

21718

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BİR ROBOTİK MANİPÜLATÖRÜN EKLEM VE
KARTEZYEN ESASLI ÖNGÖRÜLÜ KONTROLU

DOKTORA TEZİ

Y. Müh. Recep KAZAN

Tezin Enstitüye Verildiği Tarih : 25 Şubat 1992

Tezin Savunulduğu Tarih : 13 Temmuz 1992

Tez Danışmanı : Doç.Dr. Can ÖZSOY

Diğer Jüri Üyeleri : Prof.Dr. Ahmet KUZUCU

Yrd.Doç.Dr. Kemal CILIZ

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

TEMMUZ 1992

ÖNSÖZ

Sunulan bu çalışmada uzun menzilli öngörülü kontrol algoritmalarından olan genelleştirilmiş öngörülü kontrol algoritması tek giriş-tek çıkışlı ve çok giriş-çok çıkışlı durumlar için üç uzuvlu bir robotik manipülatöre uyarlanmıştır. Ayrıca yorucu hesaplamalara gereksinme gösteren ters kinematik çözümü ortadan kaldıran ve manipülatörün uç elemanının direkt olarak kartezyen uzayda kontrolünü sağlayan yeni bir model önerilmiştir.

Bana bu çalışma olanağını sağlayan ve devamlı olarak destekleyerek yönlendiren değerli hocam Doç. Dr. Can ÖZSOY' a, çalışmalarımda destek gördüğüm İ.T.Ü. Sakarya Mühendislik Fakültesi Dekanı sayın Prof. Dr. İnal SEÇKİN'e ve Makina Bölüm Başkanı sayın Prof. Dr. Atilla BOZACI' ya, başta Sayın Prof. Dr. Talha DİNİBUTUN ve Prof. Dr. Ahmet KUZUCU olmak üzere İ.T.Ü. Otomatik Kontrol Birimi tüm çalışanlarına, yardımlarını gördüğüm tüm arkadaşlarıma ve de çalışmalarına büyük bir özveri ile katlanan eşime teşekkürlerimi sunarım.

ADAPAZARI, 25 . 2. 1992

Y. Müh. Recep KAZAN

İÇİNDEKİLER

SEMBOLLER	Vii
ÖZET	1
SUMMARY	ix
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. ADAPTİF KONTROL SİSTEMLERİ	15
2.1. Giriş	15
2.2. Adaptif Kontrolün Temel Yapısı	18
2.2.1. Model Referans Adaptif Kontrol	19
2.2.2. Kendi Kendini Ayarlayan Regülatörler ...	21
2.3. Kendi Kendini Ayarlayan Kontrol	
Algoritmaları	23
2.3.1. Kendi Kendini Ayarlayan Regülatörler	
(SISO)	24
2.3.1.1. Minimum-Varyans Kendi Kendini	
Ayarlayan Regülatörler	24
2.3.1.2. Direkt (Implicit) ve Endirekt (Explicit)	
Kendi Kendini Ayarlama	26
2.3.1.3. Pol Atamalı Kendi Kendini Ayarlayan	
Regülatörler	28
2.3.1.4. LQG Kendi Kendini Ayarlayan	
Regülatörler	29
2.3.1.5. Non-Minimum Faz Sistemler için Kendi	
Kendini Ayarlayan Regülatörler	30
2.3.1.6. Karma (Hybrid) Kendi Kendini Ayarlama.	30
2.3.2. Kendi Kendini Ayarlayan Regülatörler	
(MIMO)	31
2.4. Çok değişkenli Kendi Kendini Ayarlayan	
Bir Regülatör Dizaynı	33
2.4.1. Genel Bilgiler	33
2.4.2. Problemin Formülasyonu	33
2.5. Parametre Kestirme Algoritmaları	38
2.5.1. Genel Bilgiler	38
2.5.2. Problemin Formülasyonu	39
2.5.3. Ardışık Parametre Kestirme Metodları ...	40
2.5.3.1. Ardışık En Küçük Kareler Metodu	40
2.5.3.2. Ardışık Kare Kök Algoritması	41
BÖLÜM 3. UZUN MENZİLLİ ÖNGÖRÜLÜ KONTROL ALGORİTMALARI	43
3.1. Giriş	43
3.2. Yapılan Çalışmalar	46
3.3. Değişik Uzun Menzilli Öngörülü Kontrol	
Algoritmaları	48
3.3.1. Model Algoritmik Kontrol (MAC)	48

3.3.2. Dinamik Matris Kontrolü (DMC) Metodu ...	53
3.3.3. EPSAC Metodu	56
3.3.4. EHAC Metodu	59
3.3.5. GPC Metodu	61
BÖLÜM 4. GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROL (GPC)	63
4.1. Tek Giriş-Tek Çıkışlı (SISO) Genelleştirilmiş Öngörülü Kontrol Algoritması	63
4.1.1. Genel Bilgiler	63
4.1.2. Öngörü Kavramı	65
4.1.3. Diophantine Denkleminin Ardışık Hale Getirilmesi	68
4.1.4. Öngörülü Kontrol Kanunu	70
4.1.5. Kontrol Ufuku	76
4.1.6. Çıkış ve Kontrol Ufuklarının Seçimi	78
4.1.7. GPC' nin Özellikleri ve Diğer Yaklaşımlar ile İlişkisi	79
4.2. Çok Giriş-Çok Çıkışlı (MIMO) Genelleştirilmiş Öngörülü Kontrol Algoritması	82
4.3. Genelleştirilmiş Öngörülü Kontrol Akış Şeması	84
BÖLÜM 5. GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROL ALGORİTMASI İLE İLGİLİ GENEL SİMÜLASYON ÇALIŞMALARI	87
5.1. SISO Simülasyon Çalışmaları	87
5.1.1. SISO Genelleştirilmiş Öngörülü Kontrol Algoritmasında Ayar Parametrelerinin Etkisi	87
5.1.2. Farklı Sistemler Karşısında Genelleştirilmiş Öngörülü Kontrol Algoritması	94
5.1.2.1. Ayırık Zamanlı Deterministik Sistemler	94
5.1.2.2. Ayırık Zamanlı Stokastik Sistemler	95
5.1.2.3. Sürekli Zamanlı Sistemler-Hatalı Model Mertebesi	102
5.1.2.4. Sürekli Zamanlı ve Zamanla Değişen Sistemler	105
5.2. MIMO Simülasyon Çalışmaları	106
5.3. Sonuçlar	116
5.3.1. SISO Genelleştirilmiş Öngörülü Kontrol Algoritmasında Ayar Parametrelerinin Etkisi	116
5.3.2. Farklı Sistemler Karşısında Genelleştirilmiş Öngörülü Kontrol Algoritması	117
5.3.2.1. Ayırık Zamanlı Deterministik Sistemler	117
5.3.2.2. Ayırık Zamanlı Stokastik Sistemler ...	118
5.3.2.3. Sürekli Zamanlı Sistemler-Hatalı Model Mertebesi	118
5.3.2.4. Sürekli Zamanlı ve Zamanla Değişen Sistemler	
5.3.3. MIMO Sistemler için Simülasyon Sonuçları	119

BÖLÜM 6. ROBOT KOLU KİNEMATİĞİ VE DİNAMİĞİ	120
6.1. Robot Kolu Kinematığı	120
6.2. Robot Kolu Dinamiğı	126
6.2.1. Giriş	126
6.2.2. Hareketin L-E Denklemlerinin Çıkarılması	128
6.2.3. Bir Robot Kolunun Hızı	129
6.2.4. Bir Robot Kolunun Kinetik Enerjisi	133
6.2.5. Bir Robot Kolunun Potansiyel enerjisi	137
6.2.6. Hareketin L-E Denklemleri	137
BÖLÜM 7. ROBOT SİMÜLASYONU	140
7.1. Genel Bilgiler	140
7.2. Kullanılan Robot Modeli	143
7.2.1. Robot Modelinin Jacobian Matrisinin Teşkili	143
7.2.2. Sımülasyonda Kullanılan Eklem Referans Yörüngeleri	147
7.3. Robot Kontrolu İçin Kendi Kendini Ayarlayan Kontrolör Dizaynı	149
7.3.1. Sayısal Örnekler	151
7.4. Öngörülü Kontrol Algoritmasının Robotik Bir Manipülatöre Uygulanması	160
7.4.1. SISO Öngörülü Kontrolör Dizaynı	160
7.4.1.1. Sayısal Örnekler	164
7.4.2. MIMO Öngörülü Kontrolör Dizaynı	176
7.4.2.1. Sayısal Örnekler	178
7.5. Kartezyen Uzayda Öngörülü Kontrolör Dizaynı	180
7.5.1. Kartezyen Uzayda Kullanılacak Ayrık Zaman Modelinin Teşkili	190
7.5.2. Kartezyen Uzayda Geliştirilen Ayrık Zaman Modelinin Robotik Bir Manipülatöre Uygulanması	192
7.5.3. Sayısal Örnekler	195
GENEL SONUÇLAR VE ÖNERİLER	208
KAYNAKLAR	216
EKLER	227
EK A. MIMO Sistemler için Öngörülü Kontrol Algoritmalarında Kullanılan E, F ve G Polinom Parametrelerinin Hesaplanması ve Kontrol Büyüklüğünün Teşkili	227
EK A.1. E Polinom Parametrelerinin Çıkarılması	227
EK A.2. F Polinom Parametrelerinin Çıkarılması	229
EK A.3. G Polinom Parametrelerinin Çıkarılması	230
EK A.4. u Kontrol Büyüklüğünün Teşkil Edilmesi	232
EK B. Robot Yörünge Kontrolu Sımülasyon Programları	237
EK B.1. Robot Yörünge Kontrolu Sımülasyon Programları Genel Akış Şeması	237

EK B.2. Robot Yörünge Kontrolu Simülasyon	
Programlarının Tanıtılması	238
EK B.2.1. Ana Program	238
EK B.2.2. Girdiler	238
EK B.2.3. Yörünge Hesabı	239
EK B.2.4. RUNGE Alt Programı	239
EK B.2.5. Dinamik Davranış (SYST) Alt Programı	240
EK B.2.6. Kontrol Sinyalinin Hesaplanması	240
EK B.2.7. Çıkış Dosyaları	241
EK B.2.8. Çok Giriş-Çok Çıkışlı Öngörülü Robot	
Kontrolu İçin Kullanılan Bilgisayar	
Programlarının Detaylı Akış Şeması ..	242
EK B.3. FORTRAN Dilinde Yazılmış Robot	
Yörünge Kontrolu Simülasyon Programı..	245
ÖZGEÇMİŞ	268

SEMBOLLER

- d : Proses Zaman Gecikmesi
 u : Proses Girişi
 y : Proses Çıkışı
 $\hat{y}$: Öngörülen Proses Çıkışı
 y_r : Proses Çıkışı Referans Değerleri
 ξ : Gürültü Bileşeni
 $A(q^{-1}), B(q^{-1}), C(q^{-1})$: Proses Modeli Katsayı Polinomları
 q^{-1} : Geriye Doğru Kaydırma operatörü
 Δ : Fark Alma Operatörü ($\Delta u : u(t) - u(t-1)$)
 n_a, n_b, n_c : Polinom Mertebeleri
 E : Beklenti Operatörü
 Q_1, Q_2 : Ceza Matrisleri
 μ : Unutma Faktörü
 e : Hata
 K : Kazanç
 N_1 : Başlangıç Ufuzu (Minimum Maliyet Ufuzu)
 N_2 : Öngörü Ufuzu (Maksimum Maliyet Ufuzu)
 N_u : Kontrol Ufuzu
 λ : Kontrol Ağırlığı Faktörü
 h : İmpuls Cevabı Katsayıları
 a : Basamak Cevabı Katsayıları
 τ : Zaman Sabiti
 T_s : Örnekleme Peryodu
 E, F, G : Genelleştirilmiş Öngörülü Kontrolör Parametre Polinomları
 q : Manipülatör Eklem Değişkenleri
 θ_i : Eklem Dönme Açısı (radyan)
 $\dot{\theta}_i$: Eklem Dönme Hızı (rad/sn)
 $\ddot{\theta}_i$: Eklem Dönme İvmesi (rad/sn²)
 $\theta_i, d_i, a_i, \alpha_i$: Eklem Parametreleri
 A_{i-1}^t : Homojen Koordinat Dönüşüm Matrisi
 R_{i-1}^t : Koordinat Rotasyon Matrisi

$\vec{n}$: Elin Birim Normal Vektörü
 $\vec{s}$: Elin Birim Kayma Vektörü
 $\vec{a}$: Elin Birim Yaklaşma Vektörü
 $\vec{p}$: Elin Pozisyon Vektörü
 L : Lagrangian Fonksiyonu
 K : Kinetik Enerji
 P : Potansiyel Enerji
 J_i : i Uzunluğunun Atalet Matrisi
 k_{ijk} : j - k Eksen Etrafında i Uzunluğunun Jirasyon Yarıçapı
 m_i : Uzun Kütlesi
 m : Robot Tarafından Taşınan Yük
 $\bar{x}_i, \bar{y}_i, \bar{z}_i$: i Uzunluğunun Kütle Merkezinin Koordinatları
 g : Yerçekimi İvmesi ($g=9.8062 \text{ m/s}^2$)
 $\tau(t)$: Eklem Tork Vektörü (N.m)
 $D(\theta)$: İvmelenme ile İlgili Simetrik Atalet Matrisi
 $H(\theta, \dot{\theta})$: Koriyolis ve Merkezkaç Kuvvet Vektörü
 $G(\theta)$: Yerçekimi Kuvveti Vektörü
 $S(\theta)$: Sürtünme Tork Vektörü
 $J(\theta)$: Manipülatör Jacobian Matrisi
 x, y, z : Uç Elemanın Kartezyen Koordinatları
 l_i : Her Bir Robot Uzunluğunun Uzunluğu

ÖZET

Bu çalışmada öngörülü kontrol ailesinden olan genelleştirilmiş öngörülü kontrol algoritmasının üç eklemlili bir robotik manipülatöre değişik çalışma şartlarında uyarlanması incelenmiştir.

Uzun menzilli öngörülü kontrol algoritmalarının teorisini daha iyi anlayabilmek için adaptif kontrol algoritmaları ve bu konuda yapılan çalışmalar 2. bölümde geniş bir şekilde araştırılmış, 3. bölümde de öngörülü kontrol algoritmaları genel özellikleri ile incelenmiştir. Öngörülü kontrol algoritmaları içerisinde değişik açılardan en etkilisinin genelleştirilmiş öngörülü kontrol algoritması olduğuna karar verildikten sonra 4. bölümde genelleştirilmiş öngörülü kontrol algoritmasının teorisi geniş bir şekilde çıkarılmıştır. 5. bölümde ise genelleştirilmiş öngörülü kontrol algoritması değişik sistemler için performans araştırması yapmak amacı ile kişisel bilgisayarlar ile simülasyon çalışmasına tabii tutulmuş, sonuçlar detaylı bir şekilde verilmiştir.

Robot kinematiği ve dinamiği, kullanılan matematik modeller ve parametre kestirme algoritmaları hakkında genel bilgiler verildikten sonra 7. bölümde üç eklemlili bir robotik manipülatör için yapılan simülasyon çalışmalarına geçilmiştir. Bir fikir vermesi amacı ile tek giriş-tek çıkışlı (SISO) durum için kendi kendini ayarlayan (Self-tuning) bir adaptif kontrolörün aynı robotik manipülatöre uyarlanması da incelenmiştir. Öngörülü kontrol algoritması robotik manipülatöre tek giriş-tek çıkış (SISO) ve çok giriş-çok çıkışlı (MIMO) durumlar için ayrı ayrı uygulanmış ve seçilen kübik ve eksponansiyel yörüngeler için yörünge kontrolü ile ilgili simülasyon çalışmaları ve sonuçları ilave edilmiştir. Ayrıca hesapsal açıdan büyük zorluklara neden olan ters kinematik problemini ortadan kaldıran ve üç elemanın direkt olarak kartezyen koordinatlarda kontrolüne imkan veren yeni bir model geliştirilerek öngörülü kontrol algoritması ile uyum içinde çalıştığı gösterilmiştir.

Elde edilen sonuçlar, genelleştirilmiş öngörülü kontrol algoritmasının tespit edilen yörüngelerin takibini çok yüksek bir hassasiyetle gerçekleştirdiğini ve direkt kartezyen uzay kontrolünün da çok başarılı olduğunu göstermektedir.

SUMMARY

JOINT AND CARTESIAN BASED PREDICTIVE CONTROL OF A ROBOTIC MANIPULATOR

In this study, SISO and MIMO predictive control (Generalized predictive control-GPC) algorithms are proposed for the trajectory control of a three-link manipulator. Manipulator is controlled both in the joint coordinate system and directly in the cartesian coordinate system.

Manipulators have been using extensively, such as in the nuclear industries, deep undersea exploration and maintenance operations, in space and increasingly in industrial automation applications. A mechanical manipulator can be defined as a multi degree of freedom open loop chain of mechanical linkages and joints. These mechanisms driven by actuators are capable of moving an object in space from initial to final locations or along prescribed trajectories. Tracking accuracy is of prime concern in assembly task. Hence, a robot controller should be able to achieve fast and at the same time accurate tracking and positioning control of the gripper. The dynamic equations of a mechanical manipulator are highly nonlinear and complex. To overcome these difficulties, several advanced adaptive control techniques for dynamic control of mechanical manipulators have been proposed, e.g., self-tuning control, model reference adaptive control, long-range predictive control.

The equations for the motion of the manipulator may be developed by the direct application of the classical Newtonian mechanics. For a manipulator with m joints the mathematical model which is consist of a set of coupled nonlinear differential equations may be written in the joint coordinate system as follows:

$$D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) = u(t) \quad \dots (1)$$

where θ , $\dot{\theta}$ and $\ddot{\theta}$ are m -dimensional vectors signifying the joint position, velocity and acceleration, respectively. $D(\theta)$, a $(m \times m)$ symmetric matrix, includes the acceleration related coefficients of the joints and the effects of link inertia. $H(\theta, \dot{\theta})$, a m dimensional vector, signifies Coriolis and centrifugal torques. $G(\theta)$, a m -dimensional vector, represents the torques due to the gravity. The m -dimensional vector $u(t)$ is the system input. The expressions of $D(\theta)$, $H(\theta, \dot{\theta})$ and $G(\theta)$ contain trigonometric functions. A digital simulation of (1) is very time consuming due to a larger number of

mathematical operations. More over, the parameter values vary from task to task.

If the model in (1) is first properly linearized about a nominal trajectory and then discretized by Euler's method, a multivariable discrete-time model is obtained in the form.

$$y(t) = a_0 + A_1 y(t-1) + A_2 y(t-2) + B_1 u(t-1) + B_2 u(t-2) + e(t) \quad \dots (2)$$

where A_i and B_i , $i=1,2$ are $(m \times m)$ matrices, a_0 is the m -dimensional vector, and the vector $e(\cdot)$ represents modelling errors. The direct use of (2) is difficult because the calculation of the model parameters in the coefficient matrices is a tedious task.

Motivated by the form of (2), a CARIMA (Controlled Auto-Regressive Integrate Moving Average) model is proposed here to model the motion of a manipulator and to design a controller for the system. Such a model will be determined on the basis of the measured input-output data of a manipulator system. The parameters of the model will be determined so that the best fit of the model to the input-output data will be obtained in the least-squares error sense. As is well known, the parameters in such an assumed model can be estimated on-line using recursive equations. They can be obtained by minimizing the sum of the squared errors.

The CARIMA model is assumed to have the same number of inputs and outputs. The CARIMA model will be written in the following general form:

$$y(t) = A(q^{-1}) y(t) + B(q^{-1}) u(t-1) + e(t) \quad \dots (3)$$

where the m -dimensional output vector y and the input vector u have as the i th component the output y_i and the input u_i of joint i , respectively, and $i=1,2,\dots,m$. The equation error vector $e(t)$ in the CARIMA model is considered to be of the form:

$$e(t) = \xi(t)/\Delta \quad \dots (4)$$

where Δ is the differencing operator $1-q^{-1}$ and $\xi(t)$ is an uncorrelated random sequence. The argument q^{-1} is backward shift operator. The $(m \times m)$ matrices A and B are polynomials defined by:

$$A(q^{-1}) = A_1 q^{-1} + \dots + A_{n_a} q^{-n_a} \quad \dots (5)$$

$$B(q^{-1}) = B_0 + B_1 q^{-1} + \dots + B_{n_b} q^{-n_b} \quad \dots (6)$$

Long-range predictive control (LRP) has received

much interest over the last few years as an alternative to one-step-ahead and pole-placement design methods in parameter adaptive control. The well-known Generalized Predictive Control (GPC) has been found to possess features essential for successful application of self-tuning control to real processes, e.g., robustness against violation of modelling assumptions (model order, system delay), applicability to a wide class of systems including open-loop unstable and/or nonminimum phase, and possibly time-varying processes.

The majority of Generalized Predictive Control algorithm is basically a combination of a predictor, a cost function, an algorithm to choose a control sequence which minimizes this cost and a parametric identification scheme.

The controller minimizes the quadratic cost function

$$J = \sum_{j=N_1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \sum_{j=1}^{N_u} \lambda \Delta u(t+j-1)^2 \quad \dots (7)$$

where $y(t+j)$ is the j -step-ahead prediction of the plant. The quantities N_1 , N_2 and λ are called the initial horizon, prediction horizon and the control weighting respectively. If we subject the minimization to the assumption that $\Delta u(t+j) = 0$, for $j > N_u$, the control horizon, we arrive at the following solution:

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (y_r - f) \quad \dots (8)$$

where:

$$\Delta u = [\Delta u(t), \Delta u(t-1), \dots, \Delta u(t+N_u-1)]^T$$

$$y_r = [y_r(t+1), y_r(t+2), \dots, y_r(t+N_2)]^T$$

$$f = [f(t+1), f(t+2), \dots, f(t+N_2)]^T$$

$$G = \begin{bmatrix} g_0 & 0 & \dots & \dots & 0 \\ g_1 & g_0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2-1} & \vdots & \vdots & \vdots & g_{N_2-N_u} \end{bmatrix},$$

and g_j is the j th value of the model step response. The term $f(t+j)$ is the free response of the plant at time $t+j$, i.e. the output of the model at this time assuming zero future control increments. $y_r(t)$ is the set point at time t .

The generalized predictive control algorithm will become:

- 1)- Measure the plant output and external set-point,
- 2)- Estimate the parameters of the plant using RLS,
- 3)- Compute the free response $f(t+j)$,
- 4)- Compute g_i parameters and $\Delta u(t)$ using Eqn. (8),
- 5)- Shift the data ready for the next sample.

In order to evaluate the behaviour of the generalized predictive control strategies, four sets of simulation studies were carried out (Chapter 5). For each set the purpose of the simulation study, the description of the simulated system, the criterion for selecting the tuning parameters, the set point changes and disturbances are described.

From the simulation studies it can be concluded that the generalized predictive control algorithms offer interesting features making them suitable for real-life-applications.

This study presents SISO and MIMO generalized predictive control algorithms (GPC) for the trajectory control of a three link manipulator. The performance of the predictive controller was tested for several reference trajectories and under stochastic and deterministic disturbances.

Separate Joint Control (SISO) : The problem is to design a predictive controller for each joint so as to make the angular motion of each joint follow a desired path specified by discrete points. It is assumed that interactions between the joints are small. Thus coupling terms in the CARIMA model of the manipulator system are omitted and each joint will be controlled independently of the other. For the system, the input is the torque u_i applied to the joint motor, and the output is the velocity V_i of the joint.

Simulations indicate that the system performs well when the separate joint control is applied. The main advantage of a separate joint control is that it is simple. Moreover, the controller can be implemented using microprocessors since the number of updating system parameters is less.

Predictive Controller for Interacting Joints (MIMO):

In this control scheme, the dynamic couplings are taken into consideration in the design of multivariable GPC controller. The task here is to determine a predictive control scheme for the manipulator joints such that the motion of the manipulator joints follow a desired path as closely as possible for this system, the input is the torque u applied to the joint motor (a three-dimensional vector) and the output is a

three-dimensional joint velocity vector.

Cartesian Space Predictive Control

The control of robotic manipulators is commonly performed by servoing the angular positions of the joints, i.e., by driving the actuators with error signals which is proportional to the differences between the desired and actual angular positions. This implies that the position of the end effector expressed in the world (cartesian) coordinate system has been converted to the desired angular positions of the joints. It is accomplished by solving numerically the inverse kinematic equations, which consist of coordinate transformations containing trigonometric functions. This solution may not be unique; for example, in a manipulator with six revolute joints (six degrees of freedom) this is so, unless additional constraints are imposed. The calculations involving the coordinate transformations are usually performed many times while the manipulator is in motion; this complicates the control scheme considerably. If the manipulator were controlled directly in the world coordinate system, the calculations could possibly be simplified.

In order to control the motion of the end effector of a manipulator in the cartesian coordinates, it is necessary to have available the measurements of the gripper position in the world coordinates may be feasible to obtain these values using a light sensor system. These coordinates can also be derived from the encoder readings using the aforementioned coordinate transformation, i.e., by solving the kinematic equations for the world coordinates.

This is considerable simpler than solving the inverse kinematic equations which is the conventional approach. In performing tasks which require force (torque) servoing, and/or avoiding obstacles, the control of a manipulator directly in the world coordinates may be more desirable than the control in the joint coordinates.

Therefore is applied the predictive approach also to the case in which the manipulator is controlled directly in the cartesian coordinate systems. The outputs of the model include the velocity components corresponding to the xyz-coordinates, and the inputs are the torques into the joint motors. The predictive controller is then determined by minimizing the expected value of a quadratic criterion. This performance index penalizes the deviations of the actual path of the gripper from the desired values expressed in the cartesian coordinate system and the use of the energy associated with the input vector so that the desired path expressed at discrete points in the cartesian coordinate system is followed as closely as possible. A first order CARIMA

model was used for the predictive trajectory control of a three-link manipulator. The performance of the predictive controller was tested for trajectories and under stochastic and deterministic disturbances.

The simulation studies were performed with the same numerical values and the same values of tuning-knobs both SISO-MIMO algorithms and joint or cartesian base control algorithms.



BÖLÜM 1. GİRİŞ

Endüstrilerin rekabete dayanan yapısı, son yıllarda otomasyona geçişi hızlandırmış ve bunun sonucu olarak da en verimli otomasyon aracı olarak nitelendirilen robotlar büyük önem kazanmışlardır.

Günümüzde robot tanımı, bir robotun üç niteliğe sahip olması gerektiğini belirtmektedir. Bu nitelikler; manipülasyon yapma, yani cisimlerin yerini değiştirme ve üzerlerinde işlem yapma, programlanabilirlik ve algılama sistemi ile çevre koşullarına göre düşük düzeyde karar verebilme yetkisine sahip olabilmesidir. Robot veya manipülatörlerin kullanım alanları daha çok seri üretim gereken sahalar (kaynak, boyama işleri v.s.), montaj işleri, nükleer reaktör çekirdekleri gibi insan çalışmasının mümkün olmadığı yerler, deniz dibi araştırmaları ve uzay çalışmaları olarak özetlenebilir.

Robotların gelişen endüstrilerdeki etkinliğinin artmasının temel nedenleri olarak, ekonomik olmaları, çalışanların verimliliğini arttıran sağlık ve emniyet istekleri, can sıkıcı ve tekrarlı işlerin niteliğini değiştirme gereği v.s. gösterilebilir.

Kullanılan ticari robotik manipülatörlerin büyük bir çoğunluğu nispeten basit olan kontrol sistemleri ile donatılmışlardır. Bunlara örnek olarak konvensiyonel PID tipi kontrolörler gösterilebilir. Fakat, bu tip kontrol sistemlerinin sadece düşük hızlarda yeterli olabileceği yapılan çalışmalarla kanıtlanmış durumdadır. Bu nedenle, manipülatör performansındaki artan isteklerden dolayı gelişmiş kontrol tekniklerine gereksinme duyulmaktadır.

Manipülatör (veya robot kolu) kontrol kanununun elde edilmesindeki ilk adım, sistemin dinamik modelinin çıkarılmasıdır. Manipülatörün dinamik davranışını veren denklemler doğrusal olmayıp aralarında yüksek oranda etkileşimler olan ikinci dereceden diferansiyel denklem takımlarıdır. Manipülatörlerin dinamik kontrolü, manipülatörün istenen pozisyon ve hız referanslarına göre eklemlerin hareketini sağlayacak aktuatör giriş bilgilerinin (tork/kuvvet veya voltaj) oluşturulmasını gerektirmektedir. Dinamiği belirleyen ikinci dereceden diferansiyel denklemlerin doğrusal olmama ve etkileşimli olma özellikleri, manipülatörün izleyeceği yörüngenin kontrolünü güçleştirmektedir. Eklemler arasındaki etkileşimlerin fazla olması ve dinamiklerin doğrusal olmaması nedeni ile klasik kontrol sistemlerini kullanan endüstriyel manipülatörler, belirli bir hız limitinin üzerinde kolayca çalıştırılamamakta ve üretimin verimliliği sınırlandırılmaktadır. Ayrıca, manipülatör kontrolörlerinden beklenen ve her geçen gün artan performans istekleri de gelişmiş manipülatör kontrol tekniklerine gereksinme duyulmasına neden olmaktadır. Endüstride kullanılan manipülatör kontrolörleri yukarıda da belirtildiği gibi genellikle çok basit bir yapıya sahiptirler. Eklemler arasındaki etkileşimin önemsiz sayılabilecek düzeyde kalabilmesi için eklem hızları düşük tutulmakta ve her bir eklem ayrı ayrı, bir PID kontrolör ile denetlenmektedir. Manipülatörün çalışma hızını arttırabilmek için eklem hızları arttırıldığı zaman, etkileşimler ve robot modelindeki belirsizlikler nedeni ile istenilen yörünge ile gerçekleştirilen yörünge arasında önemli hatalar ortaya çıkmaktadır. Bu durumda ya robot kolunun modelinin mümkün olduğunca doğru olması ve kol ucunun taşıyacağı yük ağırlığının zamanla değişiminin önceden bilinmesi, ya da adaptif bir yaklaşımın kullanılması gerekmektedir. İlk durumda, yüksek bir bilgisayar gücü gerekli olduğu gibi yükün zamanla değişimini önceden bilmek mümkün de olmayabilir.

Manipölatörlerin dinamik kontrolü ile ilgili olarak yapılan çalışmaların çoğunda lineer olmayan dinamiklerin karmaşıklığı vurgulanmış ve kontrol sistemlerinin karmaşıklığını azaltmak için değişik metodlar geliştirilmiştir. Bu metodlar, genellikle dinamik denklemlerin lineerleştirilmesi ile ilgilidir ve büyük miktarda karışık hesaplamalar gerektirmektedir. Bu yüzden de, mevcut birçok kontrolör doğru yörünge kontrolünü sadece düşük hızlarda gerçekleştirmektedir.

Bir robot kolunun kontrolü için değişik metodlar önerilmiştir. Klasik kontrolörler [1,2] ve optimum kontrolörler [3,4] içeren kontrol algoritmaları teklif edilmiştir. Klasik kontrol metodlarından olan nominal bir yörünge için torkların hesaplanması [1]'de incelenmiştir. Bu kontrol metodu, büyük miktarlarda hesaplama ve hafıza yükü gerektirmektedir. İlk önerilen tekniklerden diğer bir tanesi çözümlenmiş hız (resolved rate) kontrol tekniğidir [2]. Bu teknikte eklem açısız hız değerleri, manipölatörün uç elemanının tanımlanmış bir doğrultuda hareket etmesi sağlanacak şekilde hesaplanmaktadır.

Robotik manipölatörler için kullanılan ileri kontrol tekniklerinin büyük bir kısmı, hesaplanmış tork/kuvvet kontrol tekniği içerisinde sınıflandırılmaktadır. Lineer olmayan geri beslemeli kontrol (nonlinear feedback control) [5], ileri beslemeli kontrol (feedforward control) [6], çözümlenmiş hareket kontrolü (resolved motion control) [7,2] ve konfigürasyon uzayı kontrolü (configuration space control) [8] bu kategori içerisinde yer almaktadır.

İleri kontrol tekniklerine bir örnek olarak, Seraji [9] tarafından 1987'de yapılan bir çalışmada, eklem ve kartezyen koordinatlarda çok eklemli robot manipölatörlerinin çok değişkenli kontrolü için basit bir kontrol stratejisi sunulmuştur. Eklem kontrolü, iki adet

birbirinden bağımsız çok değişkenli ileri besleme ve geri besleme kontrolörü ile sağlanmaktadır. İleri besleme kontrolörü robot dinamiğinin lineerleştirilmiş modelinin minimal olarak tersidir ve sadece orantı-diferansiyel (PD^2) terimlerini içermektedir. İleri besleme, istenen pozisyon, hız ve ivme değerlerinden elde edilmektedir. Bu kontrolör, manipülatör eklem kontrol açıları için herhangi bir referans yörüngeyi takip etmesini sağlamaktadır. Geri besleme kontrolörü orantı-integral-diferansiyel (PID) tipindedir, pol atama esasına göre dizayn edilmiştir. Bu kontrolör, başlangıç izleme hatasını sıfıra indirmekte ve eklem açıları ile verilen basamak + eksponansiyel yörüngelerin kararlı şekilde takip edilmesini sağlamaktadır. Eklem kontrol dizaynı, Serafi tarafından daha sonra kartezyen uzayda uç elemanın hareketinin direkt kontrol edilmesini sağlayacak şekilde genişletilmiştir. Simülasyon çalışmaları ile önerilen kontrol yaklaşımının performansı gösterilmiştir.

Hesaplanmış tork tekniğine dayanan manipülatör kontrol sistemlerinde, manipülatör dinamiğinin bir matematik modeli kullanılarak hesaplanan nominal torqlar, bilinmeyen yüklerin ve modellemedeki hataların etkilerini tam olarak içermemektedir. Bilinmeyen yükleri ve parametre belirsizliklerini dengeleyici bir hesaplanmış tork tekniği yaklaşımı Goldenberg ve arkadaşları [10] tarafından 1989' da ardışık olarak tanımlanan yeni bir dinamik operatör kullanılarak gerçekleştirilmiştir. Tanılama işlemi en küçük kareler yaklaşımına dayanmaktadır. Dengelenmiş nominal torqları veren bu operatör kullanılarak sistem, eklem koordinatlarında hatanın minimize edilmesini sağlamak için kapalı çevrimde kontrol edilmektedir.

Robotik manipülatörler için geliştirilen ileri kontrol tekniklerinin diğer tipleri, öğrenen kontrol (learning control) [11] ve adaptif kontrol teknikleridir.

Son yıllarda önerilen adaptif kontrol teknikleri, robotik manipülatörlerin performansında önemli gelişmeler sağlamıştır. Adaptif kontrol, robotik manipülatöre eğitilebilir özellikler de kazandırmaktadır. Değişik şartlar altında, özellikle tekrarlı görevlerde, çok büyük performans özellikleri göstermektedirler. Yüksek çalışma hızları ve taşınan yükteki değişimler durumunda ortaya çıkacak istenmeyen etkileri dengeleme özelliğine de sahip olmaktadır.

Artan performans isteklerini karşılamak için değişik adaptif manipülatör kontrol algoritmaları önerilmiştir. Adaptif kontrol metodlarında kontrolör parametreleri, kontrol sistemini istenen performansa adapte edebilmek için gerçek zamanda değişmektedir. Adaptif kontrol algoritmalarında, kontrolör parametrelerini tanılamak için çevrim-ici model tanılama teknikleri kullanılır. Robot kontrolunda kullanılan başlıca adaptif kontrol algoritmaları ile ilgili olarak yapılan çalışmaları aşağıdaki şekilde özetlemek mümkündür.

Önerilen adaptif kontrol yöntemleri içinde en çok kullanılanlardan bir tanesi, model referans adaptif kontrol (MRAC) yöntemidir. Dubowsky ve Des Forges [12], 1979' da manipülatörün kontrolü için model referans adaptif kontrol kanunu tasarlamışlardır. Çalışmalarında manipülatör eklemleri arasındaki etkileşimi ihmal etmişler ve her bir eklem için referans model olarak lineer ikinci dereceden zamana bağlı olmayan bir diferansiyel denklem takımı kullanmışlardır. Manipülatör, modeli takip etmek için pozisyon ve hız geri besleme kazançlarının ayarlanması ile kontrol edilmektedir.

Dubowsky ve Des Forges [13] 1979' daki bir başka çalışmalarında, istenen durum vektörü ve robotik manipülatörün durum vektörü arasındaki fark olarak tanımlanan hatanın kuadratik bir fonksiyonunu minimize

eden bir adaptif kontrol algoritması geliştirmişlerdir. Kabul ettikleri model lineerdir ve bağımlı değildir. Ayrıca taşınan yükteki ani değişimlerden oluşan ilk geçici cevap değişimlerini minimize etmek için, bir öğrenme sinyali de ilave edilmiştir.

Leininger ve Wang [14], 1982' de altı serbestlik dereceli robotik bir manipülatöre pol-atamalı kendi kendini ayarlayan kontrol metodu uygulamışlardır. Bu metod, minimum-varyans kendi kendini ayarlayan kontrol algoritmasından farklı olarak sistemin minimum faz olmasını ve açık çevrimli mekanizmanın kararlı olmasını gerektirmemektedir. Ayrıca, bu yaklaşım minimum-varyans metodtan daha güvenilir olup robotik manipülatörler için daha uygundur. Her bir eklem için ayrı ayrı pol-atamalı kendi kendini ayarlayan kontrolör dizayn edilmiştir. Yapılan simülasyon çalışmaları, 2 kg.lık bir yük taşınacak ve dairesel bir yörünge izlenecek şekilde gerçekleştirilmiş ve başarılı sonuçlar elde edilmiştir.

Koivo [15], 1983' de her bir eklem için kendi kendini ayarlayan tip bir adaptif kontrolör kullanarak, bir manipülatörün pozisyon ve hız kontroluna yeni bir yaklaşım getirmiştir. Karmaşık manipülatör sistemi zaman serisi fark denklemleri ile modellenmiştir. Model parametreleri, denklem hatalarının karelerinin toplamını minimize eden çevrim-içi ardışık algoritmalar ile tespit edilmektedir. Her bir eklemin adaptif kontrolörü, fark denklemi modeline ve bir performans kriterinin seçilmesine dayanarak dizayn edilmiştir. Kontrolör kazançları sistem parametrelerinin kestirilen değerleri ile elde edilen model kullanılarak çevrim-içi hesaplanmaktadır. Simülasyon sonuçları yaklaşımın kullanılabilirliğini göstermektedir.

Tomizuka ve Horowitz [16], 1983' de mekanik manipülatörlerin dinamik kontrolü için model referans kontrol sistemi uygulamışlardır. Manipülatörün, yük

değişimlerine bağlı olan atalet karakteristiklerini ve eklemler arasındaki etkileşimleri dengeleyen bir ayrık zamanlı MRAC algoritması kullanmışlardır. Simülasyon sonuçları, uç serbestlik dereceli bir manipülatör için MRAC tekniğinin avantajlarını göstermektedir. Adaptif kontrolör sayesinde manipülatörün cevabı, taşınan yük ve konfigürasyon değişmelerinden etkilenmemektedir.

Lee ve Chung [17,18,19], geniş bir hareket sahası ve taşınan yük durumu için sürekli olarak, zamana bağlı bir yörüngeyi mümkün olduğunca kapalı olarak izleyecek adaptif pertürbasyon kontrol çalışması yapmışlardır. Önerdikleri adaptif kontrol, nominal bir yörünge civarında lineerleştirilmiş pertürbasyon denklemlerine dayanmaktadır. Kontrol edilecek sistem ayrı olarak veya birlikte hesaplanabilen ileri ve geri besleme bileşenleri ile karakterize edilmektedir. İleri besleme bileşeni, değişik eklemler arasındaki etkileşim kuvvetlerinin hepsini dengelemek için hareketin Newton-Euler denklemlerinden nominal torkları hesaplamaktadır. Geri besleme bileşeni, ardışık en küçük kareler tanılama işlemini içermektedir ve optimal bir kendi kendini ayarlayan kontrol algoritması, lineerleştirilen sistem için nominal yörünge boyunca manipülatörün pozisyon ve hız hatalarını azaltan pertürbasyon torklarını hesaplamaktadır. Formülasyonun avantajı, nominal ve varyasyonel torkların ayrı olarak veya birlikte hesaplanabilmesidir. Önerilen adaptif kontrolün performansını değerlendirmek için bir simülasyon çalışması da ilave edilmiştir.

Koivo, A.J. [20] tarafından 1985' de kartezyen baz kontrol sisteminde kendi kendini ayarlayan manipülatör kontrolü ile ilgili olarak yapılan çalışmada, robotik manipülatör ayrık zamanlı stokastik bir model ile modellenmiştir. Giriş vektörü eklem motorlarına uygulanacak voltajlardan, çıkış vektörü ise uç elemanın kartezyen $(\dot{x}, \dot{y}, \dot{z})$ hızlarından meydana gelmektedir.

Kullanılan model çok deęişkenli otoregressif bir modeldir. Modelin bilinmeyen parametreleri en küçük kareler algoritması ile ardışık olarak çevrim-içi hesaplanmaktadır. Adaptif kendi kendini ayarlayan tip kontrolör, kuadratik bir performans kriteri minimize edilerek dizayn edilmiştir. Bu performans kriteri, tutucunun gerçek yolunun kartezyen koordinat sisteminde tanımlanmış istenen deęerlerden sapmalarını ve giriş vektörü ile ilgili olarak enerjinin kullanımını kısıtlamaktadır.

Kartezyen baz koordinat sisteminde kendi kendini ayarlayan kontrol algoritması için Backes, Leininger ve Chung [21] tarafından 1986' da yapılan çalışmada, bir sayısal durum geri besleme algoritması, bir robotik manipülatörün uç elemanının yakın minimum zaman (near-minimum time) yörüngesini elde etmek için geliştirilmiştir. Bu algoritmada, lineerleştirilmiş kapalı çevrim sisteminin polleri, aktuatör torklarındaki kısıtlamaları da göz önüne alarak yakın minimum zaman cevabını oluşturabilmek için z domeninde uygun bir şekilde yerleştirilmektedir. Algoritmanın performansını gösterebilmek amacıyla üç uzuvlu bir manipülatör için simülasyon çalışması, başlangıç ve final durumlarının üç farklı kombinasyonu için gerçekleştirilmiştir.

Tomizuka ve Horowitz [22], 1986' da mekanik manipülatörler için yeni bir adaptif kontrol yaklaşımı önermişlerdir. Manipülatör denklemlerinin temel özellikleri kullanılarak, dinamik denklemlerdeki doğrusalsızlıkları dengelemek ve eklemler arasındaki dinamik etkileşimi ayırmak için bir adaptif kontrol algoritması geliştirmişlerdir. Yapılan simülasyon çalışmasıyla, adaptif lineer olmayan dengeleyici/ayırıcı kontrolör ve integral etkili durum geri besleme kontrolöründen oluşan manipülatör kontrol sisteminin performansı değerlendirilmiştir. Simülasyon sonuçları, adaptif kontrolörlü manipülatör kontrol sisteminin, konfigürasyon ve taşınan

yükteki değişimlere karşı duyarsız olduğunu göstermektedir.

Craig ve arkadaşları [23], 1987' de dinamik model yetersiz olduğu zaman lineer olmayan ve bir modeli baz olarak alan kontrolörlere de ilave edilebilen, parametrelili adaptif kontrol dizaynı önermişlerdir. Algoritma, özel olarak lineer olmayan model bazlı kontrolörler için dizayn edildiğinden konusunda tektir.

De Silva ve arkadaşları [24], 1987' de robotik manipülatörlerde yörünge kontrolu amacı ile bir ileri besleme torku ve istenen yörüngeden meydana gelecek sapmaları düzeltmek için geri besleme kazanç listeleri kullanan bir kontrol algoritması önermişlerdir. Önerilen adaptif kontrolör, yörünge hatalarını çevrim içi tanılama ile veya bir referans model kullanılmaksızın en küçük kareler hassasiyetinde elimine etmektedir. Kontrol algoritması dinamik doğrusalsızlıkları (koriyolis ve merkezci ivmeler ve taşınan yükteki değişimler), geometrik doğrusalsızlıkları (lineer olmayan koordinat tanımları) ve fiziksel doğrusalsızlıkları (lineer olmayan sönümler), manipülatörlerdeki dinamik bağlara ilave olarak hesaba katmaktadır. Simülasyon çalışmaları ile kontrol algoritmasının kullanılabilirliği gösterilmiştir.

Serafi [25], 1987' de eklem açıları ile yörünge takibini gerçekleştirmek için manipülatörlerin adaptif kontroluna yeni bir yaklaşım getirmiştir. Bu yaklaşımda temel fikir, bir ileri besleme kontrolörü olarak manipülatör dinamiğinin inversinin kullanılmasıdır. İstenilen yörünge, ileri besleme kontrolörüne giriş olarak uygulanmaktadır, kontrol çıkışı olarak ise manipülatörü hareket ettirecek torklar seçilmiştir. Kontrolör kazançları daha sonra, çalışma noktasının değişmesinden dolayı ortaya çıkacak manipülatör inversindeki değişimler karşısında etkin olabilmek için MRAC teorisinden elde edilen bir adaptasyon algoritması

ile yenilenmektedir. Adaptif bir geri besleme kontrolörü ve bir yardımcı sinyal, kapalı çevrim kararlılığını arttırmak ve daha hızlı adaptasyon gerçekleştirmek için kullanılmaktadır. Önerilen kontrol algoritması hesapsal açıdan hızlıdır ve karmaşık bir yapıdaki dinamik model ile ilgili bir ön bilgi veya manipülatörün parametre değerlerini ve taşınan yükün bilinmesini gerektirmemektedir. Sonuçlar, geliştirilen kontrolörün farklı referans yörüngeleri ve taşınan yükteki değişimlere rağmen gayet iyi performans gösterdiğini kanıtlamıştır.

Mei-Hua Liu ve We Lin [26] tarafından 1988' de çok değişkenli kendi kendini ayarlayan kontrol dizaynı ile ilgili olarak iki ayrı yaklaşım sunulmuştur. İlk yaklaşımda, eklemler arasında lineer etkileşim göz önüne alınmış ve lineer olmayan dinamik davranış, kanonik çok değişkenli bir fark denklemi kullanılarak modellenmiş ve bu modele dayanarak yeni bir kendi kendini ayarlayan kontrolör dizaynı yapılmıştır. Ayrıca eklemler arasındaki etkileşimleri tamamen kompanze edebilmek için ikinci bir kendi kendini ayarlayan kontrol yaklaşımı daha önerilmiştir. Bu kontrolde, nominal torklar bir ileri besleme oluşturmak için mevcut dinamik denklemler ile hesaplanmaktadır. İleri besleme, istenen yörünge boyunca eklemler arasındaki tüm etkileşimleri dengelemektedir. Çok değişkenli perturbasyon fark modeline dayanan geri besleme kontrolörü, ilk yaklaşıma benzer bir kendi kendini ayarlayan kontrol algoritması kullanılarak dizayn edilmiştir. Geri besleme kontrolörü sayesinde üretilen perturbasyon torkları yörünge sapmasını sıfıra götürmektedir.

Robotik manipülatörlerin kontrolü bazen karma (hybrid) kontrol uygulamaları gerektirmektedir. Karma kontrolde, hem manipülatörün uç elemanının pozisyonu ve hem de uç eleman tarafından uygulanacak kuvvet kontrol edilmektedir. Karma kontrol ile ilgili olarak yapılan çalışmalara örnek olarak Backes [21] tarafından yapılan

çalışma gösterilebilir. Çalışmada, kartezyen referans yörüngeleri kullanan eklem koordinatlı kendi kendini ayarlayan manipülatör kontrol metodu incelenmiştir. Hem pozisyon ve hem de karma kontrol gerçekleştirilmektedir. Pozisyon ve kuvvet hataları, kartezyen koordinatlardan eklemlerdeki pozisyon ve kuvvet hatalarına dönüştürülmektedir. Her bir eklemdaki pozisyon ve kuvvet hataları, pol atamalı kendi kendini ayarlayan bir algoritma kullanılarak elimine edilen bir karma hataya indirgenmektedir.

Seraji tarafından 1987' de yapılan bir çalışmada [27] ise, karma kontrol yapısı içinde adaptif kuvvet ve pozisyon kontrolörlerinin dizaynı için sistematik metodlar önerilmiştir. Kuvvet kontrolörü, istenen kuvvet referanslarının izlenmesini, pozisyon kontrolörü ise istenen pozisyon yörüngelerinin izlenmesini sağlamaktadır. Kuvvet ve pozisyon kontrolör kazançları, basit aritmetik işlemler yardımı ile adaptasyon kanunları sayesinde oluşturulmakta ve böylece kontrolör yüksek örneklem oranları ile çevrim içi çalışabilmekte ve hesapsal açıdan hızlı olmaktadır.

Robot kolunun dinamik davranışını belirleyen denklemlerdeki doğrusalsızlıklar ve etkileşimler nedeni ile ortaya çıkan kontrol güçlüklerinin, adaptif öngörülü yöntemler ile ortadan kaldırılabileceğine son yıllarda literatürlerde yer alan bir çok çalışmada işaret edilmiştir. Bu tez çalışmasında robotik bir manipülatörün kontrolü için öngörülü kontrol algoritmasının geliştirilmesi de bu amaç doğrultusunda yapılmıştır. Bu konuda yapılan önemli bazı çalışmaları aşağıdaki şekilde özetlemek mümkündür.

Lelic ve Wellstead [28], 1987' de yaptıkları bir çalışmada, robot kontrolü için çok adımlı öngörülü bir kendi kendini ayarlayıcının performansını incelemişlerdir. CARIMA ve CARMA tipi modeller için ayrı ayrı

çalışmalar yapmışlardır.

Clarke, D.W [29] tarafından 1987' de genelleştirilmiş öngörülü kontrol algoritması, tek uzuvlu bir robot koluna uygulanmış ve PID tip bir kontrolör ile yörünge takibi karşılaştırılmıştır.

Martin Lambert [30] tarafından 1987' de yapılan bir çalışmada ise yine tek uzuvlu fleksibl bir robot koluna genelleştirilmiş öngörülü kontrol algoritması uygulanmış ve simülasyon sonuçları ile algoritmanın geçerliği kanıtlanmıştır.

Tsang, T.C [31] tarafından 1990' da yine tek uzuvlu fleksibl bir robot koluna çok giriş çok çıkışlı genelleştirilmiş öngörülü kontrol algoritması uygulanmış ve başarılı sonuçlar elde edilmiştir.

Kaynak, O. [32,33,34] tarafından da öngörülü kontrol algoritmalarının robotlara uygulanması konusunda çeşitli çalışmalar yapılmıştır. Kaynak çalışmalarında [33], genelleştirilmiş öngörülü kontrol algoritmasına eşdeğer yeni bir öngörü yöntemi önermiş ve kapalı çevrim transfer fonksiyonunun kutuplarının istenilen yerlere yerleştirilmesi ile, bir maliyet fonksiyonunun minimize edilmesini hedefleyen iki ayrı kontrol algoritması sunmuştur. Elde ettiği simülasyon çalışmaları ile öngörülü kontrol algoritmalarının manipülatörlerin yörünge kontrolunda çok verimli bir şekilde kullanılabileceğini göstermiştir. Yaptığı diğer bir çalışmada [34], öngörülü kontrol algoritmasını iki serbestlik dereceli düzlemsel bir artık robota uygular. Algoritmanın dizaynında, uç elemanın izlemesi istenen yörüngeye ek olarak, eklemlere kısıtlar getiren bir başka fonksiyon tanımlanmıştır. Elde edilen simülasyon sonuçları, öngörülü kontrolün, artık robot için de çok basit bir model kullanılmasına rağmen, istenilen yörünge için büyük bir doğrulukla izlenmesini sağladığını

göstermektedir.

Gerçekleştirilen tez çalışmasında, non-minimum faz sistemler, açık çevrimli kararsız sistemler, robotik manipülatörler, değişken veya bilinmeyen model mertebesine sahip sistemler gibi gerçek sistemlerin büyük bir çoğunluğunun kararlı olarak kontrolü için rahatlıkla kullanılabilen genelleştirilmiş öngörülü kontrol (GPC) algoritması incelenmiştir.

İlk aşamada, GPC algoritması değişik dinamik sistemler için değişik etkenler altında (stokastik ve deterministik bozucular) test edilmiş ve performansı araştırılmıştır. Ayar parametrelerinin (N_2 , N_u , λ) performans üzerindeki etkileri araştırılmış ve genelleştirilmiş öngörülü kontrol algoritması için literatürlerde söz edilen düşünceler doğrultusunda sonuçlar alınmıştır [35,36,37,38,39].

İkinci aşamada, tek adım sonrasını öngören kontrol algoritmalarından olan minimum-varyans kendi kendini ayarlayan bir regülatör [40] ele alınmıştır. Bu algoritma, 3 eklemlili bir robotik manipülatörün yörünge kontrolüne ayırık eklem kontrolü durumu (SISO) için uygulanmıştır. Yapılan simülasyon çalışmalarında, Koivo'nun [15] robotik manipülatörler için geliştirdiği adaptif lineer kontrolör için bulduğu performans sonuçlarının benzerleri elde edilmiştir. Taşınan yük ve değişimi, sürtünme etkileri, ölçme hataları gibi değişik stokastik ve deterministik bozucular karşısında, tek adım sonrasını öngören bir kontrol algoritmasının nasıl çalıştığı incelenmiştir.

Daha sonra, GPC algoritması yine aynı 3 eklemlili robotik manipülatörün yörünge kontrolü için kullanılmıştır. Tek giriş-tek çıkışlı (SISO) ve çok giriş-çok çıkışlı (MIMO) olmak üzere iki ayrı durum incelenmiştir. SISO durum için 2.ci mertebeden, MIMO durum için 1.ci

mertebeden ayırık zamanlı modeller kullanılarak GPC dizaynı yapılmıştır. Bu çalışma için Clarke' nin SISO GPC algoritması [37], MIMO durum için genişletilmiştir. SISO ve MIMO GPC algoritmalarının; taşınan yük ve değişimi, sürtünme etkileri, ölçme hataları gibi değişik stokastik ve deterministik bozucular etkisinde yörünge kontrolunda performansı araştırılmıştır. Bunun için değişik yörüngeler üzerinde çalışılmıştır.

Son olarak ise, kartezyen uzayda yörünge kontroluna geçilmiştir. Bunun için, kartezyen uzayda ayırık zamanlı bir model geliştirilerek bu modele dayanan MIMO-GPC dizaynı gerçekleştirilmiştir. Böylece, ters kinematik çözüme gerek kalmadan manipülatörün uç elemanının kartezyen koordinatları direkt olarak kontrol edilebilmektedir. Benzer bir kartezyen uzay kontrolu, Koivo tarafından kendi kendini ayarlayan kontrol uygulanarak gerçekleştirilmiştir [20]. Tez çalışmasında, kartezyen uzay kontrolu için geliştirilen ayırık zamanlı modele dayanan GPC kontrolör dizaynı, 3 serbestlik dereceli bir robotik manipülatöre uygulanmıştır. Elde edilen sonuçlar, kartezyen uzaydaki ayırık zamanlı modelin ve GPC' nin seçilen zor yörüngelere rağmen uyum içinde çalışarak iyi bir kontrol performansı sağladığını göstermektedir.

Genel olarak tez çalışmaları bünyesinde yapılan simülasyon çalışmaları, robot kontrolunda öngörülü kontrol algoritmasının gerek SISO ve gerekse MIMO eklem ve kartezyen uzay kontrolunda çeşitli bozucuların etkisine rağmen çok iyi performans gösterdiğini doğrulamıştır. Yeni bir kontrol algoritması olmasına rağmen, öngörülü kontrol robotik manipülatörlerin kontrolu sahasında aydınlık bir gelecek sunmaktadır.

BÖLÜM 2. ADAPTİF KONTROL SİSTEMLERİ

2.1. Giriş

Kontrol sistemlerinin büyük bir çoğunluğunun dinamik karakteristikleri sabit değildir. Her ne kadar dinamik karakteristiklerdeki küçük değişmelerin etkileri geri beslemeli bir kontrol sisteminde azalıyorsa da, şayet sistem parametrelerindeki ve çevredeki değişmeler büyük boyutlara ulaşıyorsa, her bakımdan yeterli sayılabilecek bir kontrol sistemi adaptasyon özelliğine de sahip olmalıdır. "Adaptasyon", çevre veya yapısal şartlardaki önceden kestirilemeyen değişmeler karşısında kendi kendini ayarlama veya değişiklik yapabilme kabiliyeti anlamına gelmektedir. Adaptasyon özelliğine sahip kontrol sistemleri de "Adaptif Kontrol Sistemleri" olarak adlandırılmaktadır. Örneğin; adaptif bir regülatör, proses dinamiği ve bozuculardaki değişmelere cevap olarak davranışını değiştirebilen bir regülatördür.

Bir adaptif kontrol sisteminde dinamik karakteristikler, kontrolör parametrelerinin optimal performansı devam ettirecek şekilde ayarlanabilmesi için devamlı olarak tanılanmaktadır. Bu kavram, makul ölçülerdeki mühendislik dizayn hatalarını da ortadan kaldıracığından adaptif kontrol sistemleri, sistem dizayncıları tarafından çok rağbet görmektedir.

Günümüzde adaptif kontrol, yapısal belirsizlikler ve çevresel değişmelerin varlığında sistemlerin kontrolünü ele alan modern kontrol teorisinin önemli bir sahasını teşkil etmektedir. Sayısal donanımların sayılarının

gittikçe artarak kolayca uygulanabilir duruma gelmelerinden, adaptif kontrolörlerin kontrol alanındaki önemi gittikçe artmakta ve her geçen gün kendine yeni uygulama alanları bulmaktadır.

Mühendislik problemlerinin bir çoğu, sistem parametreleri ve giriş sinyallerindeki beklenmedik büyük değişimler ve belirsizlikler yüzünden otomatik kontrol gerektirmektedirler. Böyle durumlarda konvansiyonel bir kontrol algoritması, sistemin performansını istenen bir seviyede tutamaz. Bu yüzden parametrelerdeki ve giriş sinyallerindeki bu beklenmedik değişimleri otomatik olarak dengeleyecek özel bir kontrol sistemine gereksinime duyulmaktadır. Yüksek performanslı kontrol sistemlerinin dizaynı, kontrol edilecek proses parametreleri yeterince bilinmiyorsa veya çalışma esnasında değişiyorsa genellikle adaptif kontrol tekniklerinin kullanımını gerektirmektedir.

Adaptif kontrol ilk olarak Draper ve Li [41] tarafından 1951 yılında önerilmiştir. Draper ve Li, performans karakteristiklerindeki belirsizlikler karşısında içten yanmalı bir motorun performansını optimize edebilecek bir kontrol sistemi üzerinde çalışmışlardır. "Adaptif" terimi kontrol literatüründe ilk olarak 1954 yılında Tsien [42] tarafından insan beyninin Ashby [43] modelini tasviri sırasında kullanılmıştır. 1955' de Benner ve Drennick [44] adaptif karakteristikli bir kontrol sistemini gündeme getirmişlerdir. Adaptif kontrollerle ilgili en büyük adım 1958' de Whitaker ve arkadaşları [45] tarafından gerçek ve istenen davranış arasındaki hata sinyallerini elde etmek için bir referans modelinin kullanıldığı adaptif uçuş kontrol sistemlerinin incelenmesi ile atılmıştır. Bu hata sinyalleri, belirsizlikler ve değişen sistem dinamiğine rağmen ideal davranışı elde etmek üzere kontrolör parametrelerinde değişiklik yapılması için kullanılmıştır. Bu sistemler daha sonra "Model Referans Adaptif

Kontrol Sistemleri (MRAC)" olarak adlandırılmıştır.

Li ve Van der Velde [46] 1960' da, adaptif kontrol sisteminin parametre değişmelerini otomatik olarak dengeleyen bir başka şekli üzerinde çalışmışlardır. Bu sisteme kendi kendine salınan adaptif sistem (self-oscillating adaptive system) denmektedir.

Petrov ve arkadaşları [47], 1963' de değişken yapıllı sistemler (variable structure systems) olarak adlandırılan adaptif sistem dizaynı ile ilgilenmişlerdir.

Bellman [48] ve Fel'dbaum [49], 1960-61 yıllarında değişik belirsizliklere sahip sistemler için kontrolör dizaynında dinamik programlama kavramının kullanımı ile ilgili çalışmalar yapmışlardır.

Lainiotis [50], 1971' de adaptif kontrole hesapsal açıdan çok ekonomik ve güçlü olan parçalara ayırma (partitioning) yaklaşımını ilave etmiştir.

Aström ve Wittenmark [51], adaptif kontrolün diğer önemli bir şekli olan kendi kendini ayarlayan regülatörlerin gelişmesini sağlamışlardır. Kendi kendini ayarlayan regülatörler, mikroprosesörlerin kullanılmaya başlaması ile uygulanışı gayet kolay olan adaptif kontrol sistemleri durumuna gelmiştir.

Adaptif kontrol, önemli teorik ve algoritmik ilerlemeler neticesinde artık bir olgunluk derecesine erişmiş bulunmaktadır ve önemli uygulama alanlarına sahiptir. Uçuş kontrolü, güdüm ve füze kontrolü, proses kontrolü, elektrik motorlarının kontrolü, gemilerde rota kontrolü, metalurjik prosesler, uydu kontrolü, robot kontrolü v.s.

Adaptif kontrolörlerin dizayn ve analizi ya sürekli zaman (Laplace Dönüşümü) veya ayrık zaman (z dönüşümü) modeller kullanılarak gerçekleştirilmektedir. Yapılan çalışmalar, MRAC sistemlerinin sürekli zaman modeline, STR' lerin ise ayrık zaman modeline göre dizayn edildiğini göstermektedir. Fakat model referans adaptif kontrol sistemlerinin ayrık zaman modeller ile, kendi kendini ayarlayan regülatörlerin sürekli zaman modeller ile kullanılmaları da mümkün olmaktadır [52].

Aşağıda, model referans kontrol sistemleri ve kendi kendini ayarlayan regülatörler kısaca özetlenecek, tez içeriğine dahil olduğundan kendi kendini ayarlayan regülatörler ayrıca incelenecektir.

2.2.1. Model Referans Adaptif Kontrol

Model referans adaptif kontrol sistemleri, sistem dinamiği performansının belirlenmesi gerekmediğinden, yüksek adaptasyon hızı ile elde edilmesi nispeten kolay sistemler olarak sonuç verdiğinden önemli olmaktadır. İlk olarak Whitaker ve MIT' deki çalışma arkadaşları tarafından 1958' de adaptif otomatik pilot dizaynı için geliştirilmişlerdir [45].

Model referans adaptif kontrol sisteminde hedef; bilinmeyen sistem çıkışını, kontrol sisteminin bir parçası olan verilen referans modelin çıkışına asimptotik olarak yakınsatmaktır (Şekil 2.2).

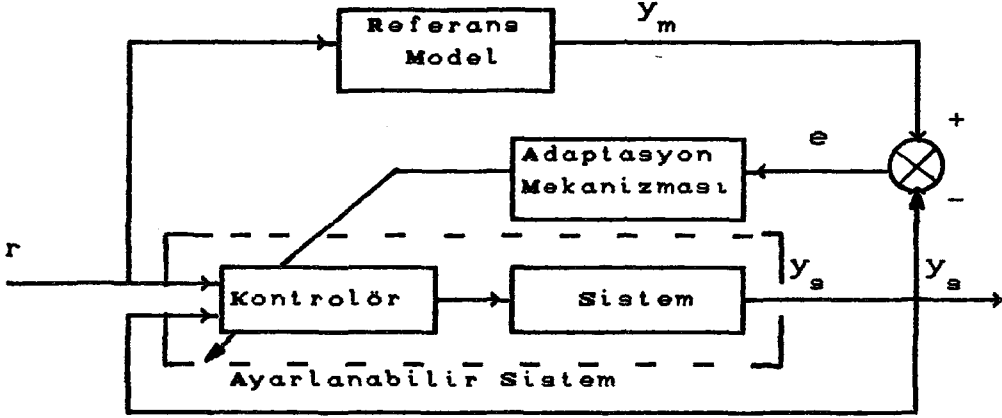
Şekil 2.2' deki model referans adaptif kontrol sisteminde bir referans performans indeksi kullanılmaktadır. Bu referans performans indeksini oluşturmak için de bir referans modeli kullanılmaktadır. Referans modeli ayarlanabilir sistem ile aynı girişi kullanmaktadır. Bu

durumda, verilen performans indeksi ve ölçülen performans indeksi arasındaki karşılaştırma, direkt olarak referans model ve ayarlanabilir sistemin çıkışları arasında yapılmaktadır. Ortaya çıkan hata, kontrolör parametrelerini değiştirmek için adaptasyon mekanizmasında kullanılmakta ve böylece hata minimize edilmektedir. Bu, parametrelili bir adaptif sistemdir (Şekil 2.2). Referans model ve ayarlanabilir sistem çıkışları arasındaki farkı minimize edecek şekilde, bir yardımcı giriş sinyali oluşturacak adaptasyon mekanizması da düzenlenebilir. Bu durumda sistem, sinyal sentezli (signal-synthesis) model referans adaptif kontrol sistemi adını almaktadır (Şekil 2.3). Temel problem, sadece hatanın sıfır olacağı değil aynı zamanda elde edilen sistemin kararlı olacağı bir adaptasyon mekanizması oluşturmaktır. Bu, zor bir problemdir ve kararlılık açısından model referans adaptif kontrol sistemlerinin dizaynında araştırmalar devam etmektedir.

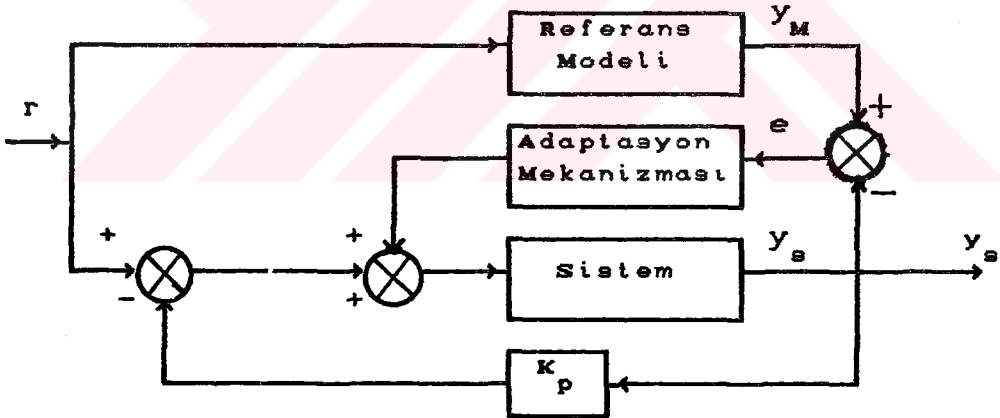
Model referans kontrol sistemlerini şu şekilde sınıflandırmak da mümkündür. Şekil 2.2' de görülen model referans adaptif kontrol sistemi "direkt" olarak adlandırılmaktadır, çünkü, kontrolör parametreleri model ve sistem çıkışları arasındaki hatayı minimize edecek şekilde direkt olarak yenilenmektedir. Kontrolör parametrelerinin dolaylı olarak yenilendiği ve bu yüzden "endirekt" olarak adlandırılan model referans adaptif kontrol sistemleri de bulunmaktadır. Endirekt kontrol durumunda sistem parametreleri kestirilmekte, kontrolör parametreleri ise bu kestirilmiş değerlere dayanarak düzenlenmektedir.

Model referans kontrol sistemleri, genellikle deterministik sistemlerde uygulanmakta, stokastik bozucular karşısındaki model referans adaptif kontrol sistemleri ile ilgili araştırmalar ise devam etmektedir. Stokastik kendi kendini ayarlayan regülatörlerin

gelişiminde ise, şu anda önemli ilerlemeler kaydedilmiş olduğunu unutmamak gerekir.



Şekil 2.2. Model Referans Adaptif Kontrol Sistemi
(Parametre Adaptasyonu)



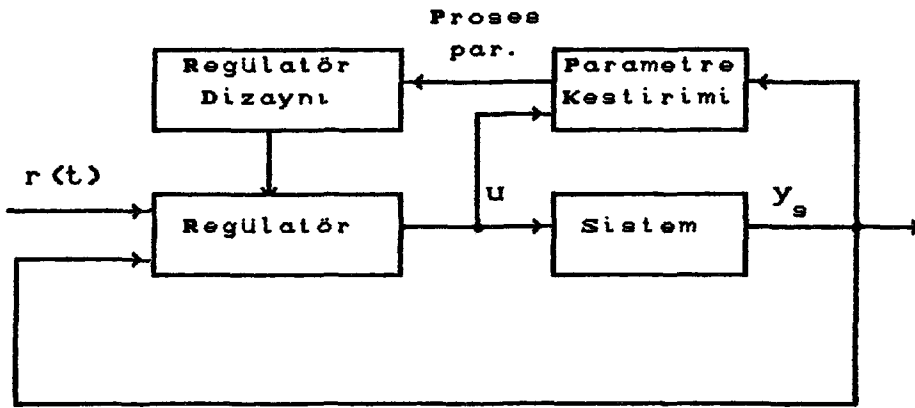
Şekil 2.3. Model Referans Adaptif Kontrol Sistemi
(Sinyal Sentezi Adaptasyonu)

2.2.2. Kendi Kendini Ayarlayan Regülatörler

Kendi kendini ayarlayan regülatörler (STR), adaptif kontrol sistemlerinin diğer önemli bir şeklidir, fikir olarak ilk defa 1973' de Aström ve Wittenmark [51]

tarafından ortaya atılmıştır. Kendi kendini ayarlayan regülatörlerde önce, bilinen sistem parametreleri için bir dizayn prosedürü seçilir. Bu prosedür, parametrelerin ardışık olarak kestirilen değerleri kullanılarak, bilinmeyen sisteme uygulanır. Böyle bir sistem şekil 2.4' de gösterilmiştir. Söz edilen kendi kendini ayarlayan regülatör, bölüm 2.3.1.2' de anlatılacağı gibi dolaylı olarak bir proses modelinin kestirilmesine dayandığından endirekt kendi kendini ayarlayan regülatörler olarak bilinmektedir. Bazen; proses, regülatör parametrelerine göre yeniden parameterize edilebilir. Bu, dizayn hesaplamalarından kaçınıldığından daha basit algoritmalar sağlamaktadır. Bu şekildeki bir kendi kendini ayarlayan regülatör, direkt regülatör olarak da tanımlanmaktadır. (Bakınız, bölüm 2.3.1.2)

Kendi kendini ayarlayan regülatörler, son yıllarda mikroprosesörler ile kullanılabilmelerinden ve değişik uygulama alanları bulmasından dolayı hayli güncel olmuştur. Kendi kendini ayarlayan adaptif kontrolörlerin, şayet iyi dizayn edilirler ise bozucular (özellikle ölçülemeyen) olması durumunda adaptasyon kabiliyeti avantajı da vardır.



Şekil 2.4. Kendi Kendini Ayarlayan (Self-tuning) Regülatör

2.3.Kendi Kendini Ayarlayan Kontrol Algoritmaları

Adaptif bir sistemde, kontrolör parametrelerinin devamlı olarak ayarlandığı kabul edilmektedir. Bu, kontrolör parametrelerinin prosesdeki değişimleri takip etmesi demektir. Fakat, böyle sistemlerde yakınsama ve kararlılık özelliklerini analiz etmek zor olmaktadır. Problemi basitleştirmek için, prosesin sabit fakat bilinmeyen parametrelere sahip olduğu kabul edilmektedir. Şayet proses biliniyorsa, dizayn işlemi istenen kontrol parametrelerinin tayin edilmesinden ibaret olmaktadır. Adaptif kontrolör, proses bilinmese bile bu parametre değerlerine yakınsamaktadır. Bu özellikli bir regülatör, istenen performansı otomatik olarak sağladığından kendi kendini ayarlayan regülatör olarak adlandırılmaktadır.

Endüstriyel prosesler için kontrolörlerin otomatik olarak ayarlanması, uzun yıllar hem teorik hem de pratik açıdan büyük ilgi uyandırmıştır. Kendi kendini ayarlayan kontrol algoritmaları, otomatik kontrol problemine yeni bir yaklaşım getirmiştir. Gerçekleştirilmeleri kolaydır ve bilinmeyen parametreler, zaman gecikmeleri, zamanla değişen proses dinamiği ve stokastik bozucular içeren değişik karakteristikli karmaşık proseslere uygulanabilmektedirler. Şekil 2.4' de görüldüğü gibi kendi kendini ayarlayan kontrol algoritması, esas olarak kontrolör dizaynı ile parametre kestiriminin birleşiminden meydana gelmektedir. Kontrolör dizaynından parametrelerin kestirilmesi işlemi ayrılmıştır. Bilinmeyen parametreler, ardışık bir parametre kestirme metodu kullanılarak çevrim-ici kestirilir. Değişik parametre kestirme algoritmaları kullanılabilmektedir (Bakınız, Bölüm 2.5). Dizayn metodlarına örnek olarak ise minimum-varyans, lineer kuadratik pol atama ve model izleme yöntemleri gösterilebilir. Parametre kestirme ve dizayn metodlarının çeşitli kombinasyonları ile farklı özelliklere sahip regülatörler elde edilebilmektedir.

Kendi kendini ayarlama kavramı ilk olarak Kalman [53] tarafından 1958' de kullanılmıştır. Kalman, algoritmasını özel amaçlı bir bilgisayarda denemiştir. Teorinin yeterince gelişmemiş olması ve bugünkü düzeyde gelişmiş bilgisayarların bulunmaması sebebi ile bu teknik, Peterka [54], 1970' deki makalesinde gürültülü sistemlerin adaptif ve sayısal olarak ayarlanması fikrini genişletinceye kadar geniş bir şekilde ele alınmamıştır. Bu konuda en büyük atılım 1973' de Aström ve Wittenmark [51] 'ın kendi kendini ayarlayan regülatörü ile başlamıştır. Teorik gelişmeler, kendi kendini ayarlama algoritmalarının formülasyonuna değişik yaklaşımlar getirerek devam etmektedir.

Kendi kendini ayarlayan kontrol algoritmaları ile ilgili olarak bugüne kadar yapılan çalışmalar, önce tek giriş-tek çıkışlı sistemler (Single Input-Single Output, SISO) ve daha sonra çok girişçok çıkışlı (Multi Input-Multi Output, MIMO) sistemler için aşağıda incelenmiştir.

2.3.1. Kendi Kendini Ayarlayan Regülatörler (SISO)

2.3.1.1. Minimum-Varyans Kendi Kendini Ayarlayan Regülatörler

İlk kendi kendini ayarlayan regülatör, yukarıda da bahsedildiği gibi stokastik minimum-varyans prensibine dayanarak Aström ve Wittenmark tarafından geliştirilmiştir [51]. Bu, optimal kontrol amaçlı bir yaklaşımdır. Sistemin minimum faz olması, zaman gecikmesi ve sistem mertebesinin bilinmesi gerekmektedir. Performanstan amaçlanan, örnekleme anlarında ölçülen proses çıkışının değişiminin minimize edilmesidir. Yapılabilecek tek değişiklik, kullanıcının örnekleme zamanını

değiştirebilmesidir. Kontrol gücü göz önüne alınmadığından, büyük kontrol sinyallerinin kararsızlığa yol açma ihtimali vardır. Kararsızlık, hızlı örnekleme veya büyük zaman gecikmelerinden dolayı da ortaya çıkabilir. Bu yüzden, bu tip bir kendi kendini ayarlayan regülatör dikkatli bir şekilde kullanılmalıdır.

Clarke ve Gawthrop [55,56] ve Gawthrop [57], yukarıdaki fikri daha geniş bir performans kriteri içerecek şekilde genişletmişlerdir. Bu yaklaşım, kontrol ve çıkış değişmelerinin bir kombinasyonunun minimize edilmesini amaçlamaktadır. Clarke' nin yaklaşımı Aström'ün kendi kendini ayarlayan regülatörü ile sağlanamayan non-minimum faz sistemleri kontrol etmek için de kullanılabilmektedir. Minimum-varyans kendi kendini ayarlayan regülatörler ile ilgili daha detaylı bir çalışma ve uygulaması Aström ve arkadaşları [58] tarafından gerçekleştirilmiştir. Grimbale [59], ağırlıklı minimum-varyans kendi kendini ayarlayan bir kontrol algoritması önermiştir. Önerilen bu algoritma, sadece çıkış değişiminin ele alındığı durumdakinden daha geniş bir kontrol aralığına izin vermektedir ve bozucuların ve dinamiklerin gerçek yapısının bilinmediği durumlarda da kullanılabilmektedir. Kontrolör, diğer kontrol kanunlarının kararsız olduğu kontrol aralığı içinde de kararlı olmaktadır. Sistem minimum faz olduğu zaman Grimbale tarafından geliştirilen algoritma Clarke ve Gawthrop tarafından geliştirilen genelleştirilmiş minimum-varyans kendi kendini ayarlayan kontrolöre eşdeğer olmaktadır. Fakat bu tip algoritmalar, sistemde non-minimum faz terimlerine müsaade edebilmek için daha karmaşık olmaktadır. Peterka [60], öngörütü (predictor) esaslı kendi kendini ayarlayan kontrol önermiştir. Toivonen [61], giriş büyüklüğü üzerinde kısıtlamaların olduğu minimum-varyans kontrol üzerinde çalışmalar yapmıştır.

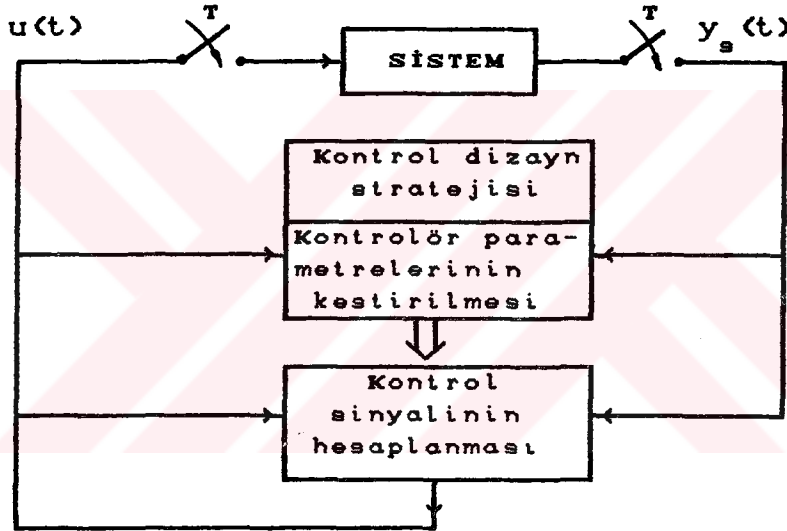
2.3.1.2. Direkt (Implicit) ve Endirekt (Explicit) Kendi Kendini Ayarlama

Kendi kendini ayarlayan regülatörler, bilindiği gibi proses parametrelerinin kestirilmesi fikrine dayanmaktadır. En çok kullanılan yaklaşım, proses transfer fonksiyonunun ve bozucuların parametrelerinin kestirilmesidir. Bu bir endirekt adaptif kontrol algoritmasını vermektedir. Regülatör parametreleri direkt olarak kestirilmemekte, proses parametrelerinin kestirilen değerlerinden dolaylı olarak elde edilmektedir. Regülatör parametrelerinin direkt olarak yenilenmesi ile direkt (implicit) kendi kendini ayarlayan kontrol algoritmaları elde edilmektedir. Direkt kendi kendini ayarlayan regülatörler, öngörülü kontrol teorisine dayanarak geliştirilmişlerdir ve sistem zaman gecikmesi hakkında bilgiye gereksinimleri vardır. Bundan kaçınmak için, zaman gecikmesinin proses dinamiğinin bir parçası olarak kestirilebildiği, endirekt kendi kendini ayarlayan regülatörler geliştirilmiştir. Endirekt kendi kendini ayarlayan kontrol algoritmasında, parametre tanımlamasını ayrı bir kontrol kanunu hesaplaması izlemektedir. Endirekt metodlar daha fazla hesaplama gerektirmektedirler.

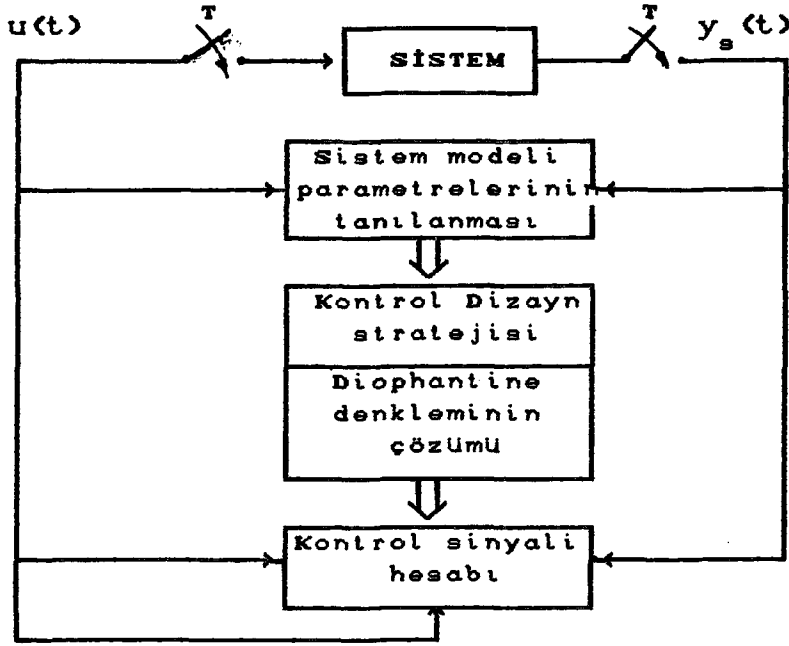
Direkt kendi kendini ayarlayan bir regülatörün asimtotik özelliklerinin analizi, "self-tuning" terimini de ilk olarak kullanan Aström ve Wittenmark tarafından 1973 yılında en küçük kareler parametre kestirme algoritması ve minimum-varyans kontrol algoritması kullanılarak yapılmıştır. Benzer kendi kendini ayarlayan regülatörler, Peterka [54] ve Clarke-Gawthrop [55] tarafından da incelenmiştir.

Endirekt kendi kendini ayarlayan regülatörler ile ilgili olarak da, ilk defa Kalman [53] tarafından 1958' de en küçük kareler parametre kestirme algoritmasına ve

"dead-beat" kontrola dayanan bir regülatör üzerinde çalışılmıştır. Daha sonra yine en küçük kareler parametre kestirme algoritmasını ve minimum-varyans kontrol algoritmasını temel alan bir kontrolör, 1971' de Wieslander ve Wittenmark [62] tarafından incelenmiştir. Ayrıca Fuchs [63] tarafından da endirekt kendi kendini ayarlayan bir kontrol yaklaşımı incelenmiştir. Şekil 2.5' de bir direkt kendi kendini ayarlayan kontrol akış şeması, şekil 2.6' de ise endirekt kendi kendini ayarlayan kontrol akış şeması görülmektedir.



Şekil 2.5. Direkt Kendi Kendini Ayarlayan Kontrol Çevrimi



Sekil 2.8. Endirekt Kendi Kendini Ayarlayan Kontrol Çevrimi

2.3.1.3. Pol Atamalı Kendi Kendini Ayarlayan Regülatörler

Pol atama esaslı, endirekt kendi kendini ayarlayan bir kontrol stratejisi Wouters [64] tarafından incelenmiştir. Bu yaklaşım Wellstead ve arkadaşları [65] tarafından biraz daha genişletilmiştir. Bu metod, sıfırlar açık çevrim pozisyonlarında kalırken kapalı çevrim pollerinin istenen geçici cevabı tanımlayacak şekilde belirlenmiş yerlere atanmasını sağlamaktadır. Çalışmadaki süreklilik (robustness) bu metodun en önemli özelliğidir. Ayrıca, bu tip kendi kendini ayarlayan regülatörler non-minimum faz sistemlere de uygulanabilir. Non-minimum faz davranışı bir çok pratik uygulamalarda karşımıza çıktığından, bu çok önemli bir avantaj teşkil etmektedir. Pol atamalı kendi kendini ayarlayan regülatörlerin diğer önemli bir özelliği, diğer metodlar ile

kolayca kullanılamayan, bilinmeyen hatta değişen zaman gecikmeli sistemleri ayar edebilme kabiliyetidir. Aström ve arkadaşları [66]' da pol atamalı kontrolörleri incelemiştir, fakat yaptıkları çalışma Wellstead ve arkadaşlarının [65] çalışmasından biraz farklıdır, en önemli fark Wellstead ve arkadaşlarının sadece servo problemi ile ilgilenmeleri ve kontrolörün elde edilmesi sırasında hiç bir stokastik bozucuyu göz önüne almamalarıdır.

