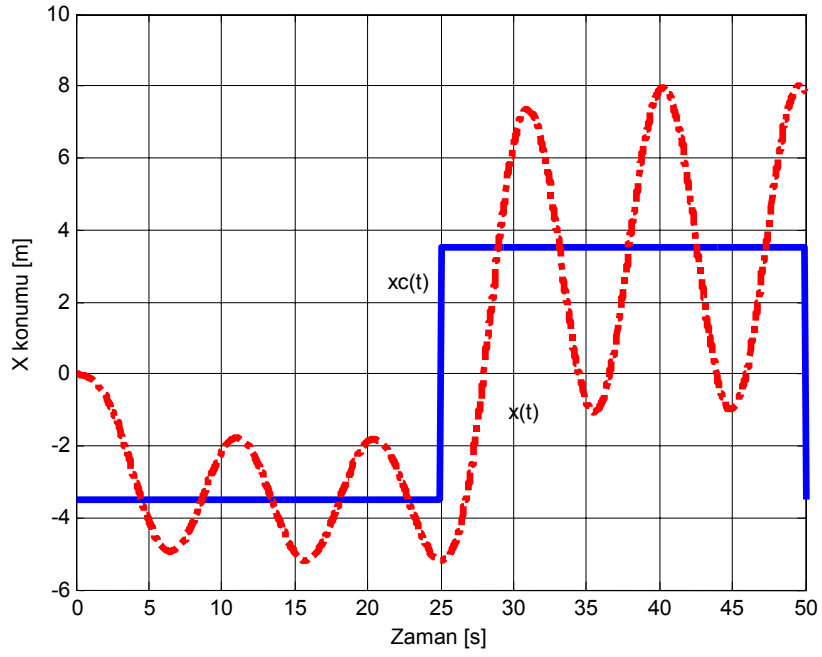
Şekil 7.35 LQR metodu 50 kg sarkaç yükünde $teta$ açısı cevabı



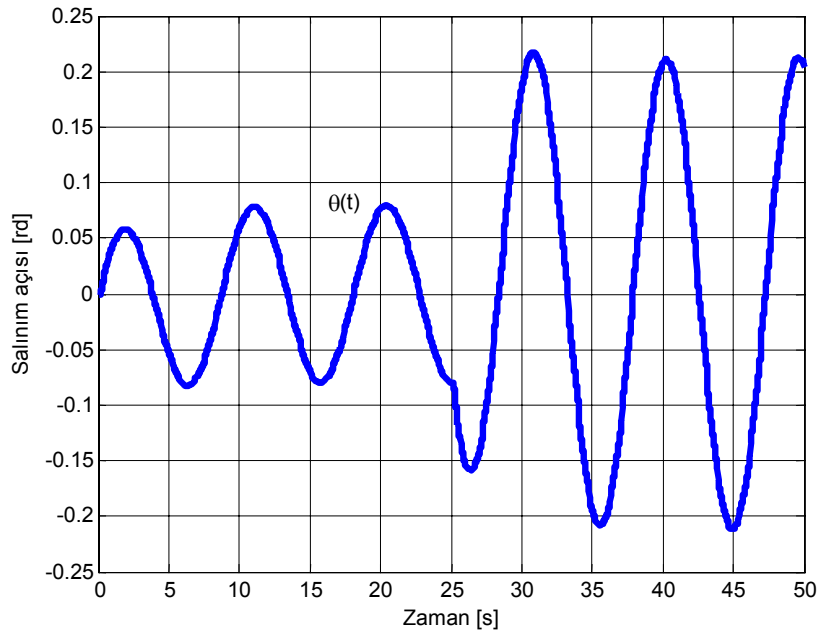
Şekil 7.36 LQR metodu $0,5 \text{ kg}$ sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.37 LQR metodu $0,5 \text{ kg}$ sarkaç yükünde $teta$ açısı cevabı



Şekil 7.38 LQR metodu 100 kg sarkaç yükünde $x_c(t)$ istenilen konum- $x(t)$ konumu cevabı



Şekil 7.39 LQR metodu 100 kg sarkaç yükünde $teta$ açısı cevabı

Grafiklerden görüldüğü gibi; LQR metodu hafif yüklerde pozisyonda bozulma olmadan sadece küçük salınım artımları ile sistemi kontrol edebilmekte; fakat yüklerin üzerinde

alıřılan 5 kg deęerini ařması ile pozisyonda bozulmalar bařladıęı ve buna iliřkili olarak salınım aısının da bozulmaya uęradıęı anlařılmıřtır. 100 kg aęırlıęında ise sistem tamamen sönümsüz hale gelmiřtir

8. SONUÇLAR

Yapılan çalışmada; bir robotik tepe vinci sisteminin 2 serbestlik derecesinde Lagrangian eşitlikleri kullanılarak matematiksel modeli elde edilmiştir. Bu model dijital durum-uzay modeline sokulduktan sonra, Simulink yazılımı yardımı ile üzerinde Kutup yerleşimi ve LQR kontrol yöntemleri denenerek sonuçları incelenmiştir.

