NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

ISBN 0-315-56096-7

Robust Adaptive Control of Robot Manipulators

Raad A Al-Ashoor

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

January 1990

ABSTRACT

Robust Adaptive Control of Robot Manipulators

Raad A Al-Ashoor, Ph D.
Concordia University, 1990

It is well-known that the dynamic behavior of even the simplest multi-link manipulators is described by highly nonlinear and strongly coupled differential equations During the last decade, many papers have been written that discuss the control of multi-link manipulators taking their dynamic behavior into account It has been noted that the use of conventional linear control techniques limits the dynamic performance of such manipulators and makes it difficult, if not impossible, to obtain uniformly high performance over a wide range of tasks in a dynamically changing environment This limitation arises primarily because of the highly nonlinear nature of the dynamic characteristics of robot manipulators and the degradation in their dynamic performance because of the unmodeled inertia properties of the objects being manipulated This has led to the development of advanced control schemes such as adaptive control which attempt to take into account the effect of the dynamics that are unmodeled or difficult to model accurately

This thesis is concerned with the problem of adaptive control of rigid-link robot manipulators The main objective of the research is to provide a powerful and robust adaptive control technique that will maintain closed loop stability and achieve trajectory tracking in the presence of model uncertainty due to unmodeled dynamics and parameter variations

An on-line adaptive control scheme for rigid link and flexible-joint robot manipulators in joint space and Cartesian space is described The scheme uses least squares identification to generate a set of piecewise linear time-invariant models on-line for the

nonlinear, time-varying dynamics of the robot manipulator. Feedforward and feedback controllers are then used, which are updated using the identified parameters, to achieve trajectory tracking in the presence of modeling uncertainties and unmodeled dynamics. The adaptive control scheme has a decoupled structure. The effects of coupling between the joints and other unmodeled dynamics are taken into account by including a correction term in the control law which attempts to reduce the errors resulting from these effects to zero. This correction term gives the adaptive controller the desired robustness characteristic which allows the reference trajectory to be tracked in the presence of payload and parameter variations. The feedback and feedforward controller parameters are selected and updated on-line to ensure asymptotic stability of the closed-loop system. Several simulation studies are carried out to show the applicability of the proposed robust adaptive control scheme.

TO MY CHILDREN

Haider and Noor

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor, Prof. R.V. Patel, who contributed a great deal of his time, effort, and ideas to the work presented in this dissertation. He has been a good advisor and teacher throughout my study at Concordia

I would like to thank my co-supervisor Prof A.J Al-Khalili, who has encouraged and advised me and has been a good friend to me

Also, I would like to thank Dr. K. Khorasani for his suggestions and contribution in the work presented on flexible joint manipulators

Finally, I wish to express my thanks to my family, particularly my wife Anan for her patience and understanding, and to my children Haider and Noor Without them I could not have stayed in Canada.

TABLE OF CONTENTS

LIST OF SYMBOLS

Unless mentioned otherwise, the following notations are used in this thesis

N	Number of degrees-of-freedom of the end-effector of the manipulator
n	Number of joints
I	$n \times n$ identity matrix
τ	$n \times 1$ Generalized joint torque vector
f	$N \times 1$ Cartesian force vector
$\theta(\theta,\dot{\theta})$	$n \times 1$ Joint position (velocity, acceleration) vector
$x(x,\dot{x})$	$N \times 1$ Cartesian (end-effector) position (velocity, acceleration) vector
$J(\theta)$	$n \times n$ Jacobian matrix of the manipulator
$M(\theta)$	$n \times n$ Positive-definite inertia matrix
$V(\theta,\dot{\theta})$	$n \times 1$ Centrifugal Coriolis torque vector
$G(\theta)$	$n \times 1$ Gravitational torque vector
$\eta(t)$	$n \times 1$ Torque vector representing unmodeled dynamics such as those due to friction, backlash, etc
$M_x(\theta)$	$N \times N$ Cartesian mass matrix
$V_x(\theta,\dot{\theta})$	$N \times 1$ Cartesian Coriolis centrifugal force vector
$G_x(\theta)$	$N \times 1$ Cartesian vector of gravitational forces
v	$N \times 1$ Cartesian velocity of the end-effector
$e(e,\dot{e})$	$n \times 1$ Generalized joint position (velocity, acceleration) error vector
$e_x(e_x,\dot{e}_x)$	$N \times 1$ Generalized Cartesian position (velocity, acceleration) error vector of the end-effector

e_θ	$n \times 1$ Prediction error vector
Γ	Generalized loss function
ν	Forgeting factor
$\hat{\Theta}$	Vector of estimated parameters
Ψ	Vector of measured variables
ω	Undamped natural frequency
ς	Damping factor
z_a	$2n \times 1$ Actual error vector in joint space
z_{a_x}	$2N \times 1$ Actual error vector in Cartesian space
$\lambda(A)$	Eigenvalue of the matrix A
ϵ	Elasticity parameter
ψ	$n \times 1$ Elastic force/torque vector at the joint
$\phi(\dot{\phi},\ddot{\phi})$	$n \times 1$ Vector of position (velocity, acceleration) of actuator motor
Υ	$n \times 1$ Integral manifold of ψ
ξ	$n \times 1$ Vector of fast dynamics corresponding to joint flexibility
Υ_0	$n \times 1$ Quasi-steady state of the elastic variable
J_m	Moment of inertia of the actuator motor
J_l	Moment of inertia of the links

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION AND LITERATURE REVIEW

1.1 INTRODUCTION

Industrial robots are multifunctional programmable devices which manipulate and transport manufacturing components in order to perform manufacturing tasks, which are physically demanding, menial, or repetitive for human operators to do efficiently. Manipulators have been used extensively in hostile environments, such as in nuclear power plant and waste handling, deep under-sea exploration, and maintenance operations, and in space. Typically in these applications, the manipulator is controlled by a remote human operator in what is often termed a teleoperator configuration, only requiring the use of relatively simple control laws. Manipulators have also been used increasingly in industrial automation applications without the involvement of human operators. In both types of applications, operating speeds and performance indices are relatively low, and relatively simple control systems have proven adequate.

Present day industrial robots operate with very simple controllers, which do not take full advantage of the inexpensive computer power that has become available. The result is that these fairly expensive mechanisms are not being utilized to their full potential in terms of the speed and precision of their movements. With a more powerful control computer, it is possible to use a dynamic model of the manipulator as the heart of a sophisticated control algorithm [33]. This dynamic model allows the control algorithm to know how to control the manipulator's actuators in order to compensate for the complicated effects of inertia, centrifugal, Coriolis, gravity, and frictional forces when the robot is in motion. The result is that the manipulator can be made to follow a desired trajectory through space with smaller tracking errors, or perhaps move faster while maintaining good tracking performance.

Direct digital control of manipulators is one of the most promising applications of this new generation of computers. With the precision, speed, programming versatility and low cost of these computers, it is expected that manipulator systems will be able to perform complex tasks at high speeds automatically, making possible a wide range of new manipulator applications in high productivity industries as well as in environmental and energy-related fields. With increased demand on manipulator performance will come the need for improved manipulator control techniques.

The use of conventional linear control techniques limits the basic dynamic performance of a manipulator. This is a consequence of several factors such as the dynamic characteristics of a general spatial manipulator which are highly nonlinear functions of position and velocities of the manipulator elements, and the degradation of the dynamic performance characteristics of manipulators by the inertia properties of the objects being manipulated [1,8,10]. Hence, in general, it is not possible to design a linear control system which will yield uniformly high system performance over a wide range of manipulator tasks. So it is necessary to apply an advanced control scheme such as adaptive control so that the control algorithm can update the model terms as well as the controller parameters with time, according to the change of the manipulator's environment thereby compensating for the nonlinearities due to the manipulator dynamics, frictional forces and disturbances acting on the manipulator.

A new algorithm is presented in this thesis that is based on the adaptive least-square identification technique. In this algorithm the complex dynamic equations and parameter values of the robot are not used in generating the control action. This is one of the major advantages of the approach. The task of an adaptive controller is to adjust its gains based on the response of the robot in such a way that the performance of the robot closely matches the desired one.

The adaptation scheme uses on-line identification of the unknown matrices of a

linear time-varying model. Its parameters are used to generate the input control torque for the manipulator. A proportional plus derivative (PD) state feedback is used to maintain stability of the closed-loop system. The feedback gains are updated linearly in terms of the identified parameters to satisfy the required tracking performance. The algorithm is robust in the sense that it takes into account the effects of unmodeled dynamics and parameter and payload variations. A new method for robustness and stability analysis is proposed to maintain stability of the resulting closed-loop system in the presence of the unmodeled dynamics. The desired path can be described either in joint space or in Cartesian space. The proposed control scheme has a simple structure, is computationally fast and does not require knowledge of the physical parameters of the robot and/or of the payload. The adaptive controller does not use any nonlinear terms in the feedforward or feedback paths.

In this thesis, for the sake of analysis and simulation, the manipulator is modeled as a set of n moving rigid bodies connected in a serial chain with one end fixed on the ground and the other end free. The bodies are joined together with revolute joints with sensors incorporated in each joint to enable the position, velocity and acceleration of each joint to be measured. An actuator is provided at each joint to apply a torque on the next neighboring link. The number of degrees of freedom, is the number of independent joint position variables, usually equal to the number of joints. In this thesis a revolute joint manipulator is considered. However the algorithm can also be applied to a manipulator that contains prismatic joints. Elasticity at the joints is also considered and the algorithm proposed for rigid links is modified for this class of manipulators. Several methods are available for formulating the dynamic behavior of a rigid-link manipulator [1,3,5]. The Newton-Euler formulation is computationally more efficient and is used in this thesis for modeling and simulation of the manipulator.

1.2 LITERATURE REVIEW

During the past decade, many schemes for direct digital control of manipulators have been proposed. Most of the approaches are based on a mathematical model derived on the basis of Newtonian mechanics [33] and are, in general, not efficient. Such schemes belong to the class of computed torque methods [8-11,91,92] which involve the computation of the inverse dynamics in the feedforward path. Paul [8] and Bejczy [91], use a computed torque method for calculating the torques required to follow a nominal trajectory. The method involves a considerable amount of computation and memory storage. The resolved motion rate control and resolved acceleration control techniques developed by Whitney [9] and Luh et al. [10], compute the joint angle rates and accelerations so as to cause the end point of the manipulator to move along a specified trajectory. They can also be classified as computed torque methods. Another approach was given by Golla et al. [92], using a variation of the computed torque method to obtain the centralized and decentralized state feedback control which involves placement of closed-loop poles. Jamshidi et al. [11] used an inverse dynamics model in the feedforward path with a PID feedback controller. These methods are computationally expensive and the convergence of the control depends on high sampling frequency. None of the above approaches show stability of the closed-loop system or rejection of disturbances.

Some manipulator control schemes use optimal control theory [12,13,16,18,93]. For example, Kahn and Roth [12], Luh and Walker [13], Snyder and Gruver [93], Saridis and Lee [18] use optimal control theory based on linearized continuous time models. For most cases results are given, for simulation studies, for a particular system only. A method in this category [16] uses a table look-up for optimal control inputs. This requires a very large memory, particularly when the trajectory or the object size changes. Another class of control schemes, such as that in [15], solves eigenvalue assignment problems for linearized models. This scheme is not based on the minimization of a performance index.

Another approach to robot control is based on sliding-mode or variable structure theory [14,19,31]. Young [14] was the first to propose a variable structure controller to control a manipulator. In this method the exact determination of the switching instances for the control input is difficult to achieve. In the approach given by Balestrino et al. [31], a linear compensator is used to produce a switching signal that ensures that all the trajectories reach the switching surface and achieve good tracking. The adaptation scheme used to determine the feedback gains is based on solving two independent inequalities. However, the high-frequency control action may excite resonances and the derivative of errors goes to zero only in the mean. These schemes do not provide any identification of parameters. The output response may be highly oscillatory.

The method of linearizing and decoupling of the nonlinear terms in the manipulator dynamics is proposed by Craig [5]. This is called model-based control which is accomplished by introducing a nonlinear controller in the feedback loop so as to cancel the nonlinear terms in the dynamics. A servo control is then constructed for the linear model. Another approach, given by Singh and Schy [7] is also based on the cancellation of nonlinear terms in the manipulator dynamics. This uses two controllers, one for the hand and one for the joint angles. It uses an inverse model based on determination of the inverse of the Jacobian. These methods assume exact cancellation of the nonlinear terms of the manipulator's inverse dynamics, which requires that the exact nonlinear terms be known a priori and should not change. However this is not possible in practice.

Another class of advanced robot control systems covers methods based on adaptive control theory. One category of such schemes uses Model Reference Adaptive Control (MRAC) techniques [17,19,20-24,31,40,67]. The resulting controller designs are generally stable, and the closed-loop state follows the desired trajectory within a small range of error. When a robot manipulator operates at high speed, however, the effects of nonlinearities, time-varying coefficients, and additional uncertainties, such as backlash and friction, may cause the system errors to become large. Dubowsky and Desforges [20]

applied MRAC for motion control of a manipulator using a second order model as a reference model for each joint. The gains are adjusted using the method of steepest descent to minimize the error between the model and the actual manipulator outputs. This model is very approximate, and does not take into account the coupling between the joints, nor any changes in the external load or the effects of disturbances due to friction. Takegaki and Arimoto [21] used a decoupled linearized perturbation model for the manipulator in Cartesian space. The model neglects coupling between the joints. The control law is designed to cancel the acceleration term in the manipulator's inverse dynamics to stabilize the system and reject disturbances. A feedforward nonlinear controller is determined based on the dynamics of the manipulator which are assumed to be known. The results given in [21] show that the control law does not compensate against the effects of changes in the inertia of the end-effector or the payload. Singh [19] proposed an adaptive model following control scheme for uncertain systems. His work is restricted to manipulators with revolute joints and only a mathematical study is presented. A dynamic compensator is used to generate the feedback elements, this requires that the compensator be stable. The approach also assumes that the manipulator's dynamical terms are bounded. On the other hand Horowitz and Tomizuka [55-57] have proposed a scheme that takes some of the manipulator dynamics into account They write the dynamics with portions that depend only on the manipulator position which is described by a set of unknown parameters that are identified on-line. Accordingly, in order for the theory to be valid, the rate at which the manipulator changes configuration must be slow compared to the adaptation time constant

Seraji [22,38,43,68] proposed an incremental MRAC scheme for robotic manipulators. The control law depends on a linearized decoupled model Its parameters are updated according to a scheme based on Lyapunov stability theory An auxiliary input is used to represent the effect of high order nonlinearities A trial and error approach is used to select the weighting matrices and scalar factors to ensure stability and obtain

good tracking performance. Such selection becomes difficult as the number of degrees of freedom increases. In the adaptive control scheme proposed by Oh et al. [17] a nonzero steady state error is obtained unless certain conditions are satisfied. Lim and Eslami [23] proposed an algorithm based on model reference and feedback controllers. The manipulator model includes the effect of friction and other disturbances. They use Lyapunov stability and solve a Riccati equation to get the updating control laws, which minimize the error between the desired output and the actual output. They make an approximation that a certain time-varying matrix is constant. As seen from their results, the error is high initially and vanishes with time along the trajectory. In the adaptive control approach of Yuan and Book [40], the reference input is directly derived from the reference model, which is not necessarily equivalent to the actual manipulator inverse dynamics. The state feedback controllers consist of constant gain matrices. The stability of the system depends on a certain matrix function whose magnitude is required to be less than 1 at all times.

Another class of adaptive control techniques uses self-tuning control [25,26,59,60,61]. Koivo et al. [26,58], proposed an adaptive self-tuning controller using an autoregressive model to fit the input-output data for the manipulator. This algorithm assumes that the interaction forces and other disturbances are negligible. The algorithm uses a linearized discrete-time invariant decoupled model for parameter identification, and the optimal input is computed by minimizing a performance criterion whose weighting matrices are selected by the designer. There is no rigorous proof of stability. Leininger [59,60] and Backes et al. [61] assume linear decoupled models for each joint and then proceed to apply self-tuning-regulator theory based on least-squares identification and pole placement control. There is no proof for closed-loop stability of these schemes. Kubc and Ohmae [62] implement an adaptive controller in Cartesian space, but make gross approximations by neglecting terms in the dynamics.

Another category of adaptive control techniques uses least-squares identification of

perturbation models [25,27,38,48,51,63]. Lee et al. [27,63] give an adaptive controller based on linearized perturbation models in Cartesian space in the vicinity of a nominal trajectory. A recursive least-squares identification technique is used to identify the parameters of the model, and an optimal adaptive self-tuning controller is designed to minimize the error. The applied joint torques consist of the nominal torques computed from the manipulator's inverse dynamics in the forward path and a correction term from the adaptive controller. The use of inverse dynamics in the feedforward path is very expensive computationally. The trajectory conversion scheme used is very complicated, and the study represents an ideal system because it neglects all high order nonlinearities. The method proposed by Lobbezoo et al. [53] is based on the estimation of the inverse Jacobian in the feedforward path. This method is computationally expensive and the estimation of the inverse Jacobian is not possible near singular points.

Another approach in adaptive control uses nonlinear model-based controllers [6,19,32,50,52] in the feedback and feedforward paths and assumes exact cancellation of some of the nonlinear terms in the manipulator dynamics. Such cancellation is not possible in practice. Craig et al. [32] have used a nonlinear model based controller. Two nonlinear terms in the feedforward and feedback paths are estimated and their values are updated according to Lyapunov theory. The method assumes a fixed value for the state feedback gains. Slotine et al. [50,66] proposed an adaptive controller based on estimation of the unknown parameters of the manipulator. A non-recursive fast algorithm is used for estimation of these parameters. The adaptation scheme involves the solution of a set of differential equations and uses a set of integrators. The technique is sensitive to measurement noise from the encoders and tachometers. They propose the use of a filter with dead zone to suppress the high frequency components. Also, solving the differential equations needs a priori knowledge of the initial values of the parameters. The response shown is highly oscillatory and converges with small steady state errors.

Recently research in adaptive control has focussed on the problem of robustness. It

was shown by Peterson et al. [46] that if poor knowledge of the system parameters exists, the control scheme may not be decoupled and linearized, but may cause the system to be unstable. The robust controller given by Singh and Schy [7] is valid only for small deviations in the joint angles about the nominal values. The approach given by Gilbert and Ha [45] used decoupled PID controllers. The method of robustness given by Spong et al. [47] assumed that the velocity terms quadratic in $\dot{\theta}$ could be bounded with linear bound. This assumption requires that the linear bound should hold over all range of $\dot{\theta}$. The method of using servocompensators [48] in the control system, requires that the compensator should remain stable along the trajectory. Craig [33] proposed a robust model based controller by specifying an upper bound for the uncertainty vector and assuming that the upper maximum bounds for all parameters are known. The approach does not ensure that the errors due to the unmodeled dynamics will converge to zero.

1.3 Motivation and Scope of the Thesis

A major requirement for robust and accurate manipulator control is that accurate values of manipulator parameters and payloads be available in implementing the controllers. The reference models used in adaptive control in general do not often represent the actual dynamics of the manipulator. Such approaches are therefore not robust to variations in the system parameters. Furthermore, the nonlinear controller often has the same order of complexity as the robot dynamics, making these techniques computationally expensive.

The need to overcome the difficulties mentioned in the preceding section has provided the motivation for the research described in this thesis. A new algorithm that uses an adaptive control technique is proposed in this thesis, which is based on explicit least-squares recursive identification to identify a piecewise linear time-varying dynamical model on-line. Based on this model, feedforward and feedback controllers are designed. The implementation of these controllers depends only on the identified model and the controller parameters are updated on-line in terms of the identified model

parameters. The algorithm takes into account the effects of coupling and other unmodeled dynamics and compensates against these effects by incorporating a robust control scheme.

A new approach is given in this thesis to ensure the required degree of robustness for the adaptive controller. This enables the controller to maintain the asymptotic stability of the closed-loop system and to achieve trajectory tracking in the presence of certain unmodeled dynamics, parameter and payload variations. The adaptive control algorithm provides on-line selection of various gain matrices and other controller parameters without requiring a trial and error approach. The implementation of the adaptive controller is based on the identified model and does not require any knowledge of the physical parameters of the manipulator or the use of any nonlinear terms such as the inverse dynamics in the feedforward or feedback paths. The adaptive controller has a relatively simple structure and an operations count shows that it is computationally fast and therefore feasible for real-time applications. The adaptive controller is implemented as decentralized control in joint space or Cartesian space. With some modifications, it is also applicable to flexible joint manipulators

The main contents of each chapter are as follows:

Chapter 2: *Manipulator Dynamical Structure*

This chapter introduces the basic dynamical equations for a rigid-link, open-chain manipulator. The nonlinear inertia, centrifugal and Coriolis, gravitational and frictional terms of the manipulator are specified. The Newton-Euler formulation of manipulator dynamics is given in both joint space and Cartesian space. The simulation methodology for the manipulator dynamics, used in this thesis, is also discussed.

Chapter 3: *Review of Adaptive Control Schemes*

This chapter introduces the basic concepts of some of the adaptive control schemes which are widely used in manipulator control and which are related to the work

presented in this thesis. In particular, model reference adaptive control and recursive least-square identification are reviewed since the proposed algorithm makes extensive use of these techniques.

Chapter 4: *Adaptive Control of Robot Manipulators: Joint-Based Control*

In this chapter the problem of robot manipulator control is presented. A new approach using adaptive control is described which determines the torque required to track desired trajectories in joint space. The controller is designed to compensate against the effects of unmodeled dynamics. A new approach is also presented to study the stability of the resulting closed-loop system. The computational complexity of the technique is discussed and an example is considered to illustrate the applicability of the proposed algorithm.

Chapter 5: *A Bounded Perturbation Approach for Robustness of the Adaptive Controller*

This chapter is concerned with the development of a new robustness method for the adaptive control scheme proposed in Chapter 4. The method is shown to be robust against the effects of unmodeled dynamics and model uncertainties. The approach is more accurate and efficient than the existing ones. An example is described to illustrate the capability of the proposed algorithm.

Chapter 6: *Cartesian Control of Robot Manipulators*

In this chapter the problem of motion control of the end-effector of a manipulator in Cartesian space is described. The algorithms developed in Chapters 4 and 5 are extended here to control the motion of the end-effector in Cartesian space, and an example is given to illustrate the main results of this chapter.

Chapter 7: *Adaptive Control of Flexible Joint Manipulators*

In this chapter the effects of flexibility at the joints of a manipulator are discussed. First the formulation of the dynamics of a flexible-joint manipulator is presented. The dynamics are then reformulated in terms of a singularly perturbated system with slow and fast modes. The adaptive control strategy described in the preceding chapters is then modified to control this type of systems. A stability analysis for the resulting controller is given and the main results are illustrated by an example of a single link flexible-joint manipulator.

Chapter 8: *Conclusions and Future Work*

General conclusions concerning the algorithms, analysis and the results of the thesis are given in this chapter. Possible extensions of the results to more specific problems or new problems in manipulator control are discussed.

CHAPTER TWO

MANIPULATOR DYNAMICAL STRUCTURE

In this chapter the general structure of the dynamic equations for a rigid-link, open-chain robotic manipulator is discussed. All the important basic definitions that will be used in the rest of this thesis are given. The focus here is only on the rigid body dynamics. The case of joint flexibility will be discuss in details in Chapter 7.

2.1 INTRODUCTION

A manipulator generally consists of an arm (to which an end-effector, or gripper is affixed), a power source and a control unit. The arm consists of rigid links connected through joints, either revolute for rotary motion or prismatic for sliding motion. Each joint is provided with position and velocity sensors to measure the joint angle and its velocity, and an actuator to apply a torque on the neighboring link. The degrees of freedom that a manipulator possesses is the number of independent position variables, usually equal to the number of joints. The kinematics of a manipulator is the study of its motion without regard to the forces which cause the motion. The equations of motion describing the kinematics, relate the joint positions, velocities and accelerations to the displacement of the end-effector in Cartesian space. This formulation is important for Cartesian control of the end-effector.

Manipulator dynamics is the study of the forces and torques required to cause motion of the manipulator links. In order to accelerate a manipulator from rest, a complex set of torque functions must be applied by the joint actuators. The exact form of the required functions of the actuator torques depends on the spatial and temporal characteristics of the path to be taken by the end-effector as well as the mass properties of the links and payload, friction of the joints, etc. One method of controlling a manipulator to follow a desired path involves calculating these actuator torque functions

using the dynamic equations of the motion of the manipulator.

2.2 Joint Space Equations for Manipulator Dynamics

Several methods are available for formulating the dynamic behavior of a rigid link manipulator [1,3,5]. The recursive Newton-Euler formulation [6] is computationally one of the more efficient methods and involves the successive transformation of velocities and accelerations from the base of the manipulator out to the end-effector, link by link, using the relationships between moving coordinate systems. Forces are then transformed back from the end-effector to the base to obtain the joint torques. The complete derivation of the Newton-Euler formulation can be found in [1,3,5]. We can write the dynamic equations of the manipulator in closed form as [6]

$$\tau = M(\theta)\,\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + J^T(\theta)\,K_x + \eta(t) \qquad (2.2.1)$$

where, K_x is the $N \times 1$ vector of external forces and moments acting on the end-effector. The i^{th} equation of (2.2.1) can be written in the sum of products form as [40]

$$\tau_i = \sum_{j=1}^{n} m_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} v_{ijk}\,\dot{\theta}_j\,\dot{\theta}_k + g_i(\theta) + \sum_{j=1}^{N} J_{ji}^T K_{x_j} + \eta_i(t) \qquad i = 1,\ldots,n \qquad (2.2.2)$$

where

m_{ij} is the $(i,j)^{th}$ element of the inertia matrix

$v_{ijk}\,\dot{\theta}_j\,\dot{\theta}_k$ is the centrifugal and Coriolis term

g_i is the gravity element

J_{ji} is the $(j,i)^{th}$ element of the Jacobian matrix $J(\theta)$

K_{x_j} is the j^{th} element of K_x

$\eta_i(t)$ is the i^{th} element of uncertainties

It is sometimes useful to write $V(\theta,\dot{\theta})$ in configuration dependent form for ease of implementation. In such a form, the centrifugal terms are separated from the Coriolis terms. The dynamic equation (2.2.1) can then be written (assuming free space motion and no uncertainties) as

$$\tau = M(\theta)\,\ddot{\theta} + B_1(\theta)\left[\dot{\theta}\dot{\theta}\right] + B_2(\theta)\left[\dot{\theta}^2\right] + G(\theta) \qquad (2.2.3)$$

where $B_1(\theta)$ is an $n \times \frac{1}{2} n(n-1)$ matrix of Coriolis terms, $\left[\dot{\theta}\dot{\theta}\right]$ is an $\frac{n(n-1)}{2} \times 1$ vector of joint velocity products given by

$$\left[\dot{\theta}\dot{\theta}\right] = \left[\dot{\theta}_1\dot{\theta}_2 \quad \dot{\theta}_1\dot{\theta}_3 \ldots . \dot{\theta}_{n-1}\dot{\theta}_n\right]^T ,$$

$B_2(\theta)$ is an $n \times n$ matrix of centrifugal terms and $\left[\dot{\theta}^2\right]$ is an $n \times 1$ vector given by

$$\left[\dot{\theta}^2\right] = \left[\dot{\theta}_1^2 \quad \dot{\theta}_2^2 \ldots . \dot{\theta}_n^2\right]^T$$

2.2.1 The Manipulator Inertia Matrix, $M(\theta)$

The manipulator is modeled as a mechanism of a set of n rigid links connected in series through joints, with one end fixed to the ground and the other end free to which is attached a gripper or end-effector. Let $\theta = \left(\theta_1 , \theta_2 , \ldots , \theta_n\right)^T$ be an $n \times 1$ vector of generalized coordinates describing the joint positions, $M(\theta)$ an $n \times n$ matrix describing the mass distribution of the manipulator, and τ an $n \times 1$ vector of joint torques. Then the kinetic energy of the manipulator can be written [3,33] as

$$K.E. = \frac{1}{2}\dot{\theta}^T M(\theta)\,\dot{\theta} \qquad (2.2.4)$$

Each element of $M(\theta)$ have units of inertia $(kg\text{-}m^2)$. Clearly, $M(\theta)$ must be positive-definite. The quantity on the right-hand side of (2.2.4) is therefore always positive and represents energy.

The potential energy of the manipulator can be described by a scalar function of positions only, say $P_r(\theta)$. Therefore, we can write the Lagrangian of the system [33] as

$$L = \frac{1}{2}\,\dot{\theta}^T M(\theta)\,\dot{\theta} - P_r(\theta) \qquad (2.2.5)$$

Using the Lagrangian method, the dynamic equation of the manipulator can be derived using the relation [33]

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}}\right] - \frac{\partial L}{\partial \theta} = \tau \tag{2.2.6}$$

Therefore,

$$\frac{d}{dt}\left[M(\theta)\,\dot{\theta}\right] - \frac{\partial}{\partial \theta}\left[\frac{1}{2}\dot{\theta}^T M(\theta)\,\dot{\theta} - P_r(\theta)\right] = \tau \tag{2.2.7}$$

or

$$\dot{M}(\theta)\,\dot{\theta} + M(\theta)\,\ddot{\theta} - \frac{1}{2}\dot{\theta}^T\left[\frac{\partial}{\partial \theta}M(\theta)\right]\dot{\theta} + \frac{\partial}{\partial \theta}P_r(\theta) = \tau \tag{2.2.8}$$

From (2.2.8) we find that the coefficient of $\ddot{\theta}$ is $M(\theta)$, that is, the manipulator inertia matrix is the kinetic energy matrix [33].

One important property of the manipulator inertia matrix is that each of its elements is a complex function depending on θ, the positions of the joints of the manipulator. Also this dependence involves sine and cosine functions of θ [3,33]. Since sine and cosine are bounded for any value of their arguments, and they appear only in the numerators of the elements of $M(\theta)$, then $M(\theta)$ is bounded for all θ. We can state several properties of $M(\theta)$:

1- It is symmetric.

2- It is positive-definite and bounded from above and below

3- Its inverse exists and is positive-definite and bounded

4- It is the kinetic energy matrix of the manipulator

2.2.2 The Centrifugal and Coriolis Terms, $V(\theta, \dot{\theta})$

The vector of centrifugal and Coriolis terms is a complex nonlinear function of θ and $\dot{\theta}$. It can be written [23,31,33] as

$$V(\theta, \dot{\theta}) = \begin{bmatrix} \dot{\theta}^T V_1(\theta)\, \dot{\theta} \\ \dot{\theta}^T V_2(\theta)\, \dot{\theta} \\ \vdots \\ \dot{\theta}^T V_n(\theta)\, \dot{\theta} \end{bmatrix} \tag{2.2.9}$$

where $V_i(\theta)$ are $n \times n$ symmetric matrices. Clearly it can be written as the product

$$V(\theta, \dot{\theta}) = B(\theta, \dot{\theta})\, \dot{\theta} \tag{2.2.10}$$

$B(\theta,\dot{\theta})$ can also be written as

$$B(\theta,\dot{\theta}) = V_p(\theta)\ V_v(\dot{\theta}) \tag{2.2.11}$$

where $V_p(\theta)$ is an $n \times \frac{1}{2} n(n+1)$ matrix whose structure depends on the kinematics of the manipulator, and

$$V_v(\dot{\theta}) = \begin{bmatrix} \dot{\theta}_1\, I_n & & \\ O_{n-1,1} & \dot{\theta}_2\, I_{n-1} & \\ O_{n-2,2} & \dot{\theta}_3\, I_{n-2} & \\ \vdots & & \vdots \\ O_{1,n-1} & & \dot{\theta}_n \end{bmatrix} \tag{2.2.12}$$

where

$O_{i,j}$ *is an* $i \times j$ null matrix

I_k *is a* $k \times k$ identity matrix

It can be shown [3,33] that the dependence on θ appears only in terms of sine and cosine functions. Therefore $V(\theta,\dot{\theta})$ has a bound that is independent of θ, but increases quadratically with $\dot{\theta}$.

From the Lagrange formulation (2.2.8), it can be shown that

$$V(\theta,\dot{\theta}) = \dot{M}(\theta)\ \dot{\theta} - \frac{1}{2}\dot{\theta}^T \left[\frac{\partial}{\partial\theta} M(\theta)\right]\dot{\theta} \tag{2.2.13}$$

from where it follows that [40]

$$\dot{M}(\theta) = 2B(\theta,\dot{\theta}) - K \tag{2.2.14}$$

where K is some skew symmetric matrix. This can be useful in that it implies the following relationship between quadratic forms

$$\frac{1}{2}\theta^T \dot{M}(\theta)\,\theta = \theta^T B(\theta,\dot{\theta})\,\theta \tag{2.2.15}$$

Now, we can state the general properties of $V(\theta,\dot{\theta})$:

1- It is quadratic in $\dot{\theta}$.

2- It can be factorized in different forms, e.g. (2.2.10-2.2.12).

3- It is related to the time derivative of the inertia matrix by (2.2.15).

2.2.3 The Friction Term, $F_r(\dot{\theta})$

Frictional forces will be quite significant for manipulators that have gears at the joints. Friction terms are complex and nonlinear and may be represented approximately by deterministic models. A simple model for friction is that for viscous friction, in which the torque due to frictional forces is proportional to the velocity of the joint motion, i.e.

$$T_{friction} = c_1 \dot{\theta} \tag{2.2.16}$$

where c_1 is the coefficient of viscous friction. Another possible model for friction is that for Coulomb friction, which is represented as

$$T_{friction} = c_2\ sgn(\dot{\theta}) \tag{2.2.17}$$

where c_2 is the Coulomb friction constant and sgn is the signum function. The value of c_2 is taken to be the static coefficient, when $\dot{\theta}=0$, and a lower value, the dynamic coefficient, when $\dot{\theta}\neq 0$. In many manipulator joints, friction also depends on the joint position because of the effect of gears which are not perfectly round. In general we can model friction as a sum of all these effects.

2.2.4 The Gravity Term, $G(\theta)$

The effect of gravity loading on the links can be included quite simply by setting the initial translational acceleration to be equal to g vertically upwards, where g is the gravity constant. This is equivalent to saying that the base of the robot is accelerating upward with 1 g acceleration. This will cause the same effect on the links as gravity would. The gravity term is a function of the joint angles θ and can be expressed [23,31,33] as

$$G(\theta) = C(\theta)\,\theta \tag{2.2.18}$$

where $C(\theta)$ is an $n \times n$ matrix containing the gravity constant g.

2.3 The Manipulator Jacobian

The Jacobian matrix is used to map velocities in joint space to the velocities in Cartesian space. The nature of this mapping depends on the joint angles and link parameters. The Jacobian in general, is a multidimensional form of the first derivative of a vector function with respect to a vector variable. Suppose we have six functions, each of which is a function of six independent variables, i.e.

$$x_1 = f_1\left[y_1, y_2, y_3, y_4, y_5, y_6\right]$$

$$x_2 = f_2\left[y_1, y_2, y_3, y_4, y_5, y_6\right]$$

$$\vdots$$

$$x_6 = f_6\left[y_1, y_2, y_3, y_4, y_5, y_6\right]$$

which can be written in vector form as

$$X = F(Y) \tag{2.3.1}$$

using the chain rule we can find the differential of x_i as a function of the differential of y_i as

$$\delta x_1 = \frac{\partial f_1}{\partial y_1}\delta y_1 + \frac{\partial f_1}{\partial y_2}\delta y_2 + \ldots + \frac{\partial f_1}{\partial y_6}\delta y_6$$

$$\delta x_2 = \frac{\partial f_2}{\partial y_1}\delta y_1 + \frac{\partial f_2}{\partial y_2}\delta y_2 + \ldots + \frac{\partial f_2}{\partial y_6}\delta y_6$$

$$\vdots$$

$$\delta x_6 = \frac{\partial f_6}{\partial y_1}\delta y_1 + \frac{\partial f_6}{\partial y_2}\delta y_2 + \ldots + \frac{\partial f_6}{\partial y_6}\delta y_6$$

which can be written in vector form as

$$\delta X = \frac{\partial F}{\partial Y}\,\delta Y \tag{2.3.2}$$

The 6×6 matrix of partial derivatives in (2.3.2) is the Jacobian J of F with respect to Y. If the functions $f_1(Y)$ through $f_6(Y)$ are nonlinear, then the partial derivatives are functions of y_i. Then we can write (2.3.2) as

$$\delta X = J(Y)\,\delta Y \tag{2.3.3}$$

By dividing both sides by the differential time element, we get the Jacobian which maps velocities in Y to those in X:

$$\dot{X} = J(Y)\,\dot{Y} \tag{2.3.4}$$

At any particular instant, Y has a certain value, and $J(Y)$ is a linear transformation Thus when Y changes with time, then so does $J(Y)$ The Jacobian is therefore a time varying linear transformation in general In robotics, Jacobians relate joint velocities to Cartesian velocities of the end-effector i e

$$\mathbf{v}_x = J(\theta)\,\dot{\theta} \tag{2.3.5}$$

where θ is the vector of joint angles of the manipulator, and $\mathbf{v}_x$ is a vector of Cartesian velocities Note that in (2.3.5), J and $\mathbf{v}_x$ written in the same coordinate frame For a six jointed manipulator with 6 degrees of freedom for the end-effector, the Jacobian is a

6×6 matrix, $\dot{\theta}$ and $\mathbf{v}_x$ are 6×1 vectors. The 6×1 Cartesian velocity vector contains a 3×1 linear velocity vector and a 3×1 rotational velocity vector stacked together.

Jacobians of any dimension (including nonsquare) may be defined. The number of rows equals the number of degrees of freedom in Cartesian space. The number of columns equals the number of joints of the manipulator (degrees of freedom in joint space). For a planar arm, there is no reason for the Jacobian to have more than three rows, although for redundant planar manipulators , there could be arbitrarily many columns (one for each joint). The Jacobian may be found [5] by directly differentiating the kinematic equations of the mechanism. This is straightforward for linear velocity. To include the orientation vector, one may use the method of velocity propagation from link to link to find the Jacobian (for details see [5]).

In the force domain the Jacobian relates joint torques to forces acting on the end-effector. This relation can be defined by equating the work done in Cartesian terms with the work done in joint terms. In a multi-dimensional case, work is the dot product of the vector of forces or torques and a vector of displacements. i.e.

$$F \cdot \delta X = \tau \cdot \delta\theta \tag{2.3.6}$$

where

F is a 6×1 Cartesian force-moment vector acting on the end-effector,

δX is a 6×1 vector of infinitesimal Cartesian displacement of the end-effector,

τ is a 6×1 vector of torques at the joints and

$\delta\theta$ is a 6×1 vector of infinitesimal joint angle displacements.

Expression (2.3.6) can be written as

$$F^T \delta X = \tau^T \delta\theta \tag{2.3.7}$$

Using (2.3.3), with θ in place of Y, in (2.3.7) yields

$$F^T J \delta\theta = \tau^T \delta\theta \tag{2.3.8}$$

which must hold for all $\delta\theta$, and so we have

$$F^T J = \tau^T$$

By transposing both side, we get

$$\tau = J^T(\theta)\, F \tag{2.3.9}$$

Thus the transpose of the Jacobian maps Cartesian forces acting at the hand into equivalent joint torques. This is useful in implementation of Cartesian control of the end-effector. Note that the Jacobian and the force are written in the same coordinate frame. When the Jacobian loses rank, there are certain directions in which the end-effector cannot exert static forces.

2.4 Cartesian Space Formulation of Manipulator Dynamic Equations

Cartesian space dynamic equations which relate the acceleration of the end-effector expressed in Cartesian space to the Cartesian forces and moments acting at the end-effector can be obtained using the manipulator Jacobian [5,34]. From (2.3.5) we have

$$\dot{X} = J(\theta)\,\dot{\theta} \qquad \text{and} \qquad \ddot{X} = J(\theta)\,\ddot{\theta} + \dot{J}(\theta)\,\dot{\theta}$$

Using these relationships, one can write the Cartesian space dynamic equations as

$$F = M_x(\theta)\,\ddot{X} + V_x(\theta, \dot{\theta}) + G_x(\theta) \tag{2.4.1}$$

where

F is the $N \times 1$ force-torque vector acting on the end-effector

X is the $N \times 1$ position and orientation vector of the end-effector

$M_x(\theta)$ is the $N \times N$ Cartesian inertia matrix

$V_x(\theta,\dot{\theta})$ is the $N \times 1$ vector of velocity terms in Cartesian space

$G_x(\theta)$ is the $N \times 1$ vector of gravity terms in Cartesian space

Note that the forces acting on the end-effector, F, could in fact be applied by the actuators at the joints using the relationship

$$\tau = J^T(\theta)\, F$$

where $J(\theta)$ is written in the same coordinate frame as F and X

In Cartesian space the kinetic energy is written as

$$\frac{1}{2}\dot{X}^T M_x(\theta)\,\dot{X} \tag{2.4.2}$$

By equating (2.2.4) and (2.4.2), and using (2.3.5), we can write the Cartesian inertia matrix as

$$M_x(\theta) = J^{-T}(\theta)\,M(\theta)\,J^{-1}(\theta) \tag{2.4.3}$$

where $J^{-T}(\theta)=[J^T(\theta)]^{-1}$. The other Cartesian space quantities can be expressed similarly in terms of joint space quantities [36] as

$$V_x(\theta\ ,\ \dot{\theta}) = J^{-T}(\theta)\left[\ V(\theta,\dot{\theta}) - M(\theta)\,J^{-1}(\theta)\,\dot{J}(\theta)\,\dot{\theta}\right] \tag{2.4.4}$$

$$G_x(\theta) = J^{-T}(\theta)\,G(\theta) \tag{2.4.5}$$

Note that when a manipulator approaches a singularity, certain quantities in Cartesian dynamics become infinite.

2.5 Manipulator Simulation Methodology

The computer simulation of the dynamics of a rigid-body can be done by many methods [1,6]. Because of the complexity of manipulator dynamics, most researchers apply simplifying assumptions to the model, such as ignoring Coriolis and centrifugal forces, or neglecting joint offsets, etc. In such cases, the results obtained are accurate only for a limited range of operations. A more general solution, can be obtained for manipulator simulation using the Newton-Euler equations, in which most relevant effects are considered.

Following the method given in [6], the computer simulation of the dynamic behavior of a manipulator can be summarized as follows. A computer subroutine SUB1 is written to compute the joint torques for a given trajectory and external forces acting on the n^{th} link of the manipulator. Based on the closed-form dynamic equation (2 2.1), the dynamics for a given set of torques can be computed. First we initialize the

procedure by setting the angular acceleration and velocity equal to zero, and the linear acceleration equal to the gravity constant. Next we let B_r to be a "bias vector" equal to the torques due to gravity, centrifugal and Coriolis accelerations, and external forces and moments acting on the n^{th} link, i.e.

$$B_r = V(\theta,\dot{\theta}) + G(\theta) + J^T(\theta)\, K_x$$

This bias vector can be easily computed by setting θ, $\dot{\theta}$ and K_x to their current state, letting $\ddot{\theta} = 0$, and calling $SUB\,1(\theta,\dot{\theta},\ddot{\theta},K_x,B_r)$. The computational complexity of this step, measured in terms of the number of multiplications and additions, is found to be

$$159\,n \quad \textit{multiplications} \quad \text{and} \quad 126\,n \quad \textit{additions}$$

where n is the number of joints. Having computed the bias vector, we can find the accelerations of the joints by solving the linear equation

$$M(\theta)\,\ddot{\theta} = \tau - B_r \qquad (2.5.1)$$

The difficult part in solving (2.5.1) is in evaluating the elements of the inertia matrix, $M(\theta)$. This is accomplished by setting θ to its current state, by letting $\theta = 0$, $K_x = 0$, defining $\ddot{\theta} = r_j$, and calling $SUB\,1(\theta,0,r_j,0,M_j)$. Here r_j is an $n \times 1$ vector with the j^{th} element equal to 1 and 0 everywhere else, and M_j is the j^{th} column of M, i.e. M_j is the torque on the joint actuators when the joint velocities are zero, there are no external forces or gravitational effects, and the joint accelerations θ are equal to r_j. The computation of the inertia matrix, one column at a time, requires

$$159\,n^2 \quad \textit{multiplications} \quad \text{and} \quad 126\,n^2 \quad \textit{additions}$$

Once the elements of the inertia matrix are obtained, the joint accelerations are computed by solving (2.5.1) for θ. The velocities and positions are obtained by applying any of the several well-known numerical integration techniques [42] to integrate (2.5.1) forward in time.

CHAPTER THREE

REVIEW OF ADAPTIVE CONTROL SCHEMES

In this chapter, a brief review of the relevant concepts in adaptive control that are related to the work presented in this thesis is given. The model reference adaptive control scheme and the recursive least-squares estimation technique are reviewed.

3.1 INTRODUCTION

When the knowledge about a system is limited, the issues of adaptive and robust control become important. One way to deal with poor knowledge of parameters in a control scheme is through techniques of adaptive control. By adaptation we mean the process of changing the parameters, structure and possibly the control of a system on the basis of information obtained during the sampling period, so as to achieve some desired behavior for a system when the operating conditions are either incompletely defined initially or changed. An adaptive system measures a certain index of performance using the inputs, states and outputs of the adjustable system and compares the measured index of performance with some given index. The adaptation mechanism modifies the parameters of the adjustable system or generates an auxiliary input in order to maintain the performance index values close to the given ones.

In general, an adaptive controller can be viewed as being composed of two parts: an identification portion, which identifies the parameters of the system itself, or parameters that appear in the controller for the system, and a control law part, which implements a control law that is in some way a function of the identified parameters [33]. Adaptive control schemes can in some cases be computationally fast and require little computer memory. Thus they are suitable for implementation in real-time control of robot manipulators. The most widely used adaptive control schemes in robot manipulator control are those based on model reference adaptive control and recursive estimation.

Since these two schemes are related to the work presented in this thesis, we will review them briefly.

3.2 Model Reference Adaptive Control (MRAC)

In model reference adaptive control, the desired closed-loop performance of the system is described by specifying a reference model that exhibits the desired dynamic response. The adaptive controller observes the output errors between the desired and the actual system and, based on these errors, updates a vector of parameters used by the controller in order to reduce the errors. Basically, there are two classes of MRAC techniques: signal synthesis methods and parameter adaptation methods. In the signal synthesis method, an auxiliary signal is synthesized by the adaptation mechanism and is injected into the system to supplement the operation of some nominal constant controllers. In the parameter adaptation method, the gains of the controllers are tuned by the adaptation mechanism and no external signals are introduced in the system. In both methods, however, the objective is to match the response of a given system with that of a predefined model. The basic configuration of a model reference adaptive system is shown in Fig.3.1.

Assume that the adjustable system is described [28] by

$$\dot{x}(t) = A(t)x(t) + B(t)\,u(t) + w(t) \qquad (3.2.1)$$

where

$x \in \mathbb{R}^{n\times 1}$ is the state vector,

$u \in \mathbb{R}^{m\times 1}$ is the input vector,

$w \in \mathbb{R}^{n\times 1}$ is the vector of auxiliary inputs to be synthesized,

$A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are matrices to be adjusted

The desired performance of the system is embodied in the definition of the reference model given as

$$x_m = A_m\, x_m(t) + B_m\, u(t) \qquad (3.2.2)$$

where $x_m \in \mathbb{R}^{n\times 1}$ and A_m and B_m are constant prespecified matrices of appropriate dimensions.

Let $e = x_m - x$ represent the generalized state error. Then the error between (3.2.2) and (3.2.1) satisfies

$$\dot{e}(t) = A_m\ e(t) + \left[A_m - A\right] x(t) + \left[B_m - B\right] u(t) - w(t) \qquad (3.2.3)$$

Now the adaptive control problem can state as follows: Given an unknown initial difference (at $t = t_o$) between the parameters of the adjustable system and those of the reference model i.e. $(A_m - A)$, $(B_m - B)$ and a known initial state error $e(t_o) = x_m(t_o) - x(t_o)$, find an adaptation law which minimize the quadratic criterion

$$P.I = \int_{t_o}^{\infty} \left\{ e^T P\ e + tr\left[\left(A_m - A\right)^T \kappa_a \left(A_m - A\right) \right] + tr\left[\left(B_m - B\right)^T \kappa_b \left(B_m - B\right) \right] \right\} dt$$

where tr denotes the trace operation, κ_a, κ_b and P are positive-definite matrices.

The adaptive control laws can be designed by minimizing the performance index or by using Lyapunov's second method. Here, we will describe briefly the MRAC approach proposed by Seraji [22], which uses Lyapunov stability to derive the adaptation laws. Defining the scalar positive-definite Lyapunov function as

$$V(t) = e^T P\ e + w^T Q_o\, w + tr\left[\left(A_m - A\right)^T Q_1 \left(A_m - A\right) \right]$$

$$+ tr\left[\left(B_m - B\right)^T Q_2 \left(B_m - B\right) \right] \qquad (3.2.4)$$

The matrices Q_o, Q_1 and Q_2 are arbitrary positive-definite constant $n \times n$ matrices and P is the solution of the Lyapunov equation

$$P\ A_m + A_m^T P = -Q \qquad (3.2.5)$$

with Q any positive-definite matrix. In (3.2.4), the terms $e^T Pe$ and $w^T Q_o\, w$ denote the distances of e and w from the origin while $(A_m - A)^T Q_1 (A_m - A)$ and

$(B_m - B)^T Q_2 (B_m - B)$ represent the distances between the adjustable matrices of the system and those of the reference model

Now, by differentiating the Lyapunov function $V(t)$ along the trajectory (3.2.3) we obtain

$$\dot{V}(t) = \dot{e}^T Pe + e^T P\dot{e} + 2w^T Q_0 \dot{w} + 2tr\left[(A_m - A)^T Q_1 (\dot{A}_m - \dot{A})\right] + 2tr\left[(B_m - B)^T Q_2 (\dot{B}_m - \dot{B})\right]$$

or

$$\dot{V}(t) = e^T \left[PA_m + A_m^T P\right] e - 2w^T P\dot{e} + 2x^T \left[\dot{A}_m - \dot{A}\right]^T Pe + 2w^T Q_0 \dot{w} + 2u^T \left[\dot{B}_m - \dot{B}\right]^T Pe + 2tr\left[(A_m - A)^T Q_1 (\dot{A}_m - \dot{A})\right] + tr\left[(B_m - B)^T Q_2 (\dot{B}_m - \dot{B})\right]$$

Using the identities $a^T b = b^T a = tr[ab^T]$ and $tr[ab^T M] = tr[Mab^T]$ for any $n \times 1$ vectors a and b and $n \times n$ matrix M, the above equation can be simplified to

$$\dot{V}(t) = -e^T Q e - 2w^T \left[P e - Q_0 \dot{w}\right] + 2tr\left[(A_m - A)^T (P e x^T - Q_1 \dot{A})\right] + 2tr\left[(B_m - B)^T (P e u^T - Q_2 \dot{B})\right] \quad (3.2.6)$$

Now, for the adaptation error $e(t)$ to vanish asymptotically i.e. for $x(t) \to x_m(t)$ the error equation (3.2.3) must be asymptotically stable. From Lyapunov's stability result this requires the function $V(t)$ to be negative definite. From (3.2.6) since the term $e^T Qe$ is negative-definite it is sufficient to require

$$Pe - Q_0 \dot{w} = 0 \quad (3.2.7a)$$

$$Pe x^T - Q_1 \dot{A} = 0 \quad (3.2.7b)$$

$$Pe u^T - Q_2 \dot{B} = 0 \quad (3.2.7c)$$

Therefore, the adaptation laws for the auxiliary input w and the adjustable gains A and B are found to be

$$\dot{w}(t) = Q_o^{-1} P e$$
$$\dot{A}(t) = Q_1^{-1} P e x^T$$
$$\dot{B}(t) = Q_2^{-1} P e u^T$$

with these adaptation laws, $\dot{V}(t)$ becomes equal to $-e^T Qe$ which is a negative-definite function of e only, implying that $e(t) \rightarrow 0$ as $t \rightarrow \infty$; i.e. $x(t) \rightarrow x_m(t)$. However, this does not imply that $A \rightarrow A_m$, $B \rightarrow B_m$ or $w \rightarrow 0$. Also, the behavior of $w(t)$, $A(t)$ and $B(t)$ in the adjustable system (and therefore of the adaptation scheme) depends on the choice of the matrices Q_o , Q_1 and Q_2. It may be difficult in practice to choose these matrices to achieve suitable closed-loop behavior. Seraji [22] used a trial and error approach to select these matrices. Such an approach is, of course, not suitable for on-line implementation where it may be necessary to change the values of Q_o, Q_1 or Q_2 to ensure satisfactory closed-loop response.

3.3 Recursive Least Square Identification

The basic difficulty in robot controller design is that the dynamics are not sufficiently known and/or that they are time-varying. A recursive on-line identification scheme can be used to identify the parameters of the system. These identified parameters can be used to design a suitable control scheme. One of the popular methods of identification is the recursive least squares identification method (RLSI). The basic configuration of RLSI is shown in Fig.3.2. In this method the error between the actual system and the estimated model is used to update the parameters of the model. We will present here an outline of the RLSI method since we will use it extensively throughout this thesis; (for more details the reader is refered to [29]).

Assume that a stable process is time-invariant and linearizable so that it can be described by a linear difference equation

$$y(k)+a_1 y(k-1)+\cdots+a_m y(k-m)=b_1 u(k-d-1)+\cdots+b_m u(k-d-m+1) \quad (3.3.1)$$

where m is the order of the system and d is the delay. Consider measured signals $y(k)$ and $u(k)$ up to the time instant k and process parameter estimates up to time instant $(k-1)$. From (3.3.1) we obtain the following *error* equation

$$e(k) = y(k)+\hat{a}_1(k-1)y(k-1)+\cdots+\hat{a}_m(k-m)y(k-m) \quad (3.3.2)$$

$$\hat{b}_1(k-1)u(k-d-1)+\cdots+\hat{b}_m(k-d-m+1)$$

where the error $e(k)$ arises from the noise contaminated outputs $y(k)$ and from the erroneous parameter estimates. In this equation the following term can be considered as a one-step ahead prediction $\hat{y}(k|k-1)$ of $y(k)$ at time $(k-1)$

$$\hat{y}(k|k-1) = -\hat{a}_1(k-1)y(k-1)-\cdots-\hat{a}_m(k-1)y(k-m) \quad (3.3.3)$$

$$+\hat{b}_1(k-1)u(k-d-1)+\cdots+\hat{b}_m(k-1)u(k-d-m+1)$$

or

$$\hat{y}(k|k-1) = \psi^T(k)\hat{\theta}(k-1) \quad (3.3.4)$$

with the data vector

$$\psi^T(k) = \left[-y(k-1) \quad \cdots \quad -y(k-m+1) \quad u(k-d-1) \quad \cdots \quad u(k-d-m+1) \right] \quad (3.3.5)$$

and the parameter vector

$$\hat{\theta}^T(k-1) = \left[\hat{a}_1 \quad \cdots \quad \hat{a}_m \quad \hat{b}_1 \quad \cdots \quad \hat{b}_m \right] \quad (3.3.6)$$

For the error this gives

$$e(k) = y(k) - \hat{y}(k|k-1) \quad (3.3.7)$$

where $y(k)$ is the new measurement and $\hat{y}(k|k-1)$ is the one-step ahead prediction from the model.

Now for $k = m+d, m+d+1, \ldots, m+d+n$ the $y(k)$'s are measured. Then $n+1$ equations for the unknown parameters are

$$y(k) = \psi^T(k)\hat{\theta}(k-1) + e(k)$$

which can be put in a vector equation as

$$Y(m+d+nr) = \Psi(m+d+nr)\,\hat{\Theta}(m+d+nr-1) + E(m+d+nr)$$

The algorithm of recursive estimation is used to minimize a loss function

$$\Gamma = \sum_{k=m+d}^{m+d+nr} e^2(k)$$

and therefore

$$\frac{d\Gamma}{d\Theta}\Big|_{\Theta=\hat{\Theta}} = 0.$$

With the assumption $nr \geq 2m$ and the abbreviation

$$P(m+d+nr) = \left[\Psi^T(m+d+nr)\,\Psi(m+d+nr)\right]^{-1}$$

we have the estimate,

$$\hat{\Theta}(m+d+nr-1) = P(m+d+nr)\,\Psi^T(m+d+nr)\,Y(m+d+nr) \quad (3.3.8)$$

Equation (3.3.8) represents nonrecursive parameter estimation as the parameter estimates are obtained only after the measurement and storing of all signal values. Writing the nonrecursive estimation equations for $\hat{\Theta}(k+1)$ and $\hat{\Theta}(k)$ results in the recursive parameter estimation algorithm

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + L(k)\left[y(k) - \Psi^T(k+1)\hat{\Theta}(k)\right] \quad (3.3.9)$$

The above recursive equation indicates that the estimate of the parameter vector $\hat{\Theta}$ at the instant $(k+1)$ equals the previous estimate corrected by the term proportional to $y(k) - \Psi^T(k+1)\hat{\Theta}(k)$. The term $\Psi^T\hat{\Theta}$ is the prediction of $y(k+1)$ based on the estimate of parameters $\hat{\Theta}(k)$ and the measurement vector. The correcting vector $L(k)$ is given by

$$L(k) = \frac{P(k)\Psi(k+1)}{\Psi^T(k+1)P(k)\Psi(k+1)+\nu}$$

and

$$P(k+1) = \frac{1}{\nu}\left[I - L(k)\Psi^T(k+1)\right]P(k)$$

In this identification, the term $[\Psi^T(k+1)P(k)\Psi(k+1)+\nu]$ gives a scalar value where $0 \leq \nu \leq 1$ is the forgetting factor which is a weighting factor commonly used for tracking slowly time-varying parameters by exponentially forgetting the old measurements. If $\nu < 1$, a large weight is placed on the more recent sampled data by rapidly weighting out the previous samples. If $\nu = 1$, accuracy in tracking the time varying parameters will be lost due to the truncation of measured data sequences. We can compromise between fast adaptation capabilities and loss of accuracy in parameter identification by adjusting the weighting factor ν. In most applications for tracking slowly time-varying parameters, ν is usually chosen to be $0.9 < \nu < 1$. The value of ν can be adjusted for each state variable as desired, but this requires excessive computations in computing $P(k+1)$. For real-time robot control, such adjustment is not desirable. The matrix $P(k+1)$ is computed only once at each sampling time using the given weighting factor ν. Moreover, since $P(k)$ is a symmetric positive definite matrix, only the upper triangular part of $P(k)$ needs to be computed.

To start the recursive algorithm, we set

$$\Theta(0) = 0 \quad \text{and} \quad P(0) = \alpha I$$

where α is a large positive number. The expectation of the matrix P is proportional to the covariance matrix of the parameter estimates

$$E\{P(k+1)\} = \frac{1}{\rho^2} cov\{\Delta\Theta(k)\}$$

with $\rho^2 = E\{e^T e\}$ and the parameter error $\Delta\Theta(k) = \Theta(k) - \Theta$. Hence the recursive algorithm produces the variance of the parameter estimates, being the elements of the covariance matrix.

The convergence condition required for the recursive parameter estimation algorithm is that the parameter estimates are unbiased, i.e.

$$E\{\Theta(k)\} = \Theta \qquad \text{as } k \to \infty$$

and consistent in mean square

$$\lim_{nr \to \infty} E\left(\hat{\Theta}(nr)\right) = \Theta_o$$

$$\lim_{nr \to \infty} E\left\{ \left[\hat{\Theta}(nr) - \Theta_o\right] \left[\hat{\Theta}(nr) - \Theta_o\right]^T \right\} = 0$$

Also it requires that the process order m and the dead time d be known, the input $u(k)$ and the output $y(k)$ be exactly measurable and the error $e(k)$ be uncorrelated with the elements of the data vector $\Psi^T(k)$. It is also found [44] that the convergence of the recursive algorithm depends on the choice of the starting values $P(0)$ and $\hat{\Theta}(0)$.

Next, we will show how the estimation algorithm can be programmed in a way that is suitable for adaptive on-line control applications. Let us consider a first order system model

$$y(k) + a_1 y(k-1) = b_1 u(k-1) + \bar{\xi}(k)$$

where $\bar{\xi}$ is the noise in the output. Equivalently we can write

$$y(k) = \Psi^T(k)\hat{\Theta}(k) + e(k)$$

with

$$\Psi^T(k) = \begin{bmatrix} -y(k-1) & u(k-1) \end{bmatrix}$$

$$\hat{\Theta}(k) = \begin{bmatrix} \hat{a}_1(k) & \hat{b}_1(k) \end{bmatrix}^T$$

The estimation algorithm can be described by the following steps:

a) New measurements $y(k)$ and $u(k)$ are taken at time k.

b) Calculation of the error between $y(k)$ and the estimated output which is given as

$$e(k) = y(k) - \begin{bmatrix} -y(k-1) & u(k-1) \end{bmatrix} \begin{bmatrix} \hat{a}_1(k-1) \\ \hat{b}_1(k-1) \end{bmatrix}$$

c) Finding the new parameter estimates from

$$\begin{bmatrix} \hat{a}_1(k) \\ \hat{b}_1(k) \end{bmatrix} = \begin{bmatrix} \hat{a}_1(k-1) \\ \hat{b}_1(k-1) \end{bmatrix} + \begin{bmatrix} L_1(k-1) \\ L_2(k-1) \end{bmatrix} e(k)$$

d) Inserting the measurements $y(k)$ and $u(k)$ in the vector

$$\Psi^T(k+1) = \begin{bmatrix} y(k) & u(k) \end{bmatrix}$$

e) Finding the vector

$$P(k-1)\Psi(k+1) = \begin{bmatrix} P_{11}(k-1) & P_{12}(k-1) \\ P_{21}(k-1) & P_{22}(k-1) \end{bmatrix} \begin{bmatrix} y(k) \\ u(k) \end{bmatrix}$$

$$= \begin{bmatrix} P_{11}(k-1)y(k) + P_{12}(k-1)u(k) \\ P_{21}(k-1)y(k) + P_{22}(k-1)u(k) \end{bmatrix}$$

$$\Delta \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} = \kappa$$

f) Finding

$$\Psi^T(k+1)P(k-1)\Psi(k+1) = \begin{bmatrix} y(k) & u(k) \end{bmatrix} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix}$$

$$= \kappa_1 y(k) + \kappa_2 u(k) = \iota$$

g) Using vectors κ and ι to find the corrective vector $L(k)$ as

$$\begin{bmatrix} L_1(k) \\ L_2(k) \end{bmatrix} = \frac{1}{\iota + \nu} \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix}$$

h) Using the results of steps d to g to find the new adaptive gain $P(k)$ given as

$$P(k) = \frac{1}{\nu}\left[P(k-1) - L(k)\Psi^T(k)P(k-1)\right]$$

$$= \frac{1}{\nu}\left[P(k-1) - L(k)\left[P(k-1)\Psi(k)\right]^T\right]$$

$$= \frac{1}{\nu}\left[P(k-1) - L(k)\kappa^T\right]$$

$$\frac{1}{\nu}\begin{bmatrix} P_{11}(k-1) - L_1(k)\kappa_1 & P_{12}(k-1) - L_1(k)\kappa_2 \\ P_{21}(k-1) - L_2(k)\kappa_1 & P_{22}(k-1) - L_2(k)\kappa_2 \end{bmatrix}$$

i) Replace k by $k+1$ and start again with a

To start the recursive algorithm, we use

$$\hat{\Theta}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ and } P(0) = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}$$

where α is a large positive number.