Allidina ve Hughes [67], süreklilik elde edecek ve referans takibini kolaylaştıracak şekilde pol atama ve Clarke' nin yaklaşımını birleştiren genelleştirilmiş bir kendi kendini ayarlayan kontrolör önermişlerdir. Hem kararsız hem de non-minimum faz karakteristikler ele alınabilmektedir, fakat değişen zaman gecikmeleri durumunda olumsuz sonuçlar vermektedir. Pol atama tekniklerinin hiç birisinin bilinmeyen bozucularla etkili bir şekilde uğraşamayacağını da belirtmek faydalı olacaktır.

Vogel ve Edgar [68,69], değişebilen zaman gecikmesine sahip sistemler için etkili olan ve non-minimum faz sistemlere de uygulanabilen bir adaptif pol-sıfır atamalı ölü zaman kompensatörü geliştirmişlerdir.

2.3.1.4 LQG Kendi Kendini Ayarlayan Regülatörler

Kendi kendini ayarlayan regülatör dizaynı için diğer bir yaklaşım, dizayn prosedürü lineer-kuadratik-Gaussian (LQG) kontrolü için uygulanacak şekilde, kestirilen modeli eşdeğer durum uzayı formuna dönüştürmektir. Bu şekildeki bir LQG dizaynı ilk olarak 1973' de Peterka ve Aström [70] tarafından tanıtılmıştır. Önerdikleri proses modeli, renkli gürültülü prosesleri kapsam dışı bırakmakta idi. LQG kendi kendini ayarlayıcının mikro-komputer uygulaması Zhao-Ying ve Aström [71] tarafından

incelenmiştir. LQG metodu daha fazla hesaplama yükü gerektirmektedir, fakat hesaplama yükünü azaltmak mümkündür [72]. Hem durum uzayı ve hem de giriş-çıkış modelini kullanan bir endirekt LQG regülatörü Salcudean ve Belanger [73] tarafından incelenmiştir. Grumble [74], hesaplama yükünü azaltan direkt LQG kendi kendini ayarlayıcılar üzerinde çalışmalar yapmıştır.

2.3.1.5. Non-Minimum Faz Sistemler İçin Kendi Kendini Ayarlayan Regülatörler

Daha önce de belirtildiği gibi, endüstriyel proseslerin büyük bir çoğunluğu non-minimum faz davranış göstermektedirler. Bu yüzden de kendi kendini ayarlayan regülatörlerin non-minimum faz sistemler ile de çalışabilmesi gerekmektedir. Non-minimum faz sistemler için kendi kendini ayarlayan kontrol algoritması çalışmaları Aström ve arkadaşları [75,76], Samson ve arkadaşları [77], Grumble [78], Kumar ve Moore [79] tarafından yapılmıştır. Clarke [80], non-minimum faz sistemlerde kararsızlık problemini ortadan kaldırabilmek için değişik yaklaşımları gözden geçirmiş, uygun dizayn ve mühendislik sağ duyusu ile birlikte kendi kendini ayarlayan kontrolün, non-minimum faz sistemlere de uygulanabileceğini göstermiştir.

2.3.1.6. Karma (Hybrid) Kendi Kendini Ayarlama

Karma kendi kendini ayarlama, her ikisinin avantajlarını sağlayabilmek için sürekli zaman ve ayrık zaman sistemlerin özelliklerini birleştirmektedir. Gawthrop [81], karma kendi kendini ayarlayan kontrol algoritmaları üzerinde çalışmalar yapmıştır.

2.3.2.Kendi Kendini Ayarlayan Reg lat rler (MIMO)

End striyel proseslerin b y k bir  o unlu u  ok de i kenlidir. Bu y zden, son yıllarda  ok de i kenli kendi kendini ayarlayan kontrol algoritmalarında da  nemli geli meler olmu tur. Bu b l mde, bu alanda yapılan  alı malar kısaca  zetlenecektir.

 ok de i kenli kendi kendini ayarlayan reg lat rler, ilk olarak Peterka ve Astr m [70] tarafından incelenmi  ve problem,  rnekleme anlarında ayırık Riccati denkleminin ardışık  z m  ile LQG form lasyonuna d n  t r lm  t r. Borison [40], ardışık en k  k kareler y ntemi ile parametre kestirimi yapmı  ve kestirilen bu de erlerden direkt olarak elde edilen bir lineer kontrol r kullanarak, minimum Σ varyans stratejisini temel alan  ok de i kenli kendi kendini ayarlayan bir reg lat r  nermi tir. Bu y ntem, sadece giri  ve  ıkı  b y kl kleri sayısının e it oldu u minimum faz sistemlere uygulanabilmektedir. Borison' un yakla ımı proses parametreleri yerine kontrol r parametrelerinin kestirilmesine dayanmaktadır.

Bayoumi ve El-Bagoury [82,83], giri  b y kl  u ve referans vekt r  ile  ıkı  b y kl  u arasındaki farkdan olu an, kuadratik bir maliyet fonksiyonunun minimize edilmesine dayanan bir kontrol r  incelemi lerdir. Onlar,  ok de i kenli kendi kendini ayarlayan bir reg lat r  nermi lerdir. Bu y ntemde, proses modeli parametreleri  nce en k  k kareler metodu ile kestirilmekte ve daha sonra kestirilen parametreler rasgele bir sinyal ile, arttırılmı  kontrol b y kl   n  hesaplamak i in kullanılmaktadır. Bu kombinasyonun ge ici cevap  zellikleri, temel  ok de i kenli kendi kendini ayarlayan reg lat rlere g re iyile tirilmi tir. Giri  ve  ıkı  b y kl  klerinde meydana gelen sapmalar azaltılmı tır.

Koivo [84], Clarke ve Gawthrop' un metodunu [55,56] aynı giriş ve çıkış sayısına sahip çok değişkenli sistemlere genişletmiştir. Kontrolör, maliyet fonksiyonunda kontrol büyüklüklerinin cezalandırılması ve daha sonra da kontrol kanununun hesaplanması ile elde edilmektedir. Borison [40] tarafından önerilenden biraz farklı olarak "d" adım sonrasını (d: zaman gecikmesi) öngeren bir öngörücü kullanılmıştır. Kare kök formunda bir ardışık en küçük kareler parametre kestirme yöntemi kontrolör parametrelerini tanılamaktadır. Zamanla değişen referans sinyalleri de izlenebilmekte ve sistem non-minimum faz olabilmektedir. Böylece bu metod Borison' un minimum-varyans kontrolörünün sahasını genişletmektedir.

Prager ve Wellstead [85], minimum-varyans kontrolörün dezavantajlarını ortadan kaldırmak için pol-atama fikrini çok değişkenli durum için genişletmişlerdir. Minimum-varyans kontrolörler, sistem zaman gecikmelerindeki değişmelere karşı duyarlı olmaktadır. Ayrıca minimum-varyans dizaynı gerçek sistemlerde genellikle mümkün olmayan tüm çevrimlerin aynı zaman gecikmesine sahip olmasını gerektirmektedir. Pol atamalı regülatörlerde bu güçlükler ortadan kalkmaktadır. Pol atamalı bir regülatörde en büyük dezavantaj, maksimum müsaade edilebilir örnekleme frekansını kısıtlayan hesaplama karmaşıklığından ortaya çıkmaktadır. Bu da, daha hızlı ve daha güçlü mikroprosesörlerin geliştirilmesi ile problem olmaktan çıkmaktadır.

Bezanson ve Harris [86], özdeğer atamaya dayanan çok değişkenli kendi kendini ayarlayan regülatörlerin durum değişkeni dizaynını incelemişler ve durum değişkeni geri beslemesi kullanmışlardır. Bu yaklaşım, non-minimum faz sistemler ve açık çevrimli karasız sistemler için de kullanılabilir.

2.4. Çok Değişkenli Kendi Kendini Ayarlayan Bir Regülatör Dizaynı

2.4.1 Genel Bilgiler

Tezin esas konusunu teşkil etmekte olan "öngörülü kontrol" algoritmalarına girmeden önce, yukarıdaki bölümlerde bugüne kadar yapılan adaptif kontrol çalışmaları hakkında bir ön bilgi verilmiş ve öngörülü kontrolün çıkış kaynağı olan kendi kendini ayarlayan kontrol algoritmaları hakkında da daha geniş bir kaynak araştırması yapılmıştı.

Bu bölümde ise, gerçekten önemli bir kontrol algoritması olan kendi kendini ayarlayan kontrol algoritmalarının nasıl çalıştığını daha iyi anlayabilmek için, bu konuda bir temel teşkil eden ve anlaşılması nispeten kolay olan Borison' un yaklaşımı [40] çok değişkenli durum (MIMO) için özetlenecektir.

Kontrol algoritması kurulduktan sonra, fazla bir değişikliğe gerek olmadan tek değişkenli durum (SISO) için de algoritmayı kullanmak mümkün olmaktadır. Algoritma teorik olarak aşağıda özetlenmiş, 3 eklemlili bir robotik manipülatörün eklem hareketlerinin kontrolü için gerçekleştirilen simülasyon çalışması da bölüm 7.3' de verilmiştir.

2.4.2 Problemin Formülasyonu

Cok değişkenli bir proses modeli genel olarak aşağıdaki formda gösterilmektedir (CARMA modeli, Controlled-Auto-Regressive-Moving-Average)

$$A(q^{-1})y(t) = B(q^{-1})u(t-d) + C(q^{-1})\xi(t) \quad \dots (2.1)$$

Burada;

t : Zaman,

d : Proses zaman gecikmesi,

$u(t)$: t anındaki m boyutlu giriş vektörü,

$y(t)$: t anındaki m boyutlu çıkış vektörü,

$\xi(t)$: sıfır ortalamalı, m boyutlu beyaz Gaussian gürültü vektörü,

$A(q^{-1})$, $B(q^{-1})$ ve $C(q^{-1})$ ($m \times m$) boyutlu polinom matrisleri, q^{-1} geriye doğru kaydırma operatörüdür.

Polinom matrislerini aşağıdaki şekilde yazmak mümkündür.

$$A(q^{-1}) = I + \sum_{j=1}^{n_a} A_j q^{-j}$$

$$B(q^{-1}) = B_0 + \sum_{j=1}^{n_b} B_j q^{-j}$$

$$C(q^{-1}) = I + \sum_{j=1}^{n_c} C_j q^{-j}$$

Burada, proses modelinin n_a ve n_b mertebelerinin bilindiği farz edilmekte ve $C(q^{-1}) = I$ olarak kabul edilmektedir.

$A(q^{-1})$ ve $B(q^{-1})$ ' in bilinmeyen parametrelerinin kestirilmesi gerekmektedir. Bu kestirilen değerler daha sonra, kuadratik bir maliyet fonksiyonunun minimize edilmesi ile elde edilen kontrolör algoritmasında kullanılacaktır. Kontrol algoritmasının ana hedefi, çıkış büyüklüğünü, referans büyüklüğünü gösteren y_r referans vektörüne karesel ortalama hassasiyetinde yakın elde etmektir.

Proses modelini aşağıdaki formda yazmak mümkündür.

$$\begin{aligned}
y(t) &= B_0 u(t-d) + \dots + B_{n_b} u(t-d-n_b) - A_1 y(t-1) - \\
&\quad - \dots - A_{n_a} y(t-n_a) + \xi(t) \\
&= \theta^T \phi(t-1) + \xi(t) \quad \dots (2.2)
\end{aligned}$$

Burada;

$$\theta = [B_0, B_1, \dots, B_{n_b}; -A_1, -A_2, \dots, -A_{n_a}]^T \quad \dots (2.3)$$

$$\begin{aligned}
\phi(t-1) &= [u^T(t-d), u^T(t-d-1), \dots, u^T(t-d-n_b), y^T(t-1), \dots, \\
&\quad y^T(t-n_a)]^T \quad \dots (2.4)
\end{aligned}$$

T indisi ters alma (transpoze) işlemini göstermektedir.

Bir parametre kestirme algoritması kullanılarak (2.3) ifadesindeki θ parametreleri kestirilmektedir.

(t) anına kadar gözlenmiş giriş-çıkış bilgilerinden ve kestirilen parametrelerden oluşan model,

$$y(j) = \hat{\theta}^T(t) \phi(j-1), \quad j = 1, 2, \dots, t \quad \dots (2.5)$$

şeklinde elde edilir. $\hat{\theta}(t)$ matrisi, t anına kadar gözlenmiş önceki giriş ve çıkış büyüklüklerinden elde edilen bilgilere dayanarak θ parametre matrisinin kestirilen değerlerini göstermektedir.

Dizayn edilecek kontrolörün amacı, aşağıdaki kuadratik maliyet fonksiyonunu minimize etmektir.

$$J = \| E(y(t+d|t) - y_r) \|_{Q_1}^2 + \| u(t) \|_{Q_2}^2 \quad \dots (2.6)$$

Burada, Q_1 pozitif yarı tanımlı bir simetrik

matris ve Q_2 pozitif tanımlı ve simetrik bir matristir. $\|X\|_Q^2 = X^T \cdot Q \cdot X$ olarak tanımlanmaktadır. E beklenti operatörü olarak isimlendirilmektedir. $y(t+d|t)$, $(t-1)$ anına kadar önceden gözlenen değerlere dayanarak (t) anında iken $(t+d)$ anında beklenen çıkış büyüklüğüdür.

Maliyet fonksiyonunda (2.6)' dan da görüldüğü gibi, kontrol büyüklüğü $u(t)$ ile kestirilen çıkış büyüklükleri ve referans vektörü arasındaki sapmalar minimize edilmektedir.

$(t+d)$ anında beklenen çıkış vektörü aşağıdaki şekilde yazılabilir.

$$\begin{aligned} E(y(t+d|t)) &= \hat{y}(t+d|t) = \hat{\theta}^T(t) \hat{\phi}(t+d-1) \\ &= \hat{B}_0 u(t) + \hat{B}_1 u(t-1) + \dots + \hat{B}_{n_b} u(t-n_b) - \\ &\quad - \hat{A}_1 \hat{y}(t-1+d|t) - \dots - \hat{A}_{d-1} \hat{y}(t+1|t) - \hat{A}_d y(t) - \\ &\quad - \dots - \hat{A}_{n_a} y(t-n_a+d) \quad \dots \quad (2.7) \end{aligned}$$

Referans büyüklüğü y_r ile kestirilen çıkış büyüklüğü arasındaki sapma miktarı,

$$\hat{y}(t+d|t) - y_r = \hat{B}_0 u(t) + \hat{C}(t) \quad \dots \quad (2.8)$$

şeklinde ifade edilebilir. Burada,

$$\hat{C}(t) = \sum_{i=1}^{n_b} \hat{B}_i u(t-i) - \sum_{j=1}^{d-1} \hat{A}_j \hat{y}(t-j+d|t) - \sum_{k=d}^{n_a} \hat{A}_k y(t-k+d) - y_r \quad \dots \quad (2.9)$$

(2.8) nolu ifade (2.6)' da yerine konursa,

$$J = \| \hat{B}_0 u(t) + \hat{C}(t) \|_{Q_1}^2 + \| u(t) \|_{Q_2}^2 \quad \dots \quad (2.10)$$

elde edilir. Buradan da J' yi minimize eden kontrol

$$u(t) = -(\hat{B}_0^T Q_1 \hat{B}_0 + Q_2)^{-1} \hat{B}_0^T Q_1 \hat{C}(t) \quad \dots (2.11)$$

olarak bulunur.

(2.11) ifadesi ile verilen optimal kontrol kanunu her bir adımda aşağıdaki şekilde kolayca elde edilebilir: (2.5) modelinin uygulanması ile (t) anına kadar gözlenen giriş ve çıkış büyüklükleri kullanılarak $t+1, t+2, \dots, t+d$ anındaki çıkışlar (2.7) yardımı ile kestirilir. $(t+1)$ anında kestirilen çıkış büyüklüğü $(t+2)$ anındaki çıkış büyüklüğünü kestirmek için kullanılır ve işlem bu şekilde devam eder). (2.9) bağıntısı kullanılarak $\hat{C}(t)$ elde edilir. Daha sonra elde edilen bu $\hat{C}(t)$ ifadesi (2.11)' de $u(t)$ kontrol büyüklüğünü hesaplamak için kullanılır.

Dikkat edilecek bir nokta, Q_2 ceza matrisinin büyük değerlerinin kararsızlığa yol açmasıdır, bu yüzden Q_2 matrisinin elemanları dikkatli bir şekilde seçilmelidir.

Yukarıda da görüldüğü gibi Borison' un yaklaşımı olan algoritmada, (t) anına kadar ölçülmüş olan çıkış büyüklüğü değerleri kullanılarak $t+1, t+2, \dots, t+d$ anlarındaki çıkış büyüklüğü değerleri elde edilmektedir. Tezin esas konusu olan ve ilerideki bölümlerde bahsedilecek olan çeşitli üstünlüklere sahip uzun menzilli öngörü (long-range prediction) algoritmaları geliştirilene kadar, sadece yukarıdaki gibi (d) adım ilerisindeki çıkış büyüklüğü değerlerini öngören kontrol algoritmaları kullanılmıştır. Bu algoritmalar genel olarak tek menzilli (single-range) veya tek adım sonrasını öngören (one-step-ahead) öngörülü algoritmalar olarak adlandırılmaktadır. Yukarıda anlatılan algoritmada da sadece sistem zaman gecikmesi olan (d) adım ötesi öngörülebilmektedir. Fakat bu tip kendi kendini ayarlayan kontrol algoritmaları, uzun menzilli öngörülü algoritmaların çıkış kaynağı olması açısından önem taşımaktadır.

2.5. Parametre Kestirme Algoritmaları

2.5.1. Genel Bilgiler

Adaptif kontrol algoritmalarında proses parametreleri bilinmiyor kabul edildiğinden, bu parametrelerin bir kestirim algoritması kullanılarak tanınması gerekmektedir. Bu tanılama işlemi, ölçülen giriş-çıkış bilgileri kullanılarak en küçük kareler metodu ile yapılabilir.

Parametre kestirme metodları arasında, en küçük kareler (least-squares, LS) metodu bilinmeyen parametrelili sistemlerin parametre kestirimlerini elde etmek için kullanılan en yaygın tekniklerden birisidir. Ardışık en küçük kareler (Recursive least-squares RLS) metodu Plackett [87] tarafından klasik LS kestirme metodunun genişletilmiş hali olarak önerilmiş ve Kalman [53] tarafından çevrim-içi tanılama ve kontrol durumunda yaygın olarak kullanılacak bir şekle getirilmiştir.

Değişik parametre tanılama metodları vardır. Örneğin; ardışık en küçük kareler (RLS), kare kökler (SQR) ardışık kare kökler (recursive square roots RSQR) (Peterka [88]), ve ardışık U-D faktörizasyon metodu (Bierman [89]). Ayrıca, parametreleri zamanla değişen sistemlerle uğraşabilmek için eski bilgileri tedricen kaldırarak şekilde en küçük kareler kriteri içinde unutmama faktörü (μ) kullanılmıştır.

Bu bölümde; en çok kullanılan ve tezdeki değişik simulasyon çalışmalarında da uygulanan RLS ve RSQR parametre kestirme metodları incelenmiş ve ana hatları ile aşağıda özetlenmiştir.

2.5.2 Problemin Formülasyonu

İncelenecek proses , aşağıdaki giriş-çıkış ilişkisi ile ifade edilmektedir:

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_0 u(t-d) + \dots + b_{n_b} u(t-d-n_b) + \xi(t) \quad \dots (2.12)$$

Burada, u giriş, y çıkış, ξ gürültü (beyaz veya renkli), d proses zaman gecikmesi, a_i ($i=1, \dots, n_a$) ve b_j ($j=0, \dots, n_b$) bilinmeyen proses parametreleridir.

Kestirilecek (2.12) modelini aşağıdaki formda yeniden yazmak mümkündür:

$$y(t) = \hat{\theta}^T(t) \phi(t) + e(t) \quad \dots (2.13)$$

$$\hat{\theta} = [-\hat{a}_1, -\hat{a}_2, \dots, -\hat{a}_{n_a}; \hat{b}_0, \hat{b}_1, \dots, \hat{b}_{n_b}]^T$$

parametre vektörü ve

$$\phi(t) = [y(t-1), \dots, y(t-n_a); u(t-d), \dots, u(t-d-n_b)]^T$$

giriş ve çıkış verilerini içeren veri vektörüdür. $e(t)$, kestirme hatasıdır.

$\hat{\theta}(t)$, parametre kestirim vektörü, çevrim içi veya çevrim dışı olarak genellikle ardışık en küçük kareler algoritması veya benzer yöntemlerle (t) anına kadar mevcut giriş-çıkış verileri kullanılarak elde edilir.

$\phi(t)$ veri vektörü kullanılacak proses modeline bağlı olarak değişik formlarda ortaya çıkabilir. (Örneğin; ARMA modeli, CARMA modeli, CARIMA modeli, v.s)

Parametreler kestirilirken (2.13) ifadesindeki hata terimi $e(t)$ ' nin karesel toplam hassasiyetinde küçük

olması istenir. Hata kriteri, Θ parametre vektörüne bağlı olarak

$$E(\Theta) = \frac{1}{N+1} \sum_{i=1}^N e_i^2(t) \quad \dots (2.14)$$

şeklinde seçilmektedir. N ölçüm sayısını göstermektedir ve $e_i(t)$, e hata vektörünün i . bileşenidir. Problem, Θ parametre vektörüne göre $E(\Theta)$ 'yı minimize etme işlemidir. Problemin çözümü farklı algoritmalar için aşağıdaki şekillerde elde edilmektedir.

2.5.3 Ardışık Parametre Kestirme Metodları

2.5.3.1 Ardışık En Küçük Kareler Metodu

Bu metod, kendi kendini ayarlayan kontrolörlerde parametre kestirme aşamasında en çok kullanılan yöntemlerden biridir (Aström ve Wittenmark [51], Borison [40], Keviczky ve Hetthessy [90], Bayoumi [82,83]).

Özet olarak işlemler, aşağıdaki ardışık sıra içerisinde gerçekleştirilmektedir.

$$1- \text{Kazanç, } K(t) = \frac{P(t-1)\phi(t)}{\mu + \phi^T(t) P(t-1)\phi(t)} \quad \dots (2.15)$$

$$P(t) = [I - K(t)\phi^T(t)] \frac{P(t-1)}{\mu} \quad \dots (2.16)$$

şeklinde hesaplanmaktadır.

$$2- \text{Hata, } e(t) = y(t) - \hat{\Theta}(t-1)\phi(t) \quad \dots (2.17)$$

$$3- \text{Kestirimler, } \hat{\Theta}(t) = \hat{\Theta}(t-1) + \\ + K(t)[y(t) - \hat{\Theta}^T(t-1)\phi(t)]$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t) e(t) \dots (2.18)$$

ifadesi ile hesaplanmaktadır.

Başlangıç değeri olarak $P(0) = I_N / \delta$ olarak seçilmektedir. I_N tanım matrisi ve δ küçük bir değerdir ($\delta > 0$). μ unutma faktörüdür. $0 < \mu \leq 1$ değerlerini almaktadır.

2.5.3.2 Ardışık Kare Kök Algoritması

Kare kök algoritması Peterka [88] tarafından $P(t)$ matrisinin her kestirme aşamasında pozitif tanım özelliğini kaybetmemesi için önerilmiştir. $P(t)$ matrisinin $P(t) = S(t)S^T(t)$ şeklinde çarpanlara ayrılmış hali kullanılmaktadır. $S(t)$, her bir örnekleme anında yinelenen üst tarafı üçgen bir matris oluşturmaktadır. Ardışık kare kök kestirimcisi aşağıdaki ardışık sıra içinde çalışmaktadır.

$$1- \text{ Kare kök, } f = S^T(t-1)\phi(t-1) \dots (2.19)$$

$$\sigma_j = (\sigma_{j-1}^2 + f_j^2)^{1/2} \quad j=1, \dots, N$$

$$\sigma_0 = \sqrt{\mu} \dots (2.20)$$

N , $\hat{\theta}$ ve ϕ vektörlerinin boyutudur. Burada,

$$S(t)_{ij} = (\sigma_{j-1} / \sigma_0 \sigma_j) [S(t-1)_{ij} - (f_j / \sigma_{j-1}^2) \sum_{k=i}^N S(t-1)_{ik} f_k] \dots (2.21)$$

$$2- \text{ Kazanç, } K_i(t) = \sum_{k=i}^N S(t)_{ik} f_k / \sigma_N^2 \quad i=1, \dots, N \dots (2.22)$$

$$3- \text{ Hata, } e(t) = y(t) - \hat{\theta}(t-1)\phi(t-1) \quad \dots (2.23)$$

$$4- \text{ Kestirimler, } \hat{\theta}(t) = \hat{\theta}(t-1) + e(t)K^T(t) \dots (2.24)$$

Ardışık kare kök algoritması, Koivo [84] tarafından önerilen çok değişkenli kendi kendini ayarlayan kontrolörler de kullanılmıştır.

Yukarıda anlatılanlara ilave olarak değişik parametre kestirme algoritmaları bulunmasına rağmen, tez çalışmalarında kullanılmadığından ayrıca incelenmemiştir.



BÖLÜM 3. UZUN MENZİLLİ ÖNGÖRÜLÜ KONTROL ALGORİTMALARI

3.1. Giriş

Öngörülü kontrol algoritmaları, non-minimum faz, bilinmeyen veya değişen zaman gecikmeli ve değişik bozucular etkisindeki sistemlere rahatlıkla uygulanabilmeleri ve olumlu sonuçlar vermeleri nedeniyle modern kontrol teorisinin bir parçası haline gelmiştir. Kavramlara kısa bir zamanda hakim olunabildiğinden özellikle kısıtlı bir kontrol temeli olan teknik personele cazip gelmektedir. Yapısında doğal olarak bulunan ölü zaman kompenzasyonuna sahiptir. Ölçülen bozucuları dengelemek için ileri besleme kontrol işlemine doğal olarak sahip bulunmaktadır. Algoritmayı kısıtlanmış kontrol problemlerine uygulamak çok basittir. İlerisini görme özelliğine sahiptir, bu durum referans büyüklüklerinin önceden programlandığı durumlar için çok elverişlidir [91].

Öngörülü kontrol algoritmalarının minimum-varyans kontrolörlerden türetildiği söylenebilir (Aström ve Wittenmark [51], Clarke ve Gawthrop [55,56]). Bu kontrolörler, $y(t+d)$ çıkışı için sistem gecikmesi kadar bir öngörü kullanmakta ve gelecekteki hatanın fonksiyonu olan bir maliyet fonksiyonunu minimize etmektedirler. Maliyet fonksiyonu tek bir öngörülen çıkış büyüklüğü içerdiğinden bu algoritmalar, genellikle tek adımlı (single-step) veya tek adım sonrasını öngören (one-step-ahead) öngörülü algoritmalar olarak adlandırılmaktadır.

Bir adım sonrasını öngören adaptif kontrol algoritmaları, sistem minimum faz olduğu ve sistem zaman gecikmeleri tamamen bilindiği vakit kararlı ve yakınsayan kontrol sağlamaktadır (Goodwin ve Long [92]). Fakat sistem dinamiği ile ilgili detaylı bilgiler mevcut değil ve sistem non-minimum faz ise, bir adım sonrasını öngören kontrol algoritmaları yeterli olmamakta ve tahmin edilen sistem zaman gecikmeleri, model yapısı ve non-minimum faz etkilere karşı duyarlı bir hale gelmektedir.

Örneğin, bölüm 2.2.2'de sözü edilen minimum varyans kontrol algoritması [51], non-minimum faz sistemler için uygun olmamakta, hatta minimum faz sistemler için bile aşırı kontrol eforu gerektirmektedir. Kontrol büyüklüğüne bir ağırlığın ilave edildiği genelleştirilmiş minimum-varyans kontrol algoritması [55,56] ise, non-minimum faz sistemler ile uğraşabilmekte ve aşırı kontrol eforları önlenebilmektedir. Fakat bu metod yine de tek adım sonrasını öngören bir metod olarak kalmakta ve sadece $(t+d)$ anındaki çıkışın kuadratik bir fonksiyonunu minimize etmektedir. Bu metod değişen veya bilinmeyen ölü zaman durumunda güvenilirlikten uzaklaşmaktadır.

Minimum-varyans kontrolörlerin tek adım sonrasını öngörmesi genişletilerek, bir öngörüler dizisi kullanan uzun menzilli öngörülü kontrol stratejileri elde edilmiştir (Clarke ve Zhang [35], De Keyser [36], Garcia [93]). Kullanılan temel fikir, proses çıkışının uzun menzilli (long-range) olarak öngörülmesine dayanan kontrol büyüklüğünü elde etmektir. Buradaki "uzun menzil" ifadesi, öngörü ufuğunun o anki kontrol sinyali için proses cevabının önemli bir kısmını kapsayacak kadar büyütülmesi anlamını taşımaktadır.

Uzun menzilli öngörülü kontrol algoritmaları prensip olarak şu şekilde özetlenebilir (Şekil 3.1).

- Her bir $\langle t \rangle$ anında proses çıkışı $y(t+k)$, uzun menzilli bir ufuk $(k=1, \dots, N_2)$ üzerinden öngörülür. Öngörülen değerler $y(t+k/t)$ şeklinde gösterilir. Öngörü işlemi, proses dinamiği için seçilen bir modele göre yapılır. Burada N_2 , öngörü ufuğu olarak adlandırılmaktadır. Öngörü değerleri, $k=0, \dots, N_2-1$ olacak şekilde o an içinde bulunulan $\langle t \rangle$ anından başlayarak gelecekteki kontrol senaryosuna $u(t+k/t)$ bağlı olmaktadır. $k>0$ için $u(t+k/t) \neq u(t+k)$ 'dir. Burada $u(t+k/t)$, $\langle t \rangle$ anında öngörülen şekli ile gelecekteki kontrol büyüklüğünü göstermektedir. $u(t+k)$ ise gelecekteki kontrol büyüklüğünün gerçek değeridir ve $\langle t \rangle$ anında bilinmemektedir.

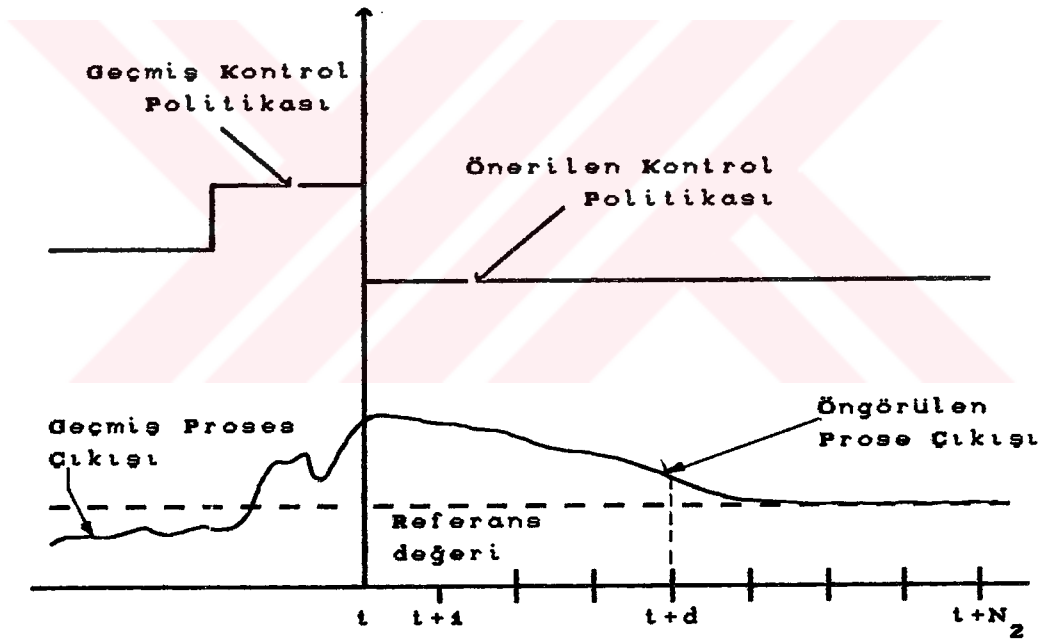
- Proses çıkışının istenen değişimini gösteren bir referans yörüngesi, $y_r(t+k/t)$, $k=1, \dots, N_2$, belirlenir.

- Gelecekteki kontrol vektörü $[u(t+k/t), k = 0, \dots, N_2-1]$, öngörü hatalarına $[y_r(t+k/t) - y(t+k/t)], k=1, \dots, N_2]$ bağlı bir maliyet fonksiyonu minimize edilerek hesaplanır.

- Elde edilen optimal kontrol vektörünün sadece ilk elemanı $u(t/t)$, gerçek sisteme uygulanır. Diğer elemanlar sisteme uygulanmaz, bir sonraki örnekleme anında kullanılmak üzere tüm dizi elemanları geriye kaydırılır, yeni bir çıkış gözlemi $y(t+1)$ elde edilir ve tüm işlemler tekrarlanır. Böylece, genel olarak önceden hesaplanan $u(t+1/t)$ 'den farklı olan yeni bir kontrol vektörü $u(t+1/t+1)$ elde edilir. Bu prensip "gerileyen ufuk stratejisi" (receding horizon strategy) olarak adlandırılmaktadır.

Uzun menzilli öngörü kontrol algoritmalarının bazı dezavantajları da vardır. Hesapsal açıdan biraz karmaşıktırlar. Gerçi bu durum, bugünkü mevcut hesaplama gücü ile özellikle sistemlerin yavaş olduğu proses kontrol uygulamalarında hiç bir problem yaratmamaktadır. Ayrıca paralel hesaplama için oldukça

uygun olduklarından mikroprosesörler yardımı ile mekanik sistemler, servo-kontrol ve robotik uygulamaları için kullanışlı olmaktadır. En büyük dezavantajları uygun bir proses modeline gereksinme duymalarıdır. Gerçi, ne tür bir algoritma kullanılırsa kullanılsın biraz modelleme veya sistem tanınması gerekli olacak ve bunun için de bir ödün vermek gerekecektir. Algoritmasının daha basit olması ve klasik kontrol metodlarının bile (örneğin basit PID algoritması) karmaşık bir proses kontrol edebilmek için bir proses modeline gereksinme göstermeleri, bu dezavantajı büyük ölçüde azaltmaktadır.



Sekil 3.1 Öngörülü Kontrol Algoritmaları Prensipleri

3.2. Yapılan çalışmalar

Çok aşamalı (multi-stage) bir maliyet fonksiyonunun minimizasyonuna dayanan bir metod ilk olarak Peterka & Aström [70] tarafından incelenmiştir. Bu metod, minimum

varyans kavramı muhafaza edilirken non-minimum faz sıfırları ile ilgili problemi ortadan kaldırmaktadır.

Durum uzayına dayanan LQG metodunda ,kestirilecek modelin eşdeğer durum uzayı gösterimine çevrilmesi, aşırı parametrelendirmeye ve hesapsal yüke sebep olmaktadır. Bu yükü hafifletmek için Lam [94], LQG yaklaşımını genişletmiş ve ortak Riccati denklemini çevrim başına sadece bir kez kullanan yeni bir metod geliştirmiştir. Clarke ve arkadaşları da [95], Lam' ın yaklaşımını, genelleştirilmiş minimum-varyans kontrolöre benzer mühendislik dizayn özellikleri ilave ederek tekrar incelemişlerdir. Clarke' nin elde ettiği algoritma N adımlı ufuk ve CARIMA modeli (Controlled-Autoregressive-Integrated-Moving-Average) kullanılmaktadır. Kullanılan CARIMA modeli, elde edilen kontrol kanununda integral etkiyi de doğal olarak içermektedir. Bu algoritma, değişken sistem zaman gecikmesi, non-minimum faz ve açık çevrimli kararsız sistemler için de etkili olabilmektedir.

Polinom domeninde uzun menzilli öngörü fikri, ilk olarak Richalet ve arkadaşları [96] tarafından IDCOM (Identification and Command) algoritması ile incelenmiştir. IDCOM algoritmasında ağırlıklı bir dizi modeli ve kontrol büyüklüklerine bir ağırlık uygulamaksızın ofsetlerden kaçınmak için "ad hoc" metodu kullanılmıştır. Model olarak impuls cevabı modeli seçilmiştir.

Cutler & Ramaker [97]' in DMC (Dynamic Matrix Control) metodunda, sınırlandırılmış bir kontrol ufuğu fikri ortaya atılmış ve bu kontrol ufuğu dışında kontrol artımlarının sıfır olduğu kabul edilmiştir. Model olarak basamak cevabı modeli kullanılmıştır. Bu metod ile non-minimum faz sistemler kontrol edilebilmekte, fakat, basamak cevabı modeline dayanan kötü parametrelendirmeden ötürü açık çevrimli kararsız sistemler kontrol edilememektedir.

Richalet [96,98] tarafından geliştirilen MAC (Model Algorithmic Control) metodunda lineer, zamanla değişmeyen bir sistemin ayırık zaman anlarındaki $y(t)$ çıkışı, ayırık impuls cevabı yardımı ile ifade edilmiştir.

De Keyser' in EPSAC (Extended Prediction Self-Adaptive Control) [99] metodu gelecekteki kontrol büyüklüklerine bir kısıtlama uygulamakta ve uzun menzilli olarak öngörülen çıkışların sadece $u(t/t)$ ' ye bağlı olarak hesaplanmasını sağlamaktadır.

Ydstie' nin EHAC (Extended Horizon Adaptive Control) [100] metodunda, proses bir giriş/çıkış modeli ile tanımlanmakta ve verilen bir maliyet fonksiyonu minimize edilerek kontrol büyüklükleri hesaplanmaktadır.

Clarke tarafından geliştirilen ve yukarıdaki uzun menzilli öngörülü kontrol algoritmalarının bir sentezi niteliğini taşıyan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control - GPC) algoritması [37, 38,39] bir gerileyen ufuklar metodudur. Robot kontrolü v.s gibi referans yörüngenin önceden programlandığı durumlar için çok kullanışlıdır.

3.3. Değişik Uzun Menzilli Öngörülü Kontrol Algoritmaları

Bu bölümde, bugüne kadar geliştirilen uzun menzilli öngörülü kontrol algoritmaları için temel fikirler izah edilecek, aşırı detaylara girilmeyecektir.

3.3.1 Model Algoritmik Kontrol (MAC)

Lineer ve zamanla değişmeyen bir sistemin ayırık zaman anlarındaki $y(t)$ çıkışı, ayırık impuls cevabı $h(j)$

yardımları ile aşağıdaki şekilde tarif edilebilir.

$$y(t) = \sum_{j=-\infty}^{\infty} h(j) u(t-j) \quad \dots (3.1)$$

Burada;

$y(t)$: t anındaki sistem çıkışı,

$u(t)$: t anındaki sistem girişi,

$h(j)$: impuls cevabının katsayıları' nı ifade etmektedir.

MAC algoritması, sistem impuls cevabı $\langle h(j) \rangle$ 'nin sonlu N adet terimden meydana gelen bir yaklaşımını kullanmaktadır. Bir iç (internal) model kullanılarak, k adım sonrasını öngören $y_{im}(t+k)$ öngörü değeri,

$$y_{im}(t+k) = \sum_{j=1}^N h(j) u(t+k-j) \quad \dots (3.2)$$

veya,

$$y_{im}(t+k) = h^T \cdot u(t+k) = u^T(t+k) \cdot h \quad \dots (3.3)$$

şeklinde yazılabilir. Burada,

$$h^T = [h(1), h(2), \dots, h(N)],$$

$$u^T(t+k) = [u(t+k-1), u(t+k-2), \dots, u(t+k-N)]$$

şeklindedir. "im" iç modeli simgelemektedir. Her bir $\langle t \rangle$ örnekleme anında aşağıdaki işlemler tekrarlanmaktadır.

İlk olarak bir model referans yörüngesi $y_{mr}(t)$, öngörü ufuğu " l " boyunca hesaplanır. Birinci mertebeden " r " zaman sabitli bir modelin, gerçek $y(t)$ çıkışından başlayarak y_r referansına doğru aşağıdaki şekilde bir değişim göstereceği önerilmiştir.

$$y_{mr}(t) = y(t)$$

$$y_{mr}(t+1) = \alpha y_{mr}(t+1-1) + (1-\alpha) y_r$$

$$i=1,2,\dots,l \quad \dots (3.4)$$

Burada;

$$\alpha = e^{-T/T_s}$$

T_s = örnekleme periyodu,

"mr" model referansı simgelemektedir.

Daha sonra, öngörü ufuğu boyunca gelecekteki giriş büyüklüğünde bir değişme olmadığı kabul edilerek çıkışın öngörülen değerleri hesaplanmaktadır. Buna göre,

$$\Delta u(t-1+i) = u(t-1+i) - u(t-2+i) = 0, \quad i=1,2,\dots,l$$

$$y_{im}(t) = h^T u(t)$$

$$y_{im} = \begin{bmatrix} y_{im}(t+1) \\ y_{im}(t+2) \\ . \\ . \\ y_{im}(t+l) \end{bmatrix} = \begin{bmatrix} u(t) & u(t-1) & \dots & u(t-N+1) \\ u(t+1) & u(t) & \dots & u(t-N+2) \\ . & . & . & . \\ . & . & . & . \\ u(t+l-1) & u(t+l-2) & . & u(t-N+1) \end{bmatrix} \cdot$$

$$\begin{bmatrix} h(1) \\ h(2) \\ . \\ . \\ h(N) \end{bmatrix}$$

..... (3.5)

$i=1,2,\dots,l$ için $\Delta u(t-1+i) = 0$ kabul edildiğinden yukarıdaki eşitlikte

$$u(t-1) = u(t) = u(t+1) = \dots = u(t+l-1)$$

yazılabilir. Ayrıca bilinmeyen yük bozucuları ile uğraşabilmek için öngörülen değerler,

$$y_{im}^*(t+1) = y_{im}(t+1) + (y(t) - y_{im}(t))$$

$$i = 1, 2, \dots, l \quad \dots (3.6)$$

şeklinde sistemin gerçek $y(t)$ çıkışına göre düzeltilir. Model referans yörüngesi $y_{mr}(t+1)$ ve ayarlanmış öngörü (adjusted prediction) $y_{im}^*(t+1)$ arasındaki fark alınarak bir hata vektörü "e" oluşturulur.

$$e = \begin{bmatrix} e(t+1) \\ e(t+2) \\ \vdots \\ e(t+l) \end{bmatrix} = \begin{bmatrix} y_{mr}(t+1) \\ y_{mr}(t+2) \\ \vdots \\ y_{mr}(t+l) \end{bmatrix} - \begin{bmatrix} y_{im}^*(t+1) \\ y_{im}^*(t+2) \\ \vdots \\ y_{im}^*(t+l) \end{bmatrix}$$

$$= y_{mr} - y_{im}^* \quad \dots (3.7)$$

Sayet, öngörü ufuğu boyunca giriş değişkenindeki değişimler göz önüne alınırsa,

$$u(t-1+i) = u(t-2+i) + \Delta u(t-1+i) \quad i = 1, 2, \dots, l$$

$$e = y_{mr} - y_{im}^* - H\Delta u \quad \dots (3.8)$$

şeklinde hata vektörü elde edilir. Burada;

$$\Delta u^T = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+l-1)],$$

$$H = \begin{bmatrix} h(1) & 0 & \dots & 0 \\ h(2) & h(1) & . & . \\ . & . & . & . \\ . & . & . & . \\ h(l) & h(l-1) & \dots & h(1) \end{bmatrix}$$

şeklindedir.

e' yi minimize eden, artımlar cinsinden düzenlenmiş kontrol büyüklüklerinin optimal bir dizisi, aşağıdaki kuadratik maliyet fonksiyonunun minimizasyonundan elde edilebilir.

$$J = e^T e + \beta^2 \Delta u^T \Delta u \quad \dots (3.9)$$

β : Artımlar cinsinden kontrol büyüklüğü için ceza faktörüdür.

En küçük kareler yöntemi ile Δu için aşağıdaki çözüm elde edilir:

$$\Delta u = (H^T H + \beta^2 I)^{-1} H^T (y_{mr} - y_{im}^*) \quad \dots (3.10)$$

Sisteme sadece ilk kontrol büyüklüğü (giriş değişkeni) $u(t) = u(t-1) + \Delta u(t)$ uygulanır.

Sayet, ardışık bir impuls cevabı tanılama prosedürü ilave edilirse, MAC algoritmasının adaptif bir versiyonu elde edilmiş olur. Bu durumda, her bir örnekleme anında impuls cevabı katsayılarının çevrim-içi kestirilmesi kontrol büyüklüğü hesabından önce gerçekleştirilir.

3.3.2.Dinamik Matris Kontrolü (DMC) Metodu

DMC algoritması, proses çıkışı $y(t)$ ' yi ayrik zaman anlarında ayrik basamak cevabı $a(j)$ yardımı ile modellemektedir.

DMC algoritmasının "l" öngörü ufuğu, göz önüne alınan basamak cevabı katsayılarının sayısı N' e eşittir. DMC tekniği, giriş değişkenindeki ardışık N_u adet değişmeyi hesaba katmaktadır ($N_u \leq N$). Burada " N_u " kontrol ufuğu olarak adlandırılmaktadır. Bu durumda, kontrol ufuğu boyunca giriş değişkenindeki ardışık değişmelerden dolayı, öngörü ufuğu boyunca model çıkışındaki değişmeler aşağıdaki şekilde tanımlanabilmektedir:

$$\begin{aligned}
 \Delta y(t+1) &= y_{im}(t+1) - y(t) \\
 &= a(1) [u(t) - u(t-1)] = a(1) \Delta u(t) \\
 \Delta y(t+2) &= y_{im}(t+2) - y(t) \\
 &= a(2) \Delta u(t) + a(1) \Delta u(t+1) \\
 &\dots \qquad \dots \qquad \dots \\
 &\dots \qquad \dots \qquad \dots \qquad \dots (3.11)
 \end{aligned}$$

Burada $y_{im}(t+k)$, $u(t+1)$ girişi uygulandığında, $0 \leq i < k$ ve $k = 1, 2, \dots, N$ şartları altında $(t+k)$ anındaki çıkışın öngörülen değeri, $\Delta u(t-1+i) : i=1, 2, \dots, N_u$ için $(t-1+i)$ anındaki giriş değişkenindeki değişme miktarı, $a(j)$ ise basamak cevabının katsayılarıdır.

(3.11) ifadesi vektörel formda aşağıdaki şekilde gösterilebilir.

$$\Delta y = A. \Delta u \quad \dots (3.12)$$

Burada;

$$\Delta y^T = [\Delta y(t+1), \Delta y(t+2), \dots, \Delta y(t+N)]$$

$$\Delta u^T = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]$$

$$A = \begin{bmatrix} a(1) & 0 & & 0 \\ a(2) & a(1) & & \\ \vdots & \vdots & & \\ a(N_u) & a(N_u-1) & \dots & a(1) \\ \vdots & \vdots & & \vdots \\ a(N) & a(N-1) & \dots & a(N-N_u+1) \end{bmatrix}$$

DMC algoritması, her bir örnekleme anında aşağıdaki yolu izlemektedir.

Öngörü ufuğu boyunca model çıkışının öngörülen değerleri, önceden hesaplanan ve uygulanan giriş değişkenindeki değişimlerin hepsi göz önüne alınarak hesaplanır. Öngörü değerleri, ölçülen gerçek $y(t)$ çıkış değerleri kullanılarak düzeltilir. Bu işlem, bilinmeyen yük bozucularına karşı etkili olabilmek için yapılmaktadır.

$$y_{im}^*(t+k) = y_{im}(t+k) + (y(t) - y_{im}(t))$$

$$k = 1, 2, \dots, N \quad \dots (3.13)$$

Öngörü ufuğu boyunca, proses çıkışının düzeltilmiş öngörütleri ve referans büyüklüğü " y_r " arasındaki fark alınarak "e" hata vektörü oluşturulmaktadır.

$$e = \begin{bmatrix} y_r(t+1) \\ y_r(t+2) \\ \vdots \\ y_r(t+N) \end{bmatrix} - \begin{bmatrix} y_{im}^*(t+1) \\ y_{im}^*(t+2) \\ \vdots \\ y_{im}^*(t+N) \end{bmatrix} = y_r - y_{im}^* \quad \dots (3.14)$$

Kontrol ufluğu boyunca, ardışık olarak yeni giriş büyüklüğü değişimleri de hesaba katılarak, hata vektörü öngörü ufluğu üzerinden minimize edilebilir.

$J = e^T e$ şeklinde kuadratik bir maliyet fonksiyonu ve $e = y_r - y_{im}^* - A \Delta u$ ifadesi kullanılarak problemin en küçük kareler formülasyonu,

$$\Delta u = (A^T A)^{-1} \cdot A^T y_r - y_{im}^* \quad \dots (3.15)$$

şeklinde elde edilir. Δu büyüklüğünü düzgünleştirmek için $(A^T A)^{-1}$ kare matrisine β ceza faktörünü katmak mümkündür, bu durumda matrisin diagonal elemanları β ile çarpılır. β faktörü bu algoritmanın ayar parametresidir. Prosese Δu dizisinin sadece ilk elemanı uygulanmaktadır.

Orjinal DMC algoritmasına, sistemin impuls cevabı modelini kestiren ardışık bir tanılama metodu ilave edilerek adaptif özellikler kazandırılabilir. İmpuls cevabı modeli ile basamak cevabı modeli arasındaki aşağıdaki bağıntı gereğince ,

$$\Delta A(q^{-1}) = H(q^{-1})$$

sistemin basamak cevabı modeli, impuls cevabı modelinden ardışık olarak basitçe elde edilebilir.

3.3.3.Epsac Metodu

EPSAC (Extended Prediction Self-Adaptive Control) metodunda, proses aşağıdaki şekilde modellenmektedir.

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-d) + V(t) \quad \dots (3.16)$$

Burada;

t : zaman değişkeni, d : zaman gecikmesi ($d > 0$), $y(\cdot)$ ve $u(\cdot)$ proses çıkış ve girişi, $V(\cdot)$ ise bozucu sinyalidir. $V(\cdot)$, ölçülemeyen bozucuların (hem stokastik gürültü, hem de basamak şeklindeki yük bozucuları) ve lineerleştirme ofsetlerinin etkilerini içermektedir.

Bir adım sonrasını öngören (one-step-ahead) çıkış büyüklüğü, aşağıdaki öngörü modeli yardımı ile hesaplanmaktadır.

$$\begin{aligned} C(q^{-1}) y^*(t+1/t) &= C(q^{-1}) y(t) + q [C(q^{-1}) - A(q^{-1})] \\ &\cdot \Delta y(t) + B(q^{-1}) \Delta u(t+1-d) \dots (3.17) \end{aligned}$$

A , B ve C polinomlarının mertebeleri sırası ile n_a , n_b ve n_c ' dir. Δ , daha önceden de bilindiği gibi fark alma operatörüdür ($\Delta y(t) = y(t) - y(t-1)$).

Parametre vektörü;

$$\begin{aligned} \theta^T &= (c_1, \dots, c_{n_c}; -a_1, \dots, -a_{n_a}; b_0, \dots, b_{n_b}), \\ \Delta y(t) &= \phi^T(t) \cdot \hat{\theta}(t) + \eta(t) \quad \dots (3.18) \end{aligned}$$

kestirme modeli yardımı ile en küçük kareler metodu kullanılarak kestirilebilir. Veri vektörü $\phi(t)$,

$$\begin{aligned} \phi^T(t) &= (e(t-1), \dots, e(t-n_c); \Delta y(t-1), \dots, \Delta y(t-n_a) \\ &\quad ; \Delta u(t-d), \dots, \Delta u(t-d-n_b)) \end{aligned}$$

şeklindedir. Burada $\varepsilon(t) = y(t) - y^*(t/t-1)$, bir adım sonrasını öngörme hatasıdır. (3.17) nolu ifade;

$$\Delta y^*(t+1/t) = \phi^T(t+1) \cdot \hat{\theta}(t) \quad \dots (3.19)$$

şeklinde tekrar yazılabilir. Burada;

$$\begin{aligned} \Delta y^*(t+1/t) &= y^*(t+1/t) - y^*(t/t) \\ &= y^*(t+1/t) - y(t) \end{aligned}$$

şeklinde ifade edilmektedir. $C(q^{-1})$ polinomu kestirilebilir veya bir dizayn polinomu olarak sabit olabilir. (En çok kullanılan özel bir durum $C(q^{-1}) = 1$ olmasıdır).

$k = 2, 3, \dots, N_2$ için (N_2 öngörü ufuğudur) proses çıkışı çok adımlı bir öngörücü yardımı ile öngörülebilir.

$$\begin{aligned} A(q^{-1}) [y^*(t+k/t) - y^*(t+k-1/t)] \\ = B(q^{-1}) \Delta u(t+k-d) \quad \dots (3.20) \end{aligned}$$

Özel q^{-1} operatörü, $y^*(t+k/t)$ 'deki argumanlardan sadece ilkinde etki etmektedir, yani;

$$q^{-1} y^*(t+k/t) = y^*(t+k-1/t)$$

Proses çıkışı $y^*(t+k/t)$, $k = 1, \dots, d, \dots, N_2$ boyunca öngörüldükten sonra yapılacak işlem, $u(t)$ 'yi hesaplamaktır. Bu aşamada, öngörülen $[y^*(t+k/t), k = d, \dots, N_2]$ değerlerinin gelecekteki kontrol politikasına $[u(t/t), \dots, u(t+N_2-d/t)]$ bağlı olduğuna dikkat etmek çok önemlidir. Çünkü; o anki (t) anında önerilmektedirler. EPSAC algoritmasında önerilen kontrol senaryosu oldukça basittir, $k > 0$ için $\Delta u(t+k/t) = 0$ olmakta veya $u(\cdot/t)$, o andan itibaren sabit kalmaktadır (Şekil 3.1). Bu strateji, sadece içinde bulunulan anda önerilen kontrol büyüklüklerine bir kısıtlama getirmektedir. Gerçek olarak elde

edilen kontrol büyüklükleri $u(t)$, $u(t+1)$, ... ise gelecek zaman dilimlerinde sabit kalmayacaktır (Gerileyen ufuk stratejisi).

Bu basit stratejinin bir sonucu olarak uzun menzilli öngörü değerleri $[y^*(t+k/t), t = 1, \dots, N_2]$, sadece belirlenen bir maliyet fonksiyonunu minimize edecek şekilde aşağıda hesaplanacak olan $u(t/t)$ ' ye bağlıdır. Kullanılan maliyet fonksiyonlarına bir örnek olarak aşağıdaki fonksiyon gösterilebilir.

$$J[u(t/t)] = \sum_{k=d}^{N_2} \gamma(k) [y_r(t+k) - P(\tilde{q}^{-1}) y^*(t+k/t)]^2 \quad \dots (3.21)$$

Burada,

$P(q^{-1}) = p_0 + p_1 q^{-1} + \dots + p_n q^{-n}$ ve $P(1) = 1$ olan bir dizayn polinomudur, y_r referans P büyüklüğüdür. $\gamma(k)$ ise ağırlık faktörüdür.

$$\gamma(k) = \begin{cases} 1 & k = d \text{ için} \\ 0 & k > d \text{ için} \end{cases}$$

şeklinde yapılan bir seçim ile, özel bir durum olarak $P(.)=1$ olduğu zaman minimum-varyans kontrolör, $P(.) \neq 1$ olduğu zaman ise model izleme kontrolörü elde edilir. Genellikle öngörü ufuğu N_2 sabit tutulur, fakat $\gamma(.)$ için uygun bir fonksiyon seçilerek etkili bir ufuk bulunabilir. Basit bir seçim eksponansiyel ağırlık faktörüdür ($\gamma(k) = \lambda^{N_2-k}$, λ dizayn parametresidir). (3.21) ifadesini minimize eden kontrol kanunu, kapalı formda aşağıdaki şekilde yazılabilir:

$$u(t/t) = \frac{\sum_{k=d}^{N_2} \gamma(k) \alpha_k [y_r(t+k) - P(\tilde{q}^{-1}) y_0^*(t+k/t)]}{\sum_{k=d}^{N_2} \gamma(k) \alpha_k^2} \quad \dots (3.22)$$

α_k ; $q^{-d} \hat{B}(q^{-1})/\hat{A}(q^{-1})$ sisteminin ayırık zaman impuls cevabı katsayılarına karşılık gelmektedir. $y_o^*(t+k/t)$ ise $\Delta u(t/t) = \Delta u(t+1/t) = \dots = \Delta u(t+N_2-d/t)=0$ kabul edilerek, (3.20) ifadesinden hesaplanan, k adım sonrasını öngören proses çıkışıdır.

3.3.4. Ehac Metodu

Proses aşağıdaki şekilde bir giriş-çıkış modeli ile tanımlanmaktadır:

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-d) \quad \dots (3.23)$$

EHAC' in temel fikri, her bir örnekleme anında aşağıdaki şartı sağlayacak giriş dizisini $[u(t), u(t+1), \dots, u(t+N_2-d)]$ hesaplamaktır.

$$E \left\{ y(t+N_2) - y_r(t+N_2) \right\} = 0, N_2 \geq d \quad \dots (3.24)$$

Hesaplanan dizinin sadece ilk elemanı prosese uygulanır ve işlemler her bir örnekleme anında tekrarlanır. (3.24)' ün çözümü tek değildir ($N_2 = d$ olmadıkça). Çözüm için değişik yaklaşımlar önerilmektedir (Ydstie [100]). Kullanılan yaklaşımlardan bir tanesi, $(t, t+N_2-d)$ aralığı boyunca kontrolü sabit kabul etmek, yani; $u(t) = u(t+1) = \dots = u(t+N_2-d)$, veya kontrol eforunun minimize edildiği bir strateji seçmek, yani; $u(t), \dots, u(t+N_2-d)$ ' yi (3.24) kısıtlaması altında,

$$J_1 = \sum_{i=0}^{N_2-d} u^2(t+i) \quad \dots (3.25)$$

ifadesini minimize edecek şekilde seçmek ile mümkün olmaktadır.

Yük bozucularından etkilenmemek için (3.25) ifadesi-
nin artımlar şeklindeki modelini kullanmak daha uygun
olmaktadır. Yani,

$$J_2 = \sum_{i=0}^{N_2-d} \Delta u^2(t+i) \quad \dots (3.26)$$

Bu amaca uygun olarak artımsal, N_2 adım sonrasını
öngören bir öngörücü kullanılır.

$$\begin{aligned} y^*(t+N_2/t) &= y(t) + F(q^{-1})\Delta y(t) \\ &+ E(q^{-1}) B(q^{-1})\Delta u(t+N_2-d) \quad \dots (3.27) \end{aligned}$$

$E(q^{-1})$ ve $F(q^{-1})$, aşağıdaki polinomial denklemi sağlaya-
cak şekilde seçilmektedir.

$$\begin{aligned} (1-q^{-N_2}) &= A(q^{-1})E(q^{-1})(1-q^{-1}) \\ &+ q^{-N_2} F(q^{-1})(1-q^{-1}) \quad \dots (3.28) \end{aligned}$$

$E(q^{-1})$ ' in mertebesi N_2-1 ' dir.

(3.26) maliyet fonksiyonunu minimize edecek kontrol
büyüklüğü dizisinin ilk elemanı, aşağıdaki şekilde
hesaplanabilir.

$$u(t) = u(t-1) + \frac{\alpha_0 (y_r(t+N_2) - y_o^*(t+N_2/t))}{\sum_{i=0}^{N_2-d} \alpha_i^2} \quad \dots (3.29)$$

Burada;

$y_o^*(t+N_2/t)$, gelecek zaman dilimlerindeki $\Delta u(t+i)$ ' lerin
 $i \geq 0$ için sıfır olduğu kabul edilerek ve (3.27) nolu
ifade kullanılarak elde edilen çıkışın N_2 adım sonrasının
öngörülen değeridir. α_i , (3.27) ifadesindeki $\Delta u(t+i)$ '
nin katsayısıdır.

Şayet, (3.23) nolu proses modelinin parametreleri biliniyorsa, (3.27)'deki parametreler ve dolayısı ile (3.29) kontrol kanunu hesaplanabilir. Ayrıca, öngörücünün parametreleri çevrim-içi kestirilebilir, böylece kontrol stratejisi adaptif duruma getirilebilir.

EHAC metodunda tek ayar parametresi, öngörü ufuğu N_2 'dir ve kestirilecek parametre sayısı öngörü ufuğuna bağlıdır.

3.3.5.GPC Metodu

Uzun menzilli öngörülü kontrol ailesinin en yeni üyesi, GPC olarak bilinen "Genelleştirilmiş Öngörülü Kontrol" (Generalized Predictive Control) algoritmasıdır [37,38,39]. Genelleştirilmiş öngörülü kontrol algoritması; non-minimum faz sistemler, açık çevrimli kararsız sistemler, robotik manipülatörler, değişken veya bilinmeyen model mertebesine sahip sistemler gibi gerçek sistemlerin büyük bir çoğunluğunun kararlı kontrolü için genel maksatlı bir algoritma oluşturmaktadır.

Genelleştirilmiş öngörülü kontrol algoritması CARIMA formunda bir proses modeli kullanmaktadır. Bu model, sonlu sayıdaki parametre takımından oluşan az sönümlenmiş veya kararsız sistemler için bile doğru çıkış öngörülerini elde etmek için kullanılabilir. GPC, kararlı bir sistem için diğer metodlardaki kontrol kalitesini sağladığı gibi ilgi alanı daha karmaşık dinamik prosesleri kapsayacak şekilde genişletilmiştir [35].

GPC, esas olarak $y(t)$ çıkışının sonlu bir ufuk üzerinden öngörüldüğü ve kontrol sinyalleri dizisinin kuadratik bir maliyet fonksiyonunu minimize edecek şekilde hesaplandığı bir algoritmadır. Uygun cevabı elde etmek için kontrol sinyallerinin beklenti

büyükliklerinde bazı tahminler yapılmaktadır, bu tahminler GPC yaklaşımının püf noktasını teşkil etmektedir.

GPC algoritmasında, ayar parametrelerinin ve ufukların özel şekilde seçilmeleri ile diğer uzun menzilli öngörülü kontrol algoritmaları ve kendi kendini ayarlayan kontrol algoritmalarını elde etmek mümkün olmaktadır [37,38]. Bu da GPC' nin avantajlarından birini teşkil etmektedir.

GPC' de kestirilecek parametre sayısı, parametrik giriş-çıkış modelini kullanmayan diğer metodlara göre (örneğin MAC, DMC, IDCOM,...) daha azdır. Bu da hesapsal yükü azaltmaktadır. GPC aşırı parametrelendirmeye karşı duyarlı değildir. 6 parametre ile elde edilebilen kapalı çevrim davranışı (3A,3B) 3 parametre ile de (örneğin 1A ve 2B) aynı şekilde sağlanmaktadır [35]. Öngörülecek değerleri oluşturmak için daha az parametre ve hesaplama gerektirdiğinden diğer uzun menzilli öngörülü kontrol algoritmalarına göre daha hızlıdır. Yapısında doğal integratör bulunması dolayısı ile kararsız ve titreşimli sistemlerde de çok iyi performans göstermektedir.

Yukarıda bahsedilen avantajları ve değişik kontrol algoritmalarını bir çatı altında toplaması dolayısı ile bu tez çalışmasında GPC algoritması kullanılmış ve değişik genel simülasyon çalışmaları ve robota uyarlanmış çalışmalar yapılmıştır.

Bundan sonraki bölümde, GPC algoritması geniş bir şekilde incelenecek ve değişik uygulama örnekleri verilecektir.

BÖLÜM 4. GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROL

Genelleştirilmiş öngörülü kontrol (Generalized Predictive Control - GPC), tek bir öngörü yerine sonlu bir öngörü ufuğu boyunca oluşturulan öngörü dizilerini kullanan daha genel bir maliyet fonksiyonuna sahip olması ile, kendinden önce yapılan bir adım sonrasını öngören (one-step ahead) çalışmalardan esaslı bir şekilde farklıdır. Öngörülü bir kontrol algoritmasının esası temel olarak; bir öngörücü, bir maliyet fonksiyonu, bu maliyet fonksiyonunu minimize edecek bir kontrol dizisini seçme algoritması ve adaptif durumda parametre tanılama algoritmasının birleşmesinden meydana gelmektedir.

4.1. Tek Giriş-Tek Çıkışlı (SISO) Genelleştirilmiş Öngörülü Kontrol Algoritması

4.1.1. Genel Bilgiler

Kontrol edilmesi istenen bir prosesin, ölçülen geçmiş zaman dilimlerindeki çıkışları ve kontrol girişleri ile birlikte dinamik bir modeli verilirse, gelecekteki zaman dilimleri için sistem çıkışlarına öngörüler yapılabilir. Gelecekteki çıkışların gelecekteki kontrol girişlerine bağlı olacağı açıkça bellidir. Bu gözönüne alınır ve lineerlik özelliği kabul edilirse,

$$\hat{y} = \{ y(t+1), y(t+2), \dots, y(t+j) \}^T$$

Öngörülerini iki ayrı öngörü dizisinin toplamı şeklinde gösterilebilir.

$$\hat{y} = \hat{y}_f + \hat{y}_d \quad \dots (4.1)$$

Bu dizilerden "serbest cevap" olarak adlandırılan ilki $\hat{y}_f$, gelecekteki kontrol büyüklüklerinden etkilenmeyen öngörülerini, yani gelecekte uygulanacak kontrol büyüklüklerinin göz önüne alınmayıp sadece o an içinde bulunan durum göz önüne alınarak yapılan öngörülerini göstermektedir. "Zorlanmış cevap" olarak adlandırılan diğer $\hat{y}_d$ dizisi ise, sadece gelecekte uygulanacak kontrol büyüklüklerinden ötürü sıfır durumundan itibaren öngörülen cevabı göstermektedir.

$\hat{y}_f$ değiştirilemeyeceğinden kontrol problemi, elde edilecek net öngörü $\hat{y}$, referans değerine yakınsayacak şekilde gelecekteki kontrol büyüklüklerinin, dolayısı ile de $\hat{y}_d$ 'nin seçimi olmaktadır.

Gelecek zaman dilimindeki kontrol büyüklükleri aşağıdaki formda bir maliyet fonksiyonunu minimize edecek şekilde seçilirler.

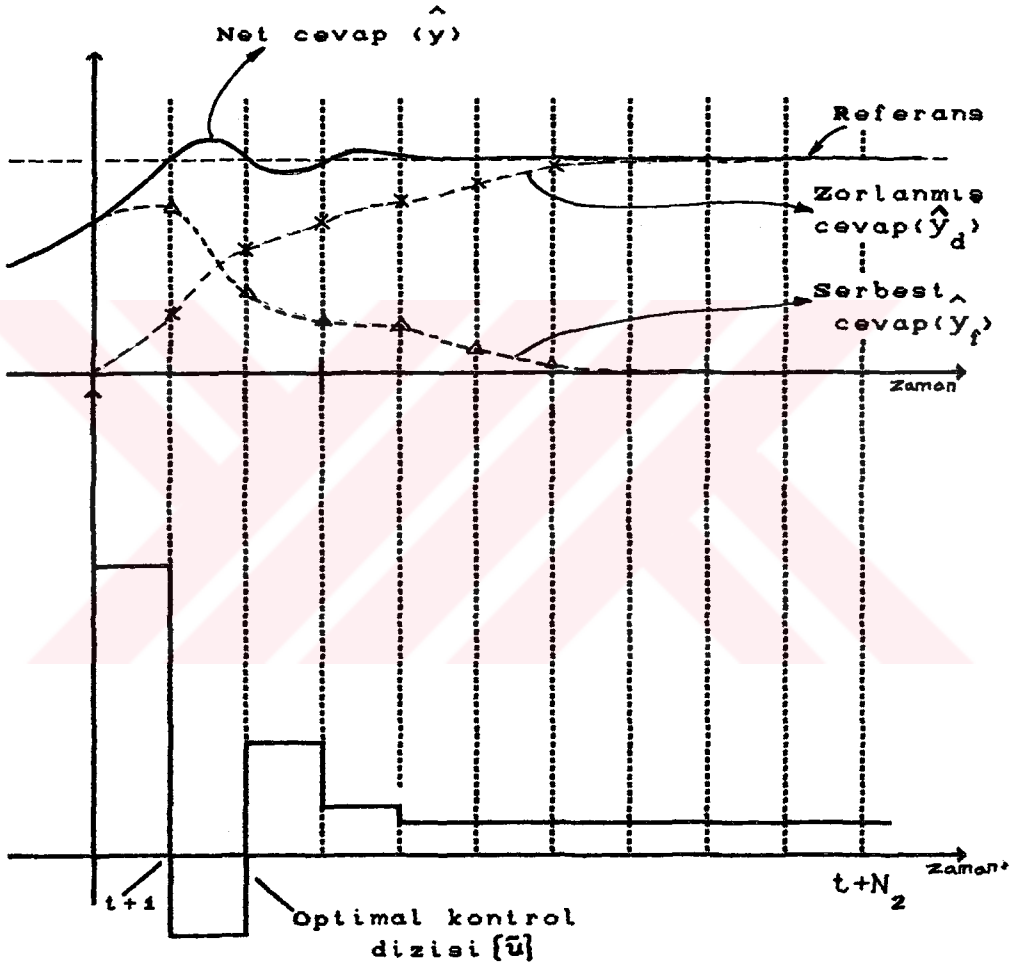
$$J = (\hat{y} - y_r)^T (\hat{y} - y_r) \quad \dots (4.2)$$

Bu, karesel ortalama hassasiyetinde, seçilen öngörülerin referans değerler dizisi y_r 'ye yakınsamasını sağlamaktadır. Yukarıda anlatılanlar şematik olarak şekil 4.1' de gösterilmiştir.

Şekil 4.1' de alt kısımdaki diyagram, "serbest cevap" in referans değerinden olan sapmalarını düzeltmek için seçilen kontrol dizisi $[\tilde{u}]$ 'yu göstermektedir. Kontrol dizisi, (4.2)'yi minimize edecek şekilde

seçildiğinden, elde edilen hafif salınım yapan öngörülen cevap mümkün olan en iyi cevap durumundadır.

Kontrol dizisi elemanlarından sadece ilki kullanılır, minimizasyon ve kontrol hesaplamaları her bir örnekleme anında tekrar edilir. Bu durum GPC'yi gerileyen ufuklu kontrol stratejisi içine dahil etmektedir (Kwon & Pearson [101]).



Şekil 4.1. Öngörülen serbest ve zorlanmış cevaplar

4.1.2. Öngörü Kavramı

Öngörülü bir kontrolör, o an içinde bulunulan (t) anına kadar gözlenen bilgilerden ve gelecekteki kontrol

büyüklikleri hakkında yapılmış tahminlerden oluşan $y(t+j)$ öngörüsünü kullanmaktadır. Clarke ve Gawthrop'un [55] öngörü şeması "Diophantine" denklemi olarak adlandırılan bir tanım aracılığı ile, bu öngörülerini elde etmek için kullanılan en genel metoddur.

Öngörü şemasını çıkarabilmek için sistem modeli olarak CARIMA modeli kullanılacaktır.

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-1) + C(q^{-1}) \xi(t) / \Delta \dots (4.3)$$

Basitlik olması açısından $C(q^{-1}) = 1$ olarak seçilmiştir. Δ fark alma operatörüdür ve $1-q^{-1}$ e eşittir.

(4.3) ifadesi temel alınarak $y(t+j)$ 'nin "j" adım sonraki öngörüsünü elde etmek için "Diophantine" denklemi olarak adlandırılan aşağıdaki bağıntıdan yararlanılmaktadır.

$$1 = E_j(q^{-1}) A \Delta + q^{-j} F_j(q^{-1}) \dots (4.4)$$

E_j ve F_j , $A(q^{-1})$ ve öngörü aralığı j verildiğinde tek olarak tanımlanabilen polinomlardır. (4.3) ifadesi $E_j \Delta q^j$ ile çarpılırsa,

$$E_j A \Delta y(t+j) = E_j B \Delta u(t+j-1) + E_j \xi(t+j)$$

elde edilir. (4.4) ifadesinden $E_j A \Delta = 1 - q^{-j} F_j$ bağıntısı çekilir, yukarıda yerine konur ve polinomlardaki q^{-1} argümanı şimdilik göz önüne alınmayacak olursa,

$$(1 - q^{-j} F_j) y(t+j) = E_j B \Delta u(t+j-1) + E_j \xi(t+j)$$

$$y(t+j) - F_j y(t) = E_j B \Delta u(t+j-1) + E_j \xi(t+j)$$

$$y(t+j) = E_j B \Delta u(t+j-1) + F_j y(t) + E_j \xi(t+j)$$

$$\dots (4.5)$$

ifadesi elde edilir.

Buradaki E_j ve F_j polinomlarının mertebesi, sırası ile $j-1$ ve n_a-1 ' dir. (n_a , A polinomunun mertebesidir). E_j ' nin mertebesi $j-1$ olduğundan gürültü bileşenlerinin hepsi gelecekteki zaman dilimi içinde kalmaktadır. Örneğin, içinde bulunduğumuz anı " t " ile gösterirsek ve j ' nin de 3 olduğunu kabul edersek gürültü bileşenleri

$$(E_1 + E_2(q^{-1})) \xi(t+3) \text{ ve}$$

$$E_1 \xi(t+3) + E_2 \xi(t+2)$$

şeklinde $(t+2)$ ve $(t+3)$ zaman diliminde bulunacaktır.

(4.5) nolu ifadede

$$e(t+j/t) = E_j \xi(t+j) \quad \dots (4.6)$$

öngörü hatası olarak kabul edilerek optimal $\hat{y}(t+j/t)$ öngörü değerleri

$$\hat{y}(t+j/t) = E_j B \Delta u(t+j-1) + F_j y(t) \quad \dots (4.7)$$

şeklinde elde edilir.

$$E_j(q^{-1}) B(q^{-1}) = G_j(q^{-1}) \text{ şeklinde tanımlanarak}$$

$$\hat{y}(t+j/t) = G_j \Delta u(t+j-1) + F_j y(t) \quad \dots (4.8)$$

elde edilir.

(4.4)' den elde edilecek

$$E_j(q^{-1}) = [1 - q^{-j} F_j(q^{-1})] / A(q^{-1}) \cdot \Delta$$

ifadesi $G_j(q^{-1}) = E_j(q^{-1}) B(q^{-1})$ 'de yerine konulacak olursa

$$G_j(q^{-1}) = [1 - q^{-j} F_j(q^{-1})] B(q^{-1})/A(q^{-1}).\Delta$$

elde edilir. Buradan da G_j' yi hesaplamamanın bir yolunun sistemin basamak cevabının "z" dönüşümünü yapmak ve ilk j terimini göz önüne almak şeklinde olduğu anlaşılmaktadır (Clarke ve Zhang [35]).

GMV self-tuning kontrolörün geliştirilmesi sırasında sadece bir öngörü $\hat{y}(t+d/t)$ kullanılmıştır, burada "d" bilindiği gibi sistem ölü zamanının kabul edilen değeridir. Çıkarmakta olduğumuz GPC'de ise "minimum ve maksimum öngörü ufuğu" olarak adlandırılan minimum bir değerden maksimum bir değere kadar olan "j" adımlık öngörü dizisinin tamamı kullanılmaktadır. $j < d$ için öngörülen $\hat{y}(t+j/t)$ değerleri tamamen bilinen bilgilere bağlıdır, fakat $j \geq d$ için gelecekteki kontrol büyüklükleri ile ilgili olarak yapılan kabullere gereksinme olmaktadır. Bu kabuller GPC yaklaşımının en önemli özelliğini teşkil etmektedir.

4.1.3. Diophantine Denkleminin Ardışık Hale Getirilmesi

Uzun menzilli öngörü elde etmek için değişik yöntemler vardır. Bu yöntemlerden bir tanesi, her bir "j" ufuğu için self-tuning öngörülerin değerlerini bir hafızada toplamaktır. Bu yöntem, De Keyser ve Van Cauwenberghe'nin [99]' ve MUSMAR metodunun (Mosca ve arkadaşları [102]) yaklaşımıdır. Diğer bir alternatif, (4.4) ifadesinin göz önüne alınacak bütün "j" 'ler için E_j ve F_j elde edilecek şekilde çözülmesidir. Fakat bu her iki metod da hesapsal açıdan masraflı olmaktadır. Bunların yerine daha basit ve daha etkili bir algoritma olarak, Diophantine denkleminin ardışık bir hale getirilmesi çalışmaları yapılmıştır. Bu durumda E_{j+1} ve

F_{j+1} polinomları, E_j ve F_j değerlerinden elde edilir.

Basitlik için $E = E_j$, $R = E_{j+1}$, $F = F_j$, $S = F_{j+1}$ notasyonu kabul edilir ve $\Delta\Delta$ yerine kısaltma maksadı ile $\tilde{A}$ kullanılarak iki Diophantine denklemi göz önüne alınırsa,

$$1 = E \tilde{A} + q^{-j} F \quad \dots (4.9)$$

$$1 = R \tilde{A} + q^{-(j+1)} S \quad \dots (4.10)$$

elde edilir. (4.10)' dan (4.9) çıkartılarak

$$0 = \tilde{A} (R - E) + q^{-j} (q^{-1} S - F) \quad \dots (4.11)$$

elde edilir. $(R - E)$ polinomunun mertebesi "j" dir ve aşağıdaki şekilde iki parçaya ayrılabilir:

$$R - E = \tilde{R} + r_j q^{-j} \quad \dots (4.12)$$

Bu değer (4.11)' de yerine yazılırsa,

$$\tilde{A} \tilde{R} + q^{-j} (q^{-1} S - F + \tilde{A} r_j) = 0 \quad \dots (4.13)$$

elde edilir. Bu eşitliği sağlayan değerler $\tilde{A} = 0$ olamayacağından

$$\tilde{R} = 0 \quad \dots (4.14.a)$$

$$S = q (F - \tilde{A} r_j) \quad \dots (4.14.b)$$

şeklindedir. $\tilde{A}$ ' nın birim başlangıç değerine sahip olduğu kabul edilerek,

$$r_j = f_0 \quad \dots (4.15)$$

$$s_i = f_{i+1} - \tilde{a}_{i+1} r_j \quad \dots (4.16)$$

tanımları, (4.12) ifadesi ve $G_j(q^{-1}) = E_j(q^{-1}) B(q^{-1})$

tanımı kullanılarak,

$$R(q^{-1}) = E(q^{-1}) + q^{-j} r_j \quad \dots (4.17)$$

$$G_{j+1} = B(q^{-1}) R(q^{-1}) \quad \dots (4.18)$$

elde edilir.

Böylece $A(q^{-1})$ ve $B(q^{-1})$ sistem polinomları ve $E_j(q^{-1})$ ve $F_j(q^{-1})$ için bir başlangıç çözümü verilirse, $F_{j+1}(q^{-1})$ ve $E_{j+1}(q^{-1})$ basit bir şekilde hesaplanabilir. İterasyonları başlatmak için "j-1" anı göz önüne alınarak,

$$1 = E_1 \tilde{A} + q^{-1} F_1$$

ve $\tilde{A}$ 'nın ilk elemanının 1 olması nedeni ile;

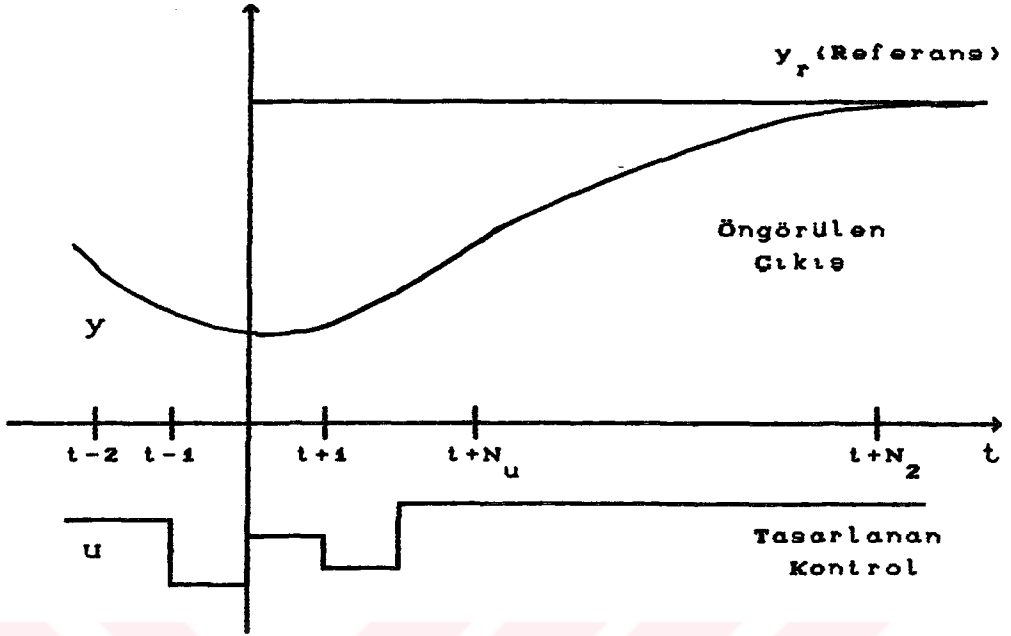
$$E_1 = 1, F_1 = q (1 - \tilde{A})$$

elde edilir.

Burada yapılan hesaplamalar, her bir "j" ufuğu için ayrı bir öngörücü kullanılmasındakinden daha doğru ve basit olmaktadır.

4.1.4.Öngörülü Kontrol Kanunu

Öngörülü kontrol kanununun amacı, gelecek zaman dilimlerindeki $y(t+j)$ sistem çıkışlarının referans değerleri $y_r(t+j)$ 'yi mümkün oldukça yakın olarak takip etmesini sağlamaktır (Şekil 4.2). Bir çok durumda istenen referans değerleri $y_r(t+j)$, sabit bir y_r değerine eşit olmaktadır, fakat robotik gibi bazı uygulama alanlarında referans değerleri $y_r(t+j)$ 'de de zaman akışı içerisinde değişimler olmaktadır.



Şekil 4.2. GPC. de Referans, Kontrol ve Çıkış büyüklükleri

Richalet ve arkadaşları [96] tarafından da önerildiği gibi, t anındaki $y(t)$ çıkışından hedeflenen referans değeri y_r ' ye aşağıdaki yaklaşım aracılığı ile de ulaşılabilir:

$$y_r(t) = y(t)$$

$$y_r(t+j) = \alpha y_r(t+j-1) + (1-\alpha) y_r \quad j = 1, 2, \dots$$

α , bir ayar parametresidir ve $0 \leq \alpha < 1$ değerlerini almaktadır. $\alpha \rightarrow 0$ hızlı kontrol, $\alpha \rightarrow 1$ ise yavaş kontrol sağlamaktadır. GPC, hem sabit hem de değişken referans yörüngeleri durumunda gayet başarılı gözükmektedir (Clarke [37]).

Öngörülü kontrol kanununun, dolayısı ile de burada

incelenen genelleştirilmiş öngörülü kontrol kanununun amacı yukarıda da açıklandığı gibi gelecekte izlenecek yörüngeyi, istenilen yörüngeye mümkün olduğu kadar yaklaştırmak ve bunu yaparken de kontrol aktivitesini gözden uzak tutmamaktır. Bu amaca ulaşmak için bir gerileyen ufuk yaklaşımı kullanılmaktadır. Her bir örnekleme anında;

1- Gelecekteki referans değerleri dizisi hesaplanır.

2- Yukarıda açıklanan yöntem kullanılarak gelecekteki sistem çıkışları $\hat{y}(t+j/t)$ öngörülür. Burada $\hat{y}(t+j/t)$ ' nin kısmen hesaplanması gereken gelecekteki kontrol büyüklükleri $u(t+j)$ ' ye bağlı olacağı unutulmamalıdır. ($j > d$ için).

3- Belirli bir kontrol ufuğundan sonra kontrol büyüklüklerindeki artım sıfır kabul edilerek gelecekteki hataların ve kontrol büyüklüğünün fonksiyonu olan bir kuadratik maliyet fonksiyonu minimize edilir.

4- Hesaplanan kontrol büyüklükleri dizisinin ilk elemanı olan $u(t)$ sisteme uygulanır ve gelecek örnekleme anında tekrarlanmak üzere veri vektörleri bir kaydırılır.

Adım 3- de bahsedilen kuadratik maliyet fonksiyonunun en genel ve GPC' de kullanılan şekli aşağıdaki gibidir;

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \dots (4.19)$$

Burada,

N_1 : Minimum maliyet ufuğu

N_2 : Maksimum maliyet ufuğu (Öngörü ufuğu)

$\lambda(j)$: Kontrol ağırlıkları dizisi

E : Beklenti operatörüdür.

