NAG-1-623

LANGLEY

IN-37-CR

ON MOTION PLANNING AND CONTROL OF MULTI-LINK LIGHTWEIGHT ROBOTIC MANIPULATORS

161074

P-273

A DISSERTATION

presented to

The Faculty of Division of Graduate Studies

By

Sabri Cetinkunt

(NASA-CR-181226) ON MOTION PLANNING AND CONTROL OF MULTI-LINK LIGHTWEIGHT ROBOTIC MANIPULATORS (Georgia Inst. of Tech.) 273 p CSCL 13I

N89-11234

Unclas

G3/37 0161074

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Mechanical Engineering

Georgia Institute of Technology

November, 1987

ON MOTION PLANNING AND CONTROL OF MULTI-LINK LIGHTWEIGHT ROBOTIC MANIPULATORS

Approved:

Dr. Wayne J. Book, Chairman

Dr. Stephen L. Dickerson

Dr. Benson H. Tongue

Dr. Abraham H. Haddad

Dr. Antony Calise

Date Approved by the Chairman: Nov 16, 1987

DEDICATION

To

my sister, Safiye, whose help and encouragement in the very early years of my education has been invaluable.

TABLE OF CONTENTS

Acknowledgment iii

List of Tables vii

List of Figures viii

Summary xvii

CHAPTER 1 Introduction 1

1.1 Objective of the Research 1
1.2 Subject Area and General Introduction 1
1.3 The Problem Statement 5
1.4 Previous Work 8
1.5 Contributions of This Work 17
1.6 Organization of the Thesis 19

CHAPTER 2 Symbolic Modeling of Flexible Manipulators 21

2.1 Introduction 21
2.2 Lagrangian-Assumed Modes Method 26
2.3 Symbolic Implementation of the Lagrangian-Assumed Modes Method . 30
2.4 Applications and Discussion of Simulation Results 38
2.5 Conclusion 42

CHAPTER 3 Open and Closed Loop Dynamics: Linear Analysis . 51

3.1 Introduction: 51
3.2 Open Loop Eigenstructure Analysis: 52
3.2.1 Linearization of the Nonlinear Model 52
3.2.2 Open Loop Eigenvalues Root Locus 54
3.3 Closed Loop Dynamics Under Joint Variable Feedback:
Limitations and Sensitivity 65
3.3.1 Previous Work and Results Based on Frequency Domain Approach 65
3.3.2 Results of Assumed Modes Based Models 70
3.4 Summary of Results and Conclusions 72

ACKNOWLEDGMENTS

I wish to express my appreciation to my advisor, Dr. Wayne J. Book, for his invaluable guidance, patience, and encouragement over the course of this work. It has been a privilege to work with him. I am also grateful to the members of the thesis committee, Dr. A.J. Calise, Dr. S.L. Dickerson, Dr. A.H. Haddad, and Dr. B.H. Tongue, for their suggestions and input through the classes I have taken over the years.

Thanks to many friends at Georgia Tech, especially visiting scholars Dr. Y-C. Chung, Dr. B. Siciliano, former graduate students, Dr. T. Alberts, Dr. G. Hastings, G. Letchworth, and current graduate students, B. Yuan, P. Fitzsimons, G. Flowers, K. Gu, J. Huggins, D. Kwon, J. Lee.

My special thanks go to my brother Sakin, and my wife Mary Ellen for their moral support.

This work was supported in part by the National Aeronautics and Space Administration under grant NAG-1-623, and the National Science Foundation under grant MEA-8303539.

CHAPTER 4 Gross Motion Control 91

4.1 Introduction 91
4.2 Non-adaptive Control Algorithms 94
4.2.1 Computed Torque Method (CTM) 94
4.2.2 Decoupled Joint Control (DJC) 99
4.2.3 Model Following Form of CTM and DJC 101
4.3 Adaptive Model Following Control (AMFC) 104
4.3.1 Preliminaries: Linear Perfect Model Following Control (LPMFC) 104
4.3.2 Adaptive Model Following Control - Hyperstability Based Design 106
4.3.3 Adaptive Model Following Control Revisited: Generalized Inertia Matrix Based AMFC 114
4.3.4 Generalized Inertia Matrix BAsed AMFC: Application to Flexible Manipulators 116
4.4 Comparative Simulations and Discussion of Results 121
4.4.1 Simulations with Rigid Manipulator Model 121
4.4.2 Comparative Control Algorithm Simulations on Rigid and Flexible Manipulator Model 138
4.4.3 Gross Motion Control Simulations of Flexible Manipulator Model 145
4.5 Summary 152

CHAPTER 5 Combined Motion Control 166

5.1 Introduction 166
5.2 Fine Motion Control: Optimal LQR 168
5.2.1 Preliminaries: General Variation of a Functional 168
5.2.2 LQR with Prescribed Degree of Stability 173
5.2.3 Model Following LQR with Prescribed Degree of Stability . . 175
5.3 Combined Control Simulations and Discussion of Results 180
5.4 Summary of Results and Conclusion 213

CHAPTER 6 Conclusions and Recommendations for Future Work 214

6.1 Conclusions 214
6.2 Recommendations for Future Work 218

APPENDIX A Flexible Manipulator Arm Parameters 220

APPENDIX B On the Solution of $Ax = b$ 222

APPENDIX C Stability 224

C.1 Lyapunov Stability 224

C.2 Hyperstability and Positivity Concepts 227

APPENDIX D Linear Analysis Results 231

D.1 Linear Dynamic Model of the Flexible Manipulator 236

D.2 LQR Feedback Gains and the Resultant Closed Loop Eigenvalues . 236

Bibliography 246

Biography 255

LIST OF TABLES

Table	Title	Page
3.1	Asymptotic eigenvalues of assumed modes model under high position feedback gain.	60
4.1	Closed loop system and desired motion bandwidth.	122

LIST OF FIGURES

Figure	Title	Page
1.1	Examples of industrial robots . . . . . a) Cincinnati Milacron T3, b) Unimation PUMA 600.	2
1.2	Block diagram of manipulator control system	2
1.3.a	Remote orbital servicing system	3
1.3.b	Robot-aided structural assembly	3
1.4.a	Trajectory transitions	10
1.4.b	Planned motion with smooth transitions	10
1.4.c	Straight line motion in the task space	10
1.4.d	Finding minimum time trajectories	12
1.4.e	Path in task space	12
1.5	Basic structure of an adaptive model folowing control (AMFC) system	16
2.1	One link flexible arm	22
2.2	A flexible serial manipulator	23
2.3	(4x4) Homogeneous coordinate transformations	24
2.4	Two link flexible arm example	39
2.5	Two link rigid model joint angles	44
2.6	Two link flexible model joint angles, clamped-free mode shapes . . . . . a) $EI_i = 10.\ Nt - m2$, b)$EI_i = 100.\ Nt - m^2.\ i = 1, 2.$	45
2.7	Two link flexible model joint angles, clamped-clamped mode shapes for link 1	47

a) $EI_i = 10.\ Nt - m^2$, b)$EI_i = 100.\ Nt - m^2.$, $i = 1, 2$.

2.8 Comparisions of flexible vibration coordinate responses 49
(Clamped-free mode shapes)
a)$EI_i = 10.\ Nt - m2.$, b) $EI_i = 100.\ Nt - m2,\ i = 1, 2.$

3.1 Eigenvalue analysis cases - root locus parameters are indicated . . . 55
on the figures

3.2 Open-loop eigenvalues (pin-pin beam is the limiting case) 56

3.3 Single clamped-free beam limit behavior 59

3.4.a Open-loop eigenvalues as function of configuration 61
(θ_2) and stiffness, ($EI_1 = EI_2$)

3.4.b Closer look at the Fig. 3.4.a 62

3.4.c Closer look at the Fig. 3.4.b 63

3.5 Illustration of limitations imposed on the closed loop 67
servo control system performance due to arm flexibility

3.6. Limitations of servo control due to flexibility for two- 68
beam,one-joint controlled case.

3.7 Analogy: Lumped parameter model exhibiting the same root locus . 69
of eigenvalues as function of servo gains.

3.8 Closed-loop eigenvalue root locus as function of servo gains - 71
Assumed mode model results.

3.9 Root locus of closed loop eigenvalues - low servo stiffness case . . . 73

3.10 Root locus of closed loop eigenvalues - medium servo stiffness case . 79

3.11 Root locus of closed loop eigenvalues - high servo stiffness case . . . 85

4.1 Computed torque control: a) standard form, ... 96
b) model following form, c) model following form

4.2 Desired motion generation 98

4.3 Decoupled joint control: a) standard form . . . . . .
b) model following form.

4.4 Basic elements of an adaptive model following control system . . . 102

4.5 Reference model used as trajectory generator. 102

4.6 Model following control: a) Linear perfect model following 107
control (LPMFC), b) Adaptive model following control,
c) Generalized inertia matrix based AMFC.

4.7 The Hyperstability problem 112

4.8 Feedback gain differences that must be taken care of by 115
adaptation algorithm.

4.9.a-b Desired trajectories: a) joint 1, b) joint 2 125
4.9.c-d CTM resultant torque history c) jont 1 d) joint 2. 126
4.9.e-f Relative importance of nonlinear forces and gravitational 127
forces along the motion. e) joint 1, f) joint 2.

4.10 Decoupled joint control performance on the rigid model- 128
Illustration of the effect of high gain feedback:
a) joint 1 response, b) joint 2 response,
c) joint 1 speed, d) joint 2 speed.

4.11 Robustness of CTM with respect to payload variations, and 130
the effect of high gain feedback: a) joint 1 angle,
b) joint 2 angle c) joint 1 speed, d) joint 2 speed

4.12 Robustness of DJC with respect to payload variations, and 132
the effect of high gain feedback: a) joint 1 angle,
b) joint 2 angle c) joint 1 speed, d) joint 2 speed.

4.13 Robustness of CTM and DJC with respect to payload variations . . 134
for relatively fast motions: a) joint 1 angle,
b) joint 2 angle, c) joint 1 speed, d) joint 2 speed.

4.14 Torque histories corresponding to motions indicated 136
in figure 4.11 through 4.12.

4.15 Flexible model repsonse, controller: CTM, desired motion: 141
Cases (a),(b), (c). a) joint 1 response for cases (a) and (b),
b) joint 2 response for cases (a) and (b),
c) joint 1 response for case (c), d) joint 2 response for case (c).

4.16 Flexible mode shape magnitude responses along the motions 143
(a), (b), and (c). a) $w_{ni} = 1/4w_{cc1}$ b) $w_{ni} = 1/2w_{cc1}$ c) $w_{ni} \cong w_{cc1}$

4.17 Flexible model response, controller: DJC, $w_{ni} = 5.5rad/sec$ 148
a) joint 1 response, desired motion: $w_{mi} = 2.75rad/sec$
b) joint 2 response, desired motion: $w_{mi} = 2.75rad/sec$
c) joint 1 response, desired motion: $w_{mi} = 5.5rad/sec$
d) joint 2 response, desired motion: $w_{mi} = 5.5rad/sec$

4.18 Flexible mode shape magnitude responses 152
a) along the motion shown in figures 4.17.a-b.
b) along the motion shown in figures 4.17.c-d.

4.19 Flexible model response, controller: AMFC 153
a) joint 1 response, desired motion: $w_{mi} = 2.75rad/sec$
b) joint 2 response, desired motion: $w_{mi} = 2.75rad/sec$
c) joint 1 response, desired motion: $w_{mi} = 5.5rad/sec$
d) joint 2 response, desired motion: $w_{mi} = 5.5rad/sec$

4.20 Flexible mode shape magnitude responses 152
a) along the motion shown in figures 4.19.a-b.
b) along the motion shown in figures 4.19.c-d.

4.21 Robustness of CTM and DJC with respect to payload variations . . 153
(Flexible model responses), controller: $w_{ni} = 5.5rad/sec$,
desired motion: $w_{mi} = 2.75rad/sec$.
a) joint 1 response, b) joint 2 response
c) flexible mode shape magnitude response under CTM.
d) flexible mode shape magnitude response under DJC.

4.22 Robustness of CTM and DJC with respect to payload variations . . 155
(Flexible model responses), controller: $w_{ni} = 5.5rad/sec$,
desired motion: $w_{mi} = 5.5rad/sec$.
a) joint 1 response, b) joint 2 response
c) flexible mode shape magnitude response under CTM.
d) flexible mode shape magnitude response under DJC.

4.23 Robustness of AMFC with respect to payload variations 157
a) joint 1 response, b) joint 2 response
c) flexible mode shape magnitude response along motion (a)
d) flexible mode shape magnitude response along motion (b)

5.1 Desired motion in task space 167

5.2 General variation of a function 169

5.3 Combined control (AMFC followed by model following LQR) simulation under perfect conditions 182
a) joint 1 response, b) joint 2 response, c) joint 1 speed,
d) joint 2 speed, e) torque history of joint 1,
f) torque history of joint 2.

5.4 Combined control (AMFC followed by standard LQR) simulation . 185
under perfect conditions.
a) joint 1 response, b) joint 2 response.

5.5 Flexible mode shape magnitude response associated with motions . 186
shown in figure 5.3

5.6 Flexible mode shape magnitude response associated with motions . 189
shown in figure 5.4

5.7 Desired joint trajectories generated by the reference models 191
of gross motion and fine motion phases
a) Joint 1 , b) Joint 2.

5.8 Robustness of combined control with respect to payload variations . 194
a) joint 1 response, b) joint 2 response

5.9 Robustness of combined control with respect to switching 195
time speed of manipulator.
a) joint 1 response, b) joint 2 response.

5.10 Robustness of combined control with respect to payload 196
variations in relatively slow motions
($mp_{actual} = mp_{lqr} = 2.0kg, mp_{amfc} = 0.0kg$).
a) joint 1 response, b) joint 2 response.

5.11 Robustness of combined control with respect to payload 197

variations in relatively slow motions
($mp_{actual} = mp_{amfc} = 0.0, mp_{lqr} = 2.0kg$).
a) joint 1 response, b) joint 2 response.

5.12 Flexible mode shape magnitude response along the motion 198
described by figure 5.8.
a) Standard LQR implementation results.
b) Model following LQR implementation results.

5.13 Flexible mode shape magnitude response along the motion 200
described by figure 5.9.
a) Standard LQR implementation results.
b) Model following LQR implementation results.

5.14 Flexible mode shape magnitude response along the motion 202
described by 5.10.
a) Standard LQR implementation results.
b) Model following LQR implementation results.

5.15 Combined control response - fine motion phase is controlled 206
by a linear partial state feedback controller.
a) joint 1 angle response.
b) joint 2 angle response.
c) joint 1 speed response.
d) joint 2 speed response.
e) flexible mode shape magnitude responses along the motion described by figure 5.15.a-d.

5.16 Combined control response - with realistic joint 209
inertial values.
a) joint 1 angle response.
b) joint 2 angle response.
c) joint 1 speed response.
d) joint 2 speed response.
e) joint 1 torque history.
f) joint 2 torque history.
g) flexible mode shape magnitude responses along the motion described by figure 5.16.a-f.

A.1 The flexible arm paramters and geometric dimensions 221

C.1 The different definitions of stability. 225

C.2 The standard Hyperstability problem. 230

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

All symbols used in this thesis are defined in the context used, and in accordance with the standard use in the litarature. Some of the symbols are defined below for convenience of the reader.

w_{mi} - Undamped natural frequency of the reference model.

w_{ni} - Undamped natural frequency of the closed loop system.

w_{cc1} - Lowest natural frequency of the flexible arm when all joints are locked and links are extended.

x_m - Reference model state.

x_p, x - Plant (Manipulator) state.

ξ_i - Damping ratio of mode i.

x_f - Final desired state, i.e. θ_f, δ_f.

A - crosssectional area of the beam.

ρA - mass per unit length.

E - Young's modulus of the material used.

I - Area moment of inertia of the beam about the bending axis.

EI - Stiffness of the beam.

m_i - mass of link i.

l_i - length of link i.

K_1, K_2 - Linear feedback control gains.

K_{pn}, K_{un} - Nominal gains in adaptive control.

$\Delta K_p, \Delta K_u$ - Adaptated gains of adaptive control.

g - gravity acceleration.

K - structural stiffness matrix of the manipulator.

mj_i, jj_i - mass and mass moment of inertia of joint i.

m_p, j_p - mass and mass moment of inertia of payload.

θ_i -i^{th} joint variable (joint angle i).

δ_{ij} - j^{th} flexible mode shape generalized coordinate of link i.

FBB - Feedback Block.

FFB - Feedforward Block.

LTI - Linear Time Invariant.

NLTV - Nonlinear Time Varying.

SPR - Strictly Positive Real.

PR - Positive Real.

PII - Popov Integral Inequality.

CTM - Computed torque Method.

DJC - Decoupled Joint Control.

AMFC - Adaptive Model Following Control.

LPMFC - Linear Perfect Model Following Control.

GIM - Generalized Inertia Matrix.

Summary

A general gross and fine motion planning and control strategy is needed for lightweight robotic manipulator applications such as painting, welding, material handling, surface finishing, and space craft servicing.

The control problem of lightweight manipulators is to perform fast, accurate, and robust motions despite the payload variations, structural flexibility, and other environmental disturbances.

Performance of rigid manipulator model based computed torque and decoupled joint control methods are determined and simulated for the counterpart flexible manipulators. A counterpart flexible manipulator is defined as a manipulator which has structural flexibility, in addition to having the same inertial, geometric, and actuation properties of a given rigid manipulator. An adaptive model following control (AMFC) algorithm is developed to improve the performance in speed, accuracy and robustness. It is found that the AMFC improves the speed performance by a factor of two over the conventional non-adaptive control methods for given accuracy requirements while proving to be more robust with respect to payload variations. Yet there are clear limitations on the performance of AMFC alone as well, which are imposed by the arm flexibility. In the search to further improve the speed performance while providing a desired accuracy and robustness, a combined control strategy is developed. Futhermore, the problem of switching from one control structure to another during the motion and implementation aspects of combined control are discussed.

CHAPTER I

1.1. Objective of the Research

The amount of literature in dynamics and control aspects of rigid robotic manipulators is large. Much less literature on research in lightweight manipulators is available and it is only on the fine motion aspect. A typical robotic application involves both gross and fine motion phases. Systematic motion planning and control methods for realistic applications of lightweight manipulators are yet to be developed. The objective of this work is to develop a general motion planning and control method for lightweight robotic manipulator applications involving a gross motion and a fine motion. Thus, a realistic base for the utilization of lightweight manipulators in industrial and space applications will be established. In the search for a control system which will keep the advantages of lightweight arms, the performance of traditional control methods will be determined when they are applied to lightweight manipulators.

1.2. Subject Area and General Introduction

Industrial robotic manipulators are mechanisms controlled by computers (Fig.1.1). The control problem of a robotic manipulator may be divided into two parts: 1. trajectory planning, which is usually done off-line, and 2. trajectory tracking which requires on-line computations (Fig.1.2). At the trajectory planning level the manipulator task is defined and, given the environmental and system

ORIGINAL PAGE IS
OF POOR QUALITY

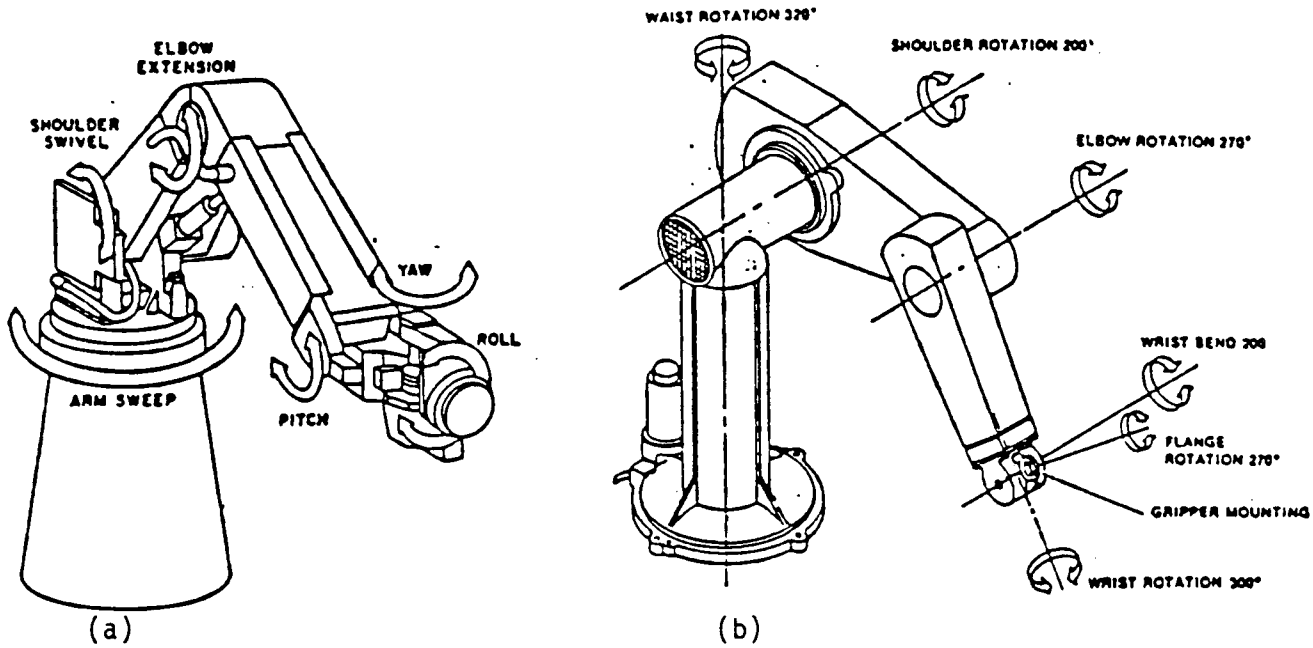


Fig.1.1 Examples of industrial robots
a) Cincinnati Milacron T3, b) Unimation PUMA 600.

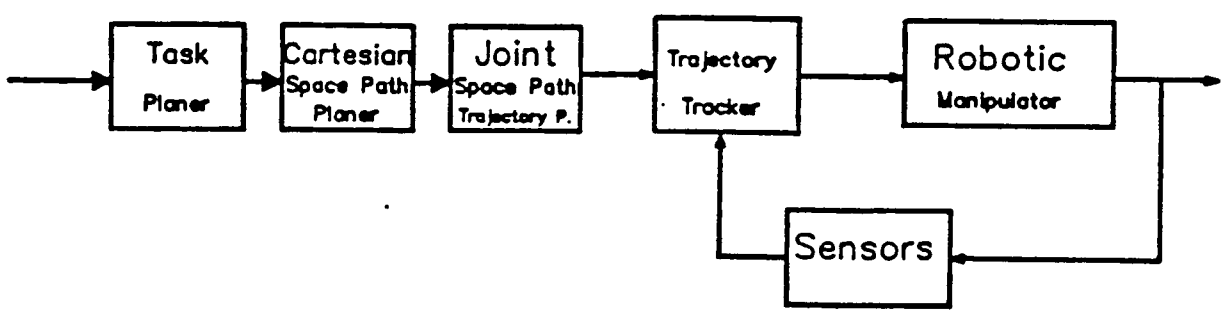


Fig.1.2 Block diagram of manipulator control system

constraints, a motion is planned off-line based on some criteria. At the tracking level, the desired trajectory command is issued the controller, and the control vector is computed based on the control law in an attempt to follow the desired trajectory planned previously.

Assuming that, at best, the controller is capable of perfectly following the desired trajectory, the best performance of the manipulator will be the planned trajectory. Thus the trajectory planning level is the one which essentially determines the upper bound of the performance. All performance requirements and system constraints must be imposed on the planned trajectory. A controller is then designed with the intent to follow that trajectory as closely as possible.

Higher productivity requirements demand manipulators that move faster and more precisely. The trajectory planning methods should utilize the system capabilities as much as possible, rather than resting on very conservative, simple planning methods. The more fundamental factors which limit the manipulator productivity are the maximum velocities and accelerations affordable by the system. These are the physical constraints of the system independent of the planning and control method. The velocity and acceleration constraints are functions of the mechanical properties of the system , such as link inertial parameters, payload, friction and the actuator capabilities. In order to increase the productivity of a robot, one may consider changing these parameters so that higher velocities and accelerations can be afforded. Payload and friction are the parameters determined by the nature of the task and the actuator types.

One option is to increase the actuator capabilities. However, in a typical

industrial robot, the actuators are located at the link joints and must be carried by the previous actuators. Therefore, increasing the actuator sizes in order to increase the system capabilities is not an ultimate answer, has a limit, and can be self-defeating. The major factor that limits the affordable speed of operations is the inertia of the manipulator. Thus the fundamental question is the following: *can the inertial parameters be reduced by the use of lightweight links, leading to a lightweight structure and making higher speed operation possible?* Reducing the link inertias is one of the most effective way of improving the manipulator speeds, which results in more productive systems.

Reducing the weight of a manipulator system makes it possible to obtain faster motions. Increased mobility, large work space and reach capabilities and lower energy consumption are additional advantages of lightweight manipulators. Unfortunately, a disadvantage is the occurrence of structural vibrations due to the lightweight nature of the manipulator. For accuracy, the structural vibrations must be kept under control. A control system must deal with the control of structural vibrations as well as joint trajectory tracking. Currently there is no convenient way of directly measuring the flexible vibration modes. They must be estimated from strain-gage or camera output signals based on some linear mathemetical model approximations. This practical problem is a major implementation problem of control algorithms for lightweight manipulators. Control algorithms which will not require the feedback information of flexible modes should be explored in order to avoid this implementation problem.

In many cases, a reasonable lightweight robotic manipulator motion, going from one position to another, would involve a gross motion followed by a fine motion.

The gross motion should be performed fast. Towards the end of the motion, a fine motion is performed. Many applications require the robot end effector to contact an object. The planning and execution of the docking motion, which involves coming into contact with an object, is an interesting and important problem to be solved. A simple example would be a spacecraft service task (Figs. 3 and 4) where the manipulator moves from its initial position to a distant object then contacts it in a controlled way, and finally works on the object. Fine motion does not neccessarly mean a slow motion (low bandwidth closed loop system). Consider a manipulator in a space craft service job. The task is to insert a peg into a hole on the object of manipulation, but the structure (on which the hole is located) vibrates with an unknown frequency. In order for the manipulator to reliably perform this task, the closed loop control system bandwidth should be considerably higher than the expected range of vibrations of the task structure.

Current motion planning and control methods of robotic manipulators cannot be directly applied to lightweight, high performance manipulators where structural flexibilities are significant. New motion planning and control methods, which take the structural flexibilities into account, are needed for lightweight manipulators and are discussed in the rest of this thesis.

1.3. The Problem Statement

A general task of a multi-link flexible robotic manipulator would consist of three phases.

Phase 1: A gross motion, typically fast for productivity, from a known initial state towards a final desired state close to an object.

ORIGINAL PAGE IS
OF POOR QUALITY

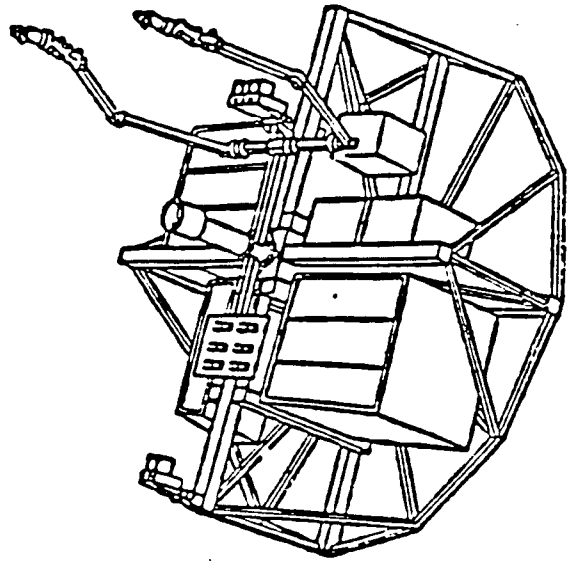


Fig.1.3.a Remote orbital servicing system

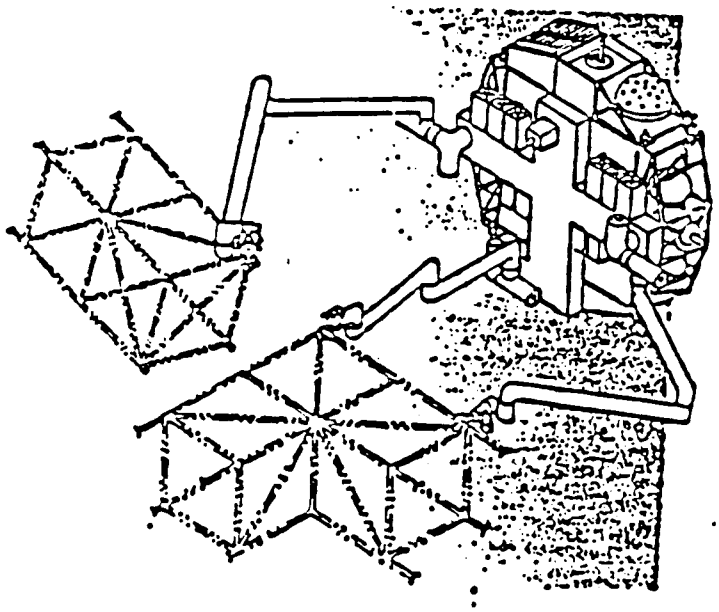


Fig.1.3.b Robot-aided structural assembly

Phase 2: Switching from gross motion to fine motion near the object and execute the fine motion.

Phase 3: Finally, contact or interact with the object.

This thesis will deal with phases 1 and 2. Phase 3 of the problem requires the monitoring of the contact forces. Position plus force feedback control has to be employed for the remaining part of the task.

The motion required by the task can be characterized in more detail as follows. At phase one, the arm is away from the object, the motion is large and has to be accomplished quickly so that the task can be performed productively. The flexible deflections and vibrations at this stage are not that important, but rather one would be satisfied with following a desired trajectory in joint space, with no explicit control action for vibration stabilization. However, the desired trajectory may be designed in such a way that if there were a perfect tracking controller, resultant vibrations would be acceptable. In phase 2, the end of the arm is close to the object and should not collide in an undesirable way. Thus, the control of flexible vibrations is important as well as accurate positioning of the joint variables. The motion may be rather slow, if necessary near the desired contact point with the object.

For a task described by phase 1 and phase 2, one needs to plan trajectories for each phase in either joint or task space as a function of time, then design controllers appropriate for each phase. Notice that every phase has a planning and control level, although in some cases the planning and control problem may be solved simultaneously. In the rigid arm case the control problem is to drive the joint

variables to follow the planned trajectories, where the number of control signals is equal to the number of controlled generalized coordinates. When the structural flexibility is significant, two control problems exist: 1. joint space control, and 2. suppression of flexible vibrations. It is the phase 2 of the motion where control problem 2 is important.

Among the goals of this work are the following:

1. Determine the best performance possible from rigid model based control methods and the limitations of these methods when applied to flexible manipulators.

2. Develop new high-speed, high-precision, robust control algorithms for light weight manipulators. Along that line, AMFC techniques, as well as a combination of different control methods, will be studied.

1.4. Previous Work

Dynamics of industrial robots are governed by second order, coupled, highly nonlinear differential equations [A9]. When the structural flexibilities are considered, the complexity of the dynamics increases. Nonetheless, after some modal truncations, the flexible system dynamics is still governed by similiar types of equations [A11]. However an important difference is that when the structural flexibility is included in the dynamic analysis, the number of inputs becomes less than the number of generalized coordinates controlled. The motion planning and control problem is a difficult task due to: 1. nonlinearity, 2. strict constraints imposed on the system, i.e. actuator saturation, and collision avoidance problems and 3. high

system order.

Because of these difficulties, earlier work took a very conservative approach toward solving the problem. For example, a desired trajectory, either in joint or task space, is planned as a collection of constant velocity profiles. The transition from one constant velocity segment to another is determined by continuity requirements. Maximum allowable acceleration bounds were imposed based on the worst possible cases [B1, B2, C8, Fig.1.4.a]. The corner points of the constant velocity segments are never exactly reached unless an overshoot is allowed (Fig. 1.4.b). Apparently such a planning scheme rarely and only instantaneously uses the full manipulator capabilities, and does not consider the manipulator dynamics, resulting in low performance and productivity. Taylor (1979) developed a method to execute straight line paths in task space [C10]. The method determines the number of intermediate points necessary so that the deviations from the path due to linear interpolations are bounded by a pre-assigned value (Fig.1.4.c). Another method was developed by Lin et al (1983) to find minimum time trajectories in joint space by means of cubic splines [C7]. A desired task is defined as a sequence of N points in the cartesian coordinates. The corresponding joint variables are found via the solution of the inverse kinematic problem. These N points in joint space are then connected to each other with cubic splines that minimize the total travel time with no constraint violations. These trajectory planning methods are developed for rigid robotic manipulators and do not consider structural flexibilities.

Bobrow et al, (also Shin and McKay) have incorporated the full nonlinear dynamics of the manipulator to the minimum time trajectory planning level, where the cartesian coordinate path and actuator constraints are given (C1, C2,

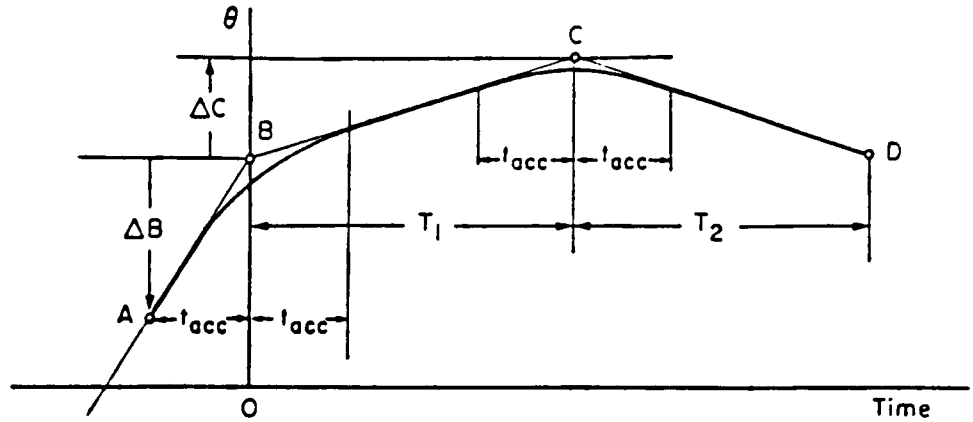


Fig.1.4.a Trajectory transitions

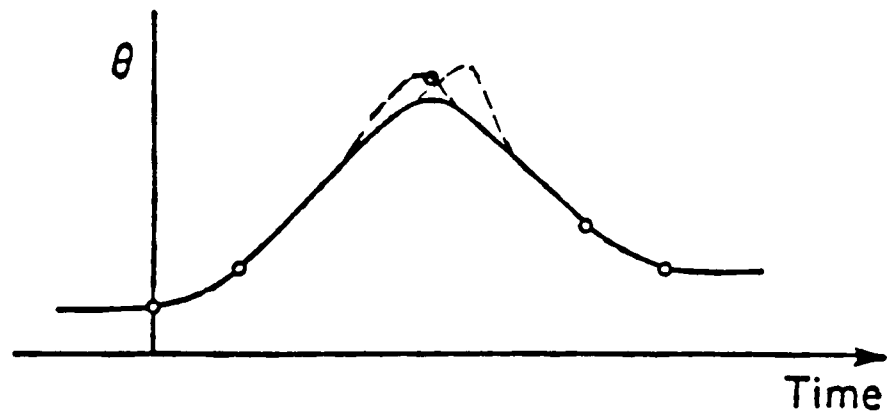


Fig.1.4.b Planned motion with smooth transitions

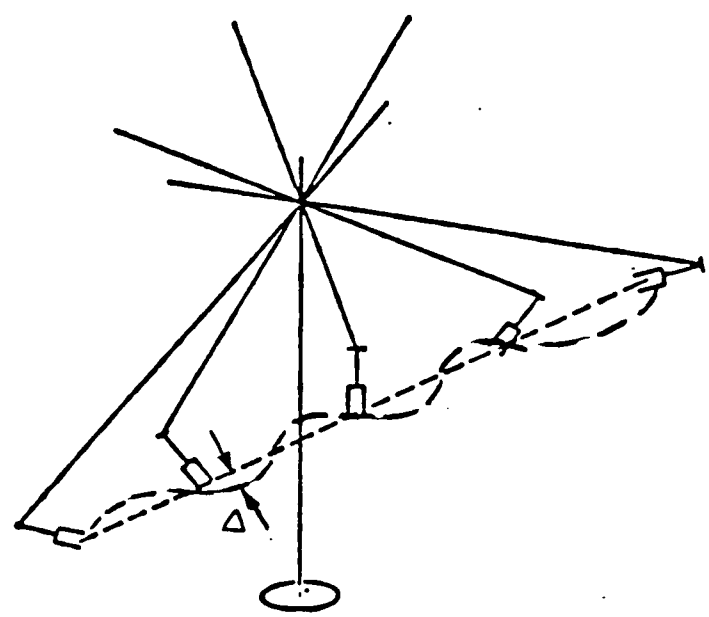


Fig.1.4.c Straight line motion in the task space

C3, C4, Fig 1.4.d. and 1.4.e). The method essentially reduces a set of n coupled second order nonlinear differential equations to a single second order nonlinear differential equation in path coordinate system, and uses direct numerical integration to a find minimum time trajectory in the task space. Notice that if a manipulator motion is constrained to follow a predefined path, then effective degree of freedom of the manipulator is only one. That is how the n-set second order equations in joint space can be reduced to a single second order equation in path space.

The second step in the manipulator control system design is to find an appropriate control law which will realize the planned motion. This is the lowest level in the control system hierarchy [E8]. Today the majority of industrial robots are used as positioning devices. If the robot end effector is to move from one position to another and the path followed is not important, each joint can be moved sequentially while the others are all locked. In this case each joint can be controlled by a simple position servo, since every joint control problem is a second order linear system, with a gravity load offset. Although such a motion makes the control problem easy, it is very inefficient. When all joints are allowed to move simultaneously, the performance of the simple position controllers drastically deteriorates due to the inertial coupling, gravitational torque variations, friction, centrifugal and coriolis torque effects.

Conventional controllers cancel some of these coupling effects via feedforward compensation. The inertial coupling and gravitational torques are the major disturbances and can be canceled based on the dynamic model of the manipulator. The friction effect is a nondeterministic phenomenon and compensation is made

based on some experimental average values.

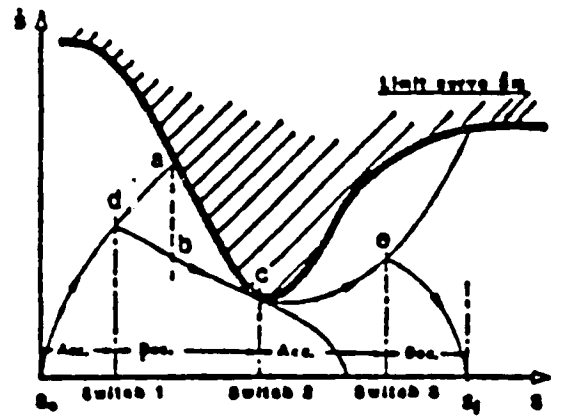


Fig.1.4.d Finding minimum time trajectories

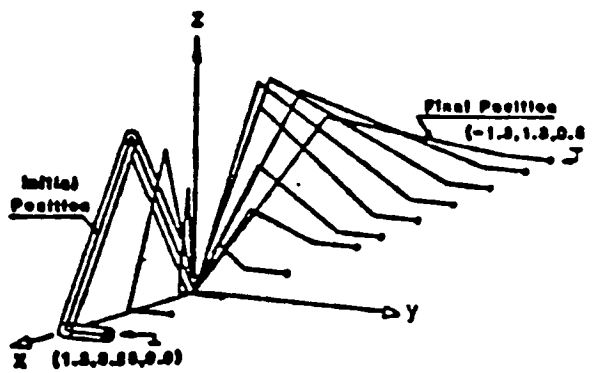


Fig.1.4.e Path in task space

The centrifugal and coriolis effects become important at high speed and are approximately compensated at each joint based on the dynamic model of the manipulator. Notice that the whole purpose of the feedback compensation of nonlinear centrifugal and coriolis, and gravity effects is to reduce the system back to a simple second order linear form so that linear feedback controllers can be used. However, almost all of the feedforward compensation is based on the manipulator dynamic model or its simplified forms. This so called *inverse problem* or *computed torque method* relies heavily on an accurate prior knowledge of the dynamic model, system parameters and their variation, and all other external disturbances. The more accurate the prior knowledge of the system dynamics and paramters is, the more successful the computed torque method will be.

In robotic applications parameters can be in the range of 50-200 % of average values. External disturbances and the nature of the friction are never accurately known in advance. The payload may drastically vary from one task to another without advance knowledge. Moreover, the dynamic characteristics of the system may change in time. Clearly, computed torque[E1,E2] methods are not so suitable for applications where external disturbances, large unknown payload variations and uncertainties exist. It is important to note that the *resolved rate* and *resolved acceleration* methods are also computed torque based methods[B4,B5]. The difference is that they generate reference trajectories in joint variables which are resolved from a desired task space trajectory.

It is very desirable to have a control system which has the following properties:

1. good tracking accuracy (transient and steady state)

2. fast adaptation, if necessary, due to

   a) the variations in the system parameters (insensitive to parameter variations)

   b) disturbances (disturbance rejection)

3. does not require precise knowledge of the model parameters,

4. is stable in the large (Global Asymptotic Stability).

The design method for the control algorithm should not require a precise knowledge of manipulator dynamics and parameters and should guarantee a stable resultant control system. Furthermore, finding the appropriate parameters of design which will yield good tracking and robustness should be relatively easy.

These requirements call for adaptive control methods. Adaptive control methods may be divided into three major categories:
1. gain scheduling, 2. self tuning regulators, and 3. model reference adaptive controllers (gradient methods, Lyapunov and Hyperstable design). Gain scheduling and self tuning regulators are direct generalizations of linear control laws, and will not be discussed here due to their serious drawbacks. For example, gain scheduling methods require storage of the control law parameters and use the appropriate parameters as the operating range changes. There are two major drawbacks. First is the problem of switching from one gain to another (how does it affect the system performance and stability?). Second and more importantly, if the system dimension and possible range of operating conditions are large, the storage requirements may become prohibitive. Self-tuning regulators are considered to be inappropriate due

to the *persistent excitation requirements*, which is a severe requirement in robotics. The MRAS (Model Reference Adaptive Systems) are attractive since they do not have the above drawbacks and globally asymptotically stable designs are possible. The difference between the methods in this category originates in the way the adaptation mechanism is designed (Fig.1.5). In recent years a tremendous amount of research has been conducted and results published on the model following adaptive control methods. An early work by Dubowsky (1979) showed the promise offered by MRAS in robotics [D8]. However this work suffered from the lack of a global stability proof. Balastrino et al (1983) developed a globally stable adaptive model following control method based on the hyperstability approach [D3]. Horowitz and Tomizuka [D19] proposed a control algorithm which has two parts. One part compansates for the inertial and nonlinear centrifugual and coriolis terms adaptively, the other part is a linear position and velocity feedback control. The adaptation algorithm for the adjustment of inertial and nonlinear terms is based on the hyperstability. Craig et.al. developed a similiar method based on the Lyapunov approach [D12]. Unfortunately, none of these techniques can explicitly specify the transient response in the design process. Lim and Eslami introduced an auxilary input signal to speed the convergence of the adaptation algorithm [D20].

When a comparison is made between Lyapunov and Hyperstability based adaptation law design methods, it is seen that theoretically they offer the same solutions for systems having bounded, piecewise continuous input signals [D15]. However, finding alternative Lyapunov functions is known to be very difficult and is usually done by trial and error, whereas Hyperstability and Positivity based methods offer a wider class of admissable control laws which

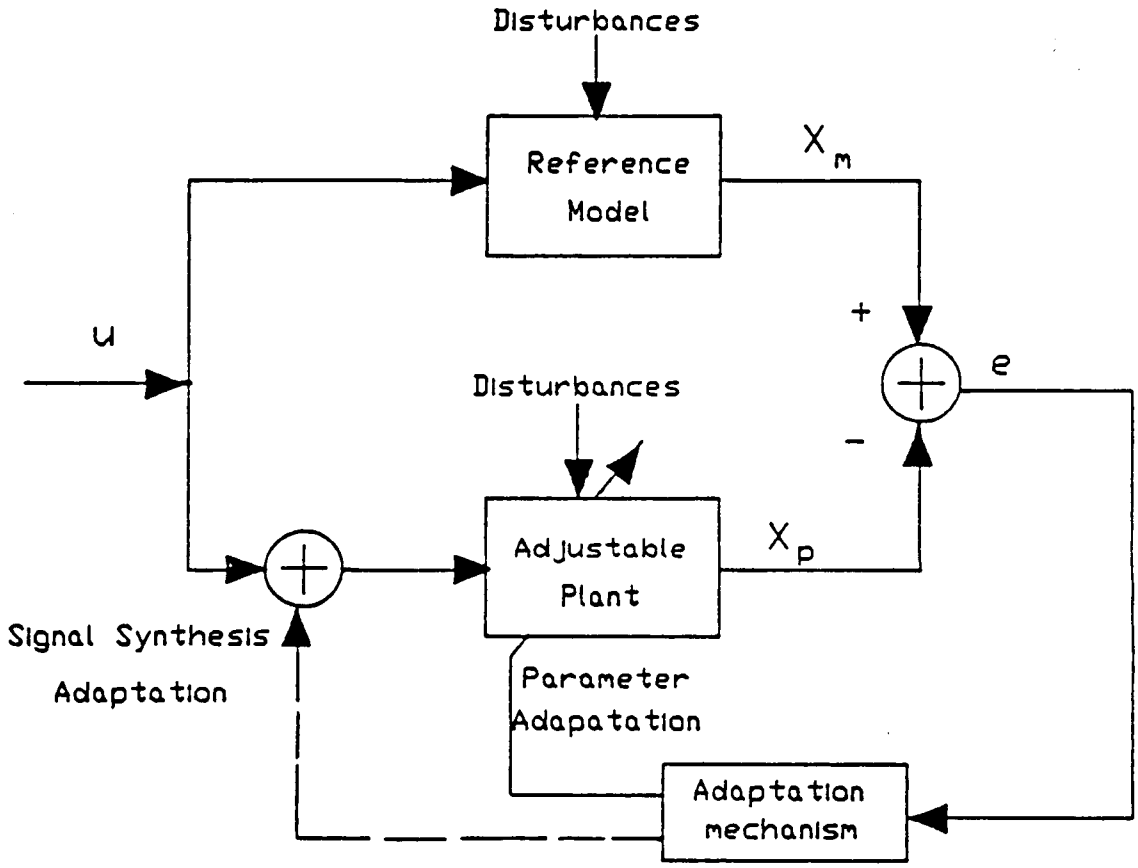


Fig.1.5 Basic structure of an adaptive model folowing control (AMFC) system

guarantee the global asymptotic stability of the system [D1]. Besides that, the reference model and the commanded reference input serve very efficiently as the on-line trajectory planning method with no complications, and result in very little computational burden for trajectory planning. Furthermore, powerful on-line control computers are not required, which reduces the cost of the control system.

It is important to note that all of the previous trajectory planning and controller design methods are for rigid manipulators. An important contribution of this thesis will be to devise a methodology which allows the application of these methods to flexible robotic manipulators and to determine their performance limitations.

1.5. Contributions of This Work

1. A new symbolic modeling method for lightweight robotic manipulators is developed based on Lagrangian-assumed modes method and implemented with a commercially available symbolic manipulation program (SMP [A19]) on a VAX-11/750 mini computer.

2. Limitations of joint variable feedback control algorithms on flexible manipulators are determined and the results agree very well with the previous linear analysis results in the literature.

3. The relative performance of a group of popular control methods in robot motion control and newly developed adaptive control methods are tested and compared in terms of maximum speed, accuracy and robustness with respect to payload variations. It is shown that the only way joint variable feedback based non-adaptive algorithms can provide robustness is to use them in high-gain feedback form. That

is to say, the commanded motions must be substantially slower than the closed loop system bandwidth.

4. AMFC, algorithms using only joint variable feedback, improve the maximum speeds while providing accuracy and robustness comparable with non-adaptive schemes. Due to the self-adaptation capability of feedback gains as a function of tracking error, controller design can be less conservative in the face of expected parameter variations. Tracking errors are taken care of on-line through the adaptation of controller parameters by an adaptation algorithm. It is shown that the speed performance for comparable accuracy and robustness criteria can be improved by a factor of two. However, as high speed performance requirements are further increased, joint variable feedback AMFC results in very lightly damped structural vibrations, which defines the upper limit of performance for AMFC using joint variable feedback.

5. In order to overcome the problems of lightly damped structural vibration modes, while retaining the advantages of AMFC, a combined control approach is proposed. Large motion part is controlled by the AMFC. Before the vibrations start to dominate while the arm is trying to stop, the control algorithm is switched to one that uses flexible mode information explicity in the feedback. The combined control approach not only improves the performance, but also has attractive implementation advantages.

6. The AMFC, which uses joint variable feedback only, is developed in a new way such that one of the two major assumptions of AMFC design methods is replaced by a much less restrictive condition. Previous design methods require

that the manipulator motion be slow enough so that the closed loop dynamics under *constant linear feedback control* (nominal control) stays constant during the adaptation. The AMFC design procedure presented here requires the manipulator motion be slow enough so that the closed loop dynamics under *variable nonlinear feedback control* (nominal control) stays constant during the adaptation.

1.5. Organization of the Thesis

Symbolic modeling of flexible manipulators is discussed in Chapter 2. Chapter 3 discusses the linear analysis results of closed loop dynamics of flexible manipulators under joint variable feedback control. Limitations of joint variable feedback controllers, and root locus sensitivity of closed loop dynamics as function of feedback gains are studied and results are discussed (Chapter 3).

Chapter 5 is the natural complement of Chapter 4. In Chapter 4, performance of computed torque, decoupled joint control on rigid and flexible manipulators are simulated and results are discussed refering to the results of linear anaysis in chapter 3. Furthermore, an AMFC algorithm is developed for flexible manipulators. Advantages and shortcomings of this method are determined. The need for combined control arose naturally at this point of analysis. Chapter 5 presents a combined control approach made up of AMFC for gross motion and LQR for fine motion control which involves explicit control of flexible vibrations as well as joint variable position control.

Chapter 6 presents the conclusions of this work and recommendations for future work.

Appendix A lists the lightweight manipulator parameters used in the simulations. Appendices B, and C, contain the details of the mathematical analysis of Chapter 4 for the interested readers. Appendix D provides the tabulated quantitative results for reference.

Throughout this paper, the *performance* of a control algorithm refers to the maximum speed at which a motion can be executed, while providing good tracking and flexible mode response for accuracy, for a wide range of payload variations, and noise uncertainty. *Joint variable feedback AMFC* means an AMFC algorithm which requires only joint position and velocity measurement information in real time for implementation. The control algorithm development study for flexible arms starts with characterizing what the well known joint variable based non-adaptive and adaptive methods can do, and determines the shortcomings of these approaches. Finally, a combined control approach is presented. All of the simulations are aimed at determining the performance in terms of speed and accuracy for two different implementation conditions: *first*, under perfect information conditions about the system parameters, measurement and enviorenment, and *second* under non-perfect information conditions (robustness performance). In general robustness is tested with respect to large payload/robot mass ratio variations (0 - 25 % payload/robot mass variations).

Words, such as *method*, *algorithm*, *law* and *flexible* , *lightweight* are used interchangeably throughout the thesis, unless otherwise stated.

CHAPTER II

Symbolic Modeling of Flexible Manipulators

2.1. Introduction

This chapter presents a new systematic algorithm to symbolically derive the full nonlinear dynamic equations of motion of multi-link flexible manipulators. The Lagrangian - assumed modes method is the basis of the new algorithm and it is adapted in a way suitable for symbolic manipulation by digital computers. The advantages of obtaining dynamic equations in symbolic form and of the presented algorithm are discussed. Application of the algorithm to a two-link flexible arm example via a commercially available symbolic manipulation program is presented. Simulation results are given and discussed.

The dynamics of a typical industrial manipulator, with six degrees of freedom, is governed by coupled highly nonlinear ordinary differential equations. These equations present a very complicated problem in control system design, mainly because the present state of knowledge in nonlinear control system theory is very limited. Traditionally, independent servo controllers are designed based on the assumption that nonlinear coupling terms are negligible. However, this assumption is reasonable and the control system performance may be satisfactory only if the speed of manipulator is "relatively slow". Increasing demand for higher industrial productivity requires manipulators that move faster and more accurately. As a result, the speed of manipulators must increase and the independent linear servo controllers,

designed based on the slow motion dynamics, will perform unsatisfactorily.

Background

Modeling and control of a single link flexible arm [Fig. 2.1] has been investigated by many authors [A1,A2,A3,A4]. The system is essentially modeled as Bernoulli-Euler beam and vibration coordinates are approximated by a finite number of assumed mode shapes. This allows the application of finite dimensional linear control theory to the problem.

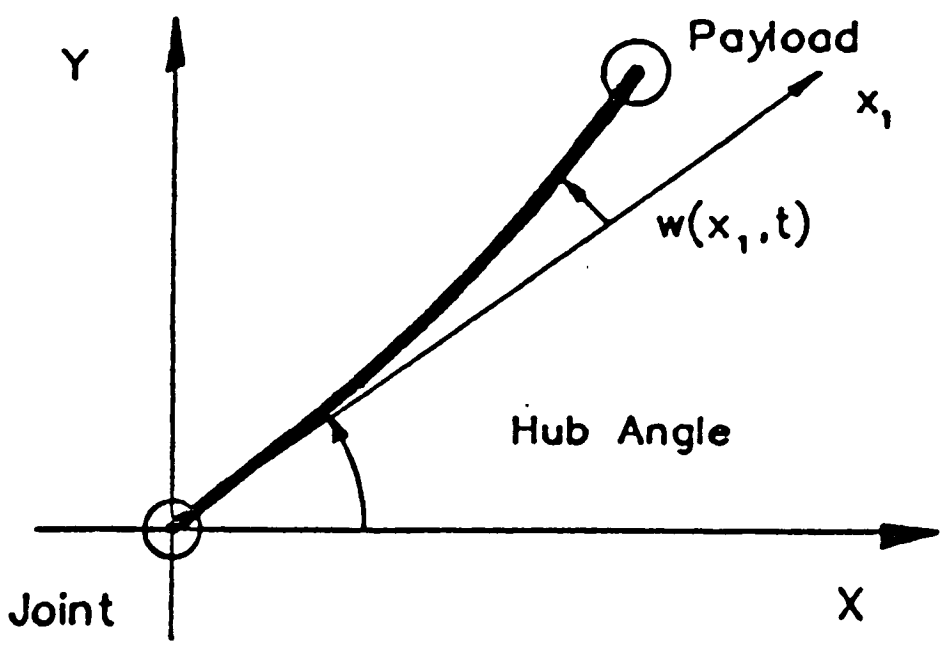


Fig.2.1 One link flexible arm

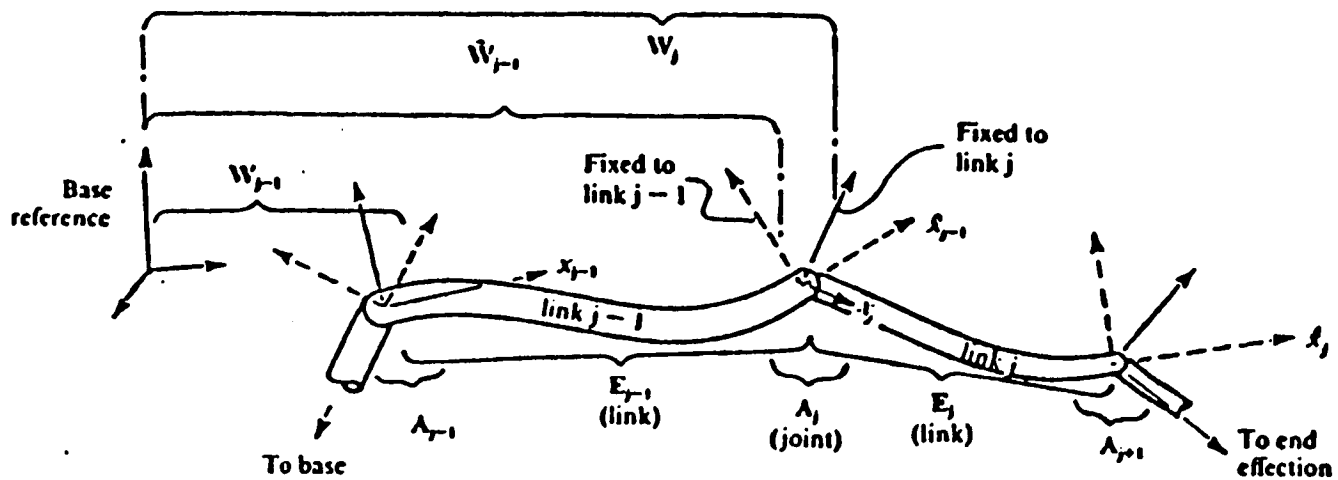


Fig.2.2 A flexible serial manipulator

The multi-link flexible manipulator [Fig. 2.2, and 2.3] modeling and control problem has not been researched to as great an extend as the single-link case. One reason for this is that the modeling problem is not a trivial one. Due to coupling between links, large configuration changes, and high speeds, the system can no longer be accurately represented by simple beam equations. An accurate dynamic model of a lightweight arm involves highly complicated algebraic manipulations and can become impossible to deal with by hand. Moreover, the possiblity of making errors along the way is very high. Making some changes in an existing model also requires long algebraic manipulations. There are two basic methods used in the modeling : 1. Lagrangian-Finite Element based methods, 2. Lagrangian- assumed

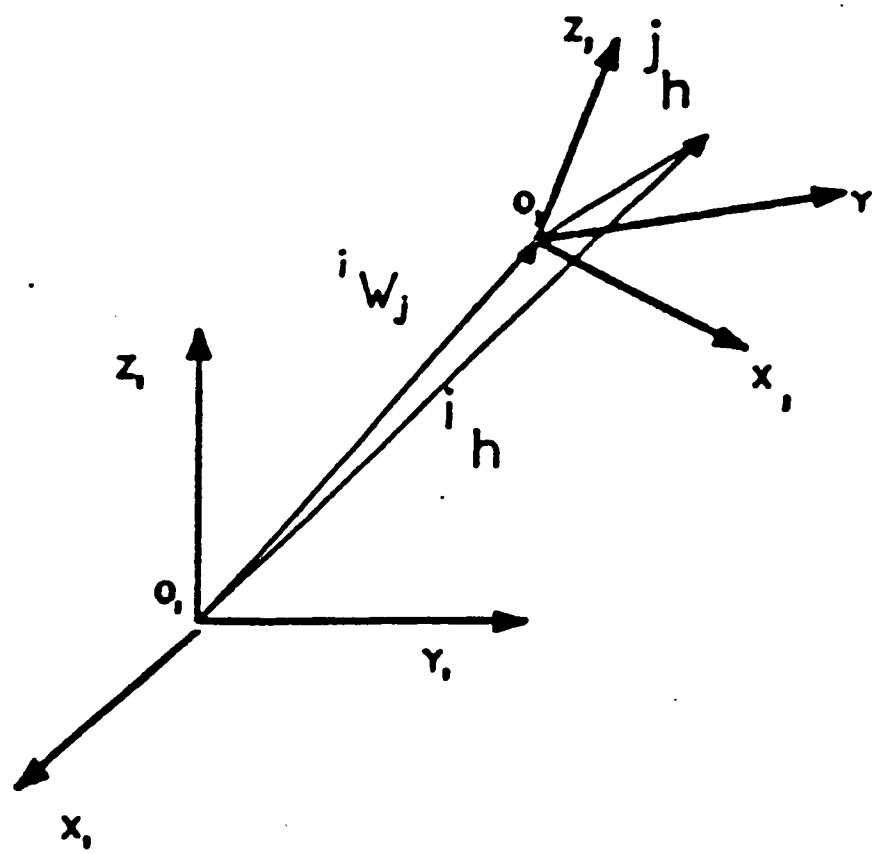


Fig.2.3 (4x4) Homogeneous coordinate transformations

mode based methods. The end result of these methods are essentially the same. Many of the finite element based works on the analysis of closed chain mechanisms can be applied to the dynamic modeling of multi- link flexible arms [A1,A2].

In Sunada and Dubowsky [7,8] the nominal joint variable time histories are asA4uA5ed to be known and the small vibration dynamic model of the manipulators and mechanisms about nominal motions are developed. In Shabana [9] this assumption is removed and full dynamic model is deriveAd. The main advantages of the Lagrangian-finite element method are: a) it is very systematic, b) it can be applied to complex shaped systems, applicable to a very wide class of problems. The disadvatages are a) it requires a substantial amount of software organization, b) it results in a constrained model, c) it does not give much insight to the dynamic structure of the system. Static deflection modes are included in the modes to improve the accuracy of models with a limited number of mode shapes [A2] . Usoro et.al. investigated the performance of LQR with a prescribed degree of stability on a two-link planar arm by digital simulations [A14].

The Lagrangian - assumed modes method is used in the modeling of a two-link robotic manipulator in [A20]. Distributed frequency domain analysis of non-planar manipulators using transfer-matrices has been developed in [A12]. A recursive method using homogeneous transformation matrices to generate full coupled nonlinear dynamics of multi-link flexible manipulators is presented in [A11].

It was the author's experience that the application of this technique to multi-link manipulators works well, but with an importa[A11] wback, namely the algebraic complexity of intermediate steps. When carried out by hand, the length of expres-

sions becomes very large and time consuming. In addition, the possibility of making algebraic errors is quite high. On the other hand, the modeling method is easy to understand, is recursive, does not require any dedicated special software and derives the full nonlinear dynamic model.

Symbolic manipulation programs eliminate the major drawback of the method. Symbolic modeling allows one to model systems with large order in a very short time, check the elements of the dynamic equations in explicit form and manipulate them very conveniently. Leu and co-workers developed programs to obtain dynamic equations for serial rigid robotic manipulators symbolically using commercially available symbolic manipulation programs [A21, A22]. Neuman et. al. generated explicit symbolic equations for the dynamics of a six degree of freedom Puma arm using the ARM symbolic program [A26]. The method presented here is more general in the sense that it can handle structural flexibilities and it contains the rigid manipulator modeling problem as a special case.

The remaining part of this chapter is organized as follows: Section 2.2 summarizes the Lagrangian - assumed Modes method. Section 2.3 presents a new algorithm which adapts this method to a form suitable for symbolic manipulation by digital computer. In section 2.4, the algorithm is applied to a two-link flexible arm example. Application details and simulation results are then discussed.

2.2. Lagrangian - Assumed Modes Method

Kinematics: The first step in the dynamic modeling of any mechanical system is to establish the kinematical relationships and to define fundamental vector quantities: position, velocity and acceleration. Consider the kinematic structure

shown in [Fig. 2.2] representing a manipulator with serial links and joints. Let the coordinate systems used for kinematics of the system be;

O_0XYZ – Fixed to base (Global Coordinate Frame) ,

O_ixyz – Fixed to the base of the link i ,

$O_i'xyz$ – Fixed to the end of link i ,

If the arms are rigid then the $O_i'xyz$ coordinates are not needed. The position vector of any point on link i with respect to O_ixyz coordinates, ${}^ih(x_i)$, can be expressed as

$$ {}^ih(x_i) = [x_i, 0, 0, 1]^T + [w_x(x_i,t), w_y(x_i,t), w_z(x_i,t), 0]^T \tag{2.1} $$

where; $w_x(x_i,t), w_y(x_i,t), w_z(x_i,t)$ are the displacements of the flexible arm due to flexibility in x, y, z directions, respectively. The dependence of the w's on the spatial coordinates makes the system infinite dimensional, leading to coupled ordinary and partial differential equations of motion. In general these are approximated by finite series consisting of spatial variable dependent functions multiplied by time-dependent generalized coordinates. Once the number of generalized coordinates to be used to represent the distributed flexibility of each link has been decided on, the w's can be approximated as;

$$ w_\beta(x_i,t) = \sum_{j=1}^{n_i} \phi_{\beta j}(x_i) \cdot \delta_j(t) \quad ; \qquad \beta : x, y, z \tag{2.2} $$

where n_i is the number of assumed mode shapes used for link i for the w_β, $\phi_{\beta j}(x_i)$ are assumed mode shape functions from an admissable class, $\delta_j(t)$ are the generalized coordinates of approximation, ${}^ih(x_i)$ is uniquely defined. Next we need to be able to transfer this position vector with respect to a global coordinate frame to obtain the absolute position vector. Let ${}^oW_i$ be the homogeneous matrix transformation from

moving coordinate frame $O_i xyz$ to fixed inertial frame $O_o XYZ$. Then the absolute position vector is given by (2.3), (Fig.2.3). It is clear that the transformation ${}^oW_i$ consists of two parts: joint variables and flexible deflections. More clearly, [Fig. 2.2]

$$ {}^oh(x_i) = {}^oW_i \cdot {}^ih(x_i) \tag{2.3} $$

$$ {}^oW_i = {}^oW_{i-1} \cdot E_{i-1} \cdot A_i \tag{2.4} $$

where;

A_i = the transformation between $O_i xyz$ and $O'_{i-1} xyz-$ joint transformation

$E_{i-1}-$ the transformation from the link (i-1) end coordinates to link (i-1) base coordinates.

${}^oW_{i-1}-$the total transformation to the base cordinates from the link (i-1) base coordinates. The form of these transformation matrices are ;

$$ {}^jW_i = \begin{bmatrix} \cdot & \cdot & \cdot & x_j \ component \ of \ O_i \\ \cdot & {}^jR_i & \cdot & y_j \ component \ of \ O_i \\ \cdot & \cdot & \cdot & z_j \ component \ of \ O_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5} $$

${}^jR_i$ is (3x3) matrix of direction cosines, 0^T (1x3);

$$ E_i = \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \sum_{j=1}^{n_i} \delta_{ij}(t) \cdot \begin{bmatrix} 0 & -\theta_{zij} & \theta_{yij} & x_{ij} \\ \theta_{zij} & 0 & -\theta_{xij} & y_{ij} \\ \theta_{yij} & \theta_{xij} & 0 & z_{ij} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.6} $$

where;

$\theta_{\beta ij}$ = rotation components of link i due to mode j, assuming small rotations due to flexible deflections,

l_i = the length of the link i.

Once the kinematic description of the system is set up, the process of obtaining the equations of motion is as follows:

1. Pick generalized coordinates (natural choices are joint variables and a finite number of assumed modes series approximation for every flexible element).

2. Form the kinetic, potential energy, and virtual work for the system.

3. Take the necessary derivatives of the Lagrangian Equations and assemble the equations.

If the system has N_j number of joints with a single degree of freedom and N_l number of flexible links with n_i modal coordinates for each element, the dynamic model of the system will be governed by a set of

$$N_j + \sum_{i=1}^{N_l} n_i \tag{2.7}$$

coupled second order ordinary differential equations.

2.3. Symbolic Implementation of the Lagrangian - Assumed Modes Method

Although the Lagrangian - assumed modes method is theoretically very well understood and documented [13], it is not in a form suitable for symbolic implementation on a digital computer, i.e. insufficient memory problems are likely to occur. Let us first specify some desired features of a modeling algorithm.

First, the mode shapes and the mode shape dependent parameters should be easily varied by the analyst. The selection of "appropriate" or "best" mode shapes for a given flexible system is not a clearly answered problem [12]. One should be able to easily simulate the effect of different mode shapes on the system behavior. For the case of a simple beam under bending vibrations the mode shapes effectively determine the natural frequencies of the system. Effective mass and spring matrix elements are functions of mode shapes as; (with simple boundary conditions)

$$m_{ij} = \int_0^{l_i} \rho A(x)\phi_i(x)\phi_j(x)dx \tag{2.8}$$

$$k_{ij} = \int_0^{l_i} EI(x)\phi_i''(x)\phi_j''(x)dx \tag{2.9}$$

If the mode shapes are orthonormalized such that $m_{ij} = 1$ for $i = j$ and 0 , for $i \neq j$, then $k_{ij} = w_i^2$ for $i = j$, 0 for $i \neq j$. The most accurate approach would be to update the mode shapes as the boundary conditions of the links vary as function of controller impedance.

Second, a recursive algorithm is very desirable. For instance, when the number of modal coordinates is increased or additional links included, the dynamic modeling process should not hace to be repeated again. *Third*, method should elim-

inate any unnecessary algebraic operations so that it would be efficient and require less memory.

The equations governing the dynamics of the system are given by;

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i}\left(\sum_{j=1}^{N} KE_j\right) - \frac{\partial}{\partial q_i}\left(\sum_{j=1}^{N} KE_j\right) + \frac{\partial}{\partial q_i}\left(\sum_{j=1}^{N} PE_j\right) = Q_i \tag{2.10}$$

where N is the total number of discrete elements in the system (joints, links, payload).

$$\sum_{i=1}^{N} PE_i = \sum_{i=1}^{N} (PE)_{i\ gravitational} + (PE)_{i\ elastic} \tag{2.11}$$

The q_i 's are the generalized coordinates which are joint variables and flexible mode shape coordinates of flexible elements. Kinetic energies for rotary joints, if considered as a mass with rotary inertia about the axis of rotation, are

$$(KE)_{joint\ i} = (1/2)m_i V_{gi}^2 + (1/2)\vec{H}_{gi} \cdot \vec{w}_i \tag{2.12}$$

where

$m_i =$ the mass of joint i,

$V_{gi} =$ the speed of joint i mass center,

$\vec{H}_{gi} =$ the angular momentum vector of joint with respect to its center of mass,

$\vec{w}_i =$ the total angular velocity vector of the joint.

The kinetic energy of the flexible links is

$$(KE)_i = 1/2 \int_0^{l_i} \rho_i(x)\dot{\vec{r}}_i \cdot \dot{\vec{r}}_i \cdot dx \tag{2.13}$$

If all the modal coordinates and associated mode shapes were given, then the integration over the spatial variable could be evaluated. However since the mode shapes and dependent parameters are to be entered later by the user for analysis purposes, we identify all possible elements that are functions of the spatial variables of link i and assign them parametric names. KE_i is spatially dependent only because of the link i flexibility. The effect of previous element flexibilities on KE_i are reflected in W terms which depend only on resulting end point motions, and thus have no spatial variable dependence. From (2.3)

$$ {}^{o}\dot{h}_i(x) = {}^{o}\dot{W}_i \cdot {}^{i}h_i(x) + {}^{o}W_i \cdot {}^{i}\dot{h}_i(x) \tag{2.14} $$

$$ \begin{aligned} \dot{\vec{r}}_i \cdot \dot{\vec{r}}_i &= {}^{o}\dot{h}_i^T(x) \cdot {}^{o}\dot{h}_i(x) \\ &= {}^{i}h_i^T(x)\, {}^{o}\dot{W}_i^T \cdot {}^{o}\dot{W}_i\, {}^{i}h_i(x) + {}^{i}\dot{h}_i^T(x)\, {}^{o}W_i^T\, {}^{o}\dot{W}_i\, {}^{i}h_i(x) \\ &\quad + {}^{i}h_i^T(x) \cdot {}^{o}\dot{W}_i \cdot {}^{o}W_i \cdot {}^{i}\dot{h}_i(x) + {}^{i}\dot{h}_i^T(x) \cdot {}^{o}W_i^T \cdot {}^{o}W_i \cdot {}^{i}\dot{h}_i(x) \end{aligned} \tag{2.15} $$

where;

$$ {}^{i}h_i^T(x) = \left[x + \sum_{j=1}^{n_i} \phi_{xij}(x)\delta_{xij}(t), \sum_{j=1}^{n_i} \phi_{yij}(x)\delta_{yij}(t), \sum_{j=1}^{n_i} \phi_{zij}(x)\delta_{zij}(x), 1 \right] \tag{2.16} $$

$$ {}^{i}\dot{h}_i^T(x) = \left[x + \sum_{j=1}^{n_i} \phi_{xij}(x)\ \dot{\delta}_{xij}(t), \sum_{j=1}^{n_i} \phi_{yij}(x)\ \dot{\delta}_{yij}(t), \sum_{j=1}^{n_i} \phi_{zij}(x)\ \dot{\delta}_{zij}(x),\ 0 \right] \tag{2.17} $$

Elements of the transformations ${}^{o}W_i$ and ${}^{o}\dot{W}_i$ are functions of the generalized coordinates and parameters of the links $k < i$, such as $\{\theta_i, \theta_i\theta_k, \phi_{\beta kj}(l_k), \delta_{\beta kl}(t), \theta_k(t),$ where $k = 1, \ldots, i-1;\ \beta = x, y, z\}$, l_k is the length of the link k.

In general for serial link robotic manipulators, the kinetic energy of link i will have the following form ; $(\star)$ is used to indicate the possible existence of terms

that are independent of spatial variable x .

$$
\begin{aligned}
(KE)_i = {} & (\star)\int \rho(x)dx + (\star)\int \rho(x)\ xdx + (\star)\int \rho(x)\ x^2dx \\
& + \sum_{\beta,\xi}\sum_{j}\int \rho(x)\phi_{\beta ij}(x)\phi_{\xi ij}(x)dx \cdot \left[(\star)\dot{\delta}_{\beta ij}\dot{\delta}_{\xi ij} + (\star)\dot{\delta}_{\beta ij}\delta_{\xi ij} + (\star)\delta_{\beta ij}\delta_{\xi ij}\right] \\
& + \sum_{\beta}\sum_{j}\int \rho(x)\phi_{\beta ij}(x)xdx\ \left[(\star)\delta_{\beta ij} + (\star)\dot{\delta}_{\beta ij}\right] \\
& + \sum_{\beta}\sum_{j}\int \rho(x)\phi_{\beta ij}(x)dx\ \left[(\star)\delta_{\beta ij} + (\star)\dot{\delta}_{\beta ij}\right]
\end{aligned}
\tag{2.18}
$$

where; β and $\xi : x, y, z,\ \ j = 1, ...n_i$. At the At this point, from a symbolic modeling point of view it is not important what these $(\star)$ terms are. But what is important is to extract all the possible combination of spatial-variable dependent terms and replace them with symbolic names so that the first objective of the modeling is accomplished. At the calculation of the absolute velocity of a differential element of a flexible member, the parameters which are functions of the spatial variable can be extracted and be given symbolic names by the symbolic manipulation program very easily. These parameters represent the elements in the dynamic model which are functions of mode shapes, link length, and mass distribution of the flexible element.

In the absolute velocity square expression (2.15), all parameters that are functions of the mode shapes can be replaced with symbolic names (2.19) at the modeling level. Then, defining the same symbolic names as in (2.20) automatically gives the kinetic energy expression for element i from (2.15). Thus the kinetic energy expression (2.18) is not evaluated explicitly, but symbolically obtained directly from (2.15).

Replace in (2.15) the following symbolic names (2.19):

$$
\begin{aligned}
nm\beta\xi ijk &\longleftarrow \phi_{\beta ij}(x)\phi_{\xi ij}(x) \\
nw\beta ij &\longleftarrow \phi_{\beta ij}(x)\cdot x \\
nq\beta ij &\longleftarrow \phi_{\beta ij}(x) \\
J_{oi} &\longleftarrow x^2 \\
m_i \cdot l_i/2 &\longleftarrow x \\
m_i &\longleftarrow 1
\end{aligned}
\tag{2.19}
$$

and in the simulation level evaluate these terms by multiplying with $\rho(x)$ and integrating over the link length.

$$
\begin{aligned}
nm\beta\xi ijk &= \int_o^{l_i} \rho(x)\phi_{\beta ij}(x)\phi_{\xi ik}(x)dx \\
nw\beta ij &= \int_o^{l_i} \rho(x)\phi_{\beta ij}(x)\cdot x dx dx \\
nq\beta ij &= \int_o^{l_i} \rho(x)\phi_{\beta ij}(x)dx \\
J_{oi} &= \int_o^{l_i} \rho(x)\cdot x^2 \cdot dx \\
m_i \cdot l_i/2 &= \int_o^{l_i} \rho(x)\cdot x \cdot dx \\
m_i &= \int_o^{l_i} \rho(x)\cdot dx;
\end{aligned}
\tag{2.20}
$$

These are the six basic parameters related to the inertia properties of the flexible element and with their use there is no longer spatialvariable dependence in the kinetic energy expressions. With this approach one can see more explicitly the effect of mode shapes and system parameters on the dynamic model, leading to a better understanding of the dynamics, which is not offered by numerical or other

modeling methods. Notice that if the mode shapes associated with a coordinate (i.e. y) are chosen to be orthonormal with respect to distributed mass and flexibility many of the above terms will be zero, such as $nm\beta\xi ijk = 1$ if $j = k$, 0 if $j \neq k$.

Similiarly for the elastic potential energy of the link i (gravitational potential energy is omited here to save space)

$$\begin{aligned}(PE)_i = (1/2) \sum_{j,k=1}^{n_i} \int_0^{l_i} \Big[& EI_y \Big(\phi''_{yij}(x)\phi''_{yik}(x)\delta_{yij}(t)\delta_{yik}(t) \Big) \\ & + EI_z \Big(\phi''_{zij}(x)\phi''_{zik}(x)\delta_{zij}(t)\delta_{zik}(t) \Big) \\ & + EA(x) \Big(\phi'_{xij}(x)\phi'_{xik}(x)\delta_{xij}(t)\delta_{xik}(t) \Big) \Big] \cdot dx \end{aligned} \tag{2.21}$$

Similiarly

$$k\beta ijk = \int_0^{l_i} EI_\beta(x)\phi''_{\beta ij}(x)\phi''_{\beta ik}(x) \cdot dx \ ; \quad j,k = 1 \ldots n_i; \qquad \beta = y, z$$

$$kxijk = \int_0^{l_i} EA(x)\phi'_{xij}(x)\phi'_{xik}(x)dx$$

$$(PE)_i = (1/2) \sum_{\beta:y,z} \sum_{j=1}^{n_i} \sum_{k=1}^{n_i} [k\beta ijk\delta_{\beta ij}(t)\delta_{\beta ik}(t) + kxijk\delta_{xij}(t)\delta_{xik}(t)] \tag{2.22}$$

The next important topic is the development of a recursive method that will not run into memory problems as the system dimension gets large and that will eliminate unnecessary algebraic operations. Moreover once a model is developed, some variations of the model should be possible without repeating the whole modeling process. As the system dimension gets larger, carrying out the derivations

using total energy expressions can easily run into memory problems. Thus,

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i}\left(\sum_j KE_j\right) - \frac{\partial}{\partial q_i}\left(\sum_j KE_j\right) + \frac{\partial}{\partial q_i}\left(\sum_j PE_j\right) = Q_i \tag{2.23}$$

$$\sum_j \left[\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i}(KE_j) - \frac{\partial}{\partial q_i}(KE_j) + \frac{\partial}{\partial q_i}(PE_j)\right] = Q_i \tag{2.24}$$

Due to the serial nature of the manipulator arm;

$$\frac{\partial}{\partial \dot{q}_i}(KE_j) = \frac{\partial}{\partial q_i}(KE_j) = \frac{\partial}{\partial q_i}(PE_j) = 0 \quad ; \quad \text{for } i > j \tag{2.25}$$

The equations of motion of the system are found to be;

$$\sum_{j=1}^{N}\left[\frac{d}{dt}\frac{\partial}{\partial \dot{q}_i}(KE_j) - \frac{\partial}{\partial q_i}(KE_j) + \frac{\partial}{\partial q_i}(PE_j)\right] = Q_i \quad ; \quad \text{for } \ i = 1, \ldots, j \tag{2.26}$$

The following algorithm, in combination with equation (2.26), can be effectively programmed in any commercially available general purpose symbolic manipulation program to obtain dynamic model equations of multi-link flexible robotic manipulators symbolically.

Algorithm:

for $j := 1$ *to* N,

for $i := 1$ *to* j,

Find and store KE_j, PE_j ; (2.18) *and* (2.22)

$$\frac{\partial}{\partial \dot{q}_i}(KE_j), \ \frac{\partial}{\partial q_i}(KE_j), \ \frac{\partial}{\partial q_i}(PE_j),$$

Next i,

Next j,

Given the results of the algorithm, substitute these to equation (2.26) and assemble the equations in a convenient form for simulation and analysis purposes. After the equations are assembled, it is very easy to program them in one of the standard scientific programming languages using the capabilities of the commercial symbolic manipulation packages [14].

Let us assume that after modeling a manipulator , it is desired to add another link to the model with m_i degrees of freedom. Based on the above algorithm one must evaluate ;

For $i := 1$ **to** $N + m_i$,

$$\frac{\partial}{\partial \dot{q}_i}(KE_{N+1}), \frac{\partial}{\partial q_i}(KE_{N+1}), \frac{\partial}{\partial q_i}(PE_{N+1}) \quad ; \qquad (2.27)$$

Next i ,

Let us assume that the previous model was assembled in the form:

$$\Big[M\Big]\ddot{q} + f = Q \qquad (2.28)$$

where the inertia matrix dimension is (NxN), q, f, Q, vector dimensions are (Nx1); N is the total number of generalized coordinates up to that point.

The additional link contribution is of the form:

$$\begin{bmatrix} m_{n,n} & m_{n,n+1} \\ m^T_{n,n+1} & m_{n+1,n+1} \end{bmatrix} \begin{bmatrix} \ddot{q}_{n,n+1} \\ \ddot{q}_{n+1,n+1} \end{bmatrix} + \begin{bmatrix} f_{n,n+1} \\ f_{n+1,n+1} \end{bmatrix} = \begin{bmatrix} Q_{n,n+1} \\ Q_{n+1,n+1} \end{bmatrix} \qquad (2.29)$$

where the inertia matrix is of dimension (N+n_i) x (N+n_i) and the vector quantities are of (N+n_i) x 1 dimension. The partition of the equation (2.29) is made so as

to clearly reflect the increase in the dimension of the system compared to (2.28). The complete equations of motion are obtained by the addition of (2.29) to (2.28), where (2.28) is extended to (2.29) dimensions with additional zeros corresponding to the new generalized coordinates q_{n+1} introduced by the new element.

The implementation adapted here has the following advantages:

a) the mode shapes and dependent parameters can be easily varied,

b) all unnecessary differentiation is avoided,

c) the technique is recursive, and

d) memory problems are not likely to occur.

2.4. Applications and Discussion of Simulation Results

Here the described modeling method is applied to a two-link planar flexible arm, with rotary joints and payload. Two mode shapes for each link are considered to represent the structural flexibilities. As noted earlier, mode shapes can be input into the simulation program and the effect of different mode shapes on the dynamic response and the accuracy of modes can be checked. Joints and the payload are considered as masses with rotary inertia. These inertial parameters can be set to zero as well [Fig. 2.4]. The system input parameters for the simulation are as follows:

Joint 1 mass and rotary inertia about its center of mass ; mj_1, jj_2.

Similiarly for joint 2 ; mj_2, jj_2, and for payload ; m_p, j_p

For link 1 and 2 ; mass per unit length, link lengths, flexural rigidity constants,

$\rho A_1, \rho A_2, l_1, l_2, EI_1, EI_2$

Assumed mode shapes and gravity vector, $\phi_{11}(x)$, $\phi_{12}(x)$, $\phi_{21}(x)$, $\phi_{22}(x)$; g_x, g_y, g_z , and

The initialization procedure.

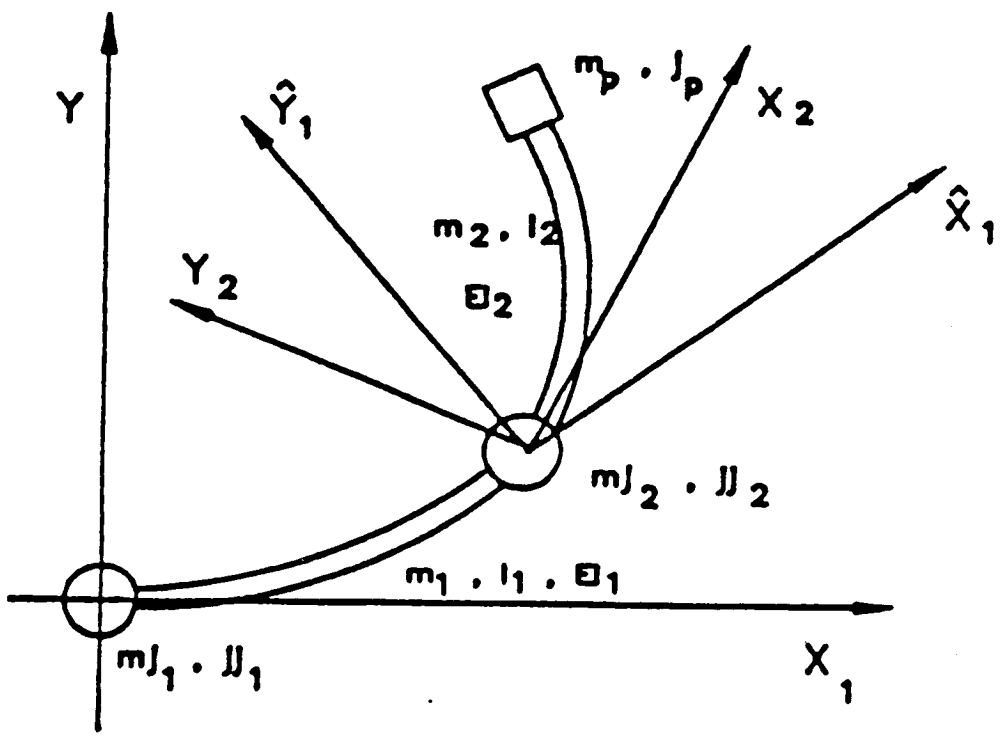


Fig.2.4 Two link flexible arm example

Time independent parameters are calculated at the initialization of the program only once per session. If mode shapes are updated as functions of changing boundary conditions, then these parameters need to be reevaluated. These parameters are:

$$nm11, nm12, nm21, nm22, nw11, nw12, nw21, nw22$$

$$nq11, nq12, nq21, nq22, kw11, kw12, kw21, kw22$$

$$\phi_{11}(l_1), \phi_{12}(l_1), \phi_{21}(l_2), \phi_{22}(l_2)$$

$$\frac{\partial}{\partial x}(\phi_{11}(x))|_{x=l_1}, \frac{\partial}{\partial x}(\phi_{12}(x))|_{x=l_1}, \frac{\partial}{\partial x}(\phi_{21}(x))|_{x=l_2}, \frac{\partial}{\partial x}(\phi_{22}(x))|_{x=l_2}$$

The objectives of digital simulations are as follows. 1. Verify that the model generated by the above algorithm is correct. 2. Demonstrate the ease of changing mode shapes and the resulting change in the dynamic response due to the different mode shapes used in the model.

1. Model verification will be done by comparing the response of the flexible arm model with that of a rigid arm, which has the same corresponding parameters.

a) Clearly as the flexural rigidity, $EI(x)$, of the links increases, the joint angle response of the flexible model should converge to that of the rigid model response. Figures (2.5) and (2.6a-b) show that the joint angle responses do indeed converge to those of the rigid arm case, as the flexural rigidity, EI, of the links is increased.

b) The same test simulation was done with clamped- clamped mode shapes for the first link. For this case, when EI is set to $100Ntm_2$, the joint angle responses were almost the same as the rigid case (See Fig. 2.5 and 2.7a-b). The reason for the faster convergence of the clamped-clamped case than the clamped-free case is that

clamped-clamped mode shapes result in a stiffer system. However, the clamped-free case provides a more accurate prediction of the system response than the clamped-clamped case, as discussed below.

c) As $EI(x)$ increases, the frequencies associated with structural flexiblity should increase, for the simple beam case natural frequencies are fuctions of EI as

$$w_i = (\gamma_i/l)^2 \sqrt{EI/\rho A} \tag{2.30}$$

where; γ_i is the characteristic value of the simple beam eigenvalue problem. Even though in the two link arm case we are considering here (2.30) does not hold exactly, it is still valid in principle and gives a quantitative idea about what to expect. Rayleigh's energy principle also supports this expectation. Figures (8a and 8b) confirm these expectations.

2. modeling method cleary reveals that mode shapes are important parameters of the system dynamics (e.g. Eqn (2.19)). What assumed mode shapes should be used? Would different shapes make an important difference in the system dynamic characteristics? Theoretically, the only constraint on the assumed mode shapes is that they must satisfy the geometric boundary conditions, but not neccessarily the natural boundary conditions nor the governing differential equations. The governing differential equations and natural boundary conditions are results of the functional variation of the Hamiltonian and are approximately satisfied in any case. The controlled end of each link, driven by a high gain feedback controller, behaves more like a clamped end [A15]. The other end condition of the intermediate links should be approximated by a mass with rotary inertia due to other links of the serial structure and payload. However, for different structures and even for different pay-

loads the resultant simple beam analysis will give different mode shapes. Given the fact that these are natural boundary conditions and will be approximately satisfied even if assumed mode shapes do not satisfy them, a clamped-free simple beam mode shape would be an appropriate choice for the assumed modes used in the model. The clamped-clamped case results in a stiffer system. As a result, the joint variable response converges to the rigid arm response much faster than the clamped-free case as a function of flexural rigidity (See Fig. 2.5, 2.6, 2.7). The frequency of flexible vibrations are significantly higher than those of the clamped-free case for the same parameters and conditions (See Fig. 2.8). This analysis further reveals the importance of mode shapes in the dynamic behavior of the system, hence the importance of keeping the mode shapes as parameters in general at the stage of model equation generation.

2.5. Conclusion

From the modeling technique point of view, it has been shown that Lagrangian - assumed modes method can be effectively used for multi-link flexible arms. The availability of general purpose symbolic manipulation programs overcomes the algebraic complexity of the derivation steps, and allows the researcher to obtain more complete models in very short time, in spite of their complexity. A new systematic algorithm based on Lagrangian-assumed mode method is presented suitable for symbolic manipulation by digital computers. The algorithm results in scalar dynamic equations of motion of the system in explict form. There is one scalar diferential equation for each generalized force. This is very useful in the parallel computation of control torques based on inverse dynamics (computed-torque) since the computation task of each of the scalar equations can be assigned to a

single processor which are independent of each other. The algorithm is applied to a two link flexible arm. Simulation results are discussed and shown that the method worked very well for this example case.

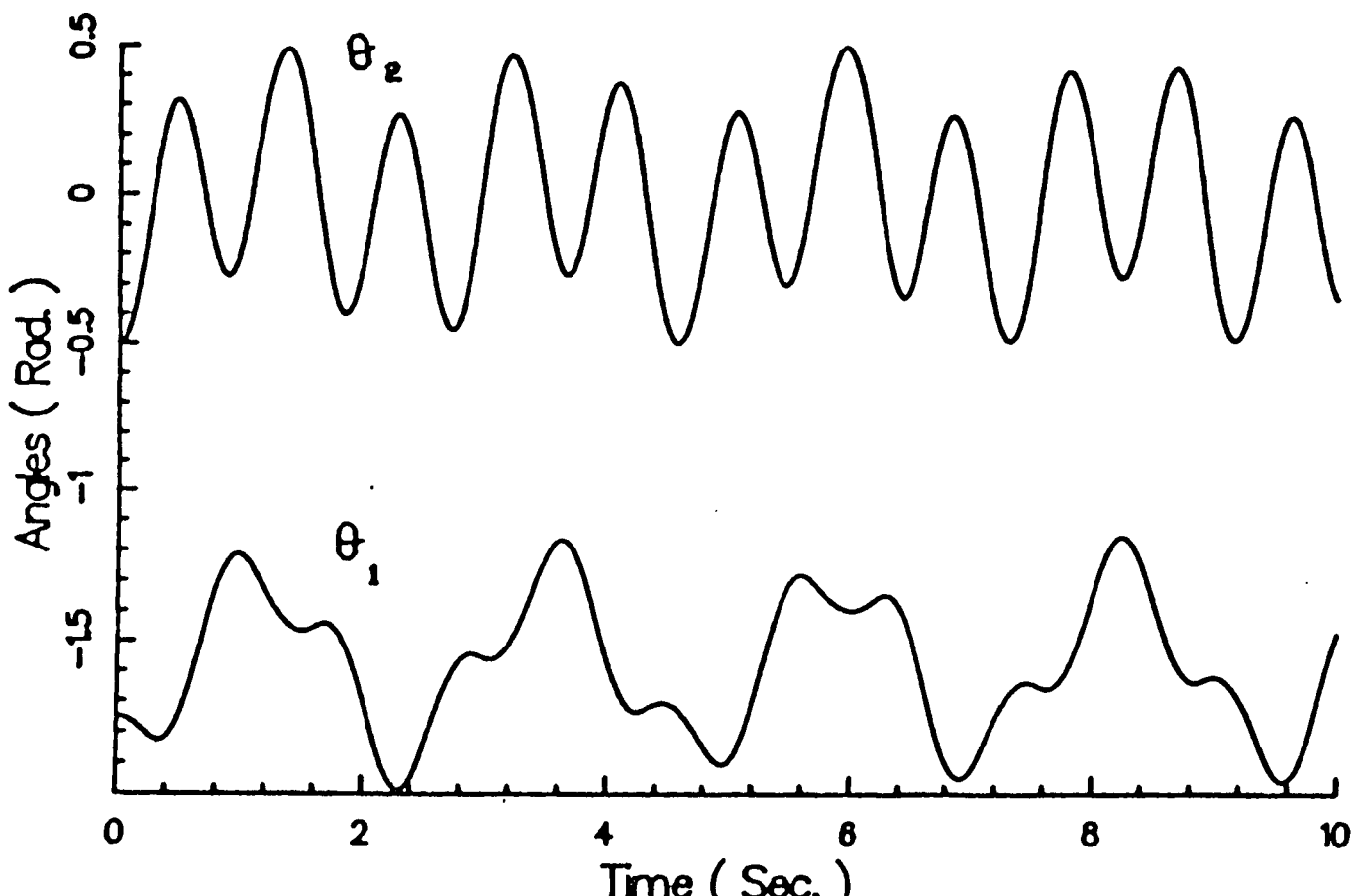


Fig.2.5 Two link rigid model joint angles

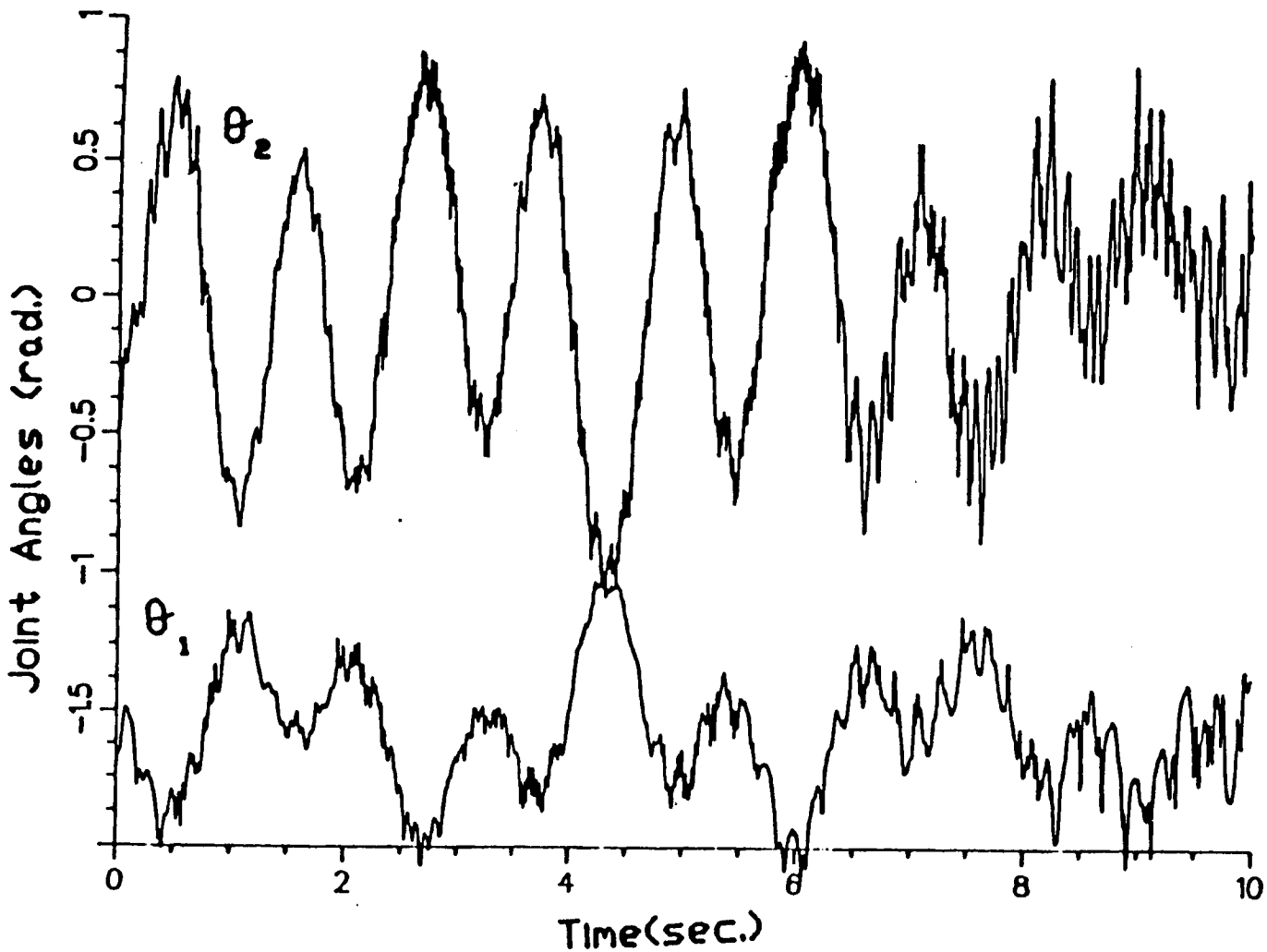


Fig.2.6 Two link flexible model joint angles, clamped-free mode shapes
a) $EI_i = 10.\ Nt.m^2$

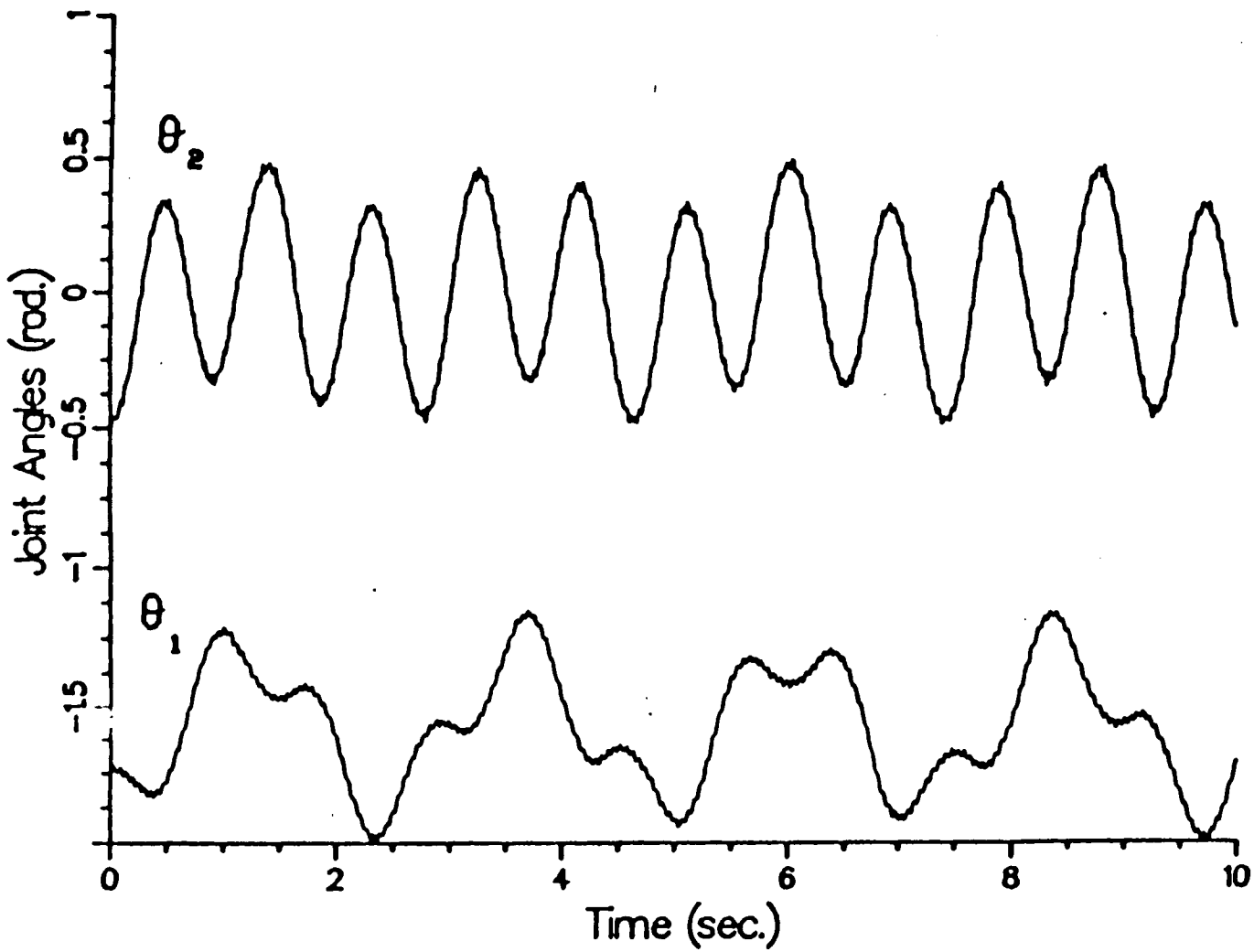


Fig.2.6 Two link flexible model joint angles, clamped-free mode shapes

b) $EI_i = 100.\ Nt.m^2.\ i = 1, 2.$

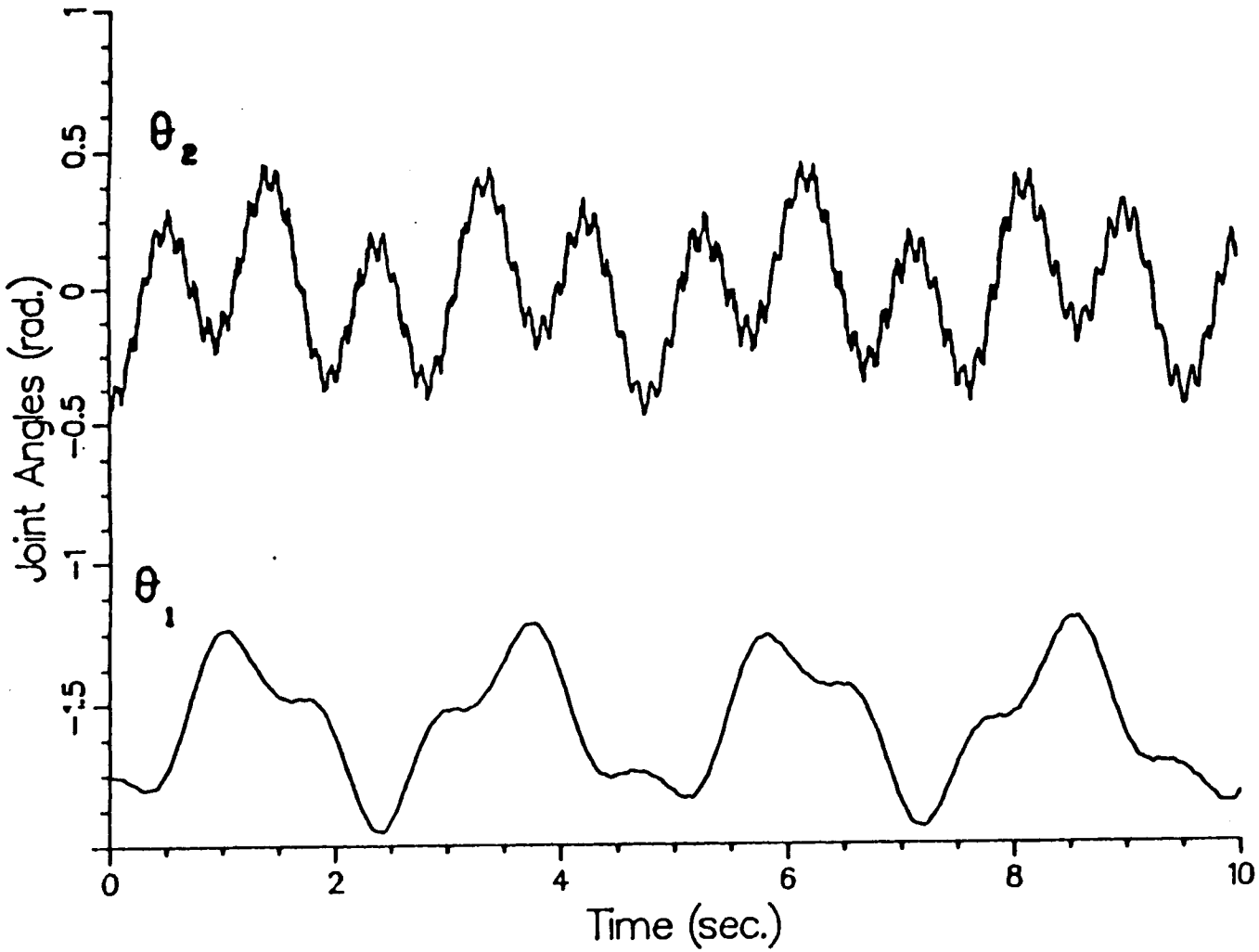


Fig.2.7 Two link flexible model joint angles, clamped-clamped mode shapes for link 1

a) $EI_i = 10.\ Nt.m^2$

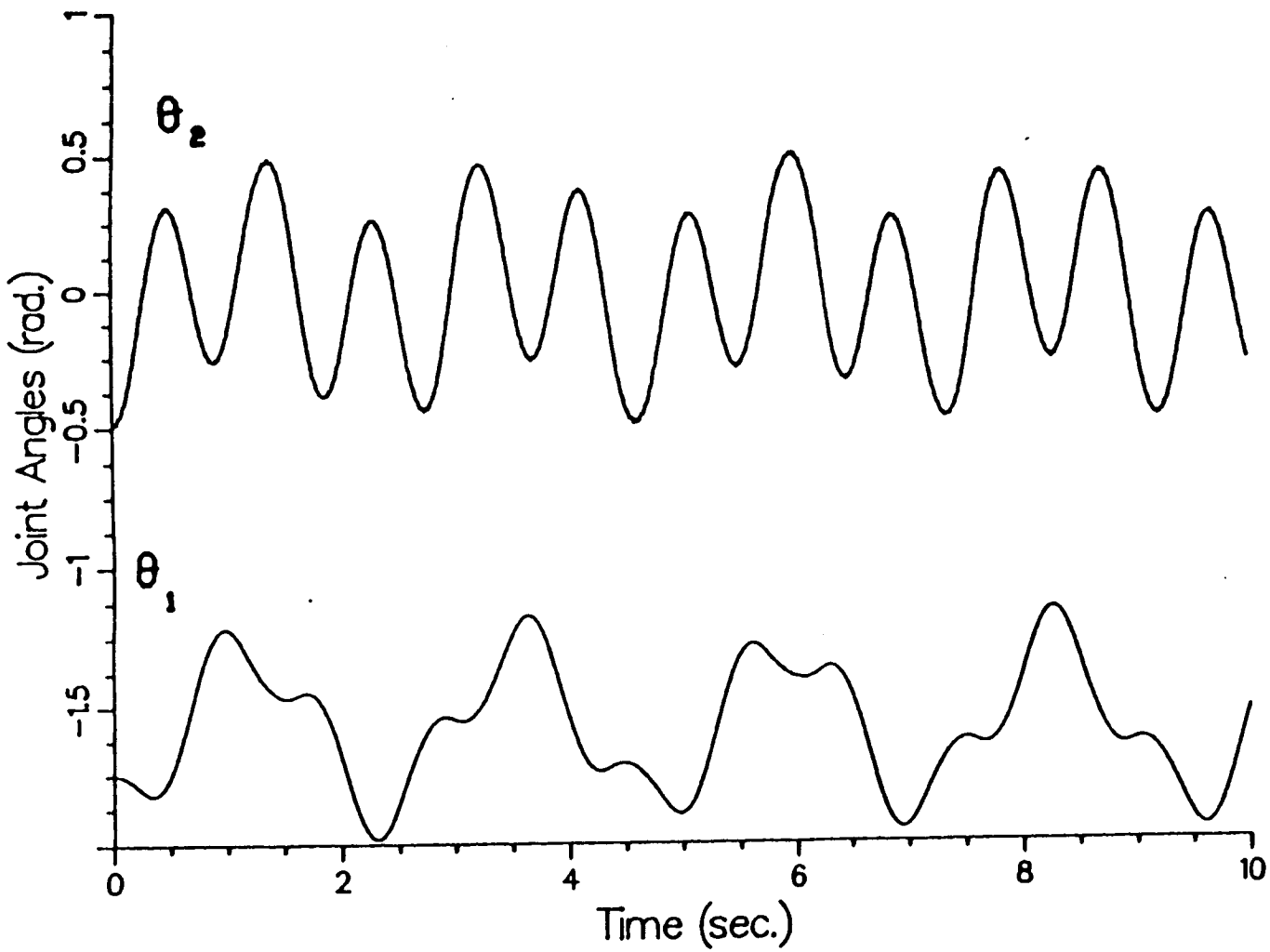


Fig.2.7 Two link flexible model joint angles, clamped-clamped mode shapes for link 1

b) $EI_i = 100.\ Nt.m^2$

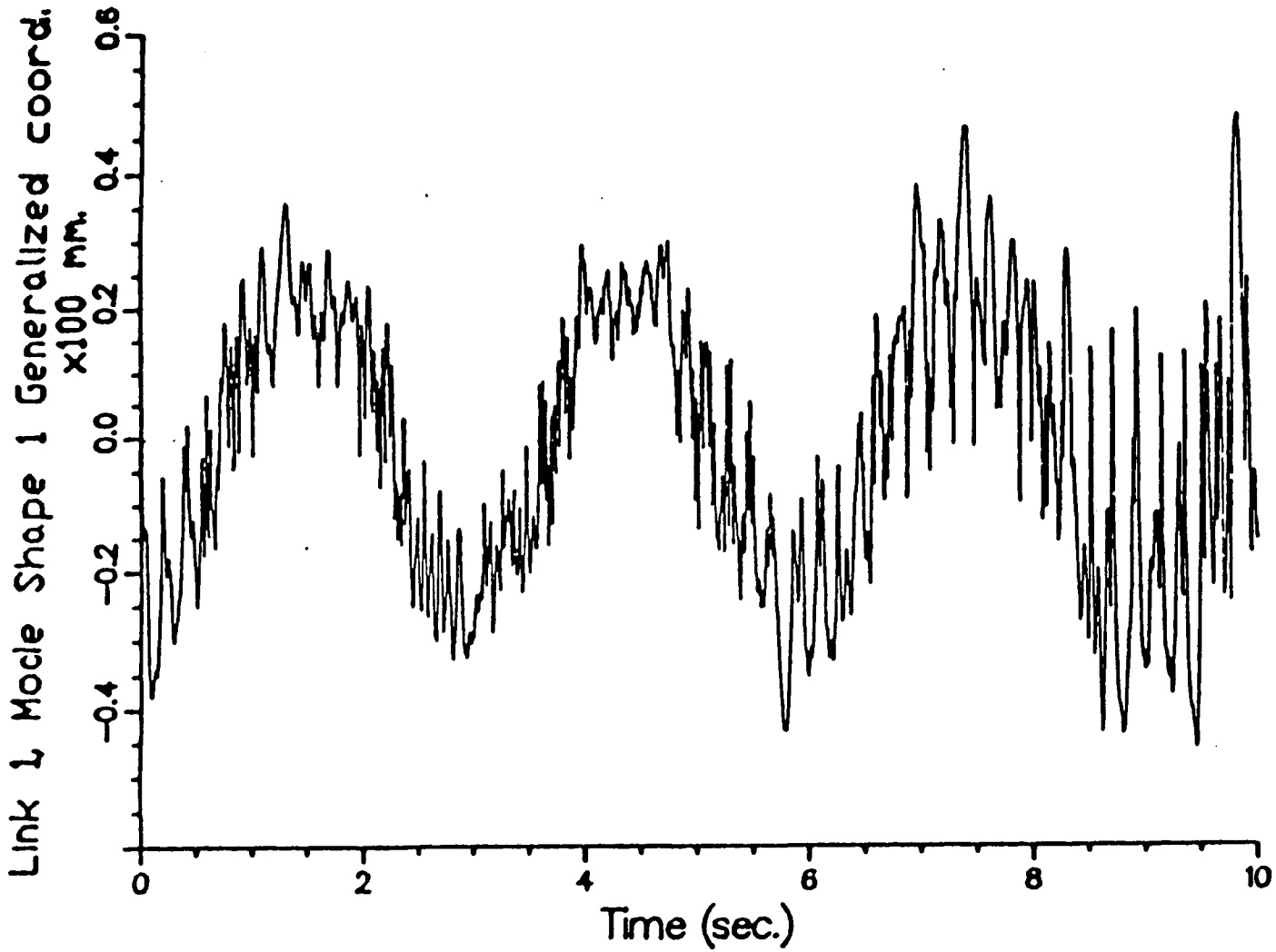


Fig.2.8 Comparisions of flexible vibration coordinate responses (Clamped-free mode shapes)
a) $EI_i = 10.\ Nt.m^2$

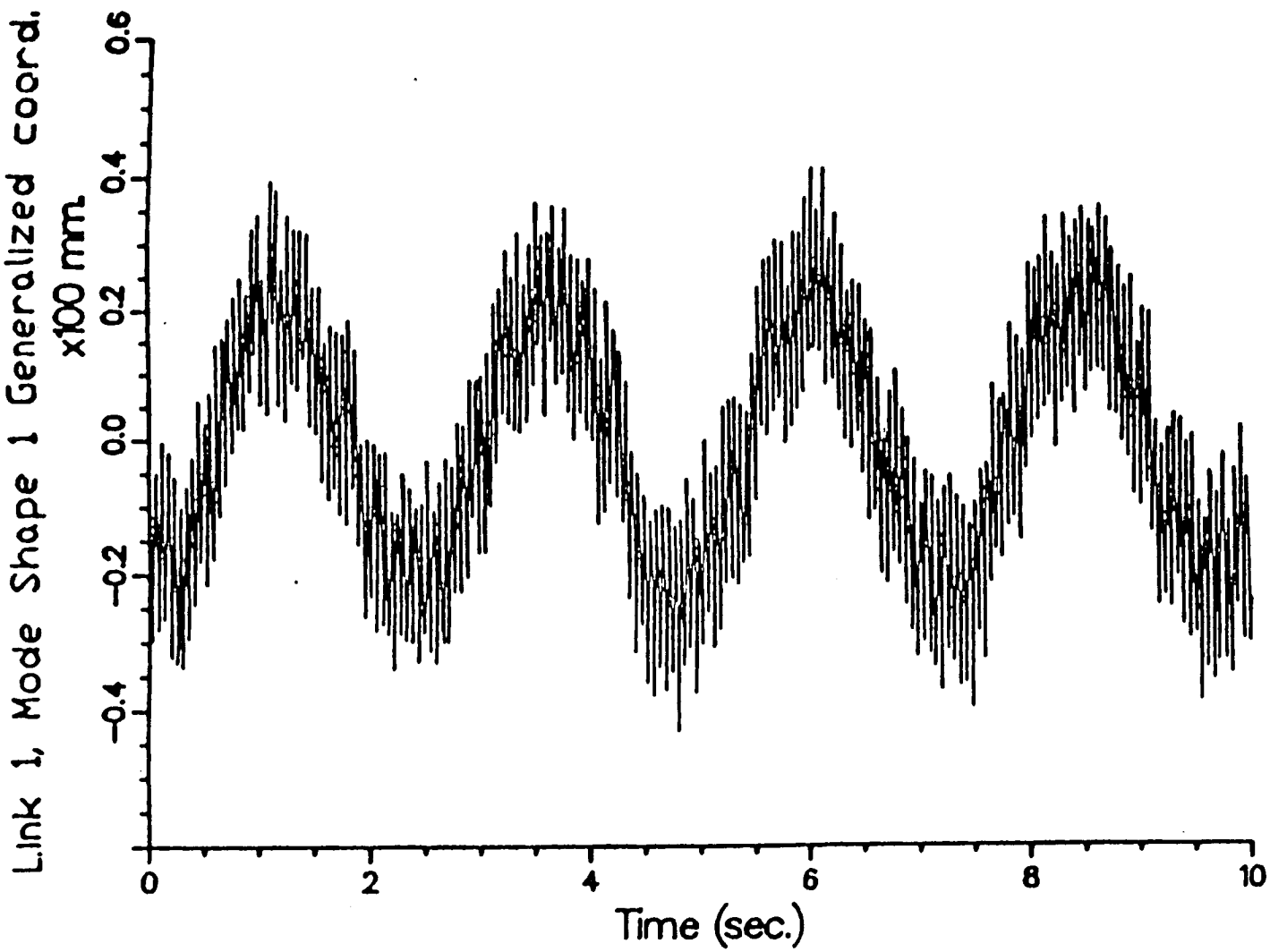


Fig.2.8 Comparisions of flexible vibration coordinate responses (Clamped-free mode shapes)
b)$EI_i = 100.\ Nt.m^2$

CHAPTER III

Open and Closed Loop Dynamics: Linear Analysis

3.1. Introduction

In this chapter open and closed loop linear dynamics of flexible manipulators are examined. The eigenstructure of the manipulator model is studied as a function of arm configuration, payload, flexibility, and joint variable feedback gains. Results are presented by means of root locus diagrams in the s-plane.

In the open loop case, joints are free of any control input. Robot manipulators go through many different configurations and deal with a wide range of payloads. Therefore it is of interest to determine how the open loop eigenvalues of the manipulator vary as a function of manipulator configurations, payload, and other manipulator parameters. The control algorithm must be robust for these variations depending on their significance.

Open and closed loop dynamics are studied for extereme values of manipulator parameters and feedback gains. Each extereme case studied has a corresponding limit system, i.e. as the joint position feedback gains are increased, two link manipulator dynamics should converge to that of a clamped single beam. The accuracy of finite dimensional assumed modes model in representing the limiting case behaviors is discussed.

Performance limitations of joint variable feedback controllers due to arm flexibility are investigated and the results are compared with the previous results reported in the literature.

In summary, it is found that open loop eigenvalues vary significantly due to configuration and payload variations. In the limiting cases, a finite dimensional assumed modes model's lower mode eigenvalues converge to the limit eigenvalues quite accurately. However, the eigenvalues associated with higher modes do not converge as accurately as the lower modes. Performance limitations due to flexibility which are predicted by an assumed modes model, agree very well with the limitations predicted by frequency domain models. The unresolved questions are identified and discussed.

3.2. Open Loop Eigenstructure Analysis

3.2.1. Linearization of the Nonlinear Model

Consider the general nonlinear dynamic model of a flexible robotic manipulator,

$$\begin{bmatrix} M_r(\theta,\delta) & M_{rf}(\theta,\delta) \\ M_{rf}^T(\theta,\delta) & M_f(\theta,\delta) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} f_r(\theta,\delta,\dot{\theta},\dot{\delta}) \\ f_f(\theta,\delta,\dot{\theta},\dot{\delta}) \end{bmatrix} + \begin{bmatrix} 0 \\ [K]\delta \end{bmatrix} + \begin{bmatrix} g_r(\theta,\delta) \\ g_f(\theta,\delta) \end{bmatrix} = \begin{bmatrix} Q \\ 0 \end{bmatrix} \quad (3.1)$$

where $M_r(\theta,\delta), M_{rf}(\theta,\delta), M_f(\theta,\delta)$ are generalized inertia matrix elements, $f_r(\theta,\delta,\dot{\theta},\dot{\delta})$, $f_f(\theta,\delta,\dot{\theta},\dot{\delta})$ are nonlinear centrifugal and coriolis terms, $g_r(\theta,\delta)$,$g_f(\theta,\delta)$ are gravitational terms, and $[K]$ is the structural stiffness matrix associated with arm flexibility and mode shape functions. θ represents the joint variables (vector), and δ represents the generalized coordinates associated with the flexible modes shapes. Q

is the pure input torque vector applied to links at the joints, and has the same dimension as the θ. The zero vector, under the Q of eqn. (3.1), is a result of using clamped mode shapes at the actuated ends of the links.

Let us express (3.1) in a more compact form as follows:

$$\dot{\underline{x}} = \underline{f}(\underline{x}) + B(\underline{x})\ \underline{u} \tag{3.2}$$

For generality, let $\underline{x}_o(t)$ and $\underline{u}_o(t)$ be the nominal states and the nominal input as function of time, and let $\Delta\underline{x}(t)$ and $\Delta\underline{u}(t)$ be the *small* variations from the nominal values. The total state and input vectors are

$$\underline{x}(t) = \underline{x}_0(t) + \Delta\underline{x}(t) \tag{3.3.a}$$

$$\underline{u}(t) = \underline{u}_0(t) + \Delta\underline{u}(t) \tag{3.3.b}$$

Expanding (3.2) into Taylor series about the nominal state and input functions yields

$$\begin{aligned}\dot{\underline{x}}_o(t) + \Delta\dot{\underline{x}}(t) = (\underline{f}(\underline{x}_o(t)) + \Delta\underline{x}(t)\frac{\partial f(x)}{\partial x}|_{\underline{x}_o,\underline{u}_o} + (\Delta\underline{x}(t)^2/2)\frac{\partial^2 f(x)}{\partial x^2}|_{\underline{x}_o,\underline{u}_o} + ...)\\ + (B(\underline{x}_o(t)) + \Delta\underline{x}(t).\frac{\partial B(\underline{x})}{\partial x} + ...)|_{(\underline{x}_o,\underline{u}_o)}(\underline{u}_o(t) + \Delta\underline{u}(t)) \quad ;\end{aligned} \tag{3.4}$$

Neglecting terms involving second and higher order values of small variations, the linearized dynamic model is obtained about a given nominal trajectory as;

$$\Delta\dot{\underline{x}} = A(t)\Delta\underline{x} + B(t)\Delta\underline{u} \tag{3.5}$$

where;

$$A(t) = \left[\frac{\partial f(x)}{\partial x} + \frac{\partial B(x)}{\partial x}\cdot\underline{u}_o(t)\right]|_{\underline{x}_o(t),\underline{u}_o(t)} \tag{3.6.a}$$

$$B(t) = B(\underline{x}(t))|_{\underline{x}_o(t)} \tag{3.6.b}$$

If $\underline{x}_o(t) = \underline{x}_o$ and $\underline{u}_o(t) = \underline{u}_o$, where $\underline{x}_o$, $\underline{u}_o$ represent constant equilibrium states, then

$$A(t) = A \quad \text{and} \quad B(t) = B \tag{3.7}$$

are constant matrices.

The form (3.4) is used for linearization about a nominal trajectory for perturbation control and (3.7) is used for local linear dynamics and control law analysis. Linearization for robotic manipulators is performed for a nominal configuration, $\underline{x}_o = [\underline{\theta}, \underline{\delta}, \dot{\underline{\theta}}, \dot{\underline{\delta}}]_o$,

$$\underline{\theta} = \underline{\theta}_{nominal} \quad ; \quad \dot{\underline{\theta}} = \underline{0} \tag{3.8.a}$$

$$\underline{\delta} = 0. \quad ; \quad \dot{\underline{\delta}} = \underline{0} \tag{3.8.b}$$

and $\underline{u}_o$ is such that,

$$0 = \underline{f}(\underline{x}_o) + B(\underline{x}_o)\underline{u}_o \tag{3.9}$$

Thus, for a given nominal equilibrium state, $\underline{x}_o$, eqn. (3.9) gives the necessary nominal input, $\underline{u}_o$, for the state to be an equilibrium state, and evaluating eqn (3.6) about these values gives the linear dynamics of the manipulator.

3.2.2 Open Loop Eigenvalues Root Locus

The locus of open loop eigenvalues is studied as a function of second joint and payload point mass properties, for the manipulator parameters given in Appendix A. Figures (3.2.a and 3.2.b) show the variation of eigenvalues as the second joint and payload point masses of values {0., 1., 2., 4., 8., 16. kg.} (only point mass values, no mass moment of inertia) are introduced to the manipulator dynamic

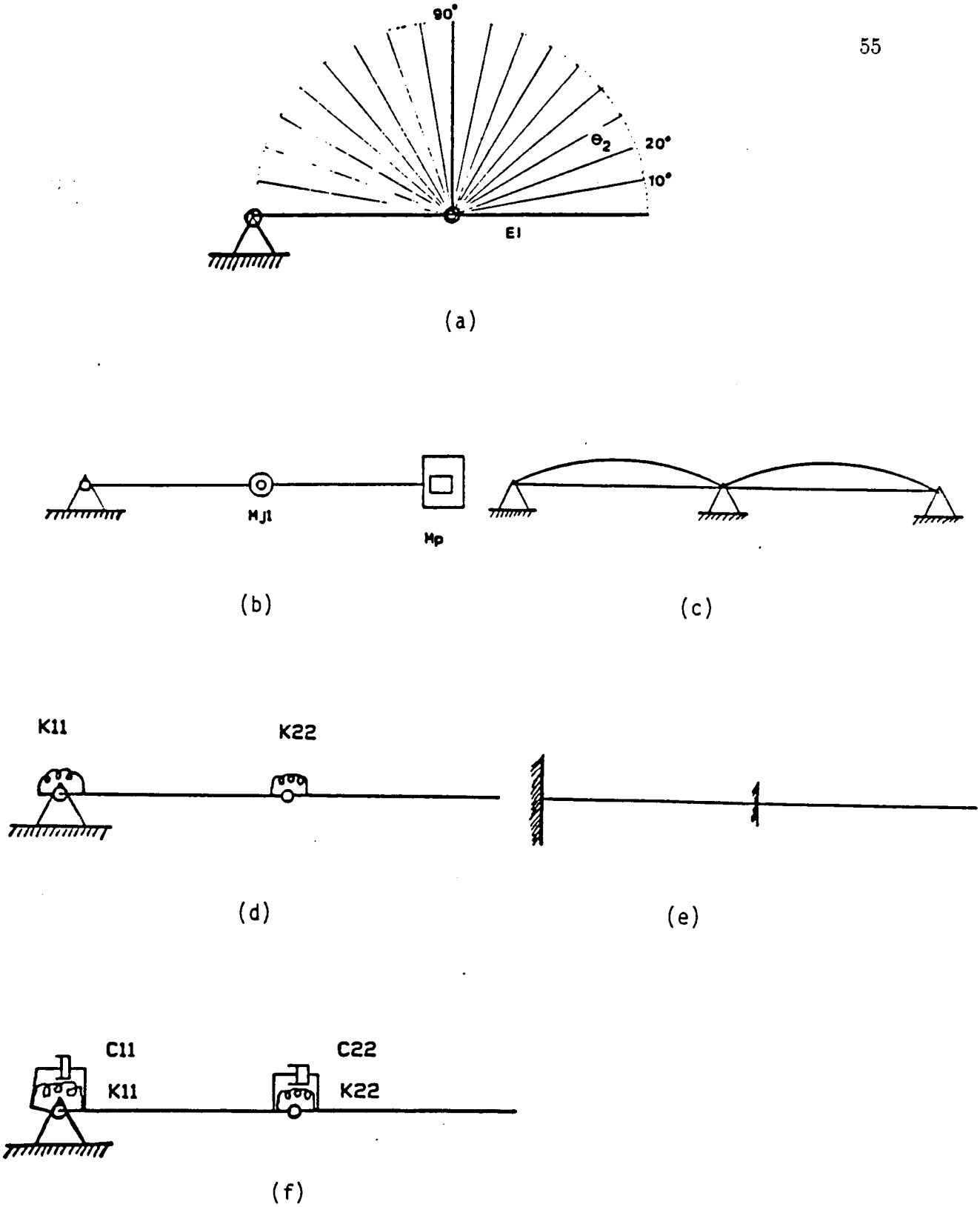


Fig.3.1 Eigenvalue analysis cases - root locus parameters are indicated on the figures

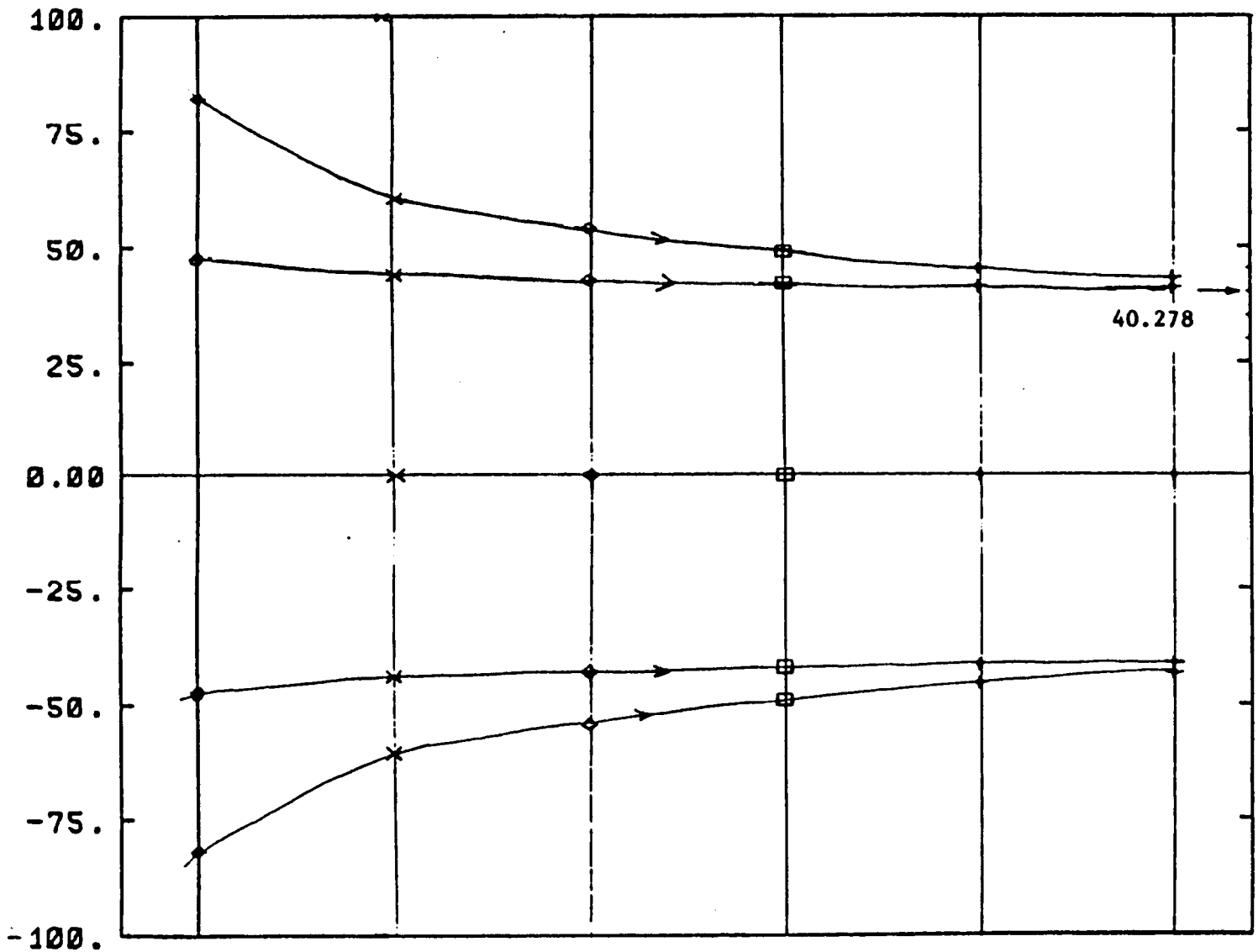


Fig.3.2 .a Open-loop eigenvalues (pin-pin beam is the limiting case)

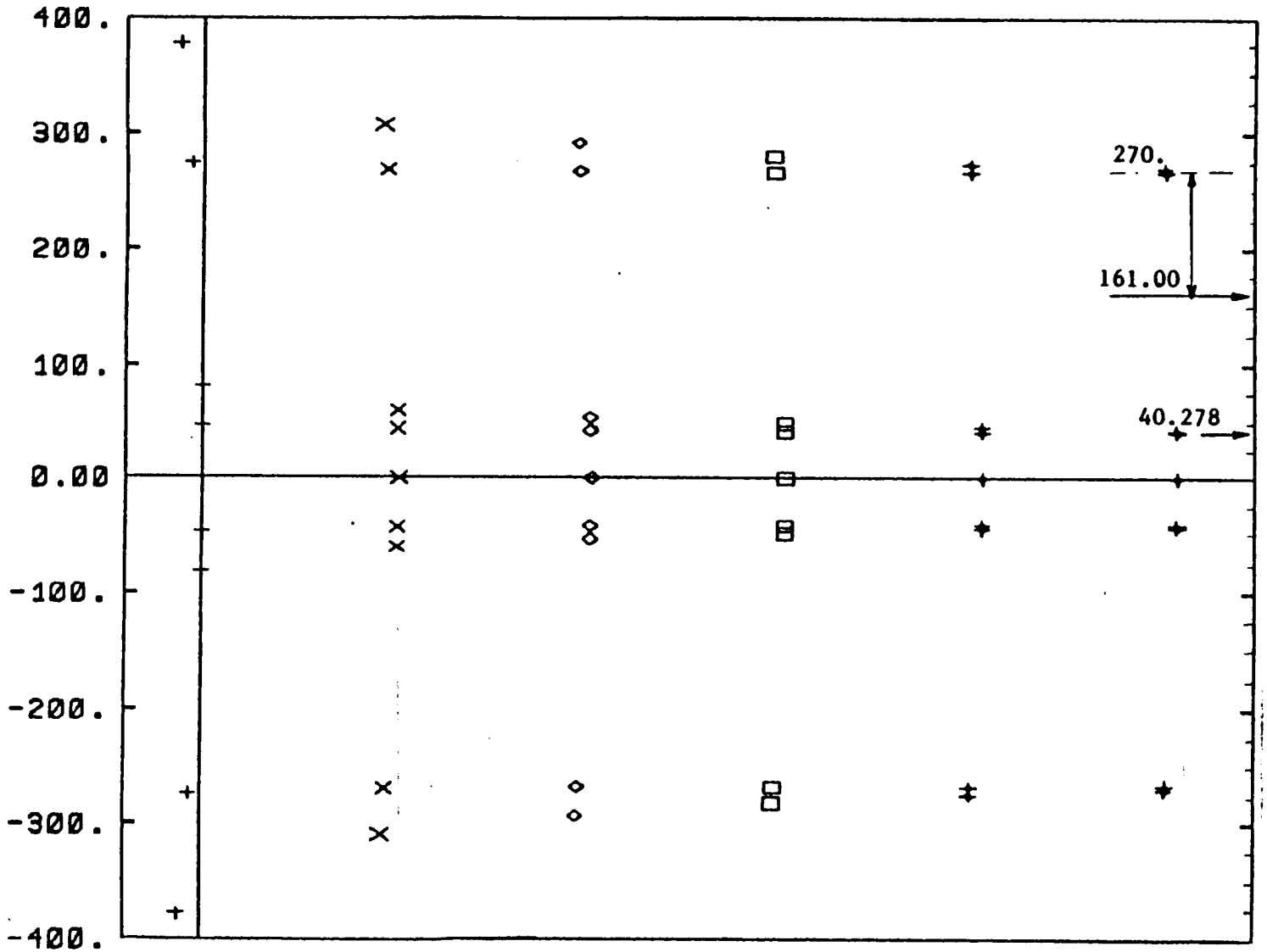


Fig.3.2.b Open-loop eigenvalues (pin-pin beam is the limiting case)

model. The eigenvalues merge to each other in pairs and converge to some limit values as second joint and payload masses increase. In the limit, system should behave like two independent pinned-pinned beams. Since beams have identical parameters, eigenvalues go to the same limits in pairs. When compared with the eigenvalues of the equivalent pinned-pinned beam, it is seen that the first pair of eigenvalues converges to the correct values accurately (our model which includes joint variables and four mode shapes (two modes for each link) converges to a value in the range of 41.15 - 43.27 rad/sec., and the equivalent ideal pinned-pinned beam first eigenvalue is 40.27 rad/sec.) However, the second pair does not converge to the correct value (our model converged to a value in the range of 266.09 - 269.71, and the corresponding pinned-pinned beam eigenvalue is 161.09 rad/sec). This is due to the fact that a finite dimensional assumed modes model can not predict the dynamic behavior for all range of parameters. At the extereme cases, the model loses accuracy in higher modes due to the truncated model order and the assumed mode shapes which may no longer be accurate under these new conditions. Due to the free rotation capabilities of the joints, the rigid body mode is still preserved, and pinned-pinned modes are imposed on that. Therefore, zero eigenvalues associated with the rigid body mode of each joint motion is retained (Fig. 3.2.b).

The limiting case of high joint position feedback gains at both joints should make the two link arm behave like a single clamped beam with a discontinuous stiffness in the middle (joint 2 stiffness, Fig.3.1.c). Open loop eigenvalues are plotted for joint 1 and 2 position feedback values of $\{10^n; n = 0., 1., 2., 3., 4., 5., 6.\}$. Fig.3.3 shows that the first 4 modes converge to the corresponding limit values very accurately (Table 3.1).

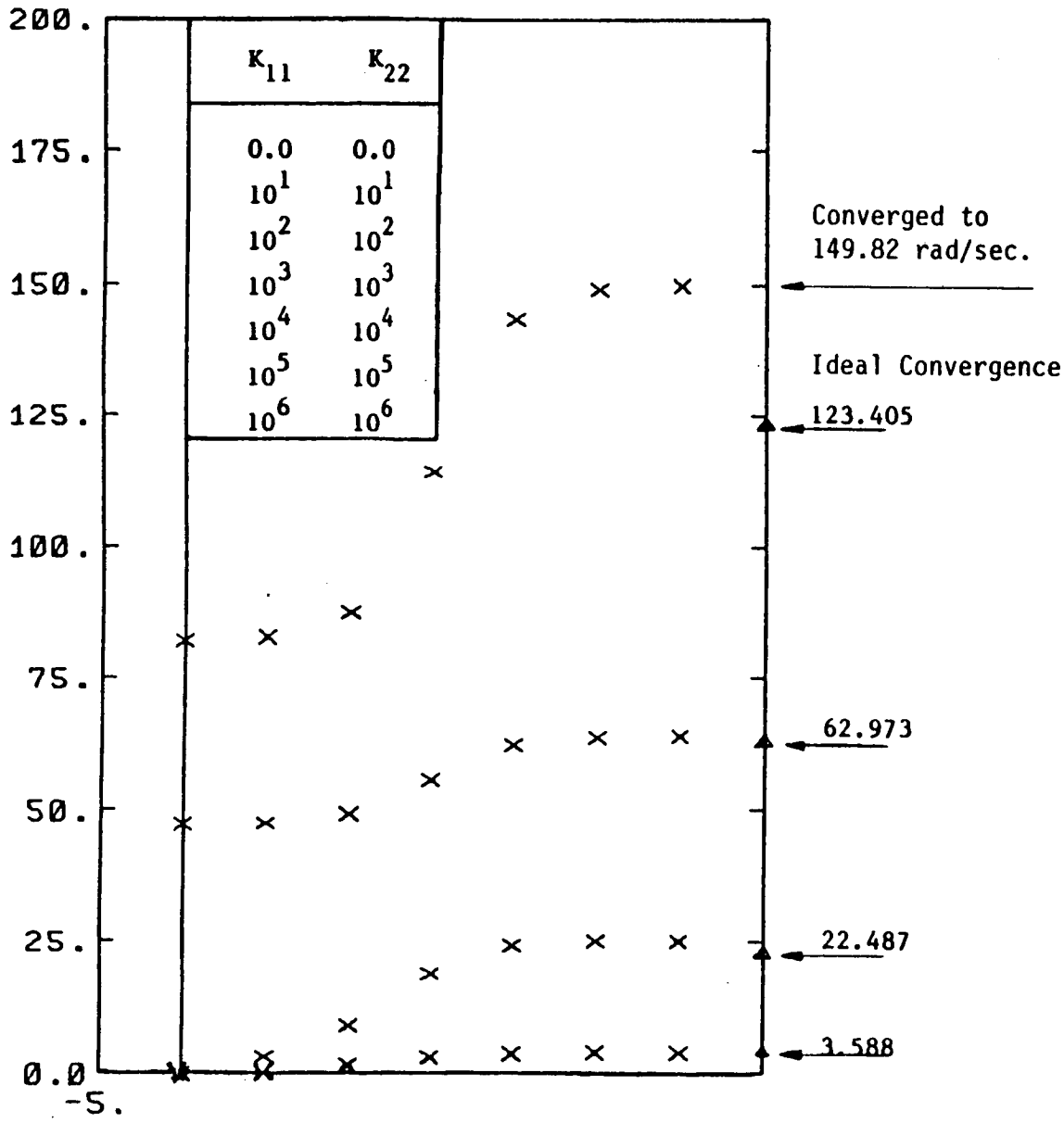


Fig.3.3 Single clamped-free beam limit behavior

Values Converged(rad/sec)	Ideal Values(rad/sec)
3.688	3.588
25.17	22.48
63.77	62.97
149.82	123.40

Table 3.1 Asymptotic behavior of eigenvalues of assumed modes model under high position feedback gain

The fifth and sixth modes did not show convergence to any values. As joint stiffness increased, they kept increasing too. The eigenvalues were 1834.0 and $3528.4 rad./sec$ for $k_{11} = k_{22} = 10^5$, and 5747.5 and $11104. rad/sec.$ for 10^6, clearly showing no convergence. However, that is not wrong nor a surprise. Recall that the model has two degrees of freedom for joint variables (the rigid body modes) and four four degrees of freedom for flexible motions. Therefore, it is expected that the two mode eigenvalues, associated primarily with the joint motions, will increase indefinitely (to infinity) as the position feedback gains increase. In other words, at limit the six modes of the finite dimensional assumed modes model will not converge to the first six modes of the corresponding limit system, as discussed above.

Fig (3.4.a) shows the variation of the eigenvalues as a function of flexural rigidity of the links. Figures (3.4.b,c) give closer look at the root locus as a function of configuration for $EI_1 = EI_2 = 533.33 Nt.m^2$ case. The noteworthy results here are as follows.

1. All eigenvalues have the closest locations to each other in pairs for $\theta_2 = 90^o$ degrees. This makes sense, for at this configuration the dynamic coupling between links is minimum, and dynamic behavior is converging towards the

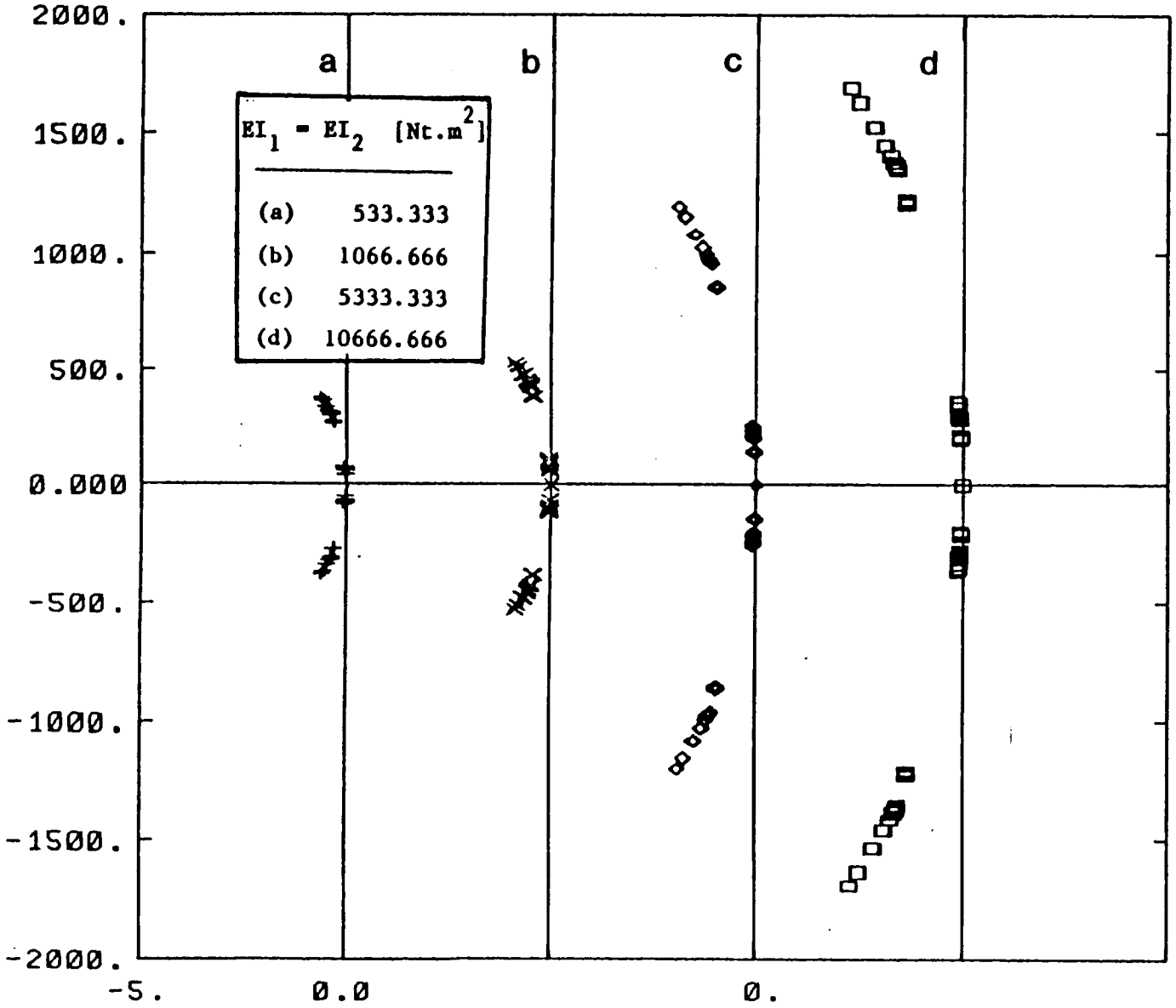


Fig.3.4.a Open-loop eigenvalues as function of configuration (θ_2) and stiffness, ($EI_1 = EI_2$)

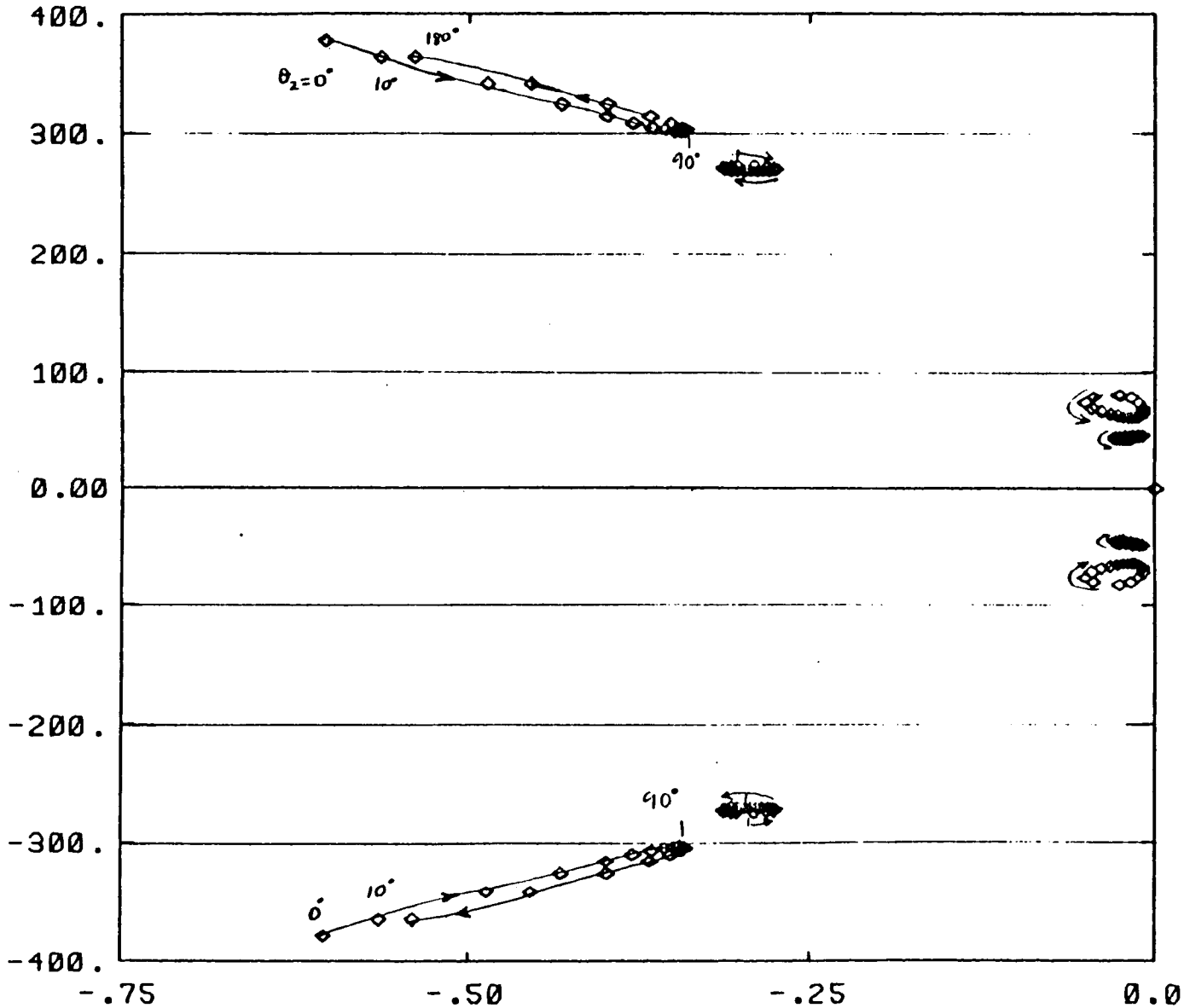


Fig. 3.4.b Closer look at the Fig. 3.4.a

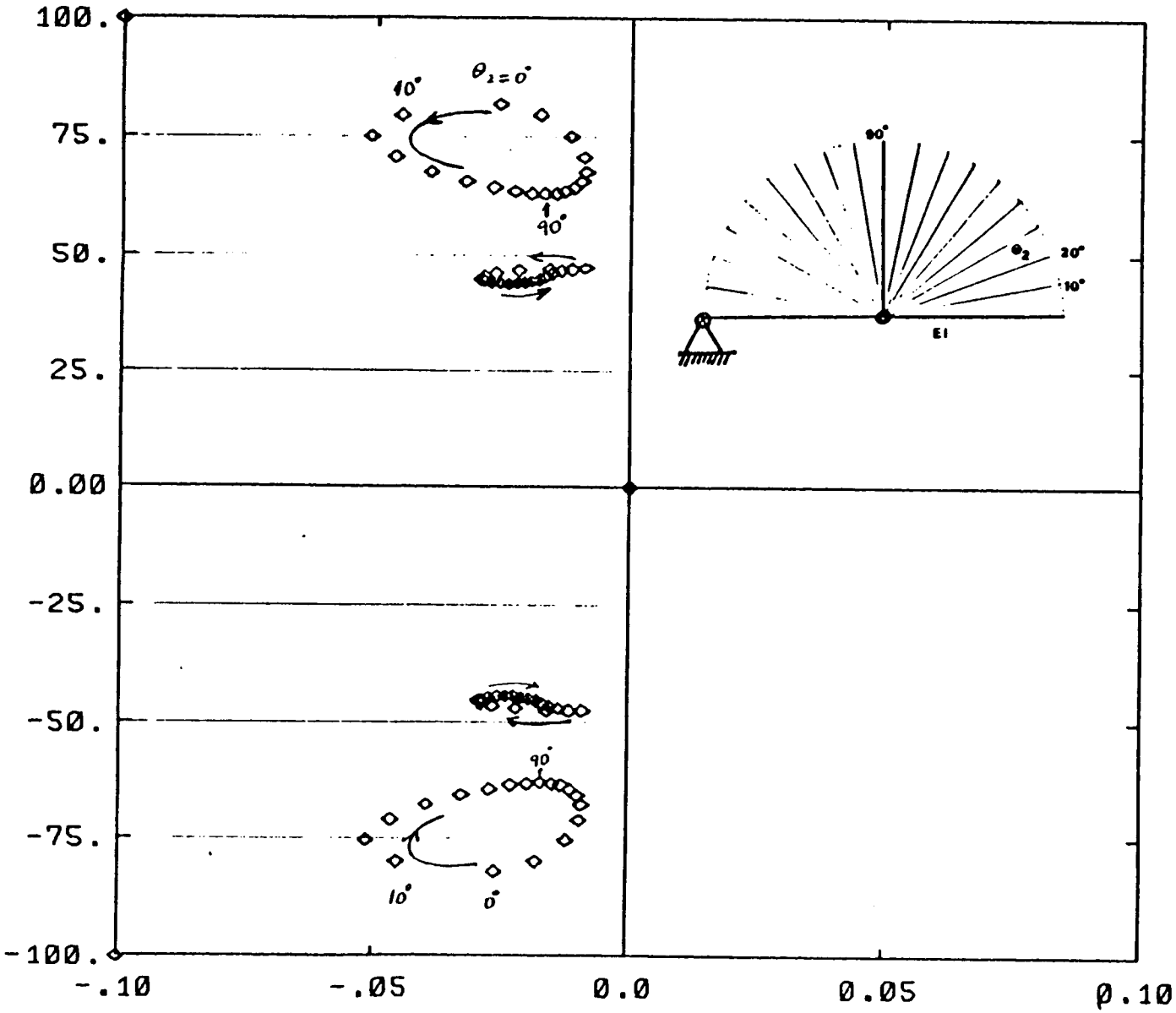


Fig. 3.4.c Closer look at the Fig. 3.4.b

dynamic behavior of a single link (Fig.3.4.b).

2. The largest variation in the location of eigenvalues occurs again for $\theta_2 = 90^o$ (Fig 3.4.c).

3. In particular, the second and fourth flexible modes are most sensitive to configuration changes. The order of 25 % variations of the nominal values was observed (Fig.3.4.c).

3.3. Closed Loop Dynamics Under Joint Variable Feedback: Limitations and Sensitivity

3.3.1. Previous work and results based on frequency domain approach

Joint variable feedback (position and velocity) is very common in robot motion control. Before attempting to develop control algorithms which use flexible state feedback as well as joint variables, one should start by determining the level of performance which can be achieved by the well known joint variable feedback control algorithms when applied to flexible manipulators. Specifically, the following questions will be studied: 1. What is the upper limit of performance that a linear joint variable feedback control law can achieve when applied to flexible manipulators? This limit is imposed by the arm flexibility on the closed loop system performance. 2. How do the closed loop eigenvalues vary as a function of joint variable feedback gains (sensitivity)?

Book [A20, A27] studied these questions using a frequency domain model of a two-link, two-joint manipulator. Transfer matrices are used to model the linear

dynamics of the distributed-lumped parameter system. The approach is limited to linear dynamic analysis. Results based on such model would be free of any model order truncation errors and may be used as a reference for other approaches. It is of interest to determine how the assumed modes method answers the same questions and to compare the results of the two different approaches. Of course, errors due to model order truncation are inherent in the results of the assumed modes model. Such a comparision will also serve the purpose of determining how many assumed modes are *accurate enough.*

The basic results can be explained by the Fig 3.5. If the beam were rigid, natural frequency of the system under position feedback regulation would be,

$$w_n = \sqrt{k/J_0}\ rad/sec. \tag{3.10}$$

where; $J_0 = (1/3)\rho A l^3$ mass moment of inertia about the joint, and as $k \rightarrow \infty$, $w_n \rightarrow \infty$.

However when the arm is flexible, this is no longer true and the dominant eigenvalue is upper bounded by

$$w_n = (1.875)^2 \sqrt{EI/(\rho A l^4)}\ rad/sec. \tag{3.11}$$

This is the simplest explanation of the limitations imposed by the arm flexibility on joint variable feedback controller performance. Book has further studied the two-beam,two-joint cases. The variation of the dominant eigenvalues (root locus) as a function of joint velocity feedback is found to be of the form shown in fig 3.6. Notice that for low servo stiffness, two complex conjugate eigenvalues break-in to the real axis,and go in the opposite directions. Further increaseing c_{22} results in

a reduced damping ratio. At some point, one more eigenvalue appears, coming from $-\infty$, and meets the dominant eigenvalue and then breaks-away from the real axis, eventually approaching point D on the imaginary axis (fig 3.6). As servo stiffness increases ((b) and (c) locii), increasing joint damping is no longer able to achieve desired damping ratios. This can be explained physically as follows: as servo position feedback increases, the arm joint gets stiffer and stiffer, making the energy dissipation more difficult. If a joint variable feedback controller is to be used for a flexible manipulator, Book suggested that the closed loop dominant eigenvalues larger than 1/2 of the lowest frequency of the arm should not be attempted. The lowest frequency of the arm is defined as the first natural frequency of the arm when all joints are clamped and links are extended.

The eigenvalue problem of the transfer matrix model has an infinite number of solutions since the model is infinite dimensional. In the root locus analysis, only the eigenvalues within a finite region of the s-plane are numerically calculated. The source of the additional eigenvalue, which enters the into the studied region at some value of feedback gains (point C in figure 3.6), is not determined. However, this phenomenon is explained [A20] by an analogy with a lumped parameter model as shown in Fig. 3.7. Fig. 3.7 shows the root locus of the eigenvalues as a function of the damping coefficient c_s.

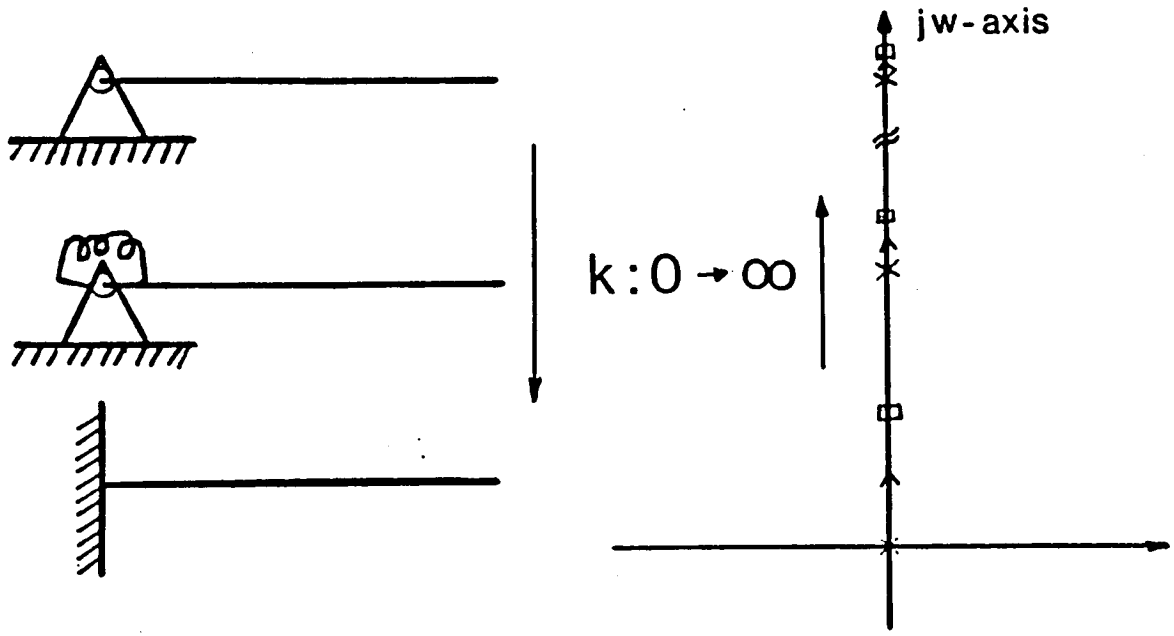


Fig.3.5 Illustration of limitations imposed on the closed loop servo control system performance due to arm flexibility

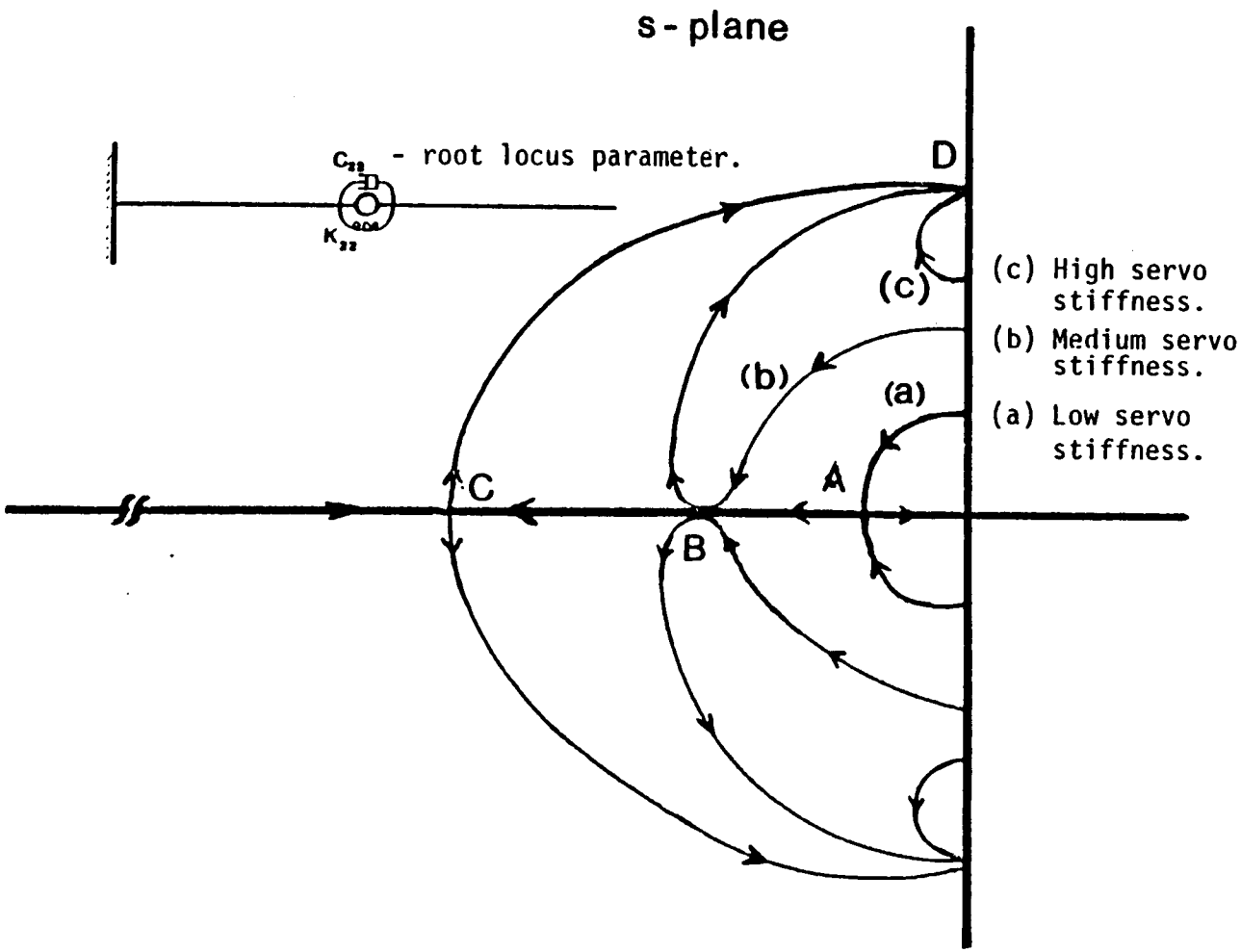


Fig.3.6. Limitations of servo control due to flexibility for two-beam,one-joint controlled case.

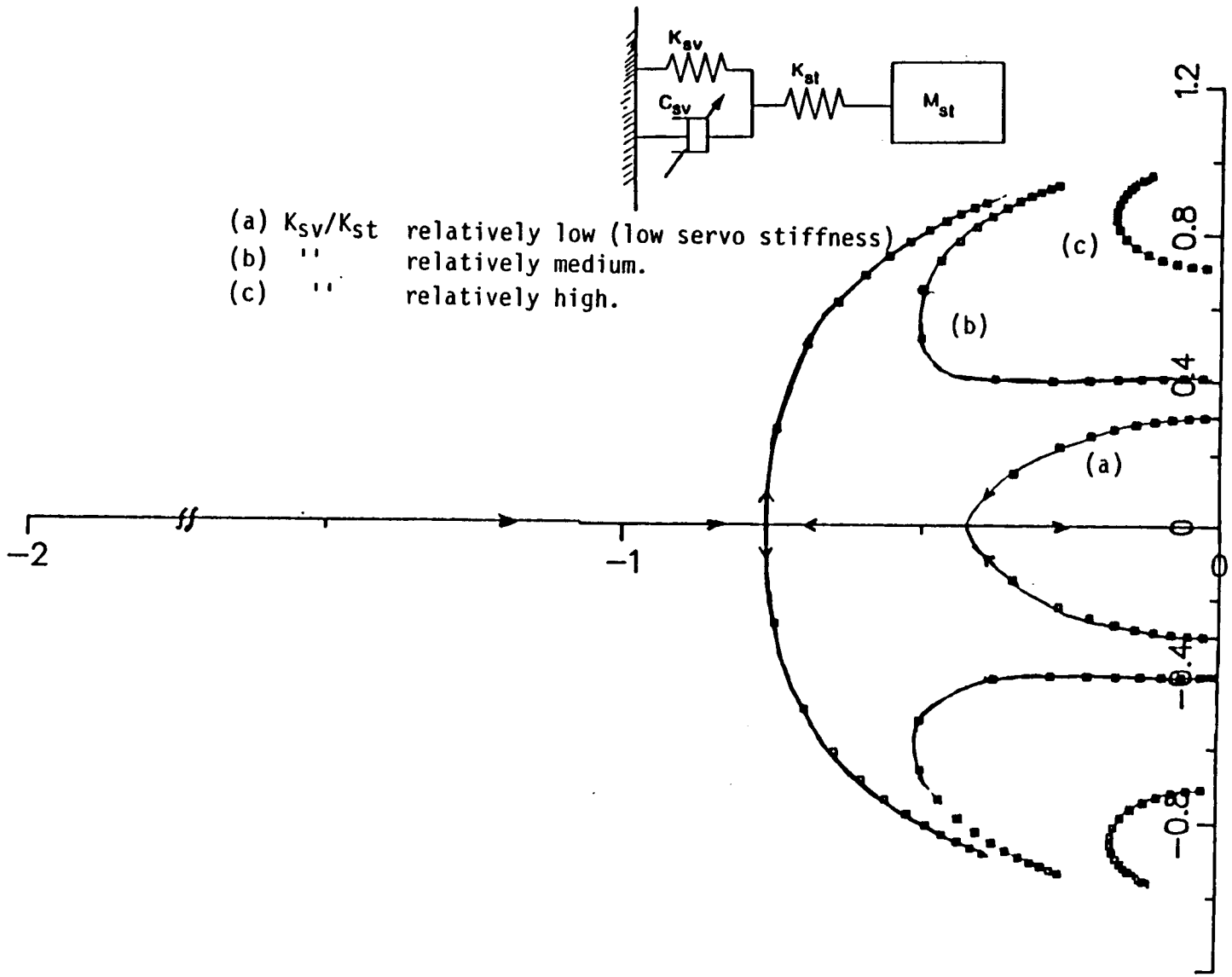


Fig.3.7 Analogy: Lumped parameter model exhibiting the same root locus of eigenvalues as function of servo gains.

3.3.2. Results of assumed modes based model

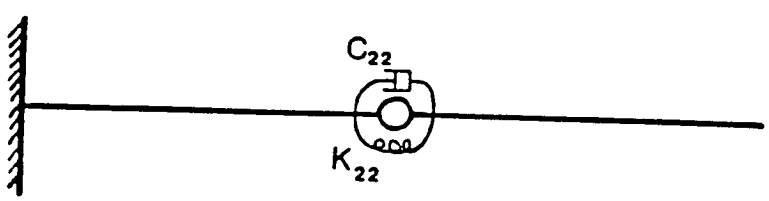
The assumed modes model predictions about the limitations of joint variable feedback control will have inherent errors introduced by the model order truncation. There is always a finite number of eigenvalue solutions. An eigenvalue appearing from the $-\infty$ direction of real axis cannot happen, or the original location of such an eigenvalue must be one of the finite number of eigenvalues already exist. The transfer matrix based analysis in the frequency domain left the question of where the additional eigenvalue came from unanswered. In fact, it did not need an answer, for the system was infinite dimensional and only those eigenvalues within some finite region of s-plane were numerically solved in the frequency domain. Nonetheless, for an assumed modes model to be acceptable, it should be able to predict the important characteristics of the system well, such as the joint feedback control limitations.

Root locus analysis will be used on the linearized model of the assumed modes method. The system model is twelfth order: one joint angle, and two flexible modes for each link. The objective is to determine a) how well this model predicts the joint variable feedback control limitations, and b) the root locus sensitivity as a function of feedback gains.

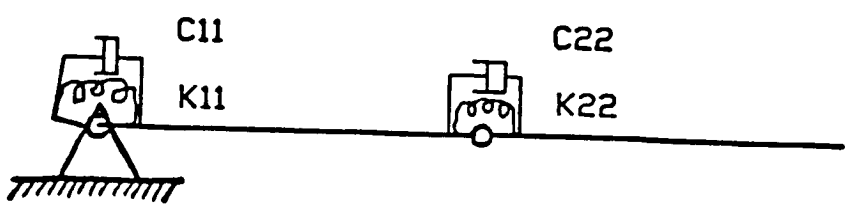
An analysis is done on cases (a) and (b) of figure 3.8. Here the results of case (a) will be discussed for its clarity and simplicity over case (b).

Figures 3.9.e, 3.10.f, and 3.11.g show the root locus of the dominant closed loop eigenvalues as a function of joint velocity feedback gain for low, medium and high servo stiffnesses. Cleary, it is seen that our truncated model (one rigid body, two flexible modes for each link) predicts the limitations of joint variable feedback

control laws due to flexibilty very well. Since the assumed modes model has only a finite number of eigenvalues, the eigenvalue which meets with one of dominant eigenvalues on the real axis (in fig.3.10.f, and 3.11.g, this eigenvalue stays on the real axis) must be associated with one of the modes. When the root locus of all the eigenvalues is checked, it is seen that the needed eigenvalue, to exhibit the phenomenon of fig 3.5, is provided by one of the flexible modes (Fig. 3.9.a,b, 3.10.a,b, 3.11.a,b,c). Physically this means joint velocity feedback alone can introduce very large damping rations to some of the flexible modes.



(a)



(b)

Fig.3.8 Closed-loop eigenvalue root locus as function of servo gains - Assumed mode model results.

3.4. Summary of Results and Conclusion

The truncated assumed modes model predicts the limitations of joint variable feedback control very well, but it may be doing so at the expense of losing accuracy in predicting the higher mode behavior. In other words, the break-in of some higher mode eigenvalues to the real axis is questionable. Also, given the results of open loop eigenvalue analysis, where accuracy is lost at higher modes for some limiting cases, it seems that the break-in of some eigenvalues to real axis may indeed be a mathematical *parasitic solution*, which does not exist in a real system. Reported results of the transfer matrix approach did not determine the source of the eigenvalue in question. The remaining questions to be answered are as follows: 1. How would the root locus behavior be if the assumed modes model included 2,4,6,8,10,....modes? 2. If this is a mathematical *parasitic solution*, that resulted from the truncation of model order, and is not a property of the dynamics of the real system, what is the error introduced to the behavior of other eigenvalues? How many modes are accurate enough to guarantee a desired accuracy for a given number of modes under partial state feedback?

Use of both frequency domain transfer matrix and assumed modes model in the analysis of this problem may prove to be effective in resolving the outstanding questions.

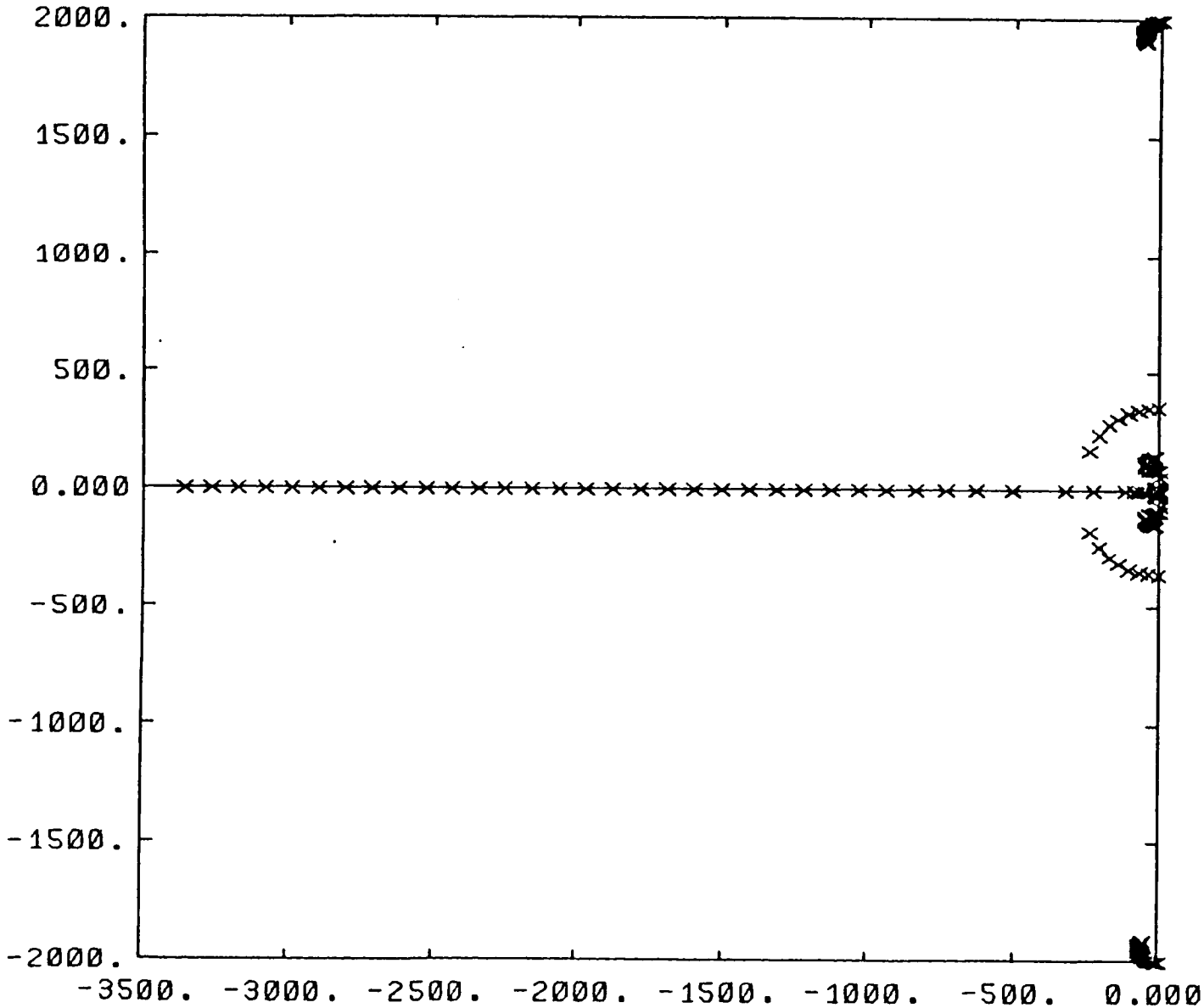


Fig.3.9.a Root locus of closed loop eigenvalues - low servo stiffness case

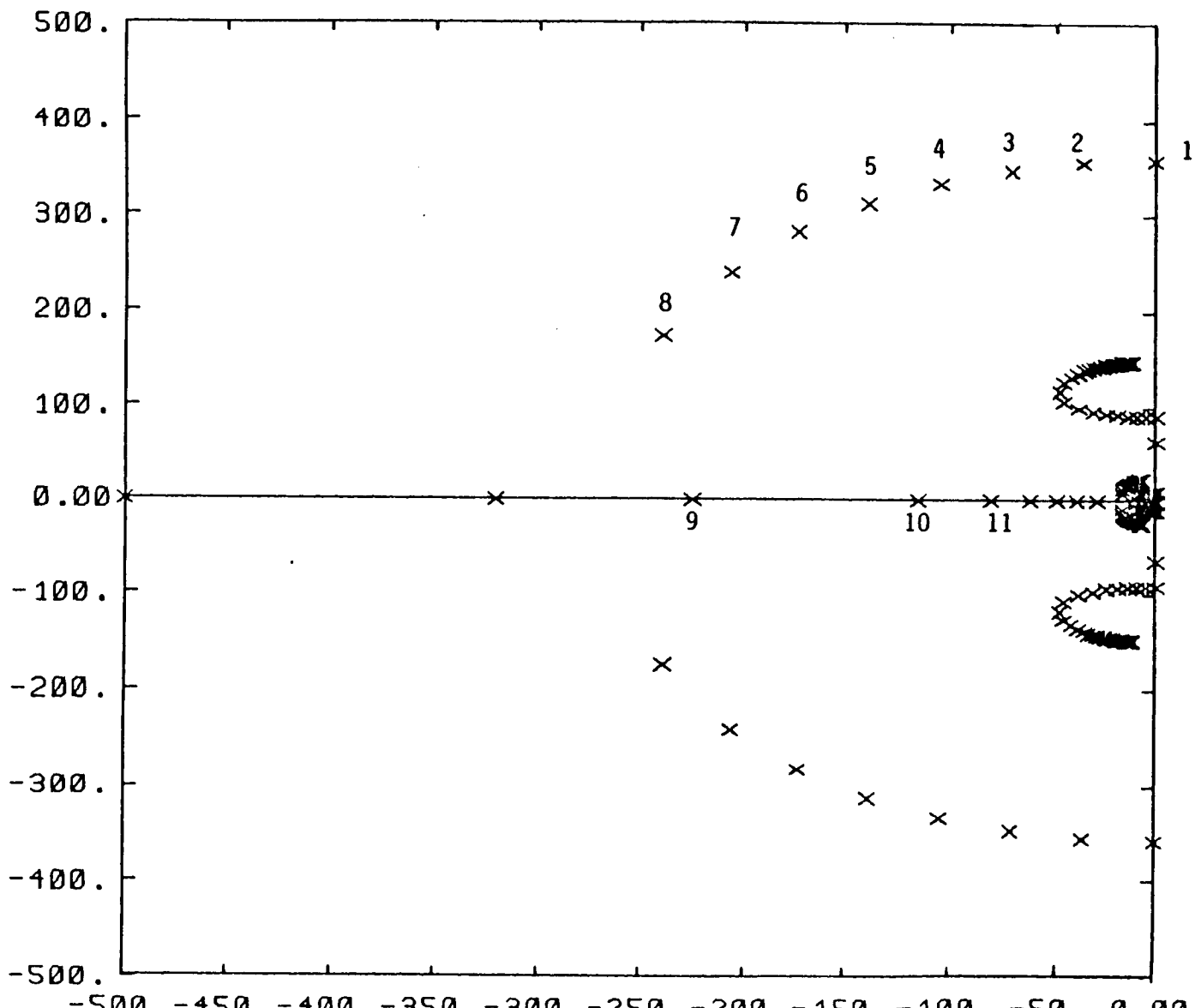


Fig.3.9.b Root locus of closed loop eigenvalues - low servo stiffness case closer look of Fig.3.9.a.

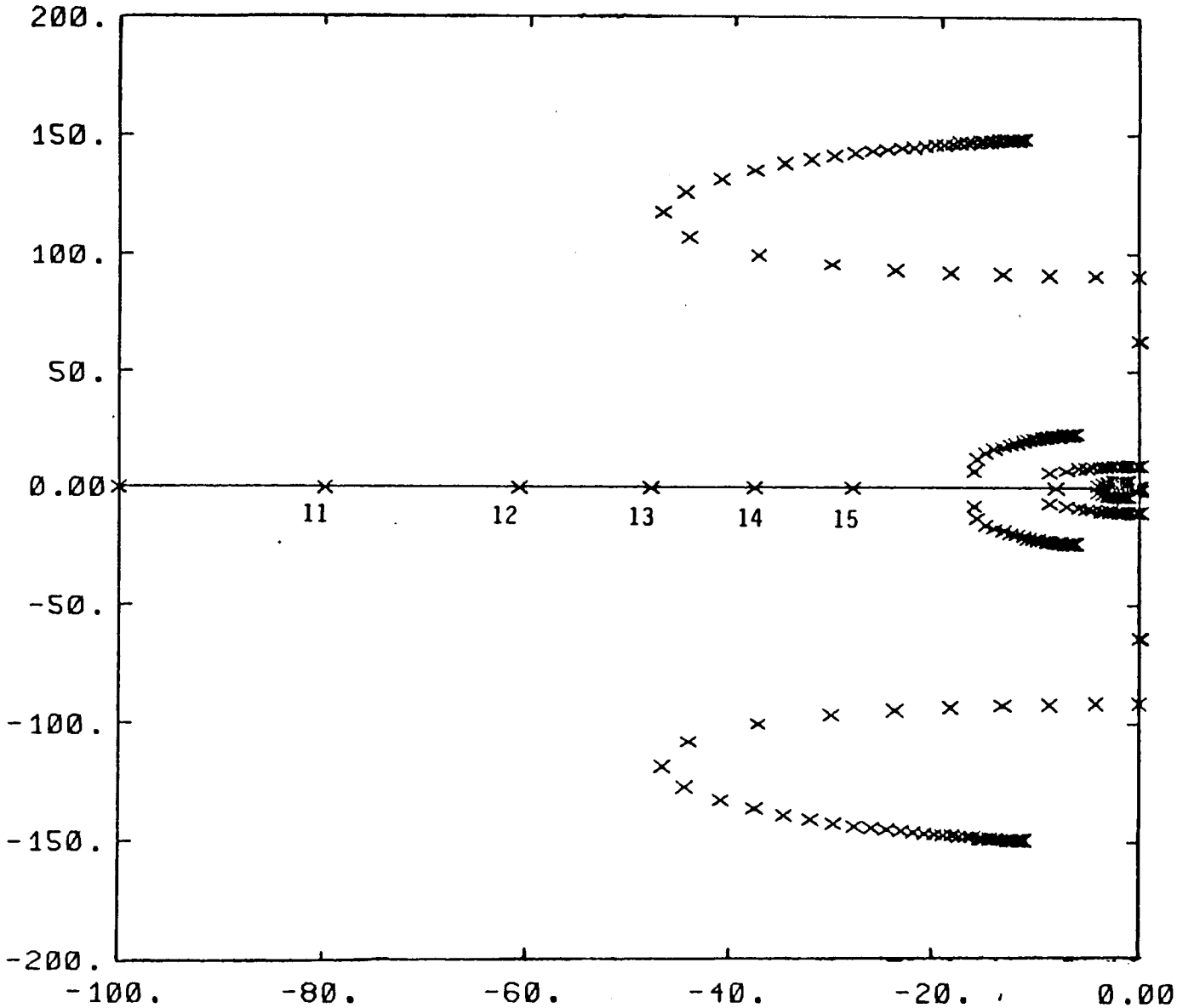


Fig.3.9.c Root locus of closed loop eigenvalues - low servo stiffness case closer look of Fig.3.9.b.

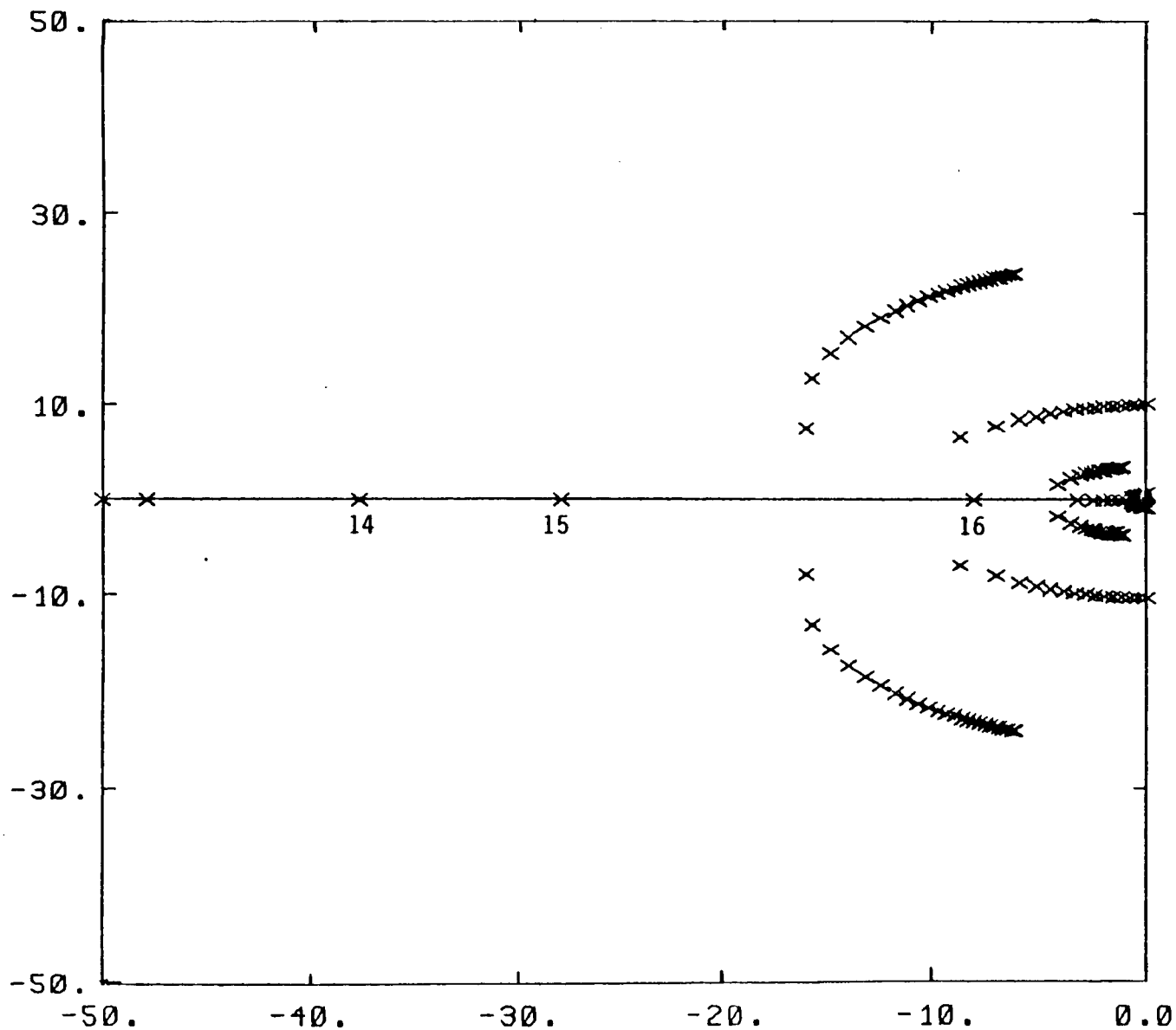


Fig.3.9.d Root locus of closed loop eigenvalues - low servo stiffness case closer look of Fig.3.9.c.

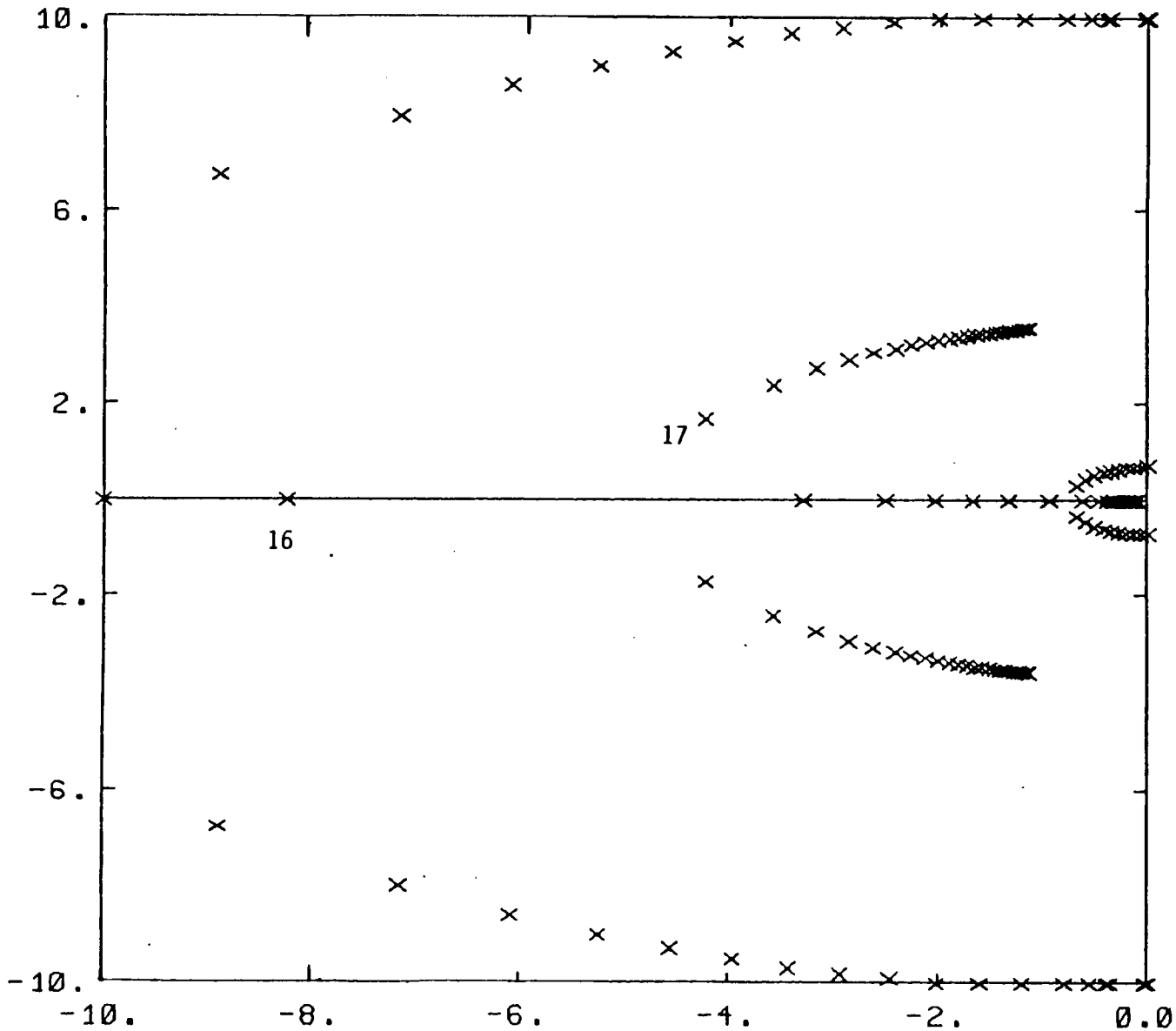


Fig.3.9.e Root locus of closed loop eigenvalues - low servo stiffness case closer look of Fig.3.9.d.

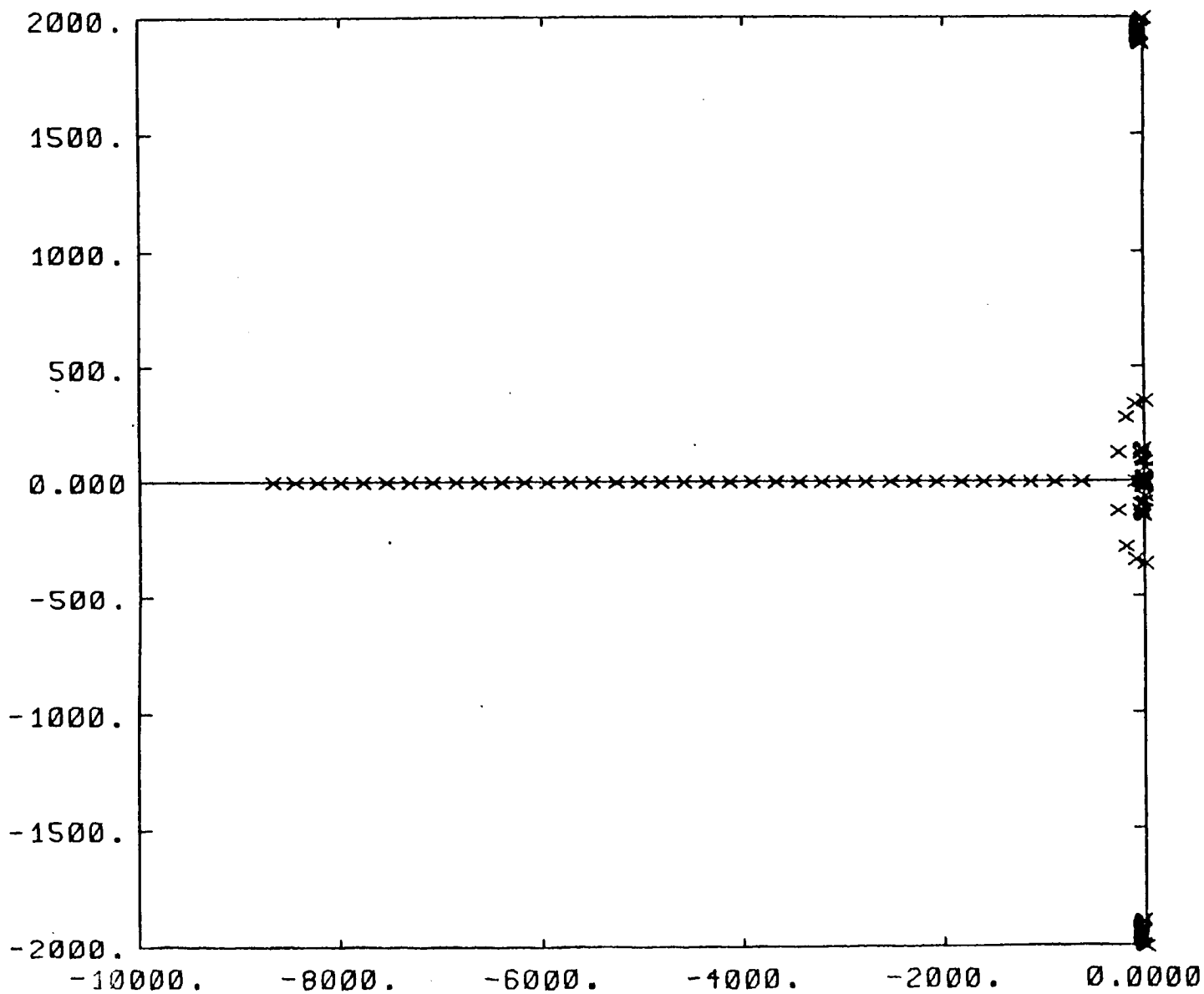


Fig.3.10.a Root locus of closed loop eigenvalues - medium servo stiffness case

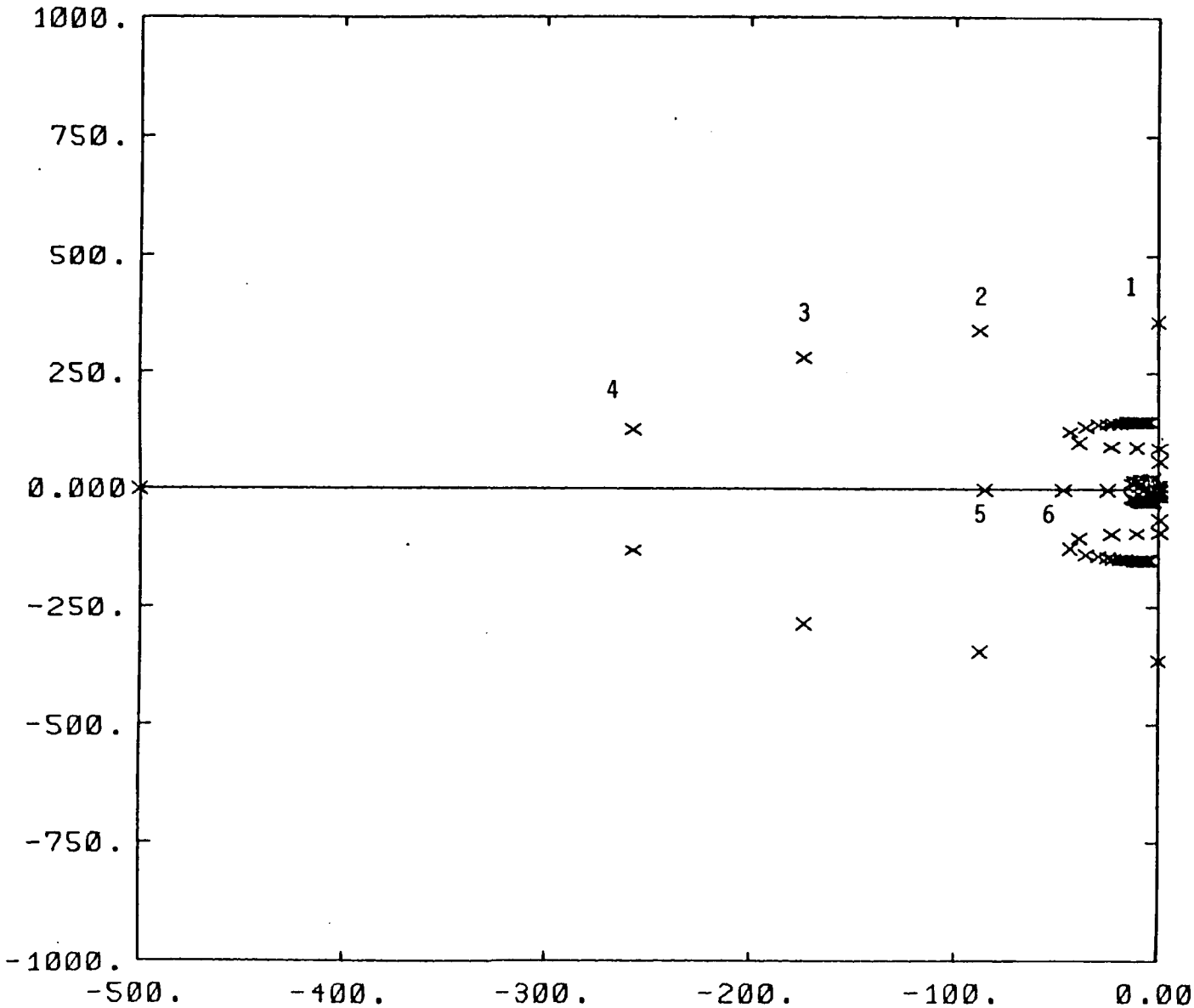


Fig.3.10.b Root locus of closed loop eigenvalues - medium servo stiffness case, closer look of Fig.3.10.a.

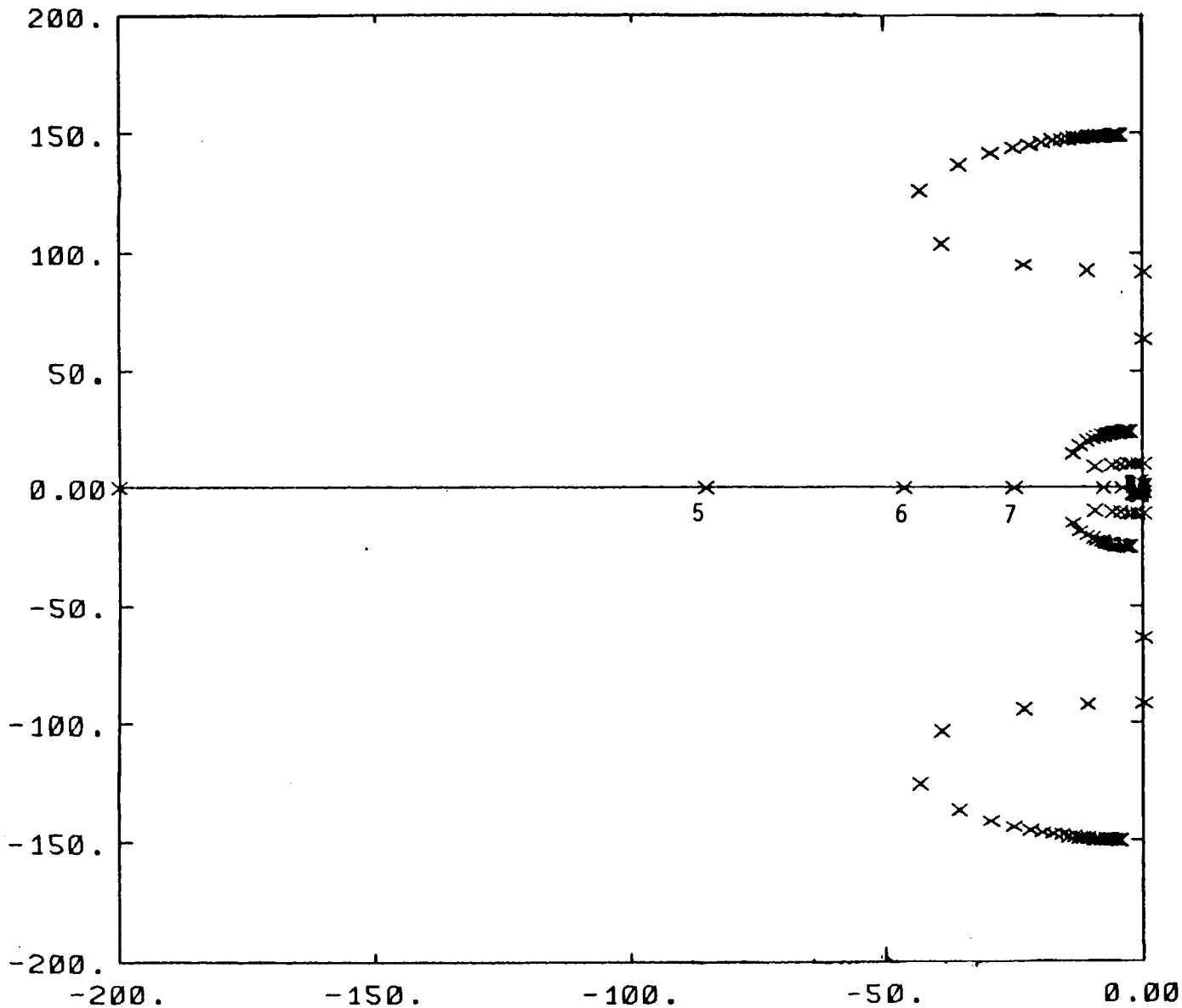


Fig.3.10.c Root locus of closed loop eigenvalues - medium servo stiffness case closer look of Fig.3.10.b.

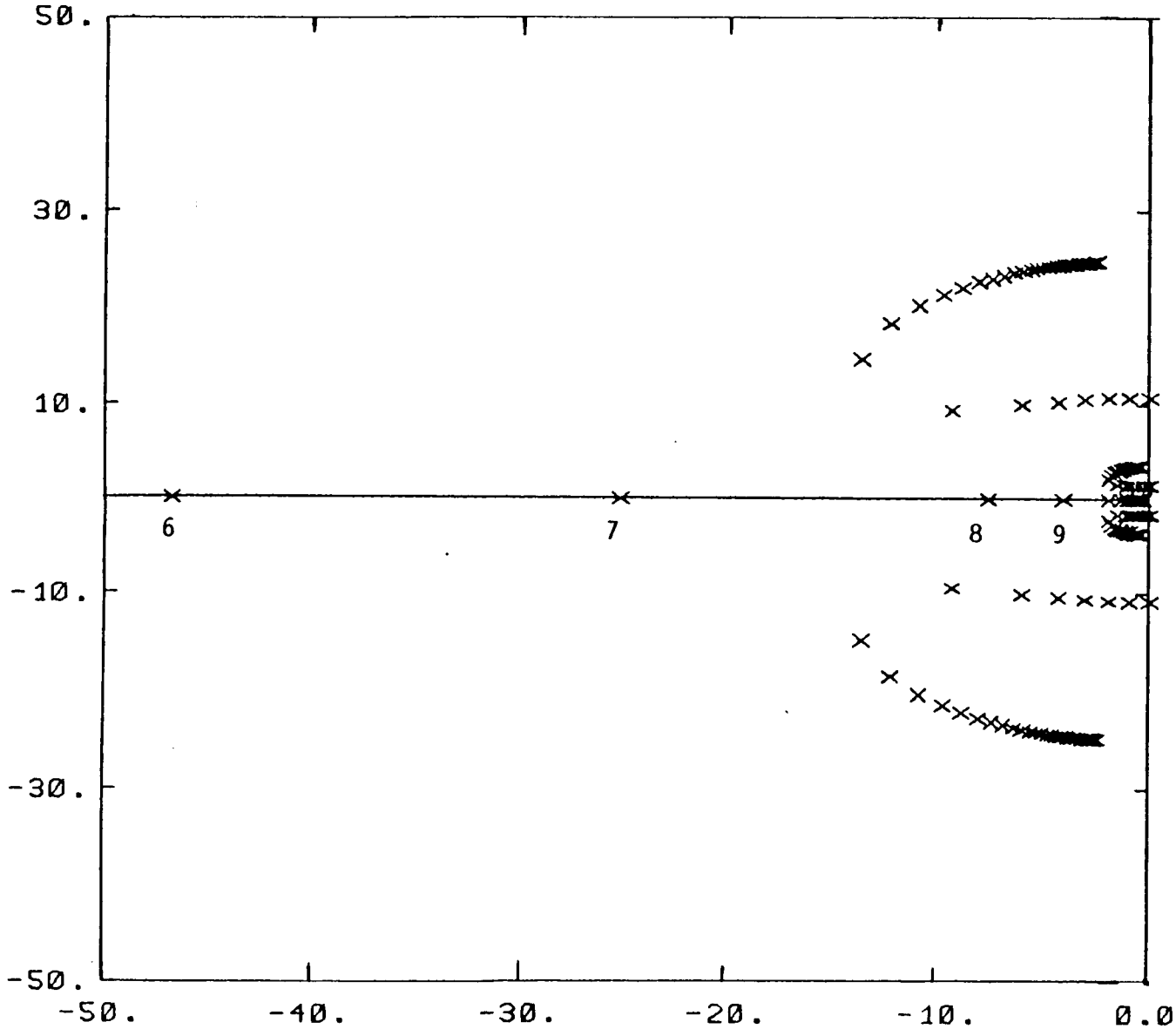


Fig.3.10.d Root locus of closed loop eigenvalues - medium servo stiffness case closer look of Fig.3.10.c.

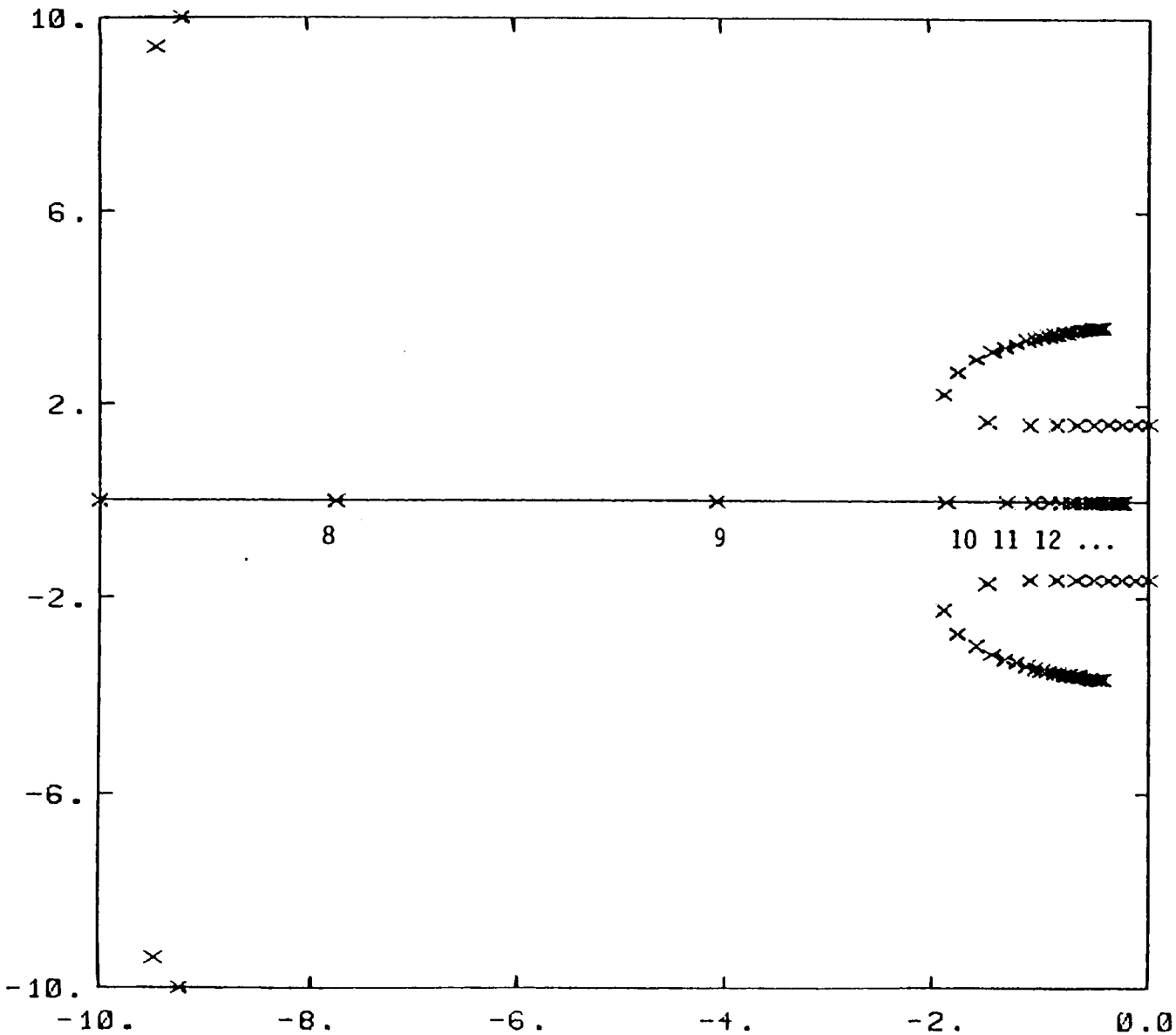


Fig.3.10.e Root locus of closed loop eigenvalues - medium servo stiffness case closer look of Fig.3.10.d.

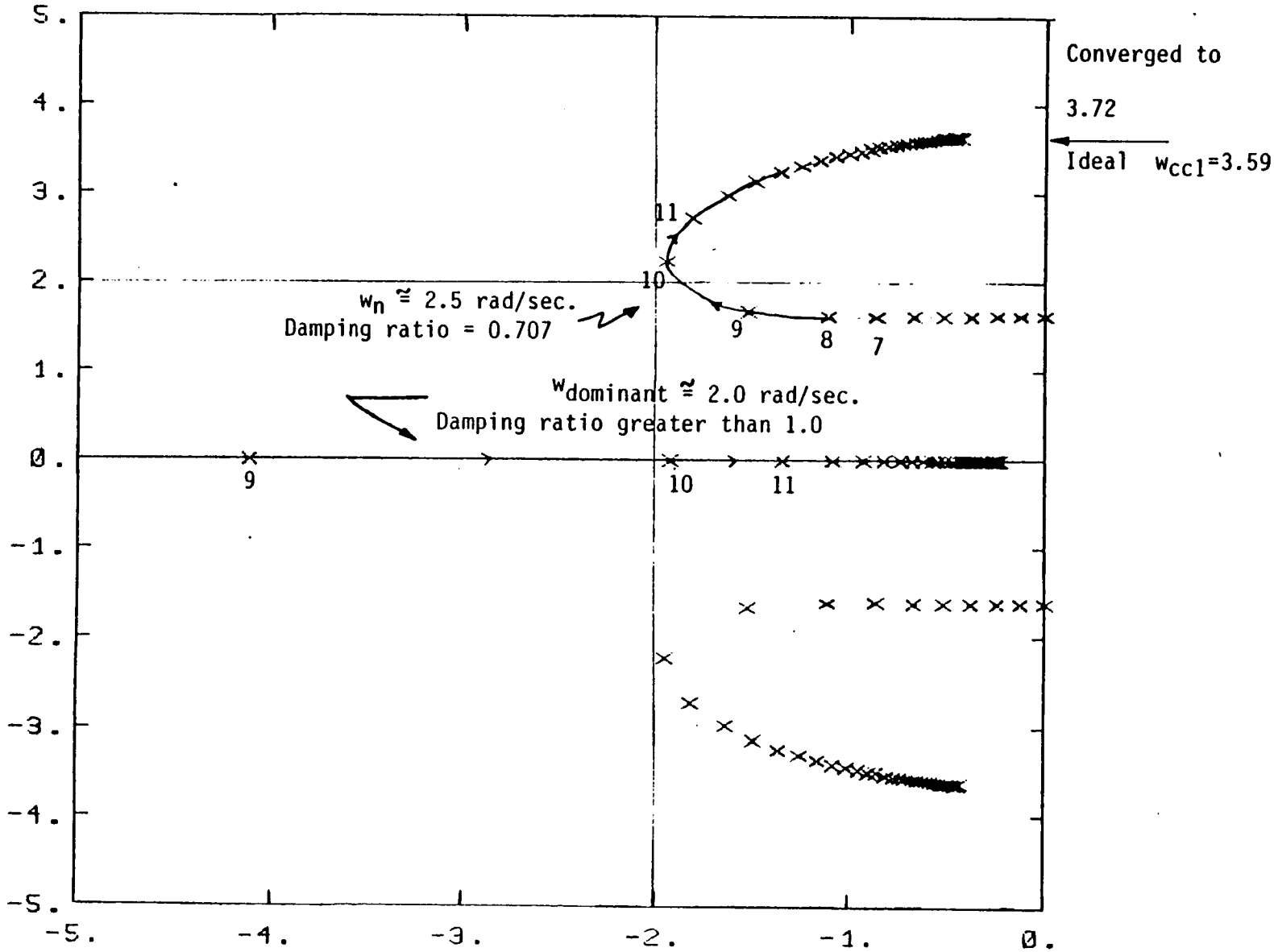


Fig.3.10.f Root locus of closed loop eigenvalues - medium servo stiffness case closer look of Fig.3.10.e.

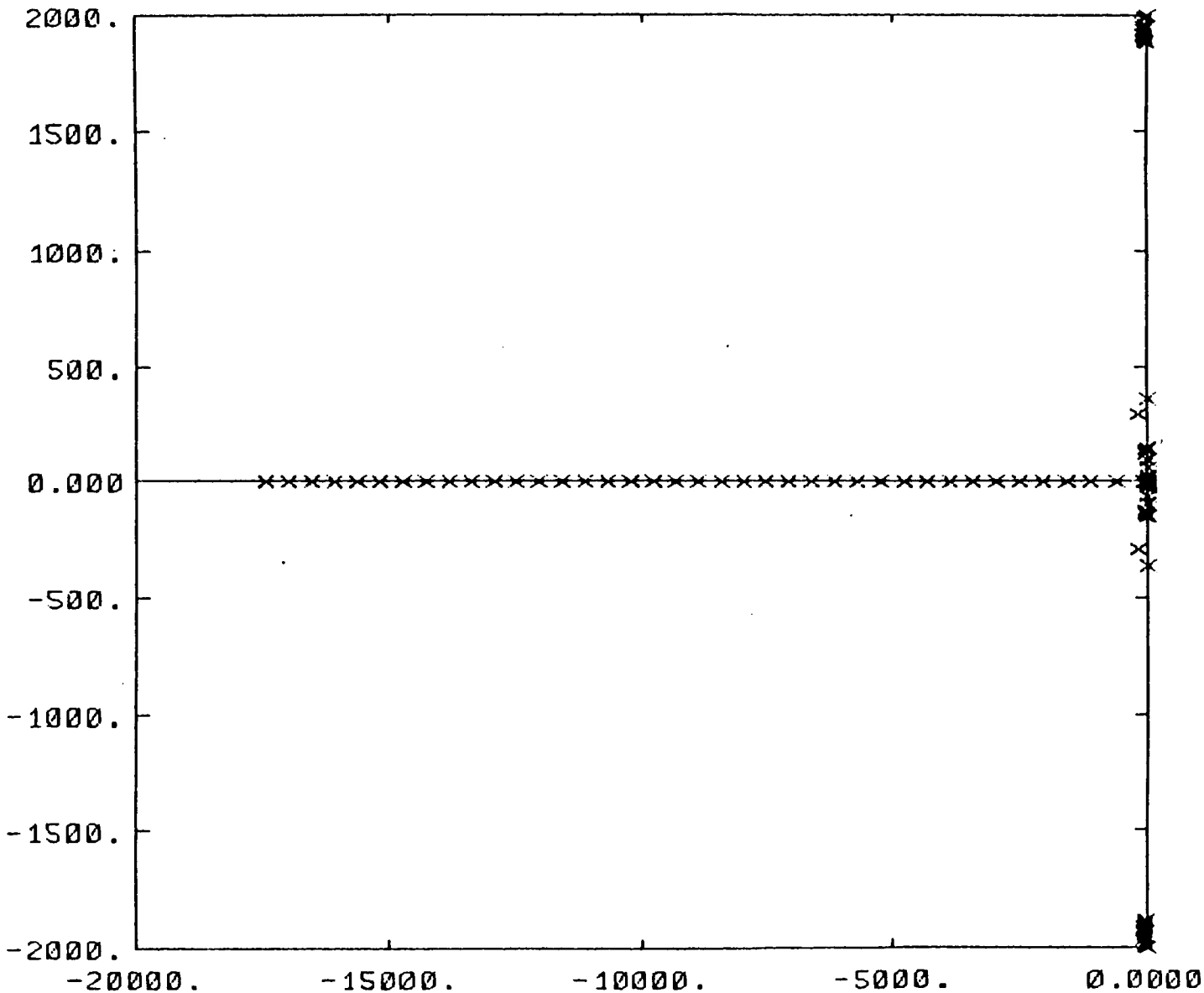


Fig.3.11.a Root locus of closed loop eigenvalues - high servo stiffness case

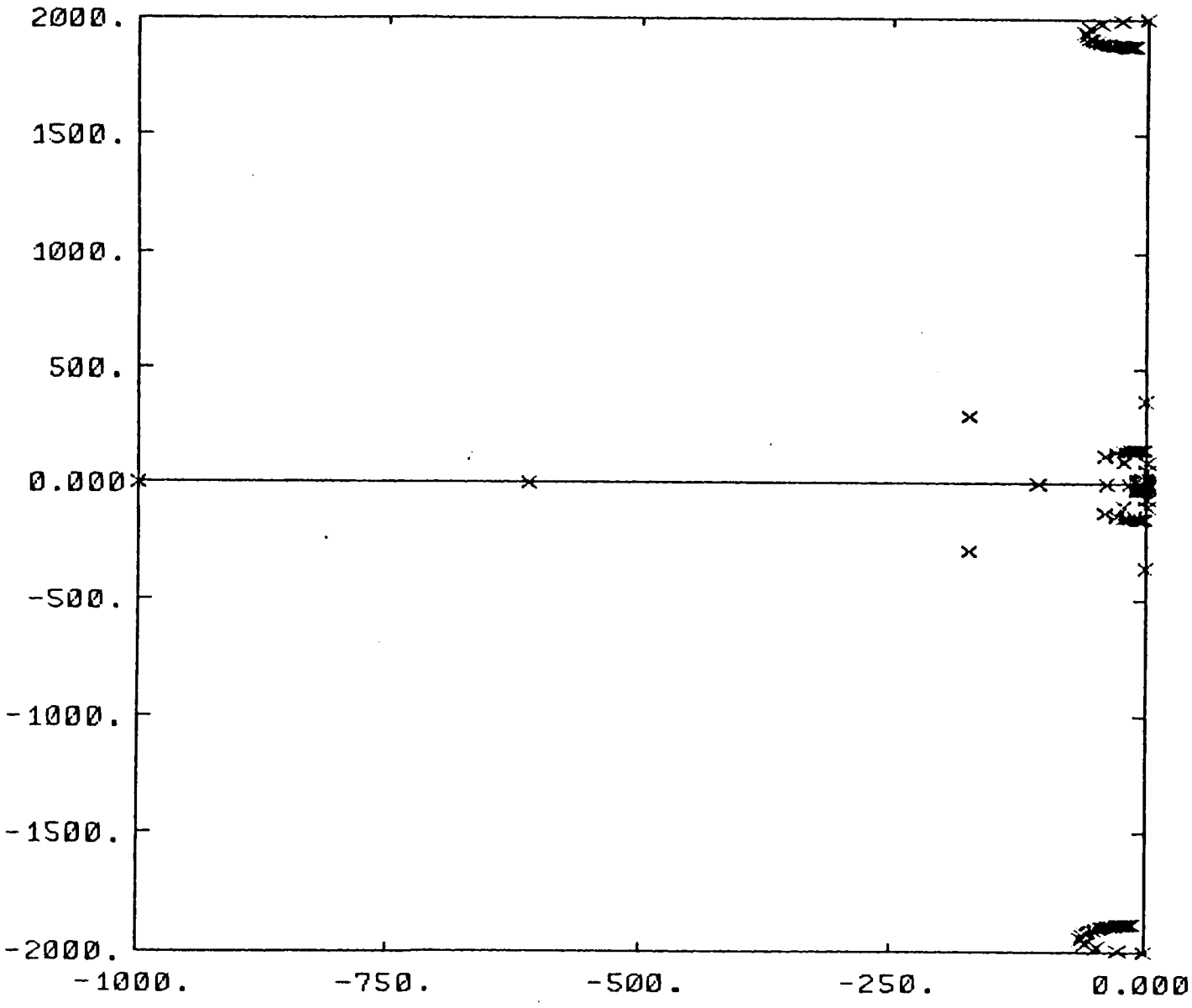


Fig.3.11.b Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.a.

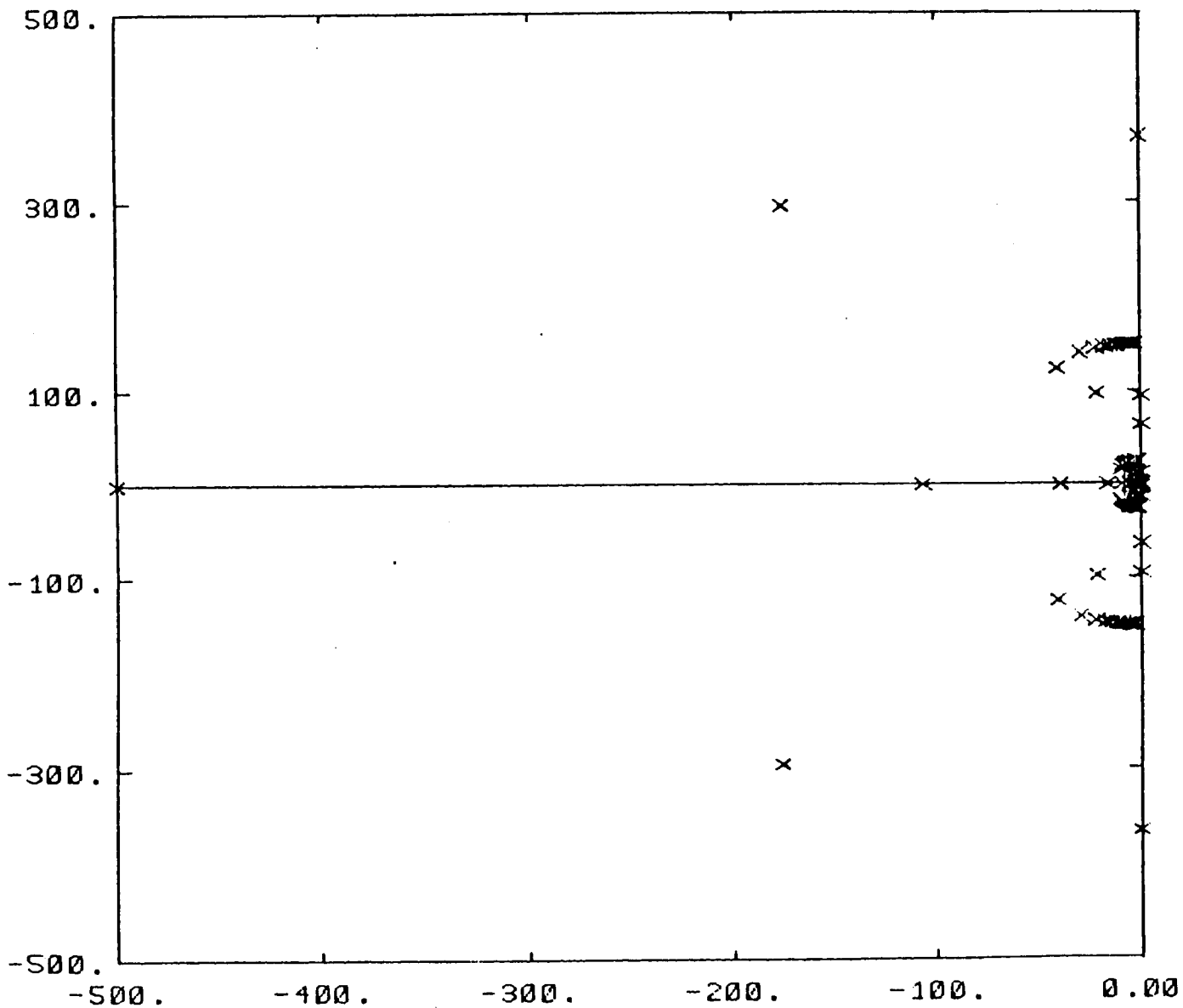


Fig.3.11.c Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.b.

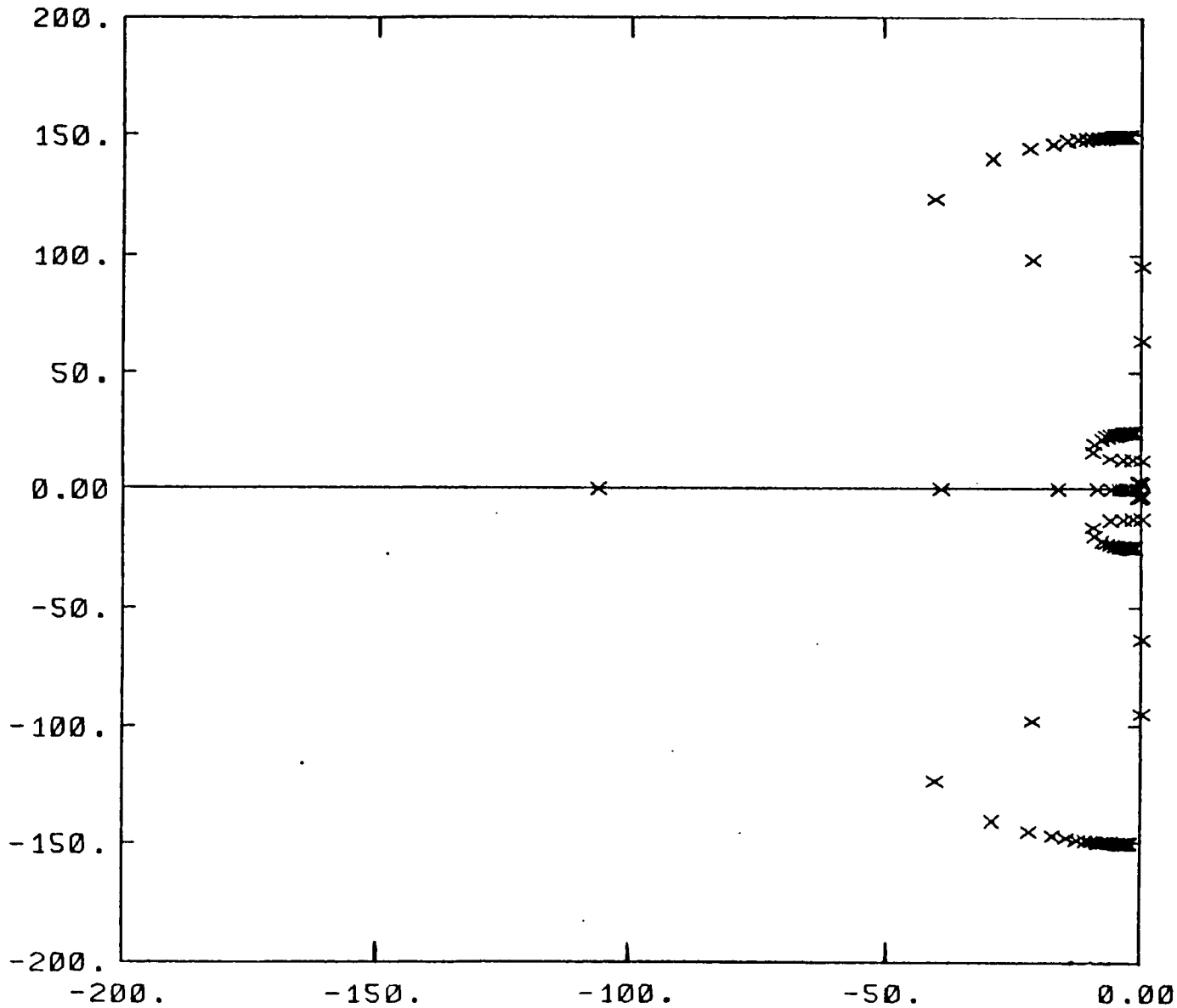


Fig.3.11.d Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.c.

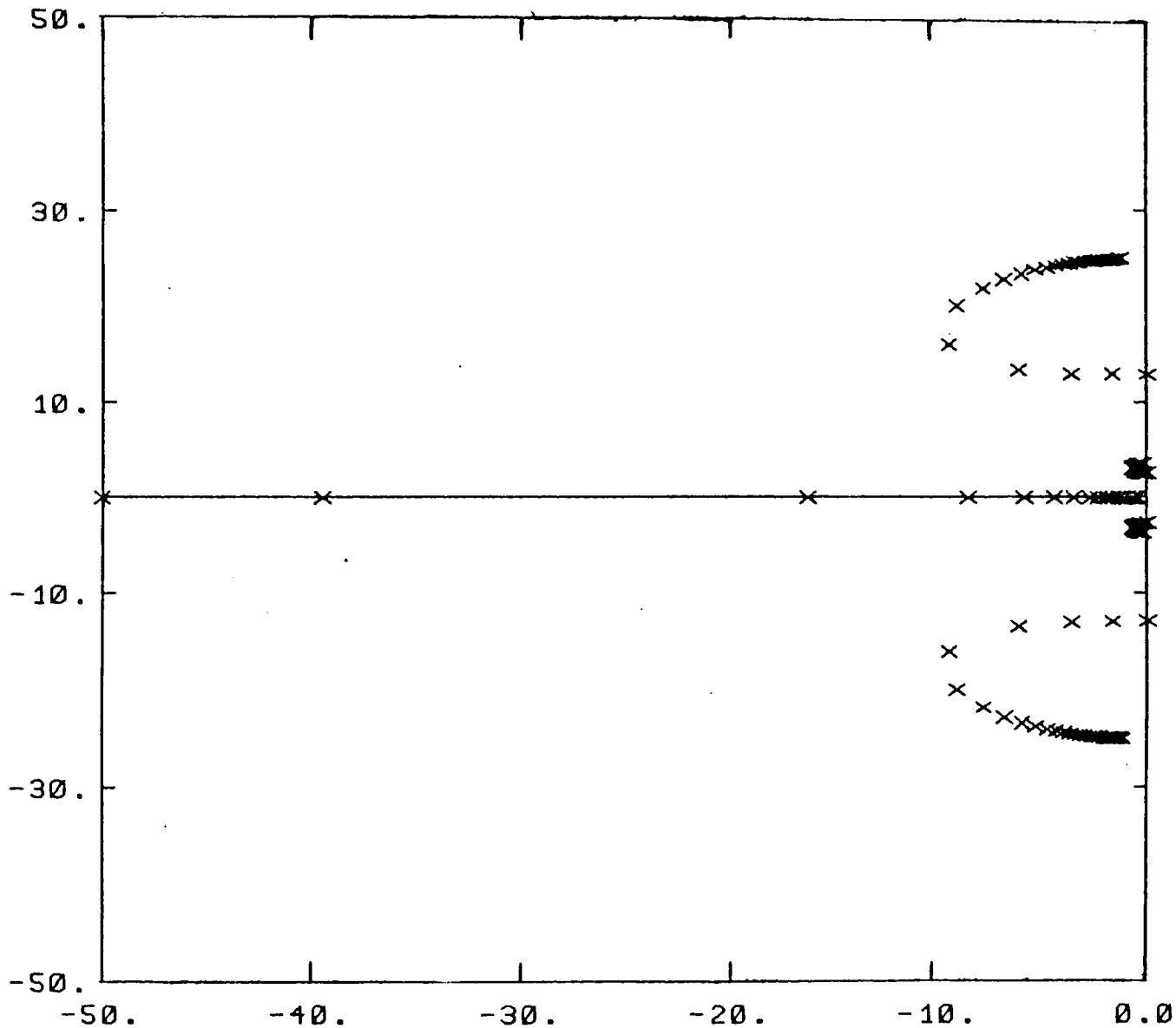


Fig.3.11.e Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.d.

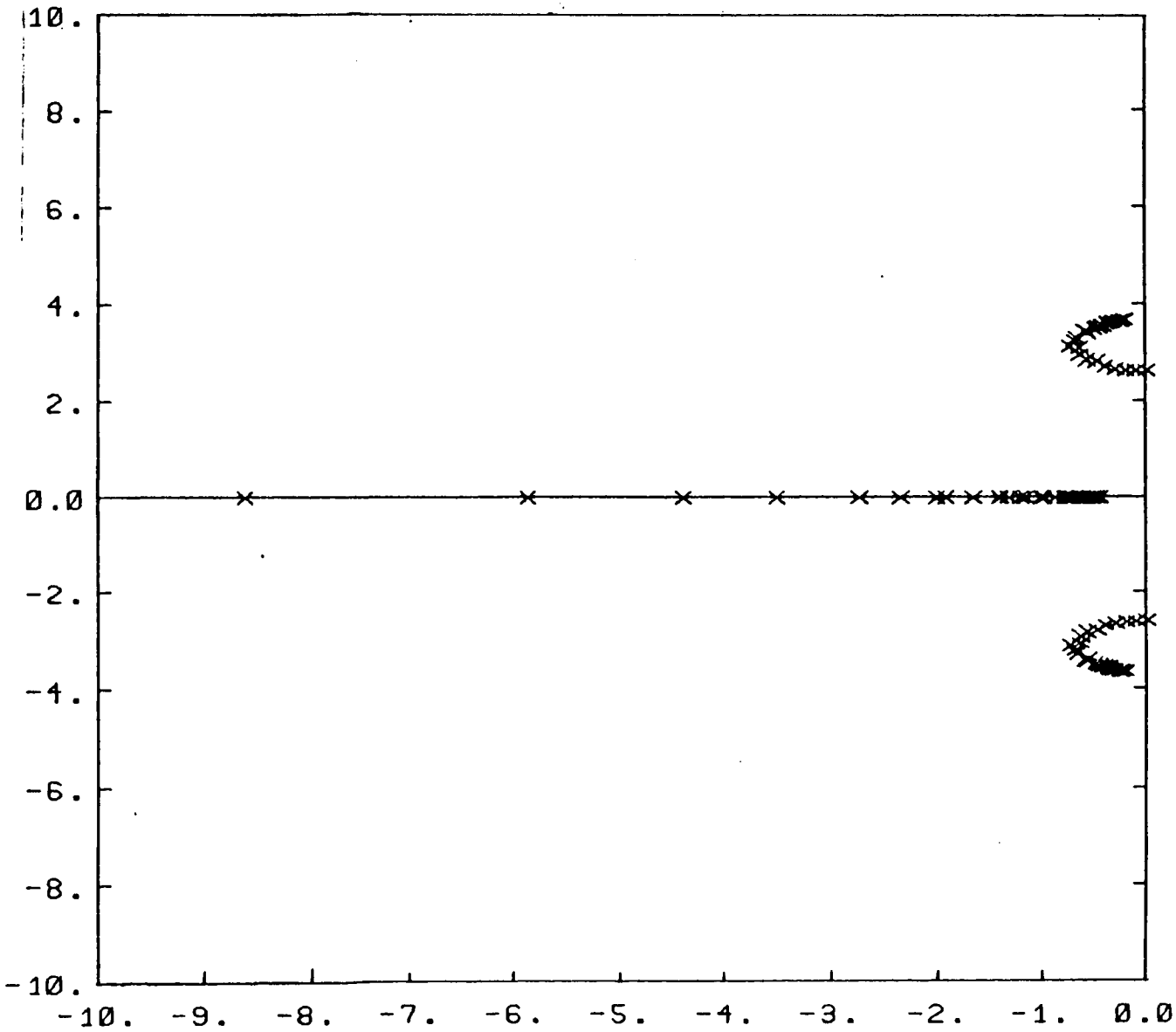


Fig.3.11.f Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.e.

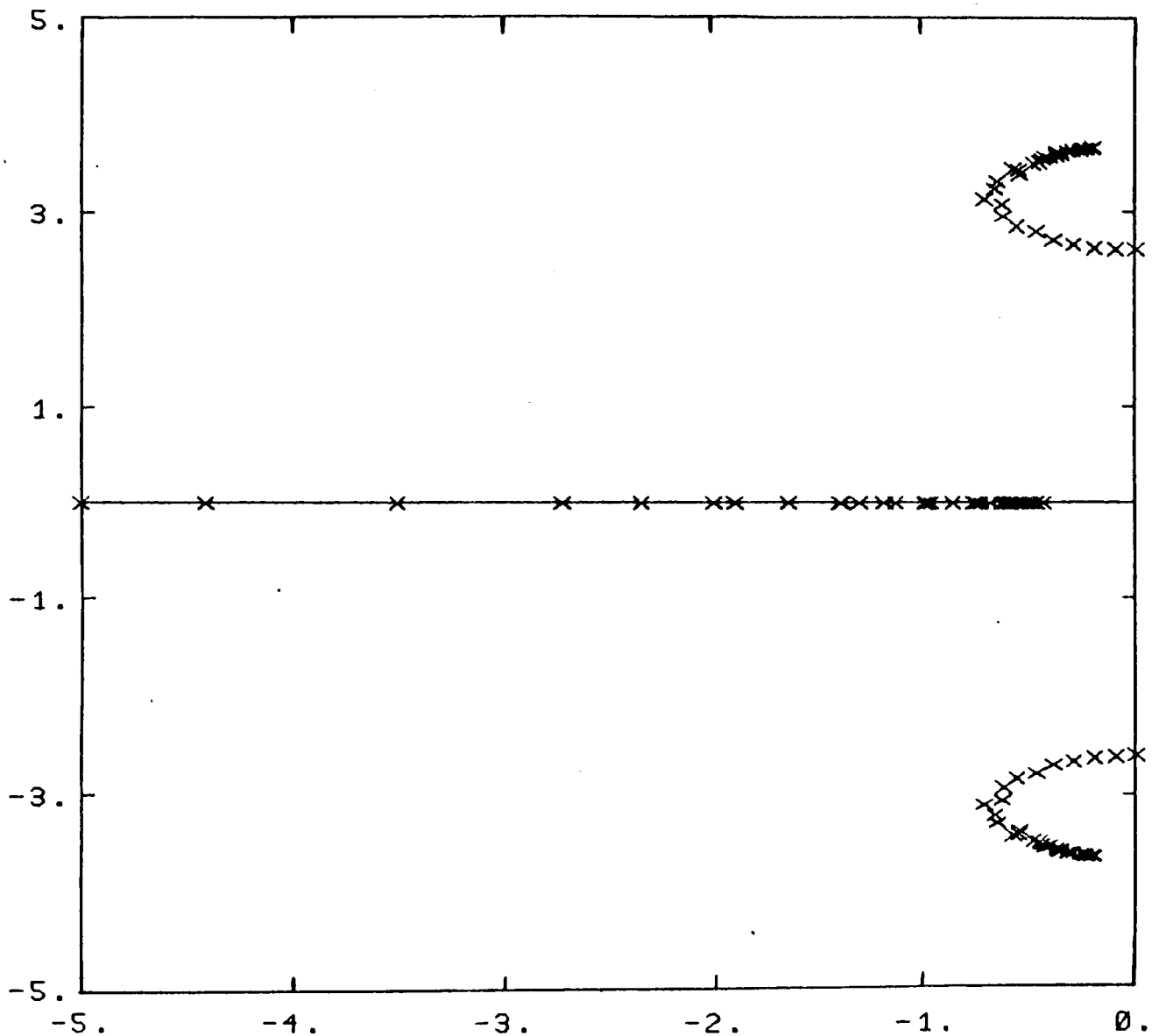


Fig.3.11.g Root locus of closed loop eigenvalues - high servo stiffness case closer look of Fig.3.11.f.

CHAPTER IV

Gross Motion Control

4.1. Introduction

A robot task is characterized as having two major parts, 1. large motion, where speed, and 2. fine motions, where precision (accuracy) is of prime importance. A control strategy is needed for lightweight manipulators , which will accomplish good tracking in joint space while keeping flexible deflections as small as possible $(\theta(t) \rightarrow \theta_d(t), \delta(t) \rightarrow 0)$ for a wide range of speeds and operating conditions. This is the control problem of lightweight manipulators in general.

Before attempting to solve the lightweight manipulator control problem with a specific control approach, let us investigate if it is possible to achieve both perfect joint tracking and vibration stabilization, $(\theta(t) \rightarrow \theta_d(t),\ \delta(t) \rightarrow\ 0)$, in general. If ideal actuators and measurement devices were available, does there exist a control law of the form (4.1)

$$u = u(\theta, \dot{\theta}, \ddot{\theta}, \delta, \dot{\delta}, \ddot{\delta}) \tag{4.1}$$

such that (4.2.a) and (4.2.b) are achieved for the manipulator described by (4.3).

$$\theta(t) \equiv \theta_d(t) \tag{4.2.a}$$

$$\delta(t) \equiv 0 \tag{4.2.b}$$

for all $t \geq t_o$, where t_o is the initial time.

$$\begin{bmatrix} m_r(\theta,\delta) & m_{rf}(\theta,\delta) \\ m_{rf}^T(\theta,\delta) & m_f(\theta,\delta) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} f_r \\ f_f \end{bmatrix} + \begin{bmatrix} 0 \\ [K]\,\delta \end{bmatrix} + \begin{bmatrix} g_r \\ g_f \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \quad ; \qquad (4.3)$$

An investigation of manipulator dynamic model (4.3) reveals that one cannot find a u which will drive $\delta(t) \rightarrow 0$ while achieving perfect tracking of any given desired trajectory in joint space. For any given $(\theta(t), \delta(t))^*$, i.e. (4.2.a) and (4.2.b), the first set of (4.3) gives a control vector history $u(t)^*$. But in general there is no guarantee that the $(\theta(t), \delta(t))^*$ will satisfy the second part of the equation (4.3). That means if $u(t)^*$ is applied to the manipulator, $(\theta(t), \delta(t))^*$ will not be achieved. However, if only $\theta(t)$ is specified (but not $\delta(t)$), *then* $\delta(t)$ is determined from the second part of the (4.3), and then both $\theta(t)$, and $\delta(t)$ determines $u(t)$ from the first part of (4.3). This $u(t)$, when applied to manipulator under ideal conditions, would result in the original $\theta(t)$, and $\delta(t)$. Notice that, in this method, one has the freedom of specifying only $\theta(t)$, but not $\delta(t)$. Therefore, a control strategy which tries to track a desired trajectory, $\theta_d(t)$, in that will result in an acceptable response in $\delta(t)$ is more realistic than a control strategy which tries to accomplish (4.2.a) and (4.2.b).

Such a control strategy alone can not achieve high speed, high precision manipulation. The precision of control must be emphasized at the fine motion level. At this level, positioning of the joint to a desired configuration is needed rather than tracking a desired trajectory. Thus, the fine motion control should achieve $\theta(t) \rightarrow \theta_{final}$, $\delta(t) \rightarrow 0$ asymptotically, where θ_{final} is the joint angle vector corresponding to the final desired configuration. The analysis presented in Chapter 3, based on the linear models of lightweight manipulators, has shown

that the system is locally controllable and the above fine motion objectives can be accomplished through linear state feedback control.

Based on this discussion, the following control strategy will be adapted: First, the gross motion control phase will attempt to control the joint variables in tracking a given trajectory and accept the outcome of this control in flexible vibrations. Then, before the end of motion is reached, the control will be switched to another algorithm that will achieve position and vibration control of the arm. We will call this *the combined control* strategy.

In this chapter, only the gross motion problem will be studied. The need for combined control will arise naturally and one specific form of combined control will be studied in Chapter 5.

The first logical step in gross motion control is to study how well the classical rigid manipulator based control schemes would perform on flexible arms, and use this as a base for further study. First the performance of Computed Torque Method (CTM) and Decoupled Joint Control (DJC) method will be studied on the rigid and the corresponding flexible arms. The comparison between the two results will help to determine the effects and the limitations due to flexibility on the performance of these algorithms. Then, an adaptive model following control (AMFC) algorithm is developed, based on hyperstability. Generalized inertia matrix plays a central role in the design of the AMFC algorithm, and has a number of significant advantages over the other design procedures currently available. Finally, results are compared and shortcomings of these joint variable feedback control schemes are determined.

Some terminology used throughout the chapter is defined as follows. A *high*

gain feedback control system is one where the ratio of the closed loop system bandwidth, w_{ni}, to the desired motion bandwidth, w_{mi}, (or equivalently the natural frequency of the reference model which generates the desired motion with a step input), is larger than 1, $w_{ni}/w_{mi} \gg 1$. A high gain feedback system implies that system capabilities are under utilized. System actuators are capable of providing faster motions, but used for operations involving much lower speeds. By *perfect condition* we mean that the information available to control algorithm about manipulator parameters are exact, and neither external disturbance nor noise exists. *Relatively slow motion with respect to arm flexibility* refers to motions with bandwidth $w_{mi}/w_{cc1} \ll 1$, and *relatively fast motion* refers to $w_{mi}/w_{cc1} \geq 1$, where w_{cc1} is the lowest frequency of the manipulator with all joints clamped. *Relatively slow and fast motion with respect to controller* (more precisely,*with respect to closed loop system dynamics*), refer to the cases of $w_{mi}/w_{ni} \ll 1$ and $w_{mi}/w_{ni} \geq 1$, respectively.

4.2. Non-Adaptive Control Algorithms

4.2.1. Computed Torque Method

The computed torque method is probably the most popular control algorithm in robot motion control. The control vector has two parts: 1. compensation of nonlinearities and gravity, 2. linear joint variable feedback (Fig.4.1.a).

$$u = \hat{f}(\theta, \dot{\theta}) + \hat{g}(\theta) + \hat{m}(\theta)\Big\{\ddot{\theta}_d + [C](\dot{\theta}_d - \dot{\theta}) + [K](\theta_d - \theta)\Big\} \tag{4.4}$$

When applied to a rigid manipulator (model) ;

$$m(\theta)\ddot{\theta} + f(\theta, \dot{\theta}) + g(\theta) = u \tag{4.5}$$

$$m(\theta)\ddot{\theta} + f(\theta,\dot{\theta}) + g(\theta) = \hat{f}(\theta,\dot{\theta}) + \hat{g}(\theta) + \hat{m}(\theta)\{\ddot{\theta}_d + [C](\dot{\theta}_d - \dot{\theta}) + [K](\theta_d,\theta)\} \quad (4.6)$$

where, $m(\theta), f(\theta,\dot{\theta}), g(\theta)$ are actual inertia matrix, nonlinearities, and gravity terms, respectively, of the real world system, which cannot be exactly determined by modeling or identification. $\hat{m}(\theta), \hat{f}(\theta), \hat{g}(\theta)$ are the values known at the control algorithm level which are always different than the actual real world values to some degree. When the CTM is applied to flexible manipulators,

$$m_r(\theta,\delta)\ddot{\theta} + f_r(\theta,\delta,\dot{\theta},\dot{\delta}) + g_r + m_{rf}(\theta,\delta) = \hat{f} + \hat{g}$$
$$\hat{m}(\theta)\left[\ddot{\theta}_d + [C](\dot{\theta}_d - \dot{\theta}) + [K](\theta_d - \theta)\right] \quad (4.7.a)$$

and the flexible body dynamics during the motion is governed by,

$$m_f(\theta,\delta)\ddot{\delta} + m_{rf}^T(\theta,\delta)\ddot{\theta} + f_f + [K]\delta + g_f = 0 \quad (4.7.b)$$

Let $e(t)$ be the error state, the difference between desired and actual joint variable states at time t,

$$e = \theta_d - \theta$$

The error dynamics are governed by,

$$\ddot{e} + [C]\dot{e} + [K]e = \left[\hat{m}^{-1}m_r - I\right]\ddot{\theta} + \hat{m}^{-1}\left[(f_r - \hat{f}) + (g_r - \hat{g})\right] + \hat{m}^{-1}m_{rf}\ddot{\delta} \quad (4.8)$$

If $\hat{m} = m_r$, $\hat{f} = f_r$, $\hat{g} = g_r$, and $\dot{\delta} = \ddot{\delta} = 0$, under ideal conditions, then the error dynamics are governed by,

$$\ddot{e} + [C]\dot{e} + [K]e = 0 \quad ; \quad (4.9)$$

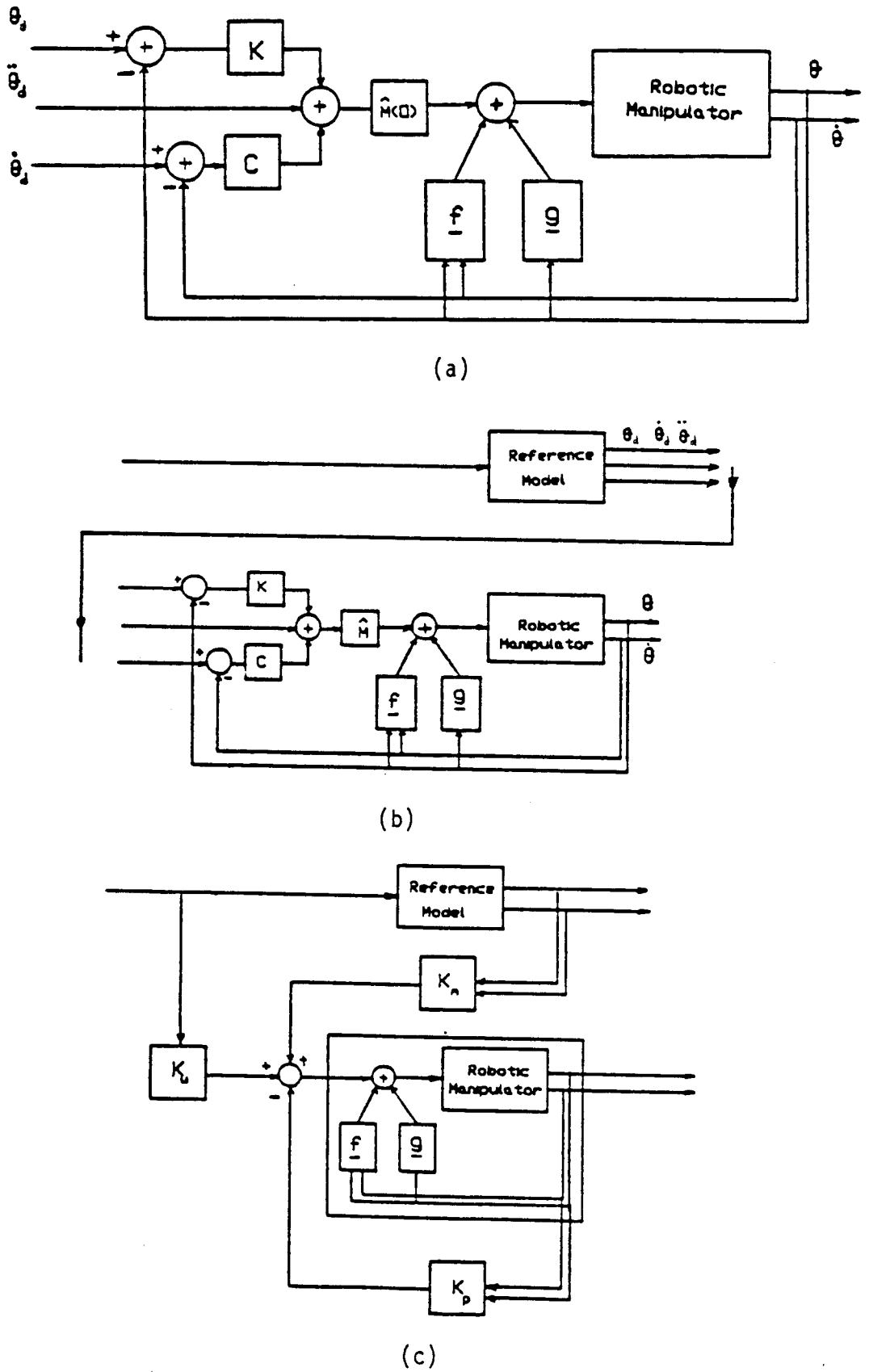


Fig.4.1 Computed torque control: a) standard form, b) model following form, c) model following form

For decoupled joint control; $[C] = diag[c_{ii}]$; $[K] = diag[k_{ii}]$,

$$\ddot{e}_i + c_{ii}\dot{e}_i + k_{ii}e_i = 0 \quad ; \tag{4.10}$$

Finding the appropriate controller feedback gain parameters to achieve a desired set of closed loop eigenvalues is trivial with the computed torque method. In order to assign a damping ratio of ξ_i (i.e. 1.0) and a natural frequency w_{ni} (i.e. $5.rad/sec$) for the closed loop dynamics of a joint variable, decoupled from other joint dynamic effects, one simply chooses;

$$c_{ii} = 2\xi_i w_{ni} \tag{4.11.a}$$

$$k_{ii} = w_{ni}^2 \tag{4.11.b}$$

The steady state error for a constant disturbance, i.e $m_{p\ design} \neq m_{p\ actual}$ acts as a constant disturbance on the control system, is as follows:

$$\lim_{t\to\infty} e_{ss} = \lim_{s\to 0} s\ \frac{W_o/s}{k_{ii}} = \frac{W_o}{k_{ii}} \tag{4.12}$$

Clearly, steady state error is reduced by use of high servo bandwidth. In practice, integral control is also included to zero out the error. In robotics, a typical source of constant disturbance in steady state is the payload variations from one task to another. Notice that as k_{ii} increases, steady state error decreases, thus high bandwidth closed loop system results in smaller steady state errors. Moreover, high closed loop bandwidth relative to the desired motion bandwidth (high gain feedback) results in better transient tracking response. Due to flexible dynamic coupling, reaching steady state value for joint variables depends on how fast the flexible vibrations are damped-out (eqn.(4.7.a)).

In order to clarify the notion of *desired motion bandwidth*, consider the figure 4.2. Let the states of the reference model be the desired trajectory, and input to the model is a unit step. The response of the model (4.13) is given by (4.14).

$$\ddot{\theta}_{mi} + 2\xi_i w_{mi}\dot{\theta}_{mi} + w_{mi}^2\theta_{mi} = w_{mi}^2 u_{mi} \tag{4.13}$$

$$\theta_{mi}(t) = \left[1 - e^{-\xi_i w_{mi} t}(\cos\sqrt{1-\xi_i^2}\; w_{mi}t + \; (\xi_i/\sqrt{1-\xi_i^2})\; \sin\sqrt{1-\xi_i^2}\; w_{mi}t)\right] \tag{4.14}$$

Thus, the bandwidth of a desired motion can be characterized by the highest frequency content of the motion, which is w_{mi} in this case.

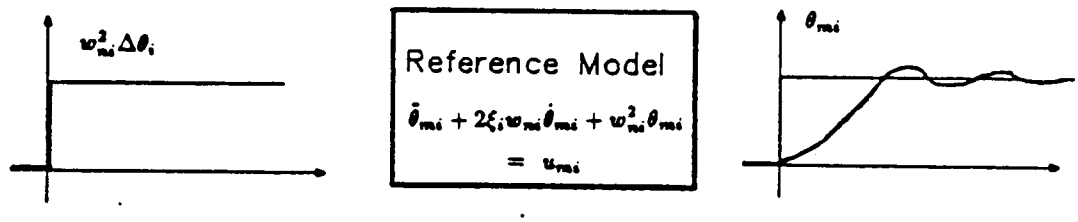


Fig.4.2 Desired motion generation

4.2.2. Decoupled Joint Control (DJC)

The decoupled joint control (DJC) is another very popular method in robot motion control. It is very similar to the computed torque method, except that it does not explicitly cancel nonlinearities (Fig.4.3.a). Therefore, implementation requires much less computational power than the computed torque method would. However, as operation speeds increase, the nonlinear forces become dominant, and the only way DJC can be successfully used in these cases is in *a high gain feedback* form. The decoupled joint control algorithm is given by,

$$u = \hat{g}(\theta) + \hat{m}(\theta)\Big\{[c_{ii}](\dot{\theta}_d - \dot{\theta}) + [k_{ii}](\theta_d - \theta)\Big\} \tag{4.15}$$

When applied to rigid manipulators, this yields:

$$m(\theta)\ddot{\theta} + f(\theta,\dot{\theta}) + g(\theta) = \hat{g}(\theta) + \hat{m}(\theta)\left[[c_{ii}](\dot{\theta}_d - \dot{\theta}) + [k_{ii}](\theta_d - \theta)\right] \tag{4.16}$$

Add and subtract $\hat{m}(\theta)(\ddot{\theta}_d - \ddot{\theta})$ from equation (4.16) in order to obtain the error dynamics,

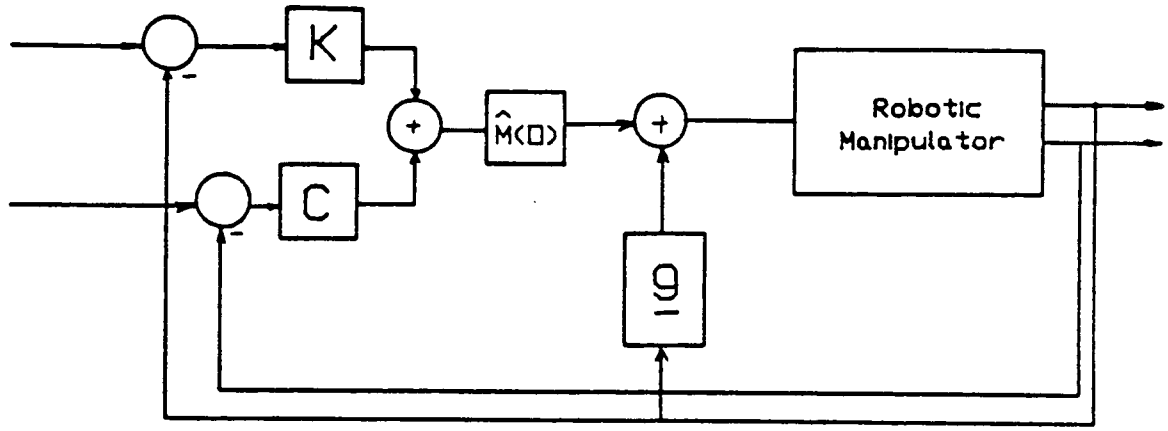
$$\ddot{e} + [C]\dot{e} + [K]e = \left[\hat{m}^{-1}(\theta)m(\theta) - I\right]\ddot{\theta} + \ddot{\theta}_d + \hat{m}^{-1}\left[(f) + (g - \hat{g})\right] \tag{4.17}$$

When applied to flexible manipulators, the error dynamics is governed by

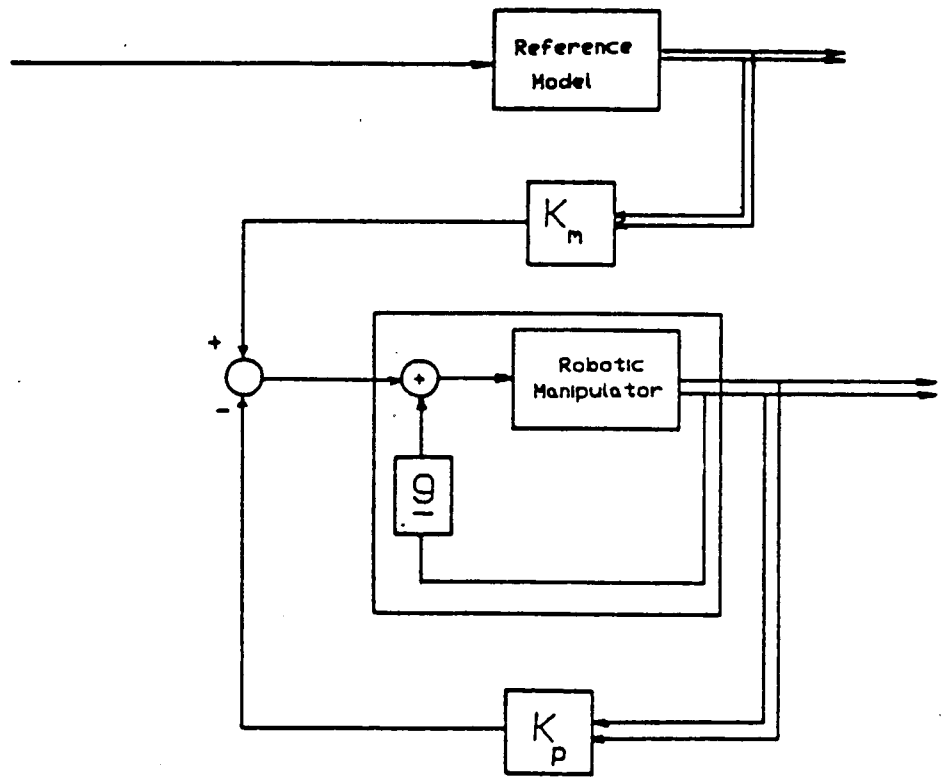
$$\ddot{e}+[C]\dot{e}+[K]e = \left[\hat{m}^{-1}m_r - I\right]\ddot{\theta}+\ddot{\theta}_d++\hat{m}^{-1}\left[(f_r) + (g_r - \hat{g})\right]+\hat{m}^{-1}m_{rf}\ddot{\delta} \tag{4.18.a}$$

and the flexible body dynamics during the motion is given by

$$m_f(\theta,\delta)\ddot{\delta} + m_{rf}^T(\theta,\delta)\ddot{\theta} + f_f + [K]\delta + g_f = 0 \tag{4.18.b}$$



(a)



(b)

Fig.4.3 Decoupled joint control: a) standard form b) model following form.

4.2.3. Model Following Form of the CTM and DJC

For comparisons between the performance of the CTM, DJC, and AMFC algorithms a common basis must be established. A reference model, which basically serves as a desired trajectory generator, is an integral part of AMFC. Thus, the CTM and DJC must be implemented such that they follow the same desired motion as does the AMFC, with the same closed loop system objectives. Here, a reference model following view point of these methods will be studied in order to establish a common basis for comparison.

In order to implement CTM, (4.4), in the model following form, one must express the $\ddot{\theta}_d, \dot{\theta}_d, \theta_d$ in terms of the reference model states elements. Let the reference model be, a linear dynamic system of the form,

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_o & -\Lambda_1 \end{bmatrix} \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_m \tag{4.21}$$

where; $\Lambda_o = diag\{w_{mi}^2\}$ $\Lambda_1 = diag\{\xi_i w_{mi}\}$ and $u_m = w_{mi}^2 \Delta\theta$.
The relationship between desired trajectory and reference model states is given by (Fig.4.5),

$$\begin{aligned} \theta_d &= \theta_m + \theta_0 \\ \dot{\theta}_d &= \dot{\theta}_m \\ \ddot{\theta}_d &= \ddot{\theta}_m \end{aligned} \tag{4.22}$$

$$\ddot{\theta}_m + \Lambda_1 \dot{\theta}_m + \Lambda_o \theta_m = u_m \tag{4.23}$$

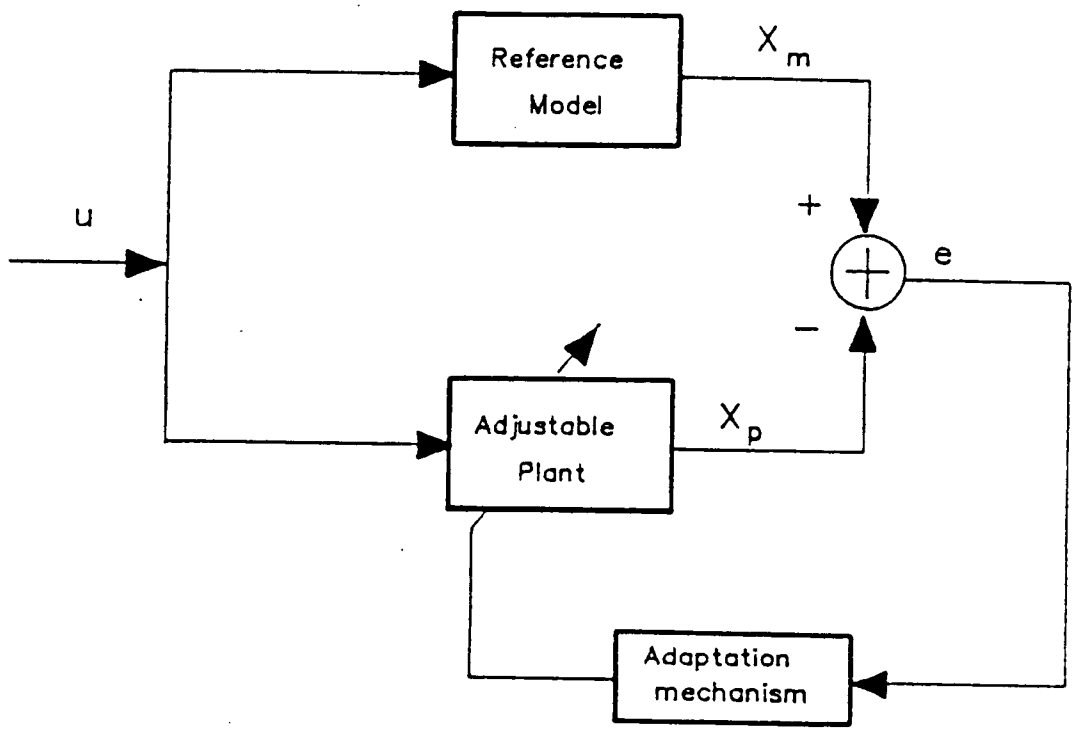


Fig.4.4 Basic elements of an adaptive model following control system

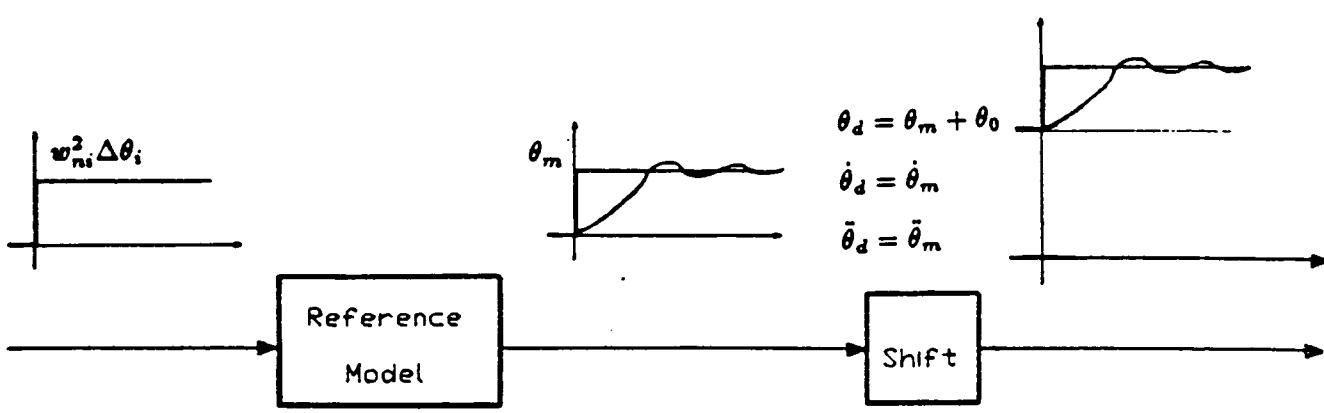


Fig.4.5 Reference model used as trajectory generator.

$$u_1 = \{\ddot{\theta}_d + [c_{ii}]\dot{\theta}_d + [k_{ii}]\theta_d\} \tag{4.24}$$

Given the relationships (4.22), and (4.23), the desired trajectory part can be expressed in terms of reference model input, states and dynamics. After some algebraic manipulation, one can show that the equivalent control law of (4.24) is given by (4.25),

$$u_1 = u_m + \{[c_{ii}] - [\Lambda_1]\}\dot{\theta}_m + \{[k_i i] - [\Lambda_0]\}\theta_m + [k_{ii}]\theta_0 \tag{4.25}$$

Thus the computed torque control law in model following implementation form; (Fig.4.1.b-c)

$$\begin{aligned} u = {} & \hat{m}_r(\theta,\delta)\Big[u_m\Big] + \hat{m}_r(\theta,\delta)\Big[\{[c_{ii}] - [\Lambda_1]\}\dot{\theta}_m + \{[k_i i] - [\Lambda_0]\}\theta_m\Big] \\ & + \hat{m}_r(\theta,\delta)\Big[[k_{ii}]\theta_0\Big] \\ & + \hat{m}_r(\theta,\delta)\{[c_{ii}]\dot{\theta} + [k_{ii}]\theta\} \\ & + \hat{f} + \hat{g} \end{aligned} \tag{4.26}$$

Notice that if $[C_{ii}] = \Lambda_1$, $[K_{ii}] = \Lambda_0 \Rightarrow w_{ni}/w_{mi} = 1$ with u_m being step input, corresponding to a *relatively fast* motion with respect to controller. Similarly, the model following implementation of DJC (Fig. 4.2.b) is given by,

$$\begin{aligned} u &= u_1 + \hat{g}(\theta) \\ u_1 &= \hat{m}_r(\theta,\delta)\Big[[c_{ii}]\dot{\theta}_m + [k_{ii}]\theta_m\Big] + \hat{m}_r(\theta,\delta)[k_{ii}]\theta_0 \\ &\quad - \hat{m}_r(\theta,\delta)\Big[[c_{ii}]\dot{\theta} + [k_{ii}]\theta\Big] \end{aligned} \tag{2.27}$$

4.3. Adaptive Model Following Control (AMFC)

The AMFC studied here is based on the Erzberger's linear perfect model following control (LPMFC) problem. First the LPMFC problem, then the AMFC problem will be studied.

4.3.1. Preliminaries: LPMFC

Consider the linear time invariant plant,

$$\dot{x}_p = A_p x_p + B_p u_p \tag{4.28}$$

where (A_p, B_p) controllable, $x_p \in R^n, u_p \in R^m$. Let the reference model be,

$$\dot{x}_m = A_m x_m + b_m u_m \tag{4.29}$$

and a control law of the form,

$$u_p = -K_p x_p + K_u u_m + K_m x_m \tag{4.30}$$

The Problem: Given the reference model (4.29) and plant (4.28) [Fig.4.6.a], does there exist K_p, K_u, and K_m such that for initial conditions $e(0) = 0$ and for all u_m that belong to a piecewise continuous, bounded class of functions,

$$e = x_m - x_p \equiv 0 \Rightarrow x_m(t) \equiv x_m(t) \quad \forall\, t \geq t_0 \tag{4.31}$$

The Solution: Let e be the error between reference model and actual states,

$$e = x_m - x_p \tag{4.32}$$

$$\dot{e} = \dot{x}_m - \dot{x}_p \tag{4.33}$$

$$= A_m x_p + B_m u_m - A_p x_p - B_p(-K_p x_p + K_u u_m + K_m x_m)$$

$$\pm\,(A_m x_p + B_p K_m x_p) \tag{4.34}$$

where last term is added to eliminate x_m from the error dynamics. Finally, the error dynamics are governed by,

$$\dot{e} = [A_m - B_p K_m]e + [(A_m - A_p) + B_p(K_p - K_m)]x_p + [B_m - B_p K_u]u_m \quad (4.35)$$

First of all, part of open loop error dynamics, A_m, will normally be chosen with desired properties by the designer, thus K_m can be set to zero without loss of generality. If later a case of $K_m \neq 0$ is desired, a modification of the design for that purpose is trivial. In order to accomplish the (4.31) for all u_m and x_p, as described in the problem, coefficients of u_m and x_p in (4.35) must be zero at all times.

$$A_p - A_m = B_p(K_p - K_m) \quad (4.36.a)$$

$$B_m = B_p K_u \quad (4.36.b)$$

If there exist K_p, K_u, K_m to exactly satisfy (4.36), then there exists a control of the form (4.30) which accomplishes (4.31), which is called *linear perfect model following control* (LPMFC) [D18]. If $(A_p - A_m) \in Span\{B_p\}$, and $B_m \in Span\{B_p\}$, then the existence of K_p, K_u, K_m is guaranteed (Appendix A). So for a given plant with a model (A_p, B_p) of the form (4.28), there exist a class of reference models $\{(A_m, B_m)\}$, satisfying (4.36), such that any element from that class can be exactly tracked by the plant using a control law of the form (4.30).

Notice that for square systems with B_p full rank, any reference model (A_m, B_m) can be perfectly followed. For a set of second order systems which can be expressed in the form of (4.37),

$$\left\{ \begin{bmatrix} 0 & I \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} 0 \\ B_2 \end{bmatrix} \right\} \quad (4.37)$$

if B_2 is full rank, any set of second order reference models of similar structure can be perfectly followed.

4.3.2. AMFC - Hyperstability based design

Consider the plant dynamics of the previous section as $(A_p(x_p,t), B_p(x_p,t))$ time varying system (so called *quasi-linear* form), instead of being *linear time invariant* (LTI). The basic idea of AMFC relies on the LPMFC of Erzberger. AMFC attempts to realize the same objective of LPMFC for a time varying system asymptotically as $t \rightarrow \infty$ (Fig. 4.6.a, and b).

Let the reference model be,

$$\dot{x}_m = A_m x_m + B_m u_m \tag{4.38}$$

and the plant dynamics be,

$$\dot{x}_p = A_p(x_p,t)x_p + B_p(x_p,t)u_p \tag{4.39}$$

with a control law of the form,

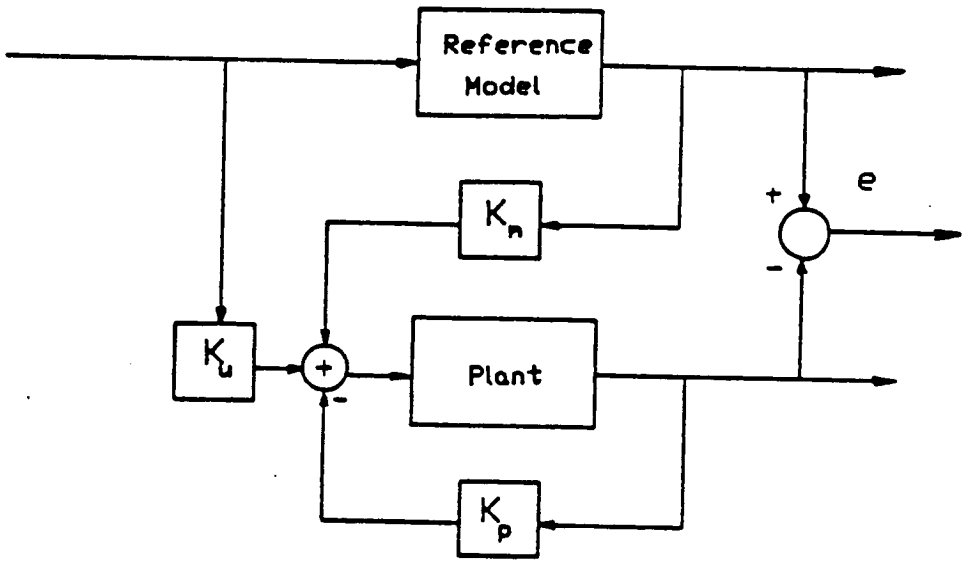
$$u_p = -K_p x_p + K_u u_m + K_m x_m \tag{4.40}$$

Clearly, as the system dynamics varies, the feedback gains must also vary in order to keep following the reference model.

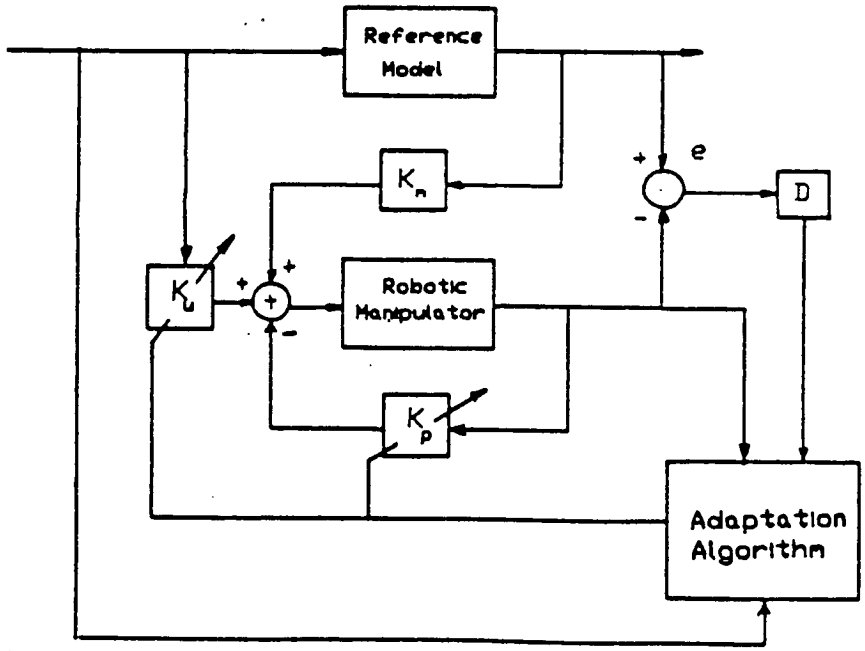
There are two basic assumptions associated with the current AMFC designs.

1. $\exists\ K_p, K_u, K_m$ *for every* $A_p(x,t)B_p(x,t)$

2. *Variations of* $A_p(X,t), B_p(x,t)$ *is slower than the speed of adaptation.*



(a)



(b)

Fig.4.6 Model following control: a) Linear perfect model following control (LPMFC), b) Adaptive model following control,

Assumption 1 is an expected *existence* condition, originating from the Erzberger's LPMFC conditions. AMFC attempts to converge to these values through adaptation as system dynamics vary. Existence of such limit values are the first requirement for the convergence, let aside whether the algorithm will be able to converge or not.

Assumption 2 is ordinarily made in the current AMFC design . This assumption will be replaced with a less restrictive one by a new design method described in the next section of the paper. Basically, this assumption says that during an adaptation interval, a time invariant approximation of the time varying plant should be accurate enough. In robotics, this depends on the speed of motion, thus AMFC requires the robot motions to be slow enough for the adaptation algorithm to work well.

In the remaining part of this section, the standard AMFC design based on hyperstability will be discussed. The next section will extend these ideas and remove assumption 2.

From (4.38), (4.39), (4.40), the error dynamics can be shown to be (following the same steps of the section 4.3.1),

$$\begin{aligned} \dot{e} &= [A_m - B_p(x_p,t)K_m]e + \\ &[A_m - A_p(x_p,t) + B_p(x_p,t)(K_p - K_m)]x_p + [B_m - B_p(x_p,t)K_u]u_m \end{aligned} \tag{4.41}$$

Letting $K_m = 0$ without loss of generality,

$$\begin{aligned} \dot{e} &= A_m e + [A_m - A_p(x_p,t) + B_p(x_p,t)K_p]x_p \\ &\quad + [B_m - B_p(x_p,t)K_u]u_m \end{aligned} \tag{4.41}$$

For $e(t) \to 0$ as $t \to \infty$, for all x_p, and u_m that belong to a piecewise continuous, bounded class of functions, the coefficients of x_p , and u_m in (4.41) must be zero.

Since A_p, and B_p are not constant, the problem is to devise an adaptation algorithm for K_p, and K_u such that as A_p, and B_p vary, K_p, and K_u are also varied to keep the coefficients of x_p , and u_m zero.

$$\begin{aligned} A_p(x_p,t) - A_m &= B_p(x_p,t)K_p \\ B_m &= B_p(x_p,t)K_u \end{aligned} \tag{4.42}$$

Let the feedback gains be

$$\begin{aligned} K_p &= K_{pn} - \Delta K_p(e,t) \\ K_u &= K_{un} + \Delta K_u(e,t) \end{aligned} \tag{4.43}$$

where, K_{pn}, and K_{un} are nominal gains, and are not affected by the adaptation algorithm. It is assumed that for every instantaneous value of $(A_p(x_p,t), B_p(x_p,t))$ and the chosen (A_m, B_m), there exist K_p, K_u such that (4.42) is satisfied (Assumption # 1).

The adaptation algorithm deals with the question of how to vary the $\Delta K_p, \Delta K_u$ so that equality (4.42) is preserved as closely as possible.

There are three basic methods of designing the adaptation algorithm, namely gradient [D19], Lyapunov [D1], and hyperstability [D1, D3] based methods. Hyperstability based design is proven to be the most powerful method [D1] and will be used here.

Recall the control law,

$$\begin{aligned} u_p &= -K_p x_p + K_u u_m + \overbrace{K_m x_m}^{0.} \\ &= \underbrace{-K_{pn}x_p + K_{un}u_m}_{Nominal\ control} + \underbrace{\delta K_p(e,t)x_p + \delta K_u(e,t)u_m}_{Adaptation\ algorithm\ control\ action} \end{aligned} \tag{4.44}$$

Ordinarily, K_{pn} and K_{un} are chosen to be constant. It will be shown that this choice is the source of assumption 2.

Substituting the control gains (4.43) into the error dynamics (4.41);

$$\begin{aligned} \dot{e} = A_m e + [(A_m - A_p) + B_p(K_{pn} - \Delta K_p(e,t))]x_p \\ + [B_m - B_p(K_{un} + \Delta K_u)]u_m \end{aligned} \tag{4.45}$$

Given the error dynamics (4.45), it is desired to find a way to adapt $\Delta K_p, \Delta K_u$ so that the coefficients of x_p and u_p go to zero asymptotically as time goes to infinity.

The hyperstability based adaptation algorithm design involves the following four standard steps: [Fig. 4.7]

1. Transfer the problem to the form of the standard hyperstability problem.

2. Determine the class of adaptation laws such that feedback block belongs to Popov class (Fig. 4.7) and choose a specific adaptation law from that class.

3. Find an output filter, D, using Kalman-Yakubovich-Popov lemma (Appendix C), such that the linear time invariant feedforward block is strictly positive real.

4. Transfer back to the original problem.

Note that the existence condition (Assumption # 1) implies that for every instant value of $A_p(x_p,t)$, $B_p(x_p,t)$, there exists K_p^* , and K_u^* such that LPMFC (4.36.a and 4.36.b) are satisfied. The values of K_p^*, K_u^* may not be known, but their

existence is!

$$A_p(x_p,t) - A_m = B_p(x_p,t)K_p^*$$
$$B_m = B_p(x_p,t)K_u^* \tag{4.46}$$

So, as $A_p(x_p,t), B_p(x_p,t)$ vary, the ideal values of K_p^*, K_u^* vary according to (4.46). The task of adaptation algorithm is to vary $\Delta K_p, \Delta K_u$ such that the actual implemented K_p, K_u converges to K_p^*, K_u^* asymptotically, thus satisfying (4.42). Substituting the ideal values into the error dynamics;

$$\dot{e} = A_m e + B_p(x_p,t)\Big\{[-K_p^* + K_{pn} - \Delta K_p(e,t)]x_p + [K_u^* - (K_{un} - \delta K_u(e,t))]u_m\Big\} \tag{4.47}$$

One can now apply the hyperstability based design procedure;

Step1:

$$\dot{e} = A_m e + B_p(x_p,t)w_1$$
$$v = De \tag{4.48.a}$$
$$w = -w_1 = [K_p^* - K_{pn} + \Delta K_p]x_p + [K_{un} + \Delta K_u(e,t) - K_u^*]u_m \tag{4.48.b}$$

Step 2:

Any choice of $\Delta K_p, \Delta K_u$ from the following class guarantees that the resultant feedback block belongs to Popov class (see Appendix C).

$$\Delta K_p(e,t) = \int_0^t \phi_1(v,t,\tau)d\tau + \phi_2(v,t) + \Delta K_p(0) \tag{4.49.a}$$
$$\Delta K_u(e,t) = \int_0^t \psi_1(v,t,\tau)d\tau + \psi_2(v,t) + \Delta K_u(0) \tag{4/49/b}$$

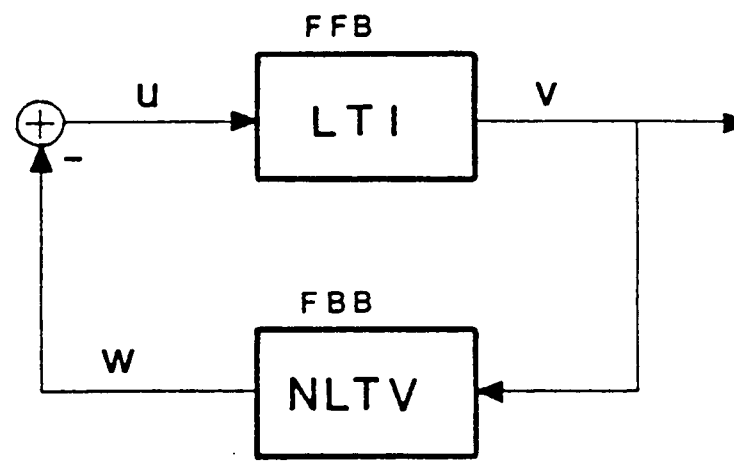
where;

$$\phi_1(v,t,\tau) = F_{p1}(t-\tau)v(\tau)[G_{p1}x_p(\tau)]^T \tag{4.50.a}$$

$$\phi_2(v,t) = F_{p2}(t)v(t)[G_{p2}(t)x_p(t)]^T \tag{4.50.b}$$

$$\psi_1(v,t,\tau) = F_{u1}(t-\tau)v(\tau)[G_{u1}u_m(\tau)]^T \tag{4.50.c}$$

$$\psi_2(v,t) = F_{u2}(t)v(\tau)[G_{u2}(t)u_m(\tau)]^T \tag{4.50.d}$$



$$\{P\} = \{ \int_{t_o}^{t_1} v^T w dt \geq -\gamma_o^2 \quad , \forall\, t_1 \geq t_o\}$$

Fig.4.7 The Hyperstability problem

where; $F_{p1}(t-\tau), F_{u1}(t-\tau)$ are positive definite matrix kernels whose Laplace transform is a positive real transfer matrix with a pole at $s = 0$, $G_{p1}, G_{u1} > 0$ positive definite matrices, and $F_{p2}(t), F_{u2}(t), G_{p2}(t), G_{u2}(t) \geq 0$ are positive semi-definite matrices.

With an adaptation algorithm from that class, the feedback block (4.48.b) becomes,

$$w = -w_1 = [\int_0^t \phi_1(v,t,\tau)d\tau + \phi_2(v,t) + \Delta K_p^0]x_p + [\int_0^t \psi_1(v,t,\tau)d\tau + \psi_2(v,t) + \Delta K_u^0]u_m \tag{4.51}$$

where ;

$$\Delta K_p^0 = K_p^* - K_{pn} + \Delta K_p(0) \tag{4.52.a}$$

$$\Delta K_u^0 = -K_u^* + K_{un} + \Delta K_u(0) \tag{4.52.b}$$

$\Delta K_p(0)$ and $\Delta K_u(0)$ can be chosen to be zero without loss of generality, for any desired values can be included in the K_{pn}, and K_{un} nominal values respectively.

The assumption #2 originates at this point. The feedback block, with choices of adaptation gains from the above class, satisfies the Popov inegral inequality, ($\int_0^t v^T.w \geq -\gamma_1^2 \quad \forall\ t \geq 0$), for constant ΔK_p^0, ΔK_u^0 [D3, D18]. ΔK_p^0, ΔK_u^0 constant implies that $(K_p^* - K_{pn})$ and $(K_u^* - K_{un})$ are constant (eqn.(4.52.a) and (4.52.b)).

If K_{pn}, K_{un} are chosen to be constant, as done currently in the AMFC design literature, K_p^*, K_u^* must be constant for the hyperstability based design to be successful (eqn. (4.52.a)). Therefore (4.46) implies that $(A_p(x_p,t), B_p(x_p,t))$ must be constant during the adaptation, or equivalently, $A_p(x_p,t)$ and $B_p(x_p,t)$ should vary slower than the speed of adaptation (Assumption 2).

4.3.3. AMFC Revisited: Generalized Inertia Matrix Based AMFC

Hyperstability based design requires $\Delta K_p^0, \Delta K_u^0$ be constant for the resultant feedback block of hyperstable design to be in Popov class. Therefore, Eqn.(4.52) implies that $(K_p^* - K_{pn})$ and $(K_u^* - K_{un})$ must be constant, *but not necessarily* K_{pn}, K_{un}. If nominal control is not constant but somewhat better in keeping the plant to track the reference model, then assumption 2 would not have to be so restrictive. So the better K_{pn} tracks K_p^* and K_{un} tracks K_u^*, the less restrictive the assumption 2 will become. Choosing variableK_{pn}, K_{un} nominal gains in the control law (as done in the decoupled joint control algorithm), assumption #2 will be replaced by the following.

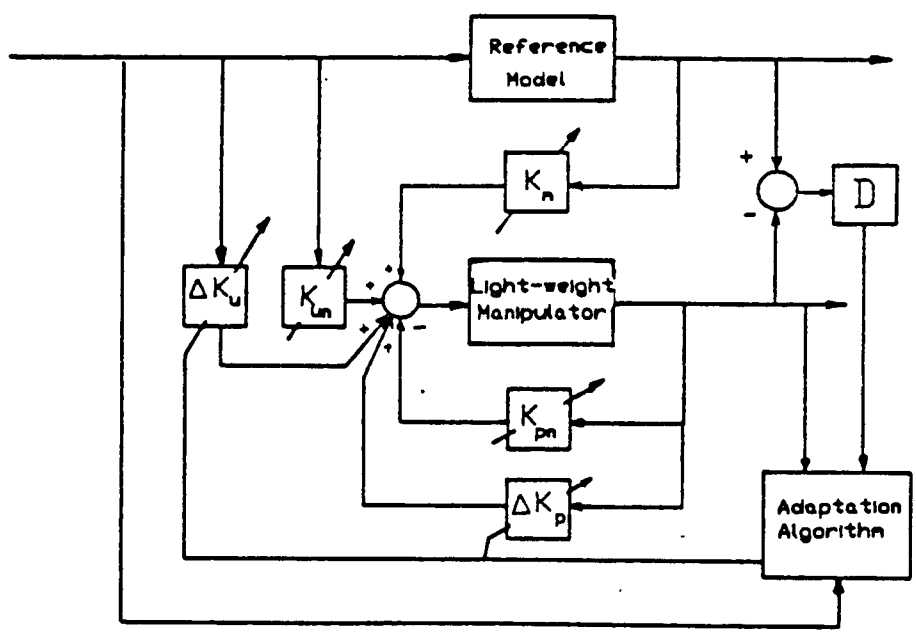


Fig.4.6.c) Generalized Inertia Matrix Based AMFC.

The previous assumption # 2 was:

> The difference between the reference model and the closed loop plant dynamics under *constant linear nominal control* should vary slower than the speed of adaptation.

The new assumption # 2 is:

> The difference between the reference model and the closed loop plant dynamics under *variable nonlinear nominal control* should vary slower than the speed of adaptation.

In other words, use of variable nominal feedback gains instead of constant nominal feedback gains make the adaptation algorithm job easier, thus extending its range of applications in robotics and other motion control systems.

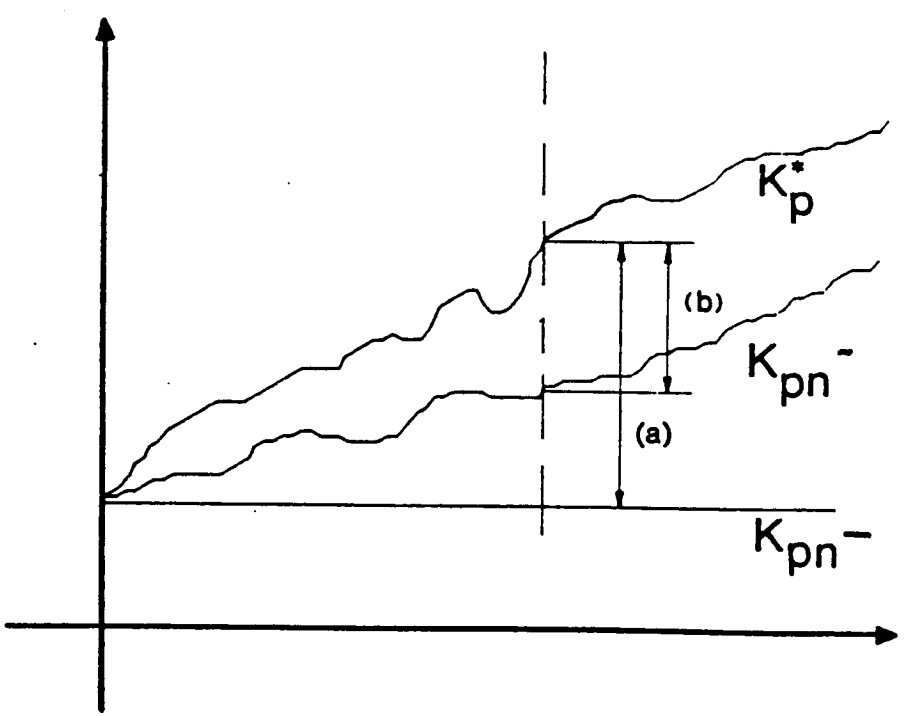


Fig.4.8 Feedback gain differences that must be taken care of by adaptation algorithm.

Furthermore, hyperstability based design results in a large class of possible adaptation laws (4.50). This may be an advantage from from an analysis point of view or exploring the possibilities. However, from the design point of view, too much freedom of choice may turn out to be a disadvantage if there is no clear reason or guidelines for choosing one adaptation law over another. It is generally agreed that more research is needed in direct methods of choosing adaptation algorithms from the admissible class. For example, currently integral and proportional adaptation are popular. The adaptation law contains a large number of parameters which can be chosen from a large class, and the choices are made in a trial and error or an ad-hoc basis.

Another contribution of this paper is to remove the uncertainty in the choice of adaptation parameters by utilizing the generalized inertia matrix, as discussed next.

4.3.4. Generalized Inertia Matrix Based AMFC †: Application to Flexible Manipulators

Consider the flexible manipulator model,

$$\begin{bmatrix} m_r(\theta,\delta) & m_{rf}(\theta,\delta) \\ m_{rf}^T(\theta,\delta) & m_f(\theta,\delta) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} f_r \\ f_f \end{bmatrix} + \begin{bmatrix} 0 \\ [K]\,\delta \end{bmatrix} + \begin{bmatrix} g_r \\ g_f \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \quad ; \tag{4.72}$$

$$m_r(\theta,\delta)\ddot{\theta} = u - [m_{rf}\ddot{\delta} + f_r + g_r] \tag{4.73}$$

† The generalized inertia matrix based AMFC algorithm, developed and presented in this thesis for the first time, is not limited to flexible arm applications, and can be readily applied to rigid manipulators without any modification.

$$u = \hat{g}_r + u_p \tag{4.74}$$

$$m_r(\theta, \delta)\ddot{\theta} = u_p + [m_{rf}\ddot{\delta} + f_r + (g_r - \hat{g}_r)] \tag{4.75}$$

During the gross motion, nonlinear terms and coupling from the flexible modes to the joint variable dynamics are treated as a disturbance and to be taken care of by the closed loop system robustness. The method does not require real time measurement or estimation information about the flexible states. This is an important advantage in terms of implementation simplicity.

Under the influence of a gravitational field, a flexible arm will deflect. Designing a control system which uses the static deflections as the nominal value for flexible states as opposed to zero would be more accurate.

Let the desired reference model be,

$$\begin{bmatrix} \dot{\theta}_m \\ \ddot{\theta}_m \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\Lambda_o & -\Lambda_1 \end{bmatrix} \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u_m \tag{4.76}$$

and the control law,

$$\begin{aligned} u_p &= -K_p x_p + K_u u_m + \overbrace{K_m x_m}^{0} \\ &= \underbrace{-K_{pn} x_p + K_{un} u_m}_{Nominal\ control} + \underbrace{\Delta K_p(e,t) x_p + \Delta K_u(e,t) u_m}_{Adaptation\ algorithm\ control\ action} \end{aligned} \tag{4.77}$$

Nominal control without the adaptation control can be chosen in the form (as used

by the computed torque method),

$$
\begin{aligned}
u_{pn} &= \hat{m}_r(\theta, \delta_{st}) \left[\ddot{\theta}_d + [c_{ii}](\dot{\theta}_d - \dot{\theta}) + [k_{ii}](\theta_d - \theta)\right] \\
&= \hat{m}_r(\theta, \delta_{st}) \left[\ddot{\theta}_d + [c_{ii}]\dot{\theta}_d + [k_{ii}]\theta_d\right] + \\
&- \hat{m}_r(\theta, \delta_{st}) \left[[c_{ii}]\dot{\theta} + [k_{ii}]\theta\right]
\end{aligned}
\tag{4.78}
$$

using (4.25)

$$
\begin{aligned}
u_{pn} &= \hat{m}_r(\theta, \delta_{st}) u_m + \hat{m}_r(\theta, \delta_{st}) \left[[c_{ii} - \Lambda_1]\dot{\theta}_m + [k_{ii} - \Lambda_0]\theta_m + [k_{ii}]\theta_0\right] \\
&- \hat{m}_r(\theta, \delta_{st}) \left[[c_{ii}]\dot{\theta} + [k_{ii}]\theta\right]
\end{aligned}
\tag{4.79}
$$

The nominal gains for the adaptive model following control algorithm based on the generalized inertia matrix is given by,

$$
\begin{aligned}
K_{un} &= \hat{m}_r(\theta, \delta_{st}) \\
K_{pn} &= \hat{m}_r(\theta, \delta_{st}) \Big[[k_{ii}] \ , \ [c_{ii}]\Big] \\
K_{mn} &= \hat{m}_r(\theta, \delta_{st}) \Big[[k_{ii}] \ - \ \Lambda_0 \ , \ [c_{ii}] - \ \Lambda_1]\Big]
\end{aligned}
\tag{4.80}
$$

If error dynamics eigenvalues are equal to those of the reference model, then $k_{ii} = \Lambda_0, c_{ii} = \Lambda_1 \Rightarrow K_{mn} = 0$.

The $\hat{m}_r(\theta, \delta_{st})$ term in the control algorithm is the key for decoupled control of joints. The adaptation algorithm should be designed such that when added to the nominal control vector u_{pn}, the decoupled nature of the control is preserved.

The adaptive part of the control is:

$$
\Delta K_p = \underbrace{\int_0^t F_{p1} v [G_{p1} x_p]^T d\tau}_{Integral\ adaptation; \Delta K_{pi}} + \underbrace{F_{p2} v [G_{p2} x_p]^T}_{Proportional\ Adaptation; \Delta K_{pp}} \tag{4.81.a}
$$

$$\Delta K_u = \underbrace{\int_0^t F_{u1} v [G_{u1} u_m]^T d\tau}_{Integral\ adaptation;\Delta K_{ui}} + \underbrace{F_{u2} v [G_{u2} u_m]^T}_{Proportional\ Adaptation;\Delta K_{up}} \qquad (4.81.b)$$

Any positive definite matrix of appropriate dimension for F_{p1}, F_{p2}, G_{p1}, G_{p2}, F_{u1}, F_{u2}, G_{u1}, G_{u2} would suffice (but is not necessary) to guarantee the global asymptotic stability of the control system with an appropriate output filter. For an n-degree of freedom system with m- number of inputs; F_{p1}, F_{p2}, F_{u1}, F_{u2}, G_{u1}, G_{u2}, $\in R^{(mXm)}, and\ G_{p1}, G_{p2} \in R^{(nXn)}$. There are too many design parameters which can be chosen arbitrarily from a large admissible class. Neither the hyperstability based design nor Lyapunov methods give any guidelines for the selection of the elements of these matrices. As the system dimension increases, finding appropriate adaptation algorithm parameters becomes a more serious design problem.

The final contribution of our AMFC design approach solves that problem. Since decoupled control calls for the use of the generalized inertia matrix, one should utilize this fact in the adaptation algorithm to direct the adaptation algorithm in the right direction. The following adaptation algorithm, which uses the generalized inertia matrix, will guarantee the global asymptotic stability of the closed loop system.

$$\begin{aligned} \Delta K_p &= \Delta K_{pi} + \Delta K_{pp} \\ &= \int_o^t p_{pi}\ \hat{m}_r(\theta_o, \delta_{st})\ v\ x_p^T\ d\tau + p_{pp} \hat{m}_r(\theta_o, \delta_{st})\ v\ x_p^T \end{aligned} \qquad (4.82.a)$$

$$\begin{aligned} \Delta K_u &= \Delta K_{ui} + \Delta K_{up} \\ &= \int_o^t p_{ui} \hat{m}_r(\theta_o, \delta_{st})\ v\ u_m^T\ d\tau + p_{up} \hat{m}_r(\theta_o, \delta_{st})\ v\ u_m^T \end{aligned} \qquad (4.82.a)$$

The generalized inertia matrix based AMFC algorithm described by (4.77), (4.80), and (4.82) has the following advantages over previous algorithms:

1. The use of the GIM immediately solves the magnitude selection problem of the adaptation algorithm, for it is naturally compatible with the problem.

2. The number of design parameters for integral adaptation is only 2, for integral plus proportional adaptation 4, no matter how many degrees of freedom the system has. Thus the design problem of finding the good adaptation parameters becomes much simpler.

3. Utilizing the GIM as an integral part of adaptation improves the decoupled response of joint variables (Fig. 4.19.a-d, 4.23.a-b).

4. The use of variable nominal gains results in less restrictive conditions on the applications of AMFC to nonlinear systems.

4.4. Comparative Simulations and Discussion of Results

In the following discussion, the *performance* of a control algorithm refers to the *speed* and *accuracy*, and *robustness* with respect to payload variations that the control algorithm can provide for a given manipulator. The performance evaluation of one algorithm relative to another is based on the maximum bandwidth that the controller can provide with predetermined or comparable accuracy over a range of payload variations, i.e. 25% (payload to robot mass ratio).

Robustness of an algorithm is tested by calculating the real time control vector based on a payload value zero, and applying it to a model with a payload of weight equal to 25

4.4.1. Simulations with a Rigid Manipulator Model

The objective of the simulations of the CTM and DJC on the rigid model are as follows:

1. Quantify how important the nonlinearities (coriolis and centrifugal forces) become relative to gravity forces as the speed of motion increases,

2. Determine the performance of the CTM and DJC on a rigid manipulator so that the results can be used as a basis for comparison of the flexible manipulator. Furthermore, show the effect of high gain feedback on the tracking and robustness performance as discussed in the development of the methods (Section 4.2).

Three different motions are simulated. The desired motion trajectories are

generated by a reference model with a step input command signal. In all cases, the reference model is a decoupled set of second order linear model with a damping ratio of $\xi_i = 0.707$ and natural frequency for cases (a) 2.75 rad/sec, (b) 5.5 rad/sec, (c) 11.0 rad/sec . Desired motions and resultant torque histories are shown in Fig.4.9.a,b,c,d.

As shown in Fig.4.9.e,f, the nonlinear forces become much more important compared to gravitational forces as the speed of motion increases. For relatively slow motions, nonlinear terms are small, and neglecting these in the controller design may result in a satisfactory closed loop performance. However, if operation speed increases the nonlinear forces become dominant and cannot be neglected.

To answer question 2, the CTM and DJC are simulated tracking a desired motion generated by a reference model. The reference model has a damping ratio $\xi_i = 1.0$, and a natural frequency $w_{mi} = 2.75 rad/sec$, with step command input signal. The rigid arm under the CTM and DJC control algorithms is simulated for three different closed loop eigenvalues as shown in table 4.1.

$w_{ni}(rad/sec)$	$w_{mi}(rad/sec)$
2.75	2.75
5.5	2.75
11.0	2.75

Table 4.1 Closed loop system and desired motion bandwidth.

Under perfect parameter information and no disturbance conditions, the CTM tracks the reference motion perfectly, as expected. The DJC does not compensate for the nonlinear terms, thus even under perfect conditions there will be

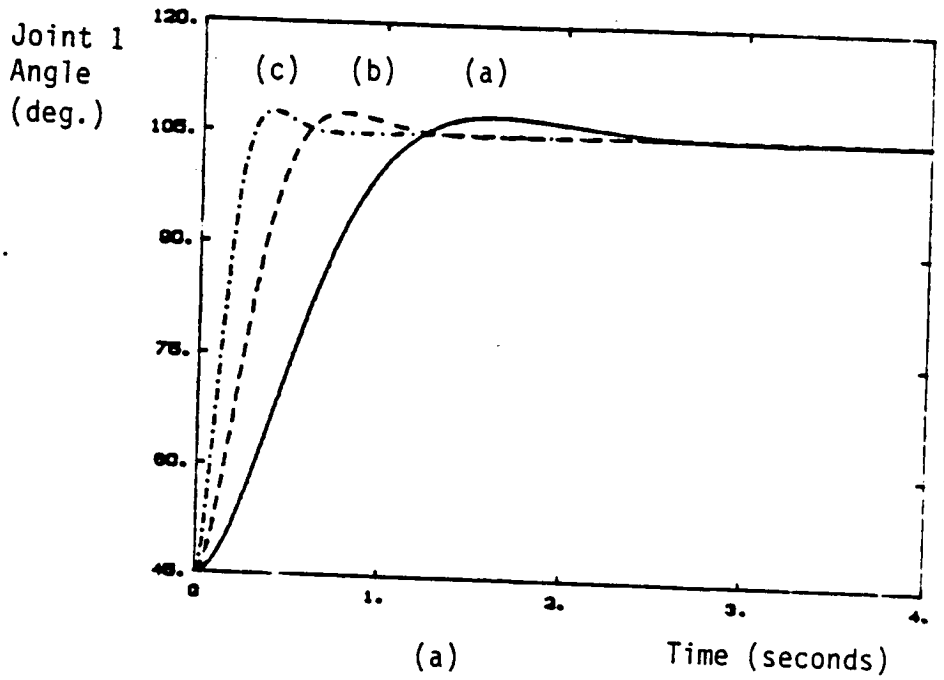
tracking errors. From figures 4.10 a-d, it is clear that tracking errors get smaller as closed loop control system bandwidth versus desired motion bandwidth ratio gets larger, ($w_{ni}/w_{mi} \gg 1$). This is the effect of high gain feedback control on the system performance. High gain feedback reduces the tracking errors and increases the robustness of the system, but the controller bandwidth must much larger than the bandwidth of the model being tracked.

The same reference model and controllers are simulated to test the robustness performance. The control algorithm makes calculations based on a payload value of zero, $m_p = 0.0kg.$, while the actual manipulator (model) has $m_p = 2.0$ kg payload (payload/robot mass = 25 %). Joint variable responses are shown in Fig.4.11.a-d and 4.12.a-d. Clearly, the cases of $w_{ni}/w_{mi} = 1.0$ (relatively fast motion with respect to controller) are unacceptably bad for both control methods. However, as the w_{ni}/w_{mi} ratio increases (high gain feedback), the system is able to compensate for the payload uncertainty. Yet, in the steady state, when the speed gets very low, the effect of nonlinearities becomes negligible and gravity acts as a constant disturbance due to the payload misinformation, resulting in a finite steady state error. The steady state error is reduced by higher feedback gains as predicted in section 4.2. Also, notice that in steady state the CTM and the DJC are equivalent and have same steady-state error, for accelerations and nonlinearities are negligible.

Finally, the rigid model is simulated with $w_{ni}/w_{mi} = 1.0$ with reference model having $5.5rad/sec$ natural frequency, in order to see the effect of higher speeds on robustness performance. The response is not good at all as seen from figures 4.13.a-d, indicating the need for high gain feedback if conditions are not perfect. Figures 4.14.a-d show the torque histories corresponding to figures 4.11

and 4.12.

In model following sense, if a non-adaptive control algorithm is designed to follow a reference model with a step input command signal, $K_m \neq 0$ must be for robustness. The case of $w_{ni}/w_{mi} = 1$ is equivalent to $K_m = 0$ case (see fig. 4.6.a, sections 4.2.3. and 4.3.1).



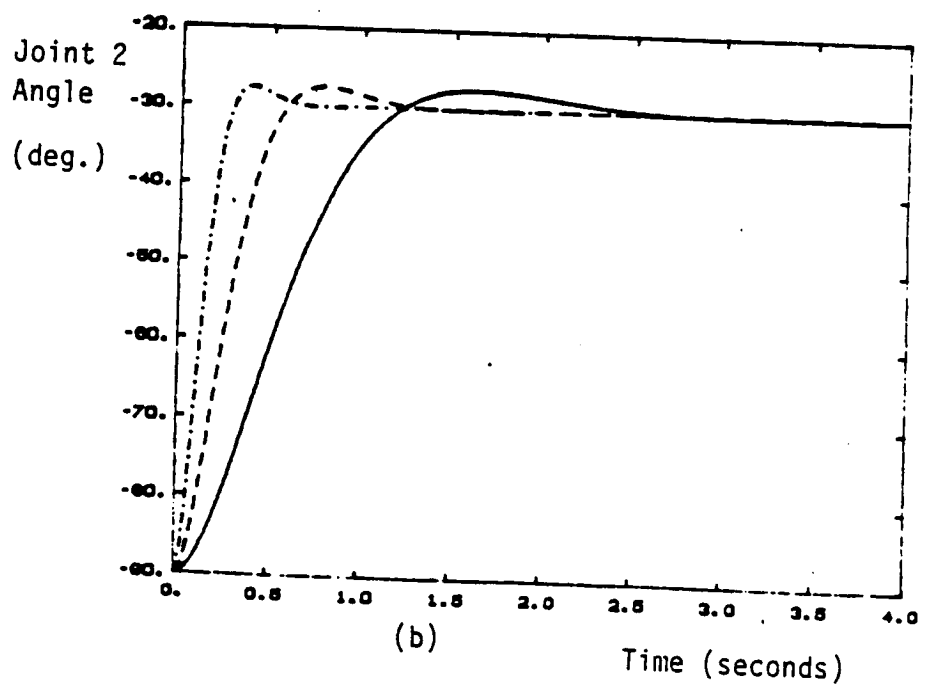
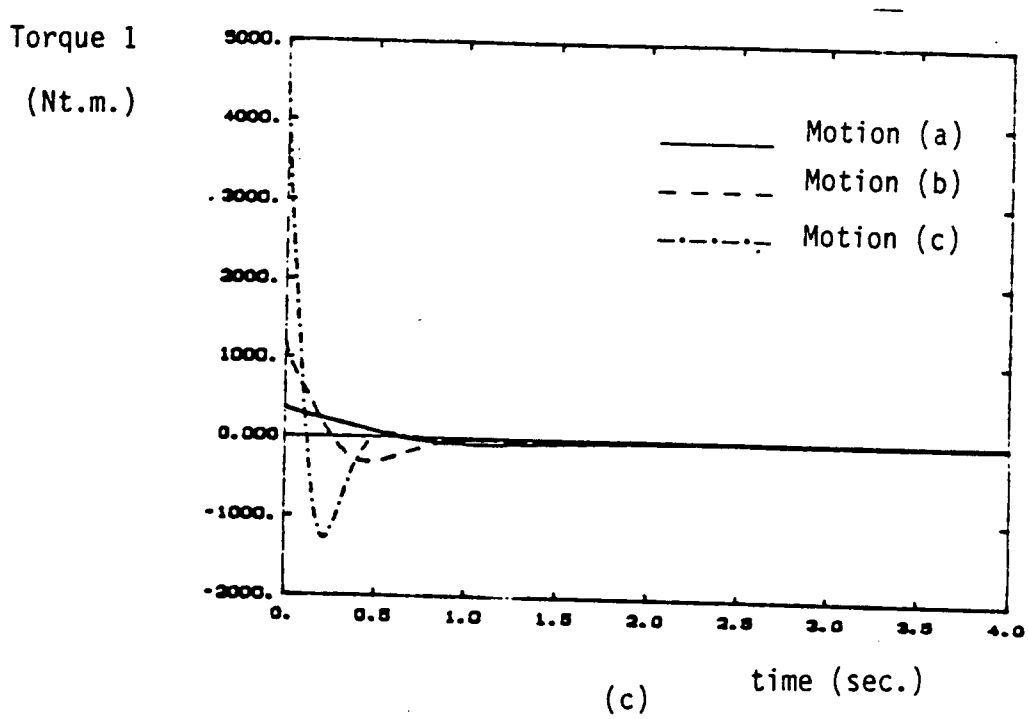


Fig.4.9.a-b Desired trajectories: a) joint 1, b) joint 2



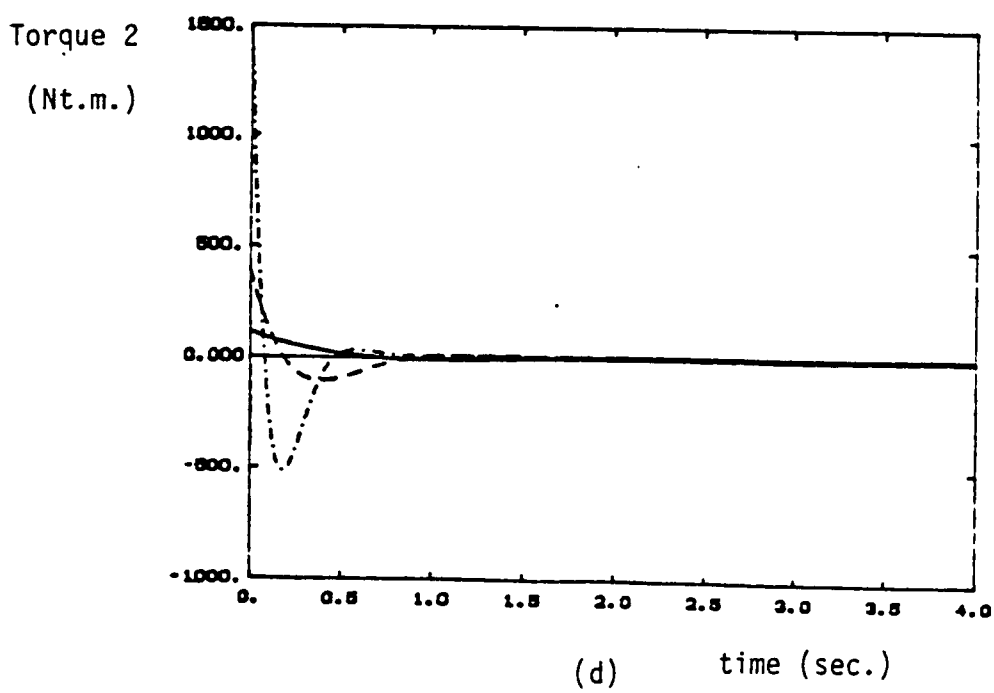
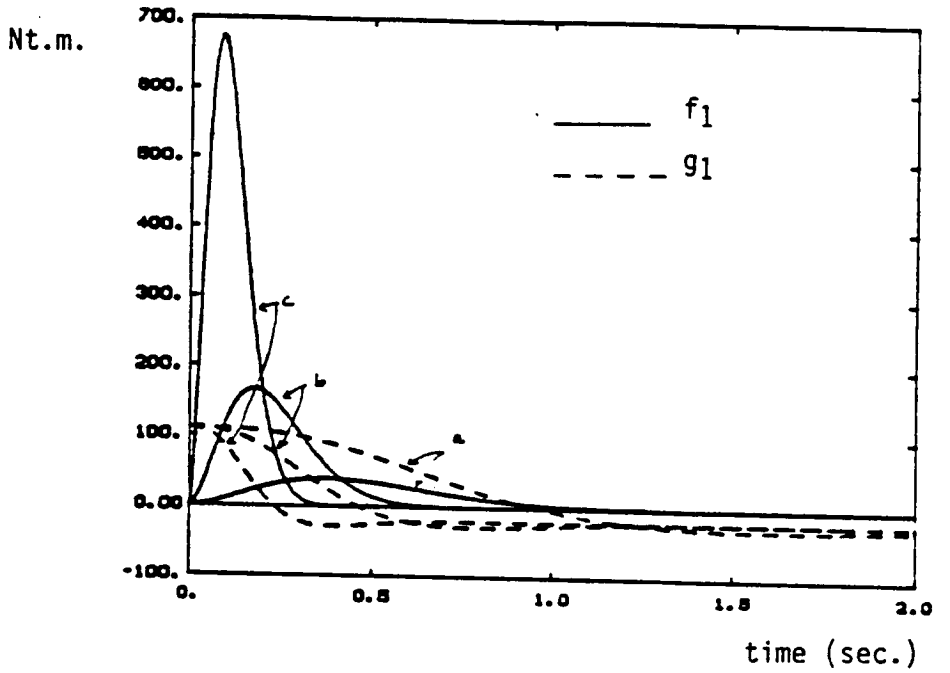
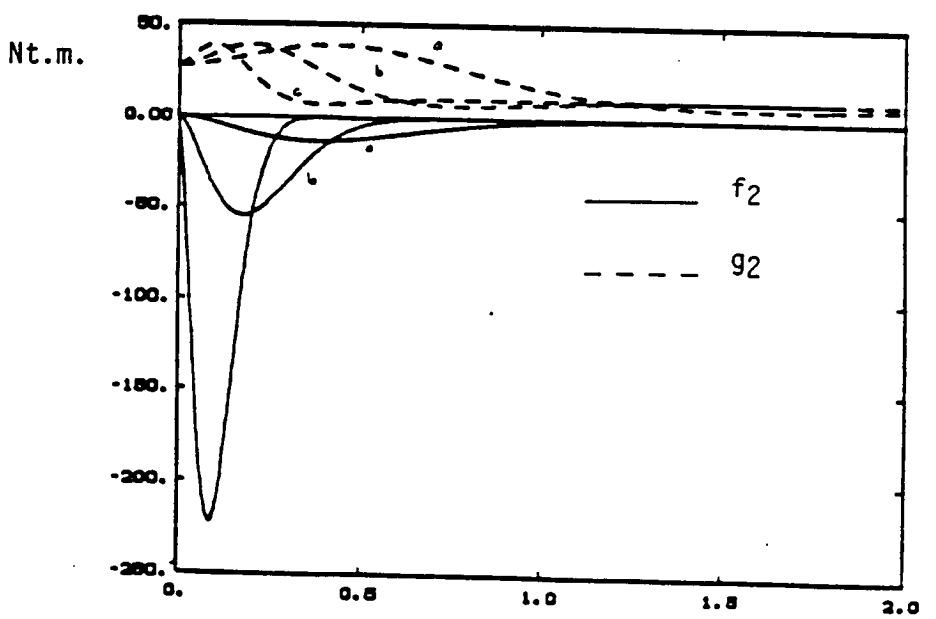


Fig.4.9.c-dCTM resultant torque history c) jont 1 d) joint 2.

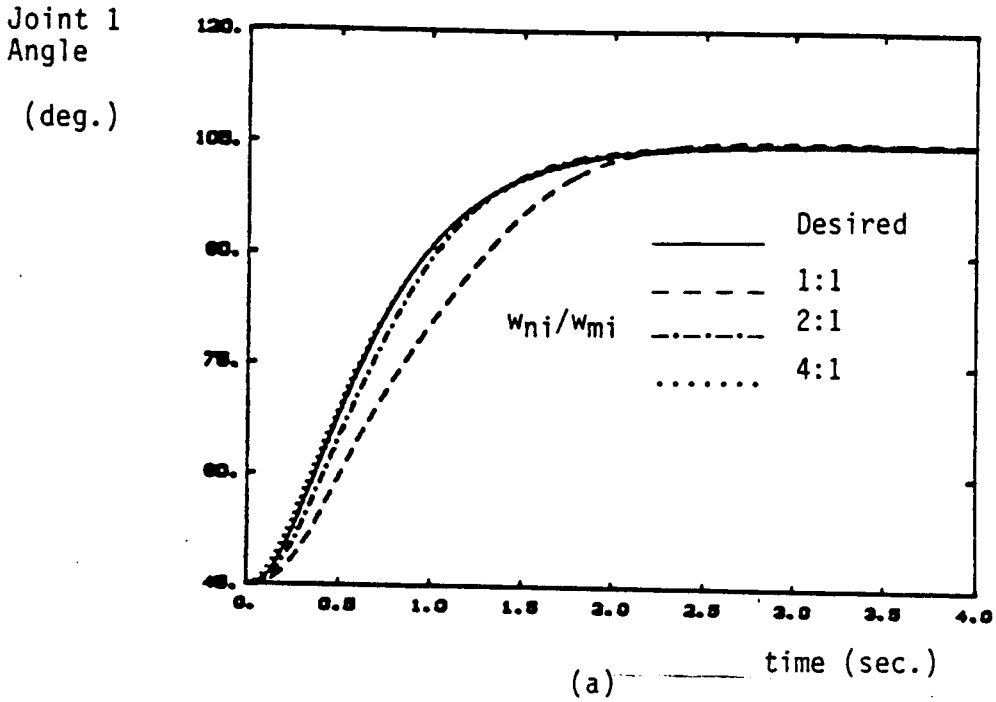


(e)



(f)

Fig.4.9.e-f Relative importance of nonlinear forces and gravitational forces along the motion. e) joint 1, f) joint 2.



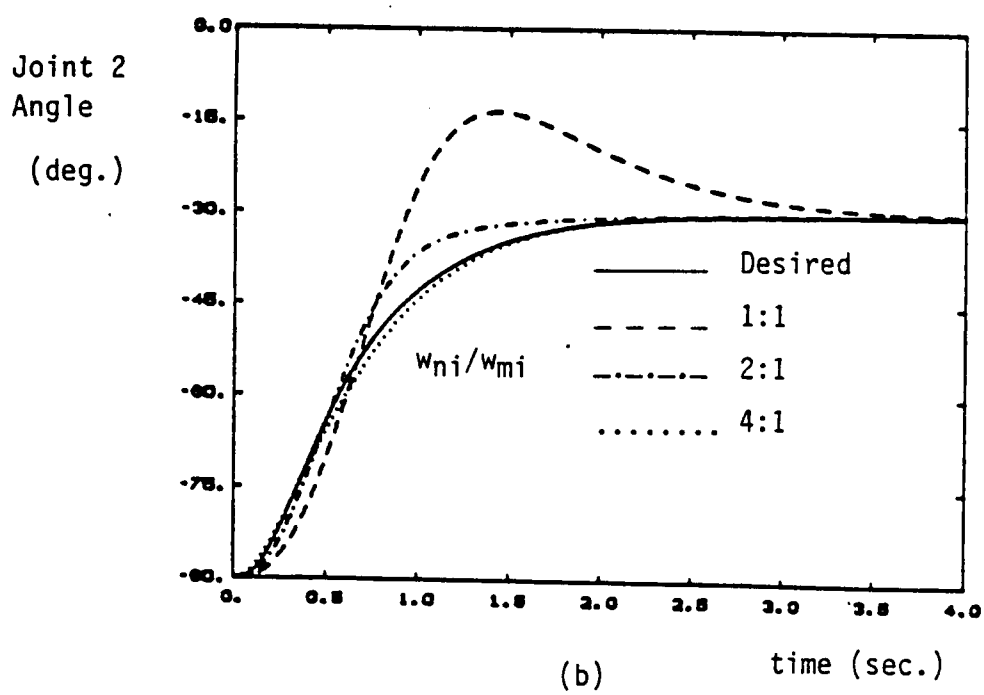
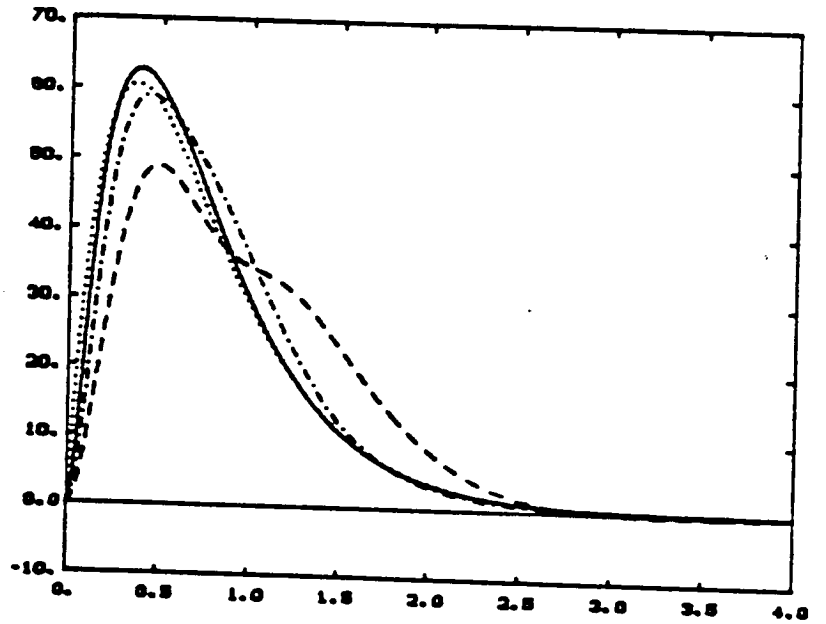
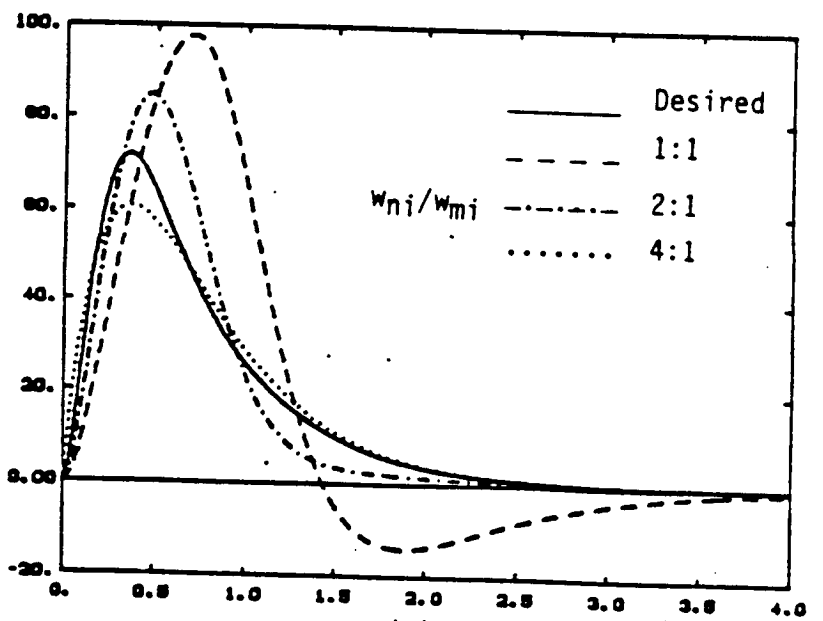


Fig.4.10.a-b Decoupled joint control performance on the rigid model- the effect of high gain feedback: a) joint 1, b) joint 2,

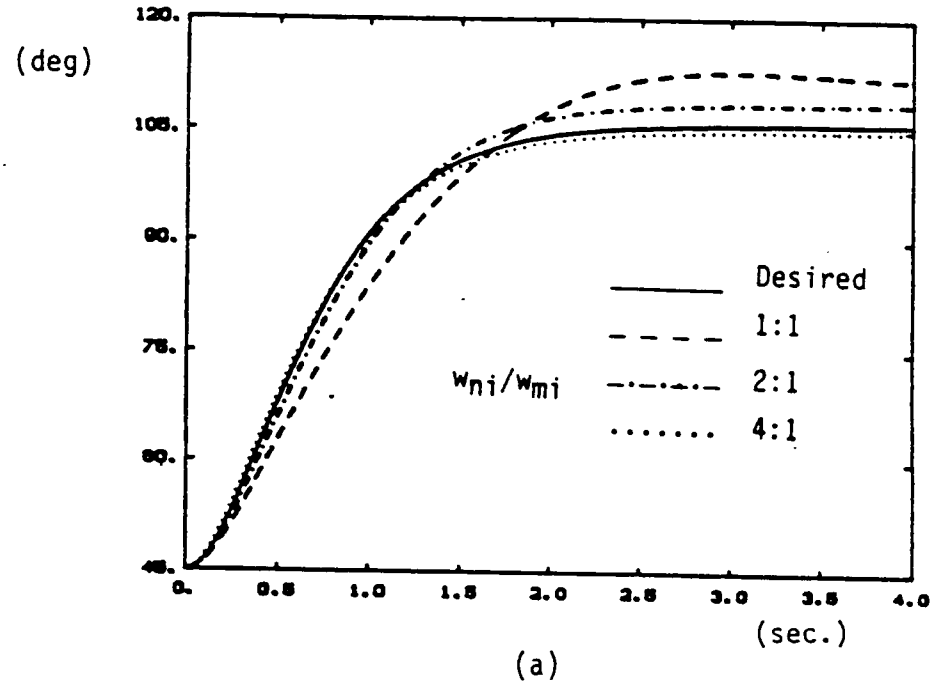


(c)

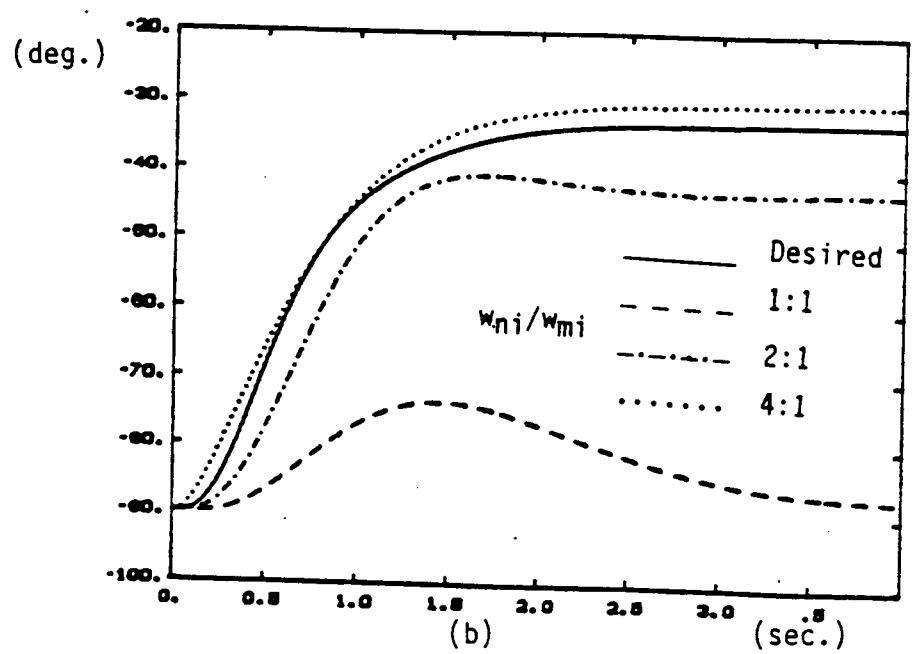


(d)

Fig.4.10.c-d Decoupled joint control performance on the rigid model- the effect of high gain feedback: c) joint 1 speed, d) joint 2 speed.

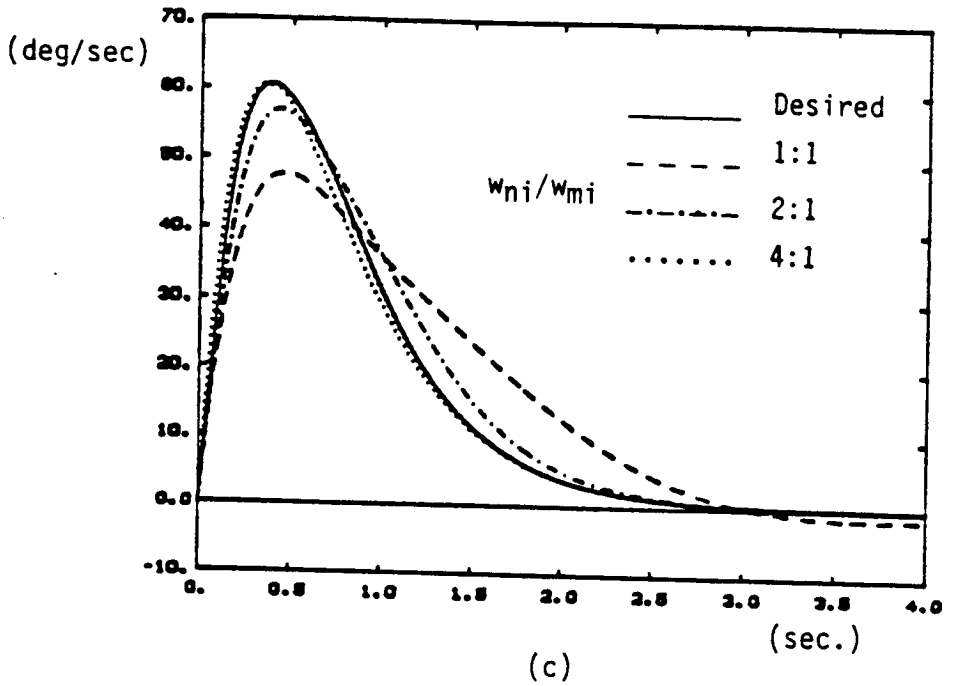


(a)

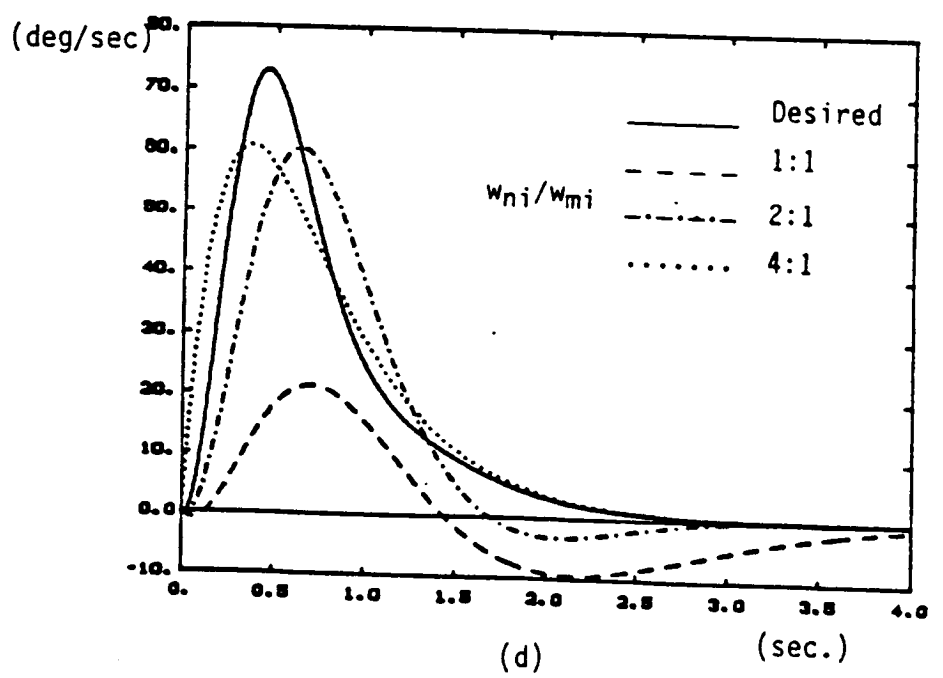


(b)

Fig.4.11.a-b Robustness of CTM with respect to payload variations, and the effect of high gain feedback: a) joint 1, b) joint 2

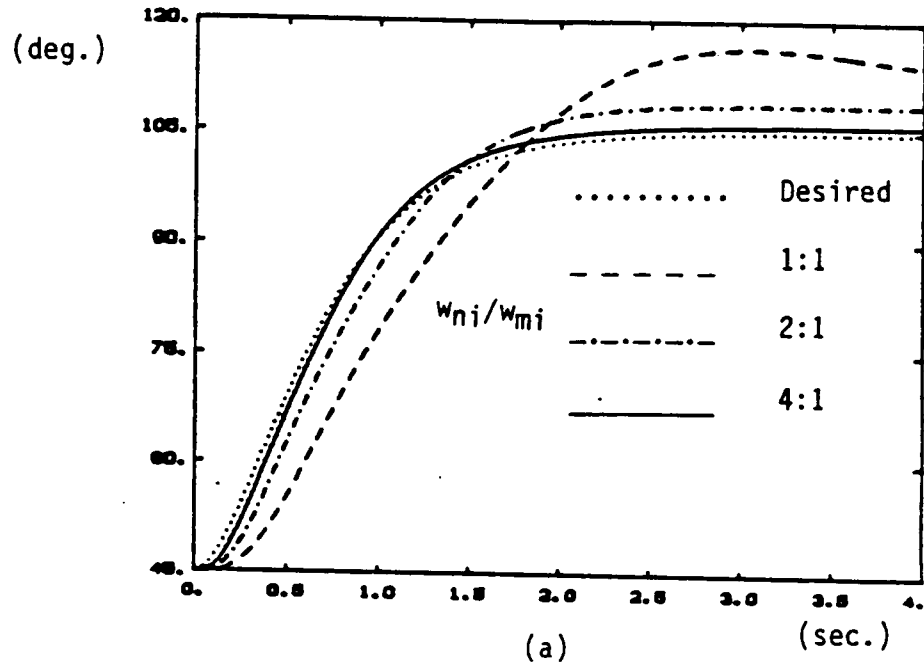


(c)



(d)

Fig.4.11.c-d Robustness of CTM with respect to payload variations, and the effect of high gain feedback: c) joint 1 speed, d) joint 2 speed.



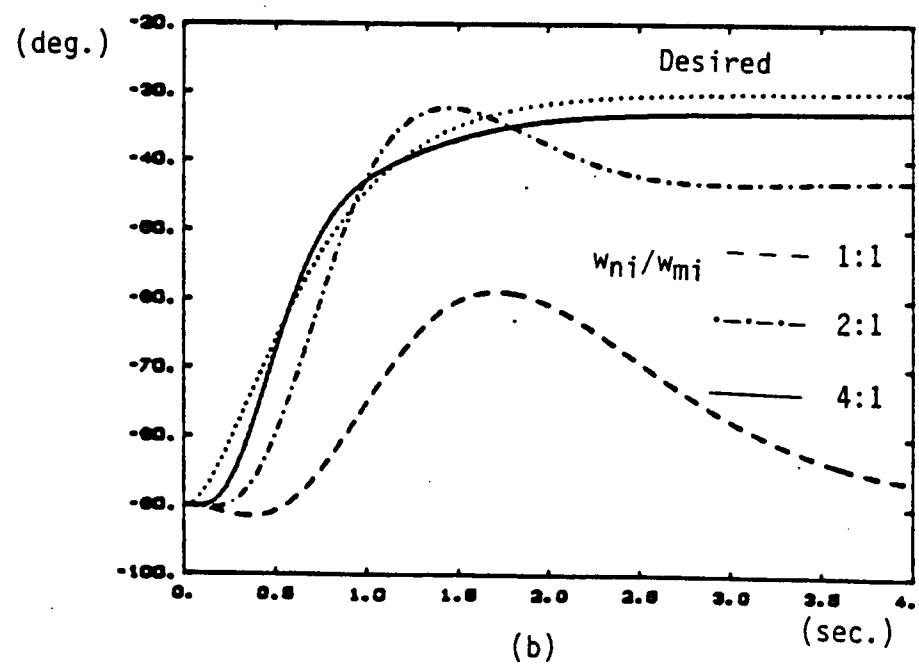
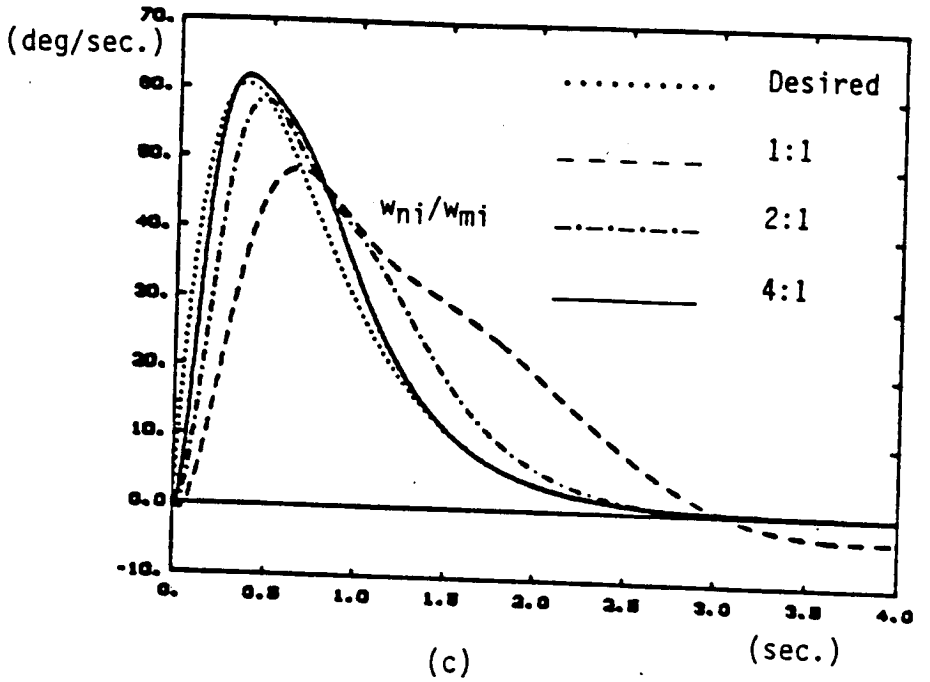


Fig.4.12.a-b Robustness of DJC with respect to payload variations, and the effect of high gain feedback: a) joint 1, b) joint 2.



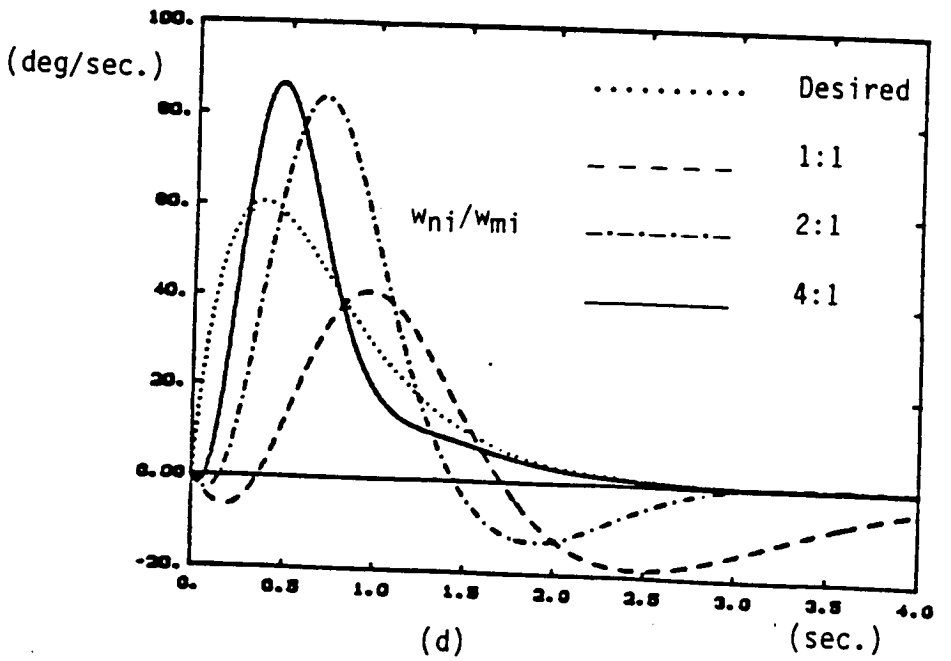
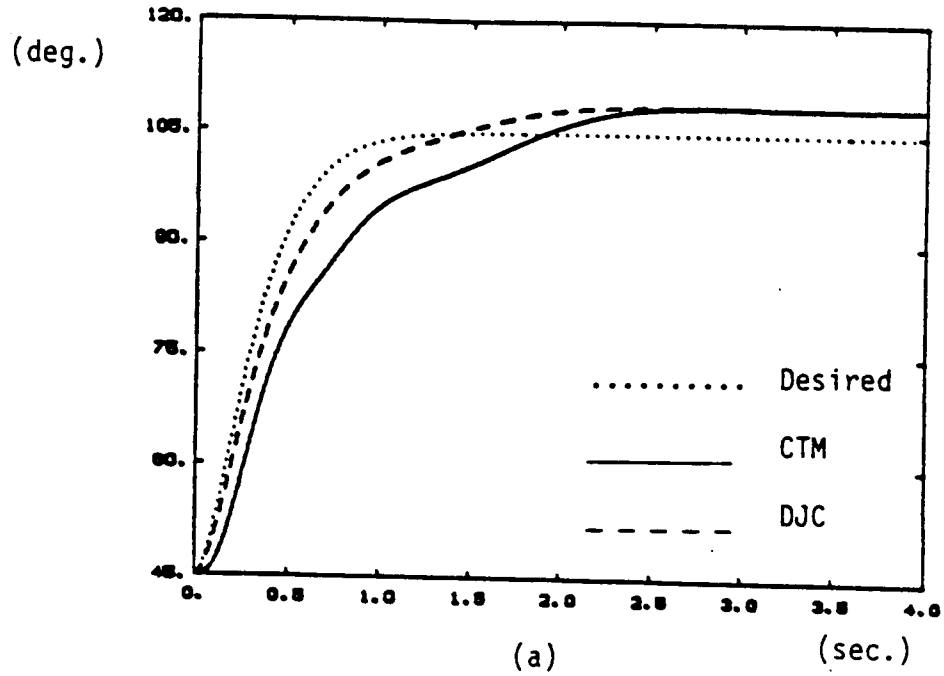


Fig.4.12.c-d Robustness of DJC with respect to payload variations c) joint 1 speed, d) joint 2 speed.



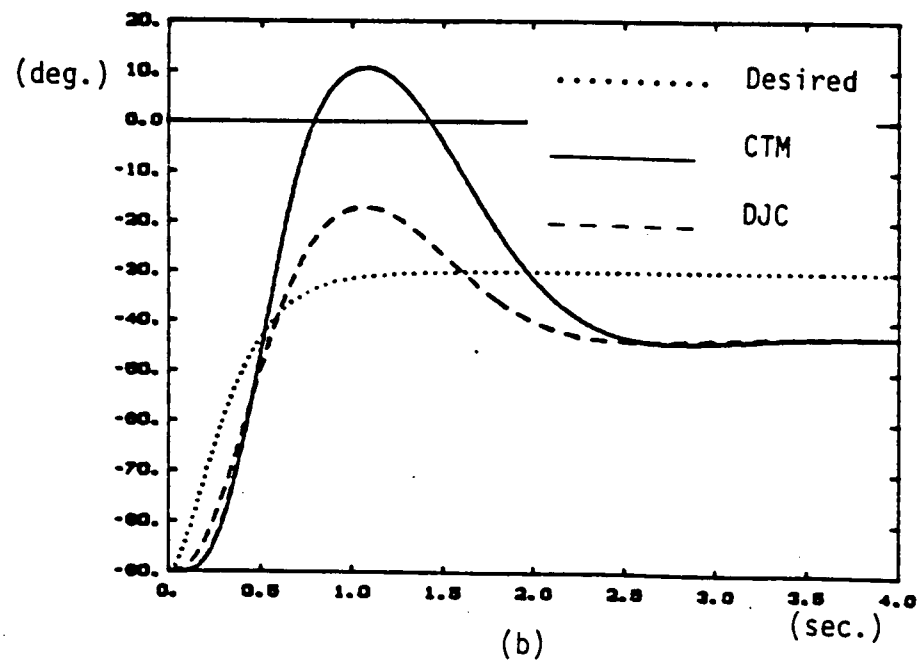
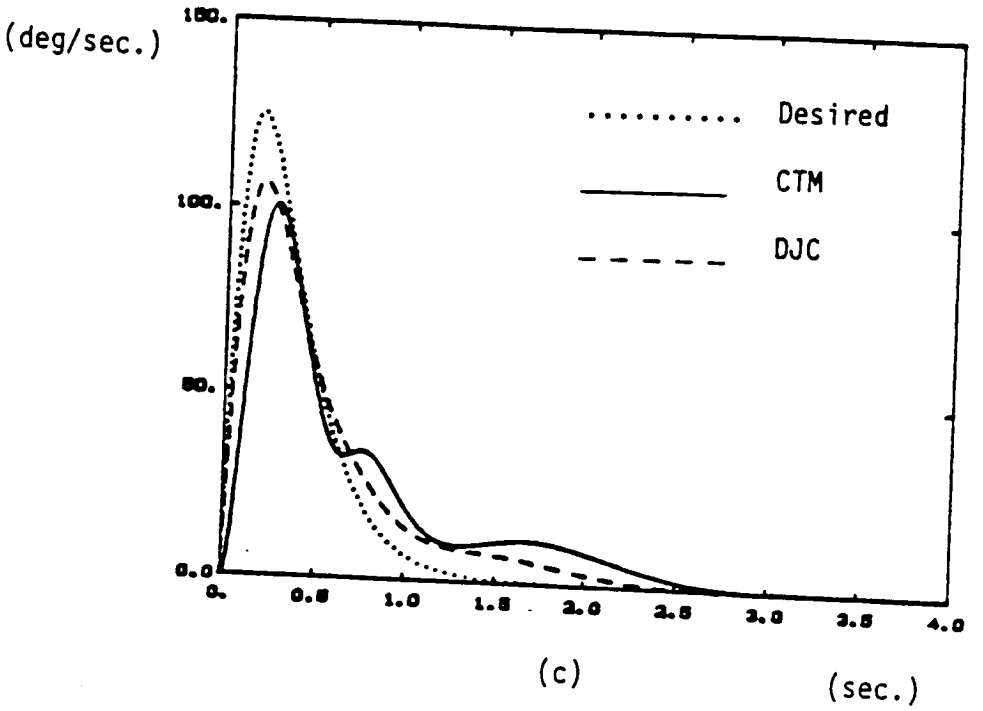


Fig.4.13.a-b Robustness of CTM and DJC with respect to payload variations for relatively fast motions: a) joint 1 angle, b) joint 2 angle.



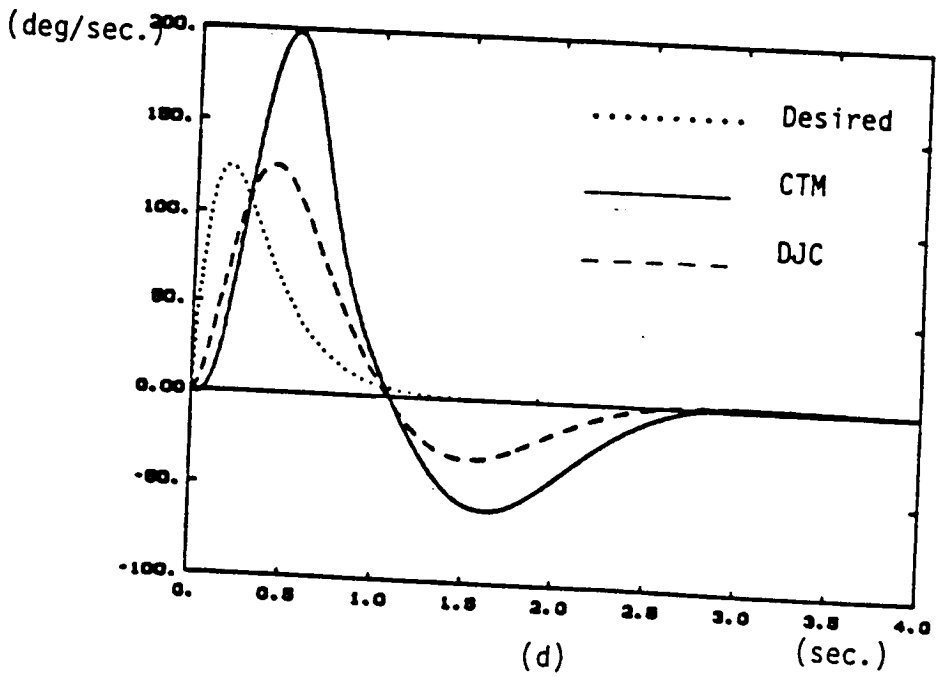
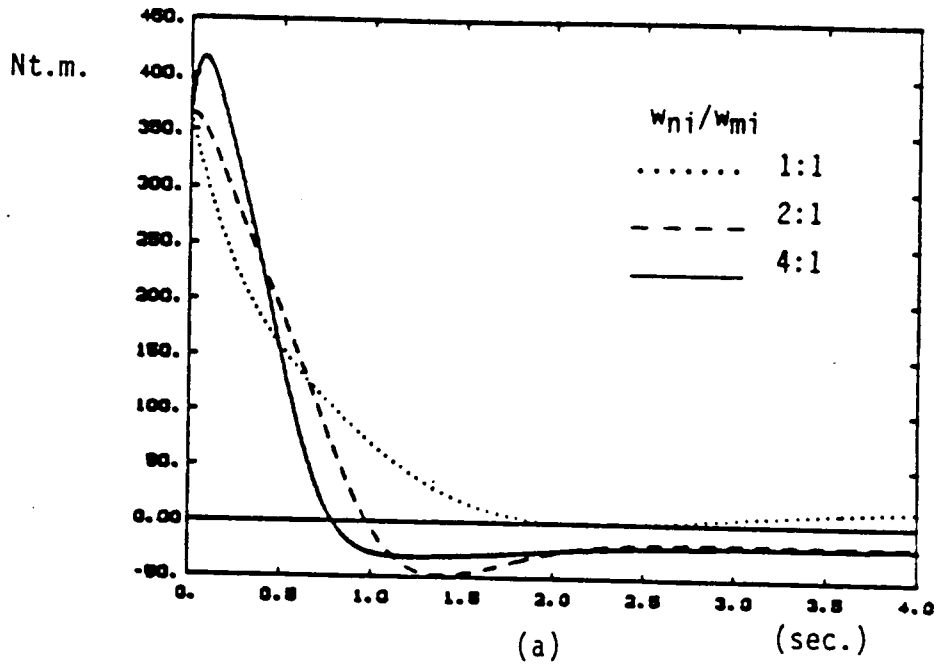


Fig.4.13.c-d Robustness of CTM and DJC with respect to payload variations for relatively fast motions: c) joint 1 speed, d) joint 2 speed.



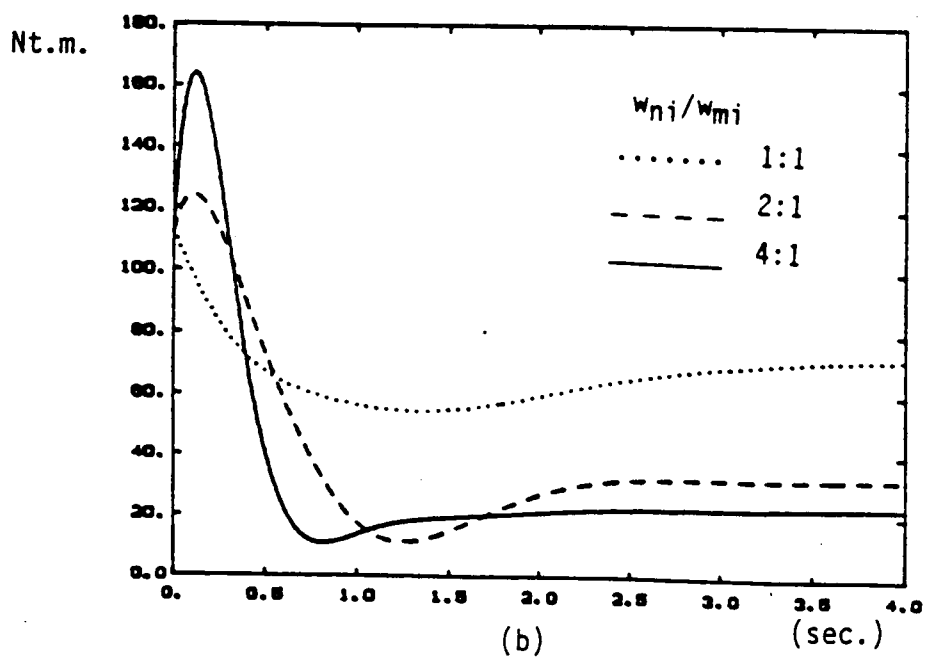
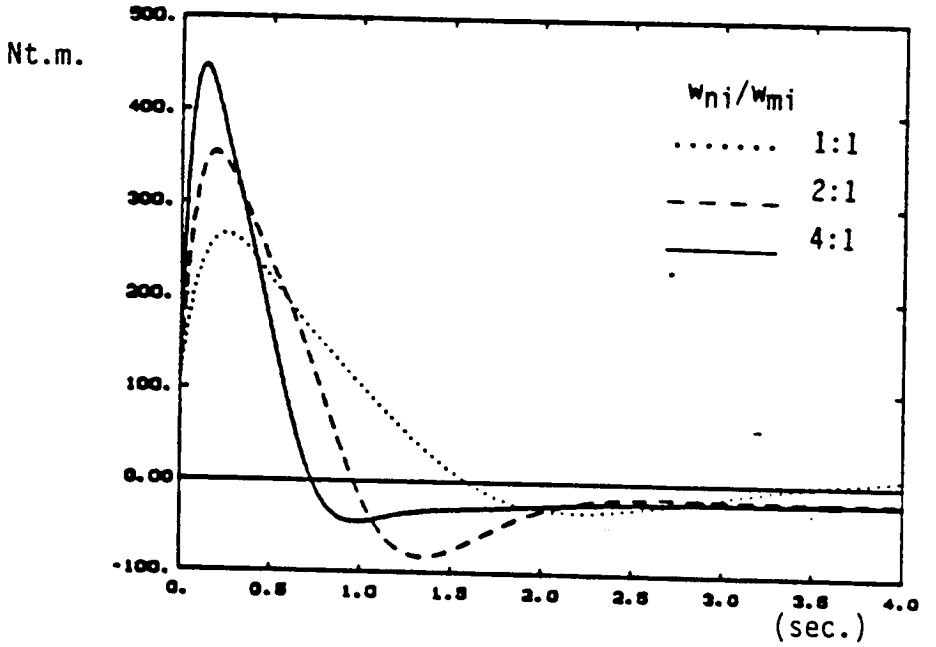
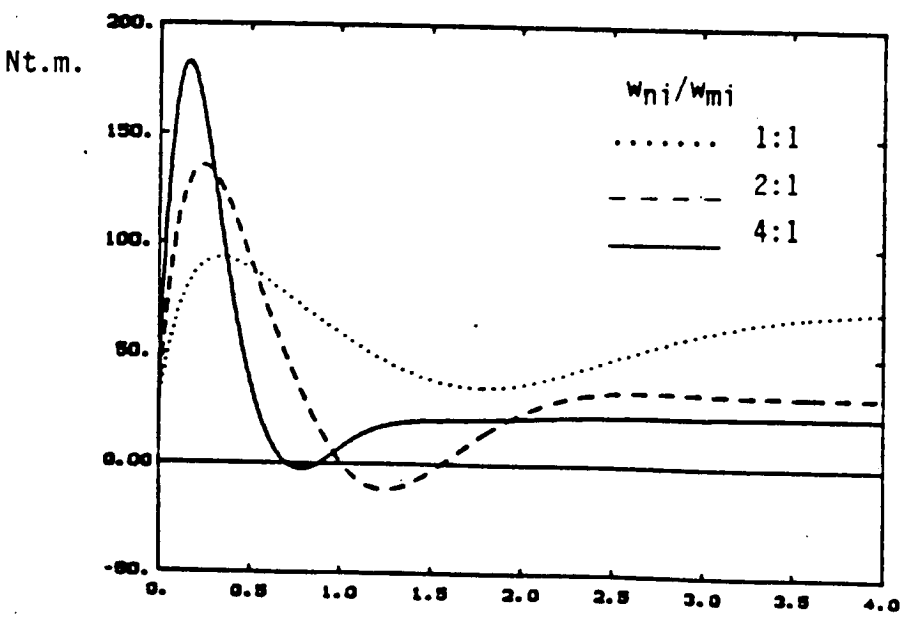


Fig.4.14.a-b Torque histories corresponding to motions indicated in Fig. 4.11



(c)



(d)

Fig.4.14.c-d Torque histories corresponding to motions indicated in Fig. 4.12

4.4.2. Comparative Control Algorithm Simulations on Rigid and Flexible Manipulator Model

The CTM and DJC have been simulated on a flexible and a corresponding rigid manipulator with the following objectives:

1. Compare the performance of the two control algorithms and determine the effect of arm flexibility on the performance.

2. Quantitatively determine when the rigid model based non-adaptive control algorithms can be successfully applied to flexible arms without arm flexibility being a significant factor.

3. Quantitatively determine when arm flexibility becomes important and what limitations are imposed on the closed loop performance, using nonlinear model simulations. Furthermore, compare the results with the results of linear analysis and see if they agree.

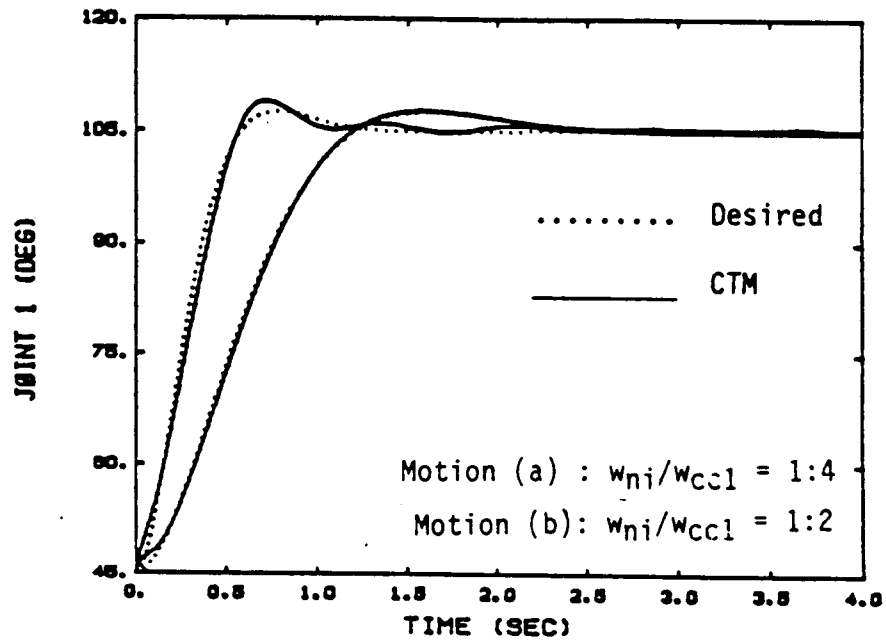
Figures 4.15.a-d show the CTM simulations. As the desired motion speed increases and the bandwidth of the motion gets closer to the clamped-clamped frequency of the arm, the tracking performance deteriorates [Fig. 4.16 a-c]. Results of linear analysis for fine motion have predicted (Chapter 3) that a computed torque type control which uses only joint feedback could achieve closed loop system bandwidth up to 1/2 of the lowest frequency of the arm. However, this conclusion is valid only within the limitations of linear analysis (small, fine motions). When such controllers are applied to fast, large motions, nonlinear effects further restrict the performance limits. The motions simulated here are *fast* with respect to both arm

flexibility and the nonlinearities. As seen in figures 4.9.e-f, at these speeds nonlinear forces become dominant, thus motions are fast with respect to the nonlinearities. Since the desired motion bandwidths $(a)2.75, b)5.5, c)11.rad/sec)$ are close to the arm lowest frequency $(11.34rad/sec.)$ the motions (b,c) are fast with respect to arm flexibility too. As shown in Fig. 4.15.c-e, when the arm is forced to follow a motion with bandwidth equal to w_{cc1}, the performance is unacceptable due to large deflections of the arm. When the arm reaches the final position, oscillations continue (Fig. 3.11) and energy is not being absorbed quickly from the lightly damped flexible modes due to high stiffness of joints. The nonlinear simulations seem to agree with the rule of thumb given by Book [A20], which says that joint variable feedback control algorithm should not attempt a closed loop bandwidth of more than 1/4 $\sim$ 1/2 w_{cc1}, lowest natural frequency of the arm when all joints clamped and links extended.

If one is also concerned with the deflections along the motions, further restrictions must be imposed on the speed to avoid excessive deflections. Damping ratio of the modes does not indicate reduced deflections, but does indicate relative stability and rapid damping of residual vibrations. The magnitude of deflections during gross motion is related to the acceleration profiles and maximum speed of motion (Fig. 4.11.a-c).

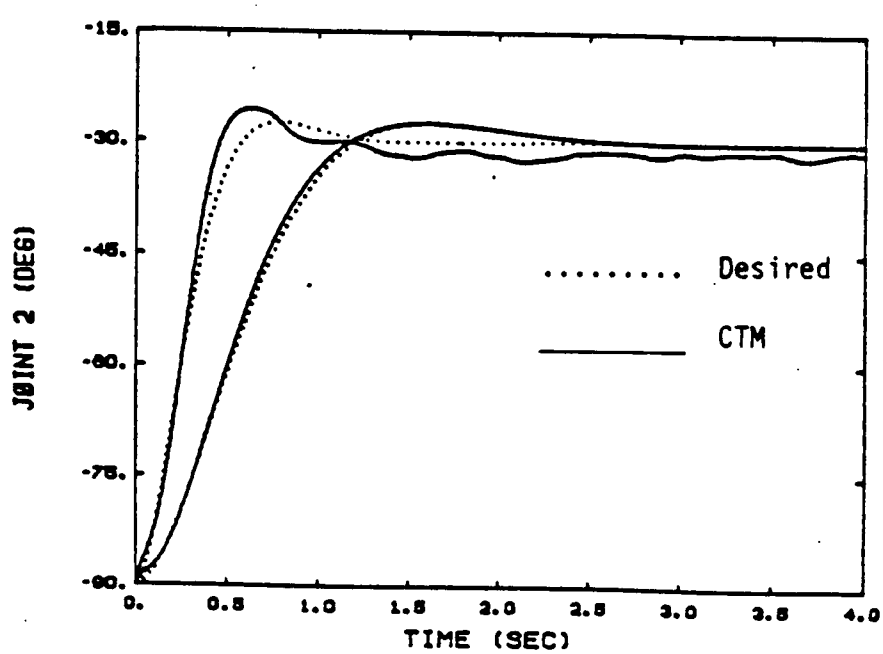
Finally, the following question will be discussed: when can the rigid model based joint variable controllers be used on flexible arms and flexibility would not be a problem? The answer may depend on the arm kinematic structure and the operating conditions. Nonetheless, for serial kinematic chain structured manipulators, a closed loop arm bandwidth of approximately 1/4 of w_{cc1} can be achieved for large motions

without arm flexibility being a significant problem.

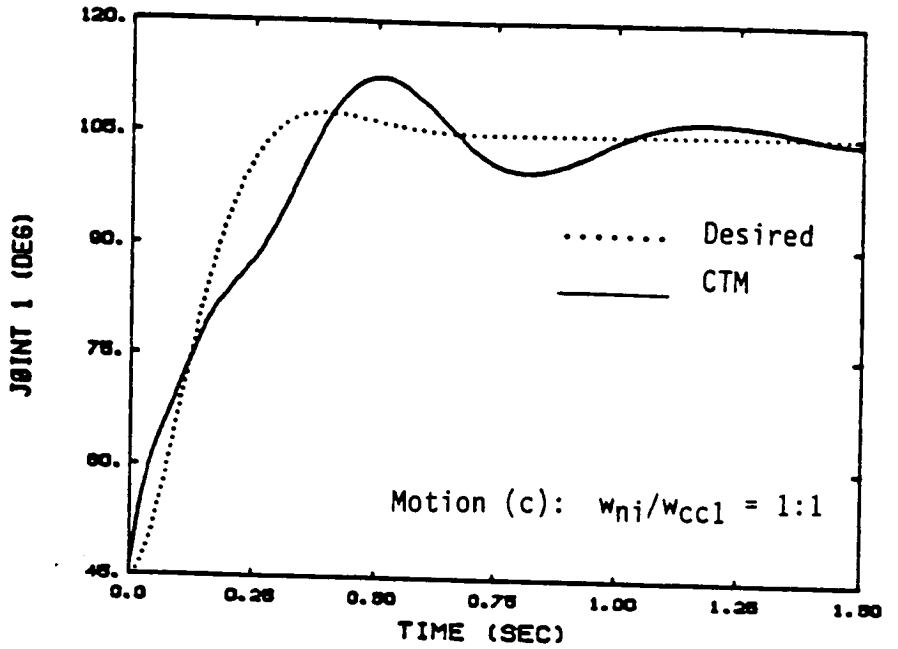


(a)

Fig.4.15.a-b Flexible model repsonse, controller: CTM,
a) joint 1 responses, b) joint 2 responses.

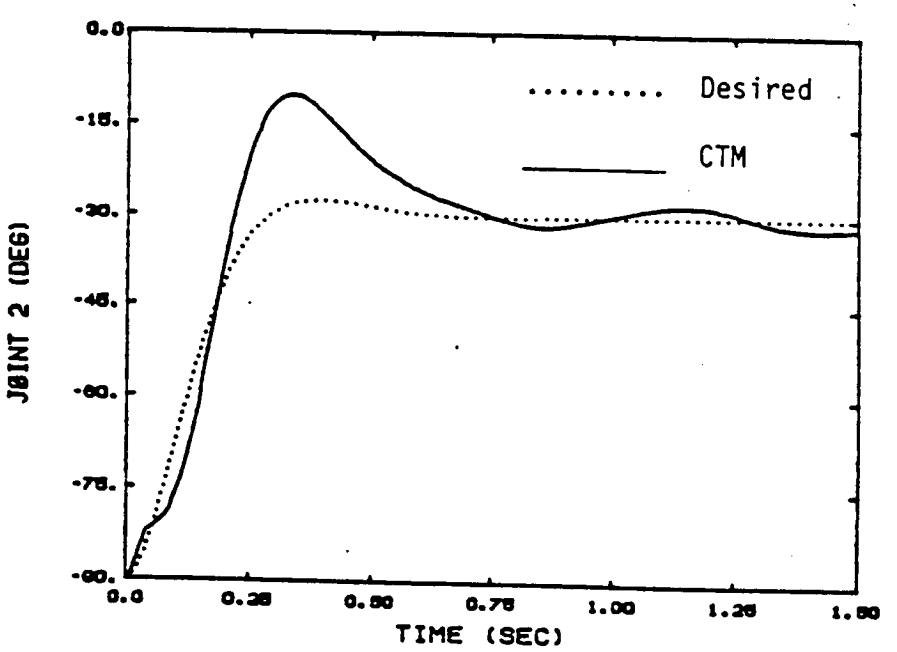


(b)



(c)

Fig.4.15.c-d Flexible model repsonse, controller: CTM,
c) joint 1 for case (c), d) joint 2 for case (c).



(d)

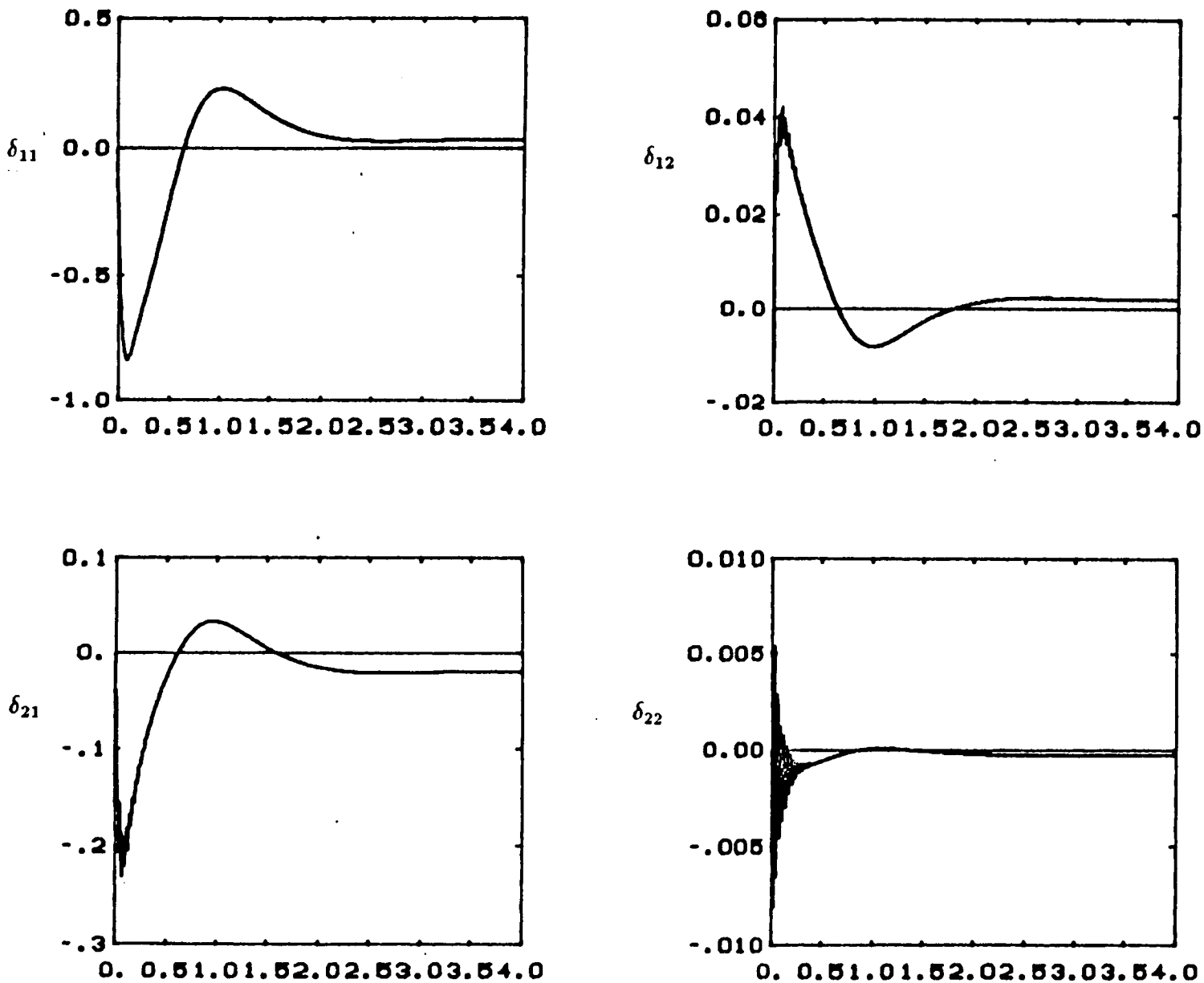


Fig.4.16.a Flexible mode shape magnitude responses along the motion $w_{ni} = 1/4w_{cc1}$

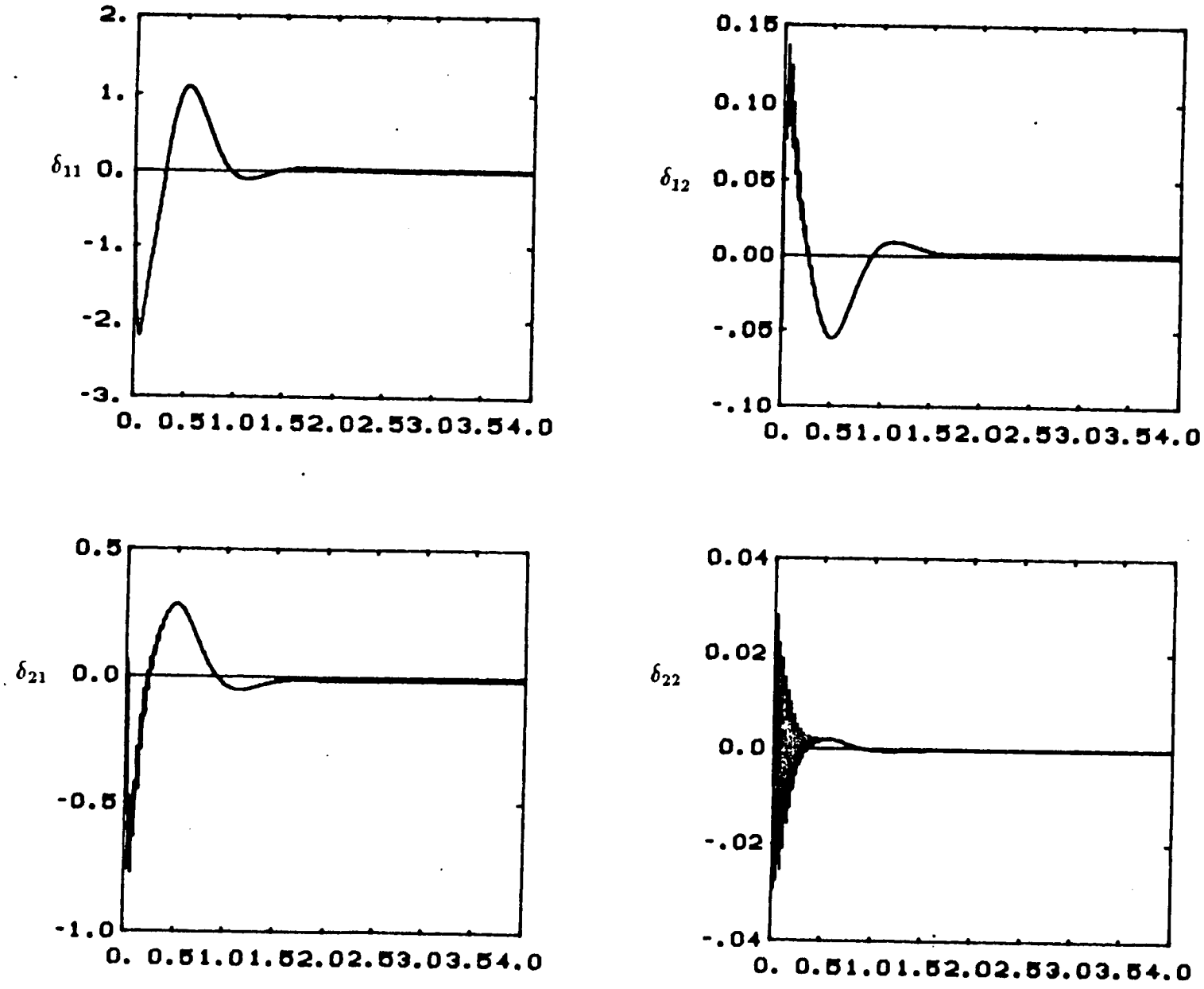


Fig.4.16.b Flexible mode shape magnitude responses along the motion $w_{ni} = 1/2w_{cc1}$

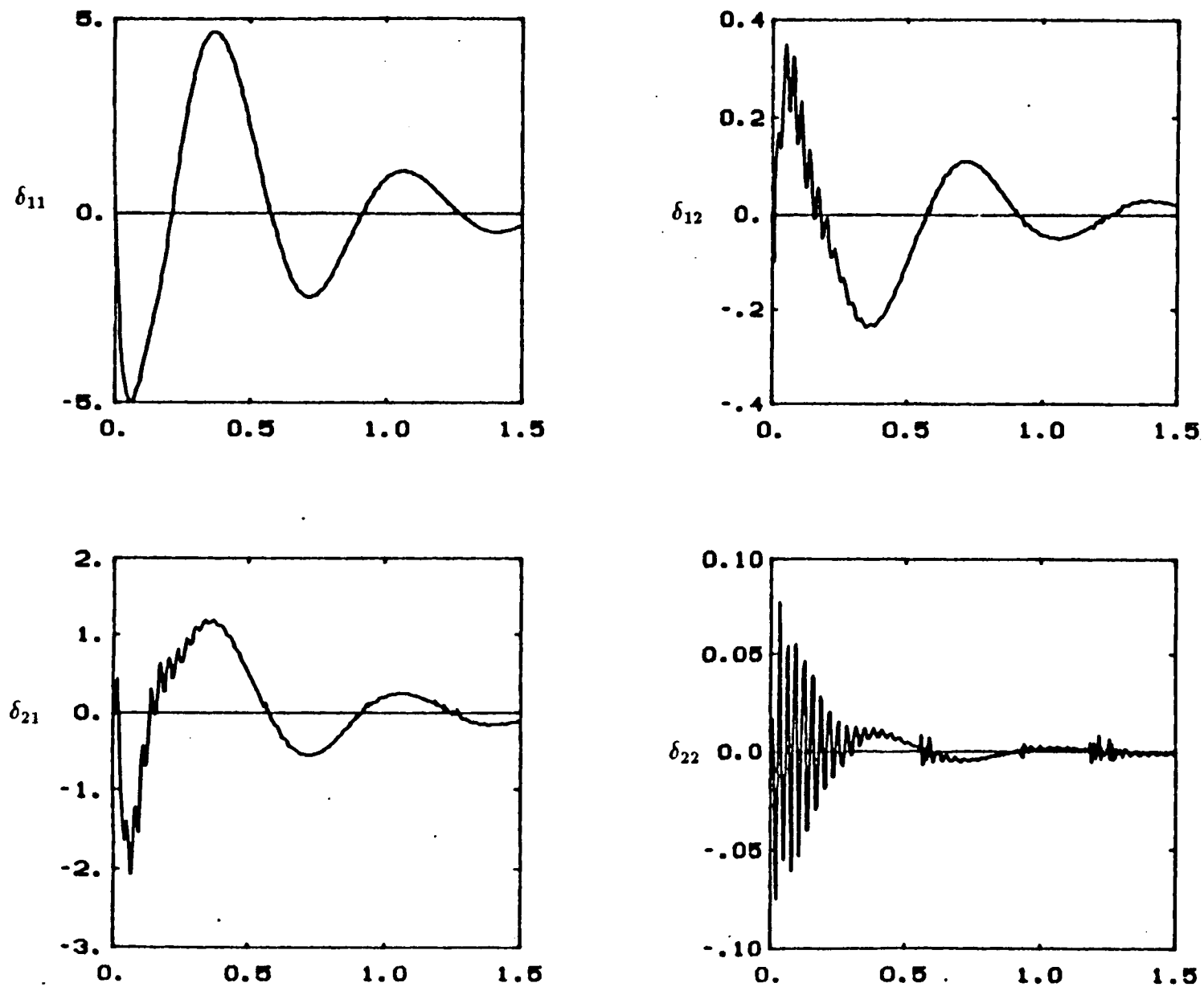


Fig.4.16.c Flexible mode shape magnitude responses along the motion $w_{ni} \cong w_{cc1}$

4.4.3. Gross Motion Control Simulations of Flexible Manipulator Model

The remaining simulations concentrate on the flexible manipulators and compare the performances of the CTM, DJC, and AMFC. First, speed and accuracy is tested under perfect conditions (perfect information about arm parameters, no disturbance). The CTM performance is already discussed above in comparison with rigid model results.

The DJC algorithm is simulated for two cases: (a) desired motion bandwidth $w_{mi} = 2.75 rad/sec$, and closed loop control system bandwidth $w_{ni} = 5.5 rad/sec$, and (b) $w_{mi} = w_{ni} = 5.5 rad/sec$. Comparing the DJC results of cases (a) and (b), Fig. 4.17 a-d, it is seen that the high gain feedback character of case(a) results in better tracking performance compared to case(b). Flexible mode responses are shown in figures 4.18.a-b. Notice the scale difference between the figures. While the general shape of flexible mode magnitudes stays the same, the magnitude of deflections increases with the speed of motion. Shown in figures 4.19.a-d are the AMFC simulation results for case (a) and (b), where AMFC is designed to perfectly match the reference model ($K_m = 0$). This comparison is a little to the advantage of the CTM, and DJC. For case (a), the bandwidths of the CTM and DJC are twice the bandwidth of the corresponding reference model, while the bandwidth of the AMFC is always equal to it. Yet the AMFC performs much better than the CTM and DJC. Notice the decoupled nature of joint responses under the generalized inertia matrix based AMFC. The decoupled response was one of the main objectives of the control algorithm design and was clearly achieved. The use of the generalized inertia matrix in the adaptation algorithm is the key in accomplishing this success. Fig.4.19.-a,b show results for two different values of adaptation parameters (slow and

fast adaptation), as indicated on the figures. Finding the appropriate adaptation parameters was a simple task for there are only two arbitrary design parameters in the integral adaptation used here.

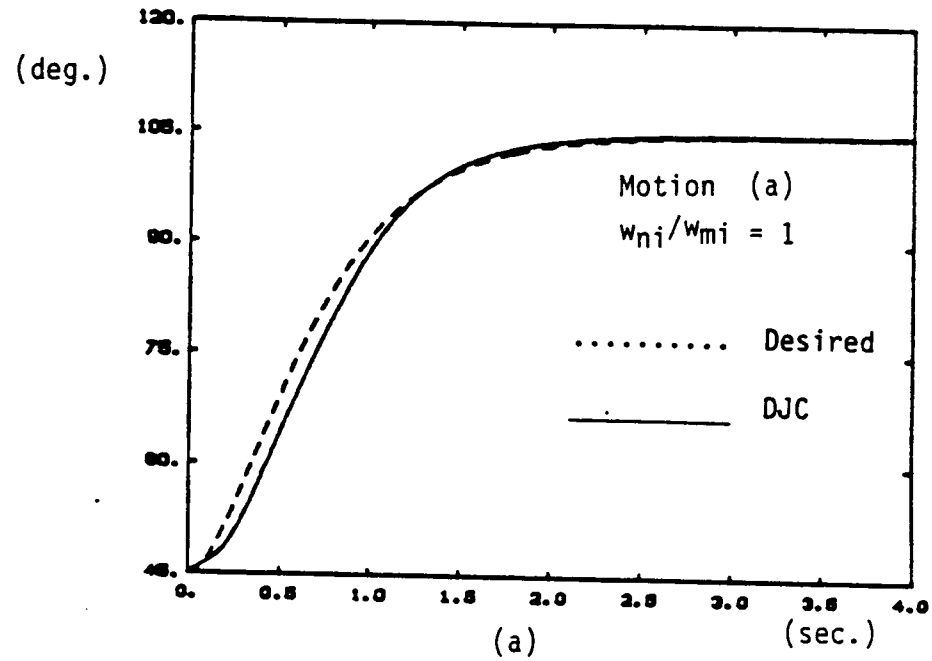
When the desired motion speed is increased, the DJC performance deteriorates [Fig.4.17.c-d], since it does not cancel nonlinear coriolis and centrifugal forces explicitly. At high speeds these forces become important (Fig. 4.9.e-f). For the same motion conditions, the AMFC does not cancel the nonlinear forces explicitly either, but it adapts its feedback gains as functions of tracking error in order to accommodate for these nonlinearities as needed. The DJC results in about 20° oveshoot in joint 2, whereas the AMFC overshoot is less than 2° and the joint responses are decoupled (compare fig. 4.17.c-d and fig 4.19.c-d).

There is a noticeable difference in the magnitude of flexible mode shape responses. For relatively slow motions†, flexible assumed mode shape magnitude responses are similar for all control schemes (Fig. 4.18.a, 4.20.a). However, as the speed increases, the AMFC results in persistent vibrations at the end of the motion (Fig. 4.20.a-b). This is explained as follows: when the speed is high, the nonlinear terms become important. The AMFC automatically increases the feedback gains based on the adaptation algorithm to compensate for these terms, and eventually generates high joint feedback gains. This results in very stiff joints and does not allow the absorption (dampout) of the energy from the flexible beam (Fig. 3.11.g). In a sense there is a trade off. The AMFC enables higher operation speeds. But if the motion gets relatively fast with respect to the arm flexibility, the AMFC fails to deal with end point vibrations due to lightly damped flexible modes. It is important

† In this content, *relatively slow motion* refers with respect to arm flexibility.

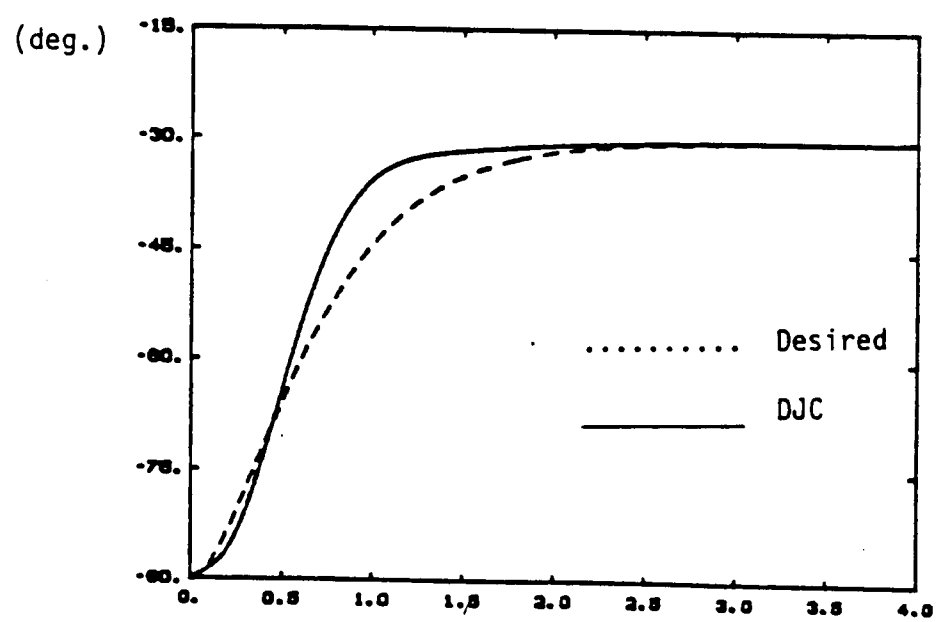
to note that the AMFC can take care of nonlinear effects during fast motions, but not the end point vibrations. If a desired motion is relatively fast with respect to nonlinearities, but not fast with respect to the arm flexibility, the AMFC will perform very well and end point vibration problem will not arise.

The remaining simulations compare the robustness of CTM, DJC, AMFC with respect to payload variations of 25% (payload/robot mass) ratio. As seen before, the only way the non-adaptive CTM, and DJC algorithms could provide robustness with respect to payload variations was to use high gain feedback. In order to have some reasonable results, the CTM, and DJC are simulated with $w_{mi} = 2.75 rad/sec$, $w_{ni} = 5.5 rad/sec$ and compared to the AMFC results with perfect model following objective ($K_m = 0.$). Shown in figures 4.21 - 4.22. are the CTM, and DJC simulation results. Steady state errors vary from 5° to 15° and large overshoot in the second joint response are very unsatisfactory. Figures 4.23.a-b show the AMFC simulation results, where the decoupled nature of the joint response is still good. The transient tracking performance is better and overshoot is not as bad as it is in the CTM, and DJC case. Nonetheless, the performance of AMFC under 25% payload/robot mass variation is not satisfactory. The main problem is again the oscillations at the end of motion, which get more severe as the speed increases. Figures 4.21.c-d, 4.22.c-d, 4.23.c-d show the flexible mode responses for the associated motions.



(a)

Fig.4.17.a-b Flexible model response, controller: DJC, $w_{ni} = 5.5 rad/sec$ Desired motion: $w_{mi} = 2.75 rad/sec$, a) Joint 1, b) joint 2.



(b)

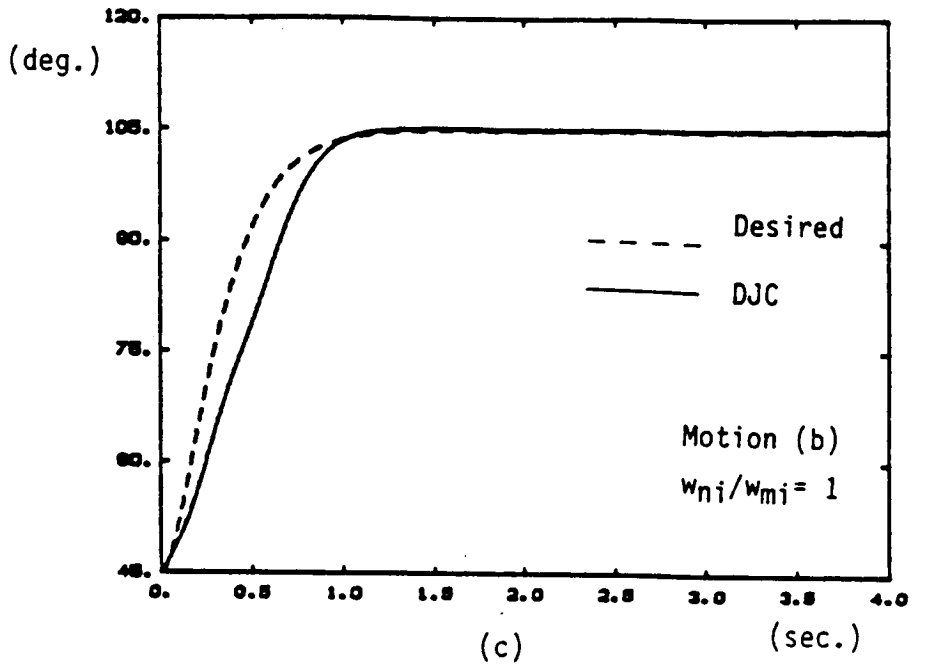
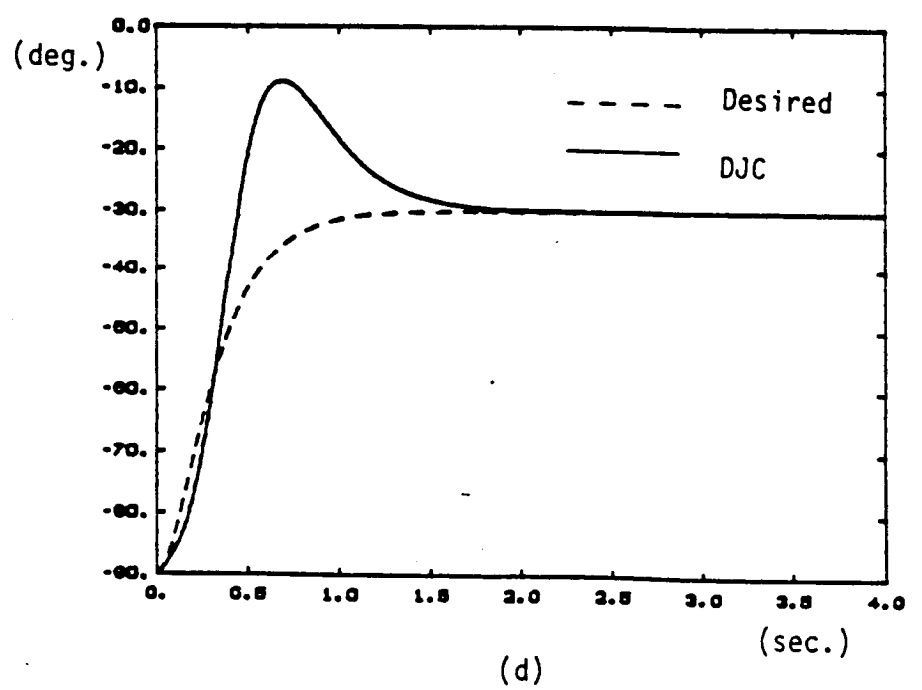


Fig.4.17.c-d Flexible model response, controller: DJC, $w_{ni} = 5.5 rad/sec$ Desired motion: $w_{mi} = 5.50 rad/sec$, a) Joint 1, b) joint 2.



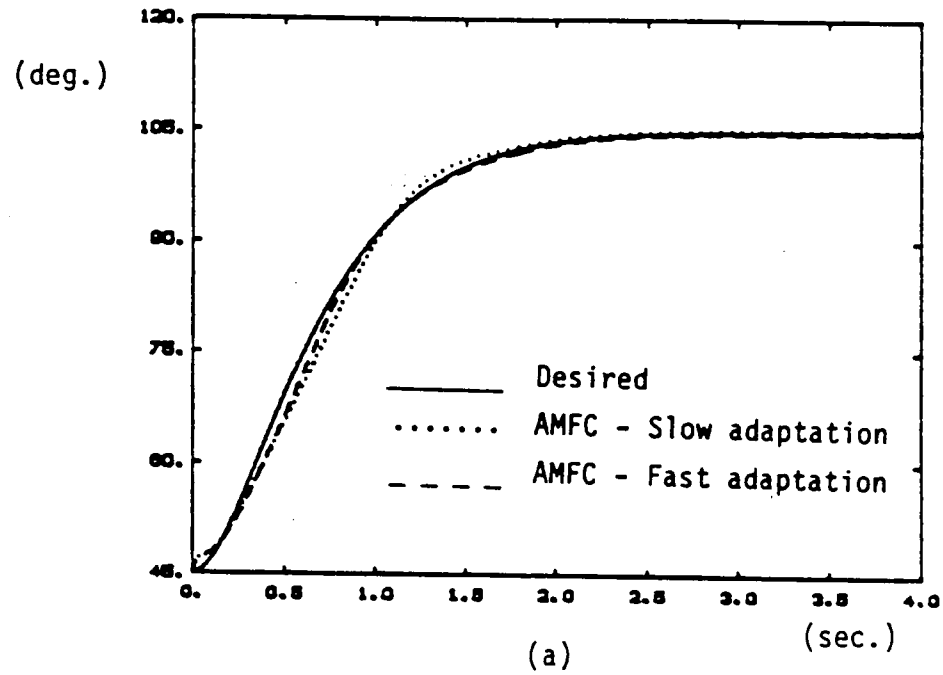
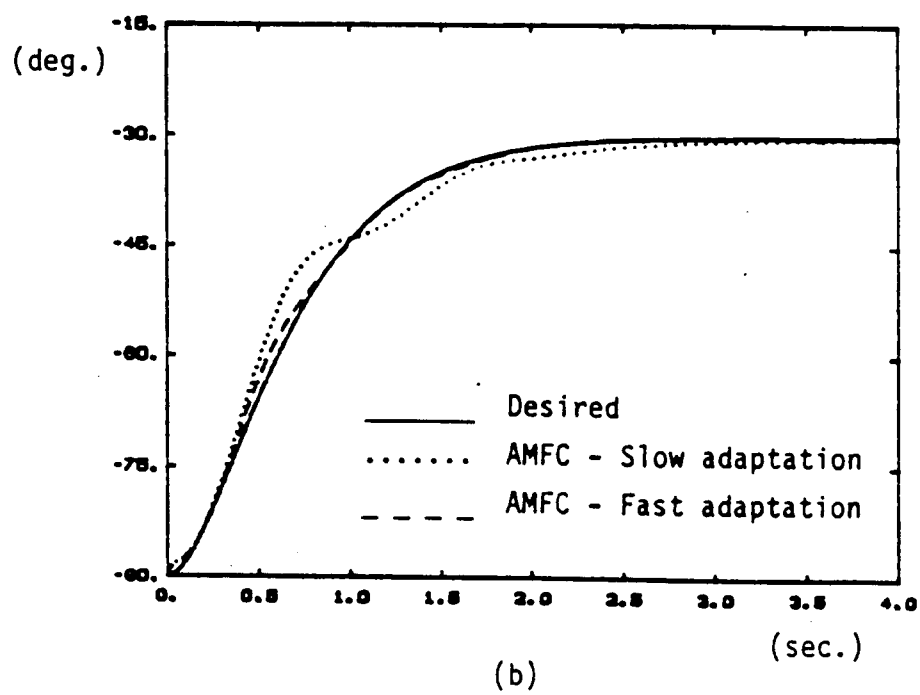


Fig.4.19.a-b Flexible model response, controller: GIM based AMFC Desired motion: $w_{mi} = 2.75 rad/sec$, a) joint 1; b) joint 2.



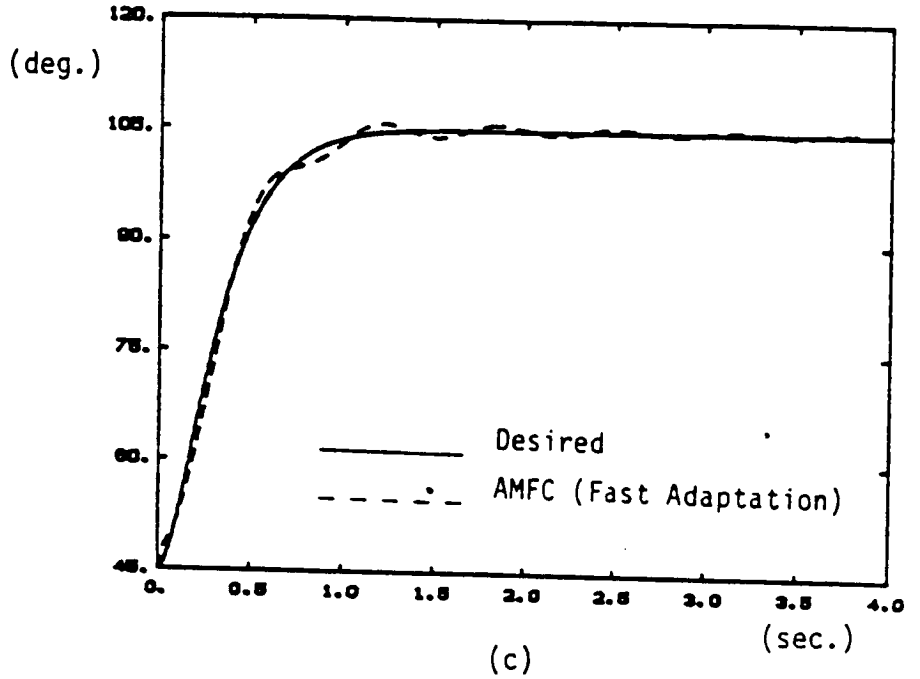
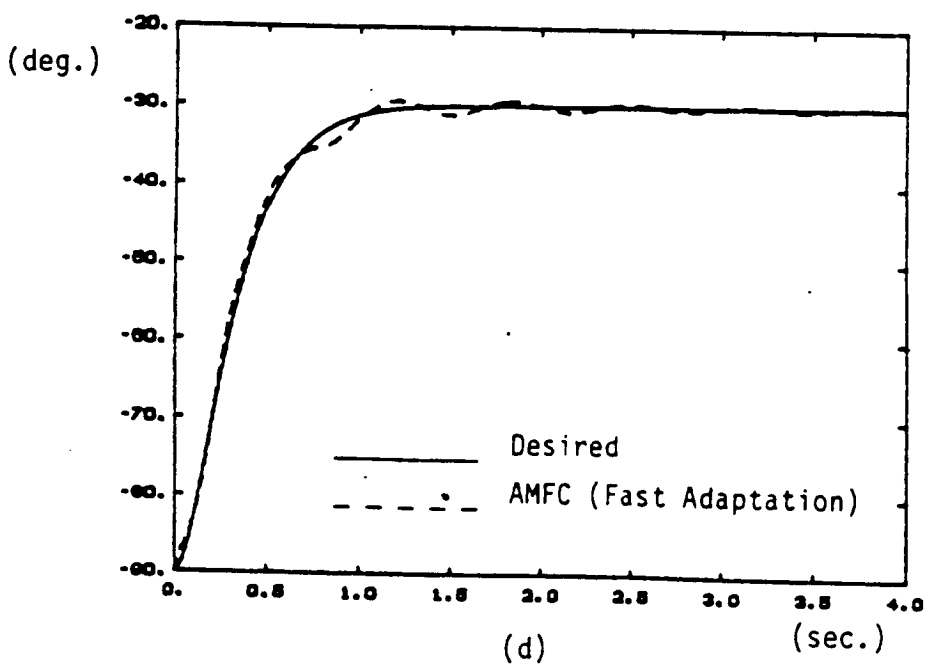


Fig.4.19.c-d Flexible model response, controller: GIM based AMFC Desired motion: $w_{mi} = 5.5 rad/sec$, c) joint 1, d) joint 2.



4.5. Summary

In summary, the non-adaptive, rigid model based CTM and DJC type control algorithms will give reasonable tracking accuracy for payload variations of 25% if closed loop control system bandwidth is about 4 time faster than the desired motion bandwidth (reference model bandwidth with step command signal). The upper limit for the closed loop bandwidth is set by the arm flexibility to approximately 1/2 of the first frequency of the arm. Nonlinear effects further restricts that to $1/4 \sim 1/2$ of the first frequency of the arm, for motions relatively fast both with respect to arm flexibility and nonlinearity effects. Therefore, these control algorithms can be used for desired motions with bandwidth less than 1/8 of the w_{cc1} on flexible arms having payload variations up to 25% of the manipulator mass.

The AMFC does not have to take such a conservative approach in order to deal with payload variations, for it can adapt its gains as needed. But the adaptation of gains during fast motions may result in fine motion oscillations due to high servo stiffness. Furthermore, due to the central role of the generalized inertia matrix in the AMFC design used here, joints always have good decoupled responses. Speeds up to 1/2 of w_{cc1} can be attained by the AMFC with comparable accuracy, if there were a way of dealing with fine motion oscillations. If one wants to take advantage of the capabilities offered by the AMFC presented here, and yet be able to damp-out vibrations at the end of motion quickly, a combination of control methods must be considered: the AMFC algorithm for gross motion, another algorithm to explicitly deal with the vibrations at the end of the motion. This, *Combined Control*, is discussed in the next chapter.

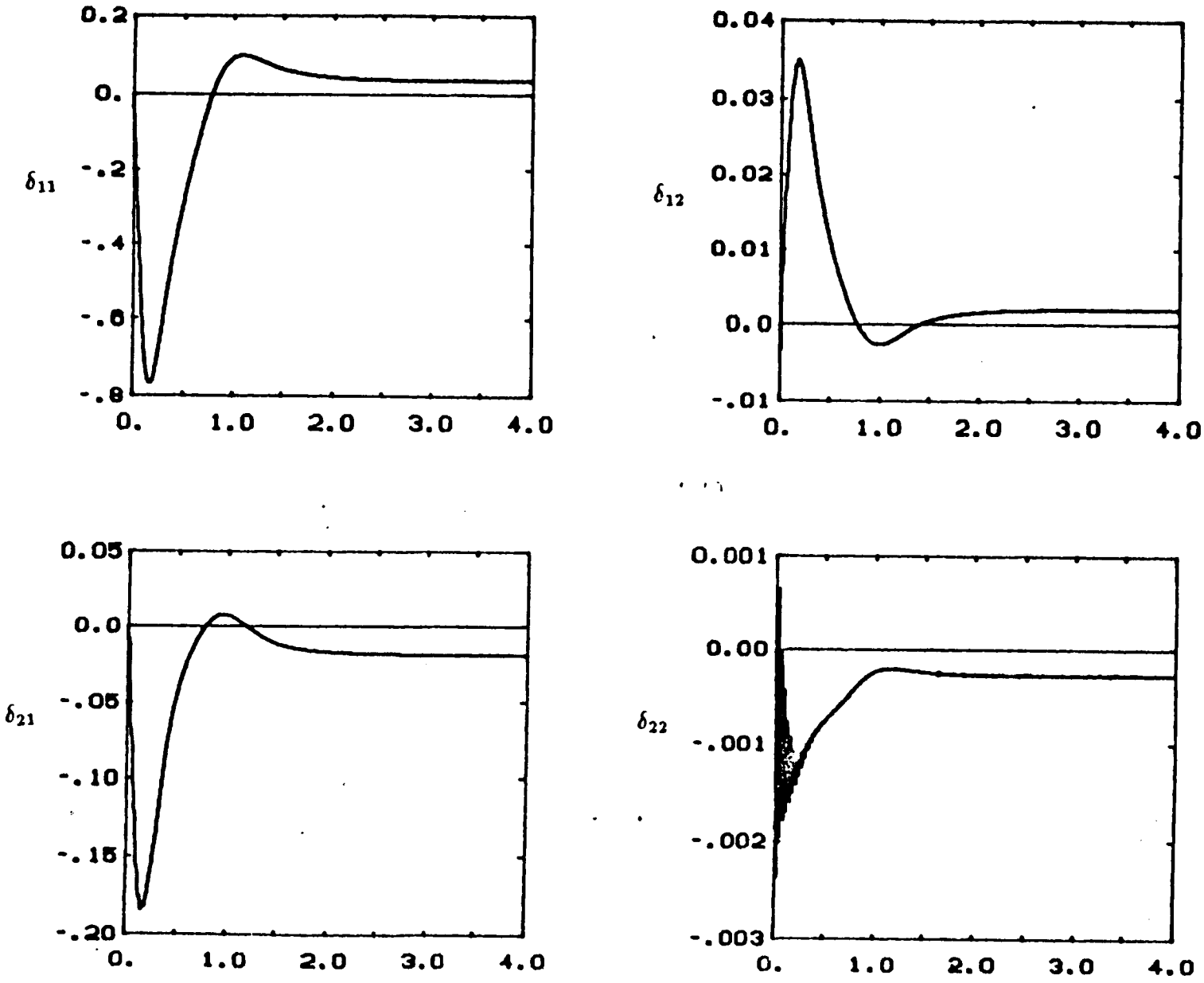


Fig.4.18.a Flexible mode shape magnitude responses along the motion shown in figures 4.17.a-b.

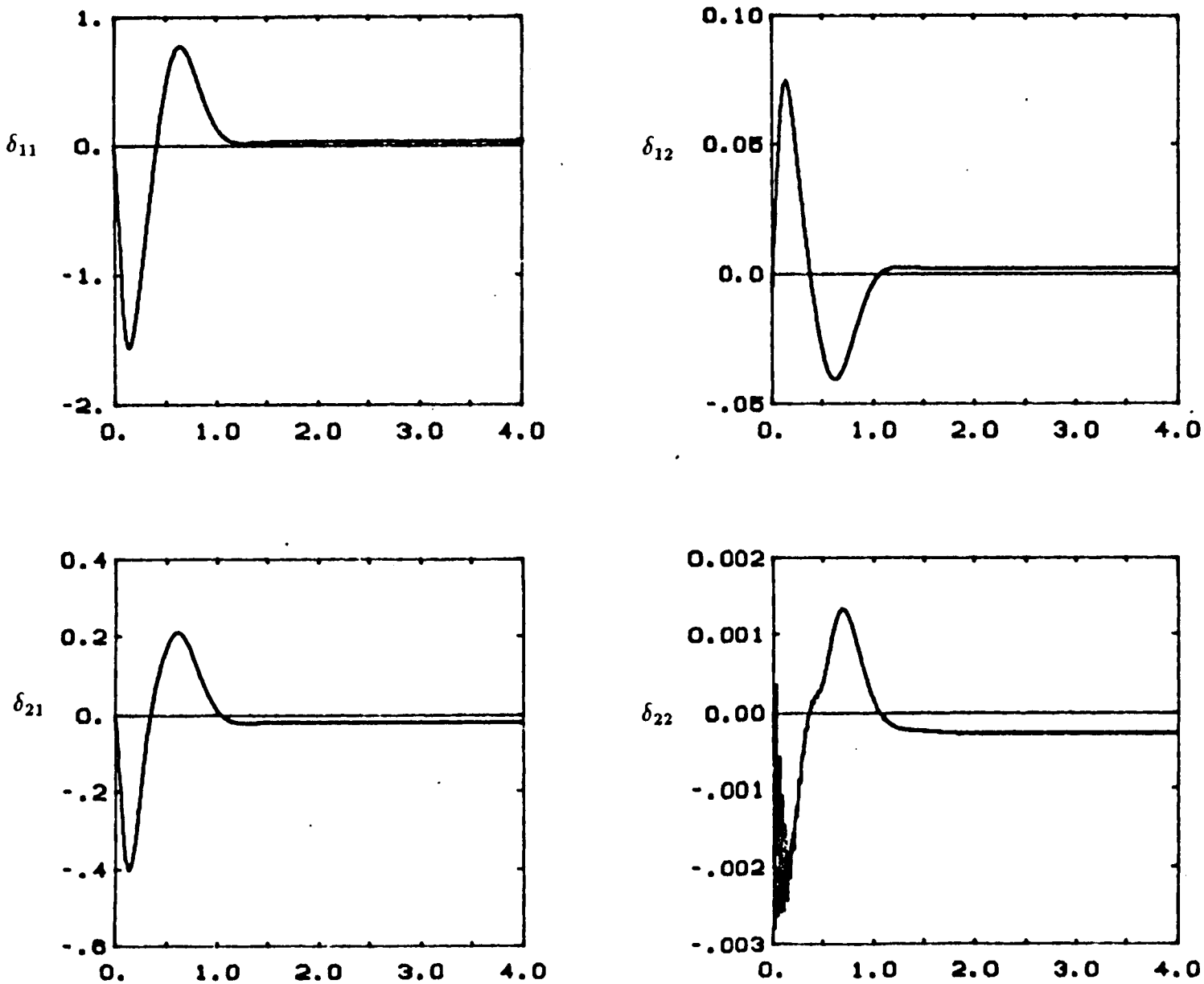


Fig.4.18.b Flexible mode shape magnitude responses along the motion shown in figures 4.17.c-d.

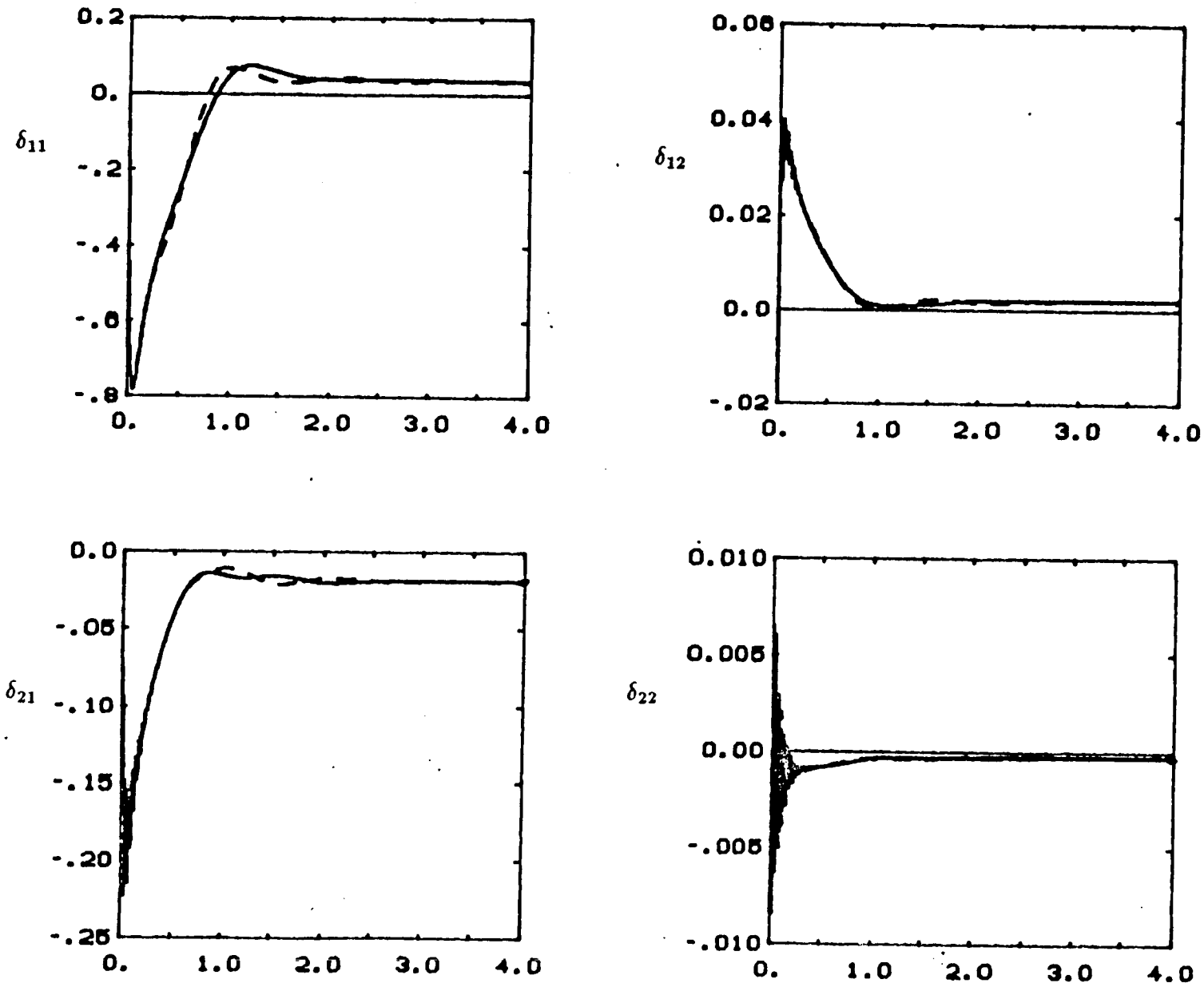


Fig.4.20.a Flexible mode shape magnitude responses along the motion shown in figures 4.19.a,b

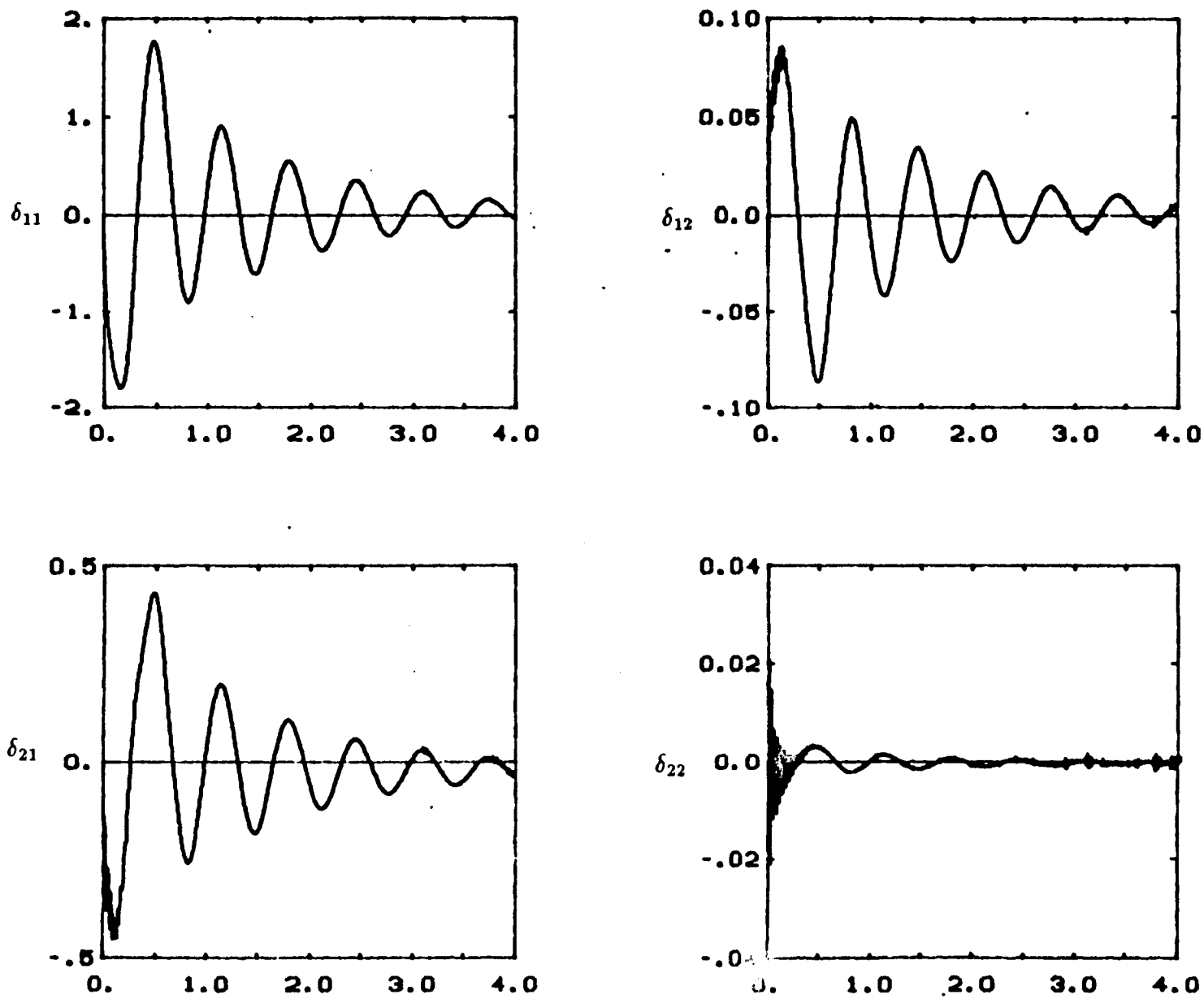


Fig.4.20.b Flexible mode shape magnitude responses along the motion shown in figures 4.19.c-d.

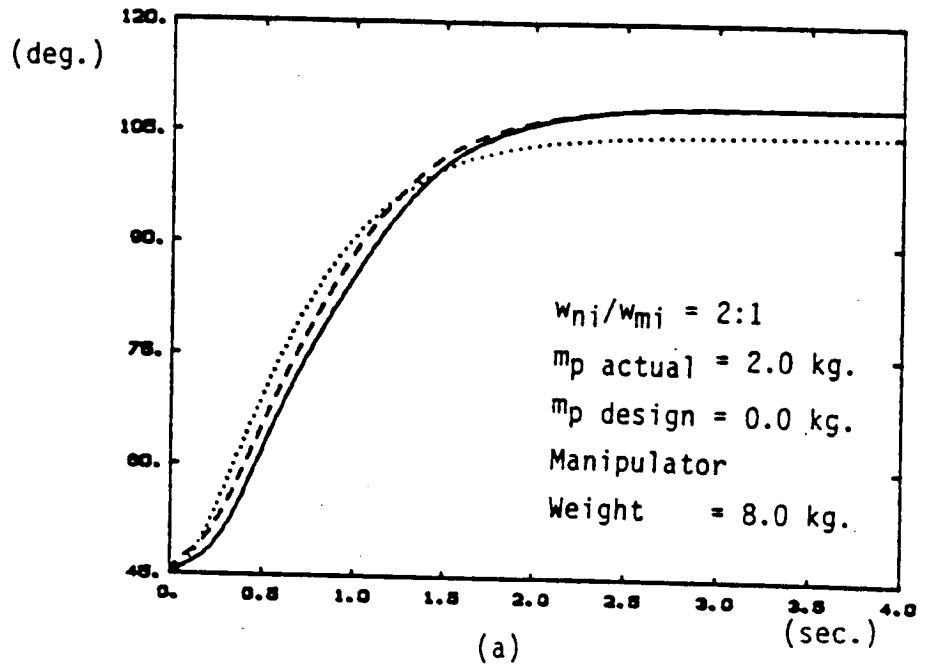
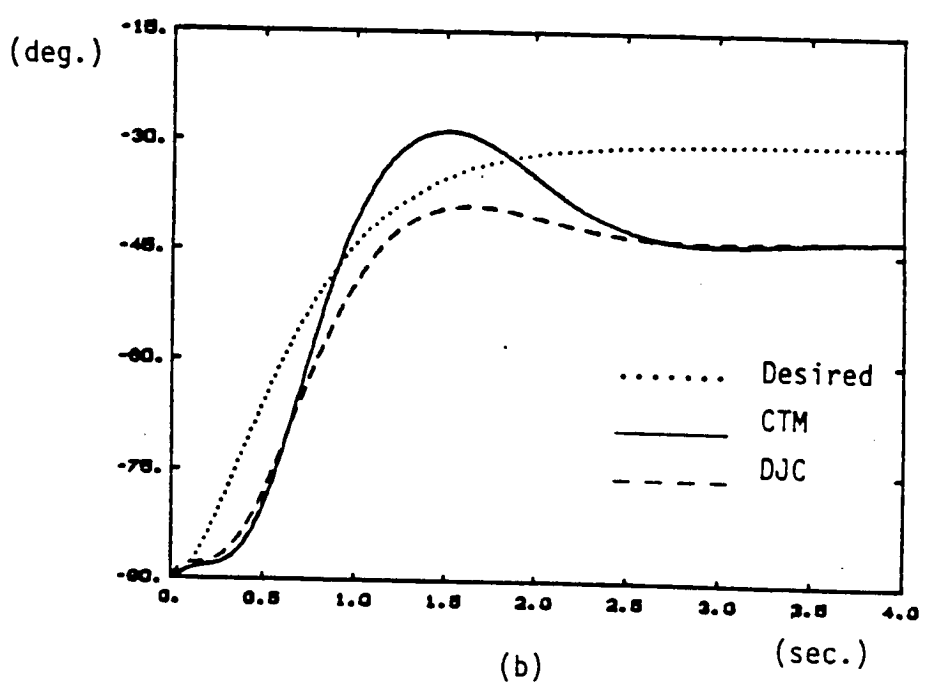


Fig.4.21.a-b Robustness of CTM and DJC with respect to payload variations (Flexible model responses), controller: $w_{ni} = 5.5 rad/sec$, desired motion: $w_{mi} = 2.75 rad/sec$, a) joint 1, b) joint 2



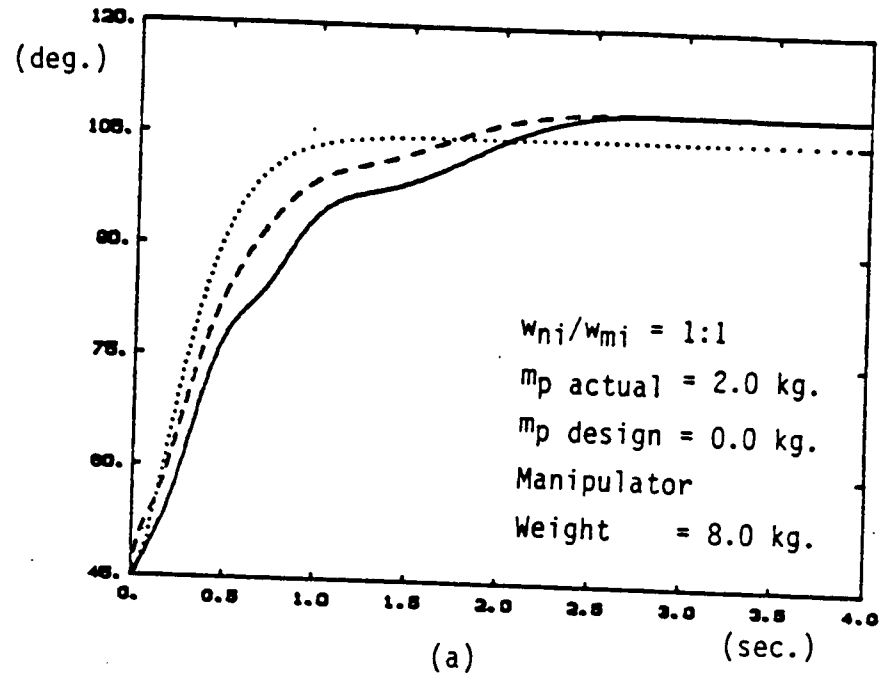
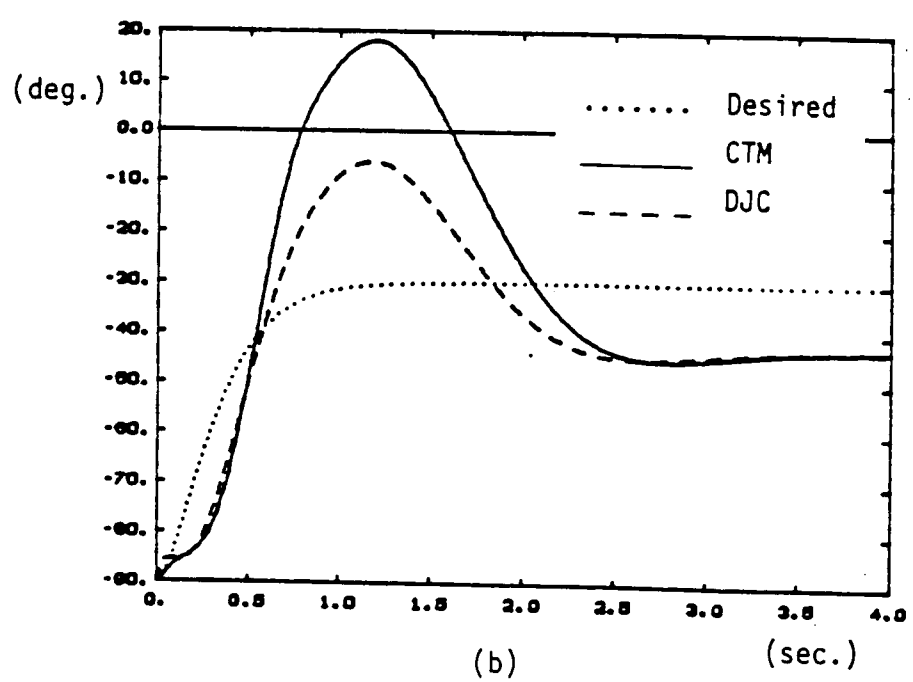


Fig.4.22.a-b Robustness of CTM and DJC with respect to payload variations (Flexible model responses), controller: $w_{ni} = 5.5 rad/sec$, desired motion: $w_{mi} = 5.50 rad/sec$, a) joint 1, b) joint 2



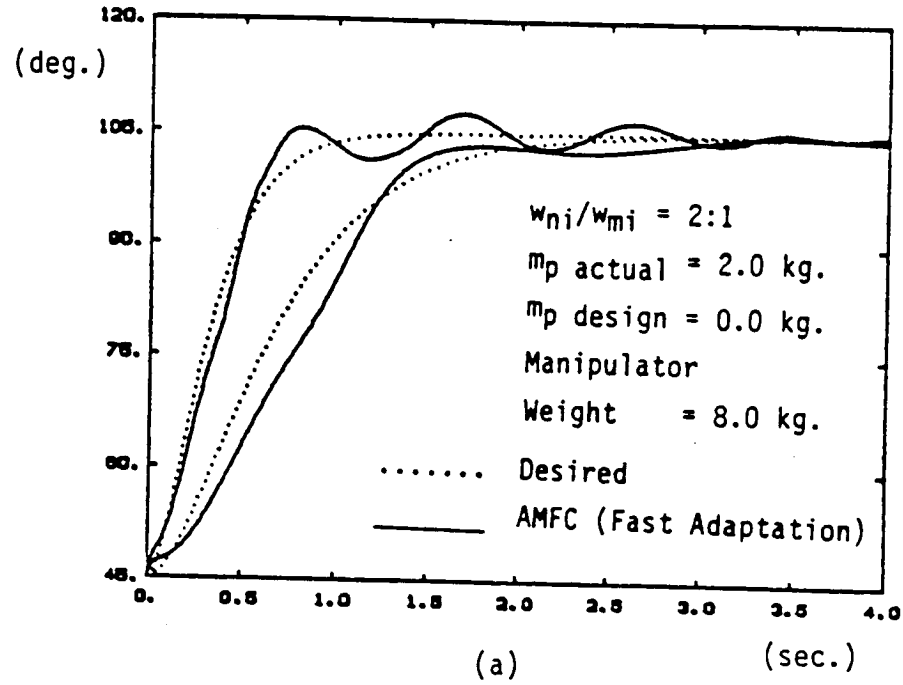
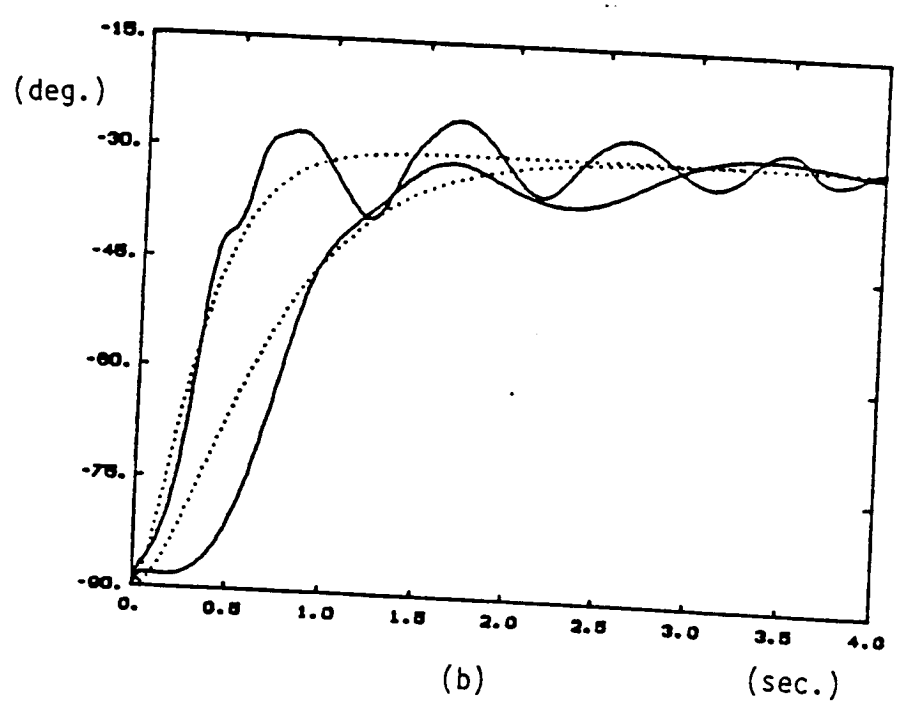


Fig.4.23.a-b Robustness of AMFC with respect to payload variations a) joint 1 response, b) joint 2 response



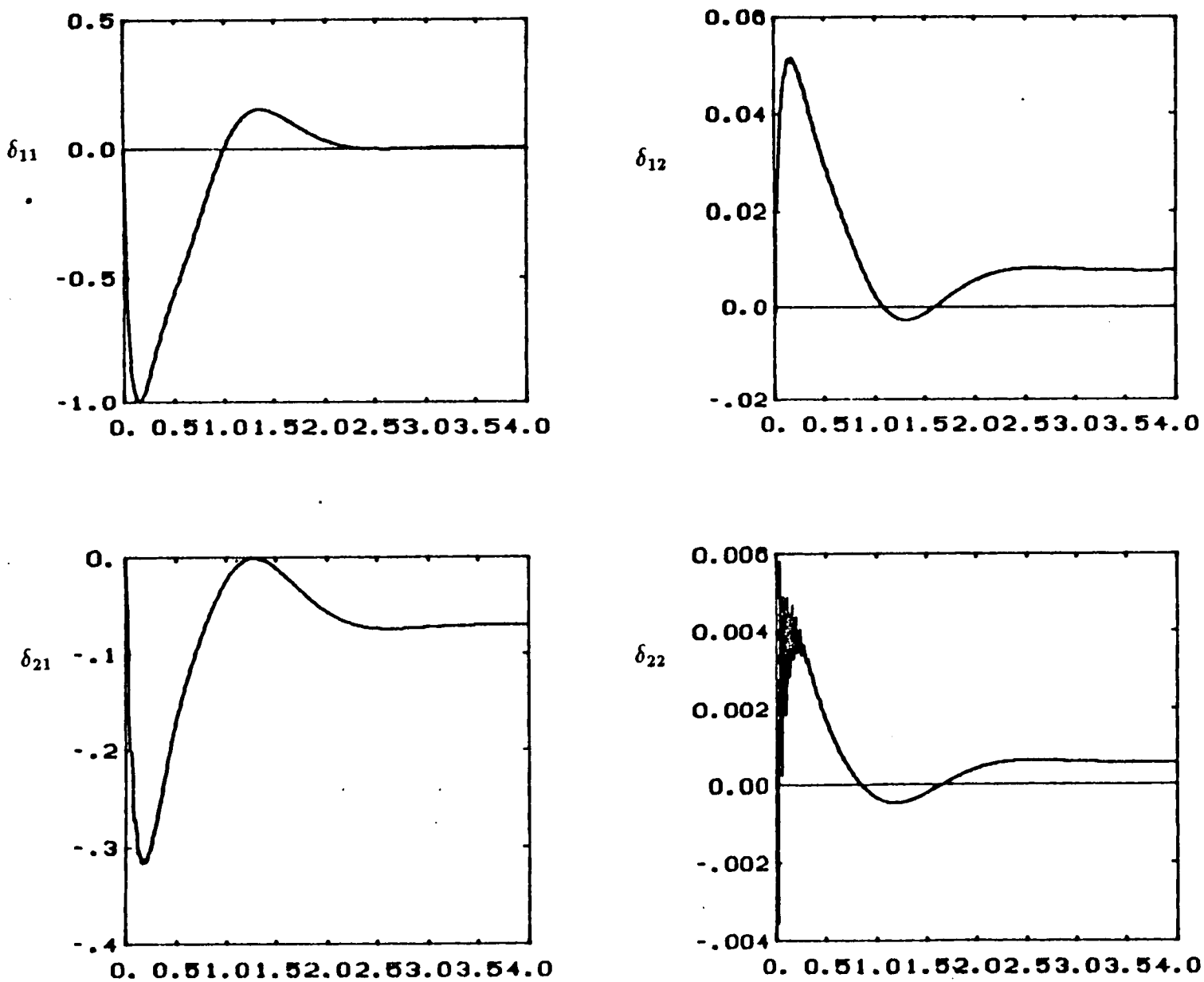


Fig.4.21.c Flexible mode shape magnitude response under CTM.

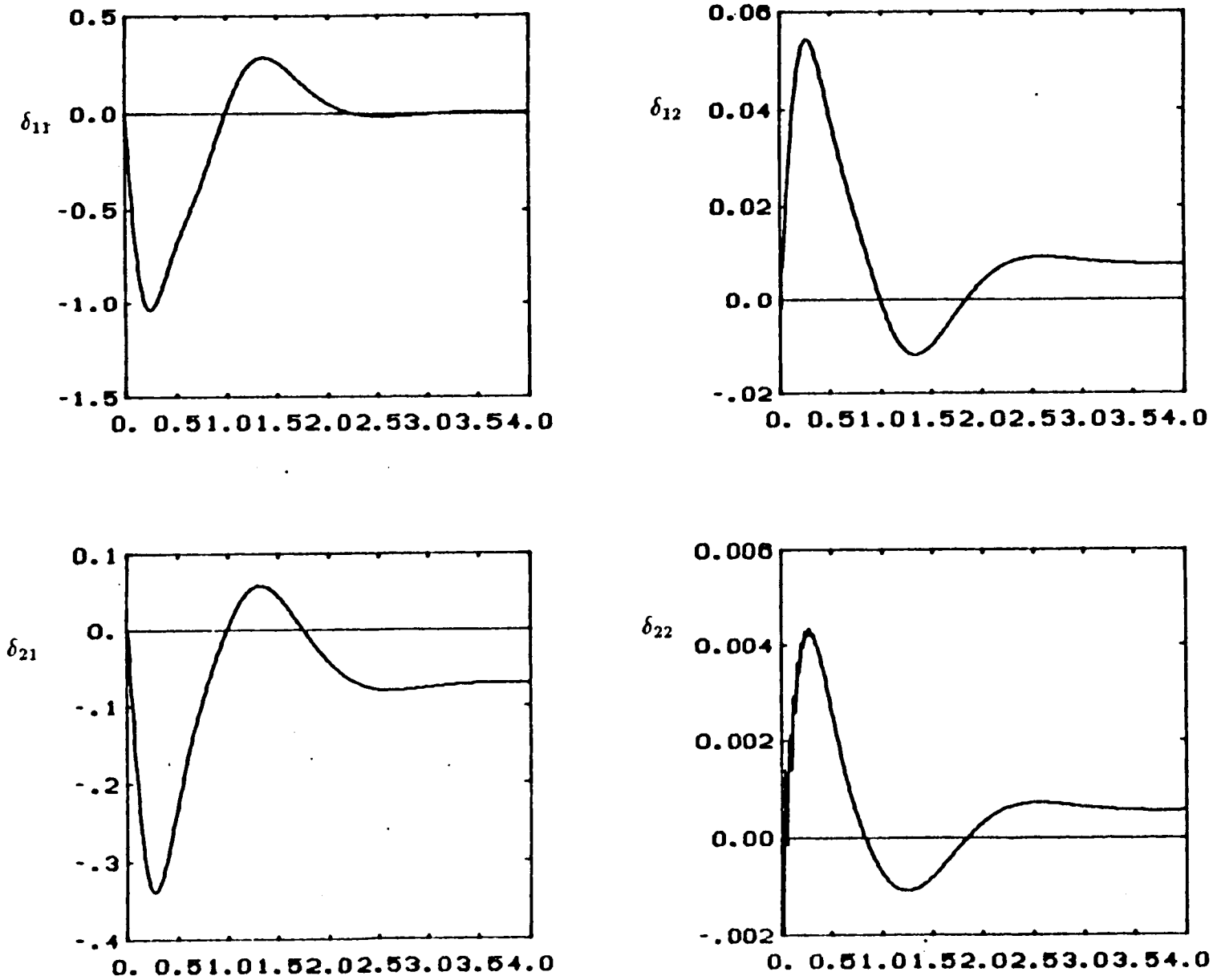


Fig.4.21.d Flexible mode shape magnitude response under DJC.

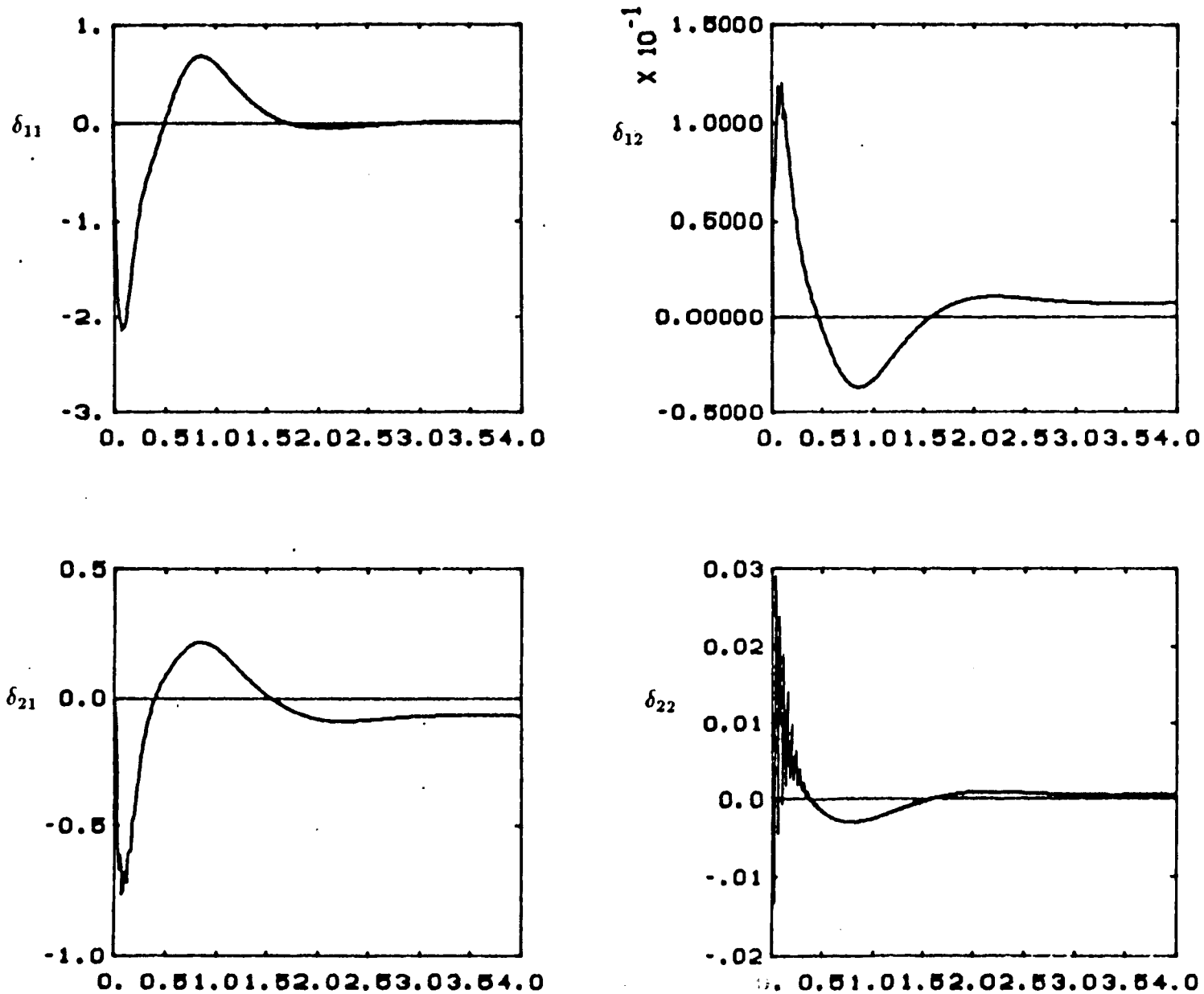


Fig.4.22.c Flexible mode shape magnitude response under CTM.

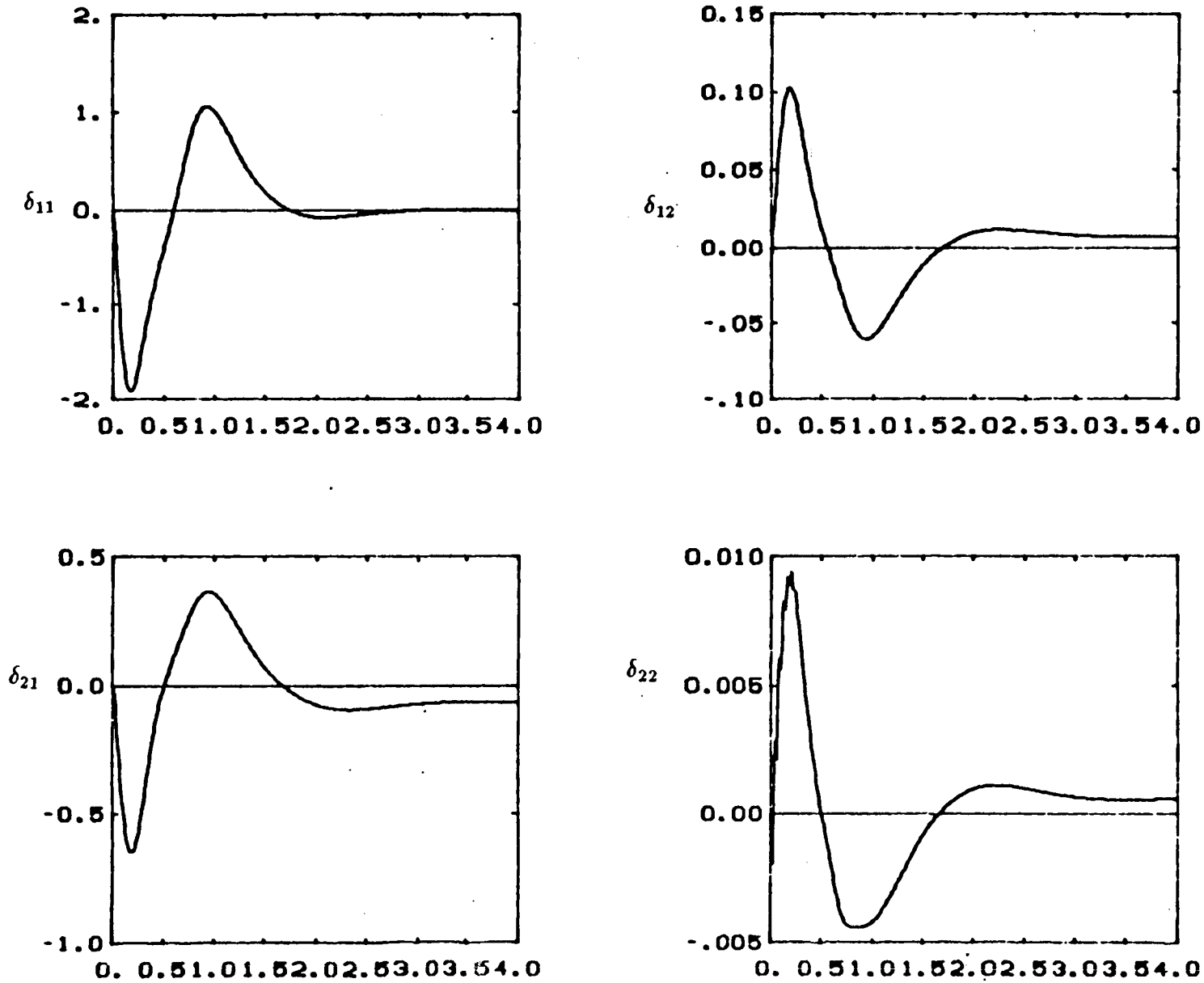


Fig.4.22.d Flexible mode shape magnitude response under DJC.

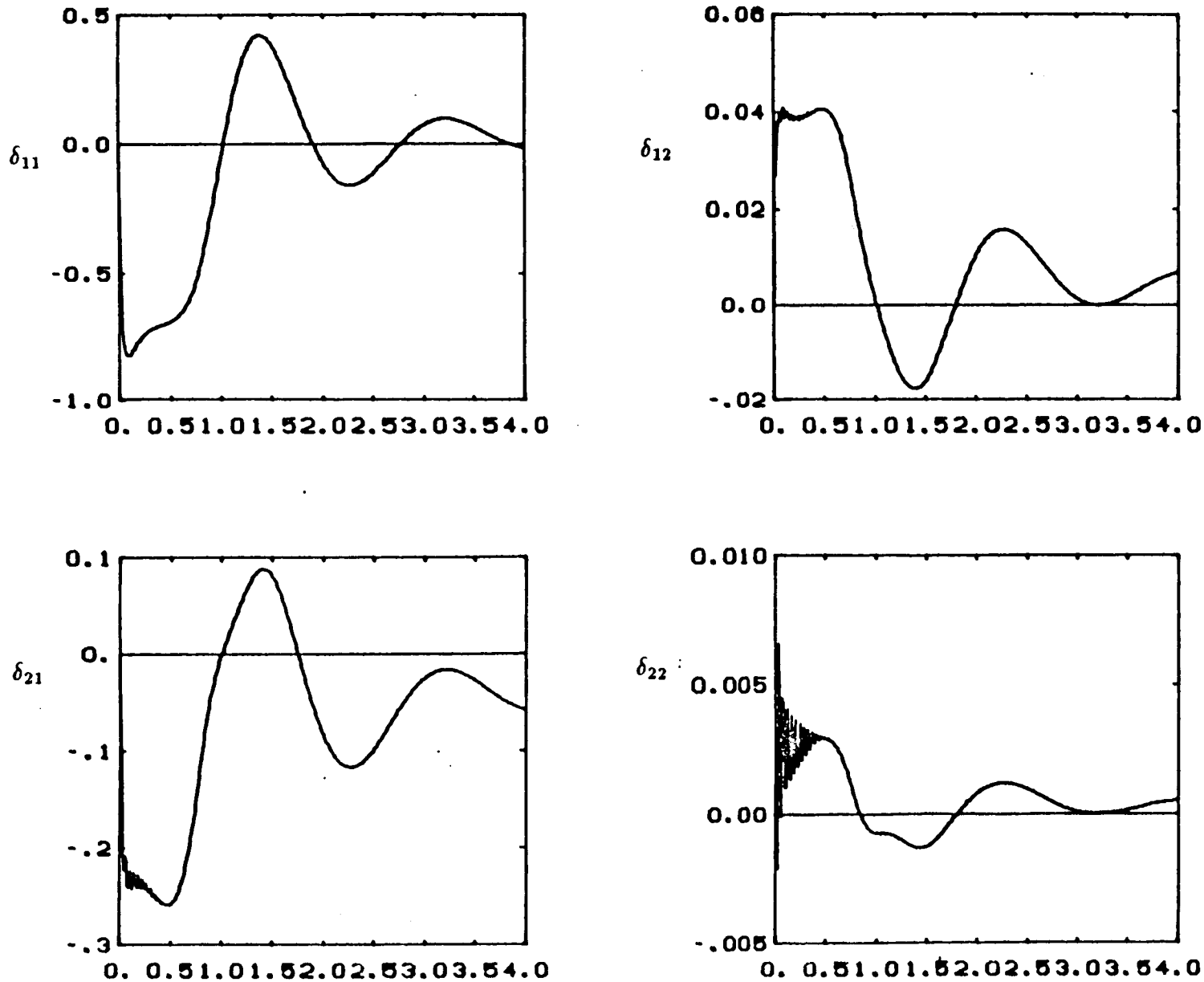


Fig.4.23.c Robustness of AMFC with respect to payload variations c) flexible mode shape magnitude response along motion (a)

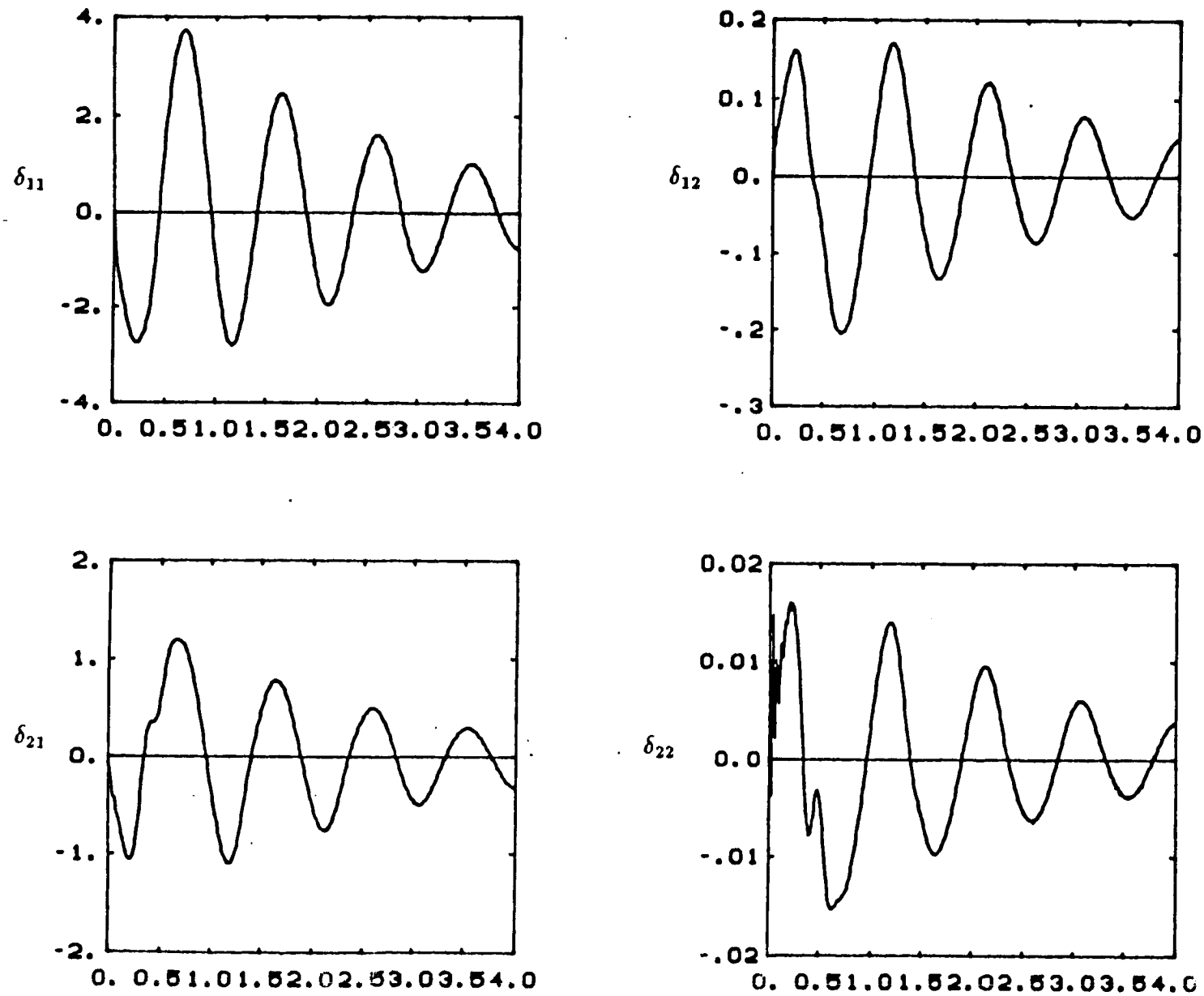


Fig.4.23.d Robustness of AMFC with respect to payload variations d) flexible mode shape magnitude response along motion (b)

CHAPTER V

Combined Motion Control

5.1. Introduction

As shown in Chapter IV, the AMFC based on joint variable feedback has improved the tracking performance of flexible manipulators compared to other control methods over a wide range of payload variations. However, as the desired motion speed is increased to the point of being relatively fast with respect to arm flexibility, the flexible vibrations persisted at the end of the motion due to high joint stiffness. In short, The AMFC provided better joint space tracking and robustness compared to the CTM and DJC, but at the expense of flexible vibrations at the end of the motion. A combined control approach, discussed in this chapter, is intended to overcome the flexible vibrations problem while retaining the advantages of the AMFC. During gross motion, the AMFC is used to control the manipulator. Before the manipulator reaches the final state, the control algorithm is switched to fine motion control, which is designed to deal with joint position and flexible vibration control. The main objective of the fine motion controler is to damp out the residual flexible vibrations as quickly as possible while positioning the joints at a desired configuration.

Since the fine motion is about a final desired state (fig. 5.1.), and is slowing down, a controller using the linear model of manipulator about the final desired

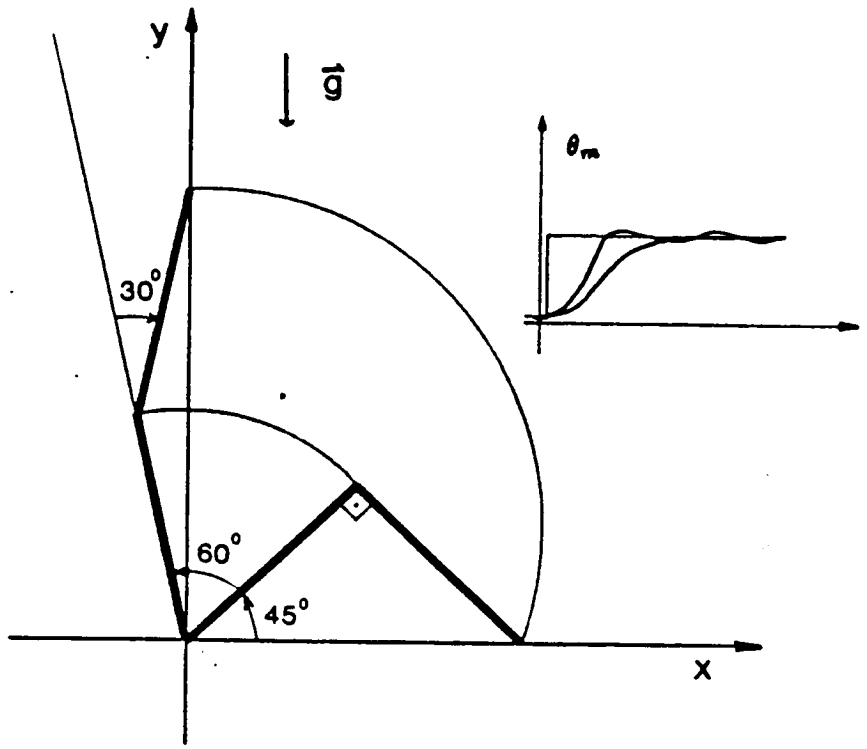


Fig.5.1 Desired motion in task space

state may be designed. However, the success of such a controller can be expected within the validity range of the linear model. For instance, if the control is switched to the fine motion control too early or if speeds at the switching time are too fast, the linear model used in the design of controller may no longer be an accurate representation of the manipulator model for the current state. In that case, the fine motion controller is unlikely to achieve the desired performance.

The question of when to switch from gross motion control to fine motion control depends on the nature of the task, environment, and the type of the gross and fine motion control algorithms used. Thus, the decision about the switching time will be very much case dependent and must be decided by a higher level of the

control system hierarchy [E8].

In this work, a linear control law is obtained using optimal LQR methods, and used for fine motion control. The main reason for choosing the optimal LQR type control is its convenience for design. Direct pole placement or eigenstructure assignment methods are not used due to the nonuniqueness problem of the solution for feedback gains [E3,E4,E5,E6,E7]. Other control methods may be used for the fine motion control as part of combined control strategy. The gross motion control is studied in chapter IV. Here, fine motion control algorithms will be developed, and combined control simulations will be studied.

5.2. Fine Motion Control: Optimal Linear Quadratic Regulators

5.2.1. Preliminaries: General Variation of a Functional

Let $J[x_1, x_2, \ldots, x_n]$ be an integral functional of n-set of independent functions $\{x_1(t), x_2(t), \ldots, x_n(t)\}$,

$$J[x_1, x_2, \ldots, x_n] = \int_{t_o}^{t_1} F(t; x_1, x_2, \ldots, x_n, \dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n)dt \tag{5.1}$$

beginning with a simple case of (5.1),

$$J[x_1] = \int_{t_o}^{t_1} F(t; x_1, \dot{x}_1)dt \tag{5.2}$$

The problem: Find $x_1(t)$ from a class of piecewise continuous, bounded functions such that $J[x_1]$ is minimized.

Developed next are the necessary conditions that must be satisfied by $x_1(t)$ for it to be the minimizing solution. These results will be directly used in the design of the LQR controllers in the following sections.

Let $x_1(t)$ and $x_1^*(t)$ be two neighboring functions, and $h(t) = x_1^*(t) - x_1(t)$; where $x_1^*(t)$ is some small variation of $x_1(t)$ and may have different end points (Fig.5.2). The $\delta J[x_1]$, the first order functional variation of $J[x1]$ due to variation of $x_1(t)$, is obtained by expanding $J[x_1^*(t)] - J[x_1(t)]$ to Taylor series about the $x_1(t)$ and neglecting the terms containing second and higher orders of $h(x)$ †.

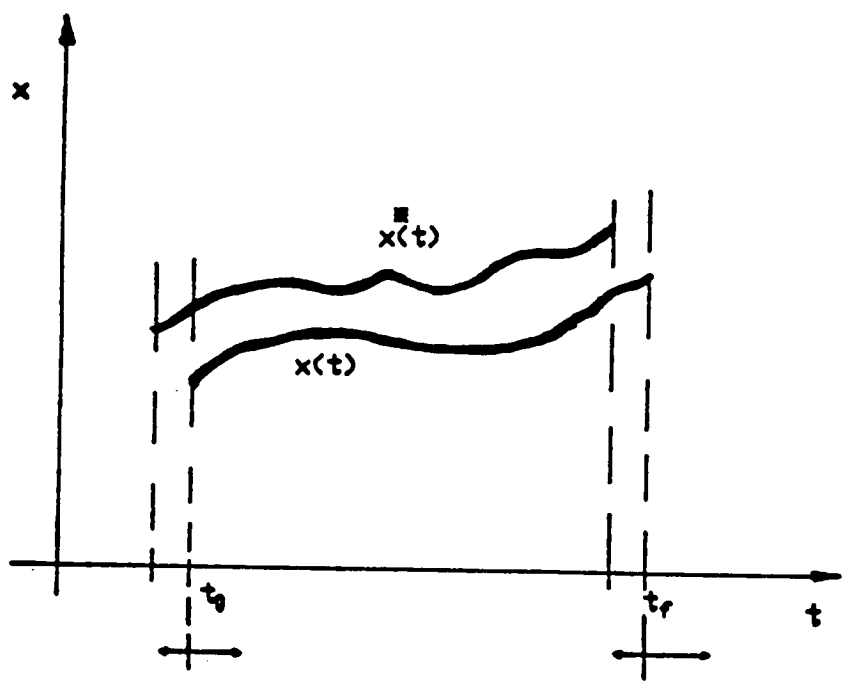


Fig.5.2 General variation of a function

† Subscript 1 will be dropped for brevity.

$$
\begin{aligned}
\Delta J &= J[x^*(t)] - J[x(t)] \\
&= \int_{t_o+\delta t_o}^{t_1+\delta t_1} F(t; x+h, \dot{x}+\dot{h})dt - \int_{t_o}^{t_1} F(t; x, \dot{x})dt \\
&= \int_{t_0}^{t_1} \left[F(t; x+h, \dot{x}+\dot{h}) - F(t; x, \dot{x})\right] dt \\
&\quad + \int_{t_1}^{t_1+\delta t_1} F(t; x_1+h, \dot{x}+\dot{h})dt - \int_{t_o}^{t_o+\delta t_o} F(t; x+h, \dot{x}+\dot{h})dt
\end{aligned}
\tag{5.3}
$$

Expanding $F(t; x+h, \dot{x}+\dot{h})$ to Taylor series about $x(t)$, and neglecting the appropriate terms,

$$
\begin{aligned}
\Delta J \cong \delta J = \int_{t_o}^{t_1} &\left[F_x(t; x, \dot{x})h(t) + F_{\dot{x}}(t; x, \dot{x})\dot{h}(t)\right] dt \\
&+ F\mid_{t_1}\delta t_1 - F\mid_{t_o}\delta t_0
\end{aligned}
\tag{5.4}
$$

Using integration by parts; $\dot{h}(t)dt = dh$;

$$
\begin{aligned}
\int_{t_o}^{t_1} F_{\dot{x}}\dot{h}dt &= \int_{t_o}^{t_1} F_{\dot{x}}dh \\
&= F_{\dot{x}}h\mid_{t_o}^{t_1} - \int_{t_o}^{t_1} h\ (d/dt)F_{\dot{x}}dt
\end{aligned}
\tag{5.5}
$$

Substituting (5.5) into (5.4);

$$
\begin{aligned}
\delta J = \int_{t_o}^{t_1} &\left[F_x(t; x, \dot{x}) - (d/dt)F_{\dot{x}}\right] h(t)dt \\
&+ F_{\dot{x}}h|_{t_o}^{t_1} + F\delta t|_{t_o}^{t_1}
\end{aligned}
\tag{5.6}
$$

Referring to figure 5.2, the relationship between $h(t)$ and $x(t)$ at the boundaries can be shown to be as follows,

$$
\begin{aligned}
h(t_o) &\approx \delta x_o - \dot{x}(t_o)\delta t_o \\
h(t_1) &\approx \delta x_1 - \dot{x}(t_1)\delta t_1
\end{aligned}
\tag{5.7}
$$

and substituting (5.7) into (5.6),

$$\delta J = \int_{t_o}^{t_1} [F_x - (d/dt)F_{\dot{x}}]\, h(t)dt \qquad (5.8)$$
$$F_{\dot{x}}\delta x|_{t_o}^{t_1} + [F - \dot{x}F_{\dot{x}}]\, \delta t|_{t_o}^{t_1}$$

Generalizing this development to the variation of a functional with n-independent variables of the form (5.1);

$$\delta J[x_1, \ldots, x_n] = \int_{t_o}^{t_1} \sum_{i=1}^{n} [F_{x_i} - (d/dt)F_{\dot{x}_i}]\, h_i(t)dt \qquad (5.9)$$
$$\sum_{i=1}^{n} F_{\dot{x}_i}\delta x_i|_{t_o}^{t_1} + \left[F - \sum_{i=1}^{n} \dot{x}_i F_{\dot{x}_i}\right] \delta t|_{t_o}^{t_1}$$

Further generalizing the functional to the case where it includes some terms outside the integral (i.e. penalizing the boundary values),

$$J = \int_{t_o}^{t_1} F(t; x_1, \ldots, x_n, \dot{x}_1, \ldots, \dot{x}_n)dt + \phi_1(t; x_1, \ldots, x_n) - \phi_0(t; x_1, \ldots, x_n) \qquad (5.10)$$

and the first order variation of this functional results in,

$$\delta J[x_1, \ldots, x_n] = \int_{t_o}^{t_1} \sum_{i=1}^{n} [F_{x_i} - (d/dt)F_{\dot{x}_i}]\, h_i(t)dt$$
$$\sum_{i=1}^{n} [F_{\dot{x}_i} + \partial\phi_{1,0}/\partial x_i]\, \delta x_i|_{t_o}^{t_1} + \left[F - (\sum_{i=1}^{n} \dot{x}_i F_{\dot{x}_i}) + \partial\phi_{1,0}/\partial t\right] \delta t|_{t_o}^{t_1} \qquad (5.11)$$

The necessary conditions for the $\{x_1(t), x_2(t), \ldots, x_n(t)\}$ to be an optimal solution of the problem , is that the first order variation of the functional about the optimal (extremum) solution must vanish, $\delta J = 0$. Thus, the necessary conditions for the optimal solution are as follows:

$$F_{x_i} - \frac{d}{dt}F_{\dot{x}_i} = 0 \ ; \ \text{for} \ \ i = 1, \ldots, n \ \ \text{and} \ \ t \in [t_0, t_1] \qquad (5.12)$$

n-set of second order differential equations that must be satisfied by $\{x_1, \ldots, x_n\}$ simultaneously, and the associated boundary conditions come from the terms out side the integral in (5.11). For example if the final time is specified, then $\delta t_1 = 0$, if not the term in brackets before δt must be zero. Either way, the problem is well defined and there are enough boundary conditions to uniquely determine the optimum solution. Notice that the boundary related terms in the performance index do not affect the necessary differential equations, but the resultant natural boundary conditions only.

Optimal control problems are always posed in the following general form,

Find the control vector $u(t)$ minimizing

$$J = \int_{t_o}^{t_1} L(x, u, t)dt + \phi_1(t_1, x(t_1)) - \phi_o(t_o, x(t_o)) \tag{5.13}$$

subject to system dynamics (and possibly boundary conditions);

$$\dot{x} = f(t; x, u)$$

$x(t_0)$ may be specified if ϕ_o does not exist in (5.13), and $x(t_1)$ may be specified if ϕ_1 does not exist in (5.13). Using the Lagrange multipliers method, the problem can be reduced to the form of (5.1). The equivalent F is;

$$F = L + \lambda^T(f - \dot{x}) \tag{5.14}$$

and independent functions are the vector functions x, u, λ.

The necessary conditions for the optimal solution are as follows:

$$\begin{aligned} F_x - (d/dt)F_{\dot{x}} &= 0 \Rightarrow \dot{\lambda} = -(\frac{\partial f}{\partial x})^T \lambda - (\frac{\partial L}{\partial x})^T \\ F_u - (d/dt)\, F_{\dot{u}} &= 0 \Rightarrow 0 = \frac{\partial L}{\partial u} + \lambda^T \frac{\partial f}{\partial u} \\ F_\lambda - (d/dt)\, F_{\dot{\lambda}} &= 0. \Rightarrow \dot{x} = f(t; x, u) \end{aligned} \tag{5.15}$$

5.2.2. LQR with Prescribed Degree of Stability

The popularity of LQR's as opposed to any other optimal control formulation is due to the fact that the resultant solution is a constant linear feedback control law, which is very desirable for implementation simplicity. In general the solution of a nonlinear optimal control problem results in a nonlinear time varying control history, which must be obtained by solving a nonlinear two point boundary value problem. For a given linear model of manipulator about a region, and a quadratic optimality criteria, there is a unique constant feedback gain under some some conditions as discussed below. It is possible to use the results of LQR in gain scheduling form. One can obtain optimal feedback gain matrices for regions of workspace of the manipulator, and store them off-line. In real-time control, simply recall these gains depending on which region of workspace the manipulator is.

Let the linearized dynamics of a flexible manipulator about a given nominal state, i.e. a final desired configuration, be,

$$\dot{x} = Ax + Bu \tag{5.16}$$

where x is the small variations of the state from the nominal state about which linearization is made, and u is the small variation of input from the nominal input.

Find u such that it minimizes the following quadratic performance index,

$$J[x,u] = 1/2 \int_0^\infty \left[x^T Q x + u^T R u\right] e^{2\alpha t} dt \tag{5.17}$$

subject to (5.16). A unique solution is guarantied if the following conditions are satisfied:

1. (A, B) controllable,

2. $(Q^{1/2}, A)$ observable,

3. $R = R^T > 0$, positive definite

4. $Q = Q^T \geq 0$, positive semi-definite.

The resultant control law guaranties that all closed loop eigenvalues have real parts further to the left of $-\alpha$ on the real axis of s-plane.

Following Anderson and Moore[E1], let

$$\overline{x} = e^{\alpha t} x \quad \text{and} \quad \overline{u} = e^{\alpha t} u \tag{5.18}$$

The problem (5.16) and (5.17) becomes,

$$\begin{aligned} J &= 1/2 \int_0^\infty \left[\overline{x}^T Q \overline{x} + \overline{u}^T R \overline{u}\right] dt \\ \dot{\overline{x}} &= \underbrace{(A + \alpha I)}_{\overline{A}} \overline{x} + B\overline{u} \end{aligned} \tag{5.19}$$

Using the Lagrange multipliers method, the equivalent problem is

$$J^*[\overline{x}, \overline{u}, \overline{\lambda}] = 1/2 \int_0^\infty \left[\overline{x}^T Q \overline{x} + \overline{u}^T R \overline{u} + \overline{\lambda}^T (\overline{A}\overline{x} + B\overline{u} - \dot{\overline{x}})\right] dt \tag{5.20}.$$

Applying the necessary conditions of optimality,

$$F_{\overline{x}} - (d/dt) F_{\dot{\overline{x}}} = 0 \;\Rightarrow\; \dot{\overline{\lambda}} = \overline{A}^T \overline{\lambda} - Q\overline{x} \tag{5.21.a}$$

$$F_{\overline{\lambda}} - (d/dt) F_{\dot{\overline{\lambda}}} = 0 \;\Rightarrow\; \dot{\overline{x}} = \overline{A}\overline{x} + B\overline{u} \tag{5.21.b}$$

$$F_{\overline{u}} - (d/dt) F_{\overline{u}} = 0 \;\Rightarrow\; R\overline{u} + B^T \overline{\lambda} = 0 \tag{5.21.c}$$

Solving (5.21.c) for $\overline{u}$ and substituting into (5.21.b) results in the following linear, two point boundary value problem.

$$\begin{bmatrix} \dot{\overline{x}} \\ \dot{\overline{\lambda}} \end{bmatrix} = \begin{bmatrix} \overline{A} & -BR^{-1}B^T \\ -Q & -\overline{A}^T \end{bmatrix} \begin{bmatrix} \overline{x} \\ \overline{\lambda} \end{bmatrix} \tag{5.22}$$

Letting

$$\overline{\lambda}(t) = S(t)\overline{x}(t) \tag{5.23}$$

in (5.22) results in the well known Matrix Riccati Equation,

$$\dot{S} + S\overline{A} + \overline{A}^T S - SBR^{-1}B^T S + Q = 0. \tag{5.24}$$

The steady state solution of the Riccati equation is needed for regulator problems, for $t_1 \rightarrow \infty$ then $\dot{S}(t) = 0$. Thus from (5.21.c.) and (5.23) and solution of (5.24) results in the control law,

$$\begin{aligned} \overline{u} &= -K\overline{x} \\ K &= R^{-1}B^T S \end{aligned} \tag{5.25}$$

In order to obtain the control law for the original problem, substitute (5.18) into (5.25),the resultant control law is same as (5.25)

$$u = -Kx \tag{5.26}$$

The weighing matrices Q, and R are selected using Bryson's rule as a starting point [B13], and varied until the good closed loop eigenvalues are obtained. Bryson's rule suggests to pick Q and R as diagonal matrices, with the following approximate values:

$$q_{ii} = max(1/(x_i^2)); \qquad r_{ii} = max(1/(u_i^2)) \tag{5.27}$$

where x_i, u_i maximum acceptable values which may result from the optimally controlled system.

5.2.3. Model Following LQR with prescribed degree of stability

The standard LQR results in a control law of the form

$$u = -Kx \tag{5.28}$$

which tries to drive the current states to the final desired values. The error or the driving term which multiplies the feedback gains is the difference between the current state and the final desired state about which the regulation is made. When the motion control is switched from the gross motion to the fine motion control algorithm, immediately taking the difference between the current state and the final desired state may lead to undesirable consequences, such as actuator saturation, and further excitation of flexible vibrations. To avoid these problems, a control law following a smooth desired trajectory in both joint and flexible variables may be a better approach than the regulation approach. This may be accomplished by,

$$u = -K(x - x_m) \tag{5.29}$$

where x_m is generated by a reference model chosen by the designer,

$$\dot{x}_m = A_m x_m \tag{5.30}$$

where $x_m = [\theta, \delta, \dot{\theta}, \dot{\delta}]_{desired}$. The reference model is driven by the initial condition which is the difference between the final desired state and the actual state of the manipulator at the switching time. In a way, the fine motion reference model looks at the state at the time of switching and the state where the manipulator is supposed to go, then generates a smooth reference trajectory to go between them.

The implementation (5.29) is no longer an *optimal* control for the optimality criteria defined by (5.17). It would be interesting to determine the performance

C-3

difference between the control (5.29), and *the optimal model following* control of the form,

$$u = -K_1 x - K_2 x_m \tag{5.31}$$

The formulation of the model following LQR with prescribed degree of stability follows.

Find u, minimizing J,

$$J = (1/2) \int_0^\infty \left[(x - x_m)^T Q (x - x_m) + u^T R u \right] e^{2\alpha t} dt \tag{5.32}$$

Subject to†,

$$\dot{x} = Ax + Bu \tag{5.33}$$

$$\dot{x}_m = A_m x_m \tag{5.34}$$

Notice that x_m is not one of the independent functions of the problem, and is completely determined by the initial conditions. So there is no δx_m, and the variation of J is not a function of δx_m.

Following the same development of the previous section, let $\overline{x} = x\ e^{\alpha t}$, $\overline{u} = u\ e^{\alpha t}$, $\overline{x}_m = x_m\ e^{\alpha t}$, and using Lagrange multiplier to adjoin (5.33) into the functional, and applying the necessary conditions of optimality for $\overline{x}, \overline{u}$, and $\overline{\lambda}$,(but not

† The problem cannot be solved by augmenting (5.33) and (5.34) for the new dynamics would be uncontrollable,

$$\left(\begin{bmatrix} A & 0 \\ 0 & A_m \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix} \right)$$

$\overline{x}_m$), results in the two point boundary value problem as follows:

$$\begin{aligned} \dot{\overline{x}} &= \overline{A}\overline{x} - BR^{-1}B^T\overline{\lambda} \\ \dot{\overline{\lambda}} &= -Q\overline{x} - \overline{A}^T\overline{\lambda} + Q\overline{x}_m \end{aligned} \tag{5.35}$$

where the coupling to these equations from $\overline{x}_m$is governed by (5.34). Thus, we have a two point boundary value problem composed of (5.35) and (5.34).

The solution can be obtained by letting

$$\overline{\lambda} = S_1\overline{x} + S_2\overline{x}_m$$

and substituting into the (5.35). After some algebraic manipulation, the following equations are obtained as optimality conditions by requiring the coefficients of $\overline{x}$ and $\overline{x}_m$ to go to zero.

$$\begin{aligned} \left(\dot{S}_1 + S_1\overline{A} + \overline{A}^T S_1 - S_1BR^{-1}B^TS_1 + Q\right) x = 0. \\ \left(\dot{S}_2 + S_2\overline{A}_m + \overline{A}^T S_2 - S_1BR^{-1}B^TS_2 - Q\right) x_m = 0 \end{aligned} \tag{5.36.a}$$

For the steady-state solution, let $\dot{S}_1$ and $\dot{S}_2 \rightarrow 0$ and the resultant equations are algebraic Riccati equation and Lyapunov equation.

$$\begin{aligned} S_1\overline{A} + \overline{A}^T S_1 - S_1BR^{-1}B^TS_1 + Q = 0 \\ S_2\overline{A}_m + (\overline{A}^T - S_1BR^{-1}B^T)S_2 - Q = 0. \end{aligned} \tag{5.36.b}$$

Notice that the S_1 of this problem is the same as the S of the standard LQR problem. Thus the standard LQR is a special case of the model following LQR.

Finally, the control law is as follows:

$$
\begin{aligned}
u &= -R^{-1}B^T\lambda \\
&= -R^{-1}B^T(S_1x + S_2x_m) \\
&= -K_1x - K_2x_m
\end{aligned}
\tag{5.37}
$$

where,

$$K_1 = R^{-1}B^TS_1$$

$$K_2 = R^{-1}B^TS_2$$

5.3. Combined Control Simulations

The gross motion phase of the combined control is performed by the AMFC, discussed in Chapter IV. In the following simulations, the AMFC for gross motion and the LQR's for fine motion are studied in the frame work of combined control strategy.

The fine motion controller (LQR) is simulated for four different cases:

1. Regulator implementation: $u = -Kx \quad ; x = [\theta - \theta_f, \delta - \delta_f, \dot{\theta}, \dot{\delta}]$

$$a) \quad \delta_f = 0.0 \tag{5.38.a}$$

$$b) \quad \delta_f = \delta_{static} \tag{5.38.b}$$

2. Model following implementation: $\delta_f = \delta_{static}$

$$a) \quad u = -K(x - x_m) \tag{5.38.c}$$

$$b) \quad u = -K_1 x - K_2 x_m \tag{5.38.d}$$

The difference between the Case 1.a and 1.b may be important in large scale manipulators with large payloads, such as the one at Georgia Tech's Flexible Automation Laboratory. For a manipulator of that type, trying to stabilize the vibrations about the static equilibrium point is a more sound approach than trying to straighten an already deflected arm. For robustness simulations, the LQR problem is solved for two different payload cases: $m_p = 0.0kg.$ and $m_p = 2.0kg.$, and applied to the manipulator model with different payloads. The $((Q, R, \alpha), (A, B))$

elements of LQR formulation and the corresponding optimal feedback gains, with the resultant closed loop eigenvalues, are tabulated in Appendix D.

Discussed first are the simulations of the combined control under perfect conditions. Figures 5.3.a-b show the results of combined control tracking a desired motion (Fig.5.7.a-b), of bandwidth $w_{mi} = 5.5 rad/sec$. The gross motion phase is controlled by the same AMFC used in Chapter IV (Fig.4.19.c-d). The fine motion phase is controlled by the LQR (5.38.c) implementation, for three different values of the prescribed degree of stability, $\alpha = 0.0,\ 2.0,$ and 5.0 . Fig.5.3.a-b, and Fig.5.5.a-c, show that the combined control has very good performance. The joint variable tracking and flexible vibration stabilization are very well accomplished. No residual vibrations at the end of motion exists, while keeping the advantages of adaptive control in gross motion (Compare these figures with Fig.4.17.c-d, 4.19.c-d, and Fig.4.18.b, Fig.4.20.b).

Shown in Fig.5.4 are the results of combined control where LQR is implemented in four different forms, (5.38.a,b,c,d), with the same gains of the previous simulation (Fig.5.3), for $\alpha = 5.0$ case. The objective is to determine the performance difference between the implementations of LQR. First of all, the (5.38.d)

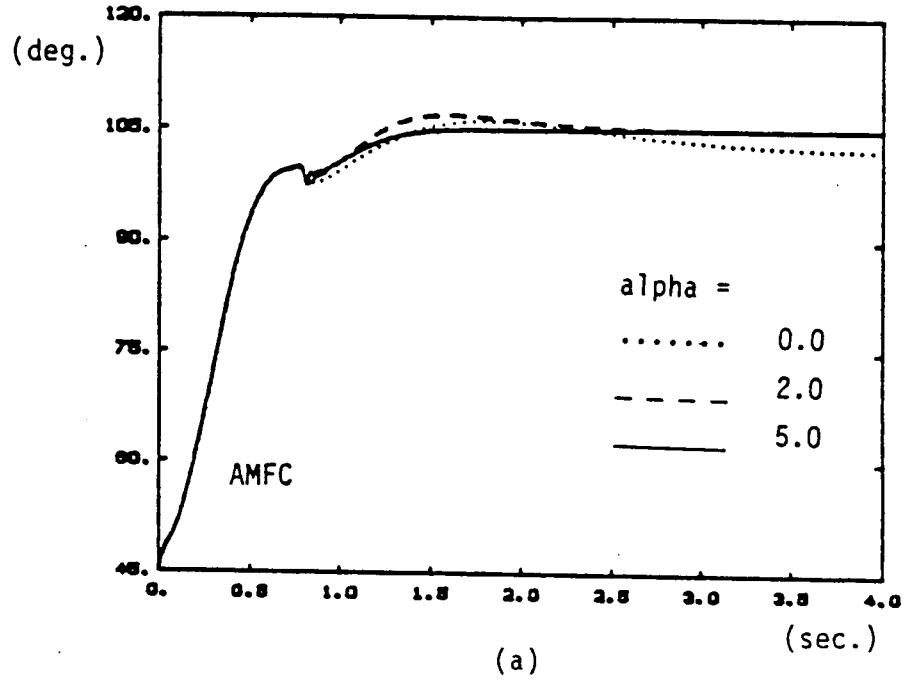
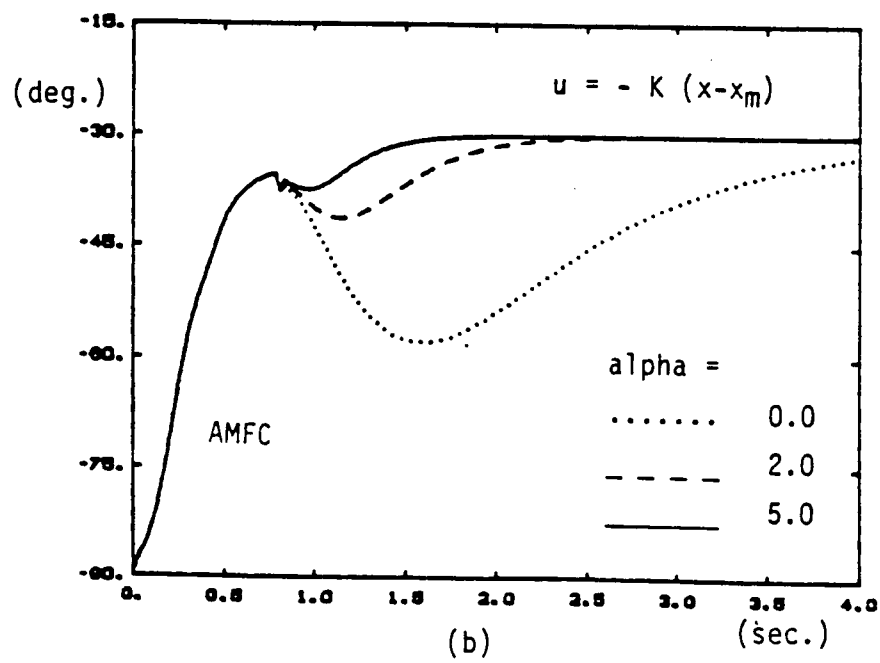


Fig.5.3.a-b Combined control (AMFC followed by model following LQR) a) joint 1 response, b) joint 2 response,



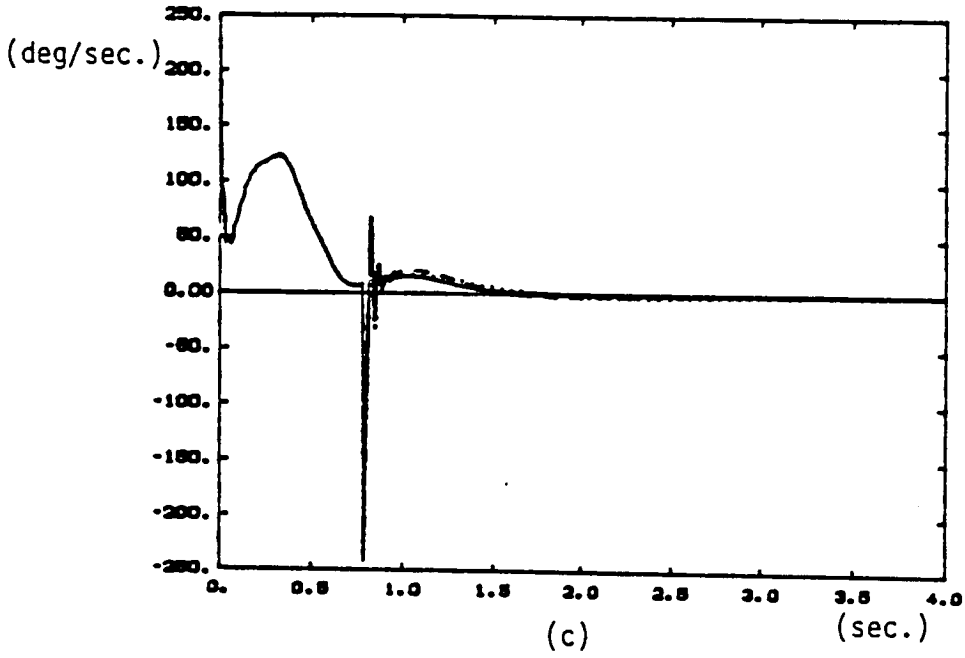
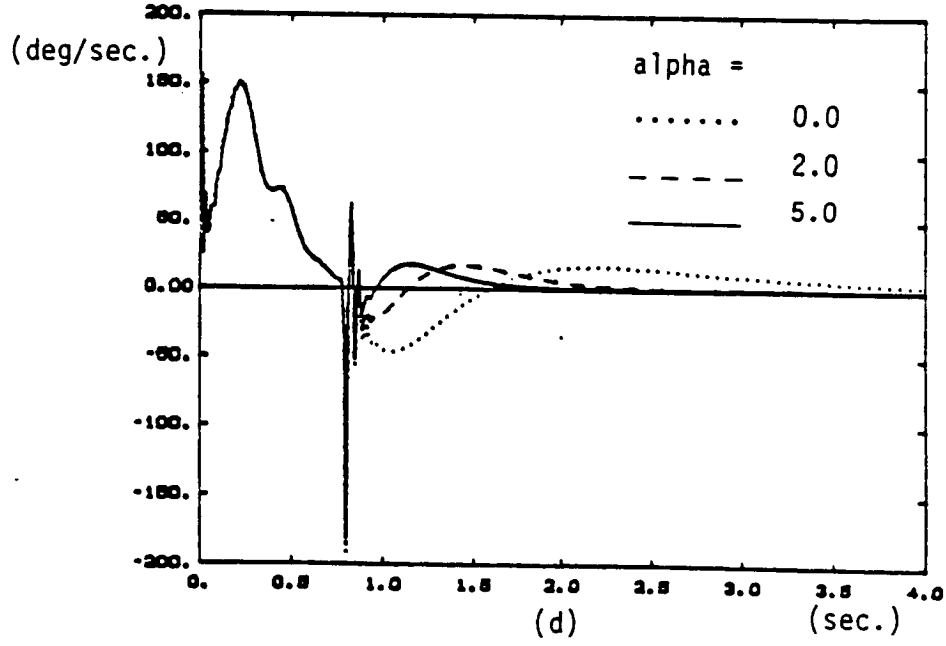


Fig.5.3.c-d Combined control (AMFC followed by model following LQR)
c) joint 1 speed, d) joint 2 speed,



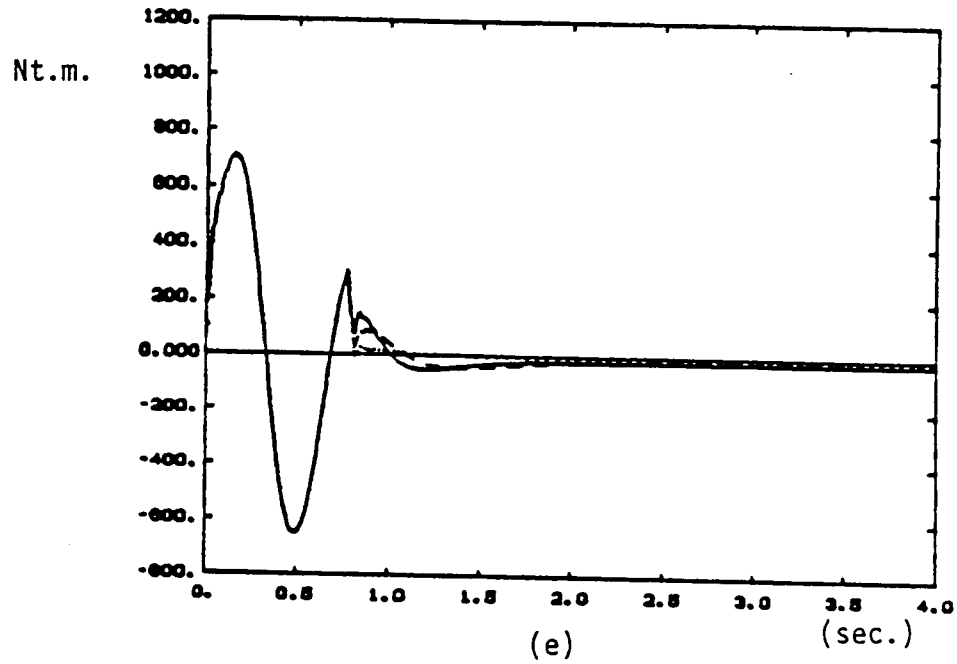
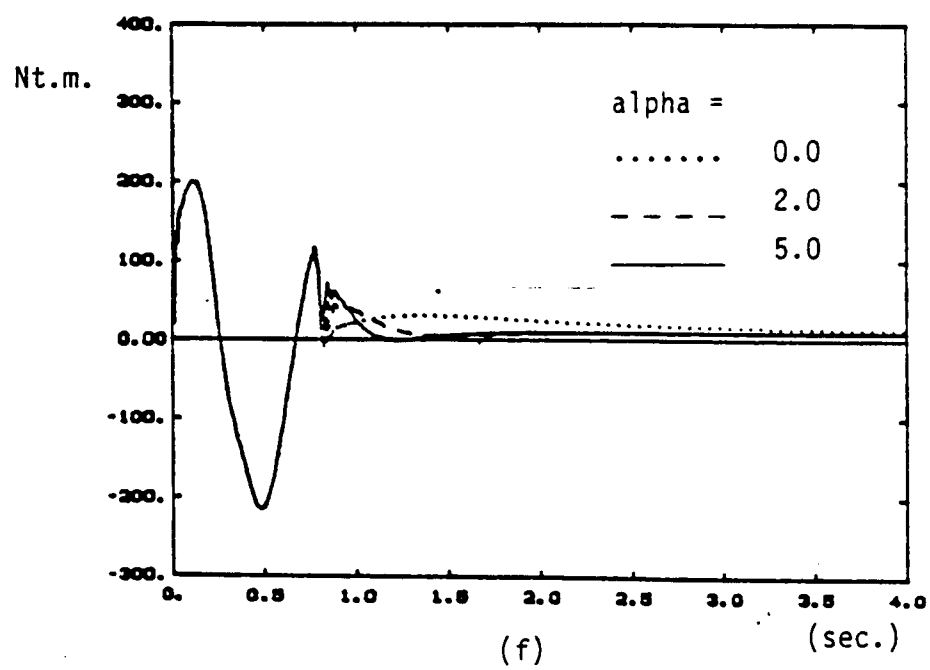


Fig.5.3.e-f Combined control (AMFC followed by model following LQR)
e) torque history of joint 1, f) torque history of joint 2



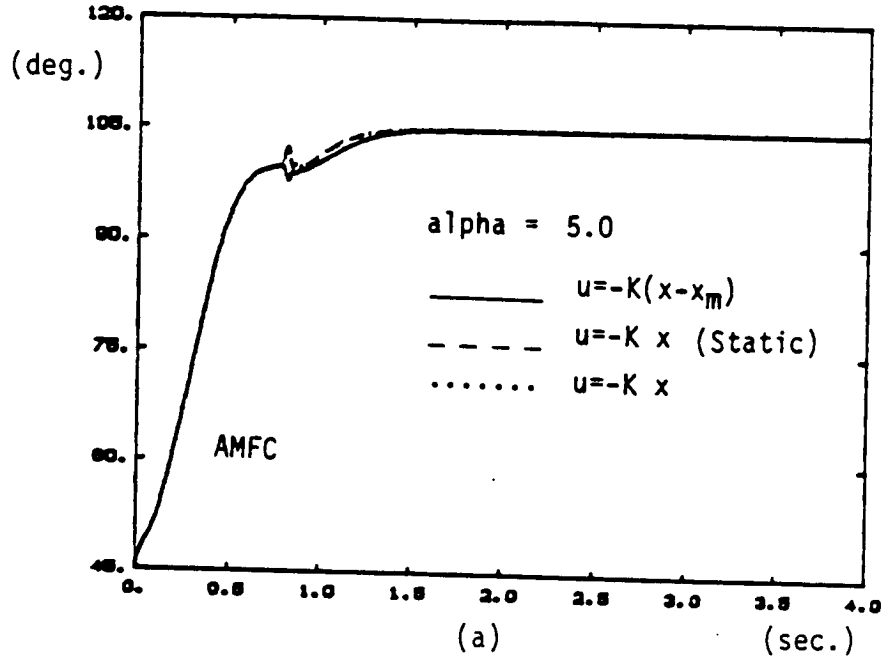
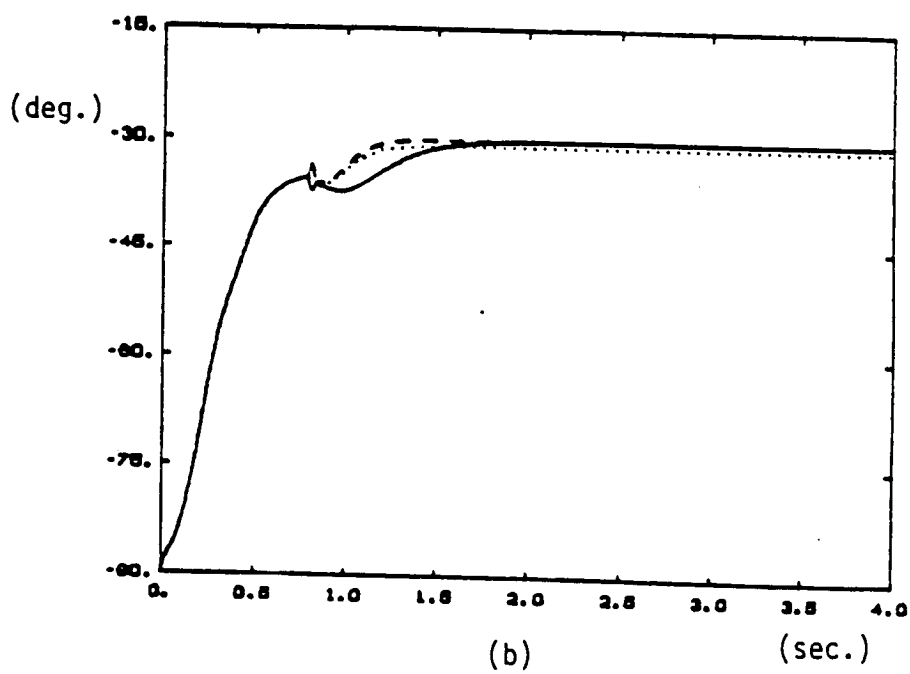


Fig.5.4.a-b Combined control (AMFC followed by standard LQR) under perfect conditions. a) joint 1 response, b) joint 2 response.



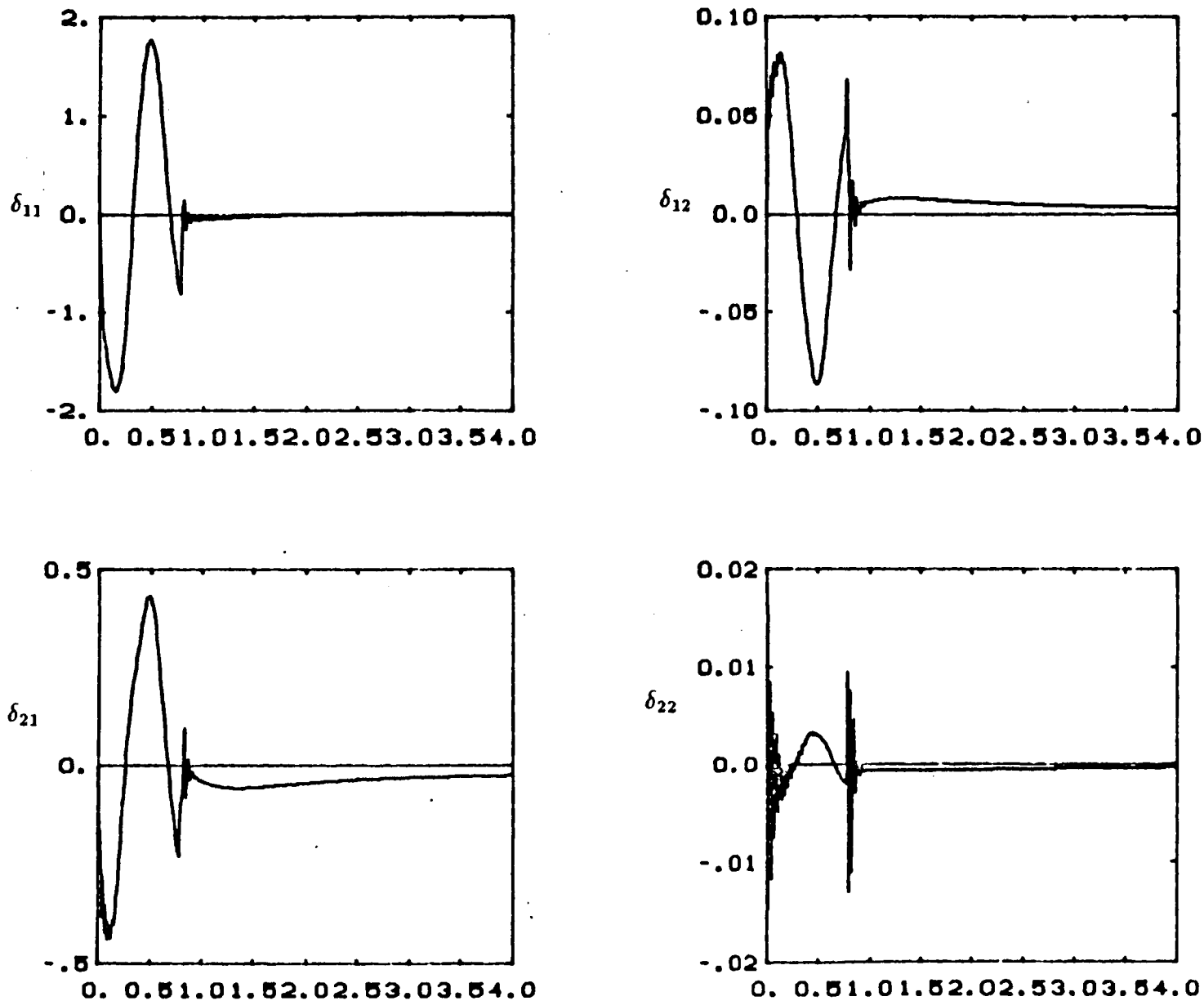


Fig.5.5.a Flexible mode shape magnitude response associated with motion shown in figure 5.3, for alpha= 0.0 case

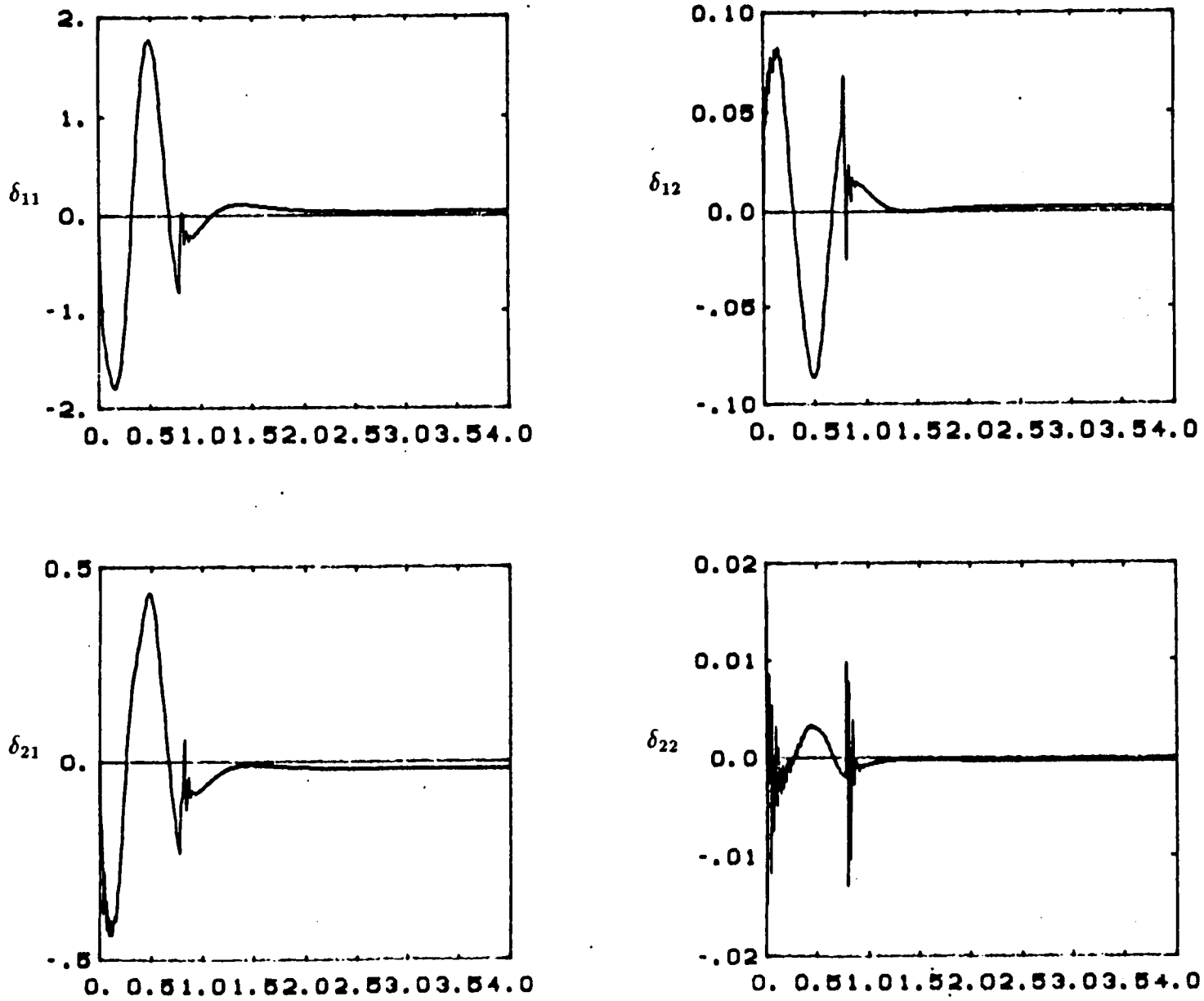


Fig.5.5.b Flexible mode shape magnitude response associated with motion shown in figure 5.3. for alpha= 2.0 case

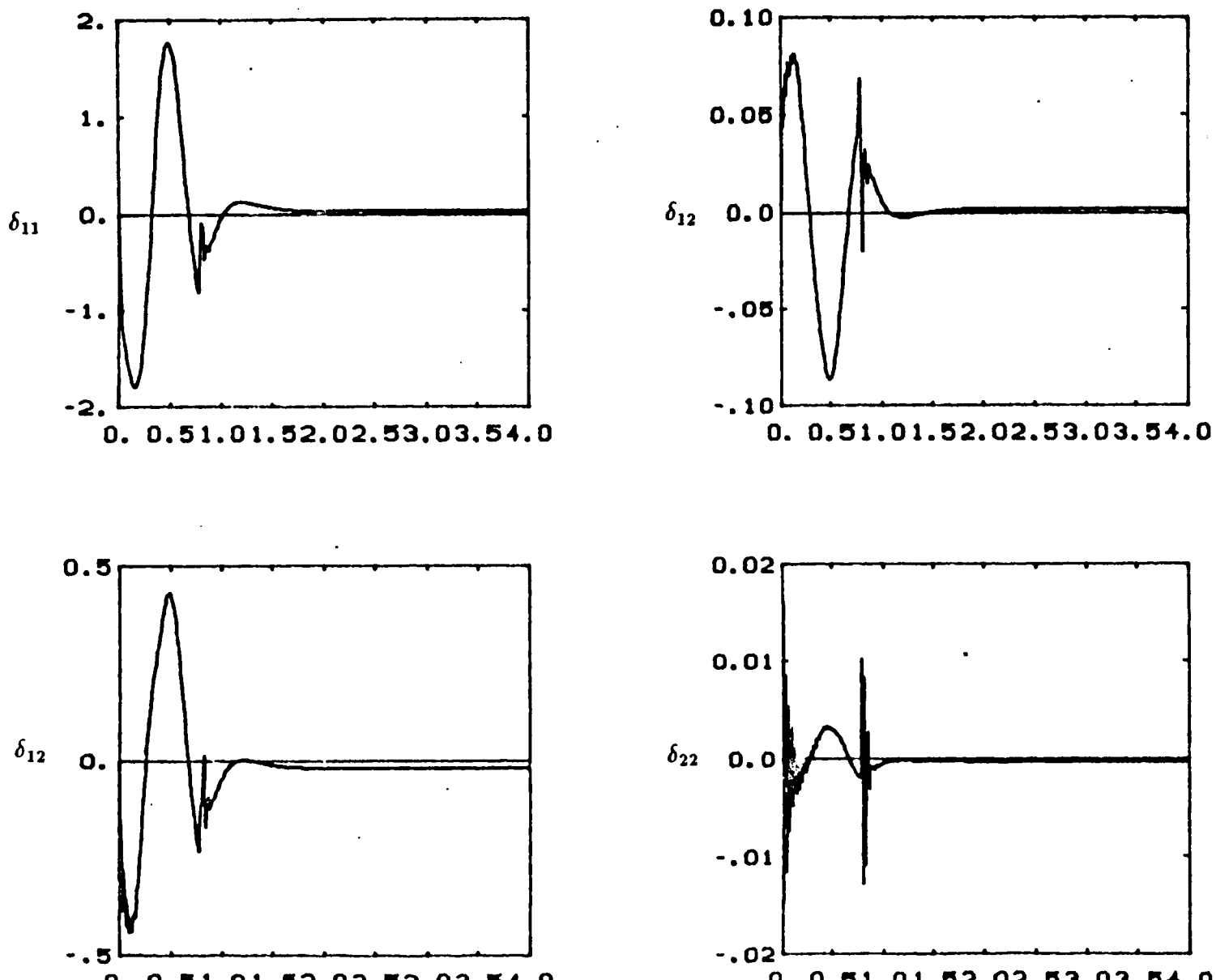


Fig.5.5.c Flexible mode shape magnitude response associated with motion shown in figure 5.3, for alpha= 5.0 case

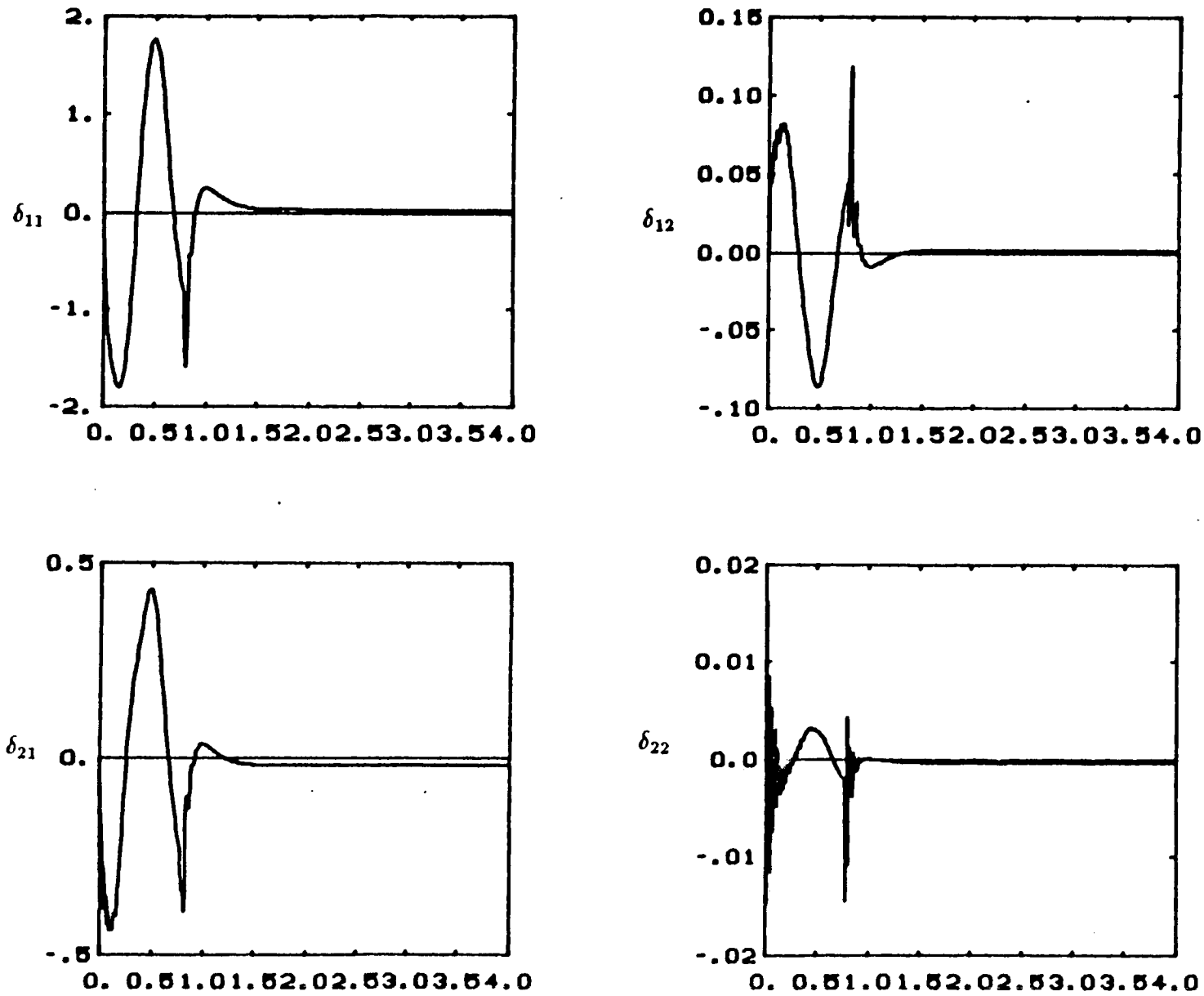


Fig.5.6.a Flexible mode shape magnitude response associated with motion shown in figure 5.4, for LQR # 1.

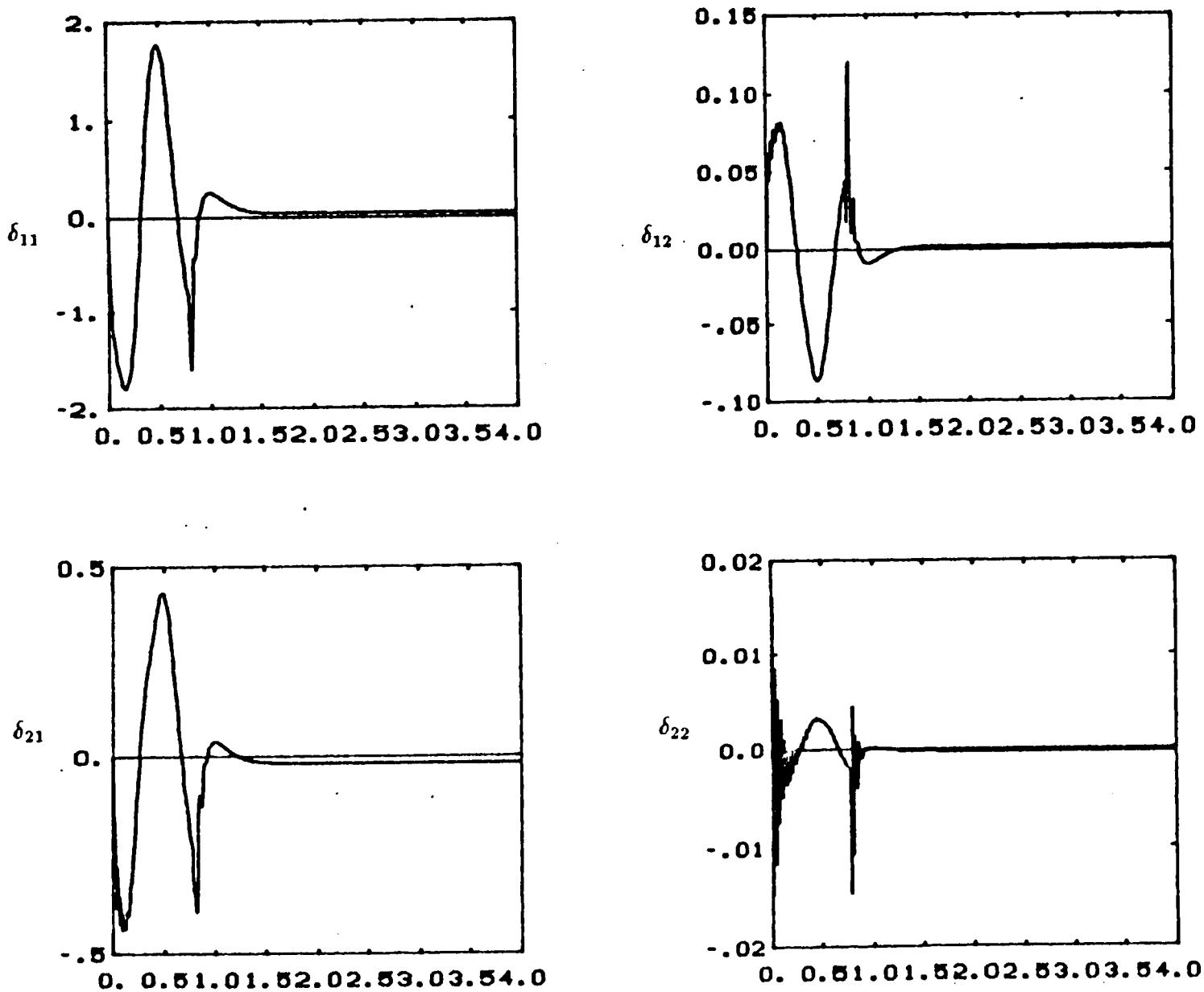


Fig.5.6.b Flexible mode shape magnitude response associated with motion shown in figure 5.4 for LQR # 2

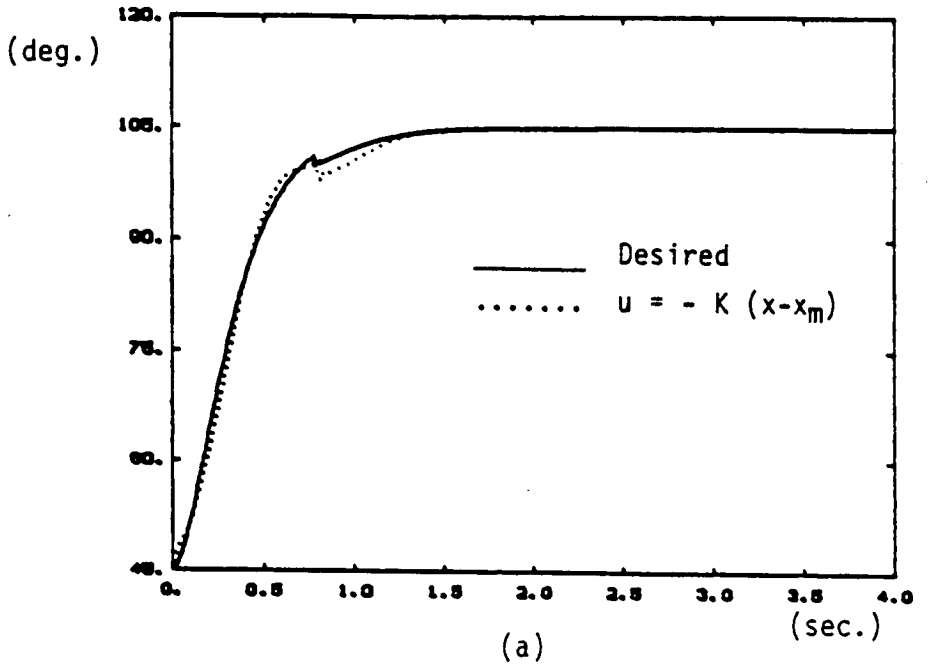
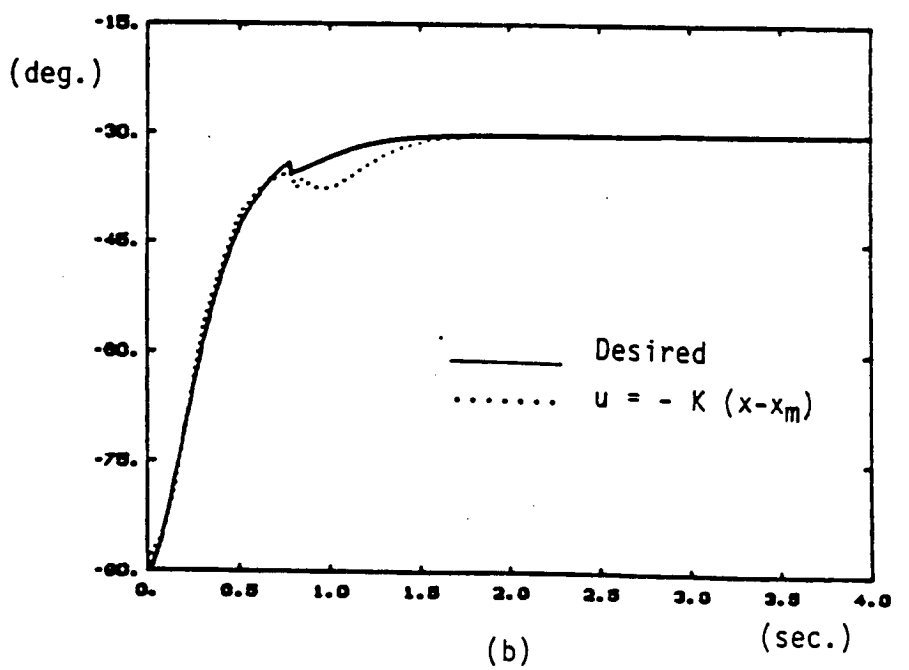


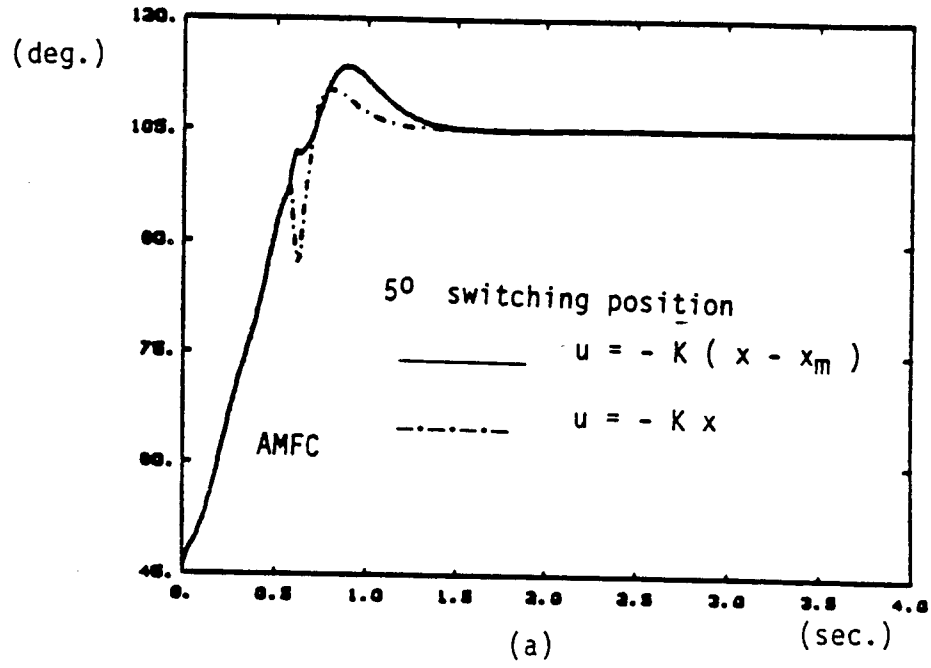
Fig.5.7.a-b Desired joint trajectories generated by the reference model of gross and fine motion phases a) Joint 1, b) Joint 2.



case was in almost all cases indistinguishable from the (5.38.b) case. If the feedback gains are examined (Appendix D), it is found that the gain K_2 is very small compared to K_1. Therefore, it is not surprising that standard regulator and optimal model following regulator implementations have almost identical results. After all, optimality with respect to some scalar criteria does not necessarily imply good time domain performance. Furthermore, there is not a significant difference between the (5.38.a) and (5.38.b) implementations of the fine motion controller. However, this largely depends on the manipulator parameters and payloads involved in the task, and would not be true in general. Finally, regulator implementation gives a faster response in the joint variables (Fig. 5.4.), but results in larger flexible mode deflections during the initial period of switching to fine motion, compared to the model following implementation (compare figures 5.6.a-b, and 5.5.c).

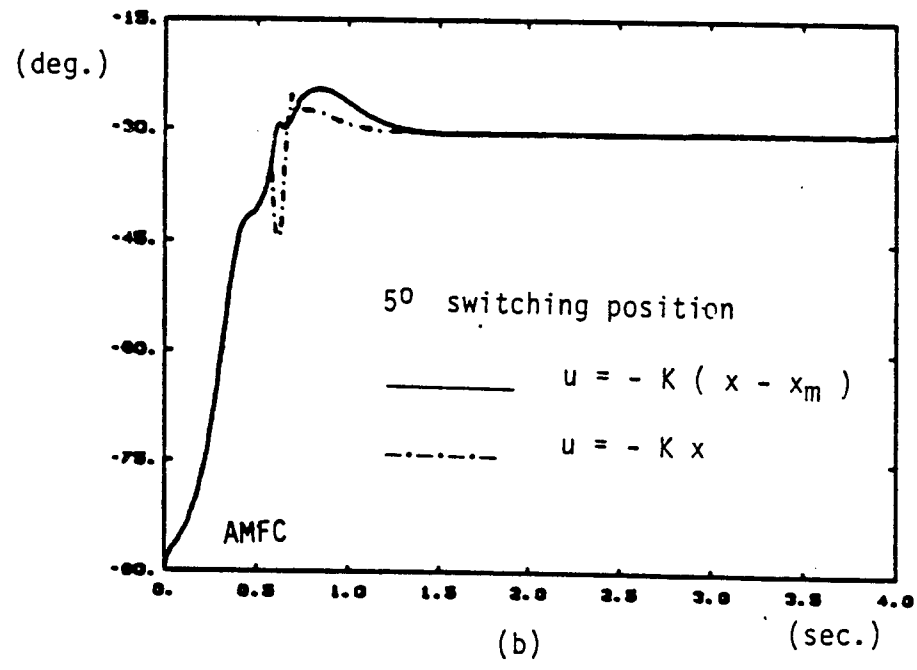
The rest of the simulations test the robustness of the combined control with respect to payload variations. The fine motion control is simulated for the (5.38.b) and (5.38.c) cases only. For the same desired motion, the manipulator model has a 2.0 kg payload. The AMFC makes calculations based on $m_p = 0.0kg.$, and LQR gains are calculated for $m_p = 2.0kg$. Figures 5.8.a-b show that the resultant performance is not satisfactory. It was thought that if the switch to the fine motion controller were made earlier so that it would have more time to position the arm and stabilize the vibrations, the resultant performance might get better. Unfortunately, figures 5.9.a-b show that switching earlier does not help improve the performance at all. Why is the fine motion control performance is poor despite the good closed loop eigenvalue locations? The answer may be found in the assumptions made during the development of this particular fine motion controller. For the linear

controller to be successful, the assumptions associated with it must be accurate enough. For example, the controller should be used in the vicinity of nominal state about which it is designed for, and the speeds should be low enough for the linear analysis to be accurate. Therefore, if the speeds before switching to linear fine motion controller are too high, then the performance will be poor as seen here. This explanation is further supported by the following simulations. The same controller is simulated for following a desired motion of bandwidth $w_{mi} = 2.75 rad/sec$. The only difference is the desired motion speed, and of course AMFC is matching with the corresponding slower reference model. Fig.5.10 shows the results. Now it is clear that when the speed before the switching time is slow enough, the fine motion controller will succeed, and the combined control will result in good performance. Fig.11 shows the case where the manipulator has no payload, but LQR designed for $m_p = 2.0kg$. payload. The results are still good. Flexible mode responses along the associated motions are given in figures 5.12 - 5.14. When the figures 5.12.a and b, and Fig.5.13.a and b, are compared, we see that regulator implementation results in much larger deflections than the model following implementation, confirming the discussion of Section 5.2.3.



(a)

Fig.5.8.a-b Robustness of combined control with respect to payload variations a) joint 1 response, b) joint 2 response



(b)

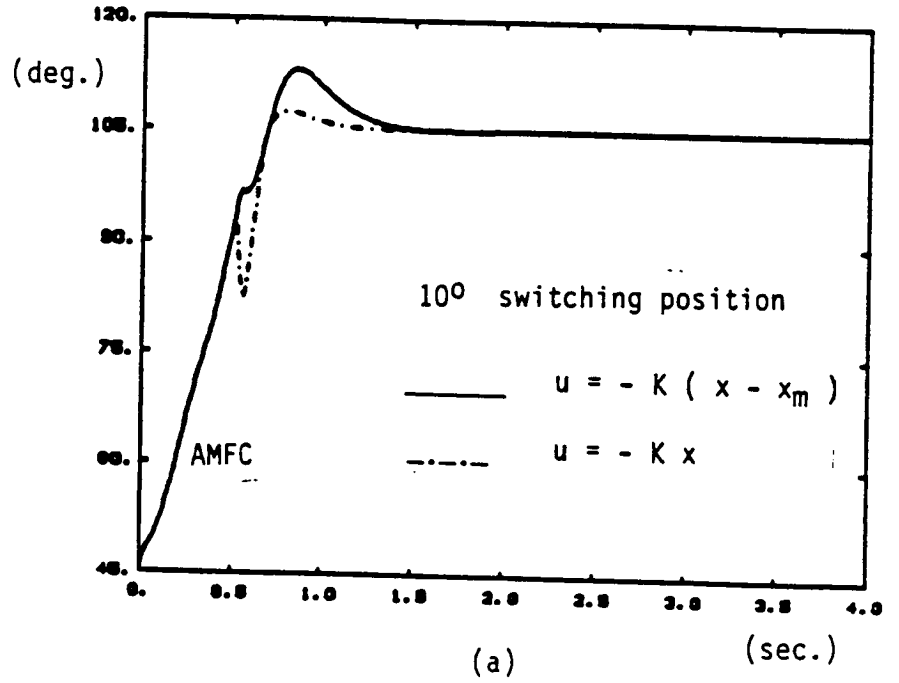
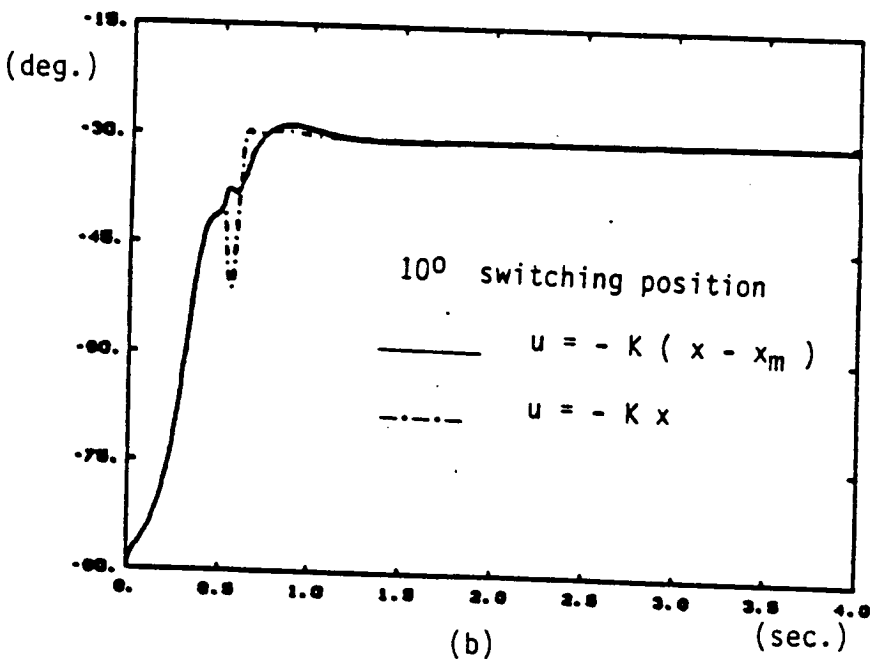


Fig.5.9.a-b Robustness of combined control with respect to switching time speed of manipulator. a) joint 1 response, b) joint 2 response.



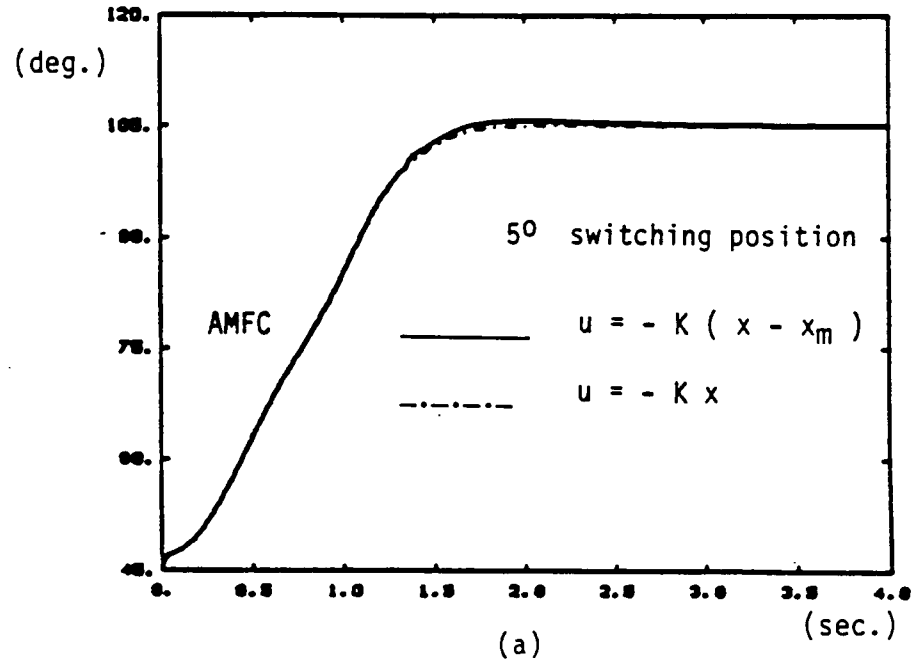
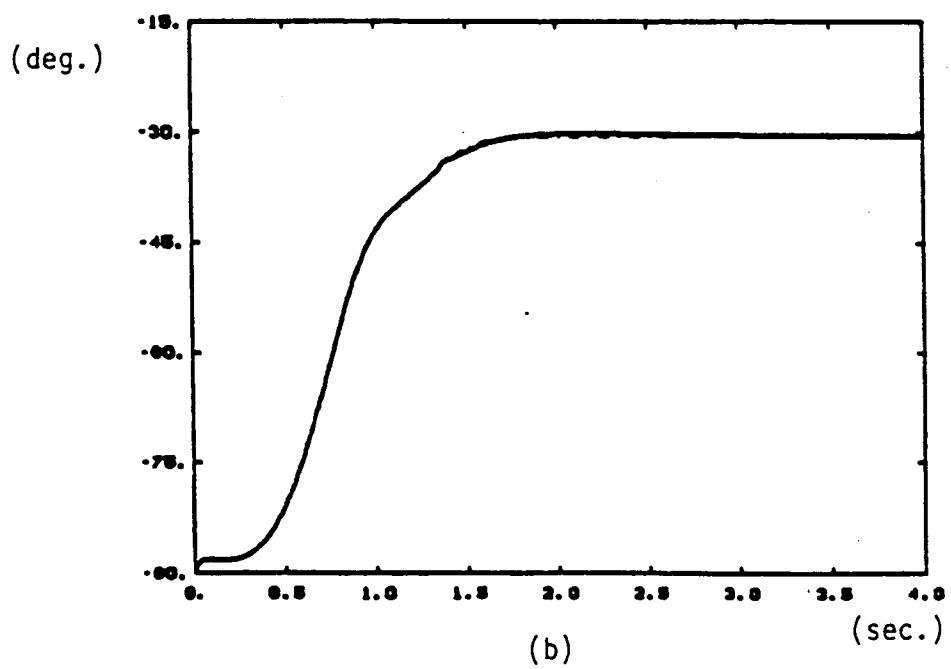


Fig.5.10.a-b Robustness of combined control with respect to payload variations in relatively slow motions ($mp_{actual} = mp_{lqr} = 2.0kg, mp_{amfc} = 0.0kg$). a) joint 1 response, b) joint 2 response.



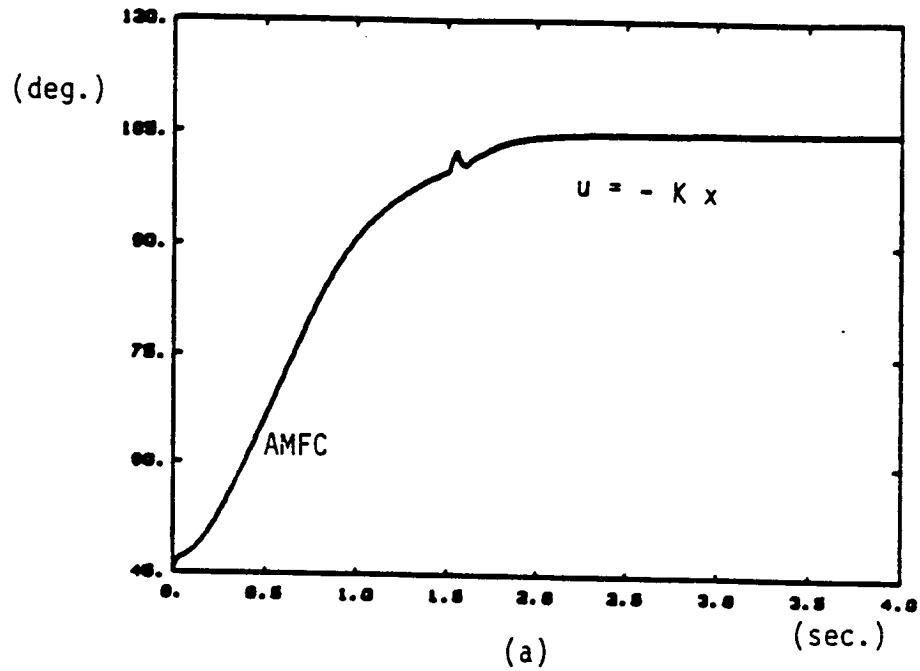
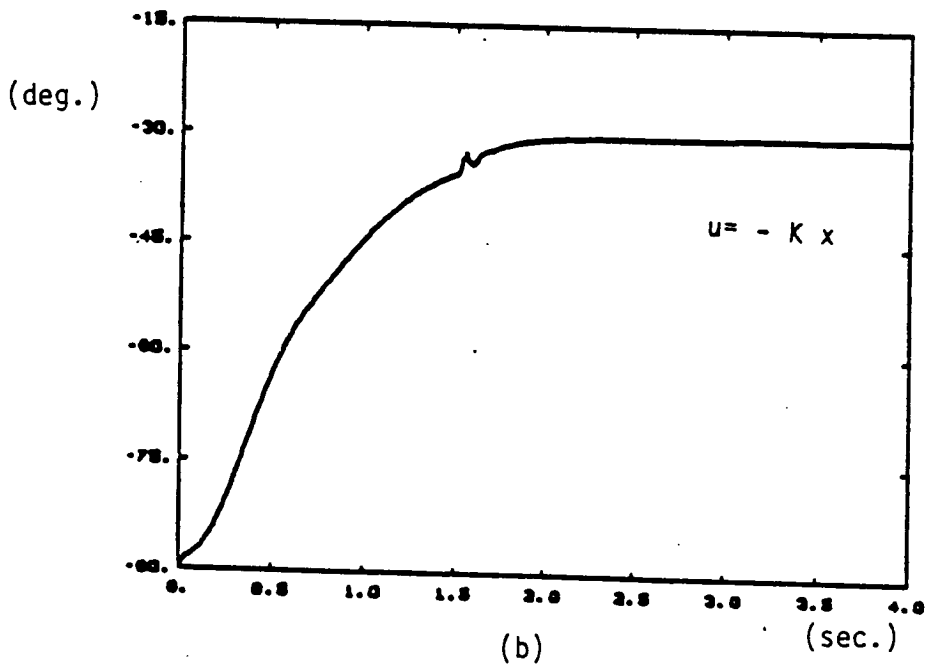


Fig.5.11.a-b Robustness of combined control with respect to payload variations in relatively slow motions a) joint 1 response, b) joint 2 response.



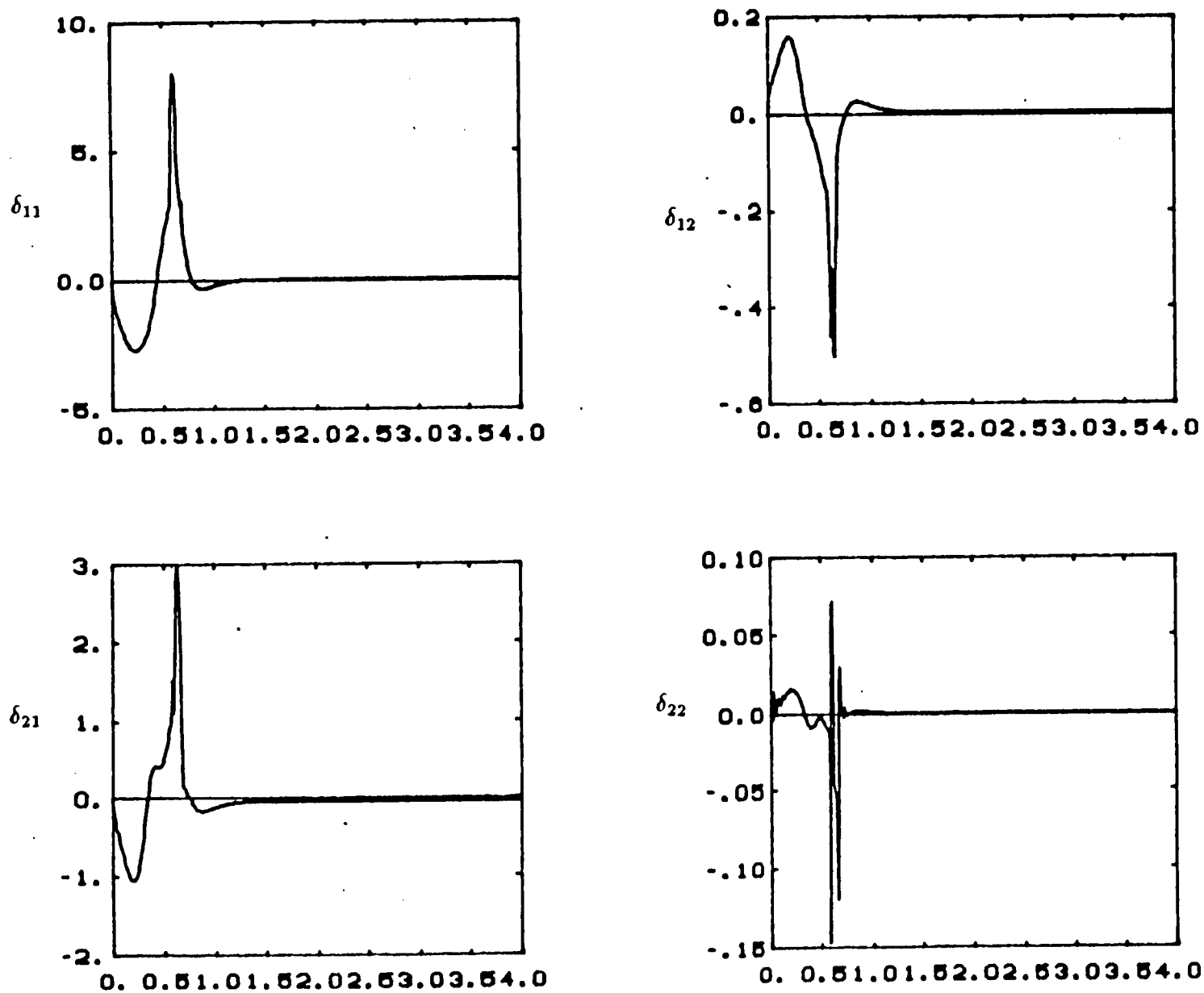


Fig.5.12.a Flexible mode shape magnitude response along the motion described by figure 5.8, a) Standard LQR results.

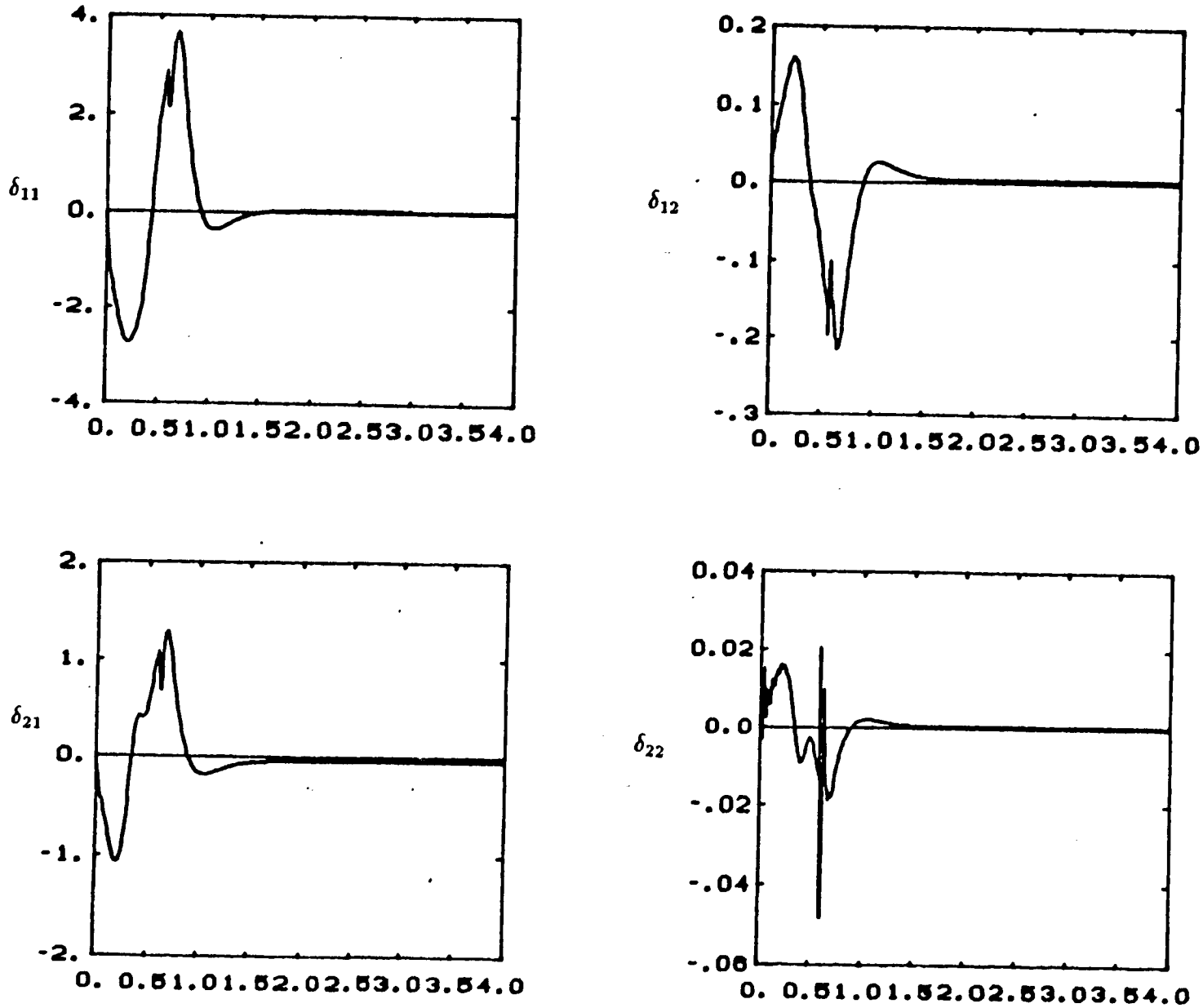


Fig.5.12.b Flexible mode shape magnitude response along the motion described by figure 5.8, b) Model following LQR results.

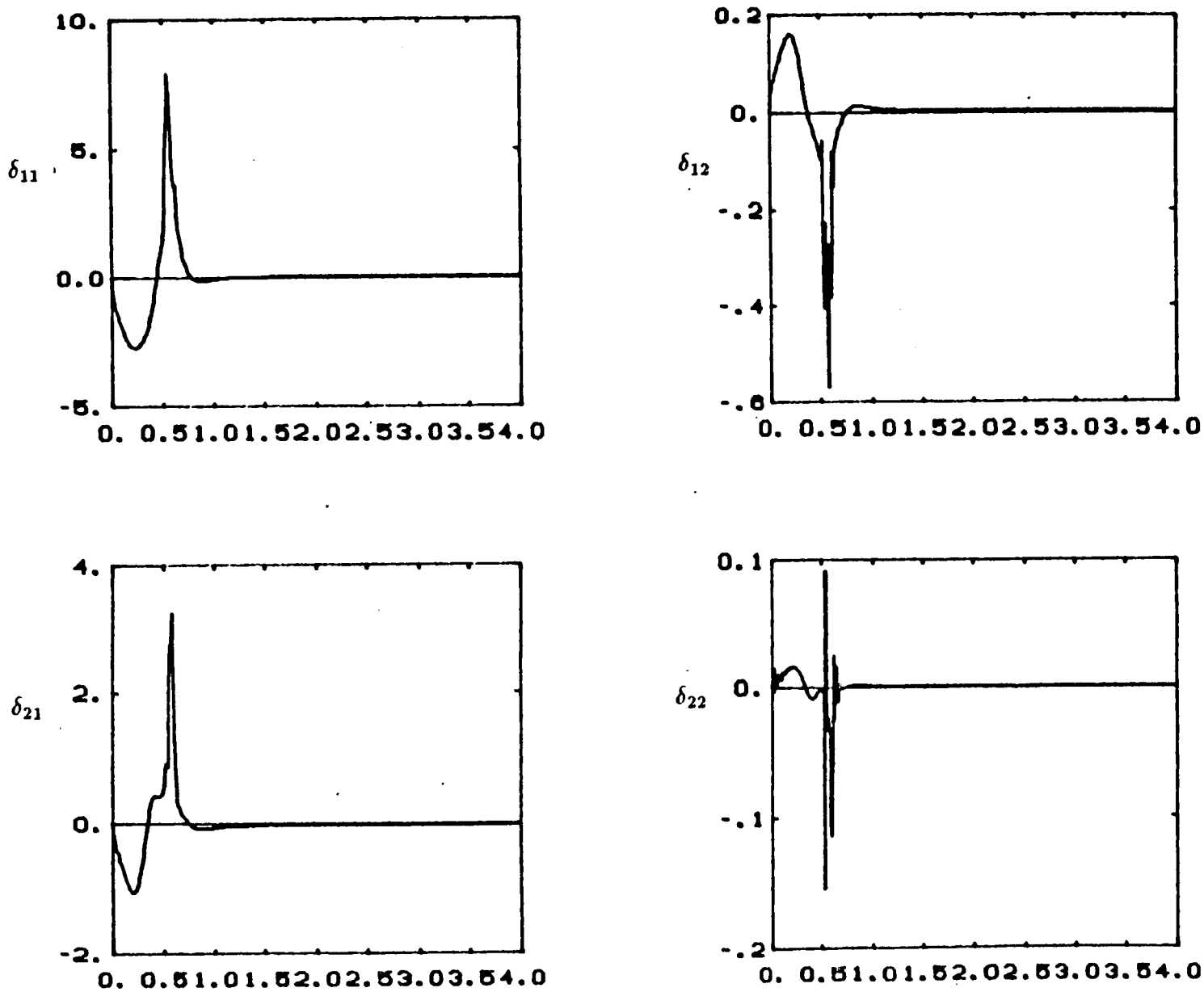


Fig.5.13.a Flexible mode shape magnitude response along the motion described by figure 5.9. a) Standard LQR results.

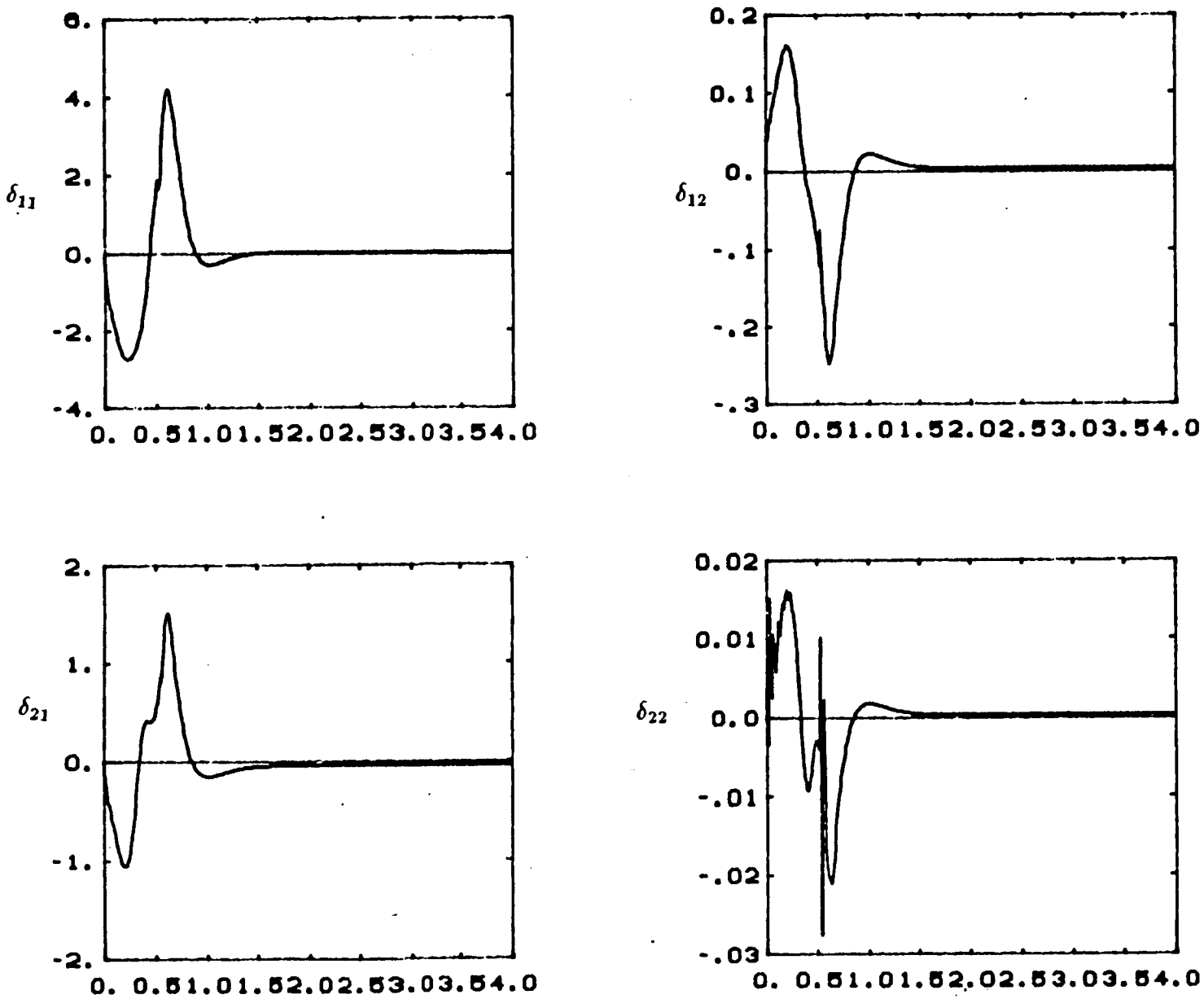


Fig.5.13.b Flexible mode shape magnitude response along the motion described by figure 5.9, b) Model following LQR results.

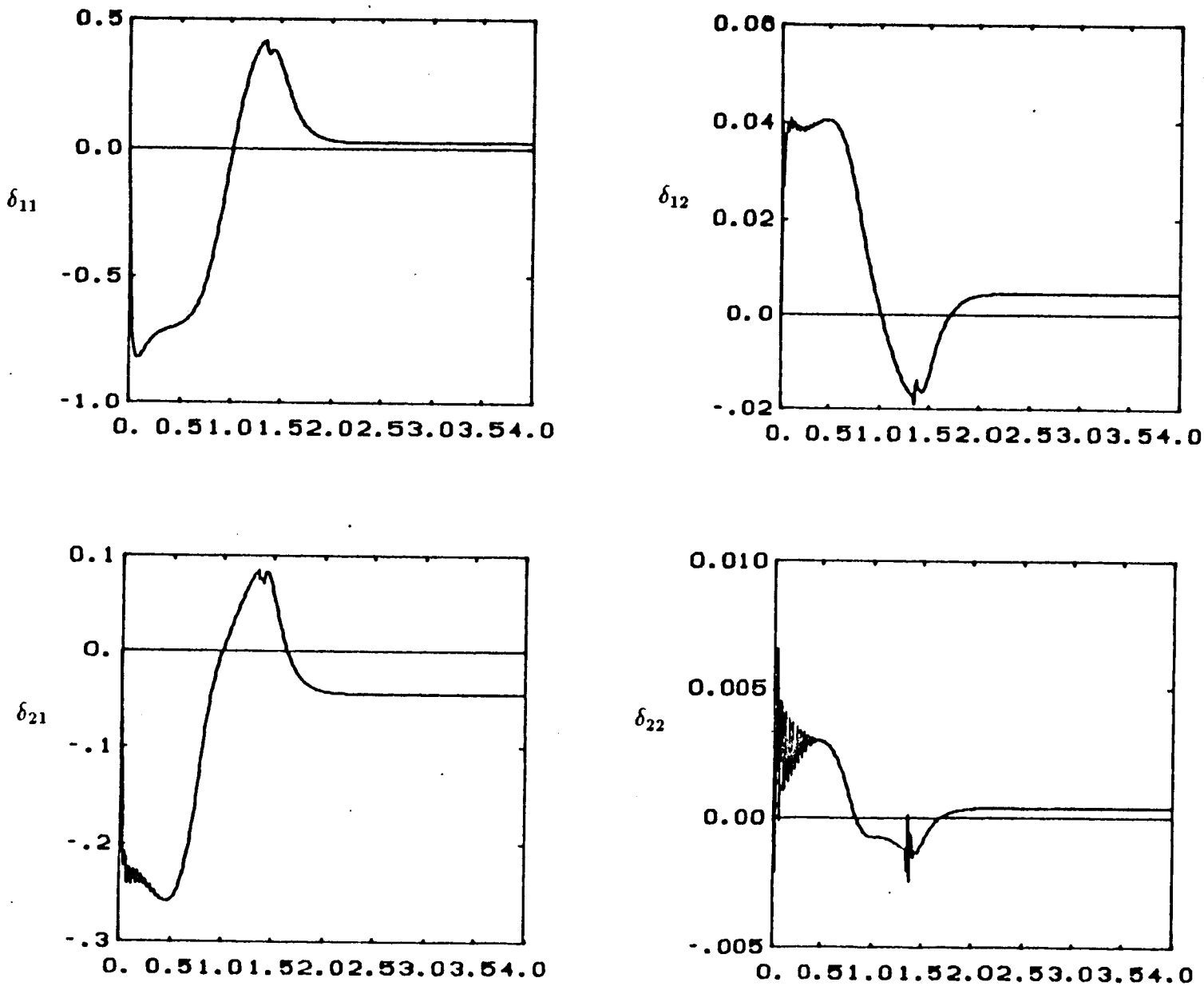


Fig.5.14.a Flexible mode shape magnitude response along the motion described by figure 5.10, a) Standard LQR results,

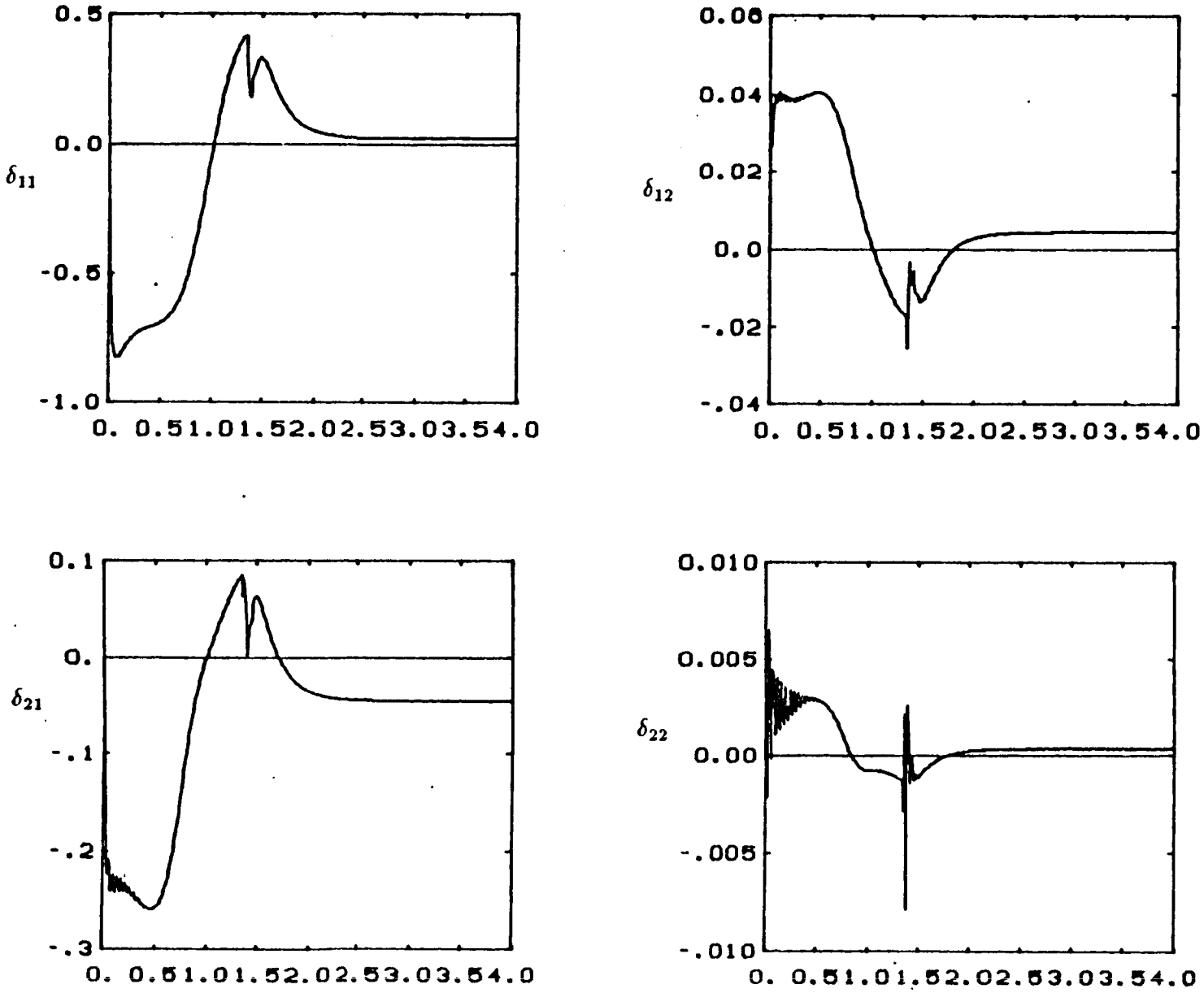


Fig.5.14.b Flexible mode shape magnitude response along the motion described by figure 5.10.

Final Note:

There are a number of issues that must be addressed before final conclusions made. First, the stability of the closed loop system under partial state feedback during fine motion phase must be studied. Second, the joint velocity response shows very sharp changes at the switching time (Fig.5.3.c-d). The practical implications of this result and if it would still happen under more realistic conditions must be determined.

Every flexible manipulator is an infinite dimensional dynamic system. Any finite dimensional state feedback is in fact a partial state feedback controller. A finite dimensional controller is designed based on an approximate finite dimensional model, and applied to an actual system which has more states that the controller design model has ignored. The actual measurements in the real world implementations will be have components from the dynamics truncated by the design model. This is so called observation spill-over. Moreover, the controller which is designed to control a finite number of states will affect the uncontrolled modes as well. This is called control spill-over effect [A18]. It is found that the closed loop system is always stable if only the joint variable feedback is used. However, if some of the flexible modes are also used in the feedback in addition to the joint variables, the closed loop system is conditionally stable. These results are obtained through the closed loop eigenvalue analysis of the system under partial state feedback laws obtained from lQR formulation. For the manipulator model used in the simulations, a linear controller that uses joint variables and first flexible modes of each link (no feedback from second flexible mode shape coordinates of link 1 and 2) is simulated in the fine motion phase. Fig.5.15 show the response of the system. The partial

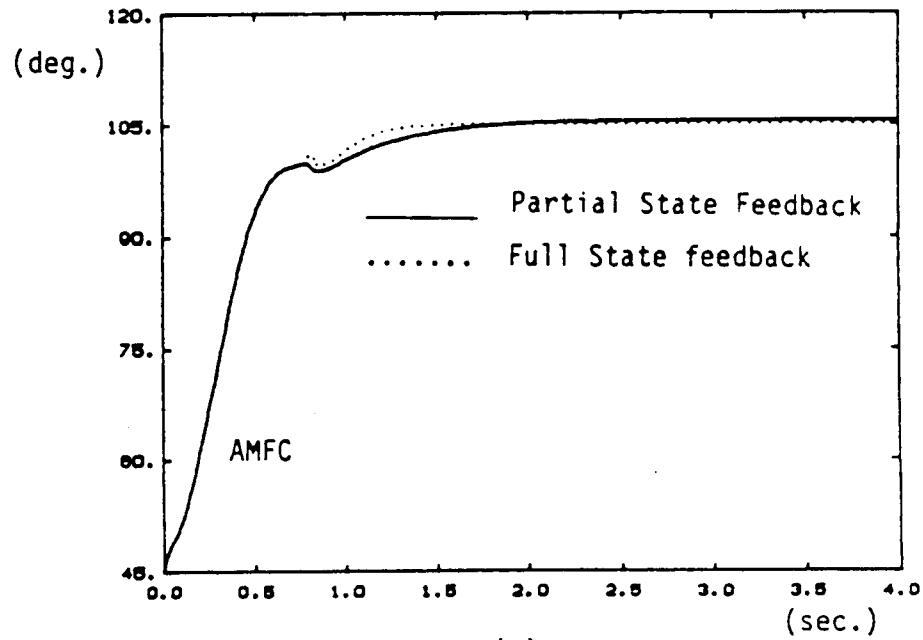
state feedback gain and the resultant closed loop eigenvalues are given on the figure. Despite the conditional stability of the partial state feedback control, it is possible to achive good closed loop response.

Regarding the second problem, it is suggested that the joint velocity discontinuity occurs because of not having any rotary inertia at the joints. When a step change in the input torque occurs at the switching time due to control law change, the base of the arm immidiately reacts to that and cause the jump in the velocity reponse. If this explanation is true, the veclocity response (Fig. 5.3.c-d) should become smoother when realistic joint inertias are added to the joints. Joint inertia properties are determined from the commercially available moving-coil, permanent magnet, D.C. motors. Motor selection is made based on the maximum torque needed for the range of motions simulated for the manipulator model. A gear ratio of 100:1 is assumed for each joint. Using the results of Sangveraphunsiri, [E9 (Appendix C)], appropriate D.C. motors can be selcted from the Electrocraft E-series. Based on the manifacturer's data, the selected motors has the following effective inertial proporties at the joints (effective mass moment of inertia = mass moment of inertia of the motor x (gear ratio squared)):

$$mj_1 = 15.kg, \qquad jj_1 = 2.0kg.m$$

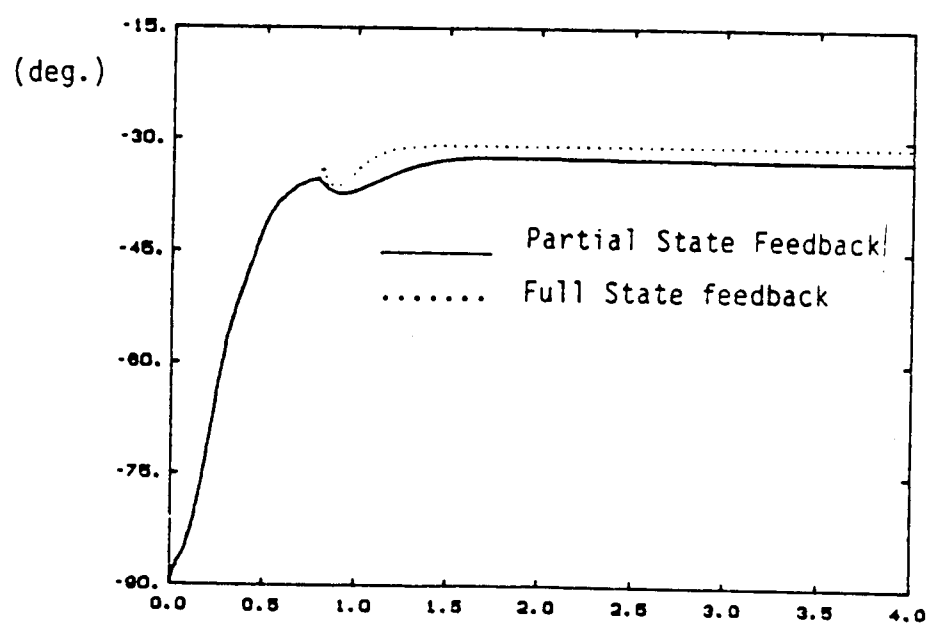
$$mj_2 = 4.0kg., \qquad jj_2 = 0.2kg.m$$

The nonlinear model is again linearized about the final desired state for the new parameters, the optimal control feedback gains are obtained using the LQR formulation. Fig.5.16 shows the response of the system with the new more realistic parameters. As seen from fig. 5.16.c-d, the velocity response no longer has step change. It can be shown that the torque step change magnitude shown in Fig. 5.16 is indeed large enough to cause such a response in the joint velocity response.



(a)

Fig.5.15.a-b Combined control response - fine motion controlled by a linear partial state feedback, a) joint 1, b) joint 2.



(b)

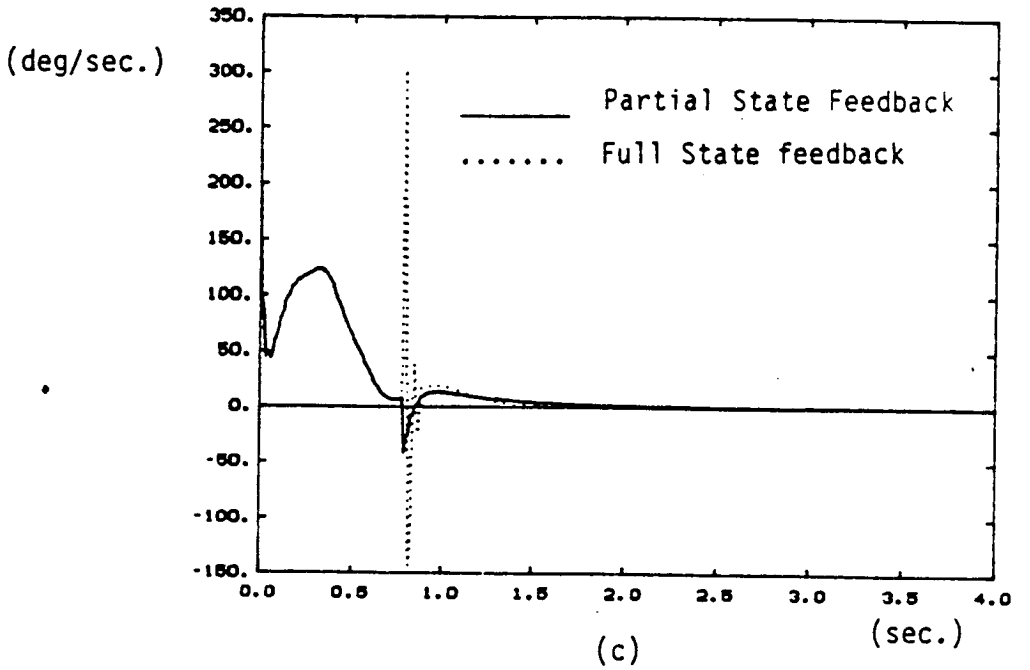
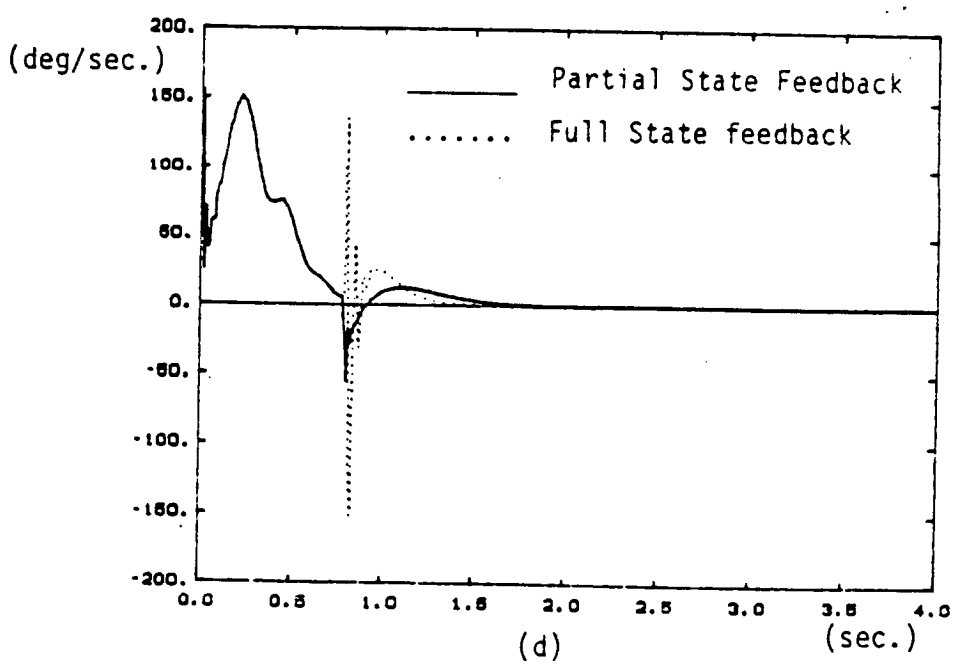


Fig.5.15.c-d Combined control response - partial state feedback c) joint 1 speed response, d) joint 2 speed response.



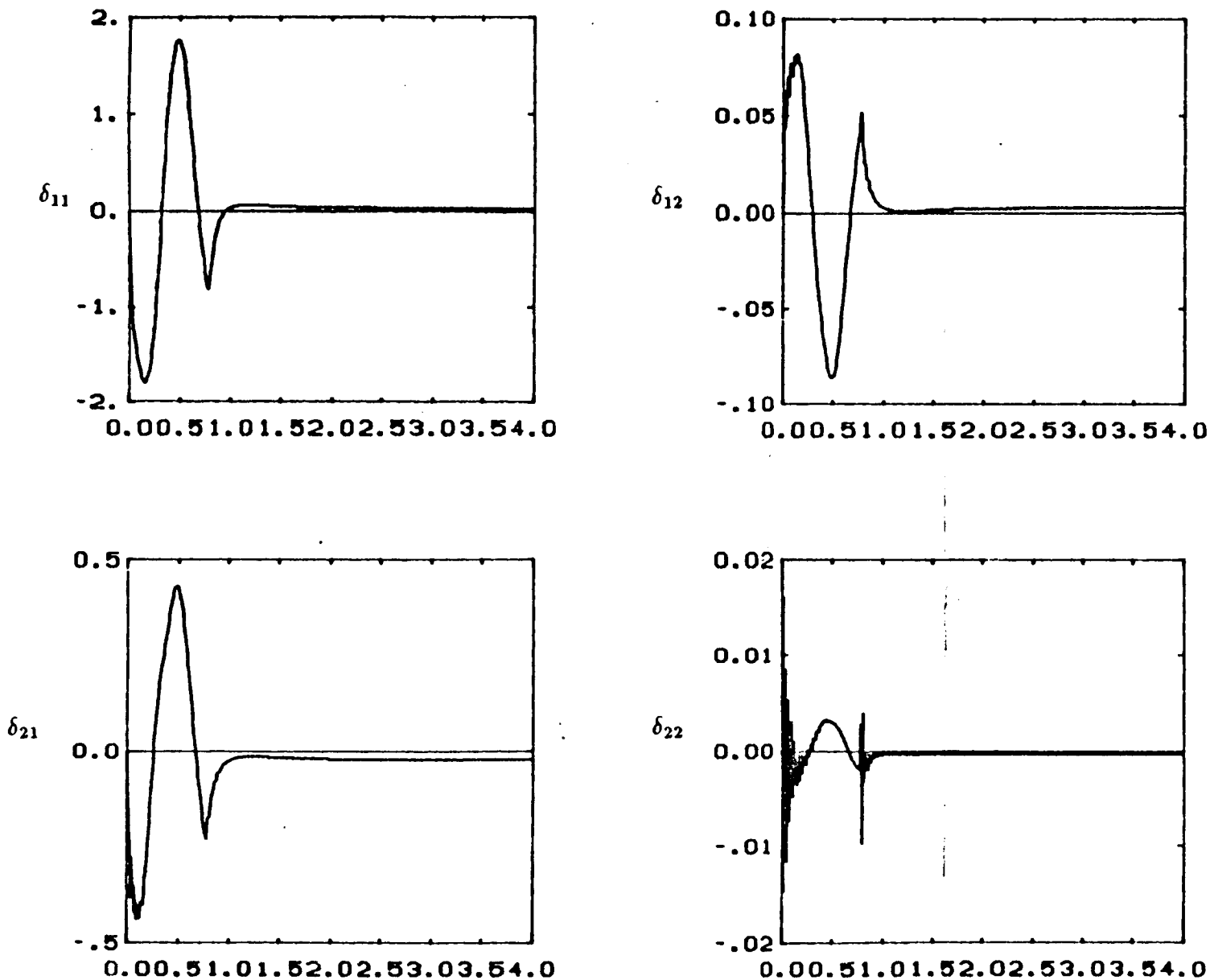


Fig.5.15.e Flexible mode shape magnitude responses along the motion described by figure 5.15.a-d.

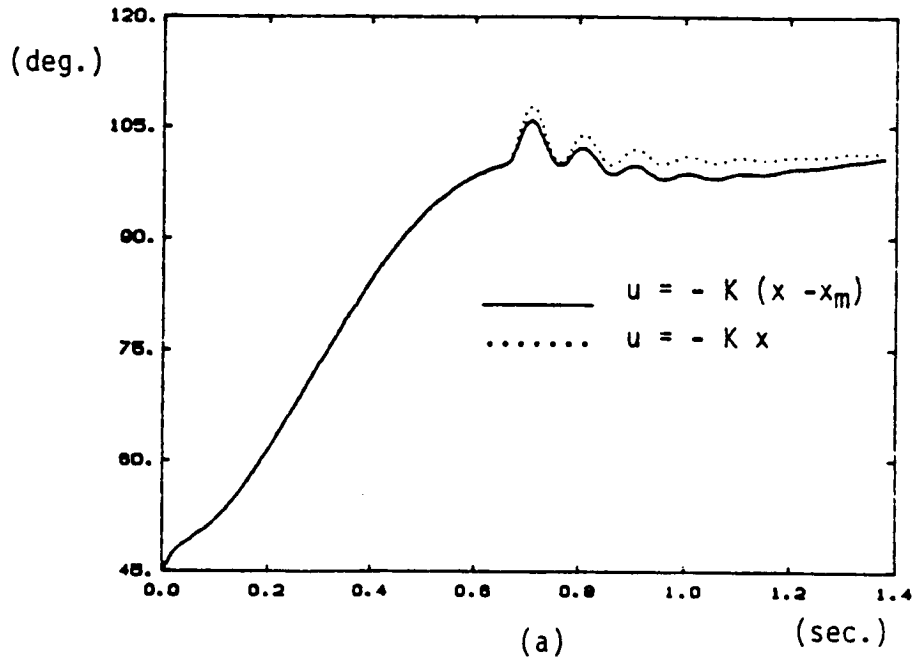
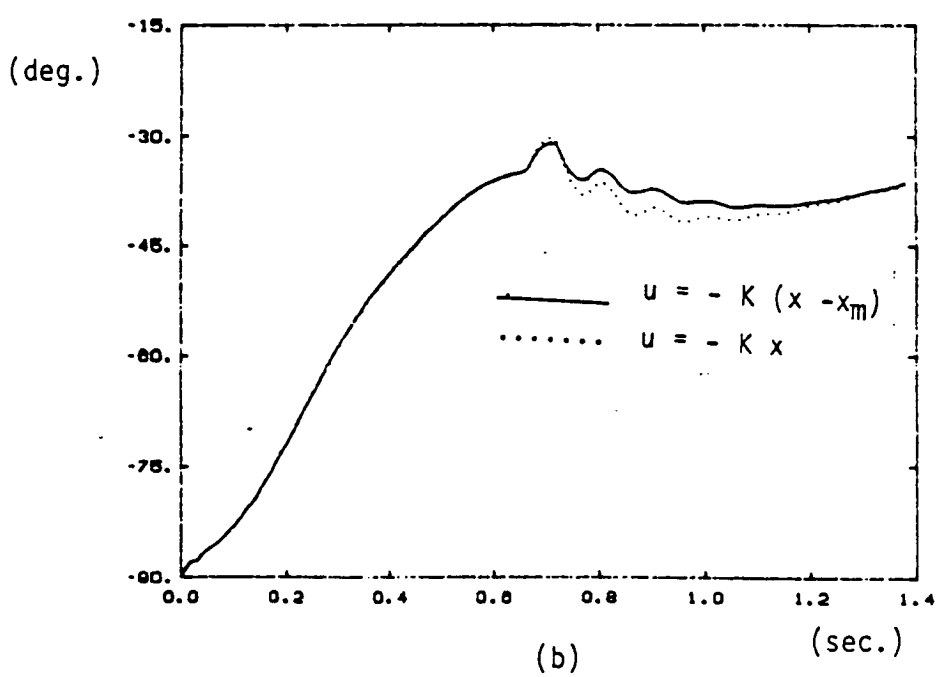


Fig.5.16.a-b Combined control response - with realistic joint inertial values.
a) joint 1 angle response, b) joint 2 angle response.



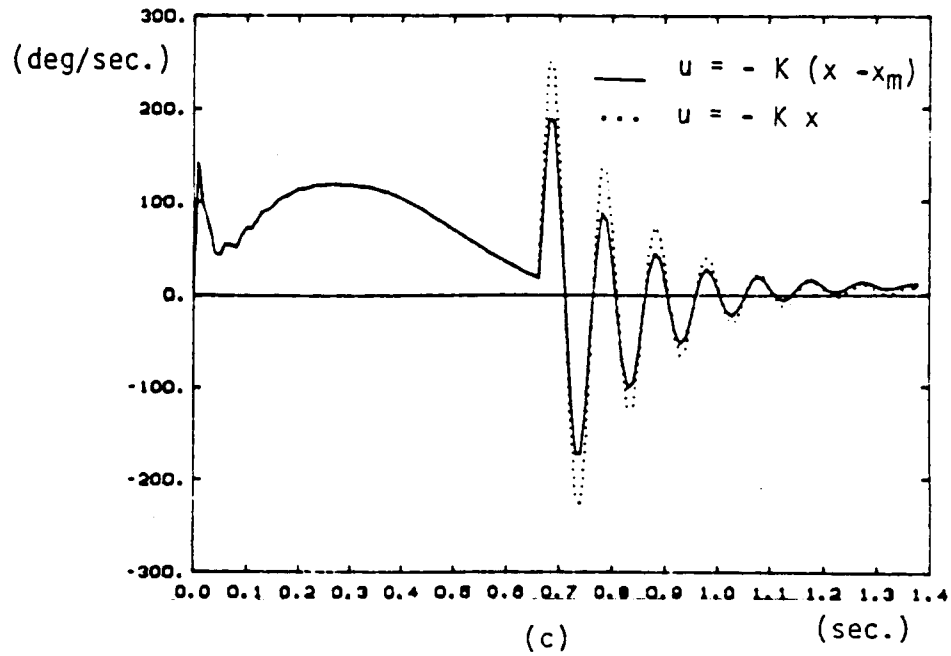
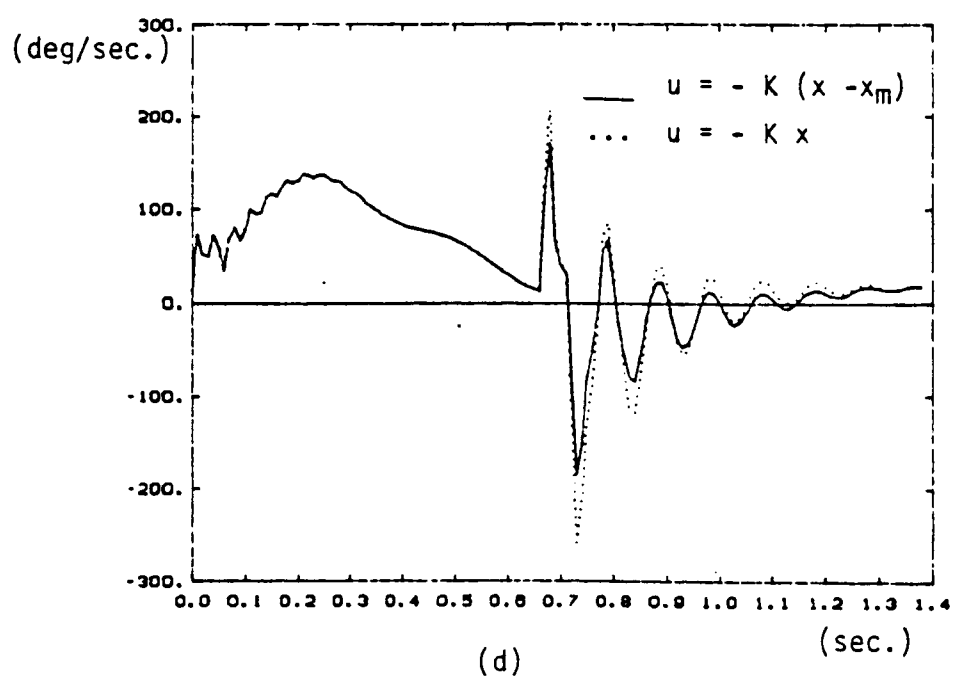


Fig.5.16.c-d Combined control response - with realistic joint inertias. c) joint 1 speed response. d) joint 2 speed response.



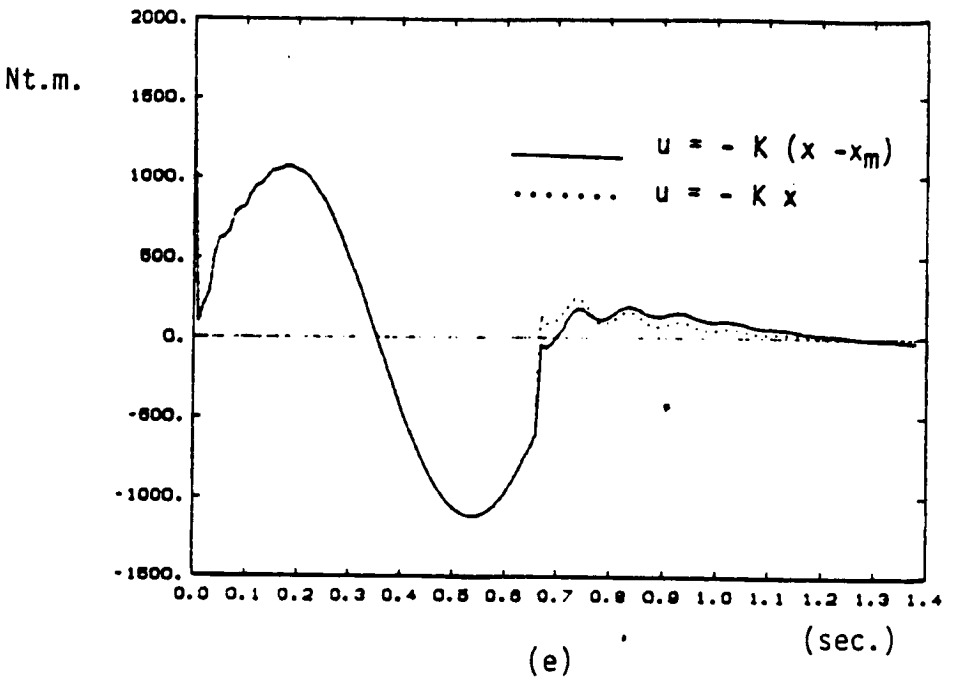
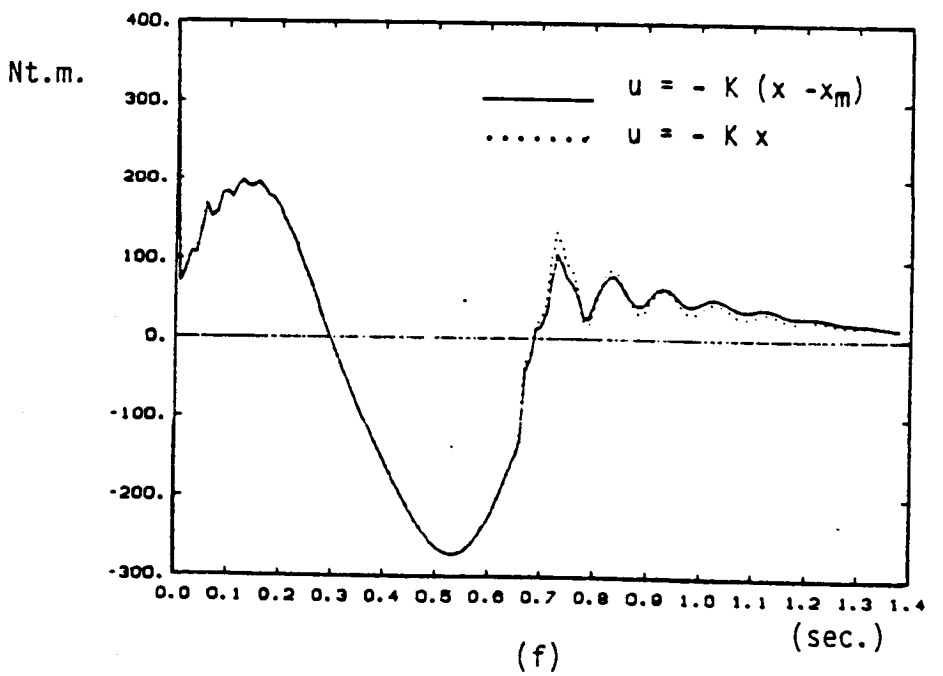


Fig.5.16.e-f Combined control response - with realistic joint inertias. e) joint 1 torque history. f) joint 2 torque history.



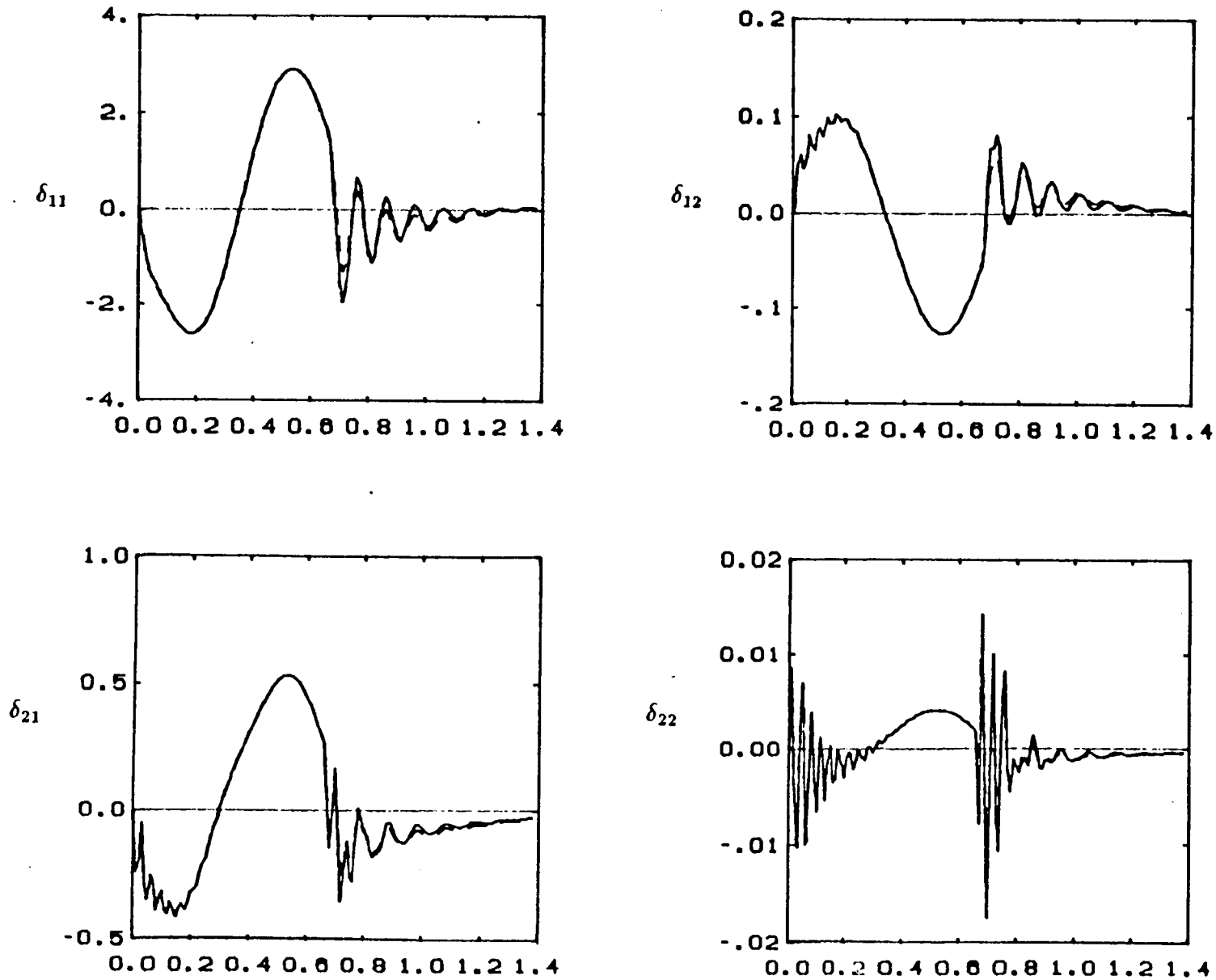


Fig.5.16.g Combined control response - with realistic joint inertial values. g) flexible mode shape magnitude responses along the motion described by figure 5.16.a

5.4. Summary of Results

The combined control has shown enough evidence of being a good strategy for high speed, high precision, robust motion control of flexible manipulators. The speed of motion just before switching to fine motion control is critical in the success of linear fine motion controllers. The model following regulator gives better results than the standard regulator.

The combined control strategy improves the performance of flexible arms over that other methods studied. It should be noted that fine motion control is not limited to linear controllers, and other methods should be studied to increase the robustness with respect to payload uncertainty as well as the state of the manipulator before the switching time.

CHAPTER VI

Conclusions and Recommendations for Future Work

6.1. Conclusions

Understanding the fundamental characteristic of the dynamics of a given system is essential in developing a control system which will *make the system do what we want it to do.* Therefore, it is important to develop methods which will provide dynamic models with desired accuracy and convenience. Along this line, a general symbolic modeling algorithm is developed for flexible manipulators. The algorithm is based on the Lagrange-assumed modes method. It can handle any degree of freedom manipulator with serial kinematic structure. Using a commercially available symbolic manipulation software package (SMP), the algorithm is successfully applied to a number of case studies, including the models used in this work. The contribution of this modeling algorithm can be summarized as follows:

1. It is convenient, fast, accurate and free of possible human error during long and tedious algebraic manipulations.

2. It handles a manipulator with any number of degrees of freedom.

3. It results in scalar explicit symbolic equations of motion. Thus, further insight of the dynamics of the system can be gained. The equations are very suitable for real-time parallel computation of the control.

4. Mode shapes of the flexible structures can be kept as parameters at the modeling level and varied at the analysis or control level.

The closed loop dynamics of flexible manipulators under joint variable feedback is examined using linear analysis tools. The results from a finite dimensional assumed modes model are compared with the results from infinite dimensional transfer matrix models. Both models agree in predicting the limitations of joint variable feedback control due to arm flexibility. However, the comparative analysis also raised new questions besides improving the current understanding. The results indicate that finite dimensional models lose accuracy in predicting the dynamic behavior of higher modes under even moderate feedback gain conditions. It is believed that any finite dimensional model results have a smaller range of accuracy than the model order anticipates. The question of how large the system order should be in order to guarantee a prescribed accuracy for a given range has not been answered. Furthermore, under feedback control, some closed loop eigenvalues of finite dimensional model go toward $-\infty$ very quickly, resulting in a *numerically stiff* system of differential equations. Efficient numerical methods to study these stiff systems should be developed. In fact, there are already existing numerical methods specifically developed for stiff systems, but they are not efficient enough for the problems faced in this work. Furthermore, the emphasis of these methods is to accurately solve the problem for all frequencies, such as in plasma dynamics. But in flexible structures, it is more desirable to be able to eliminate the very high frequency modes. These modes are very high frequency, well damped, and of little practical interest. A nonlinear model reduction approach is more desirable in robotics and large flexible structure studies.

The motion control problem of flexible manipulators consists of the tracking control in joint space and vibration control of the structural elements. In the study of control algorithm development for flexible manipulators which will provide robost, high speed, and high precision motions, the following path is taken:

1. The performance and the limitations of rigid model based motion control algorithms are determined when applied to flexible arms.

2. In an attempt to improve the performance, an AMFC design procedure is developed. Performance improvements and the limitations are determined.

3. Finally, a combined control strategy is presented as a natural way of obtaining flexible manipulator control. The combination of AMFC gross motion and LQR fine motion control is studied.

In summary, non-adaptive schemes must be used in slow motions relative to the closed loop system bandwidth if large payload variations are anticipated. The only way these schemes can provide robustness with respect to parameter variations is to have high (w_{ni}/w_{mi}) ratio. The upper limit for w_{ni} is determined by the arm flexibility,$(w_{ni} \leq w_{cc1})$. The robustness requirements further forces the non-adaptive algorithms to take conservative measures, resulting in low performance, low speed lightweight arms. The main objective of the lightweight arms in this work was to achieve higher speeds of operation. Thus, more advanced control schemes must be developed.

An AMFC based on hyperstability is developed where the generalized inertia matrix plays a central role in the design. It relaxes some of the restrictive assump-

tions made by previous AMFC design procedures, such that the use of AMFC techniques in high speed manipulation becomes possible. It simplifies the general design and parameter selection problem very significantly. The number of arbitrary design parameters that must be determined by the designer is only two no matter what the degree of freedom of the manipulator is. Previous methods had to deal with finding the appropriate values for two (mxm) matrices for an equivalent design. Due to the central role of the generalized inertia matrix in the method, decoupled joint response is preserved for almost all cases studied. The method can be readily applied to rigid manipulators as well as flexible manipulators. As far as the performance of this control on flexible manipulators is concerned, the performance is dramatically improved over the CTM, and DJC methods. Yet, the AMFC also has performance limitations due to arm flexibility. The AMFC does not have to take a conservative measure to provide robustness with respect to parameter variations in advance for it can modify its feedback gains as needed. However, if the speed of motion gets high with respect to arm flexibility, the AMFC results in very stiff joint control, and this causes persistent flexible vibrations at the end of the motion.

If one desires to keep the advantages of the AMFC and yet be able to deal with end point vibrations, a combined gross and fine motion strategy may be considered. In the combined control strategy, the same AMFC is used for gross motion control, and before the motion ends where AMFC can not cope with flexible vibrations, control algorithm is switched to one which deals with joint positioning and vibration control (fine motion control).

Studies conducted on a combined control, composed of AMFC for gross motion, and LQR for fine motion, indicate that limited success in improving the perfor-

mance can be achieved. The performance of fine motion controller (LQR) strongly depends on the state of the manipulator at the switching time. If the manipulator speed is low enough at switching time, the results are very good. However, if the speed at switching time is high for any reason, then the LQR performs poorly. Changing the switching time does not help to improve the performance. Further studies should be conducted on fine motion control algorithms which are robust with respect to manipulator state uncertainty at the switching time.

6.2. Recommendations for Future Work

1. The symbolic modeling algorithm developed in this work should be extended to handle more complicated multi-body dynamic systems, including closed loop kinematic structures as well as serial structures. The flexible body dynamics should be extended to include longitudinal vibrations in addition to the bending and torsional vibrations considered currently. The currently available finite element packages or the DSAP [A30] package may be utilized as a tool which will provide accurate mode shapes for the resultant symbolic equations of motion obtained. Such work would provide a general purpose tool which is effective and accurate in dynamic modeling and analysis of flexible multi-body systems. User interface to such a tool should be through state of the art computer graphics.

2. Closed loop dynamics of infinite dimensional systems, such as flexible manipulators, and large space structures, under partial state feedback need to be further studied. It is clear from the discussion of Chapter III, that there are unanswered questions about the closed loop dynamics. The key question in this area, in the author's opinion, is the following: how many modes should be included in

the model (what should be the order of the system) in order to guarantee a certain degree of accuracy for a given range of frequencies? It is believed that use of infinite dimensional linear models, like DSAP, and finite dimensional assumed modes models together will be an effective approach to answer this question.

3. On the robust, high speed, high precision motion control aspects, the following directions should be taken:

a) Fine motion control algorithms that are robust with respect to manipulator state uncertainty at the switching time, as well as with respect to parameter variations, should be further studied. Experiments should be conducted with example cases to provide feedback to the design and analysis of the of the control system design from the real world experience, and demonstrate pilot cases of working implementations. Improving the fine motion controller performance has more priority than further improving the AMFC performance in gross motion.

b) Dynamic parameter identification of flexible manipulators in real-time would give a new dimension to the design of adaptive control algorithms. This study would involve the following steps: i) development of analytical algorithms for parameter identification that can be implemented in real-time, ii) modifying the existing adaptive control algorithms to utilize the available information, coming from the identification scheme, iii) selecting the appropriate combination of sensors to provide the necessary information for steps (i) and (ii), (iv) putting all of the decision and calculation algorithms, sensors, computers, and the manipulator together to work.

APPENDIX A

Flexible Manipulator Arm Parameters:

Material dependent properties (Alimunum)

$$\rho_i = 2768.kg/m^3$$

$$E_i = 7.0x10^{10} Nt/m^2$$

Geometric properties:

$$l_i = 2.0m$$

$$r_1 = 18.052mm$$

$$r_2 = 9.792mm$$

$$I = 7.6190x10^{-8} m^4$$

$$A = 7.224x10^{-4} m^2$$

$$\rho A_i = 2.0kg/m$$

$$m_i = 4.0kg.$$

Lowest frequency when both both joints are clamped and link 2 is at extended position:

Chapter 3 analysis parameters:

$$EI = 533.333 Nt.m^2$$

$$w_{cc1} = 3.59 rad./sec$$

ORIGINAL PAGE IS
OF POOR QUALITY

Chapter 4 and 5 simulation parameters:

$$EI = 5333.33 Nt.m^2$$

$$w_{cc1} = 11.34 rad/sec$$

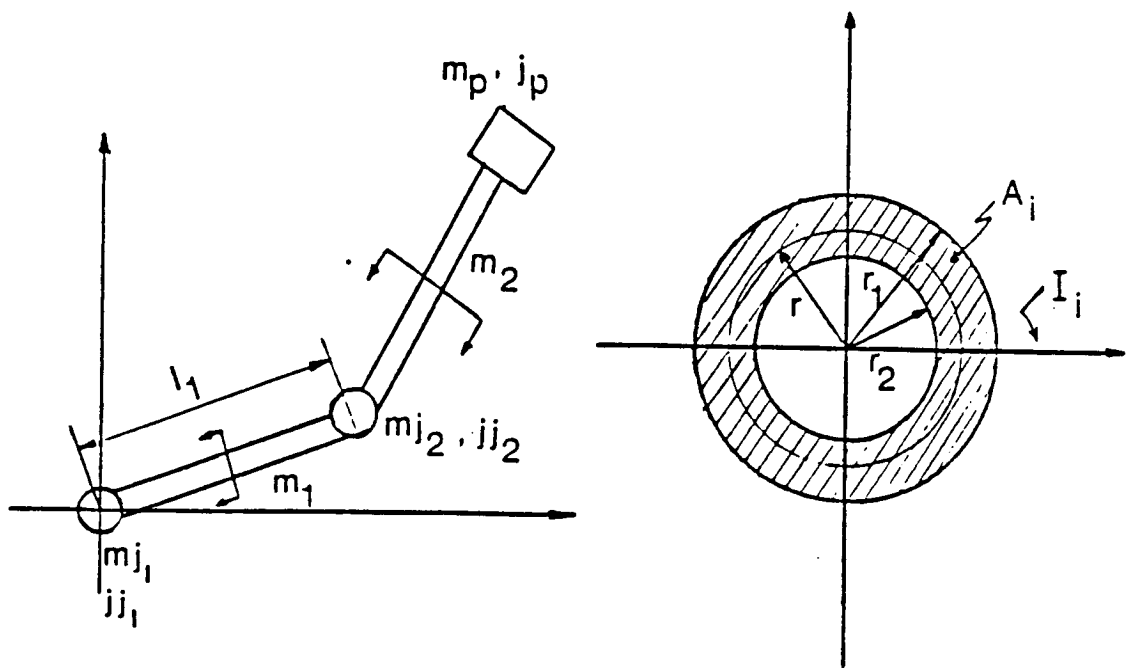


Fig.A.1 The flexible arm paramters and geometric dimensions

APPENDIX B

On the Solution of $Ax = b$

The solution of linear algebraic equations has many applications in control theory, such as controllability, observability, and model matching. Erzberger's linear perfect model following conditions are in fact the statements of the existence of a unique solution for an equation of the form

$$Ax = b \ .$$

Let $A \in R^{mxn}, x \in R^n$, and $b \in R^m$.

1. If Rank(A) = Rank$(A, b) \Longrightarrow$ $b \in$ Range(A), then there exists at least one exact solution.

i. $m = n \Rightarrow$ there is one unique exact solution , $x = A^{-1}b$.

ii. $m < n$ which means Range(A, b)<Dim(x), thus null space of A, $Null(A) \neq 0 \Rightarrow$ there exist infinitely many solutions, all of which satisfies the equation exactly, so they are all exact solutions. Among them the solution with minimum norm is obtained by ;

$$Ax = b, \quad \text{Let} \quad x = A^T z \Rightarrow x = A^T(A\ A^T)^{-1}b$$

Note that if A is of full rank (row rank in this case) $(AA^T)^{-1}$ exists. If A is not full row rank, yet $Rank(A) = Rank(A, b)$, the above statement is still true, but

the inversion must be done with singular value decomposition.

2. Rank(A) < Rank(A, b) $\Longrightarrow$ there is no exact solution. But approximate solutions can be found. The x ,for which

$$\{x : \| e \| = \| Ax - b \| \ \ is \ \ minimized\},$$

is given by

$$x = (A^T A)^{-1} A^T b$$

Again if A is not full column rank direct inversion in this equation will not be possible, yet an equivalent inversion using singular value decomposition (SVD) can be obtained , which will result in an approximate solution with minimum norm.

If a generalized inverse of A is defined as

$$A^\dagger = A^T (AA^T)^{-1} = (A^T A)^{-1} A^T \ ;$$

which either exists or found through SVD,

$$x = A^\dagger b$$

is the solution with minimum norm error. If $b \in Range(A)$ solution reduces to the exact solution with minimum norm.

APPENDIX C

Stability

The stability of a control system is always the fundamental requirement. In what follows, the concept of *stability* is clarified as its used in this work. First, stability in the sense of Lyapunov, then Hyperstability is discussed.

C.1. Lyapunov Stability

Consider a free, unforced dynamic system,

$$\dot{x} = f(x,t) \qquad x \in R^n \ , -\infty < t < \infty \tag{$C-1$}$$

Let $x = \phi(t; x_o, t_o)$ be such that,

$$x_o = \phi(t_o; x_o, t_o) \tag{$C-2$}$$

$$\dot{\phi}(t; x_o, t_o) = f\left[\phi(t; x_o, t_o), t\right] \tag{$C-3$}$$

Also let x_e be such that $f(x_e, t) = 0$ for all time, t. Then x_e is said to be an equilibrium state, x_o initial state, t_o initial time. Different types of stability of a dynamic system about such an equilibrium point are defined as follows.

Definitions: An equilibrium state x_e of $(C-1)$ is said to be:

1. Stable if $\forall\, \epsilon > 0$; $\exists\, \delta(\epsilon, t_o) > 0$ such that

$$\| x_o - x_e \| < \delta(\epsilon, t) \implies \| \phi(t; x_o, t_o) - x_e) \| \leq \epsilon \quad \forall\, t \geq t_o$$

2. Asymptotically stable if

$$\| \phi(t; x_o, t_o) - x_e \| \longrightarrow 0. \quad as \quad t \longrightarrow \infty$$

for x_o being *sufficiently close* to x_e.

3. Globally asymptotically stable, if the *sufficiently close* condition from (2.) is removed. That is, $\forall x_o$ in the space of (C-1),

$$\| \phi(t; x_o, t_o) - x_e \| \longrightarrow \quad 0 \ \ as \ t \longrightarrow \infty, \ \forall x_o \in R^n$$

4. Stable in large, If $x_o \in R_1$ where R_1 is a region of R^n and $\| \phi(t; x_o, t_o) - x_e \| \longrightarrow 0$ (a restricted form of global asymptotic stability).

5. Uniformly stable if the the conditions stated above are independent of t_o.

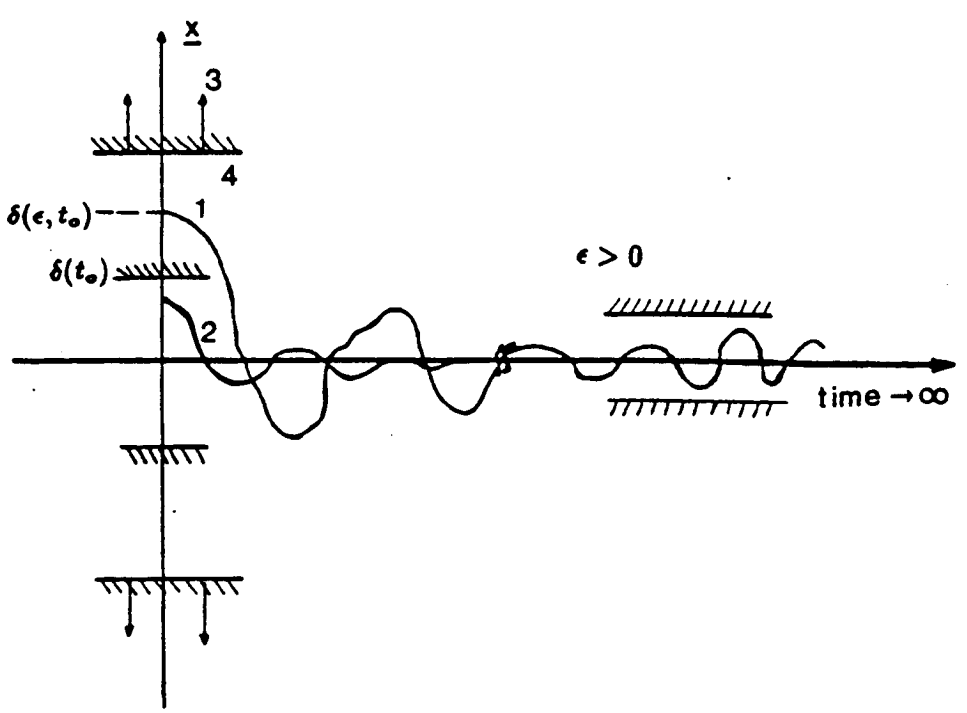


Fig.C.1 The different definitions of stability.

In general, a control system should be globally (or at least *at large*) asymptotically stable. The question which follows this statemet is the following: under what conditions an equilibrium state of a dynamic system is globally asymptotically stable (GAS). Lyapunov's second method gives sufficient conditions (but not neccessary) that must be satisfied by the dynamic system and its equilibrium point.

Lyapunov Theorem:

Consider $\dot{x} = f(x,t)$ and $f(0,t) = 0 \Longrightarrow x_e = 0$ is equilibrium state. If $\exists V(x,t)$ with continuous partial derivatives with respect to x and t, and has the following properties:

1. $V(0,t) = 0 \quad ;\text{for} -\infty < t < \infty$
2. $V(x,t) > 0 \quad \forall x \neq 0. \quad ; \; x \in R^n \quad , -\infty < t < \infty$
3. $V(x,t) \longrightarrow \infty$ as $\| x \| \longrightarrow \infty \quad , -\infty < t < \infty$
4. $\dot{V}(x,t) = dV(x,t)/dt < 0 \; , \quad \forall x \neq 0 \; , \; x \in R^n \; , \; -\infty < t < \infty$

then $x_e = 0$ is globally asymptotically stable (GAS) !

Remarks:

1. If $V(x,t) = V(x)$, not function of t, then x_e is *uniformly GAS.*

2. For linear time invariant (LTI) dynamic systems,

$$\dot{x} = Ax \qquad (x_e = 0)$$

$$V(x) = x^T P x$$

where, P is such that

$$A^T P + PA = -Q$$

for $Q = Q^T \geq \ (>) \ 0$, and $P = P^T > 0$, then $x_e = 0$ is uniformly (asymptotically) globally stable.

3. From adaptive control point of view, an adaptation algorithm is choosen based on Lyapunov so that a trail $V(x,t)$ is a Lyapunov function † for the dynamic system and the equilibrium state. The problem in design is that the correlation between the choice of Lyapunov function and the resultant control system performance (transient, steady state etc.) is not well understood yet.

C.2. Hyperstability and Positivity Concepts

Hyperstability is a different way of looking at the stability of a dynamic system. Lyapunov stability is concerned with the stability of a given dynamic sytem about an equilibrium state, or with the design of a specific feedback control so that the system is stable about the equilibrium state. Whereas, hyperstability is concerned with finding a condition (or design) so that system is stable for a class of controls, not a specific control.

Consider the figure (C.2), where FFB is linear time invariant (LTI) block, and feedback block (FBB) can be nonlinear, time varying. *The question of absolute stability:* what conditions the FFB must satisfy such that for all FBB $\in \{A\}$, the closed loop system is globally asymptotically stable, where

$$\{A\} = \{v^T w \geq 0 \quad ; \ i = 1, \ldots, m\}.$$

Popov further generalizes this question as follows: *Hyperstability:* what conditions should FFB must satisfy that for all FBB $\in \{P\}$ the closed loop system is GAS,

† Any function satisfying Lyapunov theorem conditions is said to be a Lyapunov function.

where ‡

$$\{P\} = \{ \int_{t_o}^{t_1} v^T w dt \geq -\gamma_o^2 \quad , \forall\, t_1 \geq t_o\}$$

Definition: The asymptotic hyperstability of a closed loop system means that the system is globally asymptotically stable about an equilibrium state for any choice of FBB form $\{P\}$ class.

The answer to the hyperstability questions is as follows:

Theorem: For the CLS to be (asymptotically) hyperstable, the neccessary and sufficient condition is LTI FFB has a transfer function $H(s)$ which is (strictly) positve real.

Lemma (Kalman - Yakubovich - Popov): Consider the linear time invariant FFB description,

$$\left\{ \begin{array}{l} \dot{x} = Ax + Bu \\ \\ v = Cx \end{array} \right\} \Longleftrightarrow \left\{ H(s) = C(sI - A)^{-1} B \right\}.$$

If $H(s)$ is (strictly) positive real, then $\exists P = P^T > 0$ and $Q = Q^T \geq (>) 0$ such that

$$A^T P + PA = -Q$$

$$B^T P = C$$

Thus, for a given $\dot{x} = Ax + Bu$, the design problem is to pick an output filter C according to KYP lemma, so that the resultant transfer function is (strictly) positive real. After that step, any controller design which results in the FBB $\in \{P\}$, will guarantee the global (asymptotic) stability of the closed loop system.

‡ This inequality is called Popov Integral Inequality (PII).

The last question remains in the design of hyperstable closed loop systems is what form FBB in general can be so that they belong to Popov class, $\{P\}$. Consider the standard problem of hyperstability, Fig.C.2, where v and w are of same dimension. Popov class is defined as the class of blocks stisfying the following:

$$PII: \quad \int_{t_o}^{t_1} v^T w dt \geq -\gamma^2 \ , \quad \forall t_1 \geq t_o$$

where $w(t)$ is of the following general form,

$$w(t) = \left[\int_{t_o}^{t_1} \phi_1(v,t,\tau)dt + \phi_2(v,t) + A_o \right] x(t)$$

$w(t), v(t), x(t)$ are of appropriate dimensions and piecewise continuous, bounded functions.

What is the most general form of $\phi_1(v,t,\tau), \phi_2(v,t)$ so that PII is satisfied (A_o is constant).

Lemma:(Sufficient, but not necessary)

The following choices satisfy the PII.

$$\phi_1(v,t,\tau) = F_1(t-\tau)v(\tau)\left[G_1 x(\tau)\right]^T$$
$$\phi_2(v,t) = F_2(t)v(t)\left[G_2(t)x(t)\right]^T$$

where, $F_1(v,t,\tau)$ is a positive definite square matrix kernel whose Laplace transform is any positive real transfer matrix with a pole at $s = 0$. G_1 is any positive definite, and $F_2(t), G_2(t)$ are positive semi-definite matrices. The design of an adaptation algorithm based on hyperstability results in a class of possible choices, all of which guarentees closed loop stability.

At this point, it is appropriate to clarify the concept of positive realness. In the time domain positive realness of a block means that for zero initial conditions, the integral inner product of input and output of the block is always greater than or equal to zero for every non-negative time interval.

$$\int_{t_o}^{t_1} u^T v dt \geq 0 \quad , \; t_1 \geq t_o$$

In the frequency domain, a positive real transfer matrix function implies that,

1. no poles in the open right hand plane of the s-plane,
2. if any poles exists on the imaginary axis, they are all distinct and the associated residue matrix is semi-definite and hermitian ($H(s) = H^T(s^*)$),
3. $H(jw) + H^T(-jw) \geq 0$ (hermitian) for all w on the imaginary axis †.

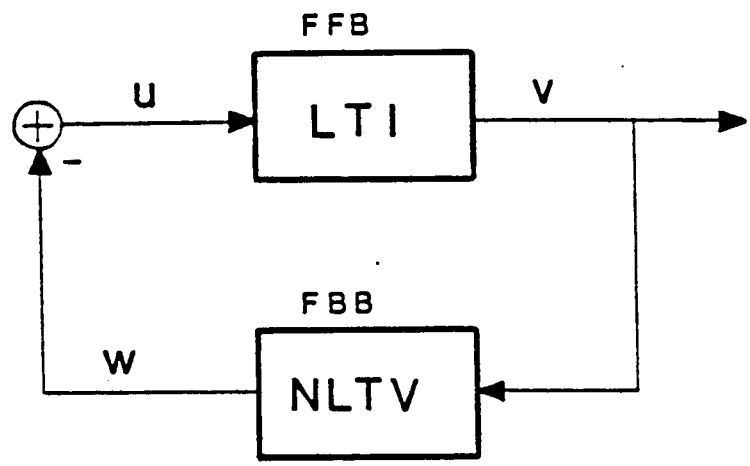


Fig.C.2 The standard Hyperstability problem.

† Until that point in this appendix the w corresponds to the output of the nonlinear feedback block of the hyperstability problem (Fig. C.1), at this point it is used to represent the frequency.

APPENDIX D

Linear Analysis Results

D.1. Linear Dynamic Model of the Flexible Manipulator

Case 1: $(A,B)_1$, $mp = 0.0kg$,

Nominal states about which nonlinear model is linearized:

$\left[\theta_1, \delta_{11}, \delta_{12}, \theta_2, \delta_{21}, \delta_{22}, , \dot{\theta}_1, \dot{\delta}_{11}, \dot{\delta}_{12}, \dot{\theta}_2, \dot{\delta}_{21}, \dot{\delta}_{22}\right] =$

[1.832596	0.0E+00	0.0E+00	-0.5235988	0.0E+00	0.0E+00
0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00]

Nominal input torque associated with the nominal state: $[u_{n1}, u_{n2}]_{nominal} =$

[-20.312114 10.15606]

Open loop (A,B) matrices of $\dot{x} = Ax + Bu$,

A Matrix:

$$A = \begin{bmatrix} 0 & I \\ A_{21} & A_{22} \end{bmatrix}$$

A_{21} :

0.0000000E+00	17473.75	98975.87	0.0000000E+00	-1708.117	-6447.686
0.0000000E+00	-404257.8	-2310347.	0.0000000E+00	34090.71	128674.4
0.0000000E+00	-58825.56	-381927.3	0.0000000E+00	10975.57	41438.56

0.0000000E+00	-5451.436	-53251.81	0.0000000E+00	25873.62	45363.5
0.0000000E+00	34090.71	431060.4	0.0000000E+00	-530607.8	-30803
0.0000000E+00	3276.280	41438.53	0.0000000E+00	-78429.96	-550128.1

A_{22} :

-0.1042042	7.6982E-02	6.9475E-02	-3.9321E-02	0.5261E-03	-4.5332E-03
2.079921	-1.781017	-1.622109	0.7848701	0.1502073	9.0468E-02
0.6693771	-0.2591318	-0.2678026	0.2525833	4.83595E-02	2.91345E-02
0.5104299	-2.3983E-02	-3.6998E-02	0.1550875	0.1140019	0.1022017
-7.632691	0.1496994	0.2969413	-2.144678	-2.337914	-2.165688
-0.7339036	1.4386795E-02	2.8545320E-02	-0.2062152	-0.3455708	-0.3867823

B Matrix:

0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
36.77985	-11.87735
-848.0787	264.5824
-122.3117	65.80717
-11.87735	66.89980
82.90250	-1255.762
7.967876	-179.6363

Case 2: $(A,B)_2$, $mp = 0.0kg$,

Nominal states about which nonlinear model is linearized:

$$\left[\theta_1, \delta_{11}, \delta_{12}, \theta_2, \delta_{21}, \delta_{22}, \dot{\theta}_1, \dot{\delta}_{11}, \dot{\delta}_{12}, \dot{\theta}_2, \dot{\delta}_{21}, \dot{\delta}_{22}\right] =$$

[1.832596	-5.6964E-02	9.769448E-03	-0.5235988	-6.8726E-02	-9.6981E-04
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00]

Nominal input torque associated with the nominal state: $[u_{n1}, u_{n2}]_{nominal} =$

[-19.64318 10.49558]

Open loop (A,B) matrices of $\dot{x} = Ax + Bu$,

A_{21} :

0.000000E+00	17741.11	101449.1	0.0000000E+00	-2254.782	-8296.033
0.0000000E+00	-409920.3	-2365059.	0.0000000E+00	47178.12	173912.4
0.0000000E+00	-60218.50	-392476.9	0.0000000E+00	12311.10	44965.71
0.0000000E+00	-6341.523	-57325.80	0.0000000E+00	25239.96	141912.7
0.0000000E+00	47177.84	483511.0	0.0000000E+00	-518024.0	-3017565.
0.0000000E+00	4428.112	44965.74	0.0000000E+00	-76832.90	-542554.8

A_{22} :

-7.3911E-02	0.101931	5.11517E-02	-3.4491E-02	-1.3046E-02	-6.1107E-03
1.382051	-2.355229	-1.194921	0.6726303	0.2729599	0.1281001
0.567287	-0.346012	-0.202339	0.236764	7.1229E-02	3.3121E-02
0.5118743	-3.6428E-02	-3.0617E-02	0.15955	0.146032	0.10453

-7.82996	0.27087	0.25966	-2.267262	-2.997148	-2.222672
-0.7698207	2.5424E-02	2.4131E-02	-0.2254684	-0.444534	-0.39960

B Matrix:

0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
37.38747	-13.70555
-861.0564	307.7827
-125.3681	70.84167
-13.70555	65.53455
109.4341	-1225.002
10.25202	-175.3719

Case 3: $(A, B)_3$, $mp = 2.0kg$,

Nominal states about which nonlinear model is linearized:

$$\left[\theta_1, \delta_{11}, \delta_{12}, \theta_2, \delta_{21}, \dot{\delta}22, \dot{\theta}_1, \dot{\delta}_{11}, \dot{\delta}_{12}, \dot{\theta}_2, \dot{\delta}_{21}, \dot{\delta}_{22}\right] =$$

[1.832596	-0.1548394	1.9869E-02	-0.52359	-0.156499	1.2656E-03
0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00]

Nominal input torque associated with the nominal state: $[u_{n1}, u_{n2}]_{nominal} =$

[-20.31215 20.31211]

Open loop (A, B) matrices of $\dot{x} = Ax + Bu$,

A_{21} :

0.0000000E+00	17830.72	102753.6	0.0000000E+00	-2727.851	-8940.832
0.0000000E+00	-411487.6	-2396300.	0.0000000E+00	60447.67	200036.9
0.0000000E+00	-61013.98	-395644.7	0.0000000E+00	11523.27	35855.89
0.0000000E+00	-7212.480	-56072.48	0.0000000E+00	20009.76	102456.1
0.0000000E+00	60447.67	452571.8	0.0000000E+00	-424433.7	-2290482.
0.0000000E+00	5093.288	35855.86	0.0000000E+00	-58319.66	-362312.8

A_{22} :

-4.4828E-02	-1.4591E-03	-1.7601E-02	-3.5321E-02	0.00E+00	-6.5856E-03
0.632382	3.3570E-02	0.4049409	0.63152	0.00E+00	0.1473439
0.54864	4.9257E-03	5.9416E-02	0.2991546	0.00E+00	2.6411E-02
0.6145965	5.9152E-04	7.1353E-03	0.2482	0.00E+00	7.5467E-02
-9.618283	-5.1357E-03	-6.19502E-02	-3.726573	0.00E+00	-1.687119
-0.906185	-4.28596E-04	-5.169976E-03	-0.3638627	0.00E+00	-0.2668718

B Matrix:

0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00

37.61354	-15.24901
-865.4055	350.0543
-126.9802	69.29288
-15.24901	51.37159
132.3949	-971.1640
11.04884	-126.6125

D.2. LQR Feedback Gains and the Resultant Closed Loop Eigenvalues

$$\underbrace{(Q, R, \underline{\alpha})}_{\textit{Given optimality criteria}} \quad , \underbrace{((A,B)_i, A_m)}_{\textit{Linear Model}} \longrightarrow \underbrace{(K_1, K_2)}_{\textit{Feedback gains}} \quad \underbrace{\{\lambda_c\}}_{\textit{Closed loop eigenvalues}}$$

The following optimal linear feedback control gains are used in the fine motion control phase ofsimulations corresponding to figures 5.3 through 5.7.

$R = I$ in all cases.

$$Q = Diag(10^2, 10^0, 10^2, 10^2, 10^0, 10^2), (10^2, 10^0, 10^2, 10^2, 10^0, 10^2)$$

$$(A, B) \rightarrow (A, B)_1$$

$$\underline{\alpha} = 0,\ 2,\ 5$$

$$u = -K_1 x - K_2 x_m$$

$$x = \left[\theta_1, \delta_{11}, \delta_{12}, \theta_2, \delta_{21}, \delta_{22}, \dot{\theta}_1, \dot{\delta}_{11}, \dot{\delta}_{12}, \dot{\theta}_2, \dot{\delta}_{21}, \dot{\delta}_{22}\right] - \left[\theta_1, \delta_{11}, \delta_{12}, \theta_2, \delta_{21}, \delta_{22}, \dot{\theta}_1, \dot{\delta}_{11}, \dot{\delta}_{12}, \dot{\theta}_2, \dot{\delta}_{21}, \dot{\delta}_{22}\right]_{nominal}$$

K_1 and K_2 are both 2x12 feedback gain matrices. In what follows, the first two rows correspond to the first row , and the last two rows correspond to the second row of the K_1 and K_2 as indicated.

$\underline{\underline{\alpha}} = 0.$

K_1:

0.99844E+01	-0.11203E+02	-0.58244E+03	0.55913E+00	-0.95506E+02	-0.12884E+03
0.29889E+02	-0.14505E+00	-0.32870E+01	0.64080E+01	0.21608E+00	-0.96800E+00
-0.55913E+00	0.19023E+02	-0.64017E+03	0.99844E+01	-0.11596E+03	-0.14800E+04
0.41799E+01	-0.50843E-01	0.20078E+01	0.12915E+02	-0.10957E+01	-0.13219E+01

K_2:

0.15532E+01	0.74038E-01	-0.20939E+00	-0.12258E+01	0.59041E-02	0.25306E-01
-0.42822E+00	0.24442E-01	-0.76887E-01	0.72653E+00	0.25634E-02	0.78150E-02
-0.17405E+01	0.28552E-01	-0.64086E+00	0.59376E+01	0.48162E-01	-0.30685E-01
0.86535E+00	0.89737E-02	-0.18333E+00	-0.30625E+01	0.13646E-01	-0.83492E-02

λ_c:

-0.35413E+00	j0.31667E+00
-0.35413E+00	-j0.31667E+00

-0.10116E+01	j0.00000E+00
-0.67083E+01	-j0.21767E-16
-0.32848E+02	j0.14659E+03
-0.32848E+02	-j0.14659E+03
-0.66526E+02	j0.20038E+03
-0.66526E+02	-j0.20038E+03
-0.50691E+03	j0.22811E-14
-0.68696E+03	j0.57072E+03
-0.68696E+03	-j0.57072E+03
-0.19880E+04	j0.00000E+00

$\underline{\underline{\alpha}} = 2.$

K_1:

0.67043E+03	0.74818E+01	-0.68444E+03	0.19657E+03	-0.96831E+02	-0.11809E+03
0.33681E+03	0.16604E+02	-0.33736E+02	0.10084E+03	0.40991E+01	0.29192E+00
0.14647E+03	0.26422E+02	-0.66724E+03	0.10402E+03	-0.13272E+03	-0.15131E+04
0.78308E+02	0.42082E+01	-0.74657E+01	0.45008E+02	0.23537E+00	-0.85650E+00

K_2:

-0.26736E+01	0.15073E-01	-0.50139E-01	0.46618E+01	0.17756E-02	0.46477E-02
-0.68231E+00	0.29634E-02	-0.98827E-02	0.11543E+01	0.35114E-03	0.91165E-03
0.59339E+01	0.55430E-02	-0.11103E+00	-0.19895E+02	0.82313E-02	-0.50201E-02
0.14810E+01	0.10900E-02	-0.21735E-01	-0.49749E+01	0.16105E-02	-0.97983E-03

λ_c:

-0.40078E+01	j0.58756E-01

-0.40078E+01	-j0.58756E-01
-0.42413E+01	j0.00000E+00
-0.90133E+01	-j0.96798E-18
-0.34920E+02	j0.14658E+03
-0.34920E+02	-j0.14658E+03
-0.68554E+02	j0.20039E+03
-0.68554E+02	-j0.20039E+03
-0.50891E+03	j0.21246E-13
-0.68896E+03	j0.57072E+03
-0.68896E+03	-j0.57072E+03
-0.19900E+04	j0.00000E+00

$\underline{\alpha} = 5.$

K_1:

0.42446E+04	0.17245E+03	-0.10887E+04	0.12871E+04	-0.66526E+02	-0.89261E+02
0.86872E+03	0.45648E+02	-0.86641E+02	0.26492E+03	0.10853E+02	0.24424E+01
0.93372E+03	0.69391E+02	-0.77322E+03	0.50370E+03	-0.15211E+03	-0.15551E+04
0.20230E+03	0.11395E+02	-0.23912E+02	0.10123E+03	0.25878E+01	-0.13918E+00

K_2 :

-0.12754E+02	0.21291E+00	-0.72657E+00	0.17249E+02	0.26729E-01	0.64561E-01
-0.15449E+01	0.31225E-01	-0.10722E+00	0.19440E+01	0.39764E-02	0.94163E-02
0.26557E+02	0.73712E-01	-0.16154E+01	-0.91314E+02	0.12104E+00	-0.76983E-01
0.30669E+01	0.10781E-01	-0.23410E+00	-0.10653E+02	0.17525E-01	-0.11097E-01

λ_c :

-0.10004E+02	j0.28840E-01
-0.10004E+02	-j0.28840E-01
-0.10101E+02	j0.00000E+00
-0.13435E+02	-j0.12620E-17
-0.38299E+02	-j0.14657E+03
-0.38299E+02	j0.14657E+03
-0.71705E+02	j0.20042E+03
-0.71705E+02	-j0.20042E+03
-0.51196E+03	j0.18565E-14
-0.69198E+03	-j0.57073E+03
-0.69198E+03	j0.57073E+03
-0.19930E+04	j0.00000E+00

$(A, B) \rightarrow (A, B)_2$, $m_p = 0.0kg.$

$\underline{\alpha} = 0.0$

K_1:

0.99853E+01	-0.11267E+02	-0.58202E+03	0.54177E+00	-0.94808E+02	-0.12837E+03
0.29930E+02	-0.14328E+00	-0.32549E+01	0.64109E+01	0.23002E+00	-0.89693E+00
-0.54177E+00	0.18342E+02	-0.65766E+03	0.99853E+01	-0.11832E+03	-0.15026E+04
0.42534E+01	-0.50947E-01	0.20612E+01	0.12930E+02	-0.10855E+01	-0.13056E+01

K_2:

0.15515E+01	0.73957E-01	-0.20890E+00	-0.12376E+01	0.58376E-02	0.25411E-01
-0.43085E+00	0.24423E-01	-0.76819E-01	0.73500E+00	0.25486E-02	0.78415E-02

-0.17388E+01	0.28973E-01	-0.64196E+00	0.59217E+01	0.48234E-01	-0.31028E-01
0.86811E+00	0.91001E-02	-0.18367E+00	-0.30695E+01	0.13667E-01	-0.84410E-02

λ_c:

-0.35334E+00	j0.31611E+00
-0.35334E+00	-j0.31611E+00
-0.10115E+01	j0.00000E+00
-0.67353E+01	-j0.19075E-16
-0.32499E+02	-j0.14704E+03
-0.32499E+02	j0.14704E+03
-0.68756E+02	j0.20140E+03
-0.68756E+02	-j0.20140E+03
-0.50965E+03	-j0.44274E-14
-0.69911E+03	-j0.55610E+03
-0.69911E+03	j0.55610E+03
-0.19766E+04	j0.14211E-13

$\underline{\alpha} = 2.0$

K_1:

0.67043E+03	0.74818E+01	-0.68444E+03	0.19657E+03	-0.96831E+02	-0.11809E+03
0.33681E+03	0.16604E+02	-0.33853E+02	0.10132E+03	0.41306E+01	0.36491E+00
0.14860E+03	0.25790E+02	-0.68467E+03	0.10467E+03	-0.13501E+03	-0.15351E+04
0.79394E+02	0.42683E+01	-0.75171E+01	0.45339E+02	0.26016E+00	-0.84068E+00

K_2:

-0.26877E+01	0.15060E-01	-0.50103E-01	0.47158E+01	0.17672E-02	0.46633E-02

-0.68568E+00	0.29609E-02	-0.98759E-02	0.11677E+01	0.34953E-03	0.91469E-03
0.59490E+01	0.56194E-02	-0.11121E+00	-0.19920E+02	0.82412E-02	-0.50725E-02
0.14845E+01	0.11049E-02	-0.21770E-01	-0.49805E+01	0.16124E-02	-0.99011E-03

λ_c:

-0.40078E+01	-j0.58538E-01
-0.40078E+01	j0.58538E-01
-0.42412E+01	j0.00000E+00
-0.90392E+01	-j0.54009E-18
-0.34572E+02	j0.14704E+03
-0.34572E+02	-j0.14704E+03
-0.70783E+02	j0.20141E+03
-0.70783E+02	-j0.20141E+03
-0.51166E+03	-j0.29231E-13
-0.70111E+03	j0.55610E+03
-0.70111E+03	-j0.55610E+03
-0.19786E+04	j0.00000E+00

$\underline{\alpha} = 5.0$

K_1:

0.42591E+04	0.17328E+03	-0.10883E+04	0.12932E+04	-0.64980E+02	-0.86588E+02
0.87170E+03	0.45836E+01	-0.87007E+02	0.26620E+03	0.10913E+02	0.25190E+01
0.94751E+03	0.69329E+02	-0.79118E+03	0.50793E+03	-0.15413E+03	-0.15762E+04
0.20520E+03	0.11563E+02	-0.24151E+02	0.10212E+03	0.26384E+01	-0.12398E+00

K_2 :

-0.12805E+02	0.21275E+00	-0.72596E+00	0.17474E+02	0.26601E-01	0.64764E-01
-0.15505E+01	0.31202E-01	-0.10714E+00	0.19694E+01	0.39584E-02	0.94450E-02
0.26597E+02	0.74855E-01	-0.16176E+01	-0.91358E+02	0.12115E+00	-0.77733E-01
0.30705E+01	0.10947E-01	-0.23441E+00	-0.10655E+02	0.17542E-01	-0.11205E-01

λ_c :

-0.10004E+02	j0.28755E-01
-0.10004E+02	-j0.28755E-01
-0.10101E+02	j0.00000E+00
-0.13458E+02	j0.12591E-17
-0.37954E+02	j0.14702E+03
-0.37954E+02	-j0.14702E+03
-0.73930E+02	j0.20144E+03
-0.73930E+02	-j0.20144E+03
-0.51470E+03	j0.46668E-14
-0.70413E+03	-j0.55611E+03
-0.70413E+03	j0.55611E+03
-0.19816E+04	j0.28422E-13

The following optimal linear feedback gains are used in fine motion phase of simulations corresponding to figures 5.8 through 5.11.

$$R = I$$

$$Q = Diag\{(10^2, 10^0, 10^2, 10^2, 10^0, 10^2), (10^4, 10^0, 10^2, 10^4, 10^0, 10^2)\}$$

$(A, B) \rightarrow (A, B)_3$, $, mp = 2.0kg$

$\underline{\alpha} = 5$

K_1 :

0.78158E+04	0.27172E+03	-0.17787E+04	0.26444E+04	0.87105E+01	-0.28412E+03
0.16552E+04	0.87491E+02	-0.18308E+03	0.60471E+03	0.28022E+02	-0.14880E+02
0.29814E+04	0.15708E+03	-0.52124E+03	0.23448E+04	-0.93971E+02	-0.12331E+04
0.67244E+03	0.39473E+02	-0.93164E+02	0.41593E+03	0.14309E+02	-0.72004E+01

λ_c :

-0.10001E+02	j0.57962E-06
-0.10001E+02	-j0.57962E-06
-0.10357E+02	j0.83468E-21
-0.35364E+02	j0.32729E+02
-0.35364E+02	-j0.32729E+02
-0.11771E+03	-j0.11391E-13
-0.30442E+02	-j0.15537E+03
-0.30442E+02	j0.15537E+03
-0.10201E+03	-j0.34038E+03
-0.10201E+03	j0.34038E+03
-0.27917E+04	-j0.88306E-14
-0.63314E+04	-j0.53803E-14

Same formulation, except the weighting matrix Q is as follows:

$$Q = Diag\{(10^2, 10^0, 10^2, 10^2, 10^0, 10^2), (10^4, 10^1, 10^3, 10^4, 10^1, 10^3)\}$$

$$\underline{\alpha} = 5.$$

K_1:

0.80388E+04	0.15699E+03	-0.54263E+04	0.27468E+04	-0.36324E+03	-0.11956E+04
0.17349E+04	0.89812E+02	-0.19396E+03	0.63669E+03	0.29608E+02	-0.19627E+02
0.28961E+04	0.17644E+03	-0.18421E+04	0.23416E+04	-0.46986E+03	-0.60577E+04
0.66524E+03	0.39293E+02	-0.93051E+02	0.42134E+03	0.12619E+02	-0.87445E+01

λ_c:

-0.10001E+02	j0.56407E-06
-0.10001E+02	-j0.56407E-06
-0.10357E+02	j0.72527E-21
-0.34115E+02	-j0.30279E+02
-0.34115E+02	j0.30279E+02
-0.90743E+02	-j0.11035E-14
-0.43792E+02	-j0.14460E+03
-0.43792E+02	j0.14460E+03
-0.14756E+03	-j0.27070E+03
-0.14756E+03	j0.27070E+03
-0.43867E+04	j0.22985E-13
-0.88560E+04	-j0.16685E-14

BIBLIOGRAPHY

A. Modelling

1. Midha, A., Erdman, A.G., Frorib, D.A., "Finite Element Approach to Mathematical Modelling of High-Speed Elastic Linkages", *Mechanism and Machine Theory*, Vol.13 , 1978, pp. 603-618.

2. Yoo, W.S., Haug, E.J., "Dynamics of Flexible Mechanical Systems" , *Third Army Conference on Applied Mathematics and Computing*, May 13-16, 1985 , Atlanta , Georgia.

3. Shabana, A.A., Wehage, R.A., "A coordinate Reduction Technique for Dynamic Analysis of Spatial Substructures with Large Angular Rotations", *J.Struct. Mech.*, 1983, pp. 401-431.

4. Sunada, W., Dubowsky, S., "The Application of Finite Element Methods to the Dynamic Analysis of Flexible Linkage Systems", *Journal of Mechanical Design*, Vol. 103, 1983, pp. 643-651.

5. Sunada, W., Dubowsky, S., "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members", *Transactions of ASME, J. Mech.,Trans., Automation and Design*, Vol 105, 1983, pp. 42-51.

6. Hurty, W.C., "Dynamic Analysis of Structural Systems Using Component Modes", *AIAA Journal*, Vol. 3, No. 4, 1965, pp. 678-685.

7. Rubin, S., "Improved Component-Mode Representation for Structural Dynamic Analysis", *AIAA Journal*, Vol 13, No. 8, 1975, pp. 995-1006.

8. Hintz, R.M., "Analytical Methods in Component Modal Synthesis", *AIAA Journal*, Vol. 13, No. 8, 1975, pp. 1007-1016.

9. Hollerbach, J.M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", *IEEE Trans. on SMC* , Vol. SMC-10, No. 11, 1980, pp. 730-736.

10. Hughes, P.C., "Dynamics of a Chain of Flexible Bodies", *The Journal of Astronautical Sciences*, Vol. XXVII, No.4, 1979, pp. 359-380.

11. Book, W.J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms", *International Journal of Robotic Research*, Vol. 3, No. 3, Fall 1984, pp.87-101.

12. Book, W.J., Majette, M., "Controller Design for Flexible, Distributed Parameter Mechanical Arms Via Combined State Space and Frequency Domain Techniques", *Journal of Dynamic Systems, Measurement, and Control* , Vol. 105, Dec. 1985, pp.245-254.

13. Book, W.J., Majette, M., Kong, M., "The Distributed System Analysis Package (DSAP) and Its Applications to Modeling Flexible Manipulators", *Final Report Submitted to Charles Stark Draper Lab., Inc.*, Subcontract No. 551, NASA Contract NAS 9-13809, (RMS Modeling and Integration Analysis), July, 1979.

14. Usoro, P.B., Nadira, R.,Mahil, S.S., "Advanced Control of Flexible Manipulators", *Phase I Final report*, NSF Award Number ECS-8260419, April 1983.

15. Hastings, G.G., Book, W.J., "Verification of a Linear Dynamic Model for Flexible Robotic Manipulators", *IEEE Control Systems Magazine, IEEE Control Systems Society*, April, 1987.

16. Alberts, T.A, Hastings, G.G., Book, W.J., Dickerson, S.L., "Experiments in Optimal Control of a Flexible arm with Passive Damping", *VPI & SU/AIAA Symposium on Dynamics and Control of Large Flexible Structures*, Blackburg, VA, June, 1985.

17. Cannon, R.H.Jr., Schmitz, E., "Initial Experiments on the End-Point Control of a Flexible One-Link Robot", *The International Journal of Robotics Research*, Vol. 3, No. 3, Fall 1984, pp.. 62- 75.

18. Balas, M.J., "Active Control of Flexible Systems", *Journal of Optimization Theory and Applications*, Vol. 25, No. 3, July 1978, pp. 415-436.

19. *SMP Reference Manual*, Inference Corporation, 1983.

20. Book, W.J., Maizza-Netto, O., Whitney , D.E., "Feedback Control of Two Beam , Two Joint Systems with Distributed Flexibility", *ASME Journal of Dynamic Systems, Measurement, and Control*, 97G, Dec. 1975.

21. Leu, M.C., Hemati, N., "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators", *Journal of Dynamic Systems, Measurement, and Control*, Vol.108, Sept. 1986, pp. 172-179.

22. Koplik, J., Leu, M.C., "Computer Generation of Robotic Dynamics Equations and the Related Isues", *Journal of Robotic Systems*, 3(3), 1986, pp. 301-319.

23. Cetinkunt, S., Siciliano, B., Book, W.J., "Symbolic Modeling and Dynamic Analysis of Flexible Manipulators", *Proc. of the 1986 IEEE International Conf. on Systems, Man, and Cybernetics*, Atlanta, Georgia, Oct 14-17, 1986, Vol. 1, pp. 798 - 803.

24. Cetinkunt, S., Book, W.J., "Symbolic Modeling of Flexible Manipulators", *Proc. of 1987 IEEE International Conf. on Robotics and Automation*, March 31-April 3, 1987, Raleigh, NC., Vol. 3, pp. 2074-2080.

25. Book, W.J., Cetinkunt, S., "Symbolic Modeling and Dynamic Analysis of Flexible Manipulators", *Technical Report*, The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA , May 1987.

26. Neuman, C.P., Murray, J.J., "The Complete Dynamic Model and Customized Algorithms of the Puma Robot", *IEEE Trans. on System, Man, and Cybernetics*, Vol. SMC-17, No. 4, July/August 1987, pp. 635-644.

B. Robotics and Non Adaptive Control

1. Luh, J.Y.S., "An Anatomy of Industrial Robots and Their Controls", *IEEE Trans. on Auto. Control*, Vol AC-28, Feb. 1983, pp. 133-153.

2. Luh, J.Y.S., "Conventional Controller Design for Industrial Robots - A Tutorial",

IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-12, No. 3 may/june 1983 , pp. 298-316.

3. Luh, J.Y.S., Fisher, W.D., Paul, R.P.C., "Joint Torque Control by a Direct Feedback for Industrial Robots", *IEEE Trans. on Auto. Control*, Vol. AC-28, No. 2, Feb. 1983, pp. 153-161.

4. Luh, J.Y.S., Walker, M.W., Paul, R.P.C., "Resolved- Acceleration Control of Mechanical Manipulators", *IEEE Trans. on Auto. Control*, Vol. AC- 25, 1980, pp. 468-474.

5. Luh, J.Y.S., Walker, M.W., Paul, R.P.C., "On-line Computational Scheme for Mechanical Manipulators", *ASME Journal of Dyn. Sys. Meas. and Control*, Vol. 102, 1980, pp. 69-76.

6. Wu, C., Paul, R.P.C., "Resolved Motion Force Control of Robotic Manipulators", *IEEE Trans. on Syst, Man and Cyb.*, Vol. SMC-12, No. 3, June 1982, pp. 266-275.

7. Shimano, B., Roth, B., "On Force Sensing Information and Its use in Controlling Manipulators", *Proc. of 8th Industrial Symposium on Industrial Robots*, pp. 119-126.

8. Whitney, D.E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators", *Journal of Dynamic System, Meas. and Control*, Vol 122, Dec 1972.

9. Walker, M.W., Orin, D.E., "Efficient Dynamic Computer Simulation of Robotic Manipulators", *Journal of Dyn. Syst. , Meas. and Control*, Sep. 1982, Vol. 104, pp. 215-211.

10. Raibert, M.H., Craig, J.J., "Hybrid Position/Force Control of Manipualtors", *Journal of Dyn. Syst., Meas. and Control*, Vol. 102, June 1981, pp. 126-133.

11. Raibert, M.H., Horn, B.K.P., "Manipulator Control Using the Configuration Space Method", *The Industrial Robot*, June 1978, pp. 69-73.

12. Takegaki, M., Arimoto,S., "A New Feedback Method for Dynamic Control of Manipulators", *Journal of Dyn. Syst. Meas. and Control*, Vol. 102, June 1981, pp. 119-125.

13. Bryson, A.E.Jr., Ho, Y-C., *Applied Optimum Control*, Hemisphere Publishing Corporation, 1975.

14. Kalman, R.E., "On the General Theory of Control", *Proc. of the First Int. Cong. on Automatic Control*, Vol. 1. London: Butterworth Scientific Institute, 1961, p.481

C. Path and Trajectory Planning (including Min-Time Problem)

1. Bobrow, J.E., Dubowsky, S., Gibson, J.S., "On the Optimal Control of Robotic Manipulators with Actuator Constraints", *Journal of International Robotics Research*, MIT Press, 1984.

2. Dubowsky, S., Shiller, Z., "Optimal Dynamic Trajectories for Robotic Manipulators", *5th CISM-IFToMM Symposium on The Theory and Practice of Robots and Manipulators*, 26-29 june 1984, Udine, Italy.

3. Dubowsky, S., Norris, M.A., Shiller, Z., "Time Optimal Trajectory Planning For Robotic Manipulators with Obstacle Avoidance : A CAD Approach", *IEEE Conference on Robotics and Automation*, 1986, pp. 1906-1912.

4. Shin, K.G., McKay, N.D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", *IEEE Trans. on Auto. Control*, Vol.AC-30, June 1985, pp. 531-541.

5. Meckl, P.H., Seering, W.P.," Active Damping in a Three- Axis Robotic Manipulator", *IEEE 1987 Int. Conf. on Robotics and Automation*, Vol. 3, pp. 1690-1695.

6. Luh, J.Y.S., Lin, C.S., "Optimum Path Planning for Mechanical Manipulators", *Transactions of ASME, Journal of Dyn. Sys, Meas.,Control*, Vol. 102, June 1981, pp. 142-151.

7. Luh, J.Y.S., Lin, C.S., Chang, P.R., "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots", *IEEE Trans. on Auto. Control*, Vol. AC-28, No. 12, Dec 1983, pp. 1066-1073.

8. Paul, R.P.C., "Manipulator Cartesian Path Control", *IEEE Trans. on Syst.,Man and Cyb.*, SMC-12, 1979, pp. 702-711.

9. Paul, R.P.C., Shimano, B., "Compliance and Control", *Proceedings of JACC*, 1976 , pp. 694-699.

10. Taylor, R.H.,"Planning and Execution of Straight Line Manipulator Trajectories", *IBM Journal of Research and Development*, 1979, pp.424-436.

D. Adaptive Control

1. Landau, Y.D., *Adaptive Control*, Marcer Dekker, Inc. 1979.

2. Goodwin, G.C., Sin, K.S., *Adaptive Filtering, Prediction and Control*, Prentice - Hall, Inc. 1984.

3. Popov, V.M., *Hyperstability of Automatic Control Systems*, Springer, New York, 1973.

4. Balestrino, A., De Maria, G., Sciavicco, L., "An Adaptive Model Following Control for Robotic Manipulators", *Journal of Dynamic Systems, Measurement and Control*, Sep. 1983, Vol. 105, pp.143-151.

5. Balestrino, A., De Maria, G., Sciavicco, L., "Hyperstable Adaptive Model Following Control of Nonlinear Plants", *Systems & Control Letters*, Volume 1, Number 4, Jan. 1982, pp.232-236 .

6. Balestrino, A., De Maria, G., Sciavicco, L., "Adaptive Control of Manipulators in the Task Oriented Space", *Technical Paper*, SME, RI, 1984.

7. Balestrino, A., De Maria, G., Zinober, A.S.I., "Nonlinear Adaptive Model-Following Control", *Automatica*, Vol. 20, No. 5, 1984, pp. 559-568.

8. Balestrino, A., De Maria, G., Siciliano, B., "Hyperstable Variable Structure Control for a Class of Uncertain Systems", *IFAC Identification and System Parameter Estimation*, 1985, pp. 1913-1919.

9. Dubowsky, S., DesForges, D.E., "The Application of Model- Refernce Adaptive Control to Robotic Manipulators", *Journal of Dynamic Systems, Measurement, and Control*, Sept. 1979, Vol. 101, pp. 193-200.

10. Annex, R.P. Jr , Hubbard, M., "Modelling and Adaptive Control of a Mechanical Manipulator", *Journal of Dynamic Systems, Measurement, and Control*, Sept. 1984, Vol. 106, pp. 211-217.

11. Lee, C.S.G., Chung, M.J., "An Adaptive Control Strategy for Mechanical Manipulators", *IEEE Trans. on Auto. Control*, 1983.

12. Koivo, A.J., Guo, T.M., "Adaptive Linear Controller for Robotic Manipulators", *IEEE Trans. Auto . Control*, Vol. AC-28, 1983, pp. 162-171.

13. Craig, J.J., Hsu, P., Sastry,S.S., "Adaptive Control of Mechanical Manipulators", *IEEE Robotics and Automation Conference*, San Fransisco, CA, 1986, pp. 190-195.

14. Hsia, T.C., "Adaptive Control of Rbot Manipulators - A Review", *IEEE Robotics and Automation Conference*, San Fransisco, CA, 1986, pp. 183-189.

15. Siciliano, B., Yuan, B., Book, W.J., "Model Reference Adaptive Control of a One Link Flexible Arm", *25th IEEE Conference on Decision and Control*, Athens, Greece, December 10-12, 1986.

16. Siciliano, B., Calise, A.J., Jonnalagadda, V.R.P., "Optimal Output Fast Feedback in Two-time Scale Control of Flexible Arms", *Proceedings of 25th Conf. on CDC*, 1986, Athens, Greece, pp. 1400-1404.

17. Narendra, K.S., Valavani, L.S., "A comparison of Lyapunov and Hyperstability Approaches to Adaptive Control of Continuous Systems", *IEEE Trans. on Auto.*

Control, Vol. AC-25, No. 2, 1980, pp. 243-247.

18. Monopoli, R.V., "Model reference Adaptive Control with an Augmented Error Signal", *IEEE Trans. on Auto Control*, Vol. AC-19, No.5, 1974, pp. 474-484.

19. Parks, P.C., "Liapunov Redesign of Model Reference Adaptive Control Systems", *IEEE Trans. on Auto Control*, Vol. AC-11, No. 3, 1966, pp. 362-367.

20. Erzberger, H., "Analysis and Design of Model Following Control Systems by State Space Techniques", *Proc. JACC*, Ann Arbor, 1968, pp. 572-581.

21. Horowitz, R., Tomizuka, M., "An Adaptive Control Scheme for Mechanical Manipulators - Compensation of Nonlinearity and Decoupling Control", *ASME J. of Dynamic Systems, Measurement and Control*, 1981.

22. Lim., K.L., Eslami, M., "Robust Adaptive Controller Design for Robot Manipulator Systems", *IEEE International Conf. on Robotics and Automation*, Los Angles, CA., 1986.

23. Seraji, H., "A New Approach to Adaptive Control of Manipulators" *Journal of Dynamic Systems, Measurement, and Control*, Sept. 1987, Vol. 109, pp. 193-202.

24. Chen, Y.H., "On the Robustness of Mismatched Uncertain Dynamical Systems", *Journal of Dynamic Systems, Measurement, and Control*, March 1987, Vol. 109, pp. 29-35.

25. Chen, Y.H., "Adaptive Robust Model Following Control and Application to Robot Manipulators", *Journal of Dynamic Systems, Measurement, and Control*, Sept. 1987, Vol. 109, pp.209-215.

E. Linear Control and Others

1. Anderson, B.D.O., Moore, B.J., "Linear System Optimization with Prescribed Degree of Stability", *Proc. IEE*, Vol. 116, No. 12, Dec. 1969, pp. 2083-2087.

2. Kreindler, E., Rothschild, D., "Model Following in Linear-Quadratic Optimiza-

tion", *AIAA Journal*, Vol. 14, No. 7, July 1976, pp. 835-842.

3. Moore, B.C., "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment", *IEEE Trans. Auto. Cont.*, Oct. 1976, pp. 689-692.

4. Porter, B., D'Azzo, J.J., "Comments on "On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment"", *IEEE Trans. Auto. Cont.*, Vol. AC-22, No. 5, Oct. 1977, pp. 888-889.

5. Porter, B., D'azzo, J.J., "Algorithm for Closed-loop Eigenstructure Assignment by State Feedback in Multivariable Linear Systems", *Int. J. Control*, 1978, Vol. 27, No. 6, pp. 943-947.

6. Silverthorn, J.T., Reid, G.J., "Computation of the Subspaces for Entire Eigenstructure Assignment via the Singular Value Decomposition", *The 19th IEEE Conf. on Decision and Control*, 1980, pp. 1206-1207.

7. Fahmy, M.M., O'Reilly, J., "On Eigenstructure Assignment in Linear Multivariable Systems", *IEEE Trans. Auto. Cont.*, Vol. AC-27, No. 3, June 1982, pp. 690-693.

8. Saridis, G.N., "Intelligent Robotic Control", *IEEE Trans. Auto. Cont.* Vol. AC-28, No. 5, May 1983, pp. 547-557.

9. Sangveraphunsiri, V., "The Optimal Control and Design of a Flexible Manipulator Arm", Ph.D. Thesis, School of Mechanical Engineering, Georgia Institute of Technology, 1984.

BIOGRAPHY

Sabri Cetinkunt was born in [redacted] on [redacted] He received the degree of Bachelor of Science in Mechanical and Aerospace Engineering in 1982 from the Technical University of Istanbul, Turkey. He was a TÜBİTAK (The Scientific and Technical Research Council of Turkey) scholar during the high school and undergraduate years.

He was enrolled in the graduate program at the School of Mechanical Engineering of Georgia Institute of Technology from June 1983 to December 1987. Graduate study has concentrated on Robotics, Automatic Control and Dynamic Systems. He received the Master of Science degree in December 1984, and continued in the Ph.D. program. While studying at Georgia Tech, he was a research and and a teaching assistant in the School of Mechanical Engineering.

His research interests include automatic control, microprocessor applications, artifical intelligence, robotics and applied mathematics.