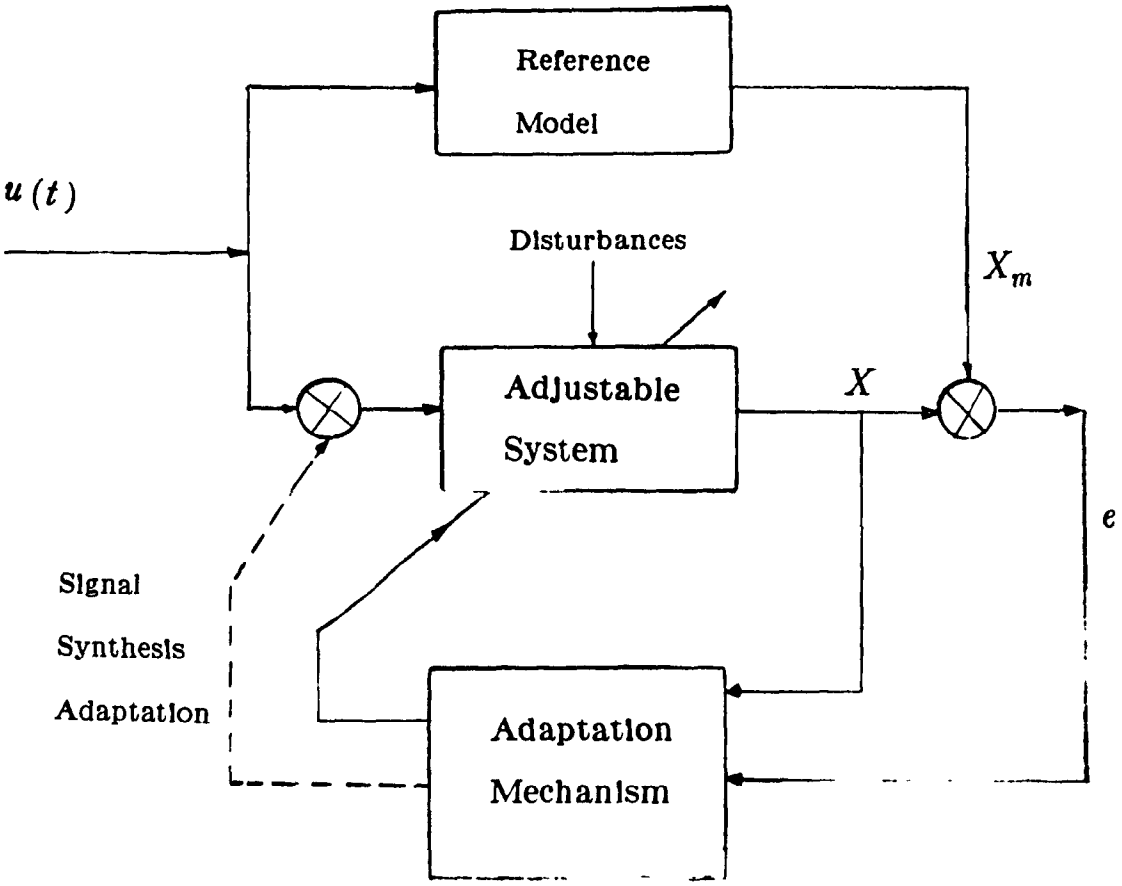

Fig.(3.1) Basic configuration of MRAS

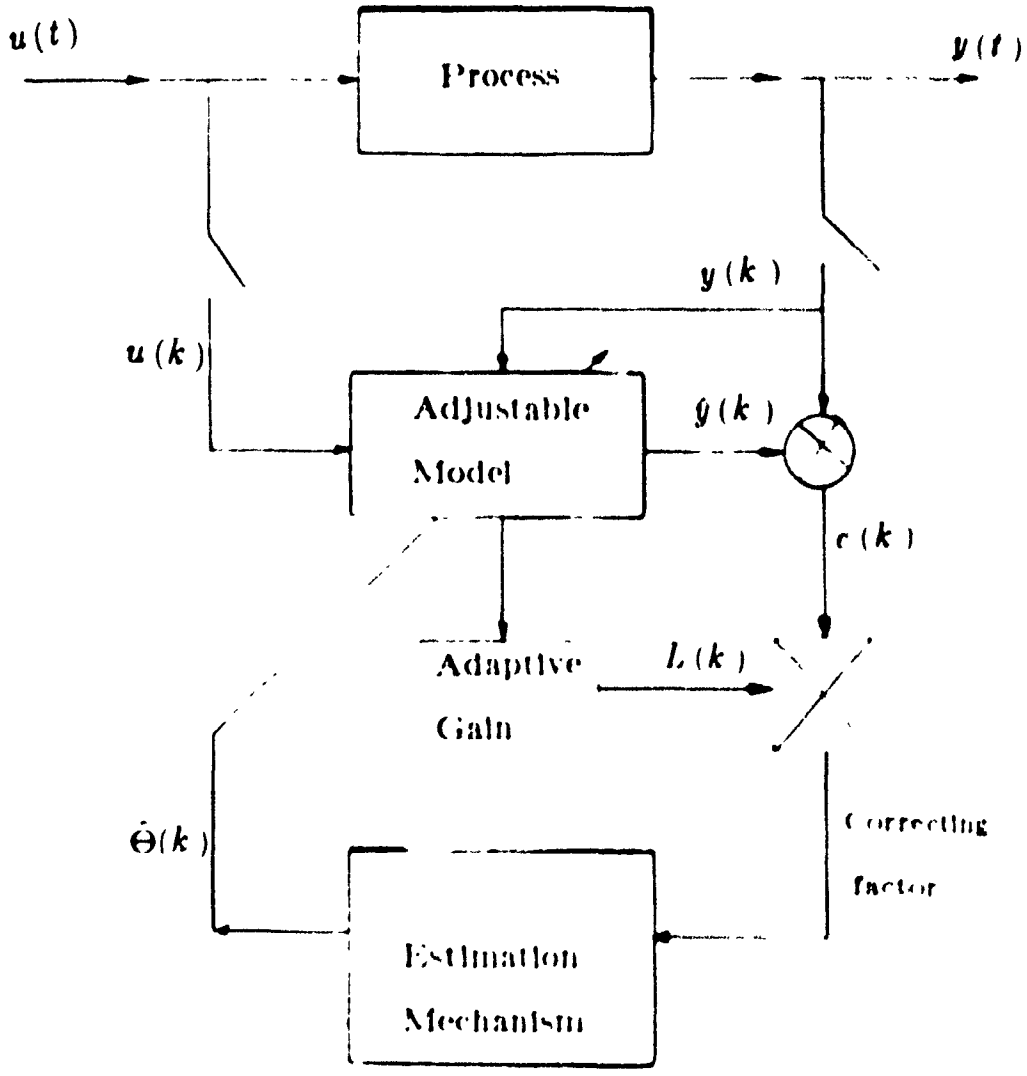

Fig.(3.2) Basic RLSI configuration

CHAPTER FOUR

ADAPTIVE CONTROL OF ROBOT MANIPULATORS: JOINT-BASED CONTROL

4.1 INTRODUCTION

In this chapter a new algorithm is presented for adaptively generating input control commands for robot manipulators to achieve trajectory tracking in joint space while maintaining stability of the resulting closed-loop system. The adaptive control scheme uses feedforward and feedback controllers to generate commands for a robot manipulator to achieve trajectory tracking in joint space. Independent joint control is used in the proposed scheme and the effect of coupling between the joints is accounted for by a special structure of the adaptive controller. The adaptation scheme uses least-squares identification on-line to identify the diagonal terms of the unknown coefficient matrices of a linear time-varying model which has the same inputs and outputs as the manipulator and a similar dynamical structure in differential equation form. Its parameters are used to generate the input control torques in the feedforward path. Proportional-derivative feedback is used to ensure stability and shape the transient behavior. An additional term obtained using Lyapunov theory is incorporated in the adaptive controller to take into account the coupling between the joints and other effects of unmodeled dynamics and parameter variations. The proposed controller, has a simple structure, since the nonlinear complex dynamics of the manipulator are not used in generating the control commands. An operations count for the computation required for the proposed scheme shows that the method is computationally fast and feasible for real-time applications. This chapter is divided into 8 Sections: Section 2 gives the state-space representation of manipulator model which is to be identified. In Section 3 the mathematical derivation of the proposed adaptive control scheme is given. New

approach to ensure stability is given in Sections 4 and 5, using Lyapunov's theorem and a pole assignment technique. In Section 6 the identification method is described. In Section 7, computational complexity of the proposed scheme is discussed. To show the capability of the proposed adaptive controller, an application to a two-link robot manipulator operating in the vertical plane is considered in Section 8.

4.2 State Space Model of Manipulator Dynamics

As discussed in Chapter 2 the dynamic model of an n-joint rigid link manipulator is given by

$$\tau(t) = M(\theta)\,\ddot{\theta}(t) + V(\theta,\dot{\theta}) + G(\theta) + \eta(t) \qquad (4.2.1)$$

For the time being, we shall ignore the term $\eta(t)$ which represents the uncertainties such as those due to friction, backlash, etc. but its effect will be accounted for later in designing the adaptive controller. In the analysis that follows the argument t will be dropped for convenience.

The vectors V and G can be written as [23,31,33]

$$V(\theta,\dot{\theta}) = B(\theta,\dot{\theta})\,\dot{\theta}$$

and

$$G(\theta) = C(\theta)\,\theta$$

where $B(\theta,\dot{\theta})$ and $C(\theta)$ are $n \times n$ matrices. Therefore the set of nonlinear equations (4.2.1) can be written as

$$\tau = M(\theta)\,\ddot{\theta} + B(\theta,\dot{\theta})\,\dot{\theta} + C(\theta)\,\theta \qquad (4.2.2)$$

This equation can be rewritten in the state-space form

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & I \\ M^{-1}(\theta)\,C(\theta) & M^{-1}(\theta)B(\theta,\dot{\theta}) \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(\theta) \end{bmatrix} \tau \qquad (4.2.3)$$

Assumption 1:

Assuming that the manipulator parameters are slowly time-varying, equation (4.2.3) can be approximated by the following piecewise linear time-varying state space model

$$\dot{\chi} = \begin{bmatrix} 0 & I \\ -A_1 & -A_2 \end{bmatrix} \chi + \begin{bmatrix} 0 \\ H \end{bmatrix} \tau \qquad (4.2.4)$$

where $\chi_1 = \theta$, $\chi_2 = \dot{\theta}$, and $\chi = \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix}$.

i.e.

$$\dot{\chi} = \Phi \chi + \Omega \tau$$

where, $\chi \in \mathbb{R}^{2n \times 1}$, $\tau \in \mathbb{R}^{n \times 1}$, $\Phi \in \mathbb{R}^{2n \times 2n}$ and $\Omega \in \mathbb{R}^{2n \times n}$. The matrices Φ and Ω are constant at a given value of time and are to be identified. Since the matrix H has full rank for all values of time, it follows that the linear model is controllable and therefore stabilizable by state feedback.

Assumption 2:

The total trajectory time is divided into a number of intervals, say r , of duration Δt_k , $k = 1,...,r$. Over the interval Δt_k , the time-varying system (4.2.4) is approximated by a linear time-invariant system

$$\dot{\chi} = \begin{bmatrix} 0 & I \\ -A_1(t_{k-1}) & -A_2(t_{k-1}) \end{bmatrix} \chi + \begin{bmatrix} 0 \\ H(t_{k-1}) \end{bmatrix} \tau \qquad (4.2.5)$$

where, t_{k-1} denotes the time at the start of interval Δt_k .

The parameters of (4.2.5) are to be estimated on-line using recursive identification. As a robot moves along the trajectory, for each Δt_k , we have a particular model of the system. This set of models represents the dynamic behavior of the manipulator along the trajectory. A feedforward-feedback controller is designed for each linear time-

invariant model.

4.3 THE ADAPTIVE CONTROL ALGORITHM

We now consider the design of a controller for the manipulator to enable it to track a desired trajectory specified in terms of a set of time-dependent desired joint position, velocity and acceleration vectors, i.e., $\theta_d(t)$, $\dot{\theta}_d(t)$ and $\ddot{\theta}_d(t)$. The goal of the controller is to minimize the deviation from the desired path, i.e, to achieve tracking while maintaining stability of the resulting closed-loop system The adaptive controller that is obtained in this section is based on the model (4.2.4) and its basic structure is shown in Figure 4.1.

The control law for the manipulator is chosen as

$$\tau = \ddot{\theta}_d + \delta\tau + w(t) \tag{4.3.1}$$

where, $\delta\tau \in \mathbb{R}^{n\times 1}$ is the correcting torque vector obtained from a feedback controller, and $w(t)$ is generated by a feedforward controller The feedback controller is chosen as a multivariable proportional-derivative (PD) controller defined by

$$\delta\tau = K_p\ e + K_v\ \dot{e} \tag{4.3.2}$$

where e , $\dot{e} \in \mathbb{R}^{n\times 1}$ are vectors of position and velocity errors, respectively

$$e = \theta_d - \theta$$
$$\dot{e} = \dot{\theta}_d - \dot{\theta}$$

and K_p , $K_v \in \mathbb{R}^{n\times n}$ are the proportional and derivative feedback gain matrices

To design the feedforward controller, let us substitute the control law (4.3.1) into the model (4.2.4). This yields

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & I\\ -A_1 & -A_2\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ H\end{bmatrix}\left[\theta_d + \delta\tau + w\right]$$

from where

$$\ddot{\theta} = -A_1\theta - A_2\dot{\theta} + H\theta_d + H\delta\tau + Hw$$

Substituting for $\theta, \dot{\theta}, \ddot{\theta}$ from the error equations, we get

$$\ddot{\theta}_d - \ddot{e} = - A_1 \left[\theta_d - e\right] - A_2 \left[\dot{\theta}_d - \dot{e}\right] + H \ddot{\theta}_d + H \delta\tau + H w$$

Using (4.3.2), we get

$$\ddot{\theta}_d + A_2\dot{\theta}_d + A_1\theta_d = \ddot{e} + A_1 e + A_2\dot{e} + H\ddot{\theta}_d + H\left[K_v \dot{e} + K_p e + w\right]$$

which can be rearranged to give

$$\ddot{e} + \left[A_2 + HK_v\right]\dot{e} + \left[A_1 + HK_p\right]e + Hw = \ddot{\theta}_d + A_2 \dot{\theta}_d + A_1 \theta_d - H\ddot{\theta}_d$$

In order to make the steady state error zero, we choose

$$H w = \ddot{\theta}_d + A_2 \dot{\theta}_d + A_1 \theta_d - H \ddot{\theta}_d$$

or

$$w = H^{-1} \left[(I - H) \ddot{\theta}_d + A_2 \dot{\theta}_d + A_1 \theta_d \right]$$

It is seen that the feedforward control w is based on the reference input, and is updated in terms of the identified model parameters. The control input to the manipulator is now given by

$$\tau = \ddot{\theta}_d + K_v \dot{e} + K_p e + H^{-1} \left[(I - H) \ddot{\theta}_d + A_2 \dot{\theta}_d + A_1 \theta_d \right] \quad (4.3.3)$$

Now, let us consider the manipulator dynamic equation (4.2.2). On applying the control law (4.3.3) to this model, we get

$$\ddot{\theta}_d + K_v \dot{e} + K_p e + H^{-1} \left[(I - H) \ddot{\theta}_d + A_2 \dot{\theta}_d + A_1 \theta_d \right] = M\ddot{\theta} + B\dot{\theta} + C\theta \quad (4.3.4)$$

From (4.2.3) and (4.2.4), during the time interval Δt_k, we have

$$A_1 = M^{-1} C \quad (4.3.5a)$$

$$A_2 = M^{-1} B \quad (4.3.5b)$$

$$H = M^{-1} \quad (4.3.5c)$$

Using (4.3.5) in (4.3.4), yields

$$\ddot{\theta}_d + K_v \dot{e} + K_p e + H^{-1}\ddot{\theta}_d - \ddot{\theta}_d + H^{-1}A_2\dot{\theta}_d + H^{-1}A_1\theta_d = H^{-1}\left[\ddot{\theta} + A_2\dot{\theta} + A_1\theta\right]$$

which can be simplified and rearranged as

$$H^{-1}\left[\ddot{\theta}_d - \ddot{\theta}\right] + H^{-1}A_2\left[\dot{\theta}_d - \dot{\theta}\right] + H^{-1}A_1\left[\theta_d - \theta\right] + K_v\dot{e} + K_p e = 0$$

from which we obtain the error equation (during the interval Δt_k)

$$\ddot{e} + \left[A_2 + HK_v\right]\dot{e} + \left[A_1 + HK_p\right]e = 0$$

which can be written in state space form as

$$\begin{bmatrix}\dot{e}\\ \ddot{e}\end{bmatrix} = \begin{bmatrix}0 & I\\ -\Lambda_p & -\Lambda_v\end{bmatrix}\begin{bmatrix}e\\ \dot{e}\end{bmatrix} \qquad (4.3.6)$$

where

$$\Lambda_v = A_2 + H\,K_v$$
$$\Lambda_p = A_1 + H\,K_p$$

The controller gains are therefore given by

$$K_v = H^{-1}\left[\Lambda_v - A_2\right]$$
$$K_p = H^{-1}\left[\Lambda_p - A_1\right]$$

The matrices Λ_p and Λ_v are obtained from the desired performance of the tracking error. The model used in the controller design is a continuous-time model The identification of such a model requires about $(4n^2)$ parameters The digital simulation of the model is very time consuming because of the large number of floating point operations involved Therefore, to simplify the construction of the controller, we drop the coupling terms due to joint interactions Such a control is called independent joint control Dropping these coupling terms will give rise to errors which will affect the stability and response of the system However, we can compensate for these errors to some extent by considering the terms that are neglected as being part of the unmodeled dynamics and by modifying the control strategy to account for them For independent joint control (neglecting joint interactions) we have

$$K_v = diag(k_{v_i}) \qquad K_p = diag(k_{p_i}) \qquad i = 1,\ldots,n$$
$$A_1 = diag(a_i) \qquad A_2 = diag(b_i) \quad H \quad diag(h_i) \qquad i = 1,\ldots,n$$

Then Λ_v and Λ_p can be written as

$$\Lambda_p = diag\,[a_i + h_i\, k_{p_i}] \qquad i=1,...,n$$

$$\Lambda_v = diag\,[b_i + h_i\, k_{v_i}] \qquad i=1,...,n$$

On neglecting joint interactions, the model equations are decoupled and can be written in the form

$$\begin{bmatrix} \dot{\theta}_i \\ \ddot{\theta}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_i & -b_i \end{bmatrix} \begin{bmatrix} \theta_i \\ \dot{\theta}_i \end{bmatrix} + \begin{bmatrix} 0 \\ h_i \end{bmatrix} \tau_i \qquad i=1,...,n \qquad (4.3.7)$$

for each joint the error equation is then given by

$$\begin{bmatrix} \dot{e}_i \\ \ddot{e}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\left(a_i + h_i\, k_{p_i}\right) & -\left(b_i + h_i\, k_{v_i}\right) \end{bmatrix} \begin{bmatrix} e_i \\ \dot{e}_i \end{bmatrix} \qquad i=1,...,n \qquad (4.3.8)$$

which implies that $e_i\ ,\ \dot{e}_i \to 0$ as $t \to \infty$ provided that $a_i + h_i\, k_{p_i} > 0$ and $b_i + h_i\, k_{v_i} > 0\ ,\ i=1,...,n$

Desired Performance

Now, let us define the desired performance of the tracking error for the manipulator. This can be done by requiring that the error satisfies a set of second-order homogeneous differential equations of the form

$$\ddot{e}_i + 2\varsigma_i \omega_i \dot{e}_i + \omega_i^2 e_i = 0 \qquad i=1,...,n \qquad (4.3.9)$$

where ς_i and ω_i are the damping factor and undamped natural frequency respectively and are to be specified by the designer. Equation (4.3.9) can be put in the state space form

$$\dot{z}_r = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} z_r \qquad (4.3.10)$$

where $\Lambda_v = diag\,(2\varsigma_i \omega_i)$, $\Lambda_p = diag\,(\omega_i^2)$ are $n \times n$ constant matrices and $z_r = \left[e_r^T \ \ \dot{e}_r^T\right]^T \in \mathbb{R}^{2n \times 1}$.

Inclusion of the Effects of Unmodeled Dynamics

The error equation (4.3.10) can be written as

$$\dot{z}_r = \Pi z_r \tag{4.3.11}$$

where, $\Pi \in \mathbb{R}^{2n \times 2n}$. Equation (4.3.11) represents the desired behavior of the closed-loop system. If the model parameters are taken to be diagonal matrices, i.e.

$$A_1 = A_{1d} \qquad A_2 = A_{2d} \qquad H = H_d ,$$

then we get the following error equation

$$\begin{bmatrix} \dot{e}_a \\ \ddot{e}_a \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} \begin{bmatrix} e_a \\ \dot{e}_a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -M_1 & -M_2 \end{bmatrix} \begin{bmatrix} e_a \\ \dot{e}_a \end{bmatrix} + \begin{bmatrix} 0 \\ \xi(t) \end{bmatrix} \tag{4.3.12}$$

where, $M_1, M_2 \in \mathbb{R}^{n \times n}$ denote the effect of coupling between the joints and e_a is the actual error. The vector $\xi(t) = H\ \eta(t) \in \mathbb{R}^{n \times 1}$ denotes the effect of disturbances and other unmodeled dynamics. Equation (4.3.12) can be written as

$$\dot{z}_a = \Pi z_a + F_a(t) \tag{4.3.13}$$

where $F_a(t)$ is a vector denoting unmodeled dynamics and disturbances and has the form

$$F_a(t) = \begin{bmatrix} 0 \\ H f_a(t) \end{bmatrix} \Delta \begin{bmatrix} 0 & 0 \\ M_1 & M_2 \end{bmatrix} \begin{bmatrix} e_a \\ \dot{e}_a \end{bmatrix} + \begin{bmatrix} 0 \\ \xi(t) \end{bmatrix}$$

Equation (4.3.13) represents the behavior of the system without compensation for the unmodeled dynamics.

Next, we assume that $F_a(t)$ is an unknown slowly time-varying function and we introduce another term in the feedback law to compensate for its effect. The manipulator dynamical model (4.2.1) can be represented in terms of the nominal values of the manipulator dynamics, $M^o(\theta)$, $B^o(\theta,\dot{\theta})$, $C^o(\theta)$ and a nonlinear term $f_a(t)$ which represents coupling between the joints and other unmodeled slowly time varying dynam

ics. Then (4.2.1) can be rewritten as

$$\tau = f_a(t) + M^o(\theta)\,\ddot{\theta} + B^o(\theta,\dot{\theta})\,\dot{\theta} + C^o(\theta)\,\theta \qquad (4.3.14)$$

In order to compensate for the effects of these unmodeled dynamics, a control term $f_c(t)$ is added to the control law (4.3.3). The total input to the manipulator becomes

$$\tau = H^{-1}\left[(I-H)\ddot{\theta}_d + A_2\dot{\theta}_d + A_1\theta_d\right] + \ddot{\theta}_d + K_v\,\dot{e}_a + K_p\,e_a + f_c(t) \qquad (4.3.15)$$

where K_p, K_v, H, A_1 and A_2 are diagonal matrices. The control scheme (4.3.15) developed in this section consists of three parts. The first part is a feedforward controller and is a special form of full dynamic compensation with three terms, corresponding to inertial, centrifugal and Coriolis, and gravitational torques/forces. This part, based on the estimation of the parameters of the manipulator, attempts to provide the nominal joint torques/forces necessary to make the desired motions (assuming that the identified dynamics are exact). The second part contains two terms representing the PD controller, which also depends on the estimated parameters. Its purpose is to regulate the trajectories resulting from the feedforward part about the desired trajectories. The third part, is a compensation term which is used to compensate for the effects of unmodeled dynamics. The overall block diagram of the above adaptive control scheme is shown in Fig. 4.1.

Remarks:

1. The control scheme (4.3.15) is simple to construct and does not require *a priori* knowledge of any nonlinear terms or complex dynamics of the manipulator. The diagonal entries of H, A_1 and A_2 are obtained using the recursive identification scheme described in Chapter 3. The convergence of the adaptive controller is independent of the initial values chosen for A_1, A_2, H, K_v and K_p.

2. It should be noted that the guaranteed convergence of the tracking errors to zero does not imply convergence of the estimated parameters to their exact values since only the

diagonal elements of the model are identified.

4.4 Stability Analysis of the Closed-Loop System

In Section 4.3, the input control torque (4.3.15) for the manipulator is derived. A compensating function $f_c(t)$ was added to compensate against the effects of unmodeled dynamics. In this section we will compute $f_c(t)$ so as to approximate $f_a(t)$. This can be derived by applying the control law (4.3.15) to the manipulator (4.3.14), and get

$$f_c(t)+K_v\dot{e}_a+K_p e_a+H^{-1}\left[\ddot{\theta}_d+A_2\dot{\theta}_d+A_1\theta_d\right]=f_a(t)+M^o\ddot{\theta}+B^o\dot{\theta}+C^o\theta \quad (4.4.1)$$

where $e_a = \theta_d - \theta$ is the actual error with independent joint control. Assuming that M^o, B^o and C^o satisfy (4.3.5), we obtain the following error equation

$$\ddot{e}_a + \left[A_2 + HK_v\right]\dot{e}_a + \left[A_1 + HK_p\right]e_a = H\left[f_a(t) - f_c(t)\right] \quad (4.4.2)$$

which can be written in state space form as

$$\begin{bmatrix}\dot{e}_a\\ \ddot{e}_a\end{bmatrix} = \begin{bmatrix}0 & I\\ -A_p & -A_v\end{bmatrix}\begin{bmatrix}e_a\\ \dot{e}_a\end{bmatrix} + \begin{bmatrix}0\\ H\end{bmatrix}\begin{pmatrix}f_a & f_c\end{pmatrix}$$

or

$$\dot{z}_a = \Pi z_a + \beta\begin{pmatrix}f_a & f_c\end{pmatrix} \quad (4.4.3)$$

where,

$$\beta = \begin{bmatrix}0\\ H\end{bmatrix}$$

Let, $f_m = f_a - f_c$, i.e. $f_a = f_m + f_c$, where f_m denotes the error in the approximation of the actual unmodeled dynamics by the computed value $f_c(t)$. In (4.4.3), we note that A_p, A_v and H are diagonal matrices. Therefore, by rearranging the elements of z_a as

$$z_a = \begin{bmatrix}z_{a_1}^T & z_{a_2}^T & \cdots & z_{a_n}^T\end{bmatrix}^T \quad (4.4.4)$$

where $z_{a_i} = [e_{a_i} \quad \dot{e}_{a_i}]^T$, we can write (4.4.3) as a set of n decoupled second order systems of the form

$$\dot{z}_{a_i} = \Pi_i \; z_{a_i} + \beta_i \, f_{m_i} \qquad i = 1,\ldots,n \tag{4.4.5}$$

where

$$\Pi_i = \begin{bmatrix} 0 & 1 \\ -\lambda_{p_i} & -\lambda_{v_i} \end{bmatrix} , \qquad \beta_i = \begin{bmatrix} 0 \\ h_i \end{bmatrix}$$

and f_{m_i} is the i^{th} element of f_m.

Theorem 4.1

Assuming that $f_{a_i}(t)$ is slowly time-varying, i.e. $\dot{f}_{a_i}(t) \approx 0$, the closed-loop system (4.4.5) is asymptotically stable if $f_{c_i}(t)$ satisfies the equation

$$\dot{f}_{c_i}(t) = \frac{1}{q_{0_i}} \beta_i^T P_i \; z_{a_i}$$

Proof

Choose a Lyapunov function

$$V_i(t) = z_{a_i}{}^T \; P_i \; z_{a_i} + q_{0_i} \; f_{m_i}{}^2 \tag{4.4.6}$$

where $q_{0_i} > 0$ and $P_i \in \mathbb{R}^{2\times 2}$ is a positive-definite matrix. Taking derivatives (with respect to t) of both sides of (4.4.6), we get

$$\dot{V}_i = \dot{z}_{a_i}{}^T P_i \, z_{a_i} + z_{a_i}{}^T P_i \, \dot{z}_{a_i} + 2 q_{0_i} \, f_{m_i} \dot{f}_{m_i}$$

Substituting for $\dot{z}_{a_i}$ from (4.4.5) yields

$$\dot{V}_i = z_{a_i}{}^T \left[\Pi_i^T P_i + P_i \Pi_i \right] z_{a_i} + 2 \left[q_{0_i} \, f_{m_i} \dot{f}_{m_i} + f_{m_i} \beta_i^T P_i \, z_{a_i} \right]$$

where P_i is the positive-definite solution of the Lyapunov equation

$$\Pi_i^T P_i + P_i \Pi_i = -2 Q_i \tag{4.4.7}$$

with $Q_i \in \mathbb{R}^{2\times 2}$ an arbitrary positive-definite matrix. Then,

$$\dot{V}_1 = -2 z_{e_1}^T Q_1 z_{e_1} + 2 \left[q_{o_1} f_{m_1} f_{m_1} + f_{m_1} \beta_1^T P_1 z_{e_1} \right] \qquad (4.4.8)$$

For stability, it is required that $\dot{V}_1(t) \leq 0$ along the trajectory $z_{e_1}(t)$ of (4.4.5). For this, it is sufficient (from (4.4.8)) that

$$f_{m_1} \left[q_{o_1} f_{m_1} + \beta_1^T P_1 z_{e_1} \right] = 0$$

which can be satisfied if $f_{m_1} = 0$ or if

$$q_{o_1} f_{m_1} + \beta_1^T P_1 z_{e_1} = 0 \qquad (4.4.9)$$

i.e., if

$$f_{m_1}(t) = -\frac{1}{q_{o_1}} \beta_1^T P_1 z_e \qquad (4.4.10)$$

The case where $f_{m_1} = 0$ is trivial because the stability of (4.4.5) then follows from the fact that H_1 is (by choice of λ_{p_1} and λ_{r_1}) an asymptotically stable matrix. Since f_{e_1} is slowly time-varying, $f_{e_1} \approx 0$. Therefore by definition of f_m, $f_{m_1} = f_{r_1}$. Then the second case is satisfied if

$$f_{r_1} = -\frac{1}{q_{r_1}} \beta_1^T P_1 z_{e_1} \qquad (4.4.11)$$

At this point we choose the adaptive law as (4.4.11). Consequently we get

$$\dot{V}_1 = -2 z_e^T Q_1 z_e \leq 0$$

Since $\dot{V}_1$ is negative semi-definite and V_1 is positive definite, we conclude that the origin is stable. Now by invoking LaSalle's Theorem (2.4.9), it can be shown that regulation of the closed-loop system takes place, i.e., $\lim_{t \to \infty} z_e(t) = 0$ and that $\lim_{t \to \infty} f_m(t) =$ constant. This completes the proof of the theorem.

Remark:

We can also achieve asymptotic stability of the overall system by redefining f_m as

$$\dot{f}_{m_i} = -\frac{1}{q_{0_i}} \beta_i^T P_i \; z_{a_i} - \sigma_{s_i} \; f_{m_i} \tag{4.4.12}$$

where σ_{s_i} is a positive scalar. Then using (4.4.12) in (4.4.8), we get

$$\dot{V}_i = -2 z_{a_i}{}^T \; Q_i \; z_{a_i} + 2 \left[q_{0_i} f_{m_i} \left(-q_{0_i}{}^{-1} \beta_i^T P_i \; z_{a_i} - \sigma_{s_i} \; f_{m_i} \right) + f_{m_i} \; \beta_i^T P_i \; z_{a_i} \right]$$

Choosing $q_{0_i} = \sigma_{s_i}$ results in

$$\dot{V}_i = -2 \; z_{a_i}{}^T \; Q_i \; z_{a_i} - 2\sigma_{s_i}{}^2 \; f_{m_i}{}^2 < 0$$

for all $z_{a_i} \neq 0$, $f_{m_i} \neq 0$, i.e. $\dot{V}_i$ is negative-definite and the system (4.4.5, 4.4.12) is asymptotically stable.

The result in Theorem 4.1 gives us a way of determining $f_{c_i}(t)$. Writing P_i as

$$P_i = \begin{bmatrix} p_{1_i} & p_{2_i} \\ p_{2_i} & p_{3_i} \end{bmatrix}$$

we can express (4.4.11) as

$$\dot{f}_{c_i} = \frac{1}{q_{0_i}} \begin{bmatrix} 0 & h_i \end{bmatrix} \begin{bmatrix} p_{1_i} & p_{2_i} \\ p_{2_i} & p_{3_i} \end{bmatrix} \begin{bmatrix} e_{a_i} \\ \dot{e}_{a_i} \end{bmatrix} \tag{4.4.13}$$

which gives,

$$\dot{f}_{c_i}(t) = \frac{h_i}{q_{0_i}} \left[p_{2_i} \; e_{a_i} + p_{3_i} \; \dot{e}_{a_i} \right] \tag{4.4.14}$$

Equation (4.4.14) can be solved for $f_{c_i}(t)$ as

$$f_{c_i}(t) = f_{c_i}(t_o) + \frac{h_i \, p_{2_i}}{q_{0_i}} \int_{t_o}^{t} e_{a_i}(t) dt + \frac{h_i \, p_{3_i}}{q_{0_i}} e_{a_i}(t) \tag{4.4.15}$$

where $f_{c_i}(t_o)$ denotes the initial value of $f_{c_i}(t)$. This gives the term $f_{c_i}(t)$ which compensate against the effects of unmodeled dynamics in the adaptive control law.

In the next section we will show how p_{2_i} and p_{3_i} can be determined.

4.5 Pole Assignment Technique

From (4.4.10) and (4.4.11) it is seen that $f_{c_i}(t)$ and $f_{m_i}(t)$ depend on p_{2_i} and p_{3_i}. In this section we will show how p_{2_i} and p_{3_i} can be determined (subject to the constraint that the corresponding Q_i is positive-definite) by placing the closed loop poles at desired locations in the left-half of complex plane. This allows us to achieve satisfactory transient behavior for $z_{a_i}(t)$ and $f_{m_i}(t)$.

Writing (4.4.5) and (4.4.10) together results in the following augmented closed-loop system

$$\begin{bmatrix} \dot{z}_{a_i} \\ \dot{f}_{m_i} \end{bmatrix} = \begin{bmatrix} \Pi_i & \beta_i \\ \alpha_i^T & 0 \end{bmatrix} \begin{bmatrix} z_{a_i} \\ f_{m_i} \end{bmatrix} \qquad i = 1,\ldots,n \tag{4.5.1a}$$

where

$$\alpha_i^T = \frac{1}{q_{0_i}} \beta_i^T P_i = \frac{h_i}{q_{0_i}} \begin{bmatrix} p_{2_i} & p_{1_i} \end{bmatrix}$$

or

$$\begin{bmatrix} \dot{e}_{a_i} \\ \dot{e}_{a_i} \\ \dot{f}_{m_i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \lambda_{f_i} & \lambda_{v_i} & h_i \\ \dfrac{h_i}{q_{0_i}} p_{2_i} & \dfrac{h_i}{q_{c_i}} p_{1_i} & 0 \end{bmatrix} \begin{bmatrix} e_{a_i} \\ e_{a_i} \\ f_{m_i} \end{bmatrix} \tag{4.5.1b}$$

The characteristic equation corresponding to (4.5.1) is given by

$$s \det \left[sI - \Pi_i \right] - \alpha_i^T \operatorname{adj} \left[sI - \Pi_i \right] \beta_i = 0$$

which can be simplified to yield

$$s^3 + \lambda_{v_i} s^2 + \left[\lambda_{f_i} + \frac{h_i^2}{q_{c_i}} p_{1_i} \right] s + \frac{h_i^2}{q_{c_i}} p_{2_i} = 0 \tag{4.5.2}$$

From (4.5.2) it is clear that we do not have enough freedom to assign all three eigenvalues arbitrarily by choice of p_{2_i} and p_{3_i} since the coefficient of s^3 is indepen

dent of p_{2_i} and p_{3_i} . However, there is additional freedom available in the choice of λ_{v_i} and λ_{p_i}, but this makes the problem nonlinear since p_{2_i} and p_{3_i} depend on λ_{v_i} and λ_{p_i} via (4.4.7). Since Π_i , P_i and Q_i are 2×2 matrices, it is not difficult to obtain explicit expressions relating λ_{v_i} , λ_{p_i} and the elements of P_i and Q_i (as we will show below). From these expressions and (4.5.2), it is possible to choose values for the elements of Q_i which will ensure that the roots of (4.5.2) are at or near desired locations in the left-half of the complex plane. This, in turn, will ensure that $z_{a_i}(t)$ and $f_{m_i}(t) \rightarrow 0$ sufficiently fast or in some desired manner.

Remark:

It should be noted that in obtaining an expression for $f_{c_i}(t)$, we have assumed that $f_{a_i}(t)$ is slowly time-varying in comparison with $f_{m_i}(t)$. The validity of this assumption can be ensured by choosing $q_{0_i}(> 0)$ sufficiently small which will tend to increase the magnitude of $\dot{f}_{m_i}(t)$.

Since the 2×2 symmetric matrix P_i is required to be positive-definite, the element p_{3_i} should be positive. Also from (4.5.2), it is necessary that $p_{2_i} > 0$ for stability. we now show how $p_{2_i} > 0$ and $p_{3_i} > 0$ can be selected. In the proposed algorithm, it is required that the closed-loop poles be placed at desired locations in the left-half of the complex plane. This can be achieved by requiring that the closed-loop characteristic equation (4.5.2) has a desired set of coefficients. Assume that the desired characteristic equation is given by

$$s^3 + a_{1_i}\, s^2 + a_{2_i}\, s + a_{3_i} = 0 \qquad (4.5.3)$$

By equating the coefficients of like powers of s in (4.5.2) and (4.5.3), we get $a_{1_i} = \lambda_{v_i}$, which can be chosen equal to $2\zeta_i \omega_i$; also, we can choose $\lambda_{p_i} = \omega_i^2$. Here, ζ_i and ω_i are the damping factor and undamped natural frequency, respectively (see Eq.(4.3.9)). Also we have

$$a_{2_o} = \lambda_{p_o} + \frac{h_1^2}{q_{o_o}} p_{3_o} \qquad \text{and} \qquad a_{3_o} = \frac{h_1^2}{q_{o_o}} p_{2_o} \tag{4.5.4}$$

from which we get expressions for p_{3_o} and p_{2_o} as

$$p_{3_o} = \frac{\left(a_{2_o} - \lambda_{p_o}\right) q_{o_o}}{h_1^2} \qquad \text{and} \qquad p_{2_o} = \frac{a_{3_o} q_{o_o}}{h_1^2} \tag{4.5.5}$$

The solution of Lyapunov equation (4.4.7) with $Q_1 = diag\,(q_{1_o} \;\; q_{2_o})$, gives

$$p_{2_o} = \frac{q_{1_o}}{\lambda_{p_o}} \qquad \text{and} \qquad p_{3_o} = \frac{q_{2_o}}{\lambda_{1_o}} + \frac{q_{1_o}}{\lambda_{p_o}\lambda_{1_o}} \tag{4.5.6}$$

by using the values of p_{2_o} and p_{3_o} in (4.5.5), we get

$$\frac{q_{1_o}}{\lambda_{p_o}} = \frac{a_{3_o} q_{o_o}}{h_1^2} \qquad \text{and} \qquad \frac{\left(a_{2_o} - \lambda_{p_o}\right) q_{o_o}}{h_1^2} = \frac{q_{2_o}}{\lambda_{1_o}} + \frac{q_{1_o}}{\lambda_{p_o}\lambda_{1_o}} \tag{4.5.7}$$

from which we can get the expressions for q_{1_o} and q_{2_o} as

$$q_{1_o} = \frac{a_{3_o}\lambda_{p_o} q_{o_o}}{h_1^2} \qquad \text{and} \qquad q_{2_o} = \frac{q_{o_o}}{h_1^2}\left[\left(a_{2_o} - \lambda_{p_o}\right)\lambda_{1_o} - a_{1_o}\right] \tag{4.5.8}$$

Therefore, we can see that to have a positive-definite diagonal Q_1 matrix we must select the desired locations for the closed-loop poles such that

$$a_{3_o} > 0 \qquad \text{and} \qquad a_{2_o} > \left(\lambda_{f_o} + \frac{a_{1_o}}{\lambda_{1_o}}\right) \tag{4.5.9}$$

Remark:

It can be shown that the above results lead to a positive definite solution of Lyapunov equation. Eq (4.4.7) can be written as

$$\begin{bmatrix} p_{1_o} & p_{2_o} \\ p_{2_o} & p_{3_o} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ \lambda_{f_o} & \lambda_{1_o} \end{bmatrix} + \begin{bmatrix} 0 & \lambda_f \\ 1 & \lambda_1 \end{bmatrix}\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} = \begin{bmatrix} 2q_{1_o} & 0 \\ 0 & 2q_{2_o} \end{bmatrix}$$

This gives

$$-2\,p_{2,}\,\lambda_{p,} = -2q_{1,}$$
$$p_{1,} - p_{2,}\,\lambda_{v,} - p_{3,}\,\lambda_{p,} = 0$$
$$2p_{2,} - 2p_{3,}\,\lambda_{v,} = -2q_{2,}$$

The solution of these equations is

$$p_{1,} = \frac{q_{1,}}{\lambda_{v,}} + \frac{q_{1,}}{\lambda_{p,}}\lambda_{v,} + \frac{q_{2,}}{\lambda_{v,}}\lambda_{p,} \tag{4.5.10a}$$

$$p_{2,} = \frac{q_{1,}}{\lambda_{p,}} \tag{4.5.10b}$$

$$p_{3,} = \frac{q_{2,}}{\lambda_{v,}} + \frac{q_{1,}}{\lambda_{p,}\lambda_{v,}} \tag{4.5.10c}$$

If $q_{1,}$ and $q_{2,}$ are positive, then for P_i to be a positive-definite matrix, the eigenvalues of P_i should be positive. The characteristic equation for P_i is

$$s^2 - (p_{1,} + p_{3,})\,s + (p_{1,}p_{3,} - p_{2,}^2) = 0 \tag{4.5.11}$$

The eigenvalues of P_i are positive, if and only if

$$(p_{1,} + p_{3,}) > 0 \qquad \text{and} \qquad (p_{1,}p_{3,} - p_{2,}^2) > 0 \tag{4.5.12}$$

Using (4.5.10) we get

$$p_{1,} + p_{3,} = q_{1,}\left[\frac{1}{\lambda_{v,}} + \frac{1}{\lambda_{p,}\lambda_{v,}} + \frac{\lambda_{v,}}{\lambda_{p,}}\right] + q_{2,}\left[\frac{\lambda_{p,}}{\lambda v_i} + \frac{1}{\lambda_{v,}}\right] \tag{4.5.13}$$

and

$$p_{1,}p_{3,} - p_{2,}^2 = 2\frac{q_{1,}q_{2,}}{\lambda_{v,}^2} + \frac{q_{1,}^2}{\lambda_{p,}\lambda_{v,}^2} + \frac{q_{1,}q_{2,}}{\lambda_{p,}} + \frac{q_{2,}^2\lambda_{p,}}{\lambda_{v,}} \tag{4.5.14}$$

Since, $\lambda_{p,}, \lambda_{v,} > 0$ and from (4.5.8) and (4.5.9) we ensure that $q_{1,}$ and $q_{2,} > 0$. Therefore, it can be seen that conditions (4.5.12) are satisfied and P_i is a positive definite matrix. Also, $p_{1,}$, $p_{2,}$ and $p_{3,} > 0$.

4.6 IDENTIFICATION TECHNIQUE

In order to estimate the model parameters on-line, a recursive least-squares param-

eter identification technique is used [28,29]. The autoregressive identification model is assumed to have the same number of inputs and outputs and the parameters of the model are assumed to be slowly time-varying but that the speed of variation is slower than the adaptation speed. This is illustrated in our simulation results The same assumption have been made in [25]. In order to simplify the identification scheme it is assumed that the measurements noise is negligible. Also, all the states of the manipulator are assumed to be measurable.

The model can be written in the general form

$$y(t+\Delta t) = \hat{y}(t+\Delta t \mid t) + e_s(t+\Delta t) \tag{4.6.1}$$

$$\hat{y}(t+\Delta t \mid t) = \Psi^T(t)\,\hat{\Theta}(t) \tag{4.6.2}$$

where e_s is the prediction error vector and y represents the output vector. This model is used to identify the diagonal matrices A_1, A_2 and H in (4.2.4) The vector $\hat{\Theta}(t) \in \mathbb{R}^m$ contains the m parameters to be identified and is written as

$$\hat{\Theta}(t) = \left[a_1 \ldots a_n \; ; \; b_1 \ldots b_n \; ; \; h_1 \ldots h_n \; ; \; c_1 \ldots c_n \right]^T \tag{4.6.3}$$

or

$$\hat{\Theta}(t) = \left[\hat{\Theta}_1 \ldots \hat{\Theta}_m \right]^T$$

where c_n is the residual error coefficient. The vector $\Psi(t)$ contains the measurements upto time t and can be written as

$$\Psi^T(t) = \left[\theta^T(t-1) \;\; \dot{\theta}^T(t-1) \;\; \tau^T(t-1) \;\; e_r^T(t-1)\right] \tag{4.6.4}$$

where e_r is the residual error vector given by

$$e_r(t+\Delta t) = y(t+\Delta t) - \Psi^T(t)\,\hat{\Theta}(t+\Delta t) \tag{4.6.5}$$

The algorithm for parameter estimation is structured so that one vector at a time is estimated (for each joint) The algorithm minimizes a loss function of the form

$$\Gamma(t) = \sum_{\sigma=t_0}^{t} \frac{e_s^2(\sigma)}{\nu + \Psi^T(\sigma)\,P(\sigma)\,\Psi(\sigma)} \tag{4.6.6}$$

The solution for the least-squares criterion, i.e. minimization of $\Gamma(t)$ relative to the estimated parameters, uses the recursive parameter estimation scheme (Fig. 3.2) given in Chapter 3 [29]:

$$\hat{\Theta}(t+\Delta t) = \hat{\Theta}(t) + L(t)\left[y(t+\Delta t) - \Psi^T(t)\,\hat{\Theta}(t)\right] \qquad (4.6.7)$$

where $L(t+\Delta t) = P(t+\Delta t)\,\Psi(t+\Delta t)$ is the correcting factor. The term $P(t+\Delta t)$ is the adaptation gain given by

$$P(t+\Delta t) = \frac{1}{\nu}\left[P(t) - \frac{P(t)\Psi(t+\Delta t)\Psi^T(t+\Delta t)P(t)}{\nu + \Psi^T(t+\Delta T)P(t)\Psi(t+\Delta t)}\right] \qquad (4.6.8)$$

where ν is the forgetting factor with $0 < \nu < 1$ and $P(t)$ is a block diagonal positive-definite matrix, each diagonal block is a 4×4 matrix represents one joint. The initial value, $P(0)$, is chosen as $P(0) = \alpha I$, α is a large positive scalar.

4.7 Computational Complexity of the Adaptive Controller

For the independent joint control scheme proposed in this chapter, an operational count shows that the computation of the joint torques requires approximately

$$2n^2 + 9n \quad \textit{multiplications} \quad \text{and} \quad 2n^2 + 11n \quad \textit{additions}$$

and the identification scheme requires approximately

$$76\,n \quad \textit{multiplications} \quad \text{and} \quad 60\,n \quad \textit{additions}$$

resulting in a total of

$$2n^2 + 85\,n \quad \textit{multiplications} \quad \text{and} \quad 2n^2 + 71n \quad \textit{additions}$$

for the adaptive controller. For a 6 degrees-of-freedom manipulator, this gives 582 multiplications and 498 additions. For full joint interaction, the same adaptive control scheme would require approximately 2565 multiplications and 2455 additions for a 6 degrees-of-freedom manipulator, which would result in a significantly lower servo rate than when independent joint control is used.

4.8 NUMERICAL SIMULATION

To illustrate the application of the proposed adaptive control scheme, we consider an example of the control of a two-link planar manipulator in a vertical plane as shown in the Fig. 4.2. The closed-form dynamic equations for the manipulator are given in [5]. The following parameters are assumed for the links [22]: For the first link, mass=15.91kg, length = 0.432m, and for the second link, mass=11.36kg, length=0.432m. The reference trajectory given in [22], i.e.,

$$\theta_1 = 7.8539\ exp\,(-t\,) - 9.4248\ exp\,(-t\,/1.2) \tag{4.8.1}$$

$$\theta_2 = 1.57 + 7.8539\ exp\,(-t\,) - 9.4248\ exp\,(-t\,/1.2) \tag{4.8.2}$$

was generated to move the joints from their initial values of $(-\pi/2,\ 0.0)$ radians at rest, to the final position of $(0.0,\ \pi/2)$ radians at rest.

The simulation was performed on a VAX 11/780 computer with an inner loop sampling rate of 500Hz. The controller gains were updated every 0.1s. The initial values of the controller gains were chosen to be K_p = diag(10,10) and K_v = diag(8,8). The desired performance of the tracking error was specified by a natural frequency (ω_i) of 4 rad/sec and a damping ratio (ζ_i) of 1 .

To illustrate the applicability of the proposed adaptive control given in this chapter, the control law (4.3.15) was first applied without feedback control. Figures (4.3a-b) show the open-loop response of joint 1 and 2. From these figures it is seen that the system is unstable. Then the control law (4.3.15) was applied without the term $f_c(t)$. The response of the system is shown in Figs. (4.4a-b) and it shows that the response is improved, but still unstable. Now, the control law (4.3.15) was applied with $f_c(t)$ obtained from (4.4.15), with $Q = qI$ in the Lyapunov equation (4.4.7) and p_2, , p_3, given by (4.5.6) The value of q was varied from 2500 to 35000. The system response is as shown in Figs (4.5a-b) for several values of q and indicate that the system response becomes stable From these figures it is clear that the system response is

greatly affected by the choice of Q . However, a value of Q which gives good tracking response is not easy to obtain and may not ensure stability in the presence of unmodeled disturbances or parameters variations. A trial and error approach has been used in [38,43] to select a suitable value for Q . A better way of choosing Q has been given in the proposed algorithm presented in this chapter.

To illustrate this approach, the control law (4.3.15) was used where $f_c(t)$ was given by (4.4.15) with p_{2_i} and p_{3_i} given by (4.5.5) with $\omega_i=4$, $\zeta_i=1$, $\lambda_{p_i}=16$ and $\lambda_{v_i}=8$. The coefficients a_{2_i} and a_{3_i} of the desired closed-loop characteristic equation (4.5.4) were then selected to satisfy the inequalities (4.5.9), while ensuring that the roots lie at or close to desired locations in the left-half of the complex plane. The values $a_{2_1}=269500$, $a_{3_1}=132120$ and $a_{2_2}=1360300$, $a_{3_2}=640000$ were obtained. These values ensured that q_{1_i} and q_{2_i} are positive and were used to compute p_{2_i} and p_{3_i} in (4.5.5) which were then updated on-line in terms of the identified parameter h_i . Figures (4.6a-b) show the behavior of $f_m(t)$ i.e. the mismatch between f_c and f_a . The system response using the proposed control law is given in Figs. (4.7a-b) and can be seen to be significantly better than the corresponding responses in Fig. (4.5a-b). The estimated values of the diagonal entries of the matrix H are shown in Figs. (4.8a-b) and indicate that the assumption that the manipulator parameters are slowly time-varying is justified. The variations of the gains $(k_{p_i}$, $k_{v_i})$ are shown in Figs. (4.9a-b) and Figs. (4.10a-b) .

In order to examine the adaptability of the control system, it was assumed that the payload changes suddenly at $t=2$ second, such that the effective mass of the second link is changed from 11.36kg to 15kg. With this change the adaptive controller still gives good performance. Figures (4.11a-b) show the response of the joint angles under this situation and it is seen that good tracking of the reference trajectory is achieved in spite of the change in the payload. The small oscillations in the responses result from the step change in the payload.

From the results of Section 4.8, it is evident that the application of the method given in this chapter results in a stable system and gives good tracking performance in the presence of unmodeled dynamics and payload variations. However, it should be noted that the accuracy of the method depends on the selections of the desired closed-loop characteristic polynomial.