(4.19) maliyet fonksiyonunda, N_2 öngörü ufuğu genellikle cevabın tamamını içerecek şekilde seçilir, en küçük değer olarak $B(q^{-1})$ ' in mertebesinden daha büyük seçmek tüm durumların maliyete yardım etmesi açısından önemli olmaktadır (Kailath [103]). Fakat daha tipik olarak sistemin yükselme zamanına (rise-time) yakın olarak seçilir. N_1 , genellikle "1" olarak seçilmektedir, şayet sistemin ölü zamanı "d" biliniyorsa hesaplamaları minimize etmek için "d" ye eşit veya daha büyük olarak da seçilebilir. Sistem modellerinin büyük bir kısmının $N_1 = 1$ ve $N_2 = 10$ değerleri ile stabilize edilebileceği gösterilmiştir. (Clarke [37]). Basitlik açısından bundan sonra $\lambda(j) \rightarrow \lambda$, $N_1 \rightarrow 1$ ve $N_2 \rightarrow N$ olarak gösterilecektir.

(4.5) ifadesi tekrar göz önüne alınarak gelecek zaman dilimlerindeki çıkışlar

$$y(t+1) = G_1 \Delta u(t) + F_1 y(t) + E_1 \xi(t+1)$$

$$y(t+2) = G_2 \Delta u(t+1) + F_2 y(t) + E_2 \xi(t+2)$$

$$\dots \quad \dots \quad \dots \quad \dots \quad \dots \quad (4.20)$$

$$\dots \quad \dots \quad \dots \quad \dots$$

$$y(t+N) = G_N \Delta u(t+N-1) + F_N y(t) + E_N \xi(t+N)$$

şeklinde elde edilir.

Yukarıdaki ifadelerden $y(t+j)$ incelenecek olursa üç ayrı terimden meydana geldiği anlaşılır: Gelecekteki

kontrol büyüklükleri ,ölçülen değişkenler ile birlikte geçmişteki kontrol büyüklükleri ve gelecekteki gürültü sinyalleri. (Öngörülerin hesaplanmasında gürültü dizisi $\hat{y}(t+j)$ öngörü hatası olarak kabul edilmekte idi, (4.6) ve (4.7) nolu ifadeler). $y(t+j)$ 'nin t anında bilinen sinyallerden oluşan bileşenleri $f(t+j)$ ile gösterilirse;

$$\mathbf{f}(t+1) = [G_1(q^{-1}) - g_{10}] \Delta u(t) + F_1 y(t)$$

$$f(t+2) = q [G_2(q^{-1}) - q^{-1}g_{21} - g_{20}] \Delta u(t) + F_2 y(t)$$

• • • • •

• • • • •

... (4.21)

elde edilir. Burada $G_i(q^{-1}) = g_{i0} + g_{i1}q^{-1} + \dots$ şeklindedir. $f(t+j)$ terimi (4.1) ifadesindeki $\hat{y}_f$ serbest cevabına karşılık gelmektedir.

(4.20) ifadeleri aşağıdaki anahtar vektör formunda tekrar yazılabilir.

$$\hat{\mathbf{y}} = \mathbf{G} \tilde{\mathbf{u}} + \mathbf{f} \quad \dots (4.22)$$

Burada,

$$\hat{\mathbf{y}} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)]^T$$

$$\tilde{\mathbf{u}} = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)]^T$$

$$\mathbf{f} = [f(t+1), f(t+2), \dots, f(t+N)]^T$$

$\hat{y}$, $\hat{u}$ ve f vektörleri $N \times 1$, G matrisi ise $N \times N$ boyutundadır. $G_i(q^{-1})$ 'deki ilk j terim, basamak cevabının parametreleridir ve $j = 0, 1, 2, \dots$ için $g_{ij} = g_j$ dir. Yani; $g_{10} = g_0$, $g_{21} = g_1$, $g_{32} = g_2$ v.s. dir.

$$G = \begin{bmatrix} g_0 & 0 & . & . & . & 0 \\ g_1 & g_0 & . & . & . & 0 \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ g_{N-1} & g_{N-2} & . & . & . & g_0 \end{bmatrix}$$

Şayet sistem ölü zamanı $d > 1$ ise G matrisinin ilk $d-1$ satırı sıfır olacaktır. Fakat N_1 , d değerine eşit alınarak ilk satırların sıfır olması önlenebilir.

Yukarıdaki vektör ve matris tanımları ve referans değerleri için,

$$y_r = [y_r(t+1), y_r(t+2), \dots, y_r(t+N)]^T$$

dizisi göz önüne alınarak (4.19) maliyet fonksiyonu aşağıdaki şekilde yeniden yazılabilir:

$$J(1,N) = E \left\{ (y - y_r)^T (y - y_r) + \lambda \tilde{u}^T \tilde{u} \right\} \dots (4.23a)$$

$$J(1,N) = E \left\{ (G \tilde{u} + f - y_r)^T (G \tilde{u} + f - y_r) + \lambda \tilde{u}^T \tilde{u} \right\}$$

$$\dots (4.23b)$$

Yukarıdaki maliyet fonksiyonunu minimize eden $\tilde{u}$ kontrol artımları dizisini bulmak için, J maliyet fonksiyonunun $\tilde{u}$ 'ya göre türevi alınarak sıfıra eşitlenir ve buradan da $\tilde{u}$ değeri çekilir. Daha basit bir gösterimle;

$$J = (G \tilde{u} + f - y_r)^2 + \lambda \tilde{u}^2$$

şeklinde yazılır ve $\tilde{u}$ 'ya göre türev alınarak sıfıra

eşitlenirse;

$$\cancel{G} (G \tilde{u} + f - y_r) G^T + \cancel{\lambda} \tilde{u} = 0$$

$$(G \tilde{u} + f - y_r) G^T + \lambda \tilde{u} = 0$$

$$G^T G \tilde{u} + G^T (f - y_r) + \lambda \tilde{u} = 0$$

$$\tilde{u} (G^T G + \lambda I) + G^T (f - y_r) = 0$$

$$\tilde{u} = \frac{G^T (y_r - f)}{G^T G + \lambda I}$$

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (y_r - f) \quad \dots (4.24)$$

elde edilir.

$\tilde{u}$ 'nun ilk elemanı olan $\Delta u(t)$ 'den, o an için geçerli olan $u(t)$ kontrol büyüklüğü,

$$\Delta u(t) = u(t) - u(t-1)$$

olduğundan

$$u(t) = u(t-1) + (G^T G + \lambda I)^{-1} G (y_r - f) \quad \dots (4.25)$$

olarak elde edilir. Görüldüğü gibi kontrol büyüklüğü sıfır offset sağlayan integral etki içermektedir.

4.1.5. Kontrol Ufuzu

(4.25) ifadesinde kullanılan G matrisi $N \times N$ boyutlarındadır. Her ne kadar adaptif olmayan durumda

$(G^T G + \lambda I)$ 'nın tersinin sadece bir kere alınması gerekecekse de, adaptif durumda her bir örnekleme anında ters alma işleminin yapılması, hesapsal açıdan büyük bir yük getirecektir. Ayrıca ölü zaman için hatalı bir değer kabul edilmişse $G^T G$ tekil olacak ve kontrol kanununun gerçekleştirilebilmesi için sıfırdan farklı bir λ ağırlığına gerek duyulacaktır. λ 'nın gerçek değeri önceden bilinmeyeceği için bu seçim kullanıcıyı bir hayli zorlayacaktır.

Yukarıda belirtilen hesaplama yükünü azaltmak $(N \times N)$ 'lık matrisin tersini almak hesapsal açıdan yük getirmektedir) ve kontrolörün kararlılık özelliklerini düzeltmek için gelecekteki kontrol büyüklükleri ile ilgili olarak DMC metodunda da [97] kullanılan bir kabul yapılmaktadır. Bu kabule göre, $N_U < N_2$ olacak şekilde seçilen bir N_U ufuğundan sonra kontrol artımlarının "0" olduğu varsayılmıştır, yani, $j > N_U$ için

$$\Delta u(t+j-1) = 0$$

Burada N_U "Kontrol Ufugu" olarak adlandırılmaktadır. N_U kontrol ufuğunun göz önüne alınması ile, (4.19) maliyet fonksiyonunda ufak bir değişiklik yapılarak sağ taraftaki ikinci terime ait toplam, $1-N_2$ arası yerine $1-N_U$ arasında gerçekleştirilecektir;

$$J(N_1, N_2, N_U) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \sum_{j=1}^{N_U} \lambda [\Delta u(t+j-1)]^2 \right\} \quad \dots (4.26)$$

Bu durum, maliyet fonksiyonu terimlerinde bir kaç gelecek zaman diliminden sonra (N_U adet), kontrol değişimleri üzerine sonsuz ağırlık uygulanmasına karşılık gelmektedir. Örneğin $N_U=1$ ise sadece tek bir

kontrol değişimi $\Delta u(t)$ göz önüne alınacak, daha sonraki $u(t+j)$ kontrol büyüklüklerinin hepsi $u(t)$ 'ye eşit alınacaktır. $N_u = 1$ alındığında kontrol kanunu genellikle düzgün ve ağır hareketler sağlamakta, N_u 'nun değeri arttıkça daha aktif kontrol elde edilmektedir.

$N_u < N_2$ seçimi, hesapsal yükü önemli ölçüde azaltmaktadır. $\tilde{u}$ vektörünün boyutu $N_u \times 1$ olmakta ve öngörü denklemleri

$$\hat{y} = G_1 \tilde{u} + f \quad \dots (4.27)$$

şeklını almaktadır. Burada;

$$G_1 = \begin{bmatrix} g_0 & 0 & \cdot & \cdot & 0 \\ g_1 & g_0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & g_0 \\ g_{N_2-1} & g_{N_2-2} & \cdot & \cdot & g_{N_2-N_u} \end{bmatrix} \quad [N_2 \times N_u]$$

şeklindedir. Bu durumda kontrol kanunu

$$\tilde{u} = [G_1^T G_1 + \lambda I]^{-1} G_1^T (y_r - f) \quad \dots (4.28)$$

olmakta ve tersi alınacak matrisin boyutu $N_u \times N_u$ 'ya indirgenmektedir. Özel olarak $N_u = 1$ ise, işlemler skaler hesaplama işlemine dönüşmektedir.

4.1.6. Çıkış ve Kontrol Ufuklarının Seçimi

N_1 - Minimum çıkış ufuğu : Sayet ölü zaman "d" biliniyorsa, gereksiz hesaplamaları önlemek için N_1 ufuğu "d" den küçük seçilmemelidir. "d" bilinmiyor veya zaman

içerisinde değişiyorsa N_1 , "1" e eşit alınabilir. Bu durumda, $B(q^{-1})$ ' in mertebesi "d" nin alabileceği tüm değerleri içerebilecek şekilde arttırılabilir.

N_2 - Maksimum çıkış ufuğu (Öngörü ufuğu) : Genellikle $B(q^{-1})$ ' in mertebesinden büyük seçilmelidir. Daha tipik olarak sistem yükselme zamanına (rise-time) eşit alınmalıdır.

N_u - Kontrol ufuğu: Önemli bir dizayn parametresidir. basit bir sistem için $N_u = 1$ değeri genellikle kabul olunur kontrol sağlamaktadır. Kompleks sistemler için N_u ' nun daha büyük değerleri uygun olmaktadır. Böyle sistemler için N_u , en küçük değer olarak kararsız veya kötü sönümlenmiş pollerin sayısına eşit alınırsa daha iyi bir kontrol sağlanmaktadır. $N_u = N_2$ iken GPC kontrolörü minimum-varyans bir kontrolöre dönüşeceğinden, N_u ' nun dikkatli bir şekilde seçilmesi gerekmektedir.

Kısaca özetlemek gerekirse, GPC iki ayrı şekilde kullanılabilmektedir: $N_1=1$, N_2 sistem yükselme zamanına eşit ve $N_u=1$ olarak alınarak proses kontrolü için uygun bir performans elde edilmektedir. Yüksek performanslı uygulamalar için ise N_u 'nun değerini arttırmak daha uygun olmaktadır.

4.1.7. GPC 'nin Özellikleri ve Diğer Yaklaşımlar ile İlişkisi

GPC, beş anahtar fikrin birleştirilmesi ile oluşturulmuştur. Bunlar:

1- CARMA sistem modeli yerine CARIMA sistem modelinin kabul edilmesi,

2- Sistem ölü zamanından daha büyük ve en küçük

değer olarak model mertebesine eşit sonlu bir ufuk üzerinden uzun menzilli öngörülerin kullanılması,

3- Diophantine denkleminin ardışık olarak kullanılması,

4- Maliyet fonksiyonunda kontrol artımlarına bir ağırlığın uygulanması,

5- Kendisinden sonra kontrol artımlarının sıfır olarak alındığı bir kontrol ufuğunun seçimi.

Bu fikirler değişik şekillerde literatürlerde incelenmiştir. GPC 'nin kuvvetli tarafı bu fikirleri mantıklı bir şekilde bünyesinde bulundurmasıdır.

Robust kontrol sağlamak düşüncesi ile, uzun menzilli öngörü kullanma fikri IDCOM algoritmasında Richalet ve arkadaşları [96] tarafından gerçekleştirilmiştir. Bu metodun bazı endüstriyel başarılarla sahip olduğu belirtiliyor ise de kararsız veya non-minimum faz açık çevrimli sistemler için uygun değildir [37]. Cutler ve Ramaker 'in DMC algoritması [97] basamak modeli cevabına dayanmakta, fakat kontrol ufuğunun temel fikrini de içermektedir. DMC algoritması non-minimum faz sistemler için etkilidir, fakat açık çevrimli kararsız prosesler için etkili olmamaktadır.

Bilinen "d" ölü zamanlı bir sistem için Clarke ve Gawthrop [55] 'un GMV yaklaşımı, GPC 'nin hem minimum ve hem de maksimum ufuklar olan N_1 ve N_2 'nin "d" 'ye eşitlendiği ve sadece bir kontrol sinyalinin $(u(t)$ veya $\Delta u(t))$ ağırlıklandırıldığı özel bir durum olarak görülebilir. Bu metod, model mertebesinin olduğundan büyük tahmin edilmesine karşı robust olarak bilinmektedir, fakat sadece kontrol ağırlığı λ 'nın dikkatle seçileceği non-minimum faz sistemlerin belirli bir sınıfını kararlı olarak kontrol edebilmektedir. Ayrıca GMV, λ büyük

değerler almadıkça, değişen ölü zamana karşı oldukça hassastır. GPC, GMV 'nin dezavantajları hariç robustluk özelliklerini paylaşmaktadır.

$N_1 = N_2 > d$ ve $N_u = 1$, $\lambda = 0$ seçilerek GPC, Ydstie [100]' nin genişletilmiş ufuk (extended horizon) yaklaşımına dönüşür. Bu, teorik olarak kararlı non-minimum faz bir sistem için, kararlı bir kontrolör olarak gösterilmektedir. Ydstie, CARMA modeli kullanmıştır ve metod açık çevrimli kararsız prosesleri stabilize edecek durumda görülmemektedir. Gerçekten de kötü sönümlenmiş pollere sahip bir sistem için yapılan simülasyon sonuçları, GPC dizaynından farklı olarak genişletilmiş ufuk metodunun kararsız olduğunu göstermektedir [37]). Bu durum, kararlılık için maliyet fonksiyonunda birden daha fazla gelecek zaman dilimindeki çıkışa gereksinme duyulmasından kaynaklanmaktadır (yani $N_2 > N_1$).

$\lambda = 0$ ve CARIMA modeli kullanılarak $N_1 = 1$, $N_2 > d$ ve $N_u = 1$ değerleri ile GPC, EPSAC algoritmasına (De Keyser ve Van Cauwenberghe [99]) indirgenmektedir. Bu, değişik araştırmacılar tarafından bazı pratik uygulamalar için özellikle faydalı bir metod olarak gösterilmiştir. GPC 'de $N_2 = 10$ alınarak yapılan simülasyon çalışmaları EPSAC 'inkine benzer sonuçlar vermiştir (Clarke [37]).

Peterka 'nın öngörülü kontrolörü [104] ana prensip olarak GPC 'ye en yakın olanıdır. CARIMA benzeri bir model ve benzer bir maliyet fonksiyonu kullanılmıştır. Esas yoğunluk $N_2 \rightarrow \infty$ olma durumunda toplanmıştır. Peterka kontrol ağırlığı için iki değer göz önüne almaktadır (İlk kontrol değerleri takımı için λ_1 ve son δB adım için λ_2), fakat $\lambda_2 = \infty$ olan GPC durumunu göz önüne almamıştır. GPC genel olarak kısıtlanmış bir kontrol ufuğu ile sınırlı aşamalı (finite-stage) bir yaklaşımdır. Bu hem gelecekteki kontrol büyüklükleri ve

hem de kontrol oranlarındaki kısıtlamaların, GPC hesaplamalarının uygun bir şekilde modifiye edilmesi ile göz önüne alınabilmesi demektir. Bu genelleştirme sınırsız aşamalı (infinite-stage) dizaynlarda mümkün olmamaktadır. Bununla beraber, Peterka 'nın metodu adaptif kontrol için önemli ve uygulanabilir bir yaklaşımdır.

4.2. Çok Giriş-Çok Çıkışlı (MIMO) Genelleştirilmiş Öngörülü Kontrol Algoritması

Çok giriş-çok çıkışlı sistemler için genelleştirilmiş öngörülü kontrol algoritmasının çıkarılmasında izlenen yol tek giriş-tek çıkışlı durum için izlenen yolun aynısıdır. Kontrol kanununun detaylı olarak çıkarılması Ek A' da adım adım verilmiştir. İşlemler, SISO durumu için değişken sayısı 1 kabul edilerek aynen geçerli olmaktadır. En önemli fark SISO algoritmadaki tek boyutlu elemanlar yerine vektörlerin kullanılması, matris elemanlarının da ayrıca matrislerden meydana geliyor olmasıdır.

Çok değişkenli bir sistemin genel CARIMA modeli aşağıdaki şekilde yazılabilir;

$$A(q^{-1}) \Delta y(t) = B(q^{-1}) \Delta u(t-1) + C(q^{-1}) \xi(t) \quad (4.29a)$$

veya (4.3) ifadesine benzetilerek

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-1) + C(q^{-1}) \xi(t)/\Delta \quad (4.29b)$$

Burada, $y(t)$, $u(t)$ ve $\xi(t)$ çıkış, giriş ve beyaz gürültü vektörleridir. $A(q^{-1})$, $B(q^{-1})$ ve $C(q^{-1})$ polinom matrisleridir ve $A(0) = I$ (birim matris)' dir.

Örnek olarak (2x2)' lik bir sistem için (iki giriş

ve iki çıkışa sahip bir sistem) (4.29) ifadesi açık formda aşağıdaki şekilde yazılabilir (Bozucu gözönüne alınmamıştır).

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \Delta y_1(t) \\ \Delta y_2(t) \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} \Delta u_1(t-1) \\ \Delta u_2(t-1) \end{bmatrix}$$

... (4.30)

SISO algoritmada olduğu gibi, çok değişkenli öngörülü bir kontrolör de aşağıdaki maliyet fonksiyonunu minimize etmeye çalışmaktadır.

$$J(N_1, N_2, N_U) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - y_r(t+j)]^2 \right\} + \sum_{j=1}^{N_U} \Delta u(t+j-1) \Lambda(j) \Delta u(t+j-1)$$

... (4.31)

Burada, $y(t+j)$, $y_r(t+j)$ ve $u(t+j)$ sırası ile $(t+j)$ anlarındaki çıkış, referans büyüklükleri ve kontrol sinyali vektörleridir. Diophantine denklemi çözülerek veya serbest cevap/zorlanmış cevap ilişkileri kullanılarak [31], (4.31) ifadesini minimize eden kontrol kanunu aşağıdaki şekilde elde edilir (Detaylı bilgi için Ek A. ya bakınız).

$$\Delta u(t) = [I \ 0 \ 0 \ \dots \ 0] (G^T G + \Lambda)^{-1} G^T (y_r - f) \dots (4.32)$$

Burada I , problemin değişken sayısına göre uygun boyutlardaki birim matrisi göstermektedir ve,

$$G = \begin{bmatrix} G_0 & 0 & . & . & 0 \\ G_1 & G_0 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & G_0 \\ G_{N_2-1} & G_{N_2-2} & . & . & G_{N_2-N_U} \end{bmatrix}$$

$$\Lambda = \text{diag} \left\{ \Lambda(1), \Lambda(2), \dots, \Lambda(N_2) \right\}^T$$

$$y_r = [y_r(t+1)^T, y_r(t+2)^T, \dots, y_r(t+N_2)^T]^T$$

$$f = [f(t+1)^T, f(t+2)^T, \dots, f(t+N_2)^T]^T$$

şeklindedir. Λ , kontrol ağırlıkları dizisidir. $f(t+j)$, sistemin serbest cevap vektörleridir [30]. Ayar parametrelerinin (N_2, N_U, λ) etkisi ve diğer yaklaşımlar ile olan ilişkiler SISO durum için belirtildiği şekilde geçerlidir, bu yüzden bu bölümde detaylı olarak anlatılmamıştır.

4.3. GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROL AKIŞ ŞEMASI

Genelleştirilmiş öngörülü kontrol algoritmalarında her bir örnekleme anında aşağıdaki işlemler tekrarlanmaktadır.

1- Gelecekteki referans dizisi hesaplanır.

2- Öngörü algoritması uyarınca gelecekteki sistem çıktıları $\hat{y}(t+j)$ öngörülür.

3- Belirli bir kontrol ufağı N_U ' dan sonraki kontrol değişmelerinin sıfır olduğu varsayılarak gelecekteki

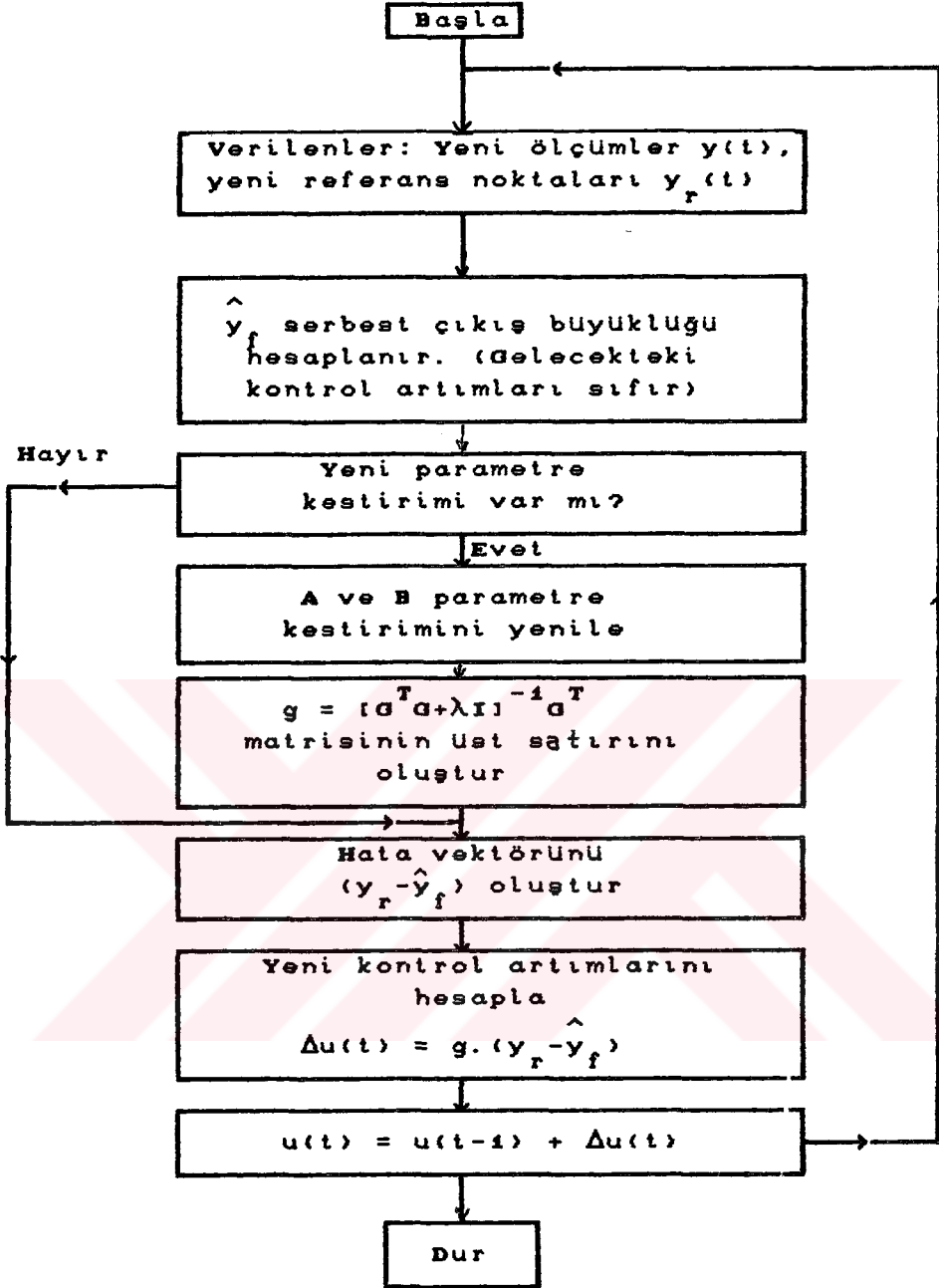
hataların uygun bir karesel formu minimize edilir.

4- Bu şekilde hesaplanan $u(t)$ kontrol dizisinin ilk elemanı sisteme uygulanır ve hesaplar gelecek örnekleme anında tekrarlanmak üzere veri vektörleri bir kaydırılır.

5- Gelecek örnekleme anında uygulanacak giriş büyüklüğünün hesaplanması için 1. adıma geri dönlür.

Bu adımlar bir akış diyagramı üzerinde şekil 4.3' de gösterilmiştir.





Sekil 4.3. Genelleştirilmiş Öngörülü Kontrol Algoritmasının Akış Şeması

BÖLÜM 5. GENELLEŞTİRİLMİŞ ÖNGÖRÜLÜ KONTROL ALGORİTMASI İLE İLGİLİ GENEL SİMÜLASYON ÇALIŞMALARI

Genelleştirilmiş öngörülü kontrol algoritmasının performansını daha iyi gösterebilmek için değişik simülasyon çalışmaları yapılmıştır. Bu çalışmalarda, tek giriş-tek çıkışlı (SISO) ve çok giriş-çok çıkışlı (MIMO) sistemler ayrı ayrı düşünülmüştür.

5.1. SISO Simülasyon Çalışmaları

SISO simülasyon çalışmaları sırasında iki ana hedef gözetilmiştir.

İlk olarak genelleştirilmiş kontrol algoritmasının bünyesinde bulunan ayar parametrelerinin (N_2 , N_U ve λ) etkilerini gösterebilmek için ayrık zamanlı bir model seçilmiş ve değişik ayar parametre değerleri için elde edilen sonuçlar yorumlanmıştır. İkinci olarak, ise farklı sistemler ve değişik bozucular için genelleştirilmiş öngörülü kontrol algoritmasının performansı incelenmiştir.

5.1.1. SISO Genelleştirilmiş Öngörülü Kontrol Algoritmasında Ayar Parametrelerinin Etkisi

Bu çalışmadan amaçlanan, ayar parametrelerinin kontrol algoritmasının performansı üzerindeki etkilerini

göstermektedir. Basitlik açısından simülasyon çalışması sırasında hiç bir bozucu etki göz önüne alınmamış ve parametre kestirimi için en küçük kareler yöntemi kullanılmıştır.

Yapılan simülasyon çalışmasında aşağıda verilen birinci mertebeden non-minimum faz bir sistem kullanılmıştır.

$$(1 + a_1 q^{-1}) y(t) = (b_0 + b_1 q^{-1}) u(t-1)$$

$$(1 - 0.9q^{-1}) y(t) = (1 + 2q^{-1}) u(t-1)$$

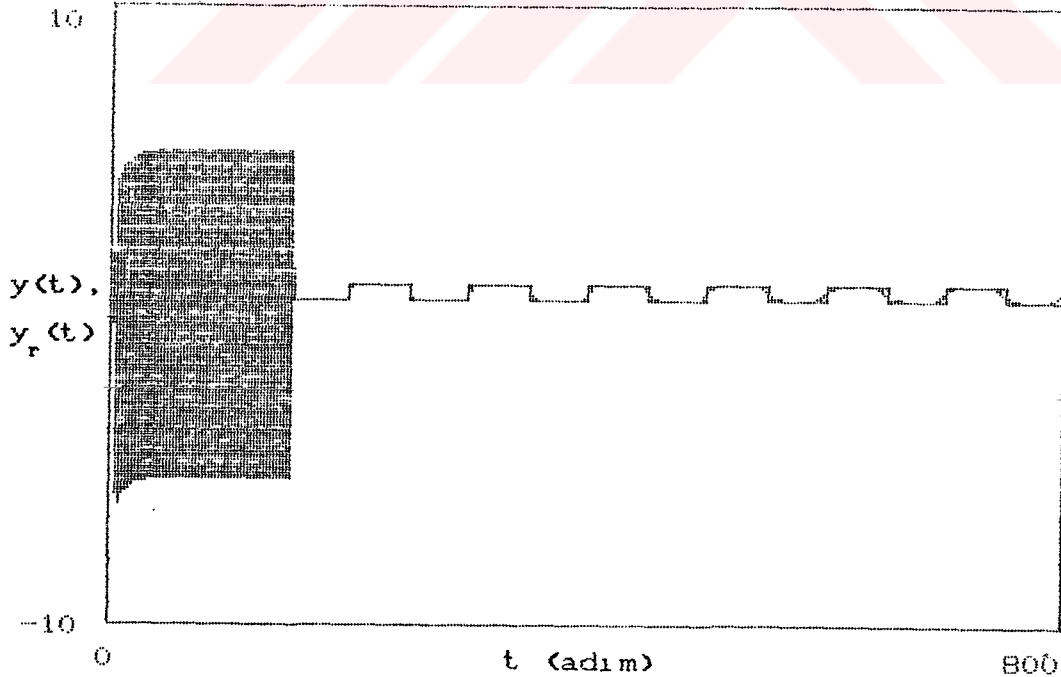
Parametre kestirme algoritmasında unutma faktörü $\mu = 0.95$, parametrelerin başlangıç değerleri $b_0 = 1$, $b_1 = 0$, $a_1 = 0$ ve $P_1(0) = 10^5 I$ olarak seçilmiştir.

Ayar parametrelerinden N_2 'nin etkisini gösterebilmek için 0.5 ve 1 değerlerinden meydana gelen bir basamak referans yörüngesi seçilmiş (Şekil 5.1) ve değişik N_2 değerleri için bu ikili basamak referans yörüngesi ardışık olarak kullanılmıştır. Her bir referans değeri periyodik olarak 50 örnekleme adımı devam etmektedir. Başlangıçta parametrelerin daha iyi yakınsayabilmesi için 50 adımlık ve 0.5 referans değerli ekstra bir simülasyon çalışması yapılmıştır. İlk referans takımında, sistemi non-minimum faz sıfırlarının iptalinden dolayı kararsız yapan $N_2 = 1$ değeri için çalışılmış ve daha sonra sıra ile $N_2 = 2, 4, 6, 8, 10, 12$ ve 14 değerleri kullanılmıştır. Her bir N_2 değeri için $N_u = 1$ ve $\lambda = 0.000001$ olarak sabit kabul edilmiştir. Sonuçlar Şekil 5.1' de gösterilmiştir. $N_2 = 1$ için kararsızlık ortaya çıktığından kararsız değişimi de gösterebilmek için Şekil 5.1.a ve 5.1.b' de ölçek büyük tutulmuş, fakat N_2 'nin diğer değerleri için ortaya çıkacak değişimleri daha belirgin olarak görebilmek için Şekil 5.1.c ve 5.1.d' de çıkış büyüklüğü 0-360 ve 350-850 adımları arasında ayrı ayrı çizdirilmiştir.

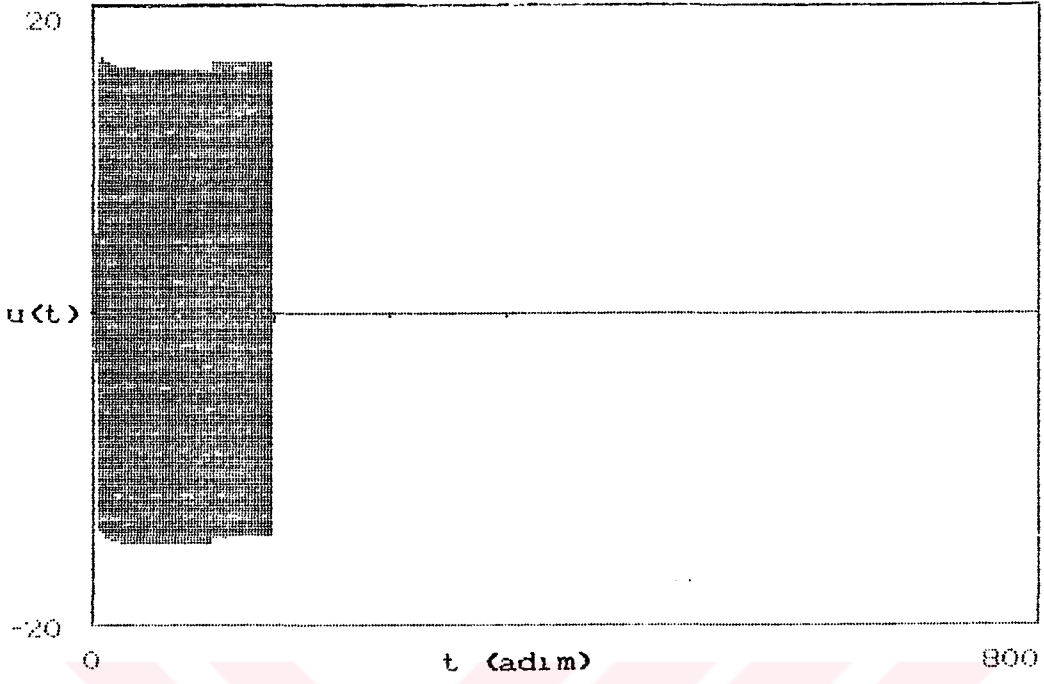
Şekil 5.1.e'de ise parametrelerin değişimi gösterilmiştir.

İkinci çalışmada $N_2 = 5$ olarak sabit alınmış ve aynı referans takımı kullanılarak $N_U = 1, 2, 3, 4$ değerleri için simülasyon çalışması yapılmıştır. Burada amaçlanan, kontrol ufuğunun kontrol performansı üzerindeki etkilerini görebilmektir. Sonuçlar Şekil 5.2' de gösterilmiştir. Her bir N_U değeri için 100 adımlık simülasyon çalışması yapılmıştır.

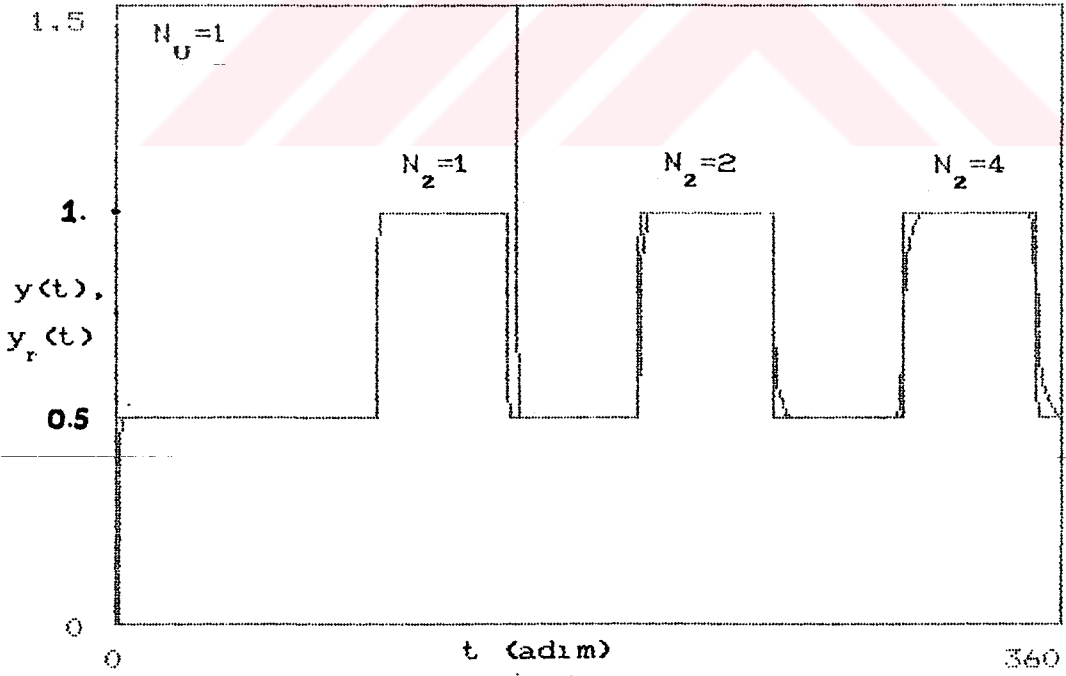
Üçüncü olarak ise, $N_2 = 5$ ve $N_U = 1$ olarak sabit kabul edilip $\lambda = 0, 0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001$ değerleri için simülasyon çalışması aynı referans yörüngesi takımı kullanılarak yapılmıştır. Burada amaçlanan ise, λ ' nin değişiminin kontrol performansı üzerindeki etkilerini göstermektir. Sonuçlar Şekil 5.3' de gösterilmiştir. Her bir λ değeri için 100 adımlık simülasyon çalışması yapılmıştır.



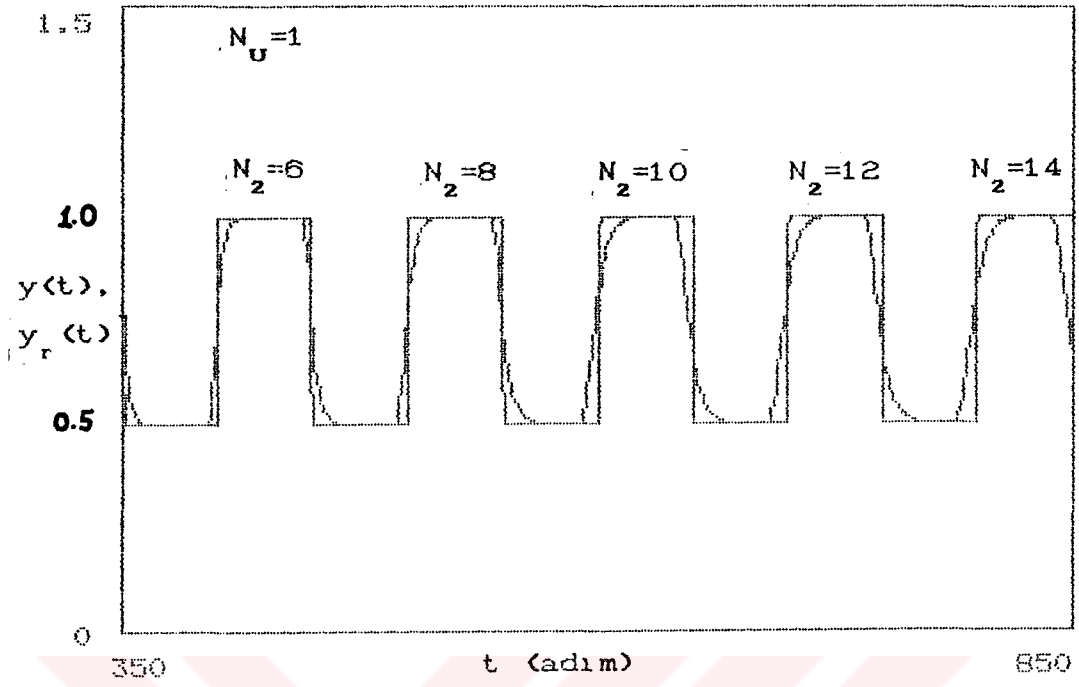
Şekil 5.1.a. N_2 ayar parametresinin kontrol performansına etkisi ($y(t)$ çıkış ve $y_r(t)$ referans büyüklüğü)



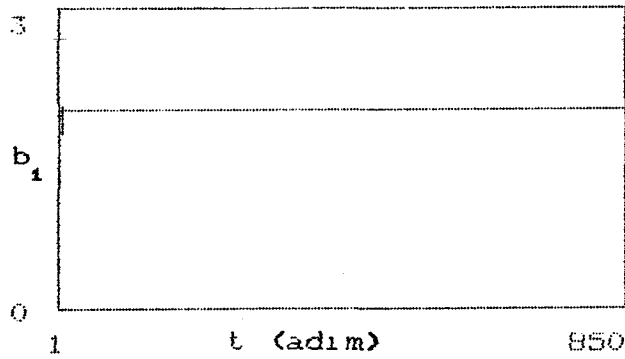
Şekil 5.1.b. $u(t)$ kontrolünün değişimi



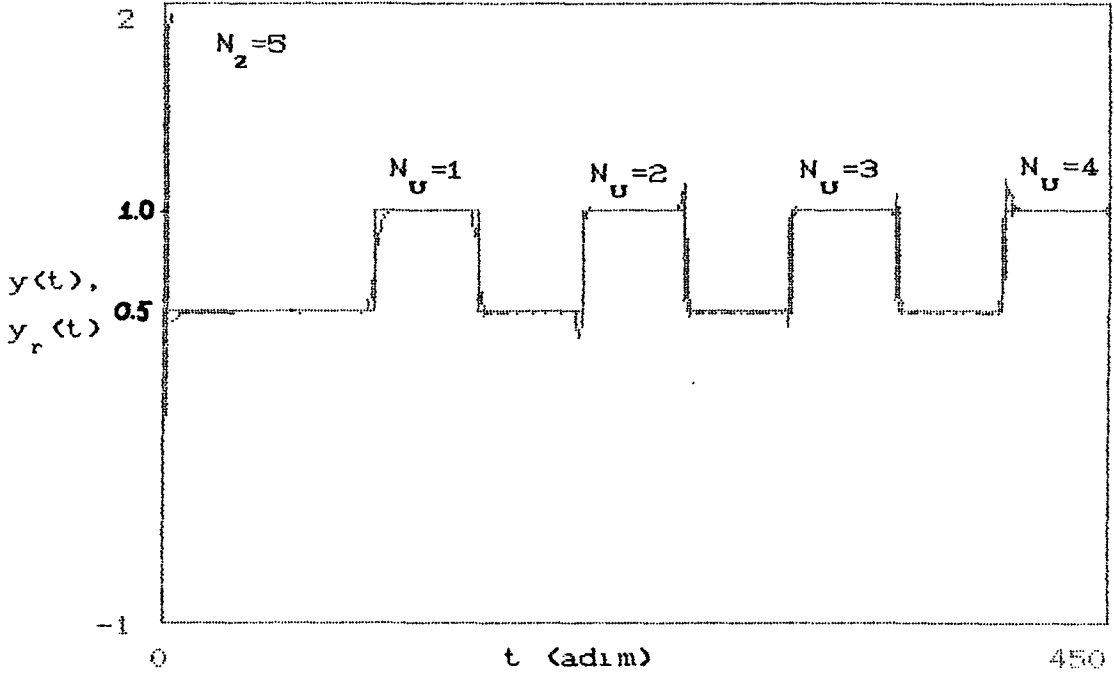
Şekil 5.1.c. $y(t)$ çıkış büyüklüğü ve referansların detaylı değişimi.



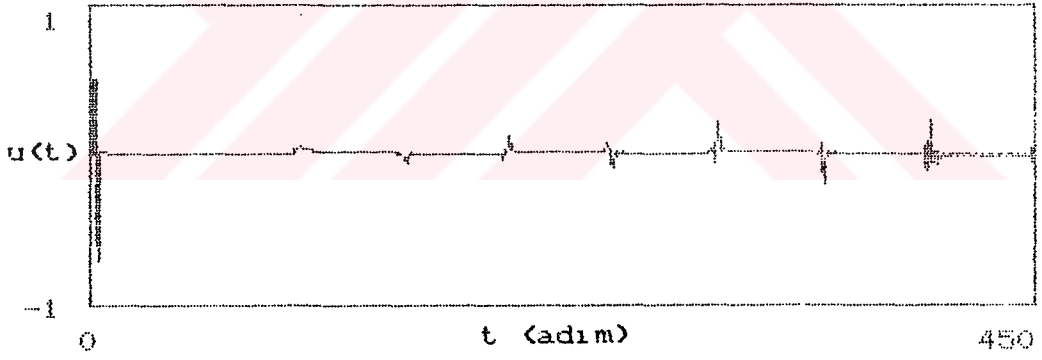
Şekil 5.1.d. $y(t)$ çıkış büyüklüğü ve referansların detaylı değişimi.



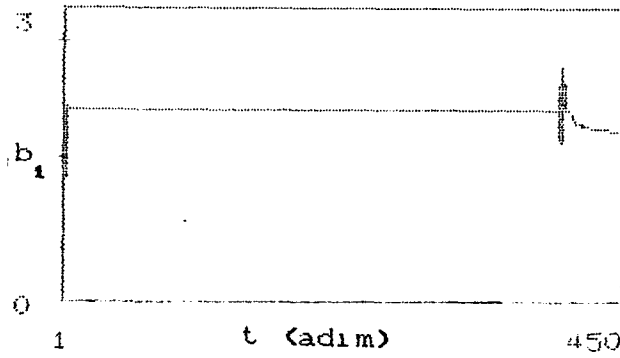
Şekil 5.1.e. b_0 ve b_1 parametrelerinin zaman ile değişimi.



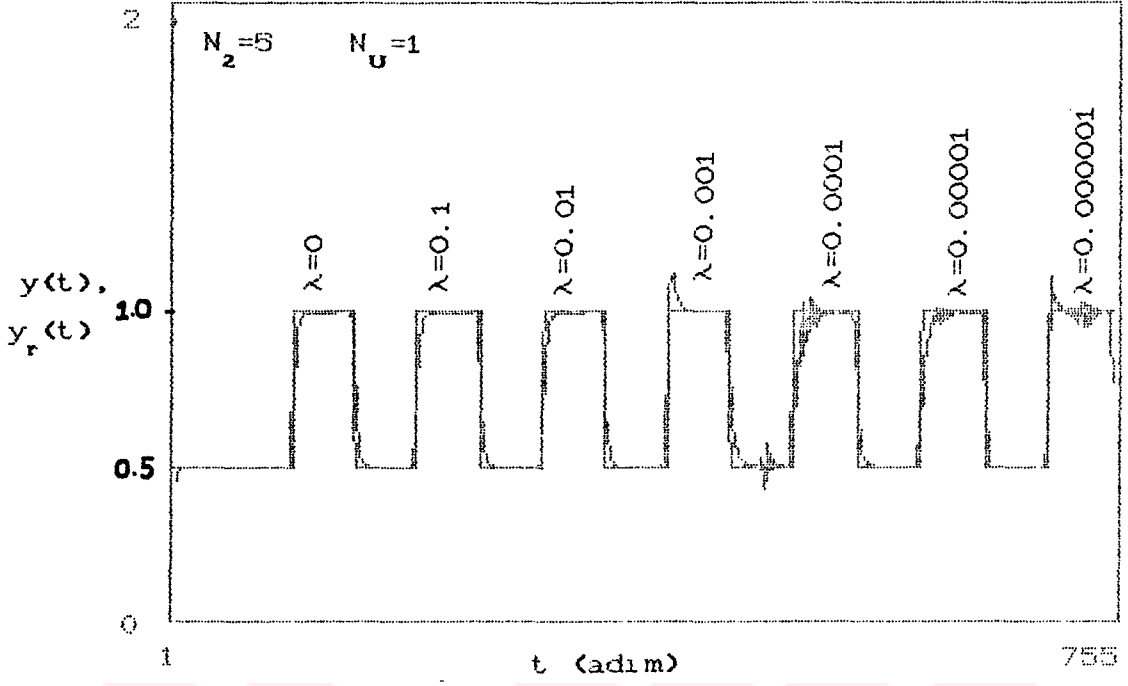
Şekil 5.2.a. N_u ' nun değişiminin kontrol performansına etkisi.
($y(t)$ ve referans değerleri)



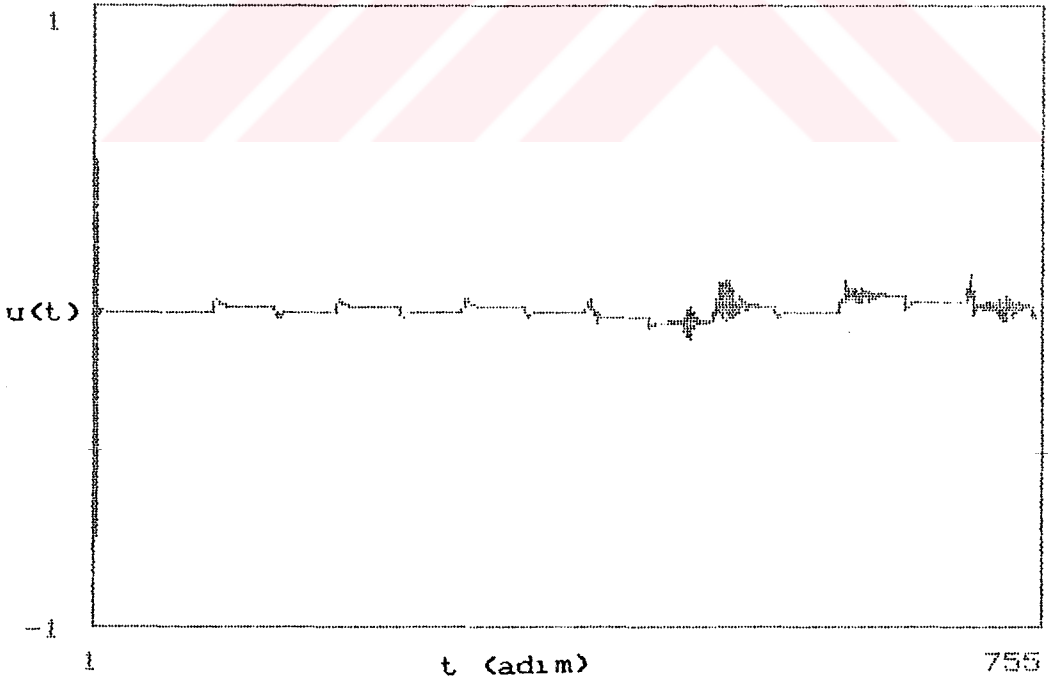
Şekil 5.2.b. $u(t)$ kontrolünün değişimi



Şekil 5.2.c. b_1 parametrelerinin zaman ile değişimi.



Sekil 5.3.a. λ 'nın kontrol performansına etkisi.
 $y(t)$ ve referans değerleri



Sekil 5.3.b. $u(t)$ kontrolünün değişimi

5.1.2. Farklı Sistemler Karşısında Genelleştirilmiş Öngörülü Kontrol Algoritması

Genelleştirilmiş öngörülü kontrol algoritmasının değişik sistemler karşısındaki performansını incelemek için 4 ayrı sistem simülasyon çalışmasına tabi tutulmuştur. İncelenen sistemlerde, $e(t)$ stokastik bozucu ve $s(t)$ ise yük bozucusunu simgelemektedir. İncelenecek sistemler:

- a)- Ayrık zamanlı deterministik sistemler,
- b)- Ayrık zamanlı stokastik sistemler,
- c)- Sürekli zamanlı sistemler, hatalı model mertebesi,
- d)- Sürekli zamanlı ve zamanla değişen sistemler,

şeklinde seçilmiştir.

5.1.2.1. Ayrık Zamanlı Deterministik Sistemler

Bu çalışmadan amaçlanan, değişik sistemler için referans değişimleri ve yük bozucularının bulunması durumunda genelleştirilmiş öngörülü kontrol algoritmasının davranışını incelemektir. Üç farklı ayrık zamanlı sistem gözönüne alınmıştır [36].

- 1)- Az sönümlü minimum faz sistem (Sistem 1)

$$\begin{aligned} y(t) &= 1.6y(t-1) + 0.8y(t-2) \\ &= 1.5u(t-1) + 0.5u(t-2) + s(t) \end{aligned}$$

- 2)- Az sönümlü non-minimum faz sistem (Sistem 2)

$$\begin{aligned} y(t) &= 1.6y(t-1) + 0.8y(t-2) \\ &= 0.5u(t-1) + 1.5u(t-2) + s(t) \end{aligned}$$

3)- Fazla sönümlü minimum faz sistem (Sistem 3)

$$y(t) = 1.489y(t-1) + 0.549y(t-2) \\ = 0.329u(t-1) + 0.269u(t-2) + s(t)$$

Simülasyon çalışmalarında her üç sistem için $N_2 = 5$, $N_u = 1$ ve $\lambda = 0.00001$ olarak sabit alınmıştır. Kullanılan parametre kestirme algoritmasında (RLS), $\mu = 0.95$, $P(0) = 10^5 I$ ve parametrelerin başlangıç değerleri olarak $b_0 = 1$ ve diğer parametreler ise 0 olarak seçilmiştir.

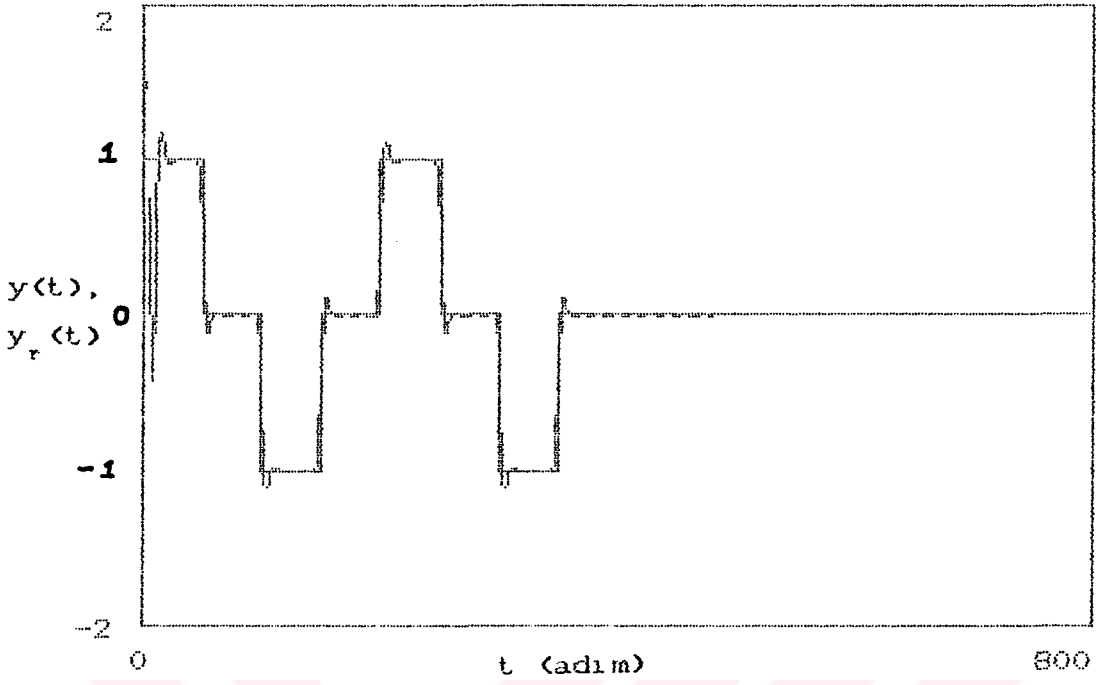
Önce, her üç sistem için de yük bozucusu $s(t) = 0$ olarak kabul edilerek -1, 0 ve +1 değerlerinden meydana gelen basamak referans büyüklükleri ve $-3 < u(t) < +3$ kısıtlaması göz önüne alınarak toplam 800 örnekleme adımı süren simülasyon çalışmaları yapılmıştır.

Daha sonra yük bozucusunun etkisini gösterebilmek için sabit ve 0.5 değerine sahip referans büyüklüğü ile toplam 300 örnekleme adımı süren bir simülasyon çalışması daha yapılmış ve 100. adımdan itibaren yük bozucusu, $s(t) = 0$ değerinden $s(t) = 0.1$ değerine değiştirilmiştir.

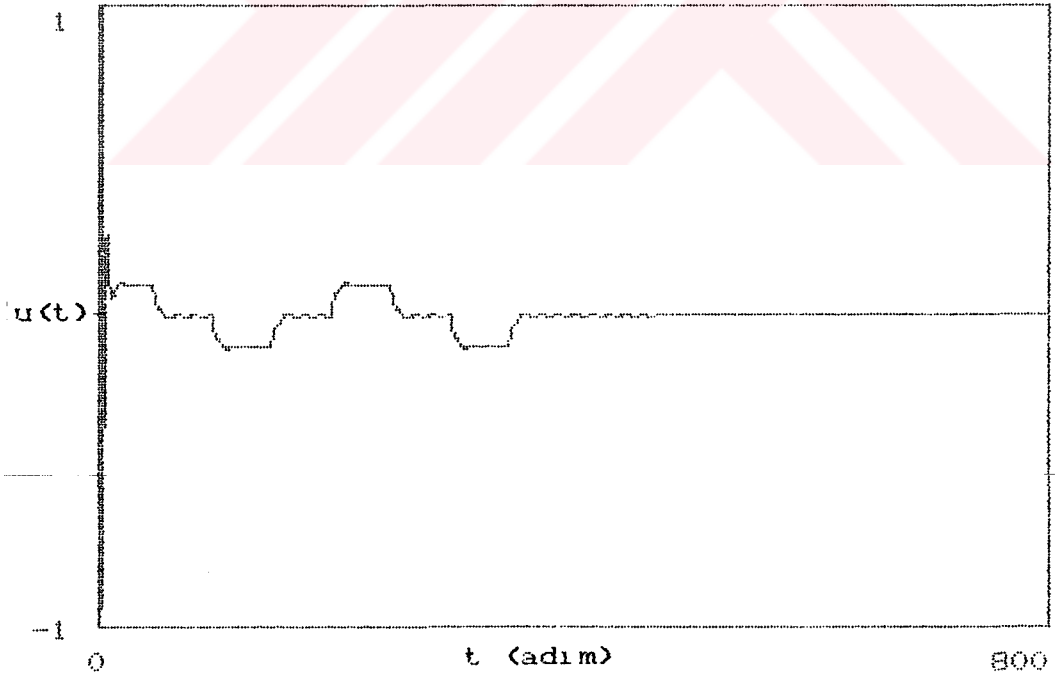
İlk çalışmanın sonuçları şekil 5.4' de, yük bozucusu ve değişimini içeren çalışmanın sonuçları ise şekil 5.5'de gösterilmiştir.

5.1.2.2. Ayırık Zamanlı Stokastik Sistemler

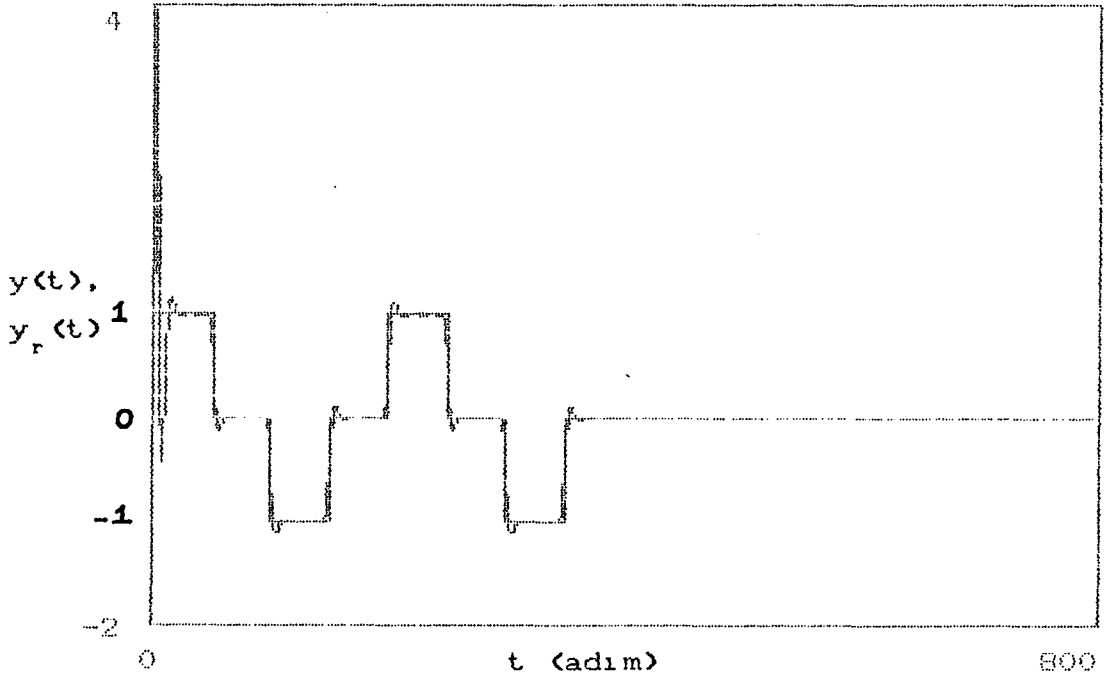
Bu çalışmadan amaçlanan ise, ayırık zamanlı sistemlerin stokastik bozucu bulunması durumunda kapalı çevrim davranışını incelemektir. İncelenen ayırık zamanlı sistem aşağıdaki şekilde seçilmiştir [36].



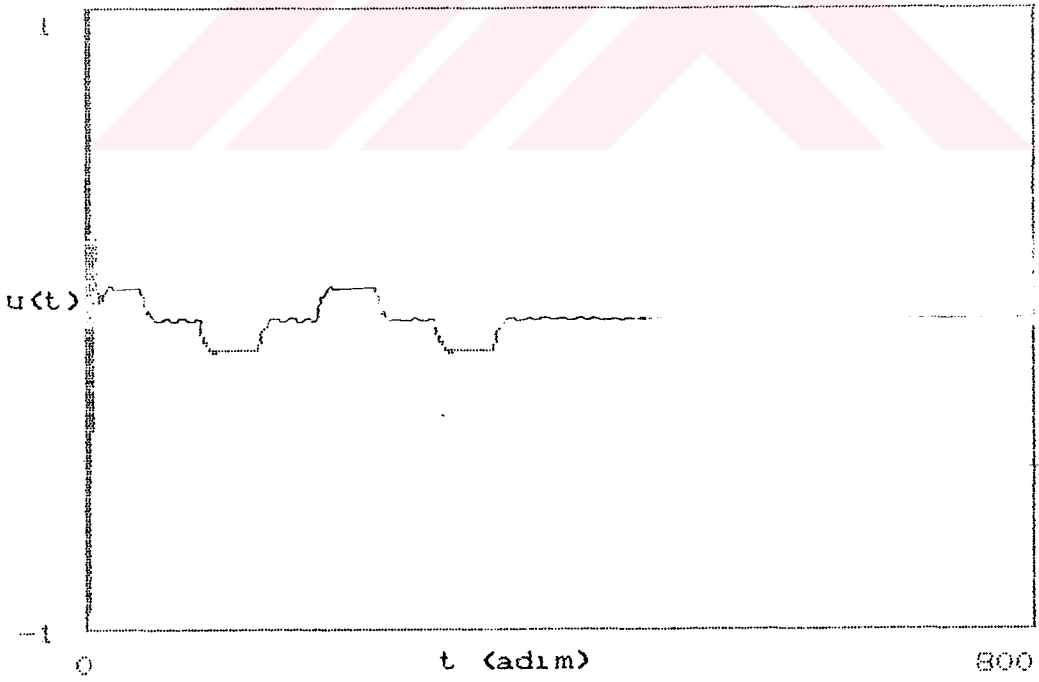
Şekil 5.4.a.1. Sistem 1 için $y(t)$ ve referans büyüklükleri, $(s(t)=0)$



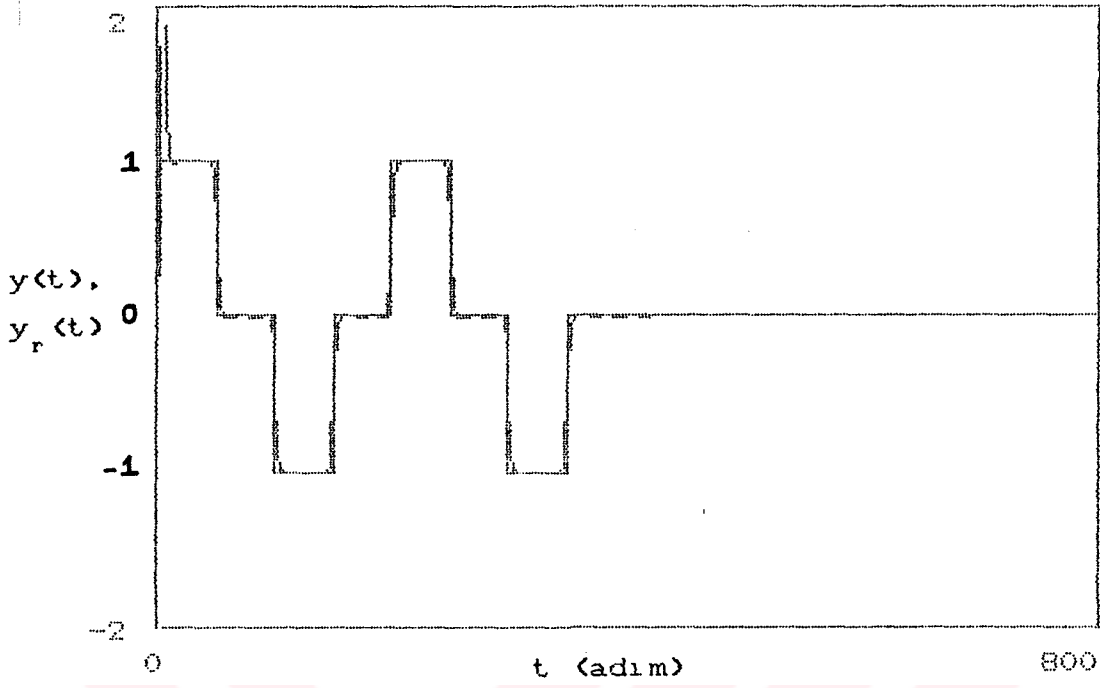
Şekil 5.4.a.2. Sistem 1 için $u(t)$ kontrolü.



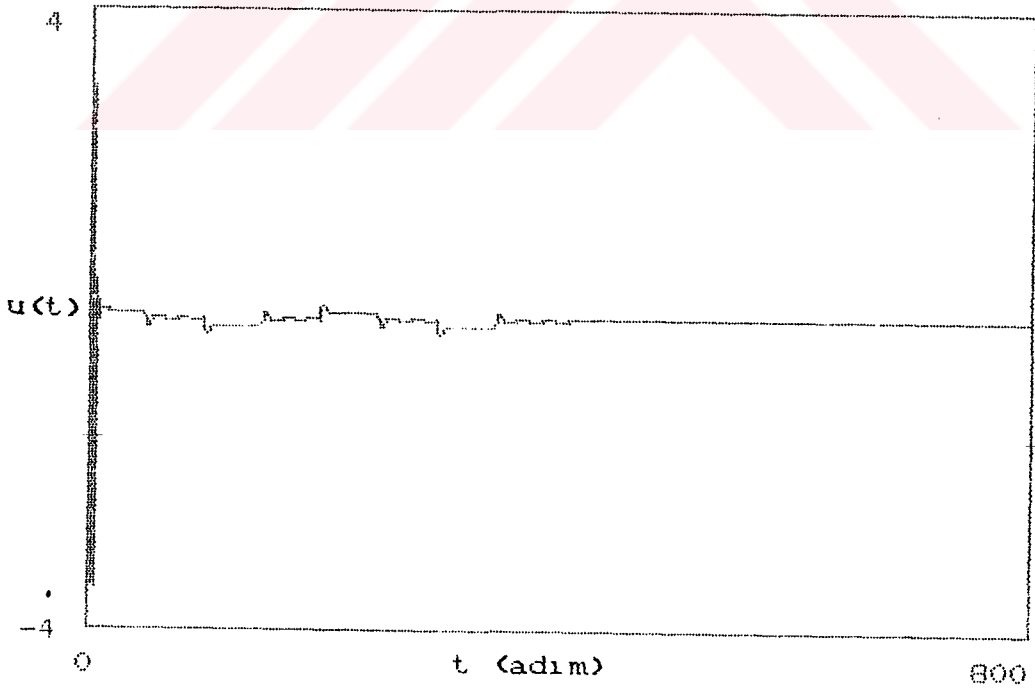
Şekil 5.4.b.1. Sistem 2 için $y(t)$ ve referans büyüklükleri, $(s(t)=0)$



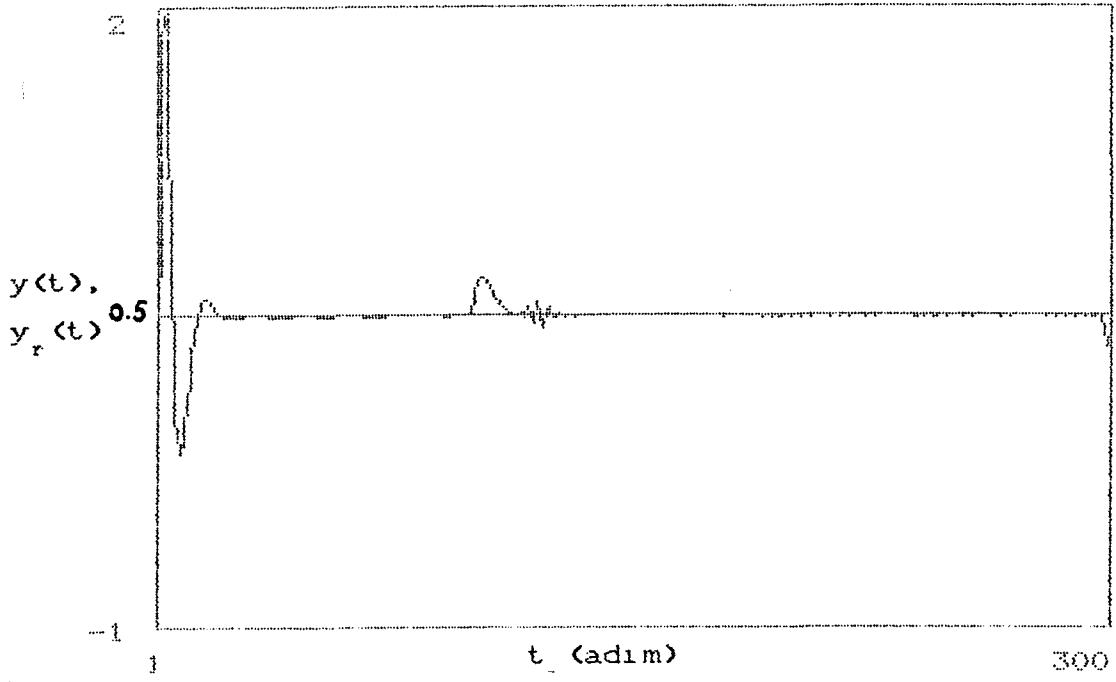
Şekil 5.4.b.2. Sistem 2 için $u(t)$ kontrolü.



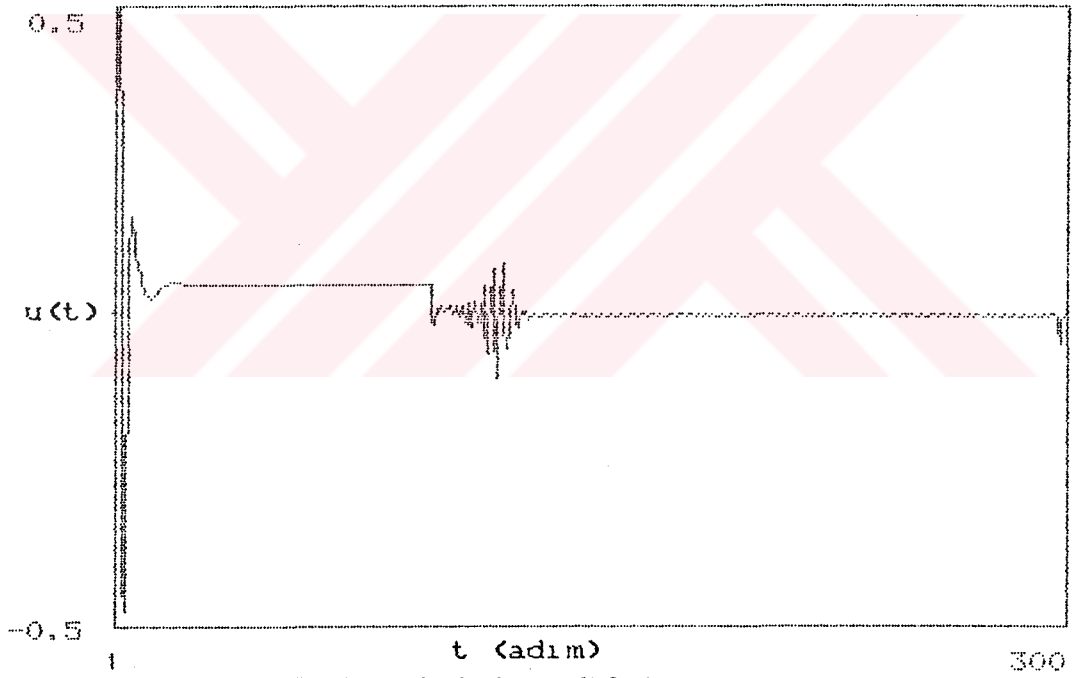
Şekil 5.4.c.1. Sistem 3 için $y(t)$ ve referans büyüklükleri, $(s(t)=0)$



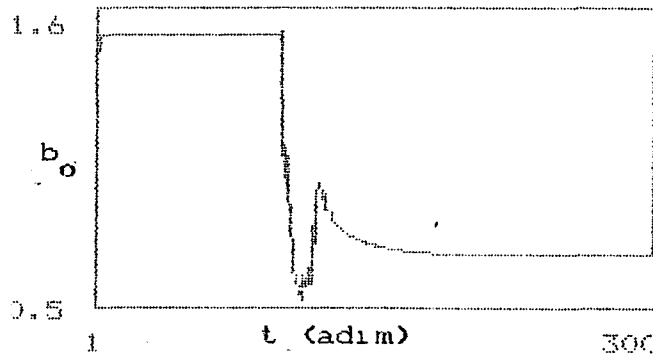
Şekil 5.4.c.2. Sistem 3 için $u(t)$ kontrolü.



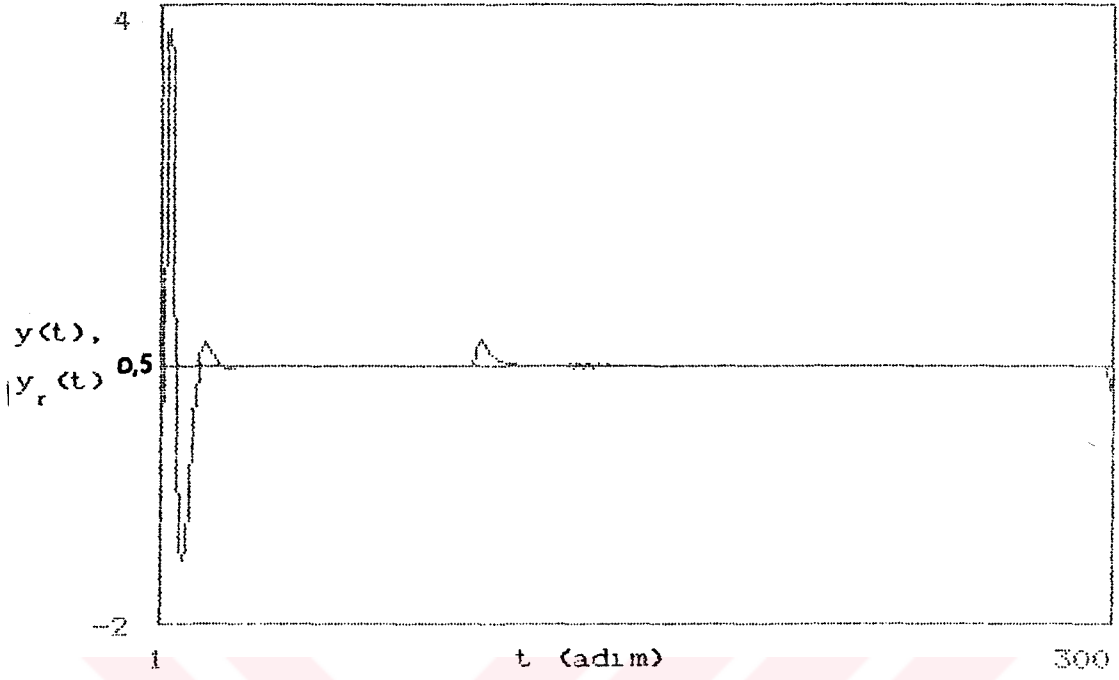
Sekil 5.5.a.1. Sistem 1 için $y(t)$ ve referans büyüklükleri, $(s(t)=0.1)$



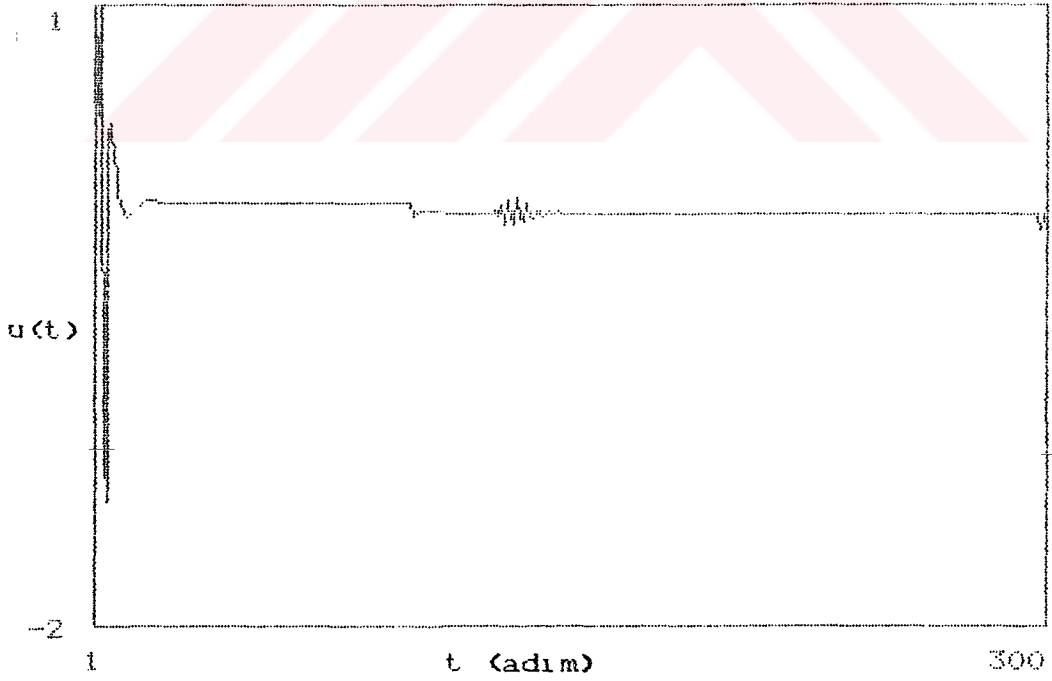
Sekil 5.5.a.2. Sistem 1 için $u(t)$ kontrolü.



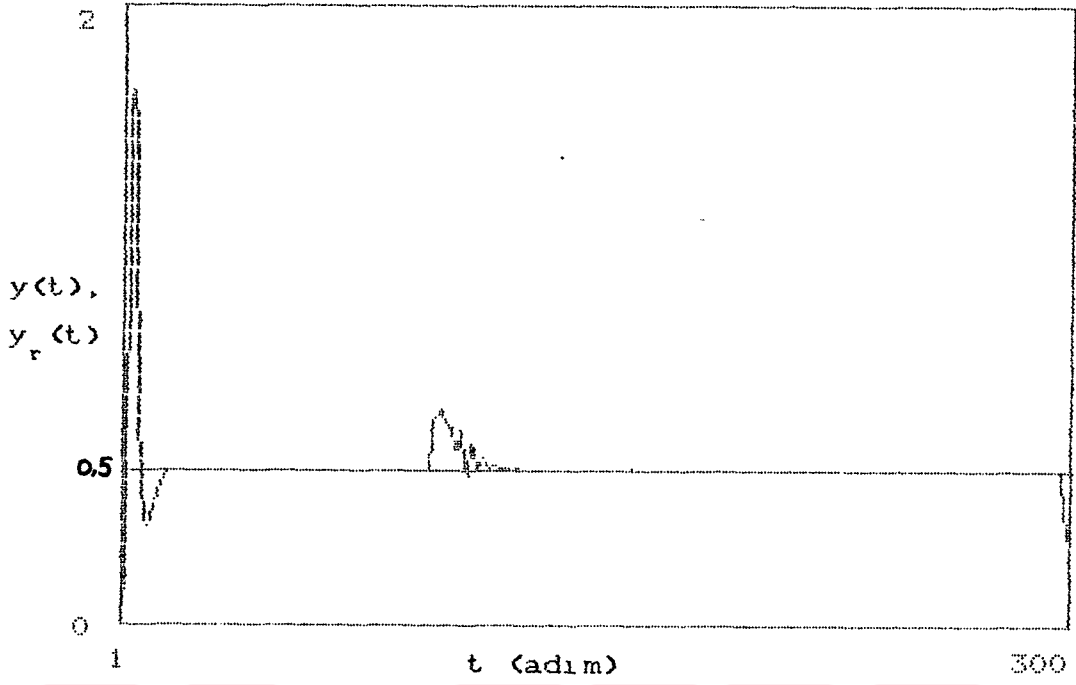
Sekil 5.5.a.3. b_0 parametresinin zamanla değişimi.



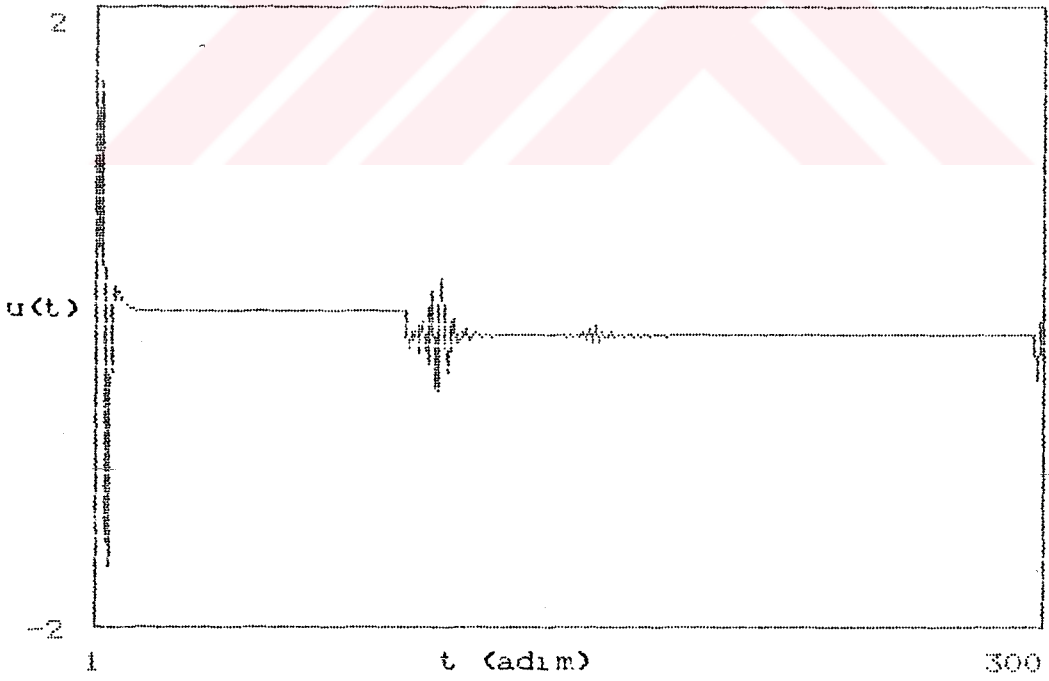
Sekil 5.5.b.1. Sistem 2 için $y(t)$ ve referans büyüklükleri, $(s(t)=0.1)$



Sekil 5.5.b.2. Sistem 2 için $u(t)$ kontrolü.



Şekil 5.5.c.1. Sistem 3 için $y(t)$ ve referans büyüklükleri, $(s(t)=0.1)$



Şekil 5.5.c.2. Sistem 3 için $u(t)$ kontrolü.

$$y(t) = 1.6y(t-1) + 0.8y(t-2) \\ = 1.5u(t-1) + 0.5u(t-2) + e(t) + 0.7e(t-1)$$

$N_2 = 5$ ve 10 alınarak iki ayrı simülasyon çalışması yapılmıştır. Her iki durum için de $N_u = 1$ ve $\lambda = 0.00001$ alınmıştır. Parametre kestirme algoritmasında kullanılan parametreler deterministik sistemlerdeki çalışma ile aynı seçilmiştir. Referans büyüklükleri de yine bir önceki çalışmadaki gibi seçilmiştir. Stokastik bozucu $e(t)$, ortalaması 0 ve standart sapması 10^{-4} olan rasgele sayılar ile karakterize edilmiştir. Sonuçlar şekil 5.6' de verilmiştir.

