A STUDY OF MANIPULATOR DYNAMICS AND CONTROL

by

ASAF AKMEHMET

B.S. in M.E., Boğaziçi University, 1981

Submitted to the Institute for Graduate Studies in Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Boğaziçi University

1984

ACKNOWLEDGEMENTS

I would like to express my sincere gratitudes to Doç.Dr. Ahmet Kuzucu for his valuable advice, guidance, encouragement, and for all the help he has given to me throughout this study.

I also wish to thank Prof.Dr. Akın Tezel and Y.Doç.Dr. Akif Eyler for their interest and helpful suggestions.

Asaf Akmehmet

A STUDY OF MANIPULATOR DYNAMICS AND CONTROL

ABSTRACT

In this work, the dynamics and control of a manipulator arm are investigated. Two different forms of the dynamic model of an n degree of freedom manipulator are studied and general computer oriented algorithms are developed to obtain the system matrices of the dynamic model. The algorithms are based on Newtonian mechanics; they are recursive and independent of the manipulator configuration.

For the purpose of control of a manipulator arm two different control schemes are proposed which are based on minimum energy optimal control. In the first method the system is decomposed into n subsystems and each subsystem is controlled independently while in the second method the dynamic coupling among subsystems is taken into account.

Computer simulations are carried out for two different manipulator models in order to investigate the effectiveness of the proposed control methods, and it is seen that the proposed suboptimal adaptive feedback law gives reasonably good results.

ROBOT KOLU DİNAMİĞİ VE DENETİMİ ÜZERİNE BİR ÇALIŞMA

KISA ÖZET

Bu çalışmada bir robot kolunun dinamiği ve denetimi incelenmiştir. N serbestlik dereceli bir robot kolunun dinamik modelinin elde edilebilmesi için iki genel hesap yöntemi geliştirilmiştir. Bu yöntemler Newton mekaniğine göre geliştirilmiş olup robot kolunun yapısından bağımsızdırlar.

Robot kolunun denetimi için iki değişik yöntem sunulmuştur. Her iki yöntem de minimum enerji optimum kontrol teorisine göre geliştirilmiş olup ilk yöntemde sistem eklem sayısı kadar (n tane) altsisteme ayrılıp her altsistem bağımsız olarak kontrol edilmekte, ikinci yöntemde ise eklemler arasındaki dinamik etkileşim gözönüne alınmaktadır.

Sunulan kontrol yöntemlerinin uygulanılabilirliğini sınamak amacıyla iki değişik robot kolunun kontrolu bilgisayarda benzetim yolu ile incelenmiş ve sunulan altoptimal kontrol yönteminin iyi sonuçlar verdiği gözlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Page No.

LIST OF TABLES

LIST OF SYMBOLS

$\vec{a}$	Approach vector of the hand
$\vec{a}_o$	Disturbance vector
a_i	Distance between the origins of coordinate systems i-1 and i measured along $\vec{x}_i$
$\vec{a}_i$	Linear acceleration of link i
$\vec{a}_{Gi}$	Linear acceleration of the mass center of link i
A	System matrix
A_i^{i-1}	Homogeneous transformation matrix between coordinate systems i-1 and i
b_i	Viscous damping coefficient associated with joint i
B	Input or control matrix
$C(\vec{q},\dot{\vec{q}})$	nxn matrix specifying centrifugal and Coriolis effects
d_i	Distance between $\vec{x}_{i-1}$ and $\vec{x}_i$ measured along $\vec{z}_{i-1}$
$\vec{f}(\dot{q}_i\dot{q}_j,\vec{q})$	nx1 vector defining centrifugal and Coriolis terms
$\vec{f}_i$	Force exerted on link i by link i-1
$\vec{F}_i$	Total force exerted on link i
$\vec{g}$	Gravitational acceleration
$\vec{g}(\vec{q})$	nx1 vector defining the gravity terms
G_k	Linear feedback matrix
$\vec{h}(\vec{q})$	nx1 vector defining the terms due to external forces and moments exerted on link n
H	Arm matrix (= A_o^n)
I_i	Inertia tensor of link i with respect to its center of mass expressed in base coordinates

$\overline{I}_i$	Inertia tensor of link i with respect to its center of mass expressed in link i's body coordinates
J	Quadratic performance index
$J(\vec{q})$	nxn symmetric, nonsingular moment of inertia matrix
K_p	Position feedback gain matrix
K_v	Velocity feedback
m_i	Mass of link i
$\vec{m}_i$	Moment exerted on link i by link i-1
$\vec{M}_i$	Total moment exerted on link i
n	Number of degrees of freedom
$\vec{n}$	Normal vector of the hand
$\vec{p}$	Position vector of the hand
$\vec{p}_i$	Vector from the base coordinate origin to the joint i coordinate origin
$\vec{p}_i^*$	Vector from coordinate origin i-1 to coordinate origin i
$\vec{q}$	Vector of generalized coordinates
$\vec{q}_d$, $\dot{\vec{q}}_d$, $\ddot{\vec{q}}_d$	Vector of desired joint coordinates, velocities, and accelerations, respectively
Q	Constant positive semi-definite output penalization matrix
R	Constant positive definite control penalization matrix
R_i^{i-1}	Rotation matrix between coordinate systems i-1 and i
$\vec{s}$	Sliding vector of the hand
$\vec{s}_i$	Vector from the base coordinate origin to the link i center of mass
t	Time

T	Horizon time (or time-to-go)
$\vec{u}$	Input or control vector
$\vec{u}_{ok}$	Open loop control vector
$\vec{v}_i$	Linear velocity of link i
$\vec{v}_{Gi}$	Linear velocity of the mass center of link i
V	nxn diagonal viscous friction matrix
$\vec{x}$	State vector
$(\vec{x}_i, \vec{y}_i, \vec{z}_i)$	Body attached coordinate system of link i
$(\vec{x}_o, \vec{y}_o, \vec{z}_o)$	Base (reference) coordinate system
α_i	Angle between the $\vec{z}_{i-1}$ and $\vec{z}_i$ axes measured in a right-hand sense about $\vec{x}_i$
$\vec{\alpha}_i$	Angular acceleration of link i
θ_i	Angle between the $\vec{x}_{i-1}$ and $\vec{x}_i$ axes measured in a right-hand sense about $\vec{z}_{i-1}$
$\vec{\tau}$	nx1 vector of input generalized forces
$\vec{\omega}_i$	Angular velocity of link i

I. INTRODUCTION

Industrial robots have become increasingly important in industrial automation in recent years. They are extensively used and are normally equipped with relatively simple control systems. Such control systems have proved adequate only at low speeds. Hence, there is a need for improved control techniques because of the increased demand on manipulator performance.

The first step in the development of a manipulator control law is the derivation of the dynamic model of the system. It becomes very difficult to derive the equation of motion analytically for a manipulator when it has more than three degrees of freedom. It is also necessary to develop a general method for the dynamic modelling of a manipulator, thus allowing to analyze various manipulator models. Those reasons motivated the researchers recently to propose various computer oriented algorithms for the real time computation of the manipulator dynamics. Those algorithms consists of recursive equations based on either a Lagrangian formulation [1,2], or a Newtonian formulation [3,4,5]. They have the common feature that, based on the information about the kinematic scheme of the mechanism, the algorithm calculates positions, velocities and accelerations, and derives differential equations of motion. Although those recursive equations are suitable for real time control, they don't give closed form differential equations that describe the dynamic behaviour of a manipulator.

Manipulators are basically positioning devices and the dynamic control of them involves the determination of the inputs for the actuators which operate at the joints so that a set of desired values for the positions and velocities of the manipulator are achieved. One of the difficulties in the control of a manipulator is that its equations are highly nonlinear and involve coupling among the multiple links. This complicates the design of a control system for high performances. Several methods are proposed for the control of a manipulator arm. The control schemes suggested include classical controllers [6,7] as well as optimum controllers [8,9]. Among the classical control methods the calculation of the torques for a nominal trajectory is presented in [6]. This control method requires a considerable amount of calculations and memory storage. Another of the early proposed techniques is the resolved rate control [7] in which the joint angle rates are computed so as to cause the end point of the manipulator to move in a definite direction. More recent studies apply optimum control theory to the manipulator control. There are also adaptive control strategies found in literature. The most notable of these is the model referenced adaptive control [10]. In this method the coupling terms between the joints are neglected and a linear second order time invariant model is used as the reference model for each degree of freedom. The manipulator is controlled by adjusting position and velocity feedback gains to follow the reference model.

In this work two general computer oriented algorithms are developed, which can be used to construct the dynamic model of any kind of open kinematic chain that consists of a combination of rotational and translational joints. With those algorithms it is possible to solve both the direct and inverse problem of mechanics. The derivation of the algorithms are explained in Chapter 2.

In Chapter 3 two different control schemes are proposed in order to control a manipulator. In both methods optimum control of a manipulator arm is investigated using an energy optimal performance index. In the first method the system is decoupled and the optimum control is found for each subsystem. In the second method a suboptimal adaptive feedback law is proposed to control the system.

Dynamic and static performances of the proposed algorithms have been tested on a typical manipulator configuration. A rectilinear trajectory in three dimensional space has been selected as the basic scenario and various computer simulations are carried out with different combinations of simulation parameters in order to emphasize the advantages and disadvantages of the proposed methods. The numerical simulation results are presented in Chapter 5.

Chapter 6, concluding this work, resumes the basic results and gives recommendations for further studies to improve the dynamic analysis and control of manipulators. The modelling algorithms have proven to be successful and the proposed control methods, in spite of the conservative behaviour of the minimum energy approach, have lead to satisfactory results.

II. MATHEMATICAL MODELLING OF A RIGID LINK SERIAL MANIPULATOR ARM

2.1 KINEMATICS OF THE MANIPULATOR ARM

A mechanical manipulator is an open loop chain which consists of a sequence of rigid bodies, called links, connected in series by kinematic joints. The joints allow relative motion of the two bodies they connect. One end of the chain is generally fastened to a support, while the other end is free to move in space.

In this study kinematic and dynamic equations are derived for a manipulator which has one degree of freedom joints which may be either revolute (rotating) or prismatic (sliding). Each joint-link pair constitutes one degree of freedom. Thus for an n degree of freedom manipulator there will be n links and n joints. The joints and links are numbered outwards, starting from the base, which is taken as link 0, to the end effector of the manipulator which is link n. Joint i is the joint which connects link i-1 to link i.

In order to develop a systematic and generalized method for the derivation of the kinematic and dynamic equations of a manipulator arm a body coordinate frame $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ is attached to each link i. Adjacent coordinate frames are related to each other by four parameters developed by Denavit and Hartenberg [11]. Those parameters are defined as follows (shown in Fig. 2.1):

θ_i = the joint angle from $\vec{x}_{i-1}$ axis to the $\vec{x}_i$ axis about the $\vec{z}_{i-1}$ axis (using the righthand rule).

d_i = the distance from the origin of the (i-1)th coordinate frame to the intersection of the $\vec{z}_{i-1}$ axis with the $\vec{x}_i$ axis along the $\vec{z}_{i-1}$ axis.

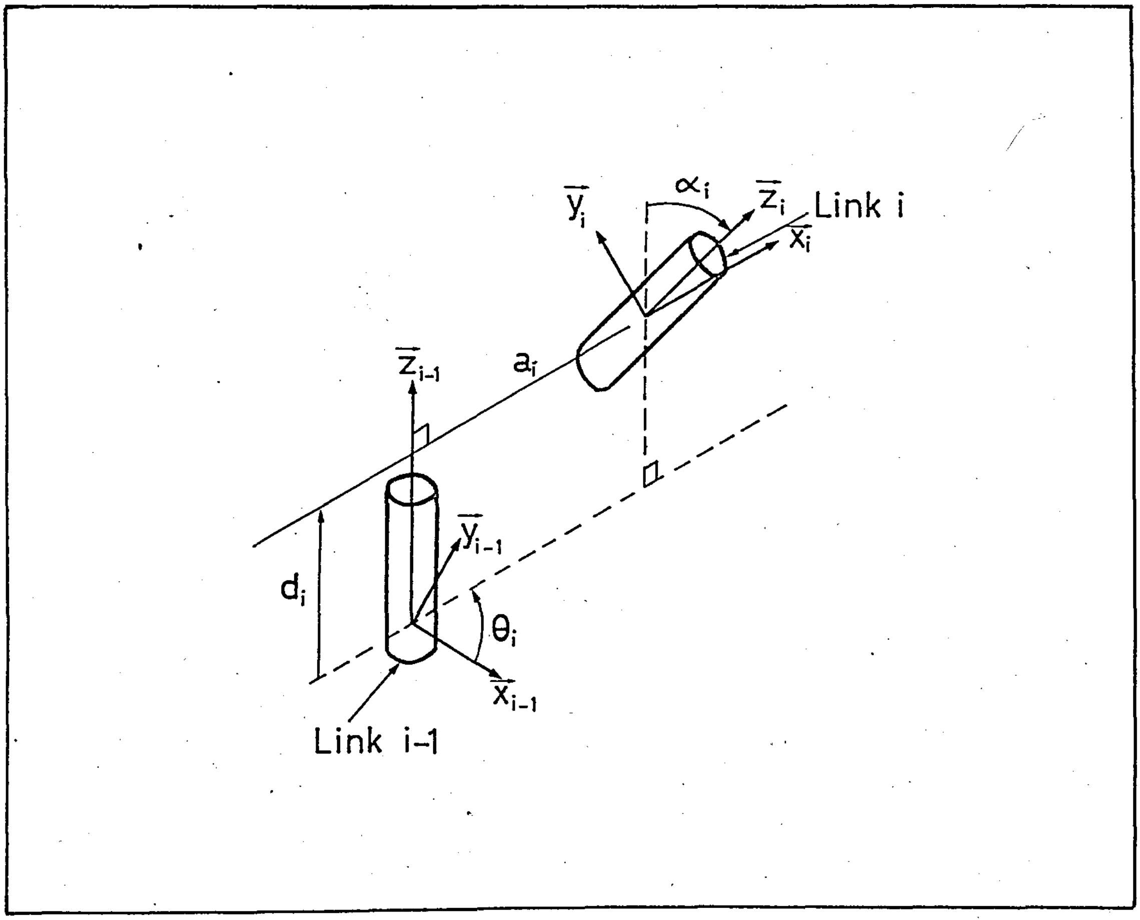


FIGURE 2.1 - Parameters relating adjacent coordinate systems.

a_i = the offset distance from the intersection of the $\vec{z}_{i-1}$ axis with the $\vec{x}_i$ axis to the origin of the ith frame along the $\vec{x}_i$ axis (or shortest distance between the $\vec{z}_{i-1}$ and $\vec{z}_i$ axes).

α_i = the offset angle from the $\vec{z}_{i-1}$ axis to the $\vec{z}_i$ axis about the $\vec{x}_i$ axis (using the righthand rule).

For a revolute joint, d_i, a_i and α_i are the joint parameters and remain constants, while θ_i is the joint variable that changes when link i rotates with respect to link i-1. For a prismatic joint, θ_i, a_i and α_i are the joint parameters and remain constants, while d_i is the joint variable. These parameters constitute a minimal sufficient set to determine the complete kinematic configuration of each link of the manipulator arm.

Every coordinate frame is determined and established on the basis of three rules [12]:

i) the $\vec{z}_{i-1}$ axis lies along the axis of motion of the ith joint;

ii) the $\vec{x}_i$ axis is normal to the $\vec{z}_{i-1}$ axis, pointing away from it;

iii) the $\vec{y}_i$ axis completes the righthand coordinate system as required.

With the above rules, reference frame $(\vec{x}_0,\vec{y}_0,\vec{z}_0)$ can be placed anywhere in the supporting base as long as the $\vec{z}_0$ axis lies along the axis of motion of the first joint. The last coordinate frame (nth frame) can be placed anywhere in the hand as long as the $\vec{x}_n$ axis is normal to the $\vec{z}_{n-1}$ axis. An algorithm for establishing consistent orthonormal coordinate systems for a manipulator is given in Appendix A.

In order to transform a vector expressed in the coordinate system of link i to the coordinate system of link i-1 homogeneous transformation matrices are used. Using Denavit-Hartenberg parameters in order to relate the two coordinate systems, the homogeneous transformation matrix that maps the coordinates of a vector expressed in the coordinate system $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ to the coordinate system $(\vec{x}_{i-1}, \vec{y}_{i-1}, \vec{z}_{i-1})$ is given as

$$A_{i-1}^{i} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

The inverse of the matrix A_{i-1}^{i} which is also used for the transformation of a vector from the (i-1)th frame to the ith frame is given by

$$[A_{i-1}^{i}]^{-1} = A_{i}^{i-1} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & \sin\alpha_i & -d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & -\sin\alpha_i \cos\theta_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

The homogeneous transformation matrix that will transform the coordinates of a vector from the coordinate system of link i to the coordinate system of link J is obtained by successive multiplication of the transformation matrices

$$A_J^i = A_J^{J+1} \cdots\cdots\cdots A_{i-2}^{i-1} \cdot A_{i-1}^i = \prod_{k=J}^{i} A_k^{k+1} \tag{2.3}$$

The upper left 3x3 submatrix of the homogeneous transformation matrix A_{i-1}^i represents the rotation matrix which is used extensively in the derivation of the dynamic model of the manipulator arm. The rotation matrix maps the coordinates of a vector from one coordinate system to another one whose origins are the same, but rotated with respect to each other. The rotation matrix for the transformation from link i coordinates to link i-1 coordinates is given by

$$R_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{2.4}$$

Since the transformation given by Eq. (2.4) is an orthonormal transformation, the inverse of the rotation matrix is equal to its transpose

$$R_i^{i-1} = (R_{i-1}^i)^T \tag{2.5}$$

A homogeneous transformation matrix geometrically represents the location (position and orientation) of a rotated coordinate system with respect to a reference frame. Given a reference frame $(\vec{x}_o, \vec{y}_o, \vec{z}_o)$ and a homogeneous transformation matrix A_0^i the column vectors of the rotation submatrix represent the principal axes of the coordinate system $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ with respect to the reference frame.

The fourth column of the homogeneous transformation matrix represent the position of the origin of the ith frame with respect to reference frame as given by

$$A_o^i = \begin{bmatrix} \vec{x}_i & \vec{y}_i & \vec{z}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

If n is substituted for i in Eq. (2.6) the obtained matrix is called the arm matrix which completely specifies the position and orientation of the hand with respect to the referene frame

$$H = A_o^n = \begin{bmatrix} \vec{x}_n & \vec{y}_n & \vec{z}_n & \vec{p}_n \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

where

$\vec{n}$ = unit normal vector of the hand,

$\vec{s}$ = unit sliding vector of the hand,

$\vec{a}$ = unit approach vector of the hand,

$\vec{p}$ = position vector of the hand.

All these vectors are defined with reference to base coordinates as shown in Fig. 2.2.

Once the link coordinate frames have been assigned to the manipulator it is possible to obtain the cartesian position and orientation of the manipulator end effector with given joint coordinates using Eq. (2.3). This is called the "direct kinematics solution" as given by

$$H = A_0^n = A_0^1 \cdot A_1^2 \,\ldots\ldots\, A_{n-1}^n \tag{2.8}$$

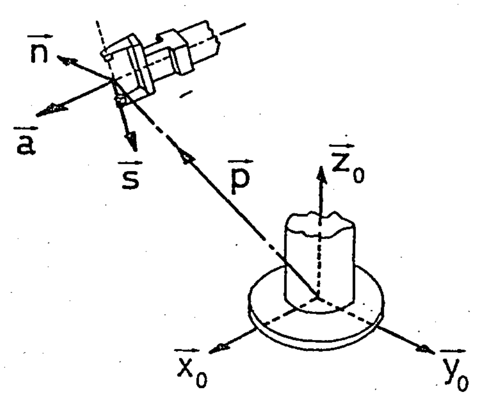


FIGURE 2.2 - Position and orientation vectors of the hand.

The control of a manipulator necessitates the inverse kinematics solution. For the solution of the inverse kinematics problem it is necessary to find the required joint coordinates given the desired position and orientation of the hand. The existence of an explicit solution to the kinematic equations for any manipulator is of great importance in evaluating the manipulator's suitability for computer control.

2.2 DYNAMICS OF THE MANIPULATOR ARM

2.2.1 Introduction

The first step for manipulator design and control is to derive its dynamic model. For manipulators with only two or three degrees of freedom the dynamic equations of the system can be derived manually. For manipulators with more than three degrees of freedom the equations are so complex that it becomes very difficult to derive them by hand. Then it is required to form such an algorithm which could automatically compose the dynamic equations, based only on input data on mechanism parameters. This helps to eliminate the problem of committing errors when forming the model by hand.

The following requirements should be considered in order to find an efficient method for manipulator modelling:

i) The method to derive the equations of motion should not be very complex, it should be easy to formulate.

ii) The model must be accurate enough to give results which satisfactorily describe the operation of the actual system yet simple enough to be of practical use for both design and real time control.

iii) The system equations must be solveable in a short time for on-line control purposes and computational efficiency. This is required also to reduce the cost of simulation.

iv) The method should solve both direct and inverse problems of dynamics. That is, given the motion of the mechanism members it should compute the necessary torques to ensure that motion, or given torques it should calculate the accelerations.

v) The method has to be general. Given only the system parameters as input, it should give all the information needed for the study of the system.

vi) The dynamic model should consider the constraints of the system. It should provide information to predict and prevent actuator overloads.

2.2.2 A General Computer Oriented Algorithm for Dynamic Modelling of a Manipulator

There are various recursive algorithms proposed in the literature for dynamic modelling of manipulators [1-5]. They generally calculate the velocities and accelerations of each link by a forward recursion starting from the base of the manipulator to the end link. Then the generalized forces are obtained by a backward recursion from the end link to the base of the manipulator. In this section a general algorithm for dynamic modelling of an n degree of freedom manipulator is developed using Newton Euler's formulation. Rather than calculating the velocities and accelerations directly, their corresponding coefficients are calculated as proposed in [3]. Thus a dynamic model of the system is obtained which can be used for the solution of both direct and inverse problem of mechanics. Also the velocities and accelerations (or corresponding coefficients) are calculated in each link's internal coordinate system as suggested in [4], thus eliminating the need for a great deal of coordinate transformations.

The dynamic model of an n degree of freedom manipulator is represented by a set of n nonlinear differential equations which describe the motion of the system in the space of internal (joint) coordinates. The equations of motion can be generated in various forms depending on whether the information on dynamics is necessary for dynamic analysis of the system or for the synthesis of control algorithms and the simulation of particular control laws.

In general, the equation of motion for an n degree of freedom manipulator can be written as:

$$\vec{\tau} = J(\vec{q})\ddot{\vec{q}} + V\dot{\vec{q}} + \vec{f}(\dot{q}_i\dot{q}_j,\vec{q}) + \vec{g}(\vec{q}) + \vec{h}\,(\vec{q}) \tag{2.9}$$

where

$\vec{q}$: nx1 vector of joint variables,

$J(\vec{q})$: nxn symmetric, nonsingular moment of inertia matrix,

V : nxn diagonal viscous friction matrix,

$f(\dot{q}_i\dot{q}_j,\vec{q})$: nx1 vector specifying Coriolis and centrifugal effects, $(i,j = 1,2,\ldots,n)$,

$\vec{g}(\vec{q})$: nx1 vector specifying the effects due to gravity,

$\vec{h}(\vec{q})$: nx1 vector specifying the effects due to external forces and moments exerted on link n,

$\vec{\tau}$: nx1 vector of input generalized forces.

In this section a method is developed to form the matrices J and V, and the vectors $\vec{f}$, $\vec{g}$ and $\vec{h}$ assuming that the vectors $\vec{q}$ and $\dot{\vec{q}}$

are known. In this method a recursive formulation based on Newton-Euler dynamics is used. First the velocities and accelerations of all the links of the manipulator are obtained by a forward recursion starting from the base of the manipulator to the end link. Generalized forces are then calculated by a backward recursion starting from the end link to the base of the manipulator.