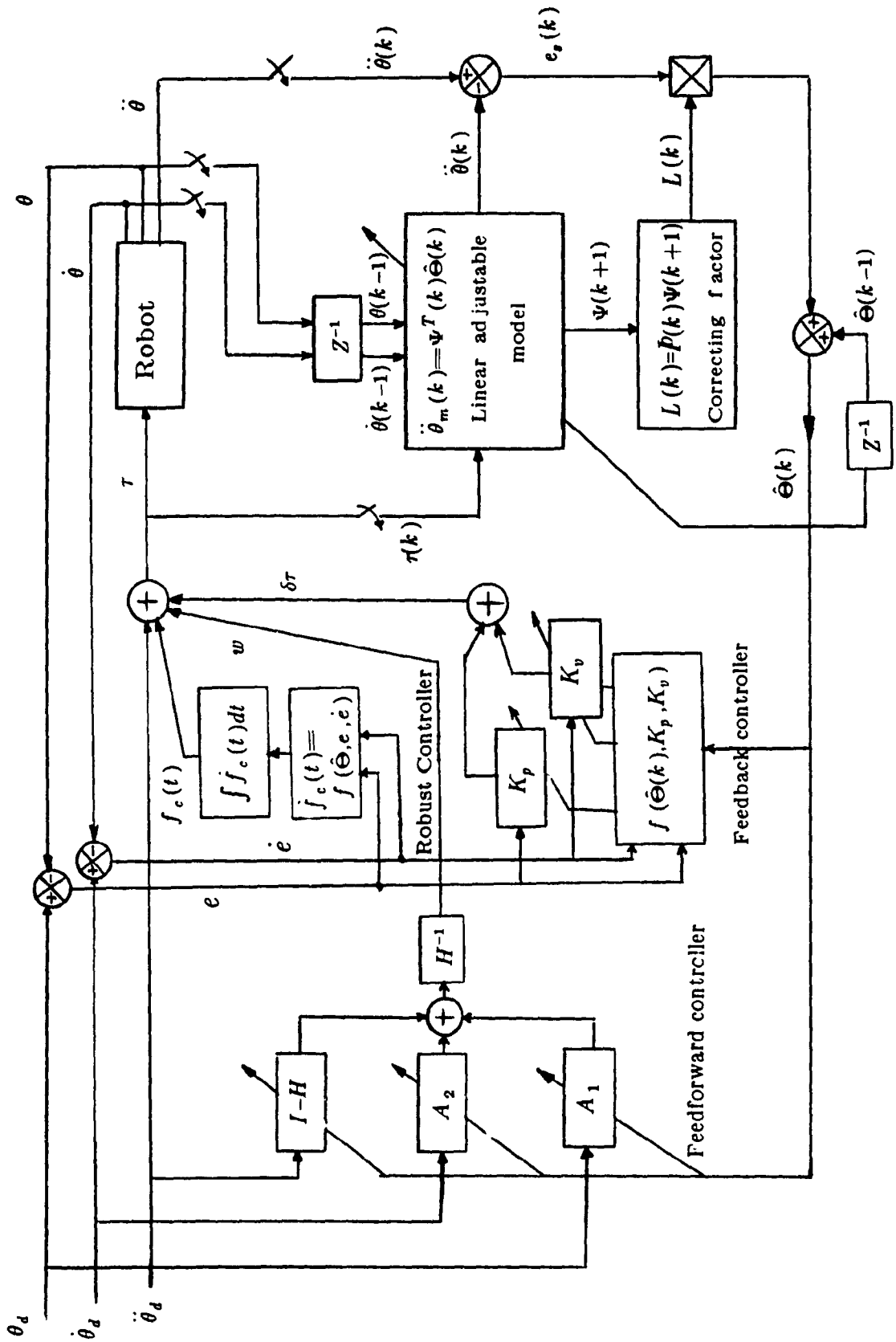

Fig.(4.1) Block diagram of the proposed adaptive control scheme

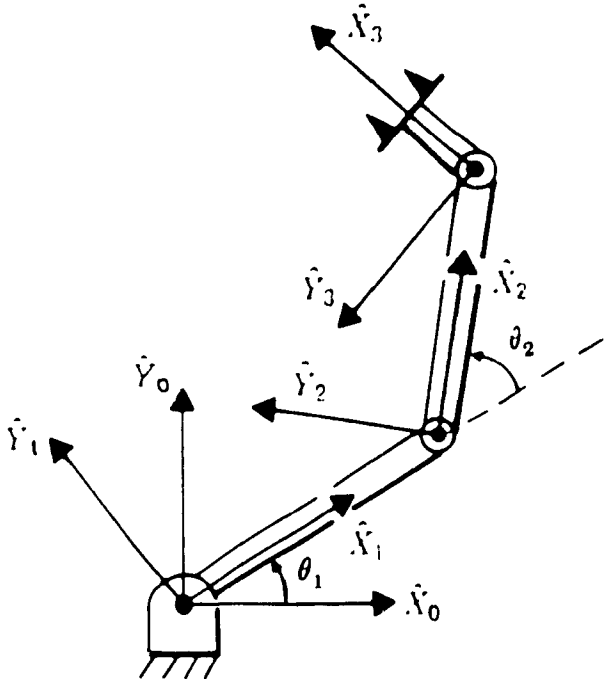

Fig.(4.2) Two link planar manipulator in vertical plane

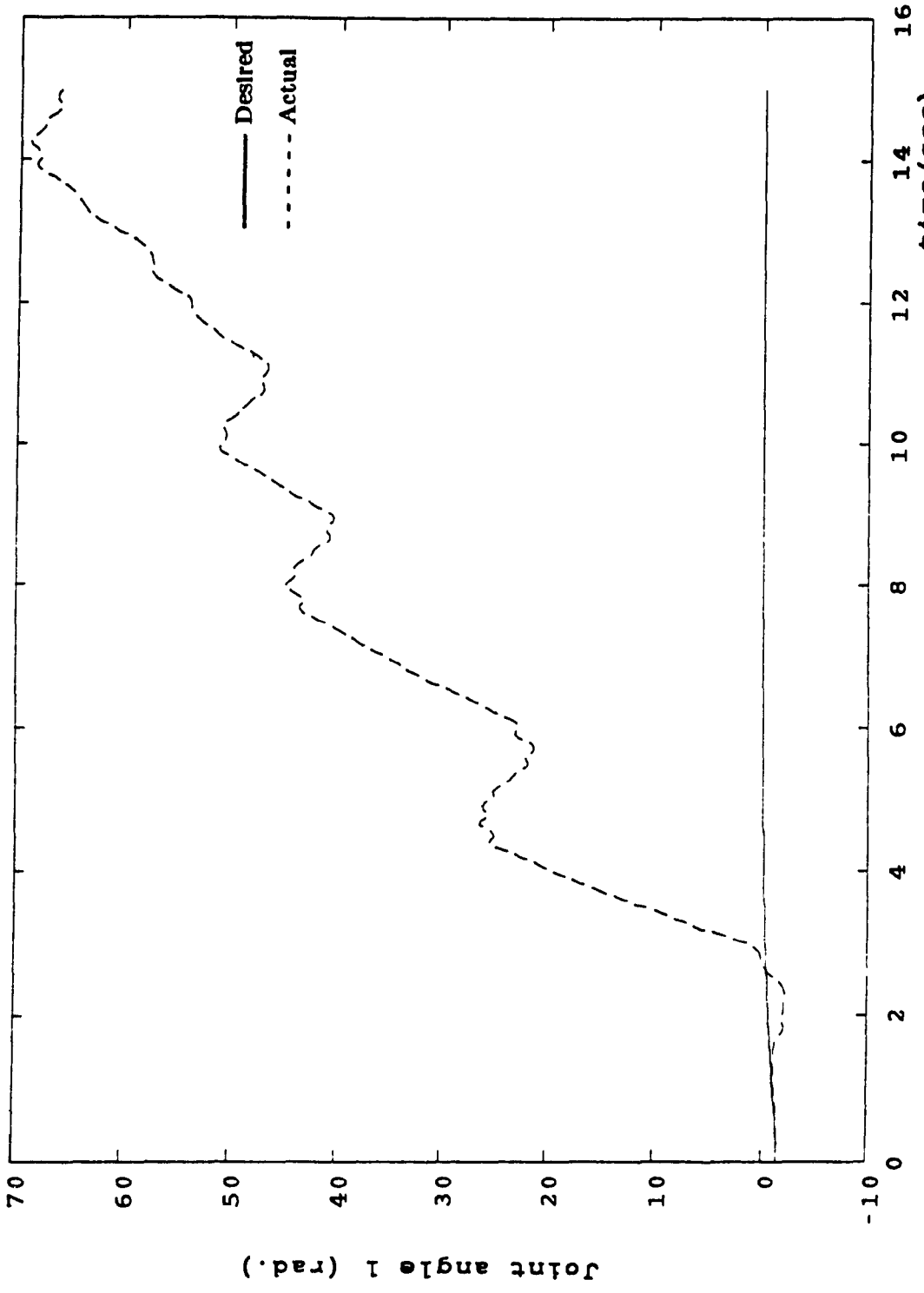

Fig.(4.3a) Open-loop response of joint 1

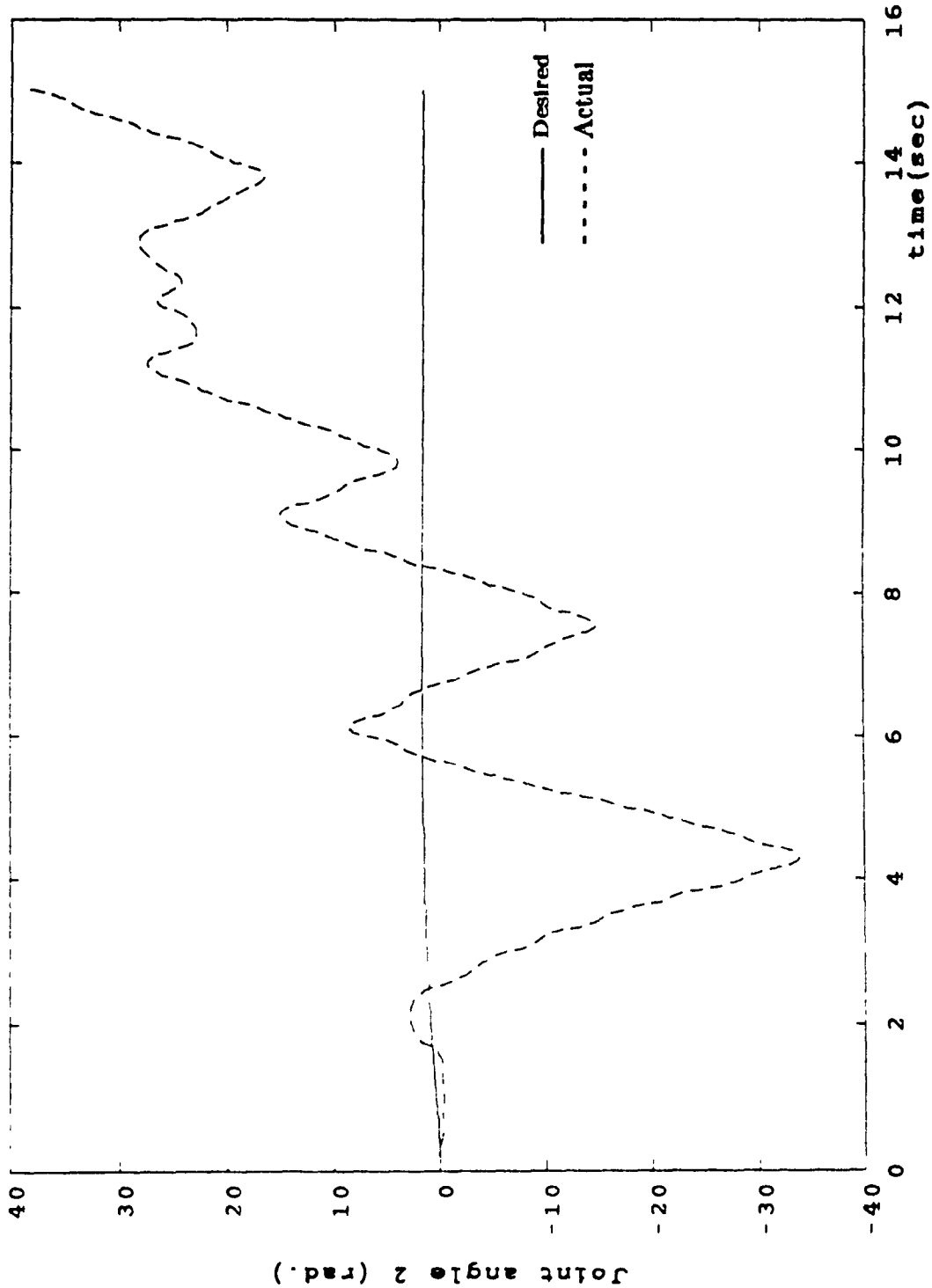

Fig.(4.3b) Open-loop response of joint 2

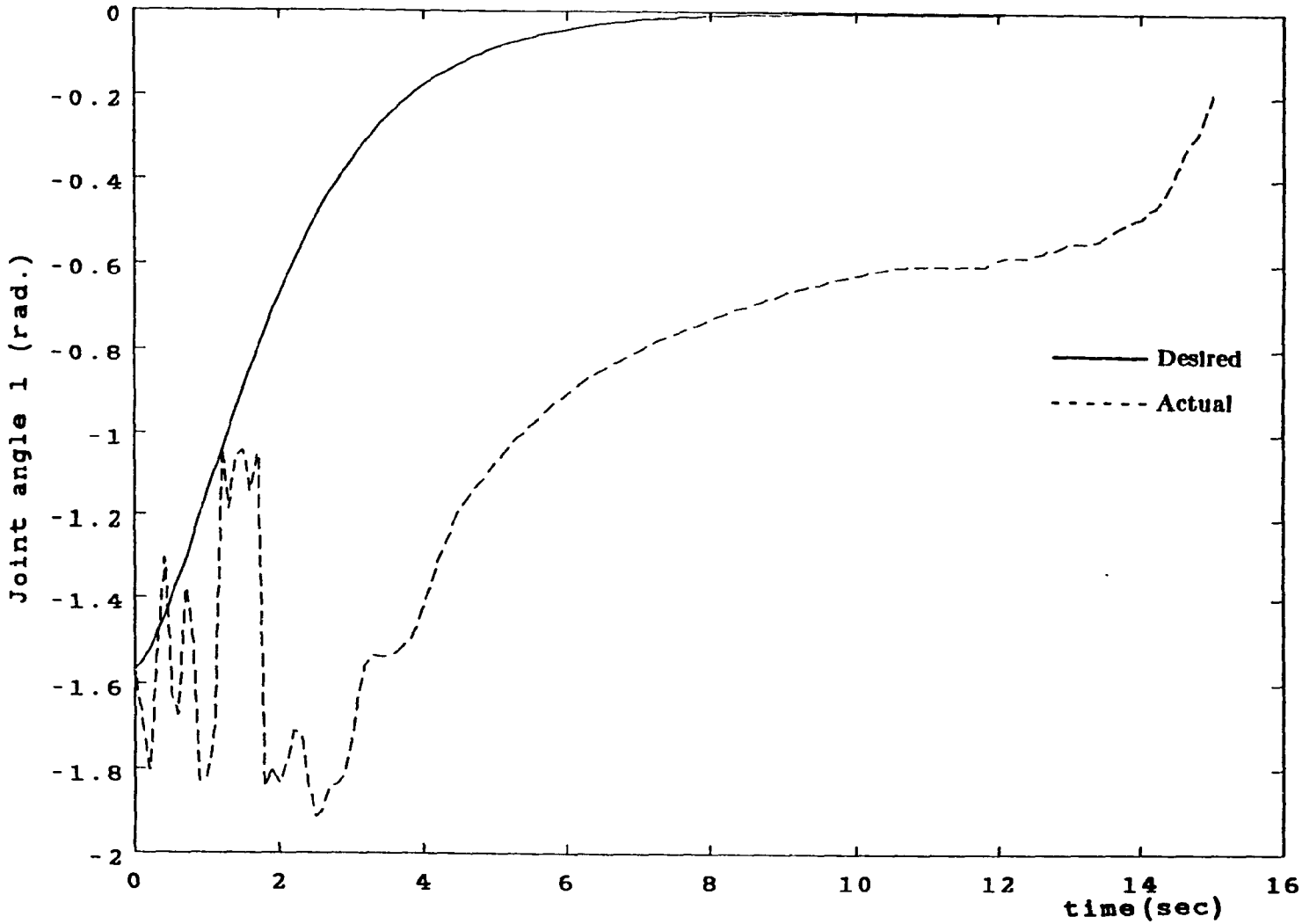

Fig.(4.4a) Closed-loop response of joint 1 without $f_c(t)$

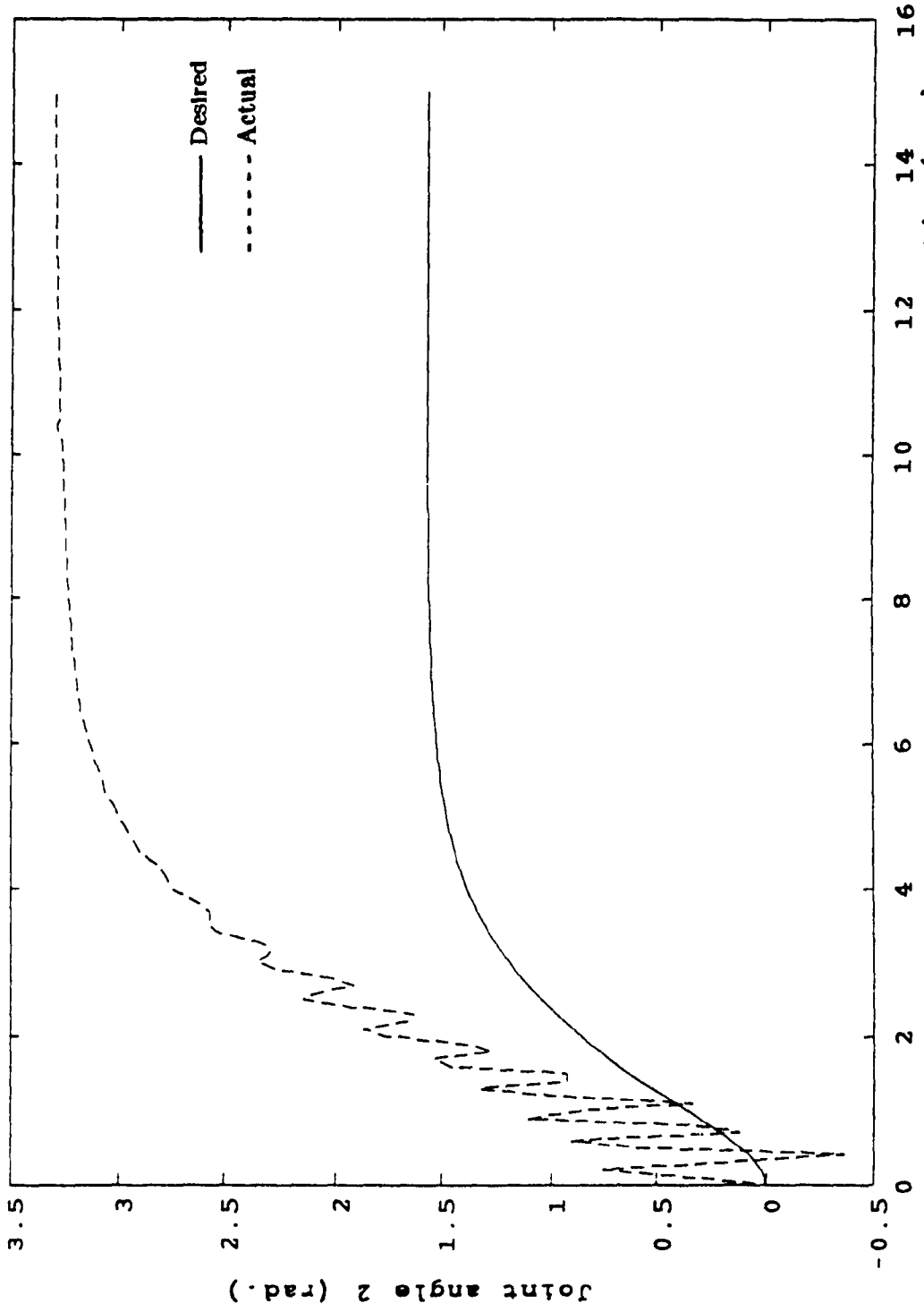

Fig.(4.4b) Closed-loop response of joint 2 without $f_c(t)$

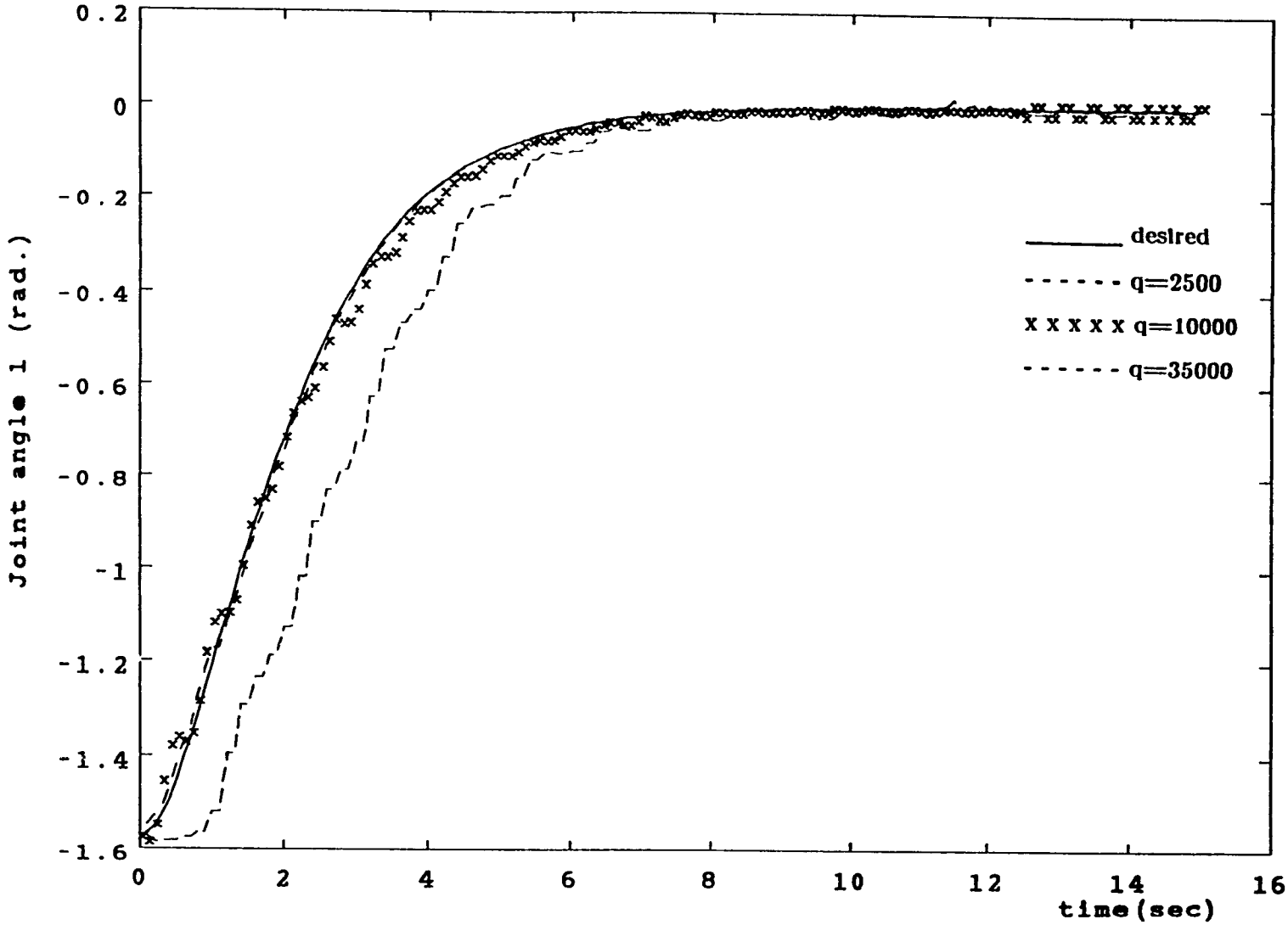

Fig.(4.5a) Closed-loop response of joint 1 for several values of q

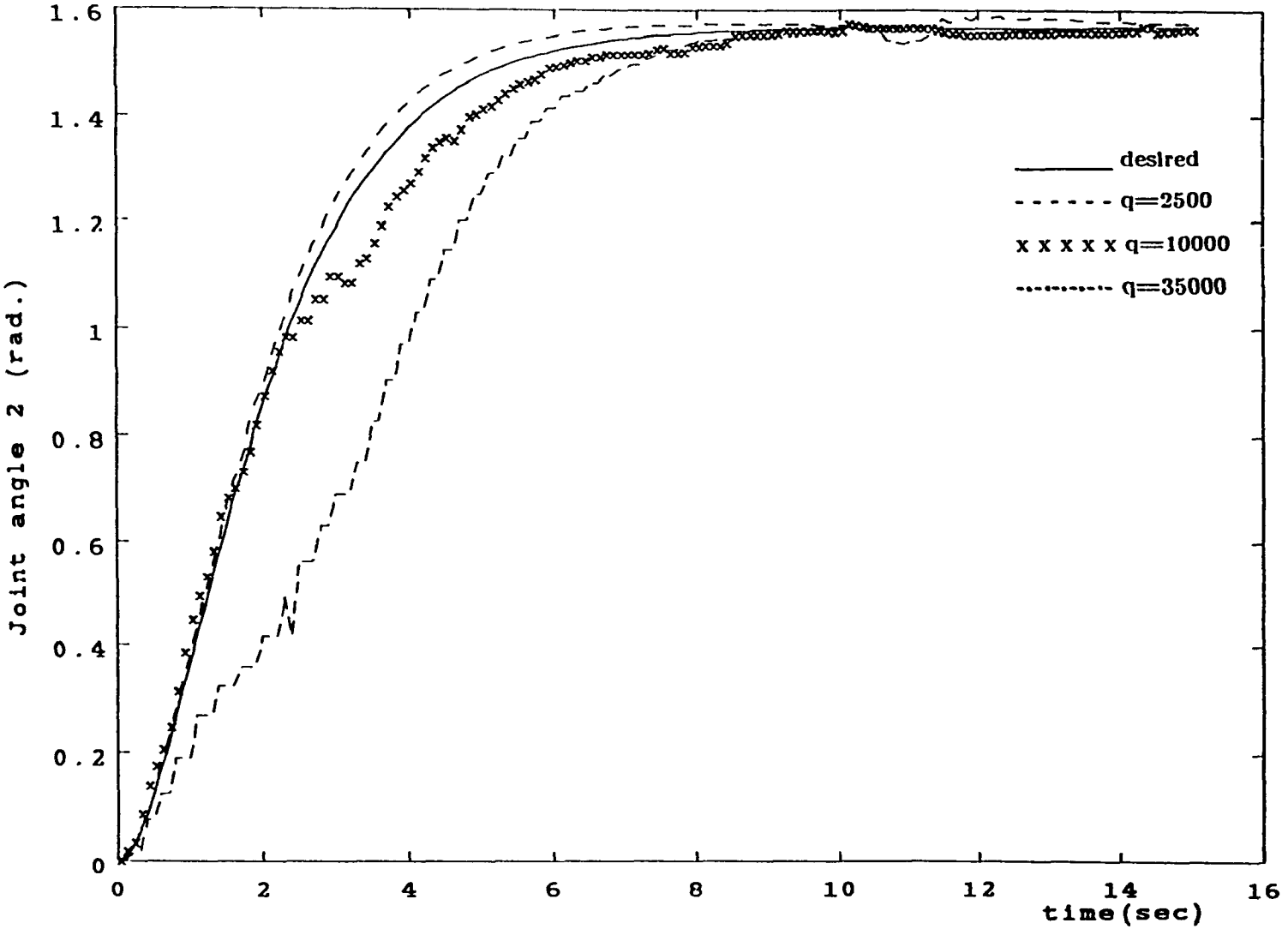

Fig.(4.5b) Closed-loop response of joint 2 for several values of q

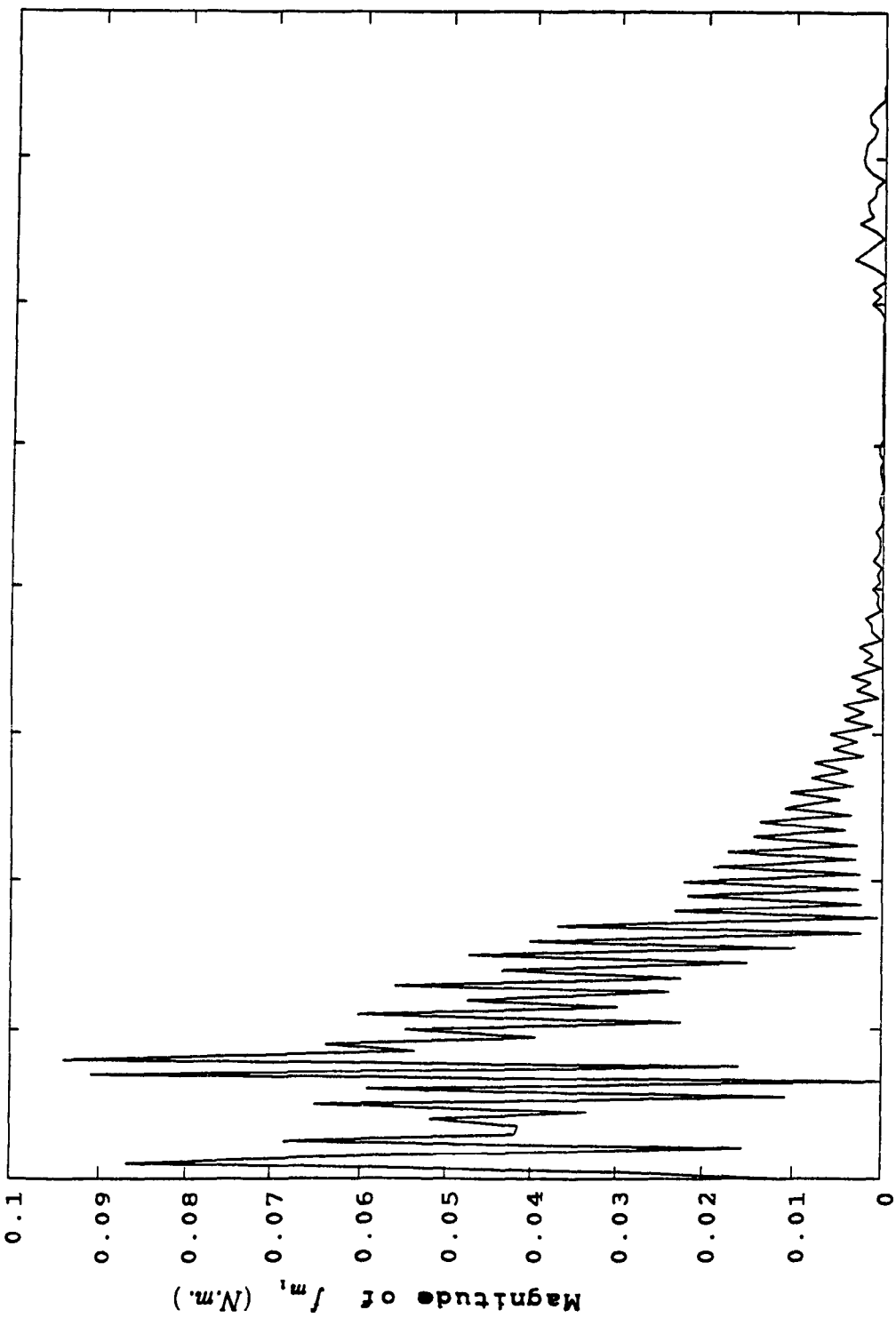

Fig.(4.6a) Behaviour of f_{m_1} for the closed-loop system

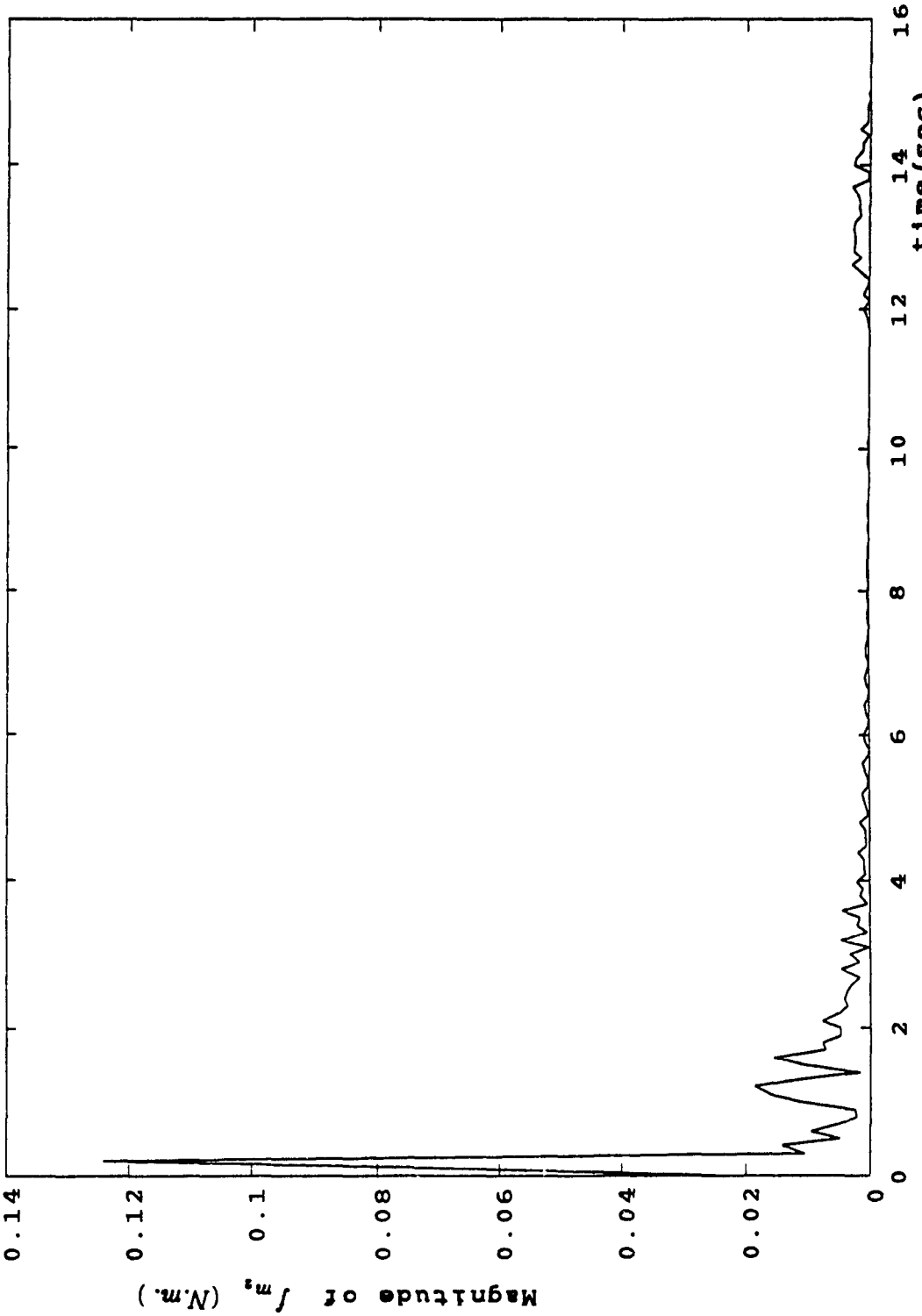

Fig.(4.6b) Behaviour of f_{m_2} for the closed-loop system

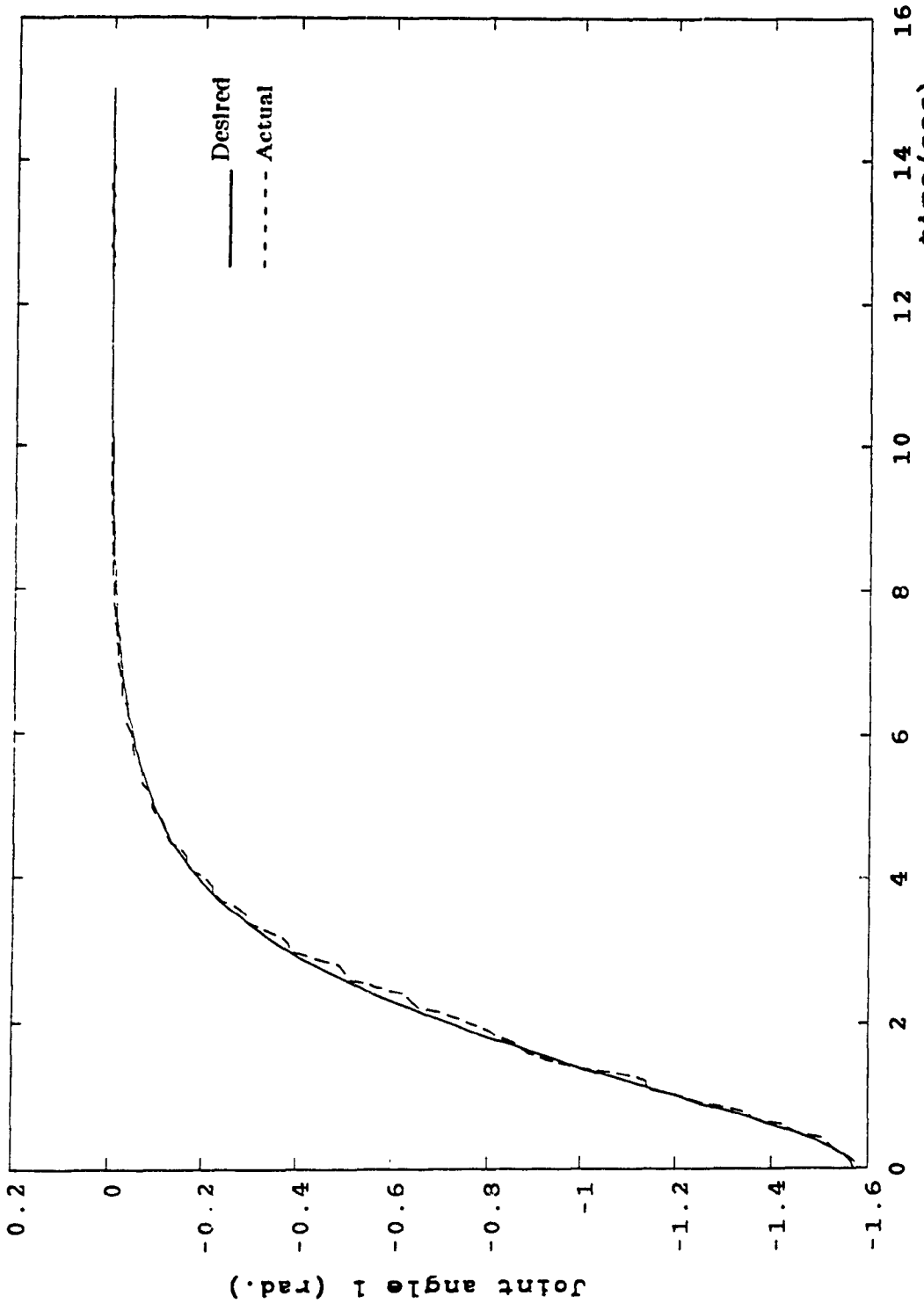

Fig.(4.7a) Closed-loop response of joint 1

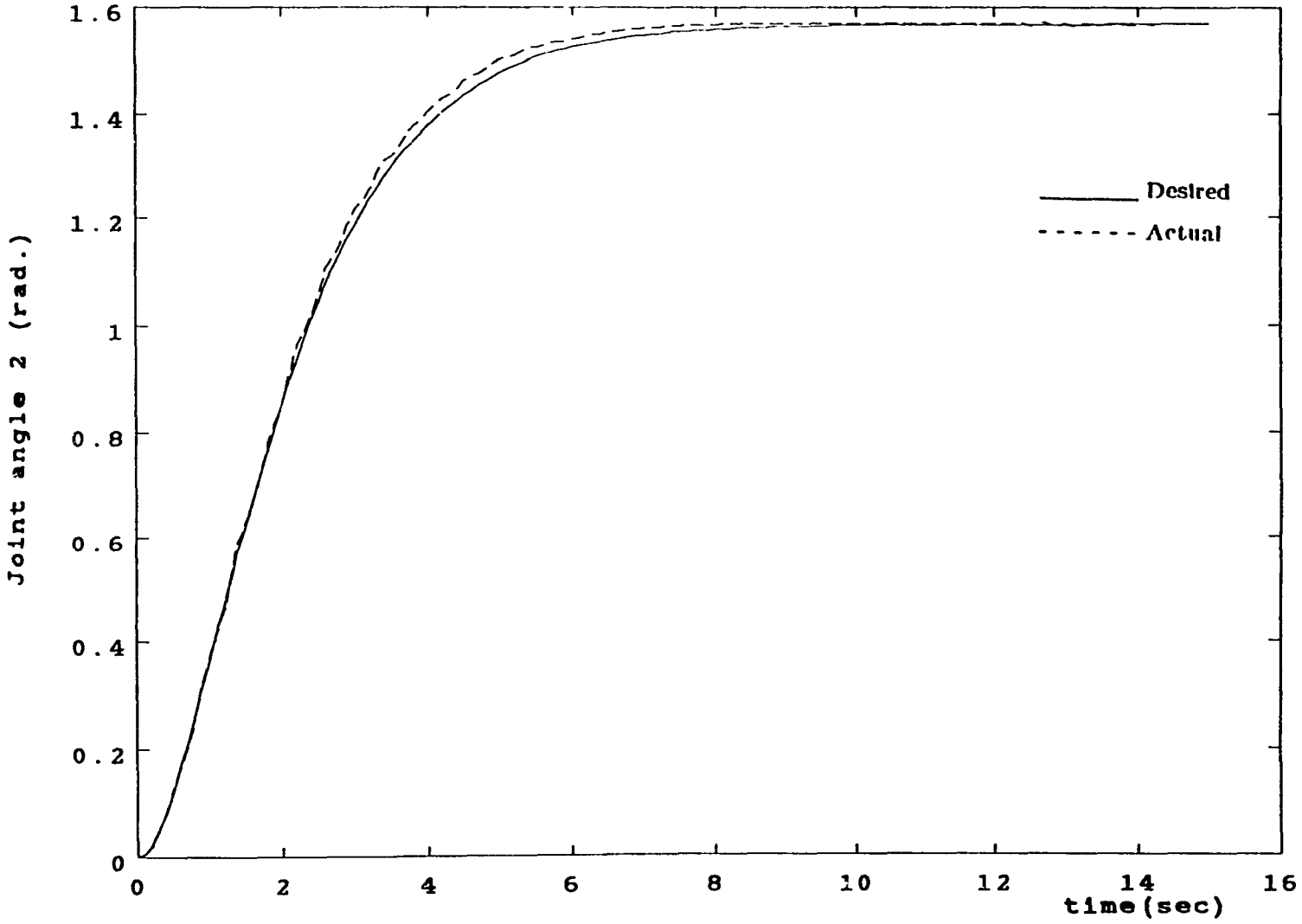

Fig.(4.7b) Closed-loop response of joint 2

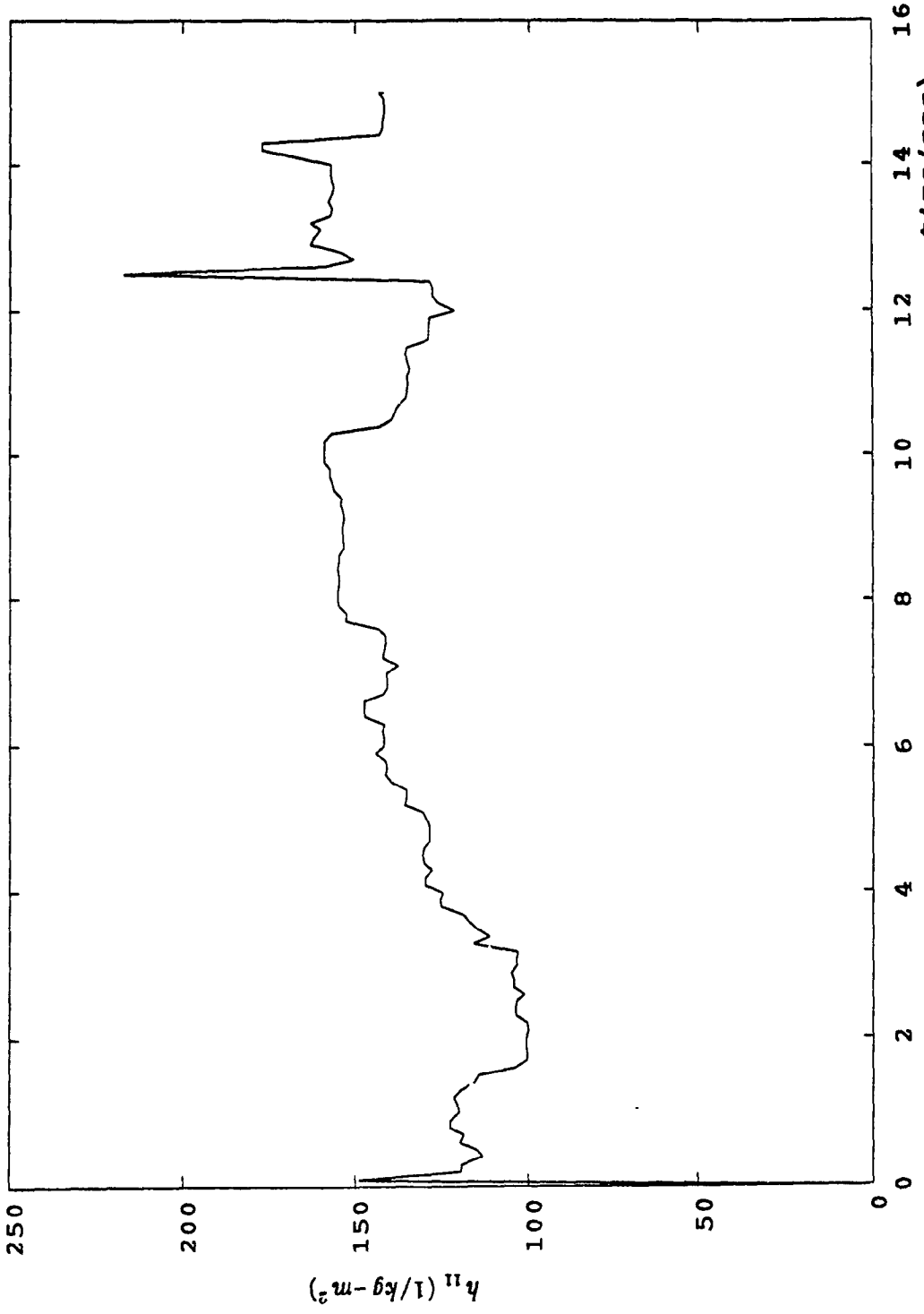

Fig.(4.8a) Estimation of the first diagonal element of h

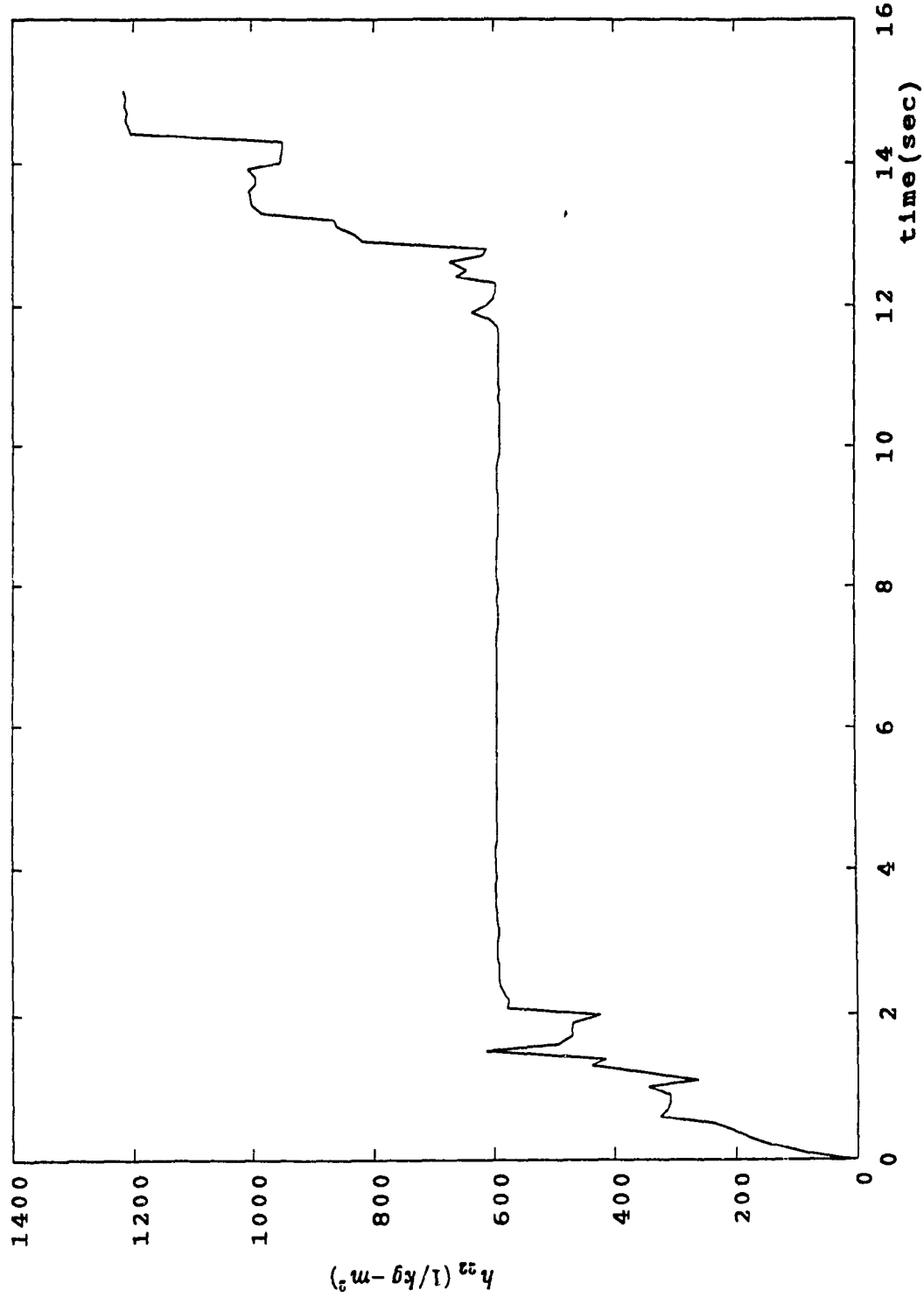

Fig.(4.8b) Estimation of the second diagonal element of H

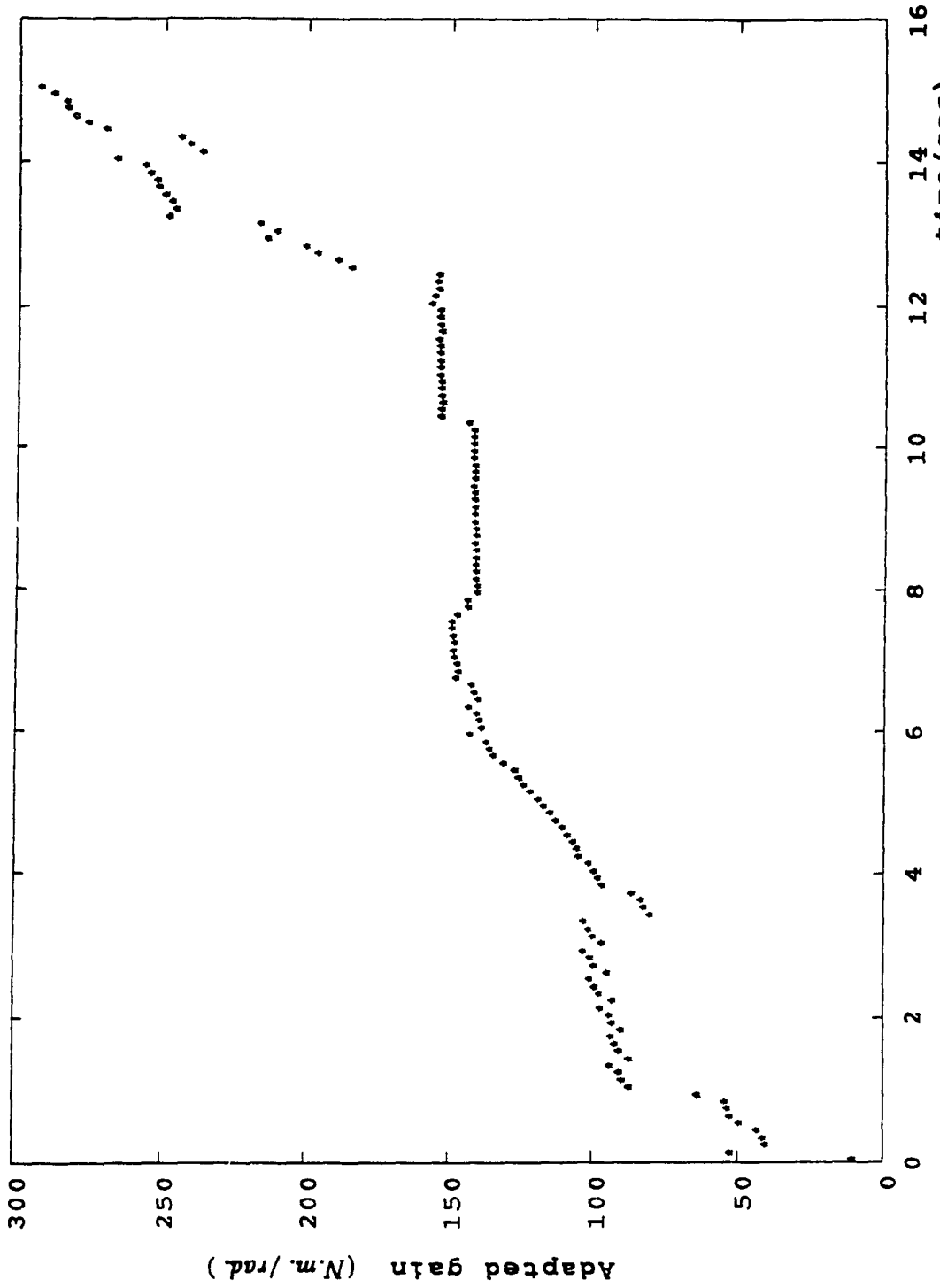

Fig.(4.9a) Adapted gain (K_{p_1}) for joint 1

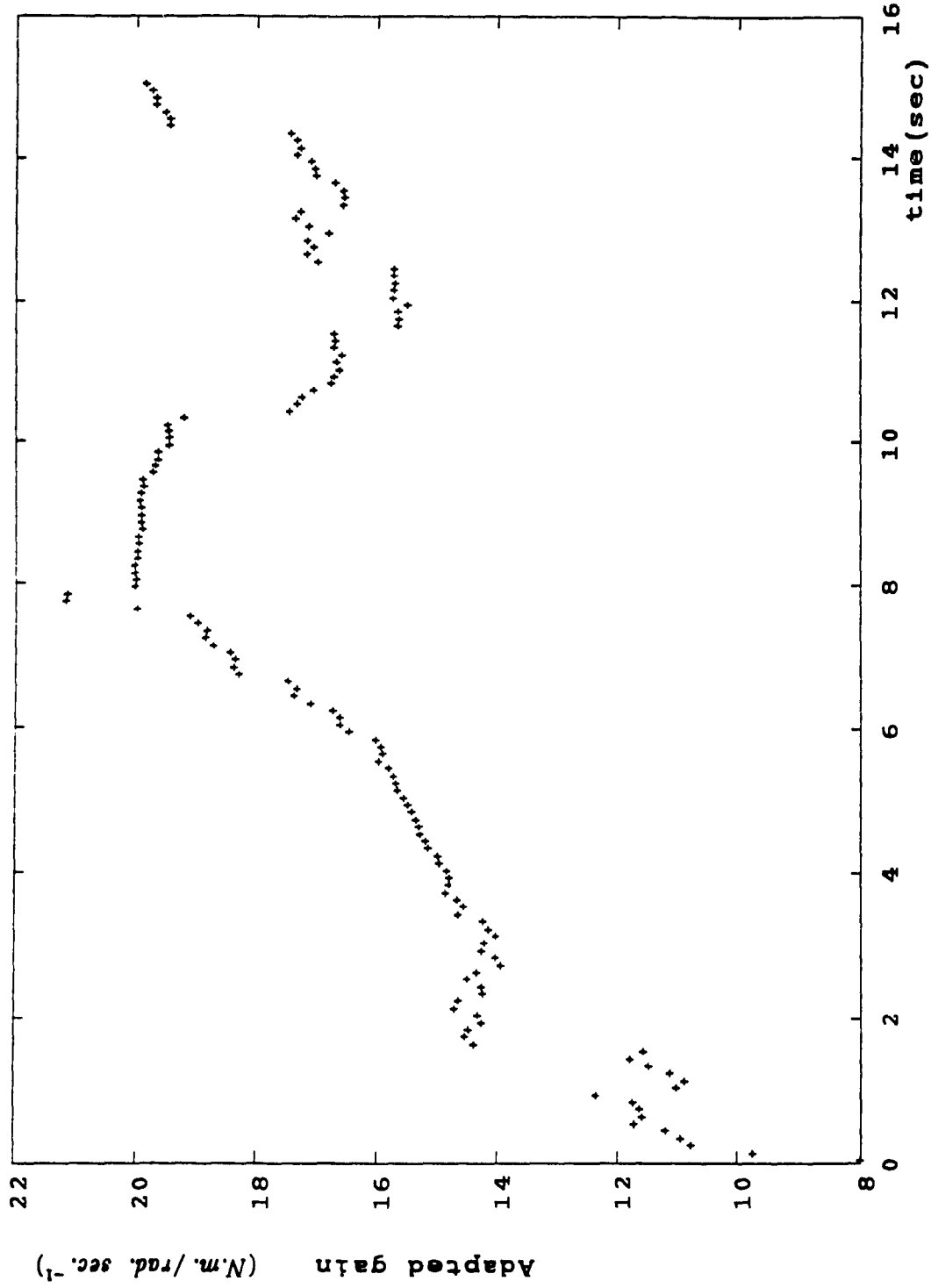

Fig.(4.9b) Adapted gain (K_{v_1}) for joint 1

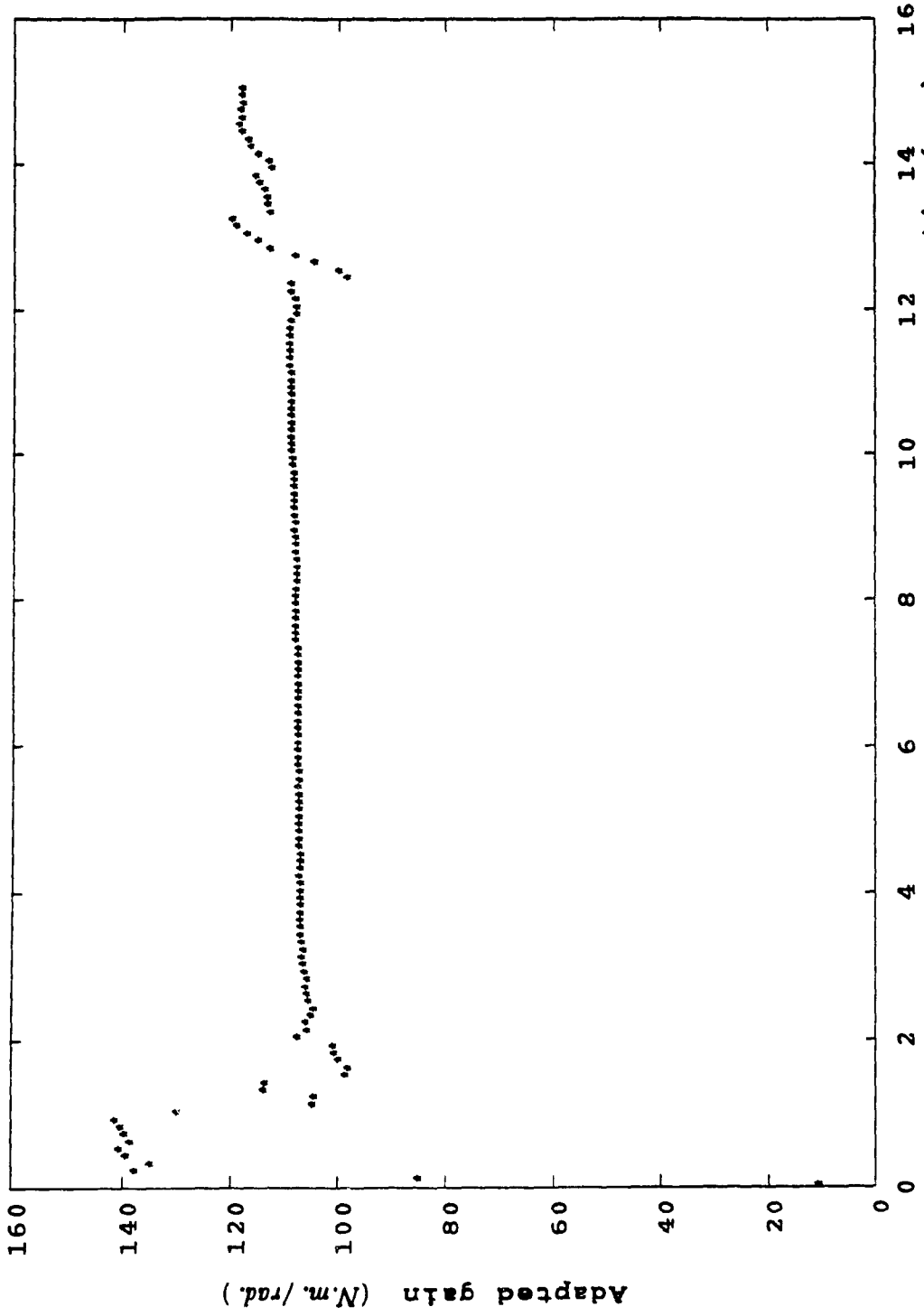

Fig.(4.10a) Adapted gain (K_{p_2}) for joint 2

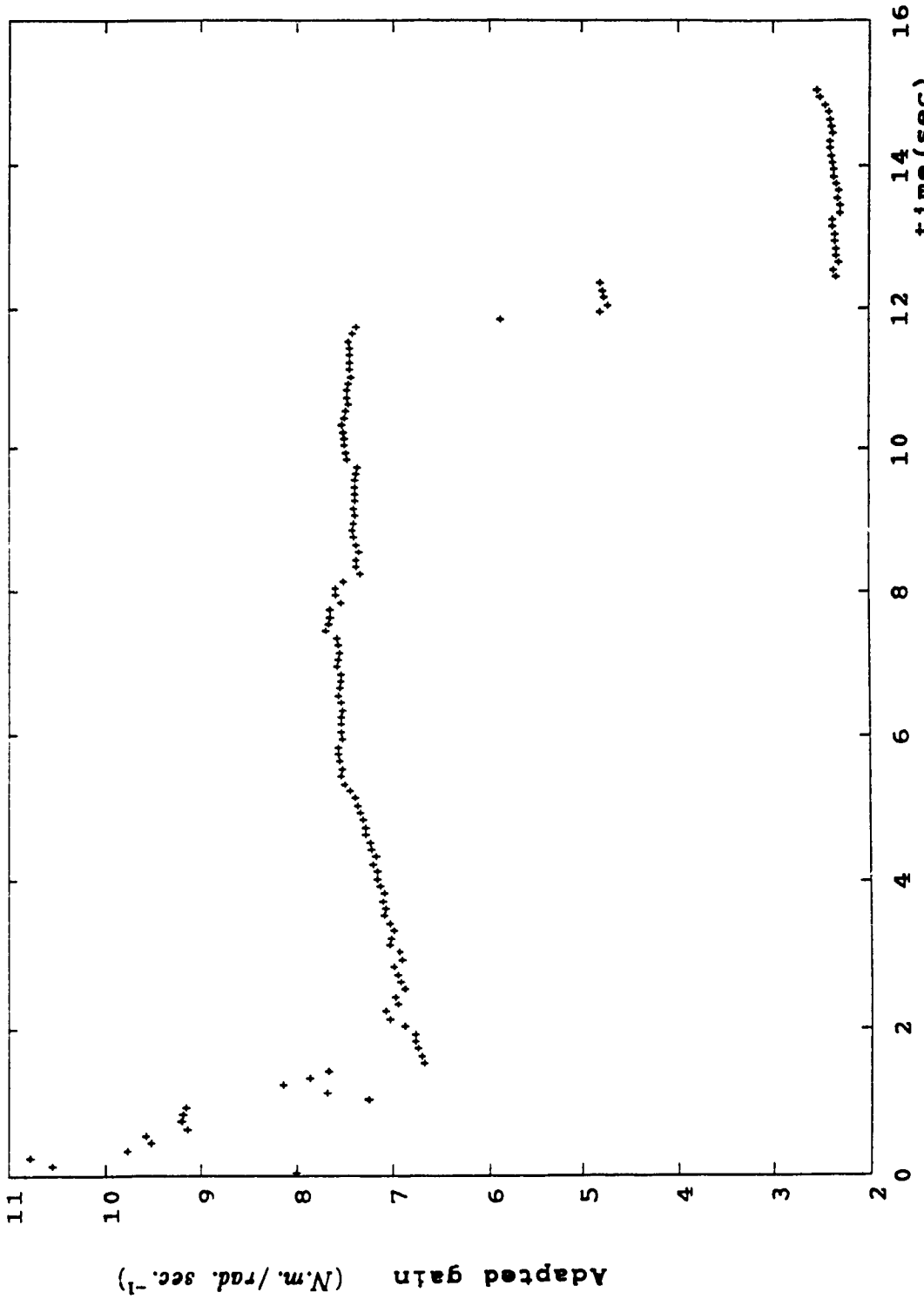

Fig.(4.10b) Adapted gain (K_{v_2}) for joint 2

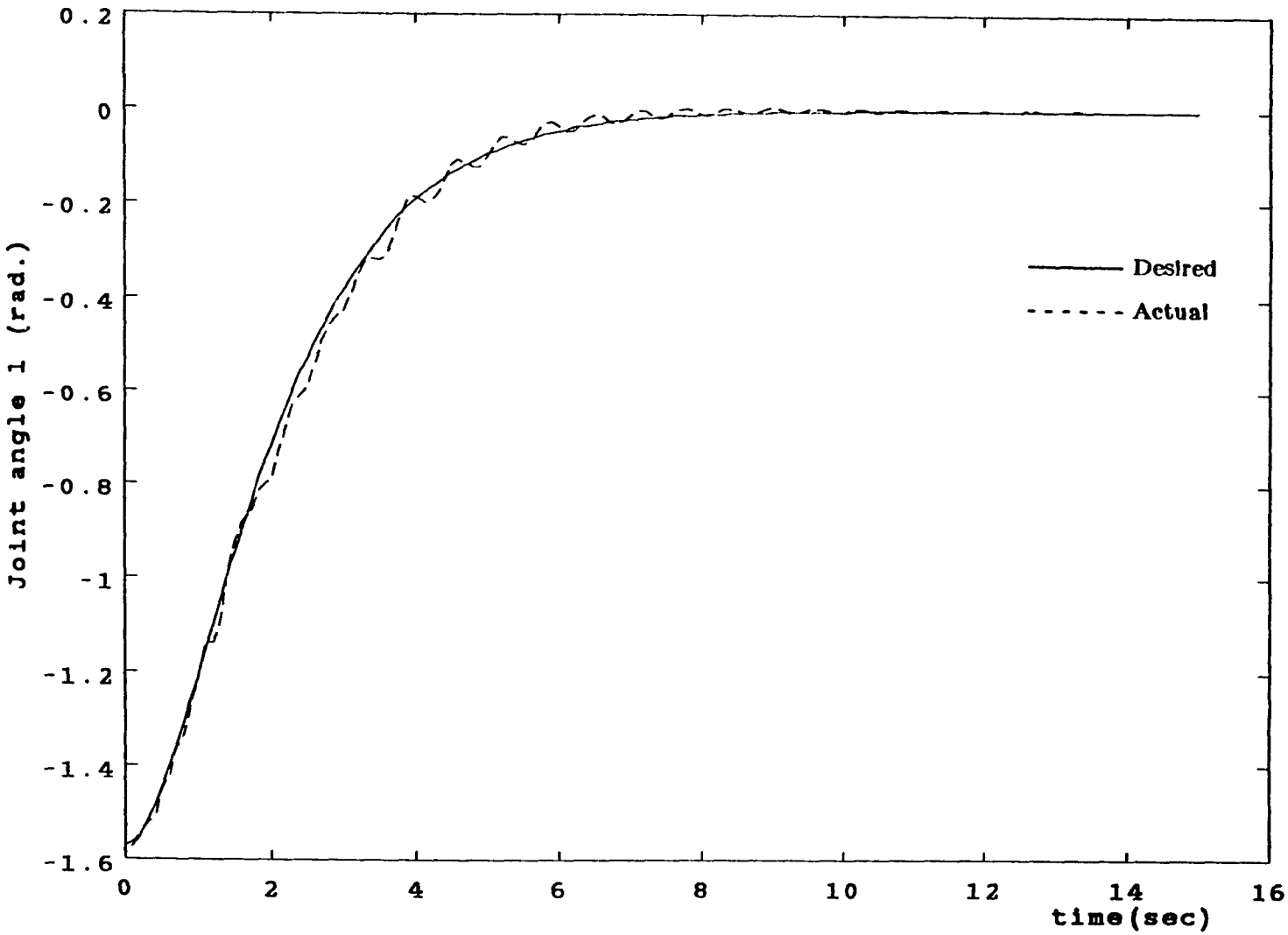

Fig.(4.11a) Closed-loop response of joint 1 under payload change

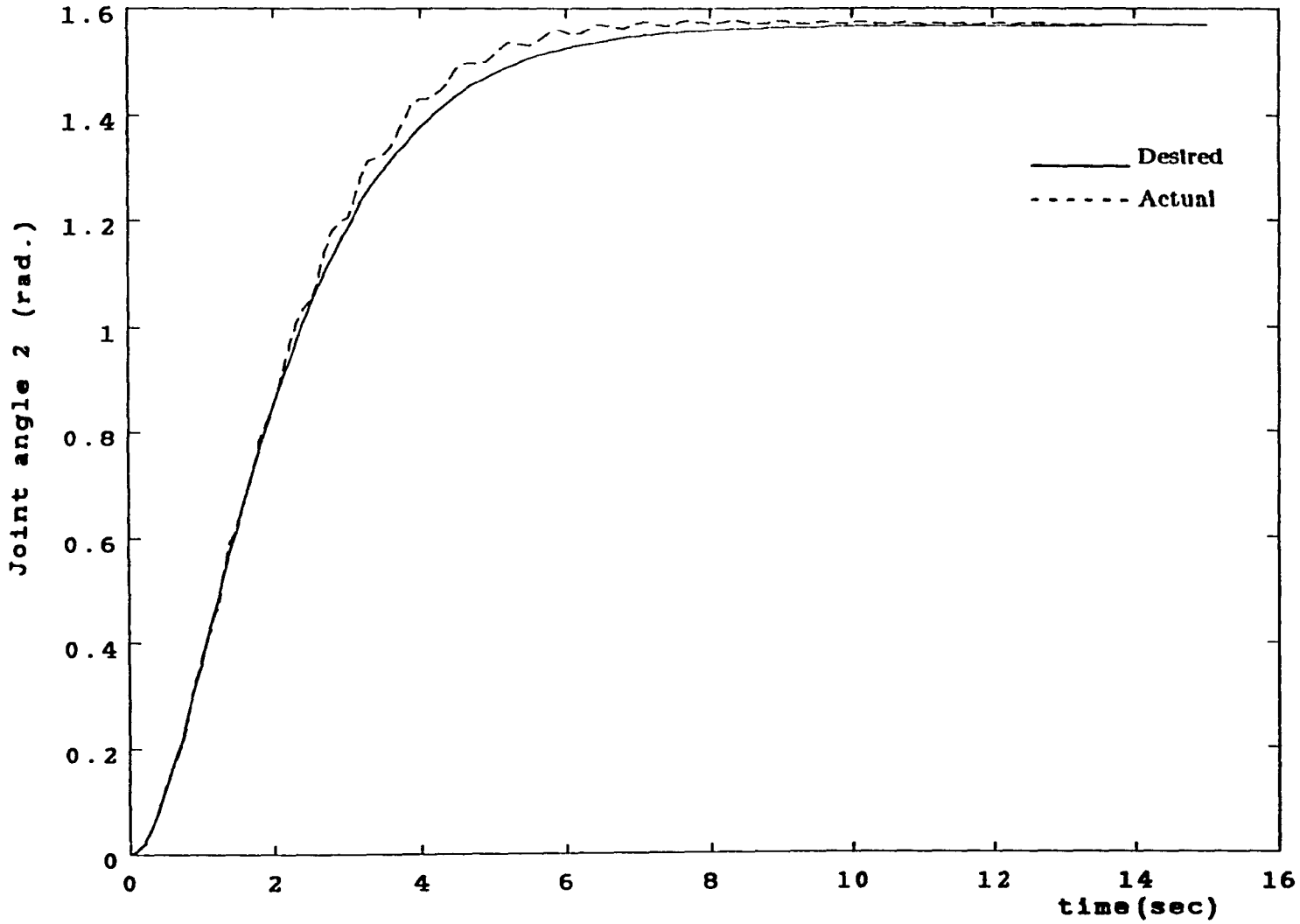

Fig.(4.11b) Closed-loop response of joint 2 under payload change

CHAPTER FIVE

A BOUNDED PERTURBATION APPROACH FOR ROBUSTNESS OF THE ADAPTIVE CONTROLLER

5.1 INTRODUCTION

Stability analysis using Lyapunov's stability theorem requires the selection of positive-definite weighting matrices and scalar parameters. This selection is still an issue. It is important because it determines how fast $e(t) \rightarrow 0$ as $t \rightarrow \infty$ and the way in which it does so. In the previous work in adaptive and robust control which used Lyapunov stability theorem, these matrices were selected arbitrarily by the designer [7,21,23,25,32,33,40,45,46,47] or by a trial and error approach [22,43]. Almost always, this selection affects the robustness of the adaptive controller. Arbitrary selection may not result in fast convergence or may result in oscillatory response. The trial and error approach is time consuming and becomes difficult as the manipulator complexity increases.

The vector $f_a(t)$ introduced in the previous chapter represents slowly time-varying unmodeled dynamics, which arise from ignoring the off-diagonal terms in the model matrices and parameter variations, payload changes and environmental disturbances. In Chapter 4 a control function $f_c(t)$ was designed to approximate $f_a(t)$. Then, a method based on the assignment of the coefficients of a cubic characteristic polynomial was developed.

In this chapter, a new technique is developed to compute the term $f_c(t)$ so as to achieve robust control i.e. to maintain trajectory tracking in the presence of a class of parameter uncertainties and payload variations. A simple iterative technique is used to select the control parameters on-line to ensure that the mismatch $f_m(t)$ is bounded for

all t of interest. This chapter contains four sections: In Section 2, the design of the robust controller is given. Section 3 discusses the method of obtaining the control parameters on-line. To show the capability of the proposed robust adaptive controller, an application to a two-link planar robot manipulator in the vertical plane is considered in Section 4.

5.2 Robust Controller Design

In Chapter 4, we obtained the dynamics of the actual closed-loop system as

$$\dot{z}_{a_i} = \Pi_i \; z_{a_i} + \beta_i \; f_{m_i} \qquad i=1,\ldots,n \tag{5.2.1}$$

where $f_{m_i} = f_{a_i} - f_{c_i}$. We have assumed that $\dot{f}_{a_i} \approx 0$, but this is not necessarily true in reality. There will always exist an error $f_{m_i}(t)$. It is therefore important to keep the magnitude of $f_{m_i}(t)$ as small as possible to maintain stability of the overall closed-loop system. Since (from (4.4.10)) the value of $f_{m_i}(t)$ is affected by the matrix P_i, we can use the positive-definite matrix Q_i in (4.4.7) to ensure (if possible) that the magnitude of $f_{m_i}(t)$ remains small. This choice is important because it determines how $z_{a_i}(t) \rightarrow 0$. Therefore our aim is to choose a positive-definite matrix Q_i such that the nonlinear term $f_{m_i}(t)$ in (5.2.1) is bounded.

The unmodeled dynamics in the closed-loop system (5.2.1) are bounded and kept below a specified level if the nonlinear term $f_{m_i}(t)$ is bounded as

$$\frac{||\beta_i f_{m_i}||}{||z_{a_i}||} \leq \mu_i$$

Since $||\beta_i|| = h_i$ and $f_{m_i}(t)$ is a scalar, we can write the above inequality as

$$\frac{h_i |f_{m_i}|}{||z_{a_i}||} \leq \mu_i \tag{5.2.2}$$

where μ_i is some (small) tolerance which can be specified in a number of ways to ensure that (5.2.1) remains asymptotically stable, eg. see [37,41]. If the result of [37] is used,

then

$$\mu_i = \frac{\min \lambda(Q_i)}{\max \lambda(P_i)} \tag{5.2.3}$$

where, P_i and Q_i are positive-definite matrices satisfying (4.4.7). Such a choice of Q_i will ensure that the effect of unmodeled dynamics is kept below a specified level in the closed-loop system. To see this let us choose Q_i as:

$$Q_i = \begin{bmatrix} Q_{1_i} & 0 \\ 0 & Q_{2_i} \end{bmatrix}$$

Then in (4.4.7), different choices of Q_{1_i} and Q_{2_i} will generate different solutions for P_i. To take advantage of the decoupled nature of the problem that arises from using independent joint control and thereby simplify computations, we choose:

$$Q_{1_i} = g_i \lambda_{p_i}{}^2 \lambda_{v_i} \quad \text{and} \quad Q_{2_i} = (s_i - g_i) \lambda_{p_i} \lambda_{v_i} \qquad s_i > g_i > 0 \tag{5.2.4}$$

Then the solution of the Lyapunov equation (4.4.7) for P_i is given by

$$P_i = \begin{bmatrix} g_i \lambda_{p_i} \lambda_{v_i}{}^2 + s_i \lambda_{p_i}{}^2 & g_i \lambda_{p_i} \lambda_{v_i} \\ g_i \lambda_{p_i} \lambda_{v_i} & s_i \lambda_{p_i} \end{bmatrix}$$

Therefore using (4.4.14), we obtain the function $f_{c_i}(t)$ from

$$\dot{f}_{c_i}(t) = \frac{h_i}{q_{o_i}} \left[g_i \lambda_{p_i} \lambda_{v_i} \, e_{a_i} + s_i \lambda_{p_i} \, \dot{e}_{a_i} \right] \tag{5.2.5}$$

i.e.

$$f_{c_i}(t) = f_{c_i}(t_o) + \frac{h_i}{q_{o_i}} \left[g_i \lambda_p \lambda_{v_i} \int_{t_o}^{t} e_{a_i}(t) dt + s_i \lambda_{p_i} \, e_{a_i} \right] \tag{5.2.6}$$

where $f_{c_i}(t_0)$ denotes the initial value of $f_{c_i}(t)$. It should be noted that $f_{c_i}(t)$ depends on the choice of the matrix Q_i, i.e. on the selection of g_i and s_i. Therefore, this selection will also affect the value of $f_{m_i}(t)$.

Applying the control law (4.3.15) (in the decoupled form with $f_{c_i}(t)$ given by (5.2.6)) to the manipulator whose dynamics are given by (4.3.14) yields the closed-loop error equation

$$\ddot{e}_{a_i} + \left[a_{2_i} + h_i k_{v_i}\right]\dot{e}_{a_i} + \left[a_{1_i} + h_i k_{p_i}\right]e_{a_i} = h_i\left[f_{a_i}(t) - f_{c_i}(t)\right] = h_i f_{m_i}(t)$$

which can be written as

$$h_i f_{m_i}(t) = \ddot{e}_{a_i} + \lambda_{v_i}\dot{e}_{a_i} + \lambda_{p_i} e_{a_i} \qquad i=1,\ldots,n \qquad (5.2.7)$$

The value of $h_i f_{m_i}(t)$ which represents the mismatch in the closed-loop system due to the unmodeled dynamics can be determined from (5.2.7) if the position, velocity and acceleration errors e_{a_i}, $\dot{e}_{a_i}$ and $\ddot{e}_{a_i}$ are known. We wish to find values of λ_{p_i} and λ_{v_i} in terms of the two independent parameters g_i and s_i such that f_{m_i} is bounded as

$$|f_{m_i}(t)| \leq \frac{\mu_i}{h_i}||z_{a_i}|| \qquad (5.2.8)$$

where, $z_{a_i} = \begin{bmatrix} e_{a_i} & \dot{e}_{a_i} \end{bmatrix}^T$, $\mu_i = \alpha\mu_{i\ \max}$, $0 < \alpha \leq 1$ and $\mu_{i\ \max}$ is the largest value of μ_i in (5.2.2) such that the closed-loop system is asymptotically stable. The inequality (5.2.8) is satisfied by appropriate choices of g_i and s_i. For example, we can let $Q_{1_i} = Q_{2_i}$ and get

$$\lambda_{p_i} = \frac{s_i - g_i}{g_i} \qquad i=1,\ldots,n \qquad (5.2.9)$$

Choosing the damping factor $\zeta_i = 1$, we have $\lambda_{v_i} = 2\sqrt{\lambda_{p_i}}$, and hence (5.2.7) can be written as

$$h_i f_{m_i}(t) = \ddot{e}_{a_i} + 2\sqrt{\frac{s_i - g_i}{g_i}}\,\dot{e}_{a_i} + \frac{s_i - g_i}{g_i}\,e_{a_i} \qquad (5.2.10)$$

From (5.2.10), g_i and s_i can be determine on-line such that $f_{m_i}(t)$ is bounded and satisfies (5.2.8). These values of s_i and q_i are then used to update the robust control term $f_{c_i}(t)$ which can be written as

$$\dot{f}_{c_i}(t) = \frac{h_i}{q_{0_i}} \left[2\frac{(s_i - g_i)^{\frac{3}{2}}}{\sqrt{g_i}} e_{a_i} + \frac{s_i (s_i - g_i)}{g_i} \dot{e}_{a_i} \right] \tag{5.2.11}$$

5.3 EVALUATION OF $f_{m_i}(t_{j+1})$

In this section we will show how s_i and g_i can be determined to satisfy (5.2.8). In the following analysis, for the sake of notational clearity, we will drop the index i since the same analysis is applied to all the links, and we will denote the time instant t_j and t_{j+1} by the arguments j and $j+1$, respectively.

