NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

K1185

AUTOMATIC CONTROL OF
ROBOT MOTION

by

George P. Kalogiros

December 1987

Thesis Advisor	George J. Thaler

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

AUTOMATIC CONTROL OF ROBOT MOTION

12. PERSONAL AUTHOR(S)
Kalogiros George, P.

13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Year, Month, Day) 1987 December	15. PAGE COUNT 261

16. SUPPLEMENTARY NOTATION

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Curve following velocity loop servo
			Computer adaptive model, Direcr-drive servo motors
			Rectangular and Revolute robot configurations

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The feasibility of controlling the three link rectangular and revolute robots with an adaptive computer simulation model, using a curve following technique, is investigated.

Both configurations are tested for different load conditions, for rejection of random disturbances and for robustness in the case of servo motor parameter variations.

The interactive nonlinear dynamics of the revolute robot, such as coupling inertia, actuator dynamics, centripetal and coriolis forces, are also investigated. First a gravity-free environment is assumed and then the robot arms are tested under gravitational torques.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL George J. Thaler	22b. TELEPHONE (Include Area Code) (408) 646-2134	22c. OFFICE SYMBOL 62Tr

1

Automatic Control of
Robot Motion

by

George P. Kalogiros
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1979

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

ABSTRACT

The feasibility of controlling the three link rectangular and revolute robots with an adaptive computer simulation model, using a curve following technique, is investigated.

Both configurations are tested for different load conditions, for rejection of random disturbances and for robustness in the case of servo motor parameter variations.

The interactive nonlinear dynamics of the revolute robot, such as coupling inertia, actuator dynamics, centripetal and coriolis forces, are also investigated. First a gravity-free environment is assumed and then the robot arms are tested under gravitational torques.

3

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

13

16

ACKNOWLEDGMENTS

I would like to express my thanks to my thesis advisor, Distinguished Professor, George J. Thaler for his guidance and assistance in this study.

I would like also to thank the Greek Navy and all the Greek tax-payers for having paid for my course of studies.

Finally, I would like to dedicate this thesis and give special thanks to my wife Katerina for her love, support, understanding and patient help throughout my studies in NPS.

I. INTRODUCTION

Current industrial robot arms tend to have been justified because of their untiring nature, predictability, precision, reliability and ability to work in relatively hostile environments. Besides, robots frequently increase industrial productivity, improve the overall product quality, allow replacement of human labor in monotonous and mainly hazardous tasks and also may save on materials or energy.

The dynamic motion of a robot arm is strongly affected by mechanical design, physical properties of the arm and the effects of gravity. Such factors as joint friction, coupling inertia, centripetal forces, actuator dynamics and gravity effects produce an interactive nonlinear dynamic system. In order to cope with the rapidly changing dynamics, a robust and flexible control algorithm is required.

The design of an adaptive algorithm consists basically of combining a particular parameter estimation technique and any feedback control law. In this study, the use of a curve following technique to control a robot arm is investigated. The very important feature of this control scheme is that the amplifier is intentionally driven to saturation so full advantage of the maximum available power can be taken while

in linear controllers driving the amplifier to saturation must be avoided.

First the three link rectangular robot is used as the simulation model and then the same tests are applied to the three link revolute robot. The same control scheme is used to drive all servo motors. An ideal second order model is chosen to approximate the servo motor and the adaptive algorithm updates the second order position, acceleration and gain constant using the servo motor position output.

Each of these configurations is tested first for different load conditions, then for random disturbance rejection and finally for robustness in the case of slight variation of the servo motor parameters. In all these tests the interactive nonlinear dynamics of the system as coupling inertia, centripetal forces and coriolis forces are also investigated. Initially a gravity-free environment is assumed and then all the tests are repeated assuming the robot arm operating under gravitational torques.

Definitions and several basic ideas about robotics and materials which are referred to in this study, are included in Chapter II. The development of the computer simulation model and the simulation studies of this model are discussed in Chapter III. Chapter IV deals with the development of the adaptive algorithm for the three link rectangular robot and Chapter V contains the modelling as well as the equations of motion this robot while the simulation results are studied

in Chapter VI. The modelling and the equations of motion of the three link revolute robot are presented in Chapter VII and the development of the adaptive model for the revolute configuration is presented in Chapter VIII. Chapter IX contains the simulation studies of the revolute robot and Chapter X concludes the studies of both configurations, indicating also some areas for further study. The detailed derivation of the mathematical models for both rectangular and revolute robots as well as the basic DSL/VS simulation programs used in the course of this thesis research are listed in Appendices A through K.

II. FUNDAMENTAL ROBOTICS

A. INTRODUCTION

The word **robotics** was invented by the Isaac Asimov, one of the best of the science fiction writers, to describe the science of dealing with robots. In one of his stories called 'Runaround', which appeared in the March 1942 issue of 'Astounding Science Fiction', Asimov propounded the famous Three Laws of Robotics.

1. A robot must not harm a human being or, through inaction, allow human being to come to harm.

2. A robot must always obey human beings unless that is in conflict with the First Law.

3. A robot must protect itself from harm unless that is in conflict with the First or Second Law.

B. DEFINITION OF ROBOT

As can be imagined from the confused usage of the word **robot** there is no international agreement about definitions. Several definitions of robot exist depending on what a robot actually consists of. Some of the most commonly used definitions are:

1. A robot is a mechanical positioning system.

2. A robot is a pick and place device.

3. The popular conception is of a mechanical man capable of carrying out tasks that a human might do and displaying some capability for intelligence.

4. A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks.

C. THE ROBOT ARM

Although mobile robots one day should be common, for the present the state to which the industrial robot has evolved is best referred to as a **robot arm**. Most of them are essentially a mechanical arm fixed somewhere or to another machine, fitted with a special end - effector which can be either a gripper or some sort of tool. The arm moves, by means of hydraulic, electric or pneumatic **actuators**, in a sequence of preprogrammed motions under the direction of a controller which these days is almost always microprocessor based and senses the position of the arm by monitoring **feedback** devices on each joint. The range of positions which the arm can reach is called **work volume** or **work space** or **work envelope**.

D. DIFFERENT ARM CONFIGURATIONS

The essential role of a robot arm is to move a gripper or tool to given orientations at a given set of points. Mathematically, to be able to orient an object in any way at any point in space requires an arm with six **degrees of freedom**, as depicted in Figure 2.1. Three **translational** (forward/back, right/left, up/down) to reach any point, and three **rotational** (pitch, roll, yaw) to get any orientation.
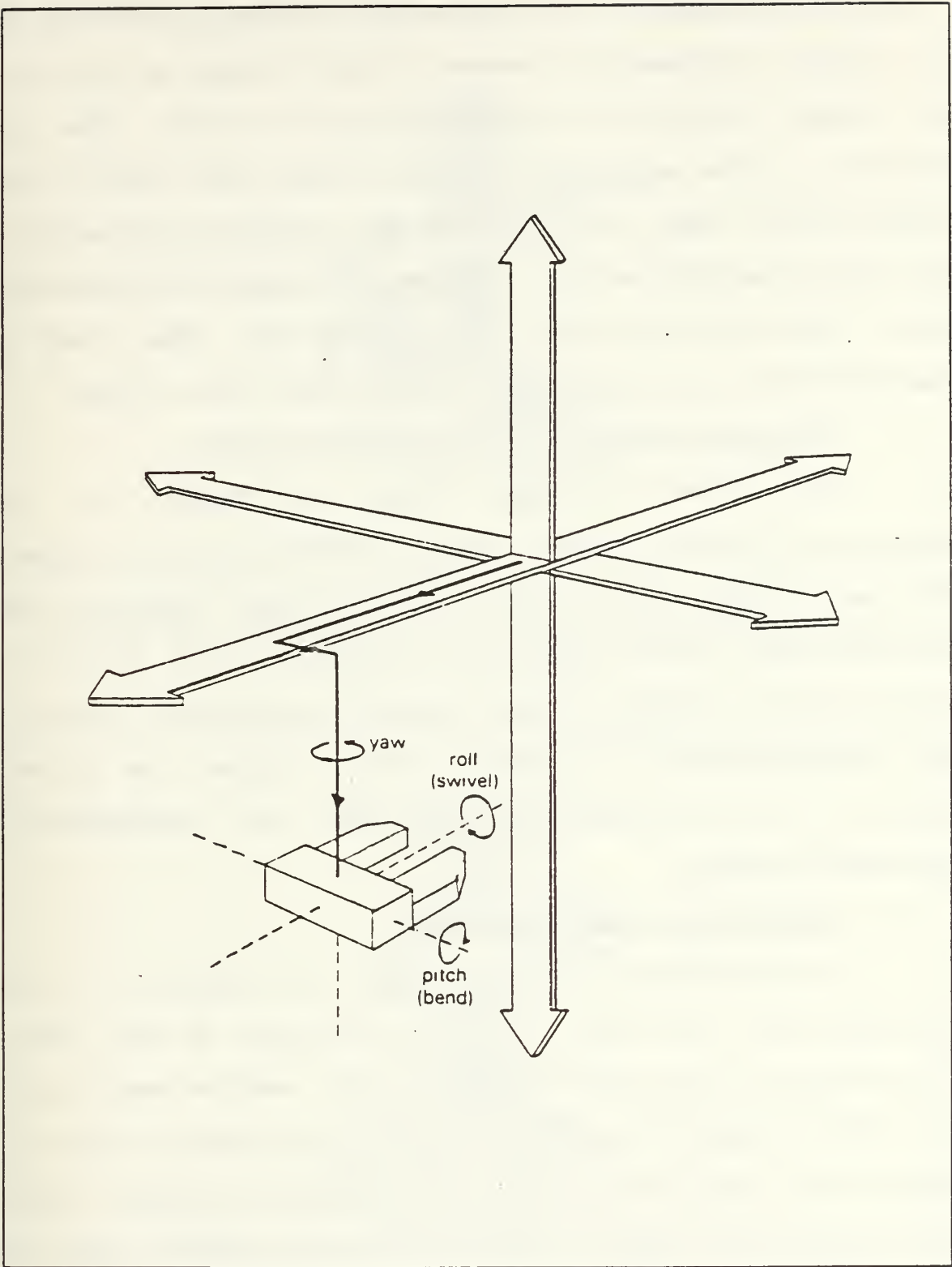
Figure 2.1   The Six Motions Required to Orient a
Gripper in any Way at any Point in Space

23

The work envelope of an arm varies in shape depending on the actual configuration chosen for the design of the arm. One common structural classification of robot arms is according to the coordinate system of the three **major** (the translational) **axes** which provide the vertical lift motion, the in/out reaching motion, and the rotational or traversing motion about the vertical lift axis of the robot. Such a classification can distinguish between four basic types.

1. Cartesian (Rectangular) Coordinate Robot

This type of robot, shown in Figure 2.2, is configured with the mutually perpendicular traversing axes. Each part of the robot's arm slides at right angles to the previous part. So the robot can reach any part of volume bounded by the length of the separate sections of the arm. This configuration is clearly ideally suited for direct usage of the mathematical **cartesian** (or **rectangular**) **coordinate system**.

2. Cylindrical Coordinate Robot

In this type of robot, shown in Figure 2.3, the horizontal arm can move in and out parallel to the base, can move up and down the vertical column, and the whole base can rotate around the vertical axis, so the end of the arm sweeps out a work envelope which is a partial cylinder. This configuration is clearly ideally suited for direct usage of the mathematical **cylindrical coordinate system**.
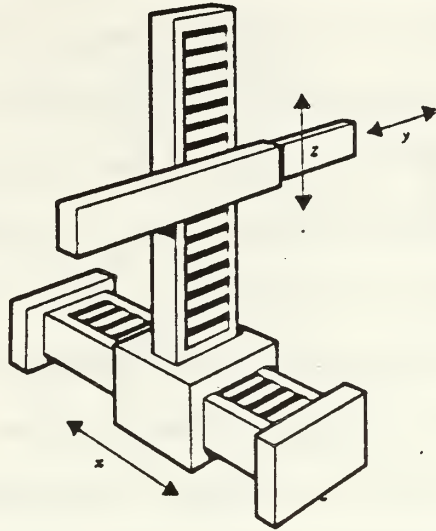
Figure 2.2   Cartesian (Rectangular) Coordinate Robot
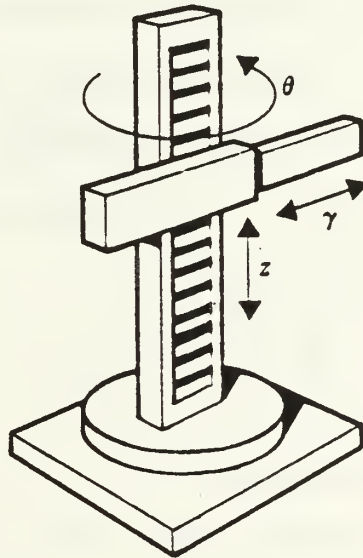
Figure 2.3   Cylindrical Coordinate Robot

3. Spherical (Polar) Coordinate Robot

The robot in Figure 2.4 has an arm capable of linear motion in and out. It is supported by two other sections, one that rotates around the base and one that rotates about an axis perpendicular to the vertical through the base. This arm can reach almost anywhere in a volume bounded by an outer and an inner hemisphere. The radii of the two hemispheres correspond to the maximum and minimum extensions of the linear section, respectively. This corresponds to the mathematical spherical (or polar) coordinate system.

4. Revolute Coordinate (Jointed Arm) Robot

An example of this fourth class of robot, sometimes known as an **anthropomorphic** robot, is shown in Figure 2.5. It consists of rotary joints called the 'shoulder' and the 'elbow'(corresponding to the human arm) all mounted on a 'waist' consisting of a rotary base which provides the third degree of freedom. This **revolute** (or **jointed arm**) configuration has the advantage of having a very large work envelope for its size, so minimizing floor space requirements.

E. TYPE OF CONTROL

1. Point-to-Point Robots

Point-to-point robots are able to move from one specified point to another but cannot stop at arbitrary points not previously designated. They are the simplest and
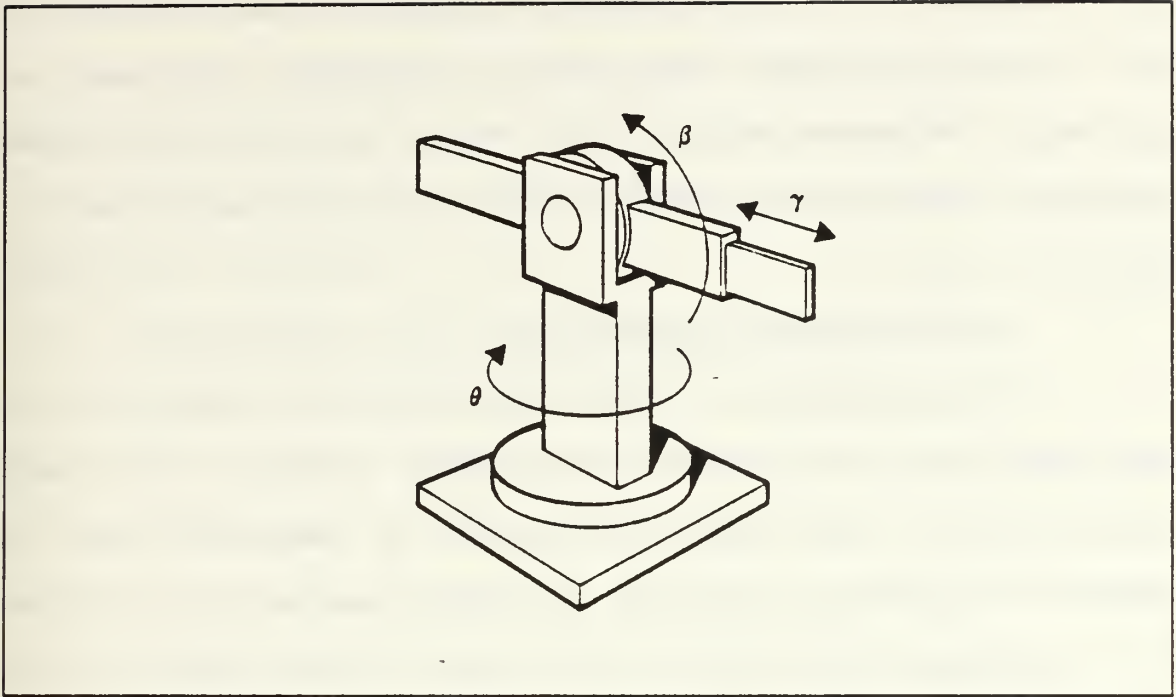
Figure 2.4 Spherical (Polar) Coordinate Robot
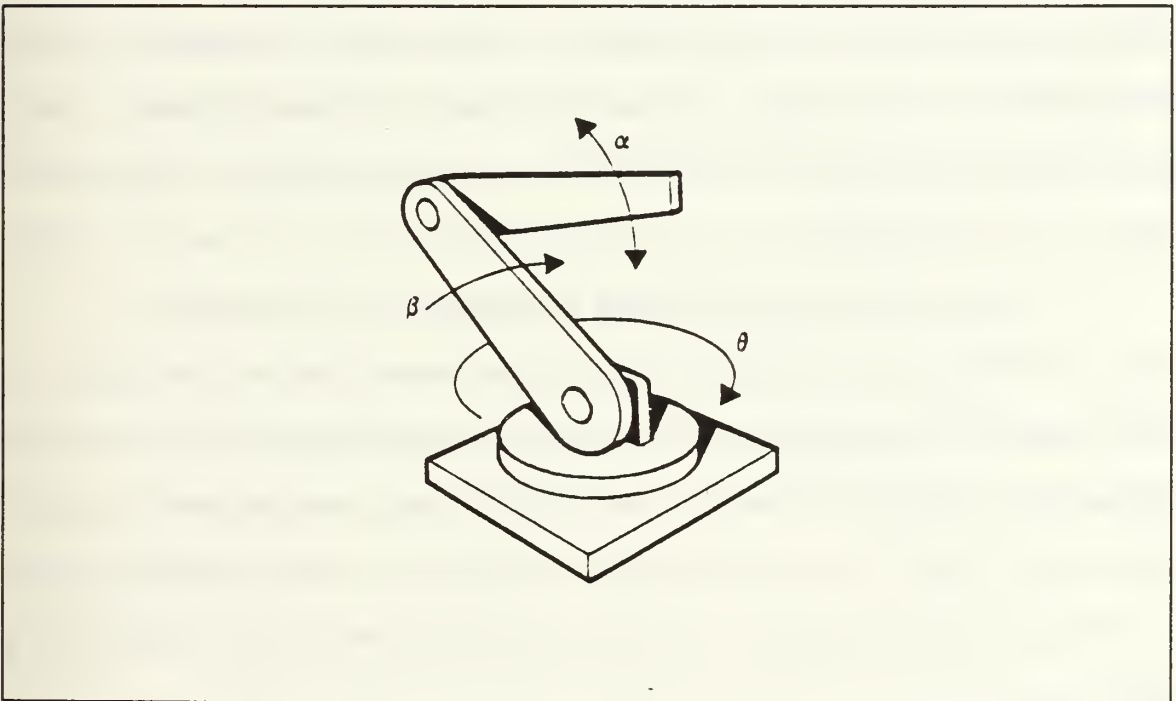
Figure 2.5 Revolute Coordinate (Jointed Arm) Robot

least expensive type of robot. Stopping points are often just mechanical stops that must be adjusted for each new operation. Point-to-point robots driven by servos are often controlled by potentiometers set to stop the robot arm at a specified point.

2. Continuous-Path Robots

Continuous-path robots are able to stop at any specified number of points along a path. However, if no stop is specified, they may not stay on a straight line or constant curved path between specified points. Every point must be stored separately in the memory of the robot.

3. Controlled-Path (Computed Trajectory) Robots

Control equipment on controlled-path robots can generate straight lines, circles, interpolated curves, and other paths with high accuracy. Paths can be specified in geometric or algebraic terms in some of these robots. Only the start and finish coordinates and the path definition are required for control.

4. Servo versus Non-Servo Robots

Servo-controlled robots have some means for sensing their position and "feeding back" the sensed position to the means of control in such a way that the control can cause a particular path to be followed. Nonservo robots have no way of determining whether or not they have reached a specified location.

28

F. CATEGORIES OF SERVO SYSTEM OPERATION

In servo control, each joint is controlled by an independent position servo, with joints moving from position to position independently. There are three possible categories of motion.

1. Sequential Joint Control

Joints are activated one at a time while the other axes are held fixed. This procedure simplifies control, but the operating time is longer than it would be if all joints operated at the same time. It provides better path control because there is no interaction between joint movements.

2. Uncoordinated Joint Control

All joints are allowed to move together, so that they follow a path determined by the relative speeds of each joint movement. There is no coordination between joints. Thus, prediction of the end-of-arm path is difficult or impossible, since varying arm loads or varying friction might change the joint velocities in a random way.

3. Terminally Coordinated Joint Control

This is the most useful type of point-to-point control but requires more control equipment. Individual joint motions are coordinated to reach their endpoints simultaneously.

G. DRIVE MECHANISM

1. Linear Drives

Examples of linear drives are the x, y and z drives of rectangular coordinate system, the vertical and horizontal positioner of cylindrical systems, and the radial drive on spherical systems. Linear motion may be generated directly by a hydraulic or pneumatic piston in a cylinder or by conversion of rotary motion to linear motion through rack and pinion gearing, lead screws, worm gears, or ball screws.

2. Rotary Drives

Most electric motors and servo drive motors generate rotary motion directly, but often at a lower torque and higher rotational velocity than is desired. It is necessary, therefore, to use some kind of gear train, belt drive, or other mechanism to convert the available high speed to lower speed with greater torque. There are also some cases in which linear hydraulic cylinders or pneumatic cylinders are used as a power source, so that the linear motion must be converted to rotary motion. The main options available for consideration are gear trains, timing belt drives, harmonic drives, rotary hydraulic motors and the recently developed direct-drive torque motors.

III. DEVELOPMENT OF THE SIMULATION MODEL

A. INTRODUCTION

The simulation model chosen was a servo motor with a curve following velocity loop, identical with the one used in Ref. 1 and Ref. 2. The velocity curve following scheme is a non linear adaptive scheme that has been used mainly in disk drive systems [Ref.1], but recent studies [Ref.2] have justified that the same scheme can also be used to control a rigid robot arm.

Although in linear controllers driving the amplifier to saturation limits must be avoided, in the case of the curve following scheme the amplifier is intentionally driven to saturation. This is a very important feature because full advantage of the motor's capabilities can be taken. Furthermore this system is simple and easily implemented in a microprocessor [Ref.3].

For a given step position command this model operates in two modes: An initial full acceleration mode and a curve following mode. The equivalent transfer function of the servo motor is [Ref.4] :

$$\frac{\Theta(s)}{V(s)} = \frac{1/K_V}{s\left(s\dfrac{JR}{K_V K_t} + 1\right)\left(s\dfrac{L}{R} + 1\right)} \tag{3.1}$$

31

where

   θ = Angular position of the shaft

   V = Applied dc voltage

   K_v = Back emf constant

   K_t = Torque constant

   J = Total inertia

   R = Armature resistance

   L = Armature inductance

When the inertia of the robot arm is added to the motor inertia, the mechanical pole of the motor becomes very small and since the ratio L/R is a small number, the electrical pole of the motor can be neglected. With the above approximations the transfer function of the robot arm and motor becomes approximately:

$$\frac{\theta(s)}{V(s)} \approx \frac{K_m}{s^2} \qquad (3.2)$$

Since in the mass production of such systems there would be substantial tolerances on the physical motors, a more complete motor model is not justified.

The second order model with a curve following velocity loop is shown in Figure 3.1. If the curve to be followed is chosen to be the deceleration curve for the idealized motor, the model will be a practical application of bang-bang control [Ref.5].
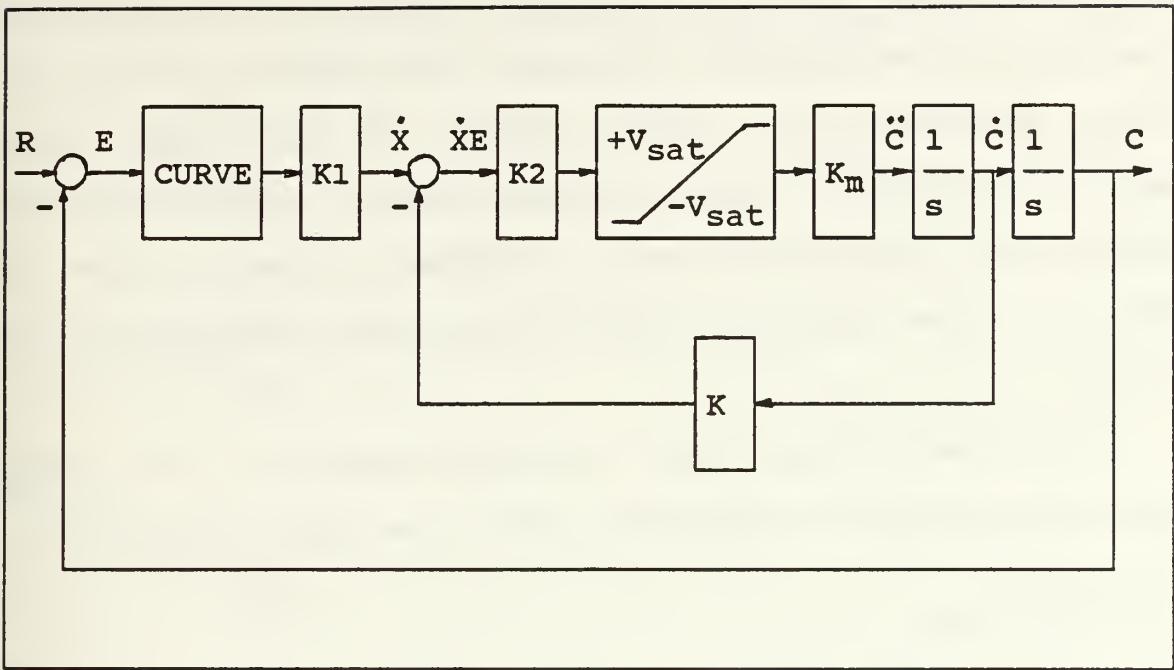
Figure 3.1 Block Diagram of the Model

B. DESCRIPTION OF THE MODEL

When a step position command is applied to the model, the error signal (E) will produce a velocity command input ($\dot{X}$). This signal saturates the amplifier and full forward drive signal is applied to the motor (full acceleration mode). As the error signal decreases, the velocity command is reduced and when it becomes equal to the velocity feedback signal ($K\dot{C}$) the system changes into curve following mode and follows the curve down until the desired position is reached.

For the system to be able to follow the curve to the desired position under different loading conditions a proper curve must be chosen. Since a parabolic curve approximates

33

the deceleration curve of an ideal motor, it was chosen as the curve to be followed. To enable the system to follow the curve accurately the curve should be below the motor deceleration curve. This is accomplished by proper selection of the gain constant K1 which reshapes the motor curve. In this case the selected value for the gain constant K1 was 0.6.

It has been shown [Ref.3] that the equation of the curve was derived from the idealized motor equations as follows:

$$C = K_m V_{sat} \tag{3.3}$$

$$C = \int C \, dt = K_m V_{sat} \, t + C(0), \quad (C(0)=0) \tag{3.4}$$

$$C = \int C \, dt = 1/2 (K_m V_{sat} \, t^2) + C(0) \tag{3.5}$$

From equation 3.4

$$t = \frac{C}{K_m V_{sat}} \tag{3.6}$$

and substituting into equation 3.5

$$C = \frac{C^2}{2 K_m V_{sat}} \tag{3.7}$$

For deceleration from initial conditions with the input R=0

$$C = - E \tag{3.8}$$

$$C = - E \tag{3.9}$$

34

Substitution of equation 3.8 and 3.9 into equation 3.7 gives

$$E = \frac{\dot{E}^2}{2\ K_m\ V_{sat}} \qquad\qquad (3.10)$$

or

$$\dot{E} = \sqrt{2\ K_m\ V_{sat}\ E} \qquad\qquad (3.11)$$

Letting

$$A = \sqrt{2\ K_m\ V_{sat}} \qquad\qquad (3.12)$$

and

$$\dot{X} = \dot{E} \qquad\qquad (3.13)$$

equation 3.11 becomes

$$\dot{X} = A\sqrt{E} = \text{velocity command input} \qquad\qquad (3.14)$$

Therefore, if the parameter A is initially calculated, then the commanded velocity curve can be generated by continuous multiplication of the calculated parameter A with the square root of the error signal (E).

The servo motor gain constant K_m is determined by the transfer function of the motor that the simulation model will control, considering the motor parameters and the effective inertia seen by the motor.

For the three link rectangular robot the chosen values of K_m were (this choice will be justified in Chapter IV)

K_{m1} = 59.29 rad/volt

K_{m2} = 90.25 rad/volt

K_{m3} = 77.44 rad/volt

and for the three link revolute robot the chosen values of K_m were (this choice will be justified in Chapter VII)

K_{m1} = 0.4225 rad/volt

K_{m2} = 0.4225 rad/volt

K_{m3} = 4.0   rad/volt

The saturation limits ($\pm V_{sat}$) of the amplifiers are determined by available power supply voltage, servo motor parameters, mechanical design of the arm, working conditions and curve constant K_1. This limit was arbitrary chosen as ±150 volts for all actuators, under different loading and working conditions.

The value of the gain constant K_2 must be chosen such that saturation of the amplifier occurs for small signals, allowing the amplifier to operate as a switch providing full forward and reverse drive signals to the motor in the curve following mode. To effectively saturate the amplifier, K_2 was given the value of 10,000.

The gain of the velocity feedback channel (K) is chosen such that X = KC when the simulation motor velocity (C) is at the desired speed for a given step position input. From Figure 3.1, for deceleration from initial conditions with R=0, and because of equation 3.9

$$\dot{X} = K_1\dot{E} = -K\dot{C} = K\dot{E} \qquad (3.15)$$

Thus, the gain of the velocity feedback channel (K) should be equal to K_1. From simulation studies, the best curve following was achieved when K was given the value of unity (K = 1).

C. SIMULATION STUDIES OF THE MODEL

To demonstrate the curve following ability of this scheme, the model was simulated using DSL/VS. Appendix A lists the DSL simulation program used for these studies. K_m was set to 59.29, a value which was found for the parameter K_{m1} of the three link rectangular robot. All the other parameters and variables used in the program are as previously discussed in this chapter. The constant CF is a factor which converts the rotary motion to linear motion. The parameter RO represents the radius of the pinion used (to be discussed in chapter IV) and was chosen to be 0.5 in.

The phase plane trajectories (velocity $\dot{C}X$ versus position CX) for a step position command of 1 inch are shown in Figure 3.2. The figure shows that the linear velocity of the model increases until the commanded velocity ($\dot{X}$) is reached. Then the model velocity follows the curve down to the commanded position. The step response of the model, depicted in Figure 3.3, shows good performance of the model.

Now since a simulation model has been found, use of this model to control the servo motors of a three link robot arm (rectangular and revolute) will be investigated.
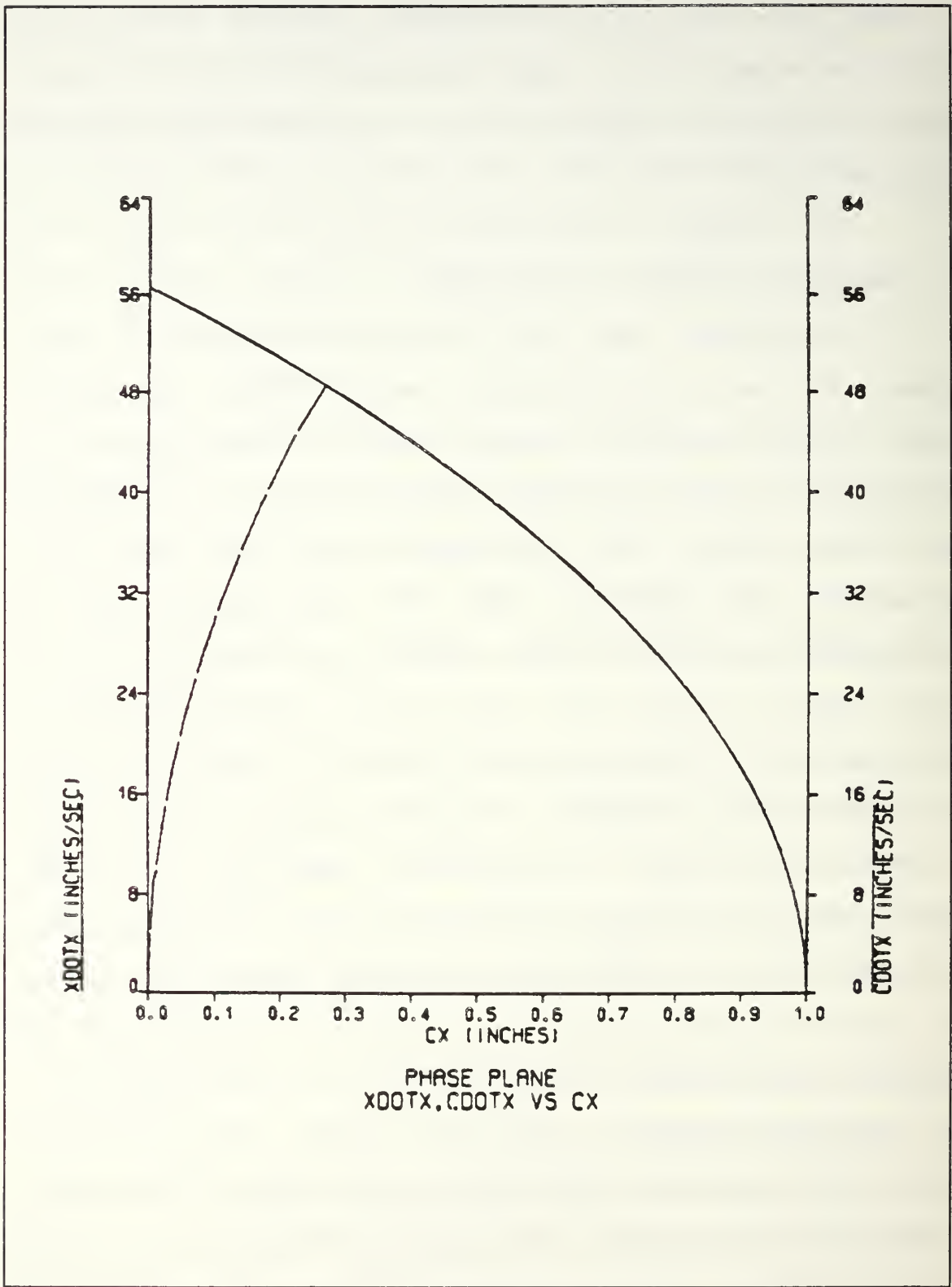
Figure 3.2   Phase Plane Trajectory of the Model
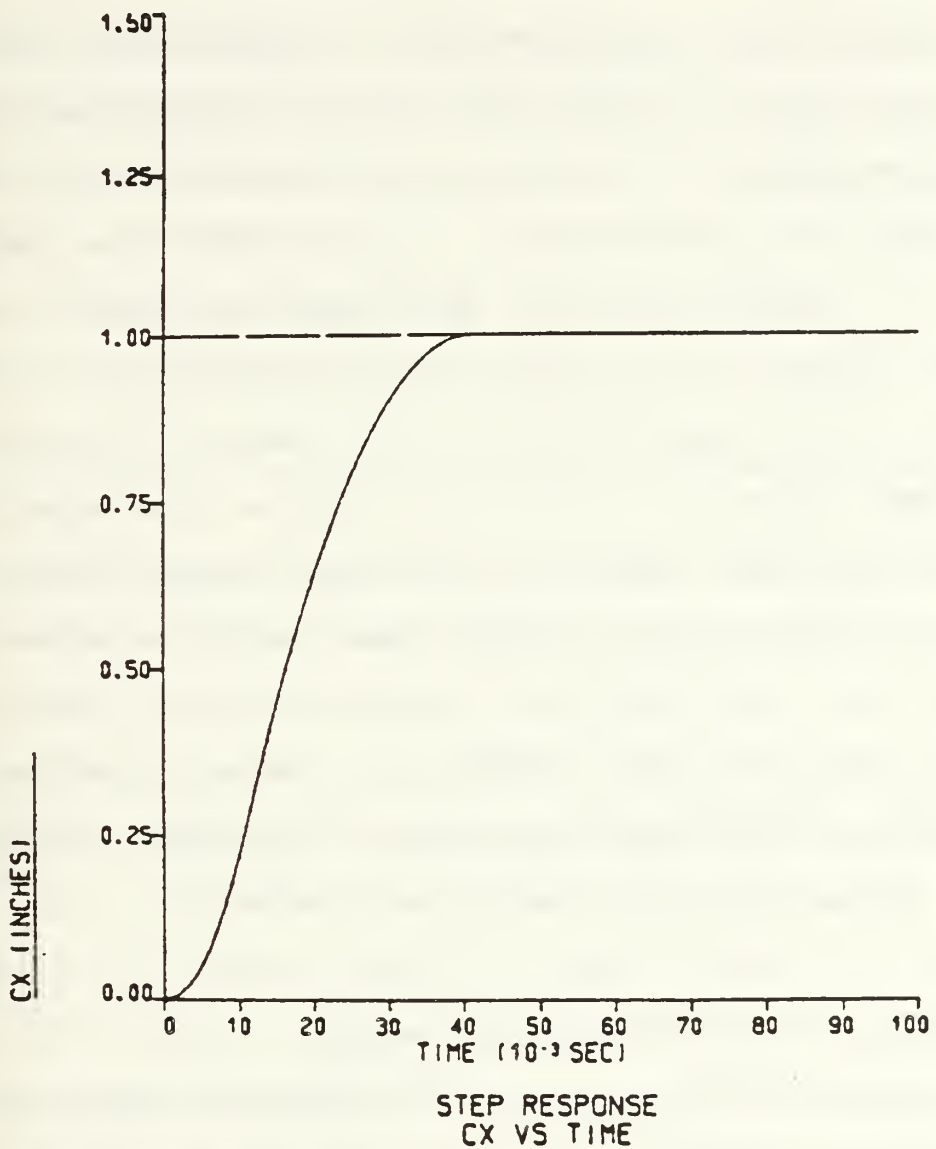
Figure 3.3   Step Response of the Model

IV. THE ADAPTIVE MODEL FOR THE THREE LINK RECTANGULAR ROBOT

A. INTRODUCTION

The second order model developed in Chapter III will be operated open loop to control its output, C, and bring it to the commanded position. But this is not the desired solution because what is really wanted is to control the output position, CR, of the real motor which must be driven to the commanded location. Due to the different dynamics of the ideal and the real motor there will be always a discrepancy between their outputs C and CR. To overcome this undesired situation, the model states and the gain constant must be adapted in a such a way that the ideal motor imitates the real one.

In this chapter, the selection of the servo motors is first discussed and then the adaptive algorithm to update the model states and gain constant is obtained.

B. SELECTION OF POSITIONING SERVO MOTORS

The selected real motor is a permanent magnet motor drive, which is widely used in industrial robots, and its data are listed in Table 1 [Ref.6]. This is a rotary motor and since a linear motion is required for the rectangular coordinate robot arm, the rotary motion is converted to linear through rack and pinion gearing.

TABLE 1

PARAMETRIC DATA FOR JOINT SERVO MOTOR

Total Inertia J_m	0.033	oz-in-sec^2/rad
Torque Constant K_t	14.4	oz-in/amp
Back emf Constant K_v	0.1012	volts-sec/rad
Damping Coefficient B_m	0.04297	oz-in-sec/rad
Ave. Terminal resistance R	0.91	ohms
Armature Inductance L	100.0	μ-henries

A rack is a long metal bar that has gear teeth cut across its width. A pinion gear is an auxiliary smaller, round gear placed so that its teeth can mesh with the teeth of the rack. Usually the rack is fixed in place. As the pinion gear rotates, it moves an attached carriage linearly along the rack. The rotary motion of the pinion gear is converted to linear motion of the carriage, as shown in Figure 4.1 [Ref.7]. Support for the carriage is provided by a guide rod or guide way. For these studies a pinion gear of radius 1.0 inch was chosen.

The same motor drive is used for the three arm joints with the same power supply. To calculate the transfer function of the servo motors, the effective joint inertias must be first calculated as follows

41
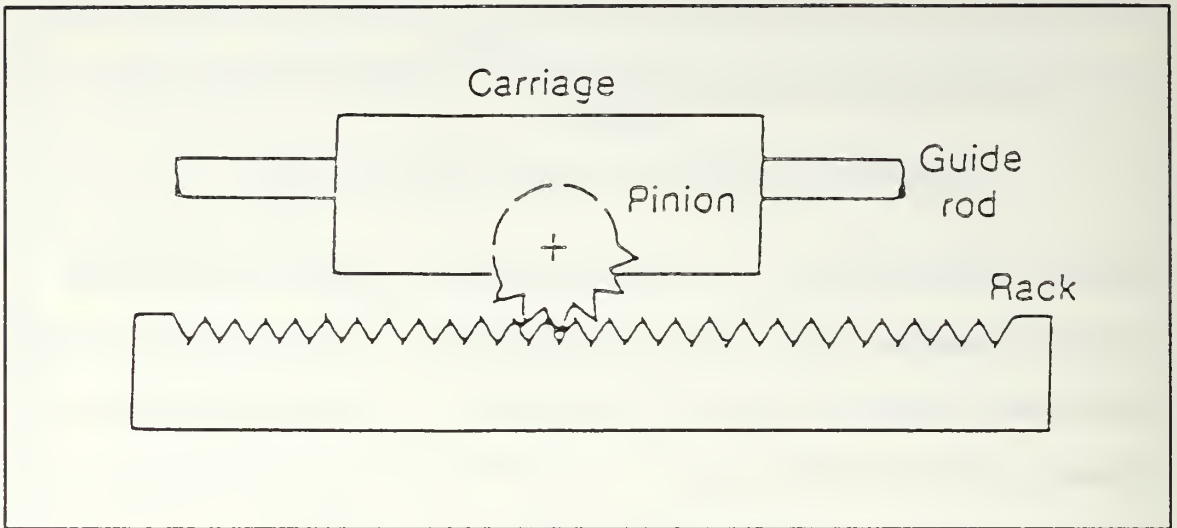
Figure 4.1 Rack-and-Pinion Gearing

$$J_{EF1} = M_1r^2 + J_{m1} \qquad (4.1)$$

$$J_{EF2} = M_2r^2 + J_{m2} \qquad (4.2)$$

$$J_{EF3} = M_3r^2 + J_{m3} \qquad (4.3)$$

Substitution of the parameters results in

$$J_{EF1} = 0.167 \quad oz.in.sec^2$$

$$J_{EF2} = 0.043 \quad oz.in.sec^2$$

$$J_{EF3} = 0.100 \quad oz.in.sec^2$$

Plugging these results and the parameters from Table 1 into equation 3.1, the transfer functions of the three servo motors become

$$G_1(s) = \frac{9.88}{s(s/9.589+1)(s/9100+1)} \quad rad/volt \qquad (4.4)$$

$$G_2(s) = \frac{9.88}{s(s/37.242+1)(s/9100+1)} \quad rad/volt \qquad (4.5)$$

42

$$G_3(s) = \frac{9.88}{s(s/16.014+1)(s/9100+1)} \text{ rad/volt} \qquad (4.6)$$

These results were used to determine the gains K_m for the model motors. Figures 4.2, 4.4 and 4.6 show the Open Loop Bode diagrams of the three real motors transfer functions respectively. From these diagrams it can be observed that the -40 db/decade slope intersects the 0 db axis at ω_1=7.7 rad/sec for the JOINT1 motor, ω_2=9.5 rad/sec for the JOINT2 motor and ω_3=8.8 rad/sec for the JOINT3 motor. Using the above frequencies as gain crossover frequencies for the second order ideal motors, we calculate

K_{m1} = 7.7^2 = 59.29 rad/volt

K_{m2} = 9.5^2 = 90.29 rad/volt

K_{m3} = 8.8^2 = 77.44 rad/volt

The Open Loop Bode diagrams of the K_m/s^2 ideal motors are shown in Figures 4.3, 4.5 and 4.7.

C. ALGORITHM TO UPDATE THE ADAPTIVE MODEL

The adaptive model depicted in Figure 4.8 is used to drive a joint servo motor. The same scheme is used for each of the three servo motors. The output of the saturating amplifier (V) is common to both the model and the servo motors. The output of the servo motor (CR) is measured, at prespecified time intervals, and fed into the Adaptive

43
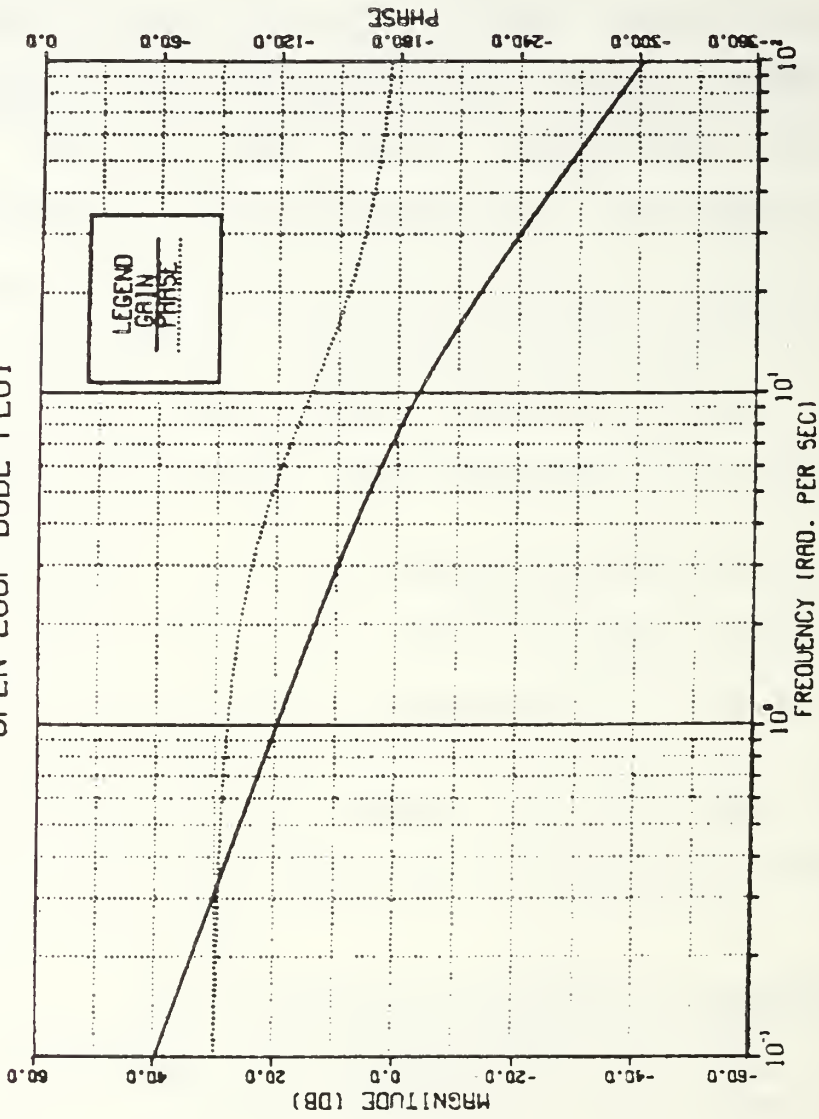
Figure 4.2   Open Loop Bode Plot of the JOINT1 Servo Motor
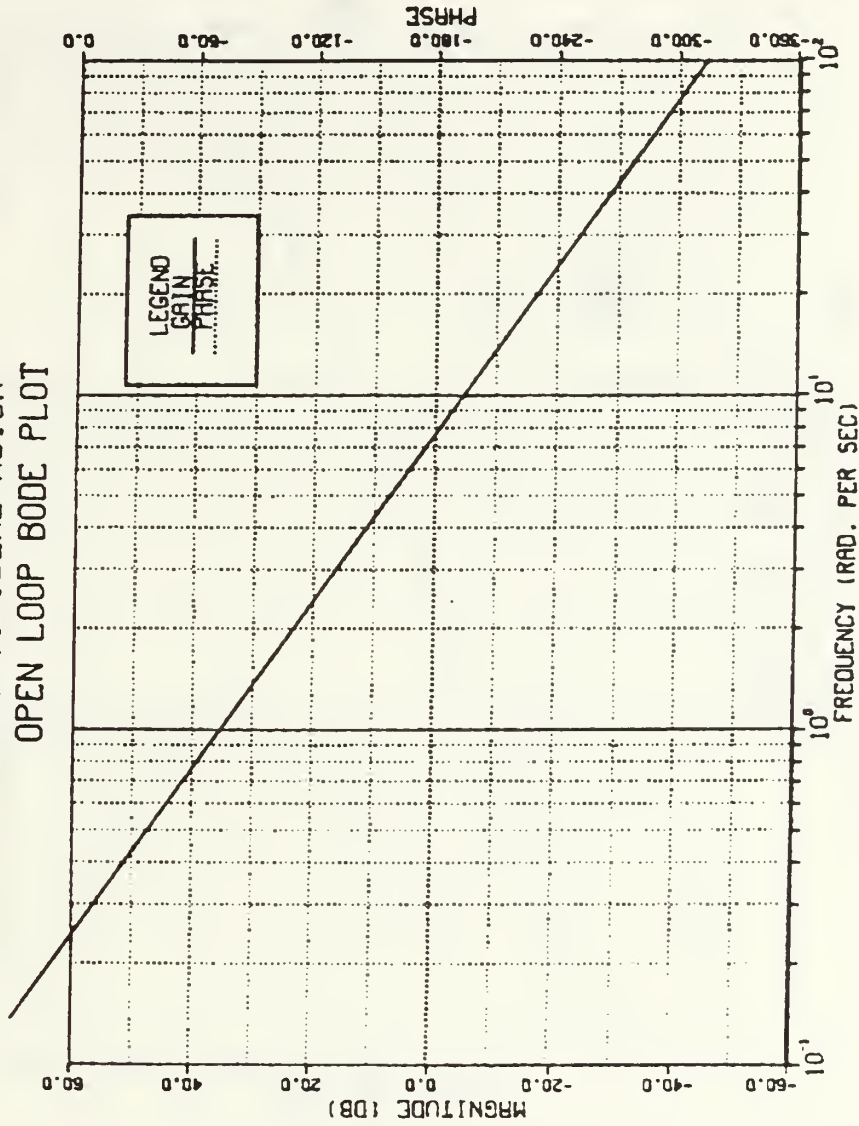
Figure 4.3   Open Loop Bode Plot of the K_{m1}/s^2 Motor
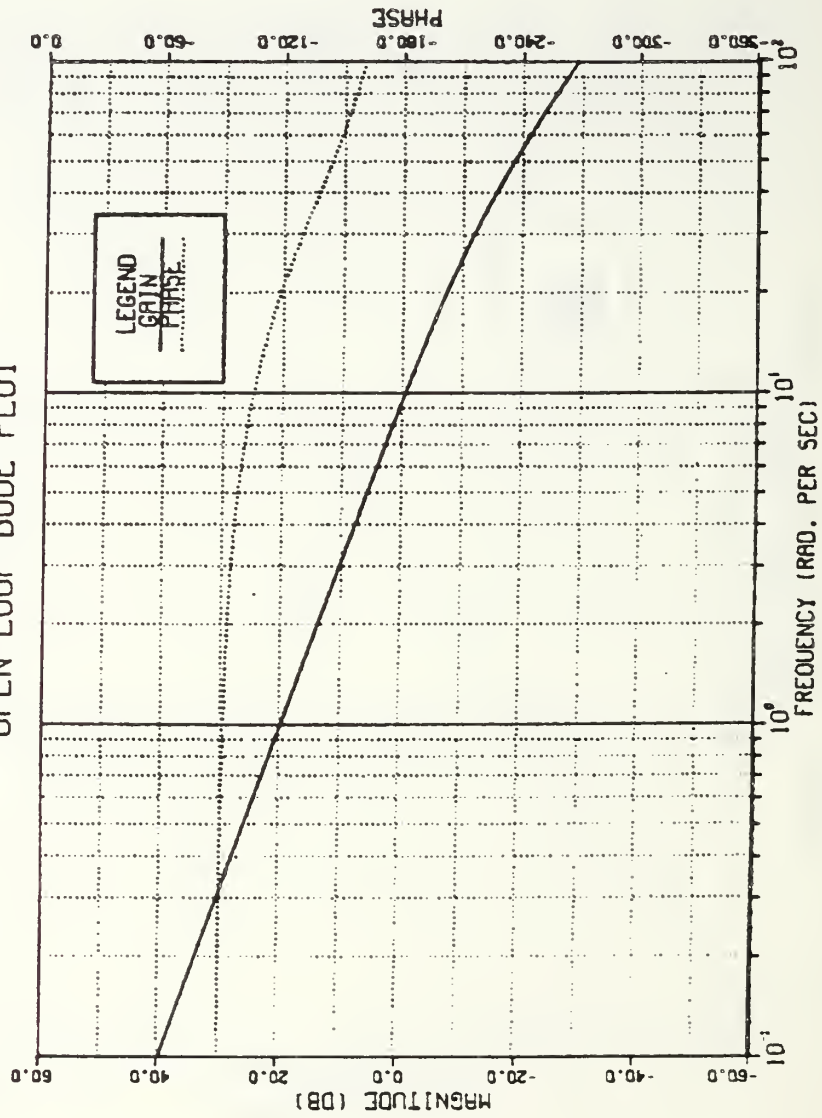
Figure 4.4   Open Loop Bode Plot of the JOINT2 Servo Motor

46
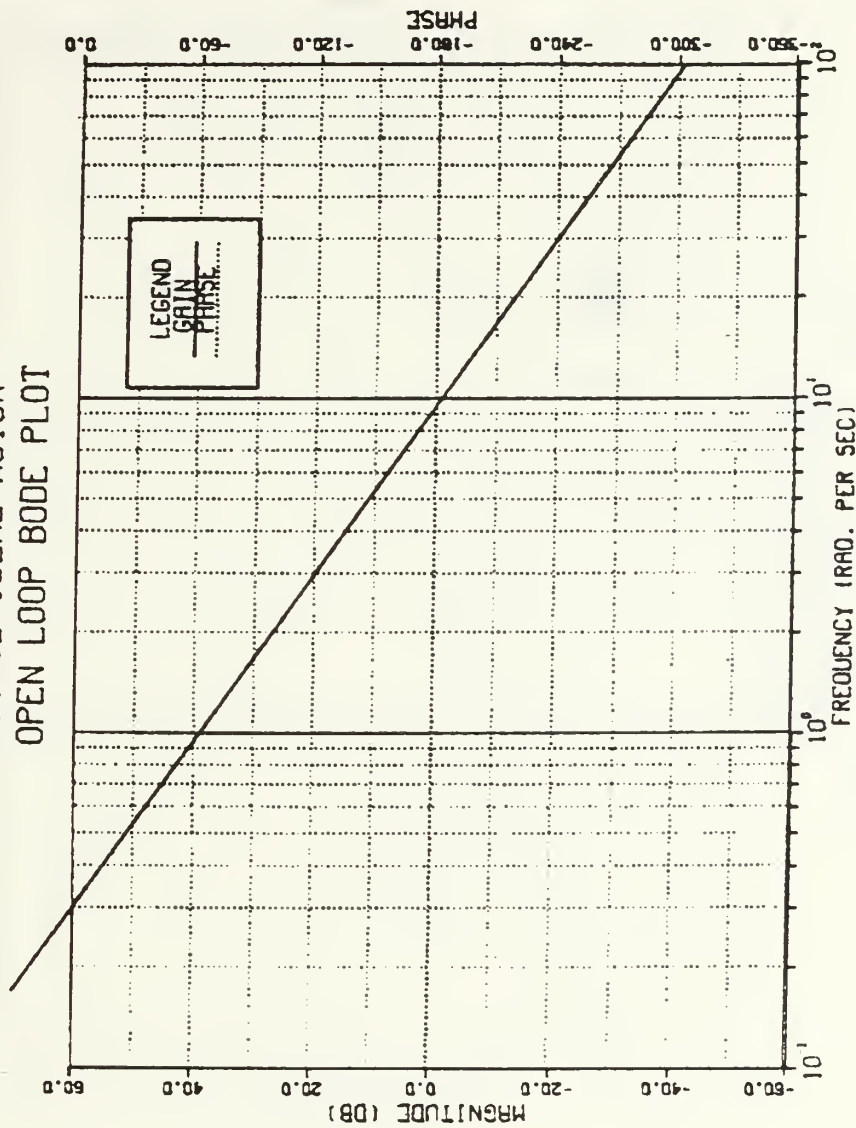
Figure 4.5   Open Loop Bode Plot of the K_{m2}/s^2 Motor

47
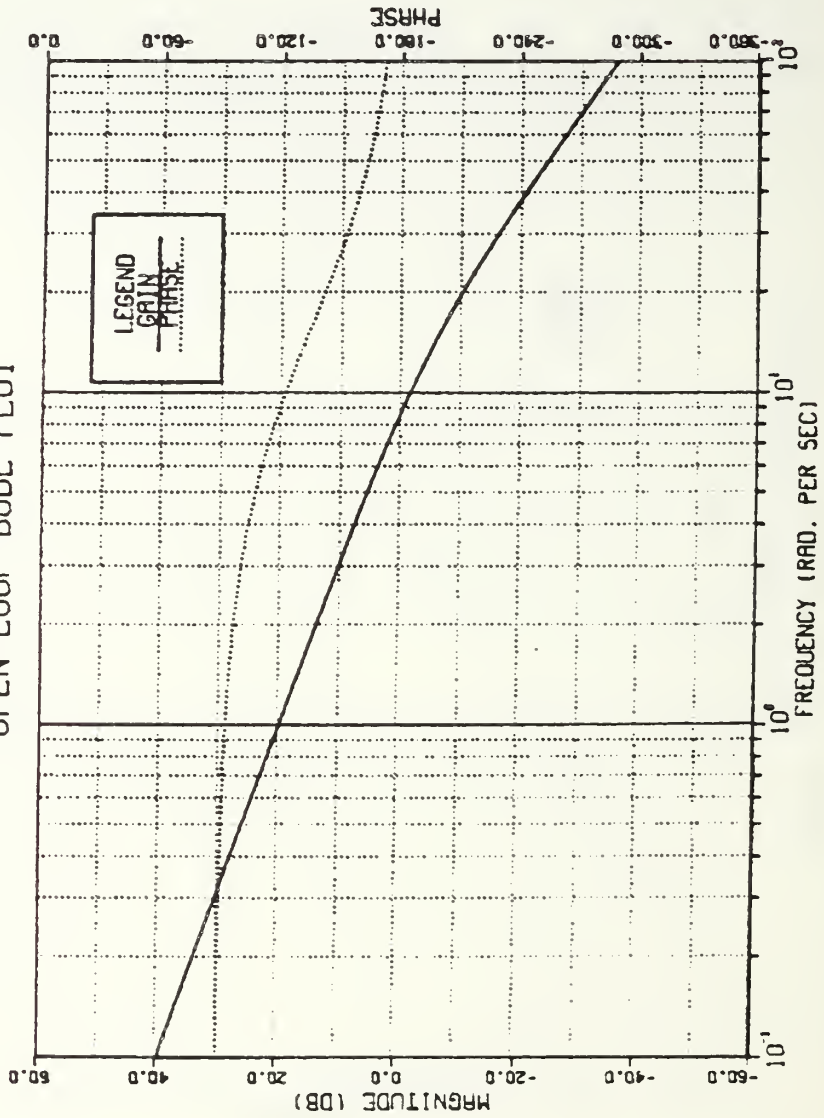
Figure 4.6   Open Loop Bode Plot of the JOINT3 Servo Motor

48
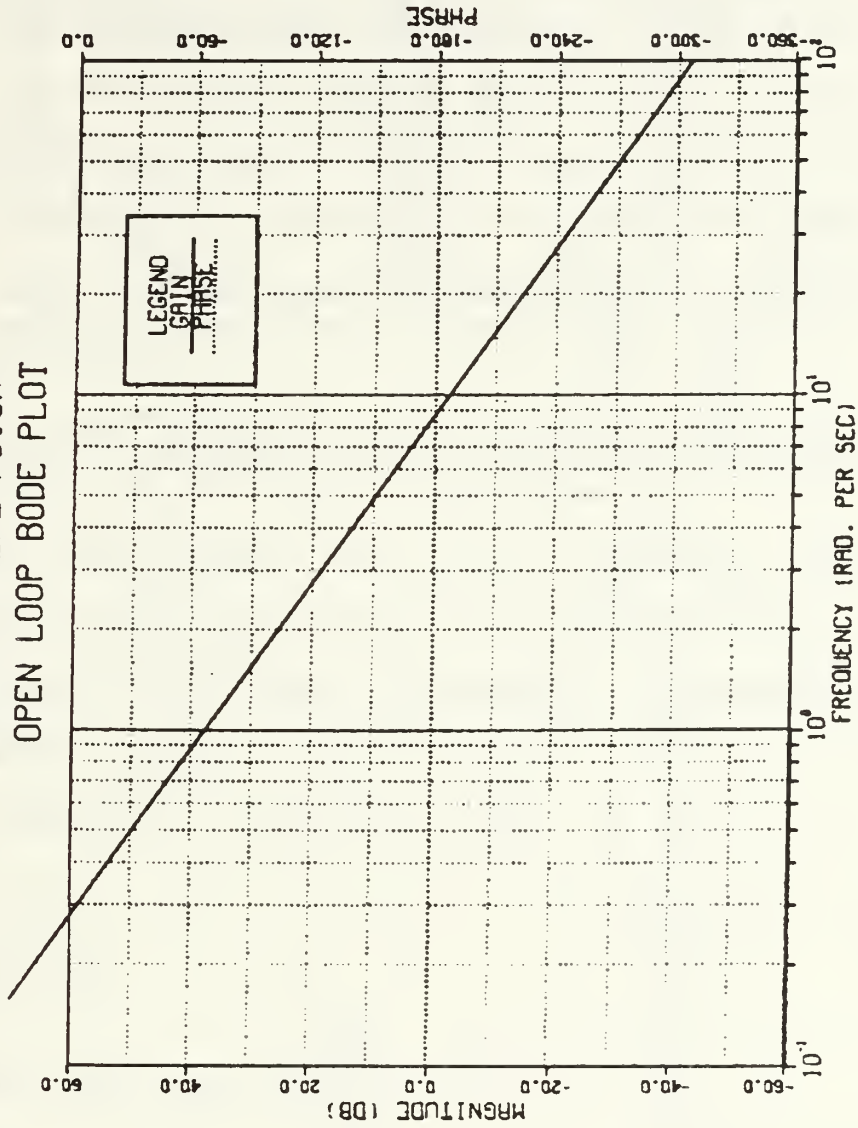
Figure 4.7   Open Loop Bode Plot of the K_{m3}/s^2 Motor
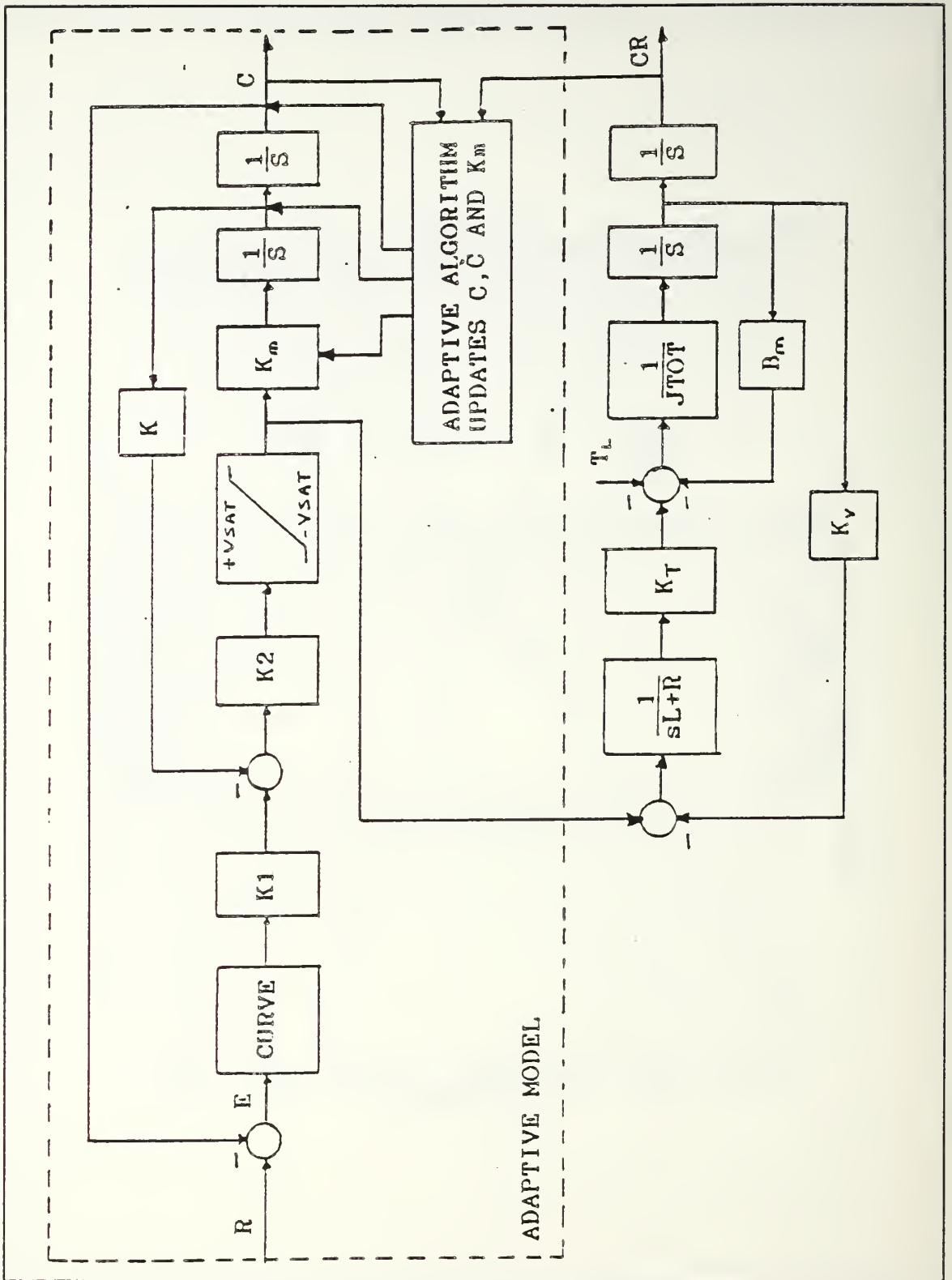
Figure 4.8   Block Diagram of the Adaptive Joint Drive System

Algorithm which updates the velocity (C) and position (C) of the model, and the gain parameter (K_m). K_m is not kept constant during the arm's motion but is adjusted in such a way as to account for the inertia reflected back from the arm. For the above reasons an estimate of the servo motor gain (K_m) as well as the linear velocity ($\dot{C}R$) of the servo motor must be determined. Two main considerations that must be taken into account in these calculations are: [Ref.1]

1. The calculations must be reasonably accurate to allow the model states to approximate the trajectory of the servo motor.

2. Since the updating of the model states and gain constant in minimum time is required, the calculations must be kept as simple as possible.

There are several different methods for these calculations. Since the method described in Ref. 1 best satisfies both of the above requirements, it is used for the calculations.

From equation 3.5 solving for K_m

$$K_m = \frac{2\ C}{V_{sat}\ t^2} \tag{4.7}$$

For discrete time intervals

$$K_m = \frac{2\ C}{V_{sat}\ (NT)^2} \tag{4.8}$$

where N is the number of samples and T is the sampling interval.

Letting C = CR

$$K_m = \frac{2 \ CR}{V_{sat} \ (NT)^2}$$

(4.9)

Equation 4.9 is valid only for the full acceleration portion of the move where the acceleration of the servo motor is constant. Therefore, the model gain K_m is updated until the velocity of the servo motor reaches the velocity curve following portion of the move.

The estimated velocity of the servo motor is [Ref.1]

$$\dot{CR}(N) = \frac{2[CR(N)-CR(N-1)]}{T} - \dot{CR}(N-1)$$

(4.10)

This calculation gives an estimate of the velocity as soon as the last position of the servo motor is known, and requires the storage of the last position measurement [CR(N-1)] and the last estimated velocity [$\dot{CR}$(N-1)]. After 2 samples of position are known, the stored velocity $\dot{CR}$(N-1) can be recalculated as

$$\dot{CR}(N-1) = \frac{CR(N) - CR(N-2)}{2T}$$

(4.11)

If the model switches from full acceleration to full deceleration (curve following mode) between samples of position, the velocity given by equation 4.10 is not self-correcting until two position samples can be obtained after

52

switching. To remove this discrepancy, the algorithm detects the switching time and stores the present value of the model velocity as CR(N-1) to be used in the next calculation.

As has been discussed earlier in this chapter, generally CR does not duplicate C, because of the different dynamics between model and real motor. However, if C and CR start with the same values, and if they have identical initial derivatives $\dot{C}$ and $\dot{CR}$, then it is expected that CR and CR will not deviate substantially from C and $\dot{C}$ over a very short time interval. If C and $\dot{C}$ are reset at the end of this interval and given the actual values of CR and $\dot{CR}$, then both model and motor start the next interval with the same states. Also the gain parameter, K_m, is updated during the full acceleration portion of the move. [Ref.8]

In the simulation of this model fixed step integration is used as the easiest numerical integration method. The integration step size parameter, DELT, is chosen to be 1/5 of the sampling interval so that 5 integration steps are completed between two successive updates of the adaptive model. The model velocity error signal, XE, is computed at each integration step and is sent to the amplifier.

V. MODELLING THE THREE LINK RECTANGULAR ROBOT

A. INTRODUCTION

In this chapter the dynamic equations of the three link rectangular robot are first developed and then solved in order to obtain the equations of motion. Although using Lagrangian mechanics the dynamic equations of any system can be obtained in a relatively simple manner, the Newton-Euler formulation of equations of motion is more convenient in the case of the cartesian coordinate configuration and for this reason is preferable over the Lagrangian method.

Since the motion of each link is independent of the motion of the other two links, there is no inertial coupling between the three joints. Besides, the only link affected by the gravitational acceleration is the vertical one while the other two links move in directions perpendicular to the direction of gravity and therefore are not affected at all.

B. MODEL DEVELOPMENT

The three link rectangular robot to be tested is illustrated in Figure 2.2. The rotary motion of the three joint servo motors is converted, through rack and pinion gearing of radius r, to linear motion, and the motions of the three links are constrained to the x, y and z directions respectively. The link moving along the y axis has a

gripper, as end effector, which can carry a load. The mass of the servo motors is denoted m_m and the masses of the three links m_1, m_2, m_3 respectively. The total masses seen by each servo motor are denoted M_1, M_2, and M_3. F_1, F_2, F_3 represent the forces acting on the three links in the directions of motion, and T_1, T_2, T_3 represent the external driving torques to the joints.

All joints can move simultaneously, so the end-of-arm follows a path determined by the relative speeds of each joint velocity. However, prediction of this path is difficult because varying arm loads or varying friction change the joint velocities in a random way.

C. EQUATIONS OF MOTION

Applying the Newton-Euler method, the motion of a rigid body can be decomposed into the translational motion of an arbitrary point fixed to the rigid body, and the rotational motion of the rigid body about that point. The dynamic equations of a rigid body can also be represented by two equations: one which describes the translational motion of the centroid, while the other describes the rotational motion about the centroid. The former is Newton's equation of motion for a mass particle, and the latter is called Euler's equation of motion.

In the specific case of rectangular coordinate robot there is no rotational motion, therefore the dynamic

equations are obtained applying only Newton's equation. If v_{ci} is the linear velocity of the centroid of the link i, then the inertial force is given by $-M_i\dot{v}_{ci}$, where M_i is the total mass seen by the servo motor i. Then, the equation of motion for the link i is obtained by adding the inertial force to the static balance of forces so that

$$F_{i-1,i} - F_{i,i+1} + M_ig - M_i\dot{v}_{ci} = 0 , \quad \text{for } i=1,2,3 \qquad (5.1)$$

where $F_{i-1,i}$ and $F_{i,i+1}$ are the coupling forces applied to link i by links i-1 and i+1, respectively, and g is the acceleration of gravity.

In the case of rectangular coordinate robot due to absence of any inertial coupling between the joints, equation 5.1 becomes

$$F_i + M_ig - M_i\dot{v}_{ci} = 0 , \quad \text{for } i=1,2,3 \qquad (5.2)$$

where F_i is the force applied to the link i by the joint motor i, in the direction of motion of the link i. The total torque about the axis of rotation of the motor i is given then by the equation

$$T_i = F_ir + J_{mi}\ddot{\theta}_i \qquad (5.3)$$

where r is the radius of the pinion gearing, J_{mi} is the inertia of motor i, and $\ddot{\theta}_i$ is the angular acceleration of the motor's i shaft.

Solving equations 5.2 and 5.3 for the three links, the following second-order nonlinear differential equations have been derived and used as the basic mathematical model for the simulation of the three link rectangular robot. The derivation of these equations is shown in details in Appendix B.

$$T_1 = (M_1 r^2 + J_{m1})\ddot{\theta}_1 \tag{5.4}$$

$$T_2 = (M_2 r^2 + J_{m2})\ddot{\theta}_2 \tag{5.5}$$

$$T_3 = (M_3 r^2 + J_{m3})\ddot{\theta}_3 - M_3 gr \tag{5.6}$$

where

$$M_1 = m_1 + m_2 + m_3 + 2m_m + \text{Load}$$

$$M_2 = m_2 + \text{Load}$$

$$M_3 = m_2 + m_3 + m_m + \text{Load}$$

$$m_1 = 0.082 \ oz/in/sec^2$$

$$m_2 = 0.041 \ oz/in/sec^2$$

$$m_3 = 0.041 \ oz/in/sec^2$$

$$m_m = 0.186 \ oz/in/sec^2$$

$$r = 0.5 \ in$$

$$g = 386.4 \ in/sec^2$$

In these equations, terms of the form $M_i r^2 + J_{mi}$ are known as the effective inertia at joint i.

VI. SIMULATION OF THE THREE LINK RECTANGULAR ROBOT

A. PLANNING THE SIMULATION STUDIES

The already developed model for the three link robot is tested for the Voltage Source Drive. First a gravity-free environment is assumed and the model is tested for different load conditions, for rejection of a disturbance applied before and after steady state condition, and finally for robustness in the case of motor parameters variation. Then the same tests are repeated but in this case gravitational torques are taken into account.

The tested arm is a point-to-point arm and the unit step input command is applied in all joints simultaneously for comparison reasons. The motions of the three joints are independent of each other, and the path that the end of the gripper follows depends on the relative speeds of each joint movement. In this thesis a free work space is assumed for the tested arm so that obstacle avoidance is not a consideration.

For every simulation program three plots are obtained:

1. Phase plane

2. Step response

3. Error versus time

In all phase plane plots the model linear velocity, C, and the actual system linear velocity, CR, are plotted as

58

ordinates versus the system linear position, CR. In step response plots the linear positions of the model, C, and the actual system, CR, are plotted as functions of time. Although the linear position and velocity of the actual system can not be observed in the above plots, they are available in simulations and are used to check the validity of the adaptive algorithm. Finally in the last group of plots the error between the commanded and the actual position of the system is plotted also as a function of time.

B. SIMULATION STUDIES OF THE ADAPTIVE SYSTEM

1. Gravity-free Environment

The adaptive system in a gravity-free environment is tested first for different load conditions. Then the capability of the system to reject a disturbance, which is applied in any time instant, is tested and finally the behavior of the system is examined for a slight variation of servo parameters.

a. Different Load Conditions

The adaptive system is tested first without carrying any load and then with different load. The DSL/VS simulation program used for these tests is listed in Appendix C. Figures 6.1, 6.2 and 6.3 are the phase plane plot, the step response and the error curve, respectively, for the unloaded arm.
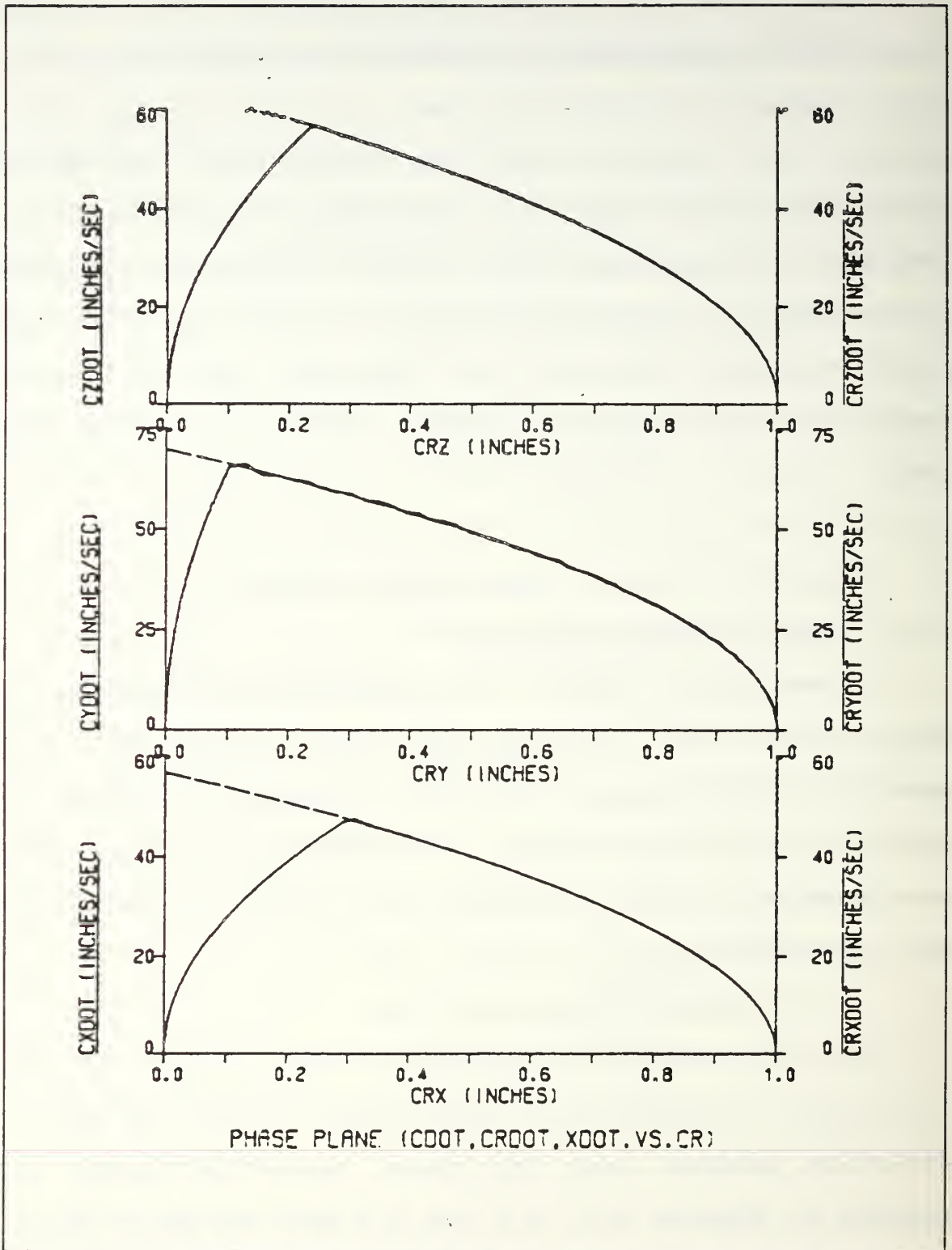
Figure 6.1 Phase Plane Trajectory for Unloaded Arm
(No Gravity)
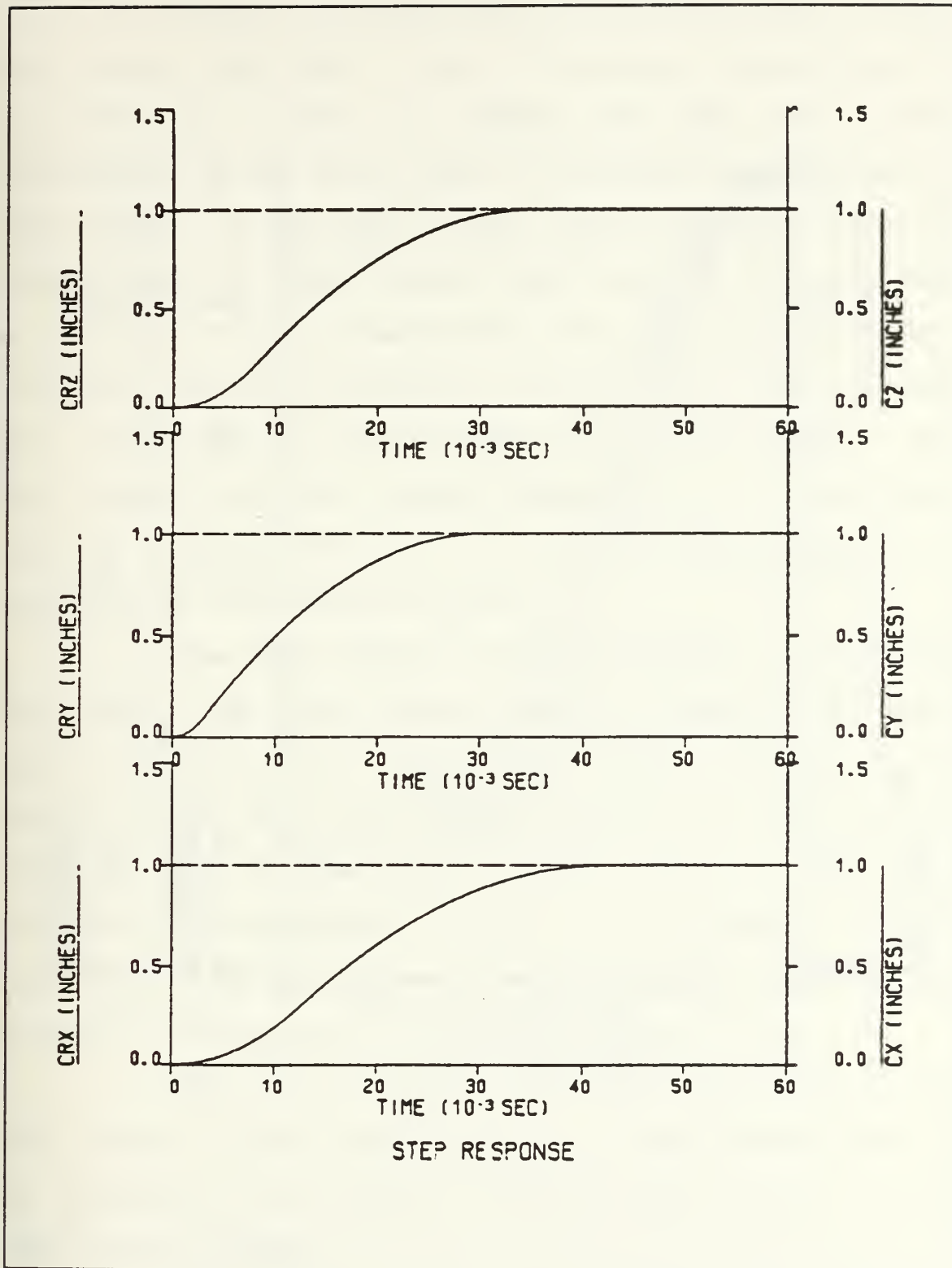
Figure 6.2   Step Response for Unloaded Arm
(No Gravity)

61
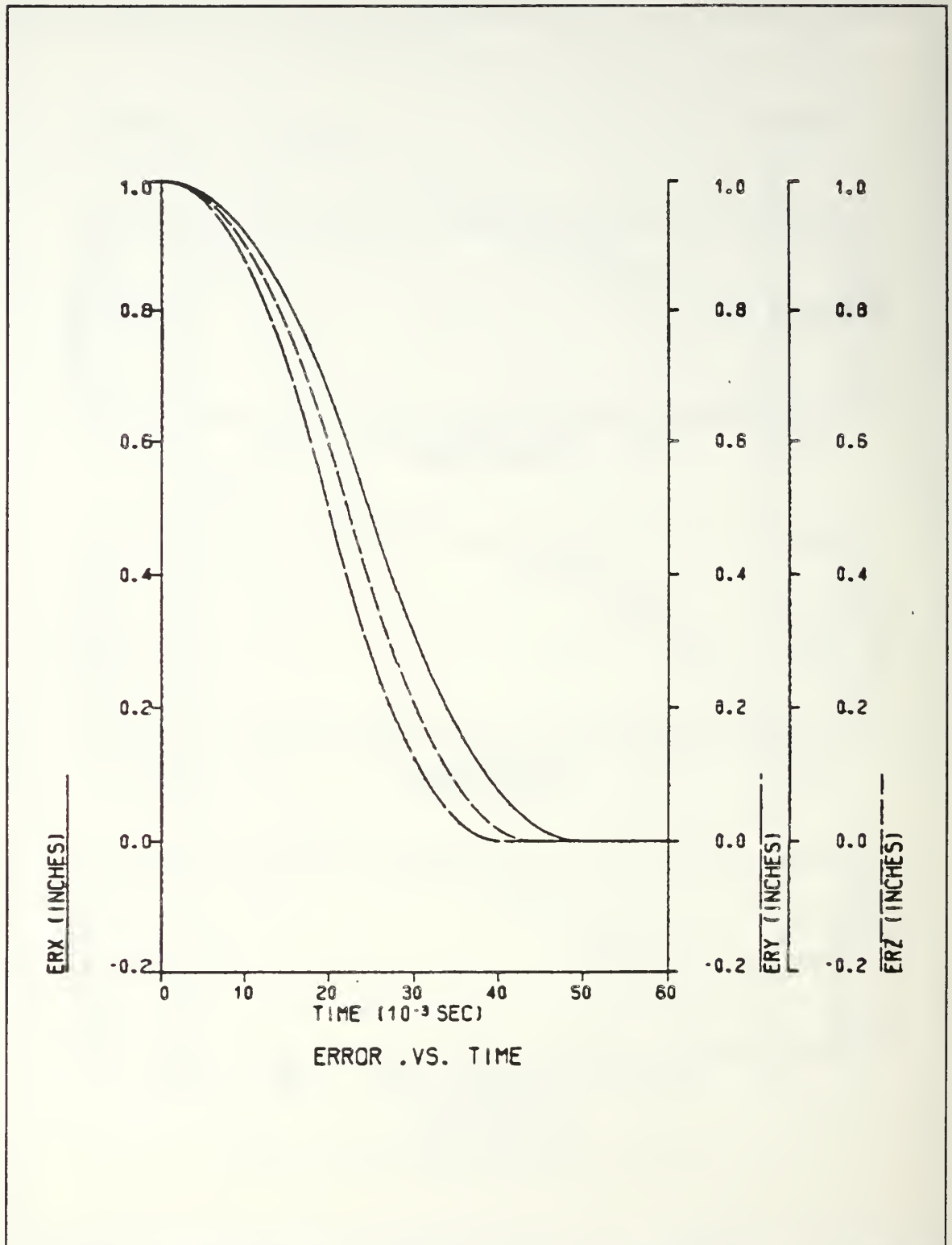
Figure 6.3   Error Curves for Unloaded Arm
(No Gravity)

Figure 6.1 shows that the adaptive model and the servo motor have good curve following characteristics for the three joints. It shows also that the linear acceleration of the three links is inversely proportional to their masses. So the link moving along the y-axis, which is the smallest one, moves faster than the other two links and is the first one that reaches the commanded velocity curve, while the link moving along the x-axis, which is the largest one, is the last one that reaches the velocity curve. The link moving along the z-axis moves with an intermediate velocity. Finally all links track the commanded velocity curve down to the desired position.

The step response curves of Figure 6.2 show the three model and servo motors tracking together to steady state condition. The y-axis link reaches the commanded position faster while the x-axis link, moving slower, is the last one that reaches the desired position. The time required for the movement of the arm to be completed is the time required by the slowest (x-axis) link to complete its movement. From Figure 6.2 this time is read 42 msec.

The error curves of Figure 6.3 show zero steady state error for all links, but an accurate observation of the simulation data reveals a steady state accuracy of the order of 10^{-5} inches.

The loaded arm under different load conditions is then simulated in Figures 6.4 through 6.12. In these
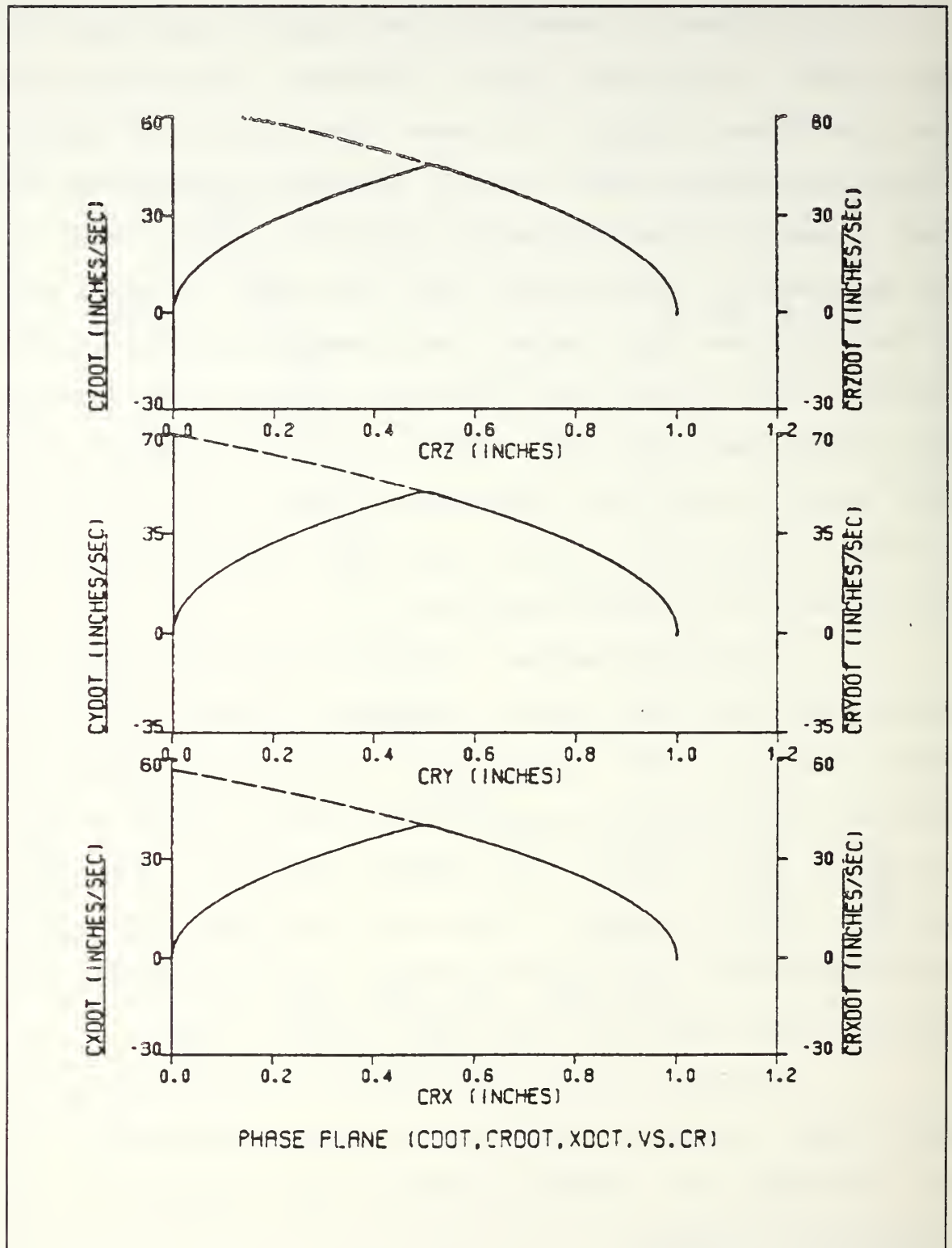
Figure 6.4   Phase Plane Trajectory for Loaded Arm
(Load=0.83 - No Gravity)
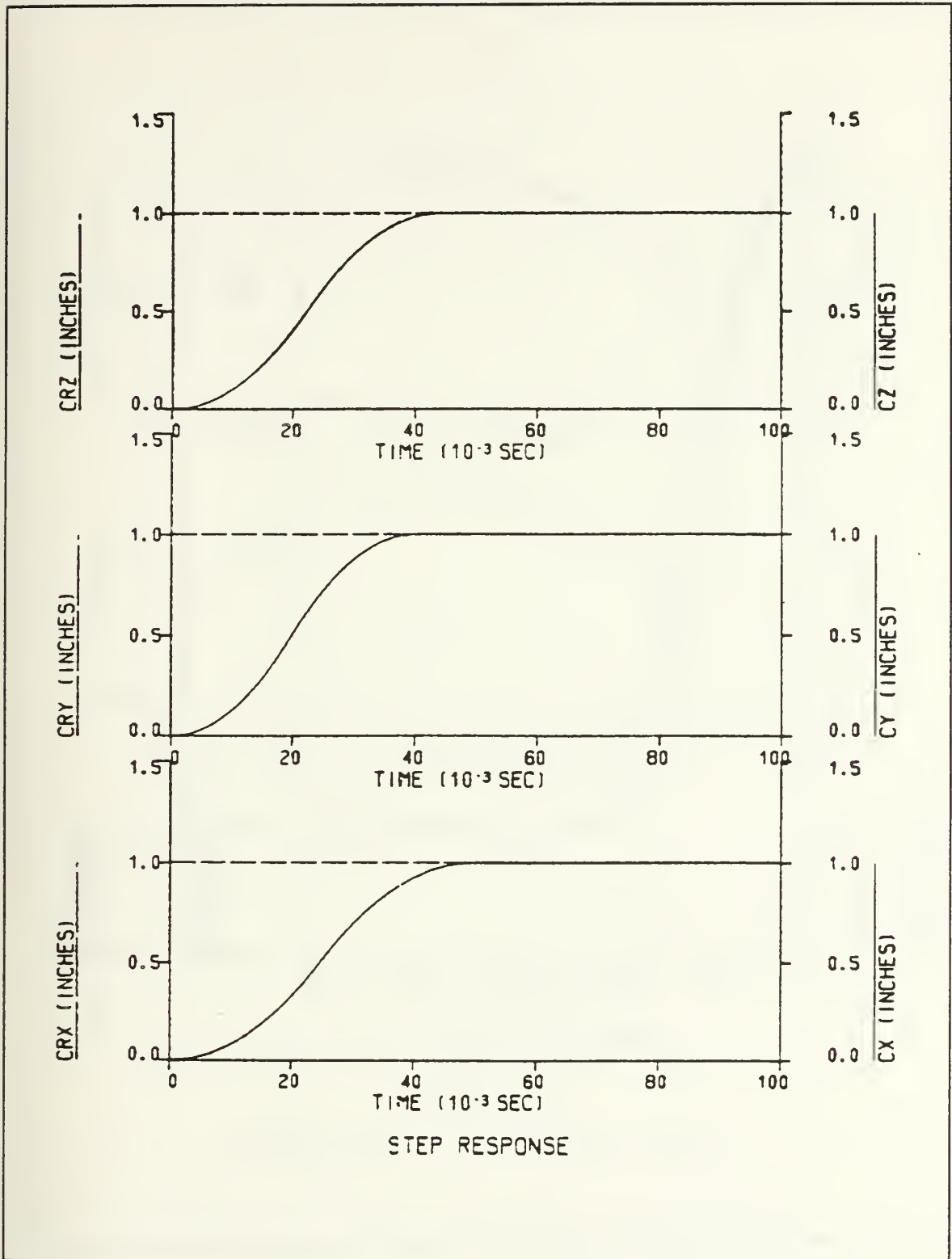
Figure 6.5   Step Response for Loaded Arm
(Load=0.83 - No Gravity)
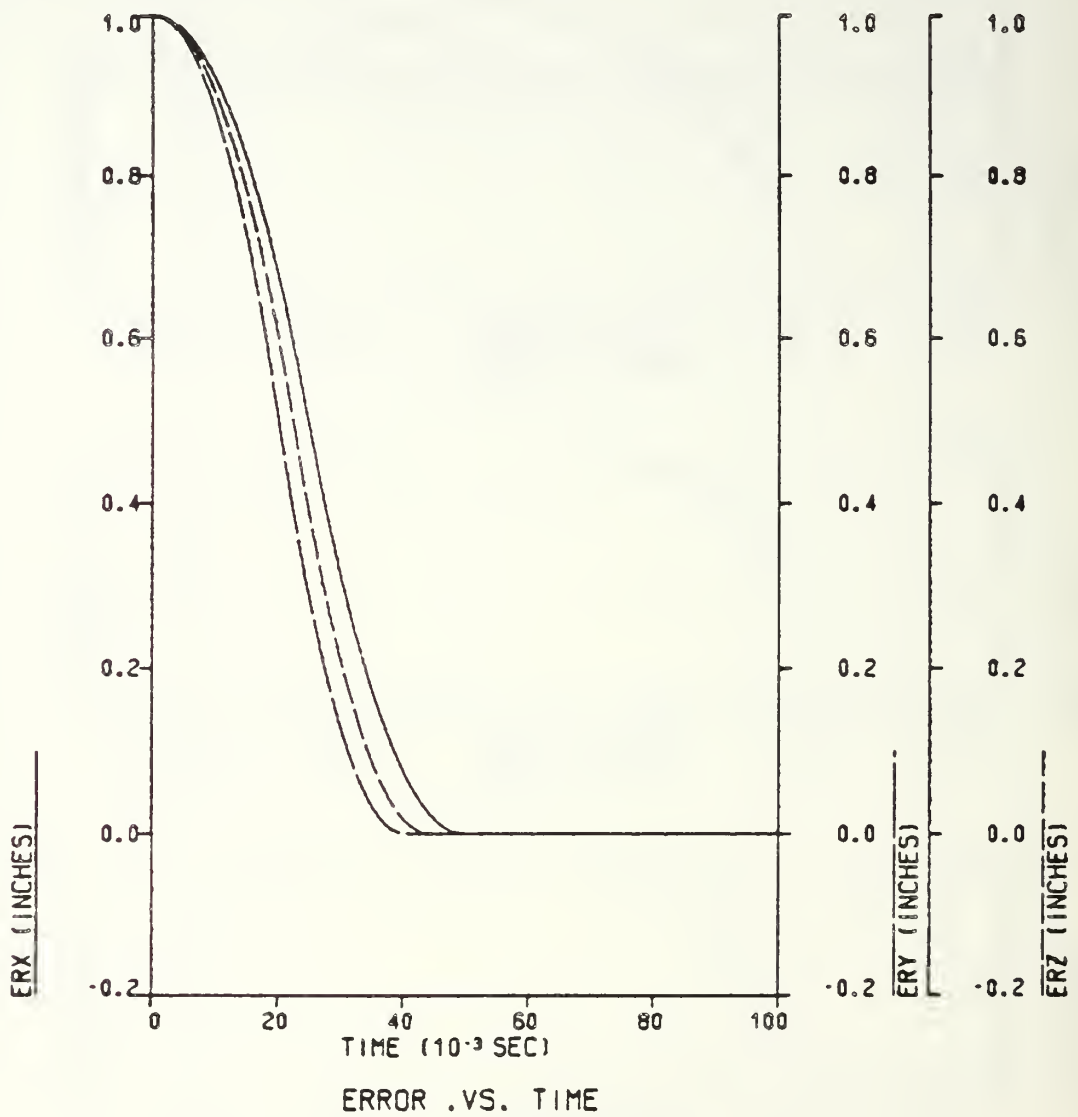
Figure 6.6   Error Curves for Loaded Arm
(Load=0.83 - No Gravity)

66
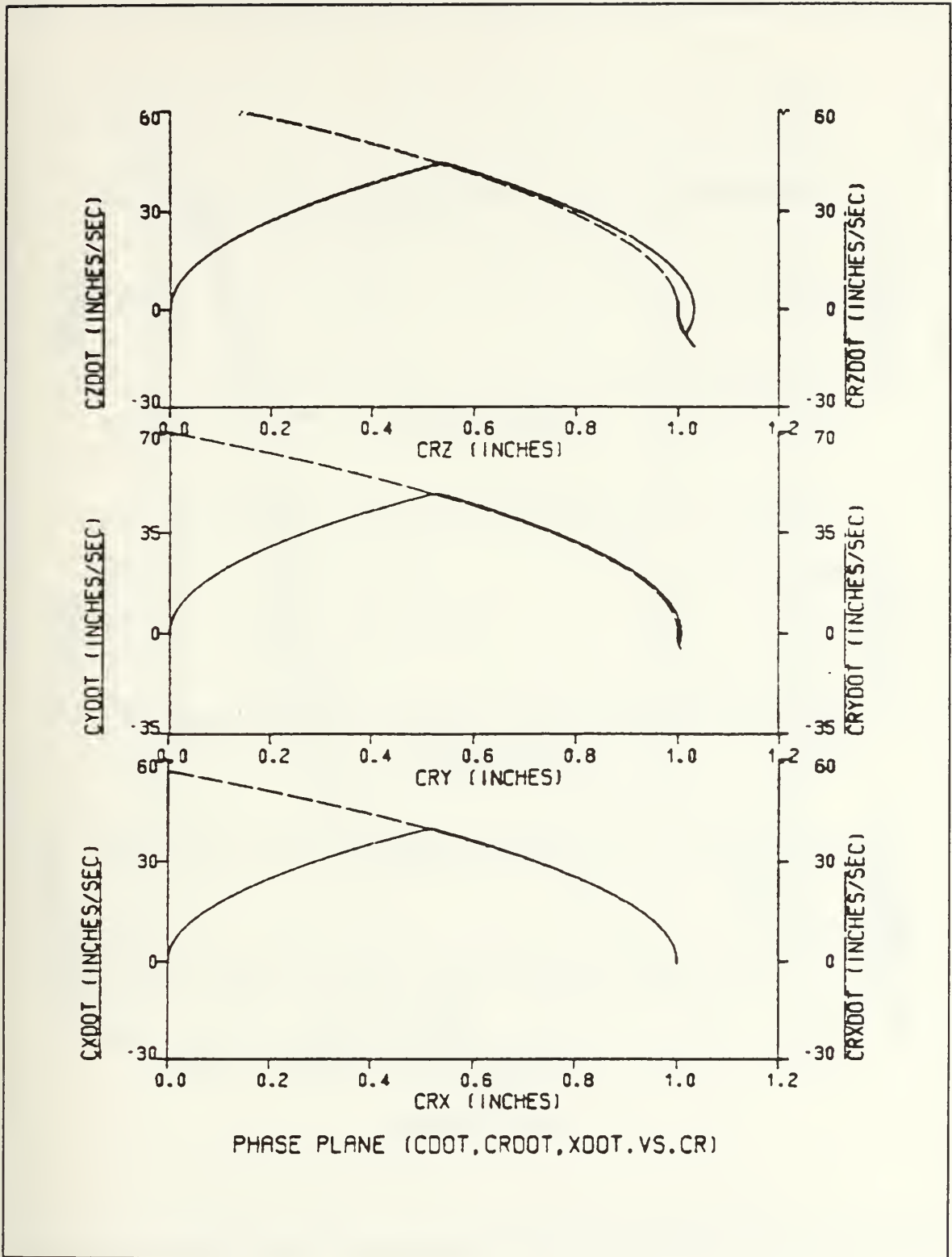
Figure 6.7   Phase Plane Trajectory for Loaded Arm
(Load=0.9 - No Gravity)
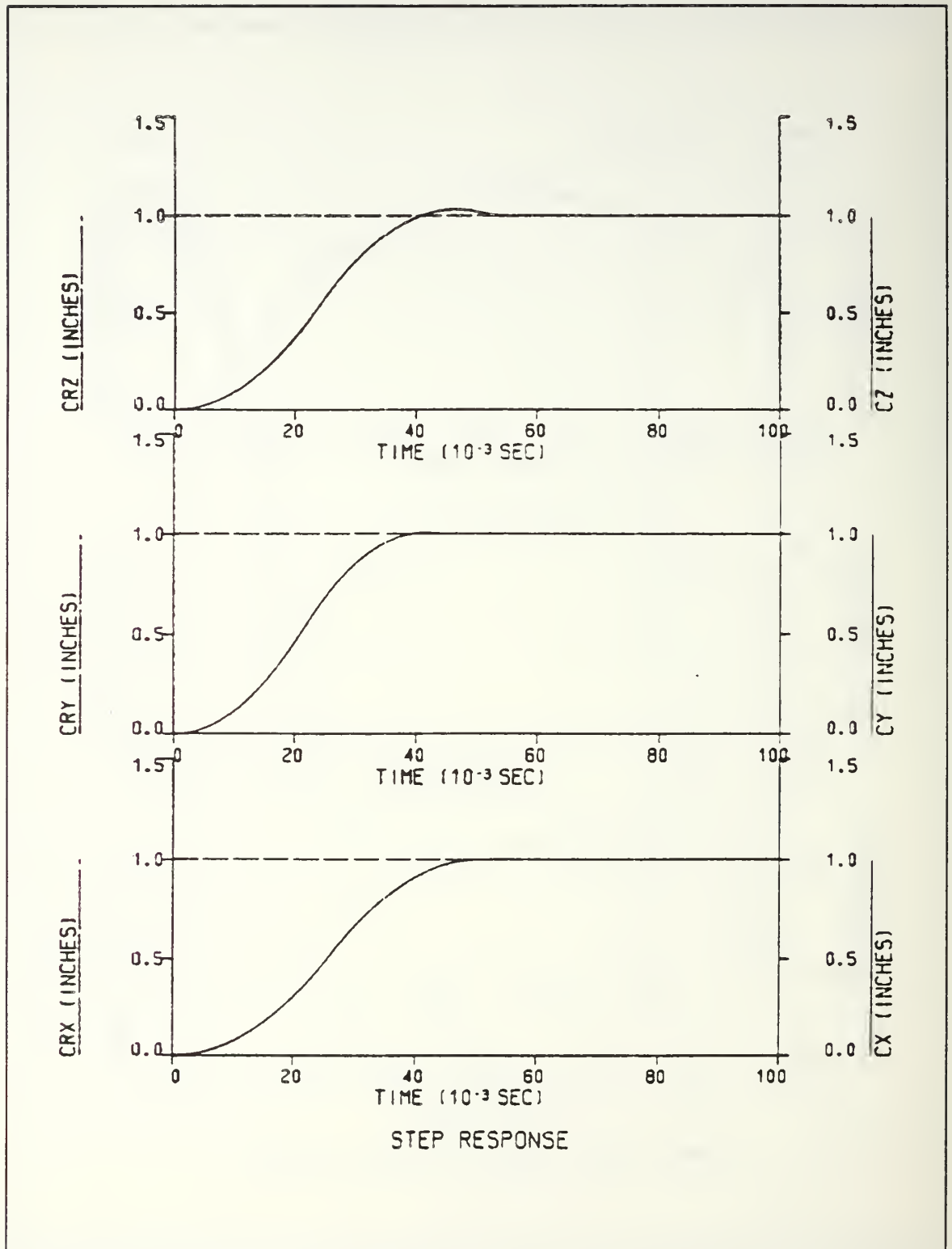
Figure 6.8 Step Response for Loaded Arm
(Load=0.9 - No Gravity)
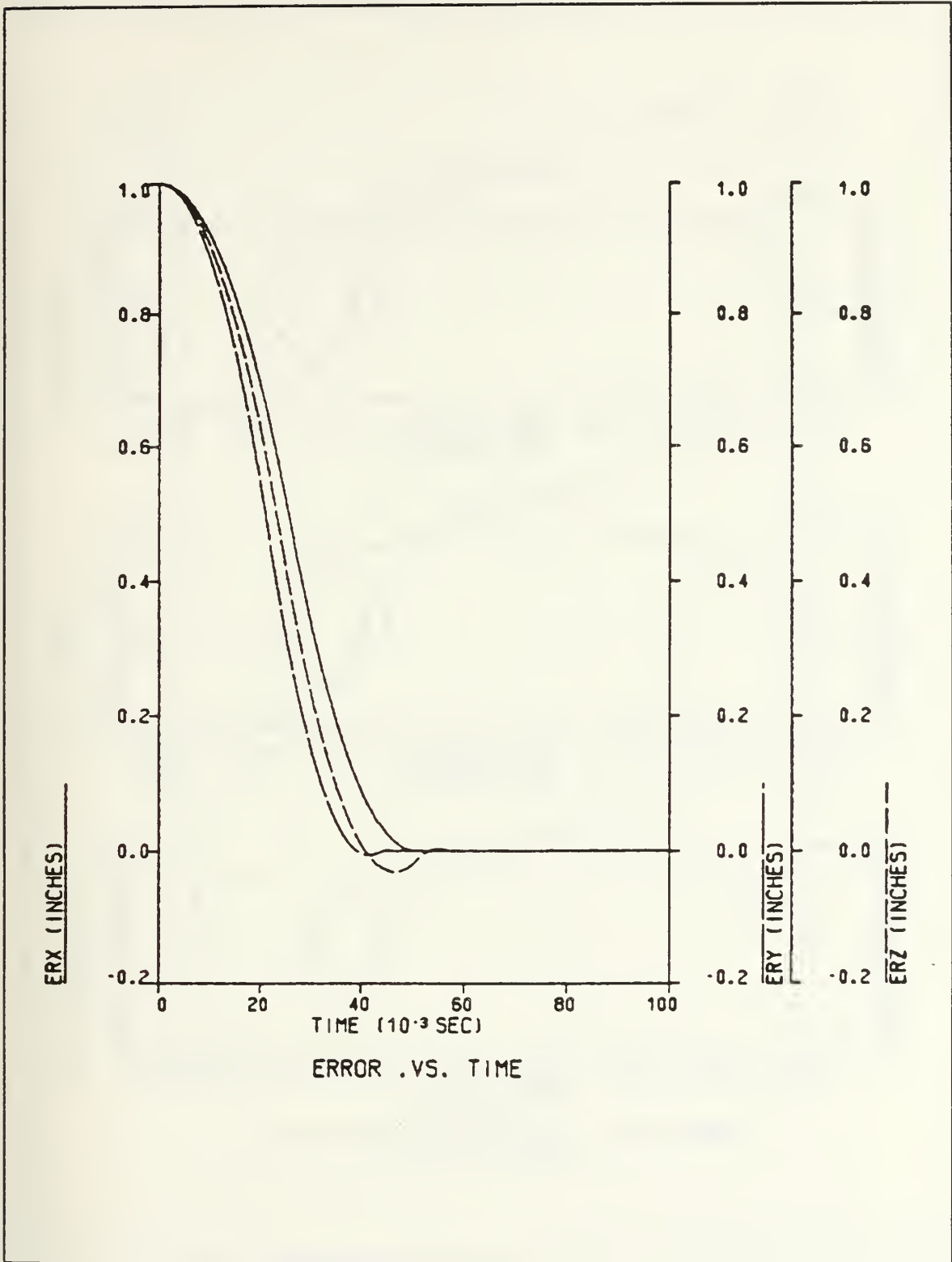
Figure 6.9   Error Curves for Loaded Arm
(Load=0.9 - Gravity)

69
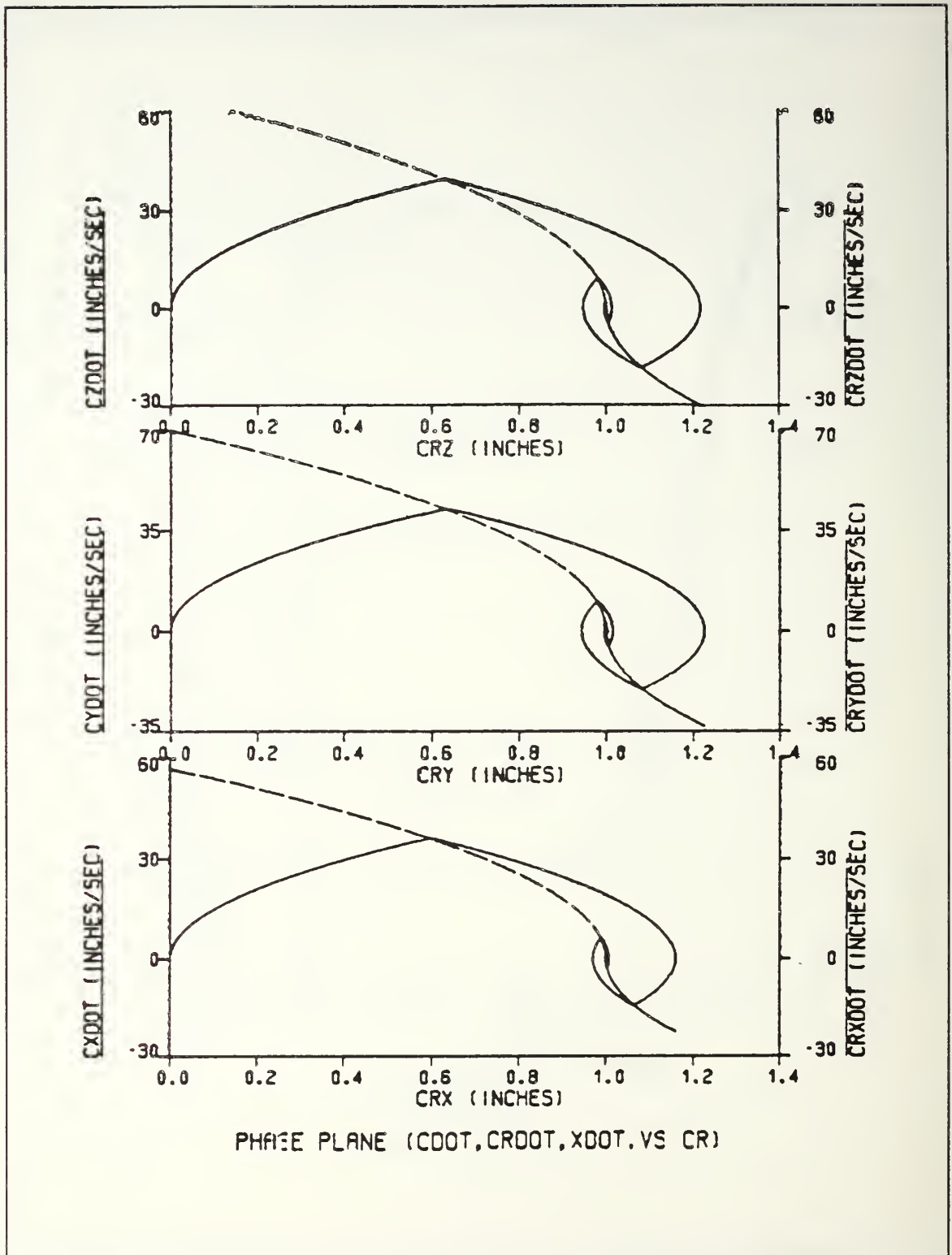
Figure 6.10   Phase Plane Trajectory for Loaded Arm
(Load=1.5 - No Gravity)

70
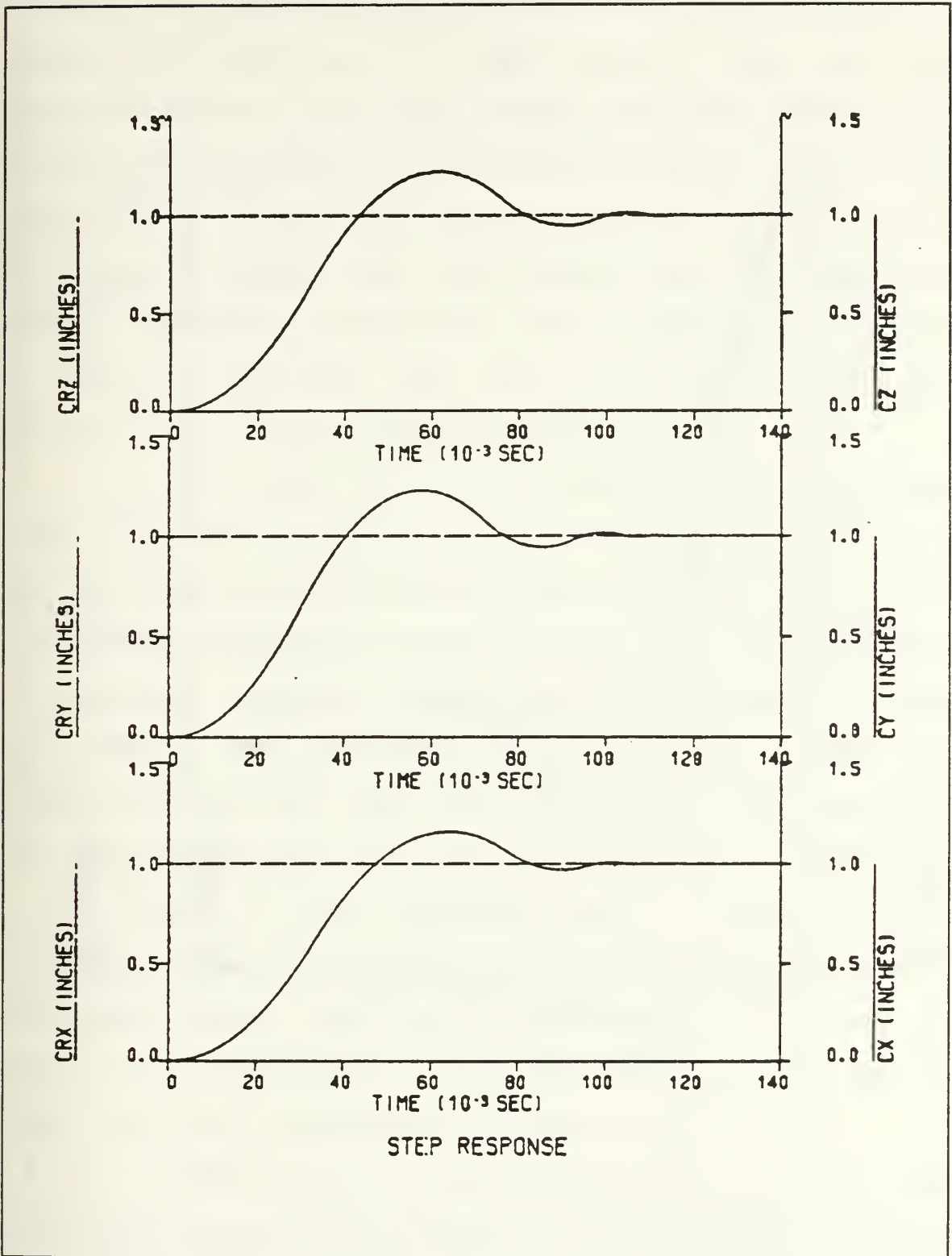
Figure 6.11   Step Response for Loaded Arm
(Load=1.5 - No Gravity)

71
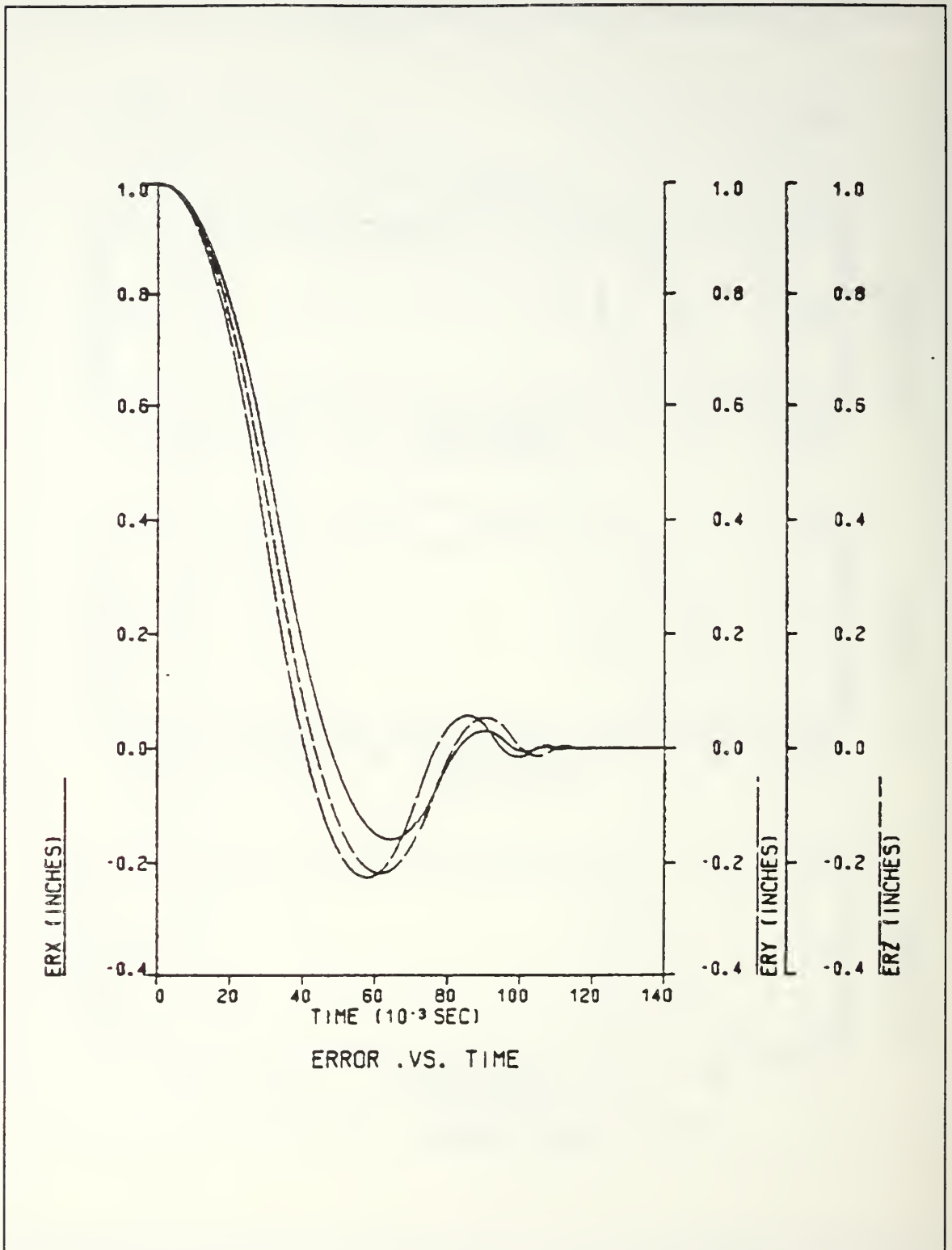
Figure 6.12   Error Curves for Loaded Arm
(Load=1.5 - No Gravity)

72

figures the effects of the added load can be observed. Although the same load is added to all links, the link mostly affected by this load is the y-axis link because this is the lightest one and its relative increment of the weight (the ratio of added load to link weight) is larger than in the other two links. Thus, the settling time for the y-axis link is increased more than the other links and as more load is added to the arm, the time difference between the settling time of the three links tends to zero.

For loads up to 0.83 oz/in/sec^2, the critical load, the system shows good curve following characteristics for all joints as depicted in Figures 6.4 - 6.6. But as the load exceeds the value of the critical load, the velocities of the three links are reduced and they can not track the deceleration curves. Therefore, when they reach the commanded positions, their velocities do not reach zero but they still have some values which are inversely proportional to the weights of the respective links as shown in Figure 6.10. As a result of this fact the three links overshoot and undershoot before they reach steady-state, as shown in Figure 6.11. Consequently, the error curves oscillate before they reach zero, as depicted in Figure 6.12.

Then, with a slight modification of the same simulation program, the unloaded arm is commanded to go to the desired position, take a load and return to the initial location. The results are shown in Figures 6.13 - 6.15.
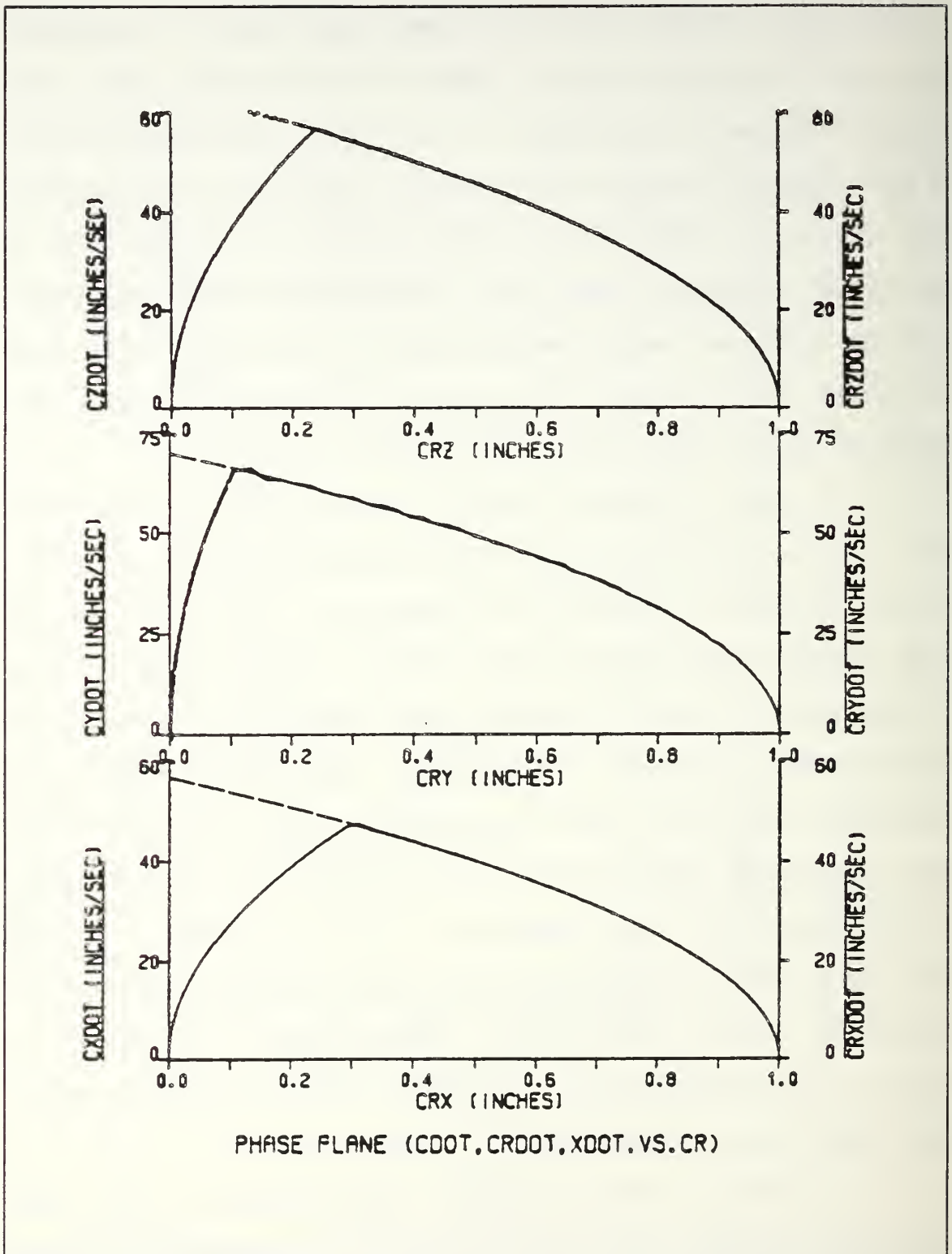
Figure 6.13   Phase Plane Trajectory for Load Pick-up
and Return (Load=0.3 - No Gravity)

74
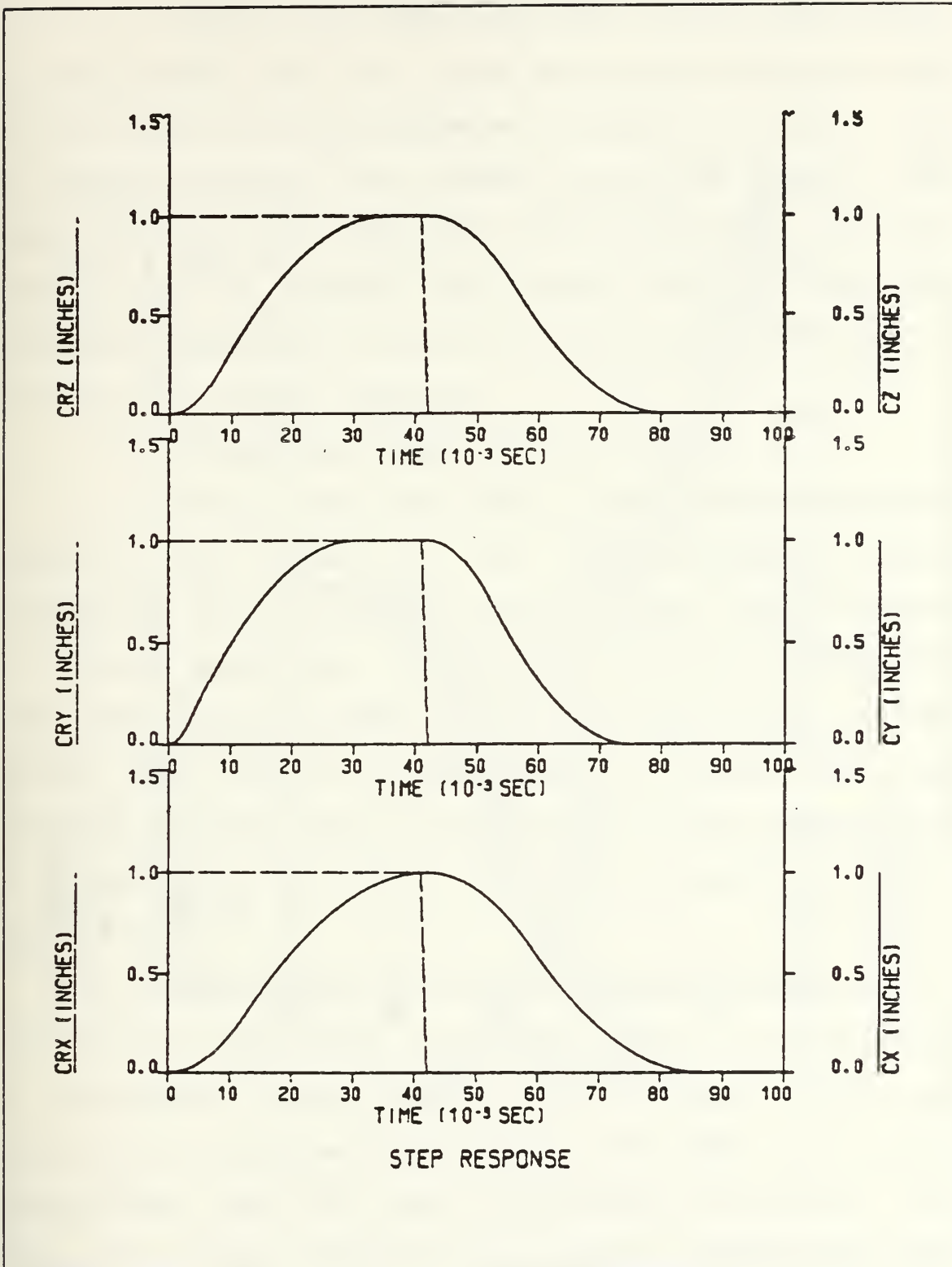
Figure 6.14   Time Response for Load Pick-up
and Return (Load=0.3 - No Gravity)

75
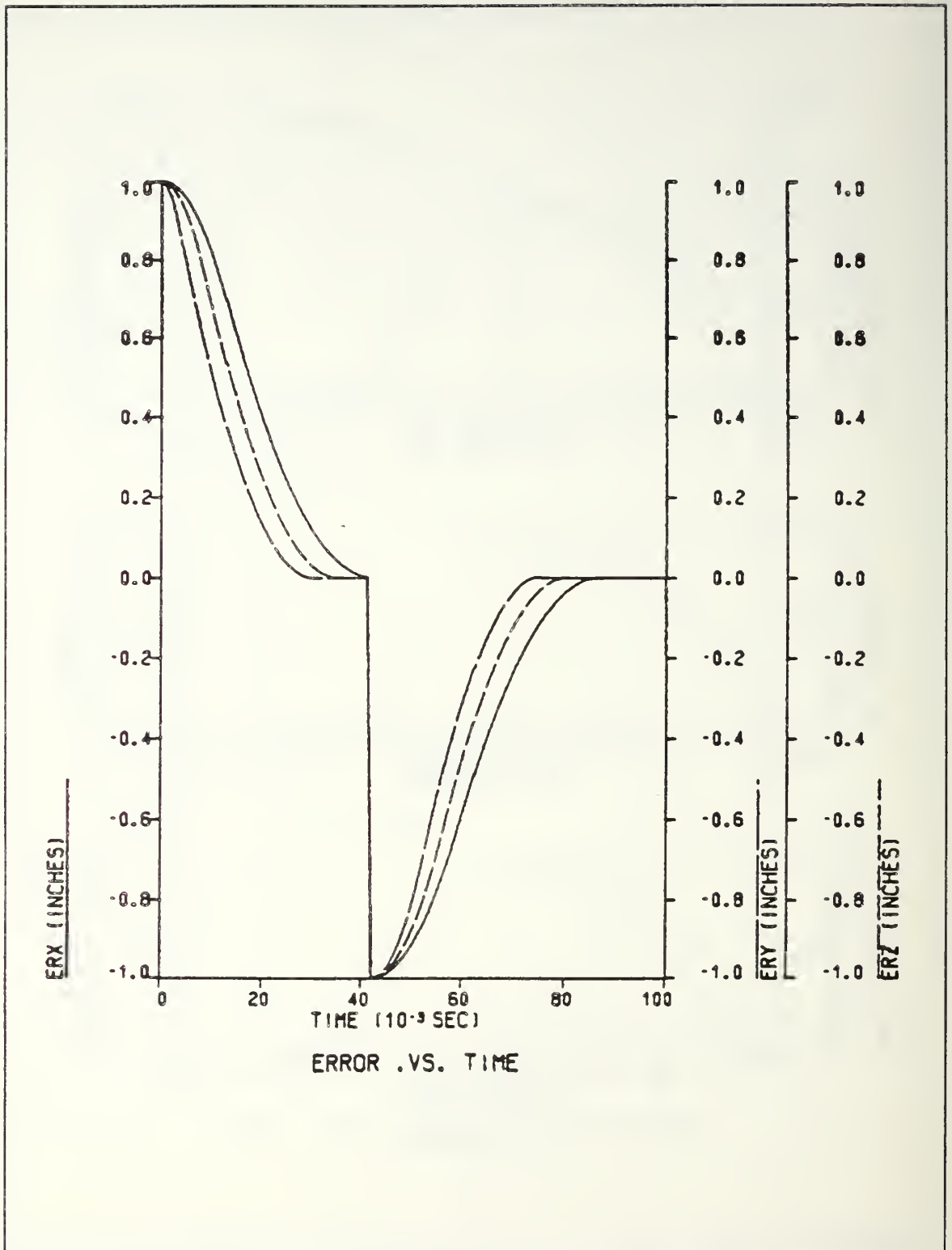
Figure 6.15   Error Curves for Load Pick-up
and Return (Load=0.3 - No Gravity)

76

Figure 6.13 shows that the second-order model and servo motor have good curve following characteristics for all joints. The step response of Figure 6.14 shows that the fastest (y-axis) link reaches the desired position and waits there till the other two links reach this position. When the slowest (x-axis) link reaches this position, the load is picked up by the robot and the three links start to move to their initial locations.

b. Disturbance Rejection

It is not rare that disturbances, although undesired, are present in the robot operation. These disturbances can be treated as random impulse inputs applied at any time during the robot motion. They can be rejected from a well designed robot, or they can cause failure to the desired robot motion if the design is poor. Next the capability of the system to reject these random disturbances is tested using the DSL/VS simulation program listed in Appendix D.

An impulse of 2 msec duration is selected to represent the disturbance and it is applied to three links in different times during their motion. First the disturbance is applied before any of the links has reached steady-state, then the same disturbance is applied just before the first (y-axis) link reaches steady-state and finally after all links have reached steady-state. The results of the three cases are shown in Figures 6.16 - 6.21.
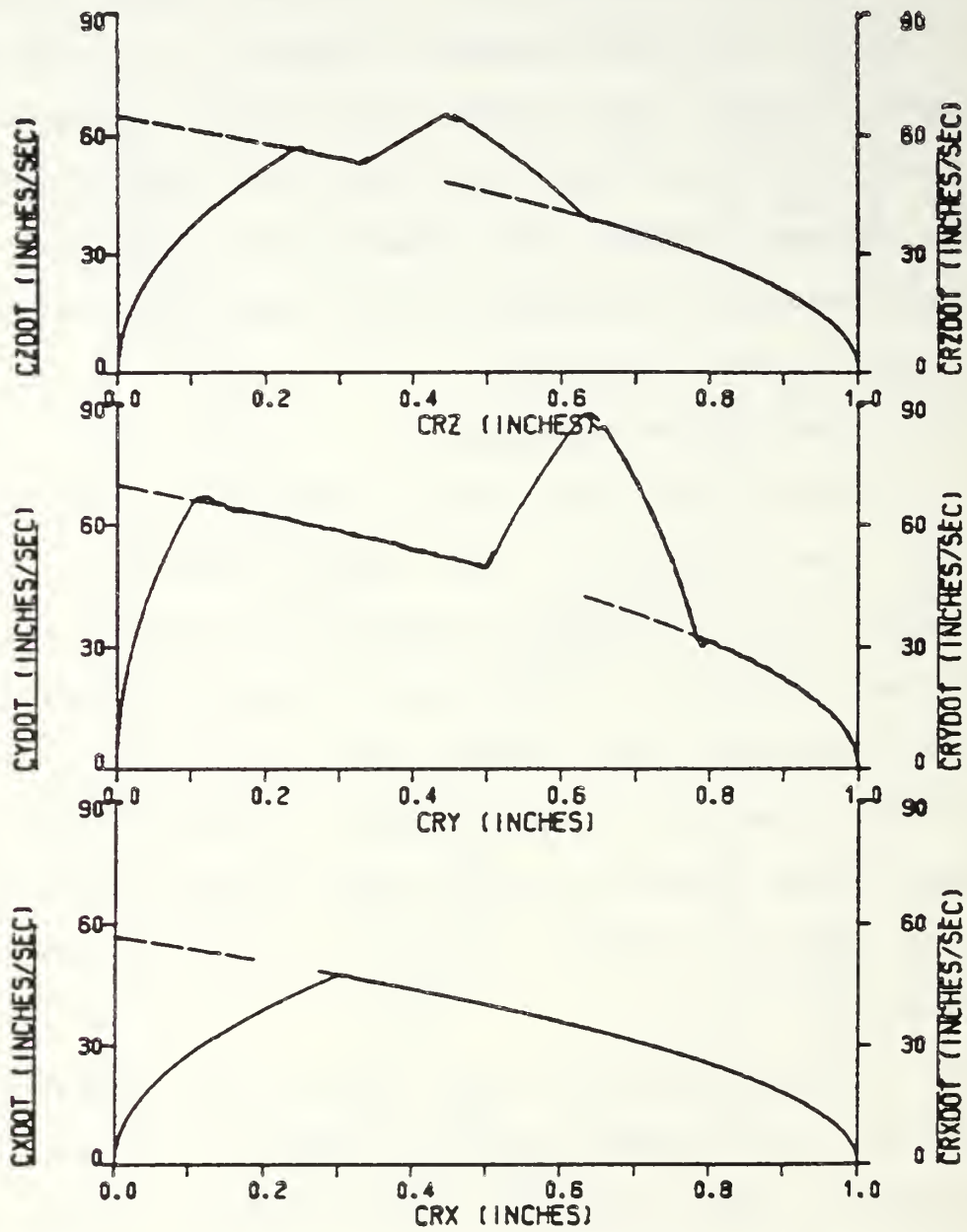
Figure 6.16   Phase Plane Trajectory with Disturbance
Applied at T=10 msec (No Gravity)
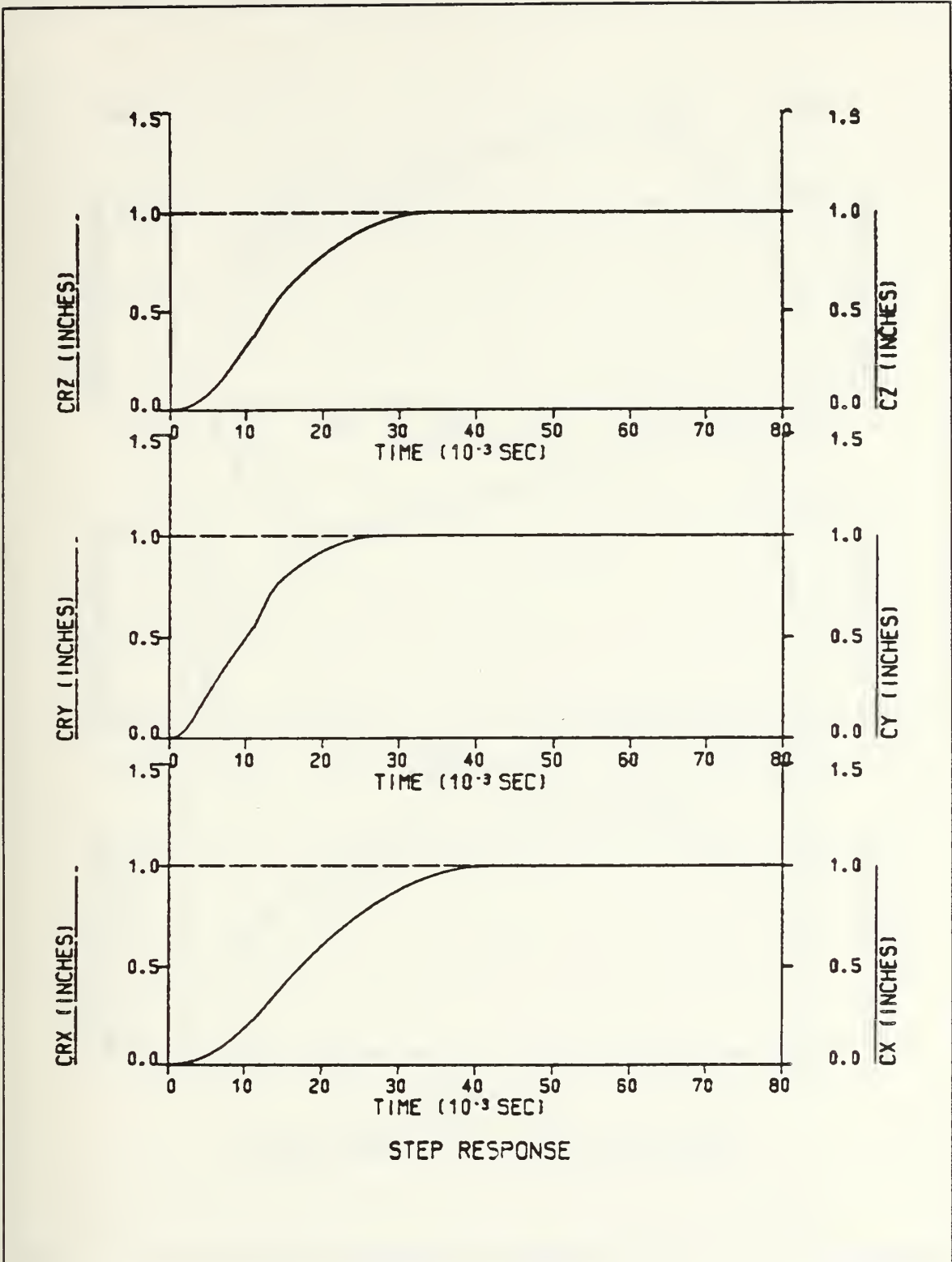
Figure 6.17   Step Response with Disturbance
Applied at T=10 msec (No Gravity)
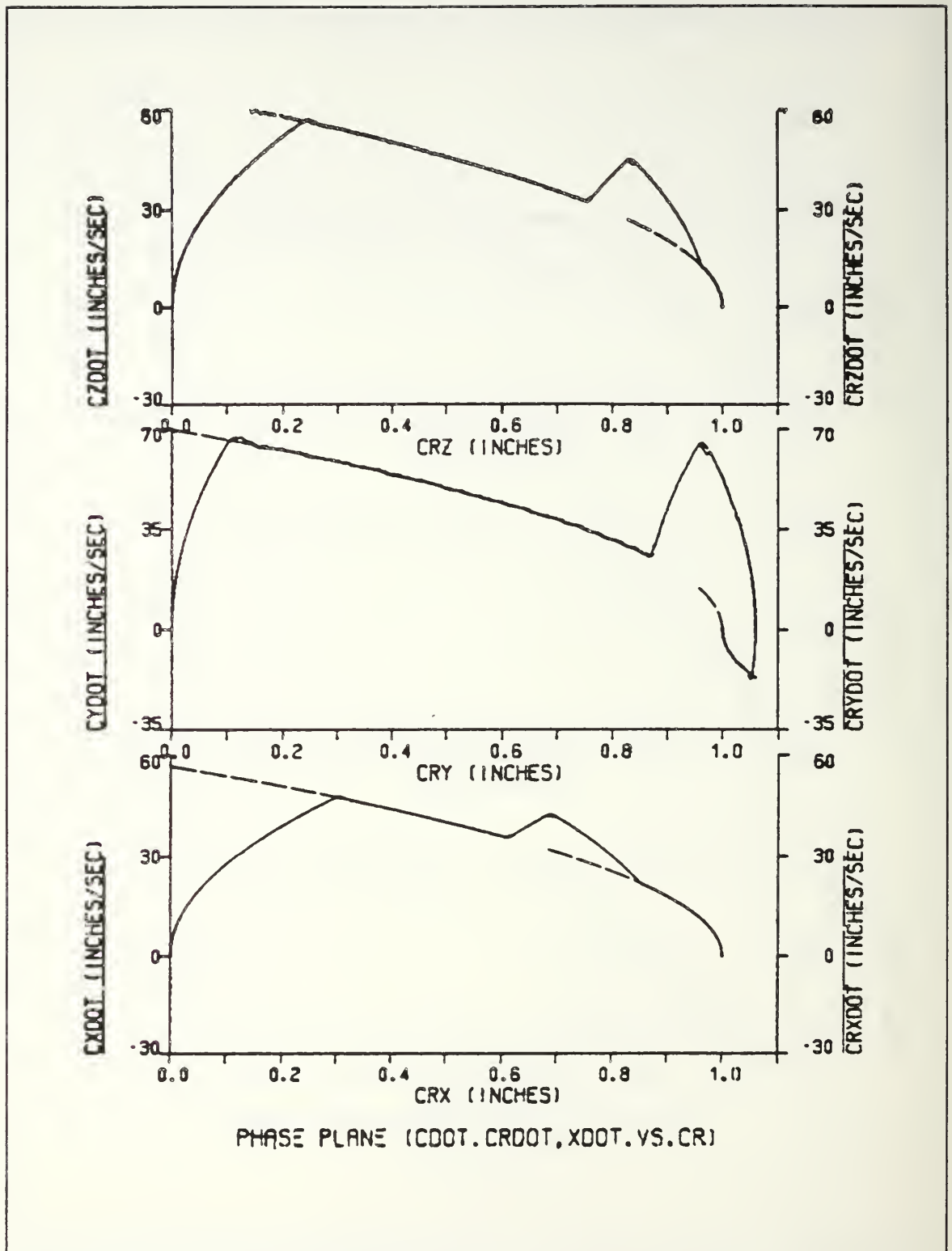
Figure 6.18   Phase Plane Trajectory with Disturbance
Applied at T=20 msec (No Gravity)
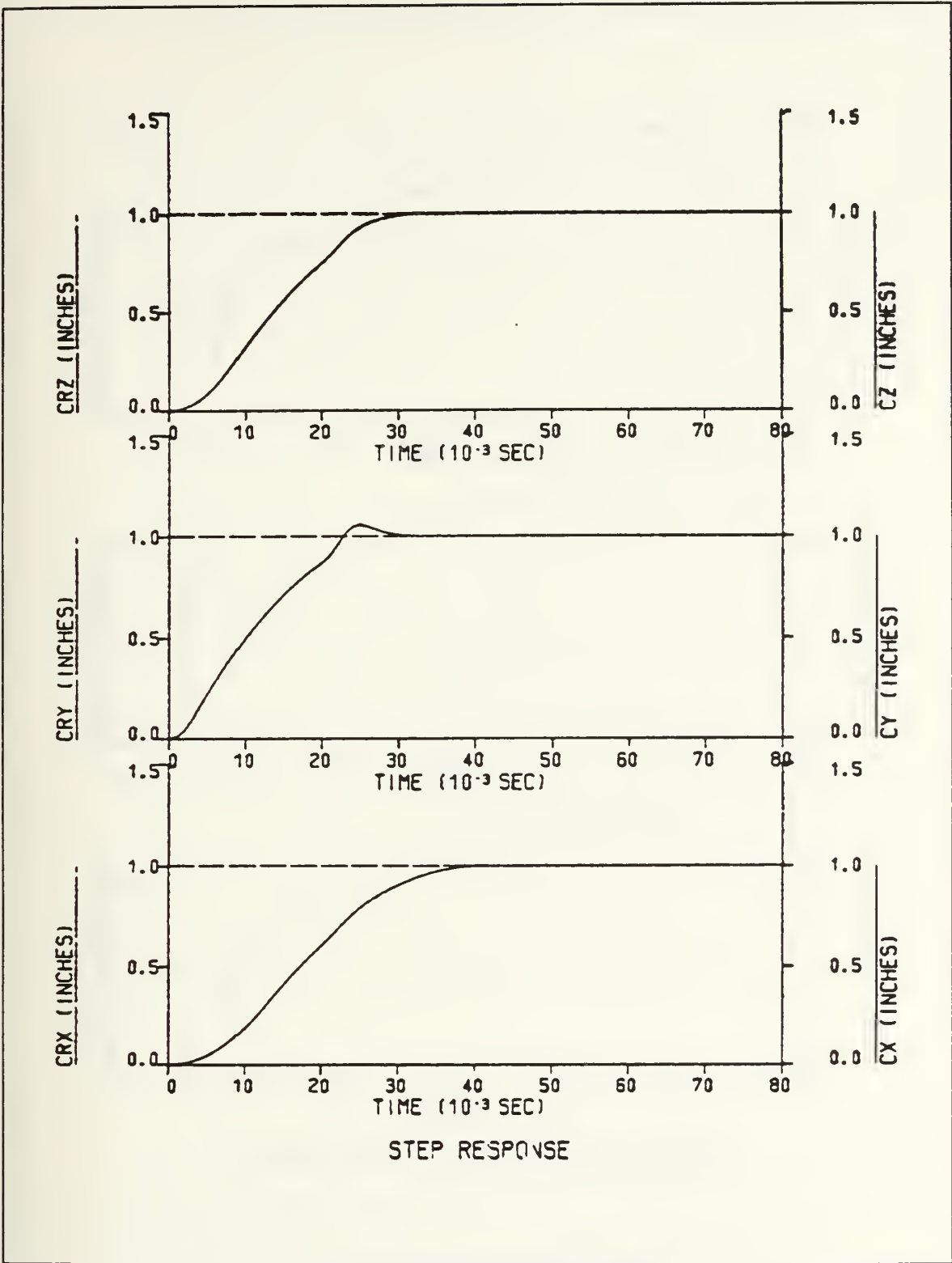
Figure 6.19   Step Response with Disturbance
Applied at T=20 msec (No Gravity)
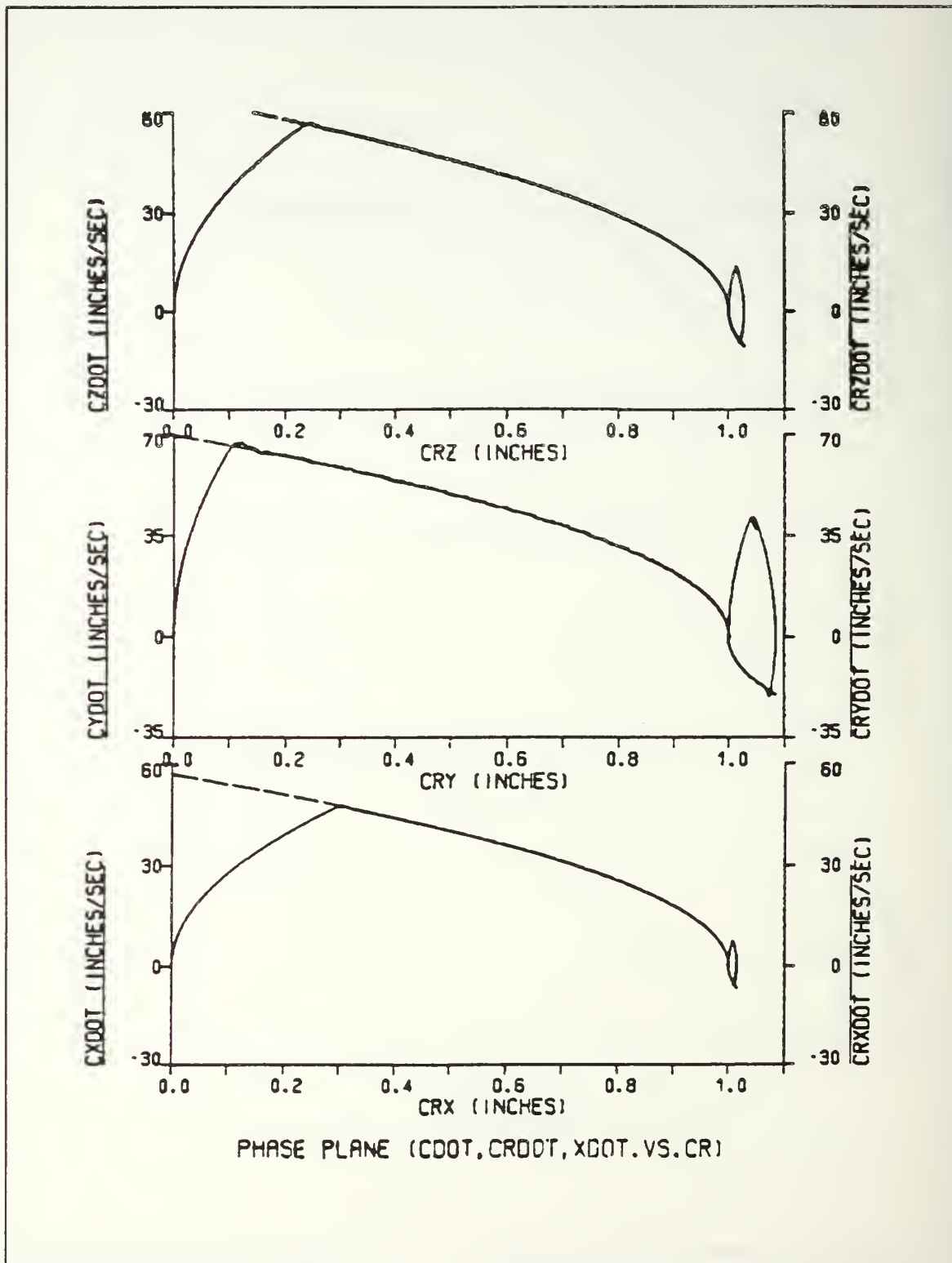
Figure 6.20 Phase Plane Trajectory with Disturbance
Applied at T=50 msec (No Gravity)
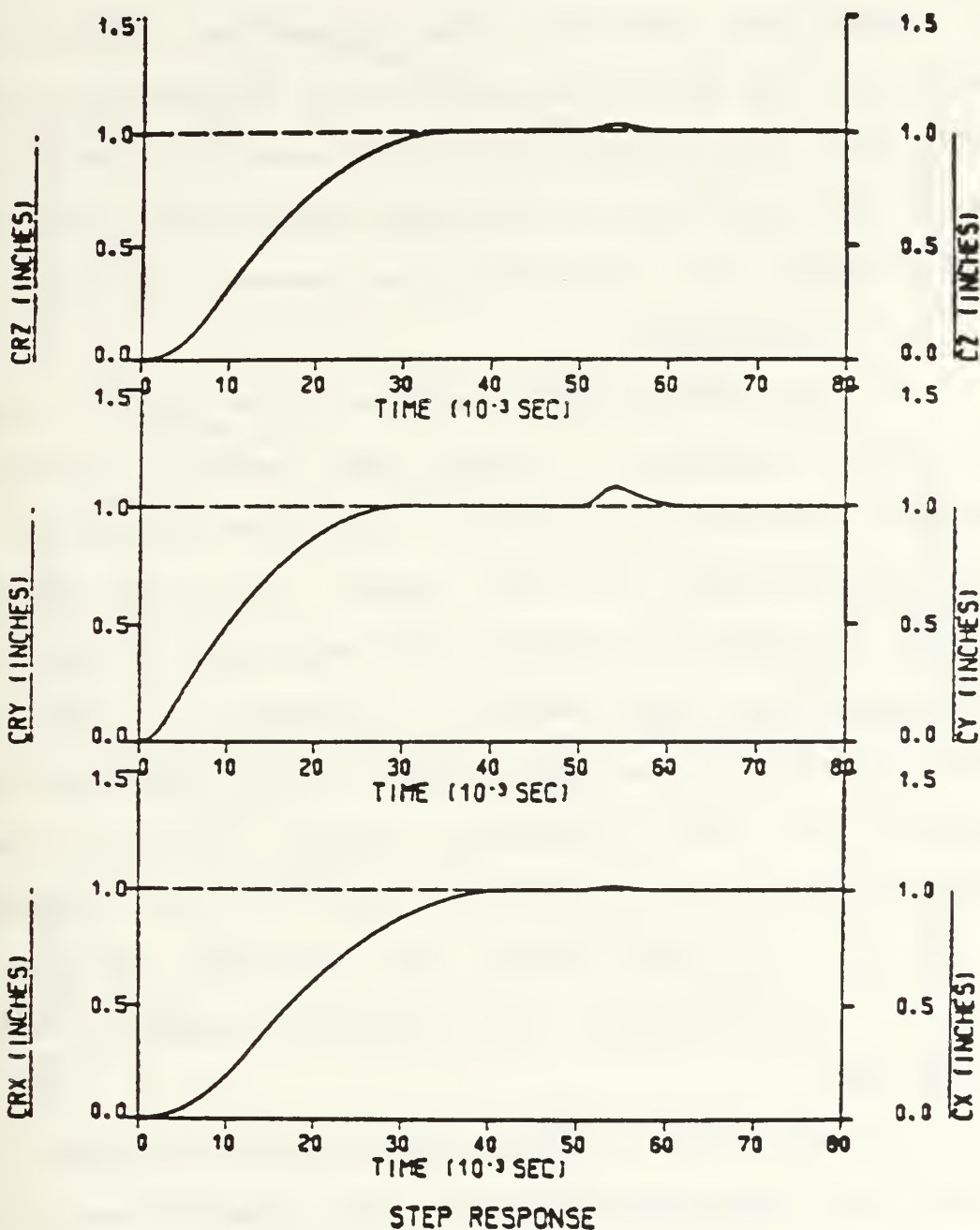
Figure 6.21   Step Response with Disturbance
Applied at T=50 msec (No Gravity)

Although the same disturbance is applied to all links, the link affected mostly is the y-axis link which is the lightest one. Generally, the lighter the link the more sensitive to any disturbance. But in all different cases the results show that the applied disturbances are rejected in a time period which is also inversely proportional to link's weight. Finally, all links go to the commanded position.

c. Robustness

The robot control problem is to design a stable and robust algorithm to enable the robot to follow a specified trajectory. Therefore, robustness is one of the main considerations in robot design. Next the adaptive algorithm is tested for slight (10%) variations of the servo motors parametric data because, although the same servo motors are used for the three joints, there are always some tolerances in their parametric values that may cause significant changes in the robot behavior if the system is not robust. For these tests the simulation program of Appendix C is used again, with different parametric values for each run.

The effects of changing the torque constant (K_t) and the back emf constant (K_v) for the unloaded arm by 10% are depicted in Figures 6.22 - 6.25. Figures 6.23 and 6.25 show a very small variation in settling time which is proportional to the links weights and has the same sign with the variation of the two parameters. Figures 6.22 and 6.24

84
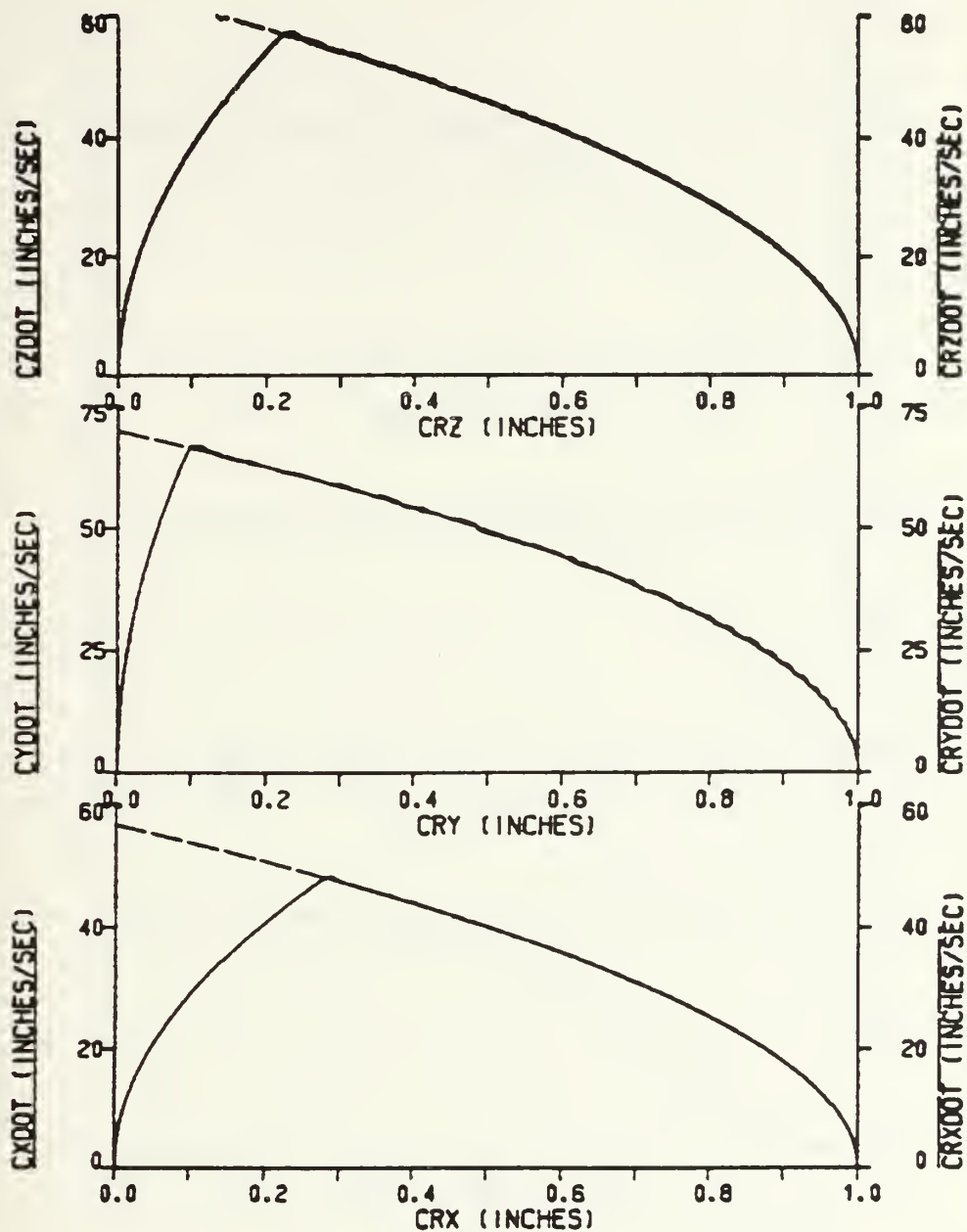
Figure 6.22   Phase Plane Trajectory for Unloaded Arm
with K_t and K_v Increased by 10% (No Gravity)

85
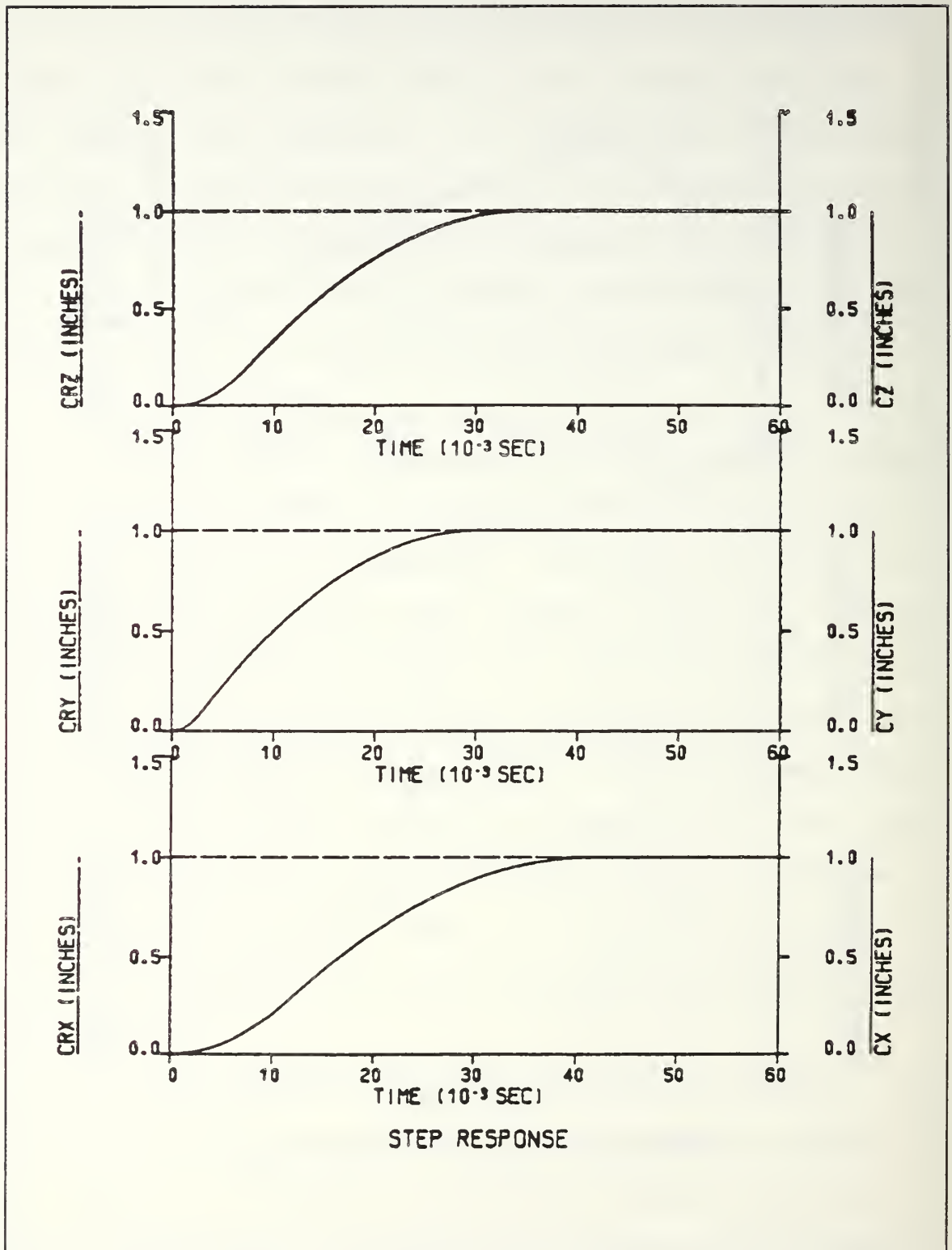
Figure 6.23   Step Response for Unloaded Arm with
K_t and K_v Increased by 10% (No Gravity)

86
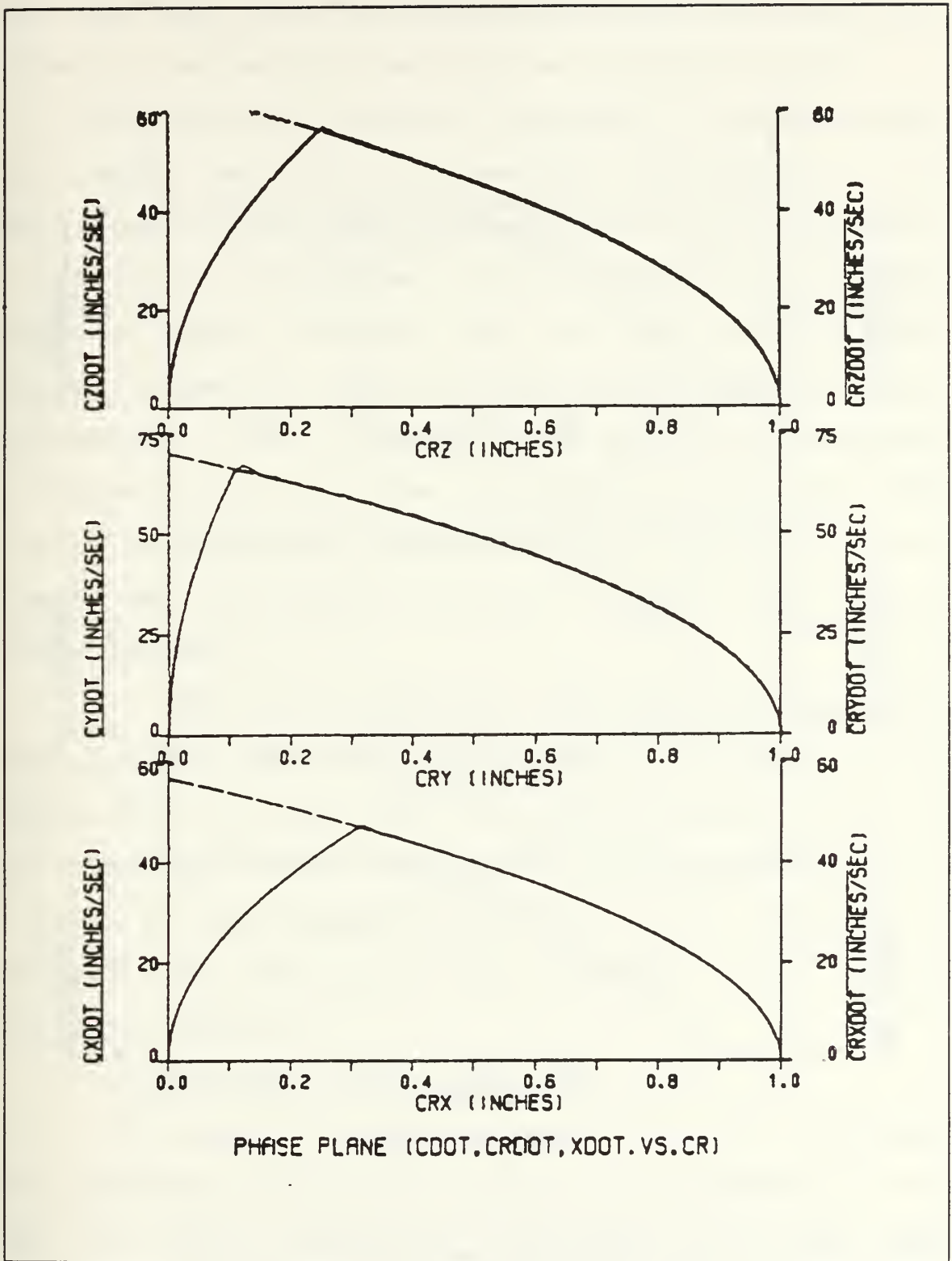
Figure 6.24   Phase Plane Trajectory for Unloaded Arm
with K_t and K_v Decreased by 10% (No Gravity)
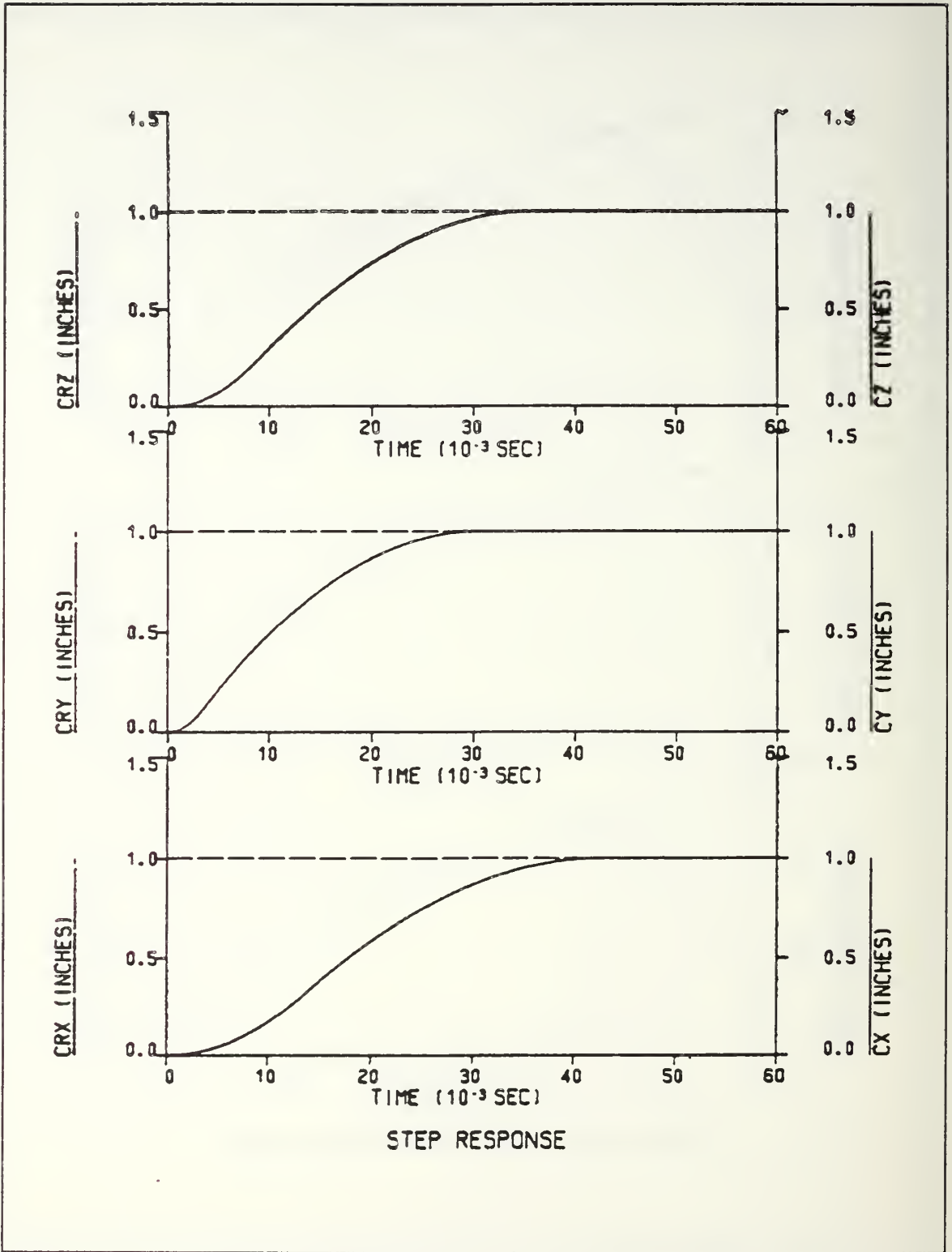
Figure 6.25 Step Response for Unloaded Arm with
K_t and K_v Decreased by 10% (No Gravity)

show very good curve following characteristics either for increase or decrease of the above two parametric values.

Then the effects of a movement of the mechanical pole towards the origin are examined. For the pole to be moved closer to the origin, the resistance (R) is decreased by 10% and the inductance (L) is increased by the same percentage amount. Figures 6.26 and 6.27, compared with Figures 6.1 and 6.2, show no change in the behavior of the unloaded arm. If load is placed on the gripper of the robot, as Figures 6.28 - 6.31 show, no overshoot occurs for loads up to 0.94 oz/in/sec^2. Comparison with Figures 6.4 - 6.8 reveals that the critical load is increased from 0.83 to 0.94 oz/in/sec^2.

With the mechanical pole moved towards the origin, if the constants K_t and K_v are increased by 10% Figures 6.32 - 35 show that the robot can carry load up to 1.08 oz/in/sec^2 without overshooting. In the opposite case, if K_t and K_v are decreased by the same amount, Figures 6.36 - 39 show that load up to 0.80 oz/in/sec^2 can be carried without overshooting.

2. Gravitational Torques Included

The adaptive system is tested again for different load conditions, disturbance rejection and robustness in the case of motor parameters variation, but this time gravitational torques are included in the simulation. Actually, the only link which senses gravitational torques
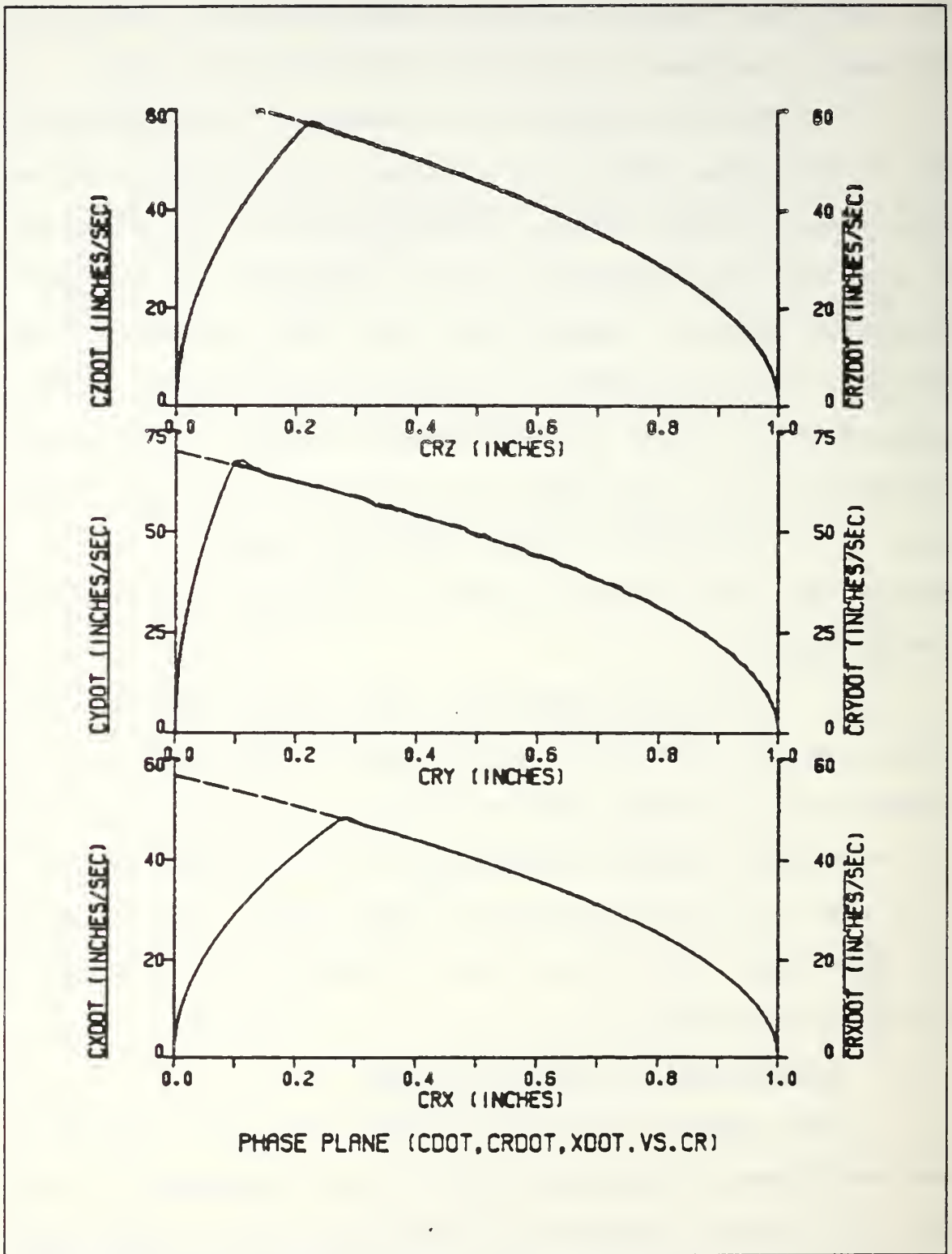
Figure 6.26   Phase Plane Trajectory for Unloaded Arm with
R Decreased and L Increased by 10% (No Gravity)

90
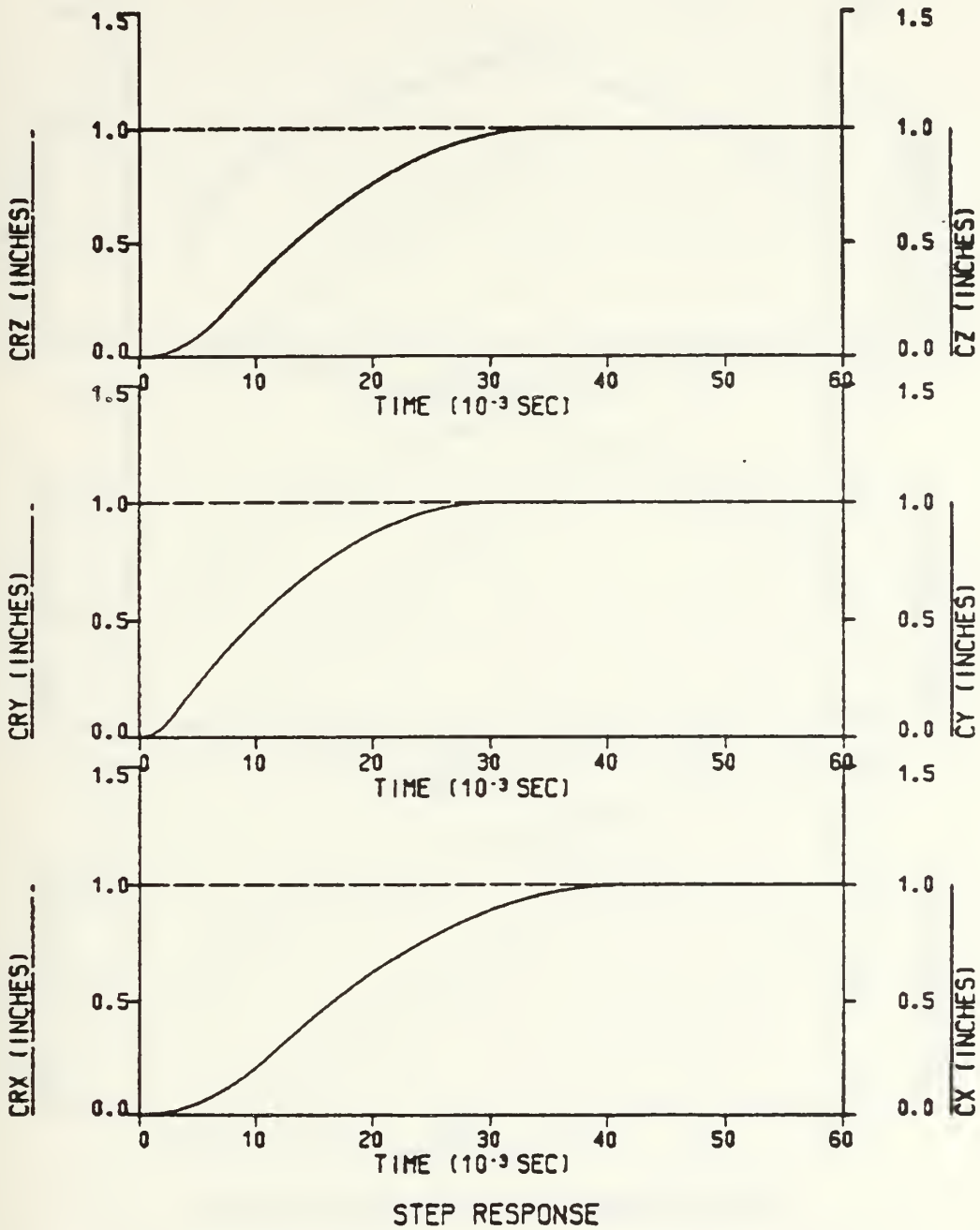
Figure 6.27   Step Response for Unloaded Arm with
Decreased and L Increased by 10% (No Gravity)

91
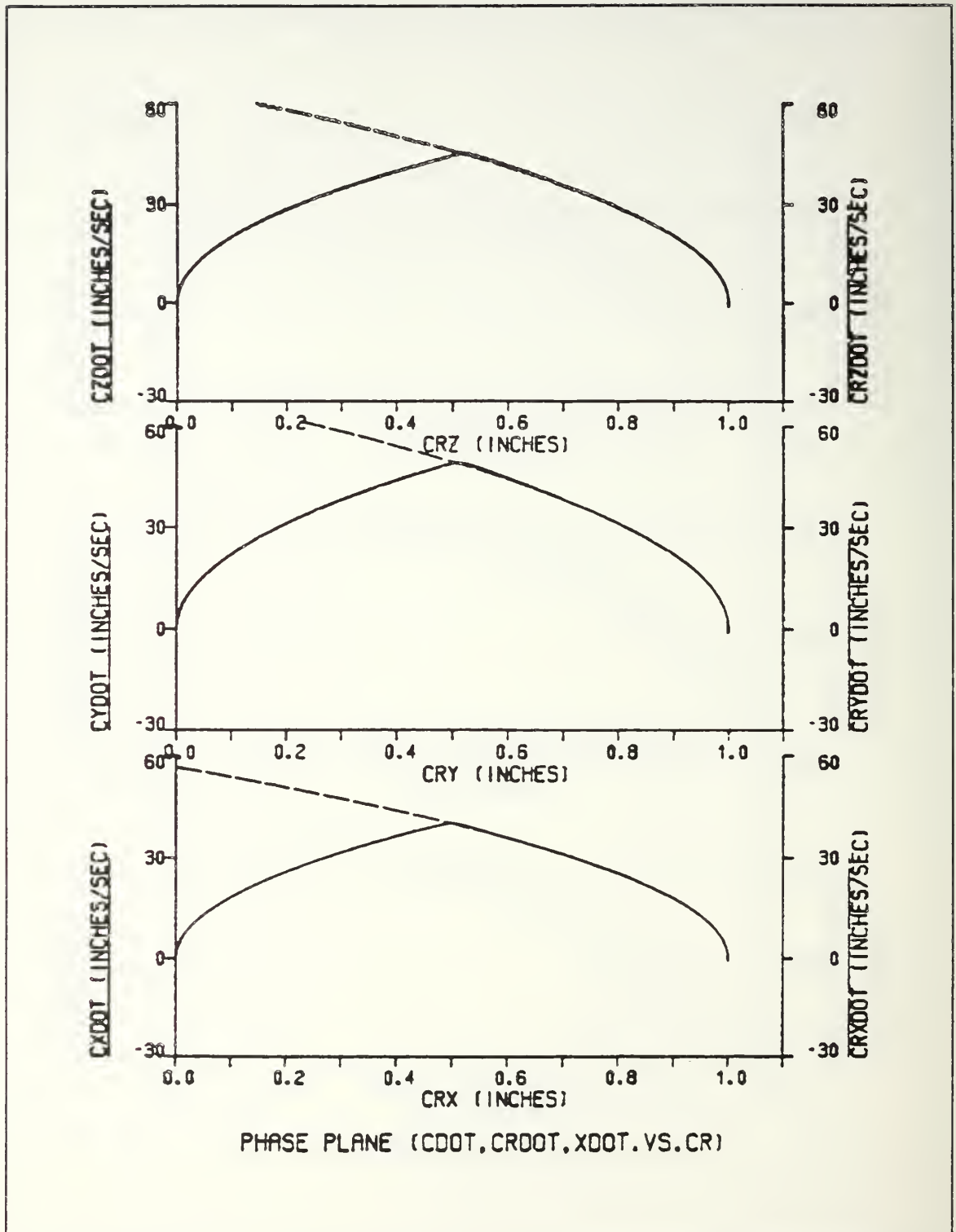
Figure 6.28   Phase Plane Trajectory for Loaded Arm
with R Decreased and L Increased by 10%
(Load=0.94 - No Gravity)
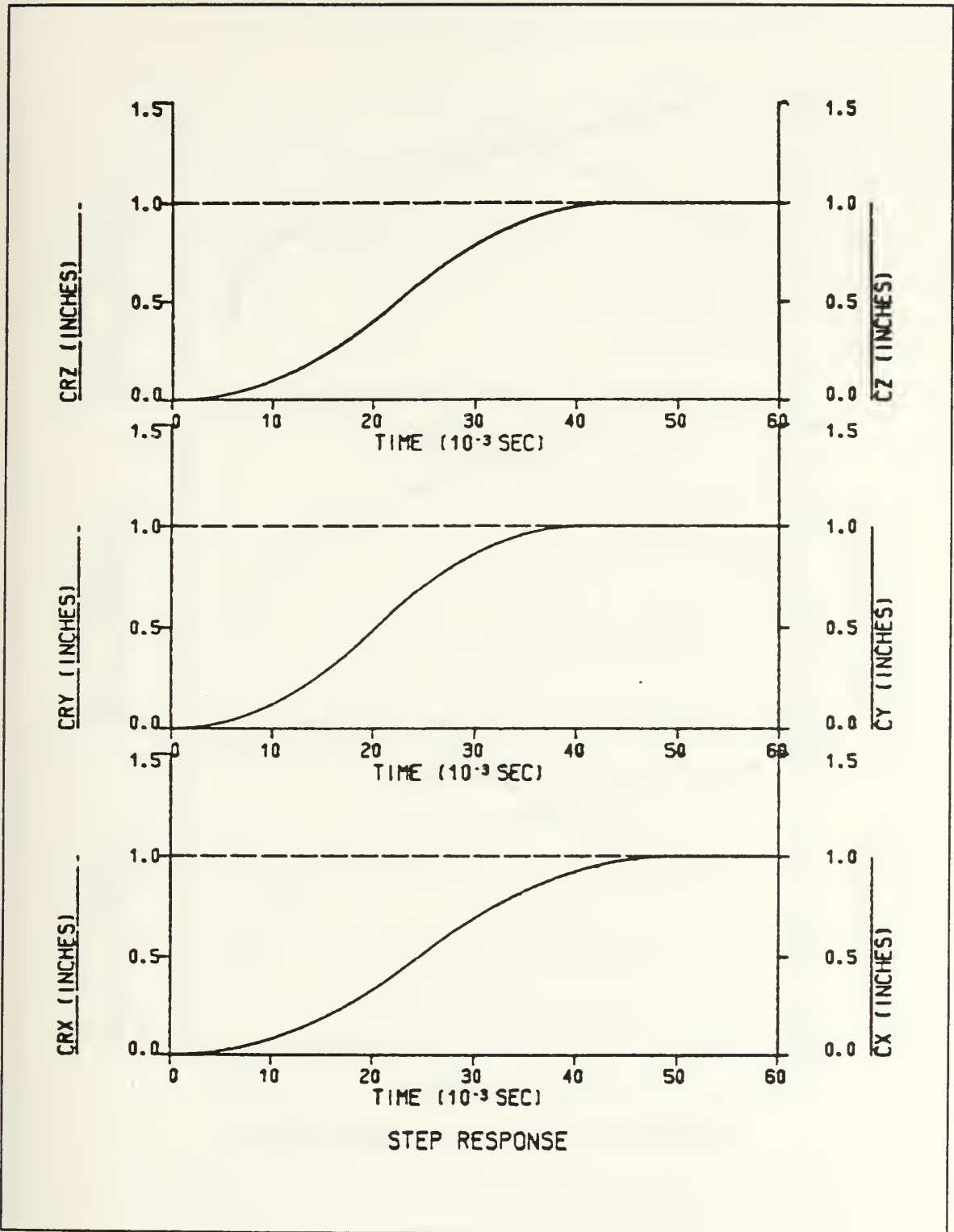
Figure 6.29 Step Response for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.94 - No Gravity)
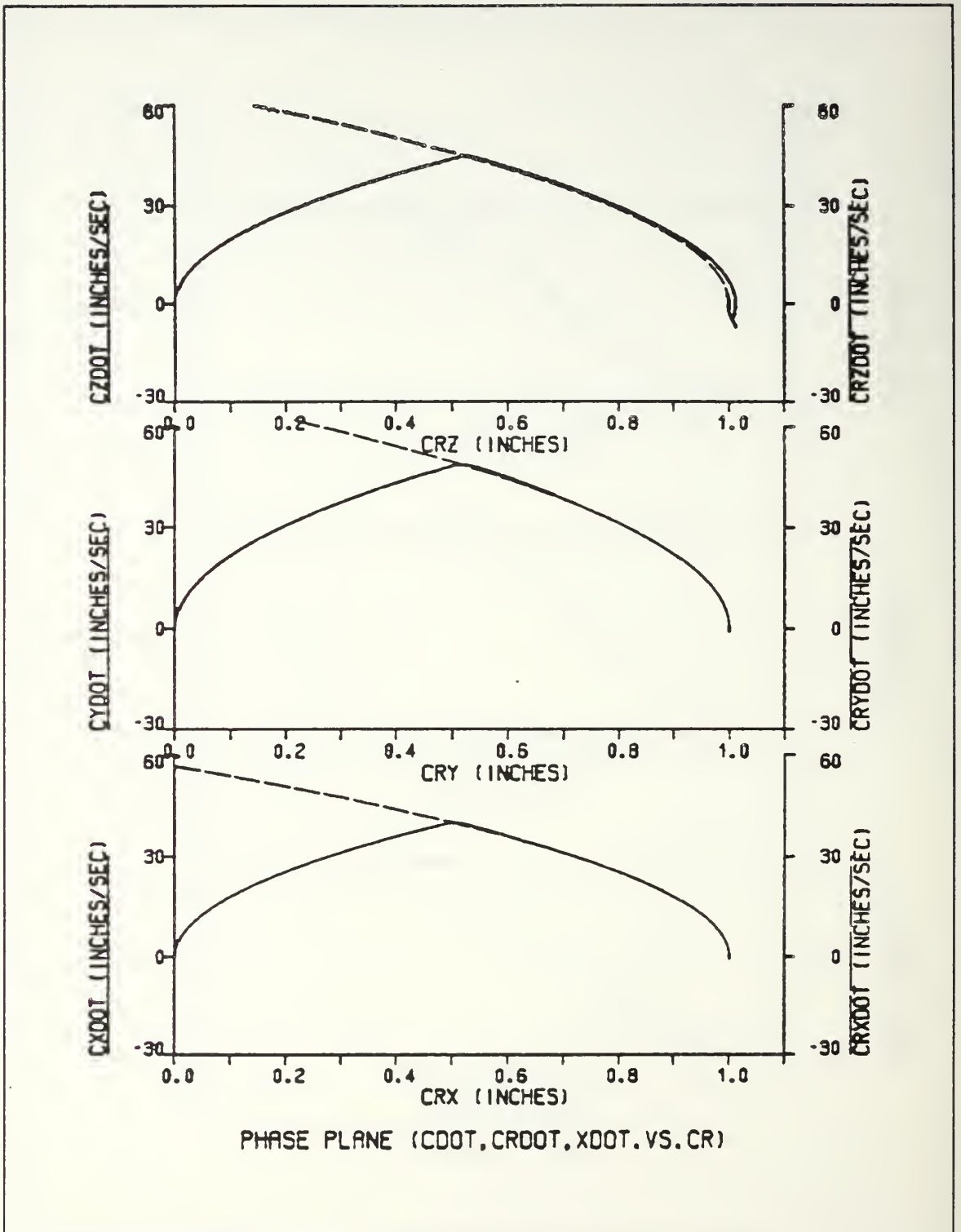
Figure 6.30   Phase Plane Trajectory for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.98 - No Gravity)

94
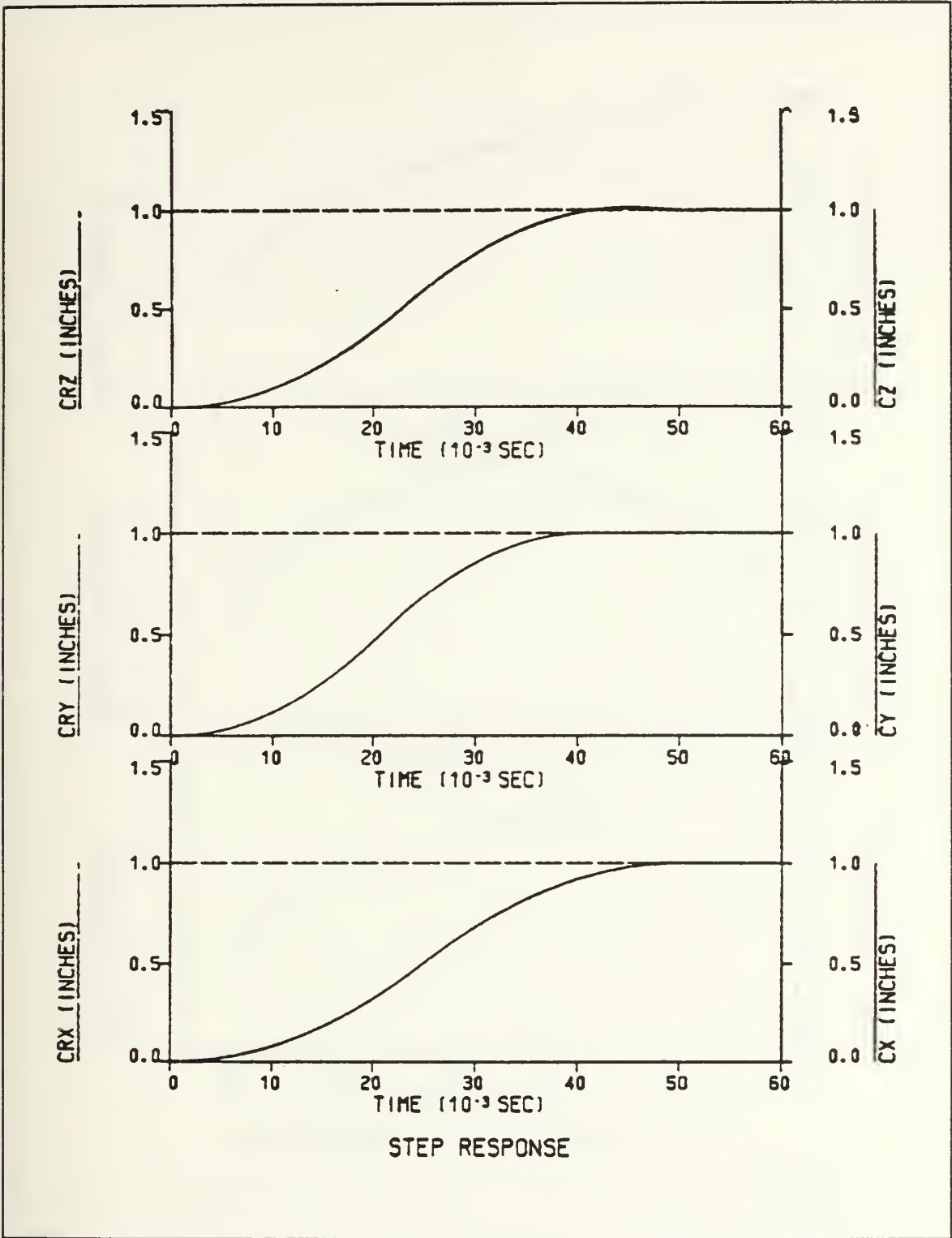
Figure 6.31   Step Response for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.98 - No Gravity)
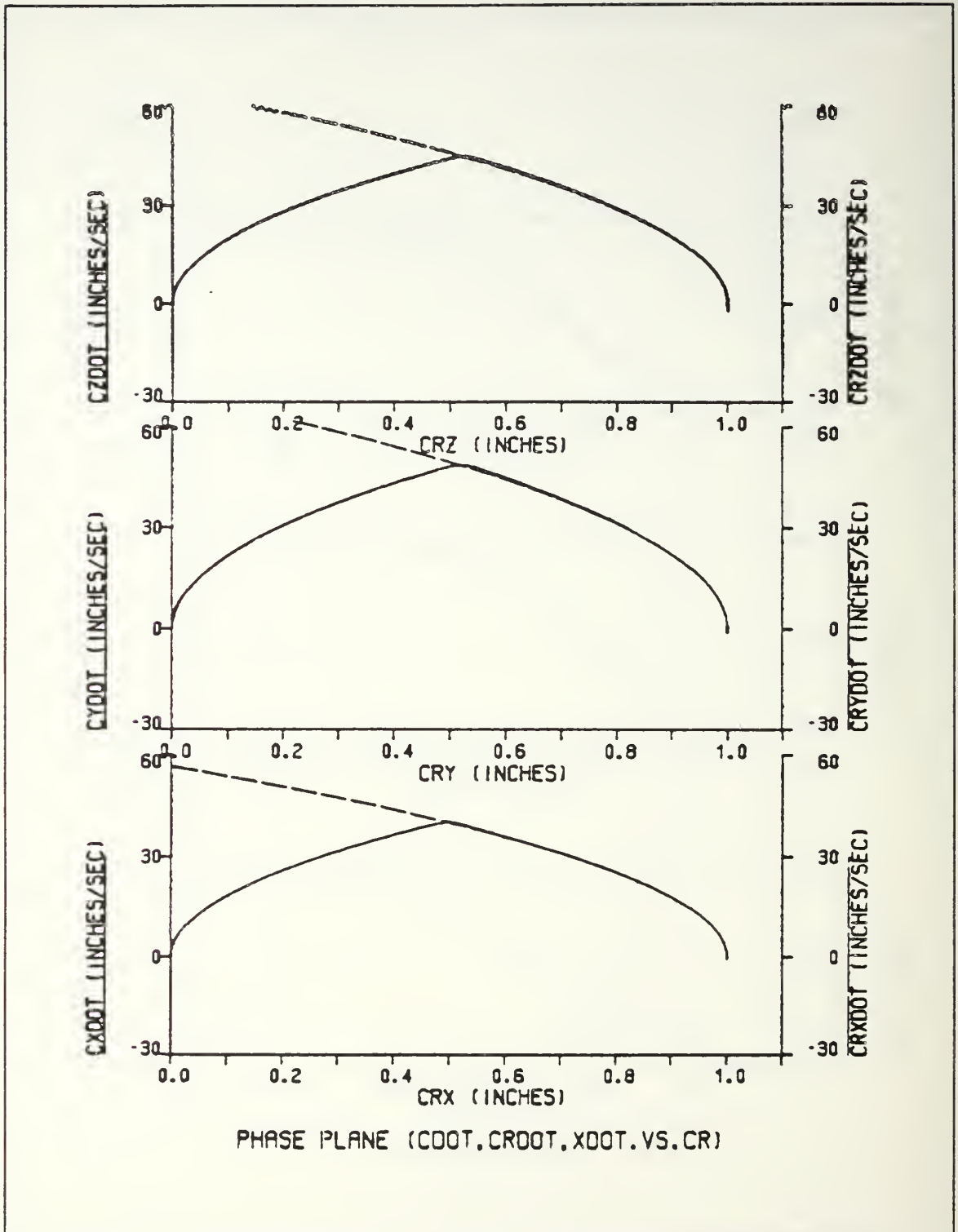
Figure 6.32 Phase Plane Trajectory for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=1.08 - No Gravity)
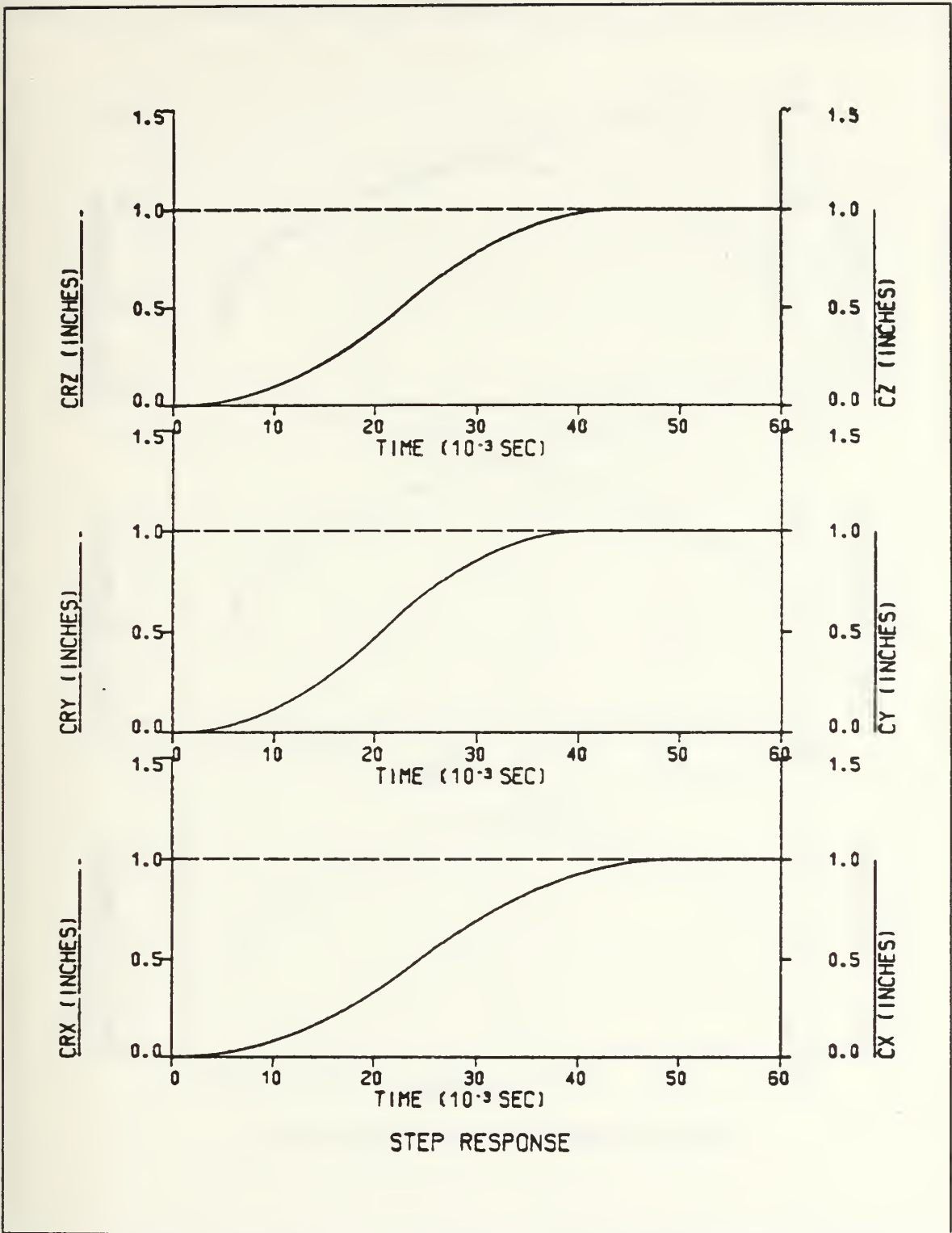
Figure 6.33   Step Response for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=1.08 - No Gravity)
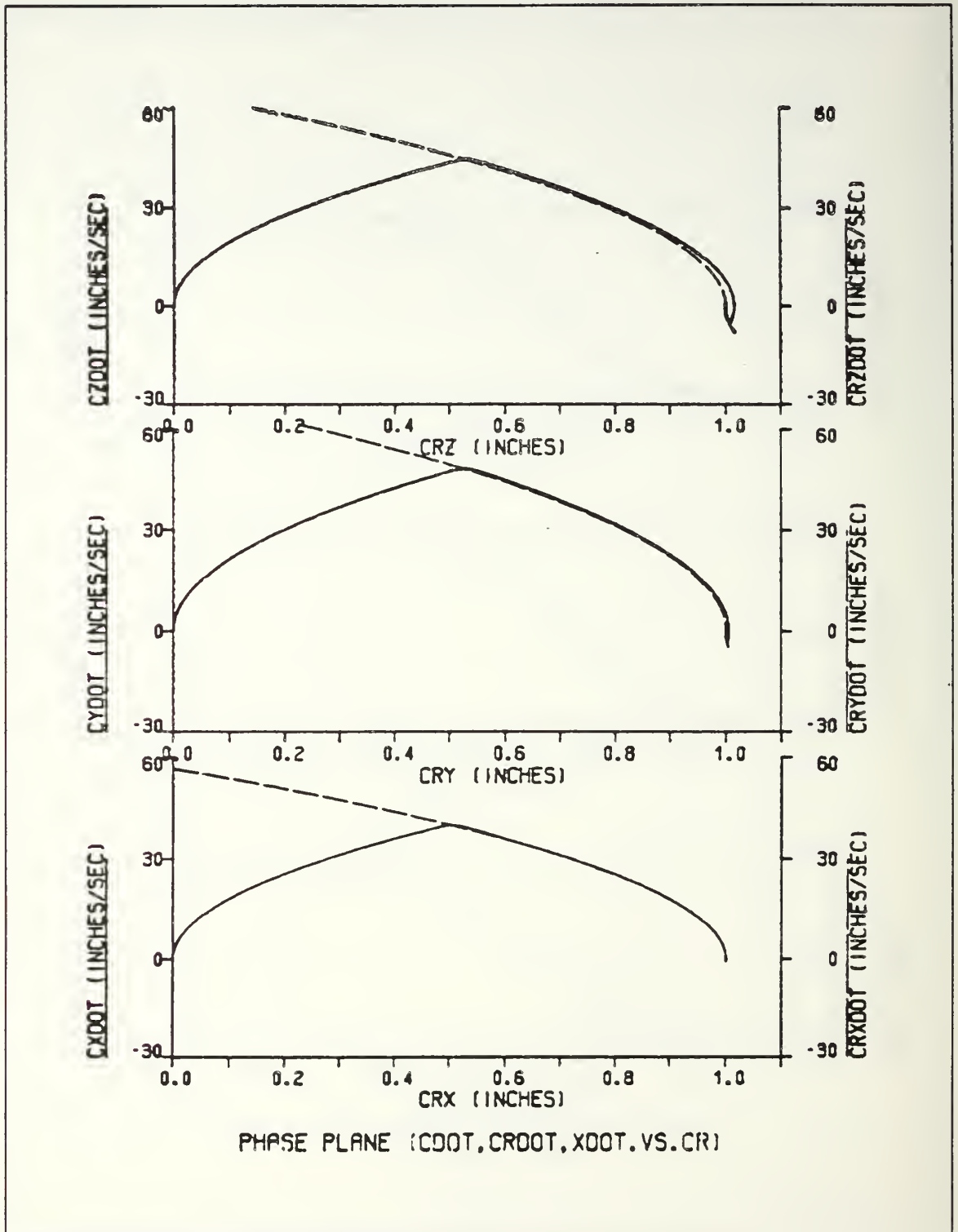
Figure 6.34   Phase Plane Trajectory for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=1.12 - No Gravity)

98
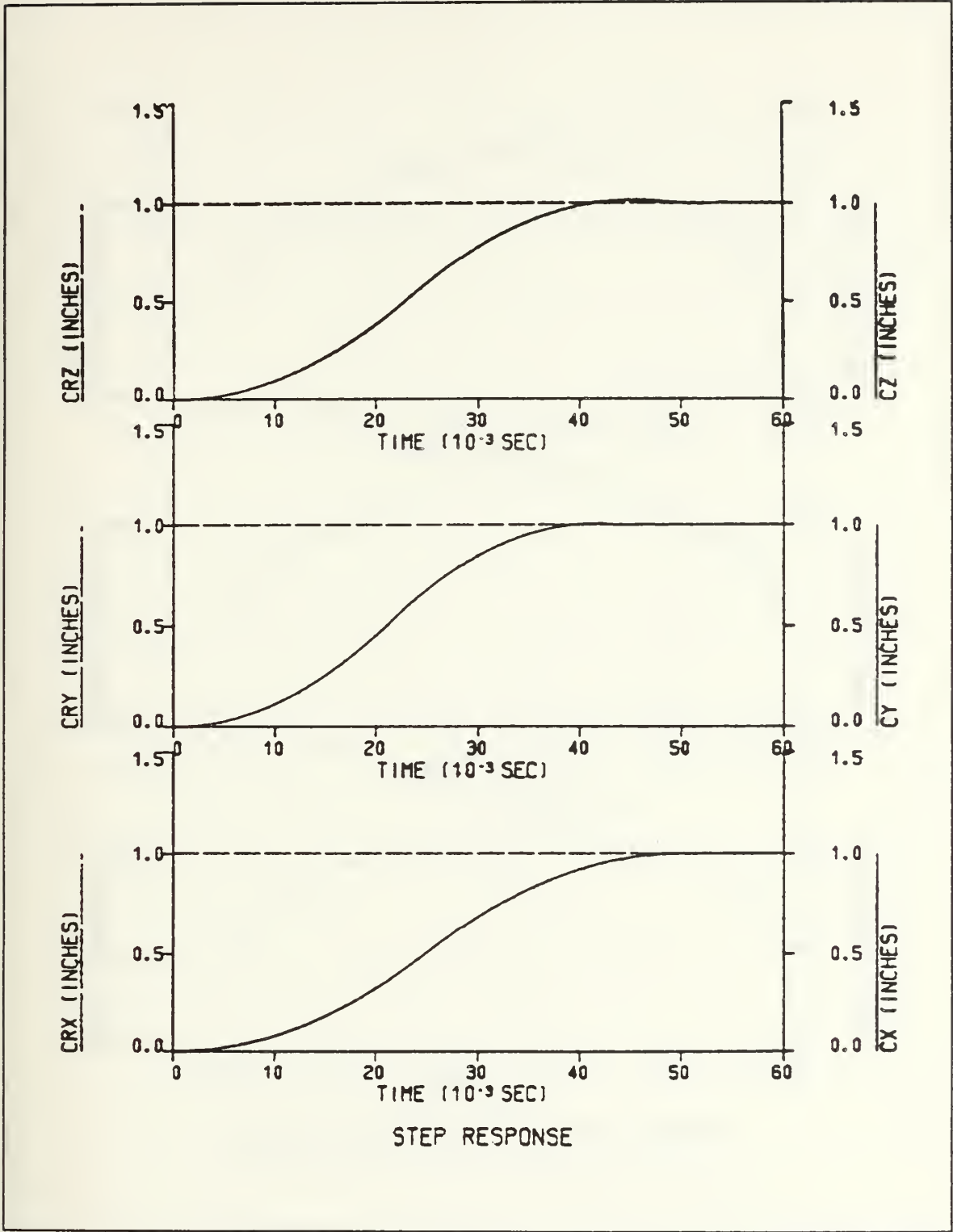
Figure 6.35   Step Response for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=1.12 – No Gravity)
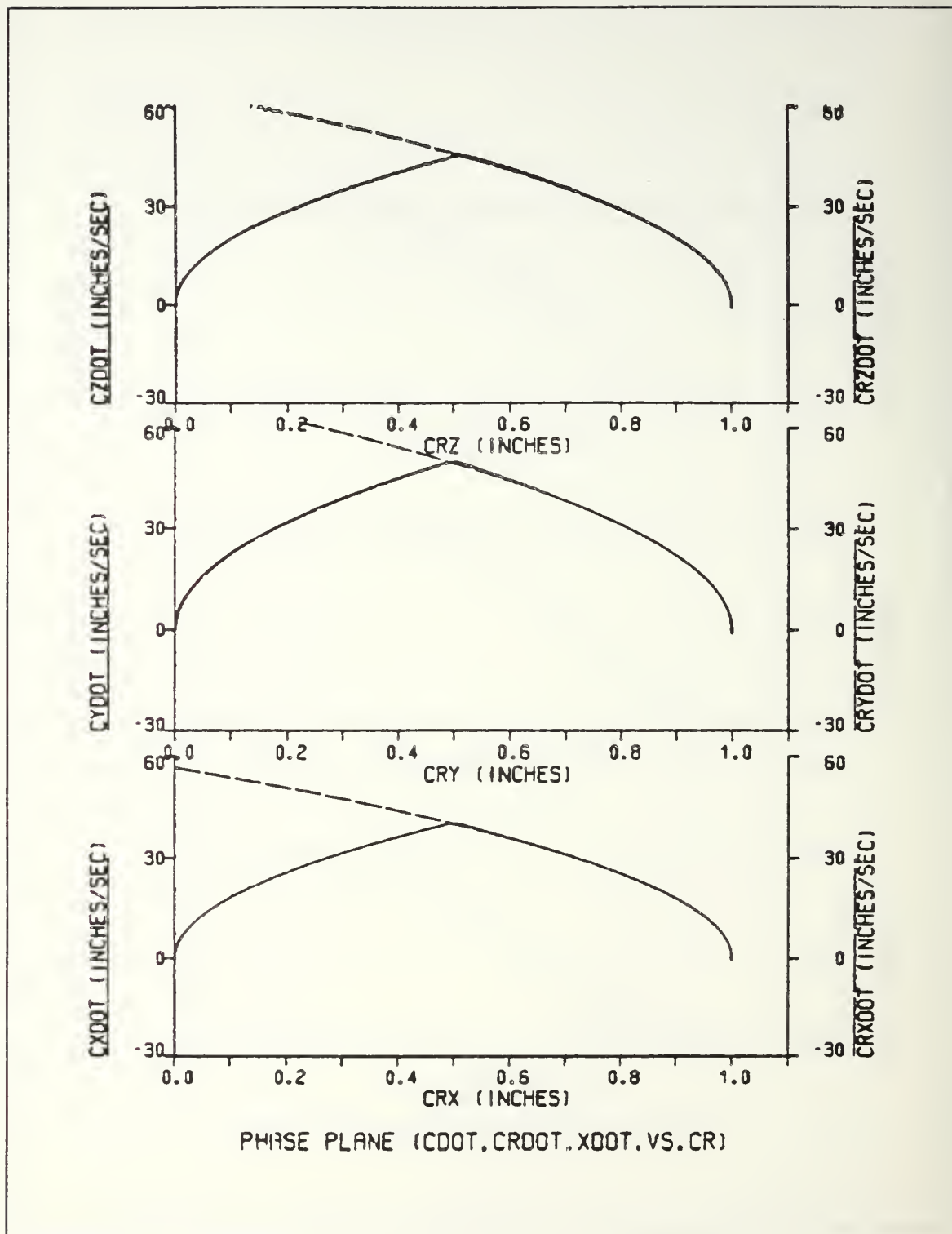
Figure 6.36   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.80 - No Gravity)
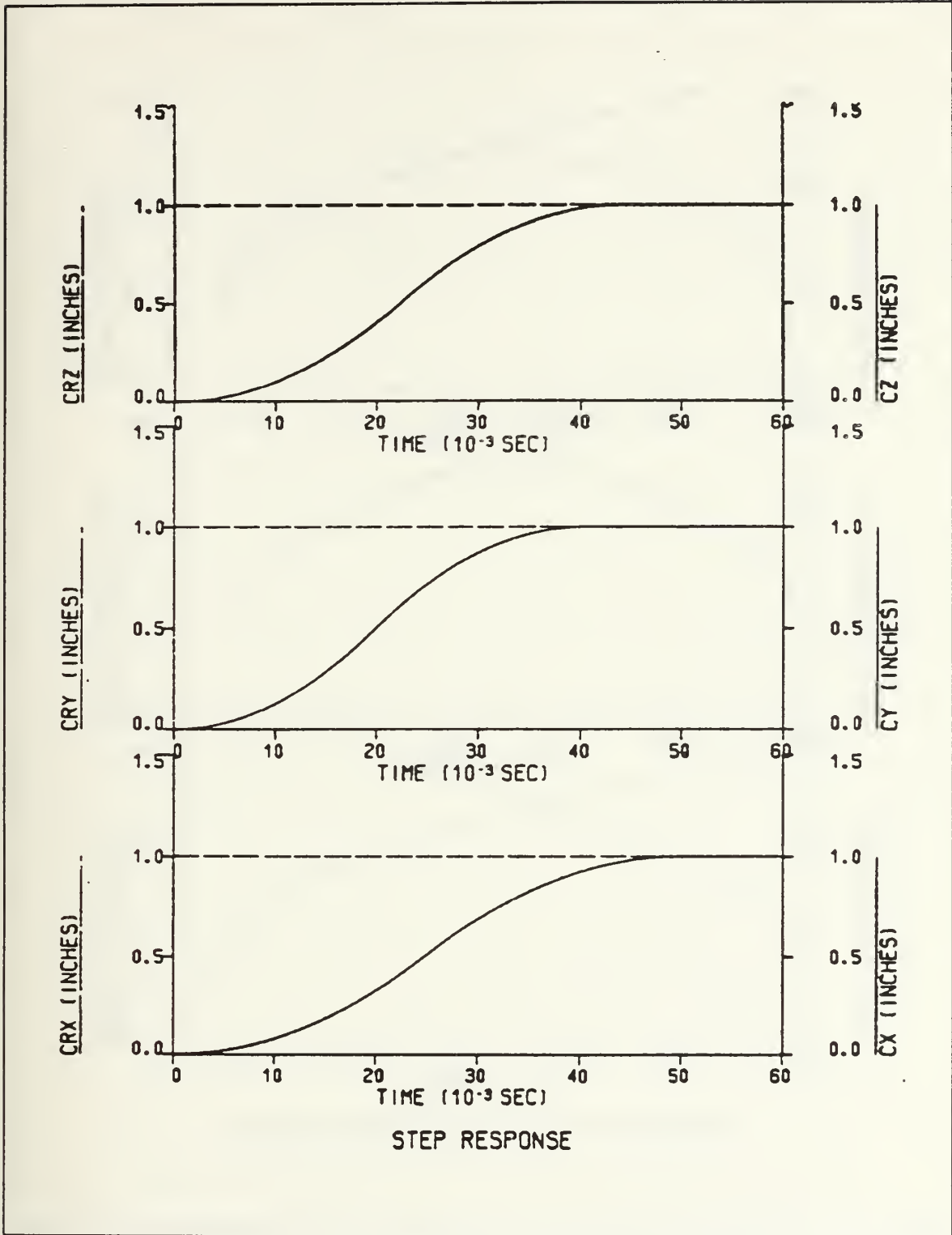
Figure 6.37 Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.80 - No Gravity)

101
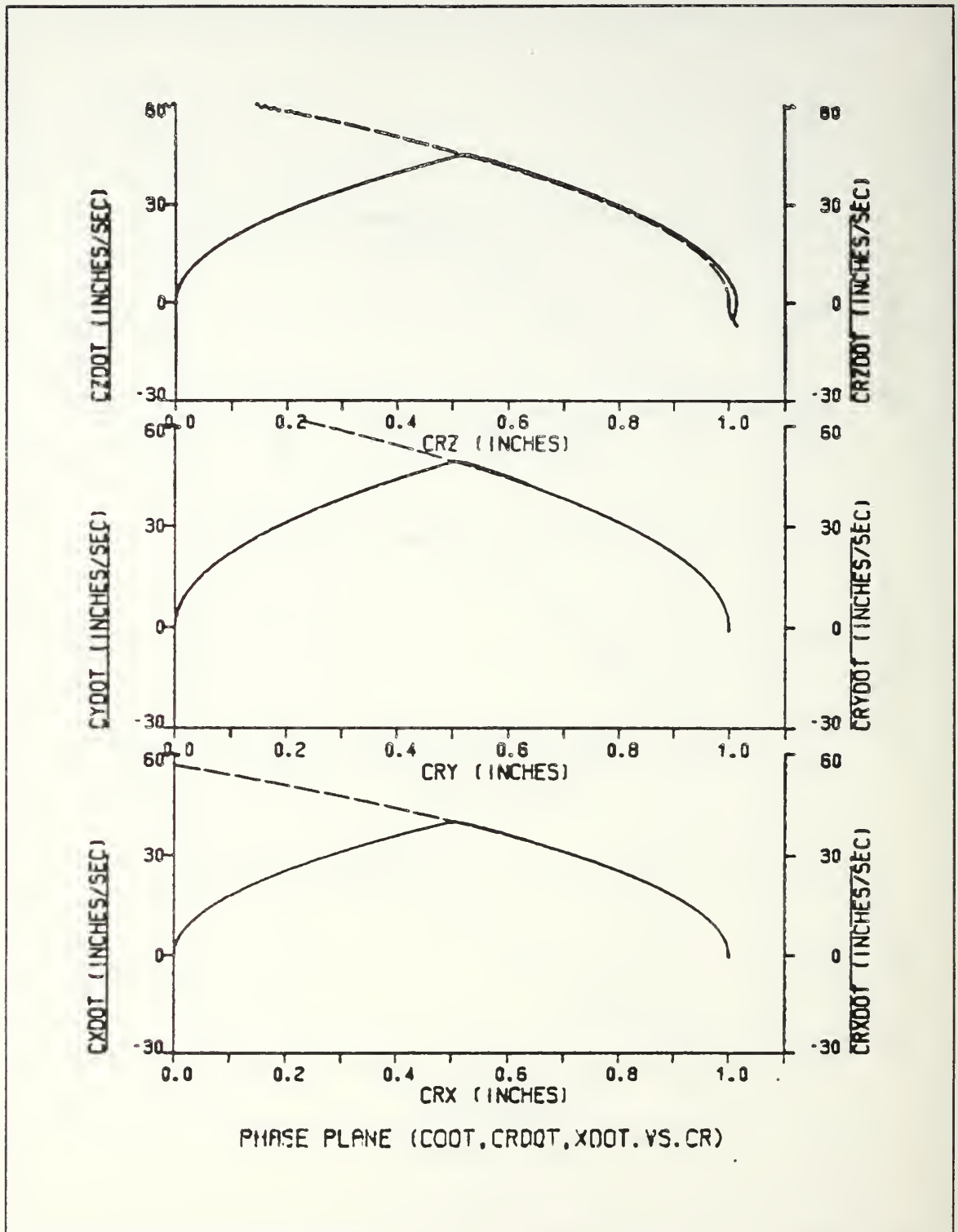
Figure 6.38   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.84 – No Gravity)
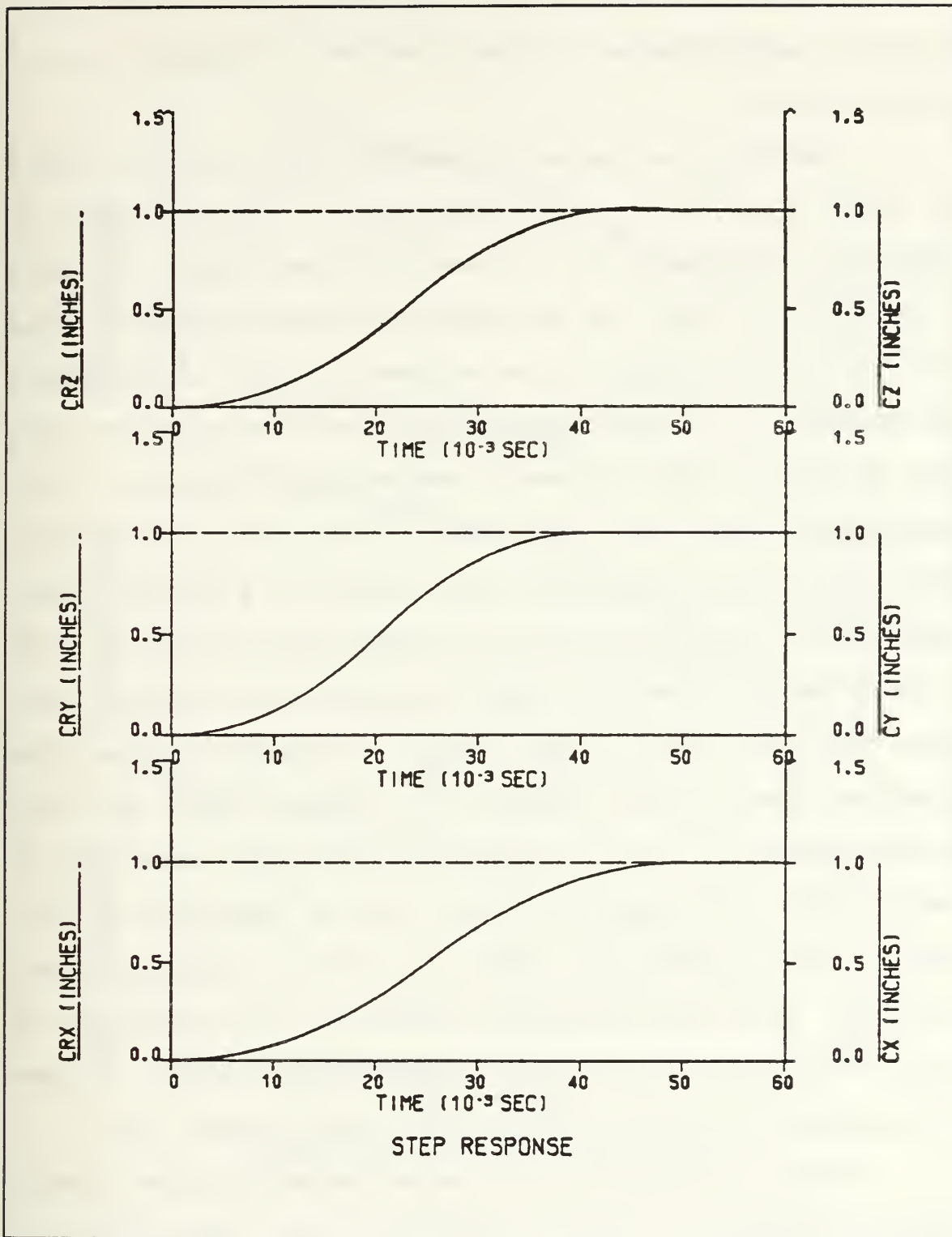
Figure 6.39   Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.84 - No Gravity)

103

is the vertical one because the other two links moving in a direction perpendicular to the direction of gravity are not affected at all.

The adaptive system is tested first without carrying any load, using the DSL/VS simulation program listed in Appendix E. Figures 6.40, 6.41 and 6.42 seem to be identical to Figures 6.1, 6.2 and 6.3 which are the respective ones for the robot operating in a no gravitational environment, and especially the step response and the error curves show that the three model and servo motors track together to the commanded position with zero steady state error. An accurate observation of the simulation data reveals a positive steady state error of the order of 10^{-5} inches for the z-axis link of the robot arm operating under gravitational torques. This means that the tip of the z-axis link never reaches the commanded position but stops 10^{-5} inches lower than the desired position. The error curves of the other two links in steady state condition oscillate with an amplitude of the order of 10^{-5} inches, as they do in the no gravity case. Therefore, the only difference observed in the behavior of the unloaded robot arm operating under gravitational torques is the steady state error of the vertical (z-axis) link.

Then the robot arm is tested under the same loaded conditions applied in the no gravity case. The simulation results show that the behavior of the robot arm under gravitational torques is almost identical to the behavior of

104
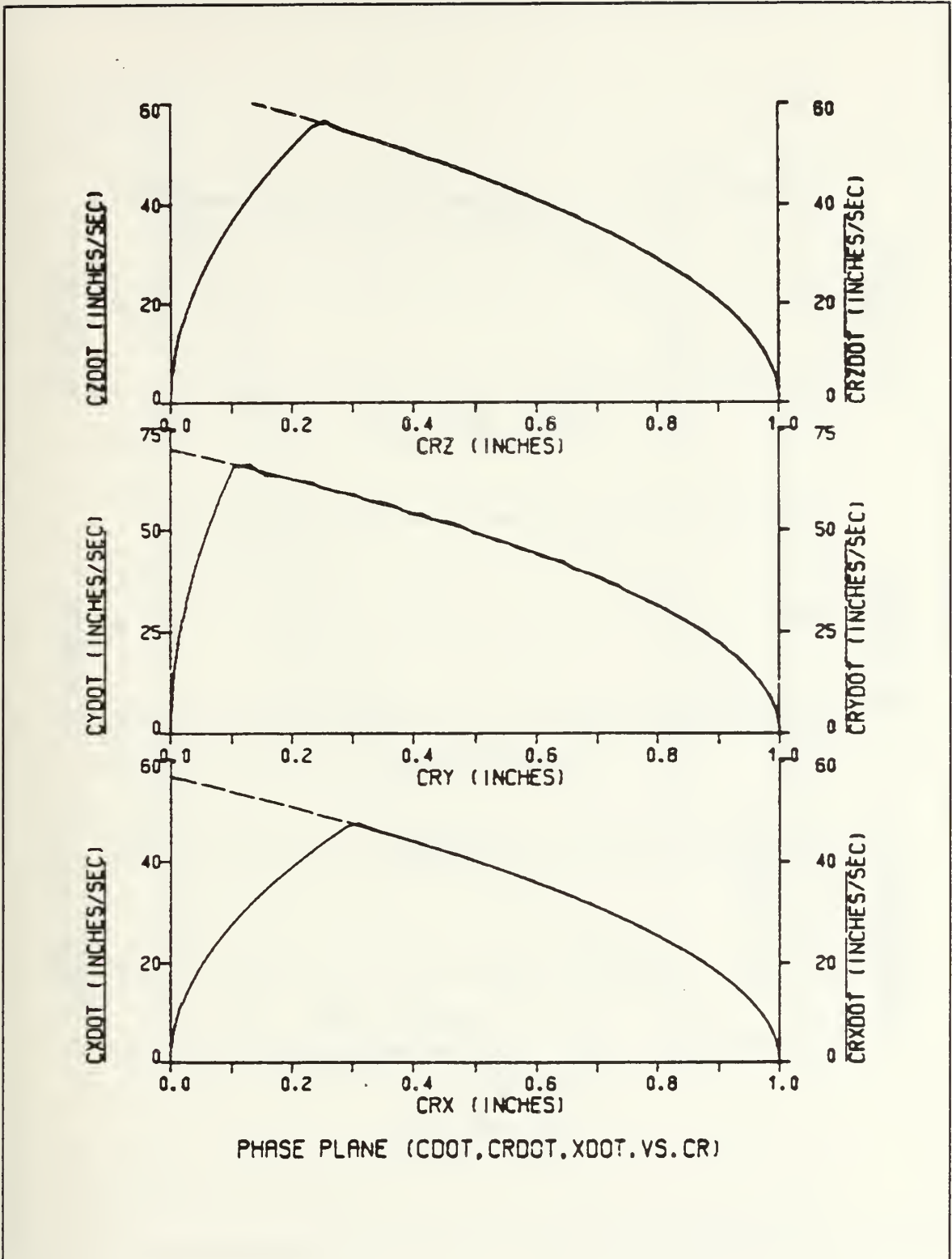
Figure 6.40   Phase Plane Trajectory for Unloaded Arm
(With Gravity)