İlk olarak sistemin tam durum gözlemcisi oluşturulmuştur. Kutup yerleşim metodu ile elde edilen geribesleme kazanç matrisi katsayılarının çok yüksek değerlerde olduğu gözlenmiştir.

Kazanç matrisi sistemin kontrol kuvvetini etkilediğinden; büyük kazanç matrisi katsayıları çok büyük giriş güçleri gerektirmektedirler. Böyle bir kontrol eforunu fiziksel bir sistemde uygulamak çok fazla enerji gerektirmekte ve hatta bazı durumlarda bu kuvvetleri elde etmek mümkün olamamaktadır. LQR metodunda bulunan katsayılar ise yaklaşık 5 de 1 oranında küçük olarak bulunmuş ve bu iki geri dönüşüm kazanç matrisi değerleri de uygulanan giriş argümanına karşı neredeyse aynı cevabı göstermişlerdir. LQR sisteminin sabit yüklerde çok daha başarılı olduğu görülmüştür. Fakat dışarıdan uygulanan distürlanslara karşı kutup yerleşim metodunun LQR sistemine göre çok daha stabil olduğu grafiklerden gözlenmiştir.

Ayrıca sistem değişken yükler altında da denenmiş; gene kazanç değerlerinin büyük değerde olmasından dolayı kutup yerleşim metodu üstünlük göstermiştir. Burada yeni sarkaç ağırlığı belirlenirken; sistemin tasarımında kullanılan 5 kg sarkaç ağırlığı %10-1000 oranında değiştirilmiştir. Yani sistemin 0.5 ve 50 kg olan yüklere karşı cevapları incelenmiş; son olarak 100kg değeri denenerek sistem hakkında daha geniş bilgi elde edilmiştir.

LQR metodunda ağırlıklar değiştirilerek daha iyi sonuçlar alınabilmektedir. Fakat ağırlıkların büyümesi kontrol eforunu da doğru orantılı olarak arttıracığından sistem davranışı kabul edilir değerlerde olan en küçük geri dönüş matrisi katsayıları seçilmiştir. Daha büyük değerlerde sistemin dış kuvvetlere karşı dayanımı artmaktadır.

LQR ve kutup yerleşim metotlarının farklarına da değinilmiştir. Kutup yerleşim metodunda; sistemin istenen kutup noktaları belirlenerek, sistemden arzulanan cevapların alınacağı öngörülmüştür. Sistemin bu belirlenen kutup noktalarında kutuplara sahip olmasına zorlanması sonucunda K_c değeri elde edilmiştir. Optimal yöntem olan LQR da ise sistem cevapları incelenerek iterasyonlar sonucu K_{lqr} değeri değiştirilerek katsayı bulunmaktadır.

KAYNAKLAR

Al-Mousa Amjed A., *Control of Rotary Cranes Using Fuzzy Logic and Time-Delayed Position Feedback Control* Virginia Polytechnic Institute and State University master thesis 2000

Anderson Brian D.O., John B. Moore *Optimal Control Linear Quadratic Methods* Prentice Hall 1989

Bagghi Arunabha, *Optimal Control of Stochastic Systems* Prentice Hall 1993

Benhidjeb A., Gissinger G.L., *Fuzzy Control of an Overhead Crane Performance Comparison with Classic Control*, Pergamon, Control Eng. Practice 1995

Chi-Cheng Cheng, Cheng Yi Chen, *Controller Design for an Overhead Crane System with Uncertainty*, Control Eng. Practice 1996

Costa Giuseppe *Robust Control For Gantry Cranes* University of New South Wales Sydney master thesis 1999

Halder Bibhrajit *Anti-swing Control Of a Suspended Varying Load With a Robotic Crane*, College Of Engineering and Technology of Ohio University 2002

Keller J.P., *Interactive Control System Design* Control Engineering Practice 14 2005

Kress R. L. *Recent Simulation Conclusions For Damped-Oscillation Control of Cranes*

Kuo Benjamin C. *Otomatik Kontrol Sistemleri Yedinci Baskı* Prentice Hall 1995

Kuo Benjamin C. *Linear Quadratic Regulator Design Example*

Lee, H.H., *Modeling and Control of a Three-Dimensional Overhead Crane* Journal of Dynamic Systems, Measurement and Control 1998