At the sampling time t_j, the errors $e_a(j)$, $\dot{e}_a(j)$ and $\ddot{e}_a(j)$ are measured, and according to these measurement, a new model $\hat{M}_j$ is identified. Also at this sampling instant $f_m(j)$ is computed using $e_a(j)$, $\dot{e}_a(j)$, $\ddot{e}_a(j)$ and $s(j)$, $g(j)$. The parameters $s(j)$ and $g(j)$ are computed during the preceding time interval Δt_{j-1}. The feedback gains k_p, k_v are updated in terms of the identified parameters a_1, a_2 and h

In order to move the system from position $\theta(j)$ to a new position $\theta(j+1)$, a new control input $\tau(j+1)$ needs to be computed during the interval Δt_j in terms of the measurements at t_j The control input $\tau(j+1)$ can be represented as

$$\tau(j+1) = \tau_0(j+1) + f_c(j+1) \tag{5.3.1}$$

where $\tau_0(j+1)$ is given by

$$\tau_0(j+1) = h^{-1}\left[\ddot{\theta}_d(j+1) + a_2\dot{\theta}_d(j+1) + a_1\theta_d(j+1)\right] + k_v e_a(j) + k_p e_a(j) \tag{5.3.2}$$

As shown in (5.3.2), $\tau_0(j+1)$ can be computed in terms of the reference trajectory, the identified parameters and the measured errors at t_j. The term $f_c(j+1)$ can be found from (5.2.11) by integration. If Euler integration is used, we get

$$\begin{aligned} f_c(j+1) &= f_c(j) + \Delta t_j \, \dot{f}_c(j+1) \\ &= f_c(j) + 2\Delta t_j \frac{[s(j+1)\, g(j+1)]^{\frac{3}{2}}}{\sqrt{g(j+1)}} e_a(j) \end{aligned}$$

$$+\Delta t_j \frac{s(j+1)\left[s(j+1)-g(j+1)\right]}{g(j+1)}\,\dot{e}_a(j) \tag{5.3.3}$$

In (5.3.3) the term $f_c(j)$ is computed during the interval Δt_{j-1} and the errors $e_a(j)$ and $\dot{e}_a(j)$ are measured at t_j. Then $f_c(j+1)$ can be changed by choosing the parameters $s(j+1)$ and $g(j+1)$ during the interval Δt_j. By adding (5.3.2) and (5.3.3), we get the total torque input at t_{j+1}:

$$\begin{aligned}\tau(j+1) = h^{-1}\left[\ddot{\theta}_d(j+1)+a_2\dot{\theta}_d(j+1)+a_1\theta_d(j+1)\right]+k_v\,\dot{e}_a(j)+k_p\,e_a(j)\\ +f_c(j)+2\Delta t_j\frac{\left[s(j+1)-g(j+1)\right]^{\frac{3}{2}}}{\sqrt{g(j+1)}}\,e_a(j)\\ +\Delta t_j\,\frac{s(j+1)\left[s(j+1)-g(j+1)\right]}{g(j+1)}\,\dot{e}_a(j)\end{aligned} \tag{5.3.4}$$

By applying the control input $\tau(j+1)$ to the manipulator at t_{j+1}, we get the following error equation:

$$\begin{aligned}h\,f_m(j+1) = \ddot{e}_a(j+1)+2\sqrt{\frac{s(j+1)-g(j+1)}{g(j+1)}}\,e_a(j+1)\\ +\frac{s(j+1)-g(j+1)}{g(j+1)}\,e_a(j+1)\end{aligned} \tag{5.3.5}$$

Our aim is to determine suitable values for $s(j+1)$ and $g(j+1)$ such that $f_m(j+1)$ is bounded and satisfies (5.2.8). In order to compute $f_m(j+1)$ during the interval Δt_j, we replace $\ddot{e}_a(j+1)$, $\dot{e}_a(j+1)$, $e_a(j+1)$ by their predicted values $\hat{\ddot{e}}_a$, $\hat{\dot{e}}_a$, $\hat{e}_a$ using the identified model $\hat{M}_j$. Now, to find the values of the predicted errors in terms of the previous measurements at t_j, we apply the torque input (5.3.4) to the model $\hat{M}_j$ and get

$$\ddot{\theta}(j+1) = -a_1\,\theta(j) - a_2\,\dot{\theta}(j) + h\,\tau(j+1) \tag{5.3.6}$$

The error between the desired input and the output of $\hat{M}_j$ is

$$\begin{aligned}\ddot{e}_a(j+1) &= \ddot{\theta}_d(j+1) - \ddot{\theta}(j+1)\\ &= \ddot{\theta}_d(j+1) + a_1\,\theta(j) + a_2\,\dot{\theta}(j) - h\,\tau(j+1)\end{aligned} \tag{5.3.7}$$

By substituting for $\tau(j+1)$ from (5.3.4), we get

$$\ddot{e}_a(j+1) = a_1\theta(j)+a_2\dot{\theta}(j)-a_1\theta_d(j+1)-a_2\dot{\theta}_d(j+1)-hk_p\, e_a(j)$$
$$-hk_v\, e_a(j)-hf_c(j)-2h\,\Delta t_j\,\frac{\left[s(j+1)-g(j+1)\right]^{\frac{3}{2}}}{\sqrt{g(j+1)}}\, e_a(j)$$
$$-h\,\Delta t_j\,\frac{s(j+1)\left[s(j+1)-g(j+1)\right]}{g(j+1)}\,\dot{e}_a(j) \qquad (5.3.8)$$

We can write $\theta(j) = \theta_d(j) - e_a(j)$ and $\dot{\theta}(j) = \dot{\theta}_d(j) - \dot{e}_a(j)$ Then (5.3.8) becomes

$$\ddot{e}_a(j+1) = a_1\left[\theta_d(j) - \theta_d(j+1)\right] + a_2\left[\dot{\theta}_d(j) \quad \dot{\theta}_d(j+1)\right]$$
$$\left[a_1 + hk_p + 2h\,\Delta t_j\,\frac{\left[s(j+1)\; g(j+1)\right]^{\frac{3}{2}}}{\sqrt{g(j+1)}}\right] e_a(j)$$
$$\left[a_2+hk_v+h\,\Delta t_j\,\frac{s(j+1)\left[s(j+1)\; g(j+1)\right]}{g(j+1)}\right] e_a(j) - hf_c(j) \qquad (5.3.9)$$

Denoting

$$a_1\left[\theta_d(j) - \theta_d(j+1)\right] + a_2\left[\theta_d(j) \quad \theta_d(j+1)\right] - hf_c(j) = \Phi(j+1)$$

$$-\left[a_1 + hk_p + 2h\,\Delta t_j\,\frac{\left[s(j+1)\; g(j+1)\right]^{\frac{3}{2}}}{\sqrt{g(j+1)}}\right] = \gamma_1(j+1) \qquad (5.3.10)$$

$$-\left[a_2 + hk_v + h\,\Delta t_j\,\frac{s(j+1)\left[s(j+1)\; g(j+1)\right]}{g(j+1)}\right] = \gamma_2(j+1)$$

we can write (5.3.9) as

$$\dot{e}_a(j+1) = \Phi(j+1) + \gamma_1(j+1)\, e_a(j) + \gamma_2(j+1)\, e_a(j) \qquad (5.3.11)$$

Using Euler integration, we can find $\dot{e}_a(j+1)$ and $e_a(j+1)$ as

$$\dot{e}_a(j+1) = e_a(j) + \Delta t_j\,\Phi(j+1) + \Delta t_j\,\gamma_1(j+1)e_a(j) + \Delta t_j\,\gamma_2(j+1)e_a(j) \qquad (5.3.12)$$

and

$$\hat{e}_a(j+1) = +\frac{3}{2}\Delta t_j^2\,\Phi(j+1) + \left[1+\frac{3}{2}\Delta t_j^2\,\gamma_1(j+1)\right] e_a(j) \qquad (5.3.13)$$
$$+ \left[\Delta t_j + \frac{3}{2}\Delta t_j^2\,\gamma_2(j+1)\right] \dot{e}_a(j)$$

From this, we can see that $\ddot{\hat{e}}_a$, $\dot{\hat{e}}_a$, $\hat{e}_a$ are functions of $[g(j+1),\ s(j+1),\ \dot{e}_a(j)\ ,\ e_a(j)]$ within the interval Δt_j . Since $e_a(j)$ and $\dot{e}_a(j)$ are assumed to be constant during this interval, $\ddot{\hat{e}}_a(j+1)$, $\dot{\hat{e}}_a(j+1)$ and $\hat{e}_a(j+1)$ change when $s(j+1)$ and $g(j+1)$ are changed. We now replace the actual error values in (5.3.5) by the predicted ones to get the predicted value of the mismatch $\hat{f}_m(j+1)$:

$$h\hat{f}_m(j+1) = \ddot{\hat{e}}_a(j+1) + 2\sqrt{\frac{s(j+1)-g(j+1)}{g(j+1)}}\ \dot{\hat{e}}_a(j+1)$$
$$+ \frac{s(j+1)-g(j+1)}{g(j+1)}\ \hat{e}_a(j+1) \qquad (5.3.14)$$

To ensure that Q_1, and Q_2, are positive-definite matrices, we need to have $0 < g < s$. Therefore, we define

$$s = \sigma^2 + \delta^2 \qquad \text{and} \qquad g = \sigma^2$$

Then using (5.3.11), (5.3.12) and (5.3.13), $\hat{f}_m(j+1)$ is given by:

$$h\ \hat{f}_m(j+1) = \left[1+2\Delta t_j\frac{\delta(j+1)}{\sigma(j+1)}+\frac{3}{2}\Delta t_j^2\frac{\delta^2(j+1)}{\sigma^2(j+1)}\right]\Phi(j+1) \qquad (5.3.15)$$
$$+\left[\frac{2\delta(j+1)}{\sigma(j+1)}+\left\{1+2\Delta t_j\frac{\delta(j+1)}{\sigma(j+1)}+\frac{3}{2}\Delta t_j^2\frac{\delta^2(j+1)}{\sigma^2(j+1)}\right\}\gamma_2(j+1)+\Delta t_j\frac{\delta^2(j+1)}{\sigma^2(j+1)}\right]\dot{e}_a(j)$$
$$+\left[\frac{\delta^2(j+1)}{\sigma^2(j+1)}+\left\{1+2\Delta t_j\frac{\delta(j+1)}{\sigma(j+1)}+\frac{3}{2}\Delta t_j^2\frac{\delta^2(j+1)}{\sigma^2(j+1)}\right\}\gamma_1(j+1)\right]e_a(j)$$

As shown above, $\hat{f}_m(j+1)$ is a function of the previous measurements $e_a(j)$ and $\dot{e}_a(j)$, $\delta(j+1)$ and $\sigma(j+1)$. It should be noted that $\gamma_1(j+1)$ and $\gamma_2(j+1)$ are also functions of $\delta(j+1)$ and $\sigma(j+1)$. The term $\hat{f}_m(j+1)$ can be changed during the interval Δt_j by changing the values of $\delta(j+1)$ and $\sigma(j+1)$. Accordingly we can use an iterative technique to obtain the best values for

$\delta(j+1)$ and $\sigma(j+1)$ which make $\hat{f}_m(j+1)$ bounded and satisfy (5.2.8). These values are then used to compute $s(j+1)$ and $g(j+1)$ and update the control term $f_c(j+1)$.

5.4 NUMERICAL SIMULATION

To illustrate the application of the proposed robust adaptive control scheme, we consider the example of the control of a two-link planar manipulator in a vertical plane that was considered in Chapter 4. The reference trajectory is given [22] by

$$\theta_1 = 7.8539exp(-t) - 0.4248exp(-t/1.2)$$

$$\theta_2 = 1.57+7.8539exp(-t) - 0.4248exp(-t/1.2)$$

and was generated to move the joints from their initial values of $(-\pi/2, 0.0)$ radians at rest, to the final position of $(0.0, \pi/2)$ radians at rest.

The simulation was performed on a VAX 11/780 computer with an inner-loop sampling rate of 500Hz. The controller gains were updated every 0.1s. The initial values of the controller gains were chosen to be K_p = diag(10,10) and K_v = diag(8,8). The desired performance of the tracking error was specified by a natural frequency (ω_t) of 4 rad/sec and a damping ratio (ζ_t) of 1.

To illustrate the approach given in this chapter, we used the control law (4.3.15) with $f_{c_1}(t)$ given by (5.2.11). For $\omega_t = 4$, $\zeta_t = 1$ and using (5.2.3), $\mu_{1_{max}}$ was calculated to be 0.3807. Figures (5.1a-b) show the behavior of $f_{m_1}(t)$ i.e. the mismatch between f_{c_1} and f_{a_1}. The system response using the proposed control law is given in Figs. (5.2a-b). It can be seen that the system is stable and the response is significantly better than the corresponding responses in Figs. (4.5a-b) and Figs. (4.7a-b). The estimated values of the diagonal entries of the matrix H (assumed diagonal) are shown in Figs. (5.3a-b) and indicate that the assumption that the manipulator parameters are slowly time-varying is justified. The values of Q are selected on-line so that the corresponding

bounds μ_i change with time. The variations of the bounds μ_i are shown in Fig. (5.4). The variations of the gains (k_{p_i} , k_{v_i}) are shown in Figs. (5.5a-b). In order to examine the adaptibility and robustness of the control system, we assumed that the payload changes suddenly at $t = 4$ second, such that the mass of the second link is changed from 11.36kg to 18.36kg . With this variation the adaptive controller still gives good performance. Figures (5.6a-b) show the response of the joint angles under this situation. It is seen that good tracking of the reference trajectory is achieved in spite of the change in the payload. The system response converged to the desired one within about 1 second of the payload change being applied. This illustrates that the robust controller designed in this chapter is able to compensate effectively against sudden variation in the payload

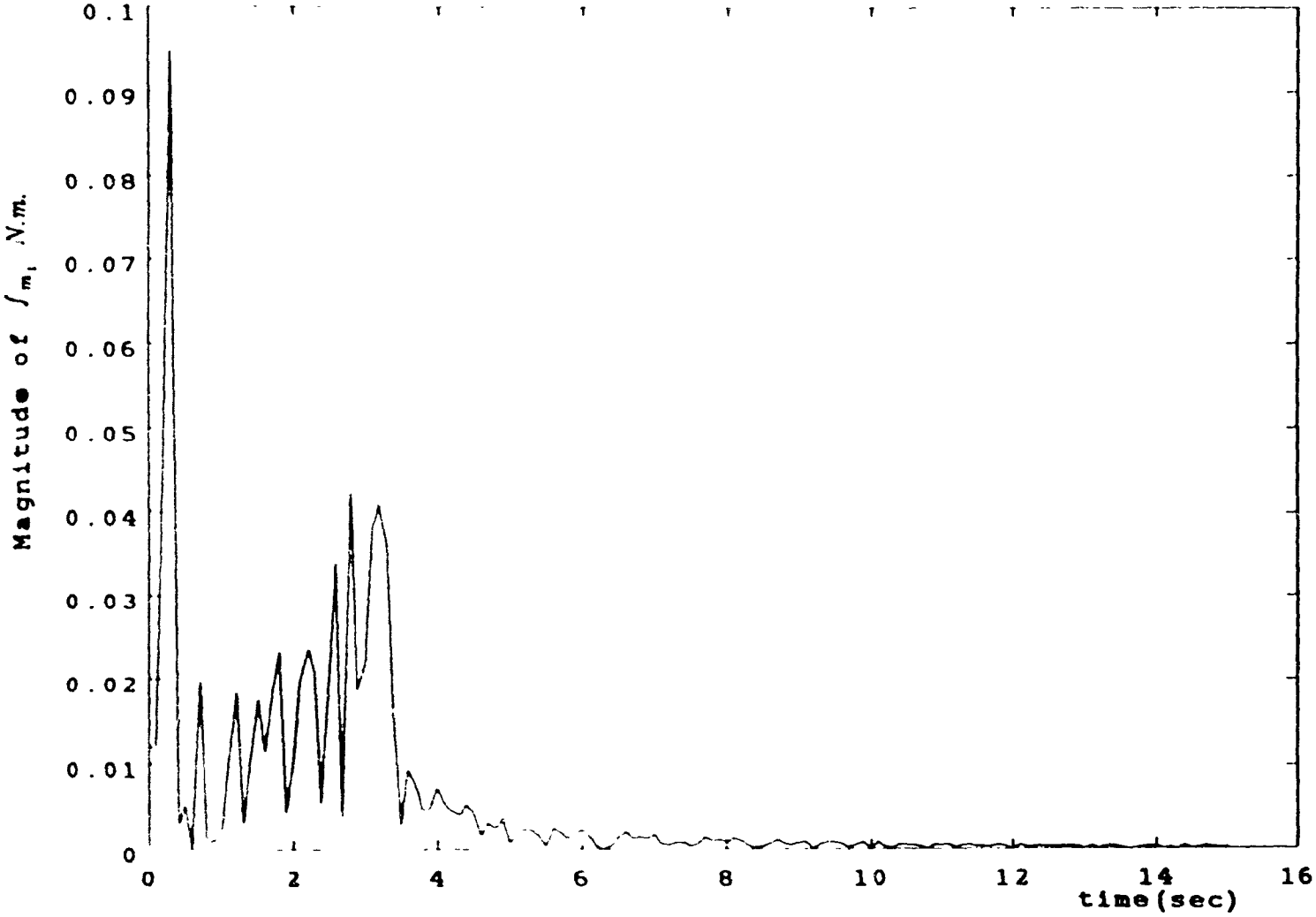

Fig.(5.1a) Behaviour of f_{m_1} for the closed-loop system

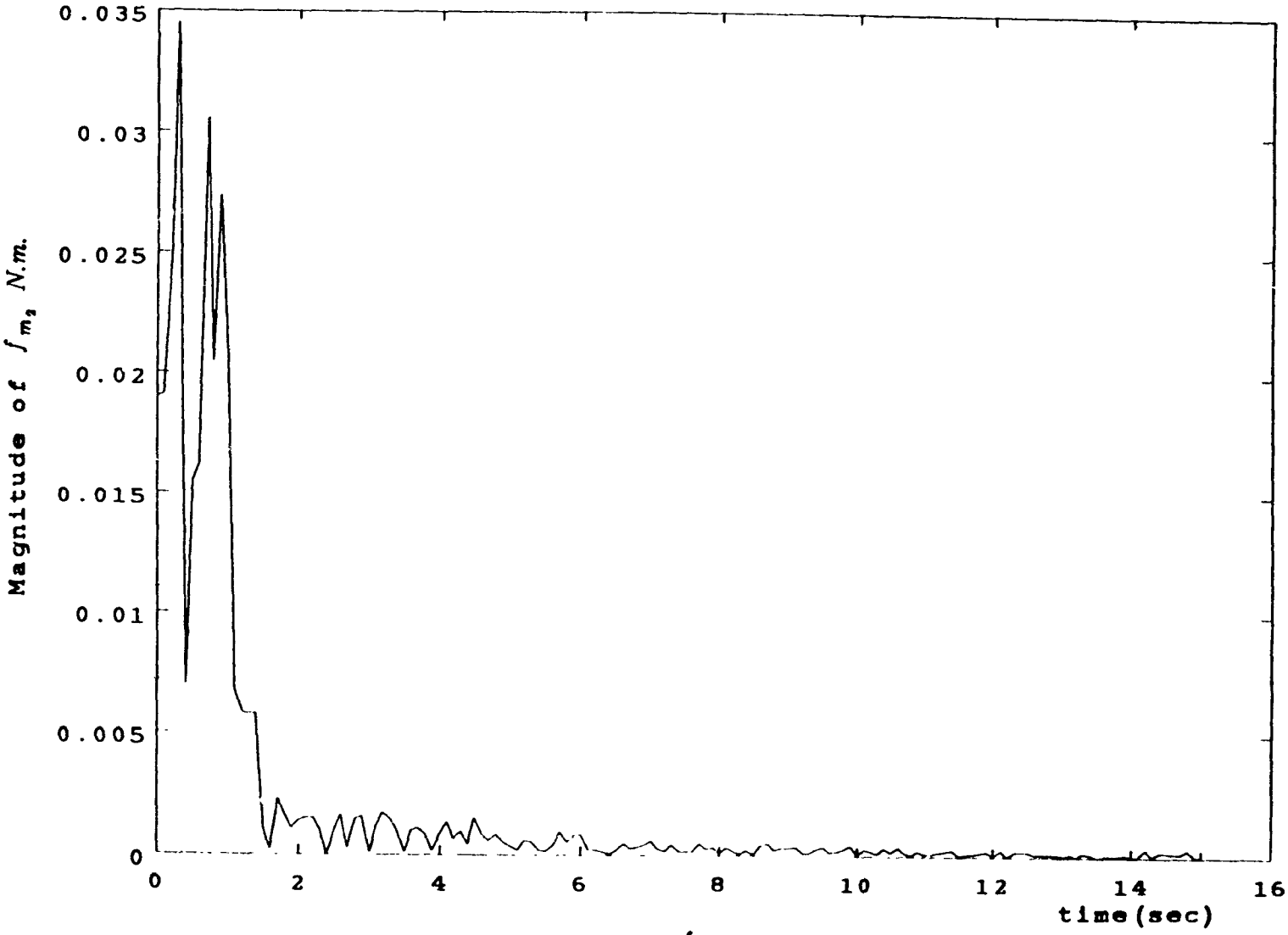

Fig.(5.1b) Behaviour of f_{m_2} for the closed-loop system

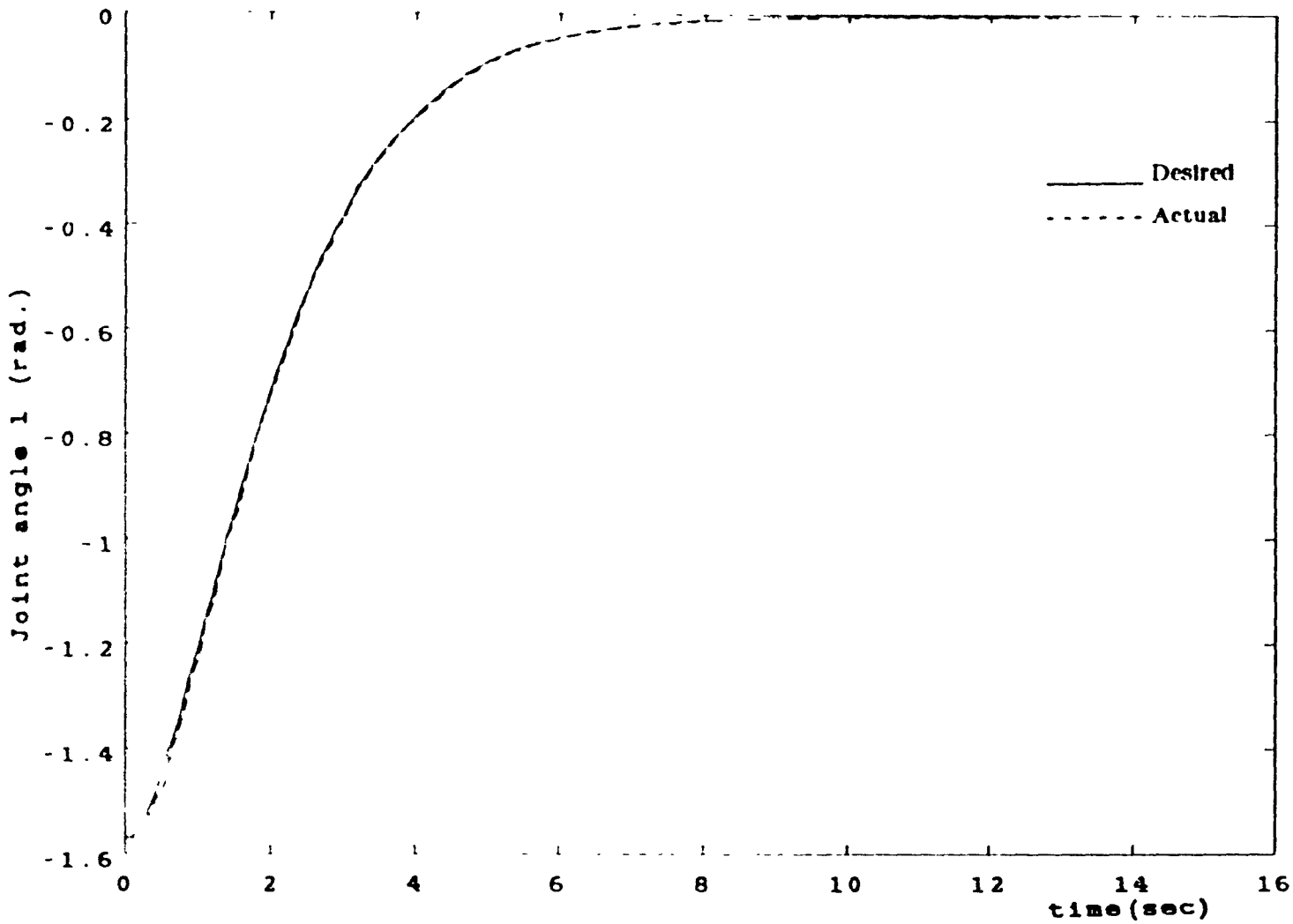

Fig.(5.2a) Closed-loop response of joint 1

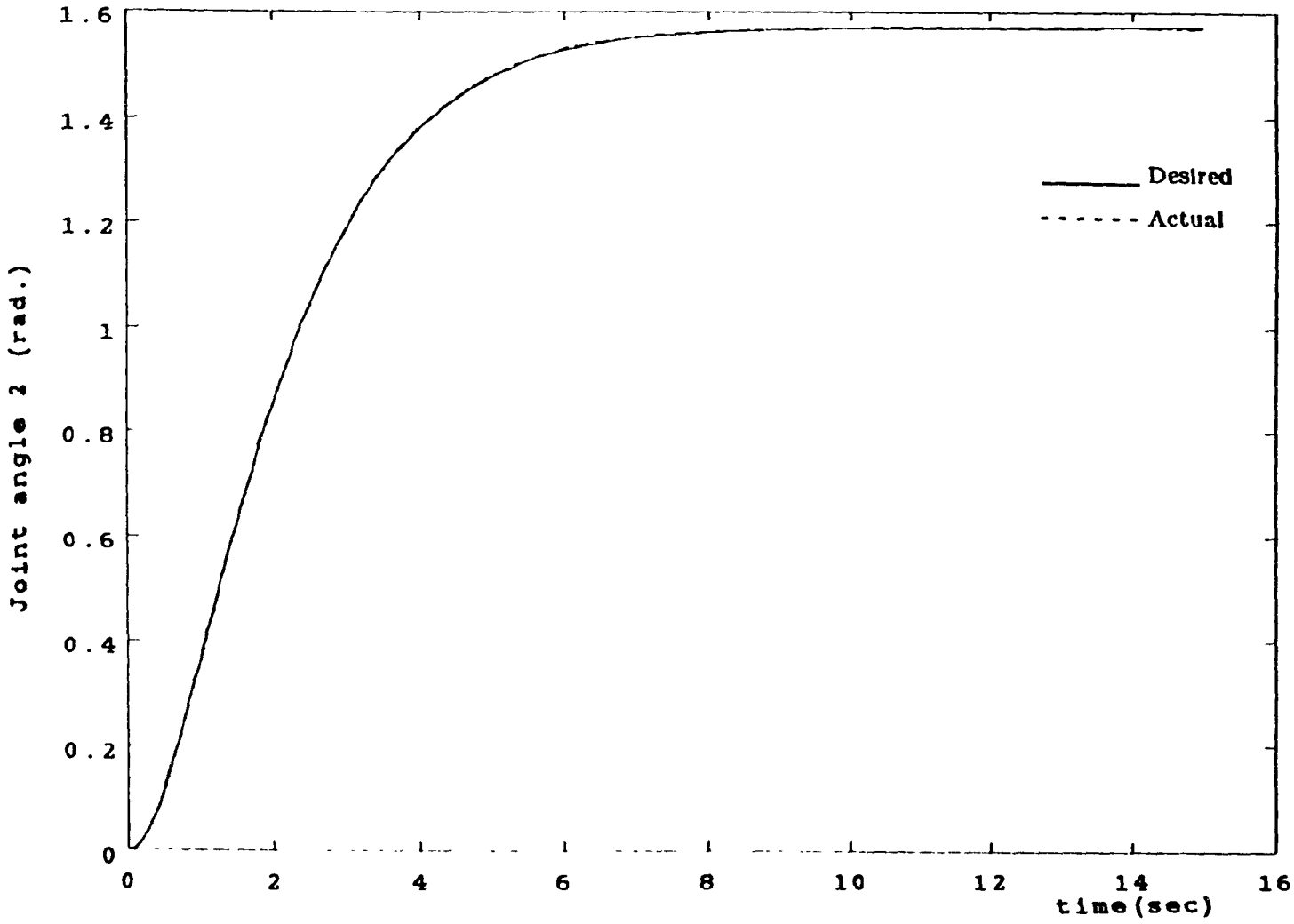

Fig.(5.2b) Closed-loop response of joint 2

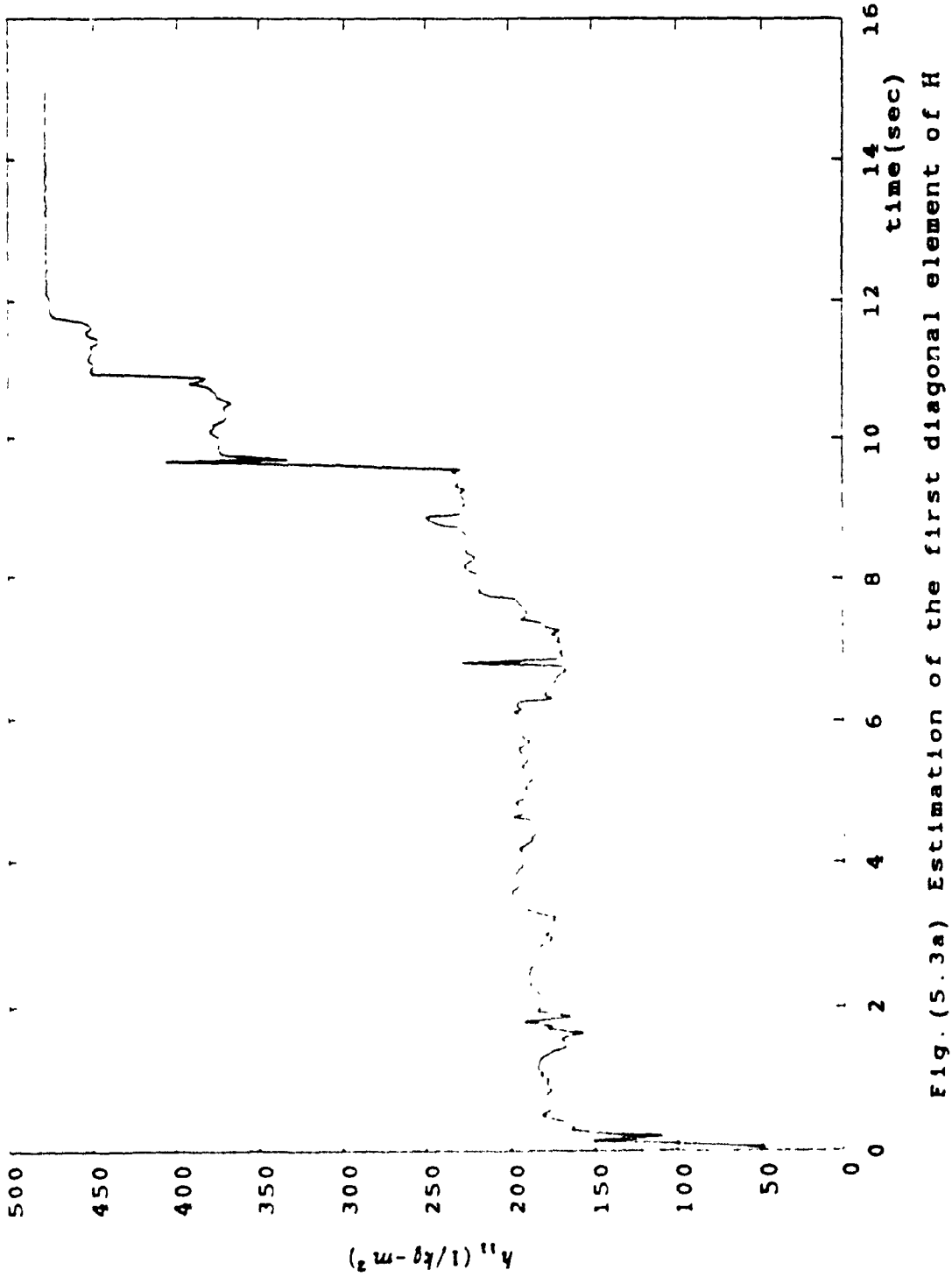

Fig.(5.3a) Estimation of the first diagonal element of H

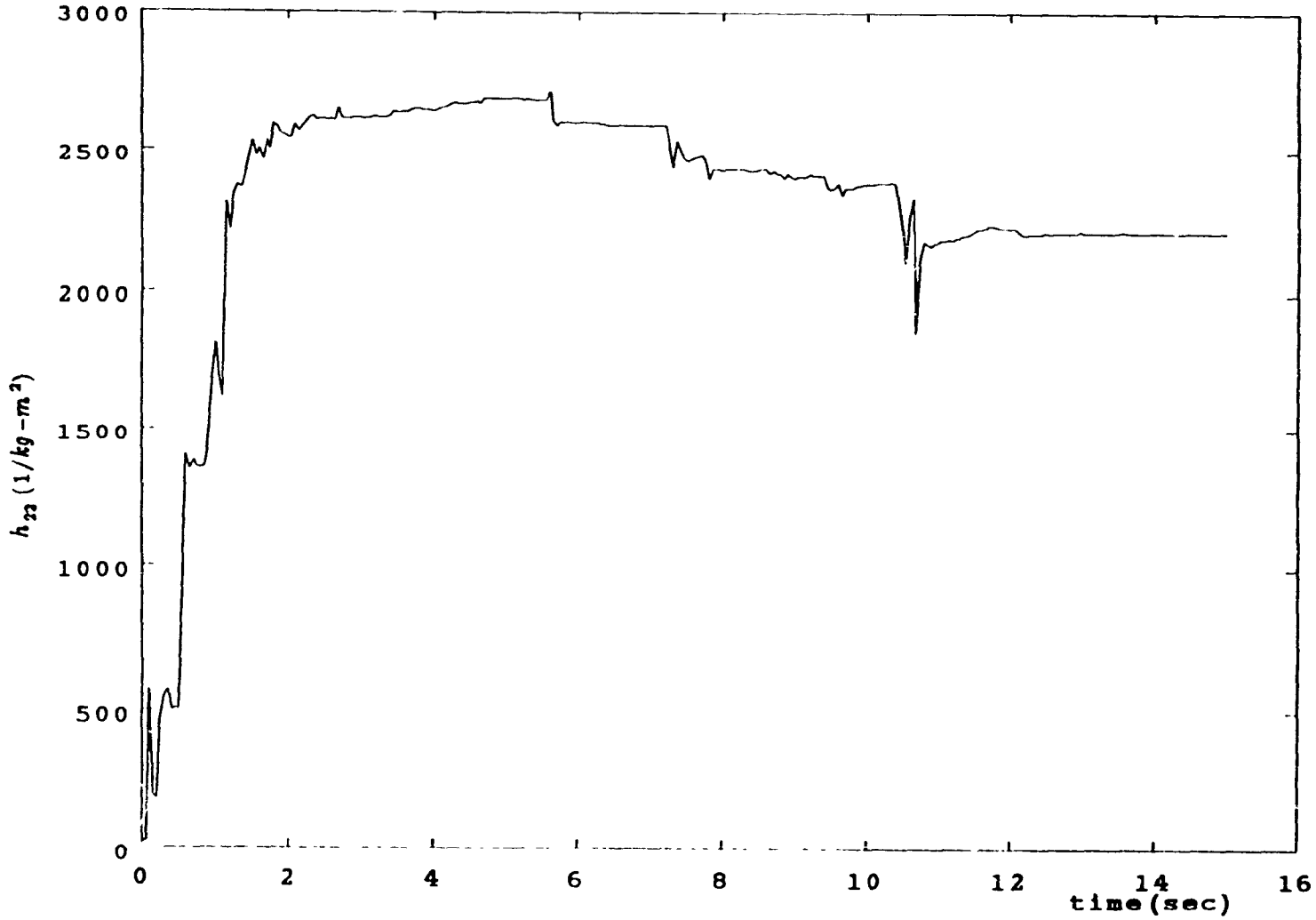

Fig. (5.3b) Estimation of the second diagonal element of H

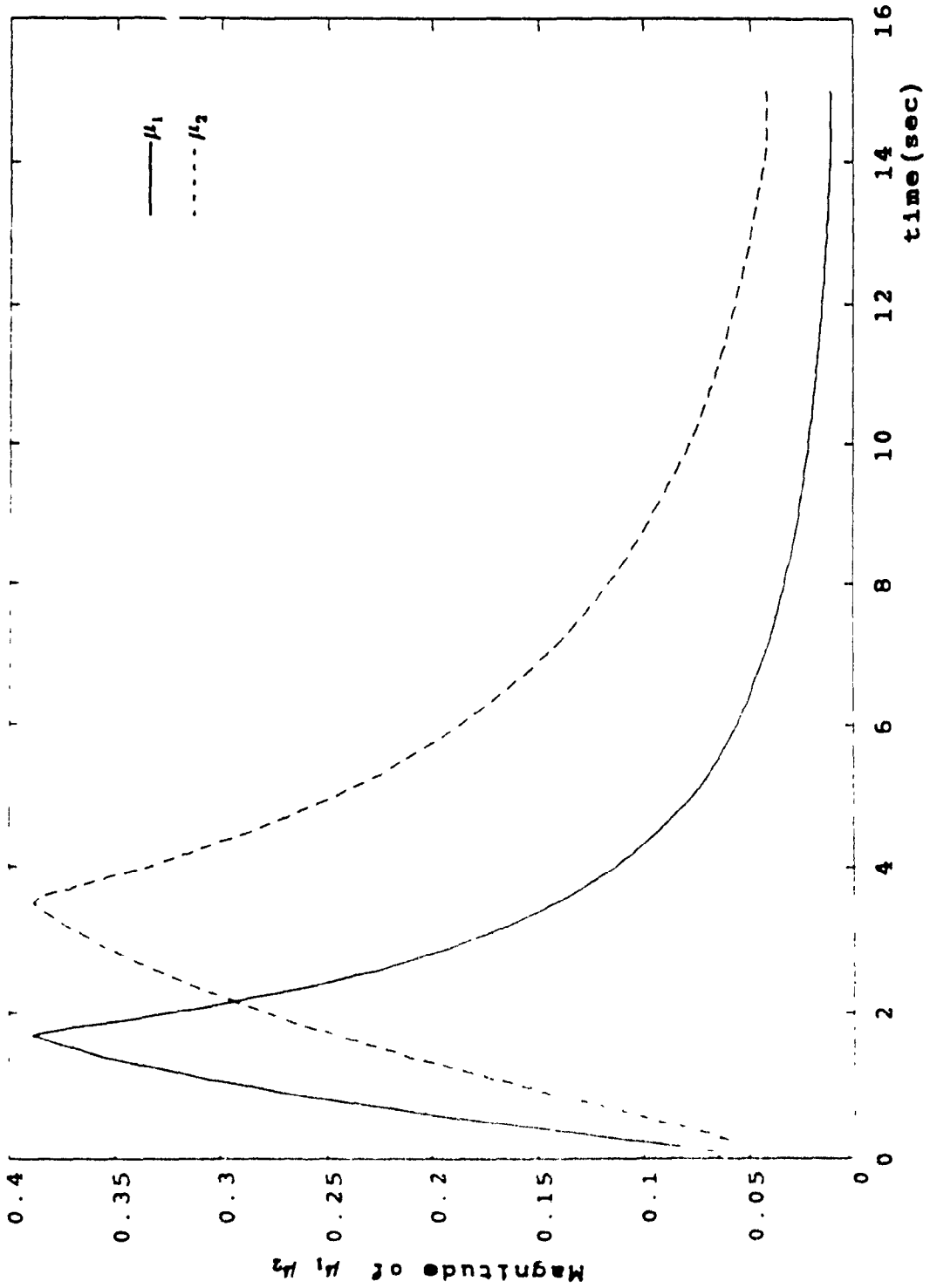

Fig.(5.4) Variation of μ_1 and μ_2

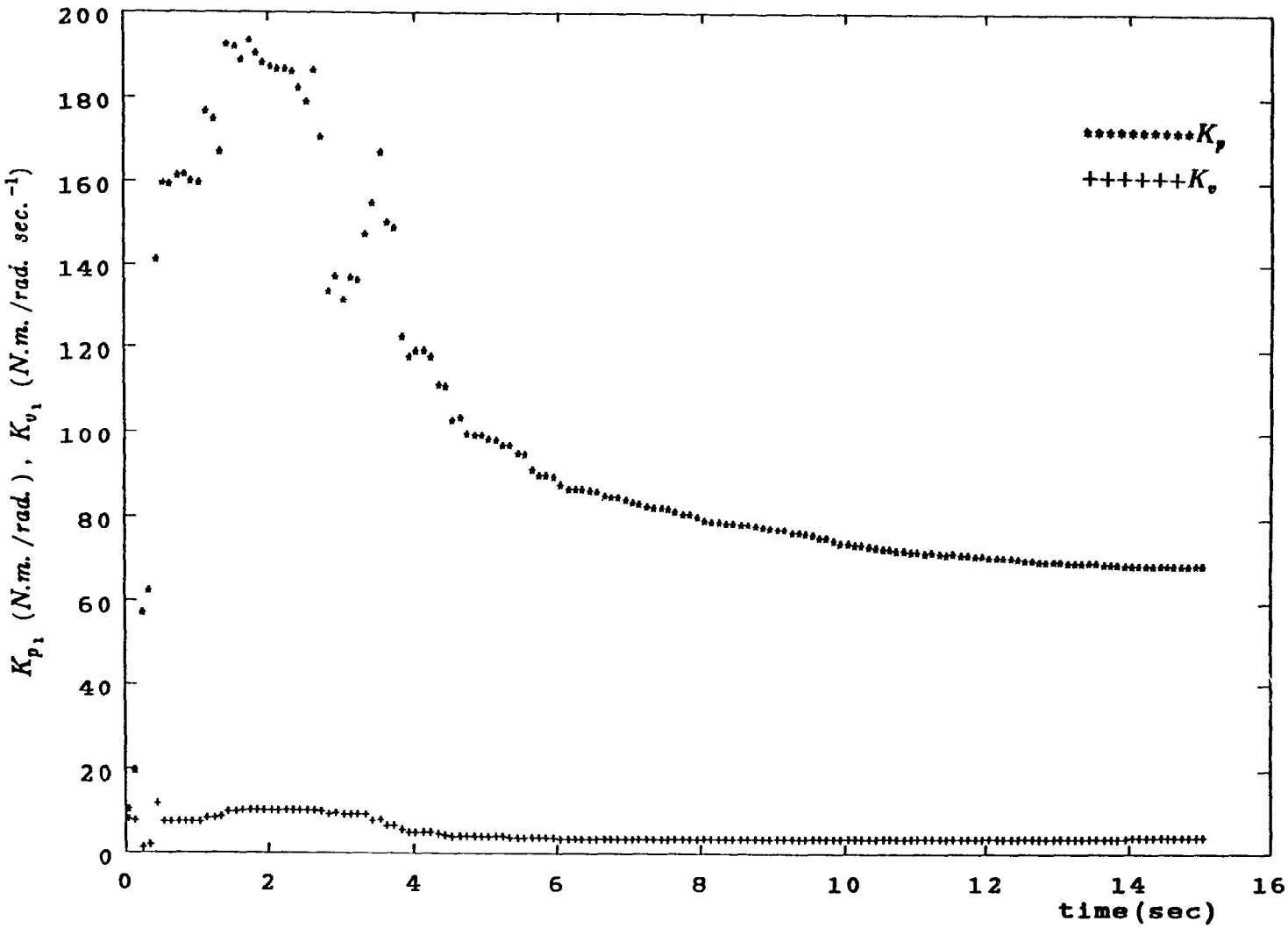

Fig.(5.5a) Adapted gains (K_{p_1}, K_{v_1}) for joint 1

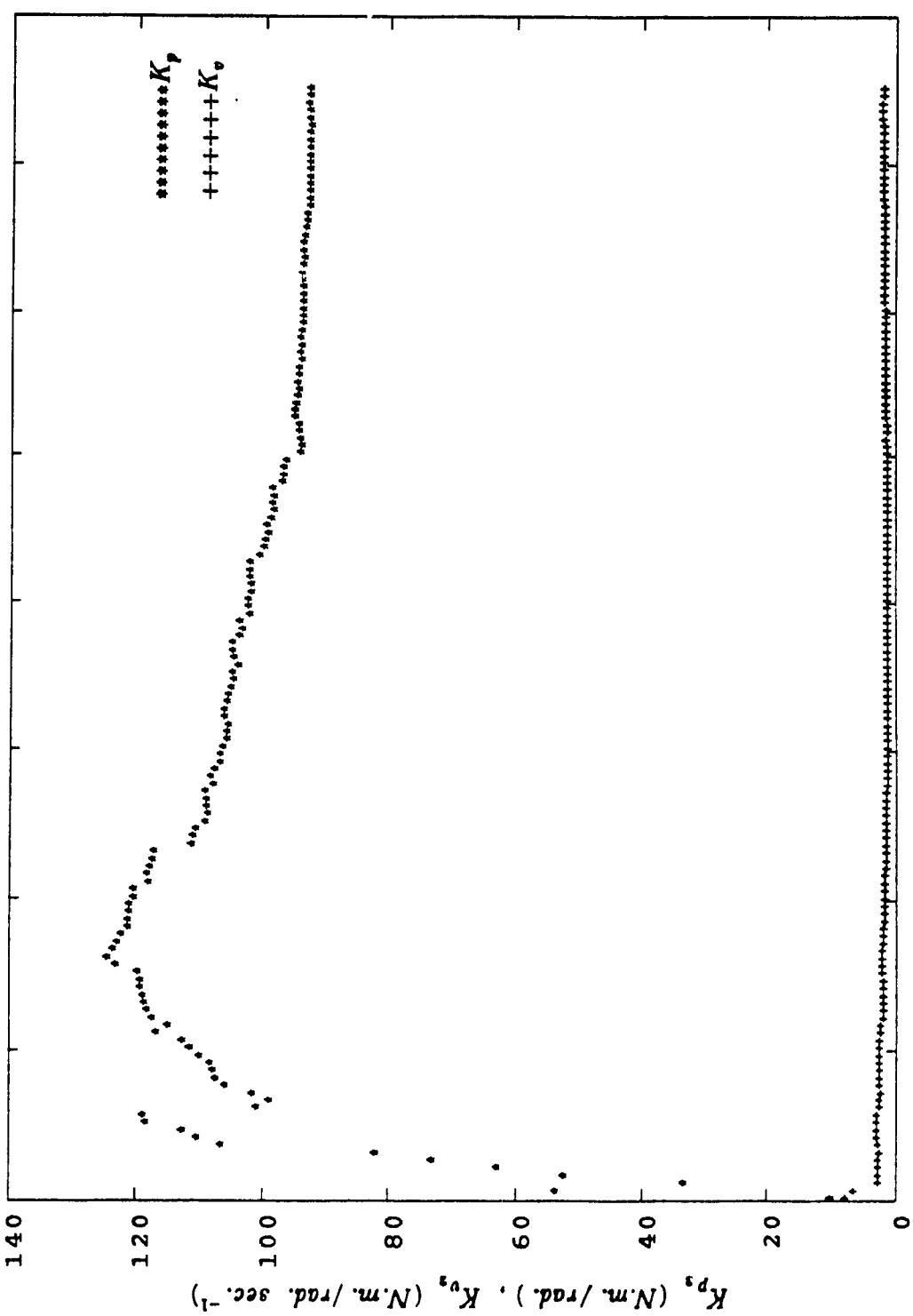

Fig.(5.5b) Adapted gains (K_{p_3}, K_{v_3}) for joint 2

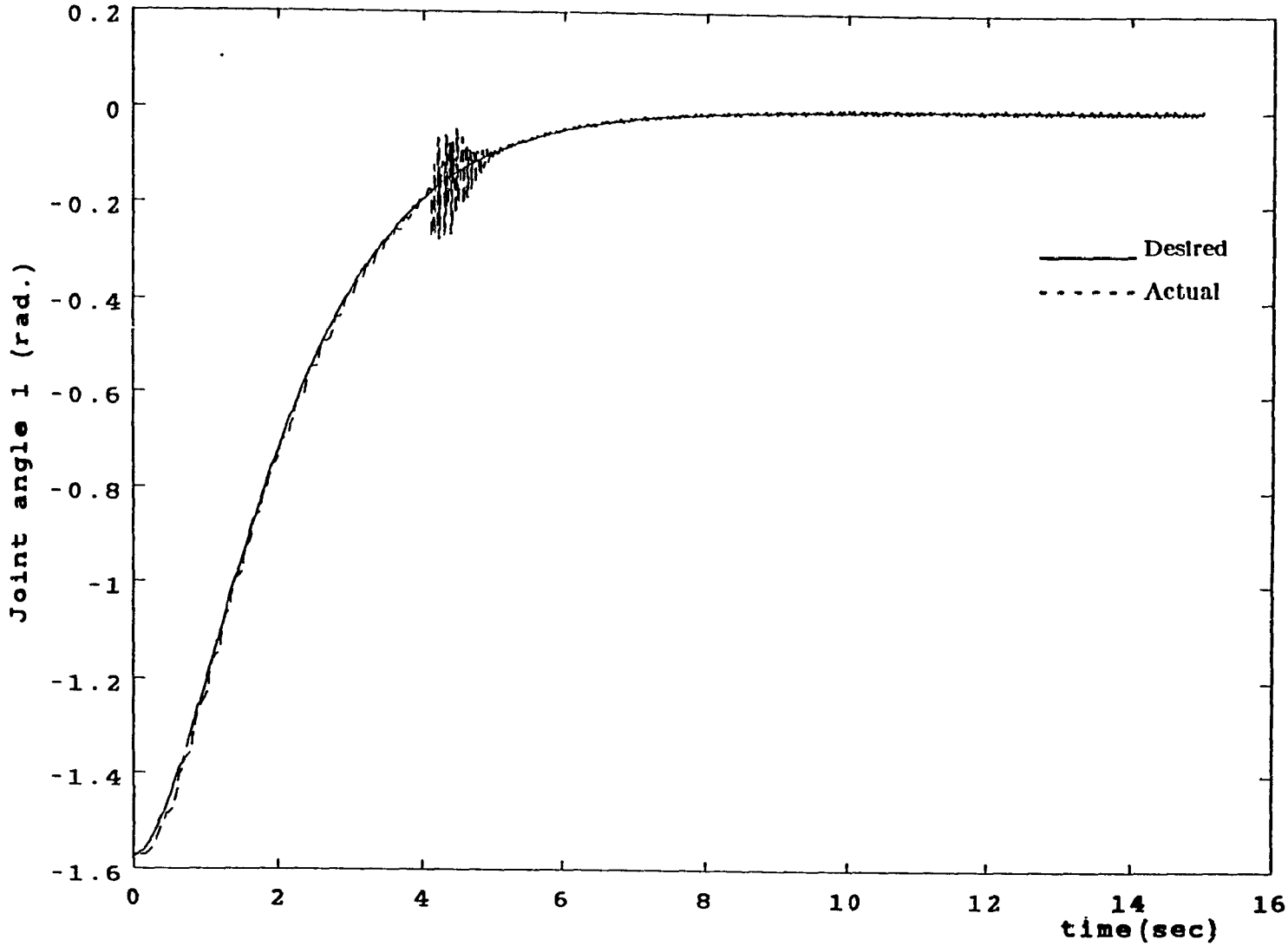

Fig.(5.6a) Closed-loop response of joint 1 under payload variation

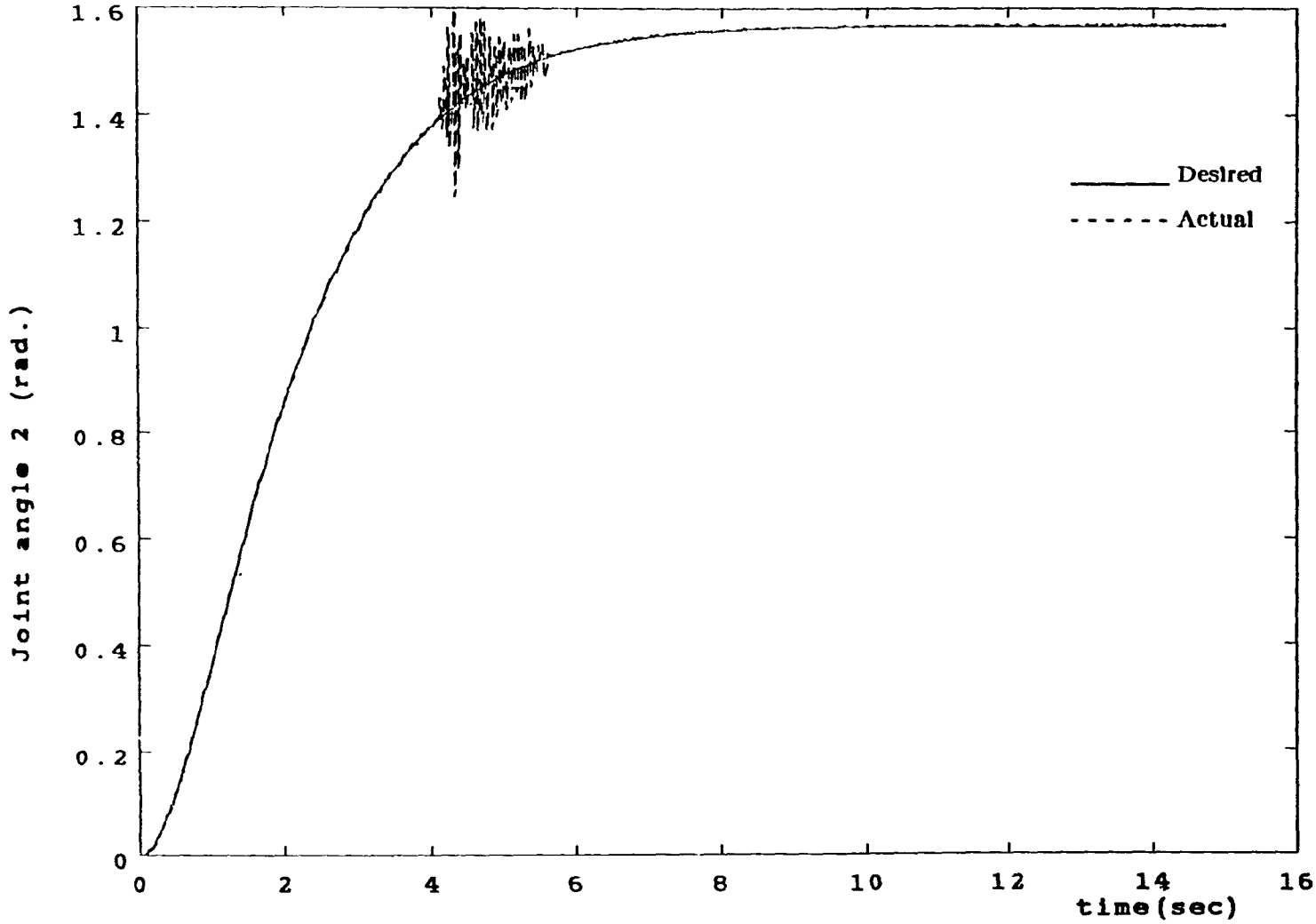

Fig.(5.6b) Closed-loop response of joint 2 under payload variation

CHAPTER SIX

CARTESIAN CONTROL OF ROBOT MANIPULATORS

In this chapter we will extend the adaptive control algorithm developed in Chapters 4 and 5 so that it can be applied in controlling the end-effector of the manipulator. We will first introduce a new coordinate system, the Cartesian coordinate system, and define the transformations necessary to transfer the system dynamic equations from joint coordinates to the end-effector coordinates by determining the forces acting on the end-effector. In the first section we discuss the basic principles of Cartesian control. In Sections 2-5 the algorithm described in Chapter 4 is modified for Cartesian control. An illustrative example is given in Section 6 to show the applicability of the algorithm.

6.1 INTRODUCTION

The motion of the end-effector of a robot manipulator is affected by the nonlinearities in the dynamics, unknown load variations, physical nonlinearities such as friction, backlash and unknown or unmodeled parameter variations and other disturbances. It is therefore necessary to focus on the design of robust controllers for robot manipulators so as to ensure that the end-effector tracks a given trajectory in Cartesian space.

Manipulator control is usually performed in joint space since the manipulator dynamics are described in terms of joint variables. So the desired trajectory in Cartesian space must be transformed using inverse kinematic relations to obtain the required joint trajectories [27,48,51]. The trajectory conversion process is quite difficult in terms of computational complexity since it requires the use of the inverse of the Jacobian as well as its derivatives (as defined in Chapter 2). Furthermore, such transformations are not possible near singular points, and also they are not in general unique unless some constraints are imposed. The inverse kinematics computations are usually performed many times during the motion of a manipulator thus making this type of control scheme very

complicated.

In order to control the motion of the end-effector in Cartesian space, it is necessary to get measurements of the end-effector motion using some form of sensing such as vision. This information can also be obtained by measuring the joint variables using encoders, and then solving the forward (direct) kinematic to get the Cartesian coordinate variables. This is a very much simpler problem than the inverse kinematics problem and gives a unique solution. In this case the desired reference trajectory can be written in Cartesian space.

In this chapter we consider such an approach and propose an adaptive control algorithm that uses least squares identification to obtain a piecewise linear Cartesian space models in differential equation form, based on which, feedforward and feedback controllers are determined to generate control commands to enable a robot manipulator to achieve trajectory tracking in Cartesian space. The adaptation scheme uses on-line identification of the diagonal terms of the unknown coefficient matrices of the piecewise linear time-invariant models which have the same inputs and outputs as the manipulator Cartesian dynamics model and similar dynamical structure. The feedforward controller is updated on-line according to the identified parameters of the model. Also a proportional-derivative feedback term which is obtained in terms of the identified parameters is used to ensure stability and yield good tracking performance. The algorithm takes into account the effects of unmodeled dynamics by adding an extra term to the controller to compensate for the effects of coupling between the subsystems and other unmodeled dynamics such as friction, backlash and payload variations.

The adaptive control scheme described in this chapter has a simple structure, is computationally fast and feasible for on-line implementation The adaptive controller does not require knowledge of the physical parameters of the robot or of the payload. The adaptive scheme does not use local linearization, inverse dynamics, or knowledge of any nonlinear terms in the feedforward or feedback paths. The algorithm allows for the

effect of unmodeled dynamics and ensures stability of the system and asymptotic convergence of tracking errors.

6.2 Cartesian State Space Model of Manipulator Dynamics

As defined in Chapter 4, the manipulator dynamic equation can be written as

$$\tau = M(\theta)\,\ddot{\theta} + B(\theta,\dot{\theta})\,\dot{\theta} + C(\theta)\,\theta \tag{6.2.1}$$

It should be noted that the above formulation relates the joint variables, θ, $\dot{\theta}$, $\ddot{\theta}$ to the joint torques τ, i.e. it is a joint-space formulation. Cartesian space dynamic equations which relate the acceleration of the end-effector expressed in Cartesian space, to the Cartesian forces and moments acting at the end-effector can be obtained using the manipulator Jacobian [5,54]. The manipulator dynamics in Cartesian space can be represented by the block diagram shown in Fig. (6.1). The Jacobian $J(\theta)$ is defined by the relationships:

$$\dot{x} = J(\theta)\,\dot{\theta} \tag{6.2.2a}$$

$$\tau = J^{T}(\theta)\,F \tag{6.2.2b}$$

where,

F is the $N \times 1$ force-torque vector acting on the end-effector

x is the $N \times 1$ position and orientation vector of the end-effector

Using (6.2.2), we can write (6.2.1) as:

$$F = M_x(\theta)\,\ddot{x} + B_x(\theta,\dot{\theta})\,\dot{c} + C_x(\theta)\,\theta \tag{6.2.3}$$

where,

$$M_x(\theta) = J^{-T}(\theta)\,M(\theta)\,J^{-1}(\theta) \tag{6.2.4a}$$

$$B_x(\theta,\dot{\theta}) = J^{-T}(\theta)\,\left[B(\theta,\dot{\theta}) - M(\theta)\,J^{-1}(\theta)\,\dot{J}(\theta)\right] \tag{6.2.4b}$$

$$C_x(\theta) = J^{-T}(\theta)\,C(\theta) \tag{6.2.4c}$$

Note that the Jacobian $J(\theta)$ is in the same frame as F and x. By using the kinematic equations [1,52]

$$x = KIN(\theta) \tag{6.2.5a}$$

$$\dot{x} = J(\theta)\,\dot{\theta} \tag{6.2.5b}$$

$$\ddot{x} = J(\theta)\,\ddot{\theta} + \dot{J}(\theta)\,\dot{\theta} \tag{6.2.5c}$$

M_x, B_x, C_x can be expressed as functions of x, $\dot{x}$ and $\ddot{x}$. i.e. (6.2.3) can be written as

$$F = M_x(x)\,\ddot{x} + B_x(x,\dot{x})\,\dot{x} + C_x(x)\,x \tag{6.2.6}$$

Equation (6.2.6) relates the motion of the end-effector to the forces acting on it. Eq. (6.2.6) can be rewritten in state-space form as

$$\dot{X} = \begin{bmatrix} 0 & I \\ -A_1 & -A_2 \end{bmatrix} X + \begin{bmatrix} 0 \\ H_x \end{bmatrix} F = \Phi_x\, X + \Omega_x\, F \tag{6.2.7}$$

where, $X \in \mathbb{R}^{2N\times 1} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$, $F \in \mathbb{R}^{N\times 1}$, $\Phi_x \in \mathbb{R}^{2N\times 2N}$ and $\Omega_x \in \mathbb{R}^{2N\times N}$ and $A_1 = M_x^{-1}C_x$, $A_2 = M_x^{-1}B_x$ and $H_x = M_x^{-1}$ and $N \leq 6$ is the number of degrees of freedom in the Cartesian space.

The matrices Φ_x and Ω_x are constant matrices at a given value of time, and are to be identified. If the matrix H_x has full rank for all values of time, it follows from the structure of (6.2.7) that the linear model is controllable and therefore stabilizable by state feedback. Assuming that the matrices M_x, B_x, C_x are slowly time-varying, we can approximate the nonlinear, time-varying model (6.2.7) by a number of piecewise linear, time-invariant models having the same structure as (6.2.7) but with constant matrices Φ_x and Ω_x. We divide the total trajectory time into a number of intervals, say r, of duration Δt_k, $k = 1,\ldots,r$ and over the interval Δt_k, the time-varying system (6.2.7) is approximated by a linear time-invariant system

$$\dot{X} = \begin{bmatrix} 0 & I \\ -A_1(t_{k-1}) & -A_2(t_{k-1}) \end{bmatrix} X + \begin{bmatrix} 0 \\ H_x(t_{k-1}) \end{bmatrix} F \tag{6.2.8}$$

where, t_{k-1} is the starting time of the interval Δt_k. Using recursive identification, the parameters of (6.2.8) are to be estimated on-line. As the end-effector moves along the trajectory, for each Δt_k, we have a particular model of the system. This set of models

represents the dynamic behavior of the end-effector of the manipulator. A feedforward-feedback controller is then designed for each linear time-invariant model.

6.3 The Adaptive Control Law

We now consider the design of a controller for the manipulator to enable it to track a desired path. The desired path is described in Cartesian space. The goal of the controller is to minimize the deviation from the desired path, i.e. to achieve tracking while maintaining stability of the resulting closed-loop system. The model used in the controller design is the piecewise time-invariant model (6.2.8). The identification of such a model requires approximately $4N^2$ parameters as described in Chapter 4. The digital simulation of the model is very time consuming because of the large number of floating point operations involved. Therefore, to simplify the construction of the controller, it is assumed that the coupling terms representing the interactions of subsystems can be neglected. Such a control is called decoupled control. Therefore, for decoupled Cartesian control, each Cartesian coordinate is considered as a subsystem of the overall system. Then the model (6.2.8) can be divided into N equations for the N interconnected subsystems and it can be written as:

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_i & -b_i \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ h_{x_i} \end{bmatrix} f_i \qquad i = 1,\ldots,N \tag{6.3.1}$$

Dropping the coupling terms will gives rise to errors which will affect the stability and response of the system. However, we can compensate for these errors to some extent by considering the terms that are neglected as being part of the unmodeled dynamics and modifying the control strategy to account for them.

For decoupled control, we choose the control law:

$$f_i = \ddot{x}_{d_i} + \delta f_i + W_{x_i}(t) \tag{6.3.2}$$

where, $\delta f \in \mathbb{R}^{N\times 1}$ is the correcting force vector obtained from a feedback con-

troller, where f_i is the i^{th} element of f, and $W_{x_i}(t)$ is generated by a feedforward controller. The feedback controller is chosen as a multivariable proportional-derivative (PD) controller defined by

$$\delta f_i = k_{p_i} e_{x_i} + k_{v_i} \dot{e}_{x_i} \tag{6.3.3}$$

where

$$e_{x_i} = x_{d_i} - x_i$$
$$\dot{e}_{x_i} = \dot{x}_{d_i} - \dot{x}_i$$

are, respectively, the i^{th} elements of the Cartesian position and velocity error vectors; and k_{p_i}, k_{v_i} are the diagonal elements of the proportional and derivative diagonal feedback gain matrices, K_p and $K_v \in \mathbb{R}^{N \times N}$.

