

14371.

İSTANBUL TEKNİK ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

MANİPULATÖR KONTROLÜ İÇİN ÇOK DEĞİŞKENLİ
ADAPTİF PID REGÜLATÖR

YÜKSEK LİSANS TEZİ

Mak. Müh. A. Selçuk Tekdemir

Tezin Enstitüye Verildiği Tarih: 21 Ocak 1991

Tezin Savunulduğu Tarih : 13 Şubat 1991

Tez Danışmanı : Doç. Dr. Can ÖZSOY

Diğer Jüri Üyeleri : Prof. Dr. A. Talha DİNİBÜTÜN
Prof. Dr. Ahmet KUZUCU

ŞUBAT 1991

T. G.
Yükseköğretim Kurulu
Dokümantasyon Me

Beni bu konuya yönelten ve değerli yardımlarını
esirgemeyen sayın hocam Doç. Dr. Can ÖZSOY ' a ve Otomatik
Kontrol Birimi elemanlarına teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	IV
SUMMARY	V
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. ROBOTLAR HAKKINDA GENEL BİLGİ	3
2.1. Robotların Karakteristik Yönleri	3
2.2. Robot Sistemi	3
2.3. Robotların Sınıflandırılması	4
BÖLÜM 3. ROBOTLARIN DİNAMİK MODELİ	6
3.1. Lagrange-Euler Formülasyonu	6
3.2. Lagrange-Euler Denklemi İçin Bazı Tanımlar	7
3.3. Robot Kolunun Kinetik Enerjisi	11
3.4. Robot Kolunun Potansiyel Enerjisi ...	12
3.5. Robot Kolunun Hareket Denklemi	13
BÖLÜM 4. N SERBESTLİK DERECELİ ROBOT İÇİN ADAPTİF PID KONTROL SİSTEMİNİN YAPISI	17
4.1. Sabit Parametrelili Kontrolörün Tasarımı	17
4.2. Adaptif Kontrolör Tasarımı	21
4.3. Lyapunov Denklemine Çözümü	23
4.2. Lyapunov Fonksiyonu Ve Adaptasyon Kuralı	25
BÖLÜM 5. SİMÜLASYON ÇALIŞMALARI	33
5.1. Simülasyon Sonuçları	35
SONUÇLAR VE ÖNERİLER	57
KAYNAKLAR	58
EKLER	60
ÖZGEÇMİŞ	69

ÖZET

Bir robot kolunun dinamik davranışı o robot kolunun dinamik performansını belirler. Robotların istenilen dinamik performansı göstermesi uygulanan kontrol algoritmasına ve robotun dinamik modelinin karmaşıklığına bağlıdır. Robotların kontrolünde robotların dinamik modeli yaklaşıklıkla bilinse dahi zamanla değişen parametreler, hesaplanması güç bozucular ve ölçme hataları gibi robotun dinamik davranışını etkileyecek bozucuların gözönüne alınacağı bir kontrol yöntemi uygulanmalıdır.

Robot kolunun dinamik modeli eklemler değişkenleri ve hızları birbirleri cinsinden ifade edildiğinden çok karmaşık lineer olmayan ifadeler içermektedir. Bu nedenle uygulanacak kontrol yönteminde her ekleme ait kontrol çevriminin diğer eklemlere ait kontrol çevrimleri ile etkileşim içersinde çalışması gerekir.

Robot kontrolü için iki temel kategori vardır. Bunlardan birincisi geribeslemeli nonlineer kontrolör yardımı ile robot dinamiğindeki nonlineerliklerin iptal edilmesi esasına dayanan robot kontrol tekniklerini içerir [1], [2], [3]. İkinci kategori gelişmiş kontrol sistemleri adaptif kontrol teorisine dayanır [4], [5], [6], [7].

Bu çalışmada, ikinci kategoride yer alan merkezi kontrol esaslı adaptif PID kontrolörü 3 serbestlik dereceli bir manipulatöre uygulanarak performansı incelenmiştir. Simülasyon çalışmaları izlenen yörüngenin değişmesi, yükteki ani değişim, uç noktaya etkiyen bir kuvvetin olması ve ölçme hatalarının yapılması durumlarında dahi adaptif PID kontrolörün performansının ne derece iyi olduğunu göstermiştir.

SUMMARY

MULTIVARIABLE ADAPTIVE PID REGULATOR FOR MANIPULATOR CONTROL

Robot arm dynamics deals with the mathematical formulations of the equations of robot arm motion. The dynamic equations of motion of a manipulator are a set of mathematical equations describing the dynamic behavior of the manipulator. Such equations of motion are useful for computer simulation of the robot arm motion, the design of suitable control equations for a robot arm.

The actual dynamic model of a robot arm can be obtained from known physical laws such as the laws of newtonian mechanics and lagrangian mechanics. This leads to the development of the dynamic equations of motion for the various articulated joints of the manipulator in terms of specified geometric and inertial parameters of the links. Conventional approaches like the Lagrange-Euler and Newton-Euler formulations could then be applied systematically to develop the actual robot arm motion equations [8].

The purpose of manipulator control is to maintain the dynamic response of a computer-based manipulator in accordance with some prespecified system performance and desired goals. In general, the dynamic performance of a manipulator directly depends on the efficiency of the control algorithms and the dynamic model of the manipulator. The control problem consists of obtaining dynamic models of the physical robot arm system and then specifying corresponding control laws or strategies to achieve the desired system response and performance.

The basic difficulty in controlling a robot manipulator arises from the fact that the dynamic equations describing the manipulator motion are inherently nonlinear and highly coupled; because of dynamic coupling effects between the joints and varying effective inertias of the links. The complexity of the mathematical model of a manipulator makes the robot control task a difficult and challenging problem.

Present day robot manipulators utilize independent joint controllers which control the joint angles separately through simple position servo loops. This basic control system enables the manipulator to perform simple positioning tasks such as pick -and -place operations. However, it is severely limited in terms of precise tracking of fast trajectories and sustaining desirable dynamic performance for different payloads.

The control of robot manipulators has been a very active area of research and two major design categories have emerged. The first category, originated by the classic works of Paul [1] , Bejczy [2] and Markiewicz [3], covers robot control techniques based on global linearization of robot dynamics by means of a nonlinear feedback controller. In these techniques, the nonlinearities in the robot robot dynamics are cancelled out by the controller and hence the nonlinear feedback controller is of the same order of complexity as the robot dynamics itself. Also in these model-based control techniques the controller requires knowledge of the full dynamic model and accurate parameter values of the manipulator and the payload. These requirements can be difficult to meet in practice since, for instance, joint friction is difficult to model accurately and is dependent on the operating condition and various adjustments. When perfect cancellations of the robot nonlinearities is not achieved due to imperfect modelling or inaccurate parameter values, dynamic performans of the robot may be degraded and a complicated stability analysis is necessitated and this situation may even lead to instability of the closed-loop system.

The second category of advanced control systems covers methods based on adaptive control theory and was initiated by the pioneering work of Dubowsky and DesForges [4]. The task of the adaptive robot controller is to adjust its gains based on the response of the robot in such a way that the performance of the robot closely matches that of a reference model defined by the designer.

Adaptive control theory is applied on many practical control problems in which we come across:

1. Changes in the plant transfer function, either in its order or in the values of some parameters due to variations in the environment.
2. Stochastic disturbances.

3. Changes in the nature of inputs.
4. Propagation of disturbances along a chain of unit processes.
5. Nonlinear behavior.
6. Appreciable dead time.
7. Unknown parameters, as when a control system for a new process is commissioned.

In all such situations a conventional controller cannot maintain the performance of the system at acceptable levels. There is, therefore, a need for a special class of control systems which can automatically compensate for these unforeseen variations in parameters and input signals.

Early attempts to overcome such problems involved designing an intentionally nonlinear control system with adjustment of, say, the loop gain. Considerable analytical effort is needed to specify the optimum parameter settings. Such a fixed compensator can be successful over a limited range of operating conditions. Outside this range the compensation is inadequate and there is degradation of the performance. Practical control systems are usually very complex and their dynamics are not known in detail. The extreme changes in the parameters and/or inputs or random fluctuations occurring within the system are often not measurable. In such situations the control engineer has to employ adaptive control systems which automatically sense and correct themselves whenever drastic disturbances and/or severe changes in the parameters occur. It is also possible to introduce a simple learning mechanism in the adaptive part of the system. Any complexity should be added to the system only if improved performance is mandatory and the costs are justifiable.

An adaptive control system has to be designed so as to guarantee stability and also robustness in the presence of disturbances. The designer has also to keep in mind the need for high speed of the adaptation algorithm to compensate for parameter variations and the need to use only a small amount of computer time and capacity.

One should explore alternative solutions before considering adaptive control. There are situations in

noticeably. Thus, the assumption of " slowly time-varying " robot mode is justified.

This multivariable adaptive PID control algorithm has been applied to a three freedom degree robot. Simulation results are presented in support of the proposed adaptive control scheme. The results demonstrate that the adaptive controller performs remarkably well for different reference trajectories, despite gross variations in the payload, end-effector and measurement errors.

The control scheme has guaranteed convergence in that the actual trajectory $q(t)$ asymptotically tracks the reference trajectory $q_r(t)$ for any arbitrary time function $q_r(t)$.

The assurance of asymptotic tracking is the main attraction of the Lyapunov design of adaptive control laws and relieves the designer from the complicated analysis of the stability problem.

The speed of adaptation depends on the magnitudes of the adaptation gains and on the reference trajectory $q_r(t)$.

Practical limitations, such as actuator saturation must be taken into consideration in the choice of the adaptation gains and the reference trajectory.

A novel feature of the proposed scheme is the utilization of adaptive feedforward control and auxiliary signal which result in improved tracking performance.

An attractive feature of the control scheme is that to generate the control action, it neither requires the complex mathematical model of the manipulator dynamics nor any knowledge of the manipulator parameters and the payload. The control scheme uses only the information contained in the actual and reference trajectories which are directly available. Furthermore, the adaptation laws generate the controller gains by means of simple arithmetic operations and hence the calculation of the control action is extremely simple and fast. These attributes make the proposed scheme suitable for implementation in on-line control of robot manipulators with high sampling rates; particularly when all physical parameters of the manipulator cannot be measured accurately and the mass of the payload can vary substantially.

BÖLÜM 1

GİRİŞ

Endüstride kullanım alanları kaynak yapma, boyama ve bir cismi alıp yerine koyma işlemleri gibi basit işlerle sınırlı robotların kontrol sistemleri genel olarak eklem açılarını (deplasmanlarını) ayrı ayrı kontrol eden bir birlerinden bağımsız konum kontrol çevrimleri içermektedirler. Önceden hesaplanmış parametreleri içeren bu konum kontrol sistemleri robotun uygulama alanını, işlem hızını ve değişik yükler altında istenen dinamik davranışı göstermesini sınırlar[1],[2],[3].

Robotları montaj, önceden verilen bir yörüngeyi izleme, hızlı ve seri işlemler yapma gibi daha karmaşık işlerde kullanmak için kullanılan kontrol sistemini geliştirmek gerekir. Bu tip kontrol sistemleri geliştirilirken eklem-ler arasındaki dinamik etkileşimler gözönüne alınmalıdır.

Robotlarda gelişmiş kontrol sistemlerinin dizaynı için oldukça geniş bir çalışma alanı vardır. Bunları iki kategoriye ayırmak mümkündür. Birinci kategori geribeslemeli nonlineer bir kontrolör yardımı ile robot dinamiğindeki nonlineerliklerin iptal edilmesi esasına dayanan robot kontrol tekniklerini içerir. Model esaslı bu kontrol tekniklerinde dinamik modelin, manipülatör parametrelerinin ve yükün tam olarak bilinmesine ihtiyaç vardır. Bu durum pratikte buluşulması güç olan bir takım talepler doğurur (Çalışma şartına bağlı olarak değişen sürtünmenin modellenmesi gibi). Aksi taktirde robottan istenen performans elde edilemez ve kapalı çevrim sistem

kararlılığı bozulabilir.

İkinci kategori, robotik manipulatorlerde olduğu gibi parametreleri bilinmeyen ve dinamik modeli hayli karmaşık olan sistemler için uygun olan adaptif kontrol teorisine dayanır. Adaptif kontrolörün görevi, robot cevabına bağlı olarak kazançlarının ayarlanarak robot performansının verilen bir referans modelin performansını göstermesini sağlamaktır[4],[5],[6],[7].

Tezin ikinci bölümünde robotların karakteristik yönleri ve sınıflandırılması anlatılmıştır.

Üçüncü bölümde N serbestlik dereceli robotlarda simülasyon için robot dinamik modelinin elde edilmesi anlatılmıştır.

Dördüncü bölümde çok değişkenli adaptif kontrol sisteminin yapısı ve adaptasyon kuralları anlatılmıştır.

Beşinci bölümde N serbestlik dereceli bir robotun adaptif kontrolü için oluşturulan simülasyon programının yapısı ve adaptif kontrolörün performansı anlatılmıştır.

Sonuçlar ve öneriler bölümünde üç serbestlik bir robota adaptif PID kontrol uygulanarak değişik çalışma şartları altında adaptif PID kontrolörün performansı hakkında bilgiler verilmiştir.

Eklerde simülasyon programının bilgisayar çıktısı ile simülasyonu yapılan üç serbestlik dereceli robotun yapısı verilmiştir.

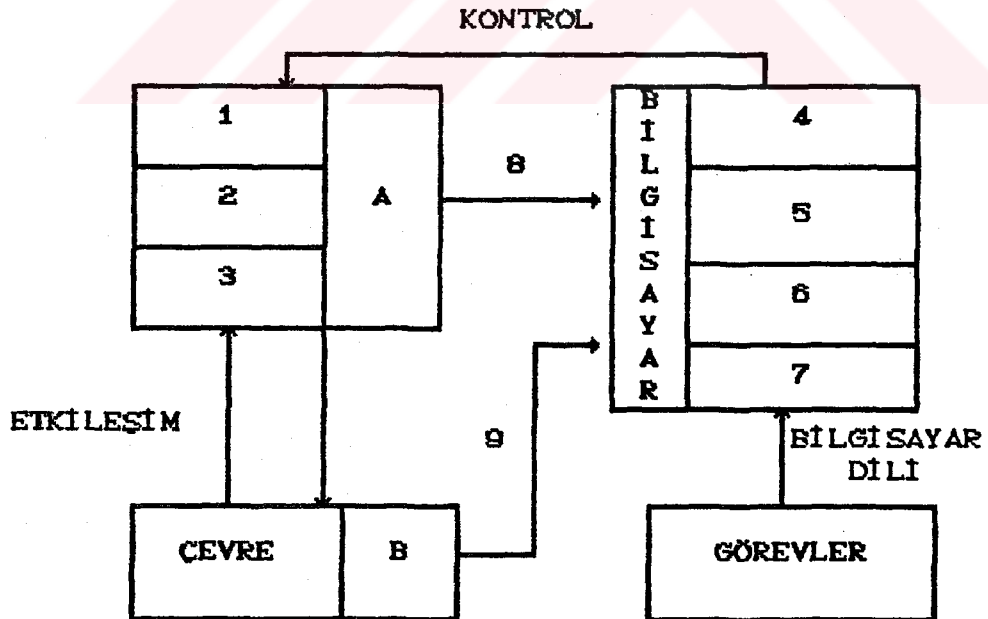
BÖLÜM 2

ROBOTLAR HAKKINDA GENEL BİLGİ

2.1 Robotların Karakteristik Yönleri

- 1- Geometrisi değişken olduğu için çok yönlüdürler.
- 2- Önceden verilen işi sonradan gelen etkilere rağmen yaptıklarından dolayı oto adaptasyon özellikleri vardır. Bu özelliği sağlayan duyarga girişleri, zekası ve hareket fonksiyonları vardır.

2.2 Robot Sistemi



Sekil 2.1 Bir robot sisteminin yapısı

- 1- eklemli mekanik sistem
- 2- transmisyon düzenleri
- 3- sürücüler (güç ileticiler)
- 4- robot modeli
- 5- çevre modeli
- 6- yapılacak görevler
- 7- kontrol algoritmaları
- 8- iç duyarga verileri
- 9- dış duyarga verileri
- A- iç duyargalar
- B- dış duyargalar

Bilgisayar bloğunda uygulanan kontrol tipine bağlı olarak robot modeli ve ayrıca endüstriyel robotlarda çevre bloğu bulunmayabilir.

2.3 Robotların Sınıflandırılması

Robotlar mekanik yapılarına (çalıştıkları koordinat sistemlerine), kapasitelerine ve uygulanan kontrol yöntemine göre sınıflandırılabilir.

Mekanik yapılarına göre robotlar dört sınıfa ayrılırlar.

- 1- kartezyen koordinatlarda çalışan robotlar
- 2- silindirik koordinatlarda çalışan robotlar
- 3- küresel koordinatlarda çalışan robotlar
- 4- polar koordinatlarda çalışan robotlar

Kapasitelerine göre robotlar beş gruba ayrılırlar.

- 1- ardışıl işlemeli (belli konumlara önceden programlanan) robotlar
- 2- play-back robotlar (Teach-Pendant ile istenilen tüm konumlara getirilerek yapılacak işler ezberletilir)
- 3- kontrollü yollu robotlar

4- adaptif robotlar (geri besleme çevreden gelen bilgilerle yapılır)

5- akıllı (çevreyi modelleyebilen, öğrenebilen ve ona göre hareket edebilen) robotlar

Uygulanan kontrol yöntemine robotlar iki gruba ayrılırlar.

1- noktadan noktaya giden robotlar

2- sürekli yol izleyen robotlar

Noktadan noktaya giden robotlar ara noktalarda duramazlar. Sürekli yol izleyen robotlar belli bir yol boyunca birçok noktada durabilirler.

BÖLÜM 3

ROBOTLARIN DİNAMİK MODELİ

Bir robot kolunun dinamik modeli robot kolunun dinamik davranışını belirleyen hareket denklemlerinden oluşur. Bu denklemlerin bilgisayar yardımı ile simülasyonu robot ve robot kontrol tasarımı açısından büyük kolaylık sağlar.

Robotun dinamik modeli Newton mekaniği, Lagrange mekaniği gibi bilinen fizik kurallarından elde edilir. Fizik kuralı ne olursa olsun elde edilecek model aynıdır. Bu çalışmada bilgisayarla dinamik modelin nümerik olarak elde edilmesi açısından büyük kolaylıklar sağladığından Lagrange-Euler formülasyonu kullanıldı [8],[9].

3.1 Lagrange-Euler Formülasyonu

Bu yöntem

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_1} \right] - \frac{\partial L}{\partial q_1} = T_1 \quad i=1, \dots, N \quad (3.1)$$

$$L = K - P \quad (3.2)$$

denklemleri ile ifade edilir. Burada

L : Lagrange fonksiyonunu,

K : Robot uzuvlarının toplam kinetik enerjisini,

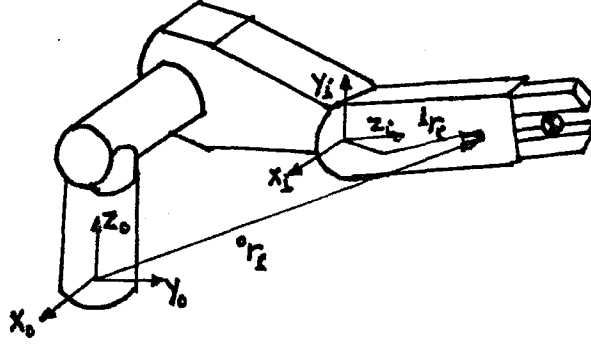
P : Robot uzuvlarının toplam potansiyel enerjisini,

T_1 : Sisteme uygulanan genelleştirilmiş moment (veya kuvveti),

q_1 : Eklem değişkenlerini,

$\dot{q}_i$: Eklem değişkenlerinin hızlarını, ifade etmektedir.

3.2 Lagrange-Euler Denklemi İçin Bazı Tanımlar



Şekil 3. 1 Robot kolunda 1 .nci uzuv üzerinde bir noktanın gösterimi

$${}^1r_1 = [x_1, y_1, z_1, 1]^T \quad (3. 3)$$

$${}^{1-1}A_1 = \begin{vmatrix} \cos \theta_1 & -\cos \alpha_1 \sin \theta_1 & \sin \alpha_1 \sin \theta_1 & a_1 \cos \theta_1 \\ \sin \theta_1 & \cos \alpha_1 \cos \theta_1 & -\sin \alpha_1 \cos \theta_1 & a_1 \sin \theta_1 \\ 0 & \sin \alpha_1 & \cos \alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3. 4a)$$

$${}^{1-1}A_1 = \begin{vmatrix} \cos \theta_1 & -\cos \alpha_1 \sin \theta_1 & \sin \alpha_1 \sin \theta_1 & 0 \\ \sin \theta_1 & \cos \alpha_1 \cos \theta_1 & -\sin \alpha_1 \cos \theta_1 & 0 \\ 0 & \sin \alpha_1 & \cos \alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3. 4b)$$

${}^{i-1}A_i$.nci uzvun (i-1) .inci uzvun koordinat sistemine göre transformasyon matrisidir. (3. 4a) döner eklemler için (3. 4b) prizmatik eklemler için transformasyon matrisleridir. Burada θ_i , α_i , a_i ve d_i Denavit-Hartenberg parametreleridir.

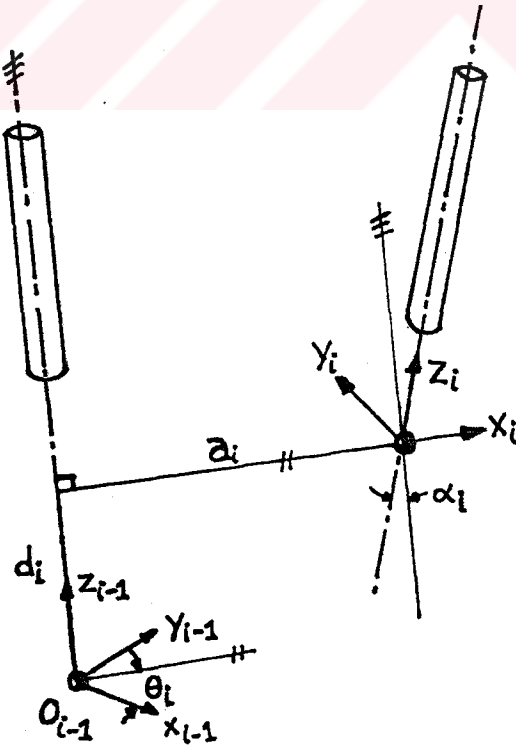
θ_i : z_{i-1} eksenini etrafında, x_{i-1} den x_i eksenine doğru olan eklem açısıdır.

α_i : x_i eksenini etrafında , z_{i-1} ekseninden, z_i eksenine kadar olan açıdır.

a_i : z_{i-1} eksenini ile x_i ekseninin kesim noktasından x_i eksenini boyunca uzanan i .nci koordinat sisteminin başlangıcına kadar olan uzaklıktır.

d_i : z_{i-1} eksenini boyunca, (i-1) .inci koordinat sisteminin başlangıcından, z_{i-1} eksenini ve x_i ekseninin kesim noktasına kadar olan uzaklıktır.

Döner eklemler için θ_i , prizmatik eklemler için d_i eklem değişkenleridir ve diğer parametreler sabittir.



Sekil 3. 2 Denavit-Hartenberg parametrelerinin gösterimi

Bir noktanın ana koordinat sistemine göre uzaklığı

$${}^0r_1 = {}^0A_1 {}^1r_1 \quad (3. 5)$$

şeklinde yazılır. Burada

$${}^0A_1 = {}^0A_1 {}^1A_2 \dots {}^{i-1}A_i \quad (3. 6)$$

0r_1 nin ana koordinat sistemine göre hızı

$$\begin{aligned} {}^0v_1 \equiv \dot{v}_1 &= \frac{d}{dt}({}^0r_1) = \frac{d}{dt}({}^0A_1 {}^1r_1) \\ &= {}^0\dot{A}_1 {}^1A_1 \dots {}^{i-1}A_i {}^1r_1 + {}^0A_1 {}^1\dot{A}_1 \dots {}^{i-1}A_i {}^1r_1 + \dots \end{aligned}$$

$${}^0\dot{A}_1 {}^1A_1 \dots {}^{i-1}A_i {}^1r_1 = \left[\sum_{j=1}^i \frac{\partial {}^0A_i}{\partial q_j} \dot{q}_j \right] {}^1r_1 \quad (3. 7)$$

şeklinde ifade edilir. Burada 1r_1 sabit olduğu için zamana göre türevi sıfırdır.

$${}^1\dot{r}_1 = 0 \quad (3. 8)$$

kA_i matrislerinin q_j ye göre kısmi türevlerinin kolaylıkla hesaplanabilmesi açısından döner eklemler için

$$Q_i \equiv \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3. 9a)$$

prizmatik eklemler için

$$Q_i \equiv \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3. 9b)$$

tanımları kullanılırsa

$$\frac{\partial^{i-1} A_i}{\partial q_i} = Q_i^{i-1} A_i \quad (3. 10)$$

yazılır.