Mahfouf M., C. H. Kee, M. F. Abbod and D. A. Linkens, *Fuzzy Logic based Antiswing Control Design for Overhead Cranes* Neural Computing & Applications 2000

Maybeck Peter S., *Stochastic models, estimation and control volume 1* Academic Press 1979

Mendez J.A., L.Acosta, L. Moreno, S. Torres, G. N. Marichal, *An Application of a Neural Self-tuning Controller to an Overhead Crane* Neural Computing & Applications 1999

Miss Yang Rong, *Geometric Techniques for Control of a 2 DOF Spherical Inverted Pendulum* Honkong University of Science and Technology master thesis 2000

Nader A. Nayfeh, *Adaptation of Delayed Position Feedback to the Reduction of Sway of Container Cranes* Virginia Polytechnic Institute and State University master thesis 2002

Naidu Desineni Subbaram *Optimal Control Systems* CRC Press 2003

Noakes, M. W., R.L. Kress, G.T. Appleton, *Implementation of Damped-Oscillation Crane Control for Existing AC induction Motor-Driven Cranes* ANS fifth Topical meeting on Robotics and Remote System 1993

Ogata Katsuhiko, *Modern Control Engineering* Fourth Edition Prentice Hall 2002

Ogata Katsuhiko, *Designing Linear Control Systems With Matlab* Prentice Hall 1994

Ogata Katsuhiko, *Discrete-Time Control Systems Second Edition* Prentice Hall 1994

Parker, G.G., *Experimental Verification of a Command Shaping Boom Crane Control System* American Control Conference 1999

Phillips Charles L., Harbor Royce D. *Feedback Control Systems Fourth Edition* Prentice Hall 2000

Real-time Workshop For use with Simulink User's Guide, The Math Works, Inc. 1997

Sivritepe Murat, *The Stabilization of the Inverted Pendulum By Matlab Simulink* Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü 2003

Stimac Andrew K. *Standup and Stabilization of the Inverted Pendulum* Massachusetts Institute of Technology thesis 1999

Vaccaro Richard J., *Digital Control A State-Space Approach* McGraw-Hill 1995

Wen John T., Dan o . Popa, Gustavo Montemiyor, Peter L. Liu, *Human Assisted Impedance Control of Overhead Cranes*

Yoon Ji Sup, Byung Suk, *Various Control Schemes for Implementation of the Anti-Swing Crane* Korea Atomic Energy Research Institute 1994

Yüksel İbrahim, *Matlab ile Mühendislik Sistemlerinin Analizi ve Çözümü* Uludağ Üniversitesi Güçlendirme Vakfı Yayını 2000