5.1.2.3 Sürekli Zamanlı Sistemler - Hatalı Model Mertebesi

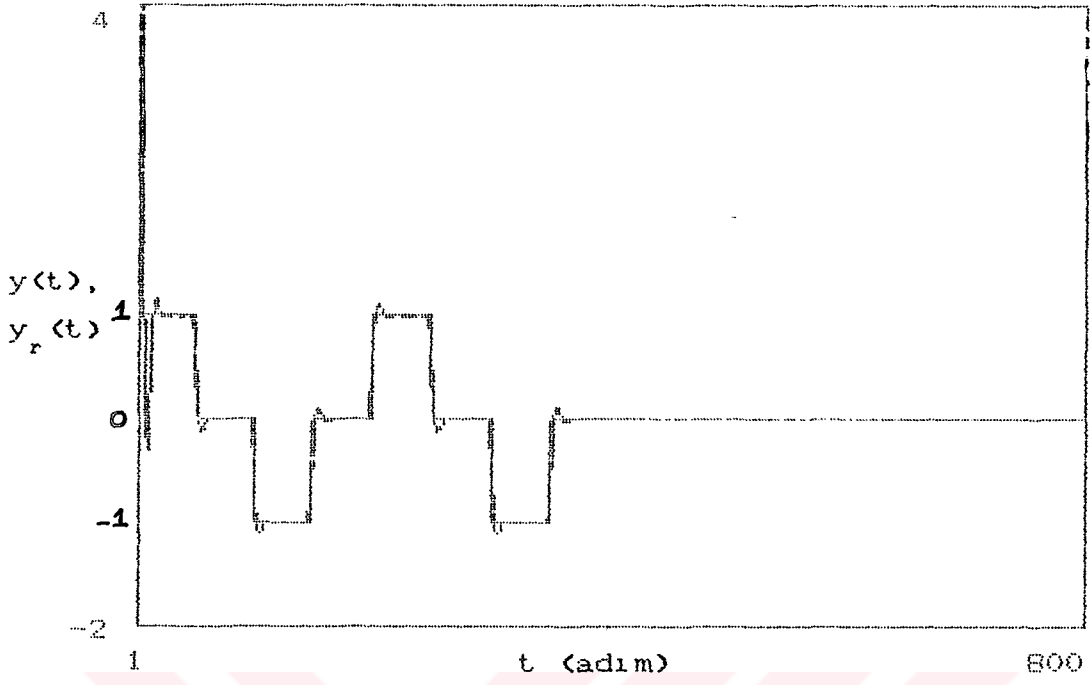
Bu çalışmada ise, sürekli zamanlı bir sistemin model mertebesinin gerçek mertebesinden farklı seçilmesi durumundaki kapalı çevrim davranışı incelenmiştir.

Simülasyonu yapılan sürekli zamanlı sistem aşağıdaki gibi seçilmiştir.

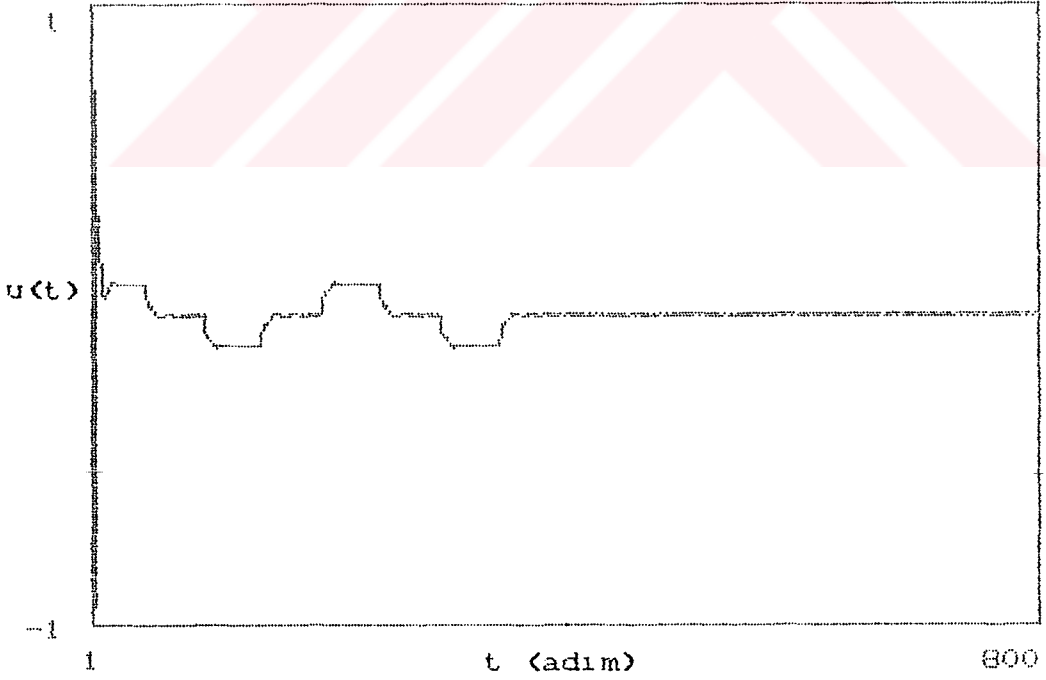
$$\ddot{y}(t) + 3\ddot{y}(t) + 3\dot{y}(t) + y(t) = u(t)$$

Diferansiyel denklem dördüncü mertebeden Runge-Kutta integrasyon yöntemi kullanılarak integre edilmiştir. Integrasyon adımı 0.1 sn., kontrolörün örnekleme periyodu ise 0.5 sn. olarak seçilmiştir.

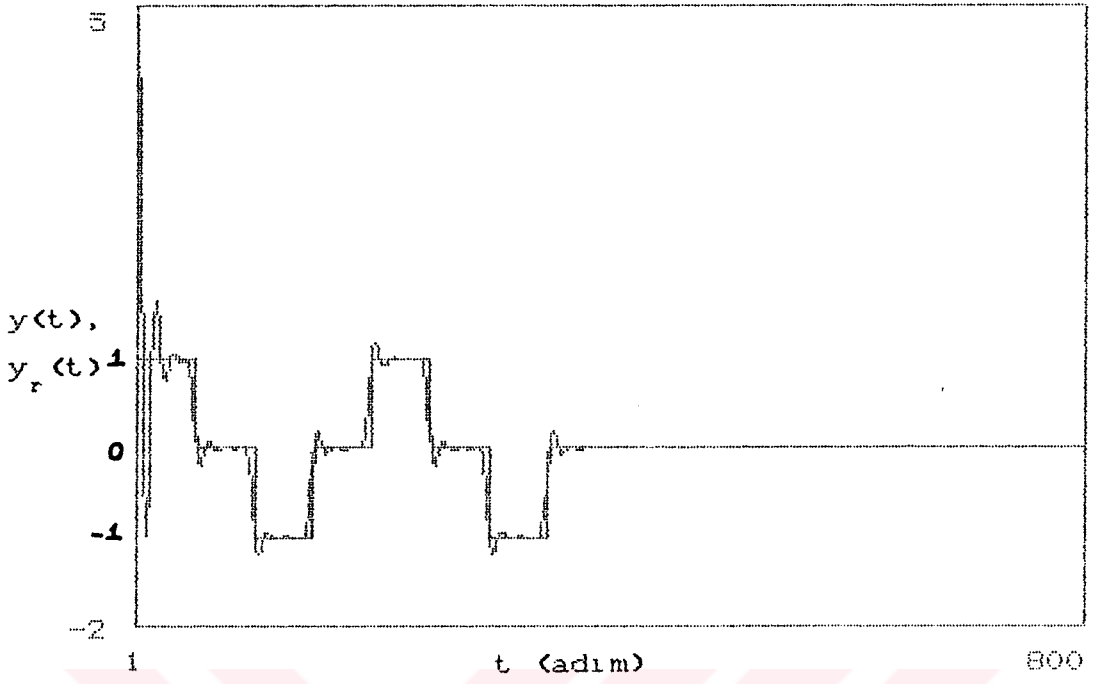
Ayar parametreleri $N_2 = 5$, $N_u = 1$ ve $\lambda = 0.000001$ olarak alınmıştır. Simülasyon toplam 500 adım devam etmekte ve 50 sn.'ye karşılık gelmektedir. Simülasyon boyunca referans büyüklüğü 1 olarak sabit kabul edilmiştir.



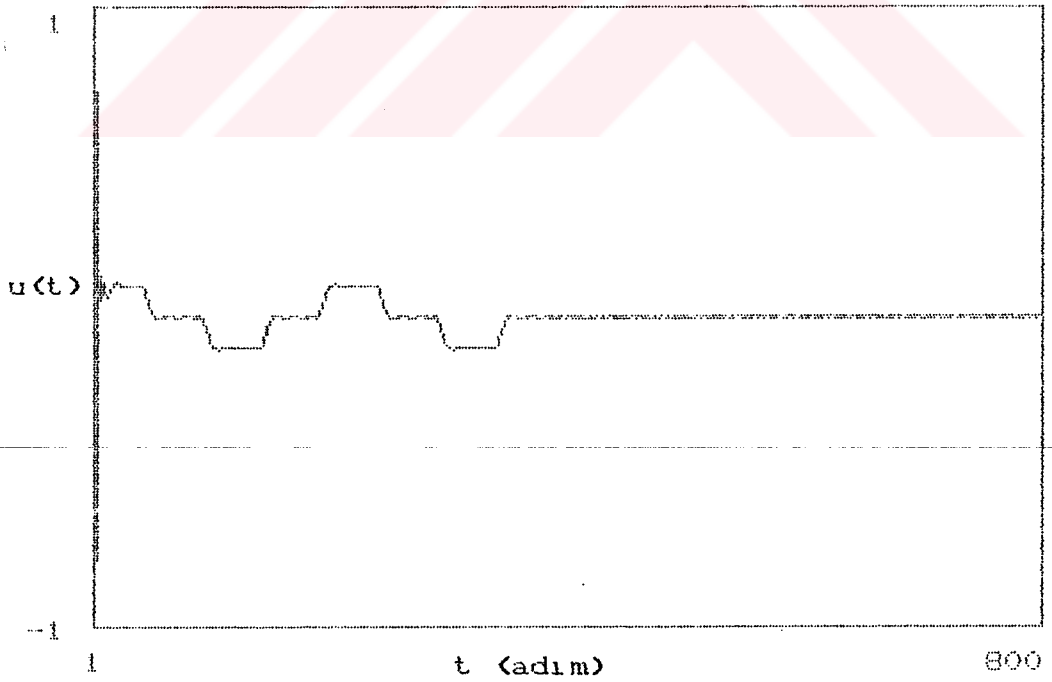
Sekil 5.6.a.1. Ayrık zamanlı stokastik sistem ($N_2=10$)
($y(t)$ ve referans değerleri).



Sekil 5.6.a.2. $u(t)$ kontrolünün değişimi



Şekil 5.6.b.1. Ayrık zamanlı stokastik sistem ($N_2=5$)
($y(t)$ ve referans değerleri)



Şekil 5.6.b.2. $u(t)$ kontrolünün değişimi

Yukarıdaki sürekli zamanlı sisteme karşılık gelen ayrık zamanlı gerçek model yapısı $n_a = 3$, $n_b = 2$ ve $d = 1$ değerleri ile sağlanmaktadır [36]. Bu bölümde iki ayrı simülasyon çalışması yapılmıştır. İlkinde, yukarıda verilen model mertebesi değerleri ile , ikincisinde ise $n_a = 1$, $n_b = 1$ ve $d = 1$ olarak seçilen hatalı ayrık zamanlı model mertebesi değerleri ile simülasyon gerçekleştirilmiştir. Her iki çalışmaya ait sonuçlar sırası ile şekil 5.7 ve 5.8' de gösterilmiştir.

5.1.2.4.Sürekli Zamanlı ve Zamanla Değişen Sistemler

Amaç, zamanla değişen sistem parametreleri karşısında kontrol algoritmasının sürekliliğini göstermektir. Kullanılan sürekli zamanlı, sistem aşağıdaki şekilde seçilmiştir [36].

$$T \ddot{y}(t) + (1+T) \dot{y}(t) + y(t) = K u(t)$$

Diferansiyel denklem dördüncü mertebeden Runge-Kutta integrasyon yöntemi kullanılarak integre edilmiştir. Integrasyon adımı 0.1 sn. ve örnekleme periyodu 0.5 sn. olarak seçilmiştir. Simülasyon, toplam 50 sn. (500 örnekleme adımı) sürmektedir.

Üç farklı simülasyon çalışması yapılmıştır.

a)- Zaman sabiti T lineer bir şekilde 5 sn.' den 10 sn.' ye arttırılmakta ve statik kazanç $K = 1$ olarak sabit tutulmaktadır.

b)- Statik kazanç K lineer bir şekilde 1' den 0.5 değerine kadar değiştirilmekte ve zaman sabiti T ise 1 olarak sabit tutulmaktadır.

c)- Hem zaman sabiti T ve hem de statik kazanç K , a

ve b şıklarındaki değerlere göre aynı anda değiştirilmektedir.

Her uç çalışmada, kontrol büyüklüğü $-10 < u(t) < +10$ şeklinde sınırlandırılmıştır. Ayar parametreleri $N_2 = 5$, $N_u = 1$ ve $\lambda = 0.000001$ olarak seçilmiştir. Sürekli zamanlı sisteme karşılık gelen ayrık zamanlı model mertebeleri $n_a=1$ ve $n_b=2$ olarak alınmıştır. Ayrıca, hatalı model mertebesi durumu da göz önüne alınarak $n_a=1$ ve $n_b=1$ seçilmiş ve her iki büyüklüğün (T ve K) değişmesi hali için ayrı bir simülasyon çalışması daha yapılmıştır. Sonuçlar şekil 5.9, 5.10 ve 5.11' de verilmiştir.

5.2, MIMO Simülasyon Çalışmaları

Tek giriş-tek çıkışlı (SISO) sistemler için geniş bir şekilde incelenen genelleştirilmiş öngörülü kontrol algoritmasının çok giriş-çok çıkışlı (MIMO) sistemler için de performansını araştırmak üzere aşağıdaki iki farklı sistem için simülasyon çalışmaları yapılmıştır.

Örnek 1

Simülasyon çalışmasında aşağıdaki çok giriş-çok çıkışlı ayrık zamanlı model kullanılmıştır [40].

$$y(t) + A_1 y(t-1) = B_0 u(t-1) + e(t)$$

Burada;

$$A_1 = \begin{bmatrix} -0.99101 & 8.80512 \times 10^{-3} \\ -0.80610 & -0.77089 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0.89889 & -4.59329 \times 10^{-3} \\ 19.390 & 0.88052 \end{bmatrix}$$

$e(t)$; ortalaması sıfır ve standart sapması 0.1 olan rasgele bozucudur.

Ayar parametreleri için yapılan değişik simülasyon çalışmaları neticesinde bulunan $N_2 = 3$, $N_U = 1$ ve $\lambda = 0.000001$ optimum değerleri kullanılmıştır. Parametre kestirme algoritmasında (RLS), $\mu = 0.95$, $P(0) = 10^5 I$, başlangıç parametreleri ise $B_0 = [I]$, diğerleri ise $[0]$ olarak alınmıştır. Simülasyon toplam 1000 örnekleme adımı devam etmektedir. Sonuçlar şekil 5.12' de görülmektedir.

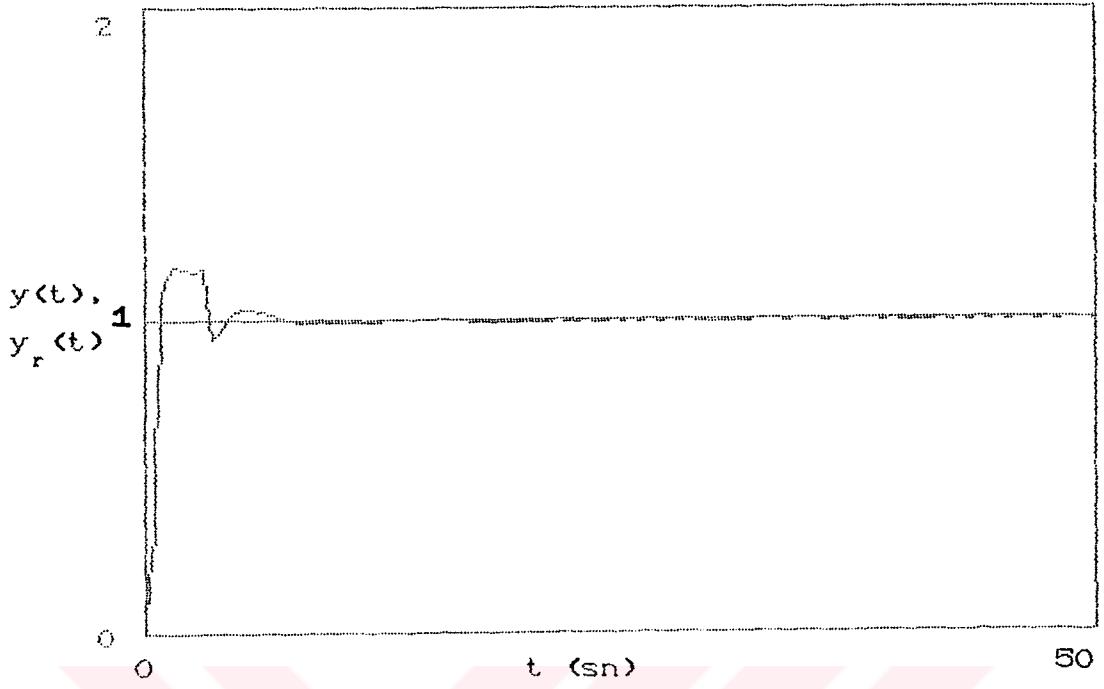
Örnek 2

Bu çalışmada ise non-minimum faz bir sistem kullanılmıştır [84].

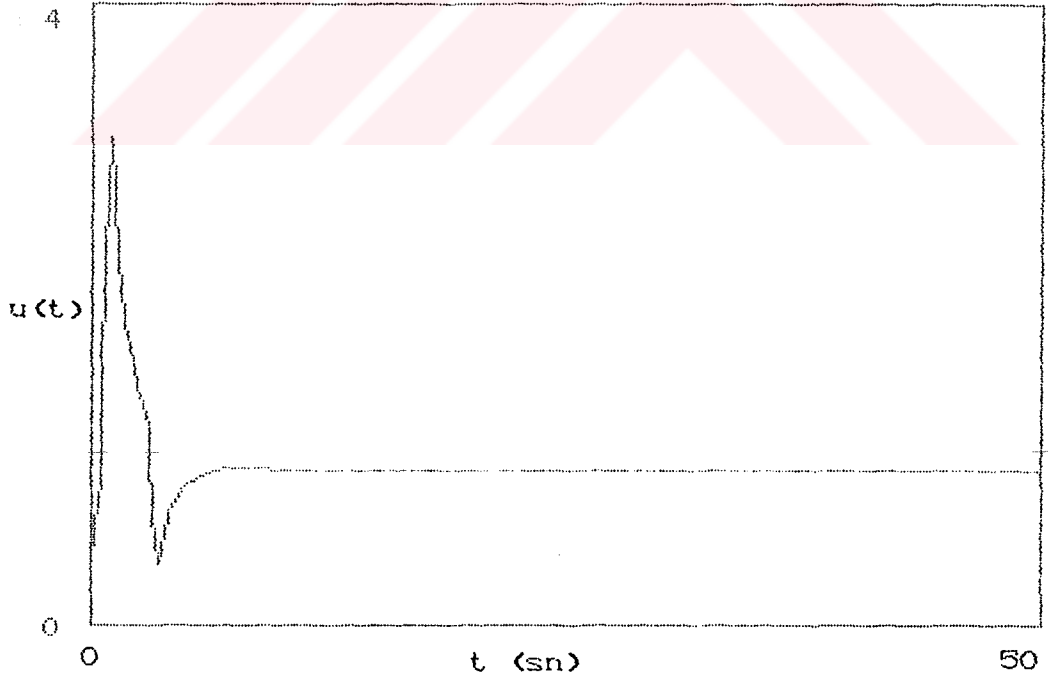
$$y(t) = A_1 y(t-1) + B_0 u(t-1) + B_1 u(t-2) + e(t)$$

Burada;

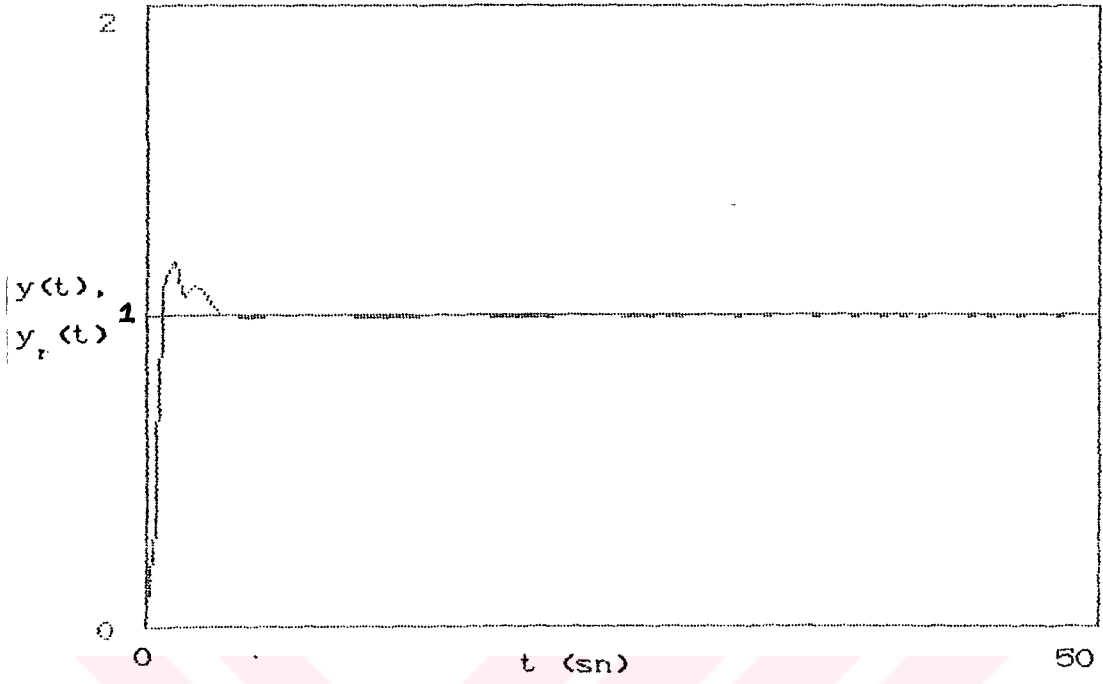
$$A_1 = \begin{bmatrix} 0.9 & -0.5 \\ -0.5 & 0.2 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0.2 & 1 \\ 0.25 & 0.2 \end{bmatrix}$$



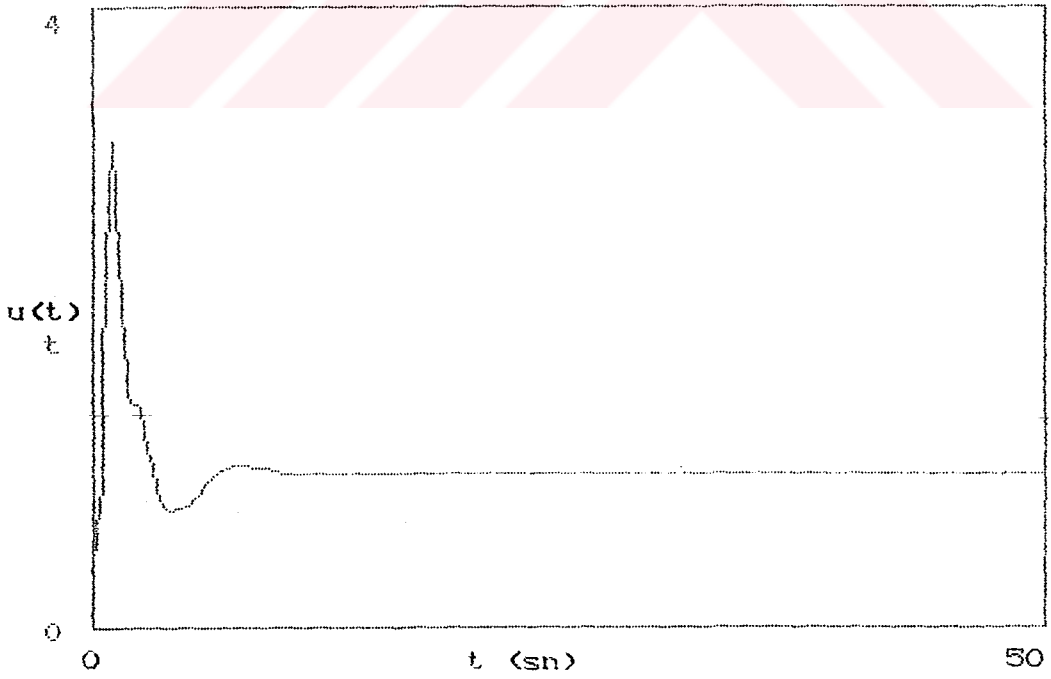
Şekil 5.7.1. Sürekli zamanlı sistem, ($y(t)$ ve referans büyüklükleri)



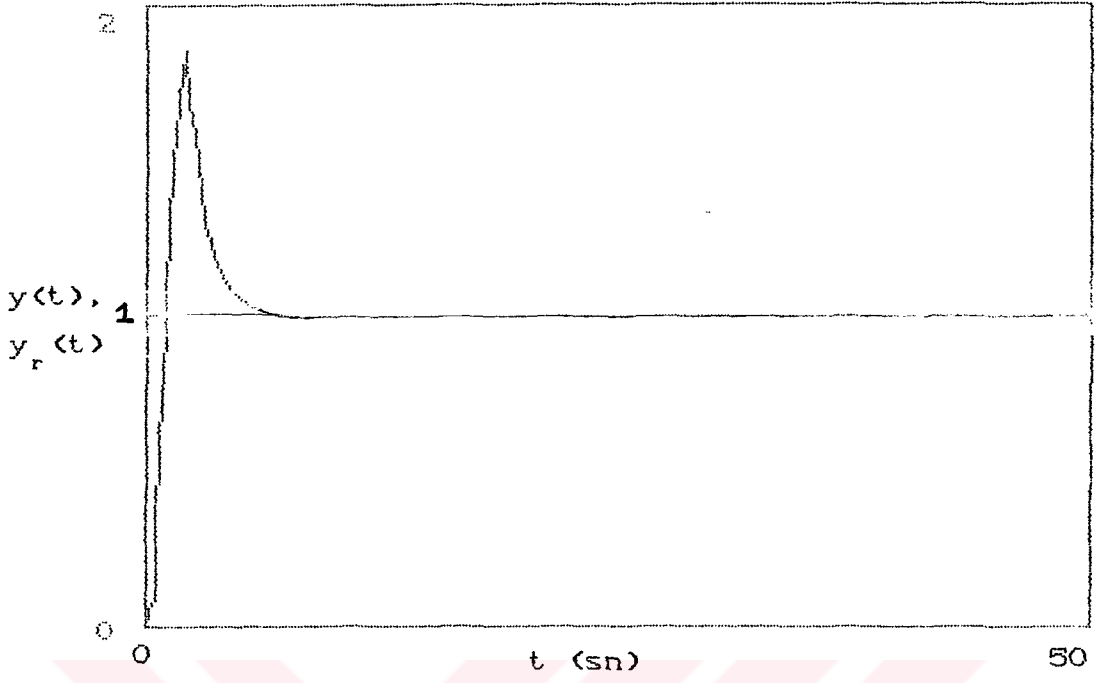
Şekil 5.7.2. $u(t)$ kontrolünün değişimi.



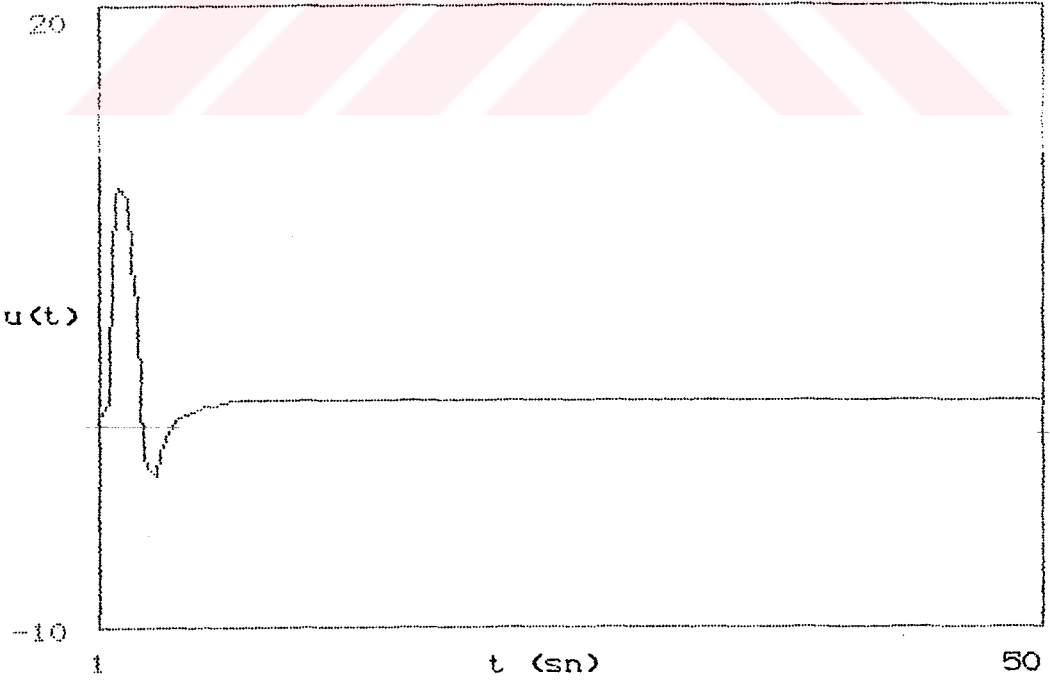
Şekil 5.8.1. Sürekli zamanlı sistem, $y(t)$ ve referans büyüklükleri.
(Hatalı model mertebesi)



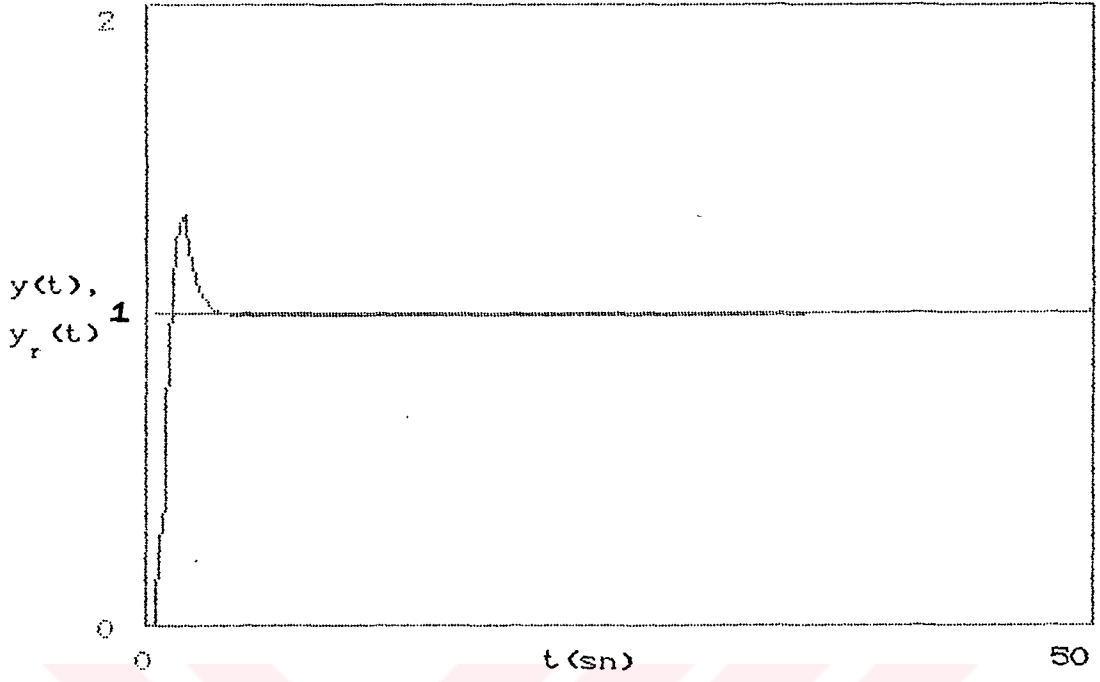
Şekil 5.8.2. $u(t)$ kontrolünün değişimi



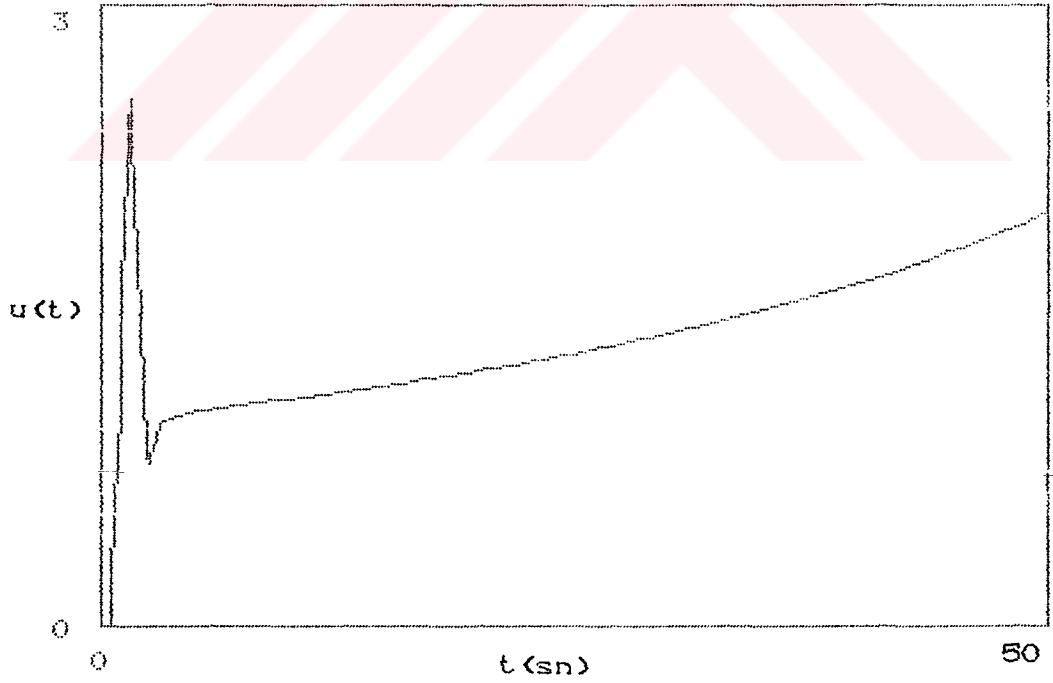
Sekil 5.9.1. Sürekli zamanlı sistem, $y(t)$ ve referans büyüklükleri. (T değişken)



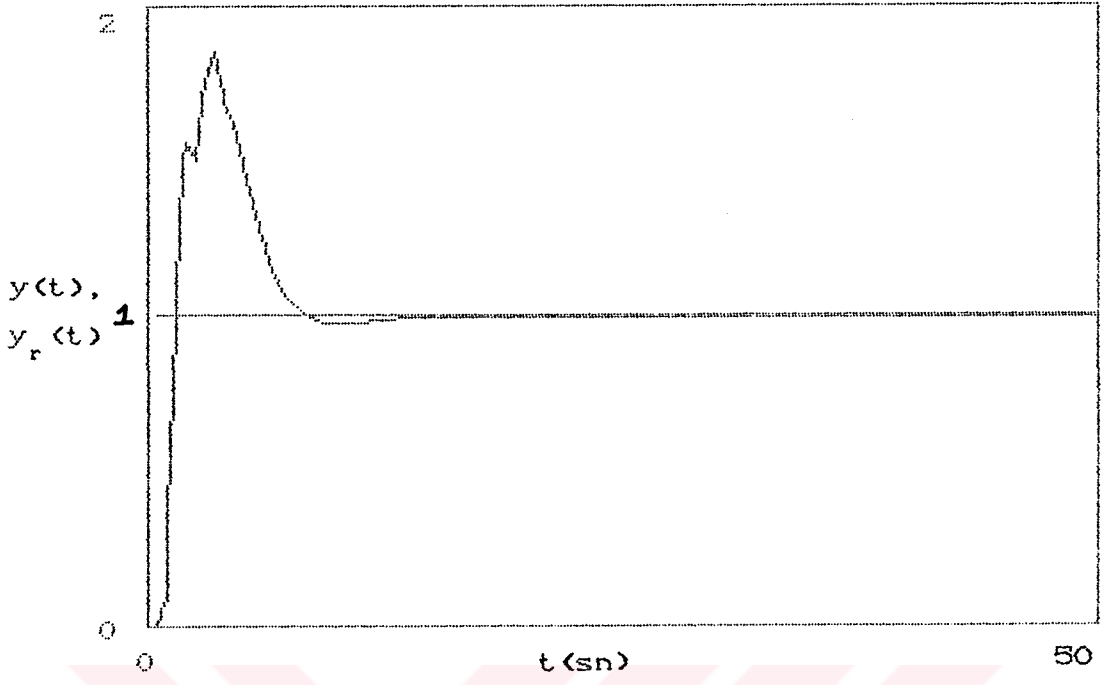
Sekil 5.9.2. $u(t)$ kontrolünün değişimi



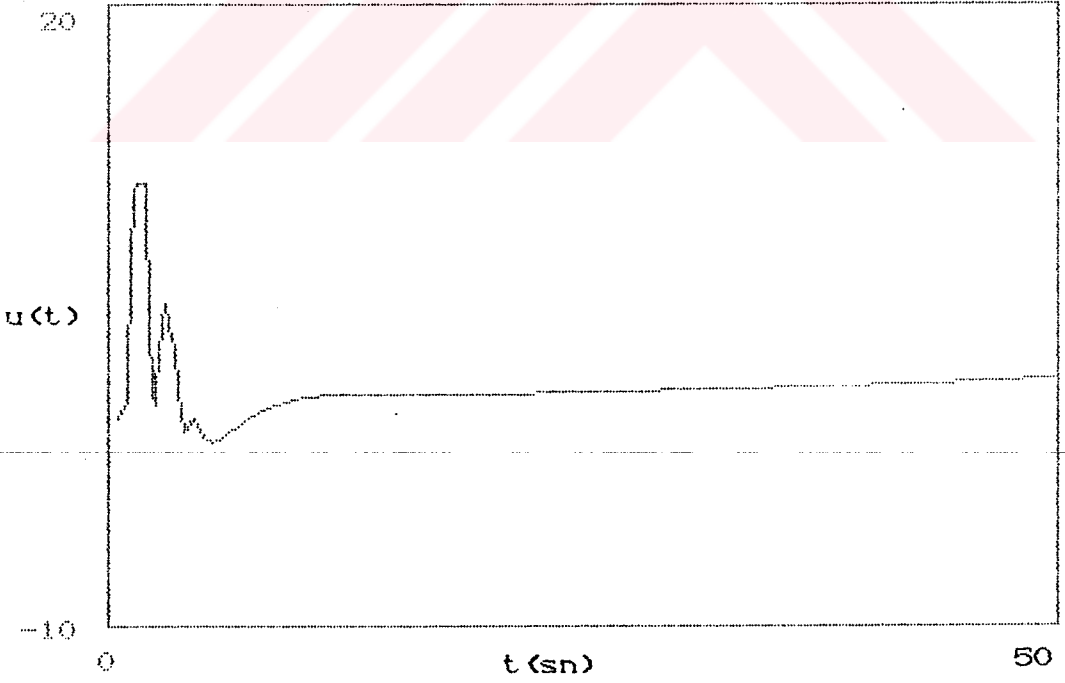
Sekil 5.10.1. Sürekli zamanlı sistem, $y(t)$ ve referans büyüklükler (K değişken)



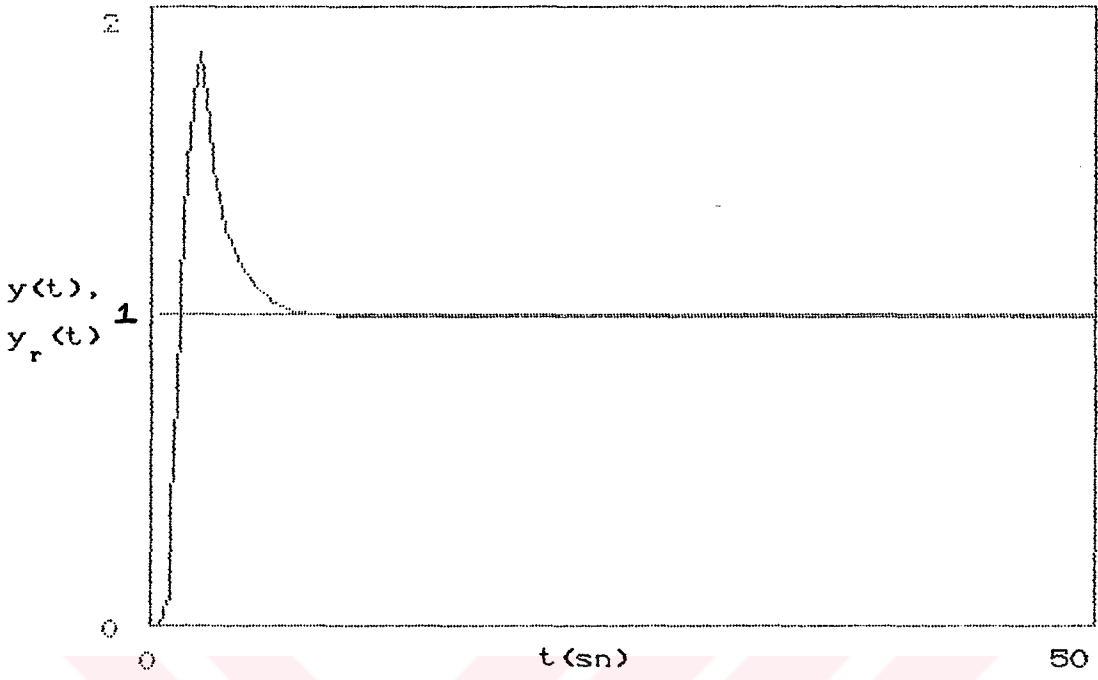
Sekil 5.10.2. $u(t)$ kontrolünün değişimi



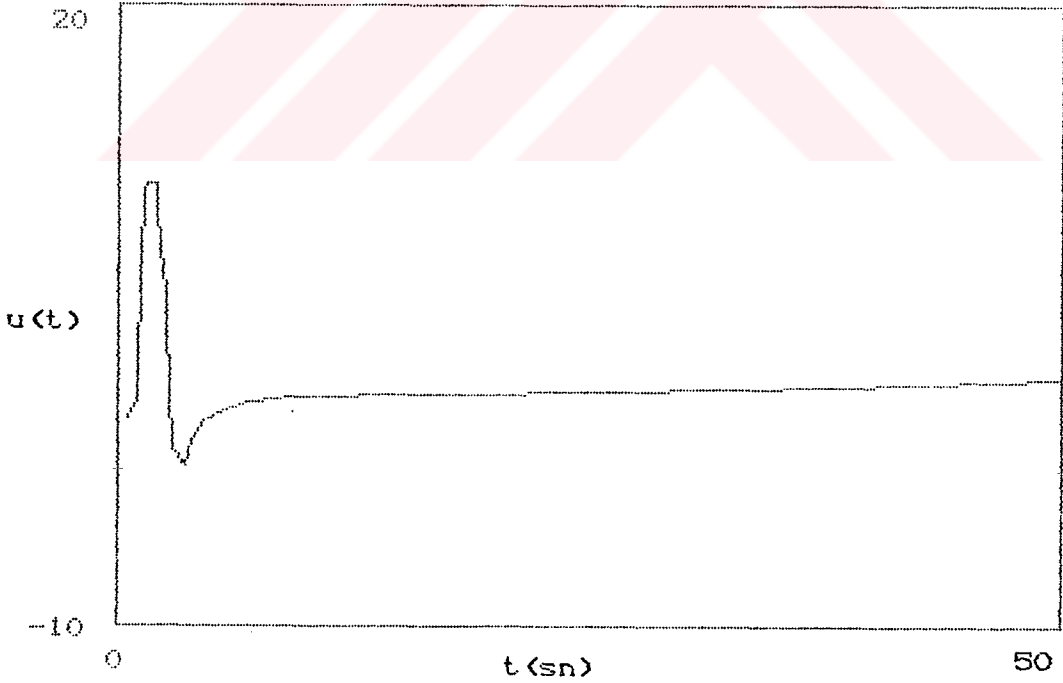
Şekil 5.11.a.1. Sürekli zamanlı sistem, $y(t)$ ve referans büyüklükleri.
(K ve T değişken), $n_a=1$, $n_b=2$



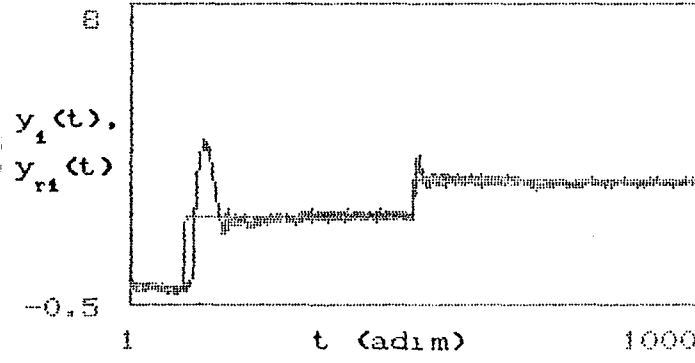
Şekil 5.11.a.2. $u(t)$ kontrolünün değişimi



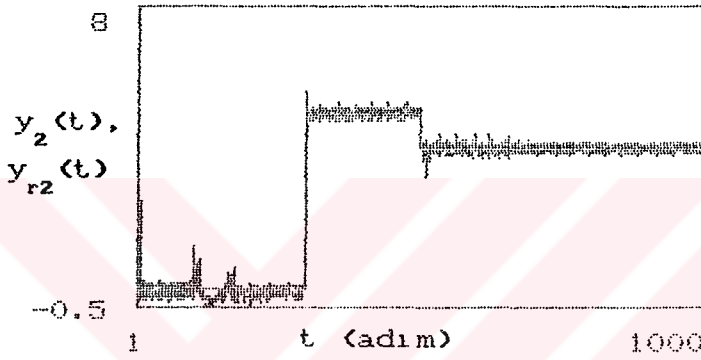
Şekil 5.11.b.1. Sürekli zamanlı sistem, $y(t)$ ve referans büyüklükleri.
(K ve T değişken), $n_a=1$, $n_b=1$



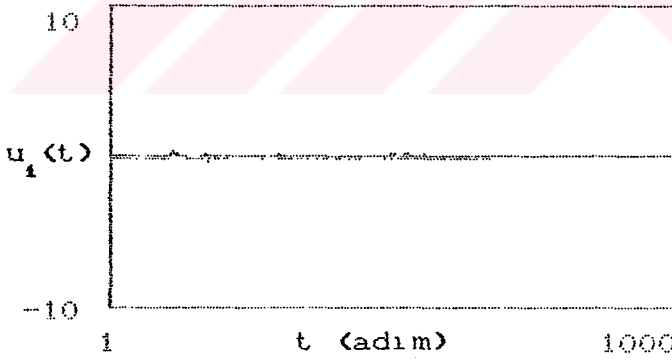
Şekil 5.11.b.2. $u(t)$ kontrolünün değişimi



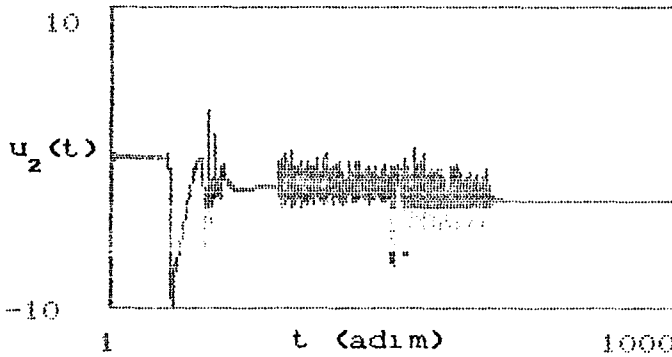
Şekil 5.12.1. $y_1(t)$ ve referans değerleri



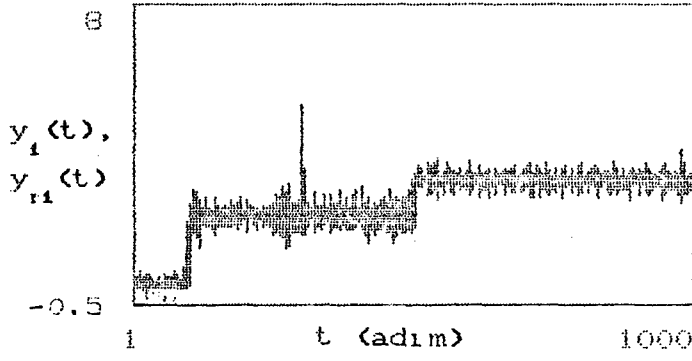
Şekil 5.12.2. $y_2(t)$ ve referans değerleri



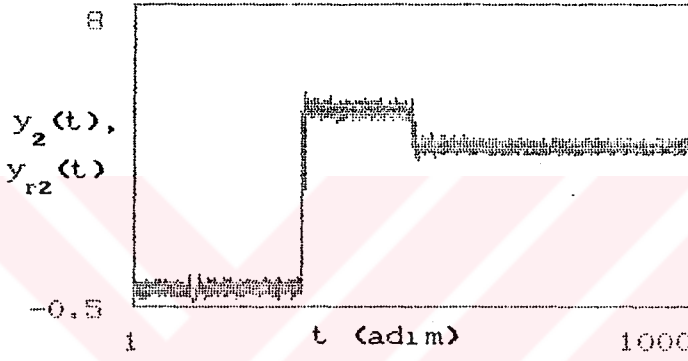
Şekil 5.12.3. $u_1(t)$ kontrolünün değişimi.



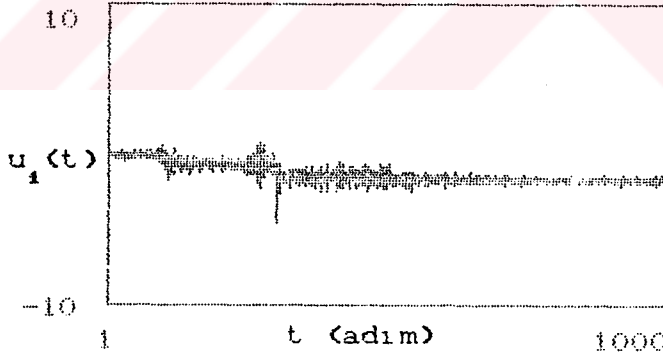
Şekil 5.12.4. $u_2(t)$ kontrolünün değişimi.



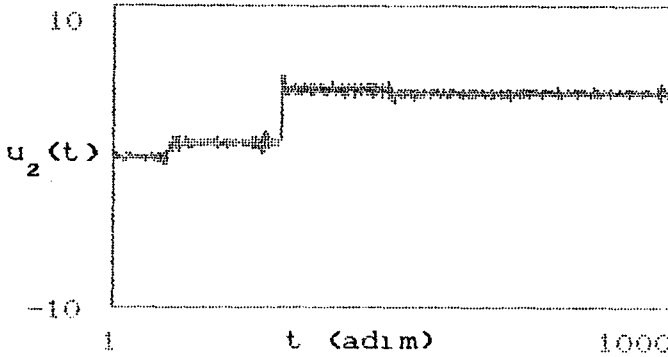
Şekil 5.13.1. $y_1(t)$ ve referans değerleri



Şekil 5.13.2. $y_2(t)$ ve referans değerleri



Şekil 5.13.3. $u_1(t)$ kontrolünün değişimi.



Şekil 5.13.4. $u_2(t)$ kontrolünün değişimi.

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$e(t)$; ortalaması sıfır, standart sapması 0.1 olan rasgele bozucuyu simgelemektedir.

Parametre kestirme algoritmasında (RLS), örnek 1' deki değerler kullanılmıştır. Referans değerleri ve ayar parametreleri yine örnek 1' deki gibi alınmıştır. Sonuçlar şekil 5.13' de gösterilmiştir.

5.3. SONUÇLAR

5.3.1. SISO Genelleştirilmiş Öngörülü Kontrol Algoritmasında Ayar Parametrelerinin Etkisi

Ayar parametrelerinin etkisini inceleyebilmek için nonminimum-faz birinci mertebeden bir sistem ele alınmıştır. Öngörü ufuğu (N_2)' nun kontrol performansı üzerine etkileri şekil 5.1'de verilmiştir. Bu durumda N_u ve λ değerleri sabit kabul edilmiştir. $N_2 = 1$ ve $N_u = 1$ için nonminimum-faz sıfırlarının iptalinden dolayı kontrolör kararsız olmaktadır. Bu durum şekil 5.1.a, b ve c' de görülmektedir. Şekil 5.1.c ve 5.1.d' den de görüleceği üzere öngörü ufuğu N_2 ' nin değeri arttırıldıkça daha yavaş bir kontrol sağlanmaktadır. Çıkış büyüklüğünün referans değerine erişmesi daha uzun zaman almaktadır. Fakat daha düzgün bir kontrol sağlanması mümkün olmaktadır. N_2 ' nin değeri düştükçe kapalı çevrimin hızı artmaktadır. Dolayısı ile daha ani bir kontrol sağlanmaktadır. Örneğin verilen nonminimum-faz sistem için $N_2 = 4$ değeri, hız ve düzgünlüğü ortakça

sağlayabilmesi açısından önerilebilir.

Kontrol ufuğunun etkileri yine aynı sistem üzerinde incelenmiştir. Şekil 5.2' den de görüldüğü gibi sabit N_2 ve λ değerleri için $N_u = 1$ durumunda yavaş fakat düzgün bir cevap elde edilmiştir. N_u değeri arttıkça çıkış büyüklüğünde sapmalar oluşmakta, fakat cevap hızı artmaktadır. Buradan da özellikle basit sistemler için $N_u = 1$ seçimi hem kontrolün düzgün olması hem de hesaplamaların minimize edilecek olması açısından tavsiye edilmektedir.

Kontrol ağırlığı λ' nin verilen nonminimum-faz sistem üzerindeki etkisi de şekil 5.3' de görülmektedir. $\lambda = 0$ için (kontrol üzerine hiç bir ağırlık uygulanmıyor) bile uygun bir cevap elde edilmektedir. λ' ya 0.1' den başlayarak gittikçe küçülecek şekilde değerler verildiğinde özellikle $\lambda = 0.000001$ için hızlı bir cevap elde edildiği, fakat salınımlar ortaya çıktığı gözlenmektedir. Sonuçlar $\lambda = 0$ 'ın bile nonminimum-faz sistemlerle kullanılabileceğini göstermektedir.

5.3.2. Farklı Sistemler Karşısında Genelleştirilmiş Öngörülü Kontrol Algoritması

5.3.2.1. Ayırık Zamanlı Deterministik Sistemler

Seçilen üç farklı ayırık zamanlı sistem için yapılan simülasyon çalışmaları sonucunda bozucu bulunmaması durumunda verilen farklı referans değerleri için gayet iyi yakınsamalar elde edildiği gözlenmiştir. Çok küçük bir öğrenme adımından sonra çıkış büyüklüğü referans değerine yakınsamaktadır. N_2 ' nin değeri küçültülerek daha da hızlı cevaplar elde etmek mümkün olmaktadır. Kontrol büyüklükleri de referans değişimleri doğrultu-

sunda kararlı bir davranış göstermektedir.

Deterministik olarak 0.1 değerinde bir yük bozucusu ilave edildiği durumda da her üç sistem için çok kısa bir zamanda sabit referans değerine erişme sağlanmaktadır. Kontrolu zor olan nonminimum-faz sistemler için de çok başarılı sonuçlar alınması, genelleştirilmiş öngörülü kontrol algoritmasının performansını göstermek açısından önemli olmaktadır (Şekil 5.5.a, b, c).

5.3.2.2. Ayrık Zamanlı Stokastik Sistemler

Ayrık zamanlı stokastik sistemler için, genelleştirilmiş öngörülü kontrol algoritmasının performansını araştırmak amacı ile seçilen sistem üzerine ortalaması sıfır ve standart sapması 10^{-4} olan rasgele bozucular uygulanmıştır. Farklı referans değerleri ve $N_2 = 10$ ve 5 olarak seçilen ayar parametreleri için ayrı ayrı yapılan simülasyon çalışmaları, bozuculara rağmen sistem cevabının referans değerlerini çok iyi takip edebildiğini göstermektedir. Kontrol büyüklükleri $u(t)$ için de gayet kararlı bir değişim gözlenmiştir. Sonuçlar şekil 5.6' da verilmiştir.

5.3.2.3. Sürekli Zamanlı Sistemler-Hatalı Model Mertebesi

Bu çalışmada sürekli zamanlı bir sistem incelenmiş ve sistem model mertebesinin gerçek mertebesinden farklı seçilmesi durumunda öngörülü kontrol algoritmasının performansı araştırılmıştır. $n_a = 3$ ve $n_b = 2$ olan gerçek model mertebesi yerine $n_a = 1$ ve $n_b = 1$ alınarak yapılan simülasyon sonuçlarının gerçek model mertebesi ile yapılan simülasyon sonuçlarına çok yakın değerler

vermesi, önerilen kontrol algoritmasının hatalı model mertebesi tanımlamalarına karşı duyarsız olduğunu göstermektedir. Sonuçlar şekil 5.7 ve 5.8' de verilmiştir.

5.3.2.4, Sürekli Zamanlı ve Zamanla Değişen Sistemler

Zamanla değişen sistem parametreleri karşısında kontrol algoritmasının performansını göstermek üzere iki farklı parametrenin (K ve T) tüm simülasyon süresi boyunca ayrı ayrı veya birlikte değiştiği düşünülmüştür. Şekil 5.9, 5.10 ve 5.11' den de görüldüğü gibi bu değişimler karşısında sistem cevabı, kısa bir öğrenme devresinden sonra referans değerini yakalamakta ve simülasyon boyunca herhangi bir salınım görülmemektedir. Bu da genelleştirilmiş öngörülü kontrol algoritmasının parametrelerdeki zamanla değişmelere karşı duyarlı olmadığını göstermektedir.

5.3.3, MIMO Sistemler İçin Simülasyon Sonuçları

Genelleştirilmiş öngörülü kontrol algoritmasının performansının MIMO sistemler üzerinde incelenmesi için minimum-faz ve nonminimum-faz iki ayrı sistem ele alınmıştır. Her iki sisteme de ortalaması sıfır ve standart sapması 0.1 olan rasgele bozucuların etki ettiği varsayılmıştır. Belirli adımlarda değişen referans büyüklükleri ile yapılan simülasyon sonuçları (Şekil 5.12 ve 5.13) her iki sistem cevabının referans değişimleri ve bozuculara rağmen referans değerlerini gayet yakından takip ettiğini göstermektedir. Özellikle nonminimum-faz sistem için elde edilen memnuniyet verici sonuçlar genelleştirilmiş öngörülü kontrol algoritmasının üstünlüğünü bir kez daha kanıtlamaktadır.

BÖLÜM 6. ROBOT KOLU KİNEMATİĞİ VE DİNAMİĞİ

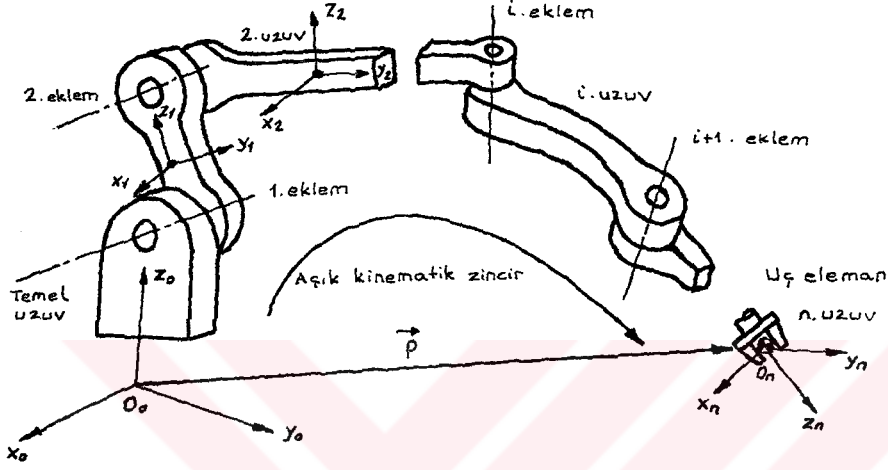
6.1. Robot Kolu Kinematığı

Bir robot kolu veya mekanik manipülatör, dönel veya kayar eklemlerle birbirine bağlanmış "uzuv" adı verilen rijid cisimlerden meydana gelen açık çevrimli bir zincirdir. Zincirin bir ucu bir desteğe (temele) bağlanmıştır, diğer ucu ise serbesttir. Eklemler birbirine bağlı iki cismin izafi hareketine izin vermektedirler.

Bu bölümde mekanik bir manipülatörün kinematik analizi yapılacaktır. Her bir eklem-uzuv çifti bir serbestlik derecesi oluşturmaktadır. n serbestlik dereceli bir manipülatör n adet uzuv ve n adet eklemden meydana gelmektedir. Eklemler ve uzuvlar temelden başlayarak dışarıya doğru numaralanırlar. Manipülatörün bağlı olduğu temel (0) uzvu, manipülatörün tutucusu (en uç uzuv) (n) uzvu olarak adlandırılmaktadır. (i) eklemi, (i-1) uzvu ile (i) uzvunu birleştirmektedir.

Kinematik problem, genel olarak iki alt problemden meydana gelmektedir: "Direkt ve Ters Kinematik Problem" Direkt kinematik problem, bir referans koordinat sistemine göre manipülatörün tutucusunun pozisyon ve oryantasyonunu bulmaktır. Manipülatörün eklem değişkenleri vektörü $q = (q_1, q_2, \dots, q_n)^T$ ve gerekli geometrik uzuv parametreleri bilinmektedir. Ters kinematik problem ise istenen oryantasyon ve pozisyonundaki bir manipülatör tutucusunun, bu oryantasyon ve pozisyonu sağlayan eklem değişkenleri vektörü (q)'yu bulmaktır. Referans koordinat sistemine göre tutucunun pozisyonu, oryantasyonu

ve gerekli geometrik uzuv parametrelerinin bilindiği varsayılmaktadır.



Şekil 6.1. Açık Kinematik Zincir

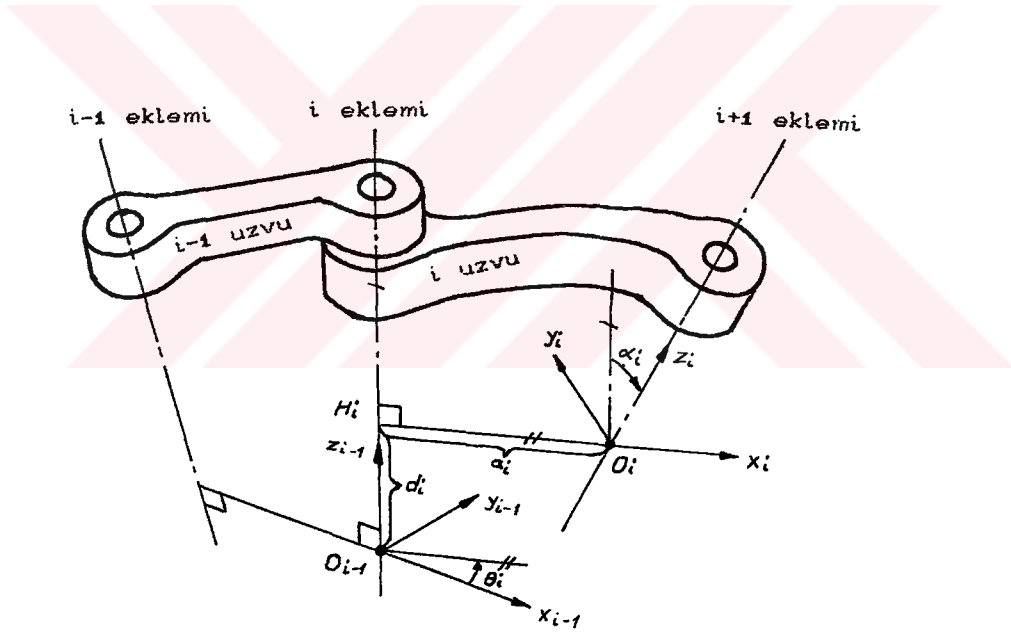
Bir manipülatörün kinematik ve daha sonra anlatılacak olan dinamik denklemlerini sistematik ve genelleştirilmiş bir şekilde çıkarabilmek için, her bir (i) uzvuna cisim koordinat sistemi $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ yerleştirilmiştir. Komşu koordinat çerçeveleri arasında Denavit ve Hartenberg [105] tarafından geliştirilen 4 parametre yardımı ile ilişki kurulmaktadır. Bu parametreler aşağıdaki şekilde tanımlanmaktadır (Şekil 6.2):

θ_i : $\vec{z}_{i-1}$ eksenini etrafında $\vec{x}_{i-1}$ ekseninden $\vec{x}_i$ eksenine kadar olan ekleml açısı (sağ el kuralı kullanılarak),

d_i : $\vec{z}_{i-1}$ eksenini boyunca (i-1). koordinat çerçevesinin orjininden $\vec{x}_i$ eksenini ile $\vec{z}_{i-1}$ ekseninin kesişme yerine kadar olan mesafe,

a_i : $\vec{x}_i$ eksenini boyunca $\vec{x}_i$ eksenini ile $\vec{z}_{i-1}$ ekseninin kesişme noktasından (i). koordinat çerçevesinin orjinine kadar olan mesafe (veya $\vec{z}_{i-1}$ ve $\vec{z}_i$ eksenleri arasındaki en kısa mesafe).

α_i : $\vec{x}_i$ eksenini etrafında $\vec{z}_{i-1}$ ekseninden $\vec{z}_i$ eksenine kadar olan açı (sağ el kuralı kullanılarak).



Şekil 6.2 Denavit-Hartenberg Notasyonu

Dönel bir eklem için d_i , a_i ve α_i eklem parametreleridir ve sabit kalmaktadır, θ_i eklem değişkenidir ve (i) uzvu (i-1) uzvuna göre döndüğünde değişmektedir. Kayar (prizmatik) bir eklem için θ_i , a_i ve α_i eklem parametreleridir ve sabit kalmaktadır, d_i ise eklem

değişkenidir. Bu parametreler, bir manipülatörün her bir uzvunun komple kinematik konfigürasyonunu oluşturmak için gerekli olan minimal sayıdaki parametre setini oluşturmaktadır.

Her bir koordinat çerçevesi aşağıdaki üç temel kurala göre yerleştirilmektedir [106]:

- 1- $\vec{z}_{i-1}$ eksenini (i). eklemin hareket eksenini boyunca yerleştirilir,
- 2- $\vec{x}_i$ eksenini $\vec{z}_{i-1}$ eksenine diktir (ucu dışarıya doğru olacak şekilde),
- 3- $\vec{y}_i$ eksenini sağ el koordinat sistemini tamamlayacak şekildedir.

Yukarıdaki kurallara uyularak, $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$ referans çerçevesi, $\vec{z}_0$ eksenini ilk eklemin hareket eksenini boyunca olacak şekilde temel uzuv üzerinde herhangi bir yere yerleştirilebilir. Son koordinat çerçevesi (n. çerçeve), $\vec{x}_n$ eksenini $\vec{z}_{n-1}$ eksenine dik olacak şekilde el üzerinde herhangi bir yere yerleştirilebilir.

(i) uzvunun koordinat sisteminde tanımlanan herhangi bir vektörü, (i-1) uzvunun koordinat sistemine taşımak için homojen dönüşüm matrisleri kullanılmaktadır. Denavit-Hartenberg parametrelerini kullanarak $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ koordinat sisteminde tanımlanan bir vektörü, $(\vec{x}_{i-1}, \vec{y}_{i-1}, \vec{z}_{i-1})$ koordinat sisteminde tanımlamaya yarayan homojen dönüşüm matrisi aşağıdaki şekilde verilmektedir.

$$A_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... (6.1)

A_{i-1}^i matrisinin tersi $([A_{i-1}^i]^{-1} = A_i^{i-1})$, herhangi bir vektörün $(i-1)$. çerçeveden (i) . çerçeveye dönüşümü için kullanılmaktadır.

Herhangi bir vektörün koordinatlarını (i) uzvunun koordinat sisteminden temel koordinat sistemine dönüştürecek homojen dönüşüm matrisi, dönüşüm matrislerinin ardışık olarak çarpılması ile elde edilir.

$$A_0^i = A_0^1 \dots A_{i-2}^{i-1} A_{i-1}^i = \prod_{j=1}^i A_{j-1}^j \dots (6.2)$$

A_{i-1}^i homojen dönüşüm matrisinin sol üst tarafında bulunan 3×3 'lük alt matris, rotasyon matrisini simgelemektedir. Rotasyon matrisi, bir vektörün koordinatlarını bir koordinat sisteminden diğerine orjinleri aynı kalacak fakat bir miktar dönecek şekilde dönüştürmektedir. (i) uzvunun koordinatlarına dönüşümü sağlayan rotasyon matrisi:

$$R_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \dots (6.3)$$

şeklinde verilmektedir.

(6.2) ile verilen dönüşüm ortonormal bir dönüşüm olduğundan rotasyon matrisinin tersi transpozese eşittir.

$$R_i^{i-1} = (R_{i-1}^i)^T \dots (6.4)$$

Homojen bir dönüşüm matrisi, geometrik olarak, referans bir çerçeveye göre dönmüş bir koordinat sisteminin yerini (pozisyon ve oryantasyon) göstermektedir. $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$ referans çerçevesi ve homojen dönüşüm matrisi A_0^i verilirse, rotasyon alt matrisinin sütun vektörleri

referans çerçeveye göre $\langle \vec{x}_i, \vec{y}_i, \vec{z}_i \rangle$ koordinat sisteminin asal eksenlerini göstermektedir. Homojen dönüşüm matrisinin 4. sütunu, referans çerçeveye göre $\langle i \rangle$. çerçevenin orjininin pozisyonunu göstermektedir.

$$A_o^i = \begin{bmatrix} \vec{x}_i & \vec{y}_i & \vec{z}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots (6.5)$$

(6.5) ifadesinde $\langle i \rangle$ yerine $\langle n \rangle$ konularak elde edilen matris, referans çerçeveye göre elin pozisyon ve oryantasyonunu tamamen belirleyen kol matrisi olarak adlandırılmaktadır.

$$H = A_o^n = \begin{bmatrix} \vec{x}_n & \vec{y}_n & \vec{z}_n & \vec{p}_n \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \dots (6.6)$$

Burada,

$\vec{n}$: elin birim normal vektörü,

$\vec{s}$: elin birim kayma vektörü,

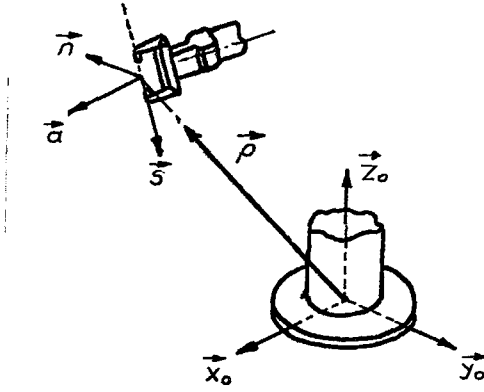
$\vec{a}$: elin birim yaklaşma vektörü,

$\vec{p}$: elin pozisyon vektörüdür.

Bu vektörlerin hepsi, şekil 6.3 'de görüldüğü gibi temel koordinatlara göre tanımlanmışlardır.

(6.2) ifadesi kullanılarak, verilen eklem koordinatları ile manipülatör tutucusunun kartezyen pozisyon ve oryantasyonunu elde etmek mümkündür. Bu işlem "direkt kinematik çözüm" olarak adlandırılmaktadır.

$$H = A_o^n = A_o^1 A_1^2 \dots A_{n-1}^n$$



Şekil 6.3 Elin pozisyon ve oryantasyon vektörleri.

Manipülatörün kontrolü ters kinematik çözüm gerektirebilir. Ters kinematik probleminin çözümü için elin istenen pozisyon ve oryantasyonunun verilip buna karşılık gelen eklem koordinatlarının bulunması gerekmektedir.

Düz kinematik için tek bir çözüm var iken, ters kinematik için birden fazla çözüm bulunabilir. Bu da ters kinematik çözüm problemini zorlaştırmaktadır.

6.2. Robot Kolu Dinamiği

6.2.1. Giriş

Manipülatör dizaynı ve kontrol için atılacak ilk adım, manipülatörün dinamik modelinin çıkarılmasıdır. Sadece iki veya üç serbestlik derecesine sahip manipüla-

törler için sistemin dinamik denklemleri el ile çıkarılabilir. Üçten daha fazla serbestlik derecesine sahip manipülatörler için dinamik denklemler, el ile çıkarmak çok zor olacak şekilde karmaşık olmaktadır. Bu yüzden dinamik denklemleri, otomatik olarak oluşturacak bir algoritma gerekmektedir. Bu algoritma, hesaplamalar el ile yapıldığı zaman oluşan hataları da yok edecektir.

Manipülatör modellemede etkili bir metod bulmak için aşağıdaki istekler göz önüne alınmalıdır.

1- Hareket denklemlerini çıkarmak için kullanılacak metod çok karmaşık olmamalı, kolayca formüle edilmelidir.

2- Model, gerçek sistem ile ilgili sonuçları doğru olarak vermelidir.

3- Sistem denklemleri, çevrim-ici kontrol ve hesapsal verimlilik açısından kısa zamanda çözülebilir olmalıdır. Bu, simülasyon maliyetini azaltmak için gereklidir.

4- Kullanılan metod, dinamiğin hem direkt, hem de indirekt problemlerini çözmelidir. Yani, mekanizma parçalarının hareketi verildiğinde gerekli torklar, torklar verildiğinde ise ivmeler hesaplanabilmelidir.

5- Kullanılan metod genel olmalıdır. Giriş olarak sadece sistem parametreleri verildiğinde, sistemin çalışması için gerekli tüm bilgiler elde edilebilmelidir.

6- Dinamik model, sistemin tüm kısıtlarını göz önüne almalıdır.

Manipülatör dinamiğini formülize etmek için Lagrange-Euler (L-E) [106-110], Recursive-Lagrange (R-L) [111], Newton-Euler (N-E) [112,113] ve Genelleştirilmiş D'Alambert Prensipleri (G-D) [109] gibi yaklaşımlar

önerilmiştir. Fakat yukarıda açıklanan isteklerde göz önüne alınarak manipülatörün dinamik modelini sistematik olarak çıkarmak için bir çok araştırmacı tarafından iki ana yaklaşım kullanılmaktadır. Bunlar; L-E ve N-E formülasyonlarıdır.

Bu tez çalışmasında, manipülatörün dinamik modelinin türetilmesinde, gayet iyi planmış, kompakt bir yapıya sahip ve çıkarılması N-E metodundakinden daha basit ve sistematik olan L-E metodu kullanılacaktır.

6.2.2. Hareketin L-E Denklemlerinin Çıkarılması

n serbestlik dereceli bir manipülatörün hareket denklemlerinin çıkarılmasında aşağıdaki bilgiler temel alınacaktır.

1- 4×4 'lük homojen koordinat dönüşüm matrisi A_{i-1}^i (6.1 ifadesine bakınız).

2- Hareketin L-E denklemleri

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, 2, \dots, n \quad (6.7)$$

Burada;

L : Lagrangian fonksiyonu = Kinetik Enerji (K) - Potansiyel Enerji (P),

K : Robot kolunun (manipülatörün) toplam kinetik enerjisi

P : " " " " "potansiyel enerjisi

q_i : " " " " genelleştirilmiş koordinatları, (Dönel eklem durumunda $q_i = \theta_i$, kayar eklem

durumunda $q_i = d_i$ 'dir),

$\dot{q}_i$: Genelleştirilmiş koordinat q_i 'nin birinci türevi,

τ_i : $\langle i \rangle$ uzvunu hareket ettiren $\langle i \rangle$ ekleminde sisteme uygulanan genelleştirilmiş kuvvet (veya tork).

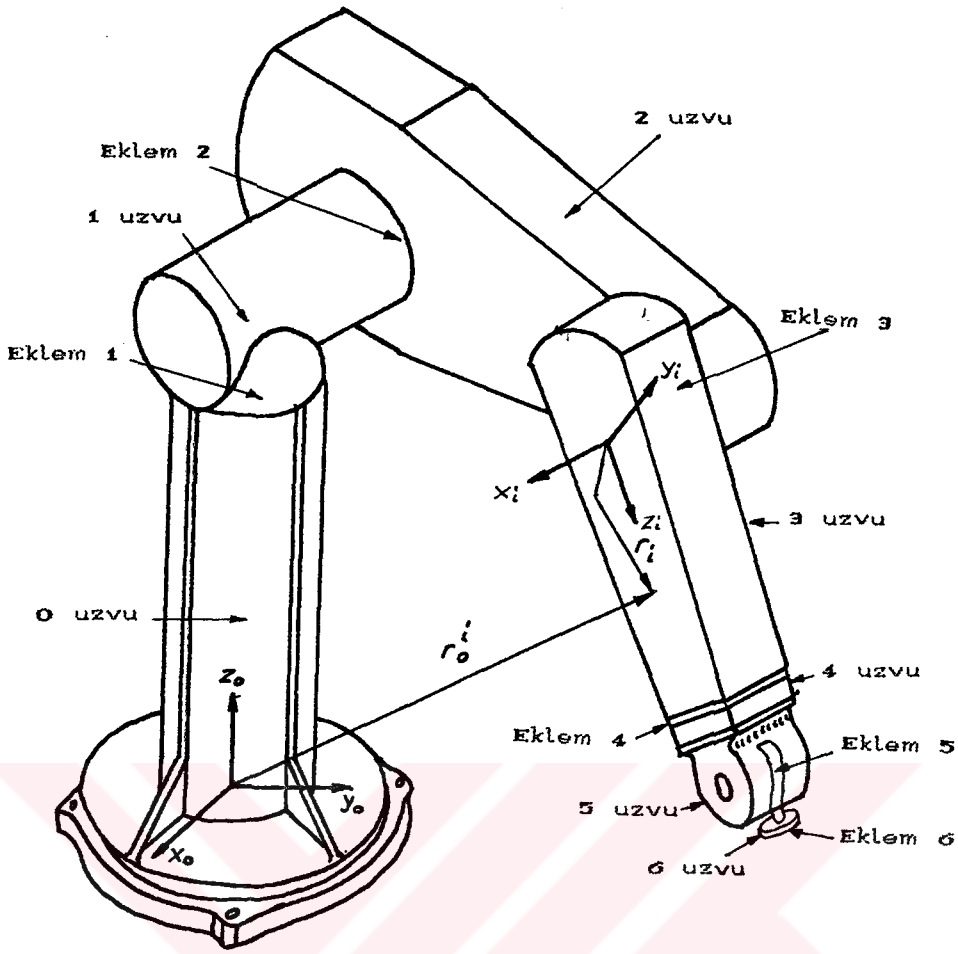
Bilgisayar kontrollu bir manipölatör için her bir eklemin açısal pozisyonu bir potansiyometre, enkoder veya diğer ölçme cihazları ile ölçülerek kolayca elde edilebilir. Böylece elde edilen eklem açısal pozisyonları, hareketin L-E denklemleri için gerekli genelleştirilmiş koordinatları oluştururlar.

6.2.3. Bir Robot Kolunun Hızı

Hareketin L-E denklemlerini yazabilmek için sistemin kinetik enerjisinin bilinmesi gerekir, bu; sıra ile her bir eklemin hızının bilinmesi demektir. Bu bölümde, $\langle i \rangle$ uzvundaki bir noktanın hızı ve bu uzuvdaki tüm noktalara diğer eklem hareketlerinin etkileri çıkarılacaktır.

r_i , $\langle i \rangle$. koordinat çerçevesine göre homojen koordinatlarda tanımlanmış ve $\langle i \rangle$ uzvunda sabit olan bir noktayı göstermek için kullanılsın (Şekil 6.4).

$$r_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = (x_i, y_i, z_i, 1)^T \quad \dots (6.8)$$



Şekil 6.5 (1) uzvundaki r_i noktası

(1). koordinat çerçevesine göre tanımlanan ve (1) uzvunda bulunan diğer noktalar gibi r_i noktasıda (1). koordinat çerçevesine göre sıfır hıza sahip olacaktır. Temel koordinat çerçevesinde r_i 'nin hızı;

$$V_i = \frac{dr_0^i}{dt} \quad \dots (6.9)$$

şeklinde hesaplanmaktadır. Burada; r_0^i , temel koordinat çerçevesine göre tanımlanan aynı r_i noktasıdır. Homojen dönüşüm matrisleri A_{i-1}^i ve A_0^i kullanılarak r_0^i aşağıdaki şekilde ifade edilebilir.

$$r_0^i = A_0^i r_i \quad \dots (6.10)$$

Burada; $A_0^i = A_0^1 A_1^2 \dots A_{i-1}^i$.dir.

A_0^i matrisinin elemanları, $(\theta_1, \theta_2, \dots, \theta_i)$.nin ve kolun kinematik yapısından bilinen α_i, a_i, d_i 'nin fonksiyonudur, θ_i veya d_i (i) ekleminin eklem değişkenleridir. Kayar ve dönel eklemlerin her ikisi içinde uygulanabilecek hareket denklemlerini çıkarabilmek için, (i) ekleminin genelleştirilmiş koordinatlarını ifade eden q_i değişkeni kullanılacaktır. q_i ; dönel eklem için θ_i , kayar eklem için ise d_i olacaktır.

Temel koordinat sistemi referans alınarak r_i noktasının mutlak hızı aşağıdaki gibi yazılabilir.

$$\begin{aligned} V_0^i = V_i &= \frac{d}{dt} [r_0^i] = \frac{d}{dt} [A_0^i r_i] \\ &= \dot{A}_0^1 A_1^2 \dots A_{i-1}^i r_i + A_0^1 \dot{A}_1^2 A_2^3 \dots A_{i-1}^i r_i \\ &\quad + A_0^1 A_1^2 \dots \dot{A}_{i-1}^i r_i + A_0^i \dot{r}_i \quad \dots (6.11a) \end{aligned}$$

$$= \left[\begin{matrix} i \\ \sum_{j=1} \frac{\partial A_0^i}{\partial q_j} \frac{dq_j}{dt} \end{matrix} \right] r_i = \left[\begin{matrix} i \\ \sum_{j=1} \frac{\partial A_0^i}{\partial q_j} \dot{q}_j \end{matrix} \right] r_i$$

$$i = 1, 2, \dots, n \quad \dots (6.11b)$$

A_0^i 'ın q_j 'ye göre kısmi türevi aşağıda tanımlanan Q_i matrisi yardımıyla kolayca hesaplanabilir.

Dönel bir eklem için;

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \dots (6.12a)$$

veya kayar bir eklem için

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \dots (6.12b)$$

Böylece;

$$\frac{\partial A_{i-1}^i}{\partial q_i} = Q_i A_{i-1}^i \quad \dots (6.13)$$

elde edilir. Buradan da;

$$\frac{\partial A_o^i}{\partial q_j} = \begin{cases} A_o^1 A_1^2 \dots A_{j-2}^{j-1} Q_j A_{j-1}^j \dots A_{i-1}^i & ; j \leq i \\ 0 & ; j > i \end{cases} \quad i = 1, 2, \dots, n \quad \dots (6.14)$$

yazılabilir. (6.14) ifadesi (1) uzvundaki tüm noktalara (j). eklem hareketinin etkisi olarak yorumlanabilir.

Notasyonları basitleştirmek için $U_{ij} \cong \partial A_o^i / \partial q_j$ tanımı kullanılarak (6.14) ifadesi $i = 1, 2, \dots, n$ için tekrar yazılabilir;

$$U_{ij} = \begin{cases} A_o^{j-1} Q_j A_{j-1}^i & ; j \leq i \text{ için} \\ 0 & ; j > i \text{ için} \end{cases} \quad \dots (6.15)$$

Bu notasyon kullanılarak V_i ifadesi $i = 1, 2, \dots, n$ için

$$V_i = \sum_{j=1}^i U_{ij} \dot{q}_j r_i \quad \dots (6.16)$$

olarak elde edilir.