There is one coordinate system attached to each link of the manipulator which moves together with the link. Considering the three coordinate systems as shown in Fig. 2.3, one can obtain the vector equation

$$\vec{p}_{i+1} = \vec{p}_i + \vec{p}^{\,*}_{i+1} \tag{2.10}$$

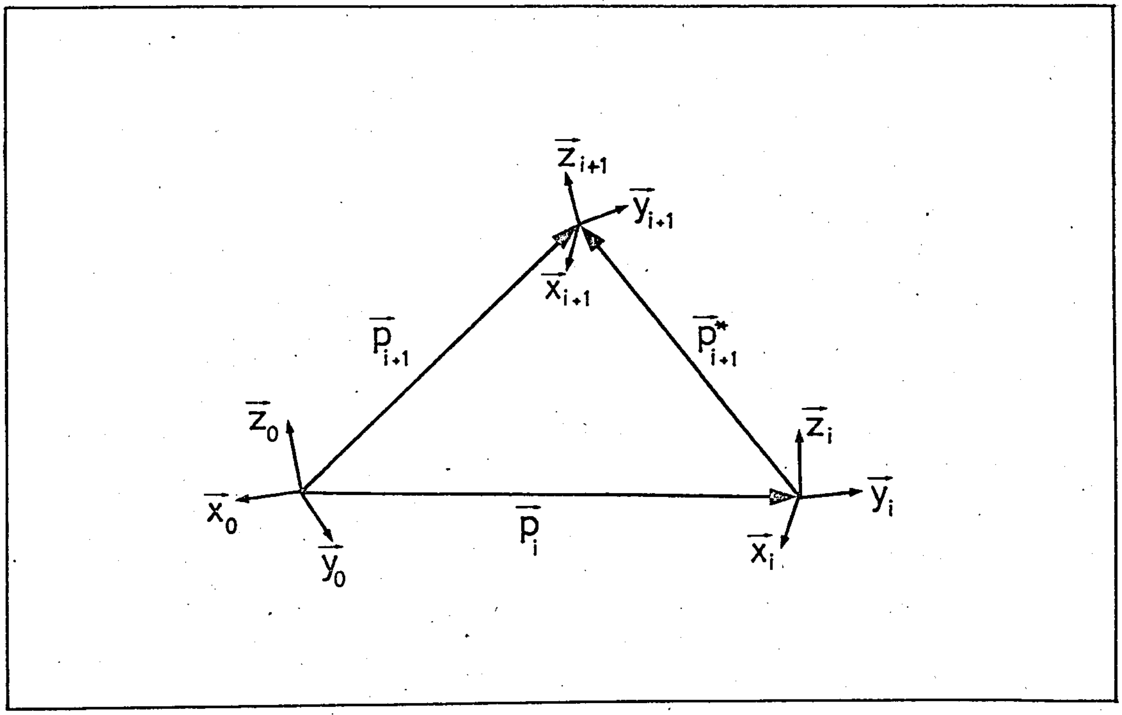


FIGURE 2.3 - Relationship between link coordinate systems.

The linear velocity of the coordinate system of link i+1 with respect to the base coordinates is obtained by the differentiation of Eq. (2.10),

$$\vec{v}_{i+1} = \vec{v}_i + \vec{\omega}_i \times \vec{p}^*_{i+1} + \left(\frac{d\vec{p}^*_{i+1}}{dt}\right)_i \tag{2.11}$$

where "x" sign denotes the cross product, and

$\vec{\omega}_i$: angular velocity of ith frame with respect to the base frame

$\left(\frac{d\vec{p}^*_{i+1}}{dt}\right)_i$: rate of change of $\vec{p}^*_{i+1}$ with respect to the ith frame

Differentiating Eq. (2.10) once more gives the linear acceleration of ith frame with respect to the base frame

$$\vec{a}_{i+1} = \vec{a}_i + (\vec{\alpha}_i \times \vec{p}^*_{i+1}) + \vec{\omega}_i \times (\vec{\omega}_i \times \vec{p}^*_{i+1}) + 2\vec{\omega}_i \times \left(\frac{d\vec{p}^*_{i+1}}{dt}\right)_i + \left(\frac{d^2\vec{p}^*_{i+1}}{dt^2}\right)_i \tag{2.12}$$

where

$\vec{\alpha}_i$: angular acceleration of ith frame with respect to the base frame.

The third and fourth terms on the right side of Eq. (2.12) represent the centrifugal and Coriolis accelerations respectively. The angular velocity of link i+1 with respect to the base coordinates is given by

$$\vec{\omega}_{i+1} = \vec{\omega}_i + \vec{\omega}^*_{i+1} \tag{2.13}$$

where $\vec{\omega}^*_{i+1}$ is the angular velocity of (i+1)th frame with respect to the ith frame. The angular acceleration of (i+1)th coordinate frame is obtained by differentiating Eq. (2.13)

$$\vec{\alpha}_{i+1} = \vec{\alpha}_i + (\vec{\omega}_i \times \vec{\omega}^*_{i+1}) + \left(\frac{d\vec{\omega}^*_{i+1}}{dt}\right)_i \tag{2.14}$$

Equations (2.11) to (2.14) are the basic relations for velocities and accelerations between links and the base of the manipulator.

If link i+1 is translational in coordinates $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ it travels in the direction $\vec{z}_i$ with a linear velocity $\dot{q}_{i+1}$ relative to link i. If it is rotational in coordinates $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$ it rotates about $\vec{z}_i$ axis with an angular velocity $\dot{q}_{i+1}$ relative to link i due to the definition of link frames according to the rules given in Section 2.1. Hence, one obtains

$$\vec{\omega}^*_{i+1} = \begin{cases} \vec{z}_i \dot{q}_{i+1}, & \text{if link i+1 is rotational} \\ \vec{0}, & \text{if link i+1 is translational} \end{cases} \tag{2.15}$$

$$\left(\frac{d\vec{\omega}^*_{i+1}}{dt}\right)_i = \begin{cases} \vec{z}_i \ddot{q}_{i+1}, & \text{if link i+1 is rotational} \\ \vec{0}, & \text{if link i+1 is translational} \end{cases} \tag{2.16}$$

$$\left(\frac{d\vec{p}^*_{i+1}}{dt}\right)_i = \begin{cases} \vec{\omega}^*_{i+1} \times \vec{p}^*_{i+1}, & \text{if link i+1 is rotational} \\ \vec{z}_i \dot{q}_{i+1}, & \text{if link i+1 is translational} \end{cases} \tag{2.17}$$

$$\left(\frac{d^2\vec{p}^*_{i+1}}{dt^2}\right)_i = \begin{cases} \left(\frac{d\vec{\omega}^*_{i+1}}{dt}\right)_i \times \vec{p}^*_{i+1} + \vec{\omega}^*_{i+1} \times (\vec{\omega}^*_{i+1} \times \vec{p}^*_{i+1}), & \text{if link i+1 is rotational} \\ \vec{z}_i \ddot{q}_{i+1}, & \text{if link i+1 is translational} \end{cases} \tag{2.18}$$

Combining Eqs. (2.11-18) yields the following recursive equations for angular and linear velocities and accelerations:

$$\vec{\omega}_{i+1} = \begin{cases} \vec{\omega}_i + \vec{z}_i\dot{q}_{i+1} & \text{, if link i+1 is rotational} \\ \vec{\omega}_i & \text{, if link i+1 is translational} \end{cases} \tag{2.19}$$

$$\vec{\alpha}_{i+1} = \begin{cases} \vec{\alpha}_i + \vec{z}_i\ddot{q}_{i+1} + \vec{\omega}_i \times (\vec{z}_i\dot{q}_{i+1}) & \text{, if link i+1 is rotational} \\ \vec{\alpha}_i & \text{, if link i+1 is translational} \end{cases} \tag{2.20}$$

$$\vec{v}_{i+1} = \begin{cases} \vec{v}_i + \vec{\omega}_{i+1}\times\vec{p}^{\,*}_{i+1} & \text{, if link i+1 is rotational} \\ \vec{v}_i + \vec{\omega}_{i+1}\times\vec{p}^{\,*}_{i+1} + \vec{z}_i\dot{q}_{i+1} & \text{, if link i+1 is translational} \end{cases} \tag{2.21}$$

$$\vec{a}_{i+1} = \begin{cases} \vec{a}_i + \vec{\alpha}_{i+1}\times\vec{p}^{\,*}_{i+1} + \vec{\omega}_{i+1}\times(\vec{\omega}_{i+1}\times\vec{p}^{\,*}_{i+1}) & \text{, if link i+1 is rotational} \\ \vec{a}_i + \vec{\alpha}_{i+1}\times\vec{p}^{\,*}_{i+1} + \vec{\omega}_{i+1}\times(\vec{\omega}_{i+1}\times\vec{p}^{\,*}_{i+1}) + 2\vec{\omega}_{i+1}\times(\vec{z}_i\dot{q}_{i+1})+\vec{z}_i\ddot{q}_{i+1} & \text{, if link i+1 is translational} \end{cases} \tag{2.22}$$

In order to obtain the velocity and acceleration of the mass center of link i, $\vec{s}_i$ is substituted instead of $\vec{p}^{\,*}_{i+1}$ in Eqs. (2.11) and (2.12)

$$\vec{v}_{Gi} = \vec{v}_i + \vec{\omega}_i \times \vec{s}_i \tag{2.23}$$

$$\vec{a}_{Gi} = \vec{a}_i + \vec{\alpha}_i \times \vec{s}_i + \vec{\omega}_i \times (\vec{\omega}_i\times\vec{s}_i) \tag{2.24}$$

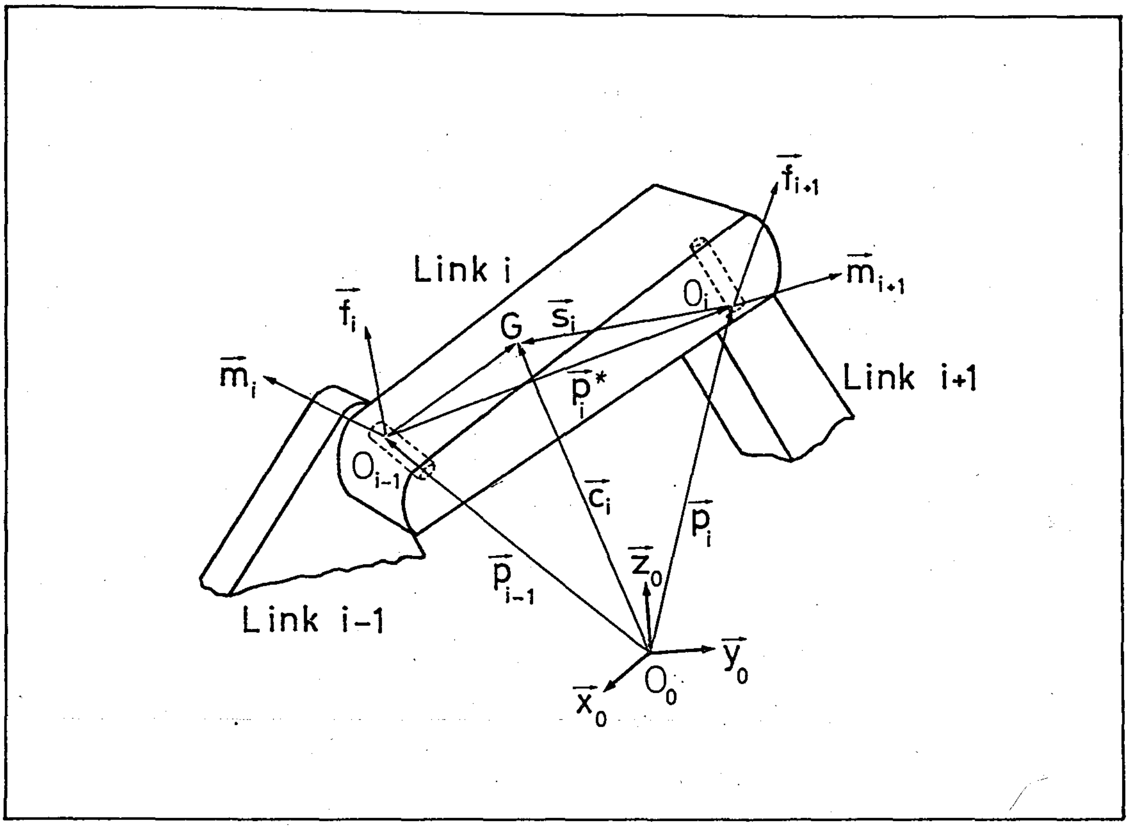


FIGURE 2.4 - Vectors related with the links.

The equations of motion for link i can be obtained by D'Alembert's Principle as

$$\vec{F}_i = m_i \vec{a}_{Gi} \tag{2.25}$$

$$\vec{M}_i = I_i \vec{\alpha}_i + \vec{\omega}_i \times (I_i \vec{\omega}_i) \tag{2.26}$$

where

$\vec{F}_i$: total external force vector exerted on link i

$\vec{M}_i$: total external moment vector exerted on link i

m_i : total mass of link i

I_i : inertia matrix of link i about its center of mass in $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$.

The equations of motion derived for link i are referenced to the base coordinate system. However the inertia matrix is dependent on the orientation of link i, which is changing. Thus the computation is quite complicated. A more efficient technique for determination of the equations of motion is to have each link's dynamics referenced to its own link coordinate system as suggested in [4]. The end result of referencing the dynamics to the link coordinates is to obviate a great deal of coordinate transformation and to allow the inertia matrix to be fixed in each link coordinate system. Hence, Eqs. (2.19-24) are rewritten in terms of each link's internal coordinate system using rotation matrices.

$$R^{o}_{i+1}\vec{\omega}_{i+1} = \begin{cases} R^{i}_{i+1}(R^{o}_{i}\vec{\omega}_i + \vec{z}_o\dot{q}_{i+1}) & \text{, if link i+1 is rotational} \\ R^{i}_{i+1}(R^{o}_{i}\vec{\omega}_i) & \text{, if link i+1 is translational} \end{cases} \tag{2.27}$$

where $\vec{z}_o$ is given by $\vec{z}_o = (0\ 0\ 1)^T$.

$$R^{o}_{i+1}\vec{\alpha}_{i+1} = \begin{cases} R^{i}_{i+1}(R^{o}_{i}\vec{\alpha}_i + \vec{z}_o\ddot{q}_{i+1} + R^{o}_{i}\vec{\omega}_i \times \vec{z}_o\dot{q}_{i+1}) & \text{, if link i+1 is rotational} \\ R^{i}_{i+1}(R^{o}_{i}\vec{\alpha}_i) & \text{, if link i+1 is translational} \end{cases} \tag{2.28}$$

$$R^{o}_{i+1}\vec{v}_{i+1} = \begin{cases} R^{i}_{i+1}(R^{o}_{i}\vec{v}_i) + R^{o}_{i+1}\vec{\omega}_{i+1} \times R^{o}_{i+1}\vec{p}^{*}_{i+1} & \text{, if link i+1 is rotational} \\ R^{i}_{i+1}(R^{o}_{i}\vec{v}_i + \vec{z}_o\dot{q}_{i+1}) + R^{o}_{i+1}\vec{\omega}_{i+1} \times R^{o}_{i+1}\vec{p}^{*}_{i+1} & \text{, if link i+1 is translational} \end{cases} \tag{2.29}$$

$$R^{0}_{i+1}\vec{a}_{i+1} = \begin{cases} R^{i}_{i+1}(R^{0}_{i}a_{i}) + R^{0}_{i+1}\vec{\alpha}_{i+1} \times R^{0}_{i+1}\vec{p}^{*}_{i+1} + R^{0}_{i+1}\vec{\omega}_{i+1}(R^{0}_{i+1}\vec{\omega}_{i+1} \times R^{0}_{i+1}\vec{p}^{*}_{i+1}), \\ \qquad \text{if link } i+1 \text{ is rotational} \\ R^{i}_{i+1}(R^{0}_{i}\vec{a}_{i} + \vec{z}_{0}\ddot{q}_{i+1}) + R^{0}_{i+1}\vec{\alpha}_{i+1} \times R^{0}_{i+1}\vec{p}^{*}_{i+1} + R^{0}_{i+1}\vec{\omega}_{i+1} \times (R^{0}_{i}\vec{\omega}_{i+1} \times R^{0}_{i+1}\vec{p}^{*}_{i+1}) \\ \quad + 2R^{0}_{i+1}\vec{\omega}_{i+1} \times R^{i}_{i+1}\vec{z}_{0}\dot{q}_{i+1}, \text{ if link } i+1 \text{ is translational} \end{cases} \tag{2.30}$$

where $R^{0}_{i}\vec{p}^{*}_{i} = (a_{i}, r_{i}\sin\alpha_{i}, r_{i}\cos\alpha_{i})^{T}$.

The linear velocity and acceleration of mass center of link i is given by

$$R^{0}_{i}\vec{v}_{Gi} = R^{0}_{i}\vec{v}_{i} + R^{0}_{i}\vec{\omega}_{i} \times R^{0}_{i}\vec{s}_{i} \tag{2.31}$$

$$R^{0}_{i}\vec{a}_{Gi} = R^{0}_{i}\vec{a}_{i} + R^{0}_{i}\vec{\alpha}_{i} \times R^{0}_{i}\vec{s}_{i} + R^{0}_{i}\vec{\omega}_{i} \times (R^{0}_{i}\vec{\omega}_{i} \times R^{0}_{i}\vec{s}_{i}) \tag{2.32}$$

where $R^{0}_{i}\vec{s}_{i}$ is the vector from the origin of the ith frame to the mass center of link i referred to link i coordinates (Fig. 2.4). Once coordinate systems are assigned to each link these vectors become the geometric parameters of the system.

Since the joint accelerations are not known, the expressions for angular and linear accelerations have to be written in a modified form in order to solve the inverse dynamics problem [3].

The angular acceleration of link i can be written as

$$R^{0}_{i}\vec{\alpha}_{i} = \sum_{j=1}^{i} \vec{\psi}_{ij}\ddot{q}_{j} + \vec{\theta}_{i} \tag{2.33}$$

where $\vec{\psi}_{ij}$ and $\vec{\theta}_{i}$ are angular acceleration coefficients. They are obtained by substituting Eq. (2.33) into Eq. (2.28). For a rotational link

$$\vec{\psi}_{ij} = R_i^{i-1}\vec{\psi}_{i-1,j} \qquad \text{for } 1 \leq j \leq i-1$$

$$\vec{\psi}_{ii} = R_i^{i-1}\vec{z}_o \tag{2.34}$$

$$\vec{\theta}_i = R_i^{i-1}\vec{\theta}_{i-1} + R_i^{i-1}(R_{i-1}^{o}\vec{\omega}_{i-1} \times \vec{z}_o\dot{q}_i)$$

For a translational link

$$\vec{\psi}_{ij} = R_i^{i-1}\vec{\psi}_{i-1,j} \qquad \text{for } 1 \leq j \leq i-1$$

$$\vec{\psi}_{ii} = \vec{0} \tag{2.35}$$

$$\vec{\theta}_i = R_i^{i-1}\vec{\theta}_{i-1}$$

The linear acceleration of each link can also be written as

$$R_i^{o}\vec{a}_i = \sum_{j=1}^{i} \vec{\beta}_{ij}\ddot{q}_j + \vec{\eta}_i \tag{2.36}$$

where $\vec{\beta}_{ij}$ and $\vec{\eta}_i$ are linear acceleration coefficients. Substituting equations (2.33) and (2.36) into Eq. (2.30) yields the recursive equations for $\vec{\beta}_{ij}$ and $\vec{\eta}_i$ for a rotational link as

$$\vec{\beta}_{ij} = R_i^{i-1}\vec{\beta}_{i-1,j} + \vec{\psi}_{ij} \times R_i^{o}\vec{p}_i^{*} \qquad \text{for } 1 \leq j \leq i-1$$

$$\vec{\beta}_{ii} = \vec{\psi}_{ii} \times R_i^{o}\vec{p}_i^{*} \tag{2.37}$$

$$\vec{\eta}_i = R_i^{i-1}\vec{\eta}_{i-1} + \vec{\theta}_i \times R_i^{o}\vec{p}_i^{*} + R_i^{o}\vec{\omega}_i \times (R_i^{o}\vec{\omega}_i \times R_i^{o}\vec{p}_i^{*})$$

and for a translational link

$$\vec{\beta}_{ij} = R_i^{i-1}\vec{\beta}_{i-1,j} + \vec{\psi}_{ij} \times R_i^0\vec{p}_i^* \qquad \text{for} \quad 1 \le j \le i-1$$

$$\vec{\beta}_{ii} = \vec{\psi}_{ii} \times R_i^0\vec{p}_i^* + R_i^{i-1}\vec{z}_o \tag{2.38}$$

$$\vec{\eta}_i = R_i^{i-1}\vec{\eta}_{i-1} + \vec{\theta}_i \times R_i^0\vec{p}_i^* + R_i^0\vec{\omega}_i \times (R_i^0\vec{\omega}_i \times R_i^0\vec{p}_i^*)$$

$$+ 2R_i^0\vec{\omega}_i \times R_i^{i-1}\vec{z}_o\dot{q}_i$$

The linear acceleration of mass center of link i can also be represented as

$$R_i^0\vec{a}_{Gi} = \sum_{j=1}^{i} \vec{\lambda}_{ij}\ddot{q}_j + \vec{\gamma}_i \tag{2.39}$$

The recursive equations for vector coefficients $\vec{\lambda}_{ij}$, and $\vec{\gamma}_i$ are obtained by substituting Eqs. (2.33), (2.36) and (2.39) into Eq. (2.32)

$$\vec{\lambda}_{ij} = \vec{\psi}_{ij} \times R_i^0\vec{s}_i + \vec{\beta}_{ij} \qquad \text{for} \quad 1 \le j \le i$$

$$(2.40)$$

$$\vec{\gamma}_i = \vec{\theta}_i \times R_i^0\vec{s}_i + R_i^0\vec{\omega}_i \times (R_i^0\vec{\omega}_i \times R_i^0\vec{s}_i) + \vec{\eta}_i$$

Equations of motion for link i can be written in coordinates of ith frame as

$$R_i^0\vec{F}_i = m_i R_i^0\vec{a}_{Gi} \tag{2.41}$$

$$R_i^0\vec{M}_i = \overline{I}_i R_i^0\vec{\alpha}_i + R_i^0\vec{\omega}_i \times (\overline{I}_i R_i^0\vec{\omega}_i) \tag{2.42}$$

where $\overline{I}_i$ is the inertia matrix of link i about its center of mass referred to its own coordinates $(\vec{x}_i,\vec{y}_i,\vec{z}_i)$.

Substituting Eqs. (2.33) and (2.39) into Eqs. (2.41) and (2.42) one gets

$$A_i^O \vec{F}_i = L_i \ddot{\vec{q}} + \vec{\ell}_i \tag{2.43}$$

$$A_i^O \vec{M}_i = N_i \ddot{\vec{q}} + \vec{n}_i \tag{2.44}$$

where L_i and N_i are 3xn matrices given as

$$L_i = m_i [\vec{\lambda}_{i1} \quad \vec{\lambda}_{i2} \quad \cdots\cdots \quad \vec{\lambda}_{ii} \quad \vec{0} \quad \cdots\cdots \quad \vec{0}]$$

$$N_i = \overline{I}_i [\vec{\psi}_{i1} \quad \vec{\psi}_{i2} \quad \cdots\cdots \quad \vec{\psi}_{ii} \quad \vec{0} \quad \cdots\cdots \quad \vec{0}]$$

and the vectors $\vec{\ell}_i$ and $\vec{n}_i$ are defined as

$$\vec{\ell}_i = m_i \vec{\gamma}_i$$
$$\vec{n}_i = \overline{I}_i \vec{\theta}_i + R_i^O \vec{\omega}_i \times (\overline{I}_i R_i^O \vec{\omega}_i)$$

Considering the forces and moments acting on link i as shown in Fig. 2.4, one gets the equations

$$\vec{F}_i = \vec{f}_i - \vec{f}_{i+1} + m_i \vec{g} \tag{2.45}$$

$$\vec{M}_i = \vec{m}_i - \vec{m}_{i+1} + (\vec{p}_{i-1} - \vec{c}_i) \times \vec{f}_i - (\vec{p}_i - \vec{c}_i) \times \vec{f}_{i+1} \tag{2.46}$$

where

$\vec{f}_i$: force exerted on link i by link i-1

$\vec{m}_i$: moment exerted on link i by link i-1

$\vec{g}$: gravitational acceleration

Since $\vec{c}_i - \vec{p}_{i-1} = \vec{p}_i^* + \vec{s}_i$ as seen from Fig. 2.4, the above equations lead to the following recursive relations for the reaction forces and moments:

$$\vec{f}_i = \vec{f}_{i+1} + \vec{F}_i - m_i\vec{g} \tag{2.47}$$

$$\vec{m}_i = \vec{m}_{i+1} + \vec{p}_i^* \times \vec{f}_{i+1} + (\vec{p}_i^* + \vec{s}_i) \times \vec{F}_i - (\vec{p}_i^* + \vec{s}_i) \times m_i\vec{g} + \vec{M}_i \tag{2.48}$$

Rewriting equations (2.47) and (2.48) in the internal coordinate system of link i gives

$$R_i^0\vec{f}_i = R_i^{i+1}(R_{i+1}^0\vec{f}_{i+1}) + R_i^0\vec{F}_i - R_i^0 m_i\vec{g} \tag{2.49}$$

$$R_i^0\vec{m}_i = R_i^{i+1}[R_{i+1}^0\vec{m}_{i+1} + (R_{i+1}^0\vec{p}_i^*) \times (R_{i+1}^0\vec{f}_{i+1})] + (R_i^0\vec{p}_i^* + R_i^0\vec{s}_i) \times (R_i^0\vec{F}_i - R_i^0 m_i\vec{g}) + R_i^0\vec{M}_o \tag{2.50}$$

All the unknown reaction forces and moments can be calculated using equations (2.49) and (2.50) by a backward recursion starting from link n to the base of the manipulator. For an n degree of freedom manipulator $\vec{f}_{n+1}$ and $\vec{m}_{n+1}$ are, respectively the force and moment exerted by link n upon an external object.