Kısa ifadeler kullanmak açısından

$$U_{ij} \equiv \frac{\partial^0 A_i}{\partial q_j} \equiv {}^0 A_{j-1} Q_j^{j-1} A_i \quad j \leq i \quad (3. 11a)$$

$$U_{ij} \equiv 0 \quad j \geq i \quad (3. 11b)$$

tanımlamaları yapılırsa (3. 5) denklemi

$$v_i = \left[\sum_{j=1}^i U_{ij} q_i \right] {}^1 r_i \quad (3. 12)$$

şeklinde yazılır.

(3. 10) ve (3. 11) denklemlerinden hareketle

$$U_{ijk} \equiv \frac{\partial U_{ij}}{\partial q_j} \equiv {}^0 A_{j-1} Q_j^{j-1} A_{k-1} Q_k^{k-1} A_i \quad i \geq k \geq j \quad (3. 13a)$$

$$U_{ijk} \equiv {}^0 A_{k-1} Q_k^{k-1} A_{j-1} Q_j^{j-1} A_i \quad i \geq j \geq k \quad (3. 13b)$$

$$U_{ijk} \equiv 0 \quad i < j \text{ veya } i < k \quad (3. 13c)$$

tanımlamaları yapılır.

3.3 Robot Kolunun Kinetik Enerjisi

dK_1 1 .nci uzuv üzerinde bir parçacığın kinetik enerjisi olmak üzere şu şekilde yazılır.

$$dK_1 = \frac{1}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) dm = \frac{1}{2} \text{trace}(\dot{\mathbf{v}}_1 \dot{\mathbf{v}}_1^T) dm = \frac{1}{2} \text{tr}(\dot{\mathbf{v}}_1 \dot{\mathbf{v}}_1^T) dm \quad (3.14)$$

(3.12) denklemi (3.14) ifadesinde yerine yazılırsa

$$\begin{aligned} dK_1 &= \frac{1}{2} \text{tr} \left[\sum_{p=1}^1 \mathbf{U}_{1p} \dot{\mathbf{q}}_p^T \mathbf{r}_1 \left[\sum_{r=1}^1 \mathbf{U}_{1r} \dot{\mathbf{q}}_r^T \mathbf{r}_1 \right]^T \right] dm \\ &= \frac{1}{2} \text{tr} \left[\sum_{p=1}^1 \sum_{r=1}^1 \mathbf{U}_{1p} \mathbf{r}_1 \mathbf{r}_1^T \mathbf{U}_{1r}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right] dm \\ &= \frac{1}{2} \text{tr} \left[\sum_{p=1}^1 \sum_{r=1}^1 \mathbf{U}_{1p} (\int \mathbf{r}_1 \mathbf{r}_1^T dm) \mathbf{U}_{1r}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right] \end{aligned} \quad (3.15)$$

elde edilir. Bu ifadenin integrali

$$K_1 = \int dK_1 = \frac{1}{2} \text{tr} \left[\sum_{p=1}^1 \sum_{r=1}^1 \mathbf{U}_{1p} (\int \mathbf{r}_1 \mathbf{r}_1^T dm) \mathbf{U}_{1r}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right] \quad (3.16)$$

şeklindedir. Burada

$$\mathbf{J}_1 = \int \mathbf{r}_1 \mathbf{r}_1^T dm \quad (3.17)$$

$$\mathbf{J}_1 = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_1 \bar{x}_1 \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_1 \bar{y}_1 \\ I_{xz} & I_{yx} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_1 \bar{z}_1 \\ m_1 \bar{x}_1 & m_1 \bar{y}_1 & m_1 \bar{z}_1 & m_1 \end{bmatrix} \quad (3.18)$$

Burada $[\bar{x}_1 \ \bar{y}_1 \ \bar{z}_1]$ i .nci koordinat sistemine göre i .nci uzvun ağırlık merkezinin koordinatlarıdır.

J_1 i .nci uzva ait atalet matrisidir. $(x_1 \ , \ y_1 \ , \ z_1)$ koordinat sisteminde m_1 rijit cisminin jirasyon yarı çapı ifadesi kullanımı ile J_1

$$J_1 = \begin{bmatrix} \frac{-k_{11}^2 + k_{22}^2 + k_{33}^2}{2} & k_{12}^2 & k_{13}^2 & \bar{x}_1 \\ k_{12}^2 & \frac{k_{11}^2 - k_{22}^2 + k_{33}^2}{2} & k_{23}^2 & \bar{y}_1 \\ k_{13}^2 & k_{23}^2 & \frac{k_{11}^2 + k_{22}^2 - k_{33}^2}{2} & \bar{z}_1 \\ m_1 \bar{x}_1 & m_1 \bar{y}_1 & m_1 \bar{z}_1 & 1 \end{bmatrix} m_1 \quad (3. 19)$$

şeklinde yazılır.

$$K = \sum_{i=1}^N dK_1 = 1/2 \sum_{i=1}^N \text{tr} \left[\begin{array}{cc} 1 & 1 \\ \sum_{p=1} & \sum_{r=1} U_{ip} J_i U_{ir}^T \end{array} \begin{array}{cc} \cdot & \cdot \\ q_p & q_r \end{array} \right] \\ = 1/2 \sum_{i=1}^N \sum_{p=1}^1 \sum_{r=1}^1 [\text{tr}(U_{ip} J_i U_{ir}^T) q_p q_r] \quad (3. 20)$$

Robot kolunun kinetik enerjisi olarak (3. 20) denklemi elde edilir.

3.4 Robot Kolunun Potansiyel Enerjisi

i .nci uzvun potansiyel enerjisi P_1

$$P_1 = -m_1 g \cdot r_1 = -m_1 g ({}^0 A_1^1 r_1) \quad i=1, \dots, N \quad (3. 21)$$

Burada $g = [0 \ 0 \ -9.81 \ 0]^T$ yerçekimi sabitidir. Robot kolunun toplam potansiyel enerjisi P şu şekilde yazılır:

$$P = \sum_{i=1}^N P_i = \sum_{i=1}^N -m_i g ({}^0A_1^i r_i) \quad (3. 22)$$

3.5 Robot Kolunun Hareket Denklemi

(3. 20) ile (3. 22) denklemleri (3. 2) de yerlerine koyulursa

$$L = 1/2 \sum_{i=1}^N \sum_{p=1}^i \sum_{r=1}^i [\text{tr}(U_{ip} J_i U_{ir}^T) \dot{q}_p \dot{q}_r] + \sum_{i=1}^N m_i g ({}^0A_1^i r_i) \quad (3. 23)$$

şeklinde Lagrange fonksiyonu elde edilir.

Lagrange fonksiyonu (3. 1) denkleminde yazılır ve (3. 10) , (3. 11) ile (3. 13) ifadeleri kullanılırsa

$$\begin{aligned} T_i = \frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} &= \sum_{j=1}^N \sum_{k=1}^j [\text{tr}(U_{jk} J_j U_{ji}^T) \dot{q}_k] \\ &+ \sum_{j=1}^N \sum_{k=1}^j \sum_{m=1}^j [\text{tr}(U_{jkm} J_j U_{ji}^T) \dot{q}_k \dot{q}_m] - \sum_{j=1}^N m_j g (U_{ji}^T r_j) \end{aligned} \quad (3. 24)$$

$$T_i = \sum_{k=1}^N D_{ik} \dot{q}_k + \sum_{k=1}^N \sum_{m=1}^N h_{ikm} \dot{q}_k \dot{q}_m + c_i \quad i=1, \dots, N \quad (3. 25)$$

şeklinde robot kolunun dinamik denklemi elde edilir [3].

$$T = D(q) \ddot{q} + h(q, \dot{q}) + c(q) \quad (3. 26)$$

$$D_{ik} = \sum_{j=\max(i,k)}^J [\text{tr}(U_{jk} J_j U_{ji}^T) \ddot{q}_k] \quad i, k=1, \dots, N \quad (3. 27)$$

$$h_i = \sum_{k=1}^N \sum_{m=1}^N h_{ikm} \dot{q}_k \dot{q}_m \quad (3. 28)$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^J \text{tr}(U_{jk} J_j U_{ji}^T) \quad i, k, m=1, \dots, N \quad (3. 29)$$

$$c_i = - \sum_{j=1}^N m_j g (U_{ji}^T J_j) \quad i=1, \dots, N \quad (3. 30)$$

Robot kolunun bir yükü taşıması durumunda (3. 26) denklemi şu şekle gelir:

$$T = D(q) \ddot{q} + h(q, \dot{q}) + c(q) + m L(q, \dot{q}, \ddot{q}) \quad (3. 31)$$

$$m L(q, \dot{q}, \ddot{q}) = m J^T(q) [J(q) \ddot{q} + \dot{J}^T(q, \dot{q}) \dot{q} + g] \quad (3. 32)$$

Burada m taşınan yükün ağırlığı, J ise 3*N jakobian matristir. g =[0 0 -9.81] biçiminde yerçekimi sabitidir.

$$J = [J_{L1} \mid J_{L2} \mid \dots \mid J_{LN}] \quad (3. 33)$$

Döner eklemler için

$$J_{Li} = \hat{B}_{i-1} \times \hat{r}_{i-1} \quad i=1, \dots, N \quad (3. 34)$$

Öteleme yapan eklemler için

$$J_{Li} = \hat{B}_{i-1} \quad i=1, \dots, N \quad (3. 35)$$

Burada $\times$ vektörel çarpımı göstermektedir. $\hat{B}_{1-1}$, z_1 eksenini doğrultusundaki vektördür ve

$$\hat{B}_{1-1} = R_0^1 R_1^2 \dots R_{1-2}^{1-1} \hat{B} \quad i=1, \dots, N \quad (3. 36)$$

olarak tanımlanır.

$\hat{B} = [0 \ 0 \ 1]^T$ tabanın z_0 eksenini doğrultusundaki birim vektördür. $\hat{B}_{1-1}$ transformasyon matrislerinden hesaplanan konum vektörüdür [10].

Eklemlerdeki viskoz ve Coulomb sürtünmeleri gözönüne alınırsa (3. 28) denklemi

$$T = D(\dot{q}) \ddot{q} + h(q, \dot{q}) + c(q) + H(\dot{q}) \quad (3. 37)$$

şekline gelir.

$$H = \begin{bmatrix} V1(1)\dot{q}_1 + V2(1)\text{sgn}(\dot{q}_1) \\ \dots \dots \dots \\ V1(N)\dot{q}_N + V2(N)\text{sgn}(\dot{q}_N) \end{bmatrix} \quad (3. 38)$$

Burada $V1(i)$ i .nci eklemdenki viskoz sürtünme katsayısı ve $V2(i)$ i .nci eklemdenki Coulomb sürtünme katsayılarıdır [8].

Robot kolunun uç noktasına $F=[F_x, F_y, F_z, M_x, M_y, M_z]$ şeklinde bir kuvvet etkimesi durumunda (3. 28) denklemi şu şekle gelir:

$$T = D(\dot{q}) \ddot{q} + h(q, \dot{q}) + c(q) + J^T(q)F \quad (3. 39)$$

Burada $J^T(q)$ $6 \times N$ jakobian matristir [10].

$$J = \begin{bmatrix} J_{L1} & J_{L2} & \dots & J_{LN} \\ J_{A1} & J_{A2} & \dots & J_{AN} \end{bmatrix} \quad (3. 40)$$

Döner eklemler için

$$J_{A1} = \dot{B}_{1-1} \quad i=1, \dots, N \quad (3. 41)$$

prizmatik eklemler için

$$J_{A1} = 0 \quad i=1, \dots, N \quad (3. 42)$$

Genel olarak bir robot kolunun hareket denklemi şu şekildedir:

$$T = D(q) \ddot{q} + h(q, \dot{q}) + c(q) + m L(q, \dot{q}, \ddot{q}) + J^T(q) F + H(q) \dot{q} \quad (3. 43)$$

BÖLÜM 4

N SERBESTLİK DERECELİ ROBOT İÇİN ADAPTİF KONTROL SİSTEMİNİN YAPISI

4.1 Sabit Parametrelili Kontrolörün Tasarımı

(3. 43) formülü ile verilen robot dinamik modeli

$$T(t) = M^*(m, q) \ddot{q} + N^*(m, q, \dot{q}) + G^*(m, q) + H(\dot{q}) \quad (4. 1)$$

şeklinde yazılabilir.

Burada

$$M^*(m, q) = D(q) + mJ^T(q)J(q) \quad (4. 2)$$

$$N^*(m, q, \dot{q}) = h(q, \dot{q}) + mJ^T(q)\dot{J}(q, \dot{q}) \quad (4. 3)$$

$$G^*(m, q) = J^T(q)F + c(q) + mJ^T(q)g \quad (4. 4)$$

(4. 1) ifadesini $t_i \leq t < t_i + \Delta t$ zaman aralığında $p = [T(t_i), q(t_i), \dot{q}(t_i)]$ çalışma noktası civarında lineerleştirerek ve aşağıdaki tanımlamalar yapılırsa (4. 7) ifadesi elde edilir.

$$A = [M^*]_p \quad ; \quad B = \left[\frac{\partial N^* + H}{\partial \dot{q}} \right]_p \quad ; \quad C = \left[\frac{\partial N^* + G^*}{\partial q} \right]_p \quad (4. 5)$$

A, B ve C matrisleri lineerleştirilmiş N*N model matrisleridir.

$$q(t) = q(t) - q(t_i) \quad (4. 8a)$$

$$\dot{q}(t) = \dot{q}(t) - \dot{q}(t_i) \quad (4. 8b)$$

$$\ddot{q}(t) = \ddot{q}(t) - \ddot{q}(t_i) \quad (4. 8c)$$

$$T_1(t) = T(t) - T(t_i) \quad (4. 8d)$$

$$T_1(t) = A \ddot{q} + B \dot{q} + C q \quad (4. 7)$$

(4. 7) denklemi s domeninde yazılır ve q çözülürse

$$q(s) = (C + Bs + As^2)^{-1} T_1(s) \quad (4. 8)$$

elde edilir.

İleri kol kontrolörü olarak Q(s) lineerleştirilmiş modelin tersi olarak seçilir.

$$Q(s) = [(C + Bs + As^2)^{-1}]^{-1} T_1(s) = (C + Bs + As^2) T_1(s) \quad (4. 9)$$

$$T_{ff}(s) = Q(s)q_r(s) = (C + Bs + As^2) q_r(s)$$

$$T_{ff}(t) = A \ddot{q}_r(t) + B \dot{q}_r(t) + C q_r(t) \quad (4. 10)$$

Burada

$$q_r(t) = q_r(t) - q_r(t_i) \quad (4. 11a)$$

$$\dot{q}_r(t) = \dot{q}_r(t) - \dot{q}_r(t_i) \quad (4. 11b)$$

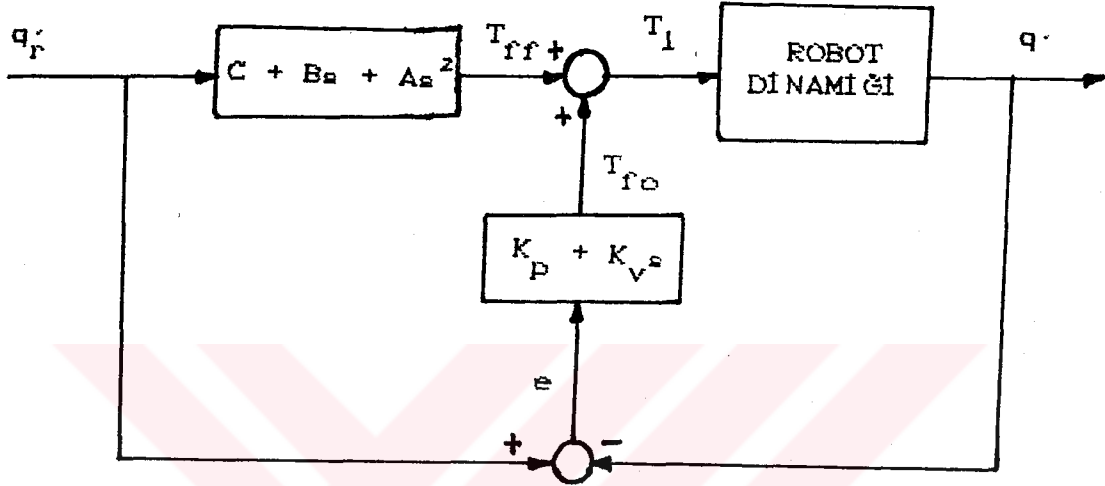
$$\ddot{q}_r(t) = \ddot{q}_r(t) - \ddot{q}_r(t_i) \quad (4. 11c)$$

Geri besleme koluna koyulacak kontrolör şu şekilde seçilir:

$$K(s) = K_p + K_v s \quad (4. 12)$$

$$T_{fo}(s) = K(s) [q_r'(s) - q'(s)] = [K_p + K_v s] e(s)$$

$$T_{fo}(t) = K_p e(t) + K_v \dot{e}(t) \quad (4. 13)$$



Sekil 4.1 Küçük değişimler için kontrol algoritmasının blok diyagramı

Küçük değişimler için toplam kontrol kuralı:

$$\begin{aligned} T_1(s) &= T_{ff}(s) + T_{fo}(s) \\ &= [K_p + K_v s] e(s) + [C + Bs + As^2] q_r'(s) \end{aligned} \quad (4. 14)$$

$$T_1(t) = K_p e(t) + K_v \dot{e}(t) + C q'(t) + B \dot{q}'(t) + A \ddot{q}'(t) \quad (4. 15)$$

Küçük değişimler için elde edilen toplam kontrol kuralı lineerleştirilmiş manipülatör modeline (4. 7) uygulanırsa

$$\ddot{e}(t) + A^{-1}(B + K_p) \dot{e}(t) + A^{-1}(C + K_v) e(t) = 0 \quad (4. 16)$$

şeklinde dinamik hata modeli elde edilir [8].

K_p ve K_v dinamik hata modelinin karakteristik denkleminin kökleri olarak seçilirse (kökler negatif olduğundan) $e(0)$ ve $\dot{e}(0)$ ne olursa olsun daima $t \rightarrow \infty$ için $e(t) \rightarrow 0$ dir. Çünkü dinamik hata modelinde eşitliğin sağ tarafı sıfırdır.

(4. 8) ve (4. 11) ifadelerinden faydalanarak $T(t)$, $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$, $q_r(t)$, $\dot{q}_r(t)$ ve $\ddot{q}_r(t)$ ifadeleri (4. 15) küçük değişimler için toplam kontrol kuralında yazılırsa toplam kontrol kuralı olarak

$$T(t) = T^*(t_i) + K_p E(t) + K_v \dot{E}(t) + Cq_r(t) + B\dot{q}_r(t) + A\ddot{q}_r(t) \quad (4. 17)$$

elde edilir. Burada

$$E(t) \cong q_r(t) - q(t) \quad (4. 18a)$$

$$\dot{E}(t) \cong \dot{q}_r(t) - \dot{q}(t) \quad (4. 18b)$$

$$\ddot{E}(t) \cong \ddot{q}_r(t) - \ddot{q}(t) \quad (4. 18c)$$

$$T^*(t_i) \cong T(t_i) - Cq(t_i) - B\dot{q}(t_i) - A\ddot{q}(t_i) \quad (4. 18d)$$

Hesaplanan K_p ve K_v değerleri izleme hatasını sifıra götürecektir. Bu şekilde seçilmesi tamamen $Q(s)$ ileri kol kontrolörünün doğru olarak bilinmesine yani robot modelindeki bütün parametrelerin bilinmesine ihtiyacı vardır. Pratikte bu parametrelerin doğru olarak bilinip modellenmesi zor olduğundan izleme hatası sifıra belli bir hata ile asimptotik olarak yakınsar [11].

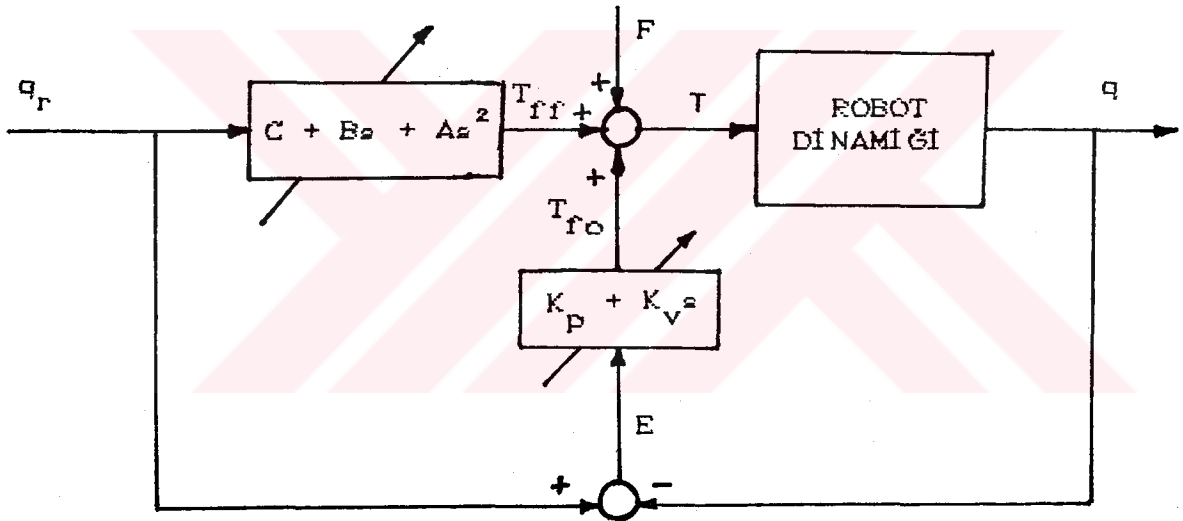
4.2 Adaptif Kontrolör Tasarımı

(4. 1) denlemini

$$T(t) = A^*(m, q) \ddot{q}(t) + B^*(m, q, \dot{q}) \dot{q}(t) + C^*(m, q, \dot{q}) q(t)$$

(4. 19)

şeklinde yazmak mümkündür. Bu parametreleri bilinmeyen $N \times N$ sistem matrisleri oldukça karmaşık ve hayli nonlineer ifadelerden oluşmaktadır. Bu sistemin belli bir yörüngeyi sıfır hata ile izlemesini sağlamak için 4.1 de elde edilen A , B , C , ve K_p , K_v $N \times N$ matrisleri adaptif olarak her t anında ayarlanmalıdır.



Sekil 4.2 Adaptif kontrol sistemi

(4. 17) ile verilen toplam kontrol kuralında

$$F(t) = T^*(t) \quad (4. 20)$$

$F(t)$ adaptif kontrol için yardımcı sinyal ifadesi kullanılırsa adaptif kontrol için toplam kontrol kuralı

$$T(t) = K_p E(t) + K_v \dot{E}(t) + A \ddot{q}_r(t) + B \dot{q}_r(t) + C q_r(t) + F(t)$$

(4. 21)

şeklinde elde edilir.

(4. 18) denklemi ile (4. 21) denklemi taraf tarafa eşitlenirse

$$A^* \ddot{q} + B^* \dot{q} + C^* q = F + K_p E + K_v \dot{E} + Aq_r + Bq_r + Cq_r \quad (4. 22)$$

elde edilir.