Eklemeler arasındaki etkileşim etkilerini bulmak için

$$\frac{\partial U_{ij}}{\partial q_k} \cong U_{ijk} = \begin{cases} A_o^{j-1} Q_j A_{j-1}^{k-1} Q_k A_{k-1}^i, & 1 \leq k \leq j \\ A_o^{k-1} Q_k A_{k-1}^{j-1} Q_j A_{j-1}^i, & 1 \leq j \leq k \\ 0, & 1 < j \text{ veya } 1 < k \end{cases}$$

... (6.17)

ifadesinden faydalanılacaktır. Örneğin $i = j = k = 1$ ve $q_1 = \theta_1$ olan bir PUMA robot kolu için,

$$U_{111} = \frac{\partial U_{11}}{\partial \theta_1} = \frac{\partial [A_o^0 Q_1 A_o^1]}{\partial \theta_1} = [A_o^0 Q_1 Q_1 A_o^1] \quad (6.18)$$

olarak elde edilir. Buradaki A_o^0 4x4 'lük birim matristir. (6.18) ifadesi, (1) uzvundaki tüm noktalara (j). ve (k). eklem hareketlerinden olan etkileri göstermektedir.

6.2.4. Bir Robot Kolumun Kinetik Enerjisi

Her bir uzvun eklem hızları elde edildikten sonra, şimdi de (1) uzvunun kinetik enerjisi bulunacaktır. K_i , i uzvunun ($i=1,2,\dots,n$) temel koordinat sistemine göre tanımlanan kinetik enerjisi olsun. m kütleli bir parçacığın kinetik enerjisi $1/2 mV^2$ olduğundan, (1) uzvundaki kütlesi dm olan bir parçacığın kinetik enerjisi dK_i ;

$$\begin{aligned} dK_i &= 1/2 (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm \\ &= 1/2 \text{Tr} (V_i V_i^T) dm \end{aligned} \quad \dots (6.19)$$

(Not: $\text{Tr } A = \sum_{i=1}^n a_{ii}$)

(6.16) daki V_i ifadesi (6.19) 'da yerine konursa,

$$\begin{aligned} dK_i &= 1/2 \text{ Tr } \left[\sum_{p=1}^i U_{ip} \dot{q}_p r_i \left(\sum_{r=1}^i U_{ir} \dot{q}_r r_i \right)^T dm \right] \\ &= 1/2 \text{ Tr } \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \langle r_i r_i^T dm \rangle U_{ir}^T \dot{q}_p \dot{q}_r \right] \\ &\dots (6.20) \end{aligned}$$

elde edilir.

U_{ij} ve $\dot{q}_i$ 'ler (i) uzvunun kütlelesel dağılımından bağımsız olduklarından, (i) uzvundaki tüm parçacıkların kinetik enerjilerinin toplamı, integraller parantez içine alınarak

$$K_i = \int dK_i = 1/2 \text{ Tr } \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \langle \int r_i r_i^T dm \rangle U_{ir}^T \dot{q}_p \dot{q}_r \right] \dots (6.21)$$

şeklinde elde edilir. Parantez içindeki integral terimi (i). koordinat çerçevesine göre (i) uzvundaki tüm parçacıkların ataletidir ve genel olarak aşağıdaki şekilde gösterilebilir.

$$J_i = \int r_i r_i^T dm = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix}$$

... (6.22)

$$I_{ij} = \int \left(\delta_{ij} \left[\sum_k x_k^2 \right] - x_i x_j \right) dm \quad \dots (6.23)$$

olarak tanımlanan I_{ij} atalet tansörü kullanılarak J_i aşağıdaki şekilde ifade edilebilir. I_{ij} ifadesindeki i, j, k (1). koordinat çerçevesinin asal eksenlerini ve δ_{ij} ise Kronecker deltayı göstermektedir ($i=j$ için $\delta_{ij}=1$ ve $i \neq j$ için $\delta_{ij}=0$ dır).

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

... (6.24)

veya (x_i, y_i, z_i) koordinat sisteminde m_i rijid cisminin jirasyon yarıçapları kullanılarak J_i atalet matrisi aşağıdaki şekilde ifade edilebilir. Burada k_{ijk} , j - k eksenini etrafında (1) uzvunun jirasyon yarıçapını

göstermektedir, $\bar{r}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, 1)^T$, (1). koordinat çerçevesinde tarif edilmiş (1) uzvunun kütle merkezinin koordinatlarıdır.

$$J_i = m_i \begin{bmatrix} \frac{-k_{i11}^2 + k_{i22}^2 + k_{i33}^2}{2} & k_{i12}^2 & k_{i13}^2 & \bar{x}_i \\ k_{i12}^2 & \frac{k_{i11}^2 - k_{i22}^2 + k_{i33}^2}{2} & k_{i23}^2 & \bar{y}_i \\ k_{i13}^2 & k_{i23}^2 & \frac{k_{i11}^2 + k_{i22}^2 - k_{i33}^2}{2} & \bar{z}_i \\ \bar{x}_i & \bar{y}_i & \bar{z}_i & 1 \end{bmatrix} \quad \dots (6.25)$$

Tüm uzuvlar için kinetik enerjilerin toplamı robot kolunun toplam kinetik enerjisi K 'yı verecektir.

$$\begin{aligned} K &= \sum_{i=1}^n K_i = 1/2 \sum_{i=1}^n \text{Tr} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r \right] \\ &= 1/2 \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \left[\text{Tr} (U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r \right] \\ &= \text{skalar bir büyüklük} \quad \dots (6.26) \end{aligned}$$

Not: J_i , (1) uzvunun kütleli dağılımına bağlıdır, uzuvların pozisyonlarına ve hareket oranına bağlı değildir ve (1). koordinat çerçevesine göre tanımlanmıştır. Bu yüzden J_i , herhangi bir robot kolunun kinetik enerjisinin belirlenmesinde sadece bir kez hesaplanmaktadır.

6.2.5. Bir Robot Kolunun Potansiyel Enerjisi

Bir robot kolunun toplam potansiyel enerjisi P ve uzuvlarının her birinin potansiyel enerjisi P_i olarak kabul edilirse

$$P_i = -m_i g r_o^i = -m_i g \langle A_o^i \bar{r}_i \rangle ; i=1,2,\dots,n \quad \dots (6.27)$$

robot kolunun toplam enerjisi, her bir uzuvdaki potansiyel enerjiler toplanarak elde edilebilir.

$$P = \sum_{i=1}^n P_i = \sum_{i=1}^n -m_i g \langle A_o^i \bar{r}_i \rangle \quad \dots (6.28)$$

Burada $g = \langle g_x, g_y, g_z, 0 \rangle$ temel koordinat sisteminde tarif edilmiş yerçekimi vektörüdür. Deniz seviyesindeki bir sistem için $g = \langle 0, 0, -|g|, 0 \rangle$ ve g yerçekimi ivmesidir ($g=9.8062 \text{ m/sn}^2$).

6.2.6. Hareketin Lagrange-Euler Denklemleri

(6.26) ve (6.28) ifadelerini kullanarak Lagrangian fonksiyonu $L = K - P$ aşağıdaki şekilde yazılır:

$$L = 1/2 \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \left[\text{Tr} \langle U_{ij} J_i U_{ik}^T \rangle \dot{q}_j \dot{q}_k \right] + \sum_{i=1}^n m_i g \langle A_o^i \bar{r}_i \rangle \quad \dots (6.29)$$

(6.7) nolu Lagrange-Euler denklemlerini (6.29) 'a uygulayarak bir manipülatörün hareketine ait dinamik denklemler elde edilir.

$$\begin{aligned}
\tau_i &= \sum_{j=1}^n \sum_{k=1}^j \text{Tr} \langle U_{jk} J_j U_{ji}^T \rangle \ddot{q}_k + \\
&+ \sum_{j=1}^n \sum_{k=1}^j \sum_{m=1}^j \text{Tr} \langle U_{jkm} J_j U_{ji}^T \rangle \dot{q}_k \dot{q}_m - \\
&- \sum_{j=1}^n m_j g U_{ji} \bar{r}_j ; i=1,2,\dots,n \quad \dots (6.30)
\end{aligned}$$

Yukarıdaki ifade daha basit bir notasyon ile

$$\tau_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m + G_i \quad i=1,2,\dots,n \quad \dots (6.31)$$

şeklinde veya matris formunda

$$\tau(t) = D(q(t)) \ddot{q}(t) + H(q(t), \dot{q}(t)) + G(q(t)) \dots (6.32)$$

şeklinde gösterilebilir.

Burada;

$D(q(t))$, ivmelenme ile ilgili simetrik atalet matrisidir, elemanları:

$$D_{ik} = \sum_{j=\max(i,k)}^n \text{Tr} \langle U_{jk} J_j U_{ji}^T \rangle \quad k=1,2,\dots,n \quad \dots (6.33)$$

şeklindedir.

$H(q(t), \dot{q}(t))$, $n \times 1$ 'lik lineer olmayan Koriolis ve merkezkaç kuvvet vektörüdür.

$$H(q(t), \dot{q}(t)) = (H_1, H_2, \dots, H_n)^T \quad \dots (6.34a)$$

$$H_i = \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m, \quad i=1,2,\dots,n \quad \dots (6.34b)$$

$$H_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr} (U_{jkm} J_j U_{ji}^T) \quad , i,k,m=1,2,\dots,n \dots (6.34c)$$

$G(q(t))$, $n \times 1$ 'lik yerçekimi kuvveti vektörüdür.

$$G(q(t)) = (G_1, G_2, \dots, G_n)^T \quad \text{ve} \quad \dots (6.35a)$$

$$G_i = \sum_{j=i}^n (-m_j g U_{ji} \bar{r}_j), \quad i=1,2,\dots,n \quad \dots (6.35b)$$

şeklindedir.

τ , $n \times 1$ 'lik eklem aktuatörlerine uygulanan tork/kuvvet'i göstermektedir.

(6.31) ifadesinden de görüldüğü gibi altı serbestlik dereceli bir robot koluna ait hareket denklemleri birbirine bağımlı, lineer olmayan, ikinci mertebeden altı adet diferansiyel denklemden oluşmaktadır. Zamanın fonksiyonu olarak eklemlere uygulanan torklar τ_i ($i=1,2,\dots,6$) verilirse, (6.31) nolu ifade integre edilerek eklem değişkenlerine $(\theta_1, \theta_2, \dots, \theta_6)$ bağılı olarak manipülatörün hareketi elde edilir. Buradan da uygun homojen dönüşüm matrisleri kullanılarak ele ait hareket yörüngesine geçilebilir. Veya eklem değişkenlerinin takip edeceği yörünge zamana bağılı olarak biliniyorsa yine (6.31) nolu ifadeden eklem momentleri (veya kuvvetleri) hesaplanabilir.

Yukarıda elde edilen L-E denklemleri, matrissel yapısından dolayı direkt durum denklemleri takımını verdiğinden kontrol açısından oldukça cazip olmaktadır. Bu form, tüm lineer olmayan etkileri kolayca dengeleyebilecek bir kontrol kanunu dizaynına olanak vermektedir. Fakat, bu denklemlerin hesapsal açıdan çok verimli olmadığını belirtmekte de fayda vardır. Bu tez çalışmasında da, robotik manipülatörün simülasyonunda L-E denklemleri kullanılmıştır.

BÖLÜM 7. ROBOT SİMÜLASYONU

7.1 Genel Bilgiler

Bu bölümde, genelleştirilmiş öngörülü kontrol algoritmasının bir uygulaması olarak, üç eklemlili bir robot için yapılan simülasyon çalışmaları verilecektir. Ayrıca bir karşılaştırma imkânı sağlamak amacı ile kendi kendini ayarlayan kontrol algoritması kullanılarak yapılan bir simülasyon çalışması da ilave edilmiştir.

Bölüm 6 ' da da açıklandığı gibi robot hareketini modellemede en uygun yöntem olarak gözüken Lagrange-Euler yöntemi kullanılmıştır. Buna göre, n eklemlili bir robotda, $(n \times 1)$ boyutlu eklem tork vektörü $\tau(t)$ ile eklem açıları $\theta(t)$ arasındaki ilişki, aşağıdaki gibi bir non-linear diferansiyel denklem takımı ile verilir.

$$\tau(t) = D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) \quad \dots (7.1)$$

Tez çalışmaları sırasında, yukarıdaki genel Lagrange-Euler denklem takımına, eklemlerdeki sürtünmeler ve robotun uç elemanında (end-effector) bir yük taşınması durumu da ayrıca ilave edilerek en genel durumda simülasyon yapılması sağlanmıştır. Eklemlerdeki sürtünme tork vektörünün ilave edilmesi ile yukarıdaki (7.1) denklemi aşağıdaki şekilde ifade edilmektedir [25].

$$\tau(t) = D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) + S(\theta) \quad \dots (7.2)$$

(7.2) denklemi taşınan yükün olmaması durumunda dinamiğini tasvir etmektedir. $S(\theta)$, sürtünme tork vektörüdür.

Şimdi, sabit bir kartezyen koordinat sistemine göre uç elemanın pozisyon ve oryantasyonunun $(n \times 1)$ boyutlu X vektörü ile gösterildiği kabul edilsin. Uç elemanın pozisyon/oryantasyon vektörü X ile eklem açısı vektörü θ arasındaki ilişki aşağıdaki gibi gösterilebilir [114].

$$X(t) = \Lambda(\theta) \quad \dots (7.3)$$

Burada; $\Lambda(\theta)$, $(n \times 1)$ boyutlarında manipölatör dinamiğini içeren ve düz kinematiği simgeleyen bir vektördür.

Eklem uzayı $\{\theta\}$ ile kartezyen uzay $\{X\}$ arasındaki dönüşümü sağlayan bağıntılar aşağıdaki şekilde verilmektedir.

$$dX(t) = J(\theta) d\theta(t) \quad \dots (7.4a)$$

$$\dot{X}(t) = J(\theta) \dot{\theta}(t) \quad \dots (7.4b)$$

$$\ddot{X}(t) = \dot{J}(\theta, \dot{\theta}) \dot{\theta}(t) + J(\theta) \ddot{\theta}(t) \quad \dots (7.4c)$$

Burada, $J(\theta) = [\partial \Lambda(\theta) / \partial \theta]$, $(n \times n)$ boyutlarında manipölatör Jacobian matrisidir.

Bu bilgilerinde ışığı altında, manipölatör dinamiğine uç eleman tarafından taşınan yükün etkisi ilave edilebilir [114]. Manipölatör uç elemanının sabit bir "m" noktasal kütesini taşıdığı kabul edilsin. Yükün $\ddot{X}(t)$ ivmesi ile hareket edebilmesi için uç eleman yüke, aşağıdaki $(n \times 1)$ ' lik $f(t)$ kuvvetini uygulamalıdır.

$$f(t) = m [\ddot{X}(t) + g] \quad \dots (7.5)$$

Burada; g $(n \times 1)$ boyutlarında yerçekimi ivmesi vektörüdür. Uç eleman kuvveti $f(t)$, ilave bir eklem torku

$$\tau_f(t) = J^T(\theta) f(t) \quad \dots (7.6)$$

gerektirmektedir. "T" indisi transpoze işlemini göstermektedir. Böylece toplam eklem tork vektörü, (7.2) ve (7.6)'nın birleştirilmesi ile aşağıdaki şekilde elde edilir:

$$\tau(t) = J^T(\theta) f(t) + D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) + S(\dot{\theta}) \quad \dots (7.7)$$

$f(t)$ 'nin (7.4c) ve (7.5) ifadeleri kullanılarak elde edilen karşılığı yukarıda yerine konursa;

$$\tau(t) = m L(\theta, \dot{\theta}, \ddot{\theta}) + D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta) + S(\dot{\theta}) \quad \dots (7.8)$$

Burada;

$$L(\theta, \dot{\theta}, \ddot{\theta}) = J^T(\theta) [J(\theta) \ddot{\theta} + \dot{J}(\theta, \dot{\theta}) \dot{\theta} + g]$$

şeklindedir. (7.8) ifadesi, manipülatör dinamiğinde taşınan "m" kütesinin etkisini açıkça göstermektedir.

$S(\dot{\theta})$ sürtünme tork vektörü ise kulomb ve viskoz sürtünme etkilerinden meydana gelmekte ve kullanılan üç eklemlili robot modeli için aşağıdaki şekilde tanımlanmaktadır [25].

$$S(\dot{\theta}) = \begin{bmatrix} V_1 \dot{\theta}_1 + V_2 \operatorname{sgn}(\dot{\theta}_1) \\ V_3 \dot{\theta}_2 + V_4 \operatorname{sgn}(\dot{\theta}_2) \\ V_5 \dot{\theta}_3 + V_6 \operatorname{sgn}(\dot{\theta}_3) \end{bmatrix} \quad \dots (7.9)$$

Burada $\{V_1, V_3, V_5\}$ ve $\{V_2, V_4, V_6\}$ sırası ile viskoz ve kulomb sürtünme katsayılarıdır. Simülasyon çalışmalarında sürtünme katsayıları $V_1 = V_3 = V_5 = 1.0$ Nt.m/rad.s⁻¹ ve $V_2 = V_4 = V_6 = 0.5$ Nt.m olarak alınmıştır [25].

$\text{sgn}(\dot{\theta}_i)$ ise;

$\dot{\theta}_i = 0$ ise $\text{sgn}(\dot{\theta}_i) = 0$

$\dot{\theta}_i > 0$ ise $\text{sgn}(\dot{\theta}_i) = 1$

$\dot{\theta}_i < 0$ ise $\text{sgn}(\dot{\theta}_i) = -1$

olacak şekilde tarif edilmektedir.

Yapılan simülasyon çalışmalarında, gerek kendi kendini ayarlayan kontrol algoritmasının gerekse öngörülü kontrol algoritmasının kullanılması durumunda yukarıdaki (7.8) nolu ifade, manipülatör dinamiğini tanımlamak için esas olarak alınmıştır.

7.2 Kullanılan Robot Modeli

Kullanılan kontrol algoritmalarının geçerliliğini göstermek için çeşitli simülasyon çalışmaları yapılmıştır. Bu amaç için, şekil 7.1' de gösterilen üç uzuvlu manipülatör kullanılmıştır [115,3].

7.2.1 Robot Modelinin Jacobian Matrisinin Teşkili

Jacobian matrisinin teşkili için ilk olarak kullanılan robot modelinin uç elemanının x,y,z koordinatları hesaplanacaktır. Robot modeli, uç elemanın x,y,z koordinatlarının hesaplanmasında yardımcı olması için şematik olarak, Şekil 7.2' de tekrar verilmiştir.

Tablo 7.1 Robot modelinin parametre deęerleri

1	1	2	3	birim
m_i	13.1339	10.3320	6.4443	kg.
a_i	0.1588	0.445	0.0	m.
α_i	$\pi/2$	0.0	$\pi/2$	radyan
$^*x_i$	-0.0493	-0.1618	0.0	m.
$^*z_i$	0.0	0.0	0.2718	m.
k_{i11}^2	5.6064	3.929	82.0644	$m^2 \times 10^{-3}$
k_{i22}^2	8.9196	47.8064	81.9353	$m^2 \times 10^{-3}$
k_{i33}^2	13.2387	45.4838	1.400	$m^2 \times 10^{-3}$

NOT: Tablo 7.1'de

m_i = i uzvunun kütlesi

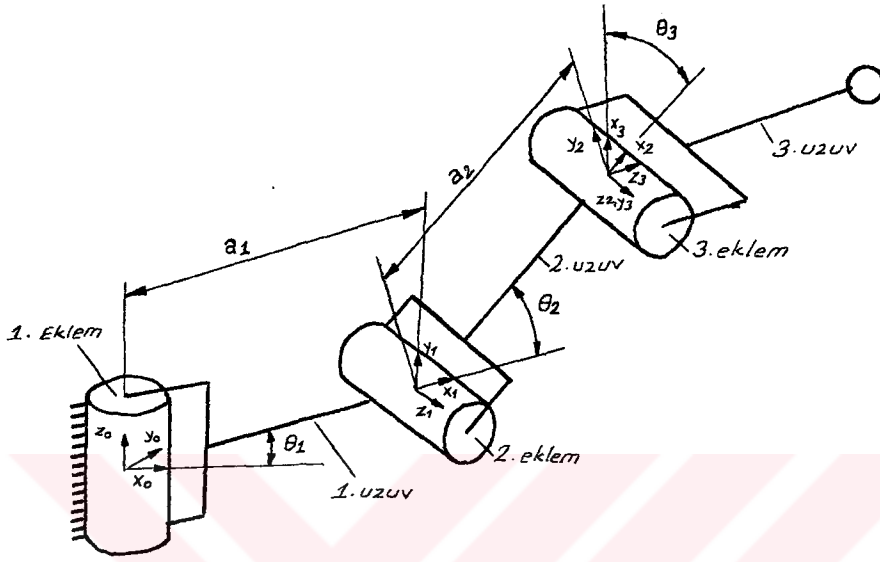
a_i = Döneren x_{i-1} eksenini boyunca i. koordinat sisteminin kayma mesafesi

α_i = z_{i-1} eksenine göre z_i ekseninin oryantasyonu

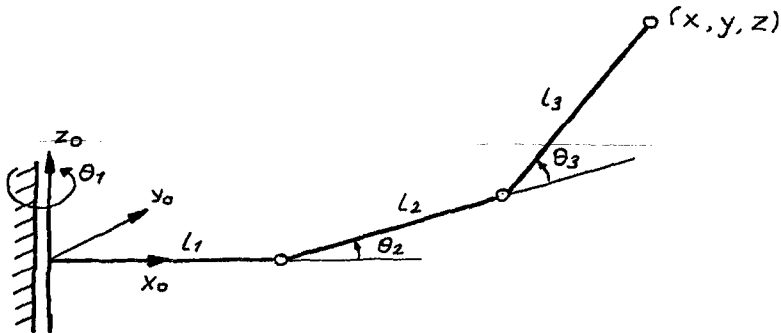
$^*x_i, ^*z_i$ = i. koordinat çerçevesine göre i uzvunun kütle merkezinin koordinatları

k_{ijj} = Jirasyon yarıçapı ($I_{ij} = m_i k_{ijj}^2$)

olarak verilmiştir.



Şekil 7.1 Uç uzuvlu robot modeli



Şekil 7.2 Robot modelinde x,y,z koordinatlarının teşkili

Şekil 7.2' nin de yardımı ile uç elemanın kartezyen koordinatları;

$$x = (l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \cos \theta_1$$

$$y = (l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \sin \theta_1$$

$$z = l_2 \sin \theta_2 + l_3 (\sin (\theta_2 + \theta_3))$$

şeklinde elde edilmektedir. Bu ifadeler de göz önüne alınarak Jacobian matrisi;

$$J = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial z}{\partial \theta_1} & \frac{\partial z}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \end{bmatrix} \quad \dots (7.10)$$

$$J(1) = -(l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \sin \theta_1$$

$$J(2) = (l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \cos \theta_1$$

$$J(3) = 0$$

$$J(4) = (-l_2 \sin \theta_2 - l_3 \sin (\theta_2 + \theta_3)) \cos \theta_1$$

$$J(5) = (-l_2 \sin \theta_2 - l_3 \sin (\theta_2 + \theta_3)) \sin \theta_1$$

$$J(6) = l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)$$

$$J(7) = -l_3 \sin (\theta_2 + \theta_3) \cos \theta_1$$

$$J(8) = -l_3 \sin (\theta_2 + \theta_3) \sin \theta_1$$

$$J(\theta) = l_3 \cos(\theta_2 + \theta_3)$$

şeklinde elde edilmektedir.

7.2.2 Simülasyonda Kullanılan Eklem Referans Yörüngeleri

Robotun her bir eklemine ait takip edilmesi istenen konum referans yörüngeleri ve hız referans yörüngeleri aşağıdaki şekilde seçilmiştir:

1- Kübik yörünge

Toplam simülasyon zamanı: 1 sn.

Örnekleme periyodu: 0.0025 sn

Toplam simülasyon adımı sayısı: 400 adım

Başlangıç şartları:

$$\theta_1 = \theta_2 = 0^\circ, \theta_3 = 30^\circ$$

$$\dot{\theta}_1 = 1.5708 \text{ rad/sn.}, \dot{\theta}_2 = \dot{\theta}_3 = 0 \text{ rad/sn.}$$

Final şartları;

$$\theta_1 = 90^\circ, \theta_2 = 45^\circ, \theta_3 = 20^\circ$$

$$\dot{\theta}_1 = 1.5708 \text{ rad/sn.}, \dot{\theta}_2 = \dot{\theta}_3 = 0 \text{ rad/sn.}$$

Eklem konum referansları

$$\theta_1(t) = 1.5708t \quad 0 \leq t \leq 1 \text{ sn.}$$

$$\theta_2(t) = -8.3776t^3 + 6.2832t^2 + 0.5236 \quad 0 \leq t \leq 0.5 \text{ sn.}$$

$$\theta_2(t) = 4.1888t^3 - 9.4248t^2 + 6.2832t - 0.2618 \quad 0.5 \leq t \leq 1 \text{ sn.}$$

$$\theta_3(t) = 12.5664t^3 - 9.4248t^2 \quad 0 \leq t \leq 0.5 \text{ sn}$$

$$\theta_3(t) = -18.15152t^3 + 40.84092t^2 - 27.22728t + 4.8869 \quad 0.5 \leq t \leq 1 \text{ sn.}$$

Eklem hız referansları

$$\dot{\theta}_1(t) = 1.5708 \quad 0 \leq t \leq 1 \text{ sn.}$$

$$\dot{\theta}_2(t) = -25.1328t^2 + 12.5664t \quad 0 \leq t \leq 0.5 \text{ sn.}$$

$$\dot{\theta}_2(t) = 12.5664t^2 - 18.8496t + 6.2832 \quad 0.5 \leq t \leq 1 \text{ sn.}$$

$$\dot{\theta}_3(t) = 37.6992t^2 - 18.8496t \quad 0 \leq t \leq 0.5 \text{ sn.}$$

$$\dot{\theta}_3(t) = -54.45456t^2 + 81.68184t - 27.22728 \quad 0.5 \leq t \leq 1 \text{ sn.}$$

2- Ekspansiyel yörünge

Toplam simülasyon zamanı: 1 sn.

Örnekleme periyodu: 0.0025 sn.

Toplam simülasyon adımı sayısı: 400 adım

Başlangıç şartları;

$$\theta_1 = \theta_2 = \theta_3 = -45^\circ (-0.78539 \text{ rad}), \quad \dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = 0 \text{ rad/sn.}$$

Final şartları;

$$\theta_1 = \theta_2 = \theta_3 = -7^\circ (-0.1247 \text{ rad}), \quad \dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = 0.77 \text{ rad/sn.}$$

Eklem konum referansları

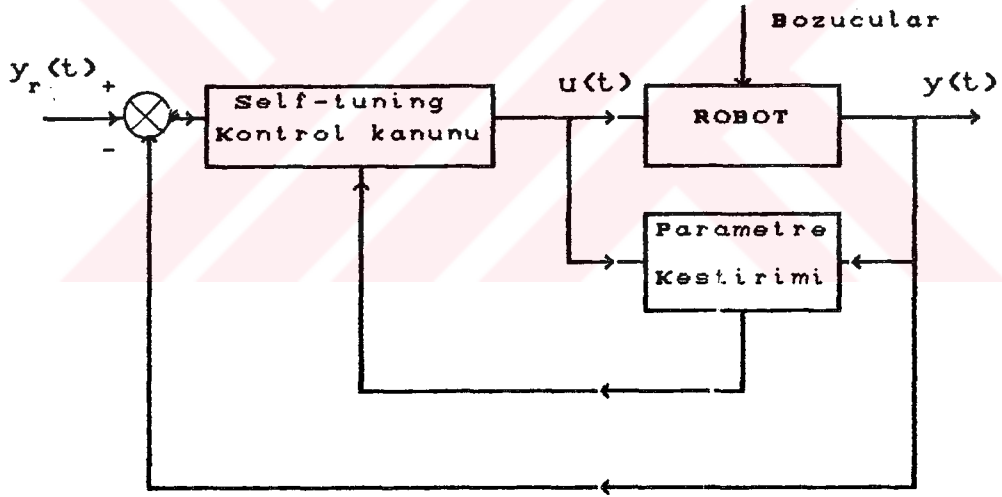
$$\theta_1(t) = \theta_2(t) = \theta_3(t) = 0.78539(1 + 6\exp(-t/0.6) - 8\exp(-t/0.8)) \quad 0 \leq t \leq 1 \text{ sn.}$$

Eklemler hız referansları

$$\begin{aligned} \dot{\theta}_1(t) = \dot{\theta}_2(t) = \dot{\theta}_3(t) = & 7.8539(-\exp(-t/0.6) + \\ & + \exp(-t/0.8)) \quad 0 \leq t \leq 1 \text{ sn.} \end{aligned}$$

7.3 Robot Kontrolü için Kendi Kendini Ayarlayan Kontrolör Dizaynı

Bu bölümde, Borison [40]' un yaklaşımını kullanan kendi kendini ayarlayan bir kontrolörün robotik bir manipülatöre uygulanması incelenecektir. Kontrolöre ait blok diyagramı şekil 7.3' de verilmiştir.



Şekil 7.3 Kendi kendini ayarlayan kontrolör blok diyagramı

Üç eklemlerli manipülatörün her bir eklemleri için ayrı bir kontrolör dizaynı düşünülmüştür, yani; her bir eklemler diğerlerinden bağımsız olarak kontrol edilmektedir (SISO-STR). Bu, manipülatör sisteminin dinamik modelindeki etkileşim terimlerinin ihmal edilmesi anlamına

gelmektedir. Kendi kendini ayarlayan kontrol algoritması tezin esas konusunu teşkil etmediğinden, çok değişkenli (MIMO-STR) kontrol durumunun incelenmesine gerek görülmemiştir. MIMO hali, öngörülü kontrol algoritması için bölüm 7.4.2' de geniş bir şekilde incelenecektir.

Gerçekleştirilen simülasyon çalışmaları, robot dinamik modelinin aşağıdaki fark denklemi ile modellenebileceğini göstermiştir.

$$(1 + a_i^1 q^{-1}) y_i(t) = (b_i^0 + b_i^1 q^{-1} + b_i^2 q^{-2}) u_i(t-d) + e(t)$$

$$i = 1, \dots, n \quad \dots (7.11)$$

Bu model için giriş $u(t)$, her bir eklem motoruna uygulanacak torkları, çıkış $y(t)$, eklem açısal hızlarını, $e(t)$ modelleme hatalarını, n eklem sayısını göstermektedir. d ölü zamandır. a polinomu 1. mertebeden ve b polinomu 2. mertebeden seçilmiştir. Bu seçim ile, 2. mertebeden nonlinear diferansiyel denklem takımı (hareketin Lagrange-Euler denklemleri) yeterince karakterize edilebilmektedir. Ölü zaman $d=1$ olarak alınmıştır.

Kontrolör dizaynında kullanılan performans kriteri aşağıdaki şekilde seçilmiştir.

$$J_i(u_i) = E \left\{ [y_i(t+d) - y_{ir}(t+d)]^2 + Q_i [u_i(t) - u_i(t-1)]^2 \right\}$$

$$i = 1, \dots, n \quad \dots (7.12)$$

Burada E , beklenti operatörüdür. y_{ir} her bir eklem için açısal hız referans yörüngesini (θ_i) göstermektedir. Q_i , 1 olarak alınmıştır.

(7.11) ifadesindeki parametrelerin kestirilmesi için ardışık en küçük kareler metodu (RLS) kullanılmıştır. Parametre vektörü,

$$\Theta_i = [b_i^0, b_i^1, b_i^2, a_i^1]$$

veri vektörü ise

$$\phi_i(t-1) = [u_i(t-1), u_i(t-2), u_i(t-3), y_i(t-1)]$$

olarak belirlenmiştir.

(7.12) nolu performans indeksini minimize eden u kontrol büyüklüğü her bir eklem için

$$u_i(t) = \frac{\hat{b}_i^0}{(\hat{b}_i^0)^2 + Q_i} \left\{ |y_{ir}(t+1) - \hat{y}_i(t+1)| + Q_i u_i(t-1) \right\}_{u(t)=0} \dots (7.13)$$

şeklinde hesaplanır [116]. " $\hat{}$ " işareti kestirilen değerleri göstermektedir.

7.3.1 Sayısal Örnekler

Simülasyon çalışmalarında bölüm 7.2' de verilen üç eklemlili manipülatör seçilmiş, (7.8) denklemi ile verilen en genel haldeki dinamik model kullanılarak değişik senaryolar için simülasyon çalışmaları gerçekleştirilmiştir.

Simülasyon sırasında her üç ekleme uygulanacak eklem torkları için $u_{min} = -225 \text{ Nm.}$, $u_{max.} = + 225 \text{ Nm.}$ sınırlaması getirilmiştir.

Q_i ağırlık faktörü zamanla değişecek şekilde $[b_i^0(t-1)]^2$ 'ye eşit alınmıştır [15]. Parametre kestirme algoritması olarak yapılan değişik simülasyon çalışmaları neticesinde en uygun sonuçları veren ardışık en küçük kareler parametre kestirme algoritması (RLS)

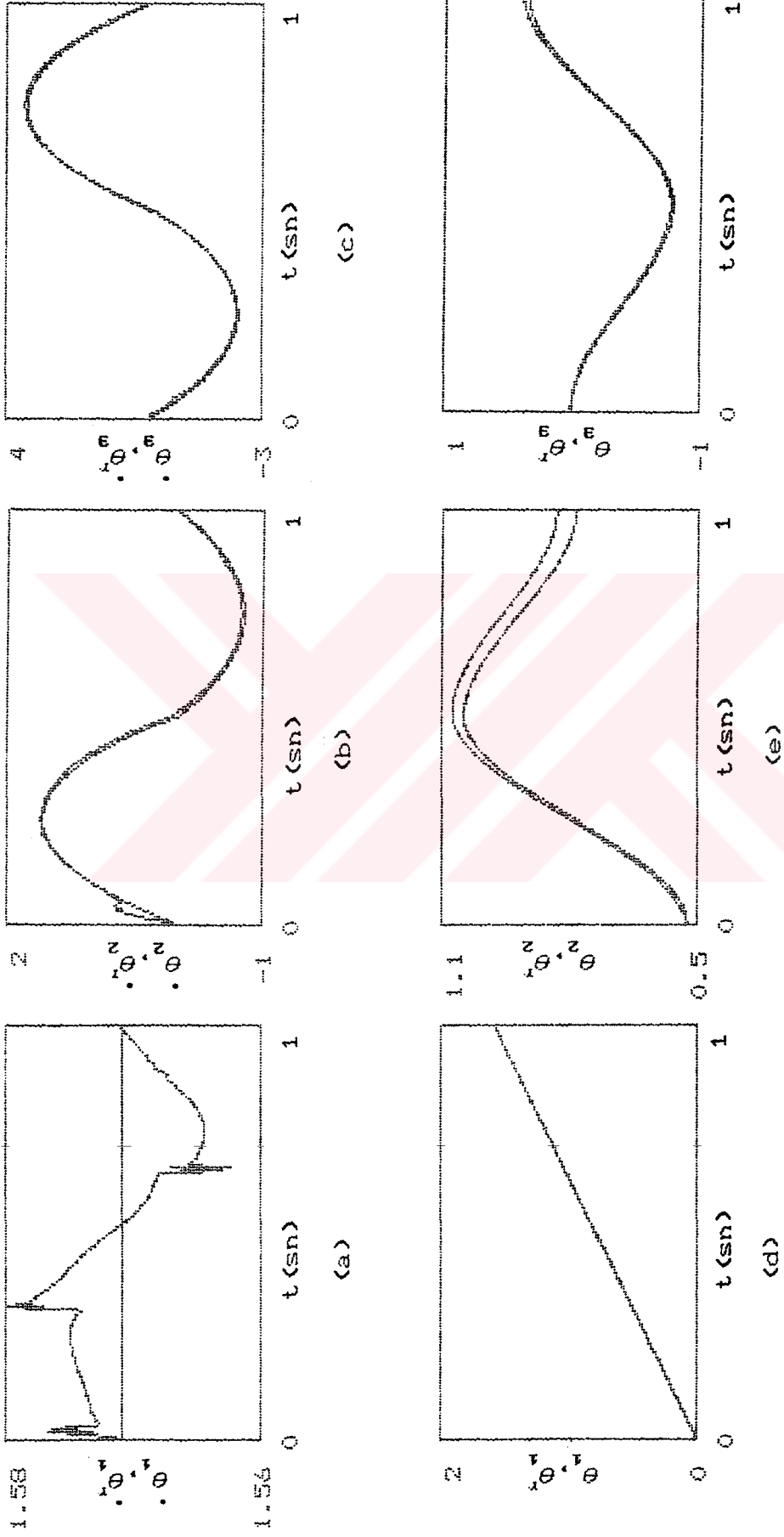
kullanılmıştır. Parametre kestirme algoritması (RLS)'de kullanılan μ unutma faktörü her bir eklem için 0.95 olarak, başlangıç değerleri $\hat{b}_i^0 = 1$, $\hat{b}_i^1 = \hat{b}_i^2 = \hat{a}_i^1 = 0$ ve $P_i(0) = 10^5 I$ olarak kabul edilmiştir. Dinamik denklemlerin çözümü için kullanılan Runge-Kutta integrasyon yönteminde, integrasyon zamanı ve parametre kestirme periyodu $\Delta t = 0.0025$ sn. olarak eşit alınmıştır.

Bölüm 7.2.2' de verilen kübik ve eksponansiyel referans yörüngeleri kullanılmış ve üç farklı simülasyon senaryosu üzerinde çalışılmıştır:

a)- (7.1) ifadesi ile gösterilen hiç bir bozucu büyüklüğün, taşınan yükün ve sürtünme etkilerinin göz önüne alınmadığı en yalın hal. Eklem açıları ve hızlarının referans değerleri ile birlikte gösterildiği simülasyon sonuçları şekil 7.4' de , her bir ekleme ait konum hatalarının karelerinin toplamı tablo 7.2'de, mutlak final konum hataları ise tablo 7.3' de verilmiştir.

b)- Yukarıdaki çalışmaya ilave olarak robot kolunun uç elemanının 10 kg.lık bir yük taşıdığı ve bu yükü 100. simülasyon adımımda (0.25 sn) bıraktığı, ayrıca eklem hızlarının üzerine, ölçme hatası olarak varsayılan ortalama değeri sıfır ve standart sapması 0.005 olan rasgele bozucular eklenmiştir. Böylece yük değişimleri ve ölçme hataları karşısında kontrolörün performansı araştırılmıştır. Simülasyon sonuçları şekil 7.5 ve tablo 7.2 ile 7.3' de gösterilmiştir.

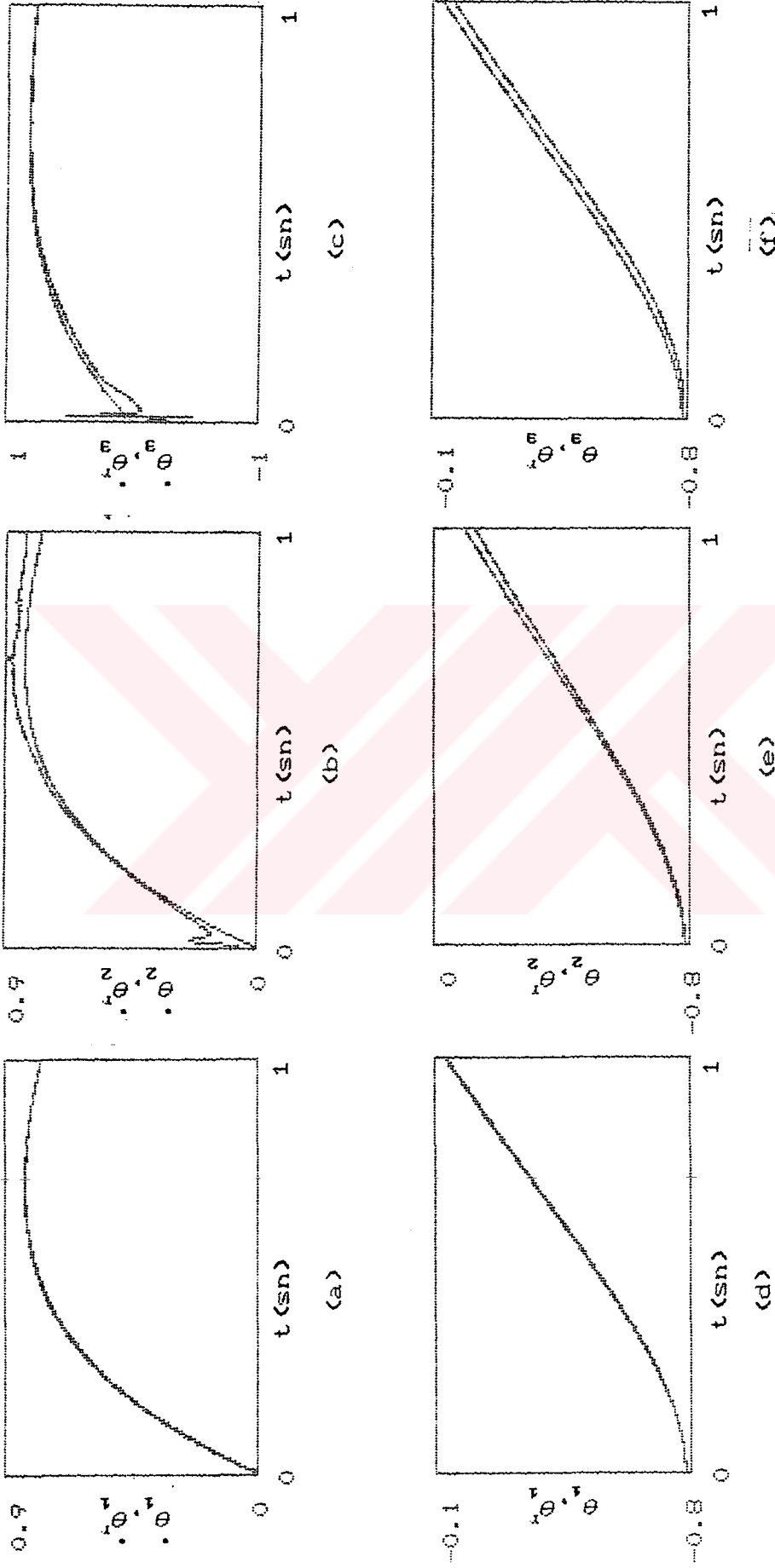
c)- b şikkında verilen çalışmaya ilave olarak, her bir eklemden sırası ile sürtünme katsayıları 1.0 ve 0.5 olan viskoz ve kulomb sürtünmelerinin mevcudiyeti varsayılarak, sürtünmenin performans üzerindeki etkisi incelenmiştir. Sonuçlar şekil 7.6 ve tablo 7.2' ve 7.3' de verilmiştir. u kontrol büyüklüğünün kübik yörünge için zamanla değişimi ise şekil 7.6.c' de verilmiştir.



Sekil 7.4.a. Self-tuning kontrolör (bozucusuz durum, kübik yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

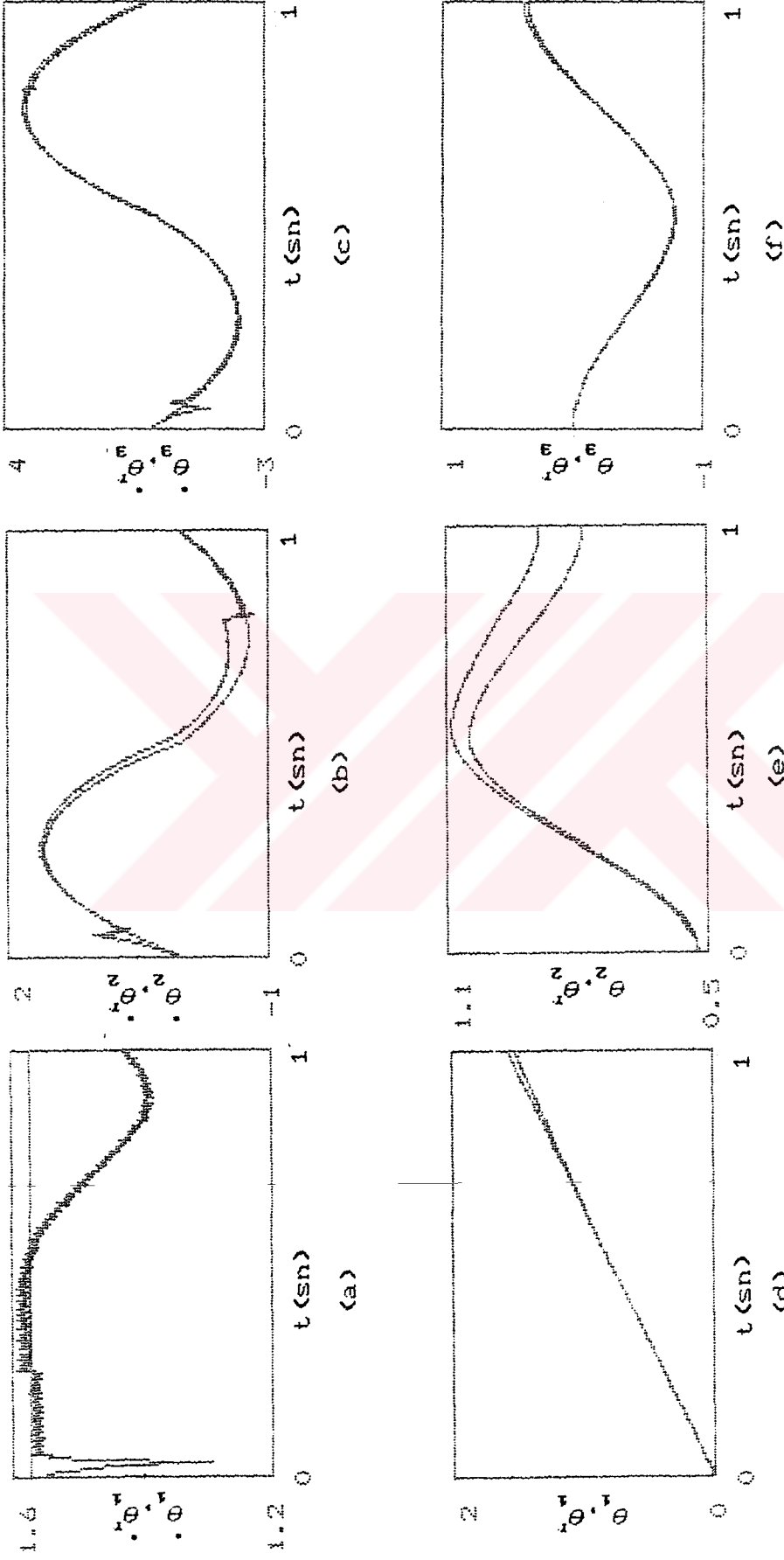
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.4.b. Self-tuning kontrolör (eksponansiyel yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

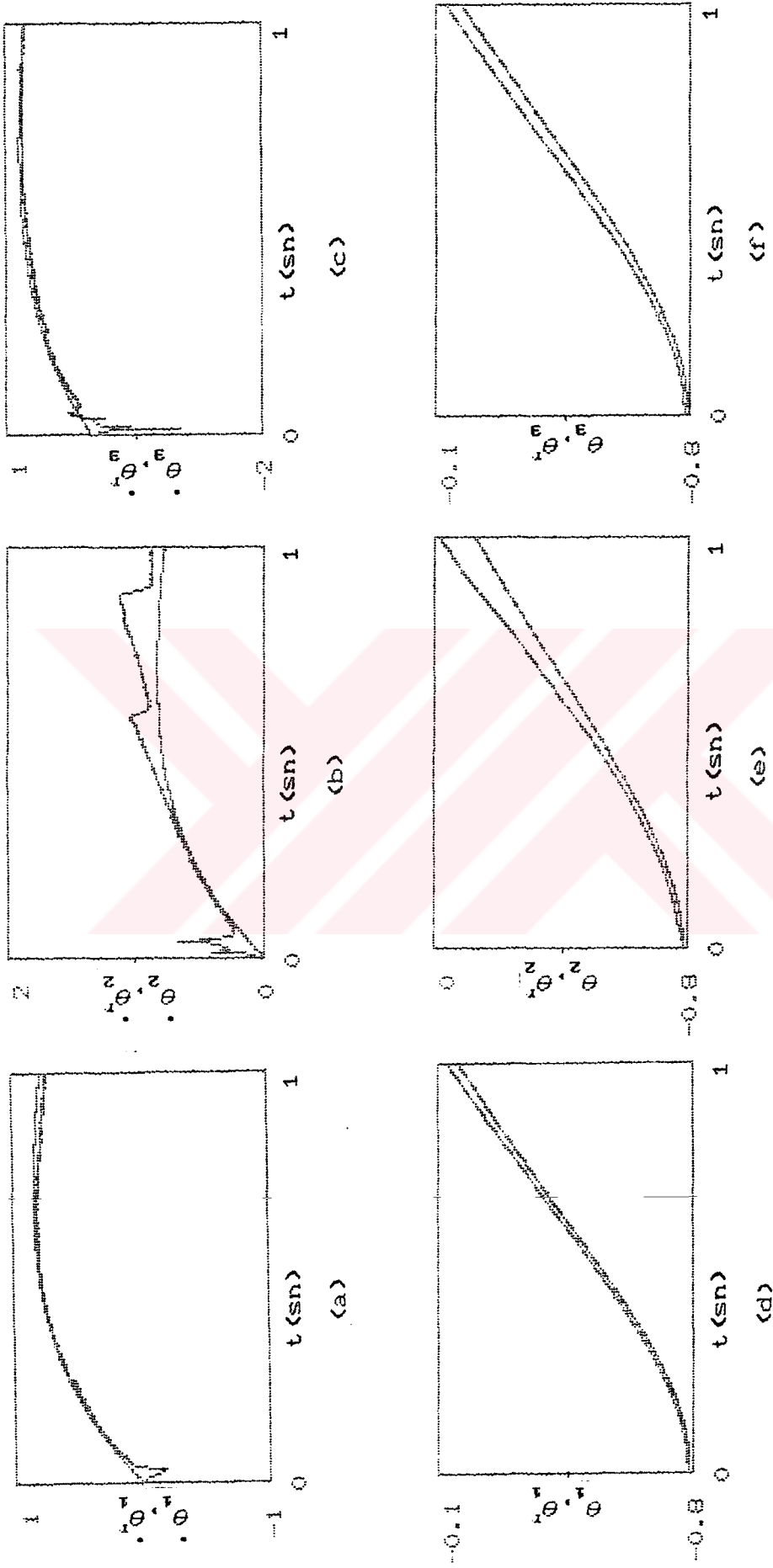
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.5.a. Self-tuning kontrolör (stokastik bozucu, taşınan yük, kübik yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

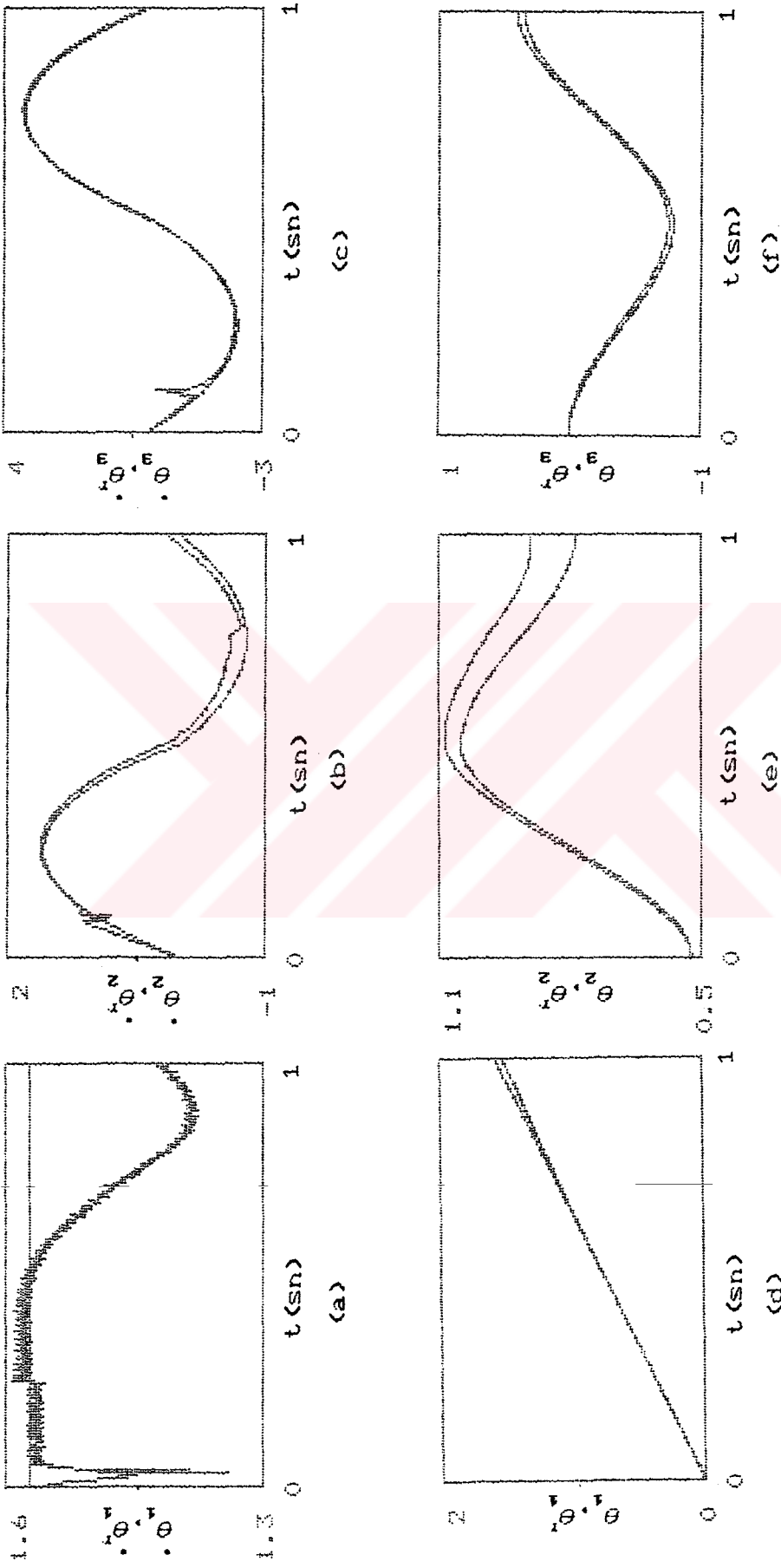
(d), (e), (f): Eklem konum ve referansları [rad]



Sekil 7.5.b. Self-tuning kontrolör (eksponansiyel yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

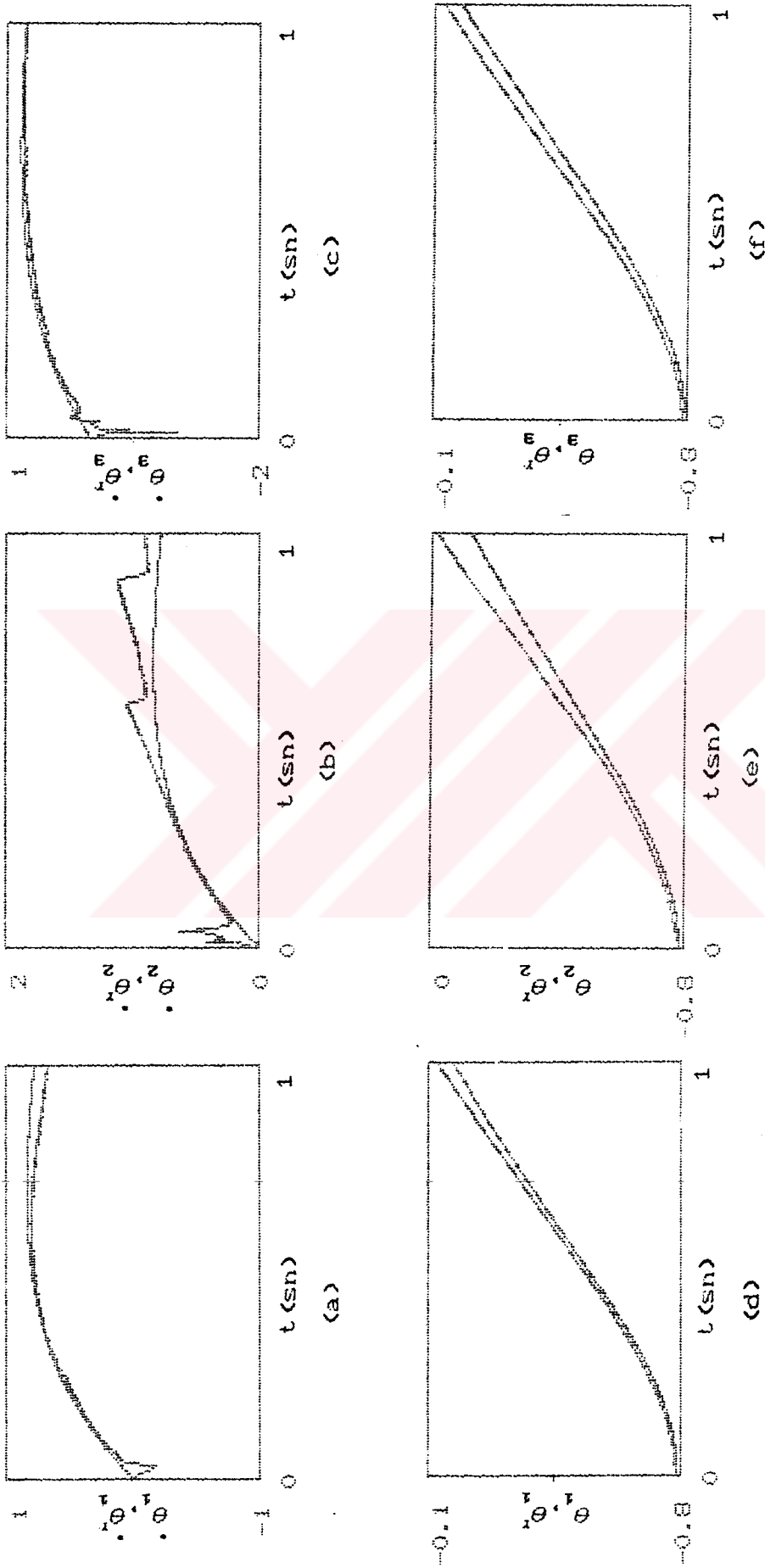
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.6.a. Self-tuning kontrolör (sürtünme, taşıyan yük, stokastik bozucu, kübik yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

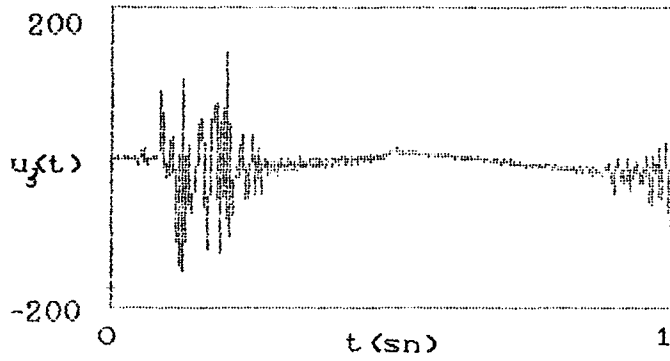
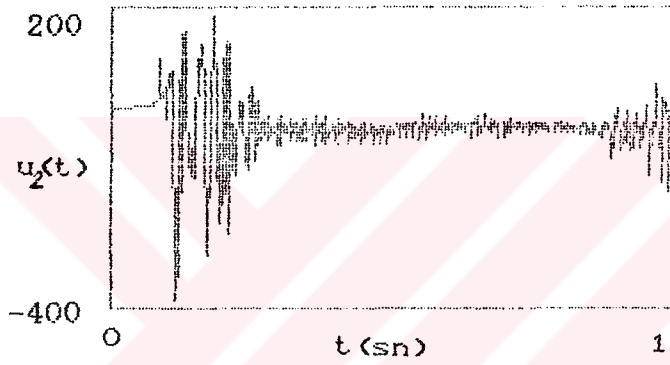
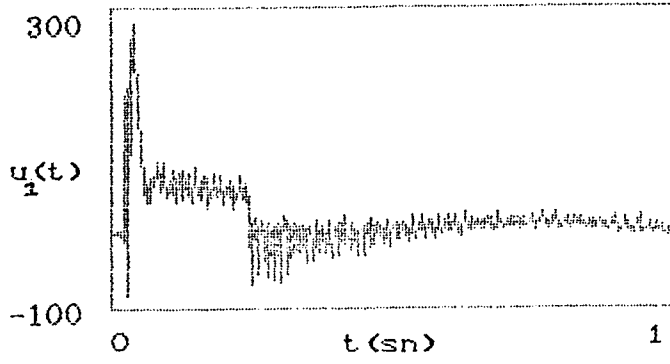
(d), (e), (f): Eklem konum ve referansları [rad]



Sekil 7.6.b. Self-tuning kontrolör (eksponansiyel yörünge)

(a), (b), (c); Eklem hız ve referansları

(d), (e), (f); Eklem konum ve referansları



Şekil 7.6.c. Kendi Kendini Ayarlayan Kontrol SISO durum için u kontrolünün zamanla değişimi (Küçük yörünge, stokastik bozucu, sürtünme, yük değişimi).

7.4 Öngörülü Kontrol Algoritmasının Robotik Bir Manipülatöre Uygulanması

Bu bölümde, şekil 7.1' de gösterilen üç eklemlilik robotik manipülatöre genelleştirilmiş öngörülü kontrol algoritması uygulanacaktır.

Simülasyon uygulamalarında üç farklı kontrolör dizaynı yapılmıştır. İlk bölümde kendi kendini ayarlayan kontrol algoritmasında olduğu gibi her bir eklem için ayrı kontrolör dizaynı (SISO-GPC) düşünülmüş, yani; her bir eklem diğerinden bağımsız olarak kontrol edilmiştir.

İkinci bölümde ise, üç eklem için tek bir kontrolör dizaynı yapılmıştır. (MIMO-GPC) dizaynının uygulandığı bu çalışmada, eklemler arasındaki etkileşimler de göz önüne alınmış olmaktadır.

Üçüncü olarak ise, robot kontrolunda zamansal ve hesapsal yük açısından büyük kolaylıklar getiren, ters kinematik çözümü gerektirmeyen kartezyen uzayda kontrol, öngörülü kontrol yaklaşımı çerçevesinde incelenmiştir.

Öngörülü kontrol algoritmasının performansı, bölüm 7.3.1'de de yapıldığı gibi eklemlerde oluşan sürtünme kuvvetleri, stokastik ölçme hataları, eldeki yükün belirli zaman aralıklarında değişmesi, farklı yörünge gibi senaryolar için incelenmiştir.

7.4.1 SISO Öngörülü Kontrolör Dizaynı

Bu çalışmadan amaçlanan, her bir eklemin ayırık noktalar ile belirlenen istenen bir yolu takip etmesini sağlayacak şekilde, her bir eklem için ayrı bir öngörülü kontrolör dizaynı etmektir. Eklemler arasındaki etkileşimler küçük kabul edilmektedir. Böylece manipülatör

sisteminin CARIMA modelindeki bağı terimleri göz önüne alınmamaktadır. Her bir eklem diğerlerinden bağımsız olarak kontrol edilecektir.

İncelenen üç eklemlili robotun her bir ekleminin hareketini karakterize etmek üzere aşağıdaki CARIMA modeli kullanılmıştır.

$$A(q^{-1}) y(t) = B(q^{-1}) u(t-1) + \xi(t)/\Delta \quad \dots (7.14a)$$

A polinomunun mertebesi n_a , B polinomunun mertebesi n_b alınarak ayırık zamanda verilen CARIMA modeli

$$y_i(t) = \sum_{j=1}^{n_a} a_i^j y_i(t-j) + \sum_{j=0}^{n_b} b_i^j u_i(t-j-1) + e_i(t)$$

$$i = 1, \dots, n \quad \dots (7.14b)$$

şeklinde de yazılabilir. Bilindiği gibi CARIMA modeli için $e(t) = \xi(t)/\Delta$ şeklindedir.

Denklem (7.14a)'daki A ve B polinomunun katsayıları, giriş ve çıkış bilgileri kullanılarak her bir örnekleme periyodunda çevrim içi olarak kestirilmekte ve bu yüzden de kullanılan öngörülü kontrol algoritmaları adaptif karakter taşımaktadır.

Öngörülü kontrolörün dizaynı için, istenen yörünge ve y sistem çıkışı arasındaki hataların kareleri toplamı ve $\lambda(j)$ ile ağırlıklandırılmış kontrol artımlarının kareleri toplamından meydana gelen bir performans indeksi kullanılmaktadır. Yörünge kontrolünde, y çıkışı olarak eklem açısal hızları, u girişi olarak ise eklemlere uygulanacak torklar alınmıştır.

$$J = \sum_{j=1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2$$

$$\dots (7.15)$$

J amaç fonksiyonunu minimize edecek u büyüklükleri, her bir eklem motoruna uygulanacak eklem torklarını verecektir. Kontrol artımları $\Delta u(t)$ ve dolayısı ile $u(t)$ ' lerin $\langle \Delta u(t) = u(t) - u(t-1) \rangle$ hesaplanma şekli bölüm 4.1.4 ve 4.1.5'de detaylı olarak verildiğinden burada tekrar incelenmeyecektir.

(7.14b) ifadesi aşağıdaki kapalı formda da yazılabilir:

$$y_i(t) = \Theta_i^T \phi_i(t-1) + e_i(t), \quad \dots (7.16)$$

Burada;

$$\Theta_i = [b_i^0, \dots, b_i^{n_b}; a_i^1, \dots, a_i^{n_a}]^T$$

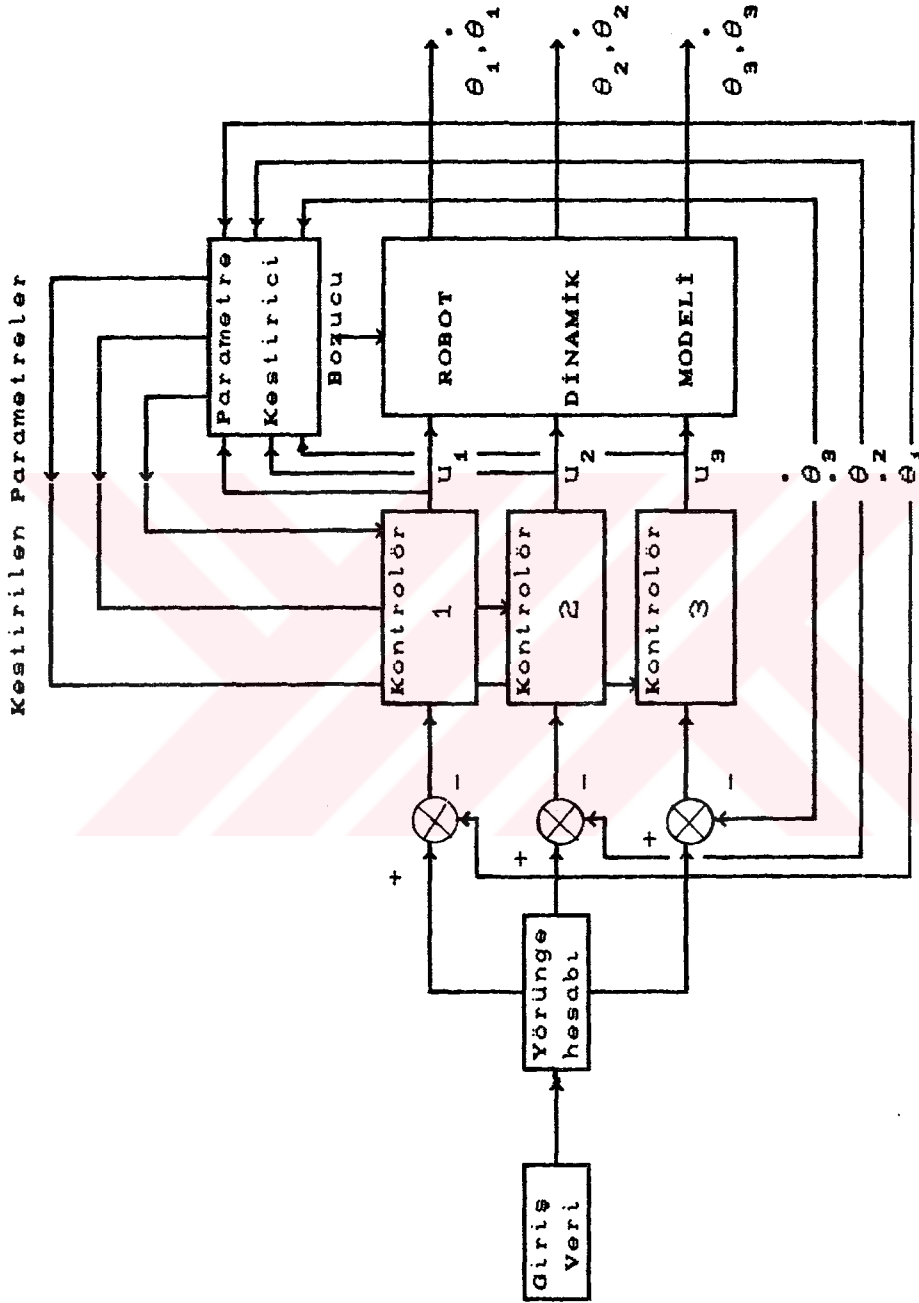
$$\phi_i(t-1) = [u_i(t-1), \dots, u_i(t-n_b); y_i(t-1), \dots, y_i(t-n_a)]^T$$

şeklindedir.

Her bir i eklemi için Θ_i parametreleri en küçük kareler parametre kestirme algoritması kullanılarak kestirilmekte ve bu değerler daha sonra i eklemi için öngörülü bir kontrolör oluşturmada kullanılmaktadır. Her bir eklem için (7.15) nolu J performans kriteri Δu ' ya göre minimize edilmektedir. Bölüm 4.1.4 veya 4.1.5' de çıkarılan,

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (y_r - f)$$

ifadesi kontrol algoritmasının temelini oluşturmaktadır. Yörünge kontrolu için tek giriş-tek çıkışlı öngörülü kontrol algoritması kullanan bir manipülatöre ait blok diyagramı şekil 7.7' de görülmektedir. Aşağıda tek giriş ve tek çıkışlı öngörülü kontrol algoritmasının robota uyarlanması ile ilgili sayısal örnekler verilmiştir.



Sekil 7.7 SISO öngörülü kontrol algoritması kullanan bir manipulator için eklem yörüngesi kontrolü blok diyagramı.

7.4.1.1 Sayısal Örnekler

Kullanılan dinamik denklemler (7.8) ifadesi kullanılarak elde edilmiş ve bu denklemlerdeki sayısal değerler Tablo 7.1' den alınmıştır. Elde edilen dinamik denklemler, sayısal simülasyon için kullanılmıştır.

Simülasyon çalışmalarında, ikinci mertebeden non-linear diferansiyel denklem takımı olan robot eklemlerinin hareketine ait matematik modeli karakterize edebilmek için CARIMA modelindeki A polinomunun derecesi $n_a = 1$, B polinomunun derecesi ise $n_b = 2$ olarak alınmıştır. Giriş u_i 'ler, eklem motoruna uygulanacak torkları ve çıkış y_i 'ler, eklemlerin açısal hızlarını ($\dot{\theta}_i$) göstermek üzere, CARIMA modeli;

$$y_i(t) = a_i^1 y_i(t-1) + b_i^0 u_i(t-1) + b_i^1 u_i(t-2) + b_i^2 u_i(t-3) + e_i(t) \quad \dots (7.17)$$

şeklinde alınabilir. Kestirilen parametreler ise;

$$\theta_i = [b_i^0, b_i^1, b_i^2, a_i^1] \quad \dots (7.18)$$

şeklindedir.

(7.15) performans indeksini minimize eden ve i eklemine uygulanacak kontrol büyüklüğü olan $u_i(t)$ ' nin ilk elemanı

$$u_i(t) = u_i(t-1) + g_1^T (y_{i,r} - f) \quad \dots (7.19)$$

olarak hesaplanmaktadır. Burada g_1^T , $(G_1^T G_1 + \lambda I)^{-1} G_1^T$ 'nin ilk satırını oluşturmaktadır.

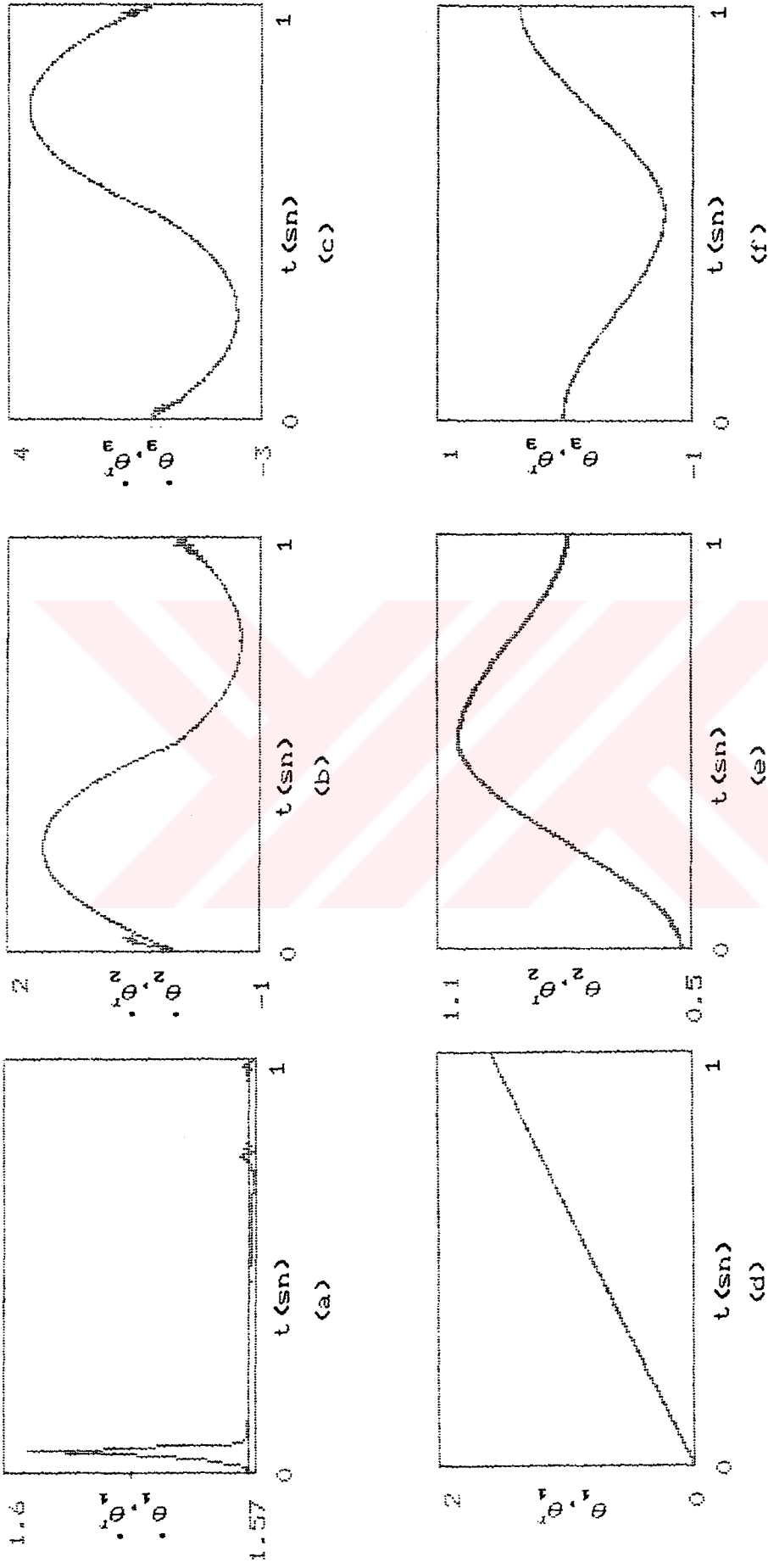
Yukarıda tarif edilen kontrolör, sayısal simülasyon çalışmaları ile test edilmiştir. Referans yörüngeler

olarak bölüm 7.2.2' de verilen eklem koordinatlarında ifade edilmiş yörüngeler kullanılmıştır.

Genelleştirilmiş öngörülü kontrol algoritmasının ayar parametreleri olan N_2 ve N_u simülasyon çalışmalarından elde edilen sonuçlara dayanarak, sırasıyla 3 ve 1 olarak alınmıştır. Ağırlık faktörü λ , her bir eklem için 0.000001 olarak alınmıştır. Parametre kestirme algoritmasındaki unutma faktörü μ , yine her bir eklem için 0.95 olarak kabul edilmiştir. Parametrelerin başlangıç değerleri $b_i^0 = 1$, $b_i^1 = b_i^2 = a_i^1 = 0$ ve $P_i = 10^5(I)$ seçilmiştir. Diferansiyel denklem çözümünde kullanılan Runge-Kutta integrasyon adımı 0.0025 sn. olarak alınmıştır [15].

Kullanılan kontrol algoritmasının performans araştırması için bölüm 7.3.1' deki referans yörüngeleri ve senaryolar kullanılmıştır. Hiç bir bozucu büyüklüğün göz önüne alınmadığı durum için simülasyon çalışmaları şekil 7.8' de, yük değişiminin ve ölçme hatalarının etkilerini gösteren durum için sonuçlar şekil 7.9'da, sürtünmenin etkilerini gösteren çalışmanın sonuçları ise şekil 7.10' da verilmiştir. Her üç senaryo için eklem hız hataları kareleri toplamı tablo 7.2'de, eklem final konum hataları ise tablo 7.3'de gösterilmiştir. Sürtünme etkilerinin de göz önüne alındığı senaryo için u kontrol değerlerinin zamanla değişimi şekil 7.10.c' de verilmiştir.

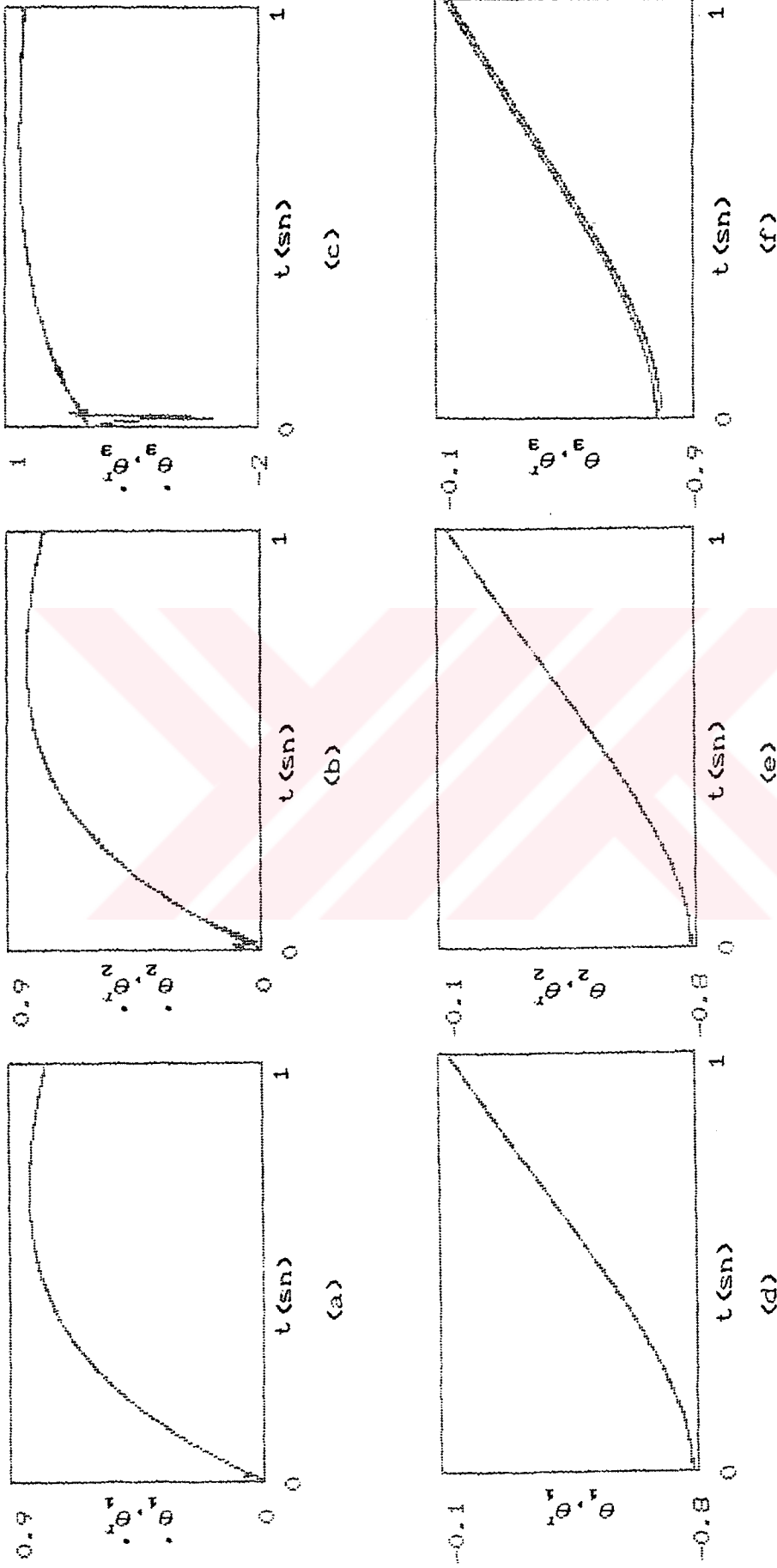
Simülasyon çalışmaları ayrıca $n_a=1$ ve $n_b=1$ alınarak da yapılmış ve model mertebesinin yanlış tanımlanmasının öngörülü kontrol algoritmasında bir sorun yaratmayacağı gösterilmeye çalışılmıştır, inceleme sadece kübik yörünge için yapılmıştır (Şekil 7.11). Bu sonuca dayanarak , kestirilecek parametre sayısının fazla olduğu MIMO sistemlerde $n_a=1$ ve $n_b=1$ alınması yeterli görülmektedir.