<http://www.engin.umich.edu/group/ctm/> Control Tutorials For Matlab

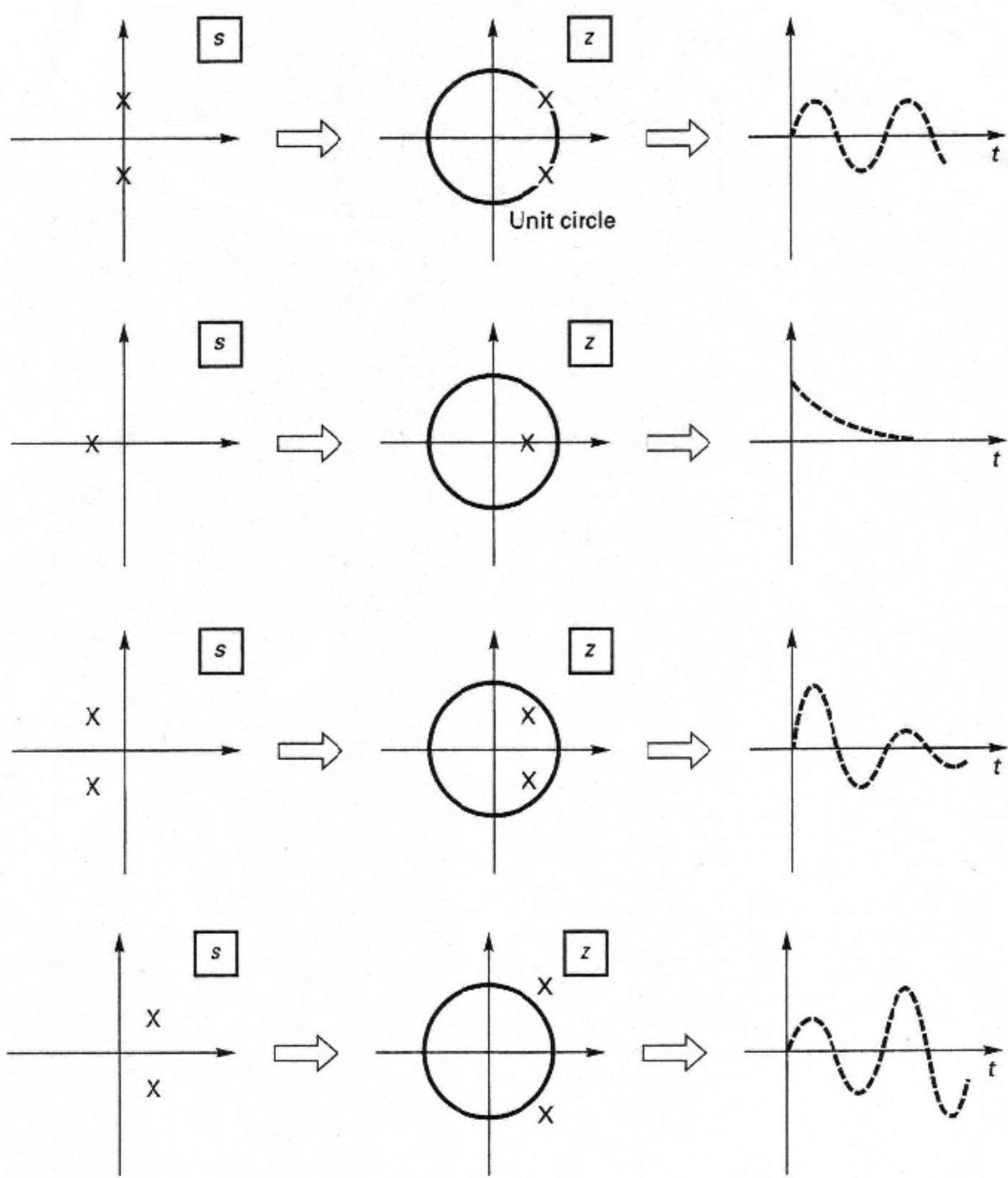
http://www.nyu.edu/classes/tuckerman/stat.mech/lectures/lecture_1/node3.html
The Lagrangian formulation of classical mechanics

http://ece.gmu.edu/~gbeale/ece_620/examples_620.html Optimal control examples

<http://www.ccec.ece.ucsb.edu/people/smith/> Lecture notes

<http://www.mathworks.com>

EKLER



Şekil 7.40 Örnek kutup noktaları ve zaman cevapları (Phillips 2000 sf. 534)

%%%

% VİNÇ TRANSFER FONKSİYONU ELDE EDİLMESİ; DURUM-UZAY GÖSTERİMİ

% VE DİJİTAL AYRIK ZAMANLI ŞEKLİ

clear

clc

%sistemin s domeni transfer fonksiyon modeli

% T1= teta/F

pay1=[-1];

payda1=[10 0 150];

T1=tf(pay1,payda1)

%T2=x/teta

pay2=[-1 0 -10];

payda2=[1 0 0];

T2=tf(pay2,payda2)

% Açık çevrim durum-uzay sürtünmesiz vinç sistemi

M = 10; m = 5;

l = 1; g = 10;

k1 = m*g/M

k2 = -(M+m)*g/(M*l);

k3 = 1/M;

k4 = -1/(M*l);

% Durum-uzay matris gösterimi

A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0]

B = [0; k3; 0; k4]