(4. 18) ifadelerinden $\ddot{q}$, $\dot{q}$ ve q terimleri E , $\dot{E}$, $\ddot{E}$, $\ddot{q}_r$, $\dot{q}_r$ ve q_r terimleri cinsinden ifade edilerek (4. 22) denkleminde yerine koyulursa

$$A^* \ddot{E} + (B^* + K_v) \dot{E} + (C^* + K_p) E = -F + (A^* - A) \ddot{q}_r + (B^* - B) \dot{q}_r + (C^* - C) q_r \quad (4. 23)$$

elde edilir.

$$Z(t) = \begin{bmatrix} E(t) \\ \dot{E}(t) \end{bmatrix} \quad (4. 24)$$

şeklindeki konum hata-hız vektörü tanımıyla (4. 23) denklemi

$$\begin{aligned} \dot{Z} = & \begin{bmatrix} 0 & I_N \\ -[A^*]^{-1} [C^* + K_p] & -[A^*]^{-1} [B^* + K_v] \end{bmatrix} Z + \begin{bmatrix} 0 \\ -[A^*]^{-1} [F] \end{bmatrix} \\ & + \begin{bmatrix} 0 \\ -[A^*]^{-1} [C^* - C] \end{bmatrix} q_r + \begin{bmatrix} 0 \\ -[A^*]^{-1} [B^* - B] \end{bmatrix} \dot{q}_r + \begin{bmatrix} 0 \\ -[A^*]^{-1} [A^* - A] \end{bmatrix} \ddot{q}_r \end{aligned} \quad (4. 25)$$

biçimine girer.

Elde edilen bu denklem artık ayarlanabilir sistem şekline dönüşmüş oldu. Adaptif kontrol için model referans denklemi şöyle verilebilir.

$$E_i(t) + 2\zeta_i\omega_i E_i(t) + E_i(t) = 0 \quad i=1, \dots, N \quad (4. 26)$$

Burada ζ_i sönüm oranı ve ω_i ise sönümsüz sistemin doğal frekansıdır. Bu denklem

$$\dot{Z}_m(t) = \begin{bmatrix} 0 & I_N \\ -D_1 & -D_2 \end{bmatrix} Z_m(t) = D Z_m(t) \quad (4. 27)$$

şekline getirilebilir.

Burada $D_1 = \text{diag}(\omega_i^2)$ ve $D_2 = \text{diag}(2\zeta_i\omega_i)$ $N \times N$ diagonal sabit bilinen matrislerdir.

$$Z_m(t) = \begin{bmatrix} E_m(t) \\ E_m(t) \end{bmatrix} \quad (4. 28)$$

4.3 Lyapunov Denkleminin Çözümü

Lyapunov denklemi

$$P D + D^T P = -Q \quad (4. 29)$$

şeklindedir.

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \quad 2N \times 2N \text{ matrisi Lyapunov eşitliğini}$$

saglayan her elemanı diagonal matrisler olan simetrik pozitif matristir.

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \quad 2N \times 2N \text{ matrisi her elemanı diagonal}$$

bilinen pozitif matristir [12].

(4. 29) Lyapunov eşitliğinde D ifadesi yerine koyulursa

$$\begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \begin{bmatrix} 0 & I_N \\ -D_1 & -D_2 \end{bmatrix} + \begin{bmatrix} 0 & -D_1 \\ I_N & -D_2 \end{bmatrix} \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} = \begin{bmatrix} -2Q_1 & 0 \\ 0 & -2Q_2 \end{bmatrix}$$

(4. 30)

elde edilir. Bu denklemin çözümü

$$P_1 = Q_1 D_2^{-1} + Q_1 D_1^{-1} D_2 + Q_2 D_2^{-1} D_1 \quad (4. 31)$$

$$P_2 = Q_1 D_1^{-1} \quad (4. 32)$$

$$P_3 = Q_2 D_2^{-1} + Q_1 D_1^{-1} D_2^{-1} \quad (4. 33)$$

P pozitif bir matris olduğu için

$$(P_{11} + P_{31}) > 0 ; (P_{11}P_{31} - P_{21}^2) > 0 \quad i=1, \dots, N \quad (4.34)$$

esitsizliklerini sağlamalıdır [13].

4.4 Lyapunov Fonksiyonu Ve Adaptasyon Kuralı

$$\dot{Z} = \begin{bmatrix} 0 & I_N \\ -\Delta_1 & -\Delta_2 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \Delta_0 \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_3 \end{bmatrix} \dot{q}_r + \begin{bmatrix} 0 \\ \Delta_4 \end{bmatrix} \ddot{q}_r + \begin{bmatrix} 0 \\ \Delta_5 \end{bmatrix} \ddot{\ddot{q}}_r \quad (4.35)$$

Burada

$$\begin{aligned} \Delta_1 &= [A^*]^{-1} [C^* + K_p] , \quad \Delta_2 = [A^*]^{-1} [B^* + K_v] , \quad \Delta_3 = [A^*]^{-1} [C^* - C] \\ \Delta_4 &= [A^*]^{-1} [B^* - B] , \quad \Delta_5 = [A^*]^{-1} [A^* - A] , \quad \Delta_0 = -[A^*]^{-1} F \end{aligned} \quad (4.36)$$

Dinamik hata modeli (4.27) ile (4.35) denklemlerinin taraf tarafa çıkarılması ile

$$\begin{aligned} \dot{e} &= \begin{bmatrix} 0 & I_N \\ -D_1 & -D_2 \end{bmatrix} e + \begin{bmatrix} 0 & I_N \\ -\Delta_1 - D_1 & -\Delta_2 - D_2 \end{bmatrix} Z + \begin{bmatrix} 0 \\ -\Delta_0 \end{bmatrix} + \begin{bmatrix} 0 \\ -\Delta_3 \end{bmatrix} \dot{q}_r \\ &\quad + \begin{bmatrix} 0 \\ -\Delta_4 \end{bmatrix} \ddot{q}_r + \begin{bmatrix} 0 \\ -\Delta_5 \end{bmatrix} \ddot{\ddot{q}}_r \end{aligned} \quad (4.37)$$

şeklinde elde edilir. Buna göre Lyapunov fonksiyonu şu şekilde yazılır [12].

$$\begin{aligned}
 V = & e^T P e + [\Delta_0 - \Delta_0^*]^T Q_0 [\Delta_0 - \Delta_0^*] + \text{tr}\{[\Delta_1 - D_1 - \Delta_1^*]^T Q_1 [\Delta_1 - D_1 - \Delta_1^*]\} \\
 & + \text{tr}\{[\Delta_2 - D_2 - \Delta_2^*]^T Q_2 [\Delta_2 - D_2 - \Delta_2^*]\} + \text{tr}\{[\Delta_3 - \Delta_3^*]^T Q_3 [\Delta_3 - \Delta_3^*]\} \\
 & + \text{tr}\{[\Delta_4 - \Delta_4^*]^T Q_4 [\Delta_4 - \Delta_4^*]\} + \text{tr}\{[\Delta_5 - \Delta_5^*]^T Q_5 [\Delta_5 - \Delta_5^*]\} \quad (4. 38)
 \end{aligned}$$

Lyapunov fonksiyonunun zamana göre türevini alırsak

$$\begin{aligned}
 \dot{V} = & -e^T Q e + 2\Delta_0^T [R + Q_0 \dot{\Delta}_0 - Q_0 \dot{\Delta}_0^*] - 2\Delta_0^{*T} Q_0 [\dot{\Delta}_0 - \dot{\Delta}_0^*] \\
 & + 2\text{tr}\{[\Delta_1 - D_1]^T [-R E^T + Q_1 \dot{\Delta}_1 - Q_1 \dot{\Delta}_1^*] - \Delta_1^{*T} Q_1 [\dot{\Delta}_1 - \dot{\Delta}_1^*]\} \\
 & + 2\text{tr}\{[\Delta_2 - D_2]^T [-R E^T + Q_2 \dot{\Delta}_2 - Q_2 \dot{\Delta}_2^*] - \Delta_2^{*T} Q_2 [\dot{\Delta}_2 - \dot{\Delta}_2^*]\} \\
 & + 2\text{tr}\{\Delta_3^T [R q_r^T + Q_3 \dot{\Delta}_3 - Q_3 \dot{\Delta}_3^*] - \Delta_3^{*T} Q_3 [\dot{\Delta}_3 - \dot{\Delta}_3^*]\} \\
 & + 2\text{tr}\{\Delta_4^T [R q_r^T + Q_4 \dot{\Delta}_4 - Q_4 \dot{\Delta}_4^*] - \Delta_4^{*T} Q_4 [\dot{\Delta}_4 - \dot{\Delta}_4^*]\} \\
 & + 2\text{tr}\{\Delta_5^T [R q_r^T + Q_5 \dot{\Delta}_5 - Q_5 \dot{\Delta}_5^*] - \Delta_5^{*T} Q_5 [\dot{\Delta}_5 - \dot{\Delta}_5^*]\} \quad (4. 39)
 \end{aligned}$$

Burada $R = -[P_2, P_3]$ le hata ağırlık vektörüdür.

$Z_m = 0$ olduğundan $e = -Z$ olur. Bu durum göz önüne alınarak

$$R = -[P_2, P_3] \dot{e} = [P_2, P_3] Z = P_2 \dot{E} + P_3 \dot{E} \quad (4. 40)$$

yazılabilir.

(4. 39) denkleminde amaç $\dot{V}$ yı negatif yapmaktır. Bunun için şu tanımlamalar yapılabilir:

$$R + Q_0 \dot{\Delta}_0 - Q_0 \dot{\Delta}_0^* = 0 \quad \Rightarrow \quad Q_0 [\dot{\Delta}_0 - \dot{\Delta}_0^*] = -R \quad (4. 41a)$$

$$-R E^T + Q_1 \dot{\Delta}_1 - Q_1 \dot{\Delta}_1^* = 0 \quad \Rightarrow \quad Q_1 [\dot{\Delta}_1 - \dot{\Delta}_1^*] = R E^T \quad (4. 41b)$$

$$-R E^T + Q_2 \dot{\Delta}_2 - Q_2 \dot{\Delta}_2^* = 0 \quad \Rightarrow \quad Q_2 [\dot{\Delta}_2 - \dot{\Delta}_2^*] = R E^T \quad (4. 41c)$$

$$R q_r^T + Q_3 \dot{\Delta}_3 - Q_3 \dot{\Delta}_3^* = 0 \quad \Rightarrow \quad Q_3 [\dot{\Delta}_3 - \dot{\Delta}_3^*] = -R q_r^T \quad (4. 41d)$$

$$R q_r^T + Q_4 \dot{\Delta}_4 - Q_4 \dot{\Delta}_4^* = 0 \quad \Rightarrow \quad Q_4 [\dot{\Delta}_4 - \dot{\Delta}_4^*] = -R q_r^T \quad (4. 41e)$$

$$R q_r^T + Q_5 \dot{\Delta}_5 - Q_5 \dot{\Delta}_5^* = 0 \quad \Rightarrow \quad Q_5 [\dot{\Delta}_5 - \dot{\Delta}_5^*] = -R q_r^T \quad (4. 41f)$$

Aynı amaçla aşağıdaki seçimler yapılabilir:

$$\Delta_0^* = -Q_0^* R \quad (4. 42a)$$

$$\Delta_1^* = Q_1^* R E^T \quad (4. 42b)$$

$$\Delta_2^* = Q_2^* R E^T \quad (4. 42c)$$

$$\Delta_3^* = -Q_3^* R q_r^T \quad (4. 42d)$$

$$\Delta_4^* = -Q_4^* R q_r^T \quad (4. 42e)$$

$$\Delta_5^* = -Q_5^* R q_r^T \quad (4. 42f)$$

Burada $Q_0^*, \dots, Q_5^*$ matrisleri yarı pozitif $N \times N$ sabit bilinen matrislerdir. (4. 39) denklemi şu şekle gelir:

$$\begin{aligned} \dot{V} = & -e^T Q e - 2R^T Q_0^* R - 2[R^T R] E^T Q_1^* E - 2[R^T R] E^T Q_2^* E - 2[R^T R] q_r^T Q_3^* q_r \\ & - 2[R^T R] q_r^T Q_4^* q_r - 2[R^T R] q_r^T Q_5^* q_r \end{aligned} \quad (4. 43)$$

Bu denklemin sağ tarafı negatif olmuş oldu. (4. 42) denklemlerinin zamana göre türevleri alınıp (4. 41) denklemlerinde yerlerine koyulursa adaptasyon kuralları olarak şu ifadeler elde edilir:

$$\dot{\Delta}_0 = -Q_0^{-1} R - Q_0^* R \quad (4. 44a)$$

$$\dot{\Delta}_1 = Q_1^{-1} [R E^T] + Q_1^* \frac{d}{dt} [R E^T] \quad (4. 44b)$$

$$\dot{\Delta}_2 = Q_2^{-1} [R E^T] + Q_2^* \frac{d}{dt} [R E^T] \quad (4. 44c)$$

$$\dot{\Delta}_3 = -Q_3^{-1} [R q_r^T] - Q_3^* \frac{d}{dt} [R q_r^T] \quad (4. 44d)$$

$$\dot{\Delta}_4 = -Q_4^{-1} [R q_r^T] - Q_4^* \frac{d}{dt} [R q_r^T] \quad (4. 44e)$$

$$\dot{\Delta}_5 = -Q_5^{-1} [R q_r^T] - Q_5^* \frac{d}{dt} [R q_r^T] \quad (4. 44f)$$

Robot model matrislerinin her örnekleme zamanı için değişiminin kontrolör kazançlarına göre çok çok küçük olduğu gözönüne alınarak şu kabul yapılır [13].

$$\frac{da_{ij}^*}{a_{ij}^*} \ll \frac{da_{ij}}{a_{ij}} \quad (4. 45)$$

Bu kabulde adaptasyon kuralında parametreleri bilinmeyen A^* , B^* , C^* robot model matrislerinin türevleri

$$\frac{da_{ij}^*}{dt} \cong 0 \quad (4. 46)$$

olur.

Bu kabule göre (4. 38) ifadeleri yardımı ile şunlar elde edilir:

$$\begin{aligned} \dot{\Delta}_1 &= [A^*]^{-1} \dot{K}_p, \quad \dot{\Delta}_2 = [A^*]^{-1} \dot{K}_v, \quad \dot{\Delta}_3 = -[A^*]^{-1} \dot{C} \\ \dot{\Delta}_4 &= -[A^*]^{-1} \dot{B}, \quad \dot{\Delta}_5 = -[A^*]^{-1} \dot{A}, \quad \dot{\Delta}_0 = -[A^*]^{-1} \dot{F} \end{aligned} \quad (4. 47)$$

$Q_0, \dots, Q_5$ ve $Q_0^*, \dots, Q_5^*$ pozitif matrisleri şu şekilde seçilebilir:

$$Q_0 = \frac{1}{\delta_1} A^* ; Q_1 = \frac{1}{\alpha_1} A^* ; Q_2 = \frac{1}{\beta_1} A^* ;$$

$$Q_3 = \frac{1}{\nu_1} A^* ; Q_4 = \frac{1}{\gamma_1} A^* ; Q_5 = \frac{1}{\lambda_1} A^* ;$$

$$Q_0^* = \delta_2 [A^*]^{-1} ; Q_1^* = \alpha_2 [A^*]^{-1} ;$$

$$Q_2^* = \beta_2 [A^*]^{-1} ; Q_3^* = \nu_2 [A^*]^{-1} ;$$

$$Q_4^* = \gamma_2 [A^*]^{-1}; \quad Q_5^* = \lambda_2 [A^*]^{-1}; \quad (4. 48)$$

Burada $[\delta_1, \alpha_1, \beta_1, v_1, \gamma_1, \lambda_1]$ pozitif adaptasyon sabitleridir. $[\delta_2, \alpha_2, \beta_2, v_2, \gamma_2, \lambda_2]$ sıfır veya pozitif adaptasyon sabitleridir.

Yukarıdaki seçimler yapılır ve (4. 44) denklemlerinde (4. 47) ve (4. 48) ifadeleri yerlerine koyulur ve gerekli düzenlemeler yapılırsa adaptasyon kuralları

$$\dot{F} = \delta_1 R + \delta_2 \dot{R} \quad (4. 49a)$$

$$\dot{K}_p = \alpha_1 [RE^T] + \alpha_2 \frac{d}{dt} [RE^T] \quad (4. 49b)$$

$$\dot{K}_v = \beta_1 [RE^T] + \beta_2 \frac{d}{dt} [RE^T] \quad (4. 49c)$$

$$\dot{C} = v_1 [Rq_r^T] + v_2 \frac{d}{dt} [Rq_r^T] \quad (4. 49d)$$

$$\dot{B} = \gamma_1 [Rq_r^T] + \gamma_2 \frac{d}{dt} [Rq_r^T] \quad (4. 49e)$$

$$\dot{A} = \lambda_1 [Rq_r^T] + \lambda_2 \frac{d}{dt} [Rq_r^T] \quad (4. 49f)$$

şeklinde elde edilir.

(4. 49) denklemlerin integralini alırsak adaptasyon kuralları

$$F(t) = F(0) + \delta_2 R(t) + \delta_1 \int_0^t R(t) dt \quad (4. 50a)$$

$$K_p(t) = K_p(0) + \alpha_2 R(t)E^T(t) + \alpha_1 \int_0^t R(t)E^T(t) dt \quad (4. 50b)$$

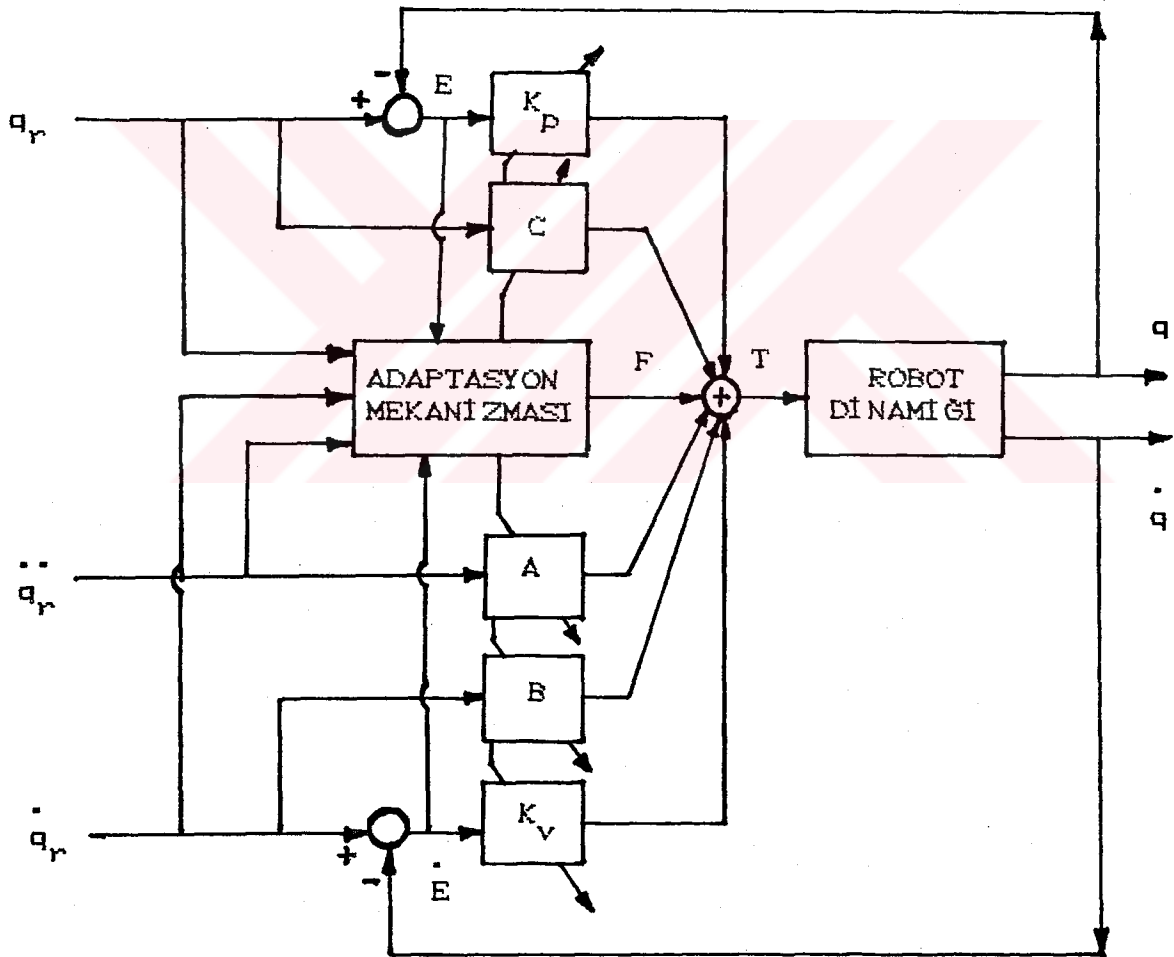
$$K_v(t) = K_v(0) + \beta_2 R(t)E^T(t) + \beta_1 \int_0^t R(t)E^T(t) dt \quad (4. 50c)$$

$$C(t) = C(0) + v_2 R(t) q_r^T(t) + v_1 \int_0^t R(t) q_r^T(t) dt \quad (4. 50d)$$

$$B(t) = B(0) + \gamma_2 R(t) q_r^T(t) + \gamma_1 \int_0^t R(t) q_r^T(t) dt \quad (4. 50e)$$

$$A(t) = A(0) + \lambda_2 R(t) q_r^T(t) + \lambda_1 \int_0^t R(t) q_r^T(t) dt \quad (4. 50f)$$

şekline girer.



Sekil 4.3 Adaptif kontrol diyagramı

$R(t) = P_2 E(t) + P_3 \dot{E}(t)$ açık ifadesi $F(t)$ yardımcı sinyal, ifadesinde yazılırsa

$$\begin{aligned} F(t) &= F(0) + \delta_2 [P_2 E(t) + P_3 \dot{E}(t)] + \delta_1 \int_0^t [P_2 E(t) + P_3 \dot{E}(t)] dt \\ &= F(0) + [\delta_2 P_2 + \delta_1 P_3] E(t) + [\delta_1 P_2] \int_0^t E(t) dt + [\delta_2 P_3] \dot{E} \end{aligned}$$

(4. 51)

elde edilir.

Bu ifadeyi (4. 21) toplam kontrol kuralında yerine koyarsak toplam kontrol kuralı

$$\begin{aligned} T(t) &= F(0) + K_P^* E(t) + K_I^* \int_0^t E(t) dt + K_D^* \dot{E} \\ &\quad + C(t) q_r(t) + B(t) \dot{q}_r(t) + A(t) \ddot{q}_r(t) \end{aligned}$$

(4. 52)

şekline dönüşür [8].

Burada

$$K_P^* = K_p + \delta_2 P_2 + \delta_1 P_3 \quad (4. 53)$$

$$K_I^* = \delta_1 P_2 \quad (4. 54)$$

$$K_D^* = K_v + \delta_2 P_3 \quad (4. 55)$$

şeklinde adaptif PID kontrolör matrisleridir.

(4. 50) denklemleri bilgisayarla şu şekilde hesaplanır:

$$K(i) = K(i-1) + \mu_2 [R(i) v(i)] + \mu_1 \frac{T_s}{2} [R(i) v(i) + R(i-1) v(i-1)]$$

(4. 56)

Burada T_s örnekleme periyodunu v sinyali ve K adaptasyon kazançlarını temsil etmektedir.

BÖLÜM 5

SİMULASYON ÇALIŞMALARI

Programın çalışması sırasında önce robot parametreleri ve adaptif kontrol için gerekli adaptasyon sabitleri bir dosyadan okunduktan sonra bir çevrim içersinde eklemlerin izlemesi istenilen referans yörüngelere ait değerler yörünge alt programından okunmaktadır. Sonra eklem değişkenleri ile referans yörüngeler arasındaki hatalar hesaplanıp adaptasyon mekanizmasında kontrolör matrisleri ve eklemler için tork (veya kuvvet) değerleri hesaplanır bu değerler Bölüm 3 de elde edilen (3. 43) denkleminde yerine koyulur ve bu denklem Runge Kutta metodu ile çözülür ve eklem değişkenleri hesaplanır ve bir dosyaya atanır. Programın akış şeması şekil (5.1) deki gibidir. Simülasyon programının bir copyası Ek A de verilmiştir.