Next we will design the feedforward controller for the end-effector by substituting the control law (6.3.2) into the model (6.3.1). This yields

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_i & -b_i \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \end{bmatrix} + \begin{bmatrix} 0 \\ h_{x_i} \end{bmatrix} \left[\ddot{x}_{d_i} + \delta f_i + W_{x_i} \right]$$

from where

$$\ddot{x}_i = - a_i x_i - b_i \dot{x}_i + h_{x_i} \ddot{x}_{d_i} + h_{x_i} \delta f_i + h_{x_i} W_{x_i}$$

Substituting for x_i, $\dot{x}_i$, $\ddot{x}_i$ from the error equations and using (6.3.3), we get

$$\ddot{e}_{x_i} + \left[b_i + h_{x_i} k_{v_i} \right] \dot{e}_{x_i} + \left[a_i + h_{x_i} k_{p_i} \right] e_{x_i} + h_{x_i} W_{x_i} = \ddot{x}_{d_i} + b_i \dot{x}_{d_i} + a_i x_{d_i} \quad h_{x_i} \ddot{x}_{d_i}$$

From this equation, it can be seen that if we choose

$$W_{x_i} = h_{x_i}^{-1} \left[(1 - h_{x_i}) \ddot{x}_{d_i} + b_i \dot{x}_{d_i} + a_i x_{d_i} \right] \tag{6.3.4}$$

then the steady state error will be zero

The feedforward control W_{x_i} is based on the reference input and is updated linearly in terms of the identified model parameters. The control input to the end-effector of the manipulator is now given by

$$f_i = \ddot{x}_{d_i} + k_{v_i}\dot{e}_{x_i} + k_{p_i}e_{x_i} + h_{x_i}^{-1}\left[(1-h_{x_i})\ddot{x}_{d_i} + b_i\dot{x}_{d_i} + a_i x_{d_i}\right] \quad (6.3.5)$$

Now, let us consider the Cartesian manipulator dynamic equation (6.2.6). The force input for each degree of freedom of the end-effector in Cartesian space can be written as

$$f_i = \sum_{j=1}^{N} m_{x_{ij}}\ddot{x}_j + \sum_{j=1}^{N} b_{x_{ij}}\dot{x}_j + \sum_{j=1}^{N} c_{x_{ij}}x_j \qquad i=1,\ldots,N \quad (6.3.6)$$

For the time being, we will assume that the interconnections between the subsystems are neglected and we will control each degree of freedom in Cartesian space independently. We will consider the interconnection terms as being part of the unmodeled dynamics later on. Therefore Eq. (6.3.6) becomes

$$f_i = m_{x_{ii}}\ddot{x}_i + b_{x_{ii}}\dot{x}_i + c_{x_{ii}}x_i \quad (6.3.7)$$

Now, by applying the control law (6.3.5) to the end-effector of the manipulator model (6.3.7), we get

$$\ddot{x}_{d_i} + k_{v_i}\dot{e}_{x_i} + k_{p_i}e_{x_i} + h_{x_i}^{-1}\left[(1-h_{x_i})\ddot{x}_{d_i} + b_i\dot{x}_{d_i} + a_i x_{d_i}\right] = m_{x_{ii}}\ddot{x}_i + b_{x_{ii}}\dot{x}_i + c_{x_{ii}}x_i \quad (6.3.8)$$

From the system model (6.2.6) and (6.2.7), during the time interval Δt_k , we have

$$a_i = m_{x_{ii}}^{-1}\, c_{x_{ii}} \quad (6.3.9a)$$

$$b_i = m_{x_{ii}}^{-1}\, b_{x_{ii}} \quad (6.3.9b)$$

$$h_{x_i} = m_{x_{ii}}^{-1} \quad (6.3.9c)$$

Using (6.3.9) in (6.3.8) and simplifying, yields

$$h_{x_i}^{-1}\left[\left(\ddot{x}_{d_i} - \ddot{x}_i\right) + b_i\left(\dot{x}_{d_i} - \dot{x}_i\right) + a_i\left(x_{d_i} - x_i\right)\right] + k_{v_i}\dot{e}_{x_i} + k_{p_i}e_{x_i} = 0$$

From which we obtain the Cartesian error equation

$$\ddot{e}_{x_i} + \left[b_i + h_{x_i}k_{v_i}\right]\dot{e}_{x_i} + \left[a_i + h_{x_i}k_{p_i}\right]e_{x_i} = 0 \quad (6.3.10)$$

which can be written in state space form as

$$\begin{bmatrix}\dot{e}_{x_i}\\ \ddot{e}_{x_i}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -(a_i + h_{x_i}k_{p_i}) & -(b_i + h_{x_i}k_{v_i})\end{bmatrix}\begin{bmatrix}e_{x_i}\\ \dot{e}_{x_i}\end{bmatrix} \qquad i=1,\ldots,N$$

which implies that e_{x_i} , $\dot{e}_{x_i} \rightarrow 0$ as $t \rightarrow \infty$ provided that $a_i + h_{x_i} k_{p_i} > 0$ and $b_i + h_{x_i} k_{v_i} > 0$, $i = 1,...,N$. Let

$$\lambda_{v_i} = b_i + h_{x_i} k_{v_i}$$
$$\lambda_{p_i} = a_i + h_{x_i} k_{p_i}$$

from which the controller gains can be found as :

$$k_{v_i} = h_{x_i}^{-1} \left[\lambda_{v_i} - b_i \right]$$

$$k_{p_i} = h_{x_i}^{-1} \left[\lambda_{p_i} - a_i \right]$$

The matrices $\Lambda_p = diag\,(\lambda_{p_i})$ and $\Lambda_v = diag\,(\lambda_{v_i})$ are obtained from the desired performance of the tracking error which can be specified in the following state space form

$$\dot{z}_{x_i} = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} z_{x_i} \qquad (6.3.11)$$

where ζ_i , ω_i are the damping factor and undamped natural frequency and $\Lambda_v = diag\,(2\zeta_i \omega_i)$ and $\Lambda_p = diag\,(\omega_i^2)$ are $N \times N$ matrices.

6.4 Compensation for the Effects of Unmodeled Dynamics

The error equation (6.3.11) can be written as

$$\dot{z}_{x_i} = \Pi_{x_i} z_{x_i} \qquad i = 1,...,N \qquad (6.4.1)$$

where. $\Pi_{x_i} \in \mathbb{R}^{2 \times 2}$ and $z_{x_i} = [e_{x_i} \quad \dot{e}_{x_i}]^T \in \mathbb{R}^{2 \times 1}$. Equation (6.4.1) represents the desired closed-loop system. Let $f_{x_{s_i}}$ denote the unmodeled dynamics due to coupling between the subsystems and other unmodeled dynamics Then the error equation (6.3.10) can be written as

$$z_{x_{s_i}} = \Pi_{x_i} z_{x_{s_i}} + f_{x_{s_i}}(t) \qquad (6.4.2)$$

Equation (6.4.2) represents the behavior of the system without compensation for the unmodeled dynamics.

Next, we assume that $f_{x_{a_i}}(t)$ is an unknown slowly time-varying function and we introduce another term in the feedback law to compensate for its effect. The manipulator end-effector dynamical model (6.3.7) can be represented in terms of the nominal values of the manipulator end-effector dynamics, $m_{x_{ii}}{}^{o}$, $b_{x_{ii}}{}^{o}$, $c_{x_{ii}}{}^{o}$, and a nonlinear term $f_{x_{a_i}}(t)$ which represents coupling terms between the Cartesian subsystems and other unmodeled slowly time-varying dynamics. Then (6.3.7) can be rewritten as

$$f_i = f_{x_{a_i}}(t) + m_{x_{ii}}{}^{o}\,\ddot{x}_i + b_{x_{ii}}{}^{o}\,\dot{x}_i + c_{x_{ii}}{}^{o}\,x_i \tag{6.4.3}$$

In order to compensate for the effects of these unmodeled dynamics, a control function $f_{x_{c_i}}(t)$ is added to the control law (6.3.5). Then the total force input for each subsystem of the manipulator end-effector becomes

$$f_i = \ddot{x}_{d_i} + k_{v_i}\,\dot{e}_{x_{a_i}} + k_{p_i}\,e_{x_{a_i}} + h_{x_i}{}^{-1}\left[(1-h_{x_i})\ddot{x}_{d_i} + b_i\,\dot{x}_{d_i} + a_i\,x_{d_i}\right] + f_{x_{c_i}}(t) \tag{6.4.4}$$

The above control computes a Cartesian force vector which should be applied to the end-effector. However, we cannot actually apply this force to the end-effector directly. Instead, we compute the joint torques needed to effectively apply this force by using the Jacobian, i.e.,

$$\tau = J^T(\theta)\,F$$

So that the input torque for each joint of the manipulator is

$$\tau_i = \sum_{j=1}^{N} J_{ji}^{T}(\theta)\,f_j \qquad i = 1,\ldots,n \tag{6.4.5}$$

The control scheme (6.4.4) developed in this section is simple to construct and does not require the inclusion of any nonlinear terms, or complex dynamics of the manipulator. The terms h_{x_i}, a_i and b_i are obtained using recursive identification. The convergence of the adaptive controller is independent of the initial values of a_i, b_i, h_{x_i} and

k_{v_i}, k_{p_i}.

6.5 Robust Control and Stability Analysis

By applying the control law (6.4.4) to the manipulator end-effector (6.4.3), we get

$$f_{x_{c_i}}(t)+k_{v_i}\dot{e}_{x_{e_i}}+k_{p_i}e_{x_{e_i}}+h_{x_i}^{-1}\left(\ddot{x}_{d_i}+b_i\dot{x}_{d_i}+a_i x_{d_i}\right)=f_{x_{e_i}}(t)+m_{x_{ii}}{}^{0}\ddot{x}_i+b_{x_{ii}}{}^{0}\dot{x}_i+c_{x_{ii}}{}^{0}x_i$$

Assuming that $m_{x_{ii}}{}^{0}$, $b_{x_{ii}}{}^{0}$, $c_{x_{ii}}{}^{0}$ satisfy (6.3.9), we obtain the following error equation

$$\ddot{e}_{x_{e_i}}+\left[b_i+h_{x_i}k_{v_i}\right]\dot{e}_{x_{e_i}}+\left[a_i+h_{x_i}k_{p_i}\right]e_{x_{e_i}}=h_{x_i}\left[f_{x_{e_i}}(t)-f_{x_{c_i}}(t)\right] \qquad (6.5.1)$$

which can be written in state space form as

$$\begin{bmatrix}\dot{e}_{x_{e_i}}\\ \ddot{e}_{x_{e_i}}\end{bmatrix}=\begin{bmatrix}0 & 1\\ -\lambda_{p_i} & -\lambda_{v_i}\end{bmatrix}\begin{bmatrix}e_{x_{e_i}}\\ \dot{e}_{x_{e_i}}\end{bmatrix}+\begin{bmatrix}0\\ h_{x_i}\end{bmatrix}(f_{x_{e_i}}-f_{x_{c_i}})$$

or

$$\dot{z}_{x_{e_i}}=\Pi_{x_i}z_{x_{e_i}}+\beta_{x_i}f_{x_{m_i}} \qquad (6.5.2)$$

where

$$\beta_{x_i}=\begin{bmatrix}0\\ h_{x_i}\end{bmatrix}$$

and $f_{x_{m_i}}=f_{x_{e_i}}-f_{x_{c_i}}$, i.e. $f_{x_{e_i}}=f_{x_{m_i}}+f_{x_{c_i}}$, where $f_{x_{m_i}}$ denotes the error in the approximation of the actual unmodeled dynamics by the computed value of $f_{x_{c_i}}$.

Let $f_{x_{m_i}}$ be given by

$$\dot{f}_{x_{m_i}}=-h_{x_i}^{-1}\beta_{x_i}{}^{T}P_i z_{x_{e_i}} \qquad (6.5.3)$$

where

$$P_i=\begin{bmatrix}p_{1_i} & p_{2_i}\\ p_{2_i} & p_{4_i}\end{bmatrix}$$

is a positive-definite solution of the Lyapunov equation

$$\Pi_{x_i}{}^T P_i + P_i \Pi_{x_i} = -2Q_i \tag{6.5.4}$$

where, Q_i is a positive-definite matrix. Equation (6.5.3) and Eq.(6.5.2) together give rise to the following augmented system

$$\begin{bmatrix} \dot{z}_{x_{s_i}} \\ \dot{f}_{x_{m_i}} \end{bmatrix} = \begin{bmatrix} \Pi_{x_i} & \beta_{x_i} \\ -h_{x_i}{}^{-1}\beta_{x_i}{}^T P_i & 0 \end{bmatrix} \begin{bmatrix} z_{x_{s_i}} \\ f_{x_{m_i}} \end{bmatrix} \qquad i=1,\ldots,N \tag{6.5.5}$$

Theorem 6.1

The closed-loop system (6.5.2) with $f_{x_{m_i}}$ satisfying (6.5.3), is asymptotically stable.

Proof

Let us choose a Lyapunov function

$$V_i(t) = z_{x_{s_i}}{}^T P_i z_{x_{s_i}} + q_{0_i} f_{x_{m_i}}{}^2 \tag{6.5.6}$$

where, $q_{0_i} > 0$. Taking derivatives of both sides of (6.5.6), we get

$$\dot{V}_i = -2z_{x_{s_i}}{}^T Q_i z_{x_{s_i}} + 2\left[q_{0_i} f_{x_{m_i}} \dot{f}_{x_{m_i}} + f_{x_{m_i}} \beta_{x_i}{}^T P_i z_{x_{s_i}} \right] \tag{6.5.7}$$

For stability, it is required that $\dot{V}_i(t) \leq 0$ along the trajectory $z_{x_{s_i}}(t)$. This can be satisfied by using (6.5.3). Then (6.5.7) becomes

$$\dot{V}_i(t) = -2z_{x_{s_i}}{}^T Q_i z_{x_{s_i}} + 2 f_{x_{m_i}} \left[\beta_{x_i}{}^T P_i z_{x_{s_i}} - q_{0_i} h_{x_i}{}^{-1} \beta_{x_i}{}^T P_i z_{x_{s_i}} \right] \tag{6.5.8}$$

Since q_{0_i} is any positive number, let us choose $q_{0_i} = h_{x_i}$. This yields

$$\dot{V}_i(t) = -2z_{x_{s_i}}{}^T Q_i z_{x_{s_i}}$$

i.e.

$$\dot{V}_i(t) \leq 0$$

Since $\dot{V}_i$ is negative semi-definite and V_i is positive-definite, then the origin of (6.5.5) is stable. As in Theorem 4.1 (Chapter 4), by invoking La Salle's Theorem, it can be shown that the system (6.5.5) is asymptoticaly stable, i.e. $z_{x_{s_i}}(t) \rightarrow 0$ *as* $t \rightarrow \infty$. This

completes the proof of the theorem.

Remark 1

Since, $f_{x_{e_i}}$ is slowly time-varying, $\dot{f}_{x_{e_i}} \approx 0$, then $\dot{f}_{x_{m_i}} = -\dot{f}_{x_{c_i}}$ and Eq. (6.5.3) becomes

$$\dot{f}_{x_{c_i}} = h_{x_i}^{-1} \beta_{x_i}^T P_i \, z_{x_{e_i}}$$

or

$$\dot{f}_{x_{c_i}} = h_{x_i}^{-1} \begin{bmatrix} 0 & h_{x_i} \end{bmatrix} \begin{bmatrix} p_{1_i} & p_{2_i} \\ p_{2_i} & p_{3_i} \end{bmatrix} \begin{bmatrix} e_{x_{e_i}} \\ \dot{e}_{x_{e_i}} \end{bmatrix}$$

resulting in,

$$\dot{f}_{x_{c_i}}(t) = p_{2_i} \, e_{x_{e_i}} + p_{3_i} \, \dot{e}_{x_{e_i}} \tag{6.5.9}$$

Equation (6.5.9) can be solved for $f_{x_{c_i}}(t)$ as

$$f_{x_{c_i}}(t) = f_{x_{c_i}}(t_o) + \int_{t_o}^{t} p_{2_i} e_{x_{e_i}} dt + p_{3_i} e_{x_{e_i}} \tag{6.5.10}$$

where $f_{x_{c_i}}(t_o)$ denotes the initial value of $f_{x_{c_i}}(t)$.

Remark 2

As we discused in Chapters 4 and 5, p_{2_i} and p_{3_i} depend on the solution of the Lyapunov equation (6.5.4) i.e. on the choice of Q_i. Similar analysis can be applied to the Cartesian control scheme for selecting Q_i i.e. using the pole-assignment technique of Chapter 4 or the bounded uncertainty approach of Chapter 5.

6.6 NUMERICAL SIMULATION

In this section, we will show the trajectory following capability of the proposed control scheme by considering the problem of controlling the Cartesian motion of the end-effector of a 3-link PUMA type manipulator shown in Fig. 6.2. The nominal param-

eters of the manipulator are assumed [7] to be as follows: The mass of the links are:

$$m_1=33.0\ kg\ ,\quad m_2=77.0\ kg\ ,\quad \text{and}\quad m_3=36.2\ kg$$

The inertia of the links in $kg-m^2$ are :

$$I_{1xx}=I_{1zz}=1.0,\quad I_{1yy}=1.5,\quad I_{2xx}=4.37$$
$$I_{2yy}=14.3,\quad I_{2zz}=10.9,\quad I_{3xx}=4.77,\quad I_{3yy}=4.79 \text{ and } I_{3zz}=0.15$$

The locations of the centers of mass of the links (in meters) are :

$$P_{c1}=[0\ \ 0\ \ 0.08]^T,\quad P_{c2}=[-0.225\ \ 0\ \ 0.217]^T \quad \text{and} \quad P_{c3}=[0\ \ 0\ \ 0.22]^T$$

The Denavit-Hartenberg parameters (as indicated in the Fig. 6.2) are :

$$a_1=a_3=0,\quad a_2=0.45,\quad d_1=d_2=0,\quad d_3=0.125,\quad d_4=0.44\quad meter$$

The twist angles are given as $[-\pi/2\ \ 0\ \ \pi/2]^T$ radians.

The kinematic equations of the manipulator [7] are :

$$p_x=c_1s_{23}d_4-s_1(d_2+d_3)+a_2c_1s_2$$
$$p_y=s_1s_{23}d_4+c_1(d_2+d_3)+a_2s_1s_2$$
$$p_z=c_{23}d_4+a_2c_2+d_1$$

and the Jacobian matrix is given by :

$$J(\theta)=\begin{bmatrix} -s_{23}s_1d_4-(d_2+d_3)c_1-a_2s_1s_2 & c_1(c_{23}d_4+a_2c_2) & c_1c_{23}d_4 \\ c_1s_{23}d_4-(d_2+d_3)s_1+a_2c_1s_2 & s_1(c_{23}d_4+a_2c_2) & s_1c_{23}d_4 \\ 0 & -s_{23}d_4-a_2s_2 & -s_{23}d_4 \end{bmatrix}$$

where, $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $s_{23} = \sin(\theta_2+\theta_3)$, $c_{23} = \cos(\theta_2+\theta_3)$

The manipulator arm is initially assumed to be at rest. A reference trajectory of cubic polynomials of the following form was adopted

$$x_i=c_{0i}+c_{1i}\,t+c_{2i}\,t^2+c_{3i}\,t^3 \qquad i=1,2,3$$

where

$$c_{01}=0.03291 \quad c_{11}=0.0 \quad c_{21}=0.04279 \quad c_{31}=-0.0057$$
$$c_{02}=0.60556 \quad c_{12}=0.0 \quad c_{22}=-0.0446 \quad c_{32}=0.00588$$
$$c_{03}=-0.4916 \quad c_{13}=0.0 \quad c_{23}=0.15018 \quad c_{33}=-0.02$$

The end-effector of the manipulator is required to move from its initial position x_o =(0.033, 0.61, -0.5) meter at rest to the final position x_f =(0.39, 0.233, 0.8) meter at rest in 5 sec. These positions correspond to the movement of the joint angles from an initial position θ_o =(75, 100, 60) degree to the final position θ_f =(15, 20, 20) degree.

The simulation was performed on a VAX 11/780 computer with an inner loop sampling frequency of 500Hz. The controller gains were updated every 0.1 sec. The initial values of the controller gains were chosen to be $K_p = diag$ (30,32,30) and $K_v = diag$ (35,34,30). The desired performance of the tracking errors was specified by a natural frequency (ω_i) of 3 rad/sec and a damping ratio (ζ_i) of 1.

To illustrate the proposed approach, the control law (6.4.4) with $f_{z_{c_i}}(t)$ given by (6.5.10) was applied to the manipulator. First $f_{z_{c_i}}(t)$ was evaluated using method of pole-assignment technique given in Chapter 4. For ω_i =3, ζ_i =1, λ_{p_i} =9, and λ_{v_i} =6, the coefficients of the desired characteristic equation (4.5.3) were selected to be a_{2_1}=8744510, a_{3_1}=445840, a_{2_2}=5778170, a_{3_2}=5568250, a_{2_3}=2545000 and a_{3_3}=8238710 to give closed-loop roots in the left half of the complex plane and satisfy inequalities given in (4.5.9). These values were used to compute q_{1_i} and q_{2_i} on-line and are updated in terms of the identified parameter h_{z_i}. Figures (6.3a-c) show the system response in 3-dimensional space. The behavior of $f_{z_{m_i}}(t)$ is shown in Figs. (6.4a-c). The estimated diagonal elements of the H_z matrix are shown in Figs. (6.5a-c) and the variation of (k_{p_i} , k_{v_i}) are shown in Figs. (6.6a-c). Figures (6.7a-c) show the response of the end-effector in 3-dimensional space under payload variation. At t =1 sec., the mass of m_3 was changed to 100 kg and the inertia was changed to I_{3zz}=4.8 and I_{3xx}=I_{3yy}=8. These results show that the control scheme is robust and the system tracks the desired trajectory in the presence of these variations.

For the same example, the method of bounded uncertainty, described in Chapter 5, was also used to evaluate $f_{x_{c_i}}(t)$. The bound $\mu_{i_{max}}$ was found to be equal 0.43. The system response is shown in Figs. (6.8a-c) and it shows a good tracking performance by the end-effector for the reference trajectory. The behavior of $f_{x_{m_i}}(t)$ is shown in Figs. (6.9a-c). The estimated values of the diagonal elements of the matrix H_x are shown in Figs. (6.10a-c). The variation of the gains K_p and K_v are shown in Figs. (6.11a-c). The robustness of the proposed controller under payload variations was tested by changing the mass of the 3rd link from 36.2 kg to 100.0kg at $t=1$ sec. The inertia was changed to $I_{3xx}=I_{3yy}=8$ and $I_{3zz}=4.8$. For the same reference trajectory as before, the system gave good tracking response as shown in Figs. (6.12a-c).

The behavior of the adaptive control scheme was also examined for the following trajectory

$$p_x = 0.5\ sin(\frac{\pi}{4}t)$$
$$p_y = 0.61\ cos(\frac{\pi}{4}t)$$
$$p_z = -0.5\ cos(\frac{\pi}{4}t)$$

The response of the end-effector is shown in Figs. (6.13a-c) and Figs. (6.14a-c). It is seen that the reference trajectory is tracked very closely.

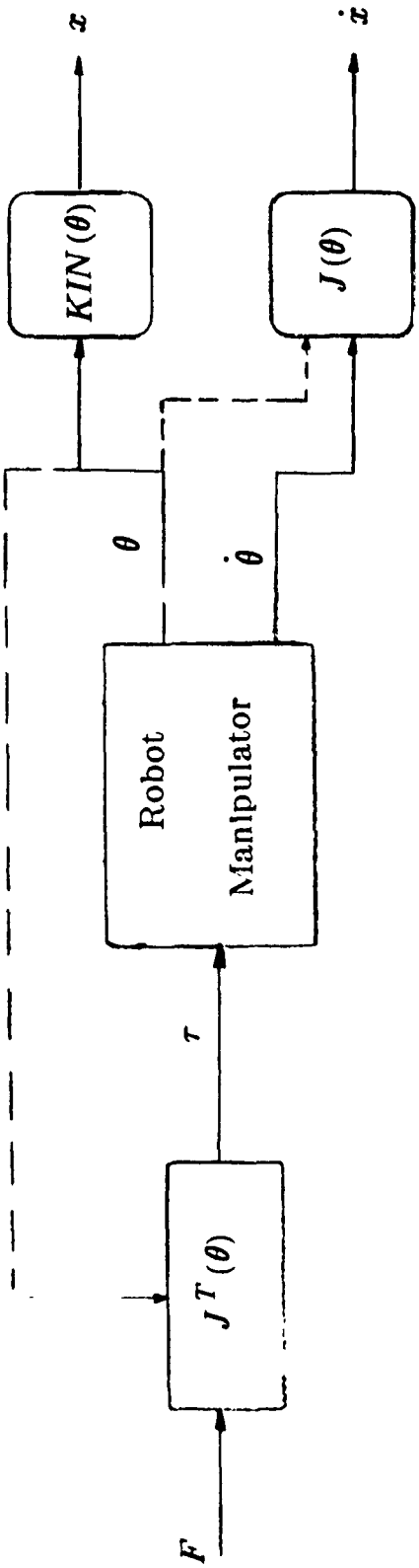

Fig.(6.1) Cartesian manipulator model

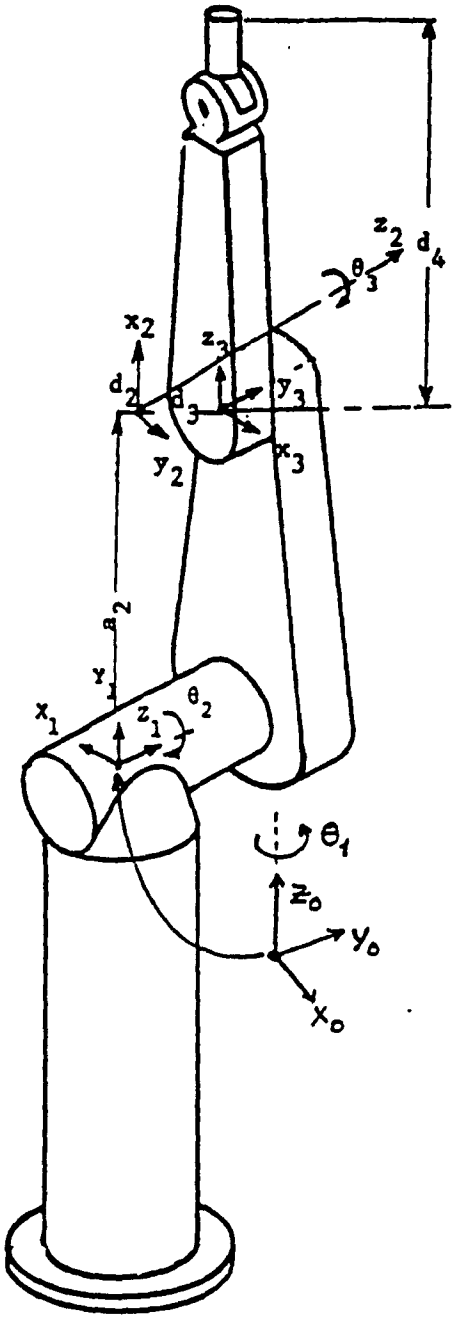

Fig.(6.2) A Three-link PUMA-type Manipulator

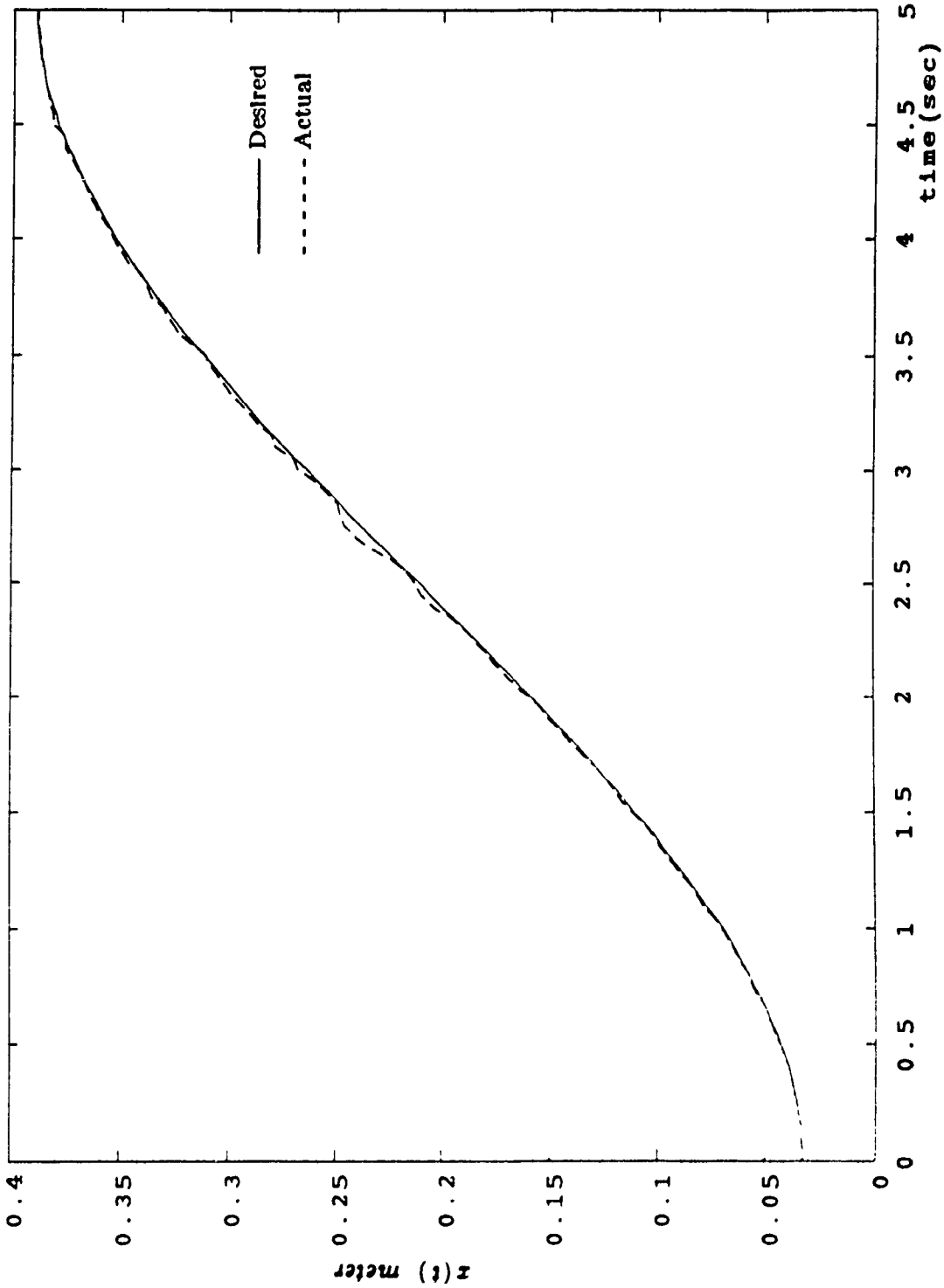

Fig.(6.3a) Response of the end-effector in x-direction

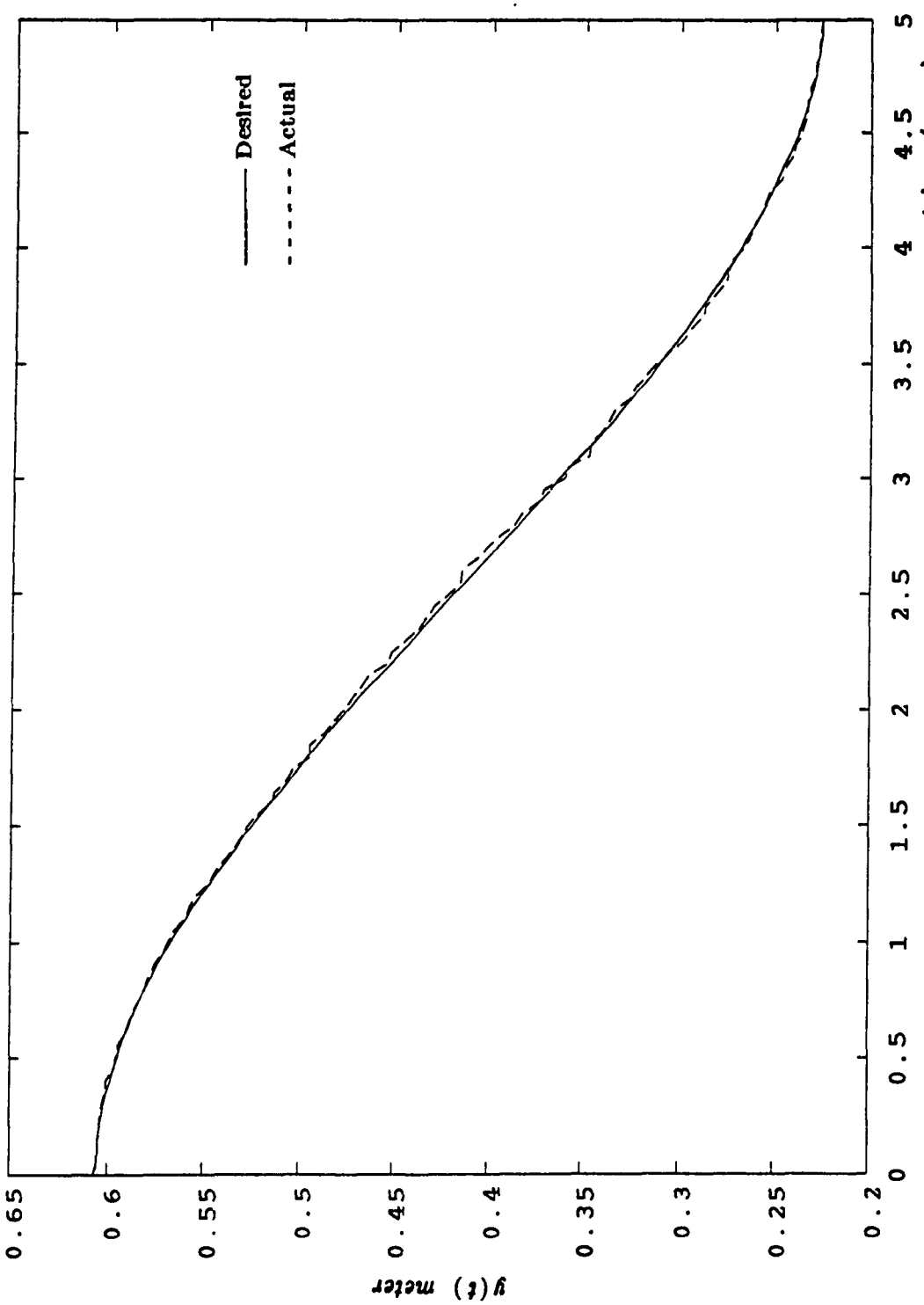

Fig.(6.3b) Response of the end-effector in y-direction

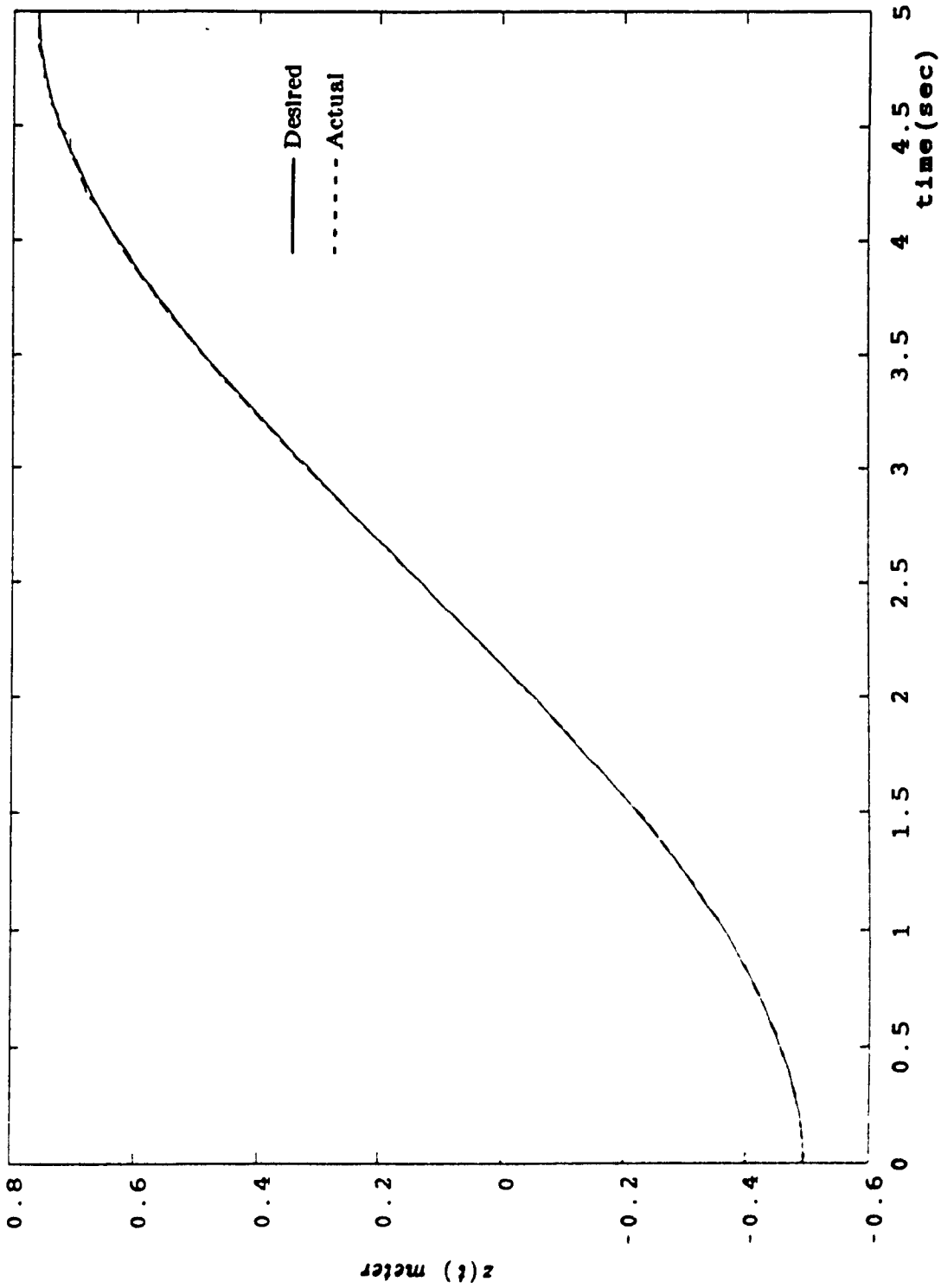

Fig.(6.3c) Response of the end-effector in z-direction

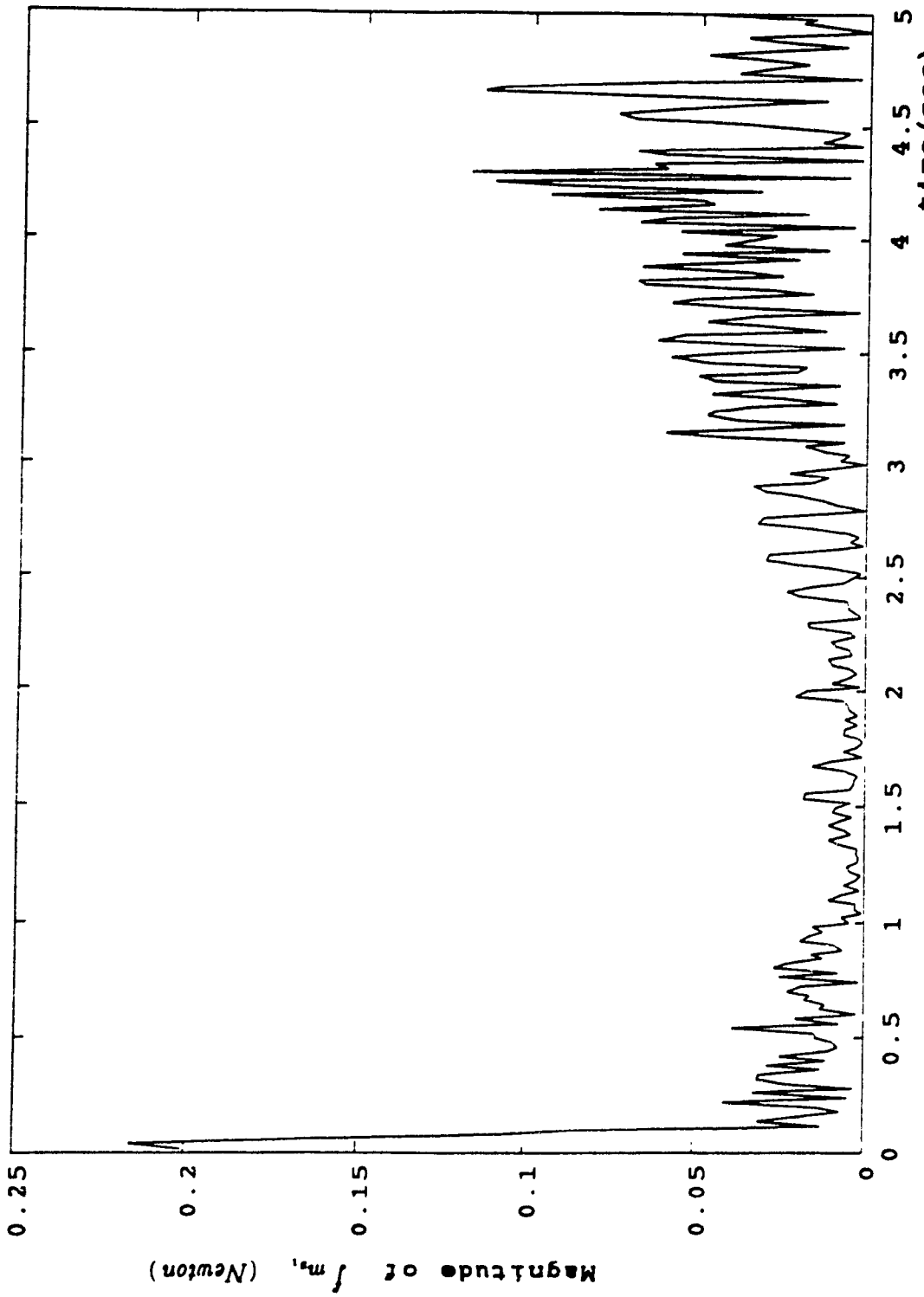

Fig.(6.4a) Behaviour of f_{m_1} for the closed-loop system

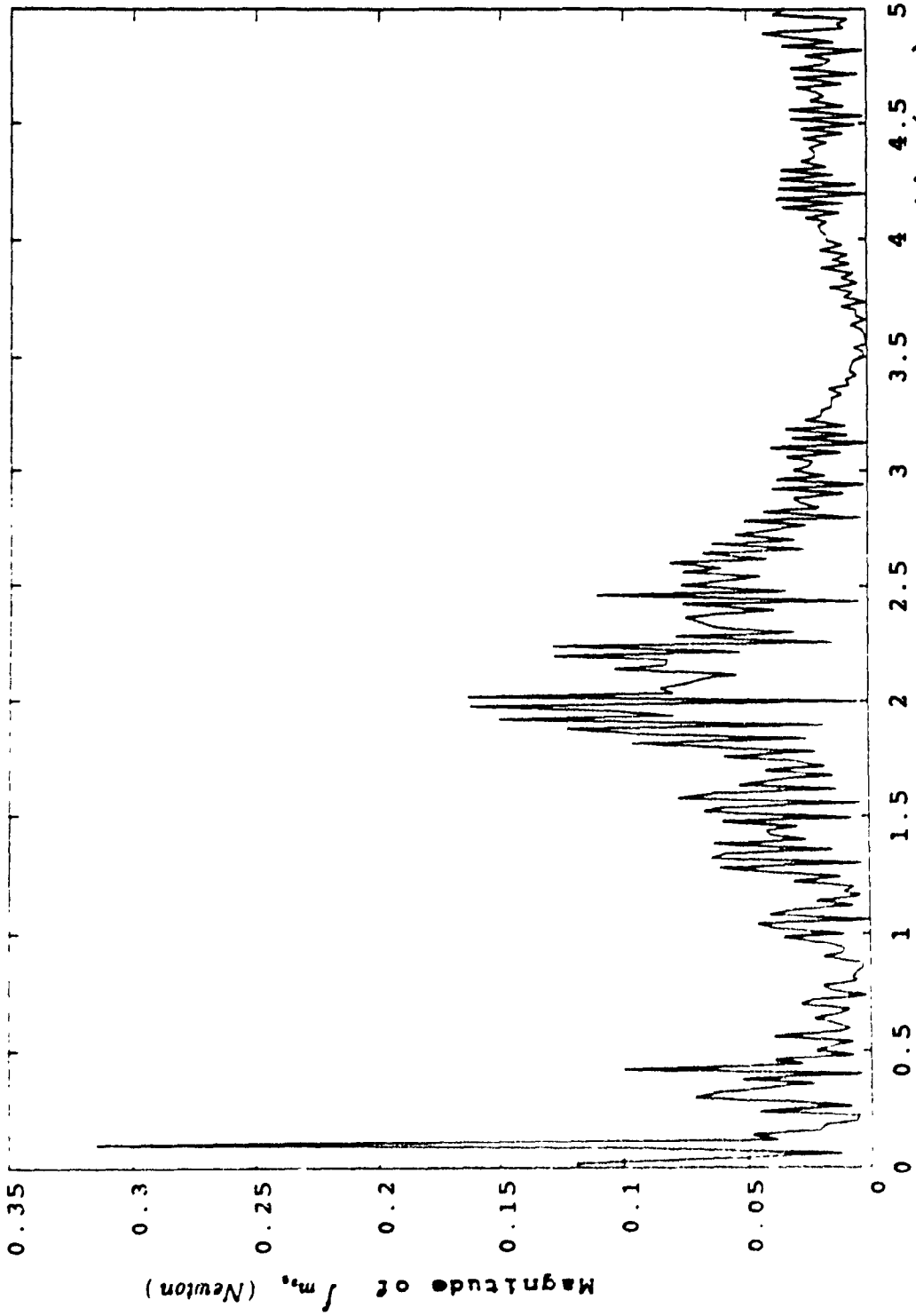

Fig.(6.4b) Behaviour of f_{m_s} for the closed-loop system

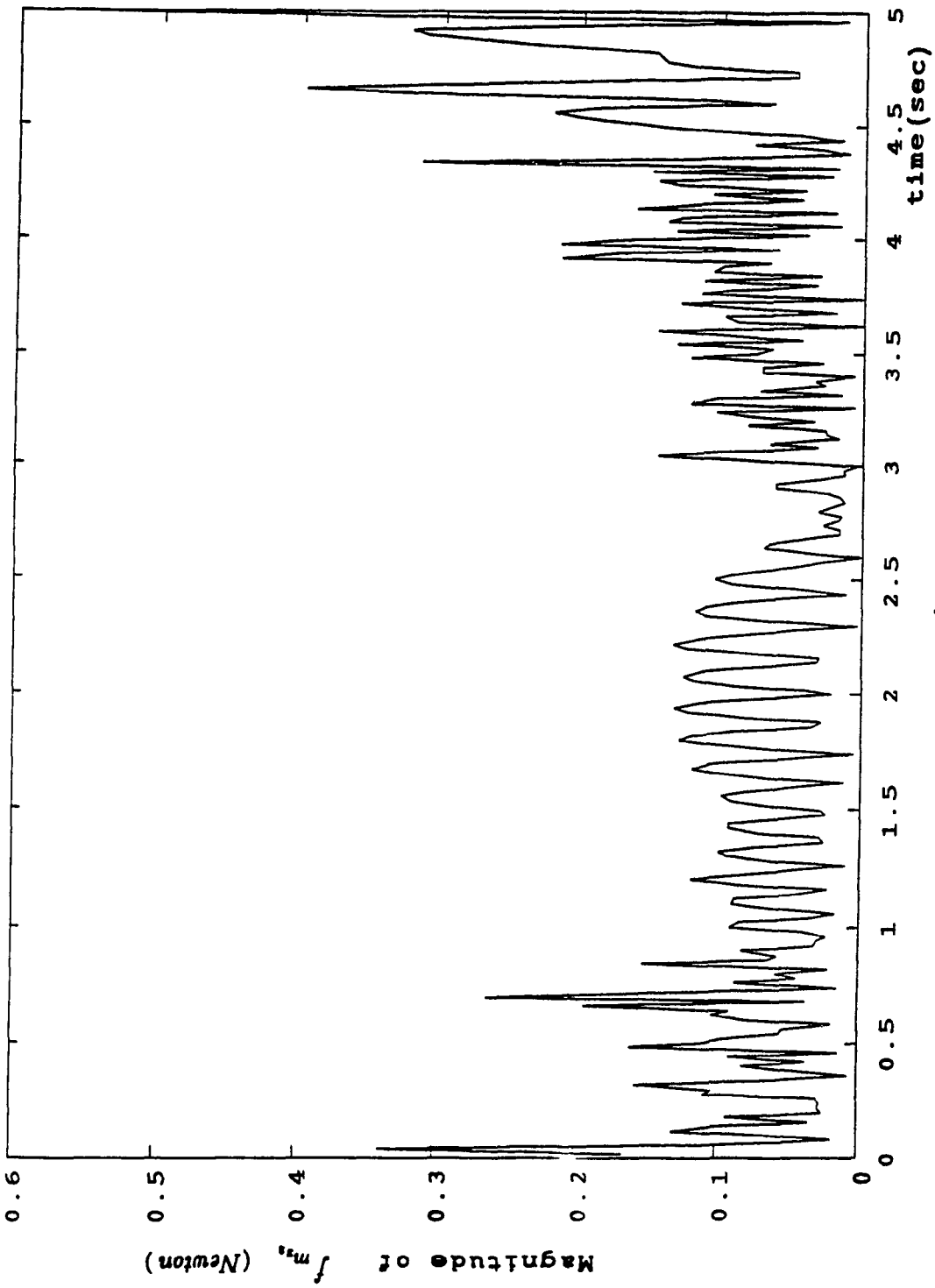

Fig.(6.4c) Behaviour of f_{m_z} for the closed-loop system

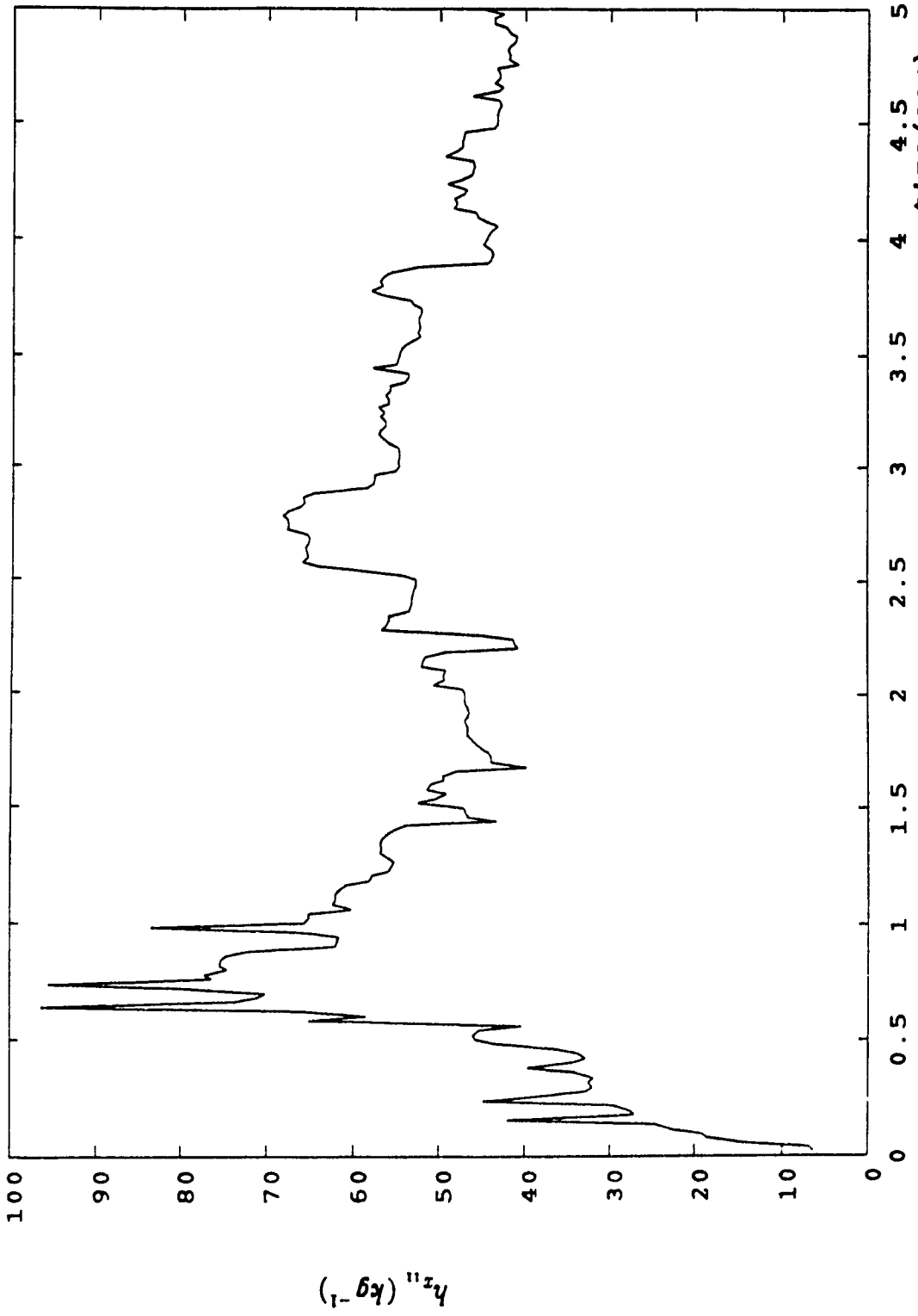

Fig.(6.5a) Estimation of the first diagonal element of H_x

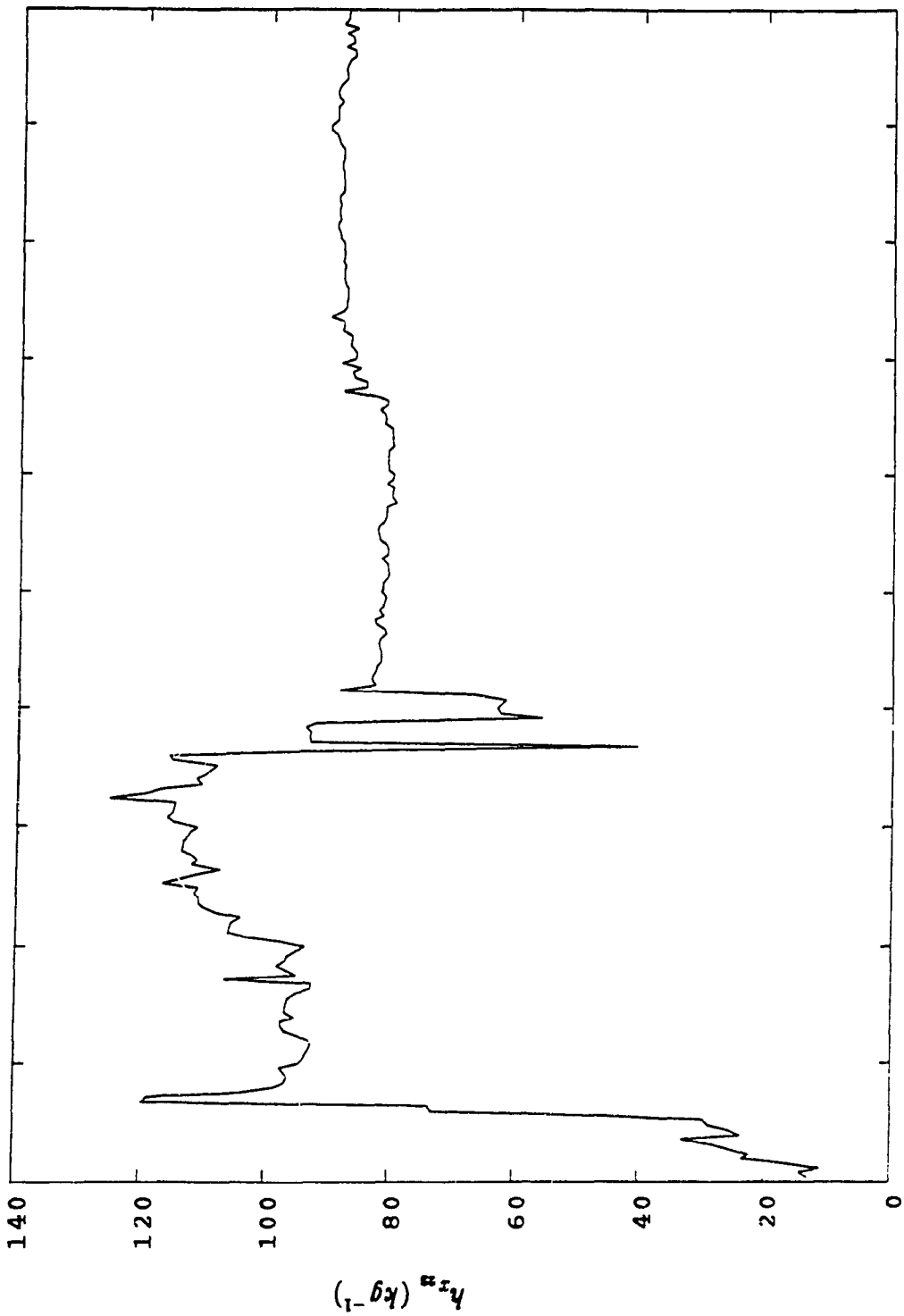

Fig.(6.5b) Estimation of the second diagonal element of H_x

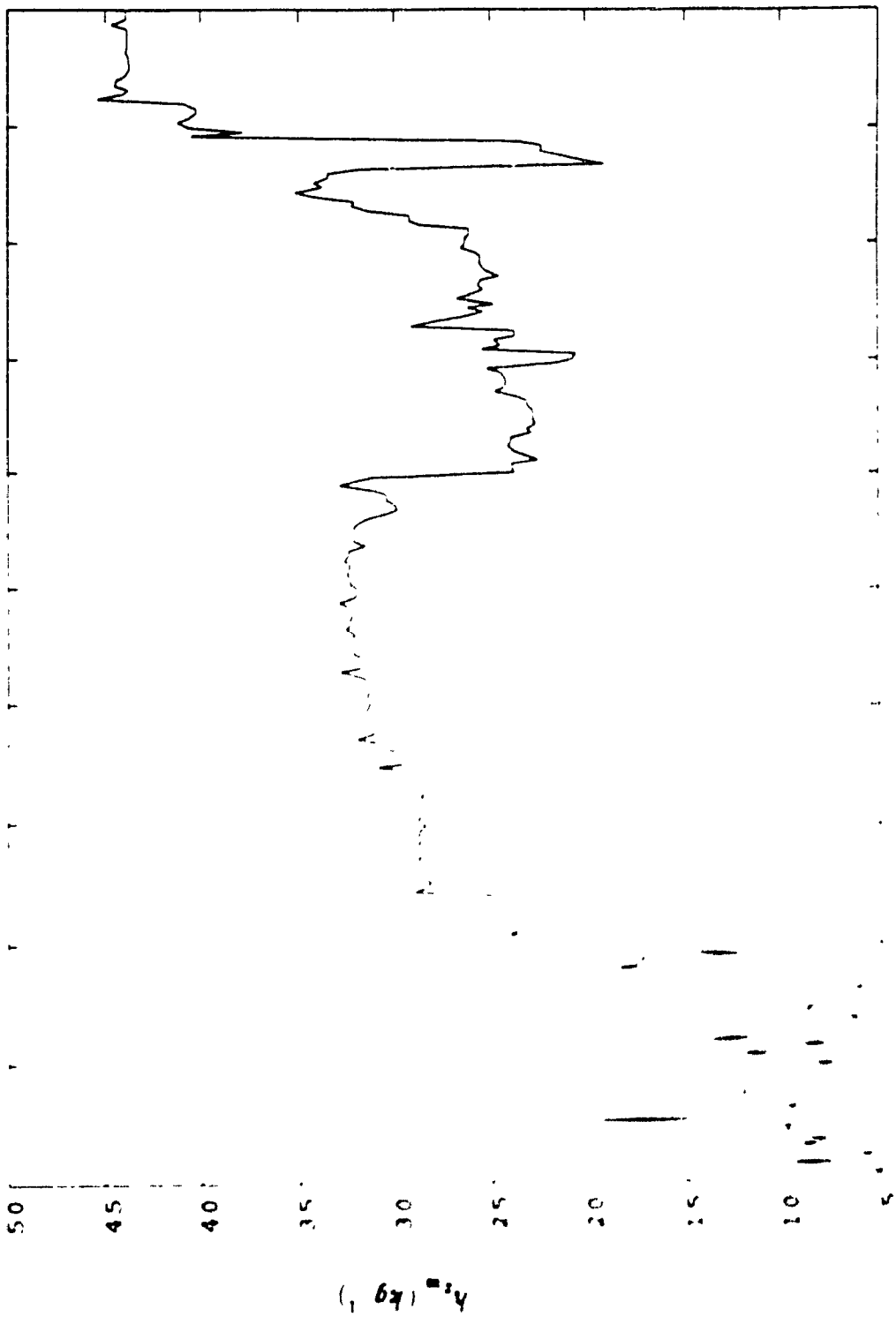

Fig (6.5c) Estimation of the third diagonal element of H_z

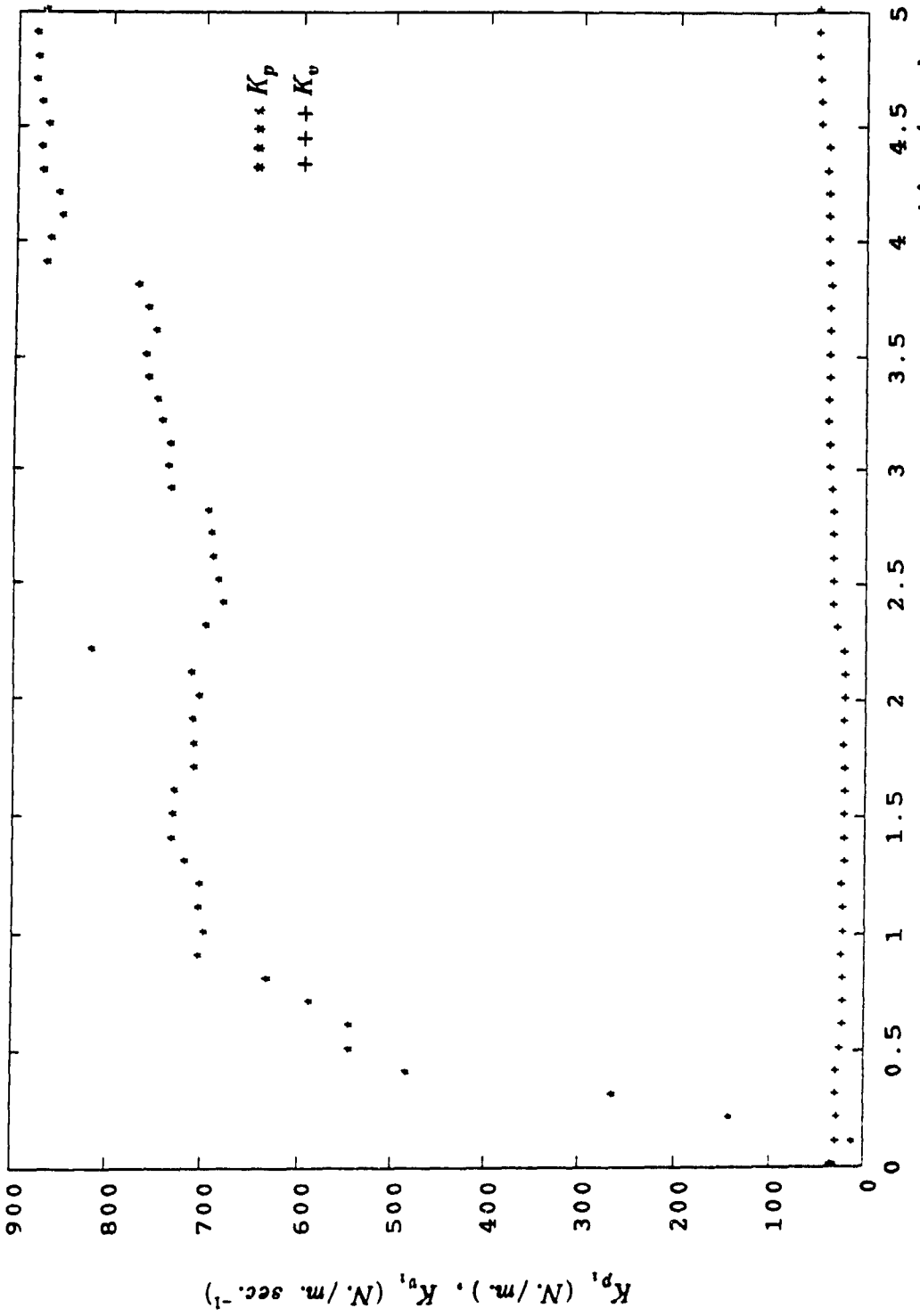

Fig.(6.6a) Variation of the adapted gains (K_{p_i}, K_{v_i})

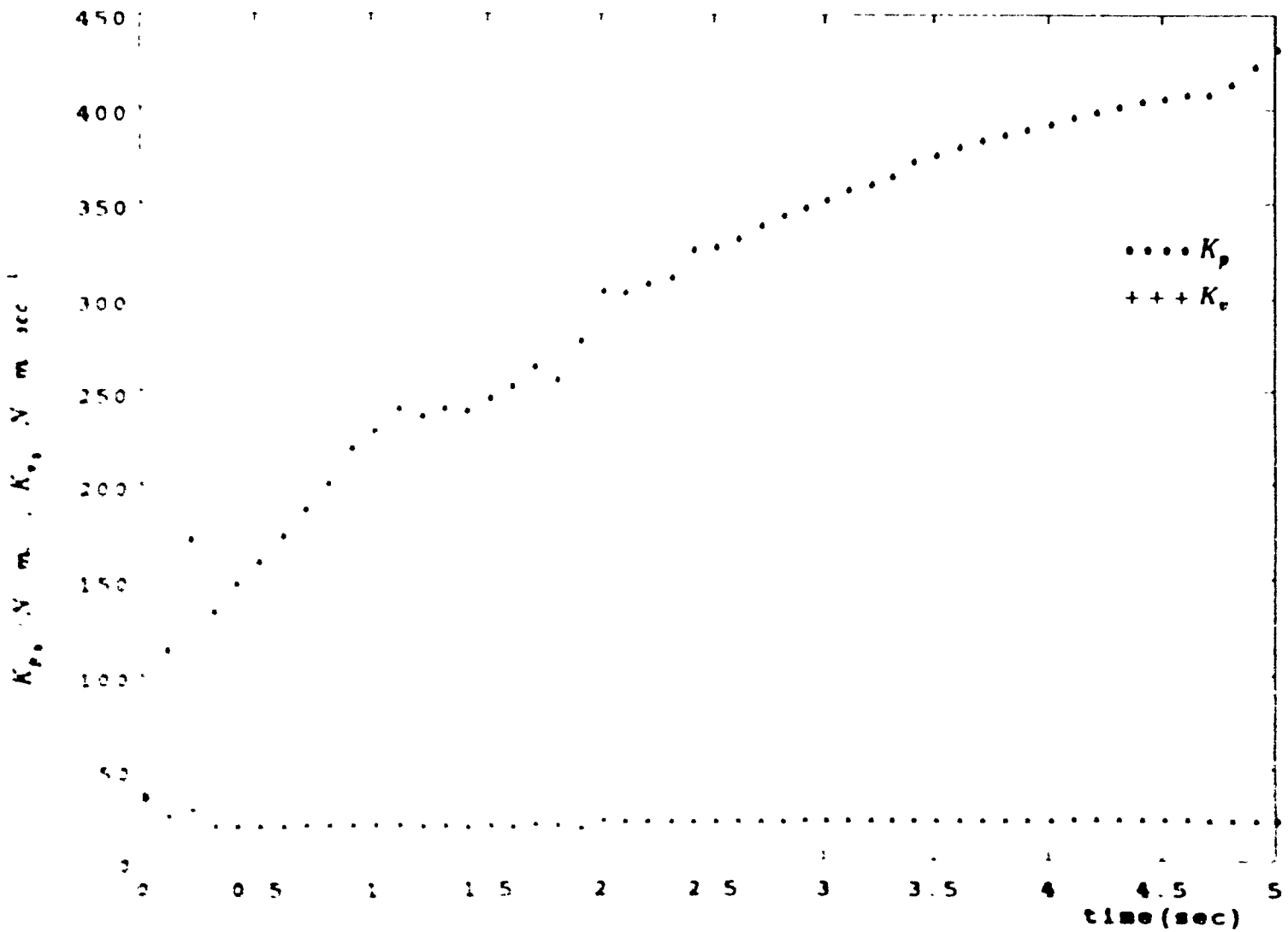

Fig (6.6b) Variation of the adapted gains (K_p, K_v)

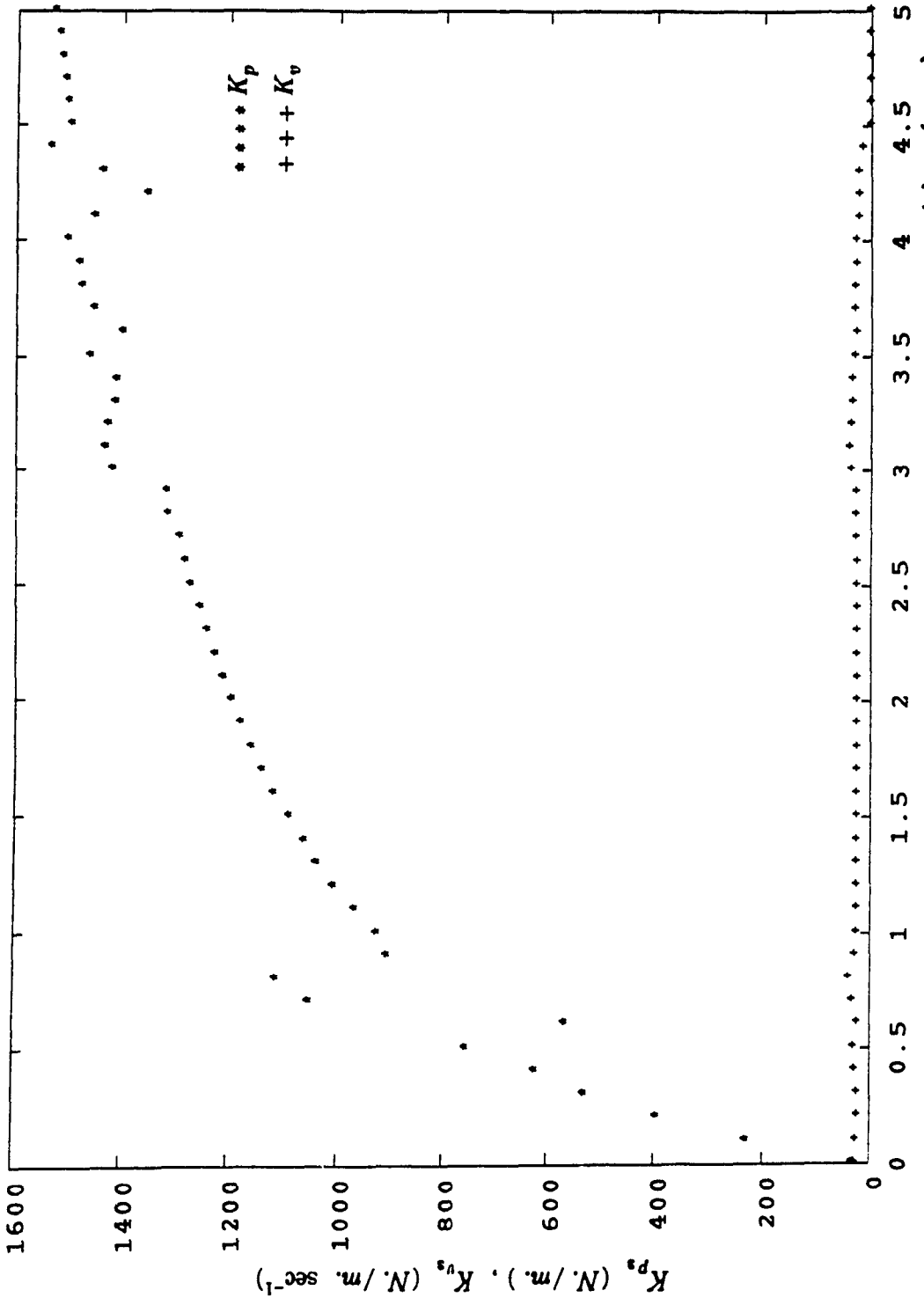

Fig.(6.6c) Variation of the adapted gains (K_{p_s}, K_{v_s})

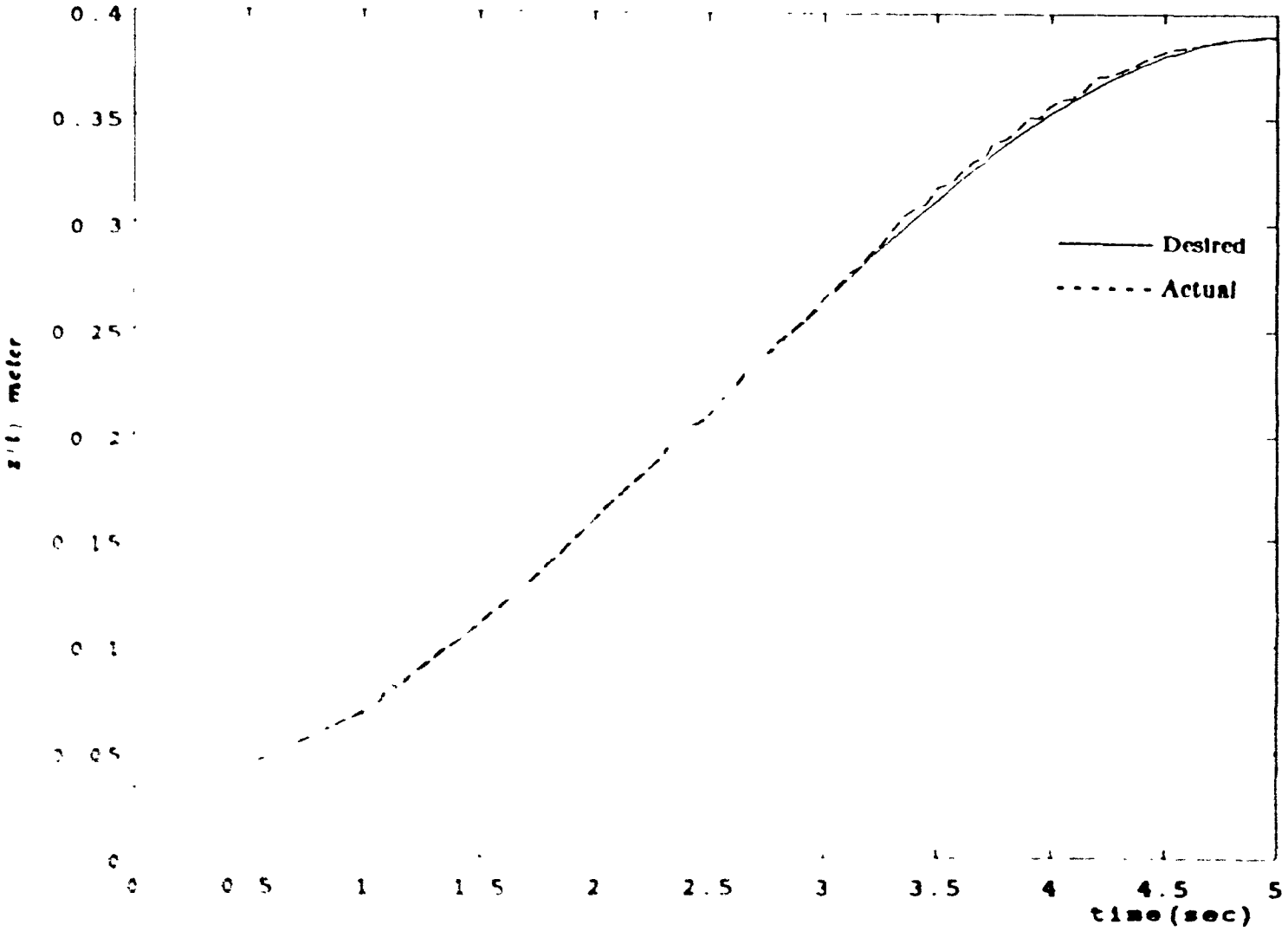

Fig (6.7a) Response in the x-direction under payload change

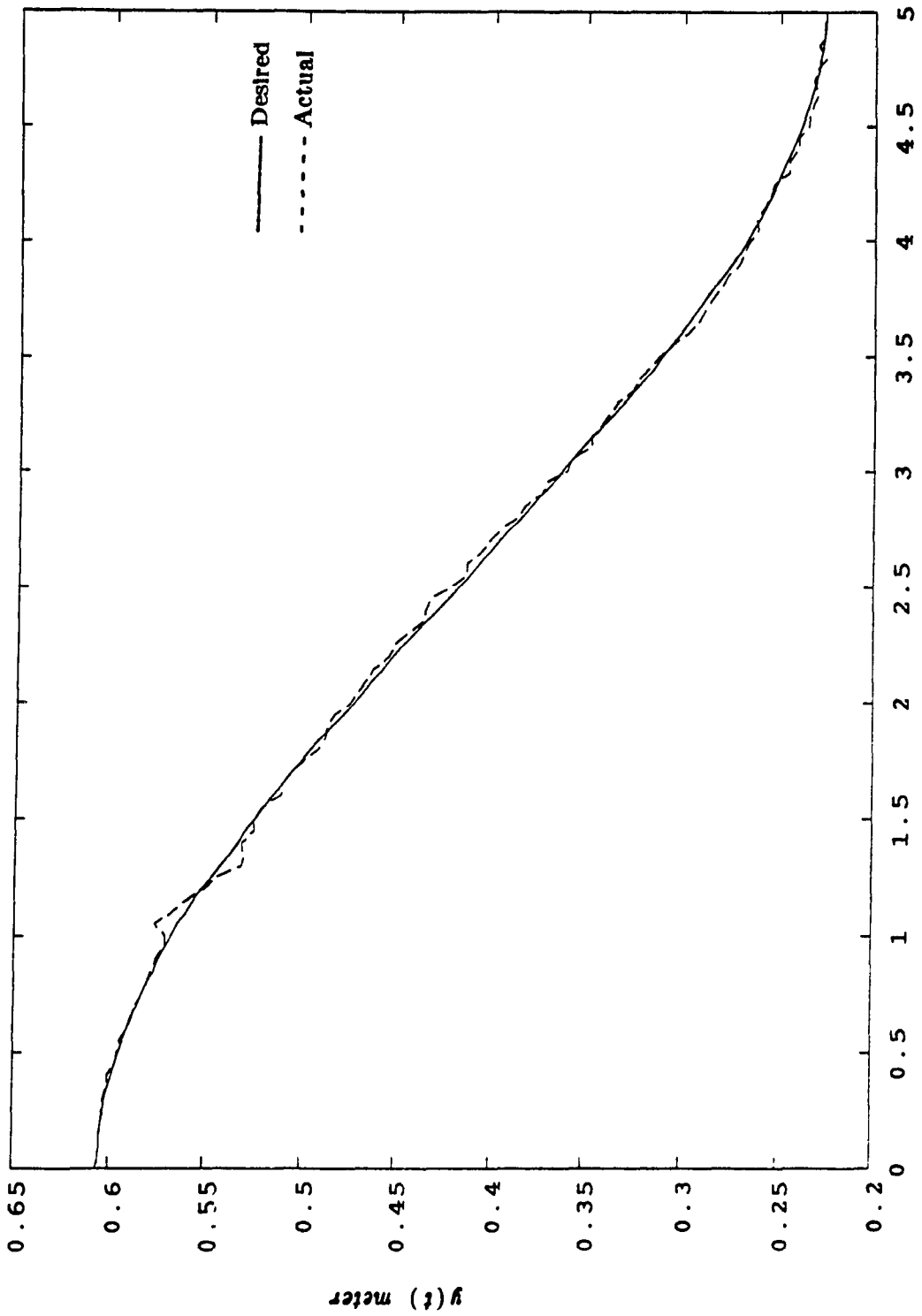

Fig.(6.7b) Response in the y-direction under payload change

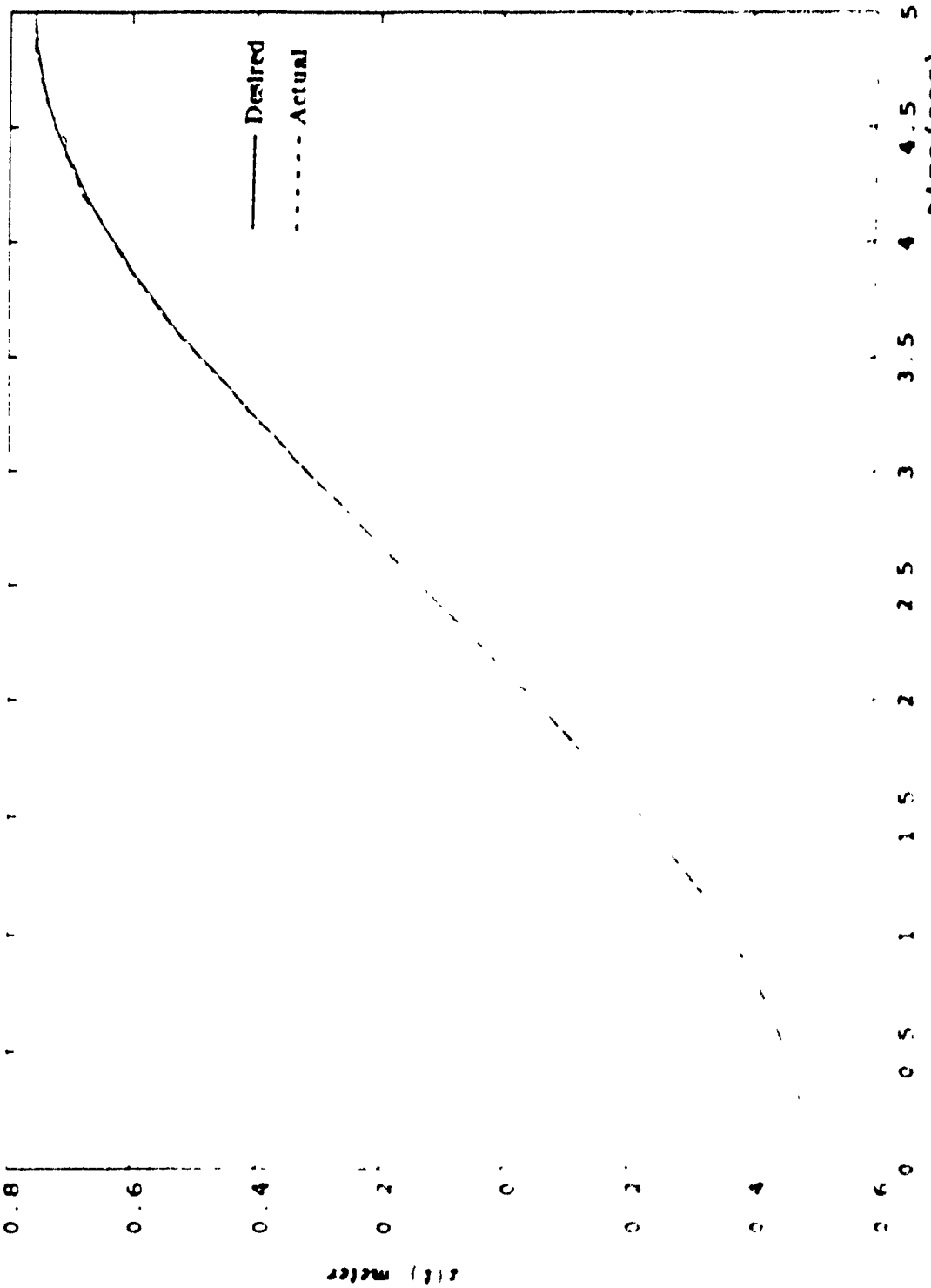

Fig (6.7c) Response in the z-direction under payload change

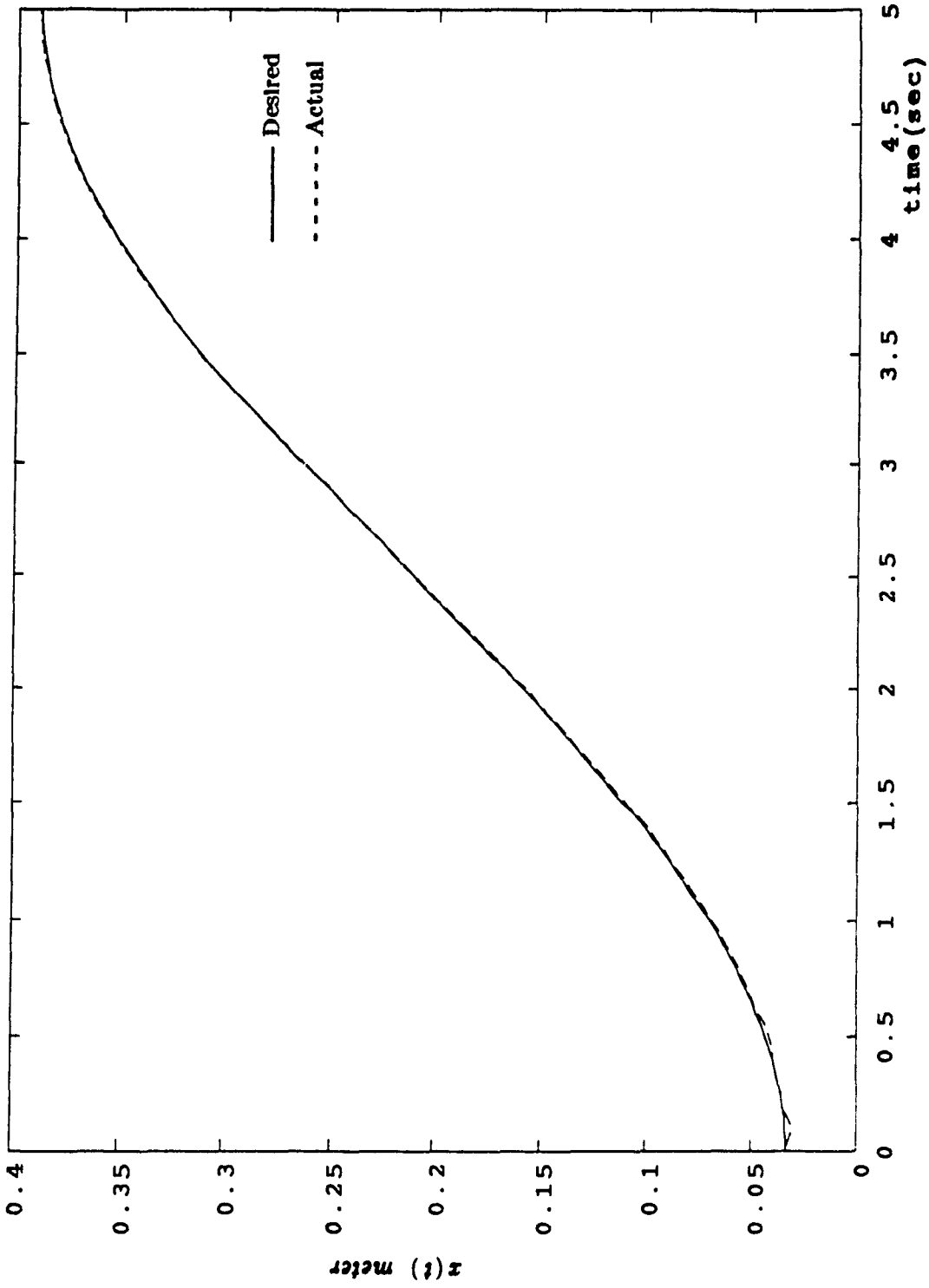

Fig.(6.8a) Response of the end-effector in x-direction

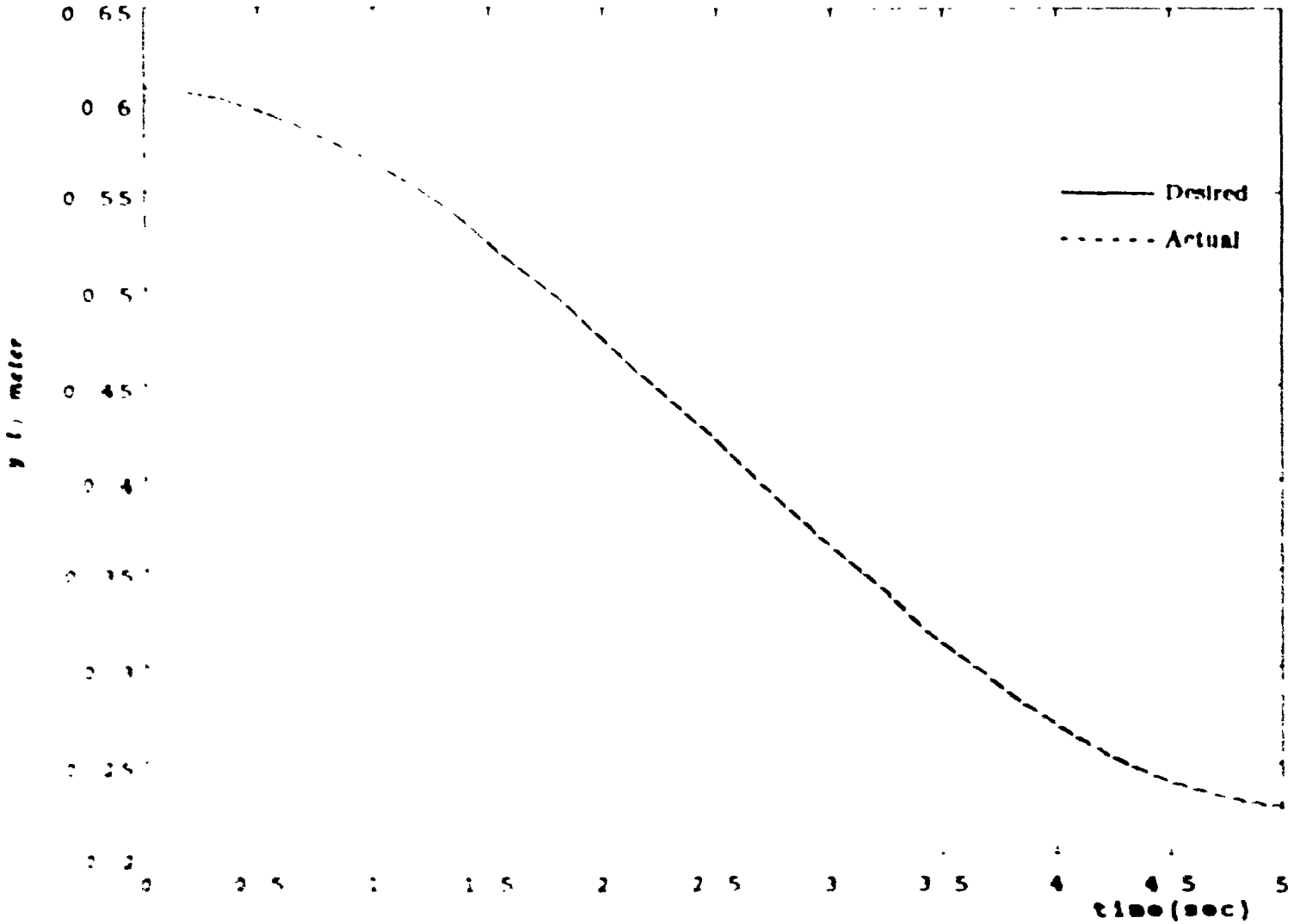

Fig (5.8b) Response of the end-effector in y-direction

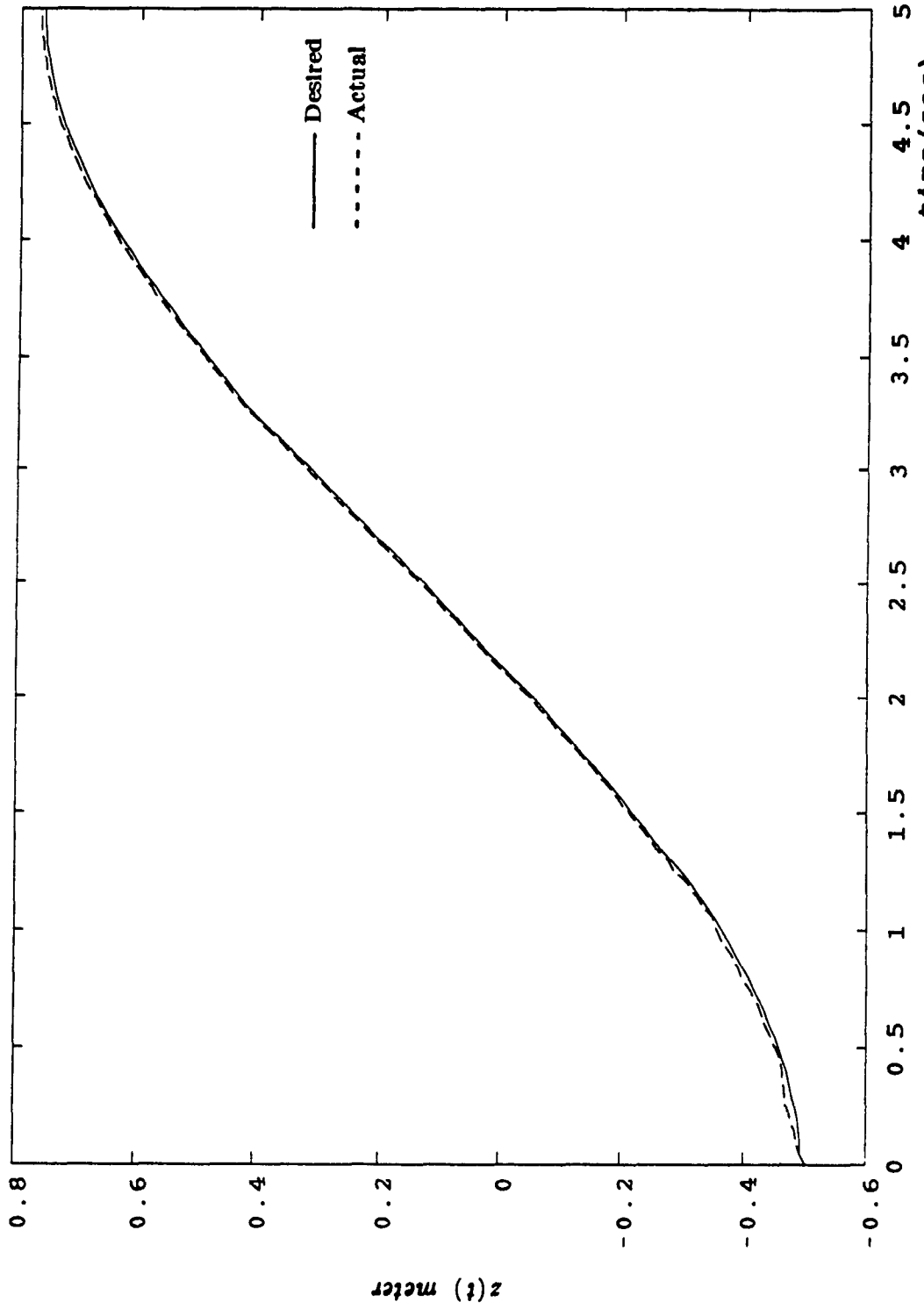

Fig.(6.8c) Response of the end-effector in z-direction

Fig.(6.9a) Behaviour of f_{m_1} for the closed-loop system

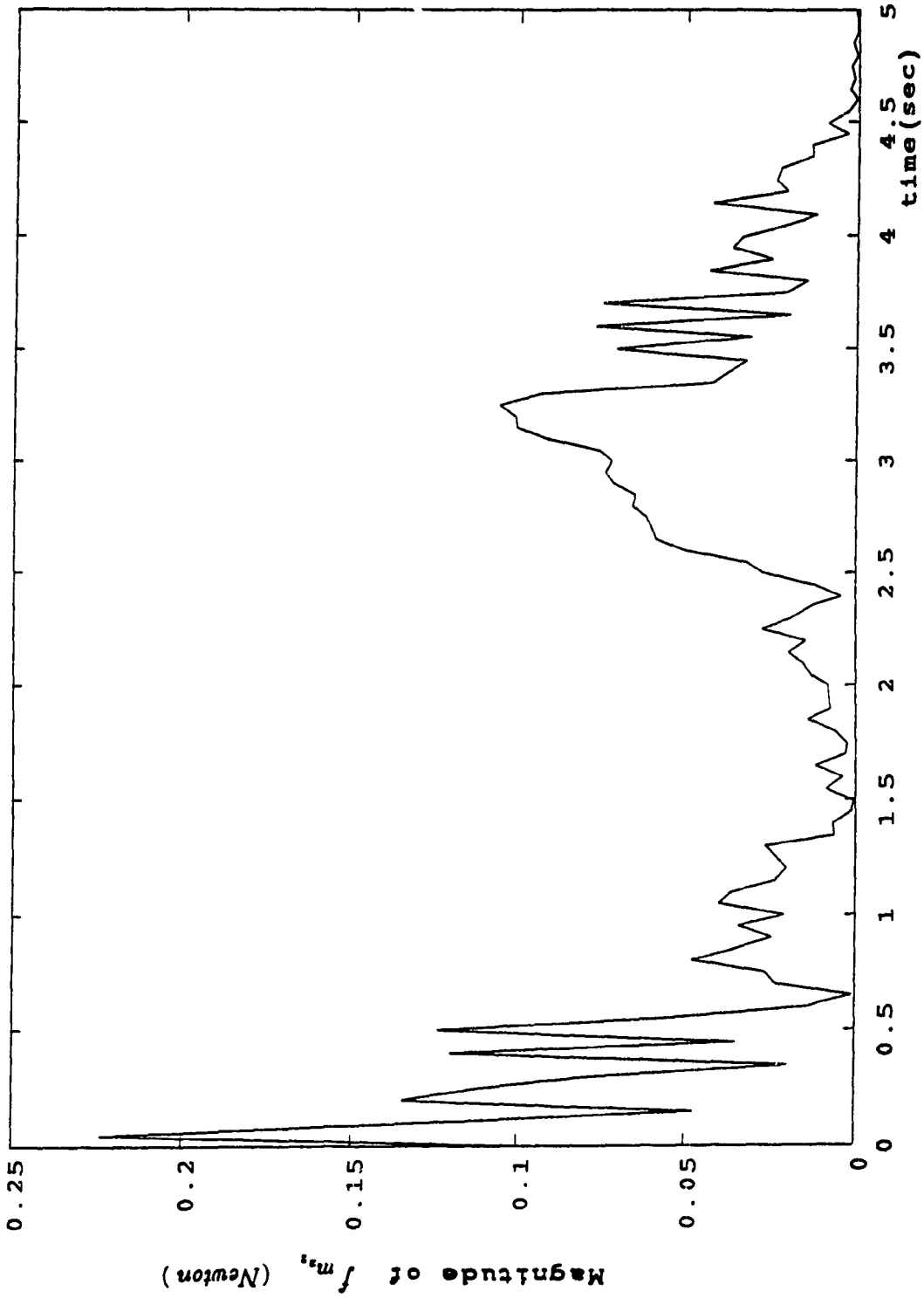

Fig.(6.9b) Behaviour of f_{m_2} for the closed-loop system

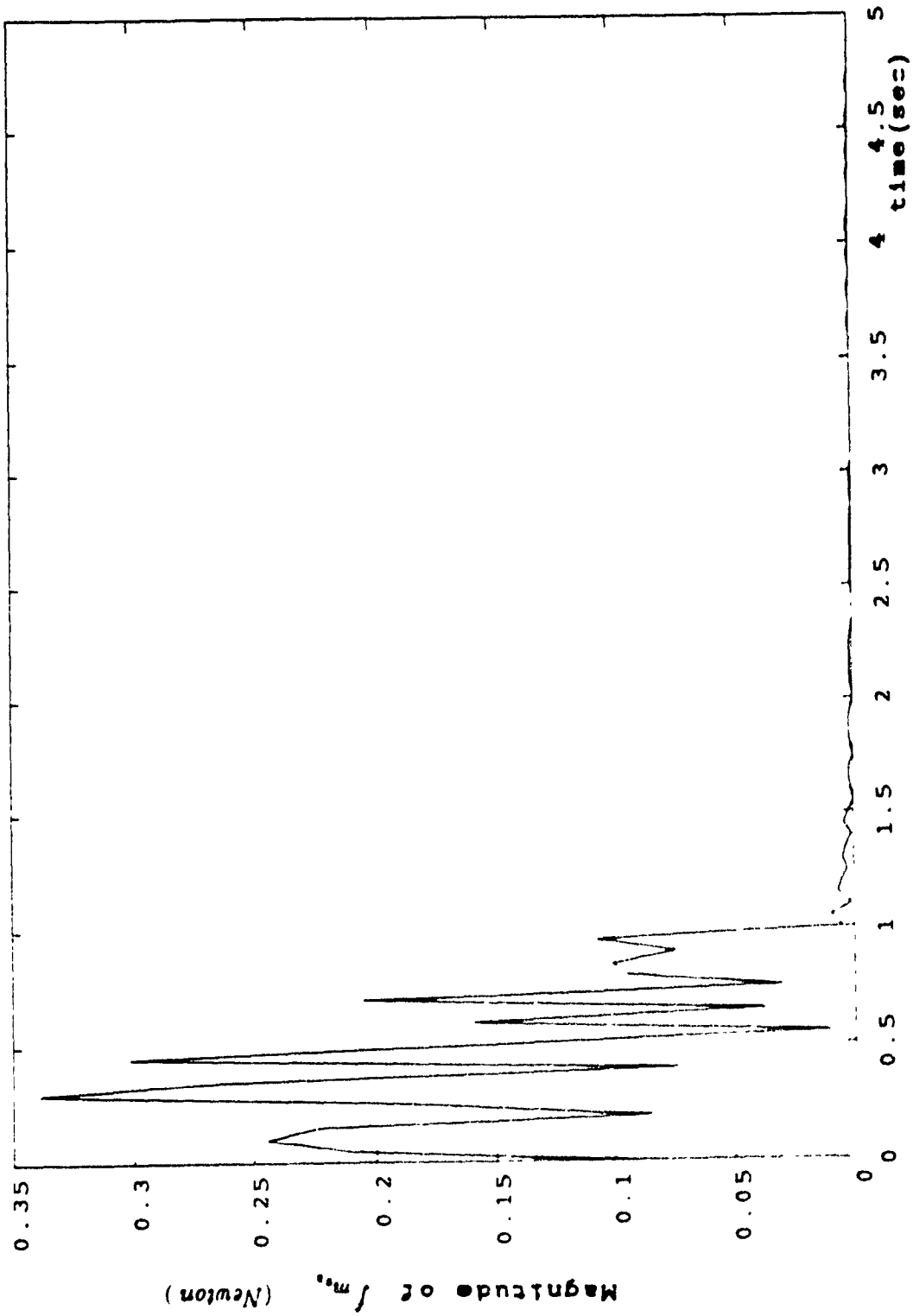

Fig.(6.9c) Behaviour of $f_{m_{12}}$ for the closed-loop system

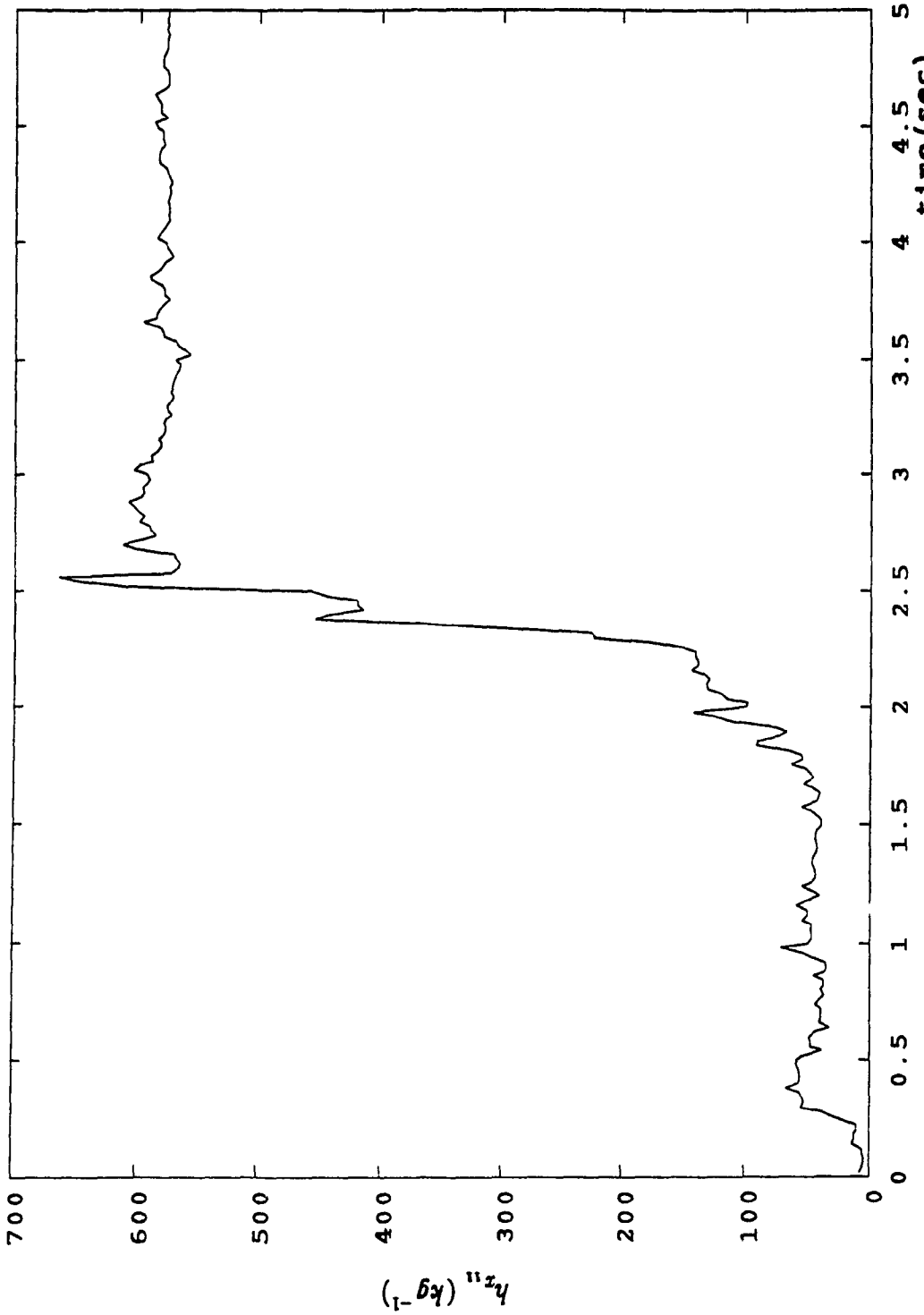

Fig.(6.10a) Estimation of the first diagonal element of H_x

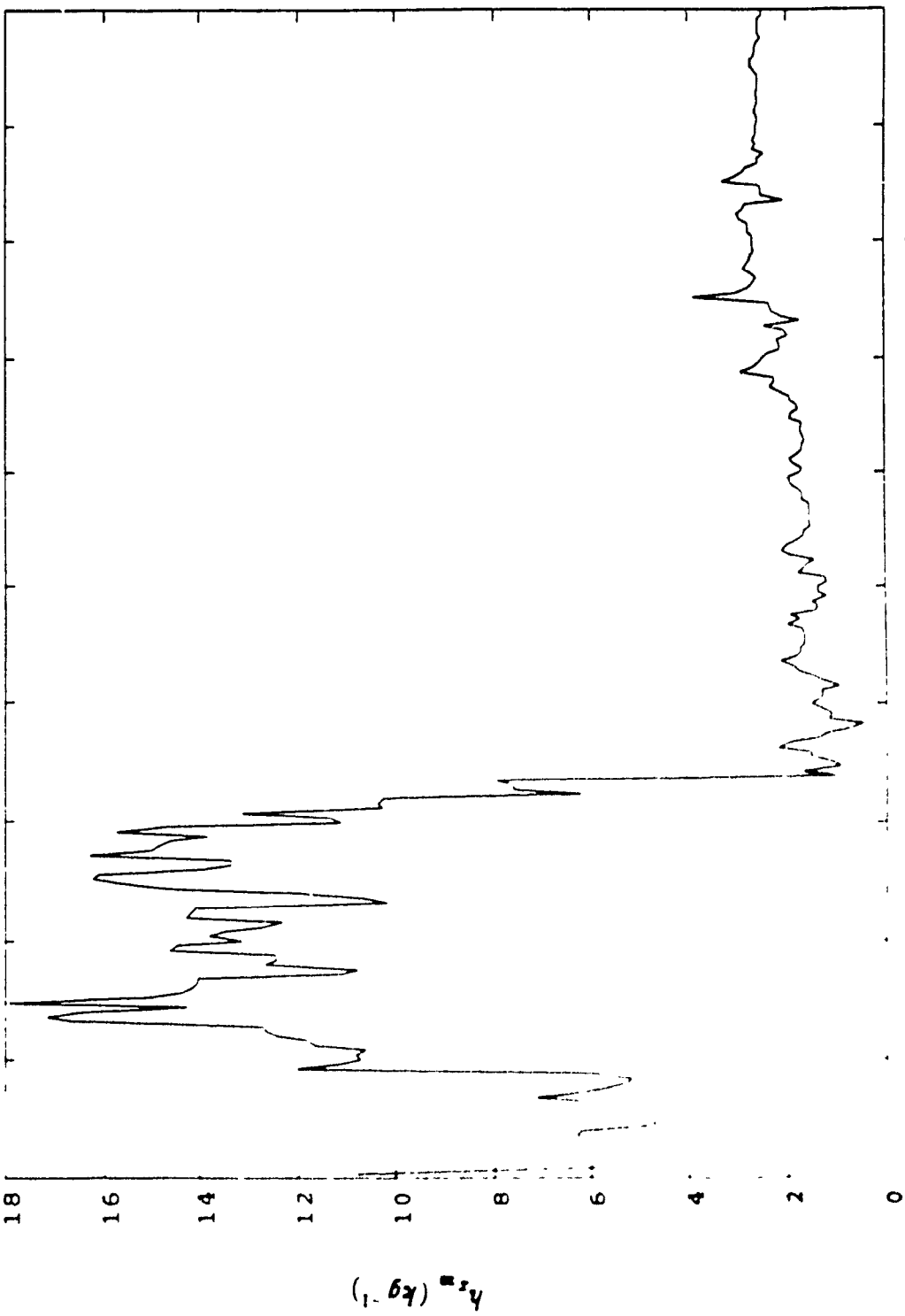

Fig.(6.10b) Estimation of the second diagonal element of H_z

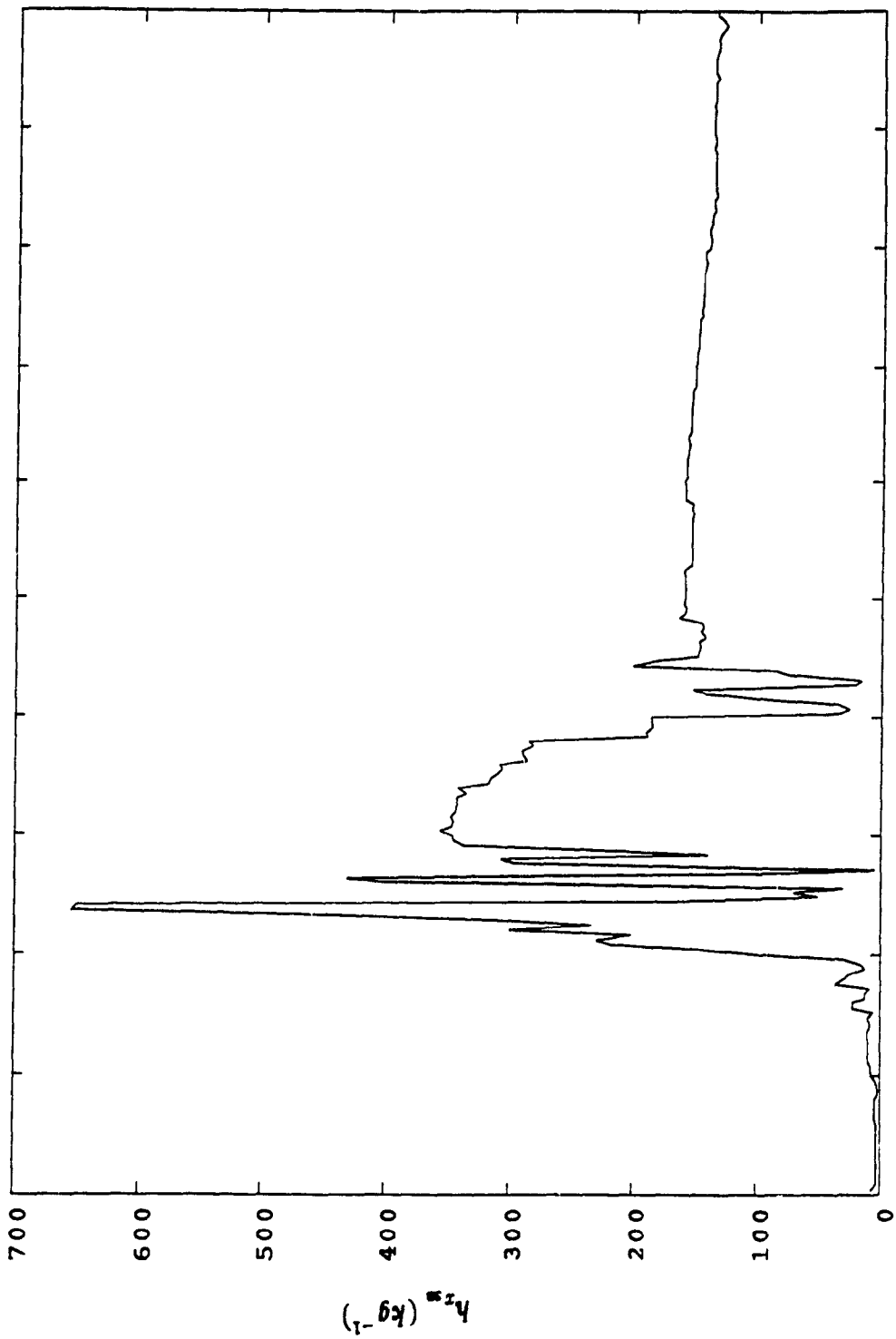

Fig.(6.10c) Estimation of the third diagonal element of H_z

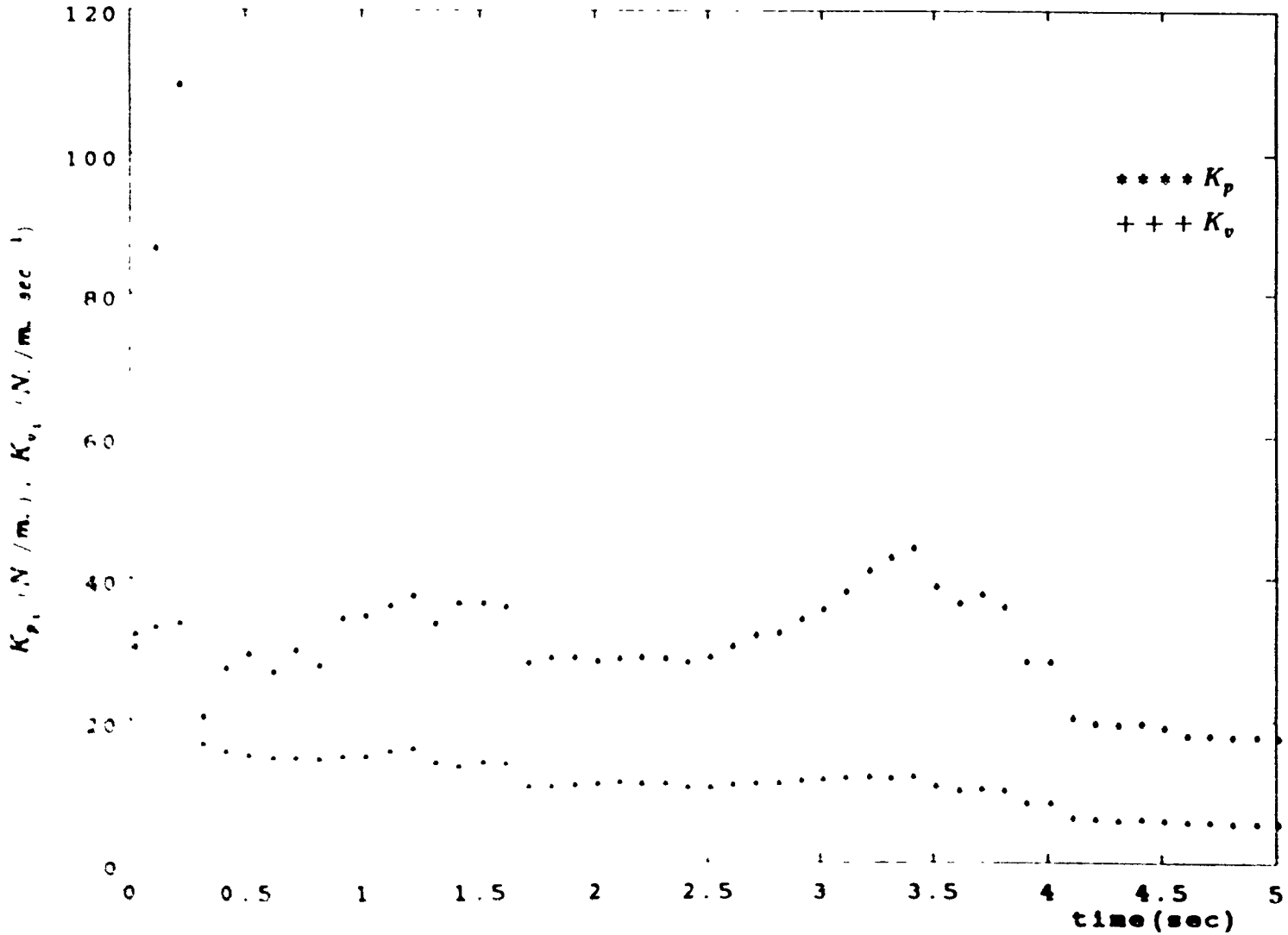

Fig (6 11a) Variation of the adapted gains (K_{p_i}, K_{v_i})

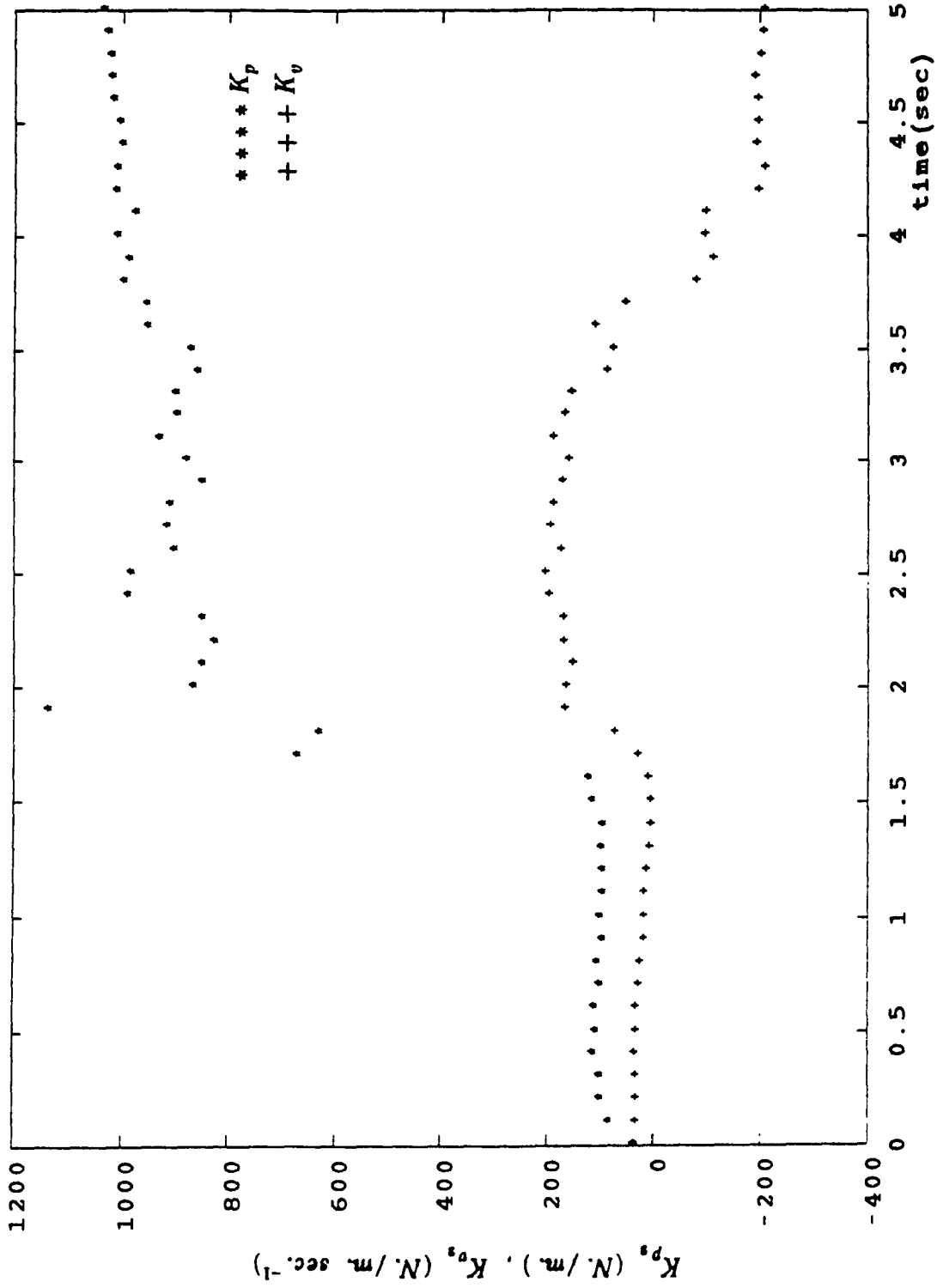

Fig.(6.11b) Variation of the adapted gains (K_{p_s}, K_{v_s})

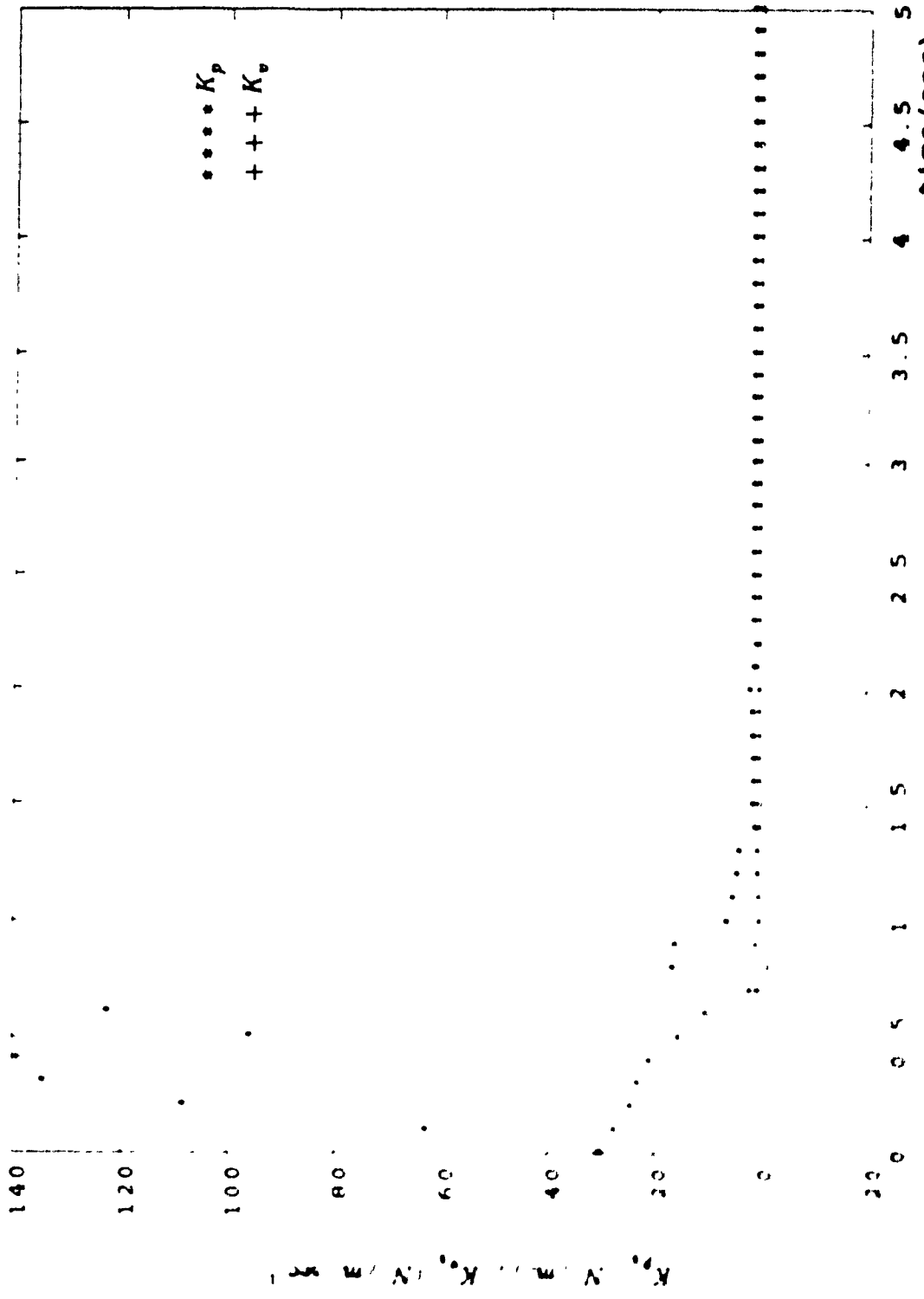

Fig (4.11c) Variation of the adapted gains (K_p, K_v)

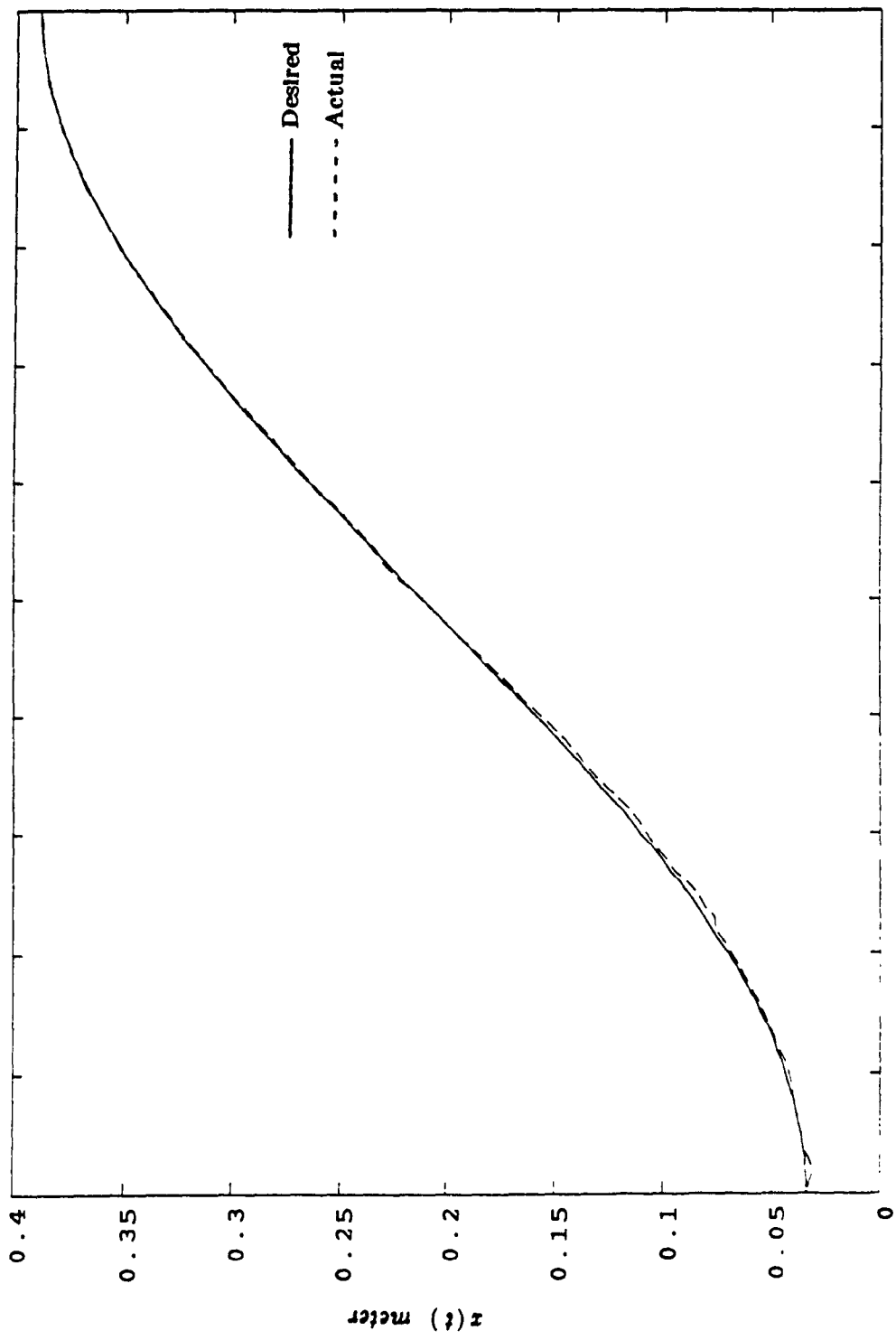

Fig.(6.12a) Response in the x-direction under payload change

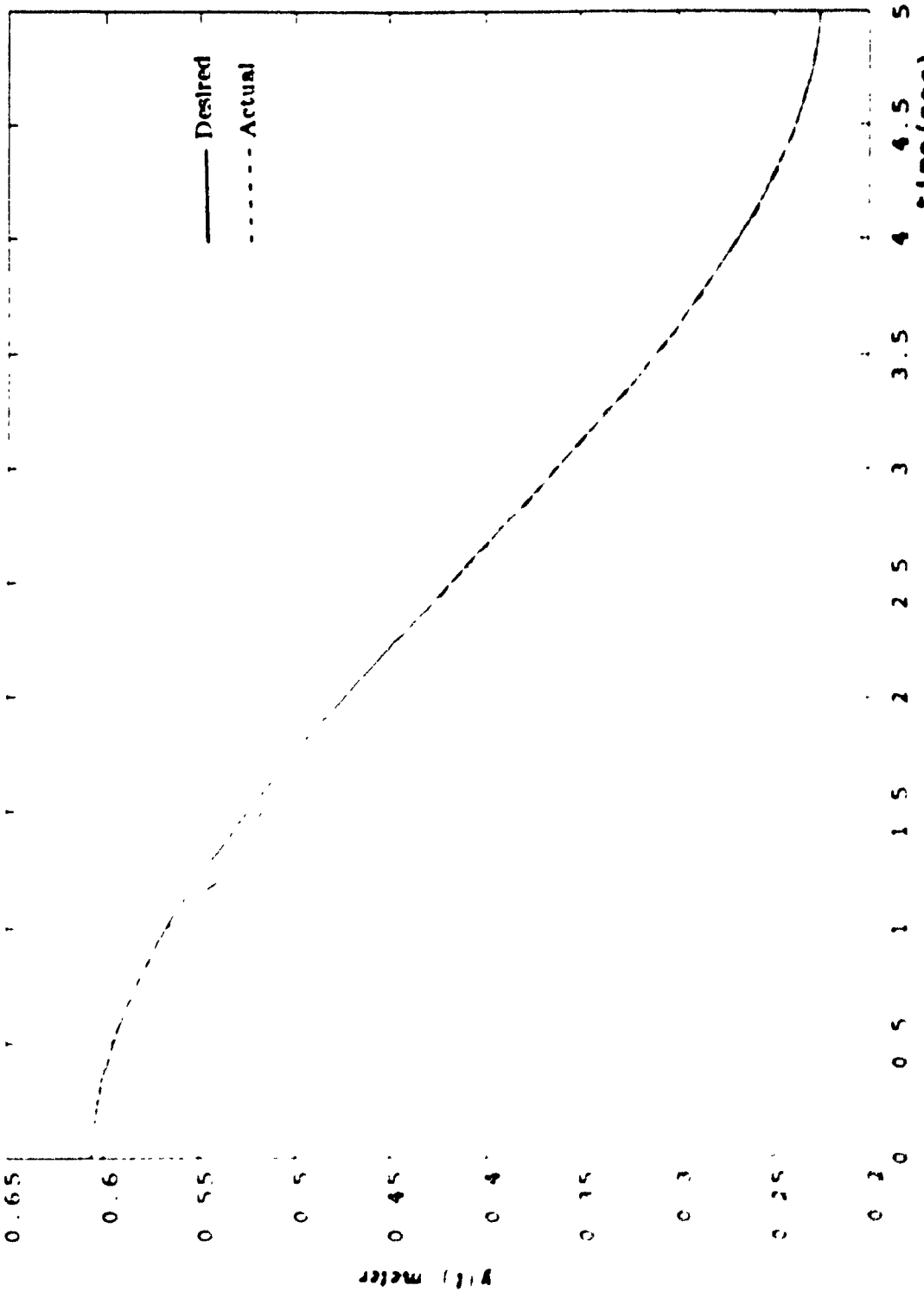

Fig (6.12b) Response in the y-direction under payload change

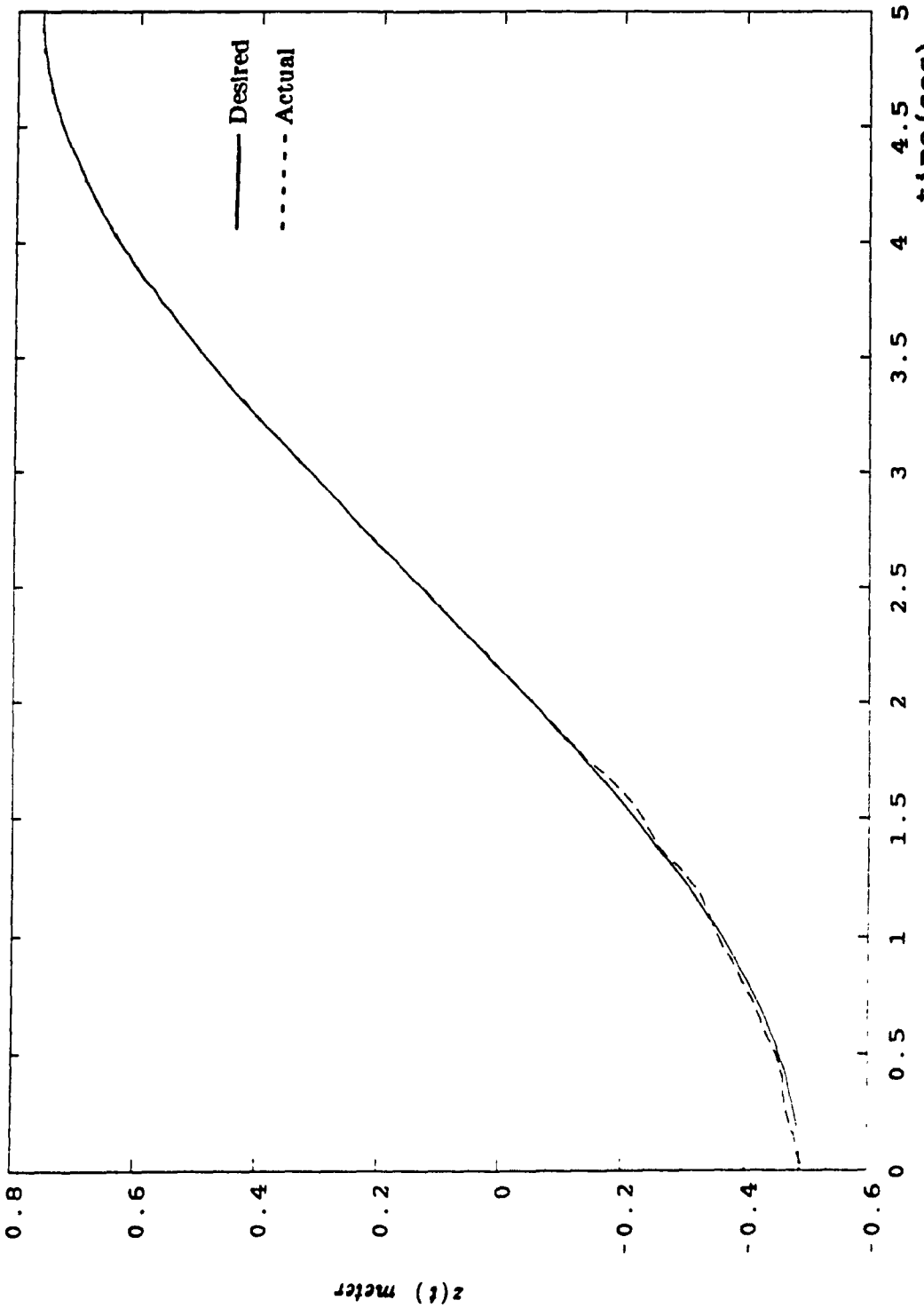

Fig.(6.12c) Response in the z-direction under payload change

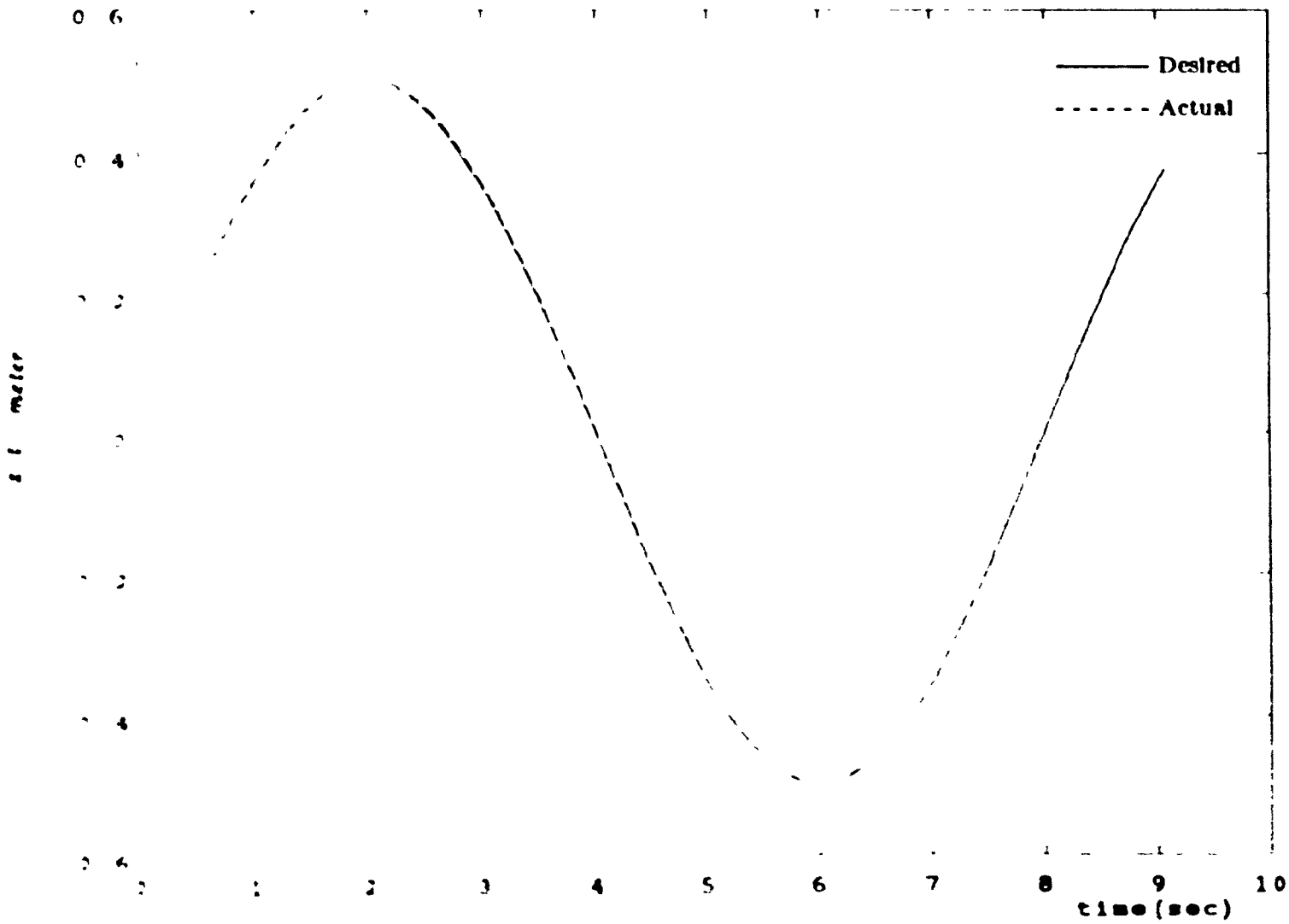

Fig (6 :1a) Response of the end-effector in x-direction

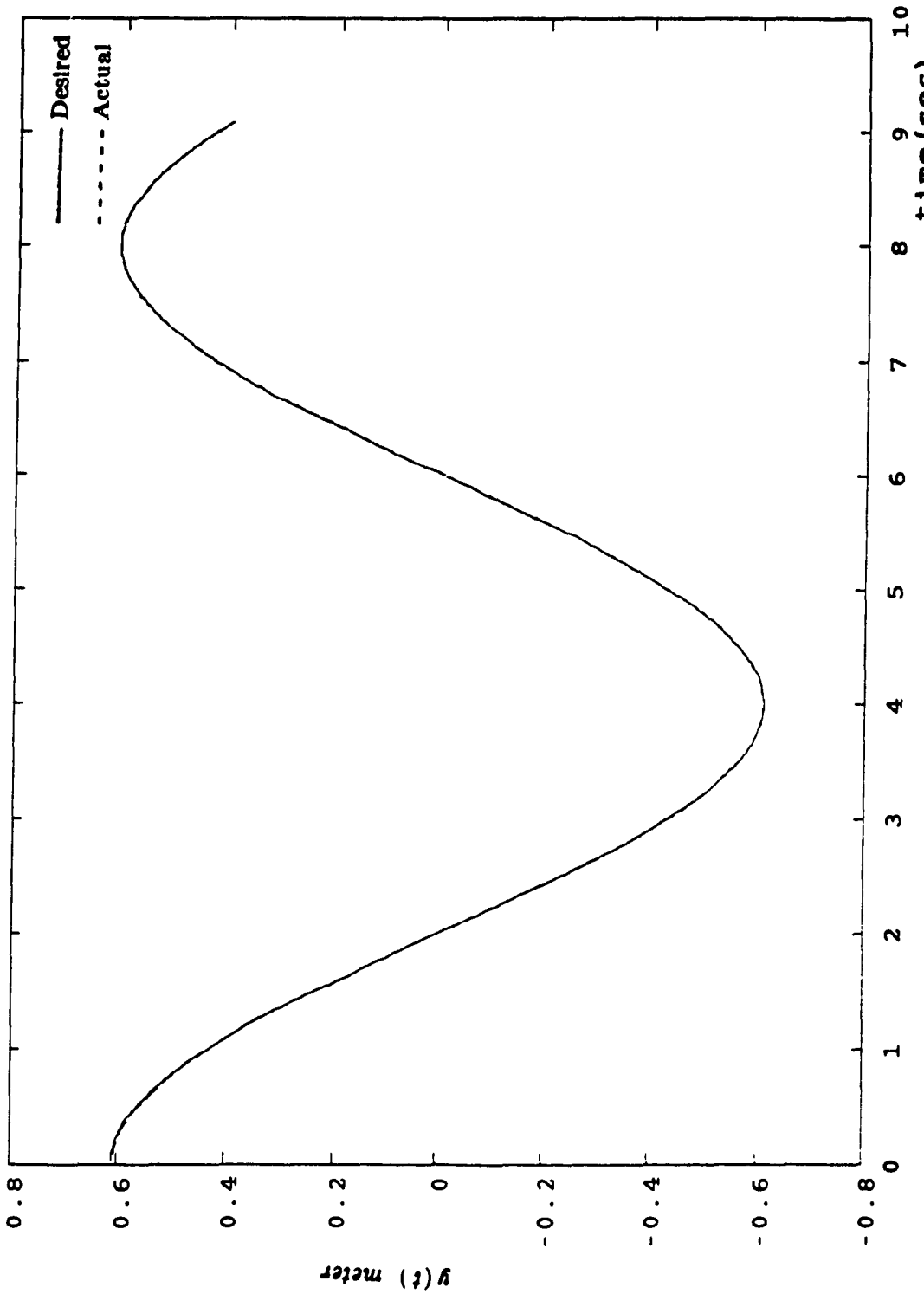

Fig.(6.13b) Response of the end-effector in y-direction

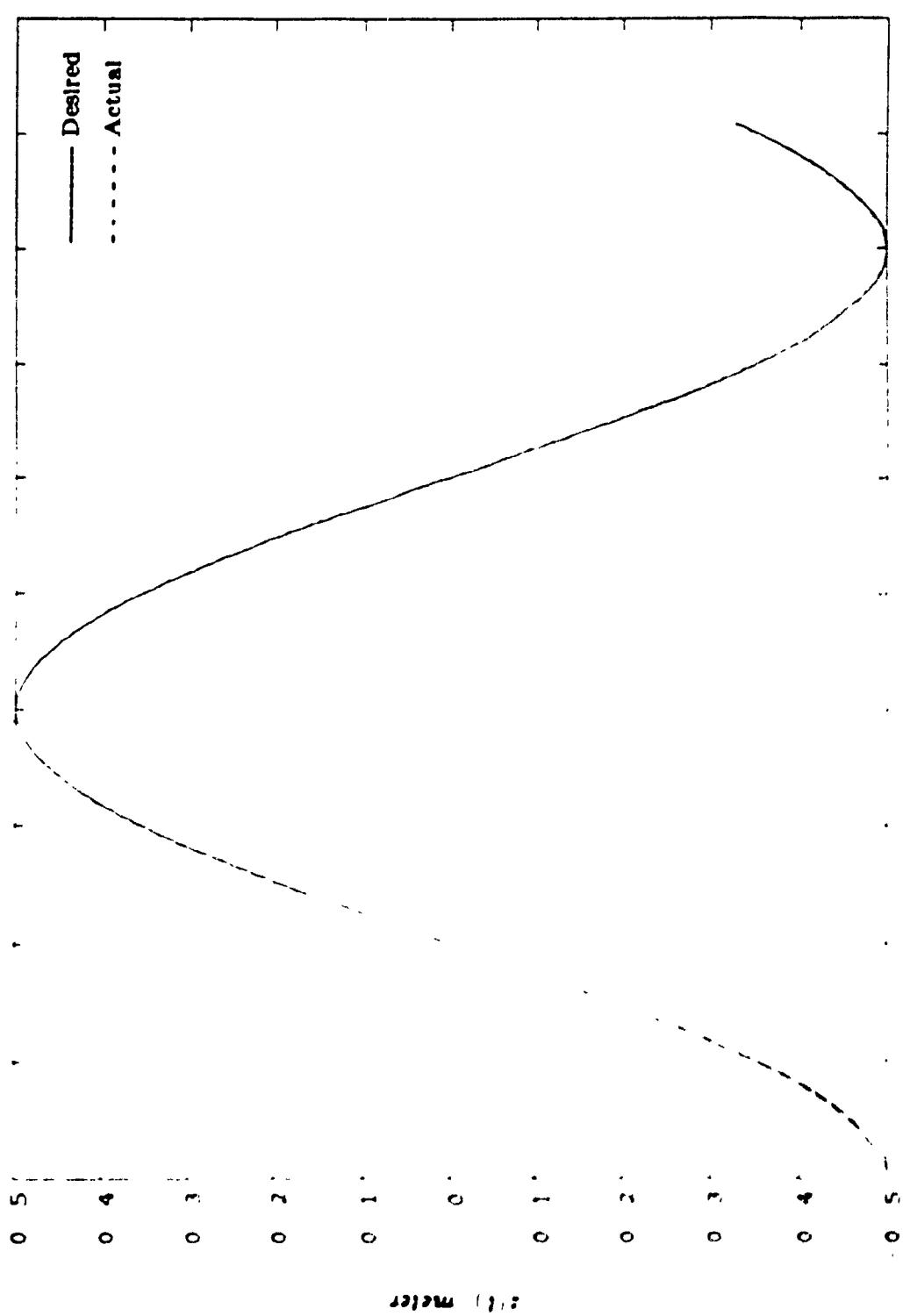

Fig (6.13c) Response of the end-effector in z-direction

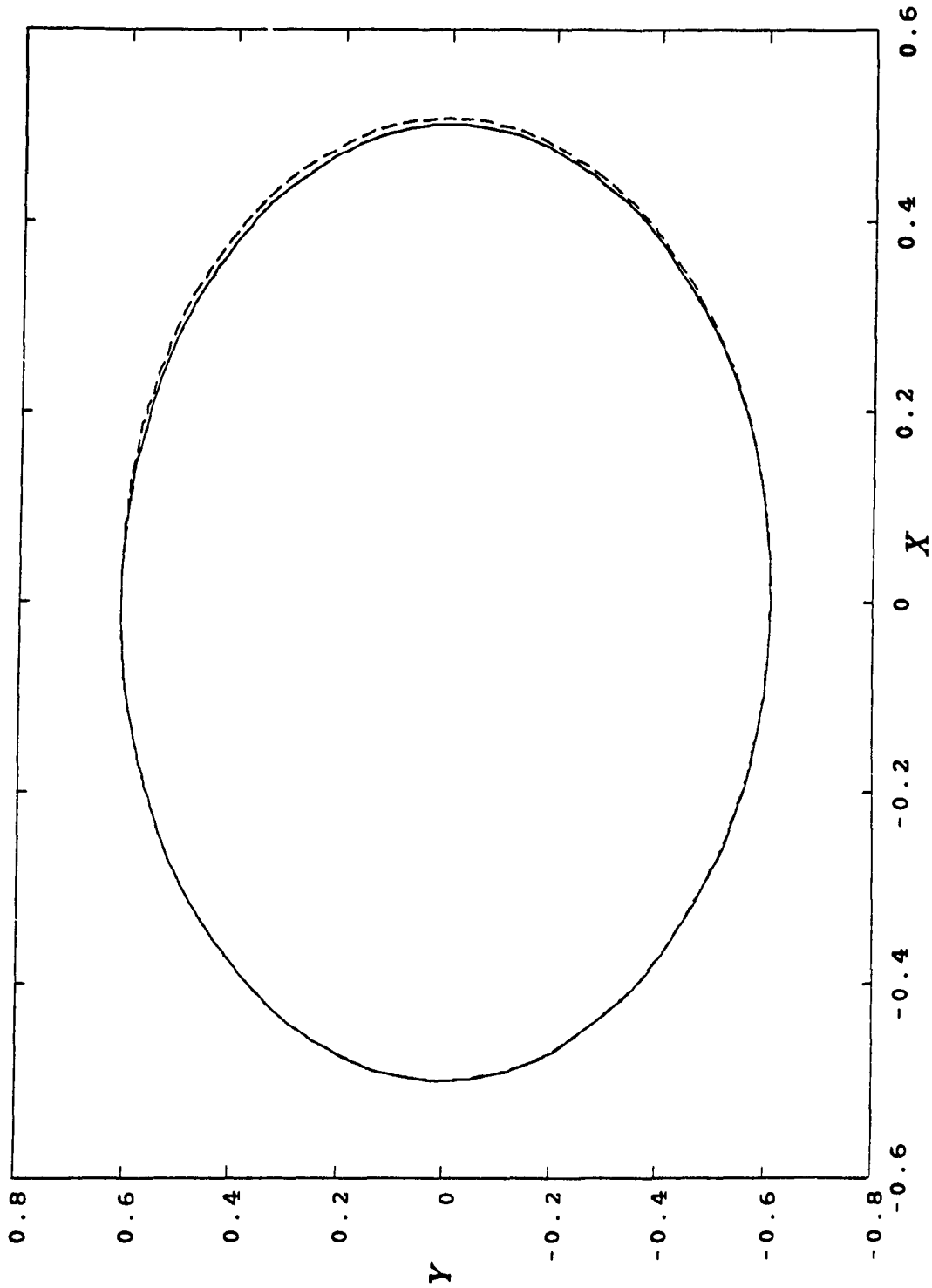

Fig.(6.14a) Projection of the trajectory on XY plane

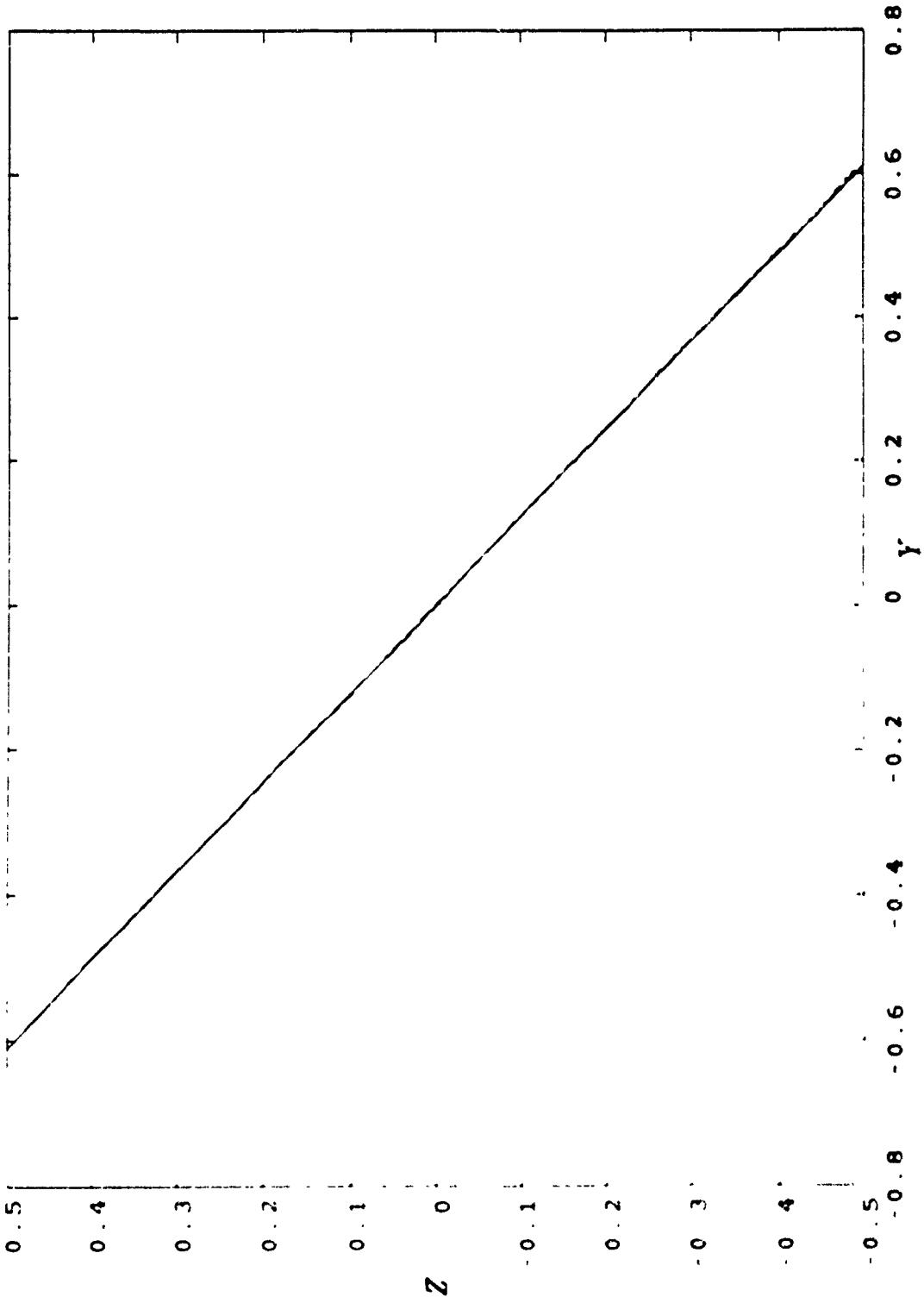

Fig.(6.14b) Projection of the trajectory on YZ plane

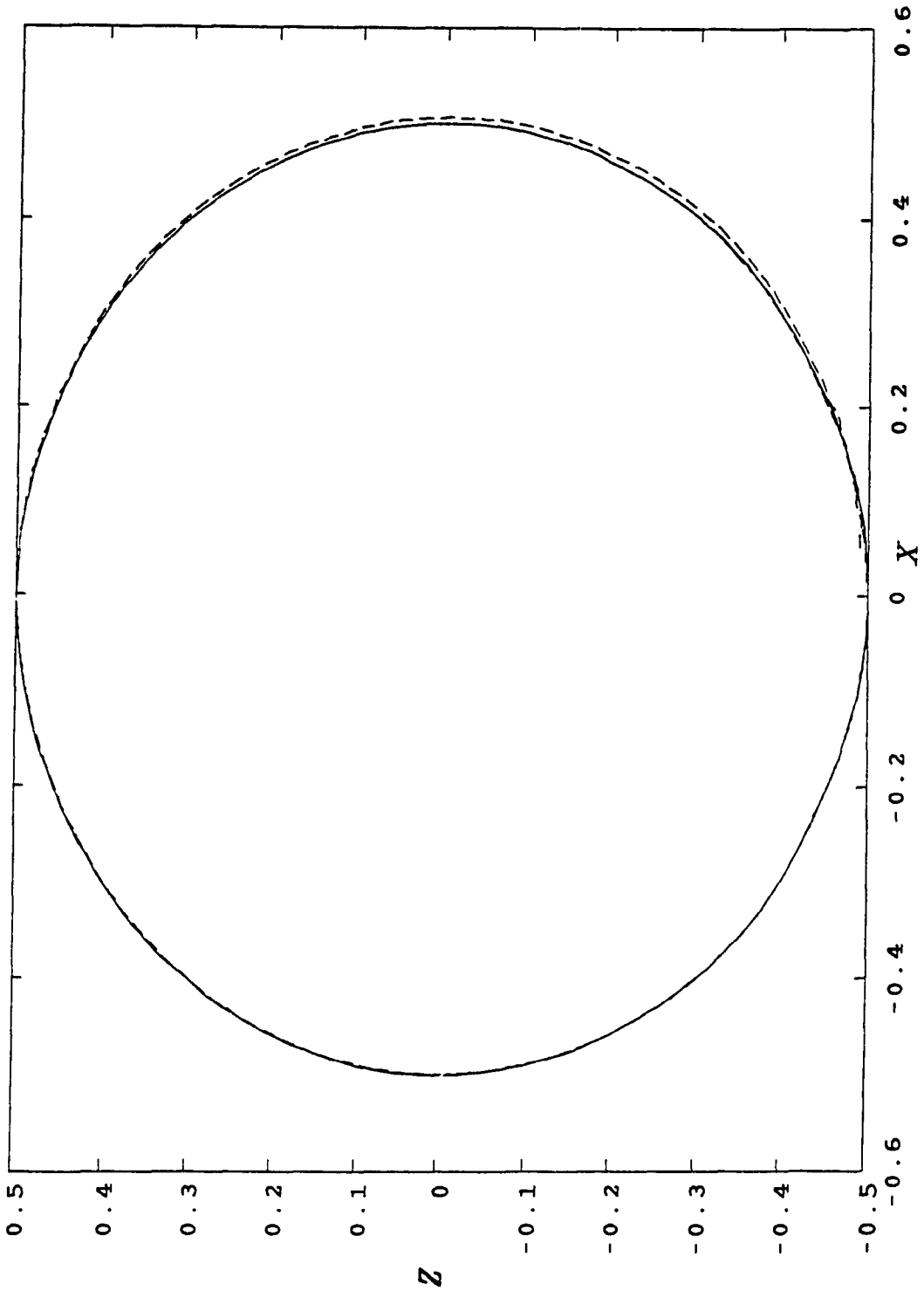

Fig. (6.14c) Projection of the trajectory on XZ plane

CHAPTER SEVEN

ADAPTIVE CONTROL OF FLEXIBLE JOINT MANIPULATORS

In this chapter the control problem for robot manipulators with flexible joints is considered. A reduced-order flexible joint model is constructed based on a singular perturbation formulation of the manipulator equations of motion. The concept of an integral manifold is utilized to construct the dynamics of a slow subsystem. A fast subsystem is also constructed to represent the fast dynamics of the elastic forces at the joints. A composite control scheme is developed based on on-line identification of the manipulator parameters, which takes into account unmodeled dynamics and parameter variations. An approach for stability analysis of the closed-loop full-order system is investigated.

This chapter is divided into the following sections: Section 1 gives a general introduction covering the problem definition for control of manipulators with joint elasticity and a brief review of the recent literature. In Section 2, a singular perturbation model of the manipulator is constructed. A control strategy and a robust controller for a reduced-order flexible subsystem (slow subsystem) are derived in Section 3. Section 4 gives the dynamics of the fast subsystem with a fast controller. Section 5 discusses the construction of the full-order controller with a composite control strategy and its stability analysis. The parameter estimation method and the control scheme implementation are discussed in Section 6. A numerical example of a single link flexible-joint manipulator is considered in Section 7 to illustrate the application of the proposed algorithm.

7.1 INTRODUCTION

In the process of modeling the dynamic behavior of robot manipulators, some physical phenomena like joint elasticity are usually neglected. Most methods assume that no compliance exists in the mechanical elements composing the manipulator arm, so that

the dynamics of the robot can be described by a rigid body model.

Elasticity implies that the position of an actuator (i.e. the angle of the motor shaft) is not equally related to the position of the driven link. From the modeling point of view, this internal deflection is taken into account by inserting a linear torsional spring at each joint. As a consequence, the rigid arm dynamic model has to be modified in order to describe completely the relation between applied torques and link motion [69]. Most industrial robots use DC or AC motors connected in series with harmonic drives (high-torque, high-ratio gear boxes) used mainly for speed reduction. In some applications, transmission belts, or long shafts in the drive system (usually in the joints) are also used. This results in lightly damped oscillatory modes in the open-loop response, and a typical resonant behavior [70]. To capture the above behavior the flexible manipulator is modeled by a chain of rigid sub-links interconnected by flexible joints [78].

It is shown in [71,72] that the control schemes which assume a rigid model for the manipulator are limited in their applicability to real robots where the assumption of perfect rigidity is never satisfied exactly. The resonant behavior in some range of frequencies imposes bandwidth limitations on any control algorithm that is designed assuming perfect rigidity and may cause stability problems for feedback control laws that neglect joint flexibility [75]. For quasi-static applications, simplified models which consider only the dynamics of the drive system, have been used by Kuntze et al. [76]. Spong [78] has investigated a simplified model which neglects the inertial coupling between the actuators and links. Models including full nonlinear dynamic interactions among joint elasticities and inertial properties of links and actuators have been introduced by Nicosia et al. [77]. It is shown by Cesareo and Marino [83], for a model of a three-link elastic joint robot, that the necessary and sufficient conditions for feedback linearizability are not satisfied.

Recently several advanced control algorithms for flexible-joint manipulators have been proposed. They use different approaches such as singular perturbation techniques

[79], sliding modes [80], pseudo-linearization [81] and model reference adaptive control [82]. All of the above methods are designed by assuming perfect knowledge of the parameters of the system, and the designed control system is unable to capture the results obtained in the nominal case for fully rigid manipulators. A method based on the concept of integral manifold was suggested by Khorasani et al. [73-74] and Spong et al. [75]. In this approach, they assume exact cancellation of the nonlinear terms of the manipulator in the case of rigid and reduced-order flexible models. Also the control scheme is designed for the slow modes only. In addition they give no stability proof for the composite system. De Luca [69] uses dynamic feedback for linearization, with no uncertainties. A theoretical study of robust control was performed by DeWit and Lys [70]. The approach uses a two-step estimation procedure for the unknown parameters of the manipulator. No stability analysis was given for the case of parameter variations. Since the estimation of the rigid body dynamics depends on the elasticity of the joints, a two-step estimation procedure for flexible-joint manipulators is not possible.

In this chapter, the concepts of linearization, singular perturbation, integral manifold, and composite control have been utilized for dynamic investigation of flexible-joint manipulators. A full-order dynamic model is constructed for the manipulator using the concepts of singular perturbation. A reduced-order model which has the same dimension as the rigid model is then obtained. This, in turn, yields an implicit expression for the computed torque that incorporates the effects of elastic distortion. An approximate reduced-order system known as the slow subsystem is constructed using a power series expansion about $\epsilon=0$ (a parameter indicating the degree of joint elasticity) A fast subsystem representing the deviation of the flexible-joint dynamics from the rigid dynamics is also derived We consider the link positions as slow variables and the elastic forces as fast variables As a consequence, we separate the full system into two subsystems, a slow subsystem and a fast subsystem

The adaptive control scheme developed in Chapter 4 is applied here to the rigid

body model (which can be constructed from the reduced-order model). A corrective control to compensate for the deviations of the dynamics of the flexible-joint system from those of the rigid system is developed using the reduced-order model. A fast control scheme is designed based on the fast subsystem model to ensure attractivity of the flexible joint dynamics to the integral manifold. To overcome the effects of uncertainties due to mismatch in the parameters and to compensate for the effects of parameter variations, a robust controller similar to that given in Chapter 4 is designed using Lyapunov theory. The overall composite control strategy consists of four parts: A feedforward computed torque with a PD feedback controller, a robust controller, a corrective controller, and a fast control scheme. The performance of these controllers depends on the accuracy of the measurement of the link motion (position, velocity, and acceleration), and hence, we assume that high precision sensors are instrumented directly on the link. The composite controller is based on the on-line identification of the unknown parameters of the manipulator. For the identification we use the reduced-order flexible-joint model, since it contains the rigid body parameters and the elastic parameter. The remaining system parameters which are used in the implementation of the control scheme can be constructed from these identified parameters as will be discuss later. Stability analysis of the closed-loop full-order system (with the composite control strategy) is investigated. The methods developed in Chapters 4 and 5 are not directly applicable because of the nature of the dynamic equations of the full-order system. A new technique which employs modified versions of these methods is used. Similarly, the Q matrix in the Lyapunov equation is selected on-line to ensure that the full-order closed-loop system is asymptotically stable.

7.2 A Singular Perturbation Model for Flexible Joint Manipulators in Joint Space

In formulating the dynamical model of a manipulator with flexible joints, we will consider the link positions (velocities, accelerations) as slow variables and the elastic

forces at the joints and their time derivatives as the fast variables. Let us consider a manipulator consisting of $n+1$ links interconnected by n flexible revolute joints and let θ_i, $i=1,\ldots,n$, denote the position of the i^{th} link, and ϕ_i, $i=1,\ldots,n$, denote the position of the i^{th} actuator. Let

$$q = \left[\theta^T \mid \phi^T\right]^T = \left[\theta_1 \ldots \theta_n \mid \phi_1 \ldots \phi_n\right]^T$$

Then the equations of motion of the manipulator can be written as [75]

$$M(q)\ddot{q} + B(q, \dot{q}) + \psi(q) = P^* u$$

In comparison with rigid body dynamics this model consists of $2n$ equations. The inertia matrix $M(q) \in \mathbb{R}^{2n \cdot 2n}$ is a positive-definite symmetric matrix with $q \in \mathbb{R}^{2n}$, $B(q, \dot{q})$ represents the Coriolis/centrifugal and gravity vector and $u=(u_1 \quad . \; u_n)^T$, where u_i is the torque/force delivered by the i^{th} actuator. The vector

$$\psi(q) = \left[\psi_1(q), \quad \ldots, \psi_{2n}(q)\right]^T$$

with

$$\psi_i(q) = \begin{cases} k_i(q_i \quad q_{n+i}) & 1 \leq i \leq n \\ k_i(q_{i-n} \quad q_i) & n+1 \leq i \leq 2n \end{cases}$$

represents the elastic force-torque vector at the joint, where k_i denotes the joint elasticity. The $2n \times n$ matrix P^* has the form

$$P^* = \begin{bmatrix} 0_{n \cdot n} \\ I_{n \cdot n} \end{bmatrix}$$

In the case of all rigid joints, $k_i \to \infty$ and the model will reduce to a rigid model of n equations (as (4.2.1)) in which $q_{n+i} = q_i$, $i=1,\ldots,n$, with ψ_i remaining finite (for more details see [75,79]).

Now consider a manipulator with flexible joints whose i^{th} link is shown in Fig (7.1) in which the flexible joint is modeled as a linear torsional spring of elasticity k_i. We will assume that all the spring constants k_i are of the same (large) order of

magnitude K, and we will take the gear ratio N = 1 for simplicity. The dynamic equations for this system are given [70] as:

$$J_l\ \ddot{\theta} + B_l\ \dot{\theta} + R_l(\theta,g) = K(\phi - \theta) \tag{7.2.1}$$

and

$$J_m\ \ddot{\phi} + B_m\ \dot{\phi} + R_m(\phi,g) = K(\theta - \phi) + \tau_m \tag{7.2.2}$$

where

J_m, J_l are the inertias of the motor and the link respectively,

B_m, B_l are the centrifugal and Coriolis terms of the motor and the link,

R_m, R_l are the gravitation and friction terms of the motor and the link,

K is the stiffness of the flexible joint,

τ_m is the total torque applied to the motor,

g is the gravity constant

Equations (7.2.1) and (7.2.2) can be written in state-space form as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ -J_l^{-1}K & -J_l^{-1}B_l & J_l^{-1}K & 0 \\ 0 & 0 & 0 & I \\ J_m^{-1}K & 0 & -J_m^{-1}K & -J_m^{-1}B_m \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -J_l^{-1}R_l \\ 0 \\ -J_m^{-1}R_m \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ J_m^{-1} \end{bmatrix} \tau_m$$

Let us define

$$\psi = K(\theta - \phi) \tag{7.2.3}$$

where, ψ is the elastic torque at the joints. By introducing the parameter $\epsilon = \frac{1}{K}$ (-in the case of rigid links, $\epsilon \to 0$ *i.e.* $K \to \infty$, at which $\theta = \phi$), we can express (7.2.1) and (7.2.2) as a singular perturbation model as follows:

From (7.2.3) we have $\phi = \theta - \epsilon\ \psi$. By substituting into (7.2.2), we get

$$J_m\ \ddot{\theta} - \epsilon\ J_m\ \ddot{\psi} + B_m\ \dot{\theta} - \epsilon\ B_m\ \dot{\psi} + R_m(\theta,\ g\ ,\ \psi) = \psi + \tau_m \tag{7.2.4}$$

Using (7.2.3), we can find $\ddot{\theta}$ from (7.2.1) as

$$\ddot{\theta} = -\ J_l^{-1}\ \psi - J_l^{-1}\ B_l\ \dot{\theta} - J_l^{-1}\ R_l \tag{7.2.5}$$

By substituting (7.2.5) into (7.2.4), we get

$$-\epsilon\,\ddot{\psi} = \left[J_l^{-1} + J_m^{-1}\right]\psi + J_m^{-1}\,B_m\,\epsilon\,\dot{\psi} + \left[J_l^{-1}\,B_l - J_m^{-1}\,B_m\right]\dot{\theta} \quad (7.2.6)$$
$$+ \left[J_l^{-1}\,R_l - J_m^{-1}\,R_m\right] + J_m^{-1}\,\tau_m$$

Equations (7.2.5) and (7.2.6) represent a singular perturbation system in terms of the slow variable θ and the fast variable ψ. The full-order model (7.2.5), (7.2.6) represents a complex highly nonlinear system which is difficult to analyse or use directly in control design.

It can be shown that the full-order model can be used to obtain the rigid model. This can be achieved by setting ϵ to zero in (7.2.6) (i.e. neglecting elasticity). This will reduce ψ to Υ_o, where Υ_o is the quasi-steady state of the fast variables (rigid manifold). From (7.2.6) we get,

$$-\left[J_l^{-1} + J_m^{-1}\right]\Upsilon_o = \left[J_l^{-1}\,B_l - J_m^{-1}\,B_m\right]\dot{\theta} + \left[J_l^{-1}\,R_l - J_m^{-1}\,R_m\right] + J_m^{-1}\,\tau_o \quad (7.2.7)$$

where τ_o is the control input for the rigid model. It has been shown in [23,31,33] that the term $J_l^{-1}\,R_l - J_m^{-1}\,R_m$ can be written as $\alpha_1(\theta, g)\,\theta$, where α_1 consists of all the terms containing the gravity constant g. Let us define

$$\begin{aligned} J_l^{-1} + J_m^{-1} &= \alpha_o \\ J_l^{-1}\,R_l - J_m^{-1}\,R_m &= \alpha_1\,\theta \\ J_l^{-1}\,B_l - J_m^{-1}\,B_m &= \alpha_2 \\ J_m^{-1} &= a_1 \\ J_m^{-1}\,B_m &= a_o \end{aligned} \quad (7.2.8)$$

Then (7.2.7) becomes

$$\alpha_o\,\Upsilon_o = a_1\,\tau_o + \alpha_1\,\theta + \alpha_2\,\theta$$

Thus Υ_o, the quasi-steady state of ψ, can be obtained as

$$\Upsilon_o = \alpha_o^{-1}\left[a_1\,\tau_o + \alpha_1\,\theta + \alpha_2\,\theta\right] \quad (7.2.9)$$

By substituting (7.2.9) into (7.2.5) and rearranging, we get

$$\ddot{\theta} = \left[J_l^{-1}\,\alpha_o^{-1}\,\alpha_2 - J_l^{-1}\,B_l\right]\dot{\theta} + \left[J_l^{-1}\,\alpha_o^{-1}\,\alpha_1 - J_l^{-1}\,R_l\right]\theta + J_l^{-1}\,\alpha_o^{-1}\,a_1\,\tau_o \tag{7.2.10}$$

Let us define

$$\begin{aligned} -J_l^{-1}\,\alpha_o^{-1}\,\alpha_2 + J_l^{-1}\,B_l &= \beta_o \\ -J_l^{-1}\,\alpha_o^{-1}\,\alpha_1 + J_l^{-1}\,R_l &= \beta_1 \\ J_l^{-1}\,\alpha_o^{-1}\,a_1 &= \beta_2 \end{aligned} \tag{7.2.11}$$

Note that β_2 is an inertia matrix which is positive-definite. Using (7.2.11), (7.2.10) becomes

$$\ddot{\theta} = -\beta_o\,\dot{\theta} - \beta_1\,\theta + \beta_2\,\tau_o \tag{7.2.12}$$

This represents the dynamics of the rigid model, which can be written in state-space form as

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & I\\ -\beta_1 & -\beta_o\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ \beta_2\end{bmatrix}\tau_o \tag{7.2.13}$$

Since β_2 has full rank, this system is controllable and thus can be stabilized by state feedback. The model (7.2.13) is of order $2n$ and is the same form as the rigid model obtained in Chapter 4. The full $(2n)th$ order system (7.2.5), (7.2.6) will be considered further in the following sections.

7.3 Reduced Order Flexible Model (slow subsystem)

As we have seen from the previous section, the full-order system (7.2.5), (7.2.6) is a complex, highly nonlinear system and is difficult to analyse. The rigid model (7.2.13) does not accurately model the system if joint flexibility is not negligible. Since the full-order system can be represented as a singularly perturbed system the theory of singular perturbations can be applied to derive the reduced-order flexible model (using the concept of an integral manifold [73-75,84,90]). The reduced-order flexible model, like the rigid model has dimension $2n$, but incorporates the effects of joint flexibility. Since the inertia matrix of the manipulator is invertible, the theory of singular perturbations [90]

allows us to obtain a reduced-order flexible model.

7.3.1 A Reduced Order Flexible Model

To derive the reduced-order flexible model, let us rewrite (7.2.6) using (7.2.8) as

$$-\epsilon\,\ddot{\psi} = \alpha_o\,\psi + a_o\,\epsilon\,\dot{\psi} + \alpha_2\,\dot{\theta} + \alpha_1\,\theta + a_1\,\tau_m \tag{7.3.1}$$

In general ψ can be written as

$$\psi = \Upsilon + \xi$$

where Υ (integral manifold) is the quasi-steady state value of ψ, i.e. it represents ψ in the integral manifold, and ξ represents the transient behavior of ψ, i.e. it is the fast part of ψ. Now let us consider the system in the integral manifold at which $\psi = \Upsilon(\theta, \dot{\theta}, \tau_m, \epsilon)$ and $\dot{\psi} = \dot{\Upsilon}(\theta, \dot{\theta}, \tau_m, \epsilon)$. By substituting for ψ in (7.3.1), we get

$$-\epsilon\,\ddot{\Upsilon} = \alpha_o\,\Upsilon + a_o\,\epsilon\,\dot{\Upsilon} + \alpha_2\,\dot{\theta} + \alpha_1\,\theta + a_1\,\tau_m \tag{7.3.2}$$

where, $\dot{\Upsilon}$ is the total differential and is calculated along the trajectories θ and $\dot{\theta}$. We can define $\dot{\Upsilon}$ as [75]

$$\dot{\Upsilon} = \frac{d}{dt}\Upsilon(\theta, \dot{\theta}, \tau_m, \epsilon) = \frac{\partial \Upsilon}{\partial \theta}\,\dot{\theta} + \frac{\partial \Upsilon}{\partial \dot{\theta}}\,\ddot{\theta} + \frac{\partial \Upsilon}{\partial \tau_m}\,\dot{\tau}_m \tag{7.3.3}$$

Noting that Υ is twice continuously differentiable, $\ddot{\Upsilon}$ can be found in the same way. Equation (7.3.2) represents a partial differential equation in the unknown Υ. Equation (7.2.5) (with $\psi = \Upsilon$) is used to substitute for $\ddot{\theta}$ so that only θ and $\dot{\theta}$ are present in (7.3.2) and no other higher terms are needed. Once $\dot{\Upsilon}$ and $\ddot{\Upsilon}$ have been found, we substitute them into (7.2.4) to get the reduced-order flexible model

$$\ddot{\theta} = \epsilon\,\ddot{\Upsilon} - a_o\,\dot{\theta} + \epsilon\,a_o\,\dot{\Upsilon} - a_1\,R_m + a_1\,\Upsilon + a_1\,\tau_m \tag{7.3.4}$$

This model has the same dimension as the rigid model and incorporates the effects of the flexibility restricted to the integral manifold Υ.

7.3.2 An Approximation for the Reduced Order Flexible Model

The reduced-order flexible model (7.3.4) is feedback linearizable [73], but the computation required for solving the partial differential equations (7.3.2) are quite difficult. An approximate reduced-order flexible model can be found by using a power series expansion about ϵ. This yields a practical computational technique for controlling the flexible model.

The control input τ_m in (7.3.4) is a composite control and is expressed as $\tau_m = \tau_s + \tau_f$. In the reduced-order flexible model only τ_s is present since on the manifold $\tau_f\ (\xi = 0) = 0$. So let us denote

$$\Upsilon = \Upsilon_0 + \epsilon\, \Upsilon_1 \tag{7.3.5a}$$

$$\tau_s = \tau_0 + \epsilon\, \tau_1 \tag{7.3.5b}$$

where τ_0 is the control input to the rigid model , and τ_1 is the corrective torque vector for compensating for the effects of Υ_1. The vector Υ_0 represents a zero-th order approximation of Υ, and Υ_1 represents the first order correction to Υ_0. Equation (7.3.5) represents an approximation up to the first power of ϵ. Note that an approximation to any power is possible. By substituting (7.3.5) into (7.3.1), we get

$$\begin{aligned} -\epsilon\, \ddot{\Upsilon}_0 - \epsilon^2\, \ddot{\Upsilon}_1 = a_0\ \epsilon\, \dot{\Upsilon}_0 + a_0\ \epsilon^2\, \dot{\Upsilon}_1 + \alpha_0\, \Upsilon_0 + \epsilon\ \alpha_0\, \Upsilon_1 + a_1\, \tau_0 \\ + \epsilon\ a_1\, \tau_1 + \alpha_2\, \dot{\theta} + \alpha_1\, \theta \end{aligned} \tag{7.3.6}$$

By equating the terms of like powers of ϵ on both sides of (7.3.6), up to the first power of ϵ, and neglecting all terms of higher power of ϵ , we get

$$-\ddot{\Upsilon}_0 = a_0\ \dot{\Upsilon}_0 + \alpha_0\ \Upsilon_1 + a_1\, \tau_1 \tag{7.3.7}$$

and

$$-\alpha_0\ \Upsilon_0 = a_1\, \tau_0 + \alpha_2\, \dot{\theta} + \alpha_1\, \theta \quad . \tag{7.3.8}$$

From (7.3.7) we obtain

$$\Upsilon_1 = -\alpha_0^{-1} \left[\ddot{\Upsilon}_0 + a_0\ \dot{\Upsilon}_0 + a_1\, \tau_1 \right] \tag{7.3 9}$$

and (7.3.8) yields

$$\Upsilon_o = -\alpha_o^{-1}\left[a_1 \tau_o + \alpha_2 \dot{\theta} + \alpha_1 \theta\right] \tag{7.3.10}$$

It can be seen from these equations that (7.3.10) relates the rigid control τ_o to Υ_o, and (7.3.9) relates the correcting control τ_1 to Υ_1. Using (7.3.9) and (7.3.10), we can get $\Upsilon = \Upsilon_o + \epsilon \Upsilon_1$ which after substitution ($\psi = \Upsilon$) into (7.2.5) gives

$$\ddot{\theta} = \left[J_l^{-1} \alpha_o^{-1} \alpha_2 - J_l^{-1} B_l\right]\dot{\theta} + \left[J_l^{-1} \alpha_o^{-1} \alpha_1 - J_l^{-1} R_l\right]\theta \tag{7.3.11}$$
$$+ J_l^{-1} \alpha_o^{-1} a_1 \tau_o + \epsilon\left[J_l^{-1} \alpha_o^{-1} a_1 \tau_1 + J_l^{-1} \alpha_o^{-1} \ddot{\Upsilon}_o + J_l^{-1} \alpha_o^{-1} a_o \dot{\Upsilon}_o\right] + O(\epsilon^2)$$

Using the notation in (7.2.11), (7.3.11) becomes

$$\ddot{\theta} = -\beta_o \dot{\theta} - \beta_1 \theta + \beta_2 \tau_o + \epsilon\left[\beta_2 \tau_1 + \beta_2 a_1^{-1} \ddot{\Upsilon}_o + \beta_2 a_1^{-1} a_o \dot{\Upsilon}_o\right] + O(\epsilon^2)$$

which can be written in state-space form as

$$\begin{bmatrix}\dot{\theta} \\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}0 & I \\ -\beta_1 & -\beta_o\end{bmatrix}\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0 \\ \beta_2\end{bmatrix}\tau_o + \begin{bmatrix}0 \\ \epsilon\beta_2 a_1^{-1}\end{bmatrix}\ddot{\Upsilon}_o + \begin{bmatrix}0 \\ \epsilon\beta_2 a_1^{-1} a_o\end{bmatrix}\dot{\Upsilon}_o \tag{7.3.12}$$
$$+ \begin{bmatrix}0 \\ \epsilon\beta_2\end{bmatrix}\tau_1 + O(\epsilon^2)$$

If the $O(\epsilon^2)$ terms are neglected, equation (7.3.12) represents an approximate reduced-order flexible model (slow subsystem), and it is clear that as $\epsilon \rightarrow 0$, (7.3.12) reduces to the rigid model (7.2.13). Note that (7.3.12) has dimension $2n$. It will be used for designing the corrective control τ_1 and the rigid control τ_o.

7.3.3 Control Algorithm for the Slow Subsystem

The corrective control τ_1 can be designed from the reduced-order flexible model by setting the ϵ terms in (7.3.12) equal to zero, i.e.

$$\beta_2 \tau_1 + \beta_2 a_1^{-1} \ddot{\Upsilon}_o + \beta_2 a_1^{-1} a_o \dot{\Upsilon}_o = 0$$

from which the corrective control is obtained as

$$\tau_1 = -a_1^{-1}\left[\ddot{\Upsilon}_o + a_o \dot{\Upsilon}_o\right] \tag{7.3.13}$$

To implement (7.3.13), we need to compute $\dot{\Upsilon}_o$ and $\ddot{\Upsilon}_o$. These can obtained by

differentiating (7.3.10) and using $\ddot{\theta}$ from the rigid model (7.2.12). By applying τ_1 to the slow subsystem (7.3.12), we obtain the rigid model

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\beta_1 & -\beta_0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \beta_2 \end{bmatrix} \tau_0 + O(\epsilon^2) \tag{7.3.14}$$

In order to design the rigid control τ_0, we first assume that the manipulator parameters are slowly time-varying during the interval Δt_j defined by $t_j < t < t_{j+1}$, i.e. a_0, a_1, α_0, α_1, α_2, β_0, β_1, β_2 are assumed to be constant during the interval Δt_j. So we can approximate the reduced-order flexible model (7.3.12) by a linear piecewise time-invariant model during the interval Δt_j. This model is identified on-line. Accordingly, we can choose the rigid control input τ_0 as

$$\tau_0 = \hat{\beta}_2^{-1} \left[\ddot{\theta}_d + K_v \dot{e} + K_p e + \hat{\beta}_0 \dot{\theta}_d + \hat{\beta}_1 \theta_d \right] \tag{7.3.15}$$

where, θ_d, $\dot{\theta}_d$, $\ddot{\theta}_d \in \mathbb{R}^n$ are the desired position, velocity and acceleration trajectories respectively; K_p, $K_v \in \mathbb{R}^{n \times n}$ are proportional-derivative gain matrices, and

$$e = \theta_d - \theta \quad , \quad \dot{e} = \dot{\theta}_d - \dot{\theta} \text{ and } \ddot{e} = \ddot{\theta}_d - \ddot{\theta} \tag{7.3.16}$$

In (7.3.15), $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2 \in \mathbb{R}^{n \times n}$ are the estimated values of β_0, β_1, β_2, respectively. Applying the control (7.3.15) to the rigid manipulator (7.3.14) (neglecting $O(\epsilon^2)$ terms) yields

$$\ddot{\theta} = -\beta_1 \theta - \beta_0 \dot{\theta} + \beta_2 \hat{\beta}_2^{-1} \left[\ddot{\theta}_d + K_v \dot{e} + K_p e + \hat{\beta}_0 \dot{\theta}_d + \hat{\beta}_1 \theta_d \right]$$

Note that a_0 and a_1 in (7.3.13), are replaced by their estimates $\hat{a}_0$ and $\hat{a}_1$, respectively. We shall discuss in Section 7.6 how these estimated parameters can be obtained. During the interval Δt_j, assuming that the estimates of the parameters converge to their true values, the error equation is

$$\ddot{e} + \left(K_v + \hat{\beta}_0 \right) \dot{e} + \left(K_p + \hat{\beta}_1 \right) e = 0 \tag{7.3.17}$$

It is required that the error satisfies a given desired second order differential equation of the form

$$\ddot{e} + \Lambda_v \dot{e} + \Lambda_p e = 0$$

from which the controller gains can be obtained as

$$K_p = \Lambda_p - \hat{\beta}_1 \qquad K_v = \Lambda_v - \hat{\beta}_o \tag{7.3.18}$$

where $\Lambda_v = diag\,(2\varsigma\omega)$ and $\Lambda_p = diag\,(\omega^2) \in \mathbb{R}^{n \times n}$ are given constant matrices, and ς and ω are the desired damping factor and undamped natural frequency respectively. Thus the total control input to the reduced-order flexible model becomes

$$\begin{aligned} \tau_s &= \tau_o + \epsilon\, \tau_1 \\ &= \hat{\beta}_2^{-1} \left[\ddot{\theta}_d + K_v \dot{e} + K_p e + \hat{\beta}_o \dot{\theta}_d + \hat{\beta}_1 \theta_d \right] - \epsilon\, \hat{a}_1^{-1} \left[\ddot{\Upsilon}_o + \hat{a}_o \dot{\Upsilon}_o \right] \end{aligned} \tag{7.3.19}$$

It is seen that the control τ_s can be implemented easily in terms of the desired trajectories, the measured slow variables θ and $\dot{\theta}$ of the link and the identified parameters of the reduced-order flexible model. Note also that $\dot{\Upsilon}_o$ and $\ddot{\Upsilon}_o$ can be computed in terms of the slow variables θ and $\dot{\theta}$. Now by applying the control τ_s to the reduced-order flexible model (7.3.12), we get the closed-loop error equation

$$\dot{z} = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} z \triangleq \Pi\, z \tag{7.3.20}$$

where $z = \begin{pmatrix} e^T & \dot{e}^T \end{pmatrix}^T \in \mathbb{R}^{2n \times 1}$. By appropriate choice of Λ_p and Λ_v we can ensure that the closed-loop poles lie in the left-half of the complex plane. This will ensure that $e(t) \rightarrow 0$ *as* $t \rightarrow \infty$.

7.3.4 Robust Control and Stability Analysis for the Slow Subsystem

In this section, we will assume that the system (7.3.12) is not completely linearizable, due to the existence of a mismatch between the actual manipulator dynamics and the estimated ones. Therefore we write the manipulator parameters as

$$\beta_o = \hat{\beta}_o + \Delta\beta_o \;,\; \beta_1 = \hat{\beta}_1 + \Delta\beta_1 \;, \text{ and } \beta_2 = \hat{\beta}_2 + \Delta\beta_2 \;, \ldots\ldots, \textit{etc.}$$

Then, the reduced-order flexible model (7.3.12) can be represented in terms of the

parameters $\hat{\beta}_o$, $\hat{\beta}_1$, $\hat{\beta}_2$, $\hat{a}_o$, $\hat{a}_1$ and a nonlinear time-varying term $\eta(t)$ which contains all the Δ terms, and order ϵ and higher order terms. Neglecting $O(\epsilon^2)$ terms, (7.3.12) can be rewritten as

$$\ddot{\theta} = -\hat{\beta}_1 \theta - \hat{\beta}_o \dot{\theta} + \hat{\beta}_2 \tau_o + \epsilon \hat{\beta}_2 \left[\hat{a}_1^{-1} \ddot{\Upsilon}_o + \hat{a}_1^{-1} \hat{a}_o \dot{\Upsilon}_o + \tau_1\right] - \eta(t) \quad (7.3.21)$$

Now, applying the control input (7.3.19) to (7.3.21), using the error equations (7.3.16) and rearranging, we get the closed-loop error equation

$$\ddot{e} + \left[\hat{\beta}_o + K_v\right] \dot{e} + \left[\hat{\beta}_1 + K_p\right] e = \eta(t) \quad (7.3.22)$$

which can be rewritten in state-space form as

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \eta(t)$$

i.e.

$$\dot{z} = \Pi z + B^* \eta(t) \quad (7.3.23)$$

where,

$$\Pi = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} \quad \text{and} \quad B^* = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Equation (7.3.23) represents the closed-loop slow subsystem without compensation for the unmodeled dynamics.

In order to compensate for the effects of the unmodeled dynamics, another control term $S(t)$ is added to the rigid controller τ_o :

$$\tau_o = \hat{\beta}_2^{-1} \left[\ddot{\theta}_d + K_v \dot{e} + K_p e + \hat{\beta}_o \dot{\theta}_d + \hat{\beta}_1 \theta_d + S(t)\right] \quad (7.3.24)$$

Applying the control τ_s, with τ_o from (7.3.24), to the slow subsystem (7.3.21) yields the error equation

$$\ddot{e} + \Lambda_v \dot{e} + \Lambda_p e = \eta(t) - S(t)$$

i.e.

$$\dot{z} = \Pi z + B^* \sigma(t) \quad (7.3.25)$$

where $\sigma(t) = \eta(t) - S(t)$. In order to compute $S(t)$ and study the stability of the system (7.3.25), we will use Lyapunov stability theory.

Let us choose

$$V(t) = z^T\ P\ z + \sigma^T\ Q_0\ \sigma \tag{7.3.26}$$

where $Q_0 \in \mathbb{R}^{2n \times 2n}$ and $P \in \mathbb{R}^{2n \times 2n}$ are positive-definite matrices and P satisfies the Lyapunov equation

$$\Pi^T\ P + P\ \Pi = -2Q \tag{7.3.27}$$

with Q a positive-definite matrix. Taking the derivatives (with respect to t) of both sides of (7.3.26) and using (7.3.25), we get

$$\dot{V} = -2z^T\ Q\ z + 2\sigma^T \left[Q_0\ \dot{\sigma} + B^{*T}\ P\ z\right]$$

We see that $\dot{V}(t) < 0$ along the trajectories of (7.3.25) and consequently the system (7.3.25) is asymptotically stable if

$$\sigma^T \left[Q_0\ \dot{\sigma} + B^{*T}\ P\ z\right] = 0$$

which is satisfied if

$$\dot{\sigma} = -\ Q_0^{-1}\ B^{*T}\ P\ z \tag{7.3.28}$$

Since we have assumed that the system parameters are slowly time-varying , and since $\eta(t)$ is a function of the system parameters, therefore $\eta(t)$ is slowly time-varying. In this case, $\dot{\eta} \approx 0$ and $\dot{\sigma} \approx -\dot{S}(t)$. Then from (7.3.28), we obtain

$$\dot{S}(t) = Q_0^{-1}\ B^{*T}\ P\ z \tag{7.3.29}$$

Writing P as

$$P = \begin{bmatrix} P_1 & P_2^T \\ P_2 & P_3 \end{bmatrix}$$

equation (7.3.29) gives

$$\dot{S}(t) = Q_0^{-1} \left[P_2\ e\ + P_3\ \dot{e}\right] \tag{7.3.30}$$

This can be solved for $S(t)$ as

$$S(t) = S(t_o) + Q_o^{-1} \left[\int_{t_o}^{t} P_2\, e \; dt + P_3\, e \right] \tag{7.3.31}$$

where $S(t_o)$ denotes the initial value of $S(t)$.

7.4 THE FAST SUBSYSTEM

In this section, we derive a control law for the fast subsystem resulting from consideration of joint flexibility. To derive the control law we first consider the dynamics of the fast subsystem.

7.4.1 Dynamics of the Fast Subsystem

The dynamics of the variable ψ from its quasi-steady state Υ are governed by the fast subsystem. Let ξ denote the fast dynamics of ψ, where $\psi = \Upsilon + \xi$. Then from system (7.3.1) we get

$$-\epsilon \left[\ddot{\xi} + \ddot{\Upsilon}_o \right] = \alpha_o \left[\xi + \Upsilon_o + \epsilon\, \Upsilon_1 \right] + \epsilon\, a_o \left[\dot{\xi} + \dot{\Upsilon}_o \right] + \alpha_2 \dot{\theta} + \alpha_1 \theta + a_1 \tau_m + O(\epsilon^2) \tag{7.4.1}$$

where τ_m is the total control input delivered to the actuator, which is

$$\tau_m = \tau_s + \tau_f \tag{7.4.2}$$

with $\tau_s = \tau_o + \epsilon\, \tau_1$, the control input to the slow subsystem, and τ_f is the control input to the fast subsystem. By applying τ_m to (7.4.1) and using Υ_o from (7.3.10) and τ_1 from (7.3.13), we obtain

$$-\epsilon\, \ddot{\xi} = \alpha_o\, \xi + \epsilon\, \alpha_o\, \Upsilon_1 + \epsilon\, a_o\, \dot{\xi} + a_1\, \tau_f \tag{7.4.3}$$

Now by defining a new fast time-scale $T = \dfrac{t}{\sqrt{\epsilon}}$, so that

$$\frac{d\xi}{dt} = \frac{1}{\sqrt{\epsilon}} \frac{d\xi}{dT} \quad \text{and} \quad \frac{d^2\xi}{dt^2} = \frac{1}{\epsilon} \frac{d^2\xi}{dT^2}$$

and denoting $\xi'' = \dfrac{d^2\xi}{dT^2}$, $\xi' = \dfrac{d\xi}{dT}$ and so on for the other variables, we can write (7.4.3) as

$$-\xi'' = \alpha_o\ \xi + \epsilon\ \alpha_o\ \Upsilon_1 + \sqrt{\epsilon}\ a_o\ \xi' + a_1\ \tau_f$$

Now by formally letting $\epsilon \rightarrow 0$, we get

$$-\xi'' = \alpha_o\ \xi + a_1\ \tau_f \tag{7.4.4}$$

Equation (7.4.4) represents the fast subsystem in the fast time-scale. We can rewrite (7.4.4) in terms of the slow time-scale as

$$-\epsilon\ \ddot{\xi} = \alpha_o\ \xi + a_1\ \tau_f \tag{7.4.5}$$

the state-space form, becomes

$$\epsilon \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & \epsilon I \\ -\alpha_o & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + \begin{bmatrix} 0 \\ -a_1 \end{bmatrix} \tau_f \tag{7.4.6}$$

Equation (7.4.6) represents the dynamics of the fast subsystem in the slow time-scale, approximated up to $O(\epsilon)$.

7.4.2 Control Law for the Fast Subsystem

As can be seen from (7.4.6), the flexible modes are undamped [75], since the fast model contains poles on the imaginary axis of the complex plane. This requires that a fast controller τ_f be designed to guarantee asymptotic stability of the flexible modes. Let

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix}$$

Then (7.4.6) can be written as

$$\epsilon\ \dot{x} = A\ x + C\ \tau_f \tag{7.4.7}$$

where

$$A = \begin{bmatrix} 0 & \epsilon I \\ -\alpha_o & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} 0 \\ -a_1 \end{bmatrix}$$

Since the pair (A, C) is controllable, then (7.4.7) can be stabilized by state feedback. Therefore, we can choose τ_f as

$$\tau_f = -k_1\ \xi - k_2\ \dot{\xi} = -K^o\ x$$

where $K^o = \begin{pmatrix} k_1 & k_2 \end{pmatrix}$. Applying τ_f to the fast subsystem (7.4.7), yields

$$\epsilon \dot{x} = \left[A - C K^o \right] x \tag{7.4.8}$$

Equation (7.4.8) represents the closed-loop fast subsystem. The terms k_1 and k_2 can be chosen such that all the closed-loop poles of (7.4.8) lie in the left-half of the complex plane.

7.5 Full Order System

7.5.1 Singularly Perturbed Model of the Full Order System

The dynamic equations of the full-order system in terms of the slow variable θ and the fast variable ξ can be obtained by considering the full-order singularly perturbed system given by (7.2.5) and (7.3.1). By replacing ψ by $\Upsilon + \xi$, and using (7.2.5), we get

$$\ddot{\theta} = - J_l^{-1} \left[\Upsilon + \xi \right] - J_l^{-1} B_l \dot{\theta} - J_l^{-1} R_l \tag{7.5.1}$$

Since $\Upsilon = \Upsilon_o + \epsilon \Upsilon_1$ and Υ_1 and Υ_o are given by (7.3.9) and (7.3.10) respectively, equation (7.5.1) becomes

$$\ddot{\theta} = J_l^{-1} \alpha_o^{-1} \left[a_1 \tau_o + \alpha_2 \dot{\theta} + \alpha_1 \theta \right] + \epsilon J_l^{-1} \alpha_o^{-1} \left[\ddot{\Upsilon}_o + a_o \dot{\Upsilon}_o + a_1 \tau_1 \right] - J_l^{-1} \xi - J_l^{-1} B_l \dot{\theta} - J_l^{-1} R_l + O(\epsilon^2)$$

By rearranging terms and using the notation given in (7.2.11), we get

$$\ddot{\theta} = -\beta_o \dot{\theta} - \beta_1 \theta - \beta_2 a_1^{-1} \alpha_o \xi + \beta_2 \tau_o + \epsilon \beta_2 a_1^{-1} \left[\ddot{\Upsilon}_o + a_o \dot{\Upsilon}_o + a_1 \tau_1 \right] + O(\epsilon^2)$$

which can be approximated up to order ϵ and put in the state-space form

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\beta_1 & -\beta_o \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} - \begin{bmatrix} 0 \\ \beta_2 a_1^{-1} \alpha_o \end{bmatrix} \xi + \begin{bmatrix} 0 \\ \epsilon \beta_2 a_1^{-1} \left(\ddot{\Upsilon}_o + a_o \dot{\Upsilon}_o + a_1 \tau_1 \right) \end{bmatrix} + \begin{bmatrix} 0 \\ \beta_2 \end{bmatrix} \tau_o \tag{7.5.2}$$

Similarly, from (7.3.1) we get

$$-\epsilon \left[\dot{\xi} + \ddot{\Upsilon}_o \right] = \alpha_o \left[\xi + \Upsilon_o + \epsilon \Upsilon_1 \right] + \epsilon a_o \left[\dot{\xi} + \dot{\Upsilon}_o \right] + \alpha_2 \dot{\theta} + \alpha_1 \theta + a_1 \tau_m + O(\epsilon^2)$$

By using (7.3.9) and rearranging terms, we obtain

$$-\epsilon\,\ddot{\xi} = \alpha_o\;\xi + \epsilon\, a_o\,\dot{\xi} + \alpha_o\,\Upsilon_o - \epsilon\, a_1\,\tau_1 + \alpha_2\,\dot{\theta} + \alpha_1\,\theta + a_1\,\tau_m + O(\epsilon^2)$$

which can be approximated up to order ϵ and put in the state-space form

$$\epsilon\begin{bmatrix}\dot{\xi}\\ \ddot{\xi}\end{bmatrix} = \begin{bmatrix}0 & \epsilon I\\ -\alpha_o & -\epsilon a_o\end{bmatrix}\begin{bmatrix}\xi\\ \dot{\xi}\end{bmatrix} + \begin{bmatrix}0 & 0\\ -\alpha_1 & -\alpha_2\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix}0\\ -\alpha_o\end{bmatrix}\Upsilon_o + \begin{bmatrix}0\\ \epsilon a_1\end{bmatrix}\tau_1 + \begin{bmatrix}0\\ -a_1\end{bmatrix}\tau_m \tag{7.5.3}$$

Equations (7.5.2) and (7.5.3) represent the dynamics of the full-order system in singularly perturbed form and in terms of θ and ξ approximated up to order ϵ. Note that Υ_o is a function of θ.

7.5.2 Control Scheme and Stability Analysis for the Full Order System

By combining the slow controller and the fast controller, we get a composite controller for the full-order system which is given by

$$\tau_m = \tau_s + \tau_f = \tau_o + \epsilon\,\tau_1 + \tau_f \tag{7.5.4a}$$

where

$$\tau_o = \hat{\beta}_2^{-1}\left[\ddot{\theta}_d + K_v\,\dot{e} + K_p\,e + \hat{\beta}_o\,\dot{\theta}_d + \hat{\beta}_1\,\theta_d + S(t)\right] \tag{7.5.4b}$$

$$\tau_1 = -\hat{a}_1^{-1}\left[\ddot{\Upsilon}_o + \hat{a}_o\,\dot{\Upsilon}_o\right] \tag{7.5.4c}$$

and

$$\tau_f = -k_1\,\xi - k_2\,\dot{\xi} \tag{7.5.4d}$$

and $S(t)$ satisfies

$$\dot{S}(t) = Q_o^{-1}\left[P_2\,e + P_3\,\dot{e}\right] \tag{7.5.4e}$$

Note that τ_m represents the total control to be applied to the manipulator. Now we will study the use of this control scheme, by applying the composite control (7.5.4) to the full-order system (7.5.2) and (7.5.3). By proceeding in the same way as in section

(7.3.4), we get the following closed-loop error equations approximated up to order ϵ^2:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_p & -\Lambda_v \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ \beta_2 a_1^{-1} \alpha_o \end{bmatrix} \xi + \begin{bmatrix} 0 \\ I \end{bmatrix} \left[\eta(t) - S(t) \right] + O(\epsilon^2) \qquad (7.5.5a)$$

$$\epsilon \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & \epsilon I \\ -\alpha_o & -\epsilon a_o \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -\hat{a}_1 k_1 & -\hat{a}_1 k_2 \end{bmatrix} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + O(\epsilon^2) \qquad (7.5.5b)$$

Neglecting the $O(\epsilon^2)$ terms, equations (7.5.5) can be written as

$$\dot{z} = \Pi z + \Gamma^* x + B^* \sigma \qquad (7.5.6a)$$

$$\epsilon \dot{x} = \left[\tilde{A} - C K^o \right] x \qquad (7.5.6b)$$

where $\tilde{A}$ is

$$\tilde{A} = \begin{bmatrix} 0 & \epsilon I \\ -\alpha_o & -\epsilon a_o \end{bmatrix}$$

and Γ^* is

$$\Gamma^* = \begin{bmatrix} 0 & 0 \\ \alpha_o - a_1 & 0 \end{bmatrix}$$

Stability Analysis

Since the control input to the full-system is the composite control which consists of the slow control and the fast control, it is necessary to ensure the asymptotic stability of the full-order closed-loop system (7.5.6). To show how this can be achieved, let us choose a composite Lyapunov function

$$U(t) = V_s + V_f \qquad (7.5.7)$$

where V_s is the Lyapunov function for the slow subsystem :

$$V_s = z^T P z + \sigma^T Q_o \sigma$$

and V_f is the Lyapunov function for the fast subsystem :

$$V_f = \epsilon x^T \tilde{P} x$$

where P and $\tilde{P} \in \mathbb{R}^{2n \times 2n}$ are positive-definite matrices satisfying the Lyapunov

equations

$$\Pi^T P + P \Pi = -2 Q_1 \tag{7.5.8a}$$

$$\left[\tilde{A} - C K^o\right]^T \tilde{P} + \tilde{P} \left[\tilde{A} - C K^o\right] = -2 Q_2 \tag{7.5.8b}$$

Taking the time derivatives of both sides of (7.5.7) along the trajectories of (7.5.6) and rearranging, yields

$$\dot{U}(t) = -2z^T Q_1 z - 2x^T Q_2 x + 2x^T \Gamma^{*T} P z + 2\sigma^T \left[B^{*T} P z + Q_o \dot{\sigma}\right] \tag{7.5.9}$$

If $\dot{\sigma}$ satisfies (7.3.28), then

$$\sigma^T \left[B^{*T} P z + Q_o \dot{\sigma}\right] = 0$$

and (7.5.9) becomes

$$\dot{U}(t) = -2z^T Q_1 z - 2x^T Q_2 x + 2x^T \Gamma^{*T} P z \tag{7.5.10}$$

For stability it is required that $\dot{U} \leq 0$. It is clear from (7.5.10) that the term $x^T \Gamma^{*T} P z$ represents the interaction between the slow subsystem and the fast subsystem. Computing $\Gamma^{*T} P$ gives

$$\Gamma^{*T} P = \begin{bmatrix} (\alpha_o - a_1) P_2 & (\alpha_o - a_1) P_3 \\ 0 & 0 \end{bmatrix}$$

where P_2 and P_3 are obtained from the solution of the Lyapunov equation (7.5.8a) i.e. $P = \begin{bmatrix} P_1 & P_2^T \\ P_2 & P_3 \end{bmatrix}$. For $Q_1 = diag\,(Q_{11}, Q_{12})$, this solution is given by

$$P_2 = Q_{11} \Lambda_p^{-1} \quad \text{and} \quad P_3 = Q_{12} \Lambda_v^{-1} + Q_{11} \Lambda_p^{-1} \Lambda_v^{-1} \tag{7.5.11}$$

To simplify the computation and the analysis, we choose

$$Q_{11} = K^* \Lambda_v \Lambda_p \quad \text{and} \quad Q_{12} = K^* \Lambda_p \tag{7.5.12}$$

where K^*, Λ_p and $\Lambda_v \in \mathbb{R}^{n \times n}$ are diagonal positive-definite matrices. By using these in (7.5.11), we get

$$P_2 = K^* \Lambda_v \quad \text{and} \quad P_3 = K^* \Lambda_p \Lambda_v^{-1} + K^* \tag{7.5.13}$$

In order to obtain $\dot{U}(t) \leq 0$, we rearrange (7.5.10) in the quadratic form

$$\dot{U} = \begin{bmatrix} z^T & x^T \end{bmatrix} \begin{bmatrix} -2Q_1 & P\,\Gamma^* \\ \Gamma^{*T}P & -2Q_2 \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} \tag{7.5.14}$$

Let $\Omega^* = \begin{bmatrix} -2Q_1 & P\,\Gamma^* \\ \Gamma^{*T}P & -2Q_2 \end{bmatrix}$. Since P is a function of Q_1, if we can choose Q_1 and Q_2 such that Ω^* is negative-semidefinite, then the stability of the full-order closed-loop system will be ensured, i.e. the equilibrium points $z = x = 0$ are stable. As in Theorem 4.1 (Chapter 4), we can show by La Salle's Theorem, that the equilibrium points of the system (7.5.6) are asymptotically stable, i.e. $z(t), x(t) \to 0$ *as* $t \to \infty$.

Now we will discuss, how the elements of Q_1, Q_2 can be selected to ensure that Ω^* is negative-definite. For decoupled control, by row and column permutations corresponding to orthogonal similarity transformations, Ω^* can be written in the form of a block diagonal matrix with n (4×4) blocks along the diagonal, i.e. $\Omega^* = diag\,(\Omega_i^*)$. Then Ω^* is negative-definite if and only if $-\Omega_i^*$ is positive-definite. According to Sylvester's criteria, $-\Omega_i^*$ is positive-definite if and only if all its principal minors are positive. By using (7.5.13) with $K^* = diag\,(k_i^*)$, $\Lambda_p = diag\,(\lambda_{p_i})$, $\Lambda_v = diag\,(\lambda_{v_i})$ and choosing $Q_2 = diag\,(q_{2i})$, then $-\Omega_i^*$ can be found to be:

$$-\Omega_i^* = \begin{bmatrix} 2k_i^*\lambda_{p_i}\lambda_{v_i} & 0 & -k_i^*\lambda_{v_i}(\alpha_{o_i}-a_{1_i}) & 0 \\ 0 & +2k_i^*\lambda_{p_i} & -k_i^*\left(\dfrac{\lambda_{p_i}}{\lambda_{v_i}}+1\right)(\alpha_{o_i}-a_{1_i}) & 0 \\ -k_i^*\lambda_{v_i}(\alpha_{o_i}-a_{1_i}) & -k_i^*\left(\dfrac{\lambda_{p_i}}{\lambda_{v_i}}+1\right)(\alpha_{o_i}-a_{1_i}) & 2q_{2i} & 0 \\ 0 & 0 & 0 & 2q_{2i} \end{bmatrix}$$

From this, it is clear that the first and second principal minors are positive. Next, we find a condition that will ensure that the third and fourth principal minors are positive.

We write

$$\det(-\Omega_i^*) = 2q_{2i}\ (3rd\ principal\ minor)$$

where the 3rd principal minor is given by

$$3rd\ principal\ minor = 8k_i^{*2}\lambda_{p_i}{}^2\lambda_{v_i}q_{2i} - 2k_i^{*3}\lambda_{p_i}\lambda_{v_i}(\alpha_{o_i}-a_{1_i})^2\left[\left(\frac{\lambda_{p_i}}{\lambda_{v_i}}+1\right)^2+\lambda_{v_i}\right]$$

Therefore, it follows that for $-\Omega_i^*$ to be positive-definite, we require

$$q_{2i} > \frac{k_i^*(\alpha_{o_i}-a_{1_i})^2\left[\left(\frac{\lambda_{p_i}}{\lambda_{v_i}}\right)^2+\lambda_{v_i}\right]}{4\lambda_{p_i}}$$

So if q_{2i} is chosen to satisfy the above inequality, then Ω^* will be negative-definite which, in turn, will ensure asymptotic stability of the full-order closed-loop system.

7.6 Control Scheme Implementation

In order to implement the adaptive control τ_m it is required that the system matrices β_0, β_1, β_2, α_0, α_1, α_2, a_0, and a_1 be known. These matrices are identified on-line using the reduced-order flexible system (7.3.12). Since these matrices are assumed to be slowly time-varying, we can approximate the system (7.3.12) by a linear piecewise time-invariant model which has the same outputs and inputs as (7.3.12) and a similar dynamical structure during the interval Δt_j :

$$\ddot{\theta} = -A_0\theta - A_1\dot{\theta} + A_2\tau_o + A_3\dot{\Upsilon}_o + A_4\ddot{\Upsilon}_o + A_5\tau_1 \qquad (7.6.1)$$

where $A_0 - A_5$ are unknown parameters to be identified on-line using least-squares identification. These parameters are then used to obtain the coefficient matrices of the system and to generate the input control to the manipulator.

The model (7.6.1) can be written as

$$\ddot{\theta} = \hat{\Theta}^T\ \Psi \qquad (7.6.2)$$

where $\hat{\Theta}$ is the vector of parameters to be identified.

$$\hat{\Theta}^T = \left[-A_o \;\; -A_1 \;\; A_2 \;\; A_3 \;\; A_4 \;\; A_5\right]$$

and Ψ is the vector of measurements at time t_j written as

$$\Psi^T = \left[\theta^T \;\; \dot{\theta}^T \;\; \tau_o^T \;\; \dot{\Upsilon}_o^T \;\; \ddot{\Upsilon}_o^T \;\; \tau_1^T\right]$$

The method of recursive identification (as described in Chapter 3) is used for identification. Note that the current measurements θ and $\dot{\theta}$ and the previous parameter estimates are used to compute $\dot{\Upsilon}_o$ and $\ddot{\Upsilon}_o$. By comparing (7.3.12) and (7.6.1), we have

$$\hat{\beta}_o = A_1 \;,\; \hat{\beta}_1 = A_o \;,\; \hat{\beta}_2 = A_2 \;,\; \hat{\epsilon}\,\hat{\beta}_2\,\hat{a}_1^{-1}\,\hat{a}_o = A_3 \tag{7.6.3a}$$

and

$$\hat{\epsilon}\,\hat{\beta}_2\,\hat{a}_1^{-1} = A_4 \;,\; \hat{\epsilon}\,\hat{\beta}_2 = A_5 \tag{7.6.3b}$$

From (7.6.3) we get

$$\hat{\epsilon} = A_5 A_2^{-1} \;,\; \hat{a}_1^{-1} = A_4 A_5^{-1} \;,\; \hat{a}_o = A_3 A_4^{-1} \tag{7.6.4}$$

By using the relationship between (7.2.8) and (7.2.11), we can write

$$\hat{\alpha}_o = A_5^2\, A_4^{-1} \left[A_5 - A_2 A_4\right]^{-1} \tag{7.6.5a}$$

$$\hat{\alpha}_2 = A_4 A_5^{-1}\,\hat{\alpha}_o \left[A_1 - A_3 A_4^{-1}\right] \tag{7.6.5b}$$

$$\hat{\alpha}_1 = G\left(\hat{\alpha}_o \,,\, \hat{a}_1, g \,,\, \theta\right) \tag{7.6.5c}$$

where g is the gravity constant. Now we can write the input controllers, τ_o and τ_1 in terms of the identified parameters as

$$\tau_o = A_2^{-1} \left[\ddot{\theta}_d + K_v\,\dot{e} + K_p\,e + A_o\,\dot{\theta}_d + A_1\theta_d + S(t)\right] \tag{7.6.6}$$

where

$$K_v = \Lambda_v - A_1 \qquad K_p = \Lambda_p - A_o$$

and

$$\tau_1 = -A_4\, A_5^{-1} \left[\ddot{\Upsilon}_o + A_3 A_4^{-1}\,\dot{\Upsilon}_o\right] \tag{7.6.7}$$

where $\Upsilon_o(\theta, \dot{\theta})$ is computed from (7.3.10) in terms of θ, $\dot{\theta}$ and the identified parameters. The fast control is

$$\tau_f = -k_1\,\xi - k_2\,\dot{\xi} \tag{7.6.8}$$

with

$$\xi = \psi - \Upsilon(\theta, \dot{\theta}) \tag{7.6.9}$$

where ψ is measured and $\Upsilon(\theta,\dot{\theta}) = \Upsilon_o + \epsilon\ \Upsilon_1$ is computed.

7.7 Numerical Example

To illustrate the use of the adaptive control scheme presented in this chapter, we consider a single link manipulator with a flexible joint as shown in Fig. 7.1. The manipulator consists of an actuator of inertia J_m and viscous damping B_m, connected through a gear box of ratio N:1, to a rigid link of mass m and length l. The elasticity in the joint is modeled as a torsional spring of stiffness k. The equations of motion for such a manipulator can be written [75] as

$$\frac{1}{3}ml^2\ddot{\theta} + B_l\ \dot{\theta} + \frac{mgl}{2}\ \sin\theta + k\,(\theta + \frac{\phi}{N}) = 0 \tag{7.7.1a}$$

and

$$J_m\ \ddot{\phi} + B_m\ \dot{\phi} + \frac{k}{N}(\theta + \frac{\phi}{N}) = \tau_m \tag{7.7.1b}$$

The nominal values of the manipulator parameters are given in [75] as: l=3 meter, m=10 kg, ϵ=0.01, N=100, B_m=0.015$N.m/rad.\ \sec^{-1}$, B_l=36$N.m./rad.\ \sec^{-1}$, J_m=0.04$kg-m^2$ and g=9.8$m/\sec^2$. Equations (7.7.1) can be converted into the standard singular perturbation form :

$$\ddot{\theta} = -J_l^{-1}\ \psi - J_l^{-1}\ B_l\ \dot{\theta} - J_l^{-1}\ R_l\ \sin\theta \tag{7.7.2a}$$

and

$$\epsilon\ \ddot{\psi} = -\left(J_l^{-1} + \frac{J_m^{-1}}{N^2}\right)\ \psi - \epsilon\ J_m^{-1}\ B_m\ \dot{\psi} - \left(J_l^{-1}\ B_l - J_m^{-1}\ B_m\right)\dot{\theta}$$
$$- J_l^{-1}\ R_l\ \sin\theta + \frac{J_m^{-1}}{N}\ \tau_m$$

or equivalently

$$\epsilon\ \ddot{\psi} = -\alpha_o\ \psi - \epsilon\ a_o\ \dot{\psi} - \alpha_2\ \dot{\theta} - \alpha_1\ \sin\theta + a_1\ \tau_m \tag{7.7.2b}$$

where

$$\alpha_o = J_l^{-1} + \frac{J_m^{-1}}{N^2} = 0.03583$$

$$\alpha_1 = J_l^{-1}R_l = 4.9$$
$$\alpha_2 = J_l^{-1}B_l - J_m^{-1}B_m = 0.825$$
$$a_o = J_m^{-1}B_m = 0.375$$
$$a_1 = \frac{J_m^{-1}}{N} = 0.25$$
$$J_l^{-1} = 0.0333$$

Simulation Results

The manipulator of Fig. 7.1 was assumed to be initially at rest with $\theta, \dot{\theta}, \ddot{\theta} = 0$. The initial condition of the fast variable ξ was set $\neq 0$, otherwise the dynamics of the fast mode will not excited at all. The initial values of the feedback gains for the slow mode were chosen to be $k_p = 16$ and $k_v = 8$. For the fast mode, we chose $k_1 = 100$ and $k_2 = 20$ such that the eigenvalues of the fast subsystem are placed far in the left-half of the complex plane. The reference trajectory given in Chapter 4 was generated. This is given by

$$\theta = 1.57 + 7.8539exp(-t) - 9.4248\exp(-t/1.2)$$

The simulation was performed on a VAX11/780 computer with an inner loop sampling rate of 2 ms.

First, the composite adaptive control law (7.5.4a), without the corrective controller (7.5.4c) was applied to the full-order manipulator system (7.7.2a-b). The response of the closed-loop system is shown in Fig. 7.2 . It is seen that the response is oscillatory. Fig. 7.3 shows the system response when the full composite control (7.5.4a) with the corrective and robust controllers was applied. This illustrates that without the corrective controller, the closed-loop system cannot have stable tracking performance and use of the corrective controller is necessary. Note that it can be seen from the figures, that tracking is achieved after an initial transient period during which the fast variables converge to the manifold. Fig. 7.4 shows the response of the system with 66% variation in the system parameters α_o and a_o . This illustrates the robustness of the control scheme.

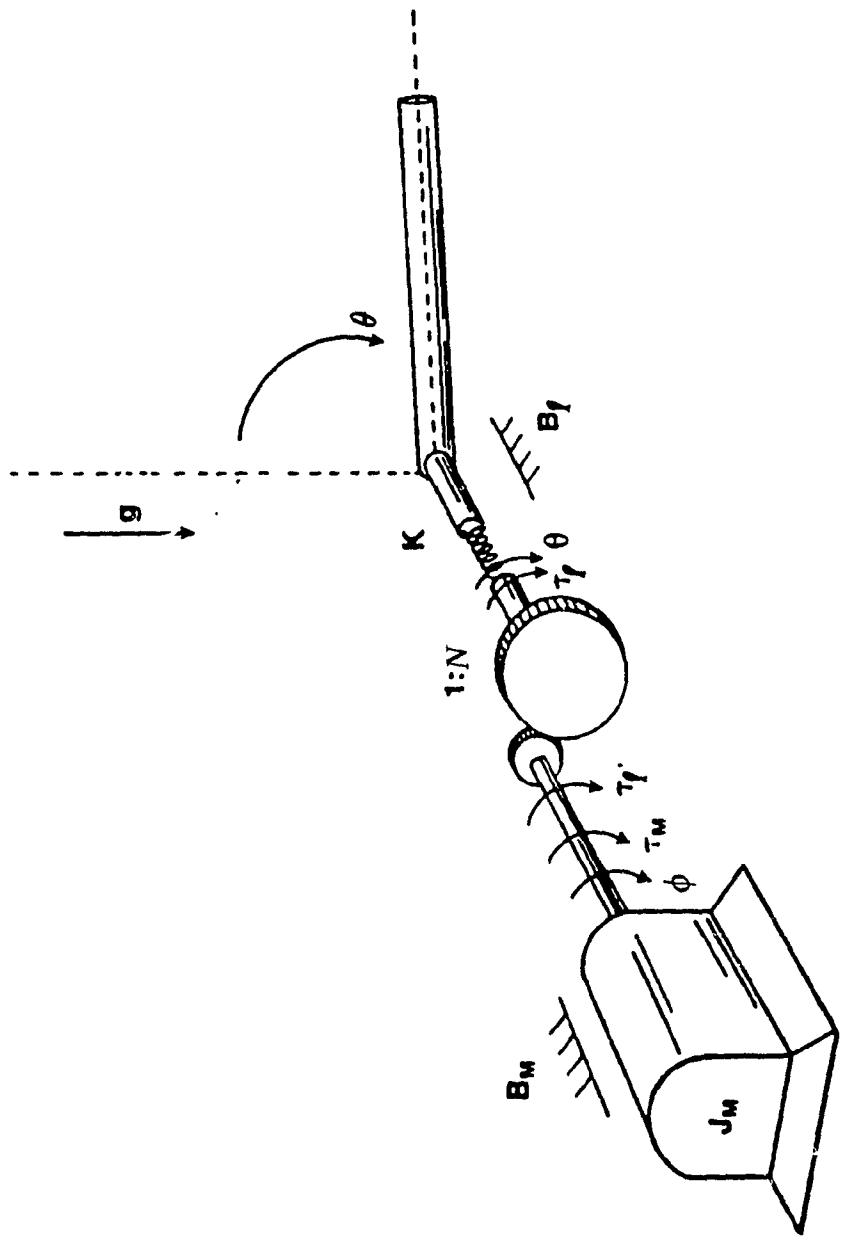

Fig.(7.1) Single-link manipulator with joint flexibility

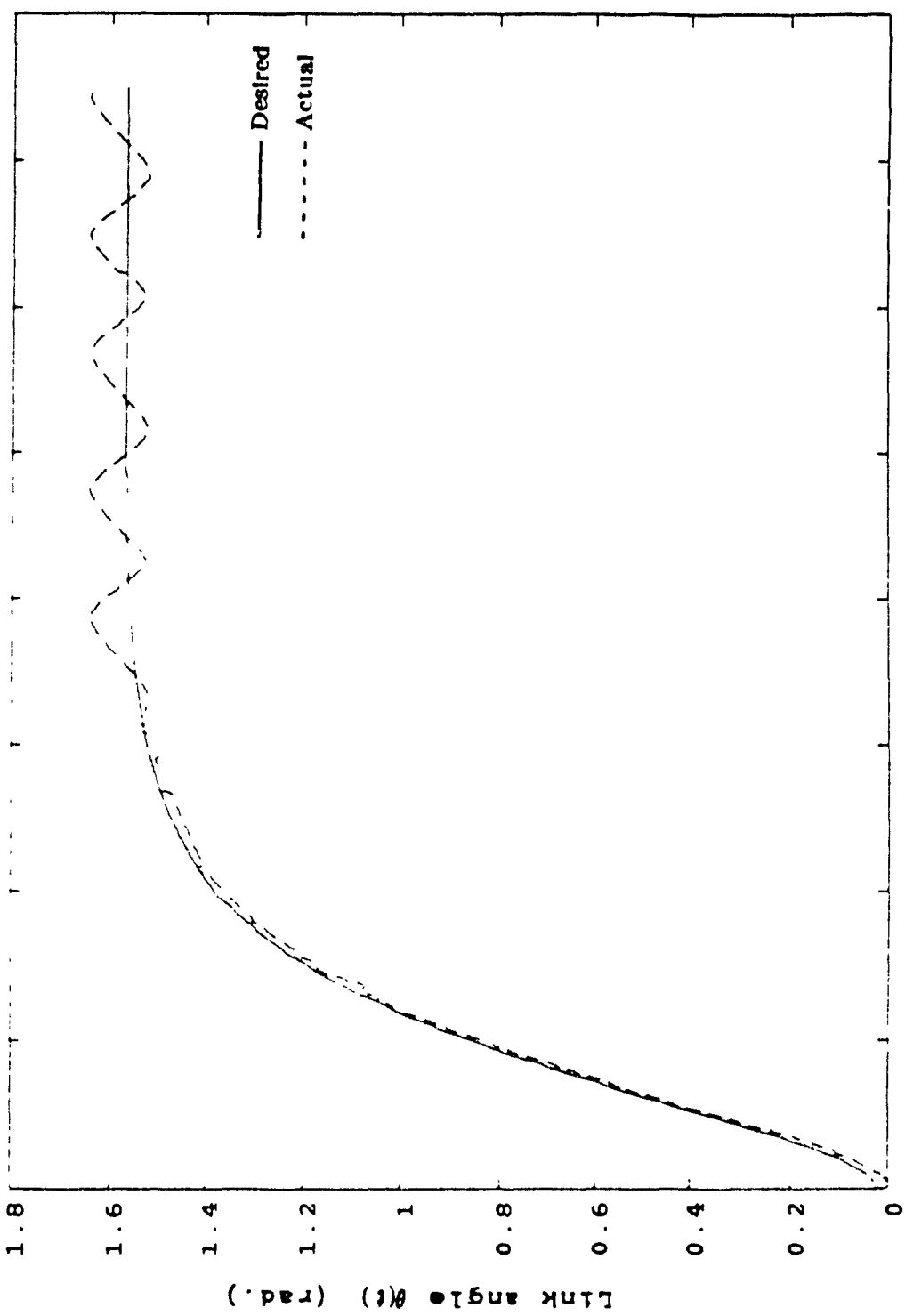

Fig.(7.2) Response of the link angle without corrective control

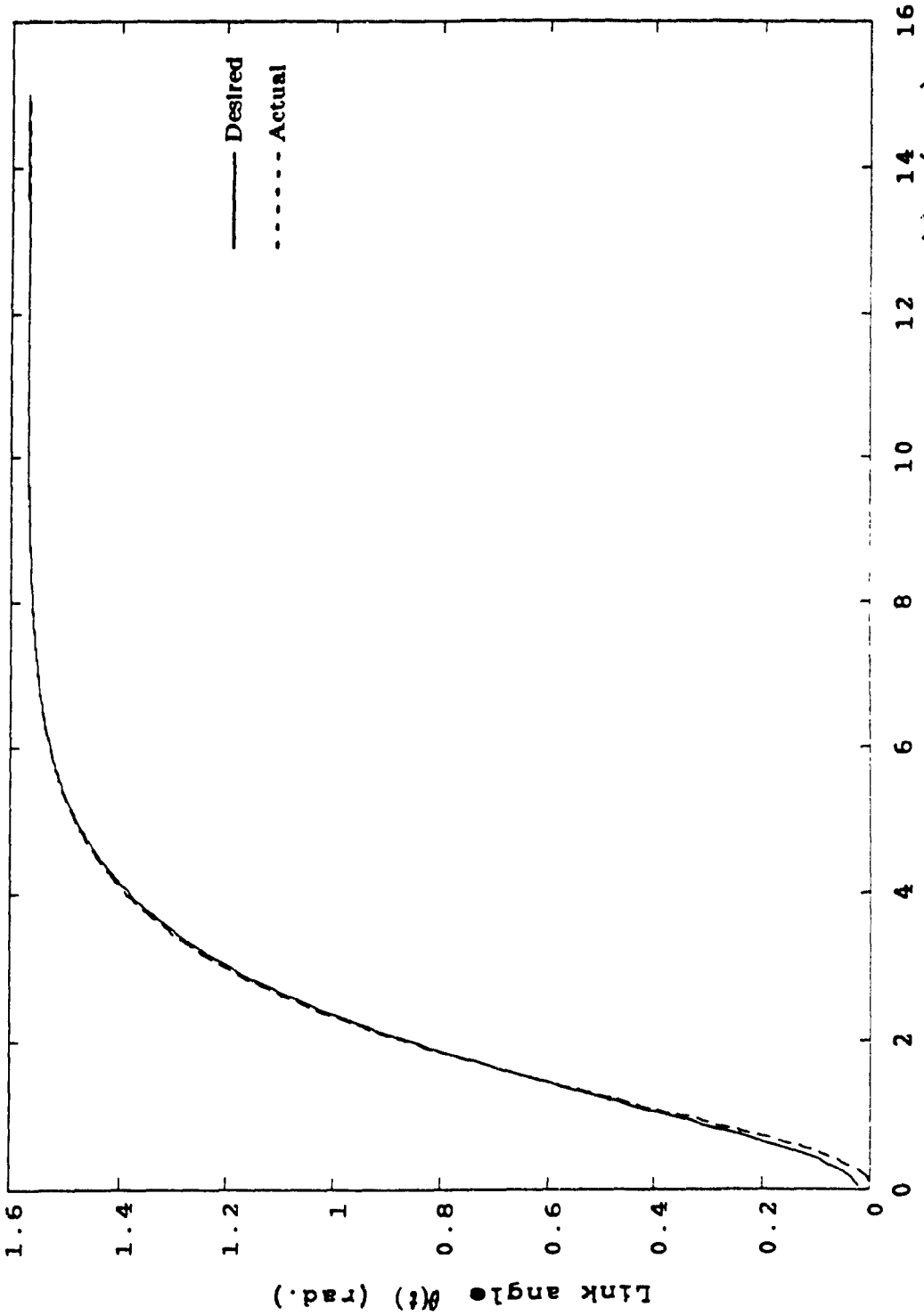

Fig.(7.3) Response of the link angle with corrective control

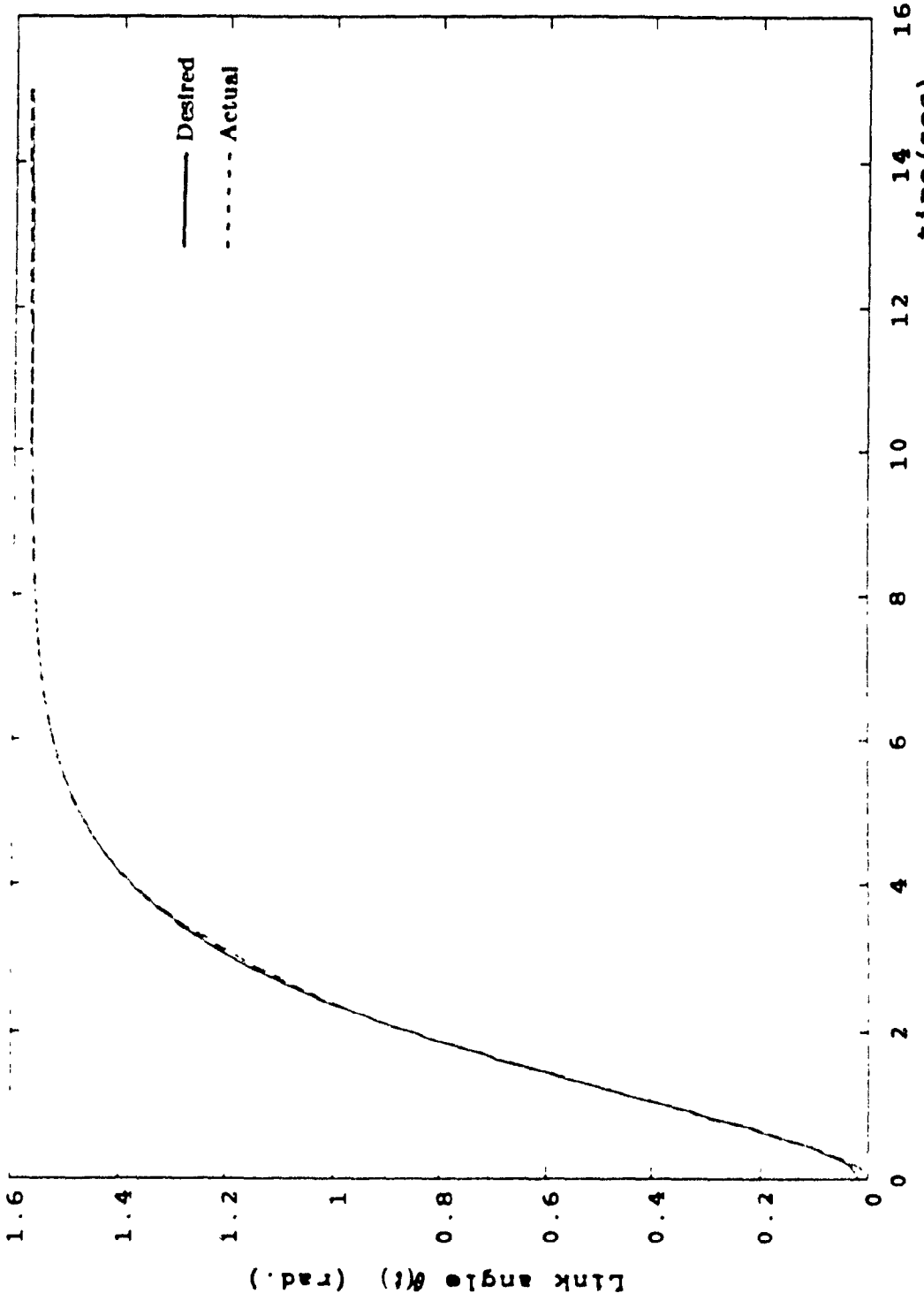

Fig.(7.4) Response of the link angle with parameter variations

CHAPTER EIGHT

CONCLUSIONS AND FUTURE WORK

8.1 CONCLUSIONS

This thesis deals with the problem of controlling robot manipulators. A new algorithm for adaptively controlling the motion of a manipulator, with special attention to robustness of the controller, is developed. The main contributions of the research described in this thesis have been to provide a reliable and powerful technique to maintain closed-loop stability and achieve trajectory tracking in the presence of unmodeled dynamics and parameter variations.

The adaptation process used least squares identification to generate a set of piecewise linear, time-invariant models on-line for the nonlinear time-varying manipulator dynamics. The identified parameters of these models were used to update the input feedforward and feedback controllers.

The adaptive controller was designed using a decentralized control The effects of coupling between the joints and other unmodeled dynamics were taken into account by including a correction term in the control law which attempts to reduce the errors resulting from these effects to zero. The gain and weighting matrices were selected and updated on-line to ensure that the errors $e(t) \rightarrow 0$ *as* $t \rightarrow \infty$.

The decoupled adaptive control scheme consisting of a number of decentralized controllers has several advantages over a single centralized controller. A major advantage is that the joint control algorithms require much less computation than algorithms for centralized control. Furthermore, due to the simplicity of the controller structure and the possibility of using parallel processing, the decoupled control scheme can be implemented on n single and fast microprocessors at a high sampling rate, thus improv-

ing the system performance.

Another major advantage of the decoupled control scheme is its reliability and failure tolerant feature. Suppose that one joint encoder gives erroneous readings of the joint position In the centralized control system, this will affect the entire control action for all joints, where-as in the decoupled system, only one control loop is effected and the remaining joint controllers are unaffected In general, however, the decoupled controller will yield a larger tracking error than the centralized controller, since it has to compensate for the unmodeled joint couplings However, by incorporating a robust control feature, we have been able to compensate against these effects to some extent Accordingly, we conclude that the decoupled control algorithm, at the price of a relatively small decrease in the manipulator performance, is considerably simpler than the centralized control law

A Cartesian robust adaptive control scheme was designed in Chapter 6 to control the motion of the end-effector of the manipulator in Cartesian space. Note that although the Cartesian controllers are independent and decoupled, the equivalent joint control law is coupled This is because of the fact that to control the end-effector motion in each Cartesian direction, appropriate torques must applied to all joints simultaneously

As can be seen from the simulation results given in Chapters 4-6, the controllers achieved tracking of the reference trajectory with very small error and in the presence of unmodeled dynamics and payload variations The results also justify the underlying assumption that the dynamic parameters are slowly time-varying Note that, in the simulation results the method of bounded uncertainty achieved better tracking of the reference trajectory with less error in comparison with the method of characteristic polynomial assignment However, more computation is needed in the former since computation of the Q matrix is done iteratively On the other hand, the accuracy of the second method depends on the selections of the desired characteristic polynomial So that

another choice of the desired characteristic polynomial may give better results.

In Chapter 7, the algorithm has been extended to the case of manipulators with flexible joints. The dynamics of the full-order system was constructed in a singularly perturbed form in terms of slow and fast variables, and a composite control strategy was designed. The composite control scheme consists of a slow controller and a fast controller. The slow controller comprises a rigid control (τ_0) based on the rigid model, and a corrective controller (τ_1) based on the corrective slow subsystem, to compensate for certain $O(\epsilon)$ nonlinearities. Since the dynamics of the fast subsystem are oscillatory, a fast control τ_f has been designed to ensure asymptotic stability of the fast subsystem and to guarantee that the fast modes will be on the slow manifold and the full system dynamics are governed by a reduced-order flexible system model.

The composite stability analysis gave us a way to design robust control for the slow subsystem to compensate for the effects of unmodeled dynamics which result from neglecting the $O(\epsilon^2)$ terms and other unmodeled dynamics or parameter variations arising from the mismatch in the estimated parameters and the actual ones. The stability of the full-order closed-loop system is achieved by selecting the weighting matrices on-line and updating them in terms of the identified parameters of the reduced-order flexible model. The control strategy is based on the identification of the reduced-order flexible system in a single step identification process. Note that the implementation of the composite control only needs measurements of θ and $\dot{\theta}$ and the reference input. This makes the scheme easy for real-time applications. Simulation results show the validity and robustness of the composite control scheme.

In summary, the controller simplicity and ease of implementation, its decoupled nature and robustness to parameter variations are desirable attributes which make the proposed adaptive control schemes feasible for solving many of the complex manipulator control problems.

8.2 SUGGESTIONS AND FUTURE WORK

The ideas in this thesis can be extended in several directions:

1- The algorithm developed in Chapter 4, can be extended to include the force control problem. Problems of position and force control can then be tackled together by designing a hybrid controller. When a manipulator is constrained by reaction surfaces in some directions (e.g. during compliant control), an interesting approach would be to modify the dynamic model of the manipulator to incorporate the dynamics of the hand and the reaction surfaces.

2- The results of Chapter 7 can be extended to control an n-link flexible joint manipulator. A fast robust controller can be added to the composite control input if the fast variable is excited along the trajectory because of uncertainties in the fast dynamics.

3- Further study is required to estimate the size of the time step for updating the identified models and the controller gains. This will result in further saving in the computational cost.

4- Another area of research is to develop a VLSI implementation of the control algorithm. This can result in significant speedup of the computations involved. This should be possible because of the relatively simple structure of the algorithm. Implementation of the algorithm involves a large number of matrix multiplications and additions, with many of the coefficients being floating point numbers. If the computations for evaluation of the models or the estimated parameters is implemented serially on a minicomputer or work station, then the computation time will be high and the system response will be slow. A special purpose system architecture is required to take advantage of the modularity of the algorithm in terms of computational complexity. In order to increase computational speed, operations have to be parallelized or pipelined using special purpose hardware with high capability of matrix multiplications. A possible system architecture for each joint is shown in Fig. 8.1. A bank of accelerators working as slaves do the number crunching simultaneously. Their number varies depending on the interaction

allowed between the joints. These accelerators may well be math-coprocessors of the master processor selected (i.e. 87 family of coprocessors for the 86 family of processors). The main processor acts as a master and has the function of a supervisor performing distribution and collection of new and computed coefficients from the accelerators, estimation of the parameters and communication with higher levels of control. Any special chips required for interfacing may be implemented in VLSI. The type of microprocessors to be used may be selected depending on detailed calculations of the required computational time. Fig. 8.2 shows the system architecture for the whole manipulator hardware, with the slaves corresponding to each joint communicating with the master in programmed I/O or interrupt mode. The above architecture will accelerate the response of the system by at least a factor of n over a standard serial implementation.

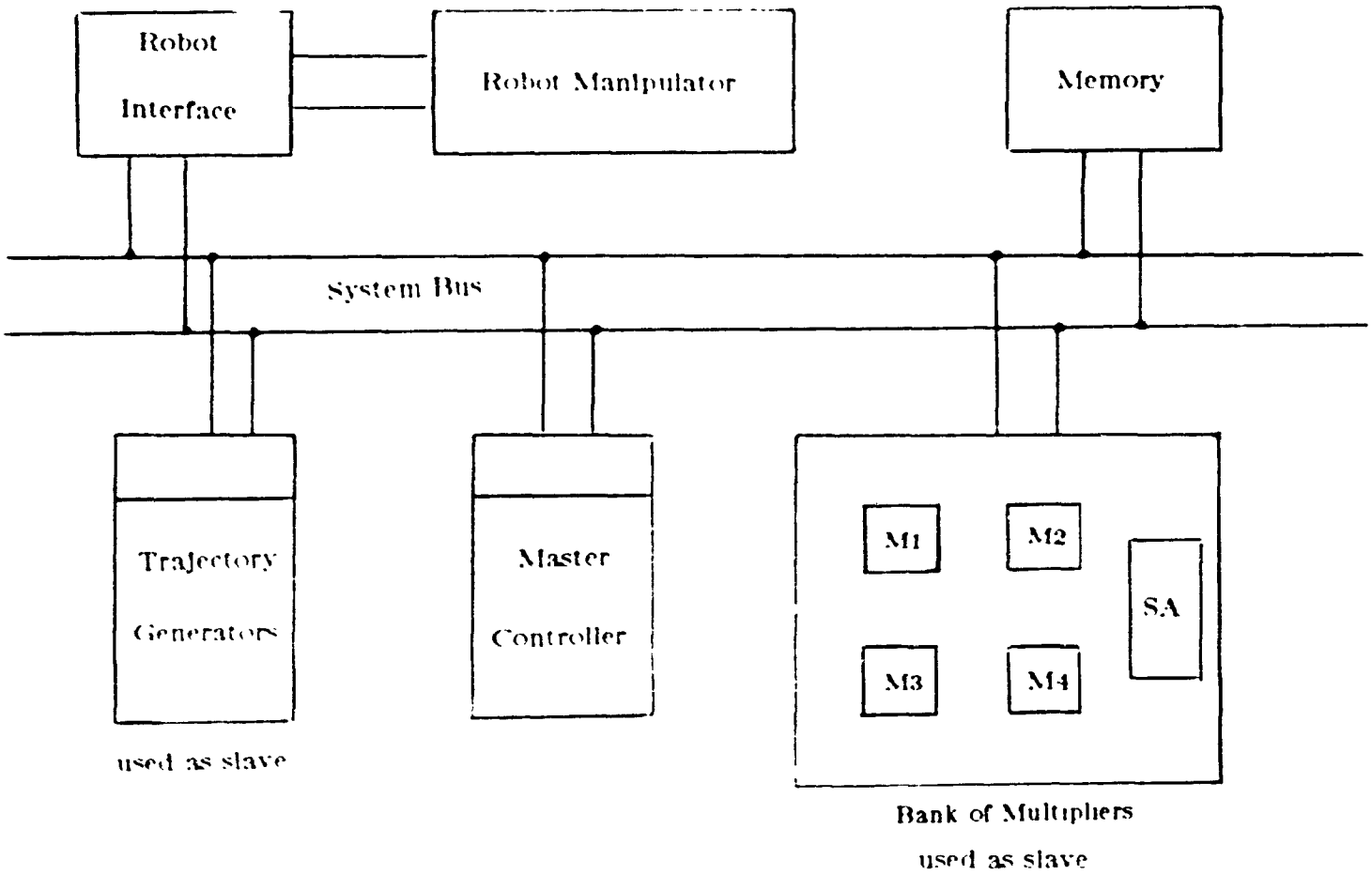

Fig.(8.1) System architecture for implementation of the adaptive controller for each joint

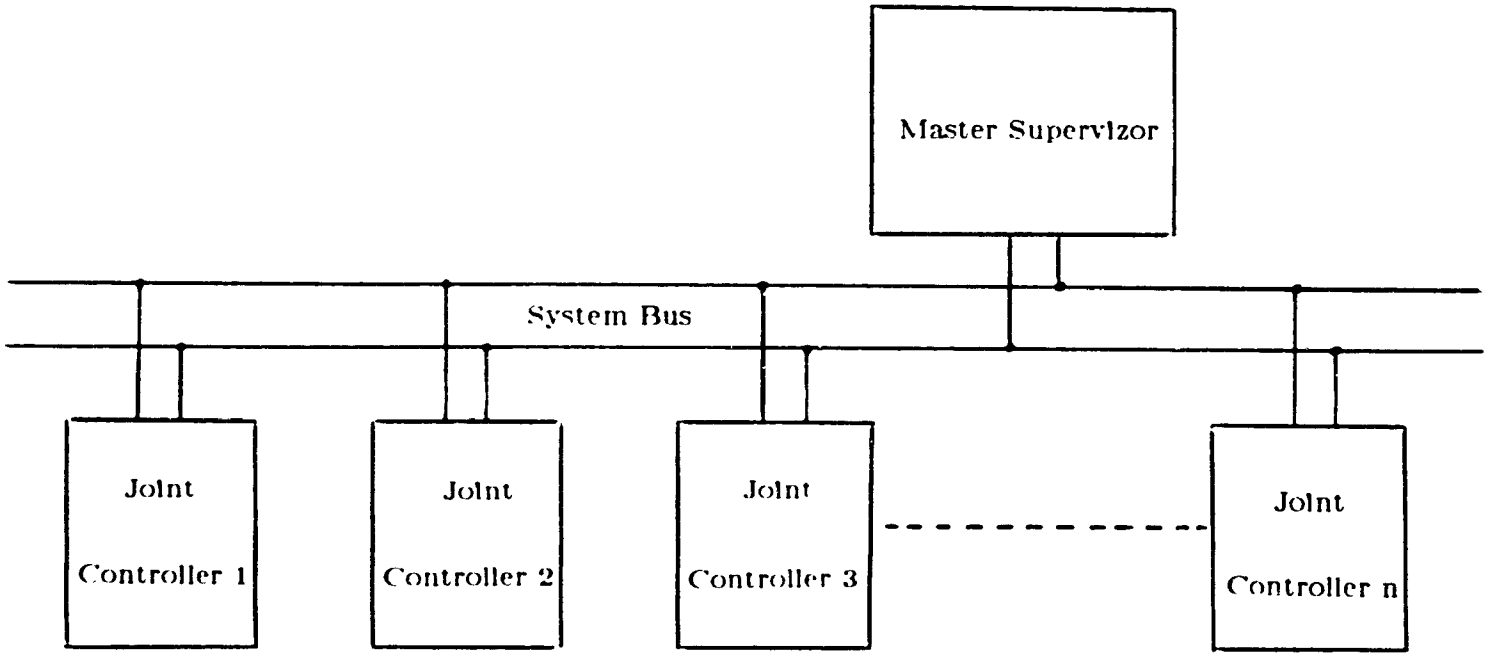

Fig.(8.2) System architecture for implementation of the adaptive controller for n-joint manipulator

REFERENCES

[1] J.Y Luh, M.W. Walker and R.P Paul, "On Line Computational Scheme for Mechanical Manipulators", ASME J Dyn. Syst. Measur & Control, Vol. 102, 1980.

[2] M Brady et al., "Robot Motion", MIT Press, Cambridge, MA and London, England, 1982

[3] R P Paul, "Robot Manipulators Mathematics, Programming and Control", MIT Press, Cambridge, MA, 1981

[4] L L Toepperwein, M T Blackman, W T Park, W R Tanner and W D Adolfson, "Robotics Application for Industry", USA 1983

[5] J J Craig "Introduction to Robotics, Mechanics and Control", Addison Wesley, 1986

[6] M W Walker and D E Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms" ASME J Dyn Syst Measur & Control, Vol 104 1982

[7] S N Singh and A A Schy "Robust Trajectory Following Control of Robotic System" ASME J Dyn Syst Measur & Control Vol 107 1985

[8] R P Paul Modeling Trajectory Calculation and Servoing of Computer Controlled arm" PhD dissertation stanford univ CA 1972

[9] D.E. Whitney "Resolved Motion Rate Control of Manipulator and Human Prostheses IEEE Trans Man MacLine System Vol MMS 10 1969

[10] J Y Luh M W Walker and R P Paul "Resolved Acceleration Control of Mechanical Manipulators" IEEE Trans on Automatic Control Vol AC 25 1980

[11] M Jamshidi H Seraji and Y T Kim Decentralized Control of Nonlinear Robot Manipulators" Int J R Robots Vol 4 No 1 4 Netherlands 1987

[12] M E Kahn and B Roth The Near Minimum Time Control of Open Loop Articulated Kinematic Chains ASME J Dyn Syst Measur and Control 1971

[13] J.Y. Luh and M. Walker, "Minimum-Time Along the Path for a Mechanical Arm", Proc. IEEE Conf. Decision Control, 1977.

[14] K.K.D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure Systems", IEEE Trans. Syst. Man. Cybern., Vol. SMC-8, 1978.

[15] W.J. Book, O. Neto and D. Whitney, "F/B Control of Two Beam Two Joint System with Distributed Flexibility", ASME J. Dyn. Syst. Measur. and Control, Dec 1975

[16] J.S. Albus,"A New Approach to Manipulator Control: The Cerebeller Model Articulation Controller (CMAC)", ASME J. Dyn. Sys. Measur and Control, 1975.

[17] B.J. Oh, M. Jamshidi and H. Seraji, "Decentralized Adaptive Control (Robot)", Proc. IEEE Int. Conf. on Robotics and Automation, Vol. 2, Philadelphia, USA, 1988.

[18] G.S. Saridis and C.S. Lee, "An Approximation Theory of Optimal Control for Trainble Manipulator", IEEE Trans Syst. Man Cybern , Vol SMC-9, 1979

[19] S.N. Singh, "Adaptive Model Following Control of Nonlinear Robotic Systems", IEEE Trans. of Automatic Control, Vol. AC-30, No 11, 1985

[20] S. Dubowsky and D.T. Deforges, "The Application of Model Referenced Adaptive Control to Robotic Manipulators", ASME J. Dyn Syst and Control, Vol 101, 1979.

[21] M. Takegaki and S. Arimoto, "An Adaptive Trajectory Control of Manipulators", Int. J. of Control, Vol 34, No 2, 1981

[22] H. Seraji, "Adaptive Control of Robotic Manipulators", JPL Engineering Memorandum 347-182, Jan 1986

[23] K.Y Lim and M Eslami, "Robust Adaptive Controller Designs for Robot Manipulator System", IEEE J of Robotics and Automation, Vol RA-3, No 1, 1987

[24] A. Liegeais, A. Fournier & M.J. Aldon, "Model Reference Control of High Velocity Industrial Robots", Joint Automatic Control Conf., San Francisco, 1980.

[25] A. Koivo & T.H Guo, "Adaptive Linear Controller for Robotic Manipulators", IEEE Trans. on Automatic Control, Vol. AC-28, No.2, 1983.

[26] A.J. Koivo, "Self-Tuning Manipulator Control in Cartesian Base Coordinate System", ASME J. Dyn Syst. Measur & Control, Vol. 107, 1985.

[27] C S G Lee and B.H Lee, "Resolved Motion Adaptive Control for Mechanical Manipulators", ASME J Dyn Syst. Measur & Control, Vol 106, 1984.

[28] I D Landau, "A Survey of Model Reference Adaptive Techniques. Theory and Applications", Automatica, Vol 10, 1974

[29] L Ljung and T Soderstrom, "Theory and Practice of Recursive Identification", MIT Press, 1983

30 K Ogata, "Modern control Engineering", Prentice-Hall Inc , NJ, 1970

31 A Balestrino, G DeMaria & L Sciavicco "An Adaptive Model Following Control for Robotic Manipulator", ASME J of Dyn Syst Measur and Control, Vol 105, 1983

32 J Craig, P Hsu and S Sastry "Adaptive Control of Mechanical Manipulators", IELE Conference on Robotics and Automation San francisco 1986

33 J Craig, "Adaptive Control of Mechanical Manipulators" Addison-Wesley USA, 1988

34 D Orin and W W Schrader "Efficient Computation of the Jacobian for Robot Manipulators" Int J of Robotics Research Vol 3 No 4 1984

35 B Noble "Applied Linear algebra" Prentice Hall 1969

36 O Khatib "Commande Dynamique dans l Espace Operationnel des Robots Manipulateurs en Presence d Obstacles" These de Docteur Ingenieur Ecole Nationle Superieure de l Aeronautique et de l Espace ENSAE Toulouse France

[37] R.V. Patel and M. Toda, "Quantitative Measures of Robustness for Multivariable Systems", Proc. Joint Automatic Control Conference, San Francsco, 1980.

[38] H. Seraji, "A New Approach to Adaptive Control of Manipulators", ASME J. Dyn. Syst. Measur. & Control, Vol. 109, 1987.

[39] R.V. Patel and N. Munro, "Multivariable System Theory and Design", Oxford, Pergamon Press, 1982.

[40] B. Yuan and W. Book, "Decentralized Adaptive Control of Robot Manipulators with Robust Stabilization Design", American Control Conference, Pittsburgh, PA, 1988.

[41] R. Yedavalli and Z. Liang, "Reduced Conservatism in the Ultimate Boundedness Control of Mismatched Uncertain Linear System", ASME J. Dyn. Syst. Measur. & Control, Vol. 109, 1987.

[42] R.L. Burden, J.D. Faires and A.C. Reynolds, "Numerical Analysis", second edition, Prindle Weber & Schmidt, USA, 1981.

[43] H. Seraji, "Adaptive Independent Joint Control of Manipulators: Theory and Experiment", IEEE Int. Conference on Robotics and Automation", 1988.

[44] P. Eykhoff, "System Identification: Parameter and State Estimation", London, NewYork, Wiley Interscience, 1974.

[45] E.G. Gilbert and I.J. Ha, "An Approach to Nonlinear Feedback Control with Applications to Robotics", Proc. IEEE Conference on Decision and Control, San Antonio, Texas, 1983.

[46] B.B Peterson and K. Narendra, "Bounded Error Adaptive Control", IEEE Trans. on Automatic Control, Vol. Ac-27, 1982.

[47] M. Spong and M. Vidyasagar, "Robust Nonlinear Control of Robot Manipulators", Proc. of 24th Conference on Decision and Control, Fort Landerdal, Fla., 1985.

[48] P. Misra, R.V. Patel and C.A. Balafoutis, "Robust Control of Robot Manipulators in Cartesian Space", American Control Conference, Philadelphia, 1988.

[49] J. La Salle and S. Lefschetz, "Stability by Lyapunov's Direct Method with Applications", New York , Academic Press, 1961.

[50] J.E. Slotine and W. Li, "Adaptive Manipulator Control: A Case Study", IEEE Trans. on Automatic Control, Vol. 33, No. 11, 1988.

[51] C.W. DeSilva and J. Van Winssen, "Least Squares Adaptive Control for Trajectory Following Robots", ASME J. Dyn. Syst. Measur. & Control, Vol. 109, 1987.

[52] M. Liu, W. Chang and L. Zhang, "Dynamic and Adaptive Force Controllers for Robotics Manipulators", IEEE Int. Conference on Robotics and Automation, Vol. 3, 1988.

[53] A. Lobbezoo and et al., "Robot Control Using Adaptive Transformations", IEEE J. of Robotics and Automation, Vol. 4, No. 1, 1988.

[54] M.H. Raibert and J.J. Craig, "Hybrid Position/Force Control of Manipulators", ASME J. Dyn. Syst. Measur. & Control, Vol.102, 1981.

[55] R. Horowitz and M. Tomizuka, "Discrete Time Model Reference Adaptive Control of Mechanical Manipulators", Comput. Eng., Vol. 2, 1982.

[56] M. Tomizuka, R. Horowitz and Y.D. Lundau, "On the use of model reference adaptive control techniques for mechanical manipulators", 2nd IASTED Symposium on Identification, Control and Robotics, Davas, Switzerland, 1982.

[57] M. Tomizuka and R. Horowitz, "Model Reference Adaptive Control of Mechanical Manipulators", Proc. IFAC Symp Adaptive Systems in Control and Signal Processing, San Francisco, 1983.

[58] A.J. Koivo, "Force-Velocity Control with Self-Tuning for Robotic Manipulators", IEEE Conference on Robotics and Automation, San Francisco, 1986

[59] G. Leininger, "Self-Tuning Control of Manipulators", Int. Symposium on Advanced Software in Robotics, Liege, belgium, 1983.

[60] G. Leininger, "Adaptive Control of Manipulators using Self-Tuning Methods", 1st ISRR, New Hampshire, 1983.

[61] P.G. Backes, G. Leininger and C.H. Chung, "Real Time Cartesian Coordinate Hybrid Control of a PUMA 560 Manipulator", IEEE Conference on Robotics and Automation, St. Louis, 1985.

[62] K. Kubo and T. Ohmae, "Adaptive Trajectory Control of Industrial Robots", 15th ISIR, Tokyo, Japan, 1985.

[63] C.S.G. Lee, M.J. Chung and B.H. Lee, "Adaptive Control for Robot Manipulators in Joint and Cartesian Coordinates", IEEE Conference on Robotics, Atlanta, 1984.

[64] K.Y. Lim and M. Eslami, "Adaptive Controller Designs for Robot Manipulator Systems Using Lyapunov Direct Method", IEEE Trans. on Automatic Control, Vol. 30, No. 12, 1985.

[65] A. Balestrino, G. De Maria and L. Sciavicco, "Adaptive Control of Robotic Manipulators", AFCET Congres Automatique, Nontes, France, 1981.

[66] J.J. Slotine, "On Modeling and Adaptation in Robot Control", Proc. of the IEEE Conference on Robotics and Automation, San Francisco, 1986.

[67] R.P. Anex, Jr and Mount Hubbard, "Modeling and Adaptive Control of a Mechanical Manipulator", ASME J. Dyn. Syst. Measur. & Control, Vol.106, 1984.

[68] H. Seraji, "Decentralized Adaptive Control of Manipulators: Theory, Simulation, and Experimentation", IEEE Trans. on Robotics and Automation, Vol. 5, No. 2, 1989.

[69] A. De Luca, "Dynamic Control of Robots with Joint Elasticity", IEEE Int. Conference on Robotics and Automation, Vol. 1, Philadelphia, PA, 1988.

[70] C. De Wit and O. Lys, "Robust Control and Parameter Estimation of Robots with Flexible Joints", IEEE Int. Conference on Robotics and Automation, Vol. 1, Philadelphia, PA, 1988.

[71] L.M. Sweet and M.C. Good, "Re-definition of the Robot Motion Control Problem: Effect of Plant Dynamics, Drive System Constraints, and User Requirements", Proc. 23rd IEEE Conf. Decision and Control, Las Vegas, NV, 1984.

[72] E.I. Rivin, "Effective Rigidity of Robot Structure: Analysis and Enhancement", Proc. American Control Conference, Boston, MA, June 1985.

[73] K. Khorasani and P.V. Kokotovic, "Feedback Linearization of a Flexible Manipulator Near its Rigid Body Manifold", Systems and Control Letters, Vol. 6, 1985.

[74] K. Khorasani and M.W. Spong, "Invariant Manifolds and Their Applications to Robot Manipulators with Flexible Joints", Proc. 1985 IEEE Int. Conference on Robotics and Automation, St. Louis, 1985.

[75] M.W. Spong, K. Khorasani and P.V. Kokotovic, "An Integral Manifold Approach to the Feedback Control of Flexible Joint Robots", IEEE J. of Robotics and Automation, Vol. RA-3, No. 4, 1987.

[76] H.B. Kuntze and A.H.K. Jacubasch, "Control Algorithms for Stiffening on Elastic Industrial Robot", IEEE J. of Robotics and Automation, Vol. RA-1, No. 2, 1985.

[77] S. Nicosia, F. Nicolo and D. Lentini, "Dynamical Control of Industrial Robots with Elastic and Dissipative Joints", 8th IFAC World Congress, Kyoto, Japan, 1981.

[78] M.W. Spong, "Modeling and Control of Elastic Joint Robots", ASME Winter Annual Meeting, Anaheim, 1986.

[79] R. Marino and S. Nicosia, "On the Feedback Control of Industrial Robots with Elastic Joints: A singular Perturbation Approach", Technical Report, Univ Rome, "Tor Vergata", R-81 01, 1981.

[80] J.J. Slotine and S Hong, "Two-Time Scale Sliding Control of Manipulators with

Flexible Joints", American Control Conference, Seattle, WA, 1986.

[81] S. Nicosia, P. Tomei and A. Tornambe, "Feedback Control of Elastic Robots by Pseudo-Linearization Techniques", 25th IEEE Conf. Decision and Control, Athens, Greece, 1986.

[82] P. Tomei, S. Nicosia and A. Ficola, "An Approach to the Adaptive Control of Elastic at Joints Robots", 3rd IEEE Int. Conf. Robotics and Automation, San Francisco, 1986.

[83] G. Cesareo and R. Marino, "On the Controllability Properties of Elastic Robotics", 6th Int. Conf. Analysis and Optimization of Systems, INRIA, Nice, 1984.

[84] P.V. Kokotovic and H.K. Khalil, "Singular Perturbations in Systems and Control", IEEE Press, New York, 1986.

[85] A.A. Sadjadi and K. Khorasani, "Block Implementation of Two Time-Scale Discreate Systems and Its Relation to Multirate Composite Control", Proc. American Control Conf., ACC87, 1987.

[86] R.G. Philips, "Reduced Order Modeling and Control of Two Time-Scale Discreate Systems", Int. J. Contr., Vol. 31, 1980.

[87] M.S. Mahmoud, "Order Reduction and Control of Discrete Systems", Proc. IEEE, Vol. 29, No. 4, 1982.

[88] P.V. Kokotovic, J.J. Allemong, J.R. Winkelman and J.H. Chow, "Singular Perturbation and Iterative Separation of Time Scales", Automatica, Vol. 16, 1980.

[89] J.H. Chow and P.V. Kokotovic, "Eigenvalues Placement in Two Time-Scale Systems," Proc. IFAC Symp. LSS, 1976.

[90] P.V. Kokotovic, R.E. O'Malley, JR. and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory-An Overview", Automatica, Vol. 12, 1976.

[91] A.K. Bejczy, "Robot Arm Dynamics and Control", Technical Memorandum 33-669, Jet Probulsion Laboratory, 1974.

[92] Golla, Garg and Hughes, "Linear State f/b Control of Manipulators", Mechanical Machine Theory, Vol. 16, 1981.

[93] W.E. Snyder and W.A. Gruver, "Microprocessor Implementation of Optimal Control for Robotic Manipulator System", Proc. 18th IEEE Conf. Decision Control, Vol. 2, 1979.

[94] J.P. La Salle, "Stability Theory for Ordinary Differential Equations", J. Diff. Equations, Vol. 4, 1968.

[95] J. La Salle and S. Lefschetz, "Stability by Lyapunov's Direct Method with Applications", Academic Press, New York, 1961.

[96] J.S. Reed and P.A. Ioannou, "Instability Analysis and Robust Adaptive Control of Robotic Manipulators", Proc. of 27th Conf. on Decision and Control, Austin, Texas, 1988.