Substituting Eq. (2.43) into Eq. (2.49), one obtains an expression for the reaction forces which is suitable for deriving the equation of motion of the manipulator.

$$R_i^0\vec{f}_i = T_i\ddot{\vec{q}} + \vec{t}_i - \vec{g}_{fi} + \vec{f}_{Hi} \tag{2.51}$$

where the elements of this equation are given by the recursive relations:

$$T_i = R_i^{i+1} T_{i+1} + L_i \tag{2.52}$$

$$\vec{t}_i = R_i^{i+1} \vec{t}_{i+1} + \vec{\ell}_i \tag{2.53}$$

$$\vec{g}_{fi} = R_i^{i+1} \vec{g}_{f_{i+1}} + R_i^o m_i \vec{g} \tag{2.54}$$

First terms of the above equations are omitted for i=n.

$$\vec{f}_{Hi} = R_i^{i+1} \vec{f}_{H_{i+1}} \tag{2.55}$$

For $i = n$ $\vec{f}_{Hi}$ is given by $\vec{f}_{Hn} = R_n^o \vec{f}_{n+1}$.

Similarly the reaction moments are also obtained by substituting Eq. (2.44) into Eq. (2.50)

$$R_i^o \vec{m}_i = S_i \ddot{\vec{q}} + \vec{s}_i' - \vec{g}_{mi} + \vec{m}_{Hi} \tag{2.56}$$

The elements of Eq. (2.56) are given by the recursive relations:

$$S_i = R_i^{i+1} S_{i+1} + C_i + D_i + N_i \tag{2.57}$$

where the kth column of the matrices C_i and D_i are given as

$$C_i^k = R_i^o \vec{p}_i^* \times (R_i^{i+1} T_{i+1})^k \tag{2.58}$$

$$D_i^k = (R_i^o \vec{p}_i^* + R_i^o s_i) \times L_i^k \tag{2.59}$$

where the superscript k indicates the kth column of the matrices involved.

$$\vec{s}_i' = R_i^{i+1} \vec{s}_{i+1}' + R_i^o \vec{p}_i^* \times R_i^{i+1} \vec{t}_{i+1} + (R_i^o \vec{p}_i^* + R_i^o \vec{s}_i) \times \vec{\ell}_i + \vec{n}_i \tag{2.60}$$

$$\vec{g}_{mi} = R_i^{i+1}\vec{g}_{m_{i+1}} + R_i^o\vec{p}_i^* \times R_i^{i+1}\vec{g}_{f_{i+1}} + (R_i^o\vec{p}_i^* + R_i^o\vec{s}_i) \times R_i^o m_i \vec{g} \tag{2.61}$$

The first two terms of equations (2.57), (2.60) and (2.61) are omitted for $i = n$.

$$\vec{m}_{Hi} = R_i^{i+1}\vec{m}_{H_{i+1}} + R_i^o\vec{p}_i^* \times \vec{f}_{Hi} \tag{2.62}$$

For $i = n$ $\vec{m}_{Hi}$ is given by $\vec{m}_{Hn} = R_n^o\vec{m}_{n+1} + R_n^o\vec{p}_n^* \times \vec{f}_{Hn}$.

The required force or torque at joint i that the actuator should provide is equal to the projection of the reaction force or moment at the joint ($\vec{f}_i$ or $\vec{m}_i$ according to whether the joint is translational or rotational) onto the $\vec{z}_{i-1}$ axis plus the friction force or moment. Hence,

$$\vec{\tau}_i = \begin{cases} [R_{i-1}^i(R_i^o\vec{m}_i)]^T\vec{z}_o + b_i\dot{q}_i \ , & \text{if link i+1 is rotational} \\ \\ [R_{i-1}^o(R_i^o\vec{f}_i)]^T\vec{z}_o + b_i\dot{q}_i \ , & \text{if link i+1 is translational} \end{cases} \tag{2.63}$$

Equations (2.51) and (2.56) are substituted into Eq. (2.63) in order to form the inertia matrix J, and the vectors $\vec{f}$, $\vec{g}$ and $\vec{h}$. Starting from the first link to the last link all elements of the inertia matrix and the vectors are formed by means of the expressions:

$$J_{ij} = \begin{cases} (R_{i-1}^i S_i)_{3j} \ , & \text{if link i is rotational} \\ \\ (R_{i-1}^i T_i)_{3j} \ , & \text{if link i is translational} \end{cases} \tag{2.64}$$

$$\text{for } i,j = 1,\ldots,n$$

where the subscripts i and j on M_{ij} indicates the corresponding element of matrix M.

$$\vec{f}_i = \begin{cases} (R^i_{i-1}\vec{s}_i)_3 & \text{, if link i is rotational} \\ \\ (R^i_{i-1}\vec{t}_i)_3 & \text{, if link i is translational} \end{cases} \tag{2.65}$$

$$\vec{g}_i = \begin{cases} (R^i_{i-1}\vec{g}_{mi})_3 & \text{, if link i is rotational} \\ \\ (R^i_{i-1}\vec{g}_{fi})_3 & \text{, if link i is translational} \end{cases} \tag{2.66}$$

$$\vec{h}_i = \begin{cases} (R^i_{i-1}\vec{m}_{Hi})_3 & \text{, if link i is rotational} \\ \\ (R^i_{i-1}\vec{f}_{Hi})_3 & \text{, if link i is translational} \end{cases} \tag{2.67}$$

where the subscript i on v_i indicates the ith component of the vector $\vec{v}$.

2.2.3 A Modified Method for Dynamic Modelling of a Manipulator

For purposes of control it may be desirable to have the equation of motion of the manipulator in the form

$$\vec{\tau} = J(\vec{q})\ddot{\vec{q}} + V\dot{\vec{q}} + C(\vec{q},\dot{\vec{q}})\dot{\vec{q}} + \vec{g}(\vec{q}) + \vec{h}(\vec{q}) \tag{2.68}$$

where C is an nxn matrix specifying Coriolis and centrifugal effects. In order to form C matrix it is necessary to reformulate the dynamic equations of the manipulator separating $\dot{q}_i$ terms. Therefore angular velocities are written in the form

$$R^o_i\vec{\omega}_i = \sum_{j=1}^{i} \vec{\xi}_{ij}\dot{q}_j \tag{2.69}$$

assuming zero velocity for the base frame. For the case of a rotational link the angular velocity coefficients are given as

$$\vec{\xi}_{ij} = R_i^{i-1}\vec{\xi}_{i-1,j} \qquad \text{for} \quad 1 \leq j \leq i-1$$

$$\vec{\xi}_{ii} = R_i^{i-1}\vec{z}_o \tag{2.70}$$

and for the case of a translational link

$$\vec{\xi}_{ij} = R_i^{i-1}\vec{\xi}_{i-1,j} \qquad \text{for} \quad 1 \leq j \leq i-1$$

$$\vec{\xi}_{ii} = \vec{0} \tag{2.71}$$

The only part of the equations for angular and linear velocities which contains $\dot{q}_i$ terms consist of the vectors $\vec{\theta}_i$, $\vec{\eta}_i$, and $\vec{\gamma}_i$ as given in equations (2.33), (2.36) and (2.39). Hence those vectors are rewritten separating the $\dot{q}_i$ terms. Assuming zero base velocity those vectors can be written as

$$\vec{\theta}_i = \sum_{j=1}^{i-1} \sum_{k=j}^{i-1} \vec{\kappa}_{ijk}\dot{q}_j\dot{q}_{k+1} \qquad \text{for} \quad i = 2,\ldots,n \tag{2.72}$$

where

$$\vec{\kappa}_{ijk} = R_i^{i-1}\vec{\kappa}_{i-1,jk}$$

$$\vec{\kappa}_{ijk} = \vec{\xi}_{jk} \times \vec{z}_o \qquad \text{for} \quad k = i-1 \tag{2.73}$$

$$\vec{\eta}_i = \sum_{j=1}^{i} \vec{\delta}_{ij}\dot{q}_j \tag{2.74}$$

For the case of a rotational link the coefficients $\vec{\delta}_{ij}$ are given as

$$\vec{\delta}_{ij} = \sum_{k=j}^{i-1} (\vec{\kappa}_{ijk} \times R_i^0 \vec{p}_i^*)\dot{q}_{k+1} + \sum_{k=1}^{i} (\vec{\xi}_{ij} \times \vec{\xi}'_{ik})\dot{q}_k + R_i^{i-1} \vec{\delta}_{i-1,j} \quad \text{for } 1 \leq j \leq i-1$$

$$\vec{\delta}_{ii} = \sum_{k=1}^{i} (\vec{\xi}_{ii} \times \vec{\xi}'_{ik})\dot{q}_k \tag{2.75}$$

For the case of a translational link

$$\vec{\delta}_{ij} = \sum_{k=j}^{i-1} (\vec{\kappa}_{ijk} \times R_i^0 \vec{p}_i^*)\dot{q}_{k+1} + \sum_{k=1}^{i} (\vec{\xi}_{ij} \times \vec{\xi}'_{ik})\dot{q}_k + 2(\vec{\xi}_{ij} \times R_i^{i-1} \vec{z}_o)\dot{q}_i$$

$$+ R_i^{i-1} \vec{\delta}_{i-1,j} \quad \text{for } 1 \leq j \leq i-1$$

$$\vec{\delta}_{ii} = \sum_{k=1}^{i} (\vec{\xi}_{ii} \times \vec{\xi}'_{ik})\dot{q}_k + 2(\vec{\xi}_{ii} \times R_i^{i-1} \vec{z}_o)\dot{q}_i \tag{2.76}$$

where $\vec{\xi}'_{ij} = \vec{\xi}_{ij} \times R_i^0 \vec{p}_i^*$.

$$\vec{\gamma}_i = \sum_{j=1}^{i} \vec{\phi}_{ij} \dot{q}_j \tag{2.77}$$

where the coefficients $\vec{\phi}_{ij}$ are given as

$$\vec{\phi}_{ij} = \sum_{k=j}^{i-1} (\vec{\kappa}_{ijk} \times R_i^0 \vec{s}_i)\dot{q}_{k+1} + \sum_{k=1}^{i} (\vec{\xi}_{ij} \times \vec{\xi}''_{ik})\dot{q}_k + \vec{\delta}_{ij}$$

$$\text{for } 1 \leq j \leq i-1$$

$$\vec{\phi}_{ii} = \sum_{k=1}^{i} (\vec{\xi}_{ii} \times \vec{\xi}''_{ik})\dot{q}_k + \vec{\delta}_{ii} \tag{2.78}$$

where $\vec{\xi}''_{ij} = \vec{\xi}_{ij} \times R_i^0 \vec{s}_i$.

The vectors $\vec{\ell}_i$, and $\vec{n}_i$ which appear in the equations for total forces and moments as given in Eqs. (2.44) and (2.45) are rewritten using Eqs. (2.72) and (2.76):

$$\vec{\ell}_i = \sum_{j=1}^{i} \vec{\phi}'_{ij}\dot{q}_j \tag{2.79}$$

where $\vec{\phi}'_{ij} = m_i\vec{\phi}_{ij}$

$$\vec{n}_i = \sum_{j=1}^{i} \vec{\mu}_{ij}\dot{q}_j \tag{2.80}$$

where

$$\vec{\mu}_{ij} = \sum_{k=j}^{i-1} \bar{I}_i\vec{\kappa}_{ijk}\dot{q}_{k+1} + \sum_{k=1}^{i} (\vec{\xi}_{ij} \times \bar{I}_i\vec{\xi}_{ik})\dot{q}_k \qquad \text{for} \quad 1 \le j \le i-1$$

$$\vec{\mu}_{ii} = \sum_{k=1}^{i} (\vec{\xi}_{ii} \times \bar{I}_i\vec{\xi}_{ik})\dot{q}_k \tag{2.81}$$

Substituting Eqs. (2.79) and (2.80) into Eqs. (2.53) and (2.57), one gets

$$\vec{t}_i = \sum_{j=1}^{n} \vec{\sigma}_{ij}\dot{q}_j \tag{2.82}$$

where

$$\vec{\sigma}_{ij} = \begin{cases} R_i^{i+1}\vec{\sigma}_{i+1,j} + \vec{\phi}'_{ij} , & \text{for} \quad 1 \le j \le i \\ \\ R_i^{i+i}\vec{\sigma}_{i+1,j} , & \text{for} \quad i < j \end{cases} \tag{2.83}$$

For $i = n$ $\quad \vec{\sigma}_{nj} = \vec{\phi}'_{nj}$.

$$\vec{s}_i' = \sum_{j=1}^{n} \vec{\nu}_{ij}\dot{q}_j \tag{2.84}$$

where

$$\vec{\nu}_{ij} = \begin{cases} R_i^{i+1}\vec{\nu}_{i+1,j} + R_i^0\vec{p}_i^* \times \vec{\sigma}_{i+1,j} + (R_i^0\vec{p}_i^* + R_i^0\vec{s}_i) \times \vec{\phi}_{ij}' + \vec{\mu}_{ij} \\ \qquad , \text{ for } 1 \le j \le i \\ R_i^{i+1}\vec{\nu}_{i+1,j} + R_i^0\vec{p}_i^* \times \vec{\sigma}_{i+1,j} \quad , \text{ for } i < j \end{cases} \tag{2.85}$$

For $i = n$ $\vec{\nu}_{nj} = (R_n^0\vec{p}_n^* + R_n^0\vec{s}_n) \times \vec{\phi}_{nj}' + \vec{\mu}_{nj}$. Using Eqs. (2.82) and (2.84) C matrix can be formed. The elements of the C matrix are given by

$$\vec{C}_{ij} = \begin{cases} (R_{i-1}^i\vec{\nu}_{ij})_3 & , \text{ if link } i \text{ is rotational} \\ \\ (R_{i-1}^i\vec{\sigma}_{ij})_3 & , \text{ if link } i \text{ is translational} \end{cases} \tag{2.86}$$

for $i,j = 1,\ldots,n$.

Two different algorithms have been developed in order to obtain the dynamic model of an n degree of freedom manipulator arm. Both algorithms consist of recursive equations, thus making them suitable for programming on a digital computer. They basically contain four stages to set up the equations of motion:

i) In the first stage the coefficients for angular and linear velocities and accelerations of the mass centers of each link is calculated in internal coordinates of links starting from the base to the end link.

ii) In the second stage total forces and total moments acting on each link are calculated. The values of linear and angular accelerations needed for this calculation have been determined in the first step.

iii) The reaction forces and moments acting on each link is determined by using D'Alembert's principle, starting from the end link to the base.

iv) The equation of motion of the manipulator is obtained using the equations found in step 3.

The main difference between the first and second algorithms is that the vector $\vec{f}$, which represents the effects of Coriolis and centrifugal forces, is obtained in an expanded form in the second algorithm as

$$\vec{f} = C(\vec{q},\dot{\vec{q}})\dot{\vec{q}}$$

Thus it is expected that more insight about the system behaviour can be obtained.

III. CONTROL OF THE MANIPULATOR ARM

The purpose of control of a manipulator arm is to maintain a prescribed motion for the arm along a desired trajectory by applying corrective compensation torques or forces to the actuators to adjust for any deviation of the arm from the trajectory. Since the dynamic model of an n degree of freedom manipulator arm consists of n highly coupled, nonlinear, second order differential equations, it is difficult to design a control system in order to achieve high performances. In this section, two simple control schemes are proposed which can be easily implemented and give satisfactory results. The two basic models, which are derived in Chapter 2, are used for the calculation of system matrices in order to obtain the feedback gains and the control vector. The first model is used when computing the control by the computed torque technique as described in Section 3.1. In Section 3.2 an adaptive control scheme is proposed where the second model is also used to consider the coupling terms due to Coriolis and centrifugal effects.

3.1 COMPUTED TORQUE TECHNIQUE

If the equation of motion of the manipulator arm (Eq. 2.9) is solved for joint accelerations, one obtains an equation of the form

$$\ddot{\vec{q}} = J^{-1}(\vec{q})[\vec{\tau} - \vec{k}] \tag{3.1}$$

where $\vec{k}$ is given by $\vec{k} = V\dot{\vec{q}} + \dot{\vec{f}}(\dot{q}_i\dot{q}_j,\vec{q}) = \vec{g}(\vec{q}) + \vec{h}(\vec{q})$. Eq. (3.1) can also be written as

$$\ddot{\vec{q}} = \vec{u} \quad , \quad \ddot{q}_i = u_i \quad , \quad i = 1,2,\ldots,n \tag{3.2}$$

where $\vec{u}$ is an nx1 control vector. Thus the system is decomposed into n subsystems, one for each degree of freedom. The manipulator is controlled using simple servo controllers that are closed separately around each degree of freedom, as given by the linear control equation

$$\vec{u} = K_p(\vec{q}_d - \vec{q}) + K_v(\dot{\vec{q}}_d - \dot{\vec{q}}) + \ddot{\vec{q}}_d \tag{3.3}$$

where $\vec{q}_d$, $\dot{\vec{q}}_d$ and $\ddot{\vec{q}}_d$ are the desired trajectories given in joint space, and K_p and K_v are nxn diagonal position and velocity feedback gain matrices. If Eq. (3.3) is written for joint i, one obtains the scalar equation

$$u_i = k_{p_i}(q_{d_i} - q_i) + k_{v_i}(\dot{q}_{d_i} - \dot{q}_i) + \ddot{q}_{d_i} \tag{3.4}$$

where k_{p_i} and k_{v_i} are the diagonal elements of the matrices K_p and K_v. Substituting Eq. (3.4) into Eq. (3.2) one gets an expression which describes the dynamic behaviour of the ith subsystem as

$$\ddot{q}_i + k_{v_i}\dot{q}_i + k_{p_i}q_i = k_{p_i}q_{d_i} + k_{v_i}\dot{q}_{d_i} + \ddot{q}_{d_i} \tag{3.5}$$

Hence, certain design requirements may be imposed upon the system by choosing appropriate values for k_{p_i} and k_{v_i}. After the control vector, u, is found,the generalized forces can be obtained in terms of the measured quantities, $\vec{q}$, and $\dot{\vec{q}}$ as

$$\vec{\tau} = J(\vec{q})[\ddot{\vec{q}}_d + K_p(\vec{q}_d - \vec{q}) + K_v(\dot{\vec{q}}_d - \dot{\vec{q}})] + \vec{k} \tag{3.6}$$

The first term of Eq. (3.6) will generate the desired torque or force for each joint if no modelling error exists and system parameters are known. However, errors due to uncertainty about system parameters, external disturbances, and time delay in the servo loop exist, making deviation from the desired trajectory inevitable. The remaining terms of Eq. (3.6) will generate the correction torque or force, depending on whether the joint is rotational or translational, to compensate for small deviations from the desired joint trajectory.

3.1.1 A Simple Adaptive Approach

If the equations of motion for the manipulator can be derived in a closed-form, then the coupling terms can be eliminated by mathematical manipulation of the equations, which is equivalent to a nonlinear transformation. This is done in [13] for a three degree of freedom manipulator arm where the position and velocity feedback gains are chosen to have a damping ratio of 0.8, and a natural frequency of 20 Hz for each link of the manipulator, and the system parameters are updated whenever their variation exceeds a predetermined tolerance value.

3.1.2 Minimum Energy Approach with Specified Final Time and State

Since Eq. (3.2) is being completely decoupled, it has a closed form solution for the optimal control problem. If the desired acceleration of each joint of the manipulator is known for a given time interval $[t_k, t_{k+1}]$ as

$$\ddot{\vec{q}}_{d_{k,k+1}} = \vec{\Gamma}_k \tag{3.7}$$

where $\vec{\Gamma}$ is given by $\vec{\Gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_n)^T$, then defining the state variables as follows

$$\begin{aligned} x_1 &\doteq q_d - q \quad , \\ x_2 &= \dot{q}_d - \dot{q} \; , \end{aligned} \tag{3.8}$$

one obtains the state model of the ith subsystem as

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u + \begin{bmatrix} 0 \\ \gamma_{ik} \end{bmatrix} , \quad t \in [t_k, t_{k+1}] \tag{3.9}$$

with the initial condition $\vec{x}(t_o) = \vec{x}_k$.

The optimal control problem is solved by using an energy optimal performance index because it leads to stabilizing feedback matrices, and the number of control parameters that should be chosen will be reduced. The performance index is given by

$$J = \frac{1}{2} \int_{t_k}^{t_{k+1}} \vec{u}^T R \, \vec{u} \, dt \tag{3.10}$$

where $R = \rho$, and the terminal condition is given by $\vec{x}(t_{k+1}) = \vec{0}$.

Solution of the optimal control problem defined by Eqs. (3.9) and (3.10) gives the control for ith joint as (detailed solution is given in Appendix B)

$$u_i(t) = \frac{6}{T^2}(q_{d_i} - q_i) + \frac{4}{T}(\dot{q}_{d_i} - \dot{q}_i) + \gamma_{ik}\ , \quad t \in [t_k, t_{k+1}] \tag{3.11}$$

where T is called time-to-go, and given by $T = t_{k+1} - t$. Substituting Eq. (3.11) into Eq. (3.2), one gets

$$\ddot{q}_i + \frac{4}{T}\dot{q}_i + \frac{6}{T^2} q_i = \frac{6}{T^2} q_{d_i} + \frac{4}{T}\dot{q}_{d_i} + \ddot{q}_{d_i}\ , \quad t \in (t_k, t_{k+1}) \tag{3.12}$$

Thus the natural frequency of the system defined by Eq. (3.12) can be adjusted by only changing time-to-go, while the damping ratio of the system is a constant having the value $\xi = 0.82$. Therefore the rigidity of the system can be increased by changing only a single parameter.

3.2 A MINIMUM ENERGY ADAPTIVE SCHEME FOR INTERACTIVE NONLINEAR MODEL

It seems that the efficiency of the control of a manipulator arm will increase as long as the original structure of the physical system is preserved in the state model. So the way in which the dynamic equations of the manipulator are converted into a state model effects the control algorithm that will be applied to the system and consequently its efficiency. Keeping that point in mind, the state model of the system is described by a nonlinear equation which takes the coupled dynamics of the joints into account, and given by

$$\dot{\vec{x}} = \vec{f}(\vec{x},\vec{a},\vec{u}) \tag{3.13}$$

where $\vec{x}$ is the state vector, $\vec{a}$ is the parameter vector, and $\vec{u}$ is the control vector. The minimum energy problem is considered given the performance index to be minimized as

$$J = \frac{1}{2} \int_{t_o}^{t_f} \vec{u}^T R \, \vec{u} \, dt \tag{3.14}$$

where R is a constant positive definite control penalization matrix. The optimal control problem formulated by Eqs. (3.13) and (3.14) is very hard to be determined and implemented. Therefore a sub-optimal linear feedback law is proposed which requires the repeated linearization of the nonlinear state equations [14]. The nonlinear state model is approximated by a linear model given as

$$\dot{\vec{x}}_m = A_k(\vec{x}_k,\vec{a}_k,\vec{u}_k)\vec{x}_m + B_k(\vec{x}_k,\vec{a}_k,\vec{u}_k)\vec{u} + \vec{a}_{ok}(\vec{x}_k,\vec{a}_k,\vec{u}_k) \tag{3.15}$$

which is valid around the measured values $\vec{x}_k$, $\vec{a}_k$ and $\vec{u}_k$ at the correction instant t_k. Thus the problem is reduced to a linear time invariant servomechanism problem if A_k, B_k and $\vec{a}_{ok}$ are considered constants as long as the nonlinear system state stays in the validity domain of the linear stationary model. The solution of this problem is given by

$$\vec{u}(t) = -R^{-1}B_k^T P(t)\vec{x}(t) - R^{-1}B_k^T\vec{p}(t) \ , \ t \in (t_k,t_f) \tag{3.16}$$

where $P(t)$ and $\vec{p}(t)$ are obtained from the adjoint variable via the linear Riccati transformation $\vec{\lambda}(t) = P(t)\vec{x}(t) + \vec{p}(t)$ and satisfy the conventional Riccati differential equations of the linear servo-mechanism problem.

Further approximation is introduced by taking $t_o = t_k$, considering only $P_k = P(t_k)$ and $\vec{p}_k = \vec{p}(t_k)$ and keeping them constants until another correction. The final time t_f may be kept unchanged or redefined at each correction for a chosen operation time Δ such that $t_f = t_k + \Delta$. Thus the suboptimal control law is given by

$$\vec{u}(t) = G_k\vec{x}(t) + \vec{u}_{ok} \tag{3.17}$$

where G_k is a linear feedback matrix and $\vec{u}_{ok}$ an open loop component of the control vector, both constants between two corrections made at t_k and t_{k+1}, and they are given as

$$G_k = -R^{-1}B_k^T P_k \quad , \quad \text{and} \quad u_{ok} = -R^{-1}B_k\vec{p}_k \quad , \quad t \in (t_k,t_{k+1}) \tag{3.18}$$

An efficient method for the computation of G_k and $\vec{u}_{ok}$ is given in [14].