Şekil 7.8.a. Öngörülü kontrolör (bozucusuz durum, kübik yörünge)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

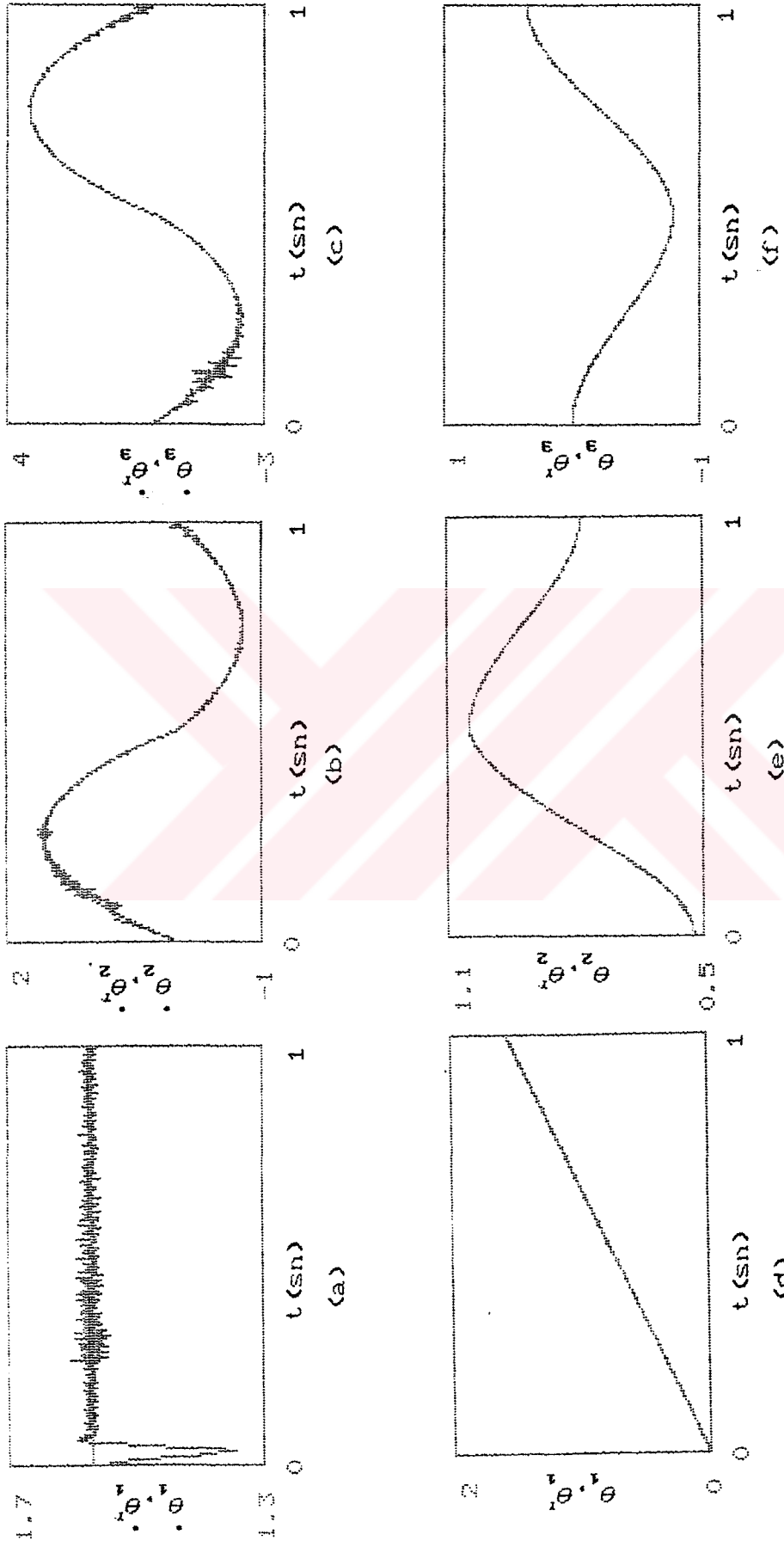
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.8.b. Öngörülü kontrolör, (eksponansiyel yörünge (SISO))

(a), (b), (c); Eklem hız ve referansları [rad/snl]

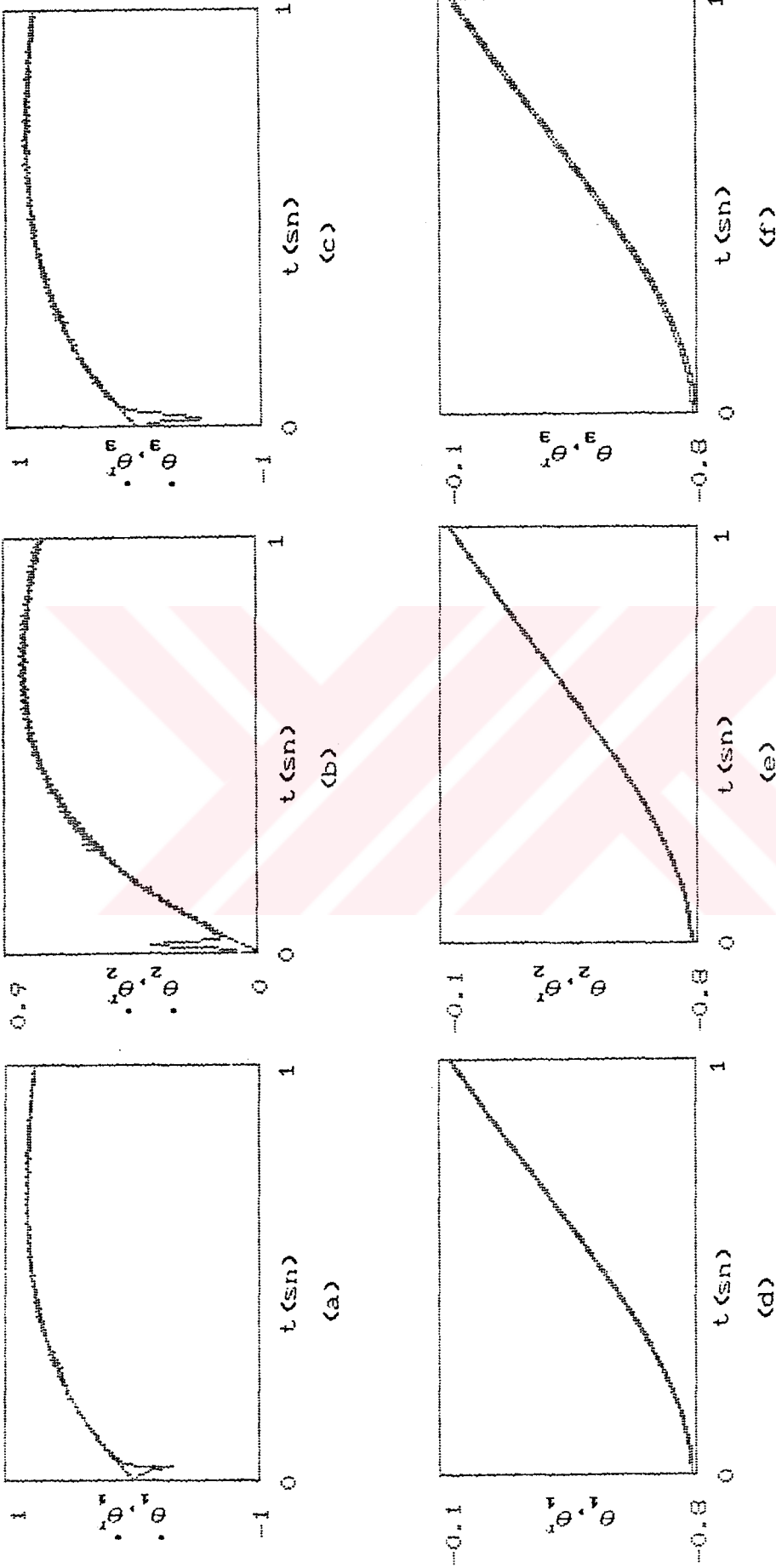
(d), (e), (f); Eklem konum ve referansları [rad]



Şekil 7.9.a. Öngörüllü kontrolör, (stokastik bozucu, taşınan yük, kübik yörünge (SISO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

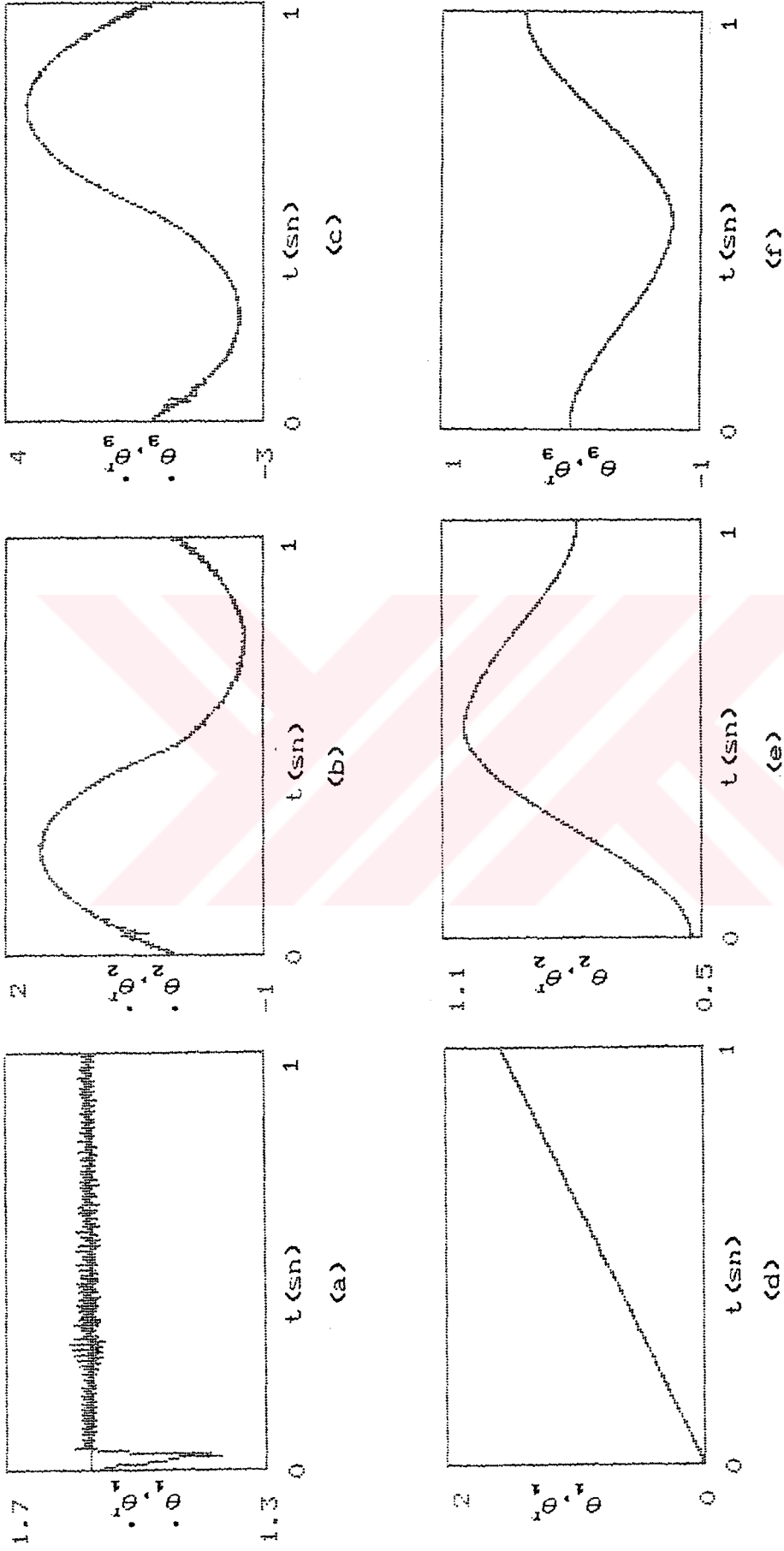
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.9.b. Öngörülü kontrolör, (eksponansiyel yörünge (SISO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

(d), (e), (f): Eklem konum ve referansları [rad]

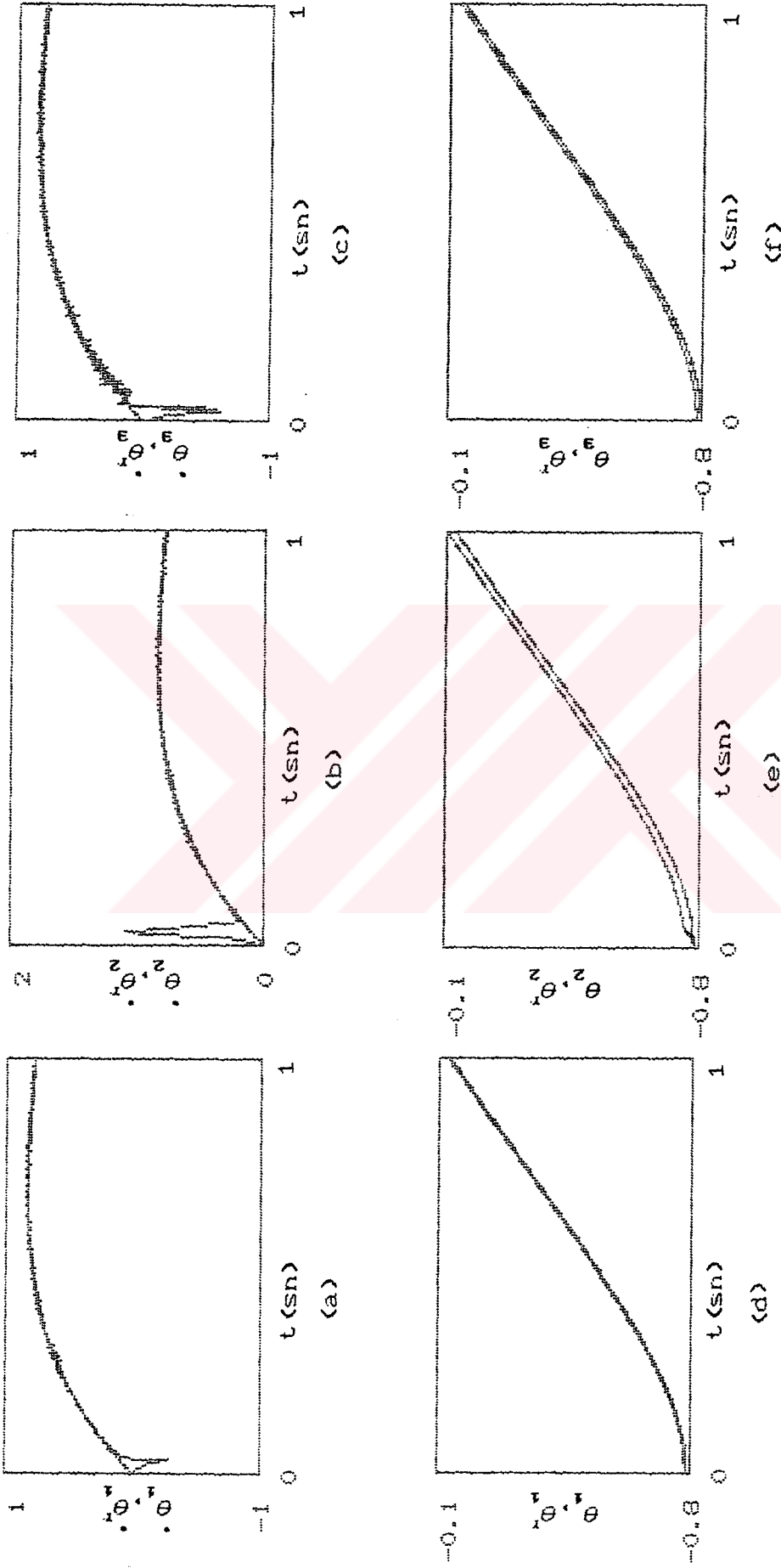


Şekil 7.10. a. Öngörülü kontrolör, (sürtünme, taşınan yük, stokastik

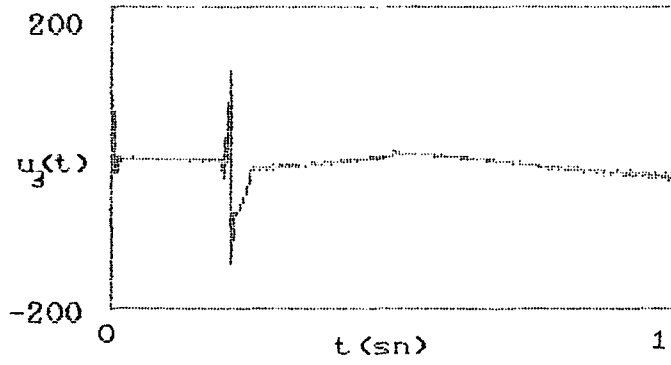
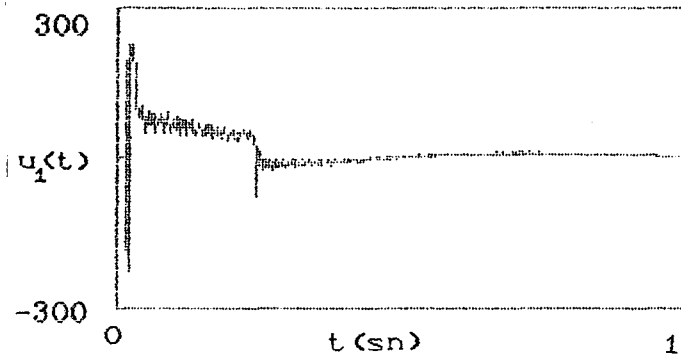
bozucu, kübik yörünge (SISO)

(a), (b), (c); Eklemler hız ve referansları [rad/sn]

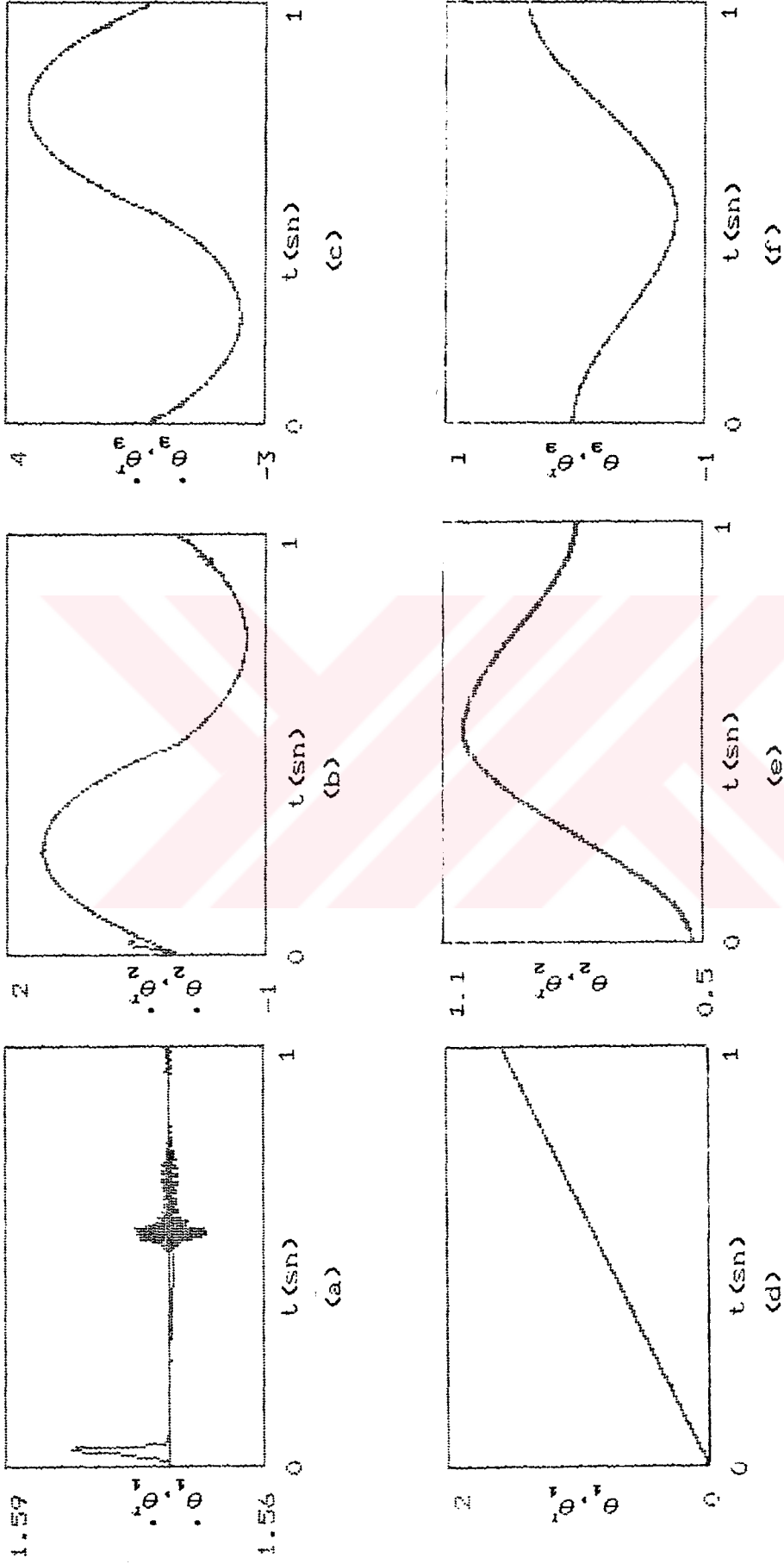
(d), (e), (f); Eklemler konum ve referansları [rad]



Şekil 7.10.b. Öngörülü kontrolör, (eksponansiyel yörünge (SISO))
 (a), (b), (c): Eklemler hız ve referansları [rad/sn]
 (d), (e), (f): Eklemler konum ve referansları [rad]



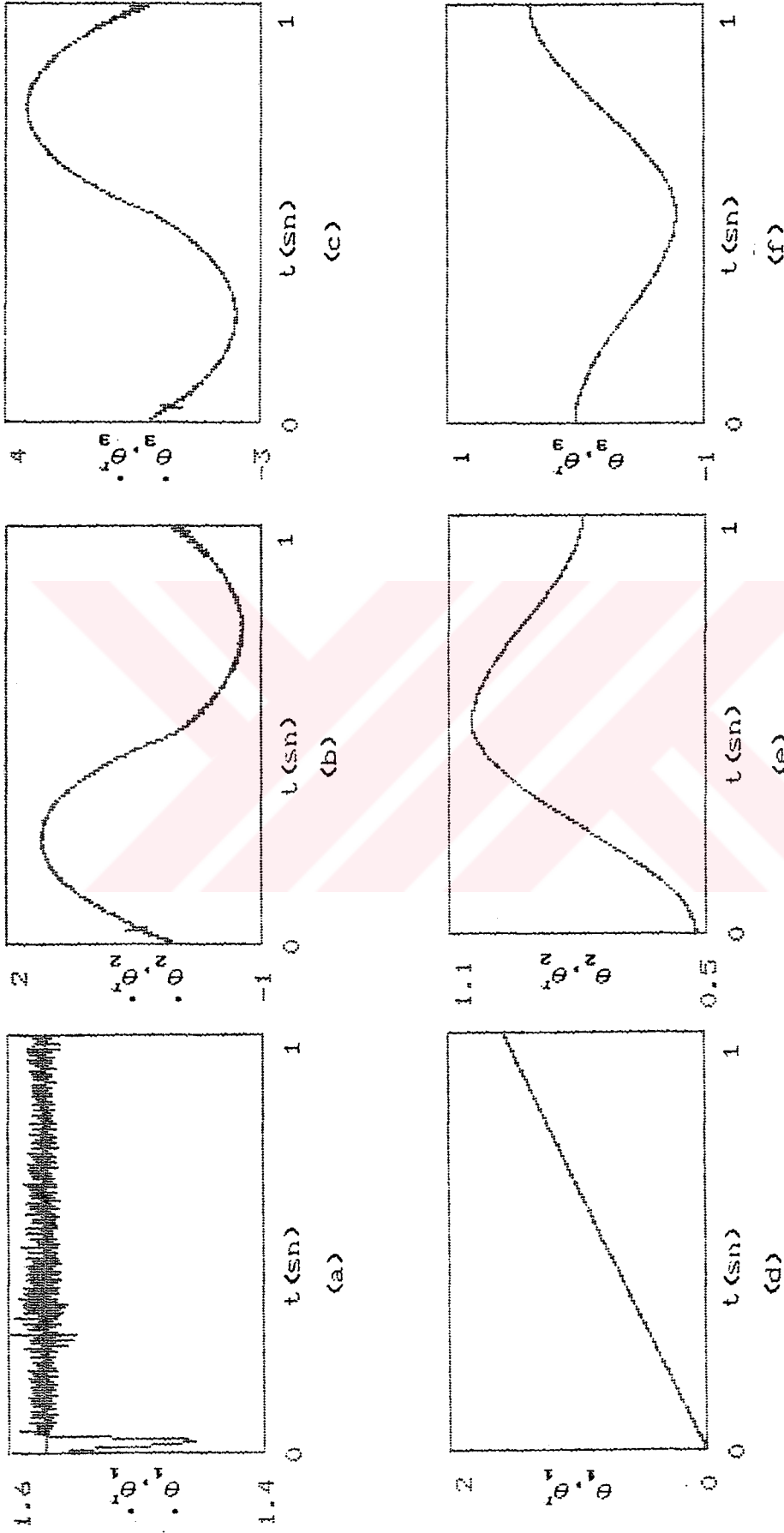
Şekil 7.10.c. Öngörülü Kontrol SISO durum için u kontrolünün zamanla değişimi (Kübik yörünge, stokastik bozucu, sürtünme, yük değişimi).



Şekil 7.11.a. Öngörülü kontrolör, (bozucusuz durum, hatalı model mertebesi, kübik yörünge (SISO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

(d), (e), (f): Eklem konum ve referansları [rad]

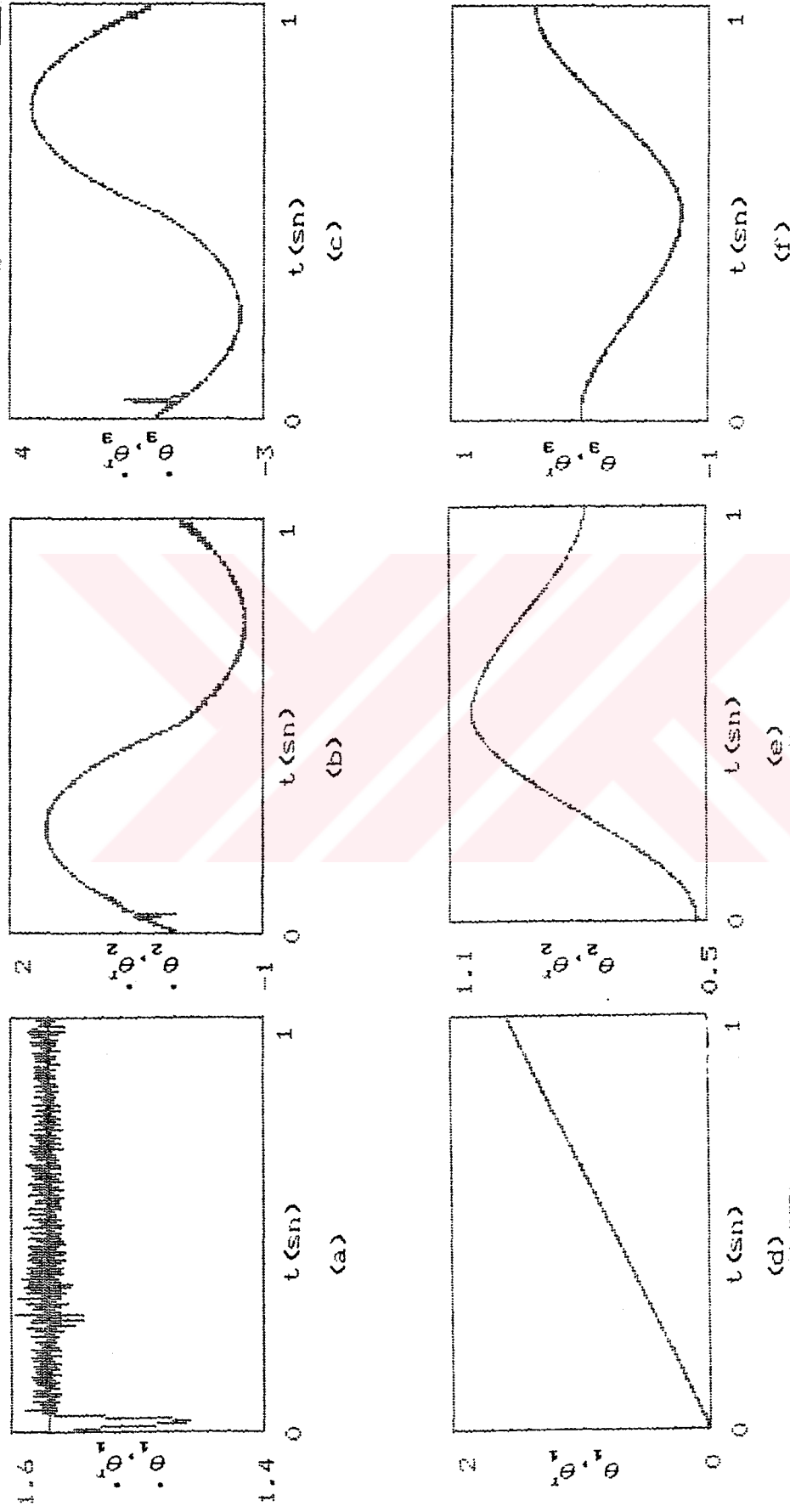


Sekil 7.11.b. Öngörülü kontrolör, (stokastik bozucu, taşınan yük, hatalı

model mertebesi, kübik yörünge (SISO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

(d), (e), (f): Eklem konum ve referansları [rad]



Sekil 7.11.c. Öngörülü kontrolör, (sürtünme, taşınan yük, stokastik bozucu, hatalı model mertebesi (SISO))
 (a), (b), (c): Eklem hız ve referansları [rad/sn]
 (d), (e), (f): Eklem konum ve referansları [rad]

7.4.2 MIMO Öngörülü Kontrolör Dizaynı

Çok değişkenli genelleştirilmiş öngörülü kontrolör dizaynında, eklemler arasındaki dinamik etkileşim de artık göz önüne alınmaktadır. Bu bölümde hedeflenen, yine manipülatör eklemlerinin istenen bir yörüngeyi mümkün olduğunca yakından izlemesini sağlayacak şekilde, tek bir kontrolör dizayn etmektir.

Problem, eklemlerin hareketi için (7.14a) CARIMA modeli kullanılarak çözülmüştür. $A_j (j=1, \dots, n_a)$ ve $B_j (j=0, \dots, n_b)$ polinom matrislerinin her biri 3×3 boyutlu olmakta ve kestirilecek 9 parametre içermektedir. Sistem çıkışı olarak kabul edilen $y(t)$ açısal eklemler hızları vektörü $y(t) = [y_1(t), y_2(t), y_3(t)]$, sistem girişi olan $U(t)$ eklemler torkları ise $u(t) = [u_1(t), u_2(t), u_3(t)]$ olarak ifade edilmektedir.

Simülasyon çalışmalarında CARIMA modeli; $n_a=1$, $n_b=1$ olacak şekilde birinci mertebeden seçilmiştir. Buna neden, çok giriş-çok çıkışlı sistemlerde kestirilecek parametre sayısı artacağından aynı performansın $n_a=1$ ve $n_b=1$ alınarak da elde edilip edilemeyeceğinin araştırılmasıdır. Örneğin; $n_a=1$ ve $n_b=1$ için model aşağıdaki formu alacaktır.

$$y(t) = A_1 y(t-1) + B_0 u(t-1) + B_1 u(t-2) + e(t) \quad \dots (7.20)$$

$e(t)$, eklemler konumlarındaki hata vektörünü göstermektedir. Denklem (7.20)' den de yararlanarak parametre matrisi Θ ve veri vektörü ϕ aşağıdaki gibi ifade edilmektedir.

$$\Theta = \begin{bmatrix} b_{11}^0 & b_{12}^0 & b_{13}^0 & b_{11}^1 & b_{12}^1 & b_{13}^1 & a_{11}^1 & a_{12}^1 & a_{13}^1 \\ b_{21}^0 & b_{22}^0 & b_{23}^0 & b_{21}^1 & b_{22}^1 & b_{23}^1 & a_{21}^1 & a_{22}^1 & a_{23}^1 \\ b_{31}^0 & b_{32}^0 & b_{33}^0 & b_{31}^1 & b_{32}^1 & b_{33}^1 & a_{31}^1 & a_{32}^1 & a_{33}^1 \end{bmatrix}^T$$

... (7.21)

$$\phi(t-1) = [u_1(t-1), \dots, u_3(t-1); u_1(t-2), \dots, u_3(t-2);$$

$$y_1(t-1), \dots, y_3(t-1)]^T \dots (7.22)$$

Θ 'nin boyutu (9×3) 'dür ve kestirilecek 27 parametre içermektedir. $n_a=2$ ve $n_b=1$ olarak alındığında ise $(9 \times 4) = 36$ parametrenin kestirilmesi gerekmektedir. Bu da parametre kestirimi her bir adımda yapıldığından hesapsal zorluklar getirecek ve yakınsama problemlerine neden olacaktır.

Performans kriteri aşağıdaki şekilde seçilmiştir:

$$J(\Delta u) = \sum_{j=1}^{N_2} \left\{ [y(t+j) - y_r(t+j)]^2 \right\} + \Lambda \sum_{j=1}^{N_u} \Delta u(t+j-1)^2$$

... (7.23)

Burada $y_r(t+j) = [y_{1r}(t+j), \dots, y_{3r}(t+j)]^T$ şeklinde her bir ekleme ait açısal hız referanslarını göstermektedir.

$$\Lambda = \text{diag} [\Lambda(1), \Lambda(2), \dots, \Lambda(N_{21})]$$

şeklinde. $\Lambda(1)$ 'ler SISO durumdaki λ ağırlık faktörüne karşılık gelmektedir.

(7.23) ifadesi minimize edilerek, eklemlere uygulanacak torkları veren sistem giriş vektörü $\Delta u(t)$ elde edilir.

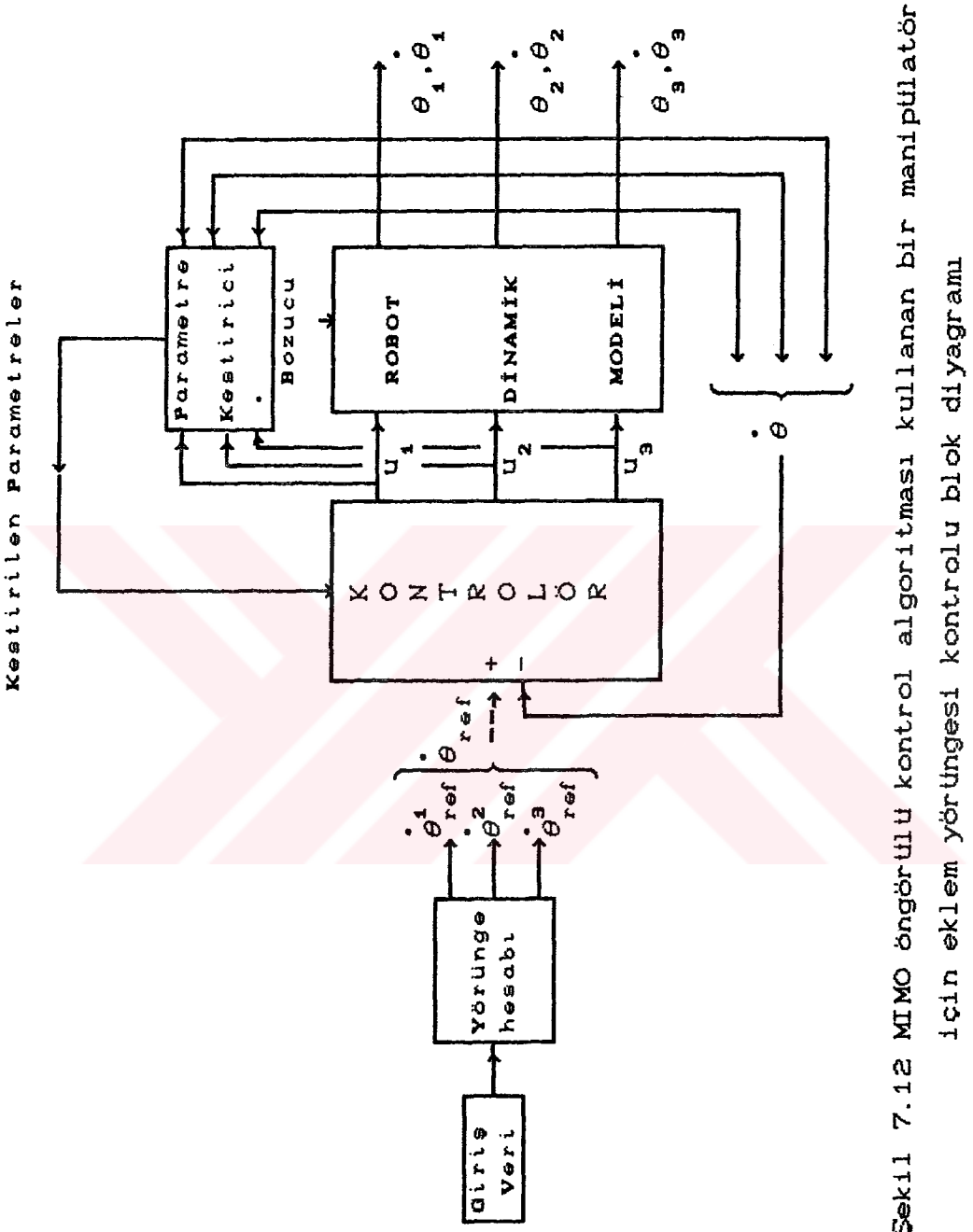
$$\Delta u(t) = [I \ 0 \ 0] (G^T G + \Lambda)^{-1} G^T (y_r - f) \dots \quad (7.24)$$

Robota uygulanan MIMO öngörülü kontrol algoritmasının blok diyagramı şekil 7.12'de gösterilmiştir.

7.4.2.1 Sayısal Örnekler

MIMO öngörülü kontrolör dizaynı ile ilgili olarak yapılan simülasyon çalışmaları, bölüm 7.4.1.1' deki aynı robot ve ayar parametreleri kullanılarak gerçekleştirilmiştir. Yapılan simülasyon çalışmaları neticesinde MIMO durum için parametre kestirme algoritması olarak kare kök algoritmasının (SQR) daha etkili sonuçlar verdiği gözlenmiş, bu nedenle de bu bölümdeki simülasyon çalışmalarında SQR algoritması kullanılmıştır. Ayırık zaman modelinin mertebeleri kestirilecek parametre sayısını azaltmak ve hesaplama gücünü arttırmak maksadı ile $n_a=1$ ve $n_b=1$ olarak seçilmiştir. Parametre kestiriminde, parametrelerin başlangıç değerleri $A_1 = 0$, $B_0 = I$, $B_1 = 0$ olarak alınmıştır. Referans yörüngeleri olarak bölüm 7.2.2 de verilen kübik ve eksponansiyel yörüngeler kullanılmıştır.

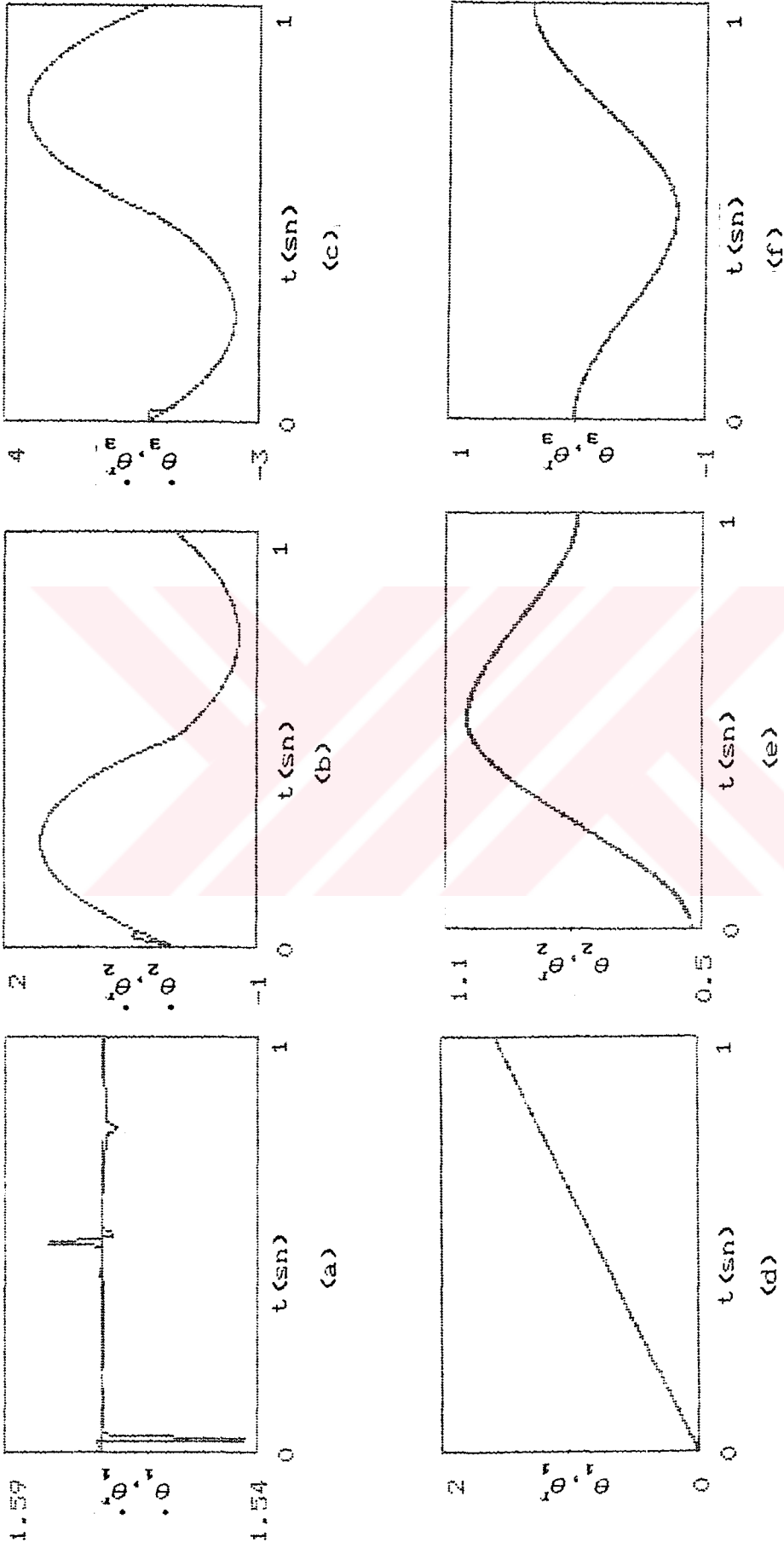
Kullanılan kontrol algoritmasının performans araştırması için bölüm 7.3.1' deki senaryolar aynen kullanılmıştır. Hiç bir bozucu büyüklüğün göz önüne alınmadığı durum ile ilgili simülasyon çalışması için sonuçlar, şekil 7.13 ve tablo 7.2 ve tablo 7.3'de, yük değişiminin ve ölçme hatalarının etkilerini gösteren sonuçlar, şekil 7.14 ve tablo 7.2 ve 7.3'de, sürtünme etkilerinin katıldığı çalışmanın sonuçları ise şekil 7.15 ve tablo 7.2 ve 7.3'de verilmiştir. Bu son senaryo için u kontrol değerlerinin değişimi de şekil 7.15.c'de verilmiştir. Ayrıca sürtünme etkilerinin göz önüne alındığı son çalışmada, integrasyon adımı $\Delta t = 0.001$



alınarak tekrar inceleme yapılmış ve ortaya çıkan çok ufak da olsa kalıcı hataların integrasyon adımını daha küçük seçerek ortadan kaldırılabilceği gösterilmiştir (Şekil 7.15.d). Fakat $\Delta t = 0.0025$ integrasyon adımı ile yapılan çalışmaların da gayet iyi sonuçlar verdiğini göz ardı etmemek gerekir. Gerçek zaman uygulamalarında $\Delta t = 0.0025$ seçimi kestirimin rahat yapılabilmesi açısından da uygun olmaktadır.

7.5 Kartezyen Uzayda Öngörülü Kontrolör Dizaynı

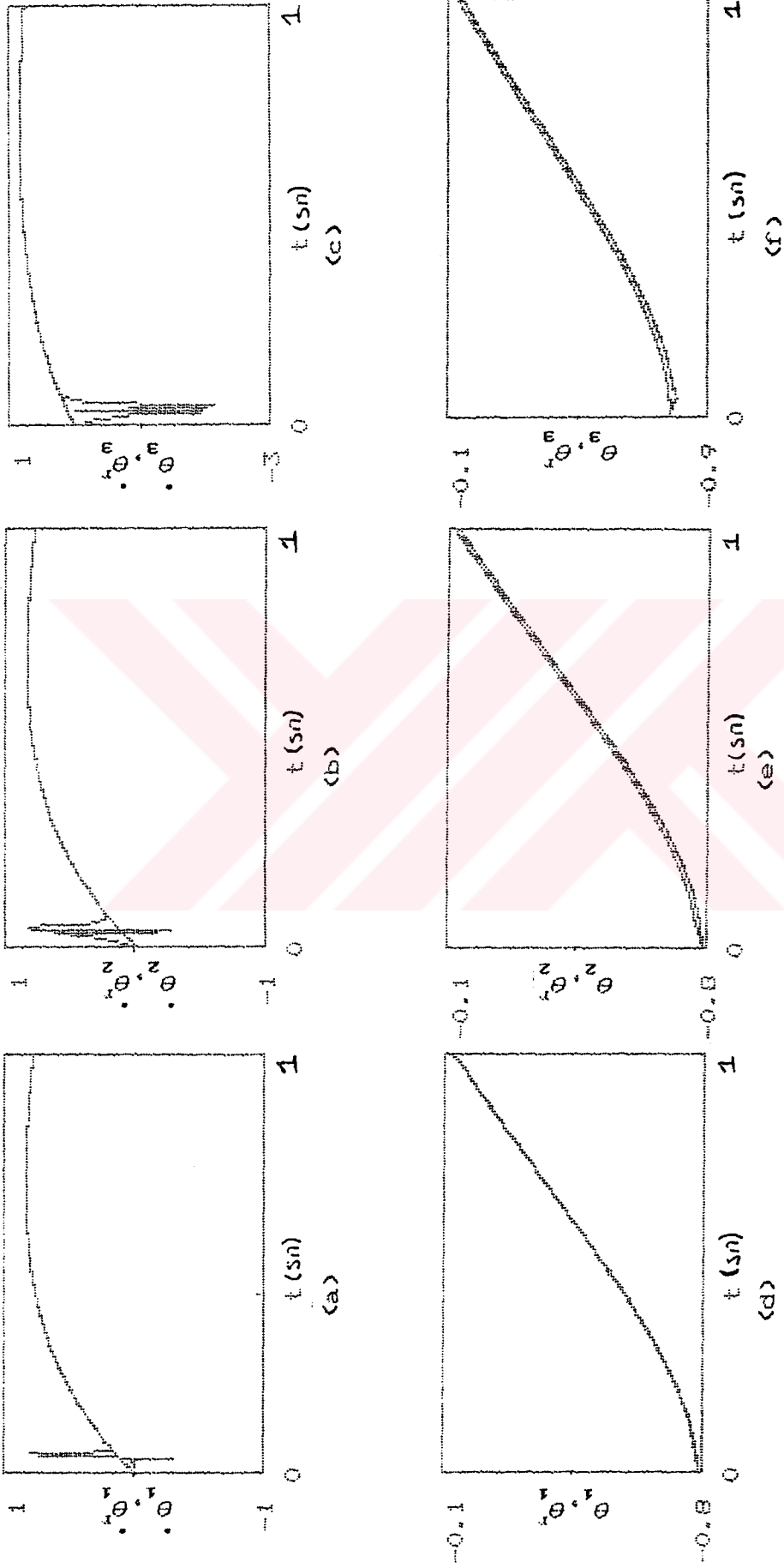
Robotik manipülatörlerin kontrolü, genellikle istenen ve gerçekleştirilen açısal pozisyonlar arasındaki farklarla orantılı olan hata sinyalleri ile aktüatörler tahrik edilerek, eklemlerin açısal pozisyonlarının kontrol edilmesi yolu ile gerçekleştirilmektedir. Bir robotik manipülatörde esas olarak tutucunun (gripper) veya uç elemanın taşıdığı yük veya takip ettiği yol önemli olduğundan, kontrolün sağlanması için kartezyen koordinat sisteminde (x,y,z) cinsinden tanımlanan tutucu veya uç elemanın pozisyonunun, eklemlerin istenen açısal pozisyonlarına dönüştürülmesi gerekmektedir. Bu işlem, trigonometrik fonksiyonlar içeren koordinat dönüşümlerinden meydana gelen ters kinematik denklemlerinin sayısal olarak çözülmesi ile yapılmaktadır. Bu çözüm tek değildir, manipülatörün tutucu veya uç elemanının tek bir (x,y,z) kartezyen koordinatına karşılık, birden fazla eklemler açılarını karşılık gelmektedir (farklı konfigürasyonlar mevcuttur). Şekil 7.16. Ancak bir takım kısıtlamalar ile çözüm sayısı azaltılabilmektedir. Koordinat dönüşümlerinin gerektirdiği hesaplamalar, genellikle manipülatör hareket halinde iken bir çok kez tekrarlanmaktadır, bu da kontrolü önemli ölçüde komplike hale getirmektedir. Bu yüzden manipülatörü direkt olarak, kartezyen koordinat sisteminde kontrol etme imkanı bulunursa, hesaplamalar büyük ölçüde azalacaktır.



Şekil 7.13.a. Öngörülü kontrolör, (bozucusuz durum, kübik yörünge (MIMO)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

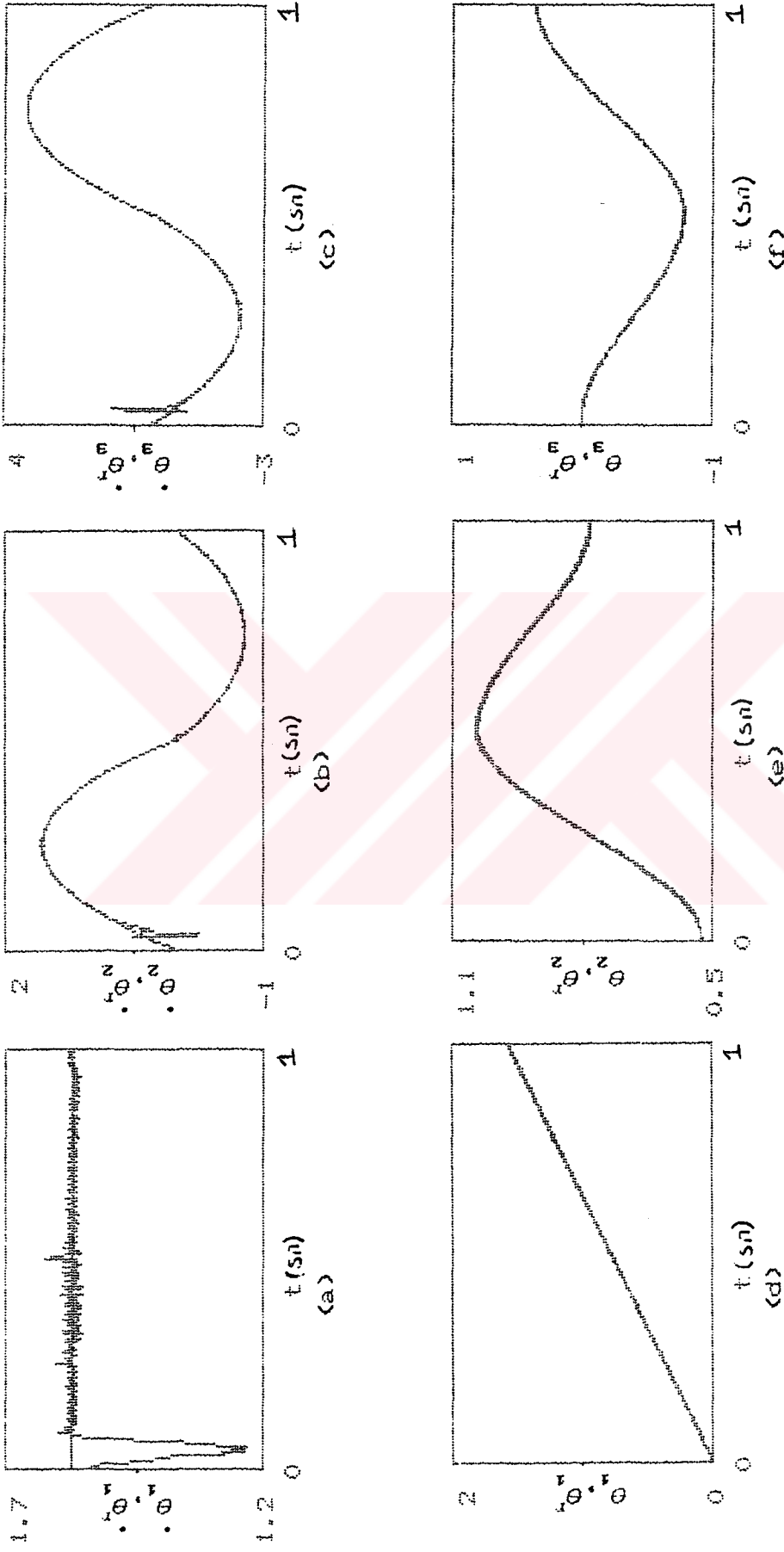
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.13.b. Öngörülü kontrolör, (eksponansiyel yörünge (MIMO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

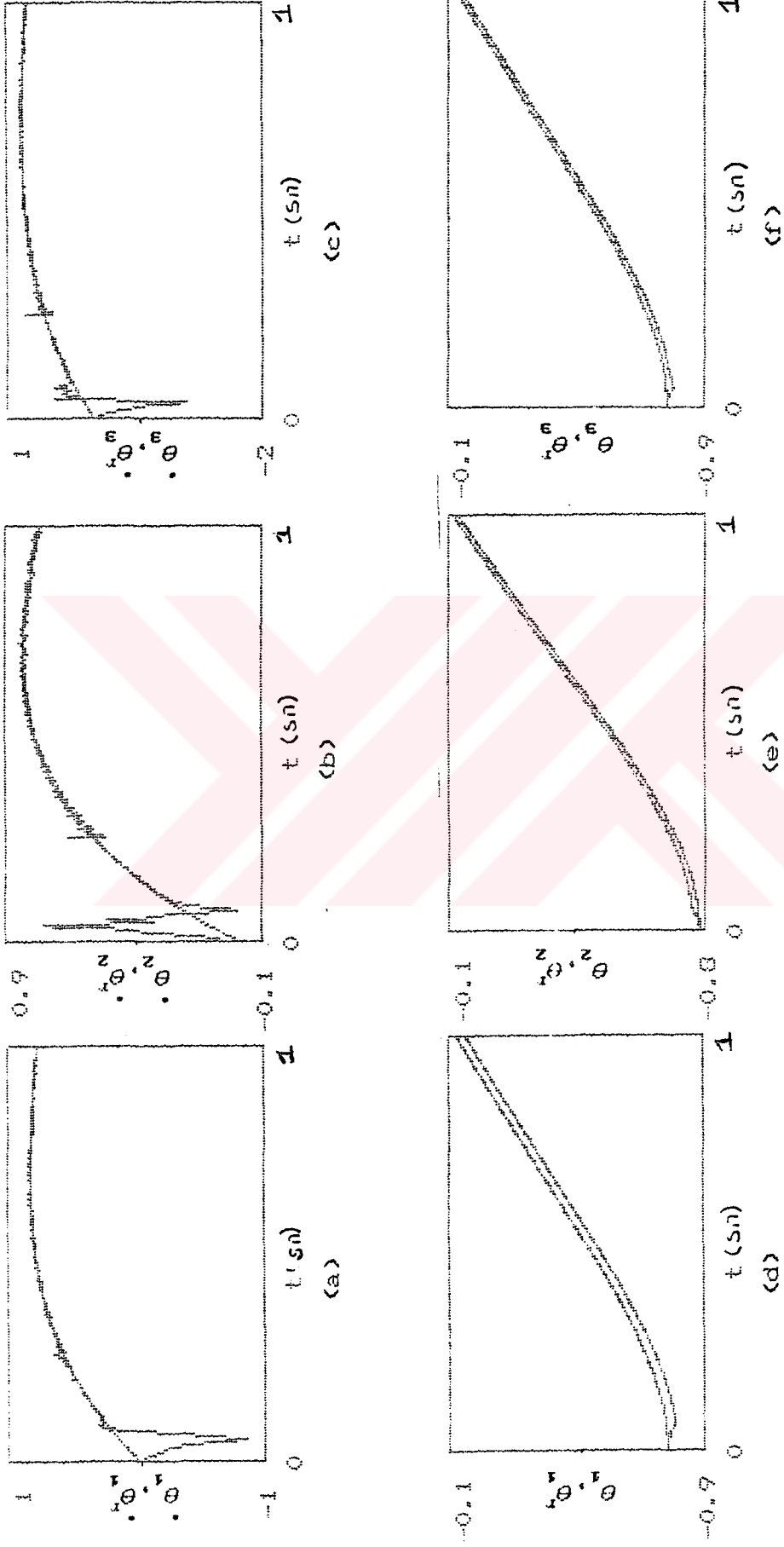
(d), (e), (f): Eklem konum ve referansları [rad]



Sekil 7.14.a. Öngörülü kontrolör, (stokastik bozucu, taşınan yük, kübik) yörünge (MIMO)

(a), (b), (c): Eklem hız ve referansları [rad/sn]

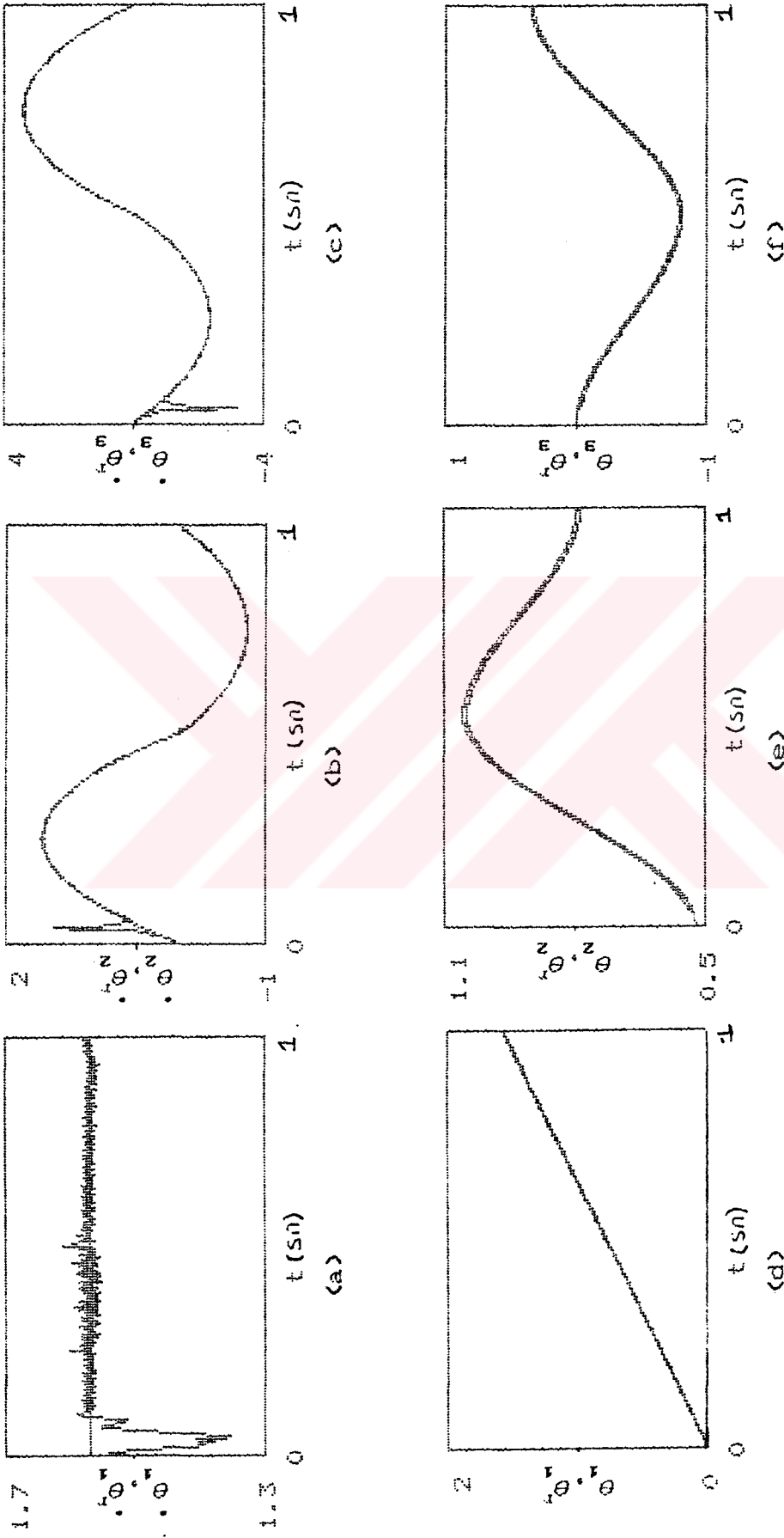
(d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.14.b. Öngörülü kontrolör, (eksponansiyel yörünge (MIMO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

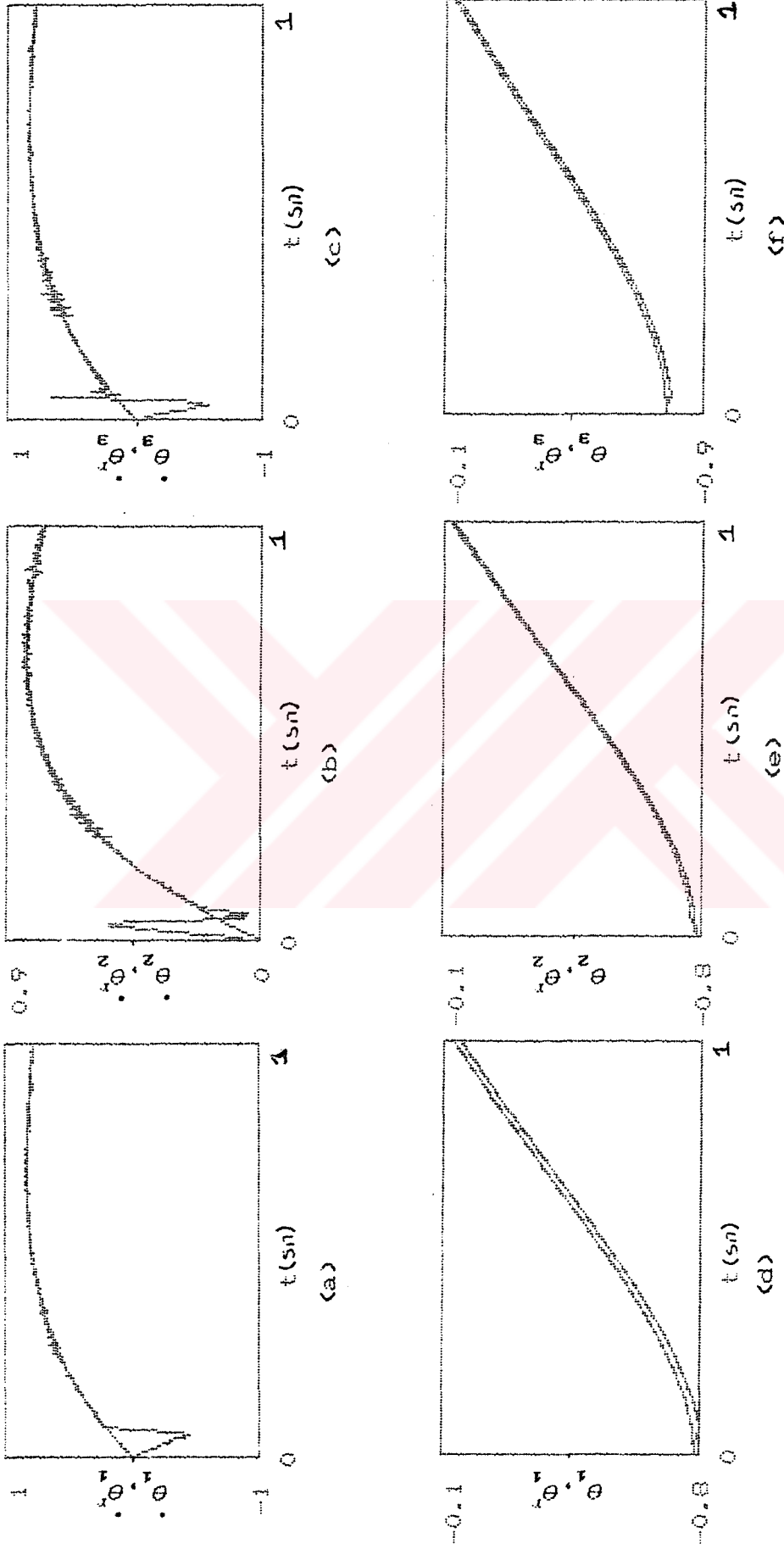
(d), (e), (f): Eklem konum ve referansları [rad]



Sekil 7.15.a. Öngörülü kontrolör, (stokastik bozucu, taşınan yük, stirtünme,küçük yörünge (MIMO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

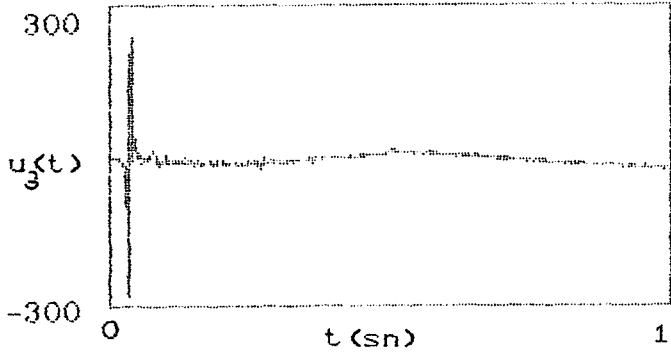
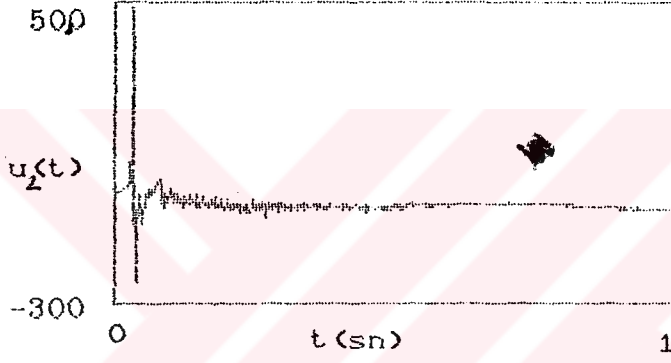
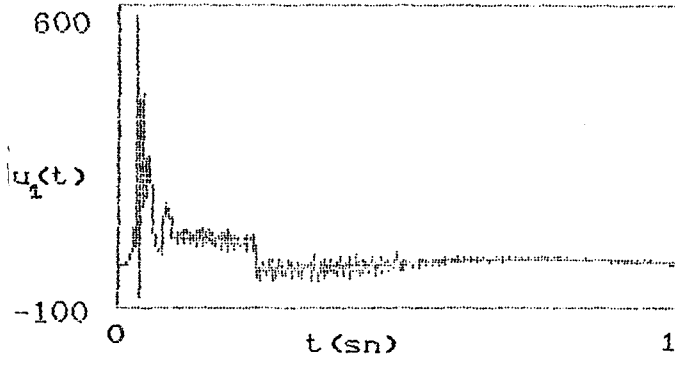
(d), (e), (f): Eklem konum ve referansları [rad]



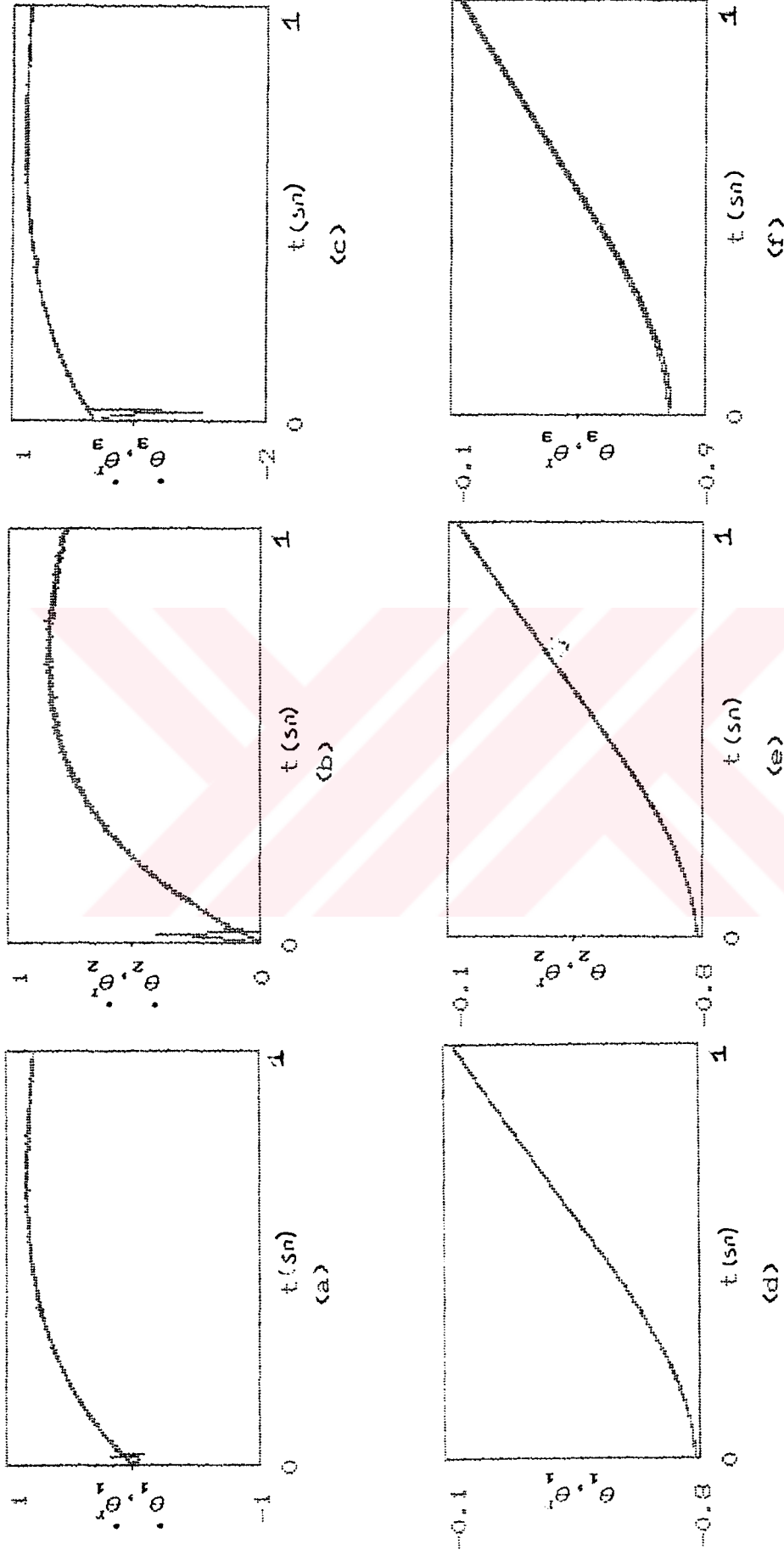
Şekil 7.15.b. Öngörülü kontrolör, (ekspansiyonel yörünge (MIMO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]

(d), (e), (f): Eklem konum ve referansları [rad]

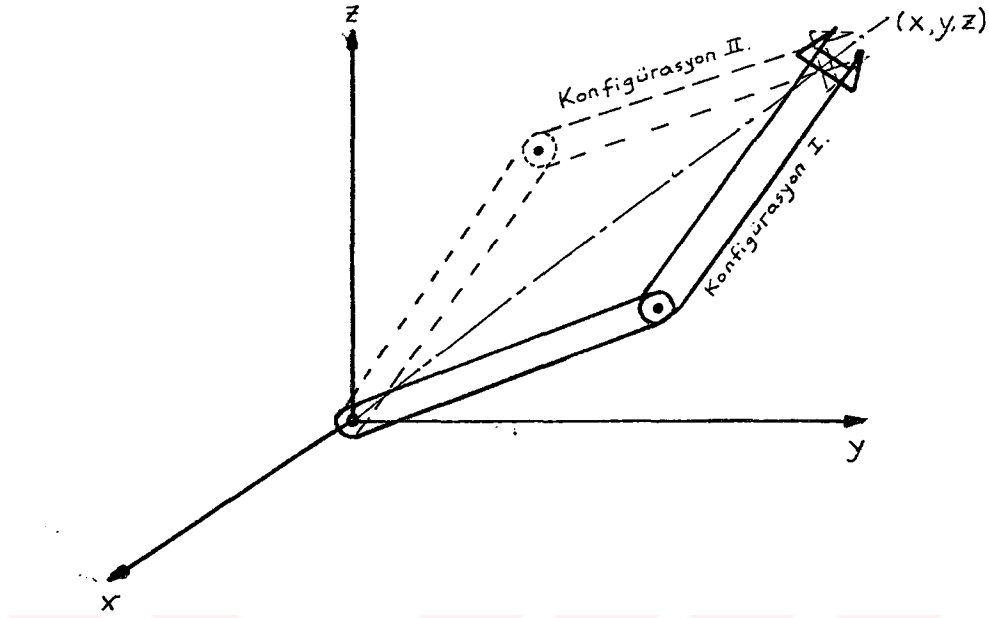


Şekil 7.15.c Öngörülü Kontrol MIMO durum için u kontrolünün zamanla değişimi (Küçük yörünge, stokastik bozucu, sürtünme, yük değişimi).



Şekil 7.15.d. Öngörülü kontrolör, (ekspansiyonlu yörünge, $\Delta t = 0.001$ (MIMO))

(a), (b), (c): Eklem hız ve referansları [rad/sn]
 (d), (e), (f): Eklem konum ve referansları [rad]



Şekil 7.16 Uç elemanın pozisyonu ve farklı konfigürasyonlar

Kartezyen koordinatlarda manipülatörün tutucusunun veya uç elemanın hareketini kontrol edebilmek için , sabit bir kartezyen koordinat sistemine göre tutucu veya uç elemanın pozisyonunun ölçülmesi gerekmektedir. Bu ölçümler basit bir sensör sistem kullanılarak gerçekleştirilebilir. Kuvvet kontrolünü veya bir engelin geçilmesini gerektiren görevlerde, direkt olarak kartezyen koordinat sisteminde manipülatör kontrolü, eklem koordinatlarında yapılan kontrol işleminden daha cazip olmaktadır.

Kartezyen koordinat sisteminde dizayn edilmiş değişik kontrolörler mevcuttur [117,118]. Fakat yapılan çalışmaların çoğunda kartezyen koordinatlarda verilen yörüngeler, ters kinematik çözümler ile ve bir takım kısıtlamalar altında eklem koordinatlarına çevrilerek kontrol yapılmakta, hesaplamalar büyük sayısal bilgisayarlarda bile çok zaman aldığından bu da büyük hesapsal

ve zamansal kayıplara yol açmaktadır.

Bu bölümde amaçlanan, bölüm 7.4.2' de sunulan çok giriş-çok çıkışlı öngörülü kontrolörü, direkt kartezyen koordinat sisteminde çalışabilecek şekilde dizayn etmektir. Bunun için koordinat dönüşümlerini de içeren bir ayırık zamanlı model geliştirilmiştir. Ayırık zamanlı modelin çıkışları, tutucu veya uç elemanın (x,y,z) koordinatlarının zamana göre birinci türevinden elde edilen kartezyen hız bileşenleri, girişleri ise eklem motorlarına uygulanacak torklar (veya voltaj) olarak tasarlanmıştır. Bu ayırık zaman modeline dayanan ve kartezyen koordinatlarda ayırık noktalarla belirlenmiş istenen bir yörüngenin yakından izlenmesini sağlayan bir MIMO öngörülü kontrolör dizaynı gerçekleştirilmiştir.

7.5.1 Kartezyen Uzayda Kullanılacak Ayırık Zaman Modelinin Teşkili

Kartezyen koordinat sisteminde bir manipülatör modeli elde edebilmek için eklem ve kartezyen koordinat sistemleri arasındaki ilişkiler, koordinat dönüşüm matrisleri cinsinden belirlenmelidir (Bakınız bölüm 6.1). Bu matrislerin kullanılması, hem eklem açılarının sinüs ve cosinüslerini hem de eklem koordinat eksenleri arasındaki mesafeleri gerektirmektedir. Çok eklemli bir manipülatör için kartezyen koordinat sistemleri ile eklem değişkenleri arasındaki ilişki, aşağıdaki genel formda gösterilmektedir.

$$\begin{bmatrix} P_x & P_y & P_z ; \phi_x & \phi_y & \phi_z \end{bmatrix}^T = W = f(\theta) \quad \dots (7.25)$$

Burada $f(\theta)$; eklem değişkenleri θ ' ya bağlı bir fonksiyonu, sol taraftaki ilk üç bileşen kartezyen pozisyonları ve son üç terim ise oryantasyon açılarını (Euler açıları) göstermektedir. (7.25) eşitliği

genellikle direkt kinematik denklem olarak ifade edilmektedir ve eklem açıları $\langle \theta_i \rangle$ biliniyorsa tutucunun pozisyon ve oryantasyonunu hesaplamaya yaramaktadır. Hesaplamalar, $f(\theta)$ fonksiyonundaki trigonometrik ifadelerin tanımlanmasından ibarettir.

Diğer taraftan, kartezyen koordinat sisteminde verilen tutucunun uç noktasının, pozisyon ve oryantasyonuna uygun eklem açılarını hesaplamak gerekirse ters kinematik problemini çözmek gerekir. Yani, (7.25) ifadesinin sol tarafı biliniyor iken sağ taraftaki $\langle \theta_i \rangle$ 'lerin hesaplanması istenmektedir. Bu da (7.25) ifadesinin geriye doğru çözümü ile sağlanmaktadır.

$$\theta = g(w)$$

... (7.26)

Burada g , f fonksiyonunun tersidir. (7.26) nolu ters kinematik denklemin çözümü, açılar için trigonometrik ifadeler içerdiğinden genellikle tek değildir. Başta da söylendiği gibi ilave bir takım kısıtlamalar ile çözüm sayısı minimize edilebilmektedir.

(7.26) nolu ters kinematik denklem $\langle \theta_i \rangle$ için çözümlere elde edilen $\langle \theta_i \rangle$ 'ler (7.8) ifadesinde yerine konursa, elde edilen diferansiyel denklem takımı kartezyen koordinat sisteminde manipülatör hareketini modelleyecektir. Fakat böyle bir modelin elde edilişi zordur ve kontrolör dizaynı için kullanışlı değildir. Bu çalışmada, trigonometrik fonksiyonlar içeren kompleks bir modelin kullanımından kaçınmak için uç noktanın hareketini karakterize eden çok değişkenli lineer bir ayrık zaman serisi modeli geliştirilmiştir. Bu modelde, koordinat sisteminde ifade edilen giriş torkları (veya voltaj) ve eklem açıları arasındaki ilişki, bilinmeyen parametrelili bir vektörel fark denklemi ile verilmektedir. Bu modeldeki parametreler, her bir örnekleme anında SQR parametre kestirme algoritması kullanılarak ardışık bir şekilde çevrim içi kestirilerek yenilenirler.

7.5.2. Kartezyen Uzayda Geliştirilen Ayırık Zaman

Modelinin Robotik Bir Manipülatöre Uygulanması

Bu tez çalışmasında, çok değişkenli lineer bir ayırık zaman modeli koordinat dönüşümlerini de içerecek şekilde geliştirilmiş ve yapılan simülasyon çalışmalarında gayet başarılı sonuçlar alınmıştır.

Kontrol algoritması olarak ise bu tezin esas konusu olan MIMO öngörülü kontrol algoritması kullanılmıştır. Geliştirilen modelde, modelin çıkış vektörü kartezyen koordinat sisteminin bağımsız değişkenlerinden, giriş vektörü ise eklem aktüatörlerine uygulanacak tork veya voltajlardan meydana gelmektedir. Oluşturulan ayırık zamanlı ve çok giriş-çok çıkışlı model aşağıdaki şekilde gösterilebilir.

$$y(t) = A(q^{-1})y(t) + B(q^{-1})u(t-d) + e(t) \quad \dots (7.27)$$

Burada $u(t)$, t anında eklemlere uygulanan tork (voltaj) girişi, $y(t)$ kartezyen koordinat sisteminde uç elemanın (V_x, V_y, V_z) hızları, $e(t)$ ise modelleme hataları olarak ifade edilmektedir.

$$A(q^{-1}) = A_1(q^{-1}) + \dots + A_{n_a}(q^{-n_a})$$

$$B(q^{-1}) = B_0 + B_1(q^{-1}) + \dots + B_{n_b}(q^{-n_b})$$

şeklinde tanımlanmıştır.

Uç eklemleri robot modeli için örnek olarak $n_a=1$ ve $n_b=1$ için (7.27) ifadesi daha açık bir şekilde aşağıdaki gibi yazılabilir.

$$\begin{aligned}
\begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} &= \begin{bmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 \\ a_{21}^1 & a_{22}^1 & a_{23}^1 \\ a_{31}^1 & a_{32}^1 & a_{33}^1 \end{bmatrix} \cdot \begin{bmatrix} v_x(t-1) \\ v_y(t-1) \\ v_z(t-1) \end{bmatrix} + \dots \\
&+ \begin{bmatrix} b_{11}^0 & b_{12}^0 & b_{13}^0 \\ b_{21}^0 & b_{22}^0 & b_{23}^0 \\ b_{31}^0 & b_{32}^0 & b_{33}^0 \end{bmatrix} \cdot \begin{bmatrix} u_1(t-d) \\ u_2(t-d) \\ u_3(t-d) \end{bmatrix} + \\
&+ \begin{bmatrix} b_{11}^1 & b_{12}^1 & b_{13}^1 \\ b_{21}^1 & b_{22}^1 & b_{23}^1 \\ b_{31}^1 & b_{32}^1 & b_{33}^1 \end{bmatrix} \cdot \begin{bmatrix} u_1(t-d-1) \\ u_2(t-d-1) \\ u_3(t-d-1) \end{bmatrix} + \begin{bmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \end{bmatrix} \\
&\dots \quad (7.28)
\end{aligned}$$

Parametre matrisi Θ ve veri vektörü ϕ aşağıdaki şekilde ifade edilir.

$$y(t) = \Theta^T \phi(t-1) + e(t) \quad \dots (7.29)$$

$$\Theta = \begin{bmatrix} b_{11}^0 & b_{12}^0 & b_{13}^0 & b_{11}^1 & b_{12}^1 & b_{13}^1 & a_{11}^1 & a_{12}^1 & a_{13}^1 \\ b_{21}^0 & b_{22}^0 & b_{23}^0 & b_{21}^1 & b_{22}^1 & b_{23}^1 & a_{21}^1 & a_{22}^1 & a_{23}^1 \\ b_{31}^0 & b_{32}^0 & b_{33}^0 & b_{31}^1 & b_{32}^1 & b_{33}^1 & a_{31}^1 & a_{32}^1 & a_{33}^1 \end{bmatrix}^T$$

$$\dots (7.30.a)$$

$$\phi(t-1) = [y(t-1); u(t-1); e(t)]^T \quad \dots (7.30.b)$$

Burada,

$$y(t) = [v_x(t), v_y(t), v_z(t)]^T$$

$$u(t) = [u_1(t), u_2(t), u_3(t)]^T$$

$$e(t) = [e_1(t), e_2(t), e_3(t)]^T$$

olarak kullanılmıştır.

Kontrolör dizaynında kullanılan minimize edilecek öngörülü kontrol performans indeksi, aşağıdaki şekilde seçilmiştir.

$$J(\Delta u) = \sum_{j=1}^{N_2} [y(t+j) - y_r(t+j)]^2 + \Lambda \sum_{j=1}^{N_U} [\Delta u(t+j-1)]^2 \quad (7.31)$$

Burada,

$$y_r(t+j) = [v_{xr}(t+j), v_{yr}(t+j), v_{zr}(t+j)]^T$$

şeklinde uç elemanın, referans kartezyen hız bileşenlerini göstermektedir.

(7.31) ifadesi $\Delta u'$ ya göre minimize edilerek, eklemlere uygulanacak torklar, (7.24) ifadesi de göz önüne alınarak

$$\Delta u(t) = [I \ 0 \ 0] (G^T G + \Lambda)^{-1} G^T (y_r - f) \quad \dots (7.32)$$

buradan da,

$$u(t) = \Delta u(t) + u(t-1)$$

şeklinde hesaplanır. G matrisi ve f vektörü Bölüm (4.2)

ve Ek A' da izah edildiği şekilde hesaplanmaktadır.

Not: Şayet altı eklemlili bir robot modeli ile çalışılacak ise, altı adet giriş ve altı adet çıkış gerekeceğinden (giriş ve çıkış sayıları birbirine eşit olmalıdır), bu durumda çıkış büyüklükleri, kartezyen koordinat sisteminde tanımlanan kayma $\langle V_x, V_y, V_z \rangle$ ve dönme $\langle \xi_x, \xi_y, \xi_z \rangle$ hızlarından meydana gelecektir. Girişler yine altı ekleme uygulanacak tork vektörleridir.

Genelleştirilmiş öngörülü kontrol algoritması için manipülatör uç elemanının, kartezyen uzaydaki yörünge kontrolü ile ilgili blok diyagramı şekil 7.17' de gösterilmiştir.