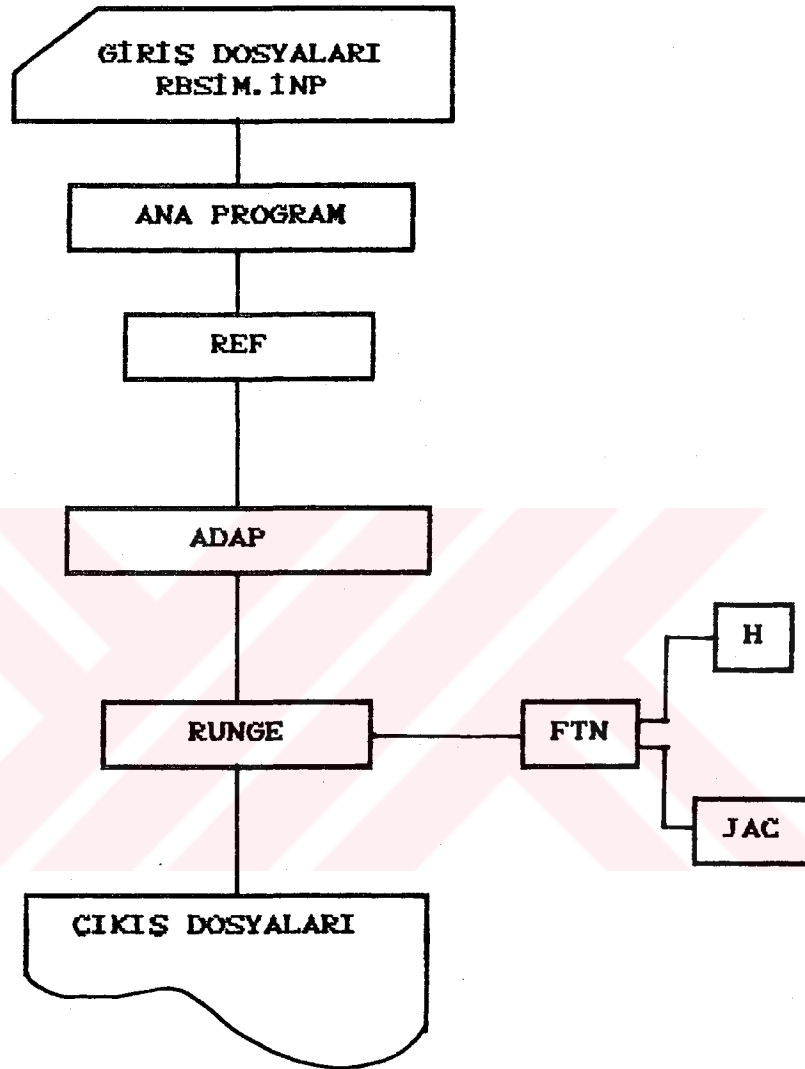
ANA PROGRAM: Robot parametrelerinin ve adaptasyon sabitlerinin okunup alt programların çağrıldığı kısımdır.

RBSIM.INP: Robot parametrelerinin ve adaptasyon sabitlerinin okunduğu dosyadır.

REF: Referans yörüngelerin okunduğu alt program.

ADAP: Adaptasyon mekanizmasına göre kontrol sinyalleri üreten alt program.

RUNGE: Sistem denklemlerinin üretilen kontrol sinyaline göre Runge Kutta metodu ile çözüldüğü alt program.



Sekil 5.1 Simülasyon programının akış şeması

FTN: Lagrange-Euler formasyonuna göre robot dinamik modelinin nümerik olarak hesaplandığı alt programdır. Burada (3. 43) denklemdeki matris veya vektörleri hesaplayan H, JAC, altprogramları çağrılmakta ve eklem değişkenlerinin ikinci türevleri hesaplanmaktadır. Burada matris çarpımlarının ve değişik matematiksel işlemlerin yapıldığı birden fazla yerde birden fazla çağrılan alt programlar da bulunmaktadır.

Simülasyon çalışmalarındaki amaç adaptif PID kontrolörün performansının araştırılmasıdır. Bu nedenle EK B de verilen üç serbestlik dereceli bir robot kolunun dinamik simülasyonu yapıldı.

Bütün senaryolar için simülasyon zamanı 1 sn olarak alındı. (4. 50) denklemlerindeki $F(0)$, $K_p(0)$, $K_v(0)$, $C(0)$, $B(0)$ ve $A(0)$ başlangıç değerleri sıfır alınmıştır.

(4. 26) denklemi ile verilen referans modelde

$$\zeta_1 = 1.0 \quad i=1, \dots, N$$

$$\omega_1 = 10.0 \text{ rd/sn} \quad i=1, \dots, N$$

olarak alınmıştır.

Sistemden örnek alma ve adaptasyon parametrelerinin hesaplanması aynı zamanda yapılmaktadır. Örneklem zamanı $T = 0.002$ sn. dir.

5.1 Simülasyon Sonuçları

Senaryo 1: Burada robotun 10 Kg. lık bir yükü taşıyarak eklem değişkenlerinin ve eklem değişkenlerinin hızlarının q_{1r} , q_{2r} ve q_{3r} ile $\dot{q}_{1r}$, $\dot{q}_{2r}$ ve $\dot{q}_{3r}$ yörüngelerini

izlemesini sağlamak için adaptasyon sabitleri bulundu.

$$[\alpha_1 \beta_1 \gamma_1 \lambda_1 \nu_1 \delta_1] = [0.5 \ 0.5 \ 2.0 \ 2.0 \ 2.0 \ 0.1]$$

$$[\alpha_2 \beta_2 \gamma_2 \lambda_2 \nu_2 \delta_2] = [0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 9.5]$$

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

$$Q_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} ; \quad Q_2 = \begin{bmatrix} 800 & 0 & 0 \\ 0 & 1300 & 0 \\ 0 & 0 & 10000 \end{bmatrix}$$

Elde edilen bu sabitlere göre adaptif kontrol algoritması ile elde edilen sistem cevabı ve referans eğrileri şekil 5.2 te verilmiştir. Şekil 5.3 de ise konum hatalarının karelerinin toplamı ile hız hatalarının karelerinin toplamı bir performans ölçütü olarak verilmiştir. Elde edilen adaptasyon sabitleri için sistem cevabı eğrilerinden de görüldüğü gibi uygun parametre takımıdır denilebilir.

Elde edilen bu parametre takımı ile çeşitli senaryolar uygulanarak simülasyon çalışmaları yapılmış ve adaptif PID kontrolörün performansı incelenmiştir.

$$q_{1r}(t) = q_{2r}(t) = 45 [1 + 6\exp(-t/0.3) - 8\exp(-t/0.4)]$$

$$q_{3r}(t) = 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 45 [-20\exp(-t/0.3) + 20\exp(-t/0.4)]$$

$$\dot{q}_{3r}(t) = 0.0$$

$$\ddot{q}_{1r}(t) = 45 [(20/0.3)\exp(-t/0.3) - (20/0.4)\exp(-t/0.4)]$$

$$\ddot{q}_{2r}(t) = \ddot{q}_{1r}(t)$$

$$\ddot{q}_{3r}(t) = 0.0$$

Senaryo 2: Referans yörüngeler, ve bunlardan türetilen hız ve ivmeler senaryo 1 deki gibi alınarak taşınan yükün 0.3 sn. sonunda 10 Kg dan 0 Kg a düştüğü (yani yükün bırakıldığı) düşünülmüştür. Yapılan simülasyon sonucunda adaptif PID kontrolörün davranışının ani yük değişimine bile çok iyi bir şekilde cevap verdiği görülmüştür. Şekil 5.4 de sistem cevabı ve referans eğrileri verilmiştir. Şekil 5.5 de konum ve hız hatalarının karelerinin toplamalarının zamana göre değişimleri bir performans ölçütü olarak verilmiştir.

Senaryo 3:

$$q_{1r}(t) = q_{2r}(t) = 45 [1 + 6\exp(-t/0.6) + 8\exp(-t/0.8)]$$

$$q_{3r}(t) = 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 45 [-10\exp(-t/0.6) + 10\exp(-t/0.8)]$$

$$\dot{q}_{3r}(t) = 0.0$$

$$\ddot{q}_{1r}(t) = 45 [(10/0.6)\exp(-t/0.6) - (10/0.8)\exp(-t/0.8)]$$

$$\ddot{q}_{2r}(t) = \ddot{q}_{1r}(t)$$

$$\ddot{q}_{3r}(t) = 0.0$$

Yukarıdaki gibi sistemin zaman sabitleri senaryo 1 ve senaryo 2 deki zaman sabitlerine göre iki misli hızlandırılarak adaptif PID kontrolörün performansı incelenmiştir. Burada taşınan yük yine 10 Kg olarak alınmıştır ve adaptif PID kontrolörün davranışının iyi olduğu gözlenmiştir.

Şekil 5.6 da sistem cevabı ve referans eğrileri görülmektedir. Şekil 5.7 de konum ve hız hatalarının karelerinin toplamalarının zamana göre değişim grafikleri bir performans ölçütü olarak verilmiştir.

Senaryo 4:

$$q_{1r}(t) = q_{2r}(t) = 45t^3 - 135t^2 + 30t - 45$$

$$q_{3r}(t) = 0.3t^3 - 0.5t + 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 135t^2 - 270t + 30$$

$$\dot{q}_{3r}(t) = 0.9t^2 - 0.5$$

$$\ddot{q}_{1r}(t) = \ddot{q}_{2r}(t) = 270t - 270$$

$$\ddot{q}_{3r}(t) = 1.8t$$

Burada adaptif PID kontrolörün performansı, yukarıdaki kübik referans yörüngeler ve bunlardan türetilen hız ve ivme referanslarıyla değişik referans yörüngeleri için incelenirken, robot kolunun bir yük taşımadığı varsayılmıştır.

Simülasyon sonucunda adaptif PID kontrolörün davranışının kübik yörüngeler için çok daha iyi olduğu görülmüştür.

Şekil 5.8 de sistem cevabı ve referans eğrileri, şekil 5.9 da ise konum ve hız hatalarının karelerinin toplamı verilmiştir.

Senaryo 5: Burada robot kolunun uç noktasına $F = [-4 \ -5 \ -6]$ N luk sabit bir kuvvet uygulandığı ve bir yük taşımadığı düşünülmüştür. Referans yörüngeler şu şekildedir:

$$q_{1r}(t) = q_{2r}(t) = 45t^3 - 135t^2 + 30t - 45$$

$$q_{3r}(t) = 0.3t^3 - 0.5t + 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 135t^2 - 270t + 30$$

$$\dot{q}_{3r}(t) = 0.9t^2 - 0.5$$

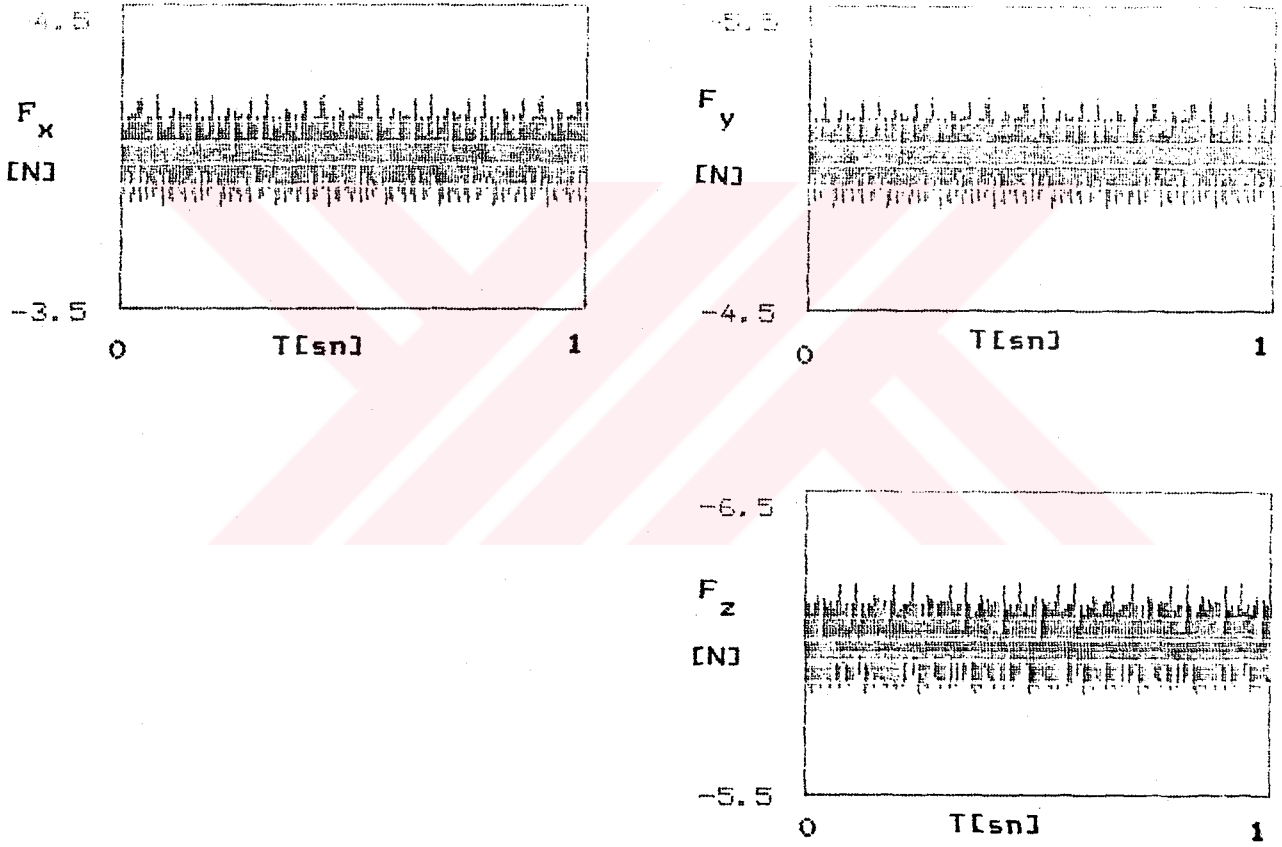
$$\ddot{q}_{1r}(t) = \ddot{q}_{2r}(t) = 270t - 270$$

$$\ddot{q}_{3r}(t) = 1.8t$$

Adaptif PID kontrolörün davranışının robot kolunun uç noktasına sabit bir kuvvet etkimesi durumunda da iyi olduğu simülasyon sonucunda görülmüştür.

Şekil 5.10 de sistem cevabı ve referans eğrileri verilmiştir. Şekil 5.11 de ise konum ve hız hatalarının karelerinin toplamalarının zamana göre değişim grafikleri performans ölçütü olarak verilmiştir.

Senaryo 6: Burada uç noktaya etkiyen kuvvetin her T örnekleme zamanı içerisinde değiştiği varsayılarak adaptif PID kontrolörün performansı incelenmiştir. Uç noktaya etkiyen kuvvetin zamanla değişimi şekil 5.12 de verilmiştir.



Sekil 5.12 Altıncı senaryo için uç noktaya etkiyen kuvvet bileşenleri

Burada konum, hız ve ivme referansları şu şekildedir:

$$q_{1r}(t) = q_{2r}(t) = 45t^3 - 135t^2 + 30t - 45$$

$$q_{3r}(t) = 0.3t^3 - 0.5t + 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 135t^2 - 270t + 30$$

$$\dot{q}_{3r}(t) = 0.9t^2 - 0.5$$

$$\ddot{q}_{1r}(t) = \ddot{q}_{2r}(t) = 270t - 270$$

$$\ddot{q}_{3r}(t) = 1.8t$$

Simülasyon sonucunda adaptif PID kontrolörün davranışının uç noktaya etkiyen kuvvetteki değişimlere rağmen iyi olduğu görülmüştür.

Sekil 5.13 te sistem cevabı ile referans eğrileri verilmiştir. Sekil 5.14 te konum ve hız hatalarının karelerinin toplamalarının zamana göre değişimleri görülmektedir.

Senaryo 7: Burada eklem değişkenlerinde ölçme hataları yapıldığı düşünülerek adaptif PID kontrolörün performansı incelenmiştir.

Döner eklemler için ± 0.001 rad, öteleme yapan eklemler için ± 0.001 m lik hata yapıldığı varsayılmıştır [14]. Robotun taşıdığı yük 10 Kg olarak alınmıştır.

Referans yörüngeler olarak şunlar alınmıştır:

$$q_{1r}(t) = q_{2r}(t) = 45t^3 - 135t^2 + 30t - 45$$

$$q_{3r}(t) = 0.3t^3 - 0.5t + 0.5$$

$$\dot{q}_{1r}(t) = \dot{q}_{2r}(t) = 135t^2 - 270t + 30$$

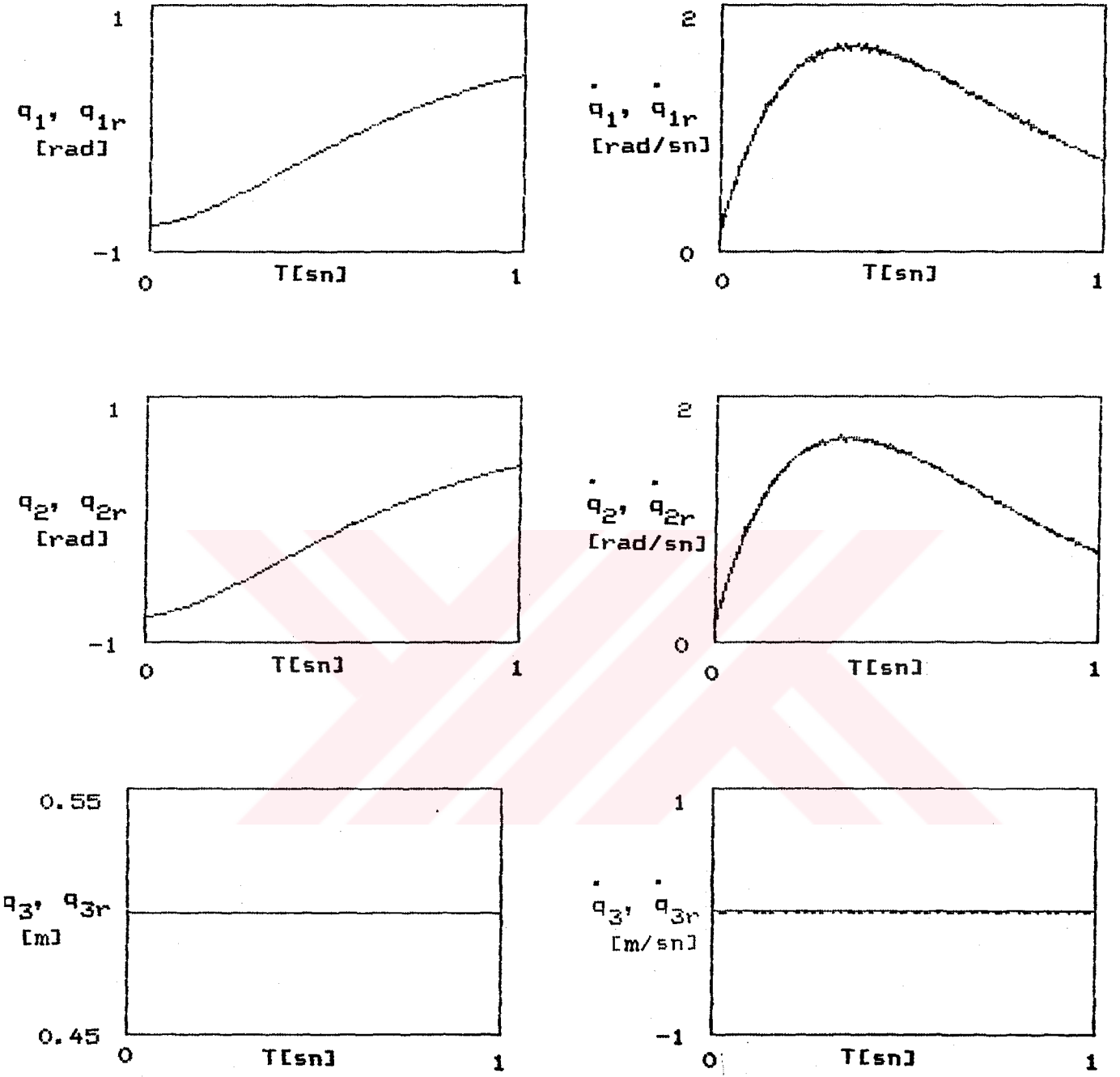
$$\dot{q}_{3r}(t) = 0.9t^2 - 0.5$$

$$\ddot{q}_{1r}(t) = \ddot{q}_{2r}(t) = 270t - 270$$

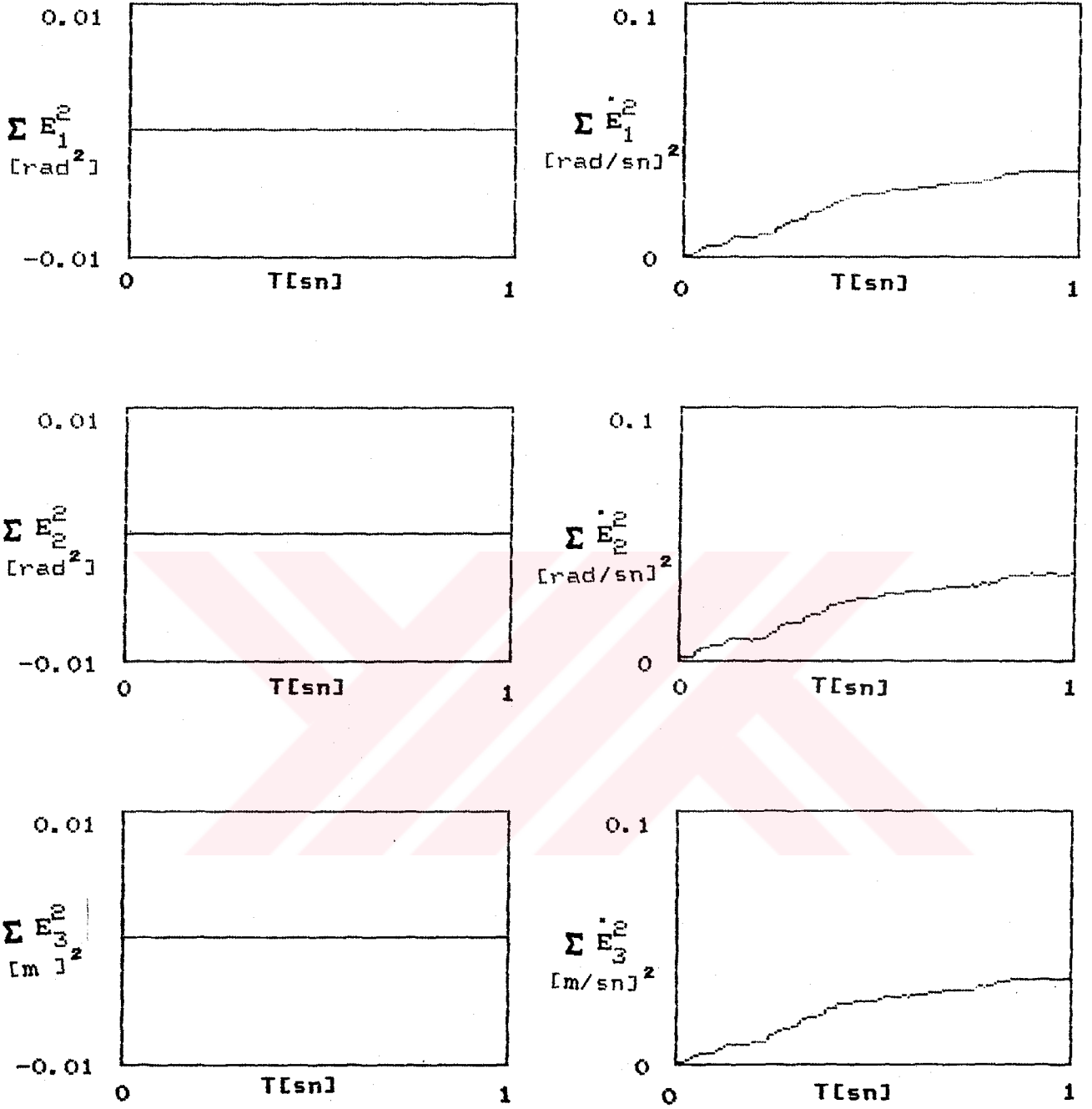
$$\ddot{q}_{3r}(t) = 1.8t$$

Sonuçta adaptif PID kontrolörün performansının ölçme hatalarına karşı duyarlı olmadığı görülmüştür.