3.2.1 Linear System Matrices and Their Generation

The linear model matrices which appear in Eq. (3.15) should be defined in order to apply the control algorithm described in Section 3.2.1. The linear model matrices which are valid at the time instant t_k are obtained by evaluating the matrices in the equation of motion of the manipulator instead of linearizing the nonlinear state equations. Two different forms of the equation of motion are used to obtain the linear model of the system.

In order to derive the state model of the system state variables are defined as

$$x_{2i-1} = q_i$$
$$x_{2i} = \dot{q}_i \qquad , \qquad i = 1,2,\ldots,n \tag{3.19}$$

where n is the degree of freedom of the manipulator arm. If Eq. (3.1) is used in order to obtain the linear model matrices with the above definition of state variables, one obtains

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 \end{bmatrix} , \quad B = \begin{bmatrix} 0 & 0 & \ldots\ldots & 0 \\ b_{11} & b_{12} & \ldots & b_{1n} \\ 0 & 0 & \ldots & 0 \\ \vdots & & & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{nn} \end{bmatrix} \tag{3.20}$$

where A is a (2nx2n) and B is a (2nxn) matrix. The elements of the B matrix are obtained from the inverse of the inertia matrix as

$$b_{ij} = [J^{-1}]_{ij} \qquad i,j = 1,2,\ldots,n \tag{3.21}$$

and the vector $\vec{a}_o$ is given by

$$\vec{a}_o = -J^{-1}\vec{k} \quad , \tag{3.22}$$

where $\vec{k} = V\dot{\vec{q}} + \vec{f}(\dot{q}_i\dot{q}_j,\vec{q}) - \vec{g}(\vec{q}) + \vec{h}(\vec{q})$. As an alternative, if a symbolic torque vector is defined as $\vec{\tau}^* = \vec{\tau} - \vec{k}$ then the vector $\vec{a}_o$ becomes equal to zero.

The second form of the equation of motion can also be used to derive the state model of the system. Solution of Eq. (2.68) for joint accelerations is given by

$$\ddot{\vec{q}} = J^{-1}(\vec{q})[\vec{\tau} - [C(\vec{q},\dot{\vec{q}}) + V]\dot{\vec{q}} + \vec{k}'] \tag{3.23}$$

where $\vec{k}' = \vec{g}(\vec{q}) - \vec{h}(\vec{q})$. With state variables defined as in Eq. (3.19), linear state model matrices can be obtained from Eq. (3.23) as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \,\cdots\cdots\, 0 \\ 0 & a_{11} & 0 & a_{12} \cdots\cdots a_{1n} \\ 0 & 0 & 0 & 1 \,\cdots\cdots\, 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 & 0 \,\cdots\cdots\, 0 \\ 0 & a_{n1} & 0 & a_{n2} \cdots\cdots a_{nn} \end{bmatrix}$$

where the elements of the A matrix is given by the relation

$$a_{ij} = [-J^{-1}C]_{ij} \qquad i,j = 1,2,\ldots,n \tag{3.24}$$

The matrix B is the same as given in Eq. (3.20), and the vector $\vec{a}_o$ is given as

$$\vec{a}_o = -J^{-1}\vec{k}' \,. \tag{3.25}$$

3.3 MODEL AND CONTROL HORIZON UPDATE POLICIES

The methods for the computation of the feedback gains and the open-loop components of the control vector, proposed in this study, are based on a linear model which has to approximate the non-linear state equations of the controlled system. Linear model matrices used to generate the control vector are valid as long as the system state stays in the validity domain of the linear model. This linear model and corresponding feedback gains have to be corrected when the approximation error introduced by the linearization procedure becomes important with respect to linear terms. Therefore the system matrices, that appear in the equation of motion of the manipulator should be updated when they no more represent the actual system. There are mainly two approaches adopted for that purpose:

i) The first one is on-line calculation of the system matrices at given instants of time with a predetermined frequency as the manipulator arm moves along the specified trajectory. The update frequency of the system matrices can be adjusted depending upon the desired trajectories and the desired quality of control. As a modification of that approach, the system matrices may be updated only when the system state moves away from the desired trajectories more than a tolerance value.

ii) The second approach is off-line calculation of the system matrices. The system matrices for desired trajectories can be calculated and memorized prior to task execution assuming that the desired trajectories are known in advance. Then during movement of the manipulator arm, required system matrices can be picked up from the memory and kept constant as long as the system state stays in the considered region of the desired trajectory. Significant departures from the planned trajectory can not be tolerated because the precomputed system matrices are valid only when the arm configuration is in the vicinity of the desired state. In order to overcome this drawback Raibert and Horn [15] proposed to calculate the system matrices for every possible configuration of the manipulator arm and store them for future use. However, this method requires considerable memory space.

Since the system matrices change during the motion of the manipulator arm along the trajectory, feedback gains should also be changed according to the changing system matrices. The system matrices and the feedback gains are updated together in most of the adaptive control schemes found in literature.

In this study the update frequency of the system matrices and the feedback gains are adjusted separately. Depending upon the followed trajectory it may not be needed to update the system matrices as frequent as the feedback gains.

The values of the control vector depend upon a horizon time, T (or time-to-go) at the end of which the controller wants to drive the system state to the desired state while minimizing the control cost. The response of the system can be improved by proper tuning of a single adjustment parameter which is the horizon time. Various policies can be adopted for its adjustment:

i) A horizon time, T is chosen and kept constant in the time interval (t,t_f) where $t_f = t + T$ defines the target point for the system to pass through. The feedback matrix and open loop control component are calculated once at time t and kept constant in the same time interval (t,t_f) as shown in Fig. 3.1.

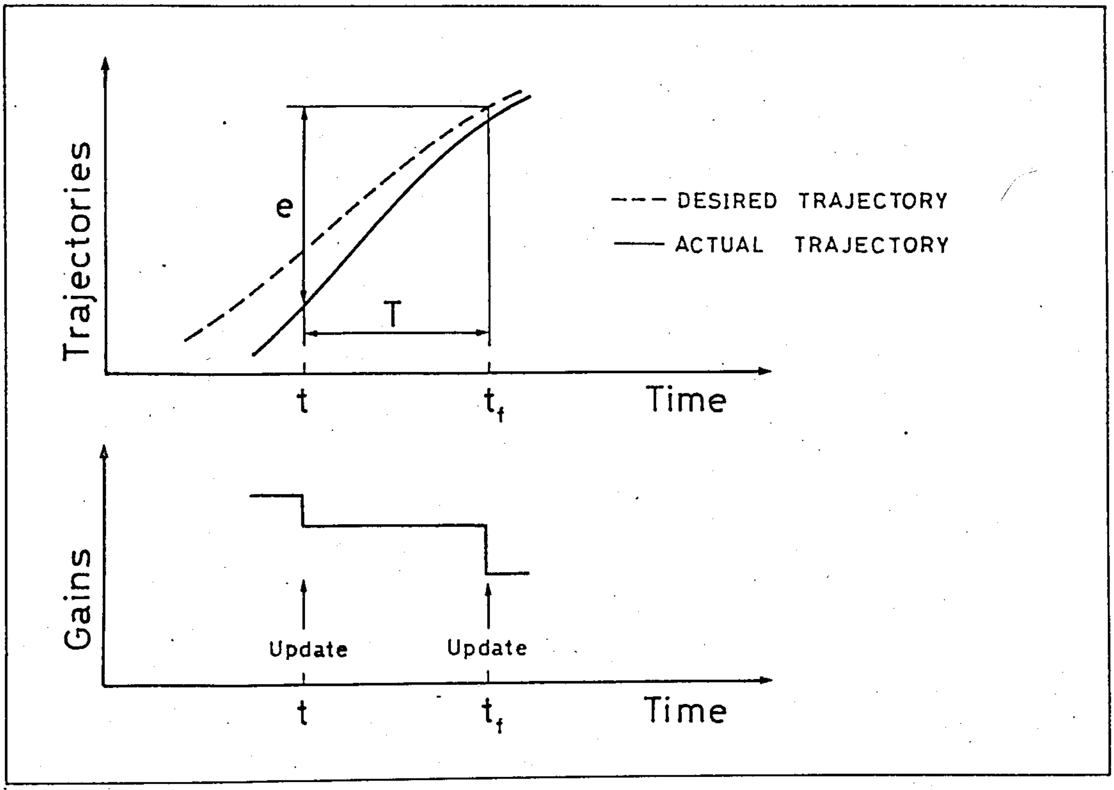


FIGURE 3.1 - Change of feedback gains with constant horizon time.

ii) Instead of keeping the horizon time fixed in the interval (t,t_f), it is updated with a predetermined frequency. Hence the control vector is also updated with the same frequency by taking errors with respect to the target point at t_f as shown in Fig. 3.2. Hence the gain values increases asymptotically in the interval (t,t_f).

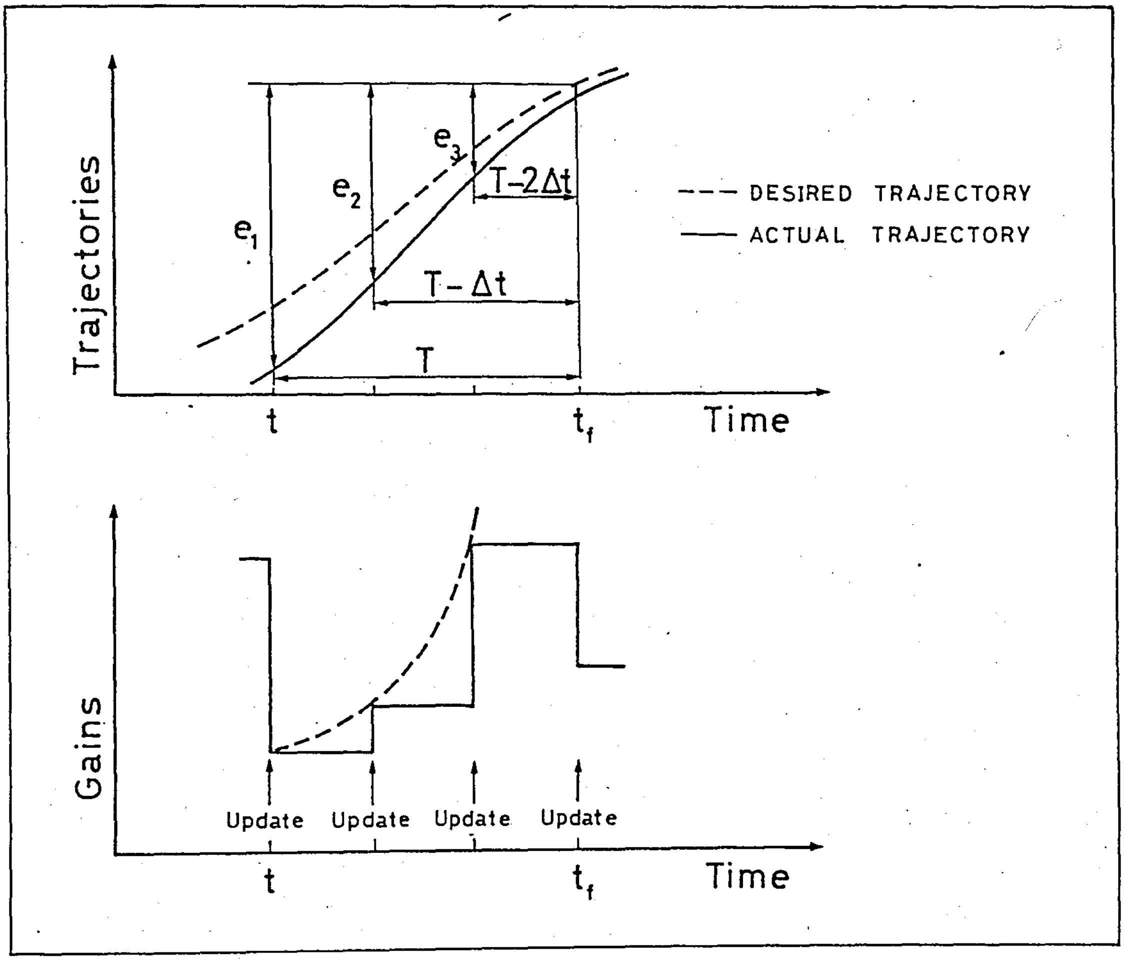


FIGURE 3.2 - Change of feedback gains with variable horizon time.

IV. SIMULATION OF MANIPULATOR BEHAVIOUR

A computer program is written to make numerical simulation studies in order to investigate the effectiveness of the proposed control schemes. The program is explained in the following sections.

4.1 MAIN PROGRAM

The general structure of the simulation program is given in Fig. 4.1. In the first part of the program off-line calculations are performed which include the inverse kinematics solution in order to determine the desired joint trajectories. The second part of the program includes the block for simulation of system dynamics with the selected control scheme.

All input data required for the program can be classified into four groups:

i) Geometric and dynamic parameters of the manipulator.

ii) Data related with the desired trajectories.

iii) Data related with the control scheme.

iv) Other data related with the simulation.

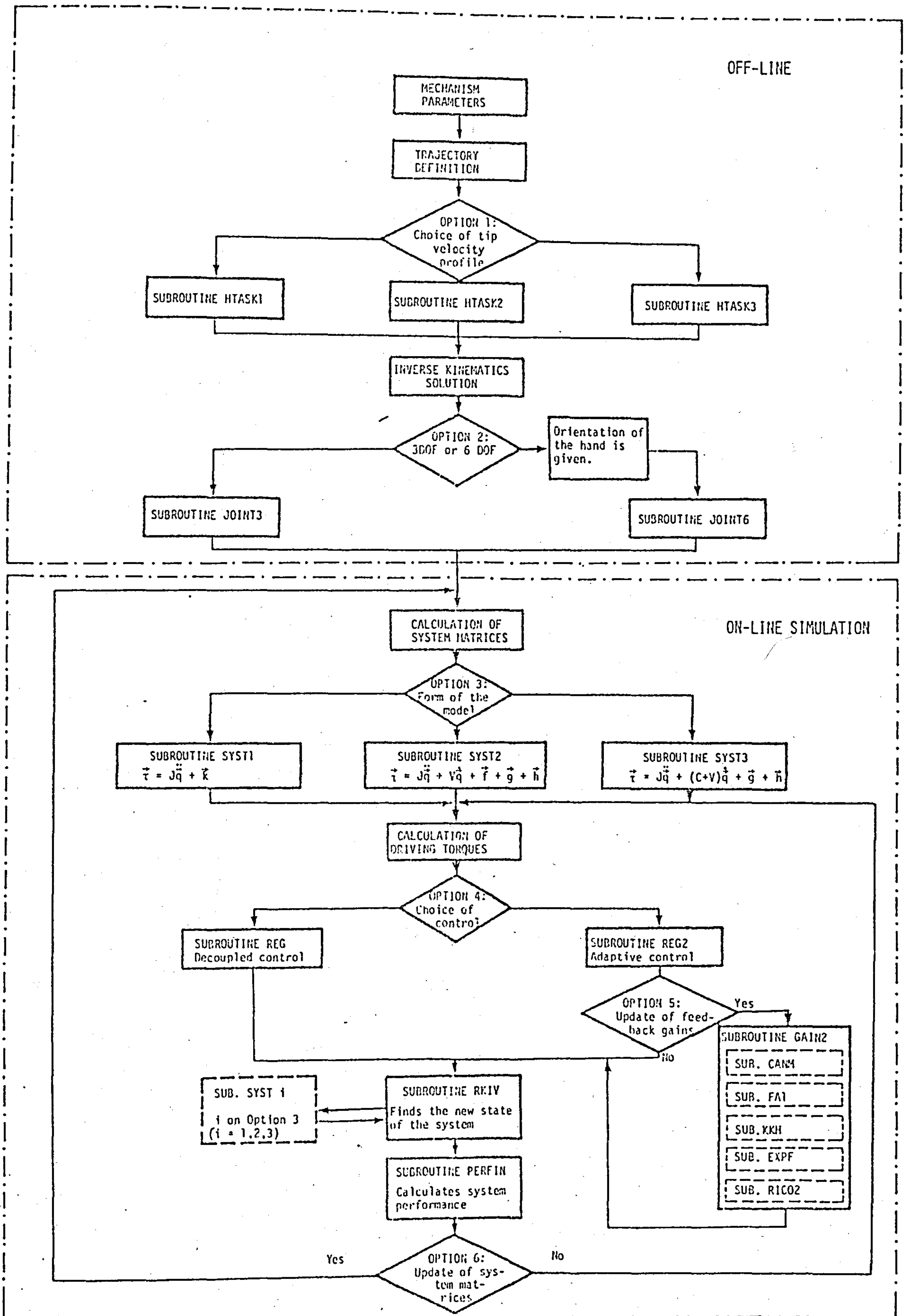


FIGURE 4.1 - Flowchart of the simulation program.

A description of input data for the simulation program is given in Appendix C.

4.2 SUBPROGRAMS FOR TRAJECTORY DEFINITION AND INVERSE KINEMATICS SOLUTION

There are three subprograms written for the calculation of position, velocity, and acceleration of the end point of the manipulator arm in order to have the manipulator tip to follow a straight line between two points given in terms of base coordinates. Those subprograms give three different acceleration profiles as shown in Fig. 4.2 and they require the end points of the straight line segment and the travel time t_f in order to generate the desired trajectories of the manipulator tip. Subprograms HTASK2 and HTASK3 also requires the acceleration time t_{ac}.

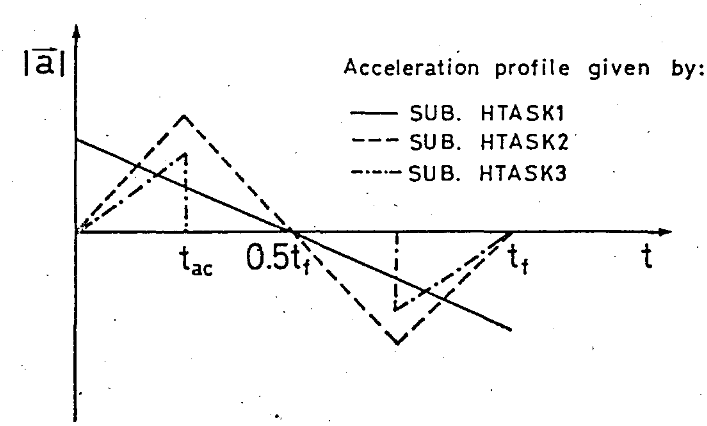


FIGURE 4.2 - Acceleration profiles given by subprograms for trajectory definition.

Given the trajectories of the manipulator tip subprograms JOINT3 and JOINT6 calculates the desired trajectories of joint coordinates for two manipulator models with three and six degrees of freedom, respectively. The kinematic properties of those models and their inverse kinematics solutions are presented in Appendix D.

4.3 SUBPROGRAMS FOR DYNAMIC MODELLING OF MANIPULATOR

Three programs have been written for dynamic modelling of an n degree of freedom manipulator. Each of them requires the geometric and dynamic parameters of the manipulator and the joint coordinates and velocities as input data. However they generate three different forms of the equation of motion.

The first subprogram which is called SYST1 generates the matrices J and V and the vectors, $\vec{f}$, $\vec{g}$, and $\vec{h}$ as given in Eq.(4.1).

$$\vec{\tau} = J(\vec{q})\ddot{\vec{q}} + V\dot{\vec{q}} + \vec{f} + \vec{g} + \vec{h} \tag{4.1}$$

The second subprogram, SYST2, generates a short form of Eq. (4.1) by forming the matrix J and the vector $\vec{k}$ as given in Eq. (4.2).

$$\vec{\tau} = J(\vec{q})\ddot{\vec{q}} + \vec{k} \tag{4.2}$$

where $\vec{k} = V\dot{\vec{q}} + \vec{f} + \vec{g} + \vec{h}$

The last subprogram written for dynamic modelling of a manipulator arm which is called SYST3 generates the matrices J, V, and C, and the vectors $\vec{g}$, and $\vec{h}$, thus giving the equation of motion in the form

$$\vec{\tau} = J(\vec{q})\ddot{\vec{q}} + [C(\vec{q},\dot{\vec{q}}) + V]\dot{\vec{q}} + \vec{g} + \vec{h} \tag{4.3}$$

4.4 SUBPROGRAMS FOR THE CONTROL GENERATION

There are mainly two subprograms written to generate the inputs for the actuators at each joint. The first one which is called REG requires the inertia matrix J and the vector $\vec{k}$ as defined in Eq. (4.2), the desired joint positions, velocities, and accelerations and the horizon time as input data.

The second subprogram, REG2 uses the values of the feedback matrix and the control vector given by subprogram GAIN2 in order to calculate the actuator inputs. The inputs required for GAIN2 includes the linear state model matrices, A, B, and the vector $\vec{a}_o$, the control penalization matrix R, desired terminal state vector $\vec{x}_d$ and the horizon time.

V. SIMULATION RESULTS

In order to investigate the effectiveness of the proposed control schemes a series of computer simulations are carried out for two revolute robot arms with three and six degrees of freedom, respectively. The latter model is obtained by adding a three degree of freedom hand to the former one. The parameters of the six degree of freedom model are given in Table 5.1 and its mechanical configuration is presented in Appendix D (Fig. D.1).

TABLE 5.1 - Parameters of the six degree of freedom manipulator model.

Link	Mass (kg)	Length (m)	I_x (kgm^2)	I_y (kgm^2)	I_z (kgm^2)
1	4.0	0.20	0.0167	0.0167	0.0067
2	2.5	0.60	0.0042	0.0771	0.0771
3	1.5	0.50	0.0025	0.0325	0.0325
4	0.5	0.12	0.0009	0.0004	0.0008
5	0.5	0.12	0.0008	0.0009	0.0004
6	0.5	0.08	0.0003	0.0005	0.0003

The basic scenario used for the simulation studies is to have the manipulator tip to move along a straight line from point P_1 to point P_2

in 1.6 seconds. The numerical values chosen for P_1 and P_2 are $P_1 = (1,0,0)$ and $P_2 = (-0.5,0.5,0.5)$ expressed in meters. A parabolic distribution is chosen for the speed of the manipulator tip, and the maximum value it takes along the desired trajectory is 1.56 m/sec.

The main parameters of the simulation are the update period of feedback gains (TGAIN), update period of system matrices (TUP), and the horizon time (or time-to-go, TTG). Computer simulations are carried out with various combinations of those parameters. The integration technique used in the simulations is the fourth order Runge-Kutta with a step size of 0.01 seconds. Simulation results are summarized in Tables 5.2-4. The values in the last two columns of those tables evaluate the tracking quality and the control cost of the system and they are obtained by calculating the quadratic performance index

$$J = \frac{1}{2} \int_{t_o}^{t_f} [(\vec{x}_d - \vec{x})^T Q (\vec{x}_d - \vec{x}) + \vec{u}^T R \vec{u}] dt$$

where $\vec{x}_d$ is the vector of desired states. Hence the term $\vec{x}_d - \vec{x}$ gives the deviation from the desired trajectories (in Tables 5.2-4 this term is given as $\vec{e}$). Q and R are diagonal penalization matrices whose elements are given by

$$[Q]_{ii} = \begin{cases} 100, & i = 1,3,\ldots,2n-1 \\ 1, & i = 2,4,\ldots,2n \end{cases} \quad ; \; [R]_{ii} = 1 \; , \; i = 1,2,\ldots,n$$

with n being the degree of freedom of the manipulator.

The first two sets of simulation are carried out for the three degree of freedom model using respectively, the computed torque technique and the adaptive feedback control, and the results are presented in Tables 5.2 and 5.3. A comparison of the values given in Tables 5.1 and 5.2 shows that the adaptive feedback control gives much better results than the computed torque technique. For both control schemes the best tracking is achieved with a horizon time of 0.04 seconds while the system matrices and feedback gains are updated at each 0.01 seconds. The values given in Table 5.3 indicates that increasing the update period of system matrices and/or feedback gains decreases system performance. However, increasing TGAIN influences system performance more than increasing TUP. Therefore it will be advantageous to increase the update period of system matrices rather than the update period of feedback gains if it is necessary to save computer time.

TABLE 5.2 - Performances obtained with the computed torque technique for the three degree of freedom model (TGAIN = 0.01 sec).

TTG (sec)	TUP (sec)	max.error on θ_i(rad)	max. error on $\dot{\theta}_i$(rad/sec)	$\int \vec{e}^T Q \vec{e}\, dt$	$\int \vec{u}^T R \vec{u}\, dt$
0.04	0.01	0.085	13.92	32.04	67592
0.04	0.04	0.216	16.22	39.61	70350
0.04	0.08	0.361	29.78	58.10	76338
0.08	0.01	0.136	14.22	17.37	15578
0.08	0.04	0.124	12.19	19.78	16047
0.16	0.01	0.241	12.79	16.39	3954
0.16	0.04	0.233	11.77	15.57	3900
0.16	0.08	0.489	19.74	21.12	4319

TABLE 5.3 - Performances obtained with the adaptive feedback law for the three degree of freedom model.