7.5.3 Sayısal Örnekler

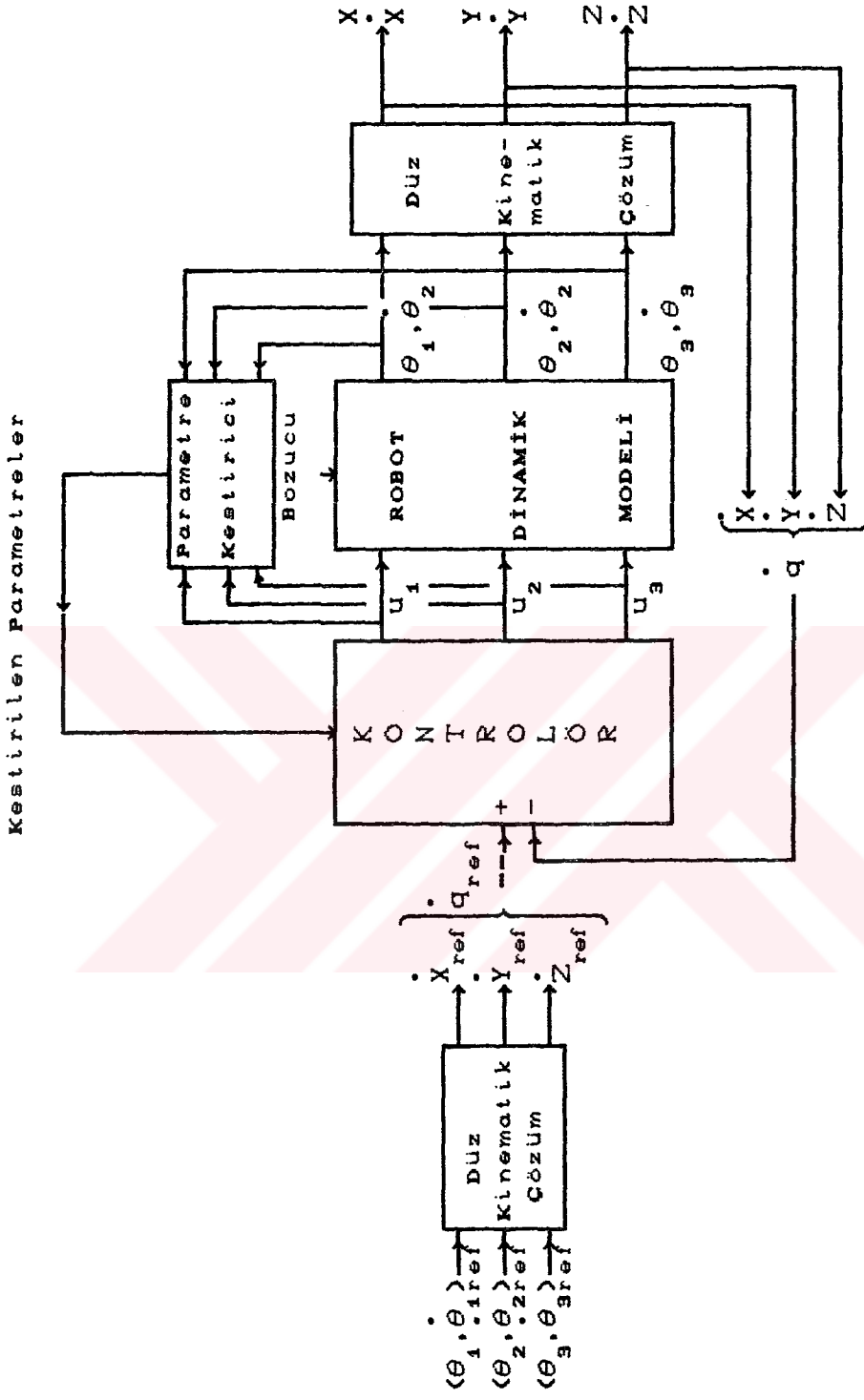
Bölüm 7.4.1 ve 7.4.2' deki aynı robot modeli kullanıldığından ilgili parametreler aynen geçerlidir. Çok giriş-çok çıkışlı genelleştirilmiş öngörülü kontrolör kullanılacağından kontrolör dizaynı ve parametre kestirme algoritması ile ilgili parametreler, MIMO bir sistem olan bölüm 7.4.2' deki gibi alınmıştır. Bölüm 7.4.2' deki eklem koordinatlarında yapılan kontrol çalışmaları ile bir karşılaştırma yapabilmek için manipülatörün uç elemanı için ayrı bir kartezyen x,y,z yörüngeleri tespit edilmemiş, bölüm 7.2.1' de verilen

$$x = (l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \cos \theta_1$$

$$y = (l_1 + l_2 \cos \theta_2 + l_3 \cos (\theta_2 + \theta_3)) \sin \theta_1$$

$$z = l_2 \sin \theta_2 + l_3 (\sin (\theta_2 + \theta_3))$$

ifadeleri kullanılarak bölüm 7.2.2' de eklem değişkenleri cinsinden verilen yörüngelerden hareketle uç elemanın



Şekil 7.17 MIMO öngörülü kontrol algoritması kullanan bir manipülör için kartezyen uzay kontrolü blok diyagramı

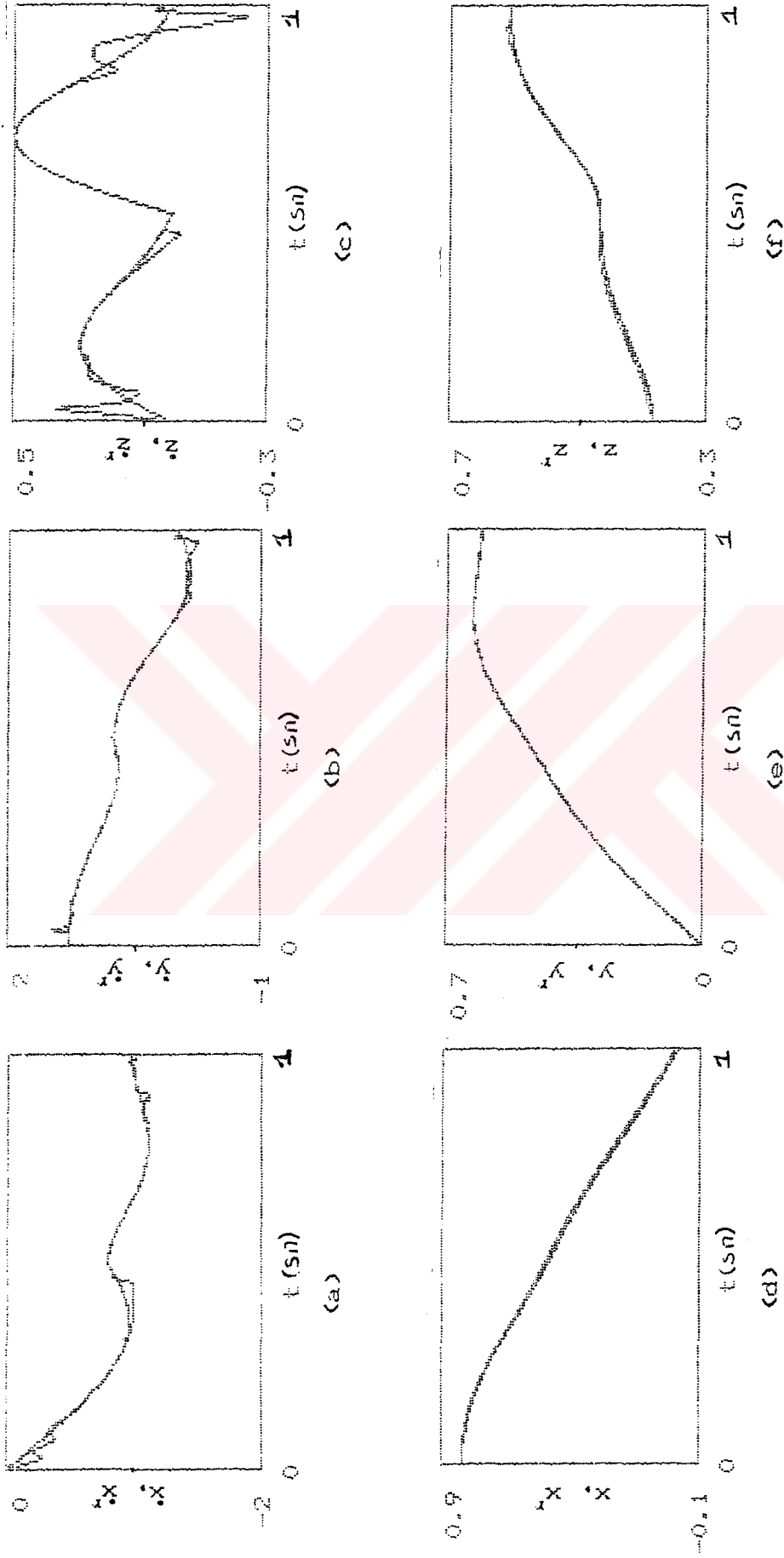
x,y,z kartezyen koordinatları ve bunun zamana göre birinci türevi olan V_x, V_y, V_z hızları hesaplanmıştır. Hesaplanan bu V_x, V_y, V_z değerleri referans büyüklükleri olarak kullanılmaktadır.

Böylece simülasyon esnasında hem kartezyen yörünge referans büyüklüğü ile karşılaştırma yapılmakta ve hem de eklem açılarının yörüngeleri de elde mevcut olduğundan, Runge-Kutta çıkışındaki θ değerleri ile θ_{ref} değerleri karşılaştırılarak eklemlerin hareketi de incelenebilmektedir.

Kartezyen uzayda çalışma durumunda, program akış şeması ayrıca bölüm Ek B' de detaylı olarak verilmiştir.

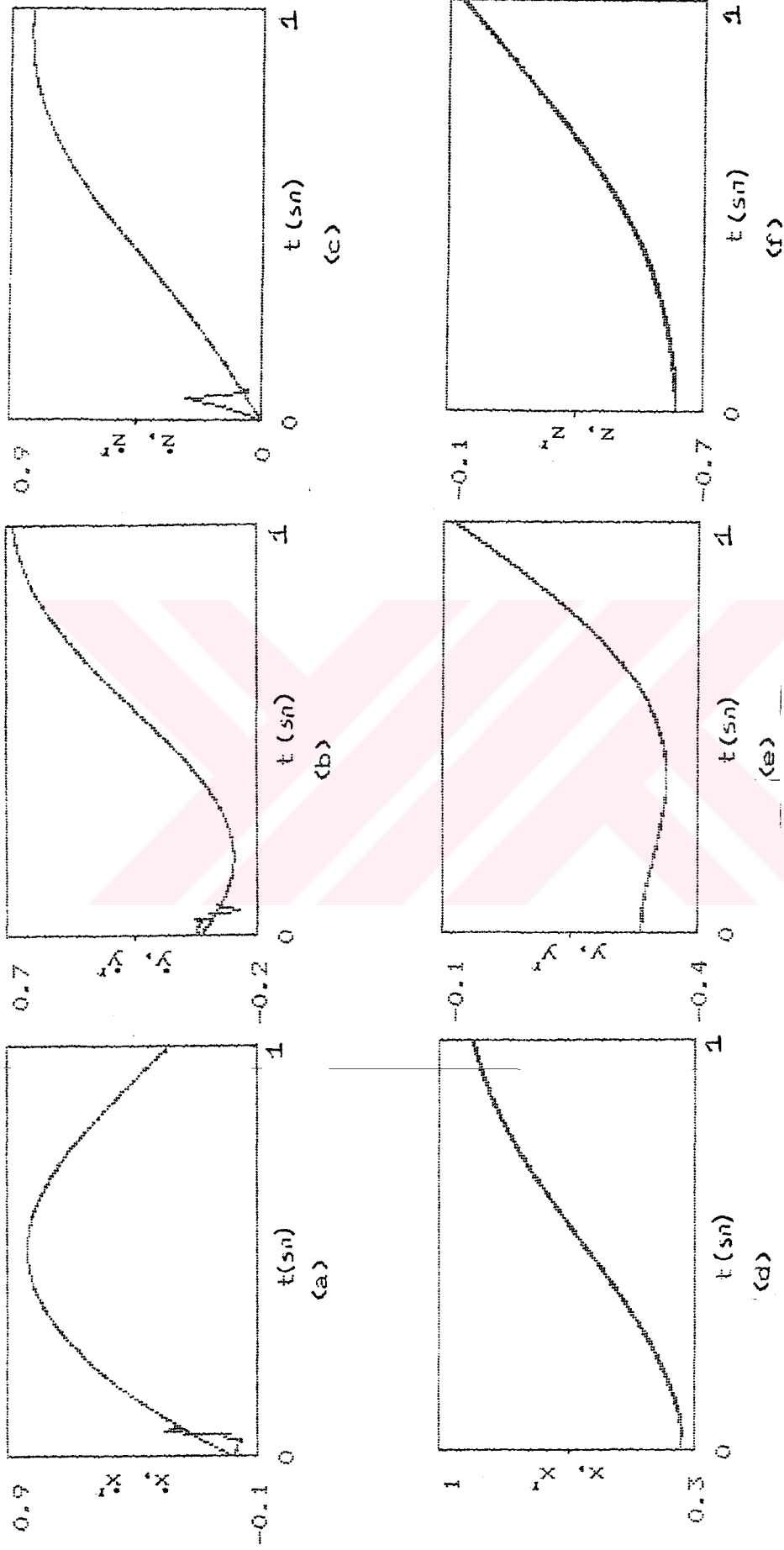
Simülasyon çalışmalarında, bölüm 7.3.1' deki senaryolar aynen alınarak performans araştırması gerçekleştirilmiştir. Sonuçlar bozucusuz durum için şekil 7.18' de, stokastik bozucu ve taşınan yük durumu için şekil 7.19' da ve sürtünmenin de ilave edildiği en genel durum için şekil 7.20' de gösterilmiştir. Uç elemanın kartezyen hızlarının hatalarının kareleri toplamı tablo 7.4' de, uç elemanın final konum hataları ise tablo 7.5' de verilmiştir. Ayrıca sürtünmenin etkilerinin de incelendiği son durum için u kontrol değerlerinin zamanla değişimi şekil 7.20.c' de verilmiştir.

GPC algoritmasının robotik bir manipülatöre uygulanması ile ilgili olarak yapılan diğer bazı çalışmalar [119,120] nolu kaynaklardan, Adaptif kontrol ve robotik ile ilgili genel bilgiler de [121,122,123,124,125] nolu kaynaklardan detaylı olarak edinilebilir.



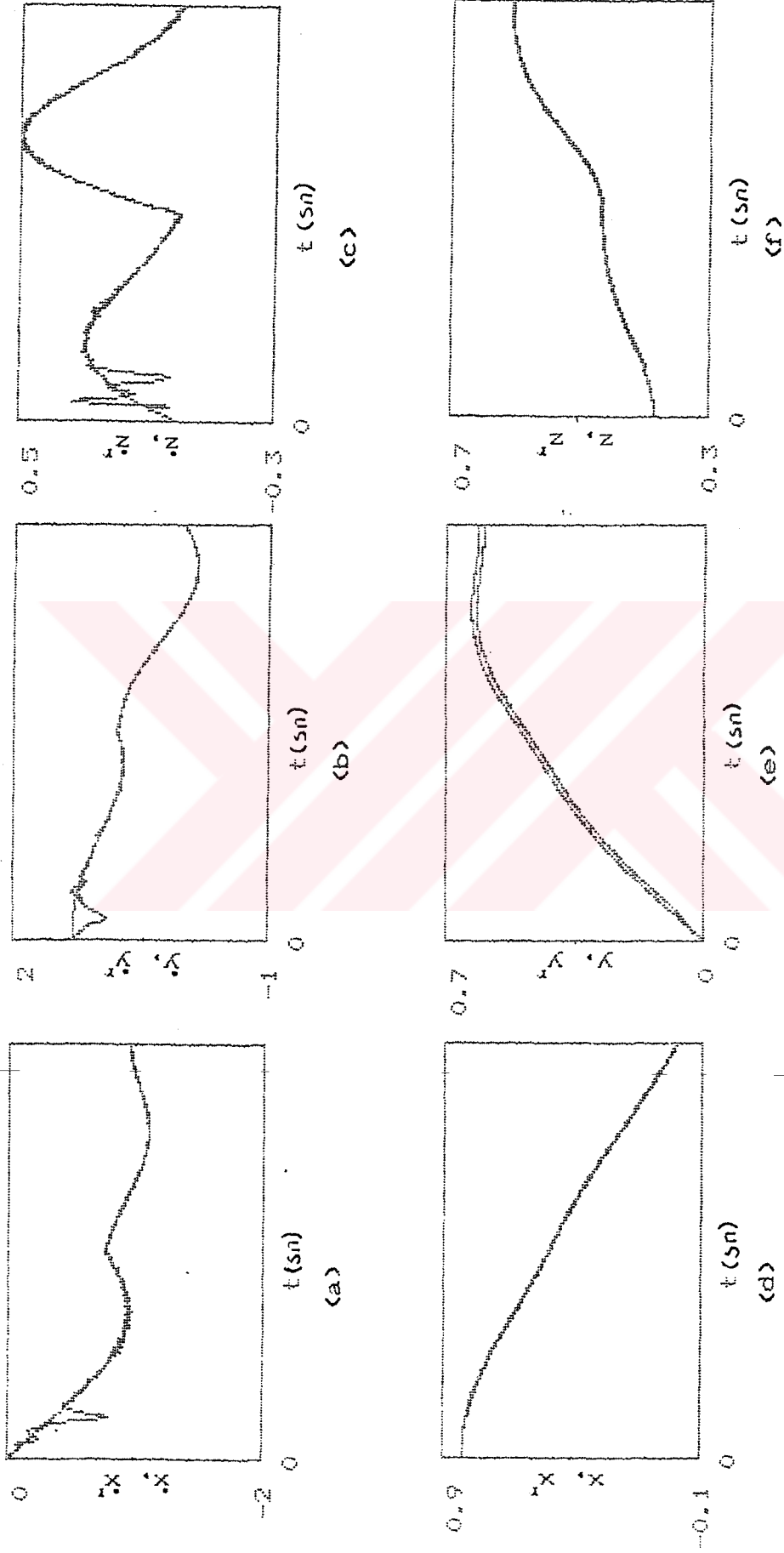
Şekil 7.18.a. Uç elemanın kartezyen uzayda kontrolü, bozucusuz durum, kübik yörünge

(a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn]
 (d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]



Sekil 7.18.b. Uç elemanın kartezyen uzayda kontrolü, bozucusuz durum, eksponansiyel yörünge

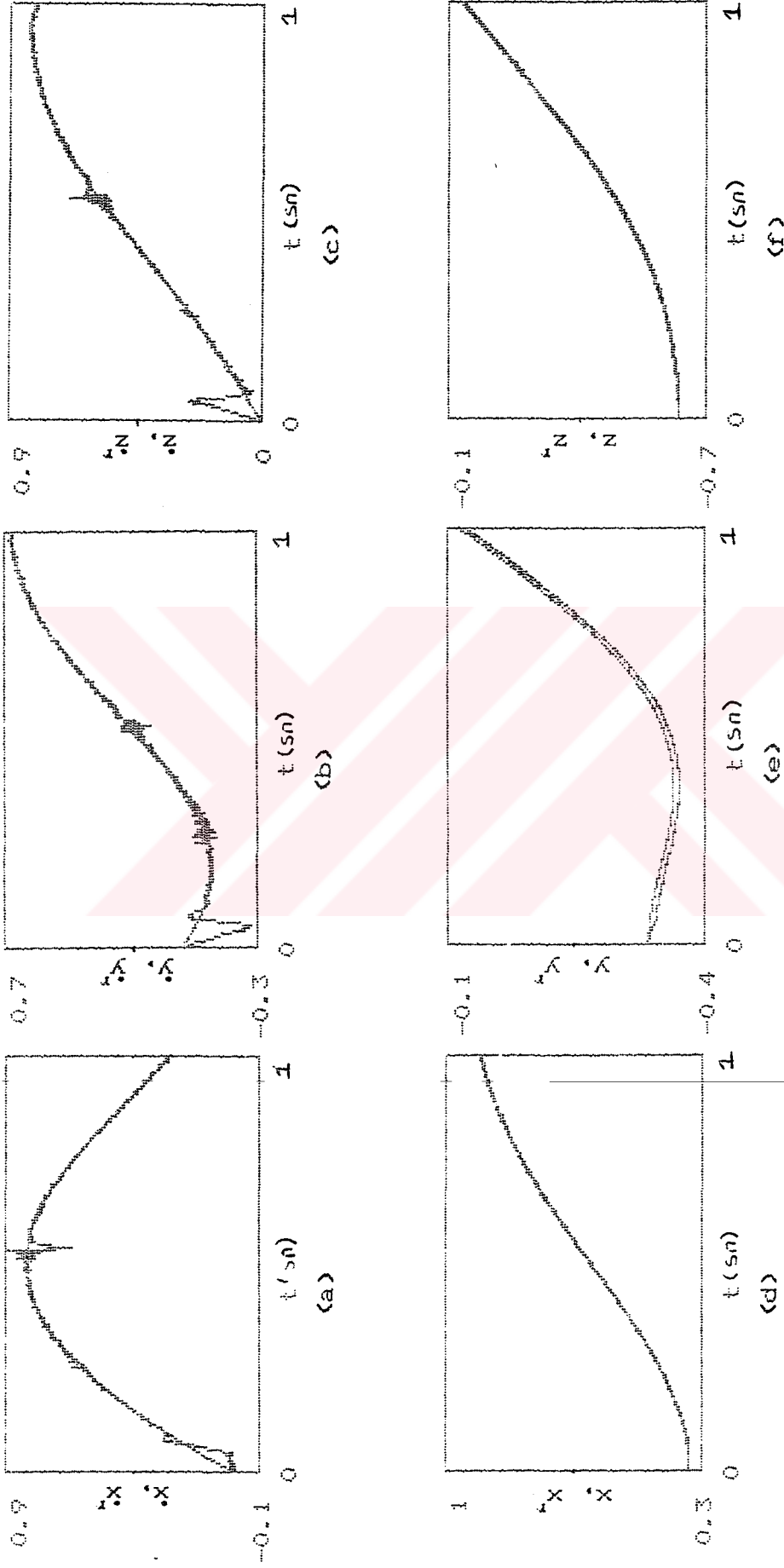
(a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn]
 (d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]



Şekil 7.19.a. Uç elemanın kartezyen uzayda kontrolü, stokastik bozucu, taşıyan yük, kübik yörünge

(a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn]

(d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]

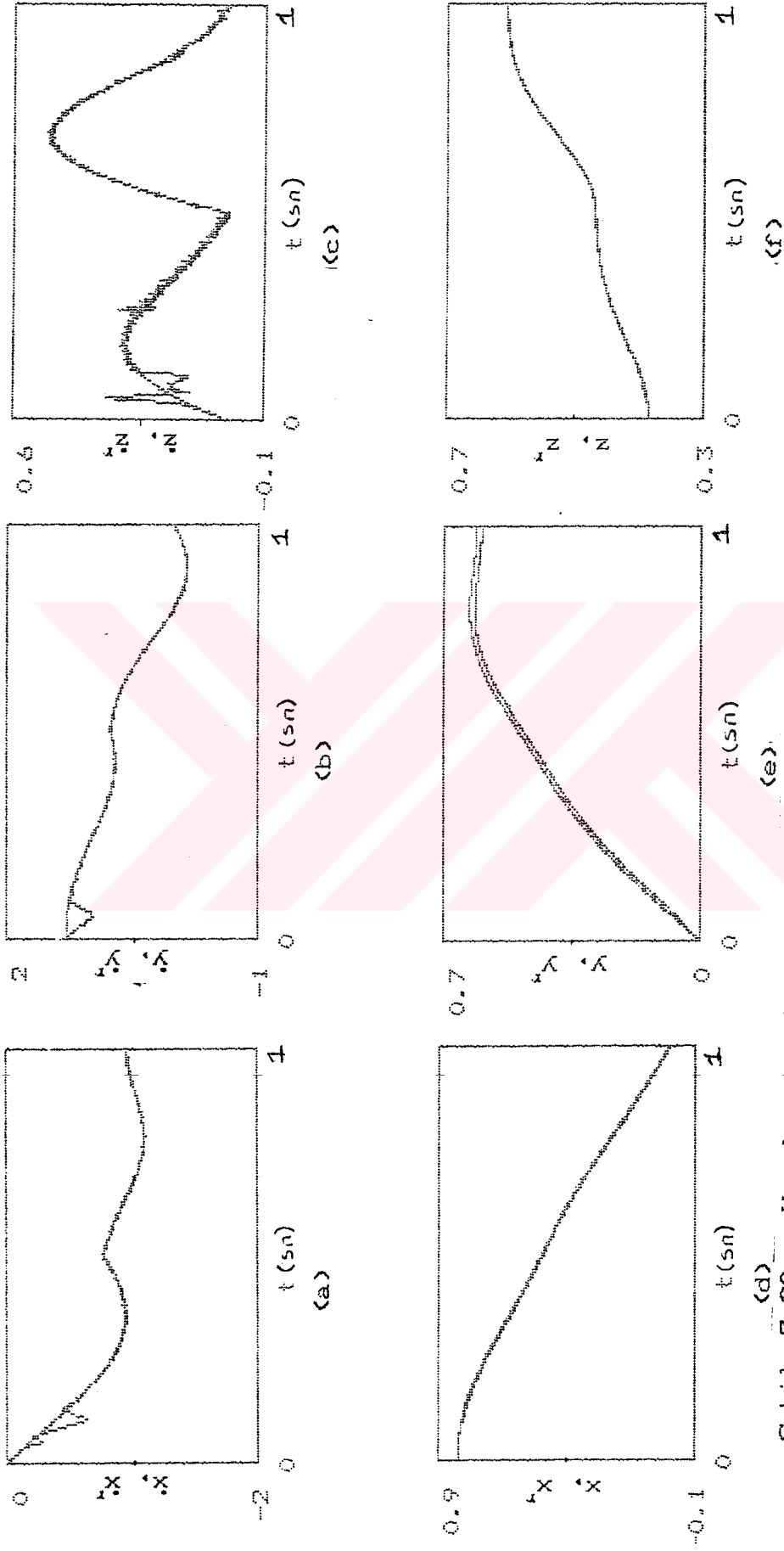


Şekil 7.19.b. Uç elemanın kartezyen uzayda kontrolü, stokastik bozucu,

taşınan yük, eksponansiyel yörünge

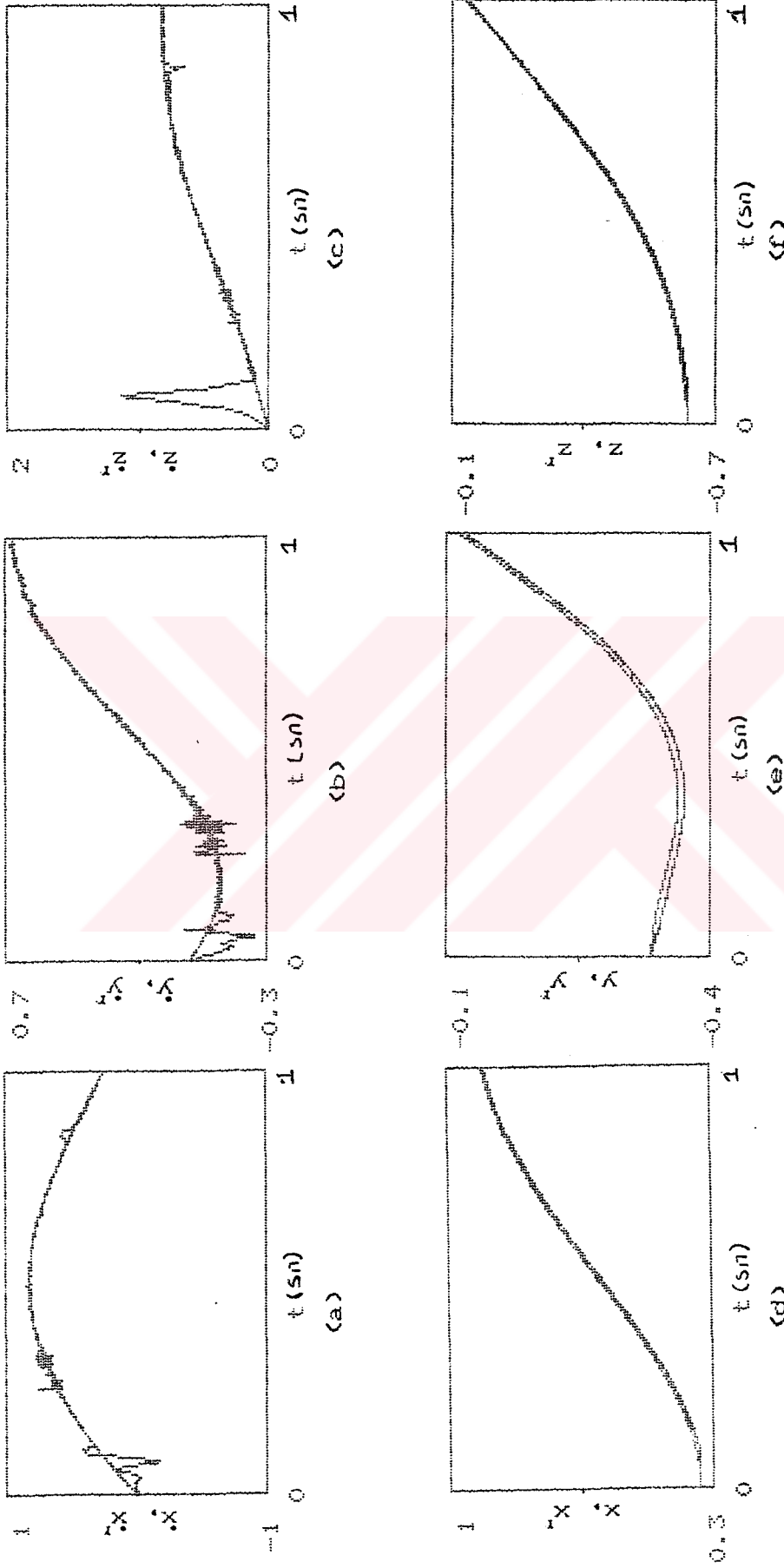
(a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn]

(d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]

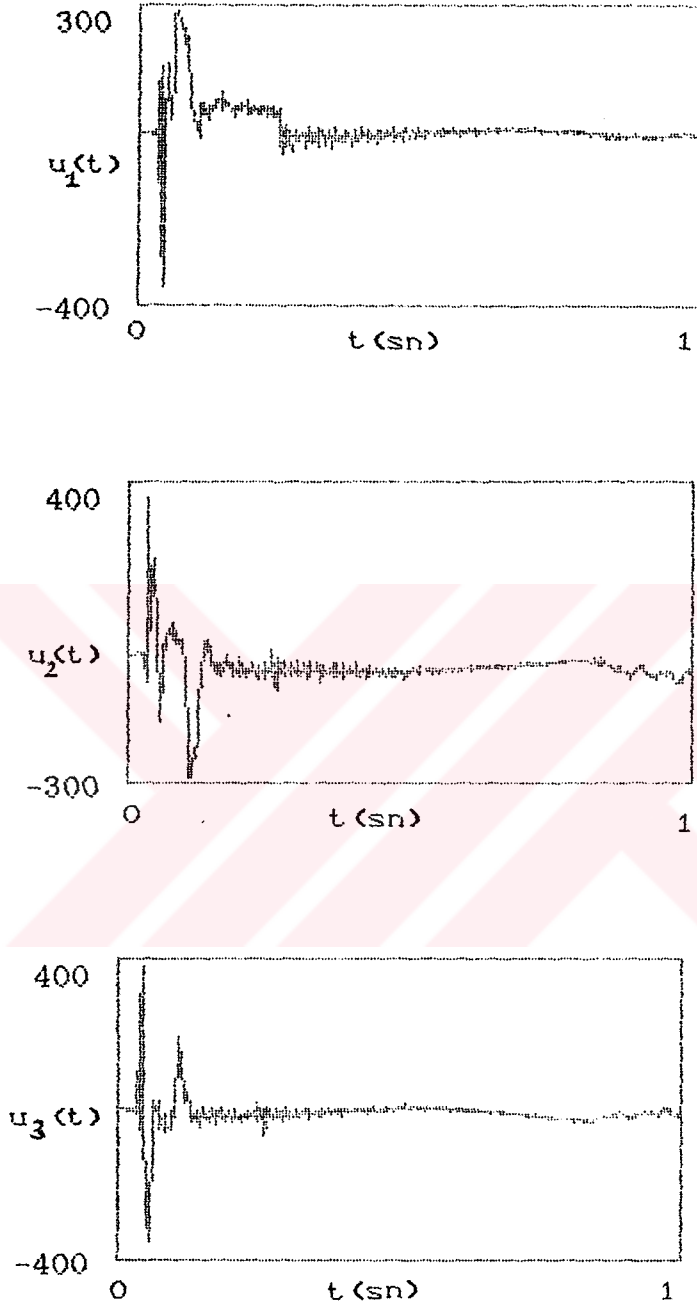


Sekil 7.20.a. Uç elemanın kartezyen uzayda kontrolü, stokastik bozucu, taşıyan yük, sürtünme, kübik yörünge

(a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn]
 (d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]



Sekil 7.20.b. Uç elemanın kartezyen uzayda kontrolü, stokastik bozucu, taşınan yük, sürtünme, ekspanansiyel yörünge
 (a), (b), (c): Uç elemanın $\dot{x}$, $\dot{y}$, $\dot{z}$ hızları ve referansları [m/sn].
 (d), (e), (f): Uç elemanın x , y , z konum ve referansları [m]



Şekil 7.20.c. Öngörülü Kartezyen Esaslı Kontrol durumu için u kontrolünün zamanla değişimi (Küçük yörünge, stokastik bozucu, sürtünme, yük değişimi).

Tablo 7.2. Eklem Hızı Hatalarının Karelerinin Toplamı

Kontrol Tipi	Senaryo	Yörünge	E-k l e m N o		
			1	2	3
Kendi Kendini Ayarlayan (SISO-STR)	A	Küçük	0.0064	1.6623	2.4990
		Ekspo.	0.0153	0.6478	1.4838
	B	Küçük	3.4222	6.6193	4.9221
		Ekspo.	0.8199	8.1085	2.5638
	C	Küçük	3.6338	5.6039	6.2209
		Ekspo.	1.4473	7.5402	2.5482
Genelleştirilmiş Öngörülü Kontrol (SISO)	A	Küçük	0.0032	0.5625	0.7291
		Ekspo.	0.0065	0.1883	6.6804
	B	Küçük	0.4290	0.7120	3.0596
		Ekspo.	0.5142	0.5090	1.6949
	C	Küçük	0.2494	0.2895	1.5019
		Ekspo.	0.4704	6.0589	2.3658
G.Ö.K (SISO) (Düşük Model Mertebesi)	A	Küçük	0.0009	0.5427	0.3629
	B	Küçük	0.1166	0.2078	0.8630
	C	Küçük	0.1091	0.4086	3.7043
Genelleştirilmiş Öngörülü Kontrol (MIMO)	A	Küçük	0.0029	0.2718	0.7521
		Ekspo.	1.2296	2.3378	22.144
	B	Küçük	1.3541	1.1534	5.6652
		Ekspo.	7.5688	2.9264	7.1049
	C	Küçük	0.5499	2.2417	13.784
		Ekspo.	3.4351	1.6471	4.2000

Tabloda Kullanılan Senaryolar:

A : Bozucu yok

B : Stokastik Bozucu + Taşınan Yük Değişimi

C : Stokastik Bozucu + Taşınan Yük Değişimi + Sürtünme

Tablo 7.3. Eklem Mutlak Final Konum Hataları [rad.]

Kontrol Tipi	Senaryo	Yörünge	E k l e m N o		
			1	2	3
Kendi Kendini Ayarlayan (SISO-STR)	A	Küçük	0.0002	0.0430	0.0452
		Ekspo.	0.0047	0.0333	0.0293
	B	Küçük	0.0406	0.1020	0.0397
		Ekspo.	0.0257	0.0949	0.0247
	C	Küçük	0.0534	0.1005	0.0533
		Ekspo.	0.0348	0.1107	0.0310
Genelleştirilmiş Öngörülme Kontrol (SISO)	A	Küçük	0.0003	0.0104	0.0029
		Ekspo.	0.0002	0.0005	0.0170
	B	Küçük	0.0054	0.0003	0.0049
		Ekspo.	0.0091	0.0711	0.0013
	C	Küçük	0.0047	0.0089	0.0100
		Ekspo.	0.0087	0.0059	0.0104
G.Ö.K (SISO) (Düşük Model Mertebesi)	A	Küçük	0.0001	0.0068	0.0033
	B	Küçük	0.0035	0.0123	0.0038
	C	Küçük	0.0033	0.0029	0.0072
Genelleştirilmiş Öngörülme Kontrol (MIMO)	A	Küçük	0.0004	0.0045	0.0047
		Ekspo.	0.0026	0.0128	0.0467
	B	Küçük	0.0134	0.0066	0.0114
		Ekspo.	0.0292	0.0148	0.0157
	C	Küçük	0.0087	0.0091	0.0233
		Ekspo.	0.0208	0.0097	0.0132

Tablo 7.4 Uç Elemanın Kartezyen Hız Hatalarının Kareleri Toplamı

Kontrol Tipi	Senaryo	Yörünge	Uç Elemanın Hızı		
			x	y	z
Genelletirilmiş Öngörülü Kontrol Kartezyen Uzayda Uç Elemanın Kontrolü	A	Küçük	0.5224	0.4478	1.4198
		Ekspo.	0.2661	0.0371	0.3345
	B	Küçük	1.0691	1.7200	0.4328
		Ekspo.	0.2550	0.5420	0.3579
	C	Küçük	0.5498	1.2614	0.2910
		Ekspo.	1.0906	0.4765	10.524

Tablo 7.5 Uç Elemanın Mutlak Final Konum Hataları [m.]

Kontrol Tipi	Senaryo	Yörünge	Uç Elemanın Konumu		
			x	y	z
Genelletirilmiş Öngörülü Kontrol Kartezyen Uzayda Uç Elemanın Kontrolü	A	Küçük	0.0148	0.0010	0.00008
		Ekspo.	0.0048	0.0011	0.0057
	B	Küçük	0.0067	0.0155	0.0022
		Ekspo.	0.0041	0.0074	0.0069
	C	Küçük	0.0072	0.0143	0.0013
		Ekspo.	0.0064	0.0072	0.0380

GENEL SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, genelleştirilmiş öngörülü kontrol algoritması üç eklemli bir manipülatöre yörünge kontrolu için uygulanmıştır. Kontrol performansı, değişik stokastik ve deterministik bozucular (ölçme hataları, sürtünme etkileri ve yük değişimi) olması durumları için araştırılmıştır. Simülasyon esnasında manipülatöre hız referansı verilmekte ve konumun zamanla değişimi kontrol altına alınmış olmaktadır. Kontrol algoritmasına ait performans indeksinde sadece eklem hızları göz önüne alınmakta, konum değerleri hesaba katılmamaktadır. Konum eğrileri ise sadece yörüngenin takibini inceleyebilmek açısından ilave edilmiştir. Her örnekleme periyodunda hız referansı vermenin yörünge takibi açısından zorluklar yaratacağı düşüncesi, genelleştirilmiş öngörülü kontrol algoritmasının sabit olmayan yörünge takibindeki etkinliği ile ortadan kaldırılmış olmaktadır.

Her bir robot eklemi için ayrı bir kontrolör dizaynı (SISO durum) ve üç eklem için merkezi tek bir kontrolör dizaynı (MIMO durum) olmak üzere iki farklı çalışma yapılmıştır. Ayrıca ters kinematik çözüme gerek kalmadan robot uç elemanının kartezyen koordinatlarının direkt olarak kontroluna olanak sağlayan yeni bir ayrık zaman modeli geliştirilmiştir.

Öngörülü kontrol algoritması ile ilgili performans araştırmalarına başlamadan önce, Borison' un önerdiği bir kendi kendini ayarlayan kontrol algoritması da [40] sadece tek giriş-tek çıkışlı (SISO-STR) durum için aynı performans araştırmalarına tabi tutulmuştur.

Kendi kendini ayarlayan kontrol algoritmasının kullanıldığı simülasyon çalışmalarına bakıldığında (şekil 7.4-5-6), kübik yörüngelerin kullanıldığı senaryolarda özellikle sabit bir değere sahip ilk eklem hızının referans yörüngeyi takipte zorlandığı görülmektedir. Hiç bir bozucunun ele alınmadığı "A" senaryosu için dahi, tüm simülasyon süresince ilk eklem hız eğrisinde sapmalar görülmektedir. Bozucuların ilave edilmesi durumunda değer olarak da büyüyen sapmalar kalıcı hata durumuna dönüşmektedirler. Yine, bozucuların ilave edilmesi durumunda, özellikle ikinci eklem hız eğrilerinde de referans değerlerine göre sapmalar ortaya çıkmaktadır. Konum eğrilerine bakıldığında ise eklem arası etkileşimlerin en çok olduğu ortadaki ikinci eklemden, bozuculu durumda daha çok olmak üzere ve gittikçe artan karakteristikte kalıcı hatalar görülmektedir. Tablo 7.3' de de görüldüğü gibi 1 sn. lik simülasyon süresi sonunda bozuculu durum için 0.1 rad. mertebesinde bir konum hatası görülmektedir. Tablo 7.2' den eklem hız hatalarının kareleri toplamı incelendiğinde de, kendi kendini ayarlayan kontrol algoritmasının kullanıldığı durumda öngörülü kontrol algoritmasına göre hataların daha büyük olduğu, özellikle bozuculara karşı self-tuning kontrolün daha duyarlı olduğu gözlenmektedir. Referans yörüngeler değiştirilerek eksponansiyel yörüngelerin kabul edilmesi de yörünge takibi açısından sonucu değiştirmemiştir. Bu durum için özellikle ikinci eklem hız eğrilerinde sapmalar görülmektedir. Bozucusuz durum için ikinci ve üçüncü eklem konumlarında kalıcı hatalar oluşmaktadır. Bozucu etkilerin devreye girmesi ile özellikle etkileşimin çok olduğu ikinci eklemden büyük değerler alarak, her üç eklem için de kalıcı hatalar ortaya çıkmaktadır. Buradan da takibi kübik yörüngelere göre daha zor olan eksponansiyel yörüngeler için kendi kendini ayarlayan kontrol algoritmasının yetersiz kaldığı sonucu çıkmaktadır. Tüm bozucu büyüklüklerin

ilave edildiği ve kübik yörüngelerin kullanıldığı "C" senaryosu için çizilen u kontrol büyüklüklerinin zamanla değişim grafiğinde de görüldüğü gibi (şekil 7.6.c), her üç eklem için öngörülü kontrol algoritması durumundakinden daha büyük salınımlar ortaya çıkmaktadır.

Öngörülü kontrol algoritmasını kullanan simülasyon çalışmaları önceden de belirtildiği gibi üç ana grupta toplanmıştır. Bunlar SISO ve MIMO eklem bazlı ve MIMO kartezyen uzay bazlı kontrol çalışmalarıdır. SISO olarak dizayn edilen kontrolöre ait simülasyon çalışması konum ve hız eğrileri şekil 7.8-9-10' da gösterilmiştir. Kendi kendini ayarlayan kontrol algoritması ile bir karşılaştırma yapılacak olursa, kübik yörüngeler için ilk eklem hızındaki sapmaların ilk adımlar hariç tamamen ortadan kalktığı görülmektedir. Tüm senaryolar için maksimum hatanın çalışmanın ilk adımlarında ortaya çıkması doğal bir sonuçtur. Bu hata daha sonra minimize edilmektedir. Çünkü, başlangıçta manipülatör durmaktadır ve kendisine referans olarak bir hız ve konum verildiğinde, bir bakıma manipülatöre bu noktaya nasıl gidersen git denilmektedir. Verilen referans ne kadar büyük olursa o kadar büyük hata oluşmaktadır. Daha sonraki adımlarda hem başlangıç, hem de bitiş noktaları için referanslar bulunacağından, yörünge konum ve hız değerleri çok yakın bir şekilde takip edilecektir. İlave edilen bozucular karşısında hız eğrilerinde ortaya çıkan çok küçük genlikli salınımlar normal karşılanmalıdır. Direkt olarak bozucuların varlığından kaynaklanmaktadır. Taşınan yükün senaryo gereği bırakıldığı 100.cü adımda (0.25. sn) doğal olarak bir salınım oluşmaktadır. Bu salınım şekil 7.10.c' de verilen u kontrol büyüklüğünün değişim grafiğinde daha belirgin bir şekilde görülmektedir. Robot dinamiğinde, yük değişiminden kaynaklanacak değişmeden dolayı, bu beklenen bir durumdur. Fakat u değerlerinin kendi kendini ayarlayan kontrolöre nazaran daha kararlı bir

davranış gösterdiği de açıkça görülmektedir. Bu değişmelere rağmen eklem konumlarında referans yörüngeye göre hiç bir sapmanın ortaya çıkmaması genelleştirilmiş öngörülü kontrol algoritmasının performans üstünlüğünü göstermektedir.

Hız eğrilerindeki baştaki dalgalanmalar, hızın zamanla değişimi kontrol altına alınmadığından ortaya çıkmaktadır. Kontrolör performans indeksinde, hız referansı verilerek konumun zamanla değişimi kontrol altına alındığından konumdaki aşmalar minimize edilmektedir. Bundan sonraki çalışmalara ışık tutması için bir öneri olarak söylemek gerekirse; manipülatöre ivme referansı da verilebilirse, performans indeksi ivme referansını da içerecek şekilde değiştirilerek hızın zamanla değişimi de kontrol altına alınacağından, ilk andaki durma anından istenen referans hızına yumuşak bir geçiş sağlanabilecektir. Buna rağmen şekil 7.8-9-10'dan da görüldüğü gibi öngörülü kontrol algoritması için kendi kendini ayarlayan kontrol algoritmasına göre çok başarılı sonuçlar alınmıştır.

Takibi nispeten daha zor olan eksponansiyel yörüngeler kullanımında, bozucuların ilave edildiği senaryolar için çok küçük mertebelerde hatalar oluşmaktadır. Bunlar önemsiz hatalar olarak gözükmemektedir. Örneğin "C" senaryosu için üçüncü eklemdeki hata 0.01 rad. mertebesinde olmaktadır. Goldenberg yaptığı bir çalışmada 0.1 rad. lık konum hatasını kabul edilir olarak göstermiştir [10].

Ayrıca; öngörülü kontrol algoritmasının, model mertebesinin hatalı seçilmesinden etkilenmediğini göstermek için model mertebesinin hatalı olarak düşük seçildiği bir duruma ait simülasyon çalışması sonuçları kübik yörüngeler için şekil 7.11' de verilmiştir. Mertebe azalmasından dolayı kestirilecek parametre sayısının da azalmasının etkisiyle, eklem açıları

referans değerlerine gayet iyi yakınsamışlardır. Bu da, model mertebesini hatalı tanımlamanın genelleştirilmiş öngörülü kontrol algoritmasında çok önemli olmadığını göstermektedir. Tablo 7.2' den eklem hız hatalarının kareleri toplamı, tablo 7.3' den eklem final konum hataları incelendiğinde hataların kendi kendini ayarlayan kontrol algoritmasına göre daha küçük değerler aldığı ve kabul edilebilir mertebelerde olduğu söylenebilir.

Simülasyon çalışmalarından elde edilen sonuçlar, birbirinden bağımsız eklem kontrolü (SISO) uygulandığı zaman sistem performansının gayet iyi olduğunu göstermektedir. Birbirinden bağımsız eklem kontrolünün en büyük avantajı basit olmasıdır. Eklemler arasındaki etkileşimler göz önüne alınmadığından daha basit bir yapıya sahiptir. Ayrıca kontrolör, kestirilen parametre sayısı az olduğundan mikroprosesör kullanımına çok elverişlidir.

Robotun bir bütün olarak ve eklemler arasında var olan tüm etkileşimler göz önüne alınarak incelendiği, merkezi tek bir kontrolör dizaynı ile ilgili olarak yapılan simülasyon çalışmaları, daha zor bir görev teşkil etmesine rağmen gayet başarılı sonuçlar vermişlerdir. Şekil 7.13-14-15' den de görüldüğü gibi eklem hızları üzerine eklenen ölçme hatası olarak varsayılan stokastik bozucular, eldeki yükün değişmesi ve eklemlerdeki sürtünmelerin ele alınması gibi bozucuların etkisine rağmen istenen referans yörüngeler gayet yakından takip edilebilmektedir. u kontrol büyüklüğünün kararlı bir değişme göstermesi de yine öngörülü kontrol algoritmasının özelliği olarak gösterilebilir (Şekil 7.15.c). Elde edilen sonuçlar, genelleştirilmiş öngörülü kontrol algoritmasının robot kontrolunda MIMO bir algoritma olarak kullanımına çok elverişli olduğunu göstermektedir. MIMO durum için tablo 7.2 ve 7.3' deki hata değerlerindeki küçük

artışlar eklemeler arasındaki tüm etkileşimlerin göz önüne alınmasından kaynaklanmaktadır. Ayrıca örnekleme periyodu ve integrasyon adımı küçültülerek $\Delta t=0.001$ sn. alınarak eksponansiyel yörünge ve "C" senaryosu için yapılan simülasyon sonuçları, eklem konum yörüngelerindeki çok küçük de olsa var olan hataların daha da azaldığını göstermektedir (Şekil 7.15.d). Hesaplamalar daha sık yapılacağından hata azalmaktadır, fakat adım sayısı artacağından ve 0.001 sn.de bir örnekleme yapmak pratik açıdan uygun olmayacağından simülasyon çalışmalarında bu integrasyon adımı tercih edilmemiştir.

Yörünge hataları karşılaştırıldığı zaman (Tablo 7.2 ve 7.4), çok değişkenli durum için elde edilen simülasyon sonuçlarının tek değişkenli durum için elde edilen sonuçlara göre açık bir iyileşme sağlamadığı görülmektedir. Bunun nedeni, MIMO kontrol algoritmasında SISO algoritmadakinden daha fazla parametrenin kestirilmesi ve bunun da yakınsama problemine sebebiyet vermesi şeklinde izah edilebilir. Bu yüzden tablo 7.2' de görüldüğü gibi SISO öngörülü kontrol algoritması için düşük model mertebesi ile çalışılması durumunda ,model mertebesinin azalması ile kestirilecek parametre sayısı azalacağından eklem hız hataları da azalmaktadır.

Simülasyon çalışmalarının son bölümünde, öngörülü kontrol algoritması, ayırık zamanlı bir model kullanılarak robotik manipülatörün uç elemanının direkt olarak kartezyen uzayda kontroluna uyarlanmıştır. Kartezyen uzayda uç elemanın hareketlerini kontrol etmek için takibi gayet zor olan yörüngeler seçilmiştir. Bu da, eklem kontrolunda kullanılan yörüngelerin takibini görebilmek amacıyla, uç elemanın kartezyen yörüngelerinin daha önceki aynı eklem yörüngelerinden düz kinematik ifadeler kullanılarak hesaplanmasından kaynaklanmaktadır. Elde edilen yörüngelerin zamanla

değişimi şekil 7.18-19 ve 20' de görülmektedir. Takibi zor yörüngelerin seçilmesine rağmen (yörüngelerde ani değişimler olduğu için) uç elemanın direkt olarak ve çok küçük hatalarla referans yörüngesini gayet yakından takip etmesi, hem genelleştirilmiş öngörülü kontrol algoritmasının ve hem de bu tez çalışmasında önerilen modelin kullanımının bir başarısı olarak ortaya çıkmaktadır. Üstelik yörüngelerin takibinin zor olması yanında stokastik bozucular, sürtünme etkileri ve uç elemanın taşıdığı yükün değişmesi durumlarının bulunması bu başarıyı daha da pekiştirmektedir. Ayrıca ters kinematik çözüm kullanılmadığı için ağır hesaplamaları ve tek çözüm bulmak için gerekli olan kısıtlamaları ortadan kaldırması da hesapsal açıdan önerilen algoritmanın üstünlüğünü göstermektedir. Hız referanslarında özellikle kübik yörüngeler için görülen sapmalar, yörüngelerin birbirinden farklı uç eğrinin toplamı şeklinde alınmasından ve hızın zamanla değişiminin kontrol altına alınmamasından ortaya çıkan yapısal zorluklardan kaynaklanmaktadır. Bunlara rağmen tablo 7.4 ve 7.5' den de görüldüğü gibi hatalar yine de çok küçük mertebelerde kalmaktadır.

Gerçekleştirilen simülasyon çalışmaları, genelleştirilmiş öngörülü kontrol algoritmasındaki öngörü yönteminin, kontrolöre bir zeka kazandırdığı ve modelleme hataları ile sistemdeki doğrusalsızlıklar ve etkileşimlerin etkilerini ortadan kaldırdığını göstermektedir. Ayrıca kartezyen uzayda uç elemanın kontrolü için gerçekleştirilen ayırık zaman modeli sayesinde ters kinematik çözüme gerek kalmamakta, hesaplamalar basitleşmekte ve gerçek zamanda kontrol'e olanak sağlanmaktadır.

Bu tez çalışmasında gerçekleştirilen genelleştirilmiş öngörülü kontrol algoritması altı serbestlik dereceli bir manipülatör için de denenmiştir. Fakat artan serbestlik derecesi ile manipülatör dinamiğinin

karmaşık bir yapı alması ve kullanılan hızı düşük bilgisayarlar ile simülasyonun uzun süreler gerektirmesi nedeni ile sonuçlar tez içeriğine dahil edilmemiştir. Fakat yapılan kısa süreli simülasyon çalışmalarından gayet olumlu sonuçlar alındığını belirtmek faydalı olacaktır. Hızlı bilgisayarlar kullanılır ve manipölatör dinamiğinin hesaplandığı programlar (SYS), programlama tekniği açısından hız kazandırılacak şekilde tekrar gözden geçirilirse, altı serbestlik dereceli manipölatörler için de gayet olumlu sonuçlar elde edilecektir. Uç elemanın kartezyen uzayda direkt olarak kontrolü için de söylenenler geçerlidir.



KAYNAKLAR

- [1] PAUL, R.P.C., Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Stanford Artificial Intelligence Laboratory Memo, No. 177, 1972.
- [2] WHITNEY, D.E., Resolved Motion Rate Control of Manipulators and Human Prostheses, IEEE Trans. Man-Machine Syst., Vol.MMS-10, pp. 47-53, 1969.
- [3] KAHN, M.E., and ROTH, B., The Near-Minimum Time Control of Open-loop Articulated Kinematic Chains, J. Dynamic Syst., Measurement, Contr., Vol.93, pp. 164-172, 1971.
- [4] SARIDIS, G.S., and LEE, C.S.G., An Approximation Theory of Optimal Control for Trainable Manipulators, IEEE Trans. Syst. Man. Cybern., Vol.SMC-9, pp. 152-159, 1979.
- [5] TOURASSIS, V.D., and NEUMAN, C.P., Robust Nonlinear Feedback Control for Robotic Manipulators, IEE Proceedings, Vol.132, No.4, pp. 134-143, July 1985.
- [6] ASADA, H., KANADE, T., and TAKEYAMA, I., Control of a Direct-Drive Arm, ASME Journal of Dynamic Systems, Measurement and Control, Vol.105, No. 3., pp. 136-142, 1983.
- [7] PAUL, R., The Mathematics of a Computer Controlled Manipulator, Proceedings of the joint Automatic Control Conference, Vol.1, pp. 124-131, 1977.
- [8] RAIBERT, M.H., and HORN, B.K.P., Manipulator Control using the Configuration Space Method, The Industrial Robot, Vol.2, pp. 69-73, June 1978.
- [9] SERAJI, H., An Approach to Multivariable Control of Manipulators, Journal of Dynamic Systems, Measurement, and Control, Vol.109, pp. 146-154, June 1987.
- [10] GOLDENBERG, A.A, APKARIAN, J.A and SMITH, H.W, An Approach to Adaptive Control of Robot Manipulators Using the Computed Torque Technique, Journal of Dynamic Systems,

Measurement, and Control, Vol.111, pp. 1-8, March 1989.

- [11] GAWRONSKI, R.R., Two Stage Learning Algorithm for Optimal Control of a Manipulator, Robotic Research: The Next Five Years and Beyond, M., 84-888, Society of Manufacturing Engineers, Dearborn, MI, August 1984.
- [12] DUBOWSKY, S., and DES FORGES, D.T., The Application of Model Referenced Adaptive Control to Robotic Manipulators, ASME Journal of Dynamic Systems, Measurement, and Control, Vol.101, pp. 193-200, Sept. 1979.
- [13] DUBOWSKY, S., and DES FORGES, D.T., Robotic Manipulator Control Systems with Invariant Dynamic Characteristics, Proc. 5th World Congress on Machines and Mechanisms, Montreal 1979.
- [14] LEININGER, G.G., and WANG, S.P., Pole-placement Self-tuning Control of Manipulators, Proc. IFAC Symp. on CAD of Multivariable Systems, West Lafayette, Ind., pp. 27-30, 1982.
- [15] KOIVO, A.J., GUO, T.H., Adaptive Linear Controller for Robotic Manipulators, IEEE Transactions on Automatic Control, Vol.28, No.2, pp. 162-171, Feb. 1983.
- [16] TOMIZUKA, M., and HOROWITZ, R., Model reference adaptive control of mechanical manipulators, Proc. IFAC Workshop on Adaptive Systems in Control and Signal Processing, San Francisco, pp. 27-32. 1983
- [17] LEE, C.S.G., and CHUNG, M.J., An Adaptive Control Strategy for Computer-based Manipulators, Proc. IEEE Conf. Decision and Control, Orlando, Fla., pp. 95-100, 1982.
- [18] LEE, C.S.G., and CHUNG, M.J., An Adaptive Control Strategy for Mechanical Manipulators, IEEE Trans. Auto. Contr., AC-29, 837-840, 1984.
- [19] LEE, C.S.G., and CHUNG, M.J., Adaptive Perturbation Control with Feedforward Compensation for Robot Manipulators, Simulation, 44, pp. 127-136, 1985.
- [20] KOIVO, A.J., Self-Tuning Manipulator Control in Cartesian Base Coordinate System, Journal of Dynamic Systems, Measurement, and Control, Vol.107, pp. 316-323, December 1985.
- [21] BACKES, P.G., LEININGER, G.G., and CHUNG, C.H.,

Joint Self-Tuning with Cartesian Set-points, Journal of Dynamic Systems, Measurement, and Control, Vol.108, pp. 146-150, June 1986.

- [22] HOROWITZ, R., and TOMIZUKA, M., An Adaptive Control Scheme for Mechanical Manipulators - Compensation of Nonlinearity and Decoupling Control, Journal of Dynamic Systems, Measurement, and Control, Vol.108, pp. 127-135, June 1986.
- [23] CRAIG, J.J., HSU, P., and SASTRY, S.S., Adaptive Control of Mechanical Manipulators, The International Journal of Robotic Research, Vol.6, No.2, pp. 16-28, Summer 1987.
- [24] DE SILVA, C.W., and VAN WINSSEN, J., Least Squares Adaptive Control for Trajectory Following Robots, Journal of Dynamic Systems, Measurement, and Control, Vol.109, pp. 104-110, June 1987.
- [25] SERAJI, H., A New Approach to Adaptive Control of Manipulators, Journal of Dynamic Systems, Measurement, and Control, Vol.109, pp. 193-202 September 1987.
- [26] LIU, M.H., and LIN, W., Multivariable Self-Tuning Control with Decoupling for Robotic Manipulators, IEE Proc. Vol.135, Pt.D, No.1, January 1988.
- [27] SERAJI, H., Adaptive Force and Position Control of Manipulators, Journal of Robotic Systems, 4(4), pp. 551-578, 1987.
- [28] LELIC, M.A., and WELLSTEAD, P.E., A Generalized Pole-placement Self-tuning Controller, Part II- an Application to Robot Manipulator Control, Int. Journal of Control, 46, No.2, pp. 569-602, 1987.
- [29] CLARKE, D.W., Generalized Predictive Control : A Robust Self-Tuning Algorithm. ACC 1987.
- [30] LAMBERT, M.R., Adaptive Control of Flexible Systems, D.Phil. Thesis, Oxford, 1987.
- [31] TERENCE, T.C.T, The Application of Predictive Control to Flexible Robot Arms, D.Phil. Thesis, Oxford 1990.
- [32] KAYNAK, O., Robot Kollarında Öngörülü Denetim, 4. Ulusal Otomatik Kontrol Sempozyumu, sayfa. 31-42, 1987, Istanbul.
- [33] KAYNAK, O., Robot Kollarının Denetiminde Uyarlamalı

ve Öngörülü Yöntemler, 5. Ulusal Otomatik Kontrol Sempozyumu, sayfa 1-17, 1989, İstanbul.

- [34] KAYNAK, O., ALPTEKİN, A.R., Bir artık robotun öngörülü denetimi, Elektrik Mühendisliği 3. Ulusal Kongresi, sayfa 666-669, Eylül 1989, İstanbul.
- [35] CLARKE, D.W., and ZHANG, L., Long-Range Predictive Control Using Weighting-Sequence Models. Proc. IEE, 134, Pt. D(3): 187-195, 1987.
- [36] DE KEYSER, R.M.C, VAN DE VELDE, P.G.A and DUMORTIER, F.A.G., A Comparative Study of Self-Adapting Long-Range Predictive Control Methods, Automatica, 24(2): 149-163, 1988.
- [37] CLARKE, D.W., MOHTADI, C., and TUFFS, P.S., Generalized Predictive Control. Part 1: The Basic Algorithm, Automatica, 23(2): 137-148, 1987.
- [38] CLARKE, D.W., MOHTADI, C., and TUFFS, P.S., Generalized Predictive Control, Part 2: Extensions and Interpretations. Automatica, 23(2) : 149-160, 1987.
- [39] CLARKE, D.W., TUFFS, P.S., and MOHTADI, C., Generalized Predictive Control: A New Robust Self-Tuning Algorithm. In Landau, I.D., and L. Dugard(Eds). Commande Adaptive - Aspects Pratiques et Theoriques, pp. 209-228, Masson, Paris, 1987.
- [40] BORISON, U., Self-Tuning Regulator for a Class of Multivariable Systems, Automatica, 15: 209-215, 1979.
- [41] DRAPER, C. S., and LI, Y. T., Principles of Optimizing Control Systems and an Application to the Internal Combustion Engine, ASME Publication, 1951.
- [42] TSIEN, H.S., Engineering Cybernetics, Mc Graw-Hill, New York, 1954.
- [43] ASHBY, W.R., Design for a Brain, John Wiley, New York, 1960
- [44] BENNER, A.H., and DRENNICK, R.F., An Adaptive Servo System, IRE Conv. Rec., Pt.4, 8-14, 1955.
- [45] WHITAKER, H.P., YAMRON, J., and KEZER, A., Design of Model Reference Adaptive Control Systems for Aircraft, Report No. R-164, Instrumentation Lab., MIT, 1958.

- [46] LI, Y.T., and Van der Velde, W.E., The Philosophy of nonlinear adaptive control, Proc. IFAC Congress, Moscow 1960.
- [47] PETROV, B.N., ULANOV, G.M., and EMEL'YANOV, S.V., Invariant Nost i Optimizacia Sistem Avtomaticheskogo Regulirovania s Zsestoko i Peremennoi Stukturoui, Proc. IFAC Congress, Basel 1963.
- [48] BELLMAN, R., Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, N.J., 1961.
- [49] FEL'DBAUM, A.A., Dual Control Theory I-IV, Auto. Rem. Contr., 21:874-880; 21:1033-1039; 22:109-121, 1960-61.
- [50] LAINIOTIS, D.G., UPADHYAY, T.N., and DESHPANDE, J.G., Optimal Adaptive Control of Linear Systems, Proc. IEEE Conf. Decision & Contr., 247-257, 1971.
- [51] ASTROM, K.J., and WITTENMARK, B., On self-tuning regulators, Automatica, 9: 185-199, 1973.
- [52] ASTROM, K.J., and WITTENMARK, B., Adaptive Control, Lund Institute of Technology, Addison-Wesley Publishing Company.
- [53] KALMAN, R.E., Design of a Self-Optimizing Control System, Trans. ASME, 80: 468-478, 1958.
- [54] PETERKA, V., Adaptive Digital Regulation of Noisy Systems, Proc. IFAC Symp. on Identification and Process Parameter Estimation, Prague, 1970.
- [55] CLARKE, D.W., and GAWTHROP, P.J., Self-Tuning Controller, Proc. IEE, 122 : 929-934, 1975.
- [56] CLARKE, D.W., and GAWTHROP, P.J., Self-Tuning Control, Proc. IEE, 126 : 633-640, 1979.
- [57] GAWTHROP, P.J., Some Interpretations of the Self-Tuning Controller, Proc. IEE, 124: 889-894, 1977.
- [58] ASTROM, K.J., BORISON, U., LJUNG, L., and WITTENMARK, B., Theory and Applications of Self-Tuning Regulators, Automatica, 13: 457-476, 1977.
- [59] GRIMBLE, M.J., Weighted Minimum Variance Self-Tuning Control, Int. J. Contr., 36: 597-609, 1982.
- [60] PETERKA, V., Predictor Based Self-tuning Control,

Automatica, 20: 39-58, 1984.

- [61] TOIVONEN, H.T., Minimum Variance Control of First Order Systems with a Constraint on the Input Amplitude, IEEE Trans. Auto. Contr., AC-26: 556-558, 1981
- [62] WIESLANDER, J., and WITTENMARK, B., An Approach to Adaptive Control Using Real Time Identification, Automatica, 7: 211-217, 1971.
- [63] FUCHS, J.J.-J., Explicit Self-Tuning Methods, Proc. IEE, 127, Pt.D: 259-264, 1980.
- [64] WOUTERS, W.R.E., Adaptive Pole-Placement for Linear Stochastic Systems with Unknown Parameters, Preprints IEEE-CDC, New Orleans, pp. 159-166, 1977.
- [65] WELLSTEAD, P.E., PRAGER, D.L., and ZANKER, P.M., Pole-Assignment Self-Tuning Regulator, Proc.IEE, 126: 781-787, 1979.
- [66] ASTROM, K.J., WESTERBERG, B., and WITTENMARK, B., Self-Tuning Controllers Based on Pole-Placement Design, Report No. TFRT-3148, Dept. of Auto. Contr., Lund Inst. of Tech., Lund, Sweden, 1978.
- [67] ALLIDINA, A.Y., and HUGHES, F.M., Generalized Self-Tuning Controller with Pole Assignment, Proc. IEE, 127, Pt.D: 13-18, 1980.
- [68] VOGEL, E.F., and EDGAR, T.F., An adaptive Dead-Time Compensator for Process Control, Proc. JACC, Paper No. TP 5-E, 1980.
- [69] VOGEL, E.F., and EDGAR, T.F., Application of an Adaptive Pole-Zero Placement Controller to Chemical Processes with Variable Dead Time, Proc. JACC, Washington, D.C., 1982.
- [70] PETERKA, V., and ASTROM, K.J., Control of Multivariable Systems with Unknown but Constant Parameters, Proc. 3rd IFAC Symp. on Identification and Process Parameter Estimation, The Hague, pp. 535-544, 1973.
- [71] ZHAO-YING, Z., and ASTROM, K.J., A microcomputer Implementation of LQG Self-Tuner, Report No. TFRT-7226, Dept. of Auto. Contr., Lund Inst. of Tech., Lund, Sweden, 1981.
- [72] LAM, K.P., Implicit and Explicit Self-Tuning Controllers, Report No. 1334/80, Eng. Lab., Oxford University, 1980.

- [73] SALCUDEAN, S., and BALANGER, P.R., An Indirect Adaptive Regulator: Computational Algorithms and Simulation Results, Proc. 6th IFAC Symp. Identification and System Parameter Estimation, Washington, D.C., pp. 253-258, 1982.
- [74] GRIMBLE, M.J., Self-Tuners Based on LQG Theory, in Encyclopedia of Systems and Control, Pergamon Press, Oxford, 1983.
- [75] ASTROM, K.J., A Self-Tuning Regulator for Non-Minimum Phase Systems, Report No. TFRT-7411, Dept. of Auto. Contr., Lund Inst. of Tech., Lund, Sweden, 1974.
- [76] ASTROM, K.J., Direct Methods for Non-Minimum Phase Systems, Proc. 19th IEEE-CDC, Albuquerque, N.M., pp. 611-615, 1980.
- [77] SAMSON, C., An Adaptive LQ Controller for Non-Minimum Phase Systems, Int. J. Contr., 35: 81-86, 1983.
- [78] GRIMBLE, K.J., A Control Weighted Minimum Variance Controller for Non-Minimum Phase Systems, Int. J. Contr., 33: 751-762, 1981.
- [79] KUMAR, R., and MOORE, J.B., On Adaptive Minimum Variance Regulation for Non-Minimum Phase Plants, Automatica, 19: 449-451, 1983.
- [80] CLARKE, D.W., Self-Tuning Control of Non-Minimum Phase Systems, Automatica, 20: 501-517, 1984.
- [81] GAWTHROP, P.J., Hybrid Self-Tuning Control, Proc. IEE, 127, Pt.D: 229-236, 1980.
- [82] EL-BAGOURY, M.A., and BAYOUMI, M.M., Multivariable Self-Tuning Augmented Regulator, in Methods and Applications in Adaptive Control, Springer-Verlag, Berlin, 1980.
- [83] BAYOUMI, M.M., WONG, K.Y., and EL-BAGOURY, M.A., A Self-Tuning Regulator for Multivariable Systems, Automatica, 17: 575-592, 1981.
- [84] KOIVO, A.J., A Multivariable Self-Tuning Controller, Automatica, 16: 351-366, 1980.
- [85] PRAGER, D.L., and WELLSTEAD, P.E., Multivariable Pole-Assignment Regulator, Proc. IEE, 128, Pt.D: 9-18, 1981.
- [86] BEZANSON, L.W., and HARRIS, S.L., State Space Design of Multivariable Self-Tuning Regulator, Int. J. Contr., 39: 395-412, 1984.

- [87] PLACKETT, R.L., Some Theorems in Least Squares, *Biometrika*, 37, 149-157, 1950.
- [88] PETERKA, V., A Square Root Filter for Real Time Multivariate Regression, *Kybernetika*, 11, 53-67, 1975.
- [89] BIERMAN, G.J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.
- [90] KEVICZKY, L., and HETTHESSY, J., Self-Tuning Minimum Variance Control of MIMO Discrete Time Systems. *Autom. Control Theory and Appl.* 3, 11-17, 1977.
- [91] DE KEYSER, R.M.C., Model Based Predictive Control Toolbox, ESPRIT/CIM Europe Workshop on "Computer Integrated Design of Controlled Industrial Systems", Commission of the European Communities, pp. 35-56, Paris, 1990.
- [92] GOODWIN, G.C., and LONG, R.S., Generalization of Results on Multivariable Adaptive Control, *IEEE Trans. Autom. Control*, AC-25(6), 1980.
- [93] GARCIA, C.E., PRETT, D.M., and MORARI, M., Model Predictive Control Theory and Practice - a Survey, *Automatica*, 25(3), 1989.
- [94] LAM, K.P., Implicit and Explicit Self-Tuning Controllers, D. Phil. Thesis, Oxford University and OUEL report 1134180.
- [95] CLARKE, D.W., KANJILAL, P.P., and MOHTADI, C., A Generalized LQG Approach to Self-Tuning Control: Part I - Aspects of Design, *Int. J. Control*, Vol.41, No.6, pp. 1509-1523.
- [96] RICHALET, J., Rault, A., Testud, J.L., and PAPON, J., Model Predictive Heuristic Control: Applications to Industrial Processes. *Automatica*, 14(5): 413-428, 1978.
- [97] CUTLER, C.R. and RAMAKER, B.L., Dynamic Matrix Control - A Computer Control Algorithm, In *JACC*, San Francisco, 1980.
- [98] RICHALET, J., RAULT, A., TESTUD, J.L., and PAPON, J., Algorithmic Control of Industrial Processes, 4th IFAC Symp. on Identification and System Parameter Estimation, Tbilisi, 1976.
- [99] DE KEYSER, R.M.C., and VAN CAUWENBERGHE, A., A Self-Tuning Multistep Predictor Application, *Automatica*, 17, 167-174.

- [100] YDSTIE, B., Extended Horizon Adaptive Control, 9th IFAC World Congress, Budapest, Hungary, 1984.
- [101] KWON, W.H., and PEARSON, A.E., On Feedback Stabilisation of Time-Varying Discrete Systems, IEEE Trans. Autom. Control, Vol. AC-23, No.3, pp. 479-481, 1978.
- [102] MOSCA, E., ZAPPA, G., and MANFREDI, C., Multistep Horizon Self-Tuning Controllers: The MUSMAR Approach, In IFAC 9th World Congress, Budapest, Hungary, 1984.
- [103] KAILATH, T., Linear Systems, Prentice-Hall, 1980.
- [104] PETERKA, V., Predictor-Based Self-Tuning Control, Automatica, 20(1): 39-50, 1984.
- [105] DENAVIT, J., and HARTENBERG, R.S., A Kinematic Notation for Lower-Pair Mechanisms, J. Applied Mechanics, Vol.22, pp. 215-221, 1955.
- [106] LEE, C.S.G., Robot Arm Kinematics, Dynamics, and Control, Computer, Vol. 15, pp. 62-80, 1982.
- [107] BEJCZY, A.K., Robot Arm Dynamics and Control, Technical MEMO, 33-669, Jet Propulsion Laboratory, 1974.
- [108] PAUL, R.P., Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm, Stanford Artificial Intelligence Laboratory MEMO AM-177, 1972.
- [109] LEE, C.S.G., LEE, B.H., and NIGAM, R., An Efficient Formulation of Robot Arm Dynamics for Control Analysis and Manipulator Design, Technical Report RSD-TR8-82, Center for Robotics and Integrated Manufacturing, University of Michigan, 1982.
- [110] PAUL, R.P., Robot Manipulators: Mathematics, Programming and Control, MIT Press., 1981.
- [111] HOLLERBACH, J.M., A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity, IEEE Trans. on. Systems, Man, and Cybernetics, vol.SMC-10, No.11. pp. 730-736, 1980.
- [112] LUH, J.Y.S., WALKER, M.W., and PAUL, R.P.C., On-Line Computational Scheme for Mechanical Manipulators, Transaction of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol.120, pp. 69-76, 1980.

- [1113] ORIN, D.E., Mc GHEE, R.B., VUKOBRATOVIC, M., and HARTOCH, G., Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods, Mathematical Biosciences, Vol.43, pp. 107-130, 1979.
- [1114] CRAIG, J.J., Introduction to Robotics-Mechanics and Control, Addison-Wesley Publishing Company, Company, MA, 1986.
- [1115] KAO, C.K., Sinha, A., and MAHALANABIS, A.K., A Digital Algorithm for Near-Minimum-Time Control of Robot Manipulators, Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control, Vol.109, pp.320-327, December 1987.
- [1116] ÖZSOY, C., and İÇLİ, H., Self-Tuning Controllers for Robotic Manipulators and Comparison with PID Controllers, Proceedings of Symposium on Electronics Technology 90, pp. 311-321, September 1990, Budapest, Hungary..
- [1117] KOIVO, A.J., Robotic Manipulator with Self-Tuning Controller Operating in Cartesian Coordinates, Proc. of the Third Yale Workshop on Applications of Adaptive System Theory, June 1983, pp. 174-178.
- [1118] LEE, C.S.G., and LEE, B.H., Resolved Motion Adaptive Control for Mechanical Manipulators, ASME Journal of Dynamic Systems, Measurement, and Control, Vol.106, No.2, pp. 134-142, June 1984.
- [1119] ÖZSOY, C., KAZAN, R., Self-tuning Predictive Controllers for Robotic Manipulators, Proceedings 2.nd Project Workshop on CIM and Robotics Applications, pp. 99-114, September 1991, Belgrad.
- [1120] ÖZSOY, C., KAZAN, R., Adaptif Öz Uyumlamalı ve Öngörülü Kontrolün Bir Robotik Manipülatöre Uygulanması ve Karşılaştırılması, 5. Ulusal Makina Teorisi Sempozyumu, sayfa 455-464, Eylül 1991, Uludağ, Bursa.
- [1121] LANDAU, Y.D., Adaptive Control, The Model Reference Approach., Marcel Dekker, Inc., 1979
- [1122] CHALAM, V.V., Adaptive Control Systems, Techniques and Applications, Marcel Dekker, Inc., 1987.
- [1123] ASTRÖM, K.J., WITTENMARK, B., Adaptive Control. Addison-Wesley Publishing Company.

- [124] FU, K.S., GONZALES, R.C. and LEE, C.S.G., Robotics: Control, Sensing, Vision, and Intelligence, Mc Graw-Hill Book Company, 1987.
- [125] INIGO, R.M., and MORTON, J.S., Simulation of the Dynamics of an Industrial Robot, IEEE Transactions on Education, Vol. 34, No.1, 89-98, February 1991.



EK A. MIMO SİSTEMLER İÇİN ÖNGÖRÜLÜ KONTROL ALGORİTMALARINDA KULLANILAN E,F VE G POLİNOM PARAMETRELERİNİN HESAPLANMASI VE KONTROL BÜYÜKLÜĞÜNÜN TEŞKİLİ

SISO sistemler için yapılan hesaplamaların bir benzeri, MIMO sistemler için tek boyutlu ifadeler yerine matrisler kullanılarak elde edilmektedir. Bu bölümde basitlik olması için 1.dereceden bir proses modeli göz önüne alınacaktır. Ayrıca proses modelinin üç giriş ve üç çıkış büyüklüğüne sahip olduğu kabul edilmiştir (Üç eklemlili robot modeline uygulamayı kolaylıkla görebilmek için). Bu şartlarda proses modeli;

$$([I] + [A_1](q^{-1})) y(t) = ([B_0] + [B_1](q^{-1})) u(t-1)$$

şeklinde ifade edilmektedir. $[I]$, (3x3) boyutlarında birim matris,

$$A_1 = \begin{bmatrix} a_{11}^1 & a_{12}^1 & a_{13}^1 \\ a_{21}^1 & a_{22}^1 & a_{23}^1 \\ a_{31}^1 & a_{32}^1 & a_{33}^1 \end{bmatrix}, B_0 = \begin{bmatrix} b_{11}^0 & b_{12}^0 & b_{13}^0 \\ b_{21}^0 & b_{22}^0 & b_{23}^0 \\ b_{31}^0 & b_{32}^0 & b_{33}^0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} b_{11}^1 & b_{12}^1 & b_{13}^1 \\ b_{21}^1 & b_{22}^1 & b_{23}^1 \\ b_{31}^1 & b_{32}^1 & b_{33}^1 \end{bmatrix}$$

(3x3) boyutlarında matrislerdir.

EK A.1. E Polinom Parametrelerinin Çıkarılması

(4.17) nolu

$$R(q^{-1}) = E_{j+1}(q^{-1}) = E_j(q^{-1}) + q^{-j}r_j$$

bağıntısı ve $E_1 = [I]$ başlangıç şartı göz önüne alınarak

$$E_1 = E_{10} = [I]$$

$$E_2 = E_1 + q^{-1}r_1 \quad (r_1 = F_{10})$$

$$E_{20} + E_{21}q^{-1} = E_1 + q^{-1}F_{10} \text{ elde edilir.}$$

Sol tarafın sağ tarafa eşit olması gerektiğinden,

$$E_{20} = E_1 = [I], \quad E_{21} = F_{10}$$

elde edilir. Aynı şekilde;

$$E_3 = E_2 + q^{-2}r_2 \quad (r_2 = F_{20})$$

$$E_{30} + E_{31}q^{-1} + E_{32}q^{-2} = E_{20} + E_{21}q^{-1} + q^{-2}F_{20}$$

$$E_{30} = E_{20} = [I], \quad E_{31} = E_{21} = F_{10}, \quad E_{32} = F_{20}$$

ve

$$E_4 = E_3 + q^{-3}r_3$$

$$E_{40} + E_{41}q^{-1} + E_{42}q^{-2} + E_{43}q^{-3} = E_{30} + E_{31}q^{-1} +$$

$$+ E_{32}q^{-2} + q^{-3}F_{30}$$

$$E_{40} = E_{30} = [I], \quad E_{41} = E_{31} = F_{10}, \quad E_{42} = E_{32} = F_{20}$$

$$E_{43} = F_{30}$$

olarak bulunur.

Genelleştirme yapılacak olursa;

$$E_{10} = E_{20} = E_{30} = \dots = E(0) = [I]$$

$$E_{21} = E_{31} = E_{41} = \dots = E(1) = FO(1)$$

$$E_{32} = E_{42} = E_{52} = \dots = E(2) = FO(2)$$

...

$E(I) = FO(I)$ elde edilir.

$E(I)$ matrisleri (3×3) boyutlarındadır. $FO(I)$ matrisleri Ek A2' de çıkarılacaktır.

EK A.2. F Polinom Parametrelerinin Çıkarılması

(4.14.b nolu

$$F_{j+1} = q(F_j - A(1-q^{-1})r_j)$$

ifadesi kullanılarak

$$F_1 = q(I - A(1-q^{-1})) \quad (\text{ilk tanım})$$

$$= q(I - (I + A_1 q^{-1})(1-q^{-1}))$$

$$F_{10} + F_{11}q^{-1} = I - A_1 + A_1q^{-1}$$

elde edilir. Sol tarafın sağ tarafa eşit olması gerektiğinden,

$$F_{10} = I - A_1 \rightarrow FO(1)$$

$$F_{11} = A_1 \rightarrow F1(1)$$

elde edilir. Benzer şekilde,

$$F_2 = q(F_1 - A(1-q^{-1})r_1) \quad (r_1 = F_{10})$$

$$F_{20} + F_{21}q^{-1} = F_{11} - A_1 F_{10} + F_{10} + A_1 F_{10}q^{-1}$$

$$F_{20} = F_{11} - A_1 F_{10} + F_{10} \rightarrow FO(2)$$

$$F_{21} = A_1 F_{10} \rightarrow F1(2) ,$$

$$F_3 = q(F_2 - A(1-q^{-1})r_2) \quad (r_2 = F_{20})$$

$$F_{30} + F_{31}q^{-1} = F_{21} - A_1F_{20} + F_{20} + A_1q^{-1}F_{20}$$

$$F_{30} = F_{21} - A_1F_{20} + F_{20} \rightarrow F0(3)$$

$$F_{31} = A_1F_{20} \rightarrow F1(3)$$

Genelleştirme yapılırsa;

$$F0(0) = I, F1(0) = 0 \quad (\text{Başlangıç şartları})$$

$$F0(I) = F0(I-1) + F1(I-1) - A_1F0(I-1)$$

$$F1(I) = A_1F0(I-1)$$

olarak elde edilir.

EK A.3. G Polinom Parametrelerinin Çıkarılması

(4.18) nolu $G_{j+1} = B(q^{-1})E_{j+1}$ ifadesi kullanılarak

$$G_1 = B E_1 \quad (E_1 = I)$$

$$G_1 = (B_0 + B_1q^{-1}) I$$

$$G_1 = G_{10} + G_{11}q^{-1} = B_0I + B_1Iq^{-1} = B_0 + B_1q^{-1}$$

elde edilir. Sol tarafın sağ tarafa eşit olması gerekeceğinden,

$$G_{10} = B_0I = B_0 \rightarrow G(0)$$

$$G_{11} = B_1I = B_1 \rightarrow G1(1)$$

Benzer şekilde devam edilerek,

$$G_2 = B E_2$$

$$= (B_0 + B_1 q^{-1}) (E_{20} + E_{21} q^{-1})$$

$$G_{20} + G_{21} q^{-1} + G_{22} q^{-2} = B_0 E_{20} + B_0 E_{21} q^{-1} + B_1 E_{20} q^{-1} + B_1 E_{21} q^{-2}$$

$$G_{20} = B_0 E_{20} = B(0) \rightarrow G(0)$$

$$G_{21} = B_1 E_{20} + B_0 E_{21}$$

$$= B_1 + B_0 F_{10} \rightarrow G(1)$$

$$G_{22} = B_1 E_{21} = B_1 F_{10} \rightarrow GI(2)$$

$$G_3 = B E_3$$

$$= (B_0 + B_1 q^{-1}) (E_{30} + E_{31} q^{-1} + E_{32} q^{-2})$$

$$G_{30} = B_0 E_{30} = B_0 \rightarrow G(0)$$

$$G_{31} = B_1 E_{30} + B_0 E_{31} = B_1 + B_0 F_{10} \rightarrow G(1)$$

$$G_{32} = B_1 E_{31} + B_0 E_{32}$$

$$= B_1 F_{10} + B_0 F_{20} \rightarrow G(2)$$

$$G_{33} = B_1 E_{32} = B_1 F_{20} \rightarrow GI(3)$$

elde edilir.

(1. mertebeden bir proses modeli için kontrolör parametrelerinin elde edilmesinde sadece G ve GI' ler gerekli olmaktadır).

Genelleştirilme yapılacak olursa,

$$G_{10} = G_{20} = G_{30} = G(0)$$

$$G_{21} = G_{31} = G_{41} = G(1)$$

$$G_{32} = G_{42} = G_{52} = G(2)$$

$$G_{11} = GI(1)$$

$$G_{22} = GI(2)$$

$$G_{33} = GI(3) \text{ bilgileri de göz önüne alınarak}$$

$$G(0) = B_0$$

$$G(I) = B_0 E(I) + B_1 E(I-1)$$

$$GI(I) = B_1 E(I-1)$$

elde edilir. Model mertebesi arttıkça Fortran dilinde yazılan programlarda GII, GIII, v.s. simgelerle gösterilen G_{43} , G_{54} , G_{65} , ... gibi ilave terimler gerekecektir. Bu durum yazılan programlara SISO durum için ilave edilmiştir. MIMO durum için de benzer ilaveler yapılabilir.