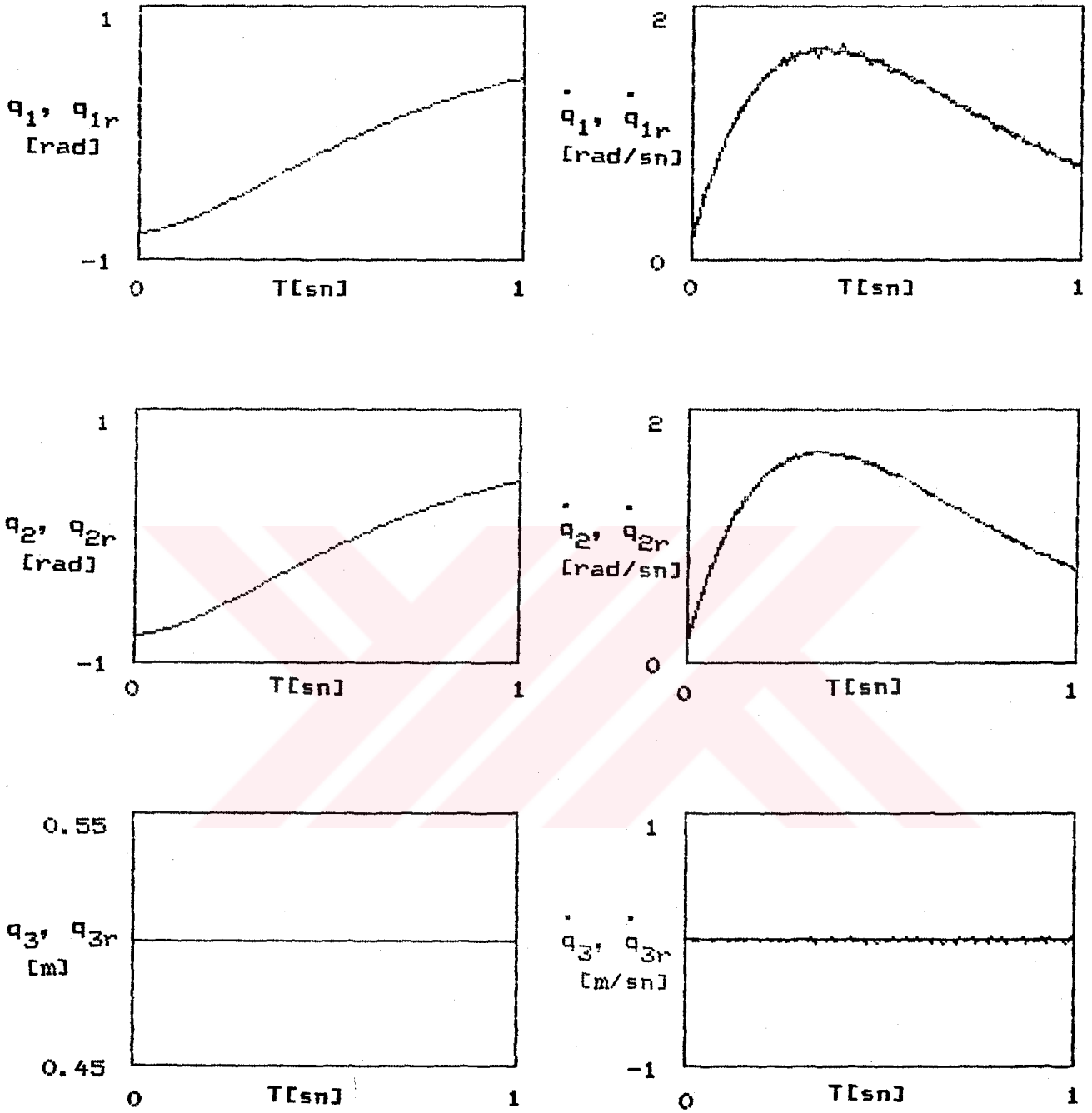
Şekil 5.15 te sistem cevabı ve referans egrileri verilmiştir. Şekil 5.16 da performans ölçütü olarak konum ve hız hatalarının karelerinin toplamaları görülmektedir.



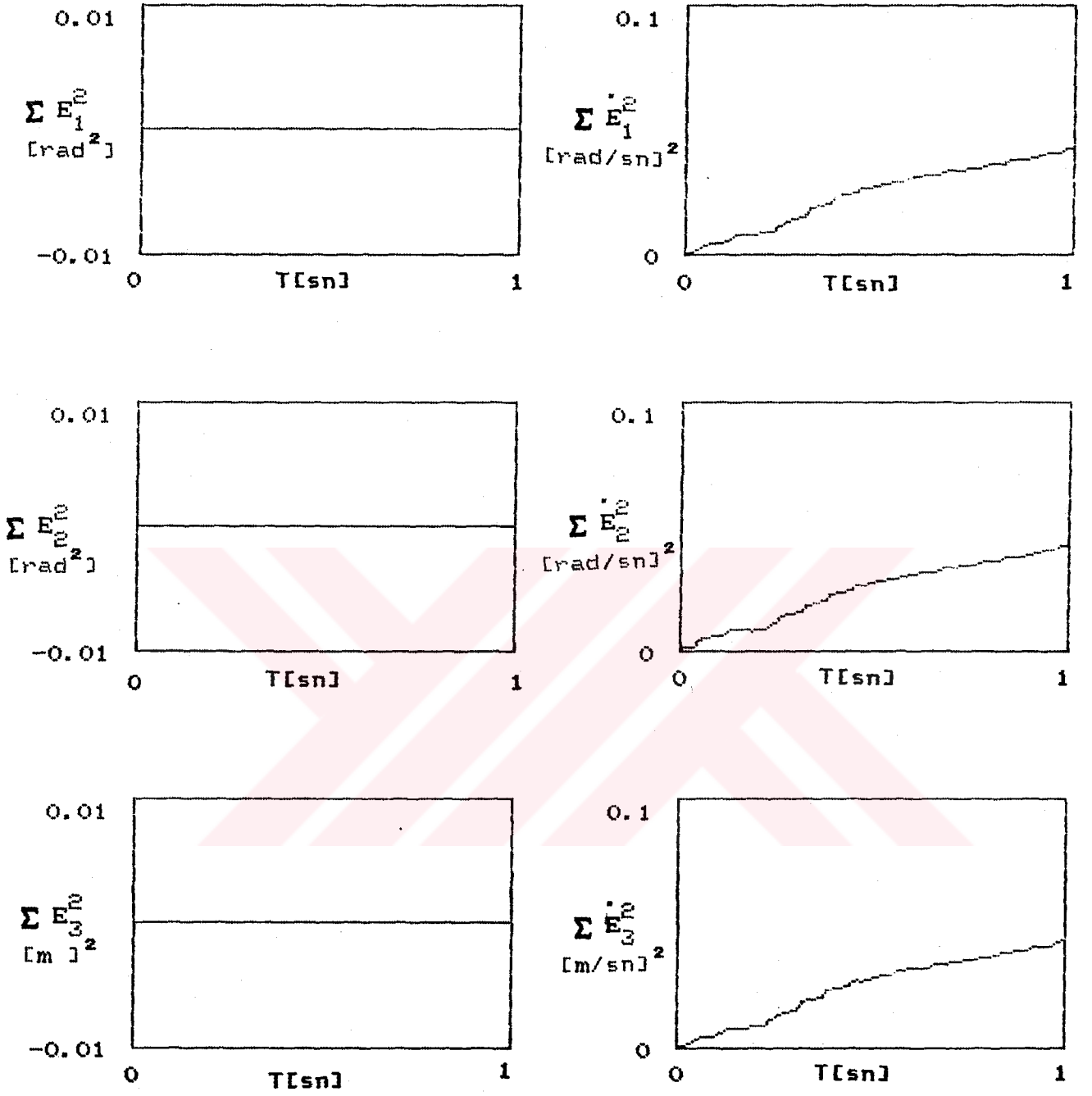
Sekil 5.2 Senaryo 1 için referans ve sistem cevabı eğrileri



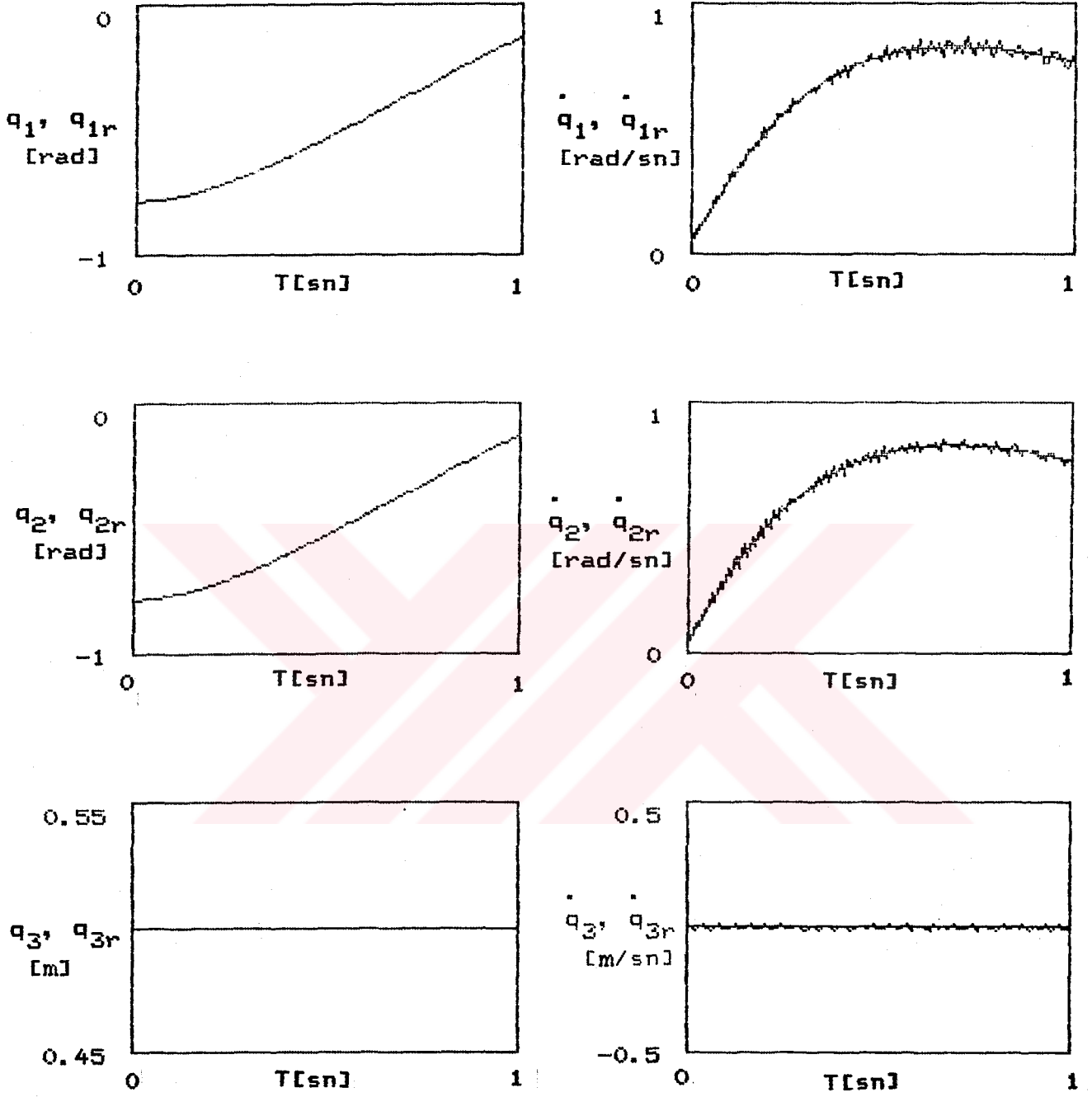
Sekil 5.3 Senaryo 1 için konum ve hız hatalarının karelerinin toplamı



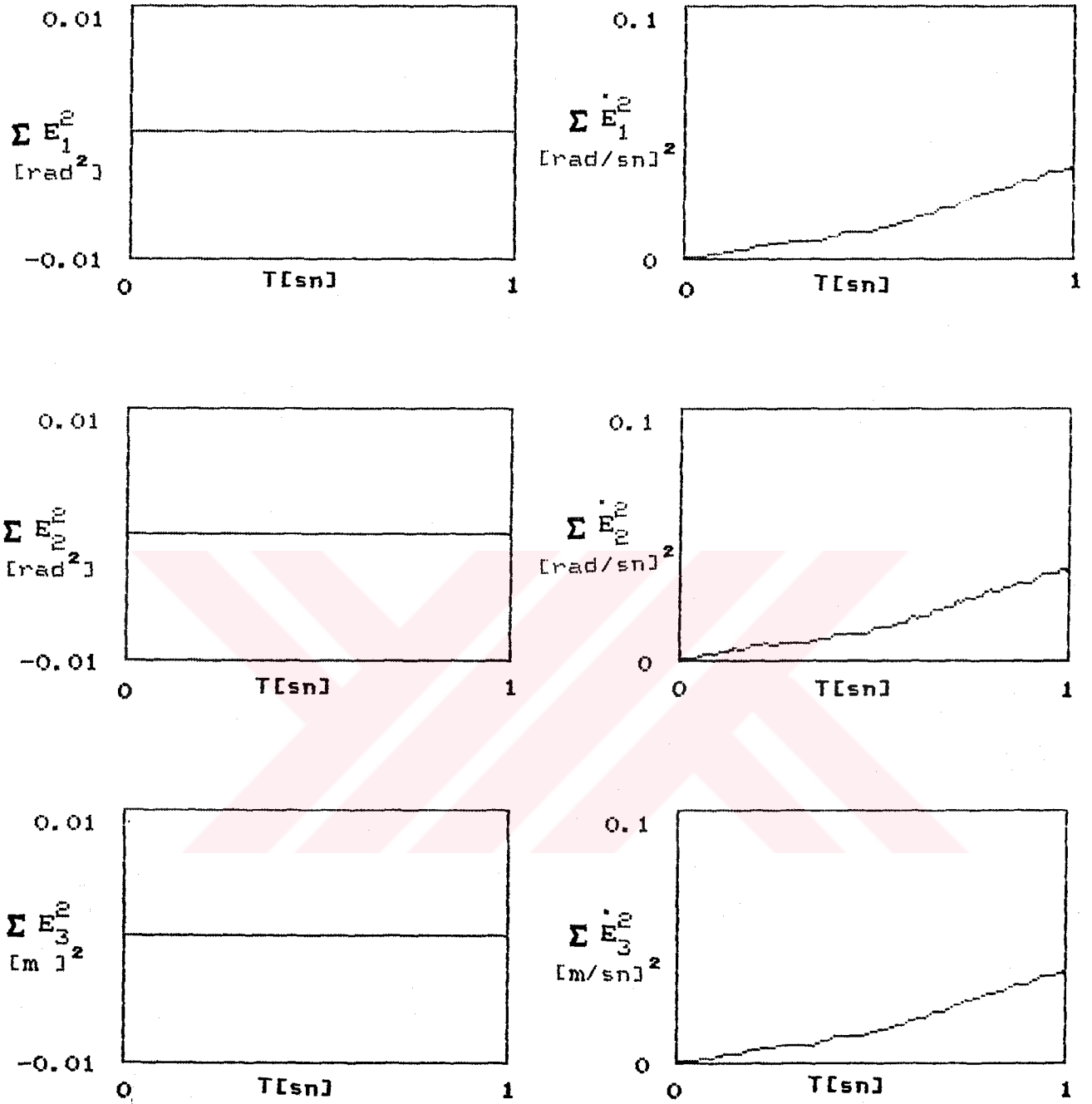
Sekil 5.4 Senaryo 2 için referans ve sistem cevabı eğrileri



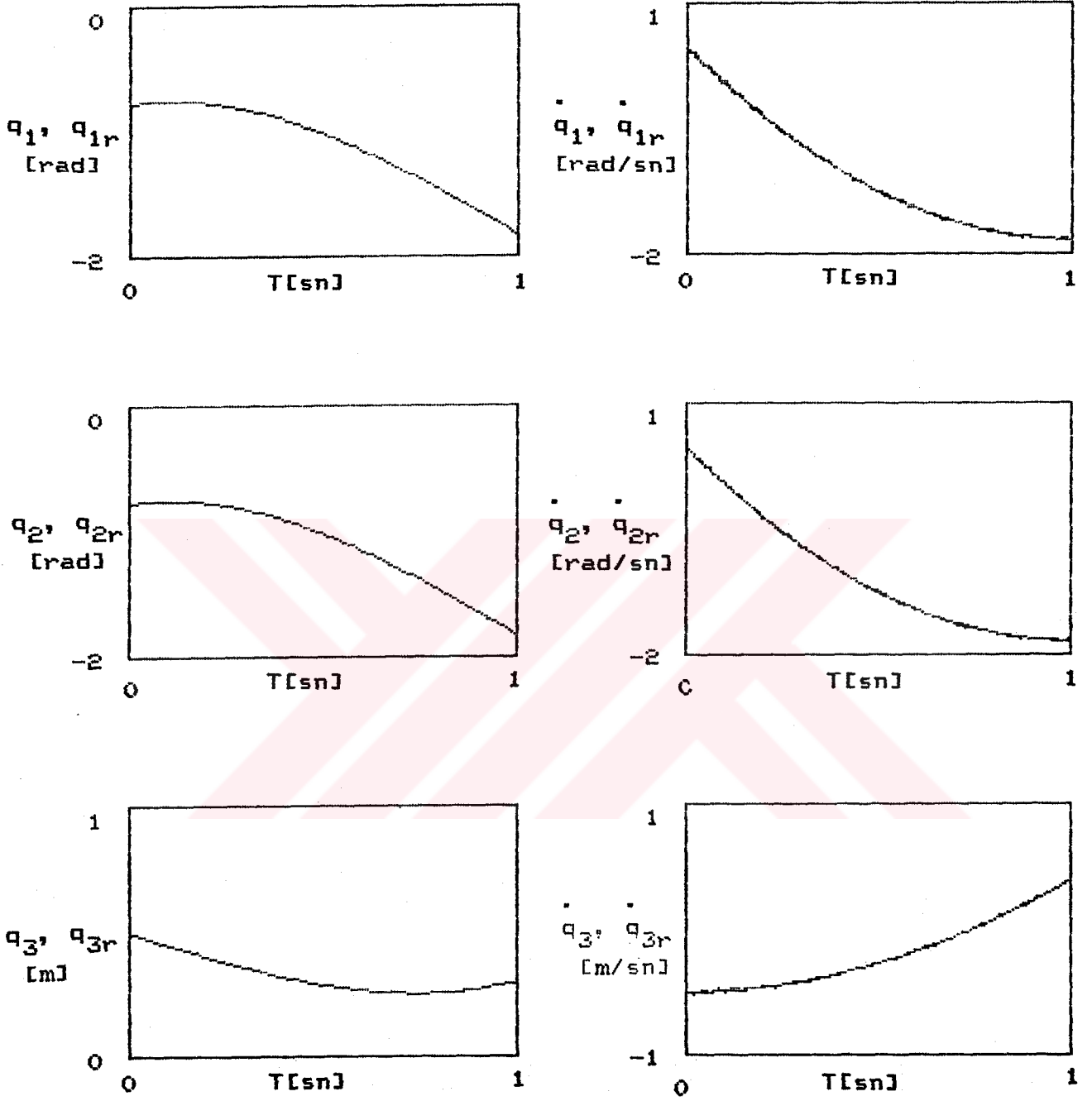
Şekil 5.5 Senaryo 2 için konum ve hız hatalarının karelerinin toplamı



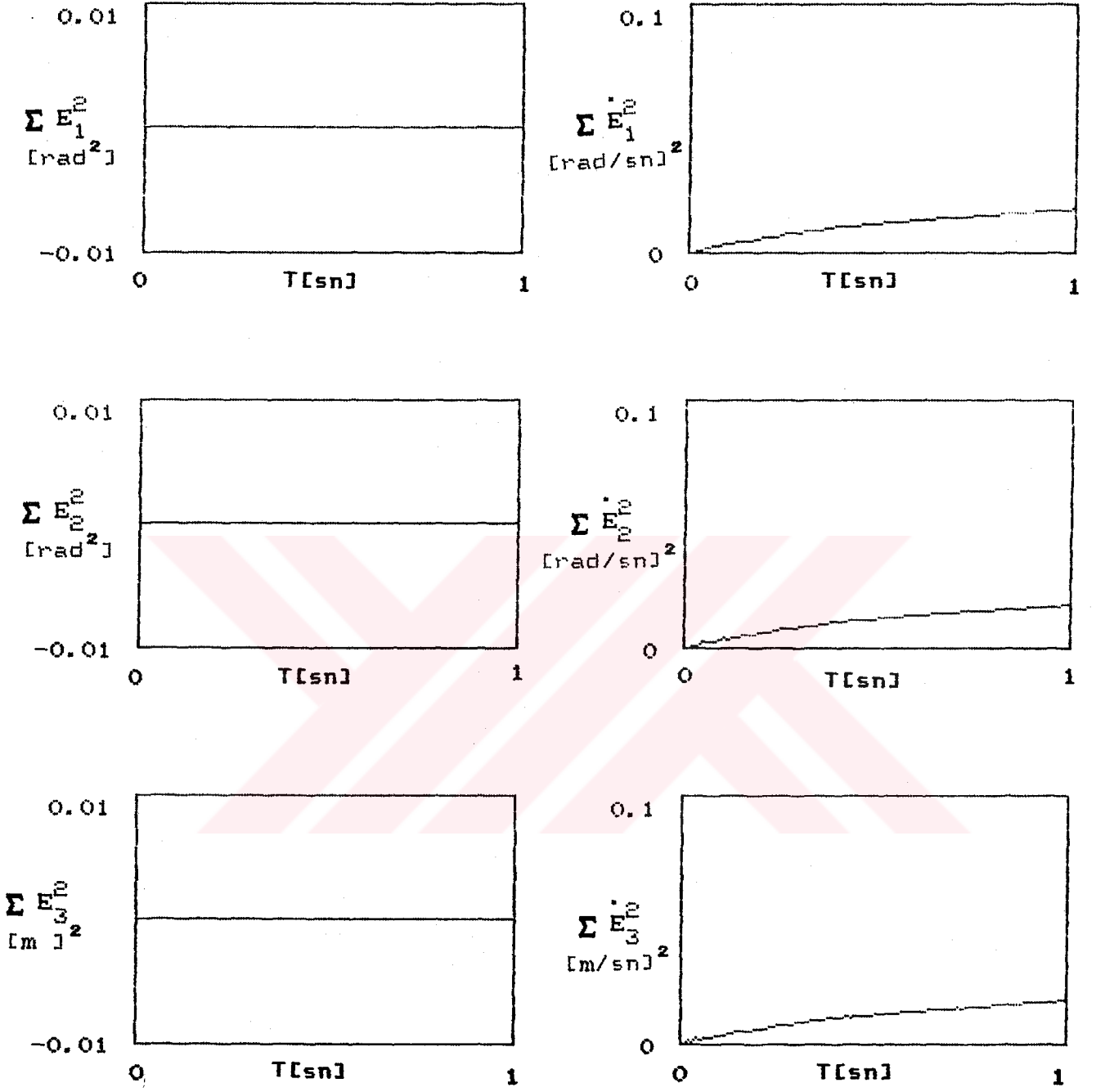
Sekil 5.6 Senaryo 3 için referans ve sistem cevabı eğrileri



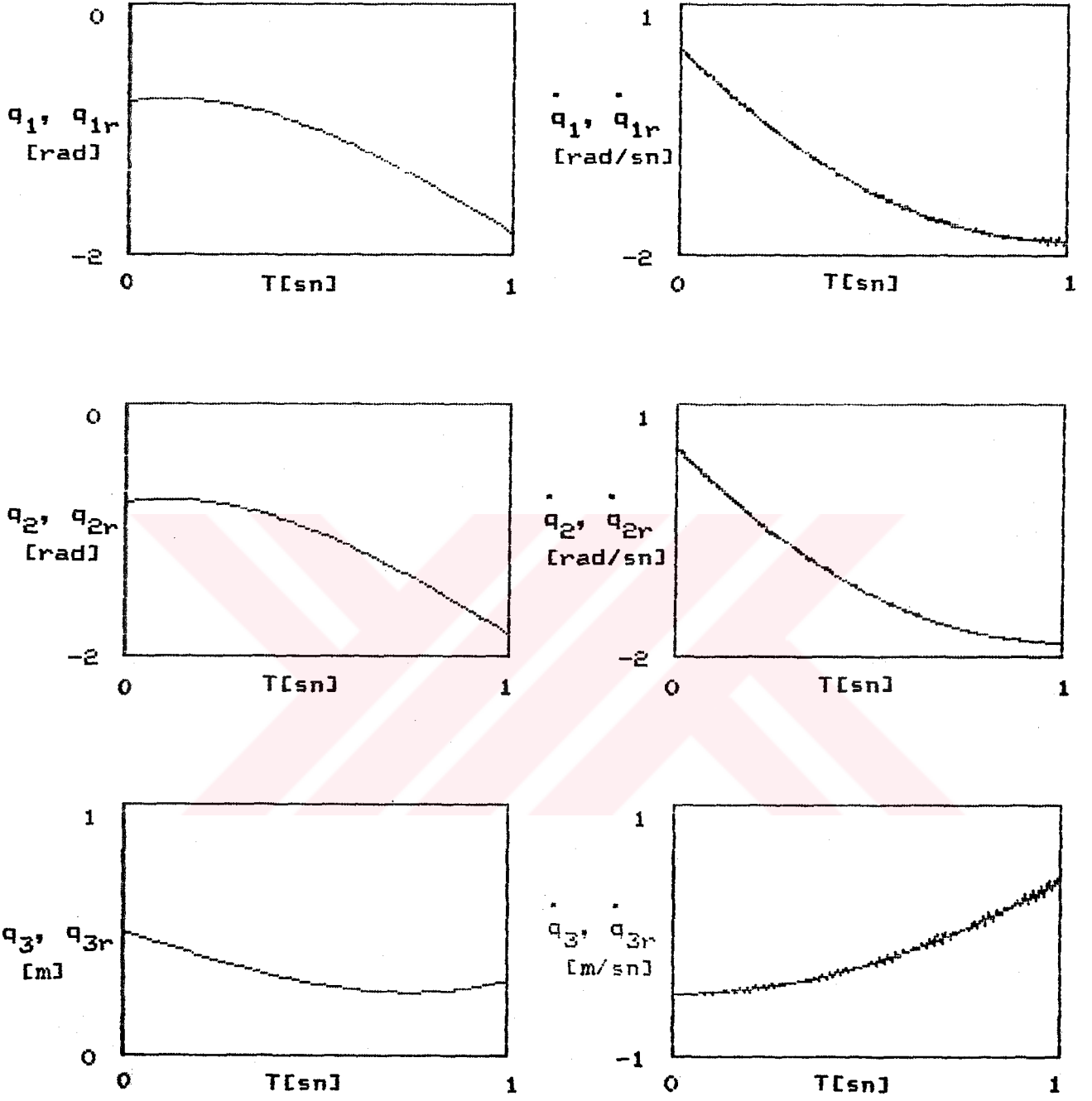
Sekil 5.7 Senaryo 3 için konum ve hız hatalarının karelerinin toplamları



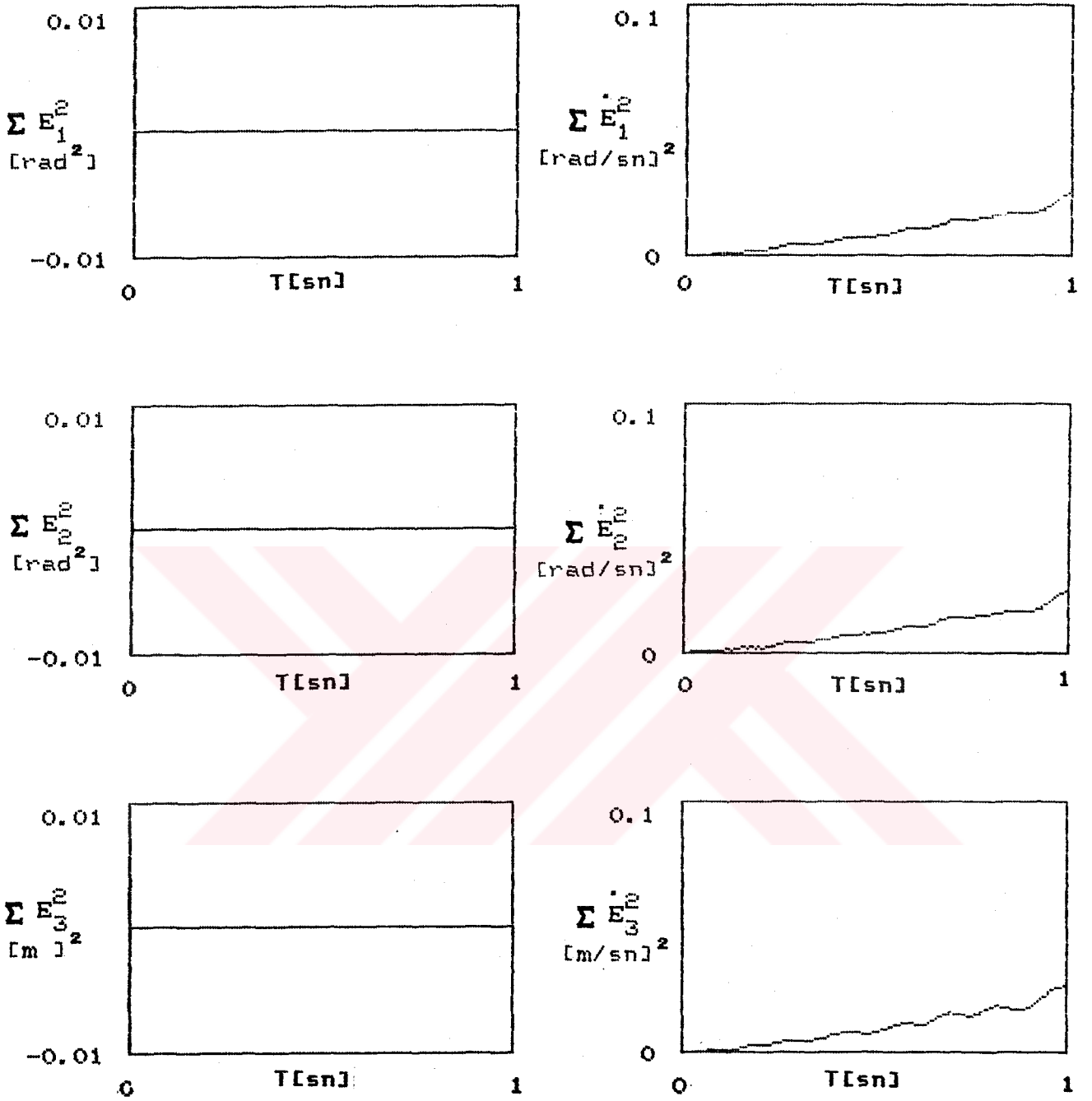
Sekil 5.8 Senaryo 4 için referans ve sistem cevabı eğrileri



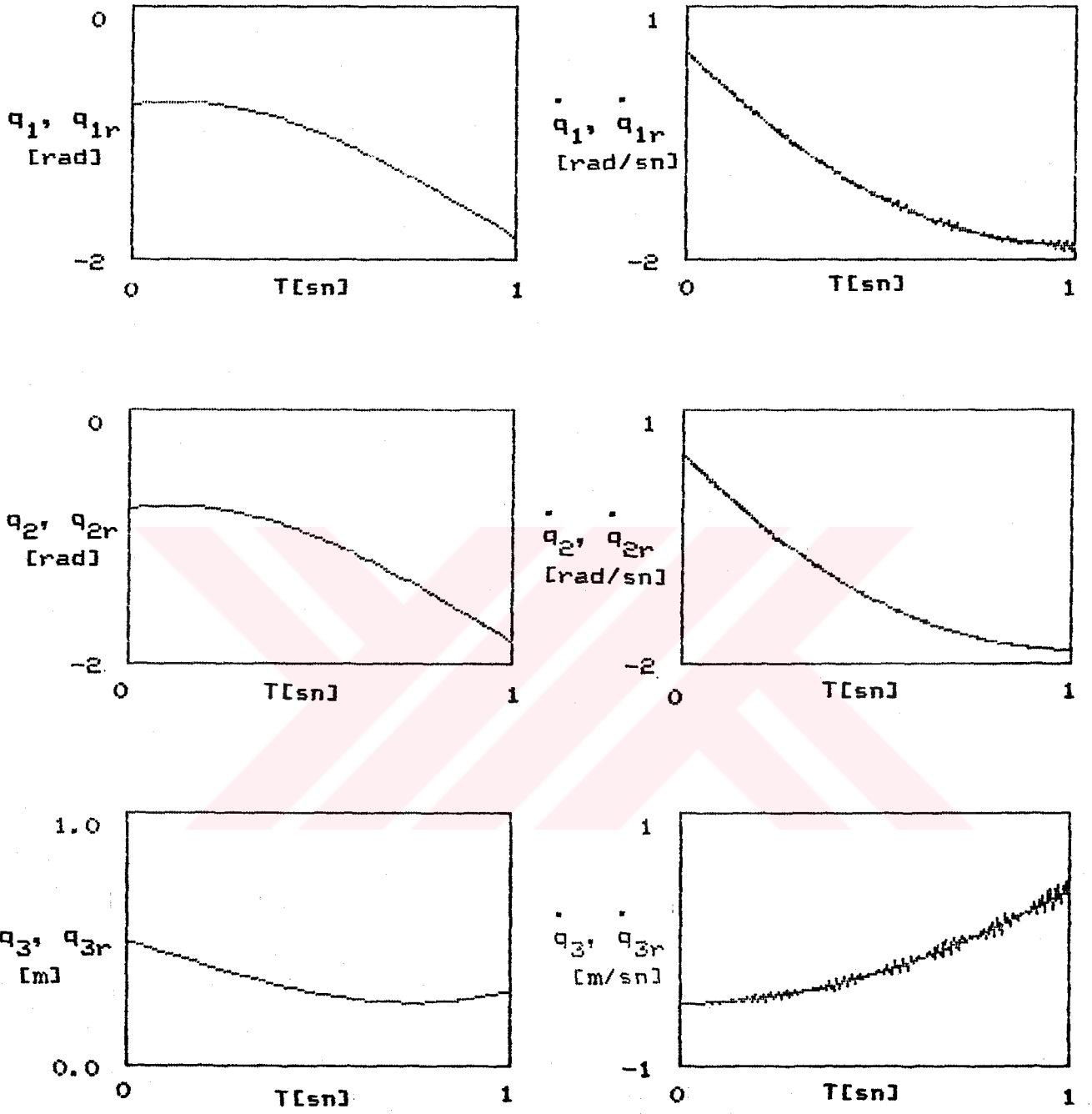
Şekil 5.9 Senaryo 4 için konum ve hız hatalarının karelerinin toplamaları



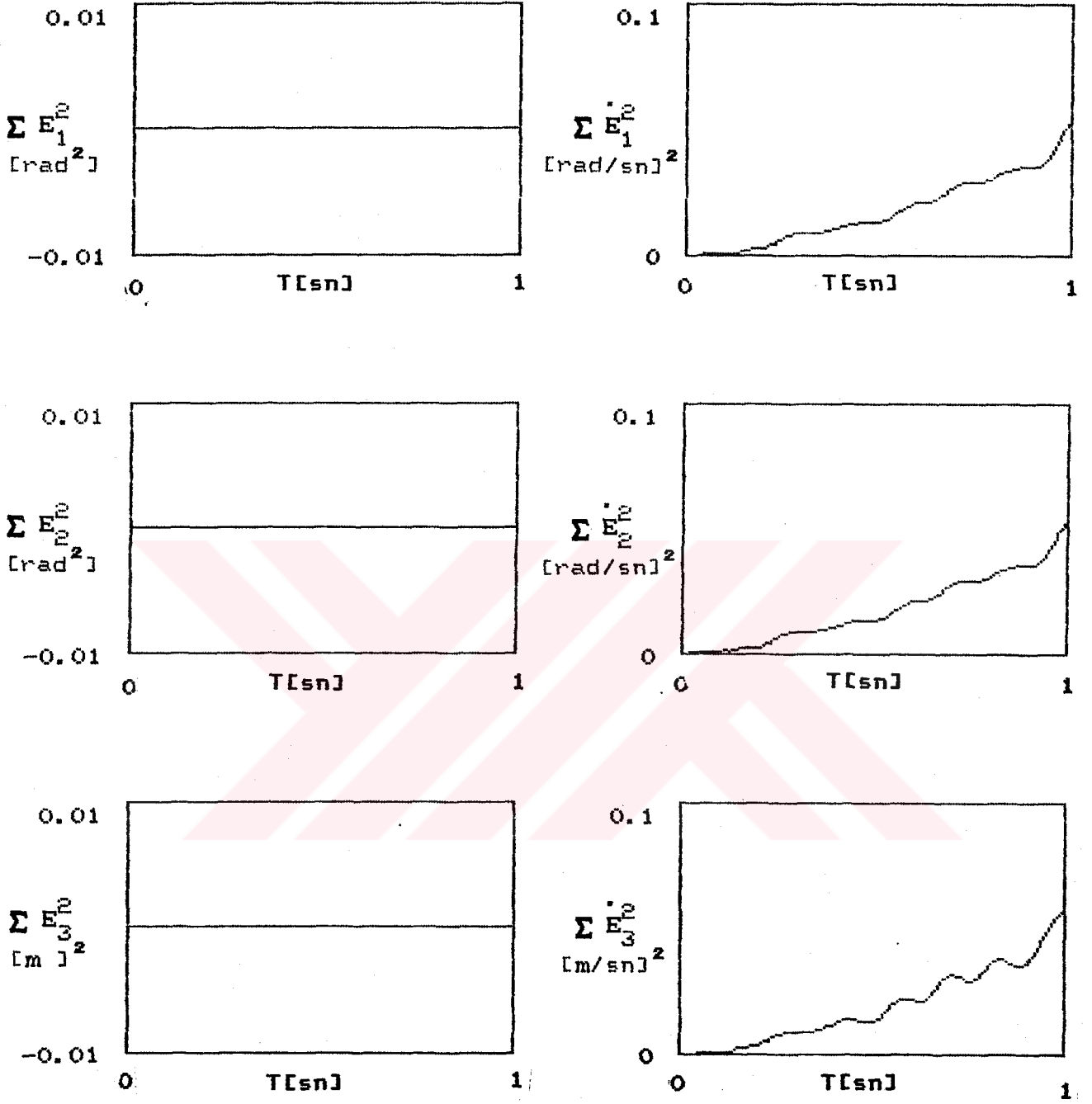
Şekil 5.10 Senaryo 5 için referans ve sistem cevabı eğrileri



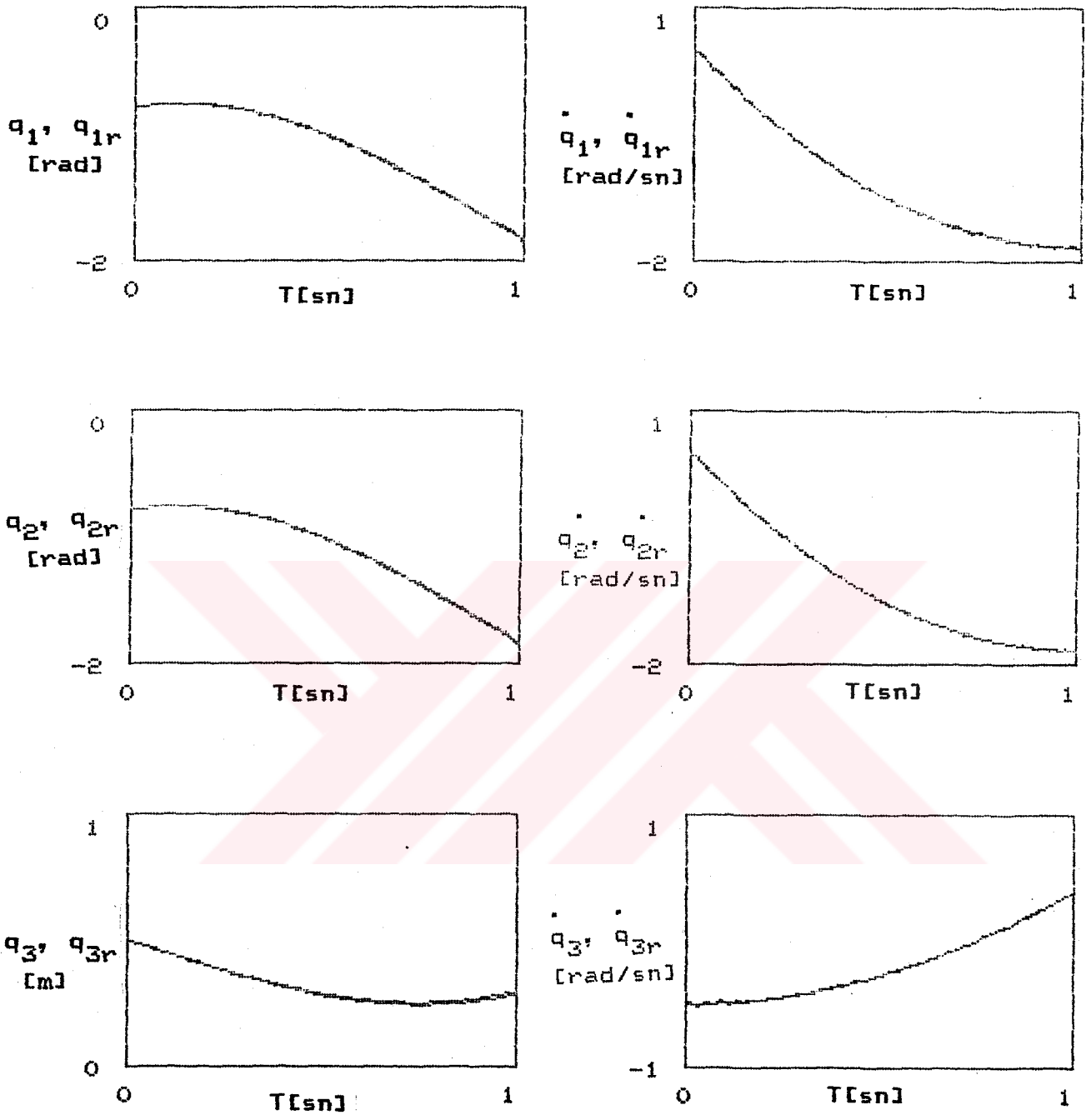
Sekil 5.11 Senaryo 5 için konum ve hız hatalarının karelerinin toplamı



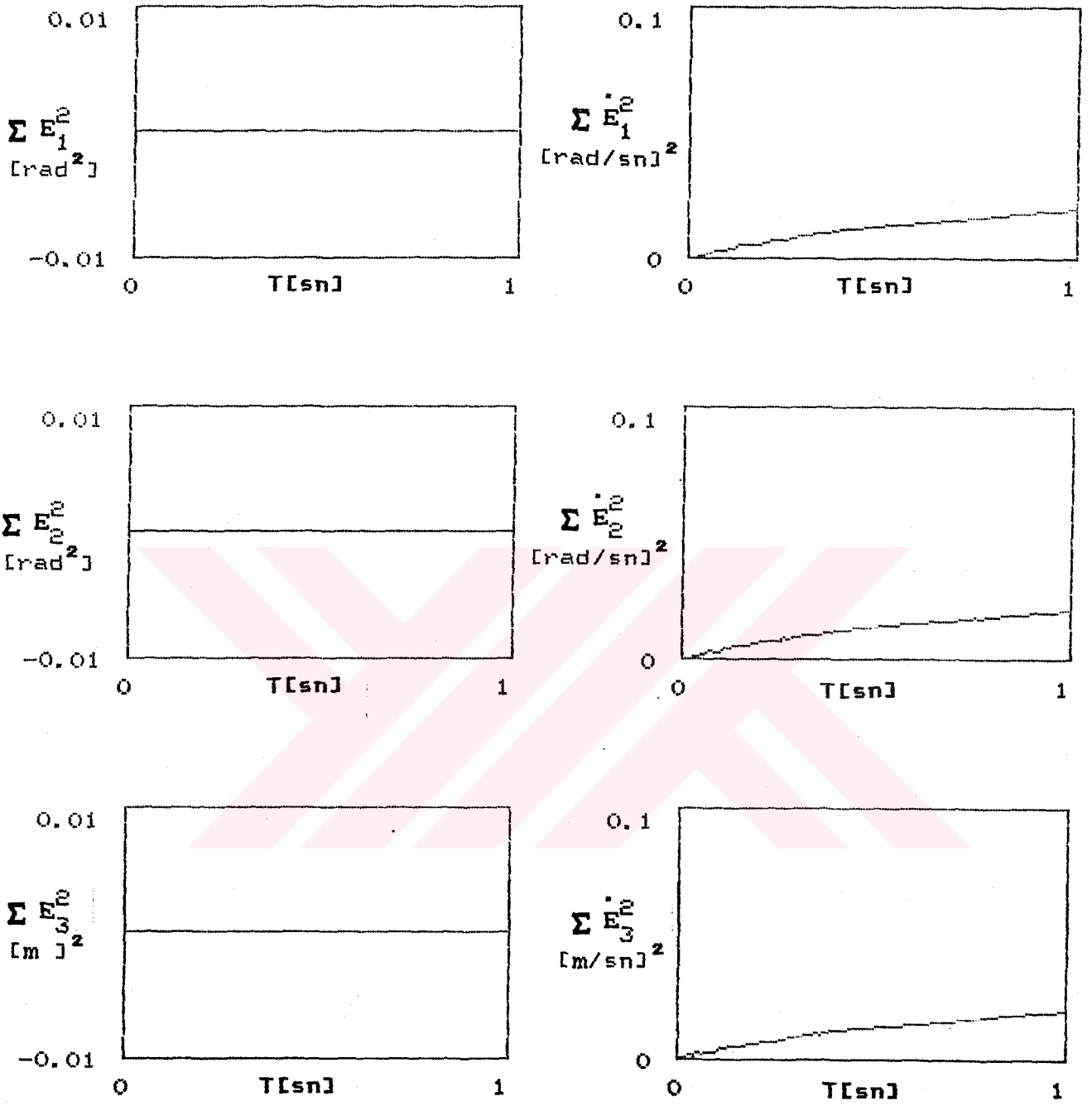
Sekil 5.13 Senaryo 6 için referans ve sistem cevabı
egrileri



Şekil 5.14 Senaryo 8 için konum ve hız hatalarının karelerinin toplamaları



Sekil 5.15 Senaryo 7 için referans ve sistem cevabı egrileri



Sekil 5.16 Senaryo 7 için konum ve hız hatalarının
karelerinin toplamları

SONUÇLAR VE ÖNERİLER

Adaptif PID kontrolör ile yörünge izleme probleminde simülasyon çalışmaları hataların karelerinin toplamı performans ölçütü alınarak şu şekilde özetlenebilir:

1- Kübik yörünge durumunda adaptif kontrolör en iyi sonucu vermiştir.

2- Eksponansiyel yörünge için daha iyi izlenebilmesi için geri besleme koluna $K_I \frac{1}{s}$ şeklinde integral kontrolör katılmasının daha iyi sonuç vereceği düşünülmektedir.

3- Eksponansiyel olarak verilen konum ve hız referans yörüngelerinde zaman sabitleri iki misli alınarak manipülatörün hızlandırılması hızlarda küçük bir takım değişimlere neden olsada yörüngelerin izlenmesi mümkün olmaktadır.

4- Yükün ani olarak değiştirilmesi durumunda kısa bir adaptasyon süresinden sonra istenilen yörünge üzerinde kalınmakta ve burada da adaptif kontrolörün yük değişimlerine karşı duyarlı olmadığı anlaşılmaktadır.

5- Üç noktaya sabit bir kuvvet etkimesi veya bu kuvvetin stokastik olarak değiştirilmesi durumunda üçüncü eklemdaki hız hatalarının artmasına karşılık adaptif kontrolör kararlı bir performans göstermektedir. Üçüncü eklemdaki bu durumun uygun parametre takımı seçilmesiyle bir ölçüde düzeltilmesi mümkün görülmektedir.

6- Konum ölçmelerinde hatalar yapıldığı varsayılarak yapılan simülasyon çalışması sonucunda da adaptif kontrolörden iyi bir performans elde edilmiştir.

KAYNAKLAR

- [1] PAUL, R., Modelling, Trajectory Calculation and servoring of a Computer-Controlled Arm, Stanford Artificial Intelligence Laboratory, A. I. Memo 177, 1972.
- [2] BEJCZY, A. K., Robot Arm Dynamics and Control, Thecnical Memorandum 33-669, Jet Propulsion Laboratory, 1974.
- [3] MARKIEWICZ, B. R., Analysis of the Computed Torque Drive Method and Comparison With Conventional Postion Servo for a Computer-Controlled Manipulator, Thecnical Memorandum 33-601, Jet Propulsion Laboratory, 1973.
- [4] DUBOWSKY, S., and DESFORGES D. T., The Aplication of Model Referenced Adaptive Control to Robotic Manipulators, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 101, 1979.
- [5] NICOSIA, S., and TOMEI, P., Model Reference Adaptive Control Algorithms for Industrial Robots, Automatica Vol. 20, No. 5, 1984.
- [6] KOIVO, A. J., and GUO, T. H., Adaptive Linear Controller for Robotic Manipulators, IEEE Trans. Aut. Control, Vol. 28, No. 2, 1981.
- [7] TOMIZUKA, M., and HOROWITZ R., Model Reference Adaptive Control of Mechanical Manipulators, Proc. IFAC Symp. Adaptive Control Systems in Control and Signal Proc., San Fransisco, 1983.
- [8] SERAJI, H., A New Approach to Multivariable Control of Manipulators, Jet Propulsion Laboratory, California Institue of Thecnology, Pasadena, Calif. 91109 Vol. 109, 1987.

- [9] PAUL, R. P., Robot Manipulators, Mathematics, Programming and Control, MIT Pres 1984.
- [10] ASADA, H., SLOTINE, J. J. E., Robot Analyssis and Control, John Wiley Sons Inc., 1986.
- [11] SERAJI, H., An Approach to Multivariable Control of Manipulators, Transactions of the ASME, Journal of Dynamic Systems, Measurment and Control, Vol. 109, 1987.
- [12] CHALAM, V. V., Adaptive Control Systems, Institue of Technology Banaras Hindu Universty Varanasi, India, 1987.
- [13] GUPTA, R. R., and CHALAM, V. V., Lyapunov Redesign of Model Reference Adaptive Control Systems, Pts. I and II, J. Inst. Electr. Tele-com. Eng., 1979.
- [14] KOIVO, A. J., and GUO, T., Adaptive Linear Controller for Robotic Manipulators, IEEE Vol. AC-28, No. 2, 1983.

EKLER

EK A

```

DIMENSION WO(6),D1(6),D2(6),ZETA(6),P1(6),P13(6),P12(6)
DIMENSION XX(12),ATA(4,4),HATA(6,2),HIZ(6,2)
DIMENSION F1(6),F2(6),F3(6),F12(6),F13(6),F23(6),QL1(6),QL2(6)
COMMON/AD1/ AUF(6,2),AUKP(36,2),AUKV(36,2),AUC(36,2),AUB(36,2)
COMMON/AD4/ AUA(36,2),P2(6),P3(6)
COMMON/AD3/ XREF(6,2),HREF(6,2),HIREF(6,2)
COMMON/AD2/ ALFA1,ALFA2,BETA1,BETA2,VU1,VU2,GAMA1,GAMA2,CAM1,CAM2
COMMON/AD5/ DELTA1,DELTA2
COMMON/AD6/ TO(6)
COMMON/P2/ XM(6),YM(6),ZM(6),CL(6),CD(6),ALF(6),TETO(6)
COMMON/A1/ ATT(21,16),FKUT(6),RMF(6,4),Q(21,16)
COMMON/JA/ QFAK(6)
COMMON/TUR/ V1(6),V2(6)
COMMON/AW/ FT,FL
COMMON/TM/ TMOM(6)

```

C

```

FT=1.
FL=1.
WRITE(*,*) 'SIM3. INP'
READ(1,*) N,DT,NSTEP
DO 106 I=1,N
106 READ(1,*) F1(I),F2(I),F3(I),F12(I),F13(I),F23(I),V1(I),V2(I)
DO 107 I=1,N
READ(1,*) TETO(I)
107 READ(1,*) XM(I),YM(I),ZM(I),FKUT(I),QFAK(I),CL(I),CD(I),ALF(I)
DO 108 I=1,N
DO 108 J=1,4
108 READ(1,*) RMF(I,J)
DO 501 ISR1=1,N
CALL ATALET(ATA,F1,F2,F3,F12,F13,F23,XM,YM,ZM,FKUT,ISR1)
ISR3=0

```

```
DO 503 IS1=1,4
DO 503 IS2=1,4
IS3=IS3+1
503 ATT(ISR1,IS3)=ATA(IS1,IS2)
501 CONTINUE
DO 104 JQ=1,N
104 CALL TUREV(Q,JQ,QFAK)
CALL REF(TT,NSTEP,DT,N,XREF,HREF,HIREF)
DO 1 I=1,N
XX(I)=XREF(I,2)
1 XX(I+N)=HREF(I,2)
DO 44 I=1,N
44 READ(1,*) WO(I),ZETA(I),QL1(I),QL2(I)
READ(1,*) (AUF(I,1),I=1,N)
READ(1,*) (AUKP(I,1),I=1,N*N)
READ(1,*) (AUKV(I,1),I=1,N*N)
READ(1,*) (AUA(I,1),I=1,N*N)
READ(1,*) (AUB(I,1),I=1,N*N)
READ(1,*) (AUC(I,1),I=1,N*N)
READ(1,*) ALFA1,BETA1,VU1,GAMA1,CAM1,DELTA1
READ(1,*) ALFA2,BETA2,VU1,GAMA2,CAM2,DELTA2
NEQ=N*N
DO 30 I=1,N
D1(I)=WO(I)*WO(I)
D2(I)=2.*ZETA(I)*WO(I)
P1(I)=QL1(I)/D2(I)+(QL1(I)*D2(I))/D1(I)+(QL2(I)*D1(I))/D2(I)
P2(I)=QL1(I)/D1(I)
P3(I)=QL2(I)/D2(I)+QL1(I)/(D1(I)*D2(I))
P13(I)=P1(I)+P3(I)
P12(I)=P1(I)*P3(I)-P2(I)*P2(I)
IF(P13(I).EQ.0.) GO TO 36
IF(P13(I).LT.0.) GO TO 36
IF(P12(I).EQ.0.) GO TO 36
IF(P12(I).LT.0.) GO TO 36
30 CONTINUE
GO TO 37
```

36 WRITE(*,*) 'BASKA Q LAR GIRIN'

STOP

37 CONTINUE

DO 2000 IKI=1,NSTEP

DO 2 IJI=1,5

CC

CALL ADAP(DT,HATA,HIZ,N,TT)

CALL RUNGE(TT,DT,NEQ,XX,N)

IUY=0

DO 99 I=1,N

HATA(I,1)=HATA(I,2)

HIZ(I,1)=HIZ(I,2)

XREF(I,1)=XREF(I,2)

HREF(I,1)=HREF(I,2)

HIREF(I,1)=HIREF(I,2)

AUF(I,1)=AUF(I,2)

DO 99 J=1,N

IUY=IUY+1

AUKP(IUY,1)=AUKP(IUY,2)

AUKV(IUY,1)=AUKP(IUY,2)

AUC(IUY,1)=AUKP(IUY,2)

AUB(IUY,1)=AUKP(IUY,2)

99 AUA(IUY,1)=AUKP(IUY,2)

CALL REF(TT,NSTEP,DT,N,XREF,HREF,HIREF)

DO 8 I=1,N

HATA(I,2)=XREF(I,2)-XX(I)

HIZ(I,2)=HREF(I,2)-XX(I+N)

8 CONTINUE

2 CONTINUE

WRITE(4,*) XREF(1,2),XREF(2,2),XREF(3,2)

WRITE(5,*) XX(1),XX(2),XX(3)

WRITE(6,*) HREF(1,2),HREF(2,2),HREF(3,2)

WRITE(7,*) XX(4),XX(5),XX(6)

WRITE(8,*) TO(1),TO(2),TO(3)

WRITE(9,*) HATA(1,2),HATA(2,2),HATA(3,2)

```
WRITE(10,*) HIZ(1,2),HIZ(2,2),HIZ(3,2)
```

```
2000 CONTINUE
```

```
STOP
```

```
END
```

C

```
SUBROUTINE ADAP(DT,HATA,HIZ,N,TT)
```

```
DIMENSION TO1(6),TO2(6),TO3(6),TO4(6)
```

```
DIMENSION ADR(6,2),HATA(6,2),HIZ(6,2),ARE(36,2),ARH(36,2)
```

```
DIMENSION ART(36,2),ARTH(36,2),ARTI(36,2),ARET(36),ARHT(36)
```

```
DIMENSION ARTT(36),ARTHT(36),ARTIT(36),AAR(6),AFF(6)
```

```
COMMON/AD1/ AUF(6,2),AUKP(36,2),AUKV(36,2),AUC(36,2),AUB(36,2)
```

```
COMMON/AD4/ AUA(36,2),P2(6),P3(6)
```

```
COMMON/AD2/ ALFA1,ALFA2,BETA1,BETA2,VU1,VU2,GAMA1,GAMA2,CAM1,CAM2
```

```
COMMON/AD5/ DELTA1,DELTA2
```

```
COMMON/AD3/ XREF(6,2),HREF(6,2),HIREF(6,2)
```

```
COMMON/AD6/ TO(6)
```

```
COMMON/AW/ FT,FL
```

C

C

```
IV1=0
```

```
IM1=0
```

```
DO 1 IX=1,N
```

```
DO 2 IZ=1,2
```

```
2 ADR(IX,IZ)=P2(IX)*HATA(IX,IZ)+P3(IX)*HIZ(IX,IZ)
```

```
1 CONTINUE
```

```
DO 11 IX=1,N
```

```
IM1=IM1+1
```

```
DO 3 IV=1,N
```

```
IM2=(IV-1)*N+IM1
```

```
DO 21 IY=1,2
```

```
ARE(IM2,IY)=ADR(IX,IY)*HATA(IV,IY)
```

```
ARH(IM2,IY)=ADR(IX,IY)*HIZ(IV,IY)
```

```
ART(IM2,IY)=ADR(IX,IY)*XREF(IV,IY)
```

```
ARTH(IM2,IY)=ADR(IX,IY)*HREF(IV,IY)
```

```
ARTI(IM2,IY)=ADR(IX,IY)*HIREF(IV,IY)
```

```
21 CONTINUE
```

```
3 CONTINUE
11 CONTINUE
    CALL INTEG (ARE, ARET, DT, N)
    CALL INTEG (ARH, ARHT, DT, N)
    CALL INTEG (ART, ARTT, DT, N)
    CALL INTEG (ARTH, ARTHT, DT, N)
    CALL INTEG (ARTI, ARTIT, DT, N)
    IM=0
    DO 4 I=1, N
        AAR(I)=DELTA1*0.5*DT*(ADR(I, 2)+ADR(I, 1))
        AUF(I, 2)=AUF(I, 1)+AAR(I)+DELTA2*ADR(I, 2)
    DO 5 J=1, N
        IM=IM+1
        AUKP(IM, 2)=AUKP(IM, 1)+ALFA2*ARE(IM, 2)+ALFA1*ARET(IM)
        AUKV(IM, 2)=AUKV(IM, 1)+BETA2*ARH(IM, 2)+BETA1*ARHT(IM)
        AUC(IM, 2)=AUC(IM, 1)+VU2*ART(IM, 2)+VU1*ARTT(IM)
        AUB(IM, 2)=AUB(IM, 1)+GAMA2*ARTH(IM, 2)+GAMA1*ARTHT(IM)
        AUA(IM, 2)=AUA(IM, 1)+CAM2*ARTI(IM, 2)+CAM1*ARTIT(IM)
5 CONTINUE
4 CONTINUE
    DO 6 IKS=1, N
6 AFF(IKS)=AUF(IKS, 2)
    CALL GMCAR(AUKP, HATA, AFF, TO1, N)
    CALL GMCAR(AUKV, HIZ, TO1, TO2, N)
    CALL GMCAR(AUC, XREF, TO2, TO3, N)
    CALL GMCAR(AUB, HREF, TO3, TO4, N)
    CALL GMCAR(AUA, HIREF, TO4, TO, N)
    IF(TT.LT.FL) GO TO 55
    DO 56 IER=1, N
56 TO(IER)=TO(IER)/FT
55 RETURN
    END
```

```
SUBROUTINE FTN(XX,YY,NEQ,TT,N)
  DIMENSION DDD(6,6),AA(21,16),DQ(36),LRS(36),MRS(36),FF1(6)
  DIMENSION BB(21,16),B(21,16),CC(21,16),C(21,16),YY(6),DJT(36)
  DIMENSION DD(21,16),D(21,16),EE(21,16),E(21,16),FDQ(6),TOJ(6)
  DIMENSION UJ(21,16),UJ1(16),UKK(91),XX(12),A(4,4),TX(6)
  DIMENSION FK(21,16),U11(16),UK1(16),UKU(91,16),HH(6),CHD(6)
```

C

```
COMMON/AS/ HTJAC(18)
COMMON/AC/ HJACT(18),SJAC(36),SJACT(36),HJAC(18)
COMMON/JA/ QFAK(6)
COMMON/TUR/ V1(6),V2(6)
COMMON/AD6/ TO(6)
COMMON/P2/ XM(6),YM(6),ZM(6),CL(6),CD(6),ALF(6),TETO(6)
COMMON/A1/ ATT(21,16),FKUT(6),RMF(6,4),Q(21,16)
COMMON/A2/ T(21,16),S(21,16),U(21,16),F(21,16)
COMMON/A3/ KI(6),U1(21,16),KK(6),SF(21,16),SSF(21,16),SFF(21,16)
COMMON/TM/ TMOM(6)
```

C

```
DO 10 JJ=1,N
  IF(QFAK(JJ).EQ.0.) GO TO 4000
  TX(JJ)=XX(JJ)
  GO TO 3000
4000 CD(JJ)=XX(JJ)
  TX(JJ)=0.
3000 CALL MATRIS(A,CL,CD,ALF,TX,JJ,TETO)
  K=0
  DO 10 I=1,4
    DO 10 J=1,4
      K=K+1
      T(JJ,K)=A(I,J)
10 CONTINUE
  DO 13 I=1,16
13 S(1,I)=T(1,I)
  IF(N.EQ.1) GO TO 142
  DO 14 I=2,N
  IC1=I-1
```

```
14 CALL CARPIM(T,S,IC1,I,I)
142 J=1
      DO 141 II=1,N
      CALL MATCAR(Q,S,AA,1,II,II)
      J=J-1
      J=J+II
      DO 141 JJK=1,16
141 U(J,JJK)=AA(II,JJK)
301 IF(N.EQ.1) GO TO 222
      DO 15 I=1,16
15 F(1,I)=T(2,I)
      IF(N.EQ.2) GO TO 162
      DO 16 I=2,N-1
      IC1=I-1
      NMJK=I+1
16 CALL CARPIM(T,F,IC1,NMJK,I)
162 CALL MATCAR(S,Q,BB,1,2,1)
      J=2
      DO 161 II=1,N-1
      CALL MATCAR(BB,F,B,1,II,II)
      J=II+J
      DO 161 JJK=1,16
161 U(J,JJK)=B(II,JJK)
      IF(N.EQ.2) GO TO 222
      DO 17 I=1,16
17 SF(1,I)=T(3,I)
      IF(N.EQ.3) GO TO 182
      DO 18 I=2,N-2
      IC1=I-1
      NMJK=I+2
18 CALL CARPIM(T,SF,IC1,NMJK,I)
182 J=4
      CALL MATCAR(S,Q,CC,2,3,1)
      DO 181 II=1,N-2
      J=J+1
      CALL MATCAR(CC,SF,C,1,II,II)
```

```
J=II+J
DO 181 JJK=1,16
181 U(J,JJK)=C(II,JJK)
    IF(N.EQ.3) GO TO 222
    DO 19 I=1,16
19 SSF(1,I)=T(4,I)
    IF(N.EQ.4) GO TO 192
    DO 20 I=2,N-3
    IC1=I-1
    NMJK=I+3
20 CALL CARPIM(T,SSF,IC1,NMJK,I)
192 J=7
    CALL MATCAR(S,Q,DD,3,4,1)
    DO 201 II=1,N-3
    J=J+2
    CALL MATCAR(DD,SSF,D,1,II,II)
    J=II+J
DO 201 JJK=1,16
201 U(J,JJK)=D(II,JJK)
    IF(N.EQ.4) GO TO 222
    DO 21 I=1,16
21 SFF(1,I)=T(5,I)
    IF(N.EQ.5) GO TO 212
    DO 22 I=2,N-4
    IC1=I-1
    NMJK=I+4
22 CALL CARPIM(T,SFF,IC1,NMJK,I)
212 CALL MATCAR(S,Q,EE,4,5,1)
    J=11
    DO 213 II=1,N-4
    J=J+3
    CALL MATCAR(EE,SFF,E,1,II,II)
    J=J+II
    DO 213 JJK=1,16
213 U(J,JJK)=E(II,JJK)
    IF(N.EQ.5) GO TO 222
```

```
CALL MATCAR(S,Q,FK,5,6,1)
CALL MATCAR(FK,T,U,1,6,21)
222 DO 666 I=1,N
666 KI(I)=0
    KI(1)=1
    DO 667 I=1,N-1
667 KI(I+1)=I+KI(I)+1
    IF(TT.NE.O.) GO TO 302
C    WRITE(3,*) 'KI', (KI(IT), IT=1,N)
302 DO 668 I=1,KI(N)
668 CALL TRANS(U,U1,I)
    IF(N.EQ.1) GO TO 555
    DO 445 ISAY=1,N-1
    IJF1=KI(ISAY)+1
    IJF2=KI(ISAY+1)
    ISAY1=ISAY+1
    DO 445 IJF=IJF1,IJF2
445 CALL MATCAR(U,ATT,UJ,IJF,ISAY1,IJF)
    IUS=1
    DO 446 ISAY=1,N-1
    IJF1=KI(ISAY)+1
    IJF2=KI(ISAY+1)
    DO 446 IJF=IJF1,IJF2
    IKF=IJF1-1
456 IUS=IUS+1
    IKF=IKF+1
    IF(IKF.GT.IJF2) GO TO 443
    IF(IJF.LT.IKF) GO TO 456
    CALL MATC1(UJ,U1,UKU,IJF,IKF,IUS)
    CALL TRACE(UKU,UKK,IUS)
    GO TO 456
443 IUS=IUS-1
446 CONTINUE
555 CALL MATCAR(U,ATT,UJ,1,1,1)
    CALL MATC1(UJ,U1,UKU,1,1,1)
    CALL TRACE(UKU,UKK,1)
```

```
KK(1)=1
DO 457 JI=1,N-1
KN=JI*JI
457 KK(JI+1)=KN+KK(JI)
```

C

```
CALL JAC(N,XX)
CALL GMPRD(HJACT,HJAC,DJT,N,3,N)
DO 458 MS=1,N
DO 458 NS=1,N
458 DDD(MS,NS)=0.
DO 459 MS=1,N
DO 459 KS=MS,N
DO 459 IS=KS,N
JJS=KK(IS)+IS*(KS-1)+MS-1
DDD(MS,KS)=DDD(MS,KS)+UKK(JJS)
459 DDD(KS,MS)=DDD(MS,KS)
CALL H(XX,CHD,HH,N,TT)
CALL SLM(FLM,TT)
KO=0
CTY=0.00005
DO 449 IO=1,N
DO 449 JO=1,N
KO=KO+1
DQ(KO)=DDD(JO,IO)+DJT(KO)*FLM
IF(ABS(DQ(KO)).LT.CTY) DQ(KO)=0.
449 CONTINUE
CALL MINV(DQ,N,OR,LRS,MRS)
CALL GMPRD(SJACT,TMOM,TOJ,N,6,1)
DO 5 IUO=1,N
5 FF1(IUO)=TO(IUO)-HH(IUO)-CHD(IUO)-TOJ(IUO)
CALL GMPRD(DQ,FF1,FDQ,N,N,1)
DO 6 IUO=1,N
YY(IUO)=XX(IUO+N)
6 YY(IUO+N)=FDQ(IUO)
RETURN
END
```

```
SUBROUTINE H(XX,CHD,HH,N,TT)
  DIMENSION QA(21,16),AS(16),Q1(16),Q2(16),S1(16),G(4)
  DIMENSION CCS(16),BBS(16),AQ(16),AOB(16),QC(16),UTA(16),XX(12)
  DIMENSION KU(6),HHH(6,6),UAJ(91,16),UA1J(91,16),HKK(91),HH(6)
  DIMENSION UC(16),UCG(16),KM(6),RM(4),CHD(6),CDH(1),Q3(16)
  DIMENSION QM(21,16),GM(3),GMF(6),SIS(6),HTX(6),DXM(6),HHJC(36)
  COMMON/HH/ UA(91,16)
  COMMON/JA/ QFAK(6)
  COMMON/AS/ HTJAC(18)
  COMMON/AC/ HJACT(18),SJAC(36),SJACT(36),HJAC(18)
  COMMON/TUR/ V1(6),V2(6)
  COMMON/A1/ ATT(21,16),FKUT(6),RMF(6,4),Q(21,16)
  COMMON/A2/ T(21,16),S(21,16),U(21,16),F(21,16)
  COMMON/A3/ KI(6),U1(21,16),KK(6),SF(21,16),SSF(21,16),SFF(21,16)
  DATA G/O.,O.,-9.81,O./
  MM=0
  DO 573 IH=1,N
  DO 574 KH=1,N
  DO 572 JH=1,N
  IF(IH.LT.KH) GO TO 573
  IF(IH.LT.JH) GO TO 574
  MM=MM+1
  IF(KH.LT.JH) GO TO 572
  IJKH=IH+JH+KH
  IF(IJKH.GT.3) GO TO 581
  DO 99 IDW=1,16
99 QM(1,IDW)=Q(1,IDW)
  CALL MATCAR(Q,Q,QA,1,1,1)
  DO 1 IV=1,16
  S1(IV)=S(1,IV)
1 Q1(IV)=QA(1,IV)
  CALL GMPRD(S1,Q1,UTA,4,4,4)
  DO 2 IV=1,16
2 UA(MM,IV)=UTA(IV)
  GO TO 572
581 IF(KH.GT.1) GO TO 591
```

```
      DO 12 IFM=1,16
12  AS(IFM)=0.
      DO 590 IFM=1,16,5
590 AS(IFM)=1.
      IF(JH.GT.1) GO TO 591
      DO 593 I=1,16
593 CCS(I)=S(IH,I)
      DO 974 IKF=1,16
      Q2(IKF)=Q(1,IKF)
974 Q1(IKF)=Q(1,IKF)
      CALL GMPRD(Q1,Q2,Q3,4,4,4)
      CALL GMPRD(CCS,Q3,UTA,4,4,4)
      DO 3 IV=1,16
      3 UA(MM,IV)=UTA(IV)
      GO TO 572
591 IF(KH.EQ.JH) GO TO 350
      IF(JH.GT.1) GO TO 342
      DO 341 IMH=1,16
341 BBS(IMH)=S((KH-1),IMH)
      GO TO 575
342 IF(JH.GT.2) GO TO 343
      DO 344 IMH=1,16
344 BBS(IMH)=F((KH-2),IMH)
      GO TO 576
343 IF(JH.GT.3) GO TO 345
      DO 346 IMH=1,16
346 BBS(IMH)=SF((KH-3),IMH)
      GO TO 576
345 IF(JH.GT.4) GO TO 347
      DO 348 IMH=1,16
348 BBS(IMH)=SSF((KH-4),IMH)
      GO TO 576
347 IF(JH.GT.5) GO TO 350
      DO 349 IMH=1,16
349 BBS(IMH)=SFF((KH-5),IMH)
      GO TO 575
```

```
350 DO 13 IX=1,16
  13 BBS(IX)=0.
    DO 17 IFM=1,16,5
  17 BBS(IFM)=1.
576 DO 744 IHM=1,16
744 AS(IHM)=S((JH-1),IHM)
575 IF(KH.GT.2) GO TO 601
    DO 602 IHM=1,16
602 CCS(IHM)=F((IH-1),IHM)
    GO TO 671
601 IF(KH.GT.3) GO TO 611
    DO 612 IHM=1,16
612 CCS(IHM)=SF((IH-2),IHM)
    GO TO 671
611 IF(KH.GT.4)GO TO 621
    DO 622 IHM=1,16
622 CCS(IHM)=SSF((IH-3),IHM)
    GO TO 671
621 IF(KH.GT.5) GO TO 631
    DO 632 IHM=1,16
632 CCS(IHM)=SFF((IH-4),IHM)
    GO TO 671
631 DO 633 IHM=1,16
633 CCS(IHM)=T(6,IHM)
671 DO 5 IV=1,16
  5 Q1(IV)=Q(JH,IV)
    CALL GMPRD(Q1,AS,AQ,4,4,4)
    CALL GMPRD(BBS,AQ,AQB,4,4,4)
    DO 6 IV=1,16
  6 Q2(IV)=Q(KH,IV)
    CALL GMPRD(CCS,Q2,QC,4,4,4)
    CALL GMPRD(QC,AQB,UTA,4,4,4)
    DO 7 IV=1,16
  7 UA(MM,IV)=UTA(IV)
```

572 CONTINUE

574 CONTINUE

573 CONTINUE

C

DO 851 IHS=1,N

KU(IHS)=KI(IHS)

IF(IHS.EQ.N) GO TO 10

DO 801 I=IHS,N-1

801 KU(I+1)=KU(I)+I*I

10 CONTINUE

DO 802 MS=1,N

DO 802 NS=1,N

802 HHH(MS,NS)=0.

DO 803 MS=1,N

DO 803 KS=MS,N

DO 804 IS=KS,N

IF(IS.LT.IHS) GO TO 804

JJS=KK(IS)+IS*(KS-1)+MS-1

CALL MATC(UA,ATT,UAJ,JJS,IS,JJS)

CALL MATC(UAJ,U1,UA1J,JJS,KU(IS),JJS)

CALL TRACE(UA1J,HKK,JJS)

HHH(MS,KS)=HHH(MS,KS)+HKK(JJS)

HHH(KS,MS)=HHH(MS,KS)

804 CONTINUE

803 CONTINUE

HH(IHS)=0.

DO 851 KHK=1,N

DO 851 MHM=1,N

851 HH(IHS)=HH(IHS)+HHH(KHK,MHM)*XX(KHK+N)*XX(MHM+N)

C

CALL SLM(FLM,TT)

DO 1011 IV=1,N

1011 DXM(IV)=FLM*XX(IV+N)

CALL SIGN(XX,SIS,N)

CALL GMPRD(HJACT,HTJAC,HHJC,N,3,N)

CALL GMPRD(HHJC,DXM,HTX,N,N,1)

```
DO 1020 IV=1,N
1020 HH(IV)=HH(IV)+HTX(IV)+V1(IV)*XX(IV+N)+V2(IV)*SIS(IV)
DO 251 ICG=1,N
KM(ICG)=KI(ICG)
IF(ICG.EQ.N) GO TO 29
DO 252 IGC=ICG,N-1
252 KM(IGC+1)=KM(IGC)+IGC
29 CONTINUE
CHD(ICG)=0.
DO 251 JCG=ICG,N
CALL DIZME(U,UC,KM(JCG))
CALL GMPRD(UC,6,UCG,4,4,1)
DO 119 ISE=1,4
119 RM(ISE)=RMF(JCG,ISE)
CALL GMPRD(RM,UCG,CDH,1,4,1)
251 CHD(ICG)=CHD(ICG)-FKUT(JCG)*CDH(1)
DO 60 IFKC=1,3
60 GM(IFKC)=-G(IFKC)
CALL GMPRD(HJACT,GM,GMF,N,3,1)
DO 1021 IV=1,N
1021 CHD(IV)=CHD(IV)+FLM*GMF(IV)
IF(TT.NE.O.) GO TO 11
WRITE(*,*) CHD(1),CHD(2),CHD(3)
11 RETURN
END
```

C

```
SUBROUTINE RUNGE(TT,DT,NEQ,XX,N)
DIMENSION YI(12),YJ(12),YK(12),YL(12),UU(12),XX(12),OO(12)
COMMON/P2/ XM(6),YM(6),ZM(6),CL(6),CD(6),ALF(6),TETO(6)
COMMON/AD6/ TO(6)
COMMON/A1/ ATT(21,16),FKUT(6),RMF(6,4),Q(21,16)
COMMON/JA/ QFAK(6)
COMMON/TUR/ V1(6),V2(6)
COMMON/TM/ TMOM(6)
DO 10 IM=1,NEQ
10 UU(IM)=XX(IM)
```

```
      CALL FTN(XX, OO, NEQ, TT, N)
      DO 20 IM=1, NEQ
        YI(IM)=OO(IM)*DT
20    XX(IM)=UU(IM)+YI(IM)/2.0
        TT=TT+DT/2.0
        CALL FTN(XX, OO, NEQ, TT, N)
        DO 30 IM=1, NEQ
          YJ(IM)=OO(IM)*DT
30    XX(IM)=UU(IM)+YJ(IM)/2.0
        CALL FTN(XX, OO, NEQ, TT, N)
        DO 40 IM=1, NEQ
          YK(IM)=OO(IM)*DT
40    XX(IM)=UU(IM)+YK(IM)
        TT=TT+DT/2.0
        CALL FTN(XX, OO, NEQ, TT, N)
        DO 50 IM=1, NEQ
          YL(IM)=OO(IM)*DT
50    XX(IM)=UU(IM)+(YI(IM)+2.0*YJ(IM)+2.0*YK(IM)+YL(IM))/6.0
        CALL MOM(TMOM, TT)
        RETURN
      END
```

C

```
      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
    A(IK)=A(IK)/(-BIGA)
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=M(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
```

C

```
      SUBROUTINE INTEG(A,B,DT,N)
      DIMENSION A(36,2),B(36)
      DO 1 I=1,N*N
1    B(I)=((A(I,2)+A(I,1))*0.5)*DT
      RETURN
      END
```

C

```
      SUBROUTINE GMCAR(A,B,C,D,N)
      DIMENSION F(6),A(36,2),B(6,2),C(6),D(6),E(36)
      IO=0
      DO 10 I=1,N
      DO 10 J=1,N
      IO=IO+1
10   E(IO)=A(IO,2)
      DO 30 III=1,N
30   F(III)=B(III,2)
      CALL GMPRD(E,F,D,N,N,1)
      DO 20 I=1,N
20   D(I)=C(I)+D(I)
      RETURN
      END
```

C

```
      SUBROUTINE SIGN(A,B,N)
      DIMENSION A(12),B(6)
      I=0
      J=N-1
2    I=I+1
      J=J+1
      IF(A(J).NE.0.) GO TO 1
      B(I)=0.
```

```
      IF(I.EQ.N) GO TO 3
      GO TO 2
1     B(I)=A(J)/ABS(A(J))
      IF(I.LT.N) GO TO 2
3     RETURN
      END
```

C

```
      SUBROUTINE ATALET(ATA,F1,F2,F3,F12,F13,F23,XM,YM,ZM,FKUT,I)
      DIMENSION F1(6),F2(6),F3(6),F12(6),F13(6),F23(6),ATA(4,4)
      DIMENSION XM(6),YM(6),ZM(6),FKUT(6)
      ATA(1,1)=0.5*(F2(I)-F1(I)+F3(I))*FKUT(I)
      ATA(1,2)=F12(I)*FKUT(I)
      ATA(1,3)=F13(I)*FKUT(I)
      ATA(1,4)=XM(I)*FKUT(I)
      ATA(2,1)=FKUT(I)*F12(I)
      ATA(2,2)=0.5*FKUT(I)*(F1(I)-F2(I)+F3(I))
      ATA(2,3)=F23(I)*FKUT(I)
      ATA(2,4)=YM(I)*FKUT(I)
      ATA(3,1)=F13(I)*FKUT(I)
      ATA(3,2)=F23(I)*FKUT(I)
      ATA(3,3)=0.5*FKUT(I)*(F1(I)+F2(I)-F3(I))
      ATA(3,4)=ZM(I)*FKUT(I)
      ATA(4,1)=ATA(1,4)
      ATA(4,2)=ATA(2,4)
      ATA(4,3)=ATA(3,4)
      ATA(4,4)=FKUT(I)
      RETURN
      END
```

C

```
SUBROUTINE TUREV(Q, JQ, QFAK)
  DIMENSION Q(21, 16), QFAK(6)
  DO 1 IIQ=1, 16
1  Q(JQ, IIQ)=0.
    IF(QFAK(JQ).EQ.1) GO TO 2
    Q(JQ, 12)=1.
    GO TO 3
2  Q(JQ, 2)=-1.
    Q(JQ, 5)=1.
3  RETURN
  END
```

C

```
SUBROUTINE MATRIS(A, CL, CD, ALF, TET, IY, TETO)
  DIMENSION A(4, 4), CL(6), CD(6), ALF(6), TET(6), TETO(6)
  A(1, 1)=COS(TET(IY))*COS(TETO(IY))-SIN(TET(IY))*SIN(TETO(IY))
  A(1, 2)=-COS(ALF(IY))*SIN(TET(IY))*COS(TETO(IY))
  A(1, 2)=A(1, 2)-COS(TET(IY))*SIN(TETO(IY))*COS(ALF(IY))
  A(1, 3)=SIN(ALF(IY))*SIN(TET(IY))*COS(TETO(IY))
  A(1, 3)=A(1, 3)+COS(TET(IY))*SIN(TETO(IY))*SIN(ALF(IY))
  A(1, 4)=CL(IY)*COS(TET(IY))
  A(2, 1)=SIN(TET(IY))*COS(TETO(IY))+COS(TET(IY))*SIN(TETO(IY))
  A(2, 2)=COS(ALF(IY))*COS(TET(IY))*COS(TETO(IY))
  A(2, 2)=A(2, 2)-SIN(TET(IY))*SIN(TETO(IY))*COS(ALF(IY))
  A(2, 3)=-SIN(ALF(IY))*COS(TET(IY))*COS(TETO(IY))
  A(2, 3)=A(2, 3)+SIN(TET(IY))*SIN(TETO(IY))*SIN(ALF(IY))
  A(2, 4)=CL(IY)*SIN(TET(IY))
  A(3, 1)=0.
  A(4, 1)=0.
  A(4, 2)=0.
  A(4, 3)=0.
  A(4, 4)=1.
  A(3, 2)=SIN(ALF(IY))
  A(3, 3)=COS(ALF(IY))
  A(3, 4)=CD(IY)
  RETURN
  END
```

```
SUBROUTINE DIZME(BDI,ADI,IDI)
  DIMENSION ADI(16),BDI(21,16)
  DO 300 KDI=1,16
300 ADI(KDI)=BDI(IDI,KDI)
  RETURN
END
```

C

C

```
SUBROUTINE MATCAR(CA,DA,SA,IAC,JAC,NMK)
  DIMENSION CA(21,16),DA(21,16),SA(21,16),EA(16),FA(16),GA(16)
  CALL DIZME(CA,EA,IAC)
  CALL DIZME(DA,FA,JAC)
  CALL GMPRD(FA,EA,GA,4,4,4)
  CALL TDIZME(GA,SA,NMK)
  RETURN
END
```

C

```
SUBROUTINE MATC1(CA,DA,SA,IAC,JAC,NMK)
  DIMENSION CA(21,16),DA(21,16),SA(91,16),EA(16),FA(16),GA(16)
  CALL DIZME(CA,EA,IAC)
  CALL DIZME(DA,FA,JAC)
  CALL GMPRD(FA,EA,GA,4,4,4)
  CALL TDIZ1(GA,SA,NMK)
  RETURN
END
```

C

```
SUBROUTINE MATC(CA,DA,SA,IAC,JAC,NMK)
  DIMENSION CA(91,16),DA(21,16),SA(91,16),EA(16),FA(16),GA(16)
  DO 4 IFS=1,16
4 EA(IFS)=CA(IAC,IFS)
  CALL DIZME(DA,FA,JAC)
  CALL GMPRD(FA,EA,GA,4,4,4)
  CALL TDIZ1(GA,SA,NMK)
  RETURN
END
```

C

```
SUBROUTINE TDIZME (AFK, BFK, IAB)
  DIMENSION AFK (16), BFK (21, 16)
  DO 301 KAB=1, 16
301 BFK (IAB, KAB) = AFK (KAB)
  RETURN
END
```

C

```
SUBROUTINE TDIZ1 (AFK, BFK, IAB)
  DIMENSION AFK (16), BFK (91, 16)
  DO 301 KAB=1, 16
301 BFK (IAB, KAB) = AFK (KAB)
  RETURN
END
```

C

```
SUBROUTINE TRACE (ATE, BTE, ITE)
  DIMENSION ATE (91, 16), BTE (91), CTE (16)
  DO 9 IJGK=1, 16
9 CTE (IJGK) = ATE (ITE, IJGK)
  BTE (ITE) = 0.
  DO 8 IITC=1, 16, 5
8 BTE (ITE) = BTE (ITE) + CTE (IITC)
  RETURN
END
```

CC

C

```
SUBROUTINE TRANS (ATR, BTR, ITR)
  DIMENSION ATR (21, 16), BTR (21, 16), CTR (16), DTR (16)
  CALL DIZME (ATR, CTR, ITR)
  I1R = -4
  IJRA = 0
  DO 9 JJRA=1, 4
  I1R = I1R + 1
  I1RA = I1R
  DO 9 IIRA=1, 4
  IJRA = IJRA + 1
  I1RA = I1RA + 4
```

```
9 DTR(IJRA)=CTR(I1RA)
  CALL TDIZME(DTR,BTR,ITR)
  RETURN
  END
```

C

C

```
SUBROUTINE CARPIM(ACA,BCA,IC1,NMJK,ICA)
  DIMENSION ACA(21,16),BCA(21,16),CCA(16),DCA(16),ECA(16)
  CALL DIZME(ACA,CCA,NMJK)
  CALL DIZME(BCA,DCA,IC1)
  CALL GMPRD(CCA,DCA,ECA,4,4,4)
  CALL TDIZME(ECA,BCA,ICA)
  RETURN
  END
```

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
        IF(A(JI).EQ.0.) GO TO 10
        IF(B(IB).EQ.0.) GO TO 10
        R(IR)=R(IR)+A(JI)*B(IB)
      10 CONTINUE
    RETURN
  END
```

```
SUBROUTINE JAC(N,XX)
  DIMENSION BVEC(6,3),KSAY(6),DJAC(6,3),XX(12),NFN(6),KMK(6),KF(6)
  COMMON/AC/ HJACT(18),SJAC(36),SJACT(36),HJAC(18)
  COMMON/AS/ HTJAC(18)
  COMMON/JA/ QFAK(6)
  COMMON/A2/ T(21,16),S(21,16),U(21,16),F(21,16)
  COMMON/HH/ UA(91,16)
  DT=0.002
  BVEC(1,1)=0.
  BVEC(1,2)=0.
  BVEC(1,3)=1.
  DO 1001 IVEC=2,N
  DO 1001 JVEC=1,3
  IVEC1=IVEC-1
  JVEC1=4*JVEC-1
  IF(QFAK(IVEC).NE.0.) GO TO 1002
  BVEC(IVEC,JVEC)=0.
  GO TO 1001
1002 BVEC(IVEC,JVEC)=S(IVEC1,JVEC1)
1001 CONTINUE
  KSAY(1)=1
  DO 1003 IVEC=1,N-1
1003 KSAY(IVEC+1)=KSAY(IVEC)+IVEC
  DO 1004 IVEC=1,N
  KKSA=KSAY(N)+IVEC-1
  DO 1004 JVEC=1,3
  JVEC1=4*JVEC
1004 DJAC(IVEC,JVEC)=U(KKSA,JVEC1)
  DO 1006 IVC=1,N
  DO 1006 JVC=1,3
  IV1=JVC+6*(IVC-1)
  SJAC(IV1)=DJAC(IVC,JVC)
  SJAC(IV1+3)=BVEC(IVC,JVC)
  JV1=JVC+3*(IVC-1)
1006 HJAC(JV1)=DJAC(IVC,JVC)
```

```
DO 1007 IVC=1,6
DO 1007 JVC=1,N
JV1=IVC+6*(JVC-1)
IV1=JVC+N*(IVC-1)
1007 SJACT(IV1)=SJAC(JV1)
DO 1008 IVC=1,3
DO 1008 JVC=1,N
JV1=IVC+3*(JVC-1)
IV1=JVC+N*(IVC-1)
1008 HJACT(IV1)=HJAC(JV1)
KMK(1)=1
DO 9 I=1,N-1
9 KMK(I+1)=KMK(I)+I*I
DO 7 I=1,N
7 KF(I)=KMK(N)+N*(I-1)
DO 17 M=N,1,-1
DO 12 I=1,M
12 NFN(I)=KF(M)+I-1
IF(M.EQ.N) GO TO 15
DO 14 NM=M+1,N
14 NFN(NM)=NFN(NM-1)+N
15 CONTINUE
DO 13 J=1,3
DO 13 K=1,N
13 HTJAC((M-1)*3+J)=UA(NFN(K),4*J)*XX(N+K)+HTJAC((M-1)*3+J)
17 CONTINUE
RETURN
END
```

C

```
SUBROUTINE REF(T,NSTEP,H,N,XREF,HREF,HIREF)
DIMENSION XREF(6,2),HREF(6,2),HIREF(6,2)
XREF(1,2)=1.5708*0.5*(1.+6.*EXP(-T/0.6)-8.*EXP(-T/0.8))
XREF(2,2)=XREF(1,2)
HREF(1,2)=1.5708*0.5*(10.*EXP(-T/0.8)-10.*EXP(-T/0.6))
HREF(2,2)=HREF(1,2)
HIREF(1,2)=15.708*((10./6.)*EXP(-T/0.6)-1.25*EXP(-T/0.8))
```

```
HIREF(2,2)=HIREF(1,2)
XREF(3,2)=0.5
HREF(3,2)=0.
HIREF(3,2)=0.
RETURN
END
```

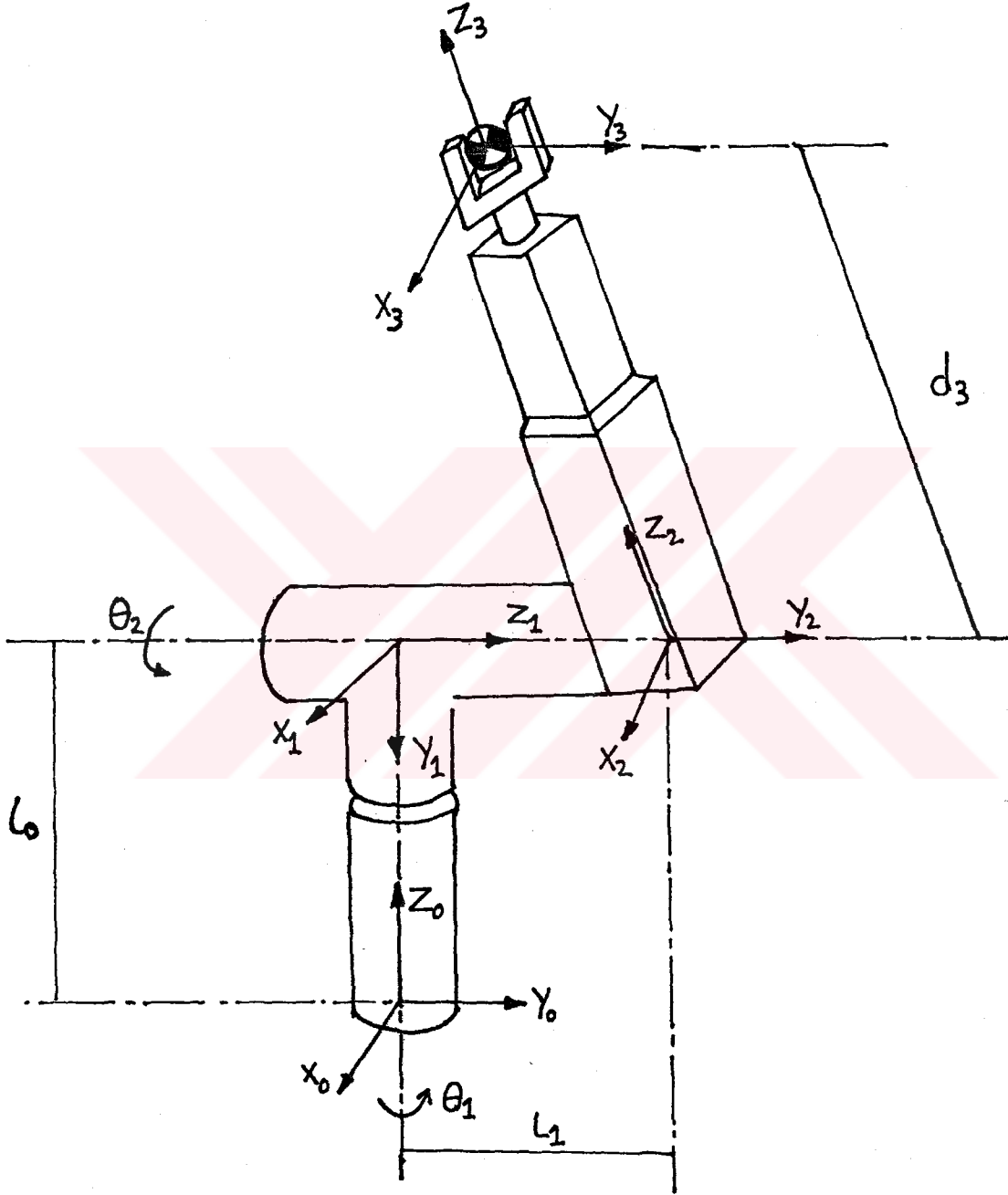
C

```
SUBROUTINE SLM(FLM, TT)
FLM=10.
RETURN
END
```

C

```
SUBROUTINE MOM(TM, TT)
DIMENSION TM(6)
TM(1)=0.
TM(2)=0.
TM(3)=0.
TM(4)=0.
TM(5)=0.
TM(6)=0.
RETURN
END
```

EK B



Sekil B.1 Robot üzerinde Denavit-Hartenberg parametrelerinin gösterimi

Tablo B.1 Denavit-Hartenberg Parametreleri

i. nci uzuv	a_i	α_i	d_i	e_i
1	0	-90	0	e_1
2	0	+90	0.21	e_2
3	0	0	d_3	0

Tablo B.2 Robotun fiziksel parametreleri

m_1 [Kg]	9.29	5.01	4.25
$\bar{x}_1$ [m]	0	0	0
$\bar{y}_1$ [m]	0.0175	-0.1054	0
$\bar{z}_1$ [m]	-0.1105	0	-0.6447
k_{111}^2 [m ⁴]	0.0297	0.0215	0.5908
k_{122}^2 [m ⁴]	0.0274	0.0035	0.5908
k_{133}^2 [m ⁴]	0.0076	0.0200	0.0015

ÖZGEÇMİŞ

A. Selçuk Tekdemir 1967 yılında İstanbul' da doğdu. İstanbul Koca Mustafa Paşa Lisesinden 1984 yılında mezun oldu. Yine aynı sene İ. T. U. Makina Fakültesi, Makina Mühendisliği Bölümüne girdi. Bu bölümde Haziran, 1988 de lisans derecesi aldı. Aynı sene İ. T. U. Fen Bilimleri Enstitüsü Robotik Programına girdi.