TTG (sec)	TUP (sec)	TGAIN (sec)	max.error on θ_i(rad)	max. error on $\dot{\theta}_i$(rad/sec)	$\int \vec{e}^T Q \vec{e}\, dt$	$\int \vec{u}^T R \vec{u}\, dt$
0.04	0.01	0.01	0.000065	0.008	0.000005	14.59
0.04	0.08	0.01	0.001296	0.161	0.000848	17.03
0.04	0.01	0.04	0.081871	3.127	1.979498	5058.77
0.08	0.01	0.01	0.000271	0.011	0.000013	14.58
0.08	0.08	0.01	0.001566	0.109	0.000516	16.15
0.08	0.16	0.01	0.006350	0.320	0.004270	20.39
0.08	0.01	0.04	0.104791	3.831	1.463104	2020.30
0.08	0.01	0.08	0.194941	2.844	1.670103	687.58
0.16	0.01	0.01	0.002100	0.044	0.000278	29.04
0.16	0.08	0.01	0.008548	0.182	0.003197	15.64
0.16	0.16	0.01	0.012209	0.426	0.006668	20.18
0.16	0.01	0.04	0.133913	3.219	1.326210	825.06
0.16	0.01	0.08	0.198421	2.684	1.910185	310.71
0.16	0.01	0.16	0.400077	2.899	5.094765	153.65

Figures 5.3-5 show the oscillating behaviour of the tracking errors of joint angles for different horizon times. It is seen that as the horizon time is increased the frequency of oscillations decreases while the magnitude of the errors increases. The torque values also oscillates when either TUP or TGAIN is increased as shown in Figs. 5.9-12 whereas they only ripple for TUP = TGAIN = 0.01 sec (Fig. 5.8) and the frequency of oscillations depends upon the horizon time. On the other hand the control energy expended during the movement of the manipulator arm decreases as the horizon time is increased.

Finally, a third set of simulations are carried out for the six degree of freedom model using the same numerical values for the motion of the manipulator tip that are used for the three degree of freedom model. The orientation of the hand is kept fixed while following the straight line from point P_1 to point P_2. Performances obtained with the adaptive control scheme are presented in Table 5.4.

TABLE 5.4 - Performances obtained with the adaptive feedback law for the six degree of freedom model.

TTG (sec)	TUP (sec)	TGAIN (sec)	max.error on θ_i(rad)	max.error on $\dot{\theta}_i$(rad/sec)	$\int \vec{e}^T Q \vec{e}\, dt$	$\int \vec{u}^T R \vec{u}\, dt$
0.04	0.01	0.01	0.00060	0.083	0.00039	54.46
0.04	0.08	0.01	0.01529	2.291	0.07093	69.19
0.04	0.01	0.04	0.07719	2.911	2.59373	14133.91
0.16	0.01	0.01	0.22322	5.924	7.26140	2606.59
0.16	0.08	0.01	0.53936	8.958	13.63615	2666.37
0.16	0.16	0.01	0.42860	7.137	10.79370	2611.62
0.16	0.01	0.16	0.25714	4.559	4.95595	1685.00

Investigating the simulation results obtained with the six degree of freedom model, it is found that the maximum tracking errors occurred in the last three joints of the manipulator. Due to the nature of the applied control scheme the joints are perturbed with a certain frequency by the actuator supplied torques. Since the inertias of the last three links, which form the hand of the manipulator arm, are smaller than the inertias of the first three links, those perturbations causes the joints of the hand to overshoot the desired trajectories.

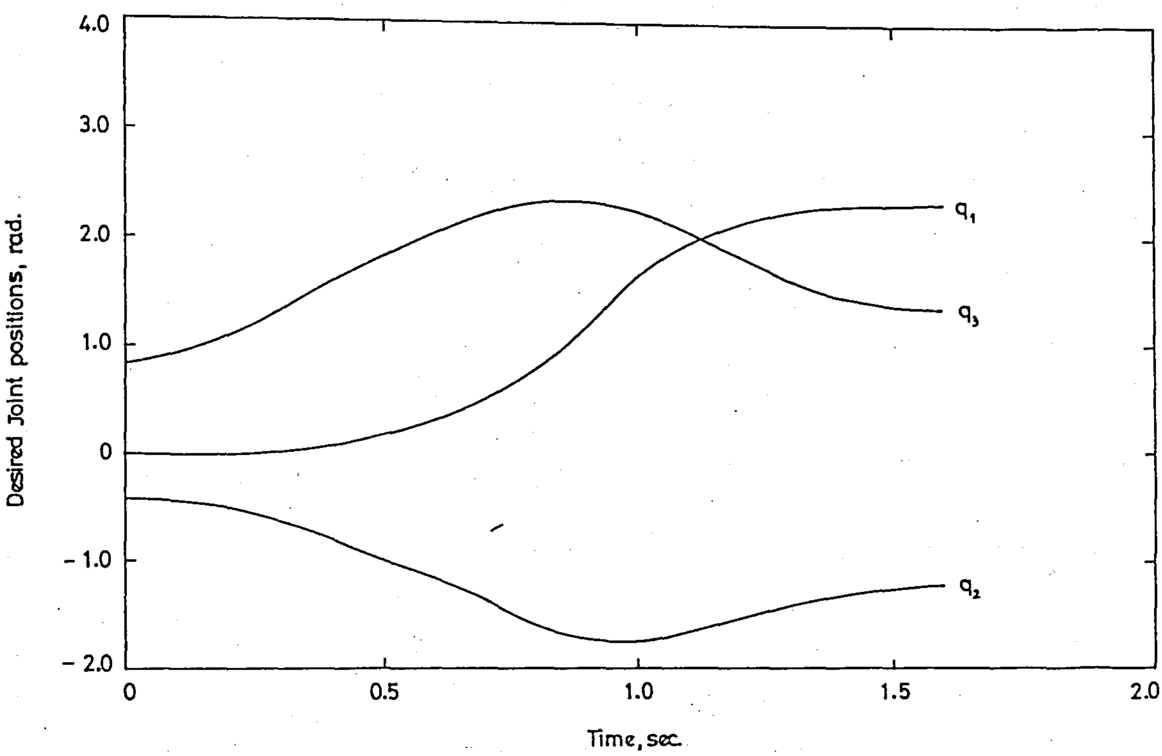


FIGURE 5.1 - Desired trajectories of joint angles for the three-degree-of-freedom model.

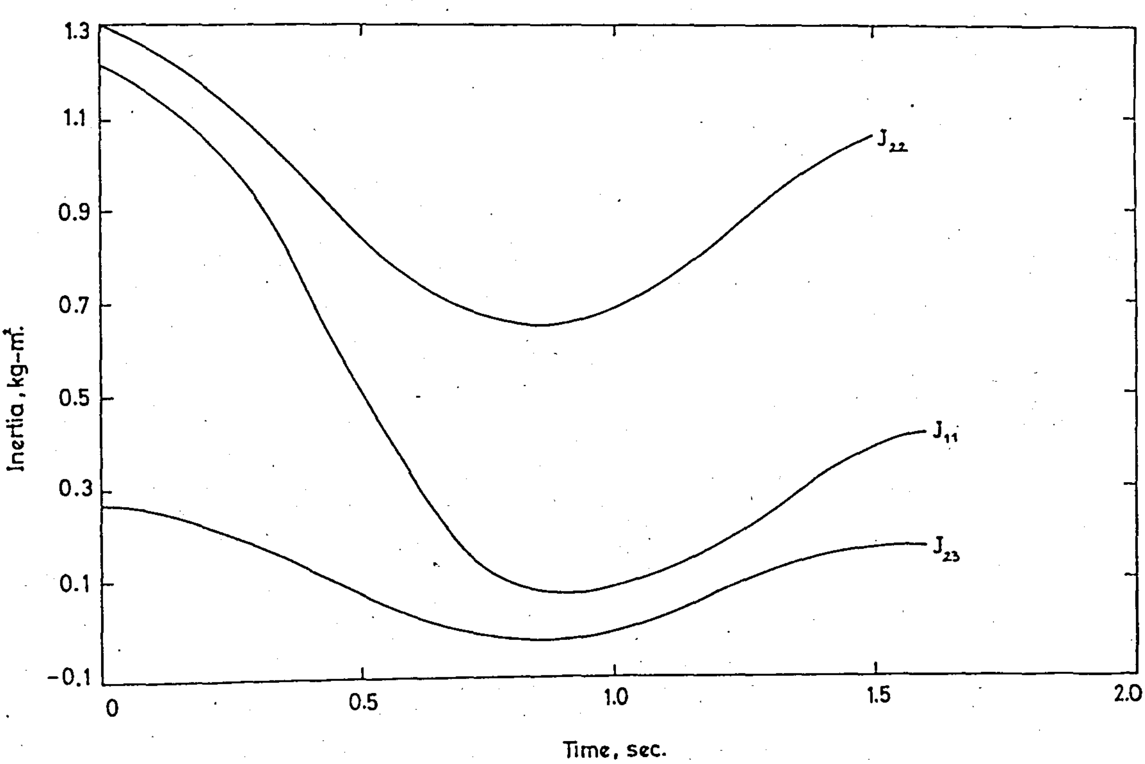
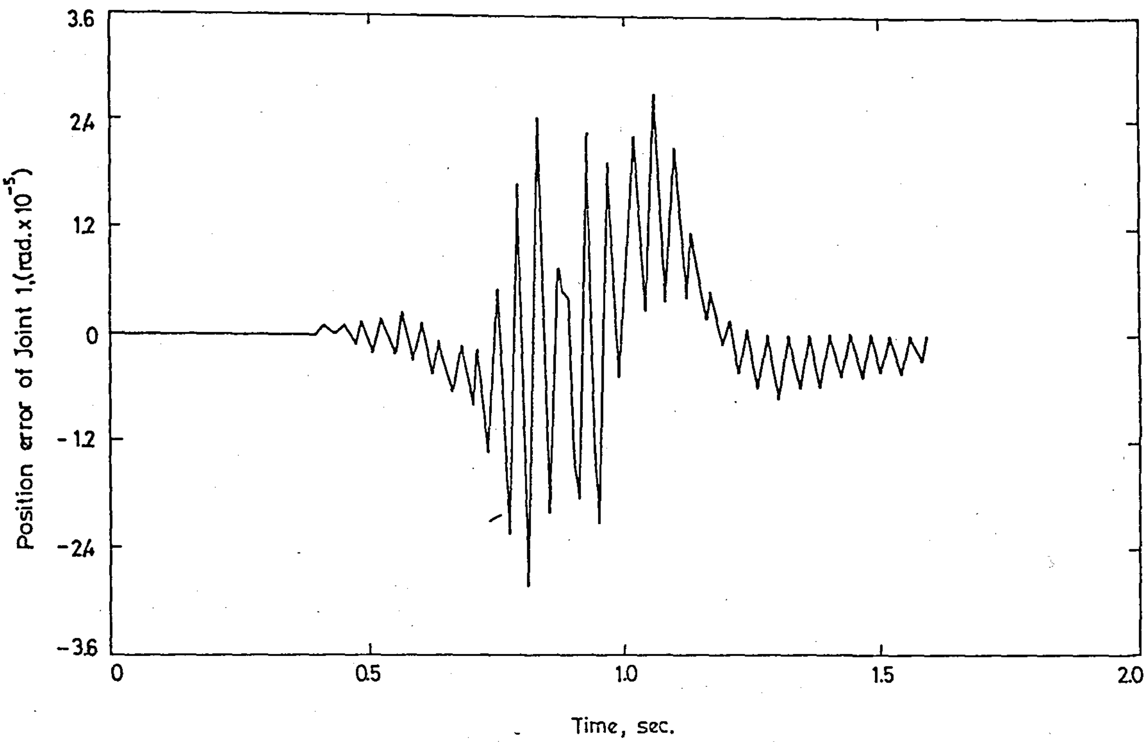
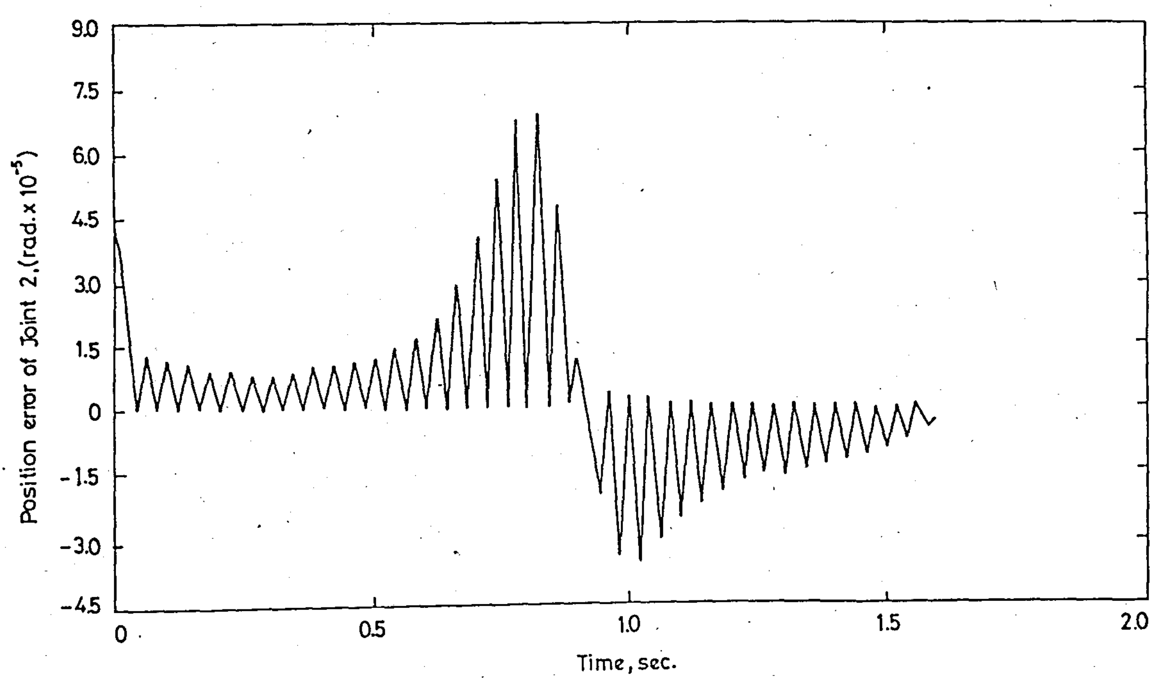


FIGURE 5.2 - Variation of the elements of the inertia matrix along the desired trajectories for the three-degree-of-freedom model.

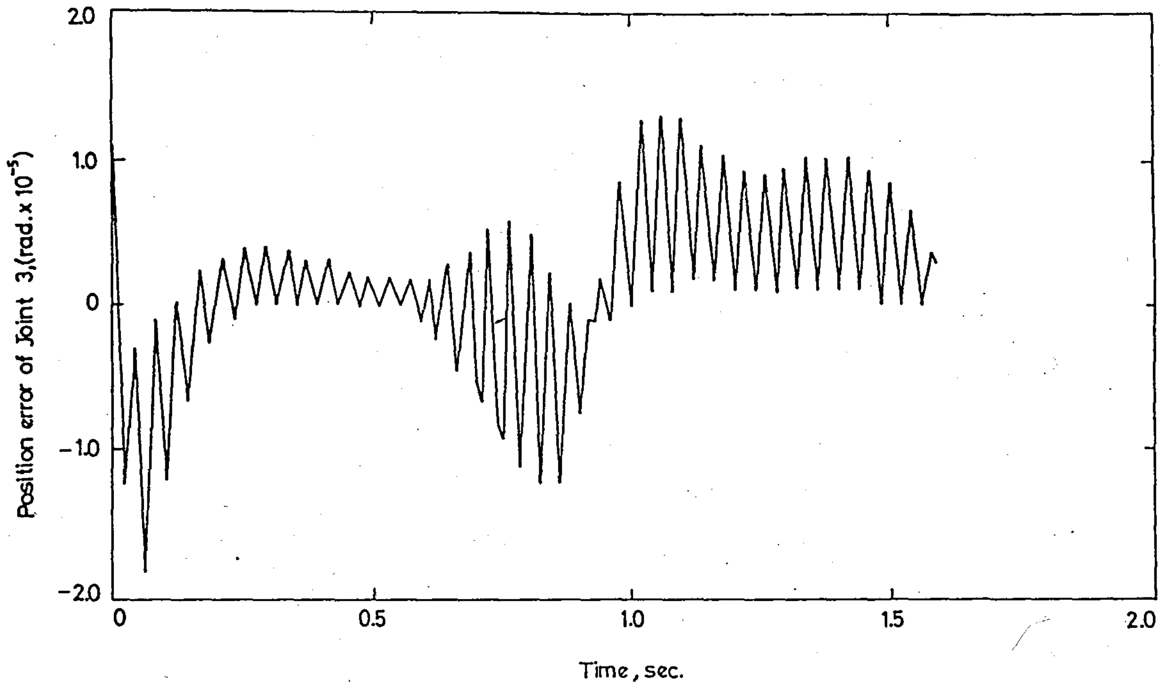


(a)



(b)

FIGURE 5.3 - Tracking errors at joint angles of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.04 sec, TUP = 0.01 sec., and TGAIN = 0.01 sec.



(c)

FIGURE 5.3 (continued).

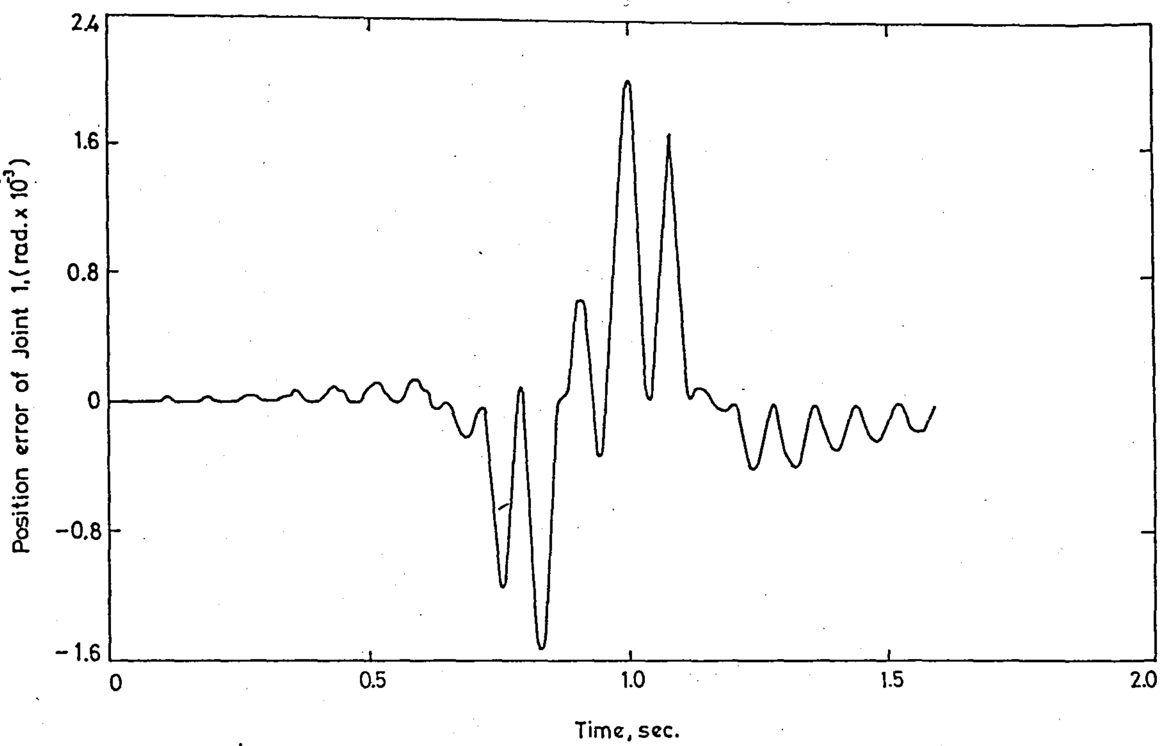


FIGURE 5.4 - Tracking error at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.08 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec.

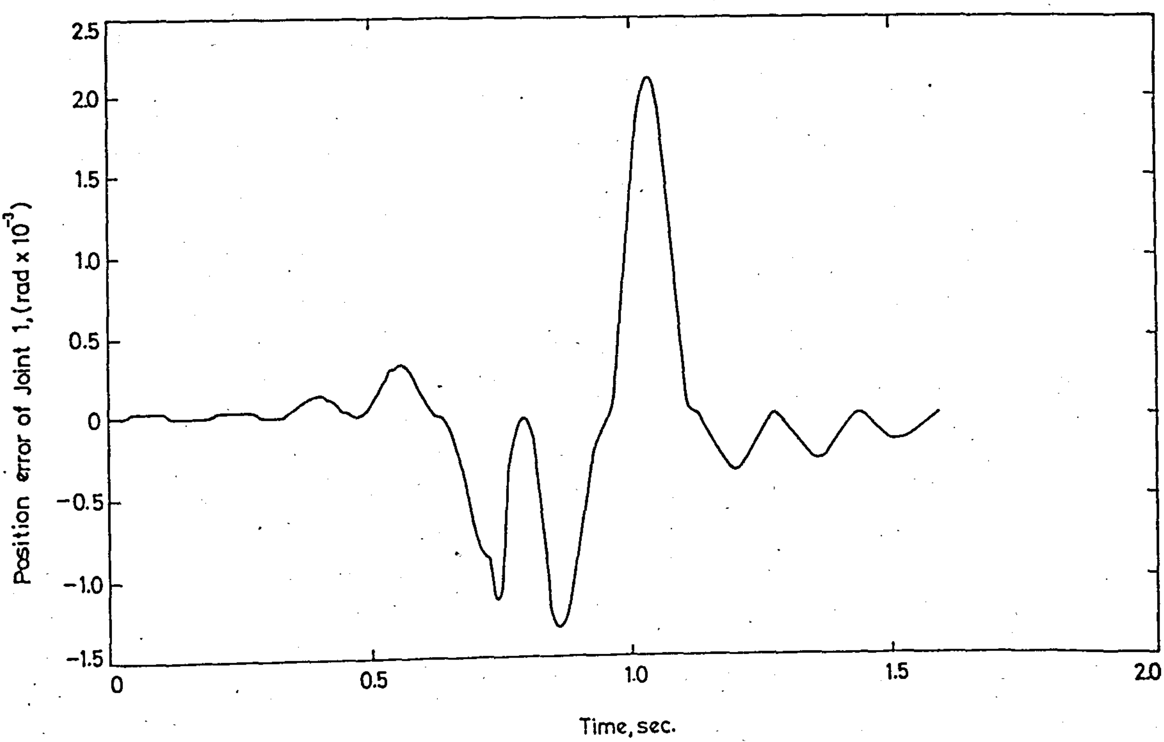


FIGURE 5.5 - Tracking error at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.16 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec.

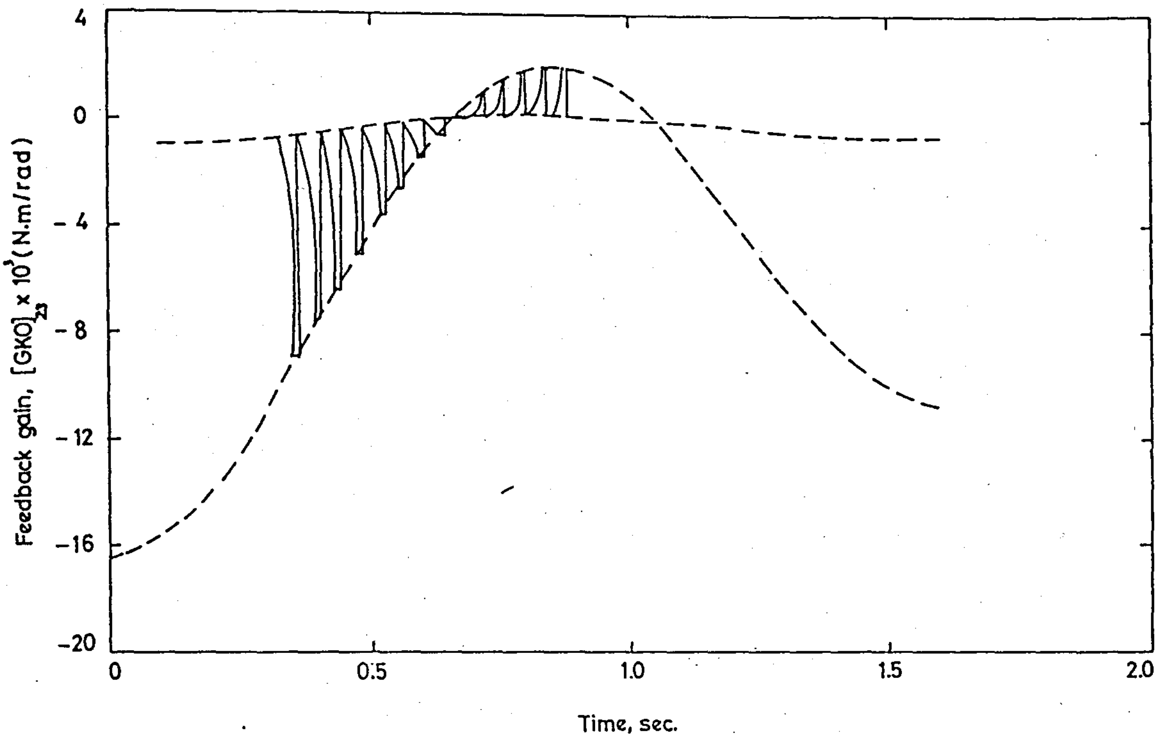


FIGURE 5.6 - Variation of the feedback gain $[GKO]_{23}$ obtained with the adaptive feedback law for TTG = 0.04 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec. (three-degree of-freedom case).