$$C = [1 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0]$$

$$D = 0$$

% Ayrık zamanlı durum-uzay modeli

$$T_e = 0.1;$$

$$[Ad, Bd, Cd, Dd] = c2dt(A, B, C, T_e, 0)$$

%%%

%%%

% GÖZLEMÇİ-TAHMİN EDİCİ DİZAYN EDİLMESİ

% Açık çevrim durum-uzay sürtünmesiz vinç sistemi

clear

clc

$$M = 10; m = 5;$$

$$l = 1; g = 10;$$

$$k1 = m * g / M;$$

$$k2 = -(M + m) * g / (M * l);$$

$$k3 = 1 / M;$$

$$k4 = -1 / (M * l);$$

% Durum-uzay matris gösterimi

$$A = [0 \ 1 \ 0 \ 0; 0 \ 0 \ k1 \ 0; 0 \ 0 \ 0 \ 1; 0 \ 0 \ k2 \ 0];$$

$$B = [0; k3; 0; k4];$$

$$C = [1 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0];$$

$$D = 0;$$

% Ayrık zamanlı durum-uzay modeli

```

Te=0.1;

[Ad,Bd,Cd,Dd]=c2dt(A,B,C,Te,0);

% Gözlenebilirlik (Observability) matrisi hesaplanması

Nobsd=obsv(Ad,Cd);

% Çıktı alınması

printmat(Nobsd,'gözlenebilirlik matrisi');

% Rank hesaplanması

rank_ = rank(Nobsd)

if rank(Nobsd)==4 disp('sistem gözlenebilirdir')

end;

% Prosesin karakteristik denkleminin elde edilmesi

phi=poly(Ad);

% Karakteristik denklemin kökleri

roots(phi);

% Durum tahmin edicisinin karakteristik denklemi

% Sönüm oranı

dr=sqrt(2)/2;

% Doğal frekans

wn=2*pi;

% Durum tahmin edicisi denklemi = p

p1 = -2*exp(-dr*wn*Te)*cos(wn*Te*sqrt(1-dr^2));

p2 = exp(-2*dr*wn*Te);

p = [0 0 1 p1 p2];

printmat(p,'Durum tahmin edicisi =')

```



```

% Sistemin kotroledilebilirliğinin incelenmesi

% Kontrol edilebilirlik matrisi

Mcontd=ctrb(Ad,Bd)

% Rank hesaplanması

rank__=rank(Mcontd);

printmat(Mcontd,'Kontroledilebilirlik matrisi =')

printmat(rank__,'rank=')

if rank(Mcontd)==4 disp('sistem kotroledilebilirdir; keyfi kutup yerleşimi yapılabilir')

else disp('sistem kontrol edilemez')

end

% Luenberger yöntemi:

% T transformasyon matrisi

n = 4;

T(:,4) = Bd;

for i = 1:n-1,

    T(:,n-i) = Ad*T(:,n-i+1)+phi(n-i+1)*Bd;

end

printmat(T,'T Transformasyon Matrisi =')

% Kobs bulunması

% Tahminedicici kazancı

kobs=[p(1,5)-phi(1,5) p(1,4)-phi(1,4) p(1,3)-phi(1,3) p(1,2)-phi(1,2)];

kobsc = kobs*inv(T);

kobsc(2,:) = kobsc(1,:);

printmat(kobsc,'tahminedicici kazanc matrisi Kobsc')

```

%%%

%%%

% KUTUP YERLEŞİM METODU

% Açık çevrim durum-uzay sürtünmesiz vinç sistemi

clear;

clc;

M = 10; m = 5;

l = 1; g = 10;

k1 = m*g/M;

k2 = -(M+m)*g/(M*l);

k3 = 1/M;

k4 = -1/(M*l);

% Durum-uzay matris gösterimi

A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0];

B = [0; k3; 0; k4];

C = [1 0 0 0; 0 0 1 0];

D = 0;

% Ayrık zamanlı durum uzay modeli

Te=0.1;

[Ad,Bd,Cd,Dd]=c2dt(A,B,C,Te,0);

% Seçilen yeni kutup noktaları

Da=[0.6 0.6 0.6 0.6]

%bu seçilen kutup noktalarındaki sistemin karakteristik denklemi

```

phic=poly(Da);

printmat(phic,' Yeni kutup noktalarının karakteristik denklemi =')

% Orijinal sistemin kontroledilebilirliği

Mcontd=ctrb(Ad,Bd)

Rank=rank(Mcontd)

% Gözlemci dizaynında hesaplanmış olan

phi = poly(Ad); %orijinal sistem

% T transformasyon matrisi

n = 4;

T(:,4) = Bd;

for i = 1:n-1,

    T(:,n-i) = Ad*T(:,n-i+1)+phi(n-i+1)*Bd;

end

printmat(T,' T Transformasyon Matrisi =')

% Durum geribesleme katsayılarının bulunması

n = 4;      % sistemin derecesi

phi = poly(Ad); %orijinal sistem

phic = poly(Da); %arzulanan sistem

for i = 1:n

    K(1,n-i+1) = phic(1,i+1)-phi(1,i+1);

end

Kc = K*inv(T);

printmat(Kc, 'durum geribesleme katsayısı Kc matrisi')

Ac=[Ad-Bd*Kc]; % yeni A değeri

```

% Kutup yerleşimi simülasyonu% step giriş

zaman= 0:0.01:5;

U=ones(size(zaman));

[y1,x1]=dlsim(Ac,Bd,Cd,Dd,U);

figure(10);

hold on;

plot(zaman,y1);

grid on

zoom;