EK A.4. u Kontrol Büyüklüğünün Teşkil Edilmesi

u kontrol büyüklüğünün çıkarılması için tek boyutlu olarak verilen $F0$, $F1$, G ve GI matrisleri aşağıdaki şekilde iki boyutlu $FF0$, $FF1$, GG ve GGI matrislerine atanmaktadır. Bu, işlemlerin kolaylığı açısından gerekmektedir. Çünkü her bir matrisin elemanları da matrislerle ifade edildiğinden ikinci bir boyuta gerek duyulmaktadır. $N_2 = 3$ için $FF0$, $FF1$, GG ve GGI matrisleri aşağıdaki şekilde tanımlanmaktadır.

$$FF0 = \begin{bmatrix} F0(1) \\ F0(2) \\ F0(3) \end{bmatrix} = FF0(I, J) = FF0(3, 9)$$

$(I=1, N_2; J=1, L \times L)$

$$FF1 = \begin{bmatrix} F1 \langle 1 \rangle \\ F1 \langle 2 \rangle \\ F1 \langle 3 \rangle \end{bmatrix} = FF1 \langle I, J \rangle = FF1 \langle 3, 9 \rangle$$

$$\langle I=1, N_2; J=1, L \times L \rangle$$

$$GG = \begin{bmatrix} G \langle 0 \rangle \\ G \langle 1 \rangle \\ G \langle 2 \rangle \end{bmatrix} = GG \langle I, J \rangle = GG \langle 3, 9 \rangle$$

$$\langle I=1, N_2; J=1, L \times L \rangle$$

$$GGI = \begin{bmatrix} GI \langle 1 \rangle \\ GI \langle 2 \rangle \\ GI \langle 3 \rangle \end{bmatrix} = GGI \langle I, J \rangle = GGI \langle 3, 9 \rangle$$

$$\langle I=1, N_2; J=1, L \times L \rangle$$

G matrisi $N_2 = 3$ ve $N_u = 1$ için AG adı verilerek,

$$AG = \begin{bmatrix} GG \langle 0 \rangle \\ GG \langle 1 \rangle \\ GG \langle 2 \rangle \end{bmatrix} \quad [9 \times 3]$$

şeklinde dizayn edilmektedir. Ayrıca GG matrisleri de

$$GG \langle 0 \rangle = \begin{bmatrix} GG \langle 0,1 \rangle & GG \langle 0,4 \rangle & GG \langle 0,7 \rangle \\ GG \langle 0,2 \rangle & GG \langle 0,5 \rangle & GG \langle 0,8 \rangle \\ GG \langle 0,3 \rangle & GG \langle 0,6 \rangle & GG \langle 0,9 \rangle \end{bmatrix}$$

şeklinde 9 elemandan meydana gelmektedir. Böylece AG matrisi;

$$AG = \begin{bmatrix} GG(0,1) & GG(0,4) & GG(0,7) \\ GG(0,2) & GG(0,5) & GG(0,8) \\ GG(0,3) & GG(0,6) & GG(0,9) \\ GG(1,1) & GG(1,4) & GG(1,7) \\ GG(1,2) & GG(1,5) & GG(1,8) \\ GG(1,3) & GG(1,6) & GG(1,9) \\ GG(2,1) & GG(2,4) & GG(2,7) \\ GG(2,2) & GG(2,5) & GG(2,8) \\ GG(2,3) & GG(2,6) & GG(2,9) \end{bmatrix}$$

şeklinde elde edilir.

Artımlar cinsinden Δu kontrol büyüklüğü (4.32) ifadesinden de yararlanarak

$$\Delta u(t) = \begin{bmatrix} \tilde{u}_1(t) \\ \tilde{u}_2(t) \\ \tilde{u}_3(t) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} AG \end{bmatrix}^T \\ [9 \times 3]^T \end{bmatrix} \begin{bmatrix} \begin{bmatrix} AG \end{bmatrix} \\ [9 \times 3] \end{bmatrix} + \begin{bmatrix} \Lambda \end{bmatrix} \end{bmatrix}^{-1} \cdot$$

$$\cdot \begin{bmatrix} \begin{bmatrix} AG \end{bmatrix}^T \\ [9 \times 3]^T \end{bmatrix} \cdot \begin{bmatrix} \begin{bmatrix} YR \end{bmatrix} \\ [9 \times 1] \end{bmatrix} - \begin{bmatrix} F \end{bmatrix} \end{bmatrix} [3 \times 1]$$

şeklinde yazılabilir. Λ kontrol ağırlığı dizisidir.

Burada; $[YR]$ giriş referans vektörüdür ve,

$$YR = \begin{bmatrix} YR_1(t+1) \\ YR_2(t+1) \\ YR_3(t+1) \\ YR_1(t+2) \\ YR_2(t+2) \\ YR_3(t+2) \\ YR_1(t+3) \\ YR_2(t+3) \\ YR_3(t+3) \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

olarak ifade edilmektedir. [F] ise (4.21) nolu ifadeden de yararlanarak

$$F = \begin{bmatrix} [GGI] \\ [GI(1)] \\ [GI(2)] \\ [GI(3)] \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta u_1(t-1) \\ \Delta u_2(t-1) \\ \Delta u_3(t-1) \end{bmatrix} + \begin{bmatrix} [FF0] & [FF1] \\ [F0(1)] & [F1(1)] \\ [F0(2)] & [F1(2)] \\ [F0(3)] & [F1(3)] \end{bmatrix} \begin{bmatrix} YC \\ y_1(t) \\ y_2(t) \\ y_3(t) \\ y_1(t-1) \\ y_2(t-1) \\ y_3(t-1) \end{bmatrix}$$

$[9 \times 3] \quad [3 \times 1] \quad [9 \times 6] \quad [6 \times 1]$

[9x1] boyutlarında elde edilmektedir. GGI, Δu, FF0, FF1

ve YC bilgisayar programlarında kullanılan notasyonları göstermektedir.

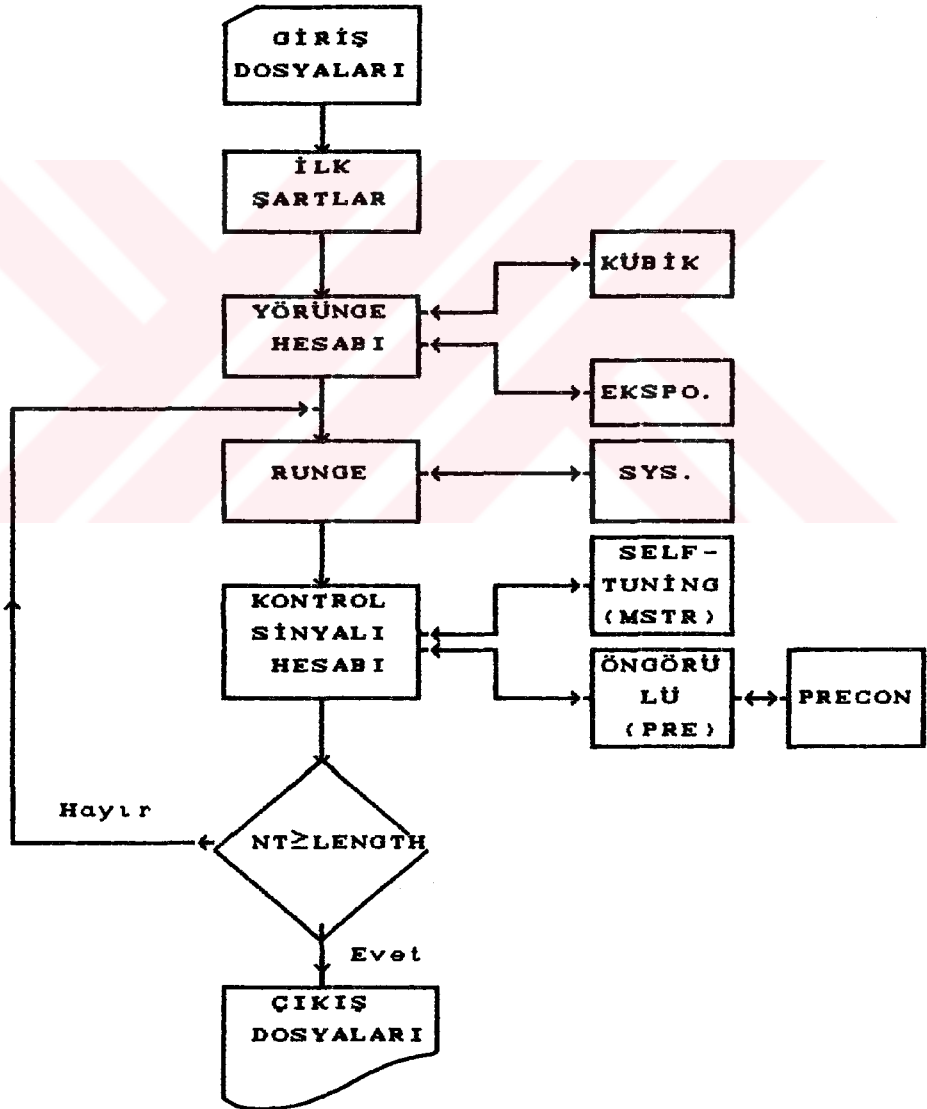
Elde edilen artımlar cinsinden ifade edilmiş Δu kontrol büyüklüklerinden

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} u_1(t-1) \\ u_2(t-1) \\ u_3(t-1) \end{bmatrix} + \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \\ \Delta u_3(t) \end{bmatrix}$$

şeklinde kontrol büyüklükleri elde edilir. Sadece ilk kontrol büyüklüğü ($u_i(t)$) işlem göreceğinden $N_u = 1$ için geçerli olan $u_i(t+2) = u_i(t+1) = u_i(t)$ eşitliğine gerek kalmayacaktır.

EK B. ROBOT YÖRÜNGE KONTROLÜ SİMÜLASYON PROGRAMLARI

EK B.1. Robot Yörünge Kontrolü Simülasyon Programları Genel Akış Şeması



Şekil Ek B.1. Robot Yörünge Kontrolü Simülasyon Akış Şeması

Ek B.2. Robot Yörünge Kontrolü Simülasyon Programlarının Tanıtılması

Ek B.2.1. Ana Program

Yörünge kontrol algoritmasının etkinliğini sınamak için kübik ve eksponansiyel yörüngelerin izlenmesi problemlerinin bilgisayarda simülasyonu gerçekleştirilmiştir. Bu iş için FORTRAN 77 programlama dili kullanılmış ve EPSON PC ve Wearnes CL286-12 kişisel bilgisayarlarından faydalanılmıştır.

İki ayrı kontrol algoritması kullanılabilmektedir.

1- Kendi kendini ayarlayan (Self-tuning) kontrol algoritması (MSTR alt programı)

2- Öngörülü kontrol algoritması (PRE alt programı)

Şekil Ek B.1' de verilen akış şemasından da görüldüğü gibi ana programda girdi okunması ve dosyalama dışında sayısal integrasyon çevrimi gerçekleştirilmektedir. Bu çevrim içerisinde manipülatörün kapalı form dinamik denklemleri integre edilmekte ve kontrol sağlanmaktadır. Ana programda alt programlar arası çağırımlar yönlendirilmekte ve çevrimin her adımında bulunan sonuç ve hata değerleri dosyalanmaktadır.

Ek B.2.2. Girdiler

Girdi olarak kullanılan manipülatör yapısal parametreleri program içerisinde şu isimlerle kullanılmıştır.

ALFA = α (Robot parametresi)
 AI = a (Robot parametresi)
 AD = d (Robot parametresi)
 AM1, AM2, AM3 = Robot uzuv kütleleri
 GG1, GG2, GG3 = Uzuvarın yerçekimi ivmesi vektörleri
 S1, S2, S3 = Uzuvarın ağırlık merkezinin koordinatları
 AKI = Uzuvarın atalet matrisi elemanları
 L1, L2, L3 = Uzuvarın uzunluğu
 V1, V2, V3, V4, V5, V6 = Eklem sürtünme katsayıları (kuru ve viskoz)
 PL = Taşınan yük
 GP = Taşınan yükün yerçekimi ivme vektörü

Bunların dışında simülasyon sırasında kullanılan ve ana programda okutulan şu girdilerde bulunmaktadır.

NEQ = Durum büyüklükleri sayısı
 DT = İntegrasyon adımı
 NS = Uzu v sayısı
 IN = Eklem tipini belirten kontrol değışkeni
 N2 = Öngörü ufuđu
 NUF = Kontrol ufuđu
 ALAMDA = λ (Kontrol ağırlığı dizisi)
 IKON = Eklem veya kartezyen uzay kontrolu seçim değışkeni
 IYOR = Yörtünge seçimi değışkeni
 LENGTH = Simülasyon uzunluđu
 L = Değışken sayısı (giriş/çıkış)
 ICON = Kontrol algoritması (Self-tuning ve öngörü lü) seçim değışkeni
 DO = Ö lü zaman
 NX = Ayrık zaman modeli parametre sayısı
 STD = Stokastik bozucu standart sapması
 AMEAN = Stokastik bozucu ortalama değeri
 IYR = Referans yörtünge değışim aralığı
 IEST = Parametre kestirme algoritması seçim değışkeni
 MO = Ayrık zaman modeli B polinomunun mertebesi
 NO = Ayrık zaman modeli A polinomunun mertebesi
 LO = Ayrık zaman modeli C polinomunun mertebesi
 PO = Ayrık zaman modeli parametreleri
 Q = Self-tuning kontrol algoritmasında ağırlık matrisi
 UMIN,UMAX = Minimum ve maksimum kontrol büyüklükleri (sınırlandırma için).

Ek B.2.3. Yörtünge Hesabı

Ana program içerisinde çevrim dışı olarak yörtünge hesabı yapılmaktadır. Tek giriş-tek çıkışlı durum (SISO) kullanıldığında hızlar YRV1, YRV2, YRV3, konumlar ise YRV4, YRV5, YRV6 olarak isimlendirilmektedir. Çok girişçok çıkışlı durum (MIMO) kullanıldığında ise hız yörtü ngeleri YRV(I,1), YRV(I,2), YRV(I,3), konum yörtü ngeleri ise YRV(I,4), YRV(I,5) YRV(I,6) olarak isimlendirilmiştir. MIMO durumda kartezyen uzayda yörtünge kontrolu yapılacak ise önce eklemlerin yörtü ngeleri konum için TETA, hız için TETAT isimleri ile hesaplanmakta, daha sonra dü z kinematik ifadeler kullanılarak YRV hız ve konumları (kartezyen hız ve konum- X,Y,Z ve $\dot{X},\dot{Y},\dot{Z}$) elde edilmektedir.

Ek B.2.4. RUNGE Alt Programı

Robotik manipülatörün Lagrange-Euler formundaki dinamik denklemleri FORTRAN 77 dilinde yazılmış Runge-Kutta IV yöntemi ile integre edilmektedir. Bu

program her bir simülasyon adımımda SYST alt programını çağırarak kapalı formdaki dinamik denklemler oluşturulmaktadır.

Ek B.2.5. Dinamik Davranış (SYST) Alt Programı

Bu alt programda Lagrange-Euler formülasyonu kullanılarak sistemin durum değişkenlerinin zamana göre türevleri bulunmakta ve Runge-Kutta IV sayısal integrasyon alt programı sürekli olarak bu programı çağırarak integrasyon yapıp konum ve hız değerlerini hesaplamaktadır. Bu hesaplama işleminde kontrol alt programlarından gelen (MSTR veya PRE) U1, U2, U3 manipülatöre verilmesi gereken sürüş moment değerleri kullanılmaktadır.

Ek B.2.6. Kontrol Sinyalinin Hesaplanması

a- SISO durum

Self-tuning ve öngörülü kontrol algoritmalarının seçilmesi mümkündür. Self-tuning algoritmasına göre kontrol sinyali hesaplanıyor ise her bir eklem için MSTR1, MSTR2, MSTR3 alt programları yardımı ile 3 adet eklem motoruna uygulanacak torklar hesaplanmaktadır. Algoritma gereği RLS, SQR veya RSQR parametre kestirme algoritmalarından biri kullanılarak parametreler kestirilmekte, SHIFT alt programı yardımı ile çıkış ve kontrol büyüklükleri ileride kullanılmak üzere geri kaydırılmakta, SAT alt programı ile kontrol büyüklüğünün belirli limitler arasında (UMIN ve UMAX) bulunması için saturasyon yapılmaktadır.

Öngörülü kontrol algoritmasının seçilmesi durumunda her bir eklem için PRE1, PRE2 ve PRE3 alt programları yardımı ile RLS, SQR veya RSQR parametre kestirme algoritmalarından biri seçilmekte, PRECON1, PRECON2 ve PRECON3 alt programları ile kontrol büyüklüğü hesaplanmakta, SHIFT alt programları ile çıkış ve kontrol büyüklükleri kaydırılmakta ve yine SAT alt programı ile kontrol büyüklüğü saturasyonu yapılmaktadır.

b- MIMO durum

Üç eklem için tek bir kontrolör dizayn edileceğinden çok giriş kullanılarak, yukarıda yapılan işlemler self-tuning kontrol algoritması kullanma durumunda MSTR ve öngörülü kontrol algoritması kullanma durumunda PRE

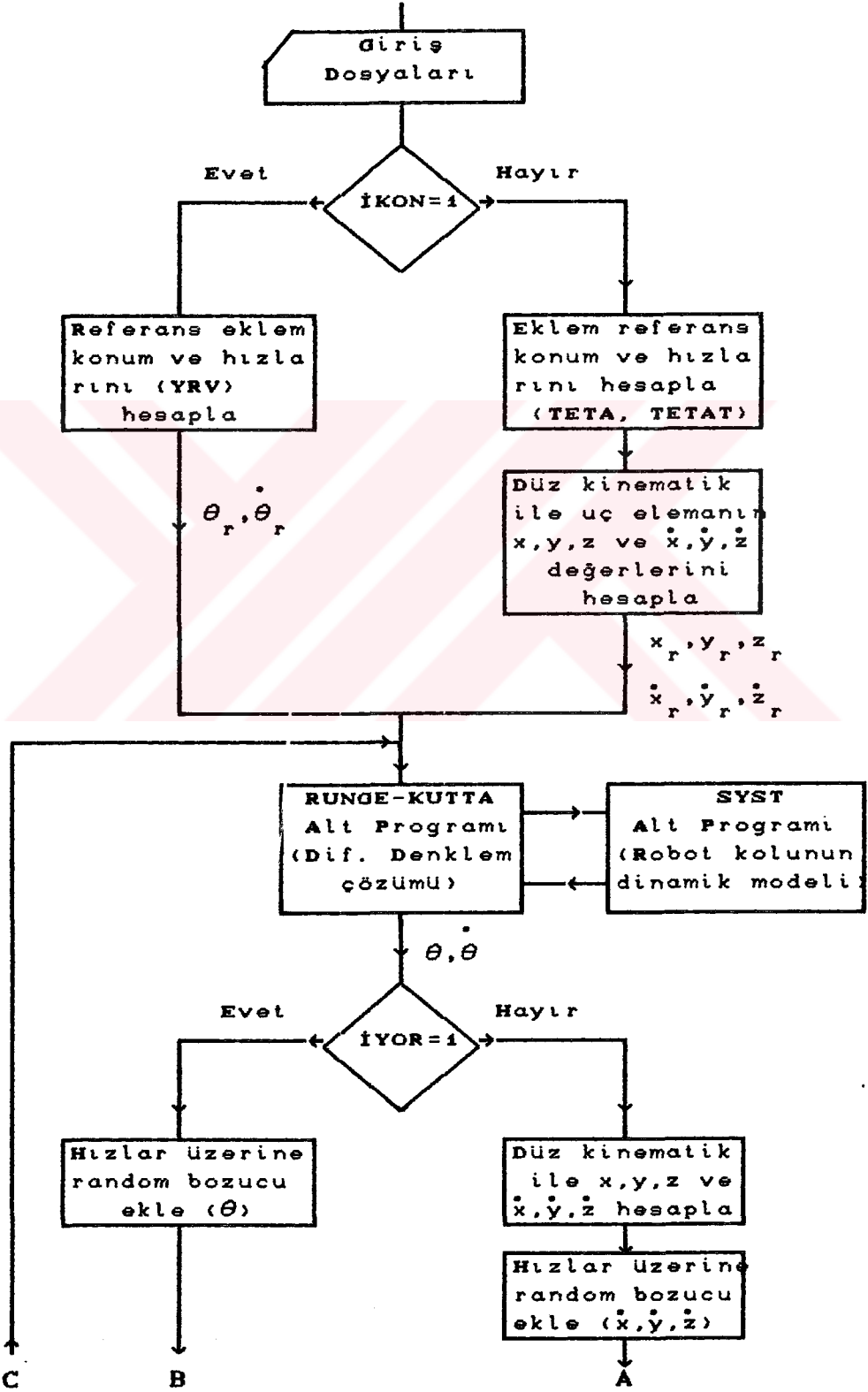
ve PRECON alt programları yardımı ile gerçekleştirilmektedir.

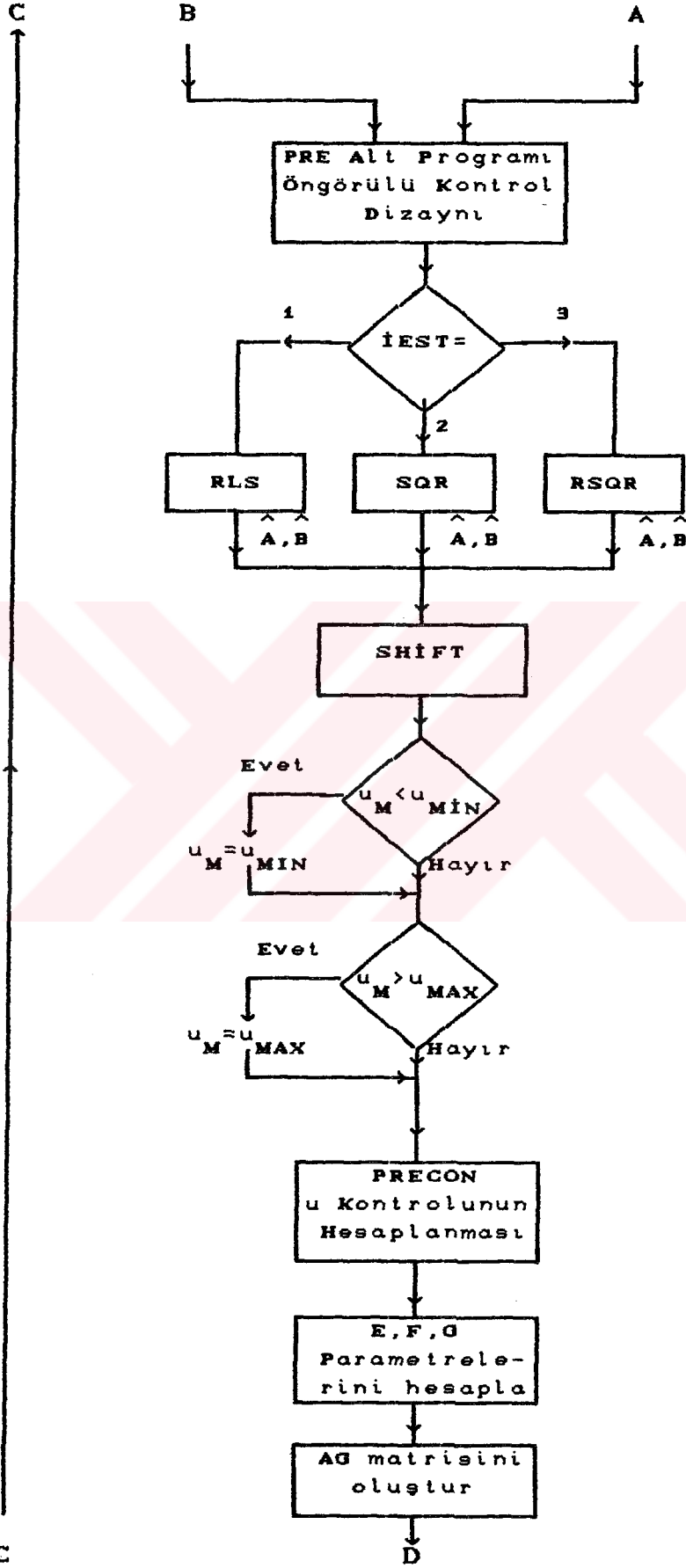
Ek B.2.7. Çıkış Dosyaları

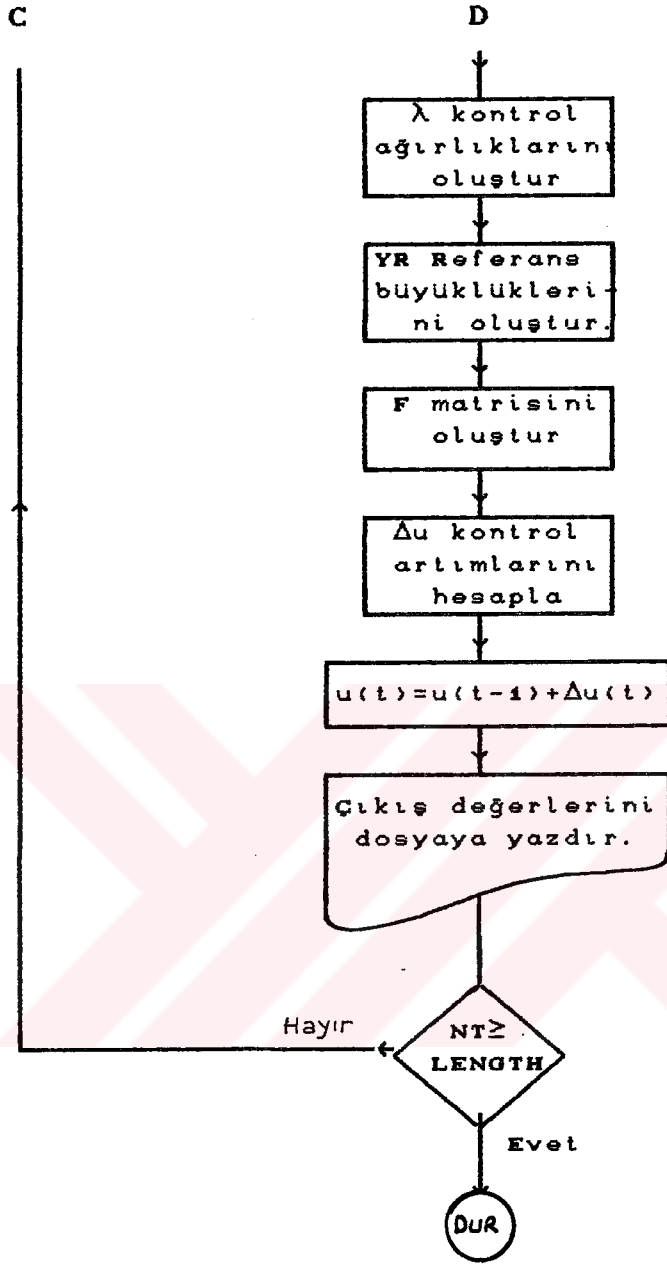
Kontrol performansını araştırmak amacı ile gerekli eğrilerin çizdirilmesinde ve hata tablolarının oluşturulmasında kullanılacak büyüklüklerin saklanacağı dosyalar olarak ifade edilmektedir.



Ek B.2.8. Çok giriş-çok çıkışlı öngörülü robot kontrolü için kullanılan bilgisayar programlarının detaylı akış şeması







EK B. 3.

```

C*****
C#
C#          ROBOT SIMULASYONU PROGRAMI (LAG-RB3P.FOR)
C# UC EKLEMLI BIR ROBOT ICIN ONGORULU KONTROL ALGORITMASINA GORE CALISIR
C#          KARTEZYEN UZAYDA DA KONTROL YAPABILMEKTEDIR
C#          RECEP KAZAN
C#
C#          KULLANILAN YARDIMCI PROGRAMLAR: MLIB, LAG-RKGP (FOR)
C#
C*****

```

```

PROGRAM LAG-RB3P
REAL P0(40),X(40),DX(40),Y(6),U(40),UDO(6),EM(40),C(40)
REAL XX(6),XD(6),L1,L2,L3,YY(6)
REAL YRV(510,6),YRP(20),YRD(20)
INTEGER D,D0,T2
COMMON/PNT/QP(40),NP,NQ,NQP,YRM1(40),NR,YRM2(40),YRM3(40)
COMMON/B1/YR(1),Q(4),W,P(40),UM(40),YM(40),R(200),S0,WMIN
COMMON/B2/D,M,N,NZ
COMMON/B3/UMIN,UMAX
COMMON/SED/ISEED1,ISEED2
COMMON/B5/A(200),B(40)
COMMON/STR/ICON,IEST
COMMON/AL/L1,L2,L3
COMMON/V/V1,V2,V3,V4,V5,V6,PL,GP(3),NT
COMMON/AA/NS,IN(3),ALFA(3),AI(3),AD(3)
COMMON/MM/AM1(1),AM2(1),AM3(1),GG1(4),GG2(4),GG3(4)
COMMON/BB/AKI(3,9),S1(4),S2(4),S3(4)
COMMON/CC/SX(3),SY(3),SZ(3)
COMMON/QPP/QP1(4),QP2(4),QP3(4)
COMMON/XXM/XX,YRDM1(6),YRDM2(6),YRDM3(6),C1(4),C2(4),C3(4)
COMMON/TTS/TS
COMMON/AK/AK1,AK2,AK3,AK4,AK5,AK6
COMMON/B/A1,A2,A3,A4,A5,A6

```

C
C
C

VERILERIN OKUTULMASI

```

READ(1,*) NEO,DT,TT
READ(1,*) (XX(I),I=1,NEO)
READ(1,*) NS,(IN(I),I=1,NS)
READ(1,*) (ALFA(I),I=1,NS)
READ(1,*) (AI(I),I=1,NS)
READ(1,*) (AD(I),I=1,NS)
READ(1,*) AM1(1),AM2(1),AM3(1)
READ(1,*) (GG1(I),I=1,4)
READ(1,*) (GG2(I),I=1,4)
READ(1,*) (GG3(I),I=1,4)
READ(1,*) (S1(K),K=1,4)
READ(1,*) (S2(K),K=1,4)
READ(1,*) (S3(K),K=1,4)
READ(1,*) ((AKI(I,J),J=1,9),I=1,3)
READ(1,*) L1,L2,L3
READ(1,*) V1,V2,V3,V4,V5,V6,PL
READ(1,*) (GP(I),I=1,3)

```

C

```

SX(1)=S1(1)
SX(2)=S2(1)
SX(3)=S3(1)
SY(1)=S1(2)
SY(2)=S2(2)
SY(3)=S3(2)
SZ(1)=S1(3)

```

c

```

SZ(3)=S3(3)
C
  READ(1,*) N2,NUF,ALAMDA
  READ(1,15) LENGTH,L,ICON,IPO,ICHS
15 FORMAT(10I5)
  GO TO (1000,2000) IPO
C
C   AYRIK ZAMANLI BIR SISTEMIN GIRIS BILGILERI
C
1000 READ(1,15) DO,M0,N0,L0
  NX=(M0+N0+L0+1)*L
  DO 10 I=1,L
10 READ(1,25) (PO(I+(J-1)*L),J=1,NX)
25 FORMAT(7F10.4)
  READ(1,25) STD
  GO TO 3000
C
C   SUREKLI ZAMANLI BIR SISTEMIN GIRIS BILGILERI
C
2000 READ(1,15) DO,NX,NSTEP
  READ(1,*) STD,AMEAN
  READ(1,15) NTYPE
C
C   REFERANS YORUNGELERININ OLUSTURULMASI
C
3000 READ(1,15) IYR
  NYR=(1+(LENGTH-DO))/IYR
  IA=0
C
C   IYOR=1 EKLEM KONTROLU
C   IYOR=2 KARTEZYEN KONTROL
C
C   IKON=1 KUBIK YORUNGE
C   IKON=2 EKSPONANSİYEL YORUNGE
C
  GO TO (700,810) IYOR
700 GO TO (800,810) IKON
800 DO 20 I=1,NYR/2,1
  TT=I*IYR*DT
C
C   HIZLAR
  YRV(I+1,1)=1.5708
  YRV(I+1,2)=-25.1328*TT**2+12.5664*TT
  YRV(I+1,3)=37.6992*TT**2-18.8496*TT
C
C   KONUMLAR
  YRV(I+1,4)=1.5708*TT
  YRV(I+1,5)=-8.3776*TT**3+6.2832*TT**2+0.5236
20 YRV(I+1,6)=12.5664*TT**3-9.4248*TT**2
  DO 21 I=NYR/2+1,NYR+N2,1
  TT=I*IYR*DT
  YRV(I,1)=1.5708
  YRV(I,2)=12.5664*TT**2-18.8496*TT+6.2832
  YRV(I,3)=-54.45456*TT**2+81.68184*TT-27.22728
  YRV(I,4)=1.5708*TT
  YRV(I,5)=4.1888*TT**3-9.4248*TT**2+6.2832*TT-0.2618
21 YRV(I,6)=-18.15152*TT**3+40.84092*TT**2-27.22728*TT+4.88695
  GO TO 801
810 DO 223 I=0,NYR+N2,1
  TT=I*IYR*DT
C
C   KONUMLAR
  YRV(I+1,4)=0.78539*(1+6*EXP(-TT/0.6))-8*EXP(-TT/0.8))
  YRV(I+1,5)=YRV(I+1,4)
  YRV(I+1,6)=YRV(I+1,4)
C
C   HIZLAR

```

```

YRV(I+1,1)=7.8539*(-EXP(-TT/0.6)+EXP(-TT/0.8))
YRV(I+1,2)=YRV(I+1,1)
223 YRV(I+1,3)=YRV(I+1,1)
GO TO 801
C
C   KARTEZYEN KONTROL
C
C   IKON=1 KUBIK YORUNGE
C   IKON=2 EKSPONANSIYEL YORUNGE
C
710 GO TO (900,910) IKON
900 DO 2220 I=0,NYR/2,1
    TT=I*IYR*DT
C   HIZLAR
    TETA1T=1.5708
    TETA2T=-25.1328*TT**2+12.5664*TT
    TETA3T=37.6992*TT**2-18.8496*TT
C   KONUMLAR
    TETA1=1.5708*TT
    TETA2=-8.3776*TT**3+6.2832*TT**2+0.5236
    TETA3=12.5664*TT**3-9.4248*TT**2
C
    AY=L1+L2*COS(TETA2)+L3*COS(TETA2+TETA3)
    YRV(I,4)=AY*COS(TETA1)
    YRV(I,5)=AY*SIN(TETA1)
    YRV(I,6)=L2*SIN(TETA2)+L3*SIN(TETA2+TETA3)
C
    BY=-L2*SIN(TETA2)*TETA2T-L3*SIN(TETA2+TETA3)*(TETA2T+TETA3T)
    YRV(I,1)=-AY*SIN(TETA1)*TETA1T+COS(TETA1)*BY
    YRV(I,2)=AY*COS(TETA1)*TETA1T+SIN(TETA1)*BY
2220 YRV(I,3)=L2*COS(TETA2)*TETA2T+L3*COS(TETA2+TETA3)*
    *(TETA2T+TETA3T)
C
DO 2221 I=NYR/2+1,NYR+N2,1
    TT=I*IYR*DT
    TETA1T=1.5708
    TETA2T=12.5664*TT**2-18.8496*TT+6.2832
    TETA3T=-54.45456*TT**2+81.68184*TT-27.22728
    TETA1=1.5708*TT
    TETA2=4.1888*TT**3-9.4248*TT**2+6.2832*TT-0.2618
    TETA3=-18.15152*TT**3+40.84092*TT**2-27.22728*TT+4.88695
C
    AY=L1+L2*COS(TETA2)+L3*COS(TETA2+TETA3)
    YRV(I,4)=AY*COS(TETA1)
    YRV(I,5)=AY*SIN(TETA1)
    YRV(I,6)=L2*SIN(TETA2)+L3*SIN(TETA2+TETA3)
C
    BY=-L2*SIN(TETA2)*TETA2T-L3*SIN(TETA2+TETA3)*(TETA2T+TETA3T)
    YRV(I,1)=-AY*SIN(TETA1)*TETA1T+COS(TETA1)*BY
    YRV(I,2)=AY*COS(TETA1)*TETA1T+SIN(TETA1)*BY
2221 YRV(I,3)=L2*COS(TETA2)*TETA2T+L3*COS(TETA2+TETA3)*
    *(TETA2T+TETA3T)
GO TO 801
930 DO 2225 I=0,NYR+N2,1
    TT=I*IYR*DT
    TETA1=0.78539*(1+6*EXP(-TT/0.6)-8*EXP(-TT/0.8))
    TETA2=TETA1
    TETA3=TETA1
    TETA1T=7.8539*(-EXP(-TT/0.6)+EXP(-TT/0.8))
    TETA2T=TETA1T
    TETA3T=TETA1T
C
    AY=L1+L2*COS(TETA2)+L3*COS(TETA2+TETA3)

```

```

YRV(I,4)=AY*COS(TETA1)
YRV(I,5)=AY*SIN(TETA1)
YRV(I,6)=L2*SIN(TETA2)+L3*SIN(TETA2+TETA3)

C
BY=-L2*SIN(TETA2)*TETA2T-L3*SIN(TETA2+TETA3)*(TETA2T+TETA3T)
YRV(I,1)=-AY*SIN(TETA1)*TETA1T+COS(TETA1)*BY
YRV(I,2)=AY*COS(TETA1)*TETA1T+SIN(TETA1)*BY
2225 YRV(I,3)=L2*COS(TETA2)*TETA2T+L3*COS(TETA2+TETA3)*
      *(TETA2T+TETA3T)

C
C   PARAMETRE KESTIRME ALGORITMASININ GIRIS PARAMETRELERI
C
801 READ(1,15) IEST
    READ(1,15) D,M,N,NR,NQ,NP
    NZ=(M+N+1)*L
    IF(ICON.LT.3) GO TO 24
    NZ=(M+N+NR+2)*L
    NQP=(NQ+NP+2)*L
24  CONTINUE
    DO 31 I=1,L
31  READ(1,*) (P(I+(J-1)*L),J=1,NZ)
    READ(1,26) S,W,SO,WMIN
26  FORMAT (4F15.4)
    IF (IEST.EQ.6) S=0.0
    J=1
    DO 32 I=1,NZ
      R(J)=S
32  J=J+NZ+1

C
C   KONTROL BILGILERI
C
    IF (ICON-3) 33,34,34
33  READ(1,*) (Q(I),I=1,L)
    GO TO 35
34  DO 36 I=1,L
36  READ(1,*) (QP(I+(J-1)*L),J=1,NQP)
35  CONTINUE
    READ(1,*) UMIN,UMAX

C
C   SIMULASYON BASLANGICI
C
    KK=-L
    TT=0.0
    DO 110 NT=1,LENGTH
      WRITE(*,*) NT
      DO 2061 I=1,NU
2061  UDO(I)=UM(I+(DO-1)*NU)
      UDO(1)=UM(1)
      UDO(2)=UM(2)
      UDO(3)=UM(3)

C
C   ROBOT DINAMIGINDEN CIKIS BUYUKLUGU (Y(T))' NIN HESAPLANMASI
C
    DO 2071 J=1,NSTEP
2071  CALL RUNGE(TT,DT,NEQ,XX,UDO)

C
C   STOKASTIK BOZUCULARIN ILAVE EDILMESI
C
    SW1=GAUSS(AMEAN,STD)
    SW2=GAUSS(AMEAN,STD)
    SW3=GAUSS(AMEAN,STD)

C
    IF ( ) GO TO

```

```

Y(1)=XX(4)+SW1
Y(2)=XX(5)+SW2
Y(3)=XX(6)+SW3
C
YY(1)=XX(1)
YY(2)=XX(2)
YY(3)=XX(3)
GO TO 1556
C
C KARTZYEN UZYDA KONTROL DURUMUNDA X,Y,Z KONUMLARI
C
1555 A1Y=L1+L2*COS(XX(2))+L3*COS(XX(2)+XX(3))
YY(1)=A1Y*COS(XX(1))
YY(2)=A1Y*SIN(XX(1))
YY(3)=L2*SIN(XX(2))+L3*SIN(XX(2)+XX(3))
C
C X,Y,Z HIZLARI
C
B1Y=-L2*SIN(XX(2))*XX(5)-L3*SIN(XX(2)+XX(3))*XX(5)+XX(6))
Y(1)=-A1Y*SIN(XX(1))*XX(4)+COS(XX(1))*B1Y+SW1
Y(2)=A1Y*COS(XX(1))*XX(4)+SIN(XX(1))*B1Y+SW2
Y(3)=L2*COS(XX(2))*XX(5)+L3*COS(XX(2)+XX(3))*XX(5)+XX(6))+SW3
C
C HIZ HATALARININ KARELERININ HESAPLANMASI
C
1556 E1=E1+(YRV(NT,1)-Y(1))**2
E2=E2+(YRV(NT,2)-Y(2))**2
E3=E3+(YRV(NT,3)-Y(3))**2
C
C GEREKLI CIKIS BILGILERININ DOSYALARA AKTARILMASI
C
WRITE(27,*) Y(1)
WRITE(28,*) Y(2)
WRITE(29,*) Y(3)
WRITE(30,*) YRV(NT,1)
WRITE(31,*) YRV(NT,2)
WRITE(32,*) YRV(NT,3)
WRITE(33,*) YY(1)
WRITE(34,*) YY(2)
WRITE(35,*) YY(3)
WRITE(36,*) YRV(NT,4)
WRITE(37,*) YRV(NT,5)
WRITE(38,*) YRV(NT,6)
WRITE(39,*) E1
WRITE(40,*) E2
WRITE(41,*) E3
C
C KONTROL SINYALININ HESAPLANMASI
C
IF(ICON.EQ.1) CALL MSTR1(Y,U,L)
IF(ICON.EQ.2) CALL PRE(Y,U,L,N2,ALAMDA,NUF,NT,YRV,NSAY)
IF (ICON.EQ.3) CALL MSTC1(Y,U,L)
C
C CIKISLARIN TOPLU OLARAK BIR DOSYAYA YAZDIRILMASI
C
DO 101 I=1,L
101 WRITE(8,78) EN(I),Y(I),YRV(NT,I),UM(I)
78 FORMAT(5X,5F8.3)
WRITE(9,68) NT
68 FORMAT (1X,I5)
DO 102 I=1,L
102 WRITE(9,69) (P(I+(J-1)*L),J=1,NZ)
69 F

```

```

110 CONTINUE
C
  WRITE(7,527)
527 FORMAT(' FINAL PARAMETERS :')
  DO 111 I=1,L
111 WRITE(7,45) (P(I+(J-1)*L),J=1,NZ)
45 FORMAT (10F8.4)
  STOP
  END
C
C   ONGORULU KONTROL ALGORITMASININ DIZAYN EDILMESI

  SUBROUTINE PRE(Y,U,L,N2,ALANDA,NUF,NT,YRV,NSAY)
    DIMENSION Y(1),U(1)
    DIMENSION Z(40),SS(40),E(40),V(40),YRV(510,6)
    DIMENSION X(6),XM(6),DUM(40),DU(40)
    INTEGER D,D2
    COMMON/B1/YR(1),Q(4),W,P(40),UM(40),YM(40),R(200),S0,WMIN
    COMMON/B2/D,M,N,NZ
    COMMON/B3/UMIN,UMAX
    COMMON/STR/ICON,IEST
    D2=D
    N1=N+D-1
    CALL FORM(UM,YM,Z,M,N,D,D2,L)
C
C   PARAMETRE KESTIRME ALGORITMASININ SECILMESI
C
  IF (IEST.EQ.1) CALL RLS(Y,Z,W,R,P,NZ,L)
  IF (IEST.EQ.2) CALL SQR(Y,Z,W,WMIN,R,S0,P,NZ,L)
  IF (IEST.EQ.6) CALL RSGR(Y,Z,W,R,E,P,NZ,L)
C
C   GENELLESTIRILMIS ONGORULU KONTROL ALGORITMASINDA U KONTROLUNUN
C   HESAPLANMASI
C
  CALL PRECON(U,Y,UM,YM,YRV,P,N2,ALANDA,NUF,N,M,D,NT,L)
C
C   KONTROL VE CIKIS BUYUKLUKLERININ HAFIZADA SAKLANMASI
C
  CALL SHIFT(YM,Y,N+1,L)
  CALL SHIFT(UM,U,M+D,L)
  CALL SHIFT(DUM,DU,M+D,L)
C
C   KONTROL BUYUKLUGUNUN SATURASYONU
C
  CALL SAT(UM,L,UMIN,UMAX)
  RETURN
  END
C
  SUBROUTINE PRECON(U,Y,UM,YM,YRV,P,N2,ALANDA,NUF,N,M,D,NT,NSAY,L)
    DIMENSION U(1),UM(1),YM(1),YR(40),Y(1),P(1),YRV(510,6)
    DIMENSION B0(9),B1(9),A1(9),AIB(9),F0(9),F1(9),E(9),G(9)
    *,F0F1(9),G1(9),B0E(9),FF0(20,9),FF1(20,9),GG1(20,9),GG(20,9)
    *,AG(900),GA(900),AGG(900),AL(90),GIL(900),GII(90),F(900),YC(6)
    *,GF(30),AA(40),DU(30),DUM(10),FF(30),LL(100),MM(100)
    INTEGER D
    J=-L
    DO 14 I=1,N2
    DO 15 K=1,L
15 YR(J+L+K)=YRV(NT+I,K)
    J=J+L
14 CONTINUE
    IY=N2*L
    IC=L*L

```

```

      DO 1 I=1,IC
1    B0(I)=P(I)
      IF(M.EQ.0) GO TO 32
      DO 2 I=IC+1,2*IC
2    B1(I-IC)=P(I)
      DO 3 I=(2*IC)+1,3*IC
3    A1(I-2*IC)=P(I)
      GO TO 556
32 DO 555 I=IC+1,2*IC
555 A1(I-IC)=P(I)
556 DO 8 I=1,IC
      8 AIB(I)=0.
      DO 9 I=1,IC,L+1
9    AIB(I)=1.
C
C    E,F,G PARAMETRELERI ICIN BASLANGIC DEGERLERININ DUZENLENMESI
C
      DO 100 J=1,IC
      F0(J)=AIB(J)
      F1(J)=0.
      E(J)=AIB(J)
100 GG(0,J)=B0(J)
C
C    E,F,G PARAMETRELERININ HESAPLANMASI
C
      DO 10 II=1,N2
      CALL GMADD(F0,F1,FOF1,L,L)
      CALL GMPRD(A1,F0,F1,L,L,L)
      CALL GMIN(FOF1,F1,F0,L,L)
      CALL GMPRD(B1,E,G1,L,L,L)
      DO 13 J=1,L*L
13  E(J)=F0(J)
      CALL GMPRD(B0,E,B0E,L,L,L)
      CALL GMADD(B0E,G1,G,L,L)
      DO 11 J=1,L*L
      FFO(II,J)=F0(J)
      FF1(II,J)=F1(J)
      GGI(II,J)=G1(J)
11  GG(II,J)=G(J)
10  CONTINUE
C
      IF(N2.EQ.1) GO TO 37
      GO TO 35
37 DO 31 I=1,L*L
31  AG(I)=GG(0,I)
      GO TO 36
35  IJ=0
      K=0
      J=0
      DO 33 LJ=1,NUF
      DO 16 KK=1,L
      IL=0
      DO 18 JK=J,N2-LJ
      DO 17 I=IL+1,IL+L
17  AG(I+K)=GG(JK,I-IL+IJ)
      IL=IL+L
18  CONTINUE
      IJ=IJ+L
      K=K+N2*L
16  CONTINUE
      J=J-1
      IJ=0
33  CONTINUE
36  M1=N2*L

```

```

      NI=L*NUF
      CALL GMTRA(AG,GA,MI,NI)
      CALL GMPRD(GA,AG,AGG,NI,MI,NI)
      DO 220 I=1,NI*NI
220  AL(I)=0.
      DO 110 I=1,NI*NI,NI+1
110  AL(I)=ALAMDA
C
      CALL GMADD(AGG,AL,GIL,NI,NI)
      CALL MINV(GIL,NI,DD,LL,MM)
      CALL GMPRD(GIL,GA,AGG,NI,NI,MI)
C
      K=0
      IL=0
      DO 19 IK=1,L
      DO 20 J=1,N2
      DO 21 I=IL+1,IL+L
21  GII(I)=GGI(J,I-IL+K*L)
      IL=IL+L
20  CONTINUE
      K=K+1
19  CONTINUE
      DO 40 I=1,L
40  DUM(I)=UM((D-1)*L+I)-UM((D-1)*L+I+L)
      CALL GMPRD(GII,DUM,GIL,MI,L,1)
      K=0
      IL=0
      DO 22 IK=1,L
      DO 24 J=1,N2
      DO 25 I=IL+1,IL+L
25  F(I)=FFO(J,I-IL+K*L)
      IL=IL+L
24  CONTINUE
      K=K+1
22  CONTINUE
      K=0
      DO 26 IK=1,L
      DO 27 J=1,N2
      DO 28 I=IL+1,IL+L
28  F(I)=FFI(J,I-IL+K*L)
      IL=IL+L
27  CONTINUE
      K=K+1
26  CONTINUE
C
      DO 210 I=1,L
      YC(I)=Y(I)
210  YC(I+L)=YM(I)
C
      CALL GMPRD(F,YC,FF,MI,2*L,1)
      CALL GMADD(GIL,FF,GF,MI,1)
      CALL GMIN(YR,GF,AA,MI,1)
      CALL GMPRD(AGG,AA,DU,NI,MI,1)
      DO 38 I=1,L
38  U(I)=DU(I)+UM(I)
      IF(NUF.EQ.1) GO TO 29
      DO 39 I=L+1,L*NUF
39  U(I)=DU(I)+U(I-L)
29  DO 30 I=L*NUF+1,L*N2
30  U(I)=U(I-L)
      RETURN
      END

```

\$debug

C

C RUNGE-KUTTA YONTEMINDE GORE DIF. DENKLEM COZUMU

C

```

SUBROUTINE RUNGE(TT,DT,NEQ,XX,UDO)
  DIMENSION YI(6),YJ(6),YK(6),YL(6),UU(6),XX(6),F(6),UDO(3)

```

C

```

  DO 10 IM=1,NEQ
10  UU(IM)=XX(IM)
    CALL SYS(XX,F,NEQ,TT,UDO)
    DO 20 IM=1,NEQ
      YI(IM)=F(IM)*DT
20  XX(IM)=UU(IM)+YI(IM)/2.0
    TT=TT+DT/2.0
    CALL SYS(XX,F,NEQ,TT,UDO)
    DO 30 IM=1,NEQ
      YJ(IM)=F(IM)*DT
30  XX(IM)=UU(IM)+YJ(IM)/2.0
    CALL SYS(XX,F,NEQ,TT,UDO)
    DO 40 IM=1,NEQ
      YK(IM)=F(IM)*DT
40  XX(IM)=UU(IM)+YK(IM)
    TT=TT+DT/2.0
    CALL SYS(XX,F,NEQ,TT,UDO)
    DO 50 IM=1,NEQ
      YL(IM)=F(IM)*DT
50  XX(IM)=UU(IM)+(YI(IM)+2.0*YJ(IM)+2.0*YK(IM)+YL(IM))/6.0
    RETURN
  END

```

C

C ROBOT DINAMIGININ HESAPLANMASI

C

```

SUBROUTINE SYS(XX,F,NEQ,TT,TD)
  DIMENSION XX(6),F(6),TD(3)

```

C

C NONLINEER DIF.DENKLEMIN KATSAYILARININ HESAPLANMASI

C

```

  REAL L1,L2,L3
  DIMENSION T(3),AM(3),TD(3),HJ(9),HJTR(9),HJTU(9),
  *AI1(16),AJ(16),AIX(6),AIYY(6),AIZZ(6),AIXY(6),AIXZ(6),AIYZ(16),
  *TDD(3),Q(16),A10(16),A21(16),A32(16),QN2(16),
  *A20(16),A30(16),A31(16),
  *A22(16),A33(16),QN(16),QN1(16),U11(16),U21(16),UG(16),U22(16),
  *U31(16),U32(16),U33(16),H(3),D(9),GR(3),Q1(16),
  *Q2(16),Q3(16),AJ1(16),AJ2(16),AJ3(16),U11T(16),
  *U21T(16),U31T(16),U22T(16),U32T(16),U33T(16),AU11(16),
  *AU21(16),AU31(16),AU22(16),AU32(16),AU33(16),G1(1),G2(1),G3(1),
  *U111(16),U211(16),U311(16),U212(16),U312(16),U313(16),
  *U222(16),U322(16),U323(16),U333(16),AK(9)
  DIMENSION LL(16),MM(16),HJTRU(9),HJTRUT(3),
  *HMJ(3),HJTRG(3),HMG(3),HG(3),HH(3),S(3),HR(3),HA(3),HJR(9),
  *HJM(9),HD(9),TA(3),TD2(3),HGP(3)

```

C

```

  COMMON/AA/NS,IN(3),ALFA(3),AI(3),AD(3)
  COMMON/MM/AM1(1),AM2(1),AM3(1),GG1(4),GG2(4),GG3(4)
  COMMON/BB/AKI(3,9),S1(4),S2(4),S3(4)
  COMMON/CC/SX(3),SY(3),SZ(3)
  COMMON/AL/L1,L2,L3
  COMMON/V/V1,V2,V3,V4,V5,V6,PL,GP(3),NT

```

C

```

  TI=3.1416/180.

```

C

```

  T(1)=XX(1)

```

```

TD(1)=XX(4)
T(2)=XX(2)
TD(2)=XX(5)
T(3)=XX(3)
TD(3)=XX(6)

```

C

```

DO 10 I=1,NS
GO TO (661,662,663),I
661 AM(I)=AM1(I)
GO TO 667
662 AM(I)=AM2(I)
GO TO 667
663 AM(I)=AM3(I)

```

C

C DONUSUM (TRANSFORMASYON) MATRISLERININ HESAPLANMASI

C

```

667 AII1(1)=COS(T(I))
AII1(2)=SIN(T(I))
AII1(3)=0.
AII1(4)=0.
AII1(5)=-COS(TI*ALFA(I))*SIN(T(I))
AII1(6)=COS(TI*ALFA(I))*COS(T(I))
AII1(7)=SIN(TI*ALFA(I))
AII1(8)=0.
AII1(9)=SIN(TI*ALFA(I))*SIN(T(I))
AII1(10)=-SIN(TI*ALFA(I))*COS(T(I))
AII1(11)=COS(TI*ALFA(I))
AII1(12)=0.
AII1(13)=AI(I)*COS(T(I))
AII1(14)=AI(I)*SIN(T(I))
AII1(15)=AD(I)
AII1(16)=1.

```

C

C ATALET MATRISININ HESAPLANMASI

C

```

DO 1221 J=1,9
1221 AK(J)=AK1(I,J)
AJ(1)=AK(1)
AJ(2)=AK(2)
AJ(3)=AK(3)
AJ(4)=AM(I)*SX(I)
AJ(5)=AK(4)
AJ(6)=AK(5)
AJ(7)=AK(6)
AJ(8)=AM(I)*SY(I)
AJ(9)=AK(7)
AJ(10)=AK(8)
AJ(11)=AK(9)
AJ(12)=AM(I)*SZ(I)
AJ(13)=AM(I)*SX(I)
AJ(14)=AM(I)*SY(I)
AJ(15)=AM(I)*SZ(I)
AJ(16)=AM(I)

```

C

```

GO TO (221,222,223),I
221 DO 231 K=1,16
231 AJ1(K)=AJ(K)
GO TO 227
222 DO 232 K=1,16
232 AJ2(K)=AJ(K)
GO TO 227
223 DO 233 K=1,16
233 AJ3(K)=AJ(K)

```

```

227 DO 20 K=1,16
20 Q(K)=0.
   IF(IN(I).EQ.1) GO TO 25
   Q(15)=1.
   GO TO 30
25 Q(2)=1.
   Q(5)=-1
30 GO TO (111,112,113),I
111 DO 121 K=1,16
121 Q1(K)=Q(K)
   GO TO 120
112 DO 122 K=1,16
122 Q2(K)=Q(K)
   GO TO 120
113 DO 123 K=1,16
123 Q3(K)=Q(K)
120 GO TO (1,2,3),I
   1 DO 35 K=1,16
35 A10(K)=A111(K)
   GO TO 10
   2 DO 40 K=1,16
40 A21(K)=A111(K)
   GO TO 10
   3 DO 45 K=1,16
45 A32(K)=A111(K)
10 CONTINUE

```

C
C
C

JACOBIAN MATRISININ TESKILI

```

AY=L1+L2*COS(XX(2))+L3*COS(XX(2)+XX(3))
BY=L2*SIN(XX(2))+L3*SIN(XX(2)+XX(3))
HJ(1)=-AY*SIN(XX(1))
HJ(2)=AY*COS(XX(1))
HJ(3)=0.
HJ(4)=-BY*COS(XX(1))
HJ(5)=BY*SIN(XX(1))
HJ(6)=L2*COS(XX(2))+L3*COS(XX(2)+XX(3))
HJ(7)=-L3*SIN(XX(2)+XX(3))*COS(XX(1))
HJ(8)=-L3*SIN(XX(2)+XX(3))*SIN(XX(1))
HJ(9)=L3*COS(XX(2)+XX(3))

```

C
C
C

JACOBIAN MATRISININ TRANSPOZESI

```

HJTR(1)=HJ(1)
HJTR(2)=HJ(4)
HJTR(3)=HJ(7)
HJTR(4)=HJ(2)
HJTR(5)=HJ(5)
HJTR(6)=HJ(8)
HJTR(7)=HJ(3)
HJTR(8)=HJ(6)
HJTR(9)=HJ(9)

```

C
C
C

JACOBIAN MATRISININ TUREVI

```

CY=L2*SIN(XX(2))*XX(5)+L3*SIN(XX(2)+XX(3))*(XX(5)+XX(6))
DY=-L2*COS(XX(2))*XX(5)-L3*COS(XX(2)+XX(3))*(XX(5)+XX(6))
HJTU(1)=CY*SIN(XX(1))-AY*COS(XX(1))*XX(4)
HJTU(2)=-CY*COS(XX(1))-AY*SIN(XX(1))*XX(4)
HJTU(3)=0.
HJTU(4)=DY*COS(XX(1))+BY*SIN(XX(1))*XX(4)
HJTU(5)=DY*SIN(XX(1))-BY*COS(XX(1))*XX(4)
HJTU(6)=-L2*SIN(XX(2))*XX(5)-L3*SIN(XX(2)+XX(3))*(XX(5)+XX(6))

```

```

HJTU(7)=-L3*COS(XX(2)+XX(3))*{XX(5)+XX(6))*COS(XX(1))+L3*
*SIN(XX(2)+XX(3))*SIN(XX(1))*XX(4)
HJTU(8)=-L3*COS(XX(2)+XX(3))*{XX(5)+XX(6))*SIN(XX(1))-L3*
*SIN(XX(2)+XX(3))*COS(XX(1))*XX(4)
HJTU(9)=-L3*SIN(XX(2)+XX(3))*{XX(5)+XX(6))

```

C
C
C

SURTUNME VEKTORUNUN HESAPLANMASI

```

CALL SGN(TD,SA,SB,SC)
S(1)=V1*TD(1)+V2*SA
S(2)=V3*TD(2)+V4*SB
S(3)=V5*TD(3)+V6*SC

```

C

```

CALL GMPRD(A10,A21,A20,4,4,4)
CALL GMPRD(A20,A32,A30,4,4,4)
CALL GMPRD(A21,A32,A31,4,4,4)

```

C

```

CALL GMPRD(Q1,A10,U11,4,4,4)
CALL GMPRD(Q1,A20,U21,4,4,4)
CALL GMPRD(Q1,A30,U31,4,4,4)
CALL GMPRD(A10,Q2,UG,4,4,4)
CALL GMPRD(UG,A21,U22,4,4,4)
CALL GMPRD(U22,A32,U32,4,4,4)
CALL GMPRD(A20,Q3,UG,4,4,4)
CALL GMPRD(UG,A32,U33,4,4,4)

```

C

D MATRISININ ELEMENLARININ HESABI

C

```

CALL GMPRA(U11,U11T,4,4)
CALL GMPRA(U21,U21T,4,4)
CALL GMPRA(U31,U31T,4,4)
CALL GMPRD(AJ1,U11T,AU11,4,4,4)
CALL GMPRD(AJ2,U21T,AU21,4,4,4)
CALL GMPRD(AJ3,U31T,AU31,4,4,4)
CALL GMPRA(U22,U22T,4,4)
CALL GMPRA(U32,U32T,4,4)
CALL GMPRD(AJ2,U22T,AU22,4,4,4)
CALL GMPRD(AJ3,U32T,AU32,4,4,4)
CALL GMPRA(U33,U33T,4,4)
CALL GMPRD(AJ3,U33T,AU33,4,4,4)
CALL ALT(U11,AU11,F1)
CALL ALT(U21,AU21,F2)
CALL ALT(U31,AU31,F3)
D(1)=F1+F2+F3

```

C

```

CALL ALT(U22,AU21,F1)
CALL ALT(U32,AU31,F2)
D(2)=F1+F2
D(4)=D(2)

```

C

```

CALL ALT(U33,AU31,F1)
D(3)=F1
D(7)=D(3)

```

C

```

CALL ALT(U22,AU22,F1)
CALL ALT(U32,AU32,F2)
D(5)=F1+F2

```

C

```

CALL ALT(U33,AU32,F1)
D(6)=F1
D(8)=D(6)

```

C

C

CALL ALT(U33,AU33,F1)
D(9)=F1

C
C
C

6 VEKTORUNUN ELEMANLARININ HESAPLANMASI

CALL GRAV(AM1,G61,U11,S1,G1)
CALL GRAV(AM2,G62,U21,S2,G2)
CALL GRAV(AM3,G63,U31,S3,G3)
GR(1)=G1(1)+G2(1)+G3(1)

C

CALL GRAV(AM2,G62,U22,S2,G1)
CALL GRAV(AM3,G63,U32,S3,G2)
GR(2)=G1(1)+G2(1)

C

CALL GRAV(AM3,G63,U33,S3,G1)
GR(3)=G1(1)

C
C
C

H VEKTORUNUN ELEMANLARININ HESAPLANMASI

CALL GMPRD(Q1,Q1,QN,4,4,4)
CALL GMPRD(QN,A10,U111,4,4,4)
CALL GMPRD(QN,A20,U211,4,4,4)
CALL GMPRD(QN,A30,U311,4,4,4)
CALL GMPRD(Q1,A10,QN,4,4,4)
CALL GMPRD(QN,Q2,QN1,4,4,4)
CALL GMPRD(QN1,A21,U212,4,4,4)
CALL GMPRD(QN1,A31,U312,4,4,4)
CALL GMPRD(Q1,A20,QN,4,4,4)
CALL GMPRD(QN,Q3,QN1,4,4,4)
CALL GMPRD(QN1,A32,U313,4,4,4)
CALL GMPRD(A10,Q2,QN2,4,4,4)
CALL GMPRD(QN2,Q2,QN,4,4,4)
CALL GMPRD(QN,A21,U222,4,4,4)
CALL GMPRD(QN,A31,U322,4,4,4)
CALL GMPRD(QN2,A21,QN,4,4,4)
CALL GMPRD(QN,Q3,QN1,4,4,4)
CALL GMPRD(QN1,A32,U323,4,4,4)
CALL GMPRD(A20,Q3,QN,4,4,4)
CALL GMPRD(QN,Q3,QN2,4,4,4)
CALL GMPRD(QN2,A32,U333,4,4,4)
CALL ALT(U111,AU11,H1)
CALL ALT(U211,AU21,H2)
CALL ALT(U311,AU31,H3)
H111=H1+H2+H3
CALL ALT(U212,AU21,H1)
CALL ALT(U312,AU31,H2)
H112=H1+H2
CALL ALT(U313,AU31,H113)

C

H121=H112
CALL ALT(U222,AU21,H1)
CALL ALT(U322,AU31,H2)
H122=H1+H2
CALL ALT(U323,AU31,H123)

C

H131=H113
H132=H123
CALL ALT(U333,AU31,H133)

C

CALL ALT(U211,AU22,H1)
CALL ALT(U311,AU32,H2)
H211=H1+H2
CALL ALT(U212,AU22,H1)

```

CALL ALT(U312,AU32,H2)
H212=H1+H2
CALL ALT(U313,AU32,H213)
H221=H212
CALL ALT(U222,AU22,H1)
CALL ALT(U322,AU32,H2)
H222=H1+H2
CALL ALT(U323,AU32,H223)
H231=H213
H232=H223
CALL ALT(U333,AU32,H233)

```

C

```

CALL ALT(U311,AU33,H311)
CALL ALT(U312,AU33,H312)
CALL ALT(U313,AU33,H313)
H321=H312
CALL ALT(U322,AU33,H322)
CALL ALT(U323,AU33,H323)
H331=H313
H332=H323
CALL ALT(U333,AU33,H333)
H(1)=H111*(TD(1)*TD(1))+H112*TD(1)*TD(2)+H113*TD(1)*TD(3)
#+H121*TD(2)*TD(1)+H122*(TD(2)*TD(2))+H123*TD(2)*TD(3)+H131*TD(1)
#+TD(3)+H132*TD(3)*TD(2)+H133*(TD(3)*TD(3))
H(2)=H211*(TD(1)*TD(1))+H212*TD(1)*TD(2)+H213*TD(1)*TD(3)+H221*
TD(2)*TD(1)+H222*(TD(2)*TD(2))+H223*TD(2)*TD(3)+H231*TD(1)*TD(3)
#+H232*TD(3)*TD(2)+H233*(TD(3)*TD(3))
H(3)=H311*(TD(1)*TD(1))+H312*TD(1)*TD(2)+H313*TD(1)*TD(3)+H321*
TD(2)*TD(1)+H322*(TD(2)*TD(2))+H323*TD(2)*TD(3)+H331*TD(1)*TD(3)
#+H332*TD(3)*TD(2)+H333*(TD(3)*TD(3))
WRITE (2,*) 'H DEGERLERI',(H(I),I=1,3)

```

C

C

```

CALL GMPRD(HJTR,HJTU,HJTRU,3,3,3)
CALL GMPRD(HJTRU,TD,HJTRUT,3,3,1)

```

C

C

```

UC ELEMENIN TASIYACABI YUKUN BOZONUNE ALINMASI

```

C

```

IF(HT.GE.100) PL=0.
DO 1004 JJ=1,3
1004 HMJ(JJ)=PL*HJTRUT(JJ)

```

C

```

CALL GMPRD(HJTR,GP,HGP,3,3,1)
DO 1005 JJ=1,3
1005 HMG(JJ)=PL*HGP(JJ)
CALL GMADD(HMJ,HMG,HG,3,1)
CALL GMADD(HG,H,HH,3,1)
CALL GMADD(HH,GR,HR,3,1)
CALL GMADD(HR,S,HA,3,1)

```

C

```

CALL GMPRD(HJTR,HJ,HJR,3,3,3)
DO 1003 I=1,9
1003 HJM(I)=PL*HJR(I)
CALL GMADD(HJM,D,HD,3,3)
DO 1002 JJ=1,9
LL(JJ)=0.
1002 MM(JJ)=0.

```

```

CALL MINV(HD,NS,DD,LL,MM)
CALL GMIN(TD,HA,TA,3,1)
CALL GMPRD(HD,TA,TD2,3,3,1)

```

C

C

```

DURUM UZAYI FORMUNUN OLUSTURULMASI

```

C

```

F(1)=

```

```

F(4)=TD2(1)
F(2)=TD(2)
F(5)=TD2(2)
F(3)=TD(3)
F(6)=TD2(3)

```

C

```

RETURN
END

```

C

```

SUBROUTINE ALT (A,B,F)
DIMENSION A(16),B(16),E(16)
CALL GMPRD(A,B,E,4,4,4)
CALL TRACE(E,F)
RETURN
END

```

C

C

```

KULLANILAN YARDINCI ALT PROGRAMLAR

```

C

```

SUBROUTINE TRACE(D,F)
DIMENSION D(16)
F=D(1)+D(6)+D(11)+D(16)
RETURN
END

```

C

```

SUBROUTINE GRAV(A,B,C,D,E)
DIMENSION B(4),C(16),D(4),F(4),H(4),G(1),A(1),E(1)
CALL GMPRD (C,D,F,4,4,1)
CALL GMPRD (B,H,4,1)
CALL GMPRD (H,F,G,1,4,1)
E(1)=A(1)*G(1)
RETURN
END

```

C

```

SUBROUTINE SGN(TD,SA,SB,SC)
DIMENSION TD(3)
IF(TD(1).LT.0.) SA=-1.
IF(TD(1).EQ.0.) SA=0.
IF(TD(1).GT.0.) SA=1.

```

C

```

IF(TD(2).LT.0.) SB=-1.
IF(TD(2).EQ.0.) SB=0.
IF(TD(2).GT.0.) SB=1.

```

C

```

IF(TD(3).LT.0.) SC=-1.
IF(TD(3).EQ.0.) SC=0.
IF(TD(3).GT.0.) SC=1.
RETURN
END

```

\$DEBUG

C İKİ MATRİSİN TOPLANMASI

```

SUBROUTINE GMADD(A,B,R,N,M)
  DIMENSION A(1),B(1),R(1)
  NM=N*M
  DO 10 I=1,NM
10 R(I)=A(I)+B(I)
  RETURN
END

```

C

C BİR MATRİSİN TRANSPOZESİNİN ALINMASI

C

```

SUBROUTINE GMTRA(A,R,N,M)
  DIMENSION A(1),R(1)
  IR=0
  DO 10 I=1,N
  IJ=I-N
  DO 10 J=1,M
  IJ=IJ+N
  IR=IR+1
10 R(IR)=A(IJ)
  RETURN
END

```

C

C İKİ MATRİSİN ÇIKARILMASI

C

```

SUBROUTINE GMIN(A,B,R,N,M)
  DIMENSION A(1),B(1),R(1)
  NM=N*M
  DO 10 I=1,NM
10 R(I)=A(I)-B(I)
  RETURN
END

```

C

C GENEL İKİ MATRİS VEYA VEKTÖRÜN ÇARPILMASI

C

```

SUBROUTINE GMPRD(A,B,R,N,M,L)
  DIMENSION A(1),B(1),R(1)
  IR=0
  IK=-M
  DO 10 K=1,L
  IK=IK+M
  DO 10 J=1,N
  IR=IR+1
  JI=J-N
  IB=IK
  R(IR)=0
  DO 10 I=1,M
  JI=JI+N
  IB=IB+1
10 R(IR)=R(IR)+A(JI)*B(IB)
  RETURN
END

```

C

C BİLGİLERİN KAYDIRILMASI

C

```

SUBROUTINE SHIFT(YM,Y,N,L)
  DIMENSION YM(1),Y(1)
  DO 10 I=L*N,L+1,-1
10 YM(I)=YM(I-L)
  DO 20 I=1,L
20 Y(I)=YM(I)

```

RETURN
END

C
C
C

BIR KARE MATRIS ILE BIR VEKTORUN CARPILMASI

```

SUBROUTINE GMPDV(A,X,Y,N)
  DIMENSION A(1),X(1),Y(1)
  DO 10 I=1,N
10 Y(I)=0.0
  IJ=0
  DO 20 J=1,N
  DO 20 I=1,N
  IJ=IJ+1
20 Y(I)=Y(I)+A(IJ)*X(J)
  RETURN
  END

```

C
C
C

MATRIS TERSININ ALINMASI

```

SUBROUTINE MINV(A,N,D,L,M)
  DIMENSION A(1),L(1),M(1)
  D=1.0
  NK=-N
  DO 80 K=1,N
  NK=NK+N
  L(K)=K
  M(K)=K
  KK=NK+K
  BIGA=A(KK)
  DO 20 J=K,N
  IZ=N*(J-1)
  DO 20 I=K,N
  IJ=IZ+I
10 IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
  L(K)=I
  M(K)=J
20 CONTINUE
  J=L(K)
  IF(J-K) 35,35,25
25 KI=K-N
  DO 30 I=1,N
  KI=KI+N
  HOLD=-A(KI)
  JI=KI-K+J
  A(KI)=A(JI)
30 A(JI)=HOLD
35 I=M(K)
  IF(I-K) 45,45,38
38 JP=N*(I-1)
  DO 40 J=1,N
  JK=NK+J
  JI=JP+J
  HOLD=-A(JK)
  A(JK)=A(JI)
40 A(JI)=HOLD
45 IF(BIGA) 48,46,48
46 D=0.0
  RETURN
48 DO 55 I=1,N
  IF(I-K) 50,55,50
50 IK=NK+I

```

```

55 CONTINUE
  DO 65 I=1,N
    IK=NK+I
    HOLD=A(IK)
    IJ=I-N
    DO 65 J=1,N
      IJ=IJ+N
      IF (I-K) 60,65,60
60  IF (J-K) 62,65,62
62  KJ=IJ-I+K
    A(IJ)=HOLD+A(KJ)+A(IJ)
65 CONTINUE
    KJ=K-N
    DO 75 J=1,N
      KJ=KJ+N
      IF (J-K) 70,75,70
70  A(KJ)=A(KJ)/BIGA
75 CONTINUE
    D=D*BIGA
    A(KK)=1.0/BIGA
80 CONTINUE
    K=N
100 K=(K-1)
    IF (K) 150,150,105
105 I=L(K)
    IF (I-K) 120,120,108
108 JQ=N*(K-1)
    JR=N*(I-1)
    DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
110 A(JI)=HOLD
120 J=N(K)
    IF (J-K) 100,100,125
125 KI=K-N
    DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
130 A(JI)=HOLD
    GO TO 100
150 RETURN
  END

```

VERILEN BİR SINYALIN SATURASYONU

```

SUBROUTINE SAT(U,L,UMIN,UMAX)
  DIMENSION U(1)
  DO 10 I=1,L
    S=U(I)
    IF (S.LT.UMIN) U(I)=UMIN
    IF (S.GT.UMAX) U(I)=UMAX
10 CONTINUE
  RETURN
  END

```

MODELİN UYGUN FORMDA DÜZENLENMESİ

```

SUBROUTINE FORM(UM,YM,Z,M,N,D1,D2,L)
  DIMENSION UM(1),YM(1),Z(1)

```

```

      INTEGER D1,D2
      K=0
      DO 10 J=0,M
      DO 10 I=1,L
      K=K+1
10  Z(K)=UM(L*(D1+J-1)+I)
      DO 20 J=0,N-1
      DO 20 I=1,L
      K=K+1
20  Z(K)=-YM(L*(D2+J-1)+I)
C    Z(2)=YM(2)
C    Z(3)=1.
      RETURN
      END

C
C    PARAMETRE KESTIRME ALGORITMASI (KARE KOK YAKLASIMI)
C
      SUBROUTINE SQR(Y,Z,W,SR,P,NZ,L)
      DIMENSION Y(1),Z(1),SR(1),P(1)
      DIMENSION G(10),SS(10)
      IVFF=0
      GO TO 1
      IVFF=1
1    CALL REFIL(Z,W,SIG2,G,SR,NZ)
C
C    UPDATE THE MODEL PARAMETER VECTOR P(T) BY
C    (1) COMPUTE GN=G/SIG2
C    (2) COMPUTE SS=Y(T)-P(T-1)*Z (MATRIX TIMES VECTOR)
C    (3) COMPUTE P(T)=P(T-1)+SS*GN (VECTOR OUTER PRODUCT)
C
      CALL GMPRD(P,Z,SS,L,NZ,1)
      DO 50 I=1,L
50  SS(I)=Y(I)-SS(I)
      K=0
      DO 60 J=1,NZ
      DO 60 I=1,L
      K=K+1
60  P(K)=P(K)+SS(I)*G(J)/SIG2
      IF (IVFF.EQ.0) GO TO 99
      S=1.
      DO 20 I=1,NZ
20  S=S-Z(I)*G(I)/SIG2
      S2=0.0
      DO 80 I=1,L
80  S2=S2+SS(I)*SS(I)
      W=1.0-S*S2/S0
      IF (W.LT.WMIN) W=WMIN
99  RETURN
      END

C
      SUBROUTINE REFIL(Z,W,SIG2,G,SR,NZ)
      DIMENSION Z(1),G(1),SR(1)
      W1=SQRT(W)
      SIG=W1
      SIG2=W
      DO 30 J=1,NZ
      F=0.0
      DO 10 I=1,J
      IJ=NZ*(J-1)+I
10  F=F+SR(IJ)*Z(I)
      A=SIG/W1
      B=F/SIG2
      SIG2=      :F

```

```

SIG=SQRT(SIG2)
C=A/SIG
JJ=NZ*(J-1)+J
G(J)=SR(JJ)*F
SR(JJ)=C*SR(JJ)
DO 20 I=1,J-1
IJ=NZ*(J-1)+I
D=SR(IJ)
SR(IJ)=C*(D-B*G(I))
20 G(I)=D*F+G(I)
30 CONTINUE
RETURN
END

```

C
C
C

SELF-TUNING KONTROL KANUNUNUN HESAPLANMASI

```

SUBROUTINE ZSSCON (YR,P,Z,Q,U,NZ,L,XM,YRDM,C)
DIMENSION YR(1),Z(1),P(1),U(1),Q(1)
DIMENSION SS(10),II(10),JJ(10)
DIMENSION XM(4),YRDM(6),C(4)
COMMON/TTS/TS

```

C

```

CALL GMPRD(P,Z,SS,L,NZ,1)
DO 110 I=1,L
110 SS(I)=(YR(I)+C(I)*(YRDM(I)-XM(I))/TS)-SS(I)
CALL GTPRD(P,SS,Z,L,L,1)
DO 120 I=1,L
120 Z(I)=Z(I)+Q(I)*U(I)
CALL GTPRD(P,P,SS,L,L,L)
J=1
DO 130 I=1,L
SS(J)=SS(J)+Q(I)
J=J+L+1
130 CONTINUE
CALL MINV(SS,L,S,II,JJ)
CALL GMPDV(SS,Z,U,L)
Q(1)=P(1)*P(1)
RETURN
END

```

C

```

SUBROUTINE GTPRD(A,B,R,N,M,L)
DIMENSION A(1),B(1),R(1)
IR=0
IK=-N
DO 10 K=1,L
IJ=0
IK=IK+N
DO 10 J=1,M
IB=IK
IR=IR+1
R(IR)=0
DO 10 I=1,N
IJ=IJ+1
IB=IB+1
10 R(IR)=R(IR)+A(IJ)+B(IB)
RETURN
END

```

C
C
C

PARAMETRE KESTIRME ALGORITMASI (ARDISIK KARE KOK YAKLASIMI)

```

SUBROUTINE RSQR(Y,Z,W,SR,E,P,NZ,L)
DIMENSION Y(1),Z(1),SR(1),E(1),P(1)
DIM YY(10),ZZ(40)

```

```

      WR=SQRT(W)
      DO 1 I=1,NZ
1     ZZ(I)=Z(I)
      DO 2 I=1,L
2     YY(I)=Y(I)
      DO 3 I=1,NZ
      DO 3 J=1,NZ
      IJ=(J-1)*NZ+I
3     SR(IJ)=WR*SR(IJ)
      DO 4 I=1,NZ
      DO 4 J=1,L
      IJ=(J-1)*NZ+I
4     E(IJ)=WR*E(IJ)
      DO 40 I=1,NZ
      II=(I-1)*NZ+I
      S=SR(II)**2+ZZ(I)**2
      IF (S.LE.0) GO TO 40
      S=SQRT(S)
C
      SS=SR(II)
      IF (SS.EQ.S) GO TO 40
      U1=SQRT(0.5*(1.-SS/S))
      U2=-ZZ(I)/(2.*U1*S)
      SR(II)=S
      ZZ(I)=0.0
      DO 10 J=I+1,NZ
      IJ=(J-1)*NZ+I
      SS=SR(IJ)
      S=2.*(U1*SS+U2*ZZ(J))
      SR(IJ)=SS-S*U1
10     ZZ(J)=ZZ(J)-S*U2
C
      DO 20 J=1,L
      IJ=(J-1)*NZ+I
      SS=E(IJ)
      S=2*(U1*SS+U2*YY(J))
      E(IJ)=SS-S*U1
20     YY(J)=YY(J)-S*U2
40     CONTINUE
      DO 60 J=1,L
      DO 60 I=NZ,1,-1
      II=(I-1)*NZ+I
      IJ=(J-1)*NZ+I
      JI=(I-1)*L+J
      IF (SR(II).EQ.0.) GO TO 60
      SS=E(IJ)
      DO 50 K=I+1,NZ
      JK=(K-1)*L+J
      IK=(K-1)*NZ+I
50     SS=SS-SR(IK)*P(JK)
      P(JI)=SS/SR(II)
60     CONTINUE
      RETURN
      END
C
C     RANDOM SINYAL URETIMI
C
      FUNCTION GAUSS(MEAN,STD)
      COMMON/SED/ISEED1,ISEED2
      REAL MEAN,STD
      S=-6.0
      DO 10 I=1,12
10

```

```

GAUSS=STD*S+MEAN
RETURN
END

```

C

```

FUNCTION RAN(DUMMY)
REAL SEED,X
DATA SEED/4.0/
X=(SEED+3.141593)**5
SEED=X-AINT(X)
RAN=SEED
RETURN
END

```

C

C

```

PARAMETRE KESTIRME ALGORITMASI (ARDISIK EN KUCUK KARELER YAKLASIMI)

```

C

```

SUBROUTINE RLS(Y,Z,W,R,P,NZ,L)
DIMENSION Y(1),Z(1),R(1),P(1)
DIMENSION RZ(150),SS(10)
CALL GMPDV(R,Z,RZ,NZ)
S=W
DO 30 I=1,NZ
30 S=S+Z(I)*RZ(I)
K=0
DO 40 I=1,NZ
DO 40 J=1,NZ
K=K+1
40 R(K)=(R(K)-RZ(I)*RZ(J))/S/W
CALL GMPDV(R,Z,RZ,NZ)
CALL GMPRD(P,Z,SS,L,NZ,1)
DO 50 I=1,L
50 SS(I)=Y(I)-SS(I)
K=0
DO 60 J=1,NZ
DO 60 I=1,L
K=K+1
60 P(K)=P(K)+SS(I)*RZ(J)
RETURN
END

```

Giris Dosyasi

6,0.0025,0.
 -0.78539,-0.78539,-0.78539,0.,0.,0.
 3,1,1,1
 90.,0.,90.
 0.1588,0.4445,0.
 0.,0.,0.
 13.1339,10.3320,6.4443
 0.,0.,-9.80621,0.
 0.,0.,-9.80621,0.
 0.,0.,-9.80621,0.
 -0.0493,0.,0.,1.
 -0.1618,0.,0.,1.
 0.,0.,0.2718,1.
 0.0056064,0.0089196,0.0132387
 0.003929,0.0478064,0.0454838
 0.0820644,0.0819353,0.0014
 0.1588,0.4445,0.3245
 1.,0.5,1.,0.5,1.,0.5,5.
 0.,9.81,0.
 3,1,0.000001,4,1
 400 3 2 2 1030
 1 6 1
 0.005,0.0
 0
 1
 2
 1 1 1 0 0 0
 1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.
 0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.
 0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.
 10000. 0.95 100000. 0.90
 0.005,1.,1.
 -225,225

ÖZGEÇMİŞ

Recep KAZAN, 1960 yılında Adapazarı'nda doğdu. İlk ve orta öğrenimini Adapazarı'nda tamamladı. Sakarya Devlet Mühendislik ve Mimarlık Akademisi Makina Mühendisliği bölümünden 1981 Yaz yarıyılında mezun oldu. Aynı yıl İ.T.Ü'de Yüksek Lisans eğitimine başladı. Bir yıl yabancı dil hazırlık sınıfında okudu. 1984 yılında İ.T.Ü Fen Bilimleri Enstitüsü Makina Ana Bilim Dalı Konstrüksiyon Programında Yüksek Lisans öğrenimini tamamladı ve İ.T.Ü Sakarya Mühendislik Fakültesi Makina Bölümünde Araştırma Görevlisi olarak çalışmaya başladı. 1984 Kış yarıyılında İ.T.Ü Fen Bilimleri Enstitüsü Makina Ana Bilim Dalı Konstrüksiyon programında doktora çalışmasına başladı. 1986 yılında doktora çalışmasına ara vererek Yedek Subay olarak askerlik hizmetini tamamladı. Halen aynı fakültede araştırma görevlisi olarak çalışmaktadır. Evli ve bir çocuk babasıdır.