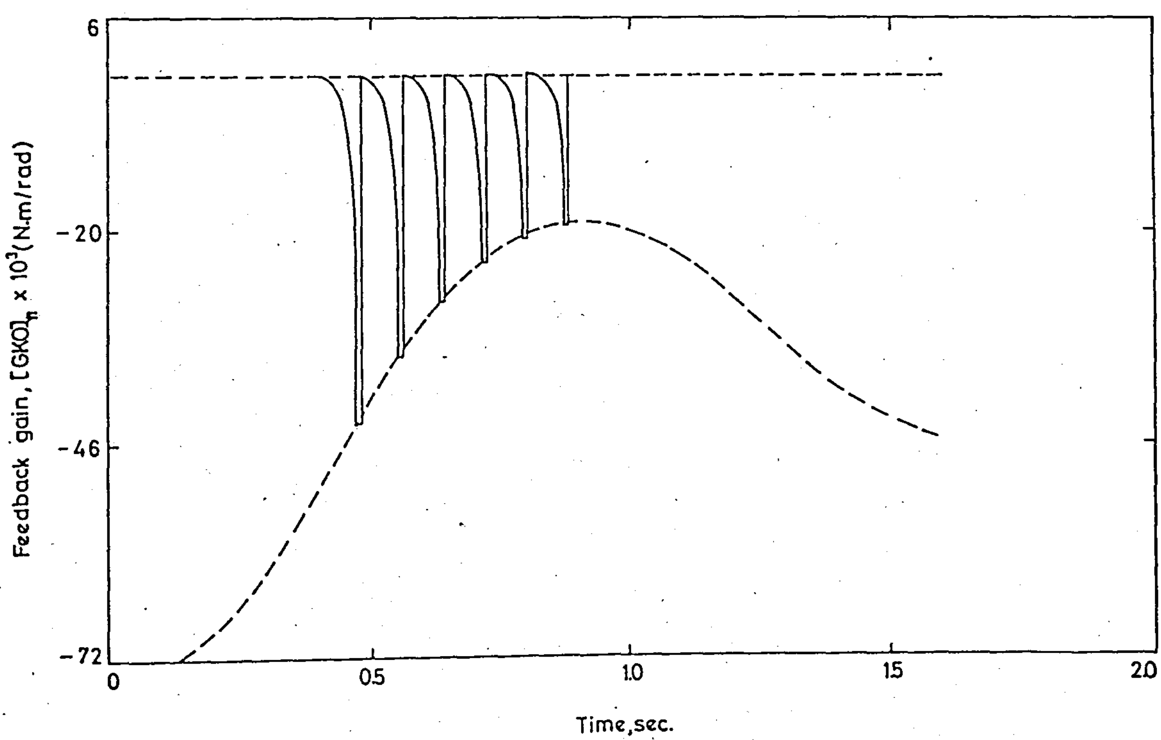
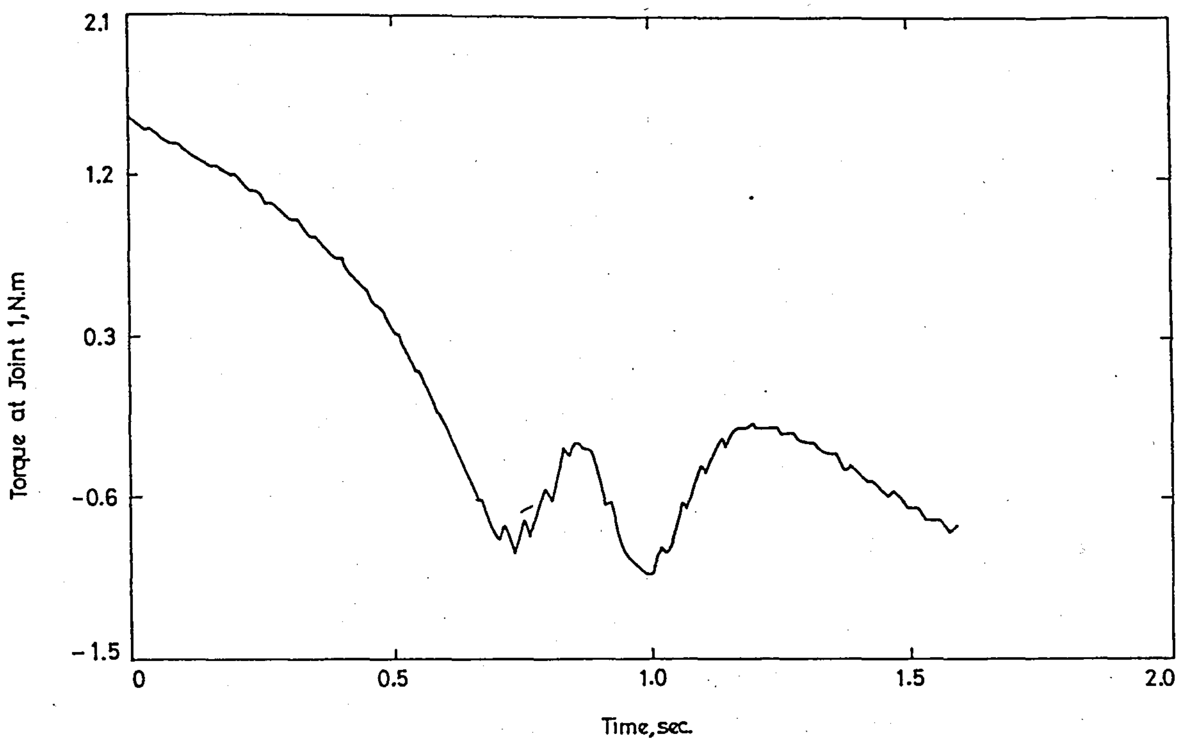
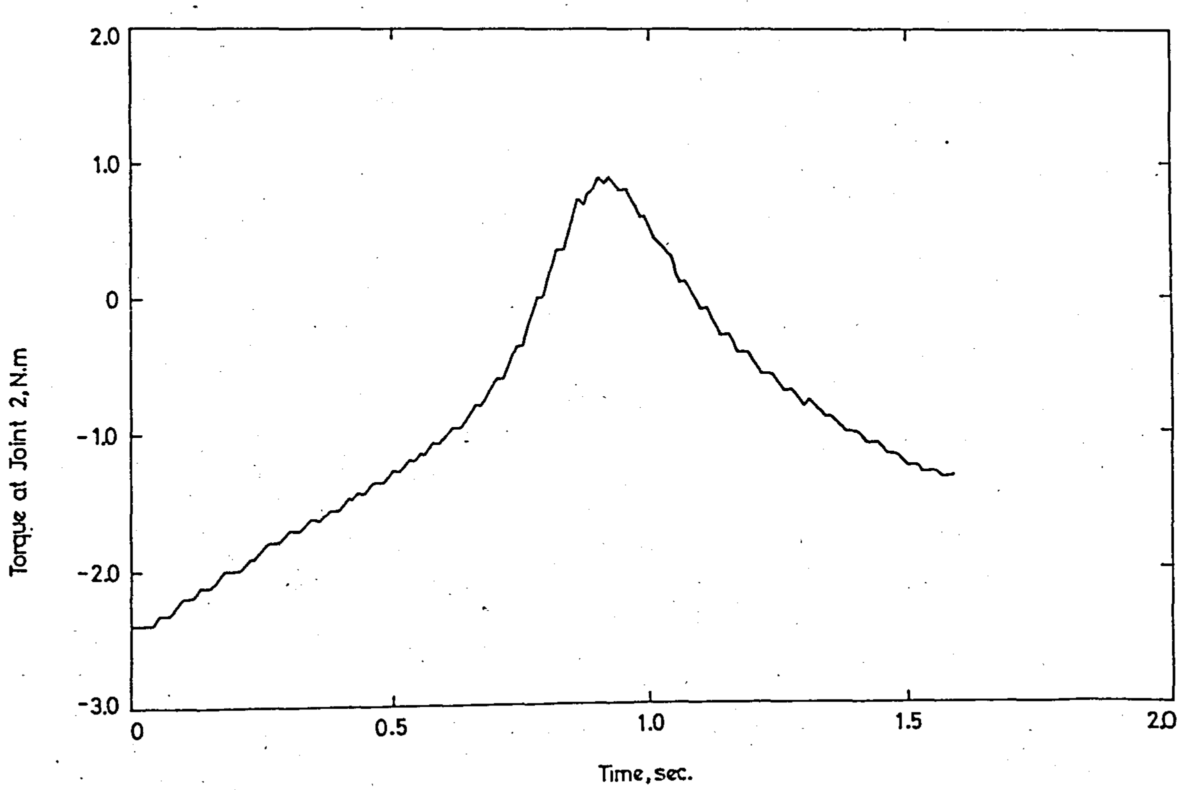


FIGURE 5.7 - Variation of the feedback gain $[GKO]_{11}$ obtained with the adaptive feedback law for TTG = 0.08 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec. (three-degree-of-freedom case).

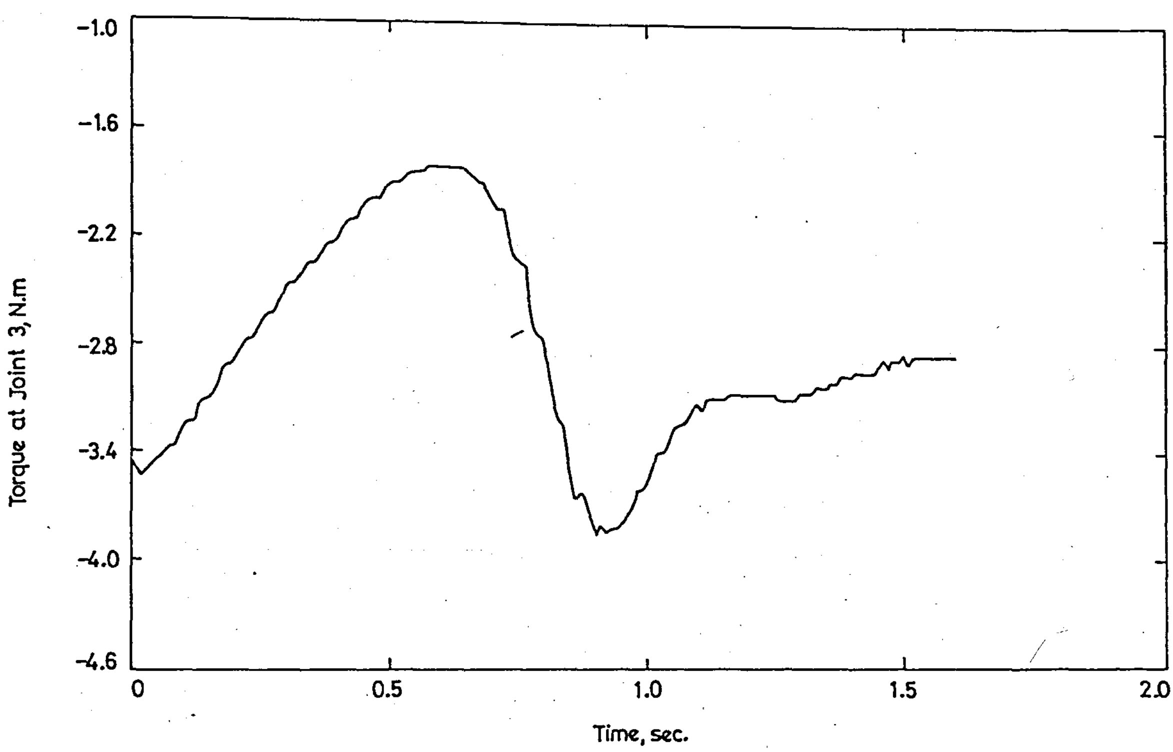


(a)



(b)

FIGURE 5.8 - Idealized torques acting at joints of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.04 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec.



(c)

FIGURE 5.8 (continued).

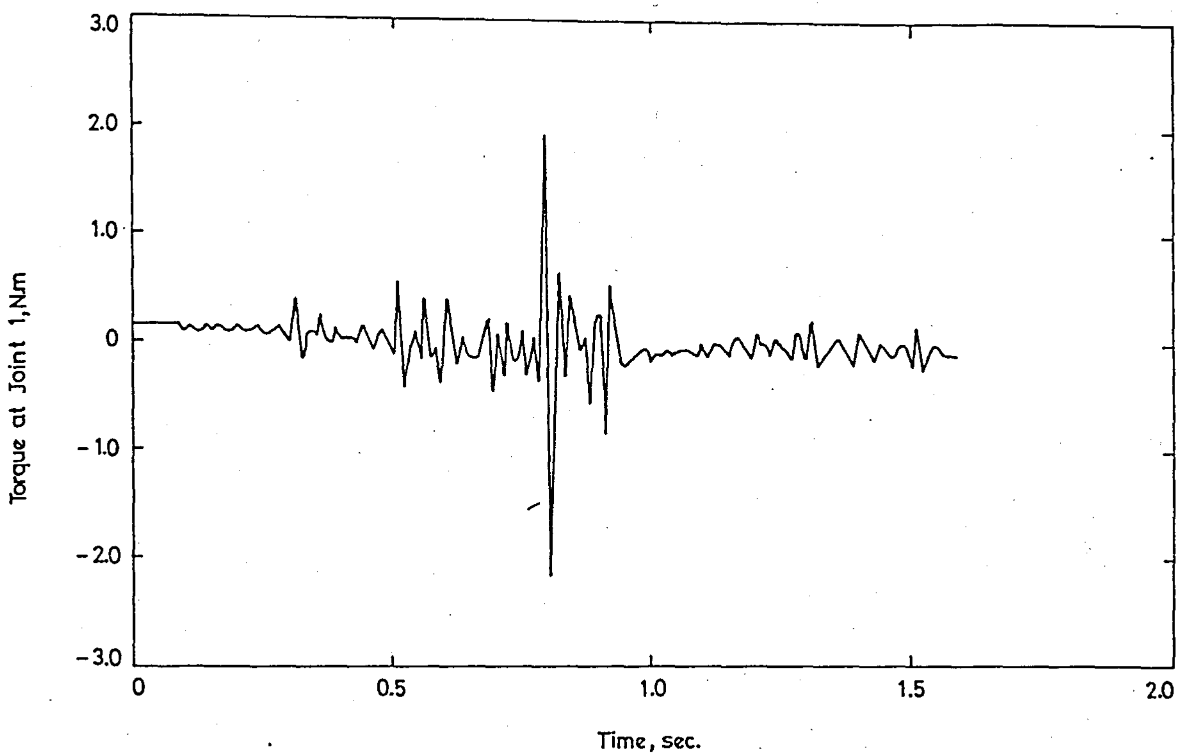


FIGURE 5.9 - Idealized torque τ_1 acting at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.04 sec., TUP = 0.10 sec., and TGAIN = 0.01 sec.

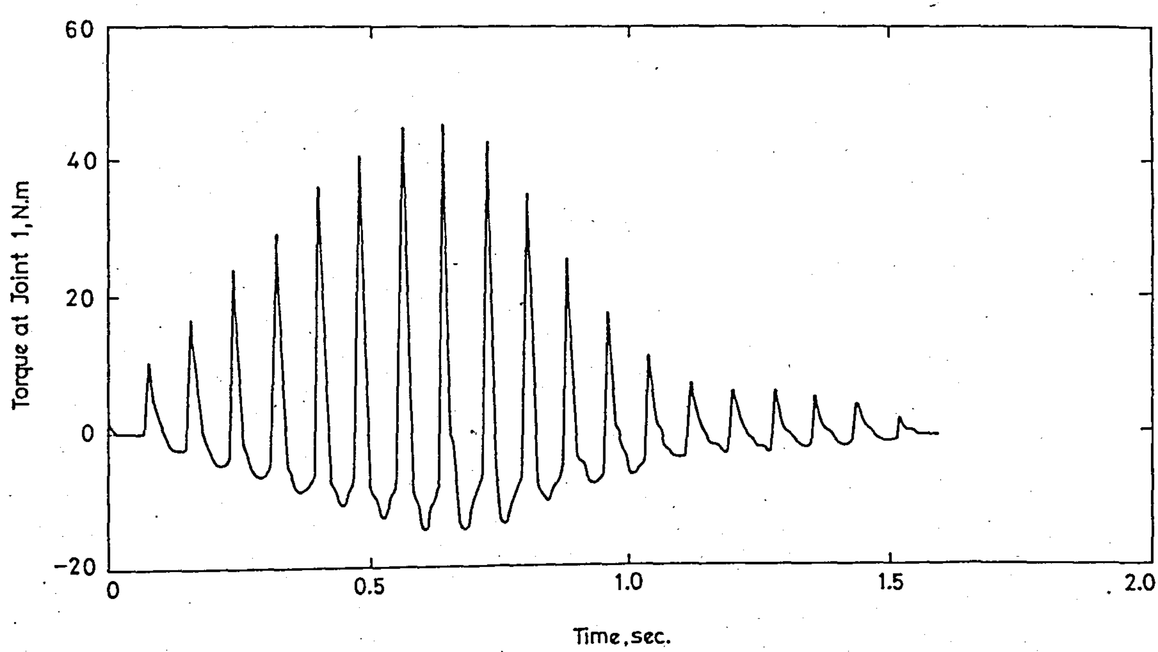


FIGURE 5.10 - Idealized torque τ_1 acting at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.08 sec., TUP = 0.01 sec., and TGAIN = 0.08 sec.

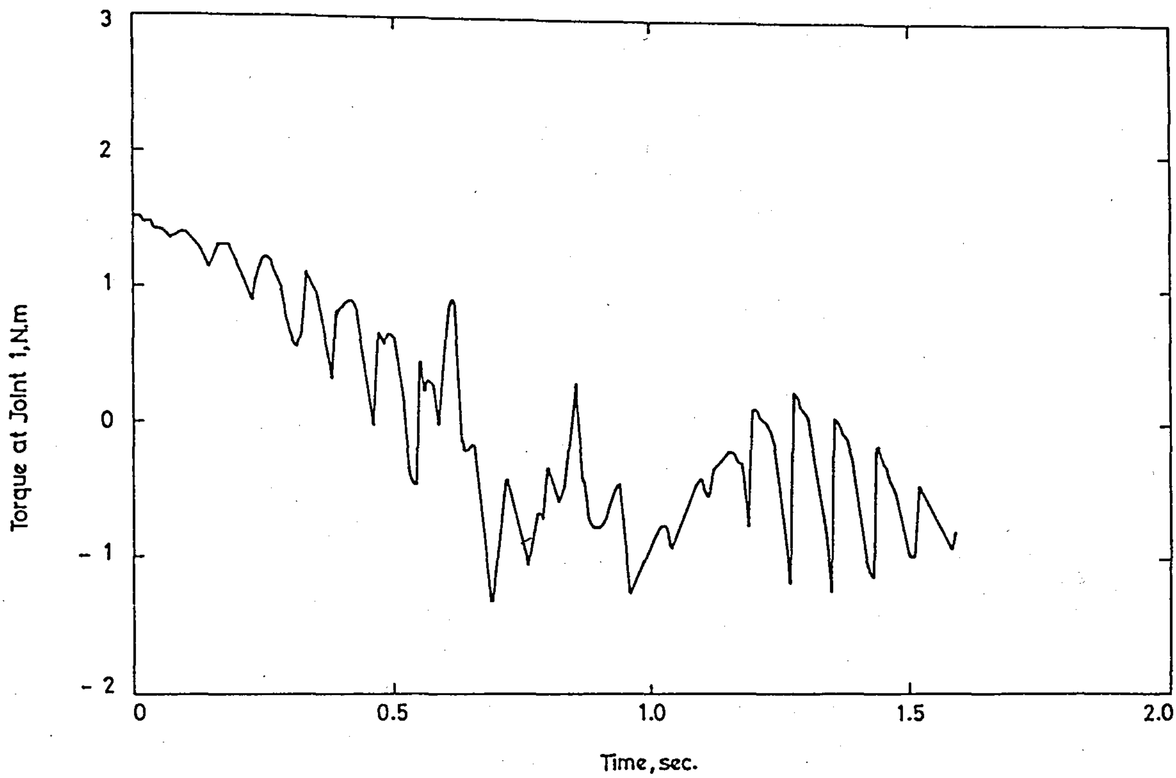


FIGURE 5.11 - Idealized torque τ_1 acting at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.08 sec., TUP = 0.08 sec., and TGAIN = 0.01 sec.

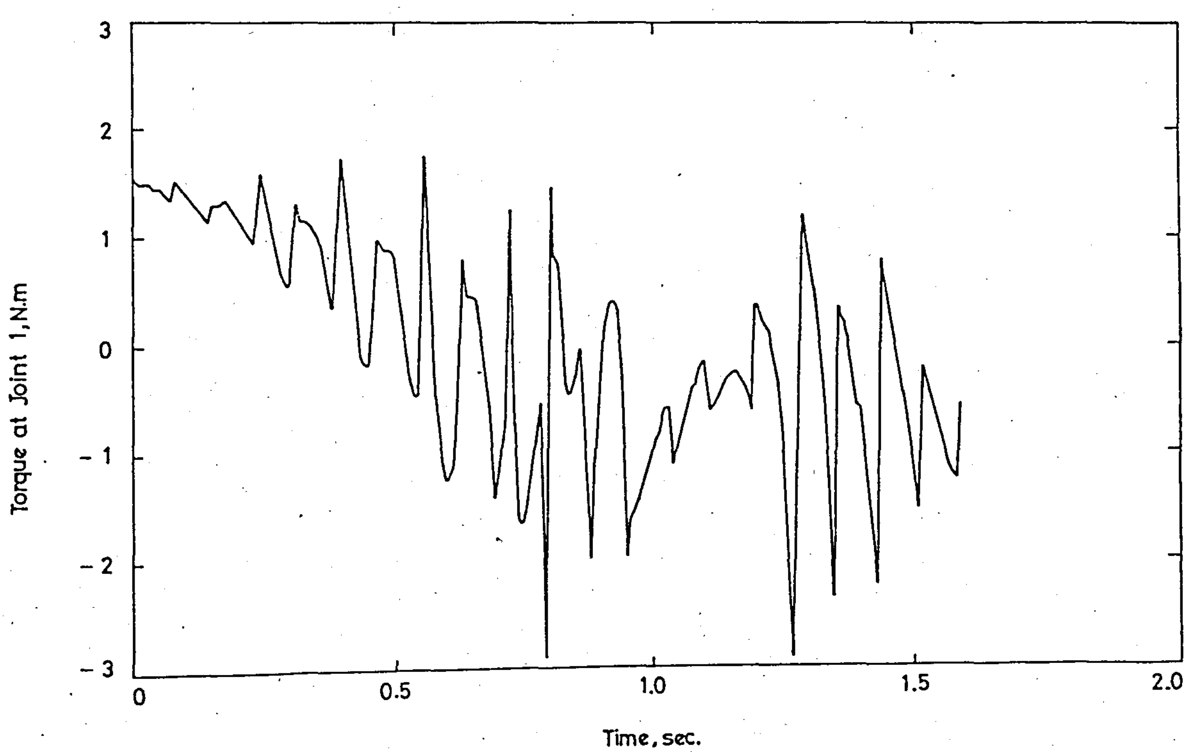


FIGURE 5.12 - Idealized torque τ_1 acting at first joint of the three-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.08 sec., TUP = 0.16 sec., and TGAIN = 0.01 sec.

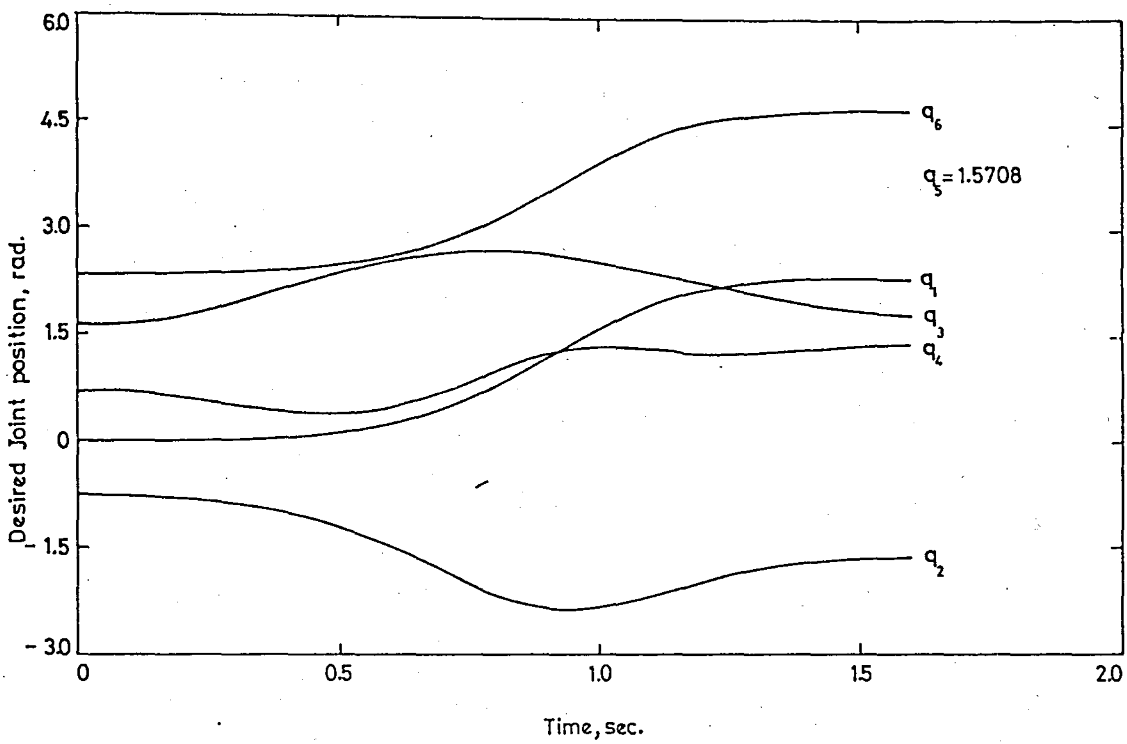


FIGURE 5.13 - Desired trajectories of joint angles for the six-degree-of-freedom model.

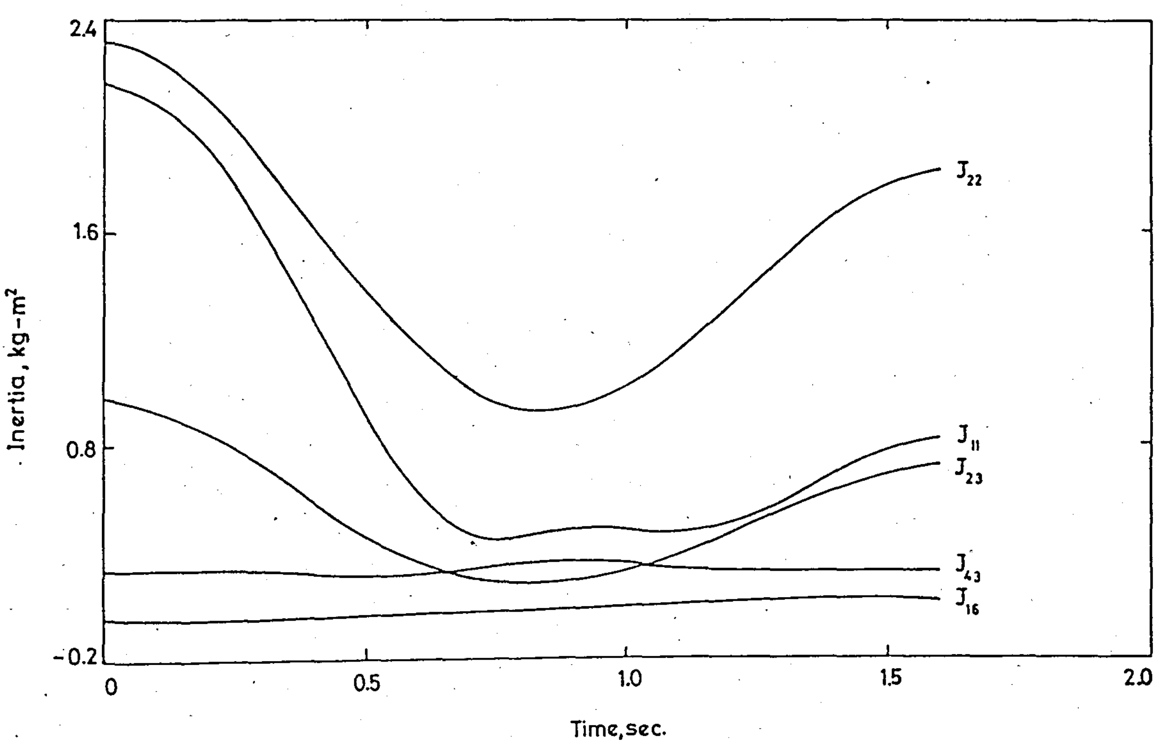
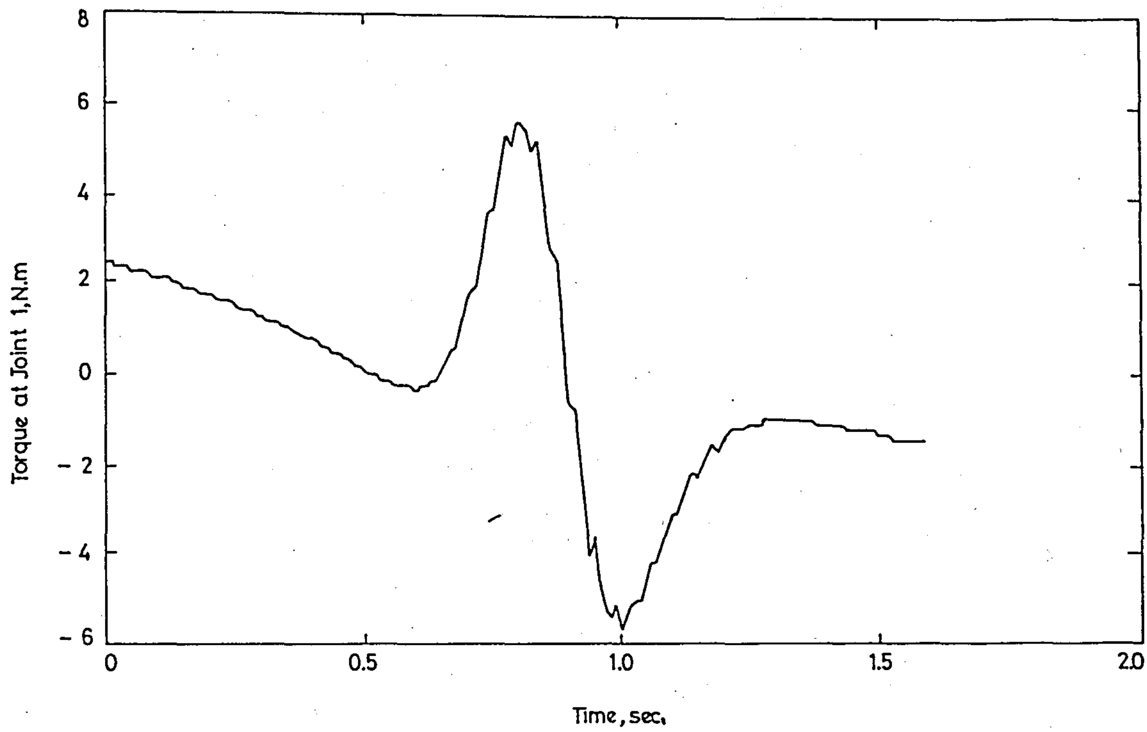
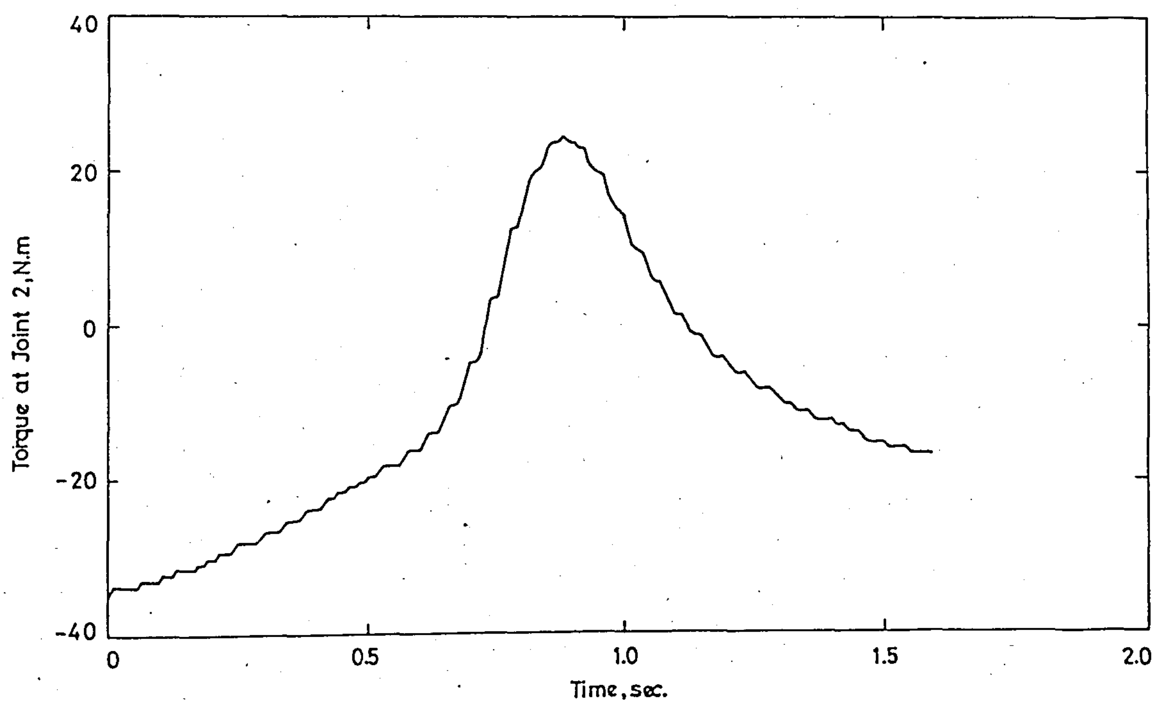


FIGURE 5.14 - Variation of the elements of the inertia matrix along the desired trajectories for the six-degree-of-freedom model.

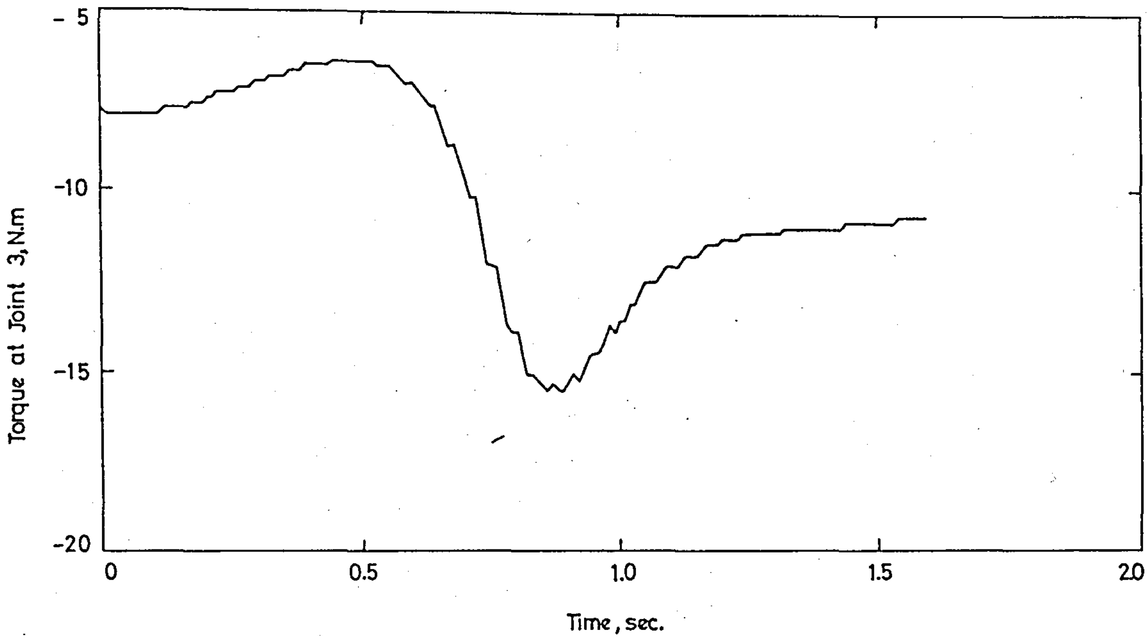


(a)

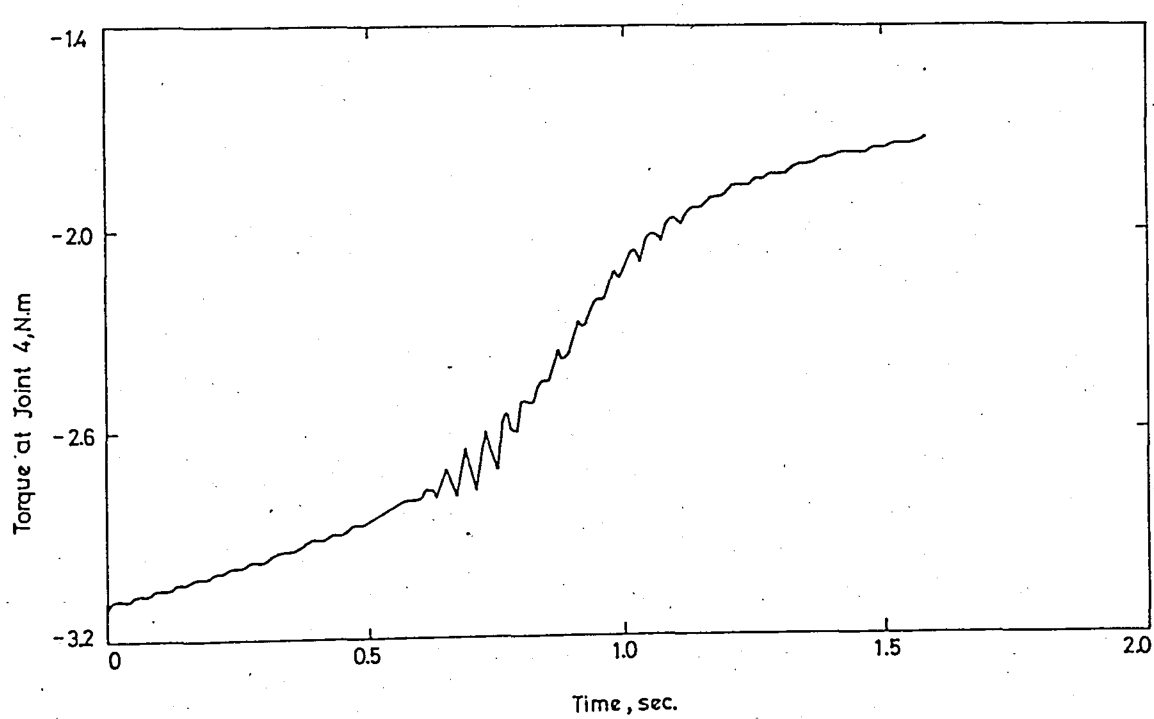


(b)

FIGURE 5.15 - Idealized torques acting at joints of the six-degree-of-freedom model obtained with the adaptive feedback law for TTG = 0.04 sec., TUP = 0.01 sec., and TGAIN = 0.01 sec.

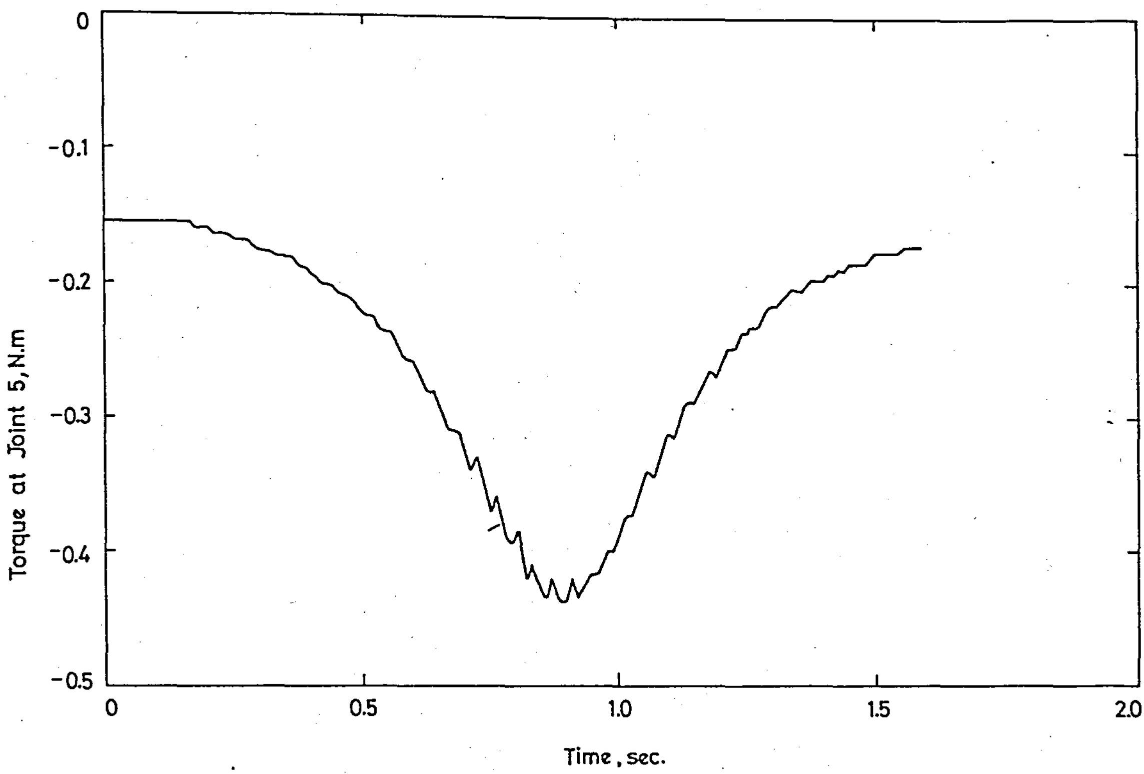


(c)

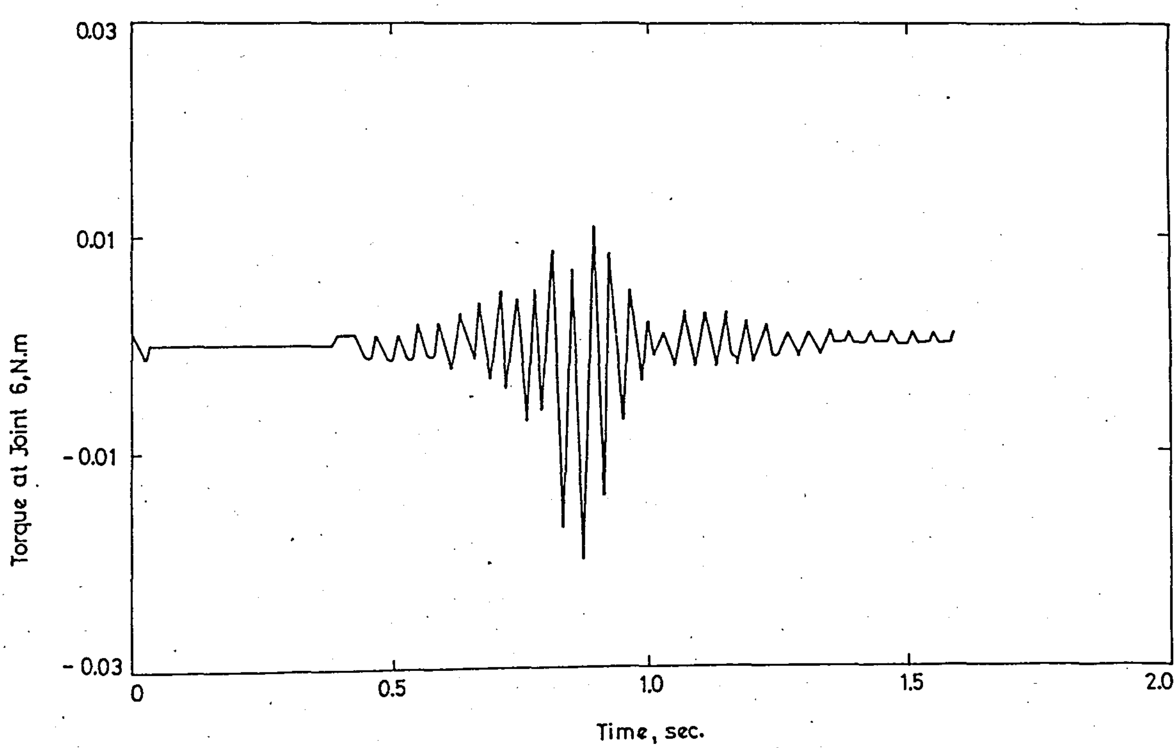


(d)

FIGURE 5.15 (continued).



(e)



(f)

FIGURE 5.15 (continued).

VI. CONCLUSIONS

The dynamic equations of a manipulator are very difficult to obtain analytically when the degree of freedom of the system exceeds three. Therefore a recursive algorithm is developed which forms the system matrices given the structure and design parameters of the system. The algorithm is general and allows the study of a kinematic chain of any degree of freedom with any combination of joints (translational or rotational). This systematic approach may also be used to obtain closed form expressions for the dynamic model in an easier way. Furthermore the second form of the dynamic model derived in Section 2.2.3 considers also the coupling terms and may lead to more efficient control algorithms.

Since a manipulator is a highly nonlinear and interactive system, control of it for position and velocity tracking is very difficult, and it has no general solution. The scheme adopted in this study is based on successive generation of a linear model using results of the dynamic modelling described in Chapter 2. Thus the nonlinear problem is solved with an adaptive scheme. Control algorithms proposed in this study have two main advantages:

i) Feedback matrix generation is fast and compatible with real time control constraints.

ii) Even though the minimum energy optimal control is adopted the tuning of the control system depends on a single basic parameter which is the horizon time, leading to a simple operation.

A general formulation of manipulator dynamics is necessary as a basis to the investigation of a manipulator with desired structure and its control. Proposed algorithms are general and may be used to investigate the dynamical behaviour of any manipulator. This general structure is obtained with a sacrifice on computer time but can serve as a software support for the CAD of industrial robots.

APPENDIX A

AN ALGORITHM FOR LINK COORDINATE SYSTEM ASSIGNMENT

Given an n degree of freedom manipulator the below algorithm assigns an orthonormal coordinate system to each link of the manipulator [12].

1. [Establish the base coordinate system]

   Establish a righthand orthonormal coordinate system $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$ at the supporting base with the $\vec{z}_0$ axis lying along the axis of motion of joint 1.

2. [Initialize and loop]

   For each i = 1,...,n perform steps 2.1 to 2.4.

   2.1 [Establish joint axis]

   Align the $\vec{z}_i$ with the axis of motion (rotating or sliding) of joint i+1.

   2.2 [Establish the origin of the i-th coordinate system]

   Locate the origin of the i-th coordinate system at the intersection of the $\vec{z}_i$ and $\vec{z}_{i-1}$ axes or at the intersection of common normals between the $\vec{z}_i$ and $\vec{z}_{i-1}$ axes and the $\vec{z}_i$ axis.

2.3 [Establish $\vec{x}_i$ axis]

Establish $\vec{x}_i = \pm(\vec{z}_{i-1} \times \vec{z}_i)/||\vec{z}_{i-1} \times \vec{z}_i||$ or along the common normal between the $\vec{z}_{i-1}$ and $\vec{z}_i$ axes when they are parallel.

2.4 [Establish $\vec{y}_i$ axis]

Assign $\vec{y}_i = (\vec{z}_i \times \vec{x}_i)/||\vec{z}_i \times \vec{x}_i||$ to complete the righthand coordinate system. Extend the $\vec{z}_i$ and $\vec{x}_i$ axes if necessary for steps 3.1 to 3.4.

3. [Find joint and link parameters]

For each i, i = 1,2,...,n perform steps 3.1 and 3.4.

3.1 [Find d_i]

d_i is the distance from the origin of the (i-1)th coordinate system to the intersection of the $\vec{z}_{i-1}$ axis and the $\vec{x}_i$ axis along the $\vec{z}_{i-1}$ axis. It is the joint variable if joint i is prismatic.

3.2 [Find a_i]

a_i is the distance from the intersection of the $\vec{z}_{i-1}$ axis and and the $\vec{x}_i$ axis to the origin of the ith coordinate system along the $\vec{x}_i$ axis.

3.3 [Find θ_i]

θ_i is the angle of rotation from the $\vec{x}_{i-1}$ axis to $\vec{x}_i$ axis about the $\vec{z}_{i-1}$ axis. It is the joint variable if joint i is revolute.

3.4 [Find α_i]

α_i is the angle of rotation from the $\vec{z}_{i-1}$ axis to the $\vec{z}_i$ axis about the $\vec{x}_i$ axis.

APPENDIX B

LINEAR FIXED END POINT MINIMUM ENERGY PROBLEM

In order to solve the optimal control problem given the linear time invariant system

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} + \vec{a}_o \tag{B.1}$$

with the initial condition $\vec{x}(t_o) = \vec{x}_o$, where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \vec{a}_o = \begin{bmatrix} 0 \\ \gamma \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \vec{u} = u ,$$

and the performance index

$$J = \frac{1}{2} \int_{t_o}^{t_f} \vec{u}^T R \vec{u} \, dt \quad , \tag{B.2}$$

with the terminal condition $\vec{x}(t_f) = \vec{0}$, the Hamiltonian is written as

$$H = \frac{1}{2} \vec{u}^T R \vec{u} + \vec{\lambda}^T [A\vec{x} + B\vec{u} + \vec{a}_o] \tag{B.3}$$

Application of the maximum principle requires that for an optimum system and unconstrained control

$$\frac{\partial H}{\partial u} = \vec{0} = R\vec{u} + B^T\vec{\lambda} \tag{B.4}$$

Solving Eq. (B.4) for the control vector $\vec{u}$ gives the relation

$$\vec{u} = -R^{-1}B^T\vec{\lambda} \tag{B.5}$$

Using the conditions $\partial H/\partial\vec{\lambda} = \dot{\vec{x}}$, and $-\partial H/\partial x = \dot{\vec{\lambda}}$, one obtains the system of equations

$$\dot{\vec{x}} = A\vec{x} - BR^{-1}B^T\vec{\lambda} + \vec{a}_o \tag{B.6}$$

$$\dot{\vec{\lambda}} = -A^T\vec{\lambda} \tag{B.7}$$

with boundary conditions given only for $\vec{x}$. Solution of Eq. (B.6) for $t \varepsilon [t,t_f]$ gives

$$\vec{x}(t_f) = e^{A(t_f-t)}\,\vec{x}(t) - \int_t^{t_f} e^{A(t_f-\tau)}BR^{-1}B^T\vec{\lambda}(\tau)d\tau + \int_t^{t_f} e^{A(t_f-\tau)}a_o d\tau \tag{B.8}$$

Solving Eq. (B.7) for $t \varepsilon [\tau,t]$ gives

$$\vec{\lambda}(\tau) = e^{-A^T(\tau-t)}\vec{\lambda}(t) \tag{B.9}$$

Replacing Eq. (B.9) in (B.8) and making a change of variables as $\tau^* = \tau-t$, and letting $T = t_f-t$ one gets

$$\vec{x}(t_f) = e^{AT}x(t) - e^{AT}\lambda(t)\int_0^T e^{-A\tau^*}\,BR^{-1}B^Te^{-A^T\tau^*}\,d\tau^* + \int_0^T e^{A(T-\tau^*)}a_o d\tau^* \tag{B.10}$$

Since $\vec{x}(t_f)$ being constrained to zero, solving Eq. (B.10) for $\vec{\lambda}(t)$ gives

$$\vec{\lambda}(t) = M^{-1}(t)\vec{x}(t) + M^{-1}(t)F_2^T(t)H(t)\vec{a}_o \ , \qquad \text{(B.11)}$$

where

$$M(t) = \int_0^T e^{-A\tau^*} BR^{-1}B^T e^{-A^T\tau^*} \, d\tau^* \ .$$

$$F_2^T(t) = e^{-AT} \qquad \text{and} \qquad H(t) = \int_0^T e^{A(T-\tau^*)} d\tau^*$$

Hence the control vector is obtained by substituting Eq. (B.11) into Eq. (B.5) as

$$\vec{u}(t) = -R^{-1}B^TM^{-1}(t)\vec{x}(t) - R^{-1}B^TM^{-1}(t)F_2^T(t)H(t)\vec{a}_o \qquad \text{(B.12)}$$

If the expression for control is evaluated using the matrices given in Eq. (B.1), and taking $R = \rho$, one obtains

$$u(t) = \frac{6}{T^2}(q_d - q) + \frac{4}{T}(\dot{q}_d - \dot{q}) + \gamma(t), \quad t \ \varepsilon \ (t_o, t_f) \qquad \text{(B.13)}$$

where T is called time-to-go.

APPENDIX C

INPUT DATA FOR THE SIMULATION PROGRAM

A description of input variables of the simulation program are given below:

i) <u>Geometric and Dynamic Parameters of the Manipulator</u>:

N	: Number of degrees of freedom (d.o.f.)
N2	: 2 x N
JT(I)	: N dimensional array which specifies joint types (JT(I) = 0 if joint-i is rotational, JT(I) = 1 if joint-i is translational.
ALFD(I), THTD(I), A(I), D(I)	: N dimensional arrays which give link coordinate parameters corresponding to α_i, θ_i, a_i, and d_i.
MASS(I)	: N dimensional array which specifies the mass of each link.
S(I,J)	: (3xN) matrix, i-th column of which gives the mass center of link i referred to its own coordinates $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$.
INR(I,J)	: (9xN) matrix i-th column of which gives the inertia matrix of link i about its center of mass referred to its own coordinates. (The entries of i-th column of INR are obtained

by storing the elements of each three columns of the inertia matrix, $\bar{I}_i$, starting from the first column).

VISFC(I) : N dimensional array which specifies the viscous friction coefficients associated with each joint.

ii) Input Data Related with Desired Trajectories:

NS : Number of straight line segments in the desired trajectory of the manipulator tip.

TF(I) : NS dimensional array which gives the travel time of each straight line segment.

TAC : Acceleration and deceleration period for the velocity profiles given by subroutines HTASK2 and HTASK 3.

POINT(I,J) : (3 x (NS+1)) matrix which stores the Cartesian coordinates of corner points of the desired path given with respect to base frame $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$.

TASK : Control variable used to select the desired acceleration profile among the three profiles shown in Fig. 4.2. (Given a value of 1,2, or 3 corresponding to the acceleration profiles given by subroutines HTASK1, HTASK2, and HTASK3, respectively).

iii) Input Data Related with the Control Scheme:

TTG : Horizon time (or time-to-go).

TUP : Update period of system matrices.

TGAIN : Update period of feedback gains and open loop control.

iv) Input Data Related with the Simulation:

HF(I), HM(I)	: 3 dimensional arrays which give the force and moment vectors exerted by the hand upon an external object given in terms of base coordinates.
HXX	: Step size of integration.
IFORM	: Control variable used to select the desired form of the dynamic model. (Given a value of 1,2, or 3 corresponding to the models formed by subroutines SYST1, SYST2, and SYST3, respectively).
IHF	: Control variable which specifies whether the load at hand will be taken into account or not while generating the control. (Load is not taken into account when it is given the value zero).

APPENDIX D

KINEMATIC EQUATIONS FOR THE MANIPULATORS

In this study two manipulator models are considered; a three degree of freedom model, and a six degree of freedom model. The geometric and dynamic parameters of the first three links of both models have the same values. The six degree of freedom model is shown in Fig. D.1 with the link coordinate systems attached to it.

D.1 THREE DEGREE OF FREEDOM MODEL

D.1.1 Solution for the Joint Angles

The homogeneous transformation matrices for the three degree of freedom model are obtained using the coordinate parameters of the first three links of the six degree of freedom model given in Table D.1.

$$A_0^1 = \begin{bmatrix} C\theta_1 & 0 & -S\theta_1 & 0 \\ S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (D.1)$$

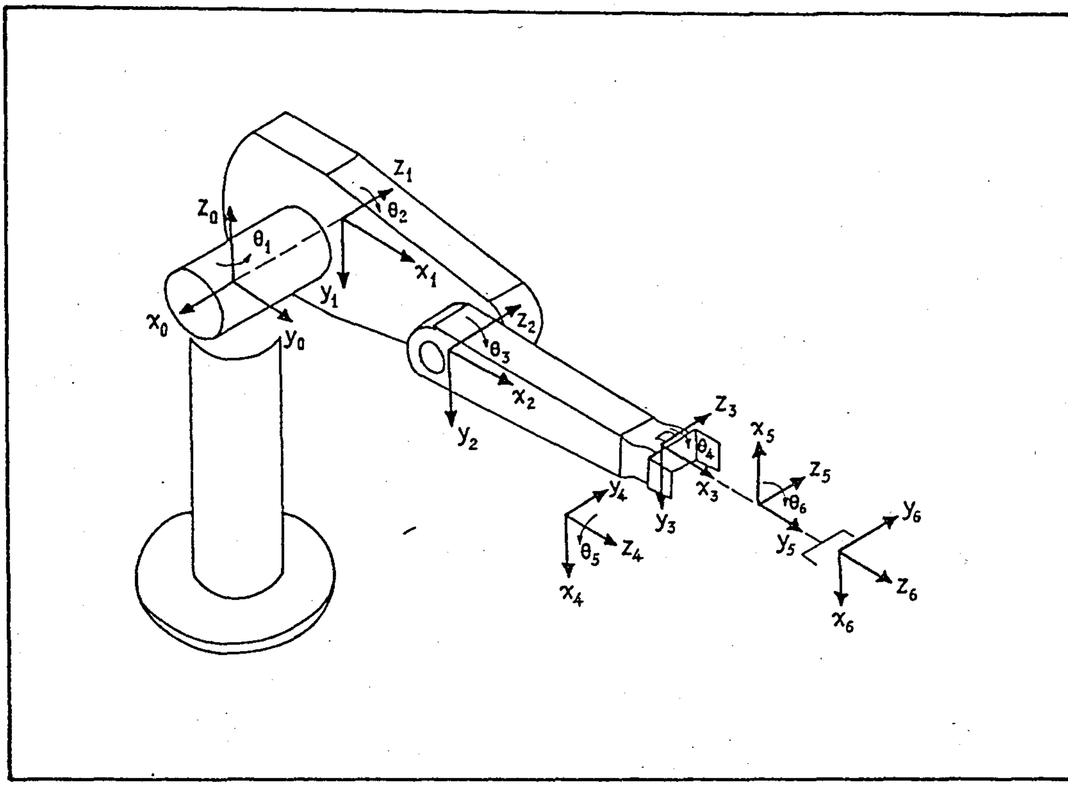


FIGURE D.1 - Link coordinate systems for the six-degree-of-freedom manipulator model.

TABLE D.1 - Link Coordinate Parameters for the Six-Degree-of-Freedom Manipulator Model

JOINT-i	α_i (deg)	θ_i	a_i (m)	d_i (m)
1	-90	θ_1	0	0
2	0	θ_2	0.6	0
3	0	θ_3	0.5	0
4	90	θ_4	0	0
5	90	θ_5	0	0.24
6	90	θ_6	0	0

$$A_1^2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.2}$$

$$A_2^3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_3C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & a_3S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.3}$$

where S and C refer to sine and cosine, respectively.

In order to find the joint coordinates corresponding to a given position of the manipulator tip it is necessary to solve the matrix equation

$$H = A_o^3 = \begin{bmatrix} \vec{n} & \vec{s} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.4}$$

where the transformation matrix A_o^3 is obtained using the matrices given by Eqs. (D.1-D.3) as

$$A_o^3 = A_o^1 \cdot A_1^2 \cdot A_2^3 = [a_{k\ell}] \qquad k,\ell = 1,\ldots,4 \tag{D.5}$$

Equating the elements of the last column of matrix A_o^3 to the components of the position vector $\vec{p}$, one obtains the basic kinematic equations for the three degree of freedom model as

$$a_{14} = a_2C\theta_1C\theta_2 + a_3C\theta_1C\theta_{23} = p_x \tag{D.6}$$

$$a_{24} = a_2S\theta_1C\theta_2 + a_3S\theta_1C\theta_{23} = p_y \tag{D.7}$$

$$a_{34} = a_2S\theta_2 + a_3S\theta_{23} = p_z \tag{D.8}$$

Solving Eqs. (D.6-D.8) for joint angles, one gets

$$\theta_1 = \tan^{-1}\frac{p_y}{p_x} \tag{D.9}$$

$$\theta_2 = \sin^{-1}\frac{-p_z}{(a_2^2 + 2a_2a_3C\theta_3 + a_3^2)^{\frac{1}{2}}} - \tan^{-1}\frac{a_3S\theta_3}{a_2 + a_3C\theta_3} \tag{D.10}$$

$$\theta_3 = \cos^{-1}\frac{c^2 + p_z^2 - a_2^2 - a_3^2}{2a_2a_3} \tag{D.11}$$

where

$$c = \begin{cases} p_x/C\theta_1 \\ p_y/S\theta_1 \end{cases}, \qquad \text{if} \quad C\theta_1 = 0 \tag{D.12}$$

D.1.2 Solution for Joint Velocities

In order to control the manipulator, it is also necessary to determine the desired joint velocities. Therefore Eqs. (D.9-D.11) are differentiated, yielding

$$\dot{\theta}_1 = \frac{p_yC\theta_1 - p_xS\theta_1}{p_xC\theta_1 + p_yS\theta_1} \tag{D.13}$$

$$\dot{\theta}_2 = -\frac{\dot{p}_z + a_3C_{23}\dot{\theta}_3}{a_2C\theta_2 + a_3C\theta_{23}} \tag{D.14}$$

$$\dot{\theta}_3 = - \frac{c\dot{c} + p_z\dot{p}_z}{a_2a_3S\theta_3} \tag{D.15}$$

where $\dot{p}_x$, $\dot{p}_y$ and $\dot{p}_z$ are components of the vector $\dot{\vec{p}}$, which defines the velocity of the manipulator tip; and c and $\dot{c}$ are given by Eqs. (D.12) and (D.16).

$$\dot{c} = \begin{cases} (\dot{p}_x + p_x\dot{\theta}_1S\theta_1)/C\theta_1 & \\ (\dot{p}_y - p_y\dot{\theta}_1C\theta_1)/S\theta_1 & , \quad \text{if} \quad C\theta_1 = 0 \end{cases} \tag{D.16}$$

D.1.3 Solution for Joint Accelerations

Equations (D.13), (D.14) and (D.15) are differentiated once more in order to obtain the desired joint accelerations as

$$\ddot{\theta}_1 = \frac{\ddot{p}_yC\theta_1 - \ddot{p}_xS\theta_1 - 2\dot{\theta}_1(\dot{p}_yS\theta_1 + \dot{p}_xC\theta_1) + \dot{\theta}_1^2(p_xS\theta_1 - p_yC\theta_1)}{p_xC\theta_1 + p_yS\theta_1} \tag{D.17}$$

$$\ddot{\theta}_2 = \frac{a_3S\theta_{23}(\dot{\theta}_2 + \dot{\theta}_3)^2 - \ddot{p}_z + a_2S\theta_2\dot{\theta}_2^2 - a_3C\theta_{23}\ddot{\theta}_3}{a_2C\theta_2 + a_3C\theta_{23}} \tag{D.18}$$

$$\ddot{\theta}_3 = - \frac{\dot{c}^2 + c\ddot{c} + \dot{p}_z^2 + p_z\ddot{p}_z + a_2a_3\dot{\theta}_3C\theta_3}{a_2a_3S\theta_3} \tag{D.19}$$

where $\ddot{p}_x$, $\ddot{p}_y$ and $\ddot{p}_z$ are components of the vector $\ddot{\vec{p}}$, which defines the acceleration of the manipulator tip; and c, $\dot{c}$, and $\ddot{c}$ are given by Eqs. (D.12), (D.16), and (D.20).

$$\ddot{c} = \begin{cases} \dfrac{p_x\dot{\theta}_1^2 + \dot{p}_x\dot{\theta}_1 S\theta_1 + (\ddot{p}_x + \dot{p}_x\dot{\theta}_1 S\theta_1 + p_x\ddot{\theta}_1 S\theta_1)C\theta_1}{C\theta_1^4} \\ \dfrac{p_y\dot{\theta}_1^2 - \dot{p}_y\dot{\theta}_1 C\theta_1 + (\ddot{p}_y - \dot{p}_y\dot{\theta}_1 C\theta_1 - p_y\ddot{\theta}_1 C\theta_1)S\theta_1}{S\theta_1^4} \end{cases}, \quad \text{(D.20)}$$

if $C\theta_1 = 0$

D.2 SIX DEGREE OF FREEDOM MODEL

D.2.1 Solution for the Joint Angles

The homogeneous coordinate transformation matrices for the first three links of the six degree of freedom model are the same as for the three degree of freedom model, given by Eqs. (D.1-D.3). The transformation matrices of the remaining links are obtained using the link parameters given in Table D.1.

$$A_3^4 = \begin{bmatrix} C\theta_4 & 0 & S\theta_4 & 0 \\ S\theta_4 & 0 & -C\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(D.21)}$$

$$A_4^5 = \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(D.22)}$$

$$A_5^6 = \begin{bmatrix} C\theta_6 & 0 & S\theta_6 & 0 \\ S\theta_6 & 0 & -C\theta_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.23}$$

In order to solve for joint angles, given the position and orientation of the manipulator tip, the transformation matrix A_1^6 is formed as follows:

$$A_1^6 = A_1^2 \cdot A_2^3 \cdot A_3^4 \cdot A_4^5 \cdot A_5^6 = [a_{k\ell}] \qquad k,\ell = 1,\ldots,4 \tag{D.24}$$

where the elements of the third column of matrix A_1^6 are given by

$$a_{31} = S\theta_5 C\theta_6; \qquad a_{32} = -C\theta_5\ ; \qquad a_{33} = S\theta_5 S\theta_6\ ; \qquad a_{34} = 0 \tag{D.25}$$

The transformation matrix A_1^6 can also be obtained from the arm matrix as

$$A_1^6 = A_1^0 \cdot A_0^6 = A_1^0 \cdot H$$

$$= \begin{bmatrix} C\theta_1 & S\theta_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.26}$$

where A_1^0 is obtained by taking the inverse of the transformation matrix A_0^1. After doing the multiplication shown in Eq. (D.26) and equating the elements of the third column of the resulting matrix to the corresponding elements given by Eq. (D.25), one gets

$$n_y C\theta_1 - n_x S\theta_1 = S\theta_5 C\theta_6 \tag{D.27}$$

$$s_y C\theta_1 - s_x S\theta_1 = -C\theta_5 \tag{D.28}$$

$$a_y C\theta_1 - a_x S\theta_1 = S\theta_5 S\theta_6 \tag{D.29}$$

$$p_y C\theta_1 - p_x S\theta_1 = 0 \tag{D.30}$$

Joint angles θ_1, θ_5 and θ_6 are solved from Eqs. (D.27-D.30), yielding

$$\theta_1 = \tan^{-1} \frac{p_y}{p_x} \tag{D.31}$$

$$\theta_5 = \cos^{-1}(s_x S\theta_1 - s_y C\theta_1) \tag{D.32}$$

$$\theta_6 = \tan^{-1} \frac{a_y C\theta_1 - a_x S\theta_1}{n_y C\theta_1 - n_x S\theta_1} \tag{D.33}$$

The remaining joint angles are solved using a geometric approach. Since the joint angles θ_5 and θ_6 are known, the position vector $\vec{p}_3$ that points from the origin of the base frame $(\vec{x}_o,\vec{y}_o,\vec{z}_o)$ to the origin of the coordinate frame $(\vec{x}_3,\vec{y}_3,\vec{z}_3)$ can be derived as

$$\vec{p}_3 = \vec{p} - d_5\vec{y}_5 \tag{D.34}$$

where $\vec{y}_5$ is given by the second column of the transformation matrix A_o^5, as

$$\vec{y}_5 = S\theta_6\vec{n} - C\theta_6\vec{a} \tag{D.35}$$

Thus, the solutions for the joint angles θ_2 and θ_3 obtained for the three degree of freedom model can be used by the substitution of the components of the vector $\vec{p}_3$ for the corresponding components of the vector $\vec{p}$ in Eqs. (D.10-D.12).

The joint angle θ_4 is obtained using the a_{14} and a_{24} elements of the matrix equation

$$A_2^6 = A_2^0 \cdot H \tag{D.36}$$

yielding

$$\theta_4 = \tan^{-1}\{C_1/C_2\} - \theta_3 \tag{D.37}$$

where

$$C_1 = wC\theta_2 - p_zS\theta_2 - a_2 - a_3C\theta_3$$

$$C_2 = wS\theta_2 + p_zS\theta_2 + a_3S\theta_3$$

with $\quad w = p_xC\theta_1 + p_yS\theta_1 .$

D.2.2 Solution for the Joint Velocities

Joint velocities $\dot{\theta}_1$, $\dot{\theta}_2$ and $\dot{\theta}_3$ are the same as for the three degree of freedom model and they are given by Eqs. (D.13-D.16) where components of the vectors $\vec{p}_3$ and $\dot{\vec{p}}_3$ should be substituted for the corresponding components of the vectors $\vec{p}$ and $\dot{\vec{p}}$ in Eqs. (D.14-D.16).

The remaining joint velocities are derived as

$$\dot{\theta}_4 = \frac{C\theta_{34}^2}{C_2}(\dot{C}_1 - \dot{C}_2\tan\theta_{34}) - \dot{\theta}_3 \tag{D.38}$$

where

$$\dot{C}_1 = (\dot{w} - p_z\dot{\theta}_2)C\theta_2 - (w\dot{\theta}_2 + \dot{p}_z)S\theta_2 + a_3\dot{\theta}_3S\theta_3$$

$$\dot{C}_2 = (\dot{w} - p_z\dot{\theta}_2)S\theta_2 + (w\dot{\theta}_2 + \dot{p}_z)C\theta_2 + a_3\dot{\theta}_3C\theta_3$$

with $\quad \dot{w} = (\dot{p}_x + p_y\dot{\theta}_1)C\theta_1 + (\dot{p}_y - p_x\dot{\theta}_1)S\theta_1$

$$\dot{\theta}_5 = \frac{(\dot{s}_y - s_x\dot{\theta}_1)C\theta_1 - (\dot{s}_x + s_y\dot{\theta}_1)S\theta_1}{S\theta_5} \tag{D.39}$$

if $\theta_5 = 0$

$$\dot{\theta}_5 = \frac{(\dot{a}_y - a_x\dot{\theta}_1)C\theta_1 - (\dot{a}_x + a_y\dot{\theta}_1)S\theta_1 - \dot{\theta}_6 S\theta_5 C\theta_6}{C\theta_5 S\theta_6} \tag{D.40}$$

if $\theta_5 = 0$, and $\theta_6 = 0$

$$\dot{\theta}_5 = \frac{(\dot{n}_y - n_x\dot{\theta}_1)C\theta_1 - (\dot{n}_x + n_y\dot{\theta}_1)S\theta_1 + \dot{\theta}_6 S\theta_5 S\theta_6}{C\theta_5 C\theta_6} \tag{D.41}$$

$$\dot{\theta}_6 = \frac{e_1 C\theta_1 - e_2 S\theta_1}{n_y C\theta_1 - n_x C\theta_1} \tag{D.42}$$

where

$$e_1 = (\dot{a}_y - a_x\dot{\theta}_1)C\theta_6^2 - (\dot{n}_y - n_x\dot{\theta}_1)S\theta_6 C\theta_6$$

$$e_2 = (\dot{a}_x + a_y\dot{\theta}_1)C\theta_6^2 - (\dot{n}_x + n_y\dot{\theta}_1)S\theta_6 C\theta_6 \quad .$$

BIBLIOGRAPHY

1. Vukobratovic, M., and Potkonjak, V., "Contribution to Computer Construction of Active Chain Models via Lagrangian Form", J. Applied Mechanics, Vol. 46, pp. 181-185, 1979.

2. Hollerbach, J.M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", IEEE Trans. Syst., Man, Cybern., Vol. SMC-10, pp. 730-736, 1980.

3. Stepanenko, Yu., and Vukobratovic, M., "Dynamics of Articulated Open-Chain Active Mechanisms", Math. Biosciences, Vol. 28, pp. 137-170, 1976.

4. Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "On-Line Computational Scheme for Mechanical Manipulators", J. Dynamic Syst., Measurement, Contr., Vol. 102, pp. 69-76, 1980.

5. Cvetkovic, V., and Vukobratovic, M., "Computer-Oriented Algorithm for Modeling Active Spatial Mechanisms for Robotics Applications", IEEE Trans. Syst., Man, Cybern., Vol. SMC-12, pp. 838-847, 1982.

6. Paul, R.P.C., "Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm", Stanford Artificial Intelligence Laboratory Memo, No. 177, 1972.

7. Whitney, D.E., "Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Trans. Man-Machine Syst., Vol. MMS-10, pp. 47-53, 1969.

8. 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.

9. 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.

10. Dubowsky, S., and DesForges, D.T., "The Application of Model-Referenced Adaptive Control to Robotic Manipulators", J. Dynamic Syst., Measurement, Contr., Vol. 101, pp. 193-200, 1979.

11. Denavit, J., and Hartenberg, R.S., "A Kinematic Notation for Lower-Pair Mechanisms", J. Applied Mechanics, Vol. 22, pp. 215-221, 1955.

12. Lee, C.S.G., "Robot Arm Kinematics, Dynamics, and Control", Computer, Vol. 15, pp. 62-80, 1982.

13. Köseoğlu, A., "Manipulator Control", Graduation Project Report, Boğaziçi University, 1982.

14. Kuzucu, A., and Roch, A., "Suboptimal Adaptive Feedback Control of Nonlinear Systems", Lecture Notes in Control and Information Sciences, Vol. 24, pp. 131-140, 1980.

15. Raibert, M.H., and Horn, B.K.P., "Manipulator Control Using the Configuration Space Method", The Industrial Robot, Vol. 5, pp. 69-73, 1978.