%%%

%%%

%LQR KONTROL AĞIRLIKLARA GÖRE SİSTEM CEVABI (ÜST AŞIM VE SÖNÜM
%ORANI) VE KLQR KAZANÇ DEĞERİ EXCEL E ATILMASI

% Sistem modeli

clear

clc

M = 10; m = 5;

d = 1;

g = 10;

k1 = m*g/M;

k2 = -(M+m)*g/(M*d);

k3 = 1/M;

k4 = -1/(M*d);

Te = 0.1;

Dd = 0;

A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0];

B = [0; k3; 0; k4];

C = [1 0 0 0; 0 0 1 0];

D=0;

%dijital

[Ad,Bd,Cd,Dd] = c2dt(A,B,C,Te,0);

%%%lqr loop

T=0:0.01:5;

U=ones(size(T));

Data=zeros(60,4); %60*4 lük sıfır matris

%kx ve kteta teker teker deneniyor

x=0

for i=5000:200:10000; %1, sütun

for j=5000:500:25000; %2. sütun

x=x+1

Data(x,1)=x;

Data(x,2)=i;

Data(x,3)=j;

%

kteta=j;

kx=i;

Q=[kx 0 0 0;0 0 0 0;0 0 kteta 0;0 0 0 1];

```

R=1;

kctm=dlqr(Ad,Bd,Q,R);

Ac=[Ad-Bd*kctm];% Geribesleme sonrası yeni A değeri

% LQR simülasyonu % step giriş

[y1,x1]=dlsim(Ac,Bd,Cd,Dd,U);

maks=max(y1);

maksx=maks(1,1);

maksteta=maks(1,2);

Mpx=(maksx-y1(500,1));

Mpteta=(maksteta-y1(500,2)) ;

Data(x,4)=Mpx;

Data(x,5)=Mpteta;

% Oturma zamanı ts

settling_percent = 0.0002; % tanım aralığı

chan1x = y1(:,1); % basamak girişinin konum cevabı

maxtimex = max(find(abs((chan1x - chan1x(end)) / chan1x(end) > settling_percent)));

% son değer 0.0002 den büyük olan değerdeki zaman

settling_time_x = 0.01*(maxtimex + 1);

% 1 sonraki zaman ts dir

Data(x,6)=settling_time_x; % x oturma zamanı

chan1teta = y1(:,2); % basamak girişinin teta cevabı

maxtimeteta = max(find(abs((chan1teta - chan1teta(end)) / chan1teta(end) >
settling_percent)));

```

% son deęerin 0.0002 den byk olan deęerdeki zaman

settling_time_teta = 0.01*(maxtmeteta + 1);

% 1 sonraki zaman ts dir.

Data(x,7)=settling_time_teta; % teta oturma zamanı

% kctm kazanç deęeri

Data(x,9)=kctm(1,1);

Data(x,10)=kctm(1,2);

Data(x,11)=kctm(1,3);

Data(x,12)=kctm(1,4);

end

end

Data

xlswrite(Data,'dongu_data')

%%%

%%%

% LQR KONTROL

% SEÇİLEN 5200-14500 Q AĖIRLIKLARI İÇİN LQR DİZAYNI

% Discrete kontrol

% Duru-uzay gsterimi

clear

clc

$$M = 10; m = 5;$$

$$d = 1;$$

$$g = 10;$$

$$k1 = m*g/M;$$

$$k2 = -(M+m)*g/(M*d);$$

$$k3 = 1/M;$$

$$k4 = -1/(M*d);$$

$$Te = 0.1;$$

$$Dd = 0;$$

$$A = [0 \ 1 \ 0 \ 0; 0 \ 0 \ k1 \ 0; 0 \ 0 \ 0 \ 1; 0 \ 0 \ k2 \ 0];$$

$$B = [0; k3; 0; k4];$$

$$C = [1 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0];$$

$$D=0;$$

% Dijital

$$[Ad,Bd,Cd,Dd] = c2dt(A,B,C,Te,0);$$

% Kontrol edilebilirlik -gözlenebilirlik

$$contd=ctrb(Ad,Bd);$$

$$rank_contd=rank(contd)$$

$$Nobsd=obsv(Ad,Cd);$$

$$rank_Nobsd=rank(Nobsd)$$

% Proses karakteristik denklemi

$$phi = poly(Ad);$$

% Stabilite%

$$poles=eig(Ad) \quad \% \text{ roots(phi) ile de bulunabiliyor}$$


```
disp('Z domeninde tüm kutuplar pozitif ve birim çemberin üzerinde; sistem sönümsüz')
```

```
% Açık çevrim basamak girişi
```

```
dstep(Ad,Bd,Cd,Dd);
```

```
%%% lqr bulunması
```

```
T=0:0.01:5;
```

```
U=ones(size(T));
```

```
Q=[5200 0 0 0;0 0 0 0;0 0 14500 0;0 0 0 1] % ağırlıklar
```

```
R=1
```

```
kctm=dlqr(Ad,Bd,Q,R)
```

```
Ac=[Ad-Bd*kctm]; % yeni A değeri
```

```
% LQR simülasyonu% step giriş
```

```
[y1,x1]=dlsim(Ac,Bd,Cd,Dd,U);
```

```
figure(23);
```

```
hold on;
```

```
plot(T,y1);
```

```
grid on
```

```
zoom;
```

```
% Maksimum değerlerin bulunması
```

```
maks=max(y1);
```

```
maksx=maks(1,1);
```

```
maksteta=maks(1,2);
```

```
Mpx=(maksx-y1(500,1));
```

```
Mpteta=(maksteta-y1(500,2)) ;
```

```
Mpx
```

Mpteta

% ts bulunması

settling_percent = 0.0002; % tanım aralığı

chan1x = y1(:,1); % basamak girişinin konum cevabı

maxtimex = max(find(abs((chan1x - chan1x(end)) / chan1x(end) > settling_percent)));

% son değer 0.0002 den büyük olan değerdeki zaman

settling_time_x = 0.01*(maxtimex + 1);

% 1 sonraki zaman ts dir

xlswrite(settling_time_x,'ts_x'); % x

chan1teta = y1(:,2); % basamak girişinin teta cevabı

maxtimeteta = max(find(abs((chan1teta - chan1teta(end)) / chan1teta(end) > settling_percent)));

% son değer 0.0002 den büyük olan değerdeki zaman

settling_time_teta = 0.01*(maxtimeteta + 1);

% 1 sonraki zaman ts dir

xlswrite(settling_time_teta,'ts_teta'); % teta

% Excel e yazdırma

xlswrite(y1,'lqr_grafikler')

%%%

%%%

% DEĞİŞKEN YÜK-SIMULINK SIMULASYONU İÇİN KATSAYILAR

% İnitilize

% Model durum-uzay gösterimi

```
clear
```

```
clc
```

```
% 50kg ağırlık için durum-uzay modeli
```

```
M = 10; m = 50
```

```
d = 1;
```

```
g = 10;
```

```
k1 = m*g/M;
```

```
k2 = -(M+m)*g/(M*d);
```

```
k3 = 1/M;
```

```
k4 = -1/(M*d);
```

```
Te = 0.1;
```

```
Dd = 0;
```

```
A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0 ];
```

```
B = [0; k3; 0; k4];
```

```
C = [1 0 0 0; 0 0 1 0];
```

```
[Ad,Bd,Cd,Dd] = c2dt(A,B,C,Te,0)
```

```
% 0.5kg ağırlık için durum-uzay modeli
```

```
m = 0.5
```

```
k1 = m*g/M;
```

```
k2 = -(M+m)*g/(M*d);
```

```
k3 = 1/M;
```

```
k4 = -1/(M*d);
```

```
Te = 0.1;
```

```
Dd = 0;
```

$$A = [0 \ 1 \ 0 \ 0 ; 0 \ 0 \ k_1 \ 0 ; 0 \ 0 \ 0 \ 1 ; 0 \ 0 \ k_2 \ 0];$$

$$B = [0; k_3; 0; k_4];$$

$$C = [1 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0];$$

$$[Ad, Bd, Cd, Dd] = c2dt(A, B, C, Te, 0)$$

% 100kg ağırlık için durum-uzay modeli

$$m = 100$$

$$k_1 = m \cdot g / M;$$

$$k_2 = -(M+m) \cdot g / (M \cdot d);$$

$$k_3 = 1/M;$$

$$k_4 = -1/(M \cdot d);$$

$$Te = 0.1;$$

$$Dd = 0;$$

$$A = [0 \ 1 \ 0 \ 0 ; 0 \ 0 \ k_1 \ 0 ; 0 \ 0 \ 0 \ 1 ; 0 \ 0 \ k_2 \ 0];$$

$$B = [0; k_3; 0; k_4];$$

$$C = [1 \ 0 \ 0 \ 0; 0 \ 0 \ 1 \ 0];$$

$$[Ad, Bd, Cd, Dd] = c2dt(A, B, C, Te, 0)$$

%%%

%%%

% SIMULINK SİMULASYONU İÇİN KATSAYILAR:

% İnitilize

%%% Model durum-uzay gösterimi %%

% Model durum-uzay gösterimi

clear

```

clc

M = 10; m = 5;

d = 1;

g = 10;

k1 = m*g/M;

k2 = -(M+m)*g/(M*d);

k3 = 1/M;

k4 = -1/(M*d);

Te = 0.1;

Dd = 0;

A = [0 1 0 0 ; 0 0 k1 0 ; 0 0 0 1 ; 0 0 k2 0 ];

B = [0; k3; 0; k4];

C = [1 0 0 0; 0 0 1 0];

[Ad,Bd,Cd,Dd] = c2dt(A,B,C,Te,0)

% Prosesin karakteristik denkleminin elde edilmesi

phi=poly(Ad); % Orijinal sistem

% Karakteristik denklemin kökleri

roots(phi);

% Gözlenebilirlik (Observability) matrisi hesaplanması

Nobsd=obsv(Ad,Cd);

% Çıktı alınması

printmat(Nobsd,'gözlenebilirlik matrisi Nobsd');

% Kontrol edilebilirlik matrisi

```

```

Mcontd=ctrb(Ad,Bd);

% Çıktı alınması

printmat(Mcontd,'kontrol edilebilirlik matrisi Mcontd ');

%%%%%%%%%%%% Gözlemci Sistemi %%%%%%%%%%%%%

disp('GÖZLEMCI DİZAYNI')

% Durum tahmin edicisinin karakteristik denklemi

% istenen gözlemci karakteristikleri

% Sönüm oranı

dr=sqrt(2)/2;

% Doğal frekans

wn=2*pi;

% Durum tahmin edicisi (gözlemcisi) denklemi = p

p1 = -2*exp(-dr*wn*Te)*cos(wn*Te*sqrt(1-dr^2));

p2 = exp(-2*dr*wn*Te);

p = [0 0 1 p1 p2];

printmat(p,'p Durum tahmin edicisi =')

% Luenberger yöntemi:

% T transformasyon matrisi

n = 4;

T(:,4) = Bd;

for i = 1:n-1,

    T(:,n-i) = Ad*T(:,n-i+1)+phi(n-i+1)*Bd;

end

printmat(T,' T Transformasyon Matrisi =')

```

```

% Kobs bulunması

% Tahminedicici kazancı

kobs=[p(1,5)-phi(1,5) p(1,4)-phi(1,4) p(1,3)-phi(1,3) p(1,2)-phi(1,2)];

kobsc = kobs*inv(T);

kobsc(2,:) = kobsc(1,:);

printmat(kobsc,'tahminedicici kazanc matrisi kobsc')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Kutup yerleşimi Sistemi %%%%%%%%%%%%%%%

disp('KUTUP YERLEŞİMİ')

% Kutup yerleşimi için seçilen yeni kutup noktaları

Da=[0.6 0.6 0.6 0.6];

printmat(Da,' Yeni kutup noktaları =')

% Bu seçilen kutup noktalarındaki sistemin karakteristik denklemi

phic=poly(Da); % Arzulanan sistem

printmat(phic,' Yeni kutup noktalarının karakteristik denklemi =')

% Durum geribesleme katsayılarının bulunması

n = 4;      % sistemin derecesi

for i = 1:n

    K(1,n-i+1) = phic(1,i+1)-phi(1,i+1);

end

Kc = K*inv(T);

printmat(Kc, 'durum geribesleme katsayısı Kc matrisi')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% LQR metodu %%%%%%%%%%%%%%%

disp('LQR METODU')

% Seçilen ağırlıklar

```

```
Q=[5200 0 0 0;0 0 0 0;0 0 14500 0;0 0 0 1]
```

```
R=1
```

```
kctm=dlqr(A,B,Q,R)
```

```
%Kod sonu
```


ÖZGEÇMİŞ

Adı soyadı	Kaan KANDEMİR	
Doğum tarihi	16.04.1981	
Doğum yeri	İSTANBUL	
İlkokul	1988-1992	Nişantaşı Işık Lisesi İlkokulu
Lise	1992-1999	Büyükşehir Anadolu Lisesi
Lisans	1999-2003	Yıldız Teknik Üniversitesi Elektrik Elektronik Fakültesi Elektrik Mühendisliği Bölümü Kontrol ve Kumanda Anabilim Dalı
Staj ve iş tecrübesi	BEDAŞ- Boğaziçi Elektrik Dağıtım A.Ş.(Staj)-İstanbul 2000 TELEKOM- Türk Telekomünikasyon A.Ş.(Staj)-İstanbul 2001 SAVEN Otomasyon ltd. (Electrical Eng.) -İstanbul 2003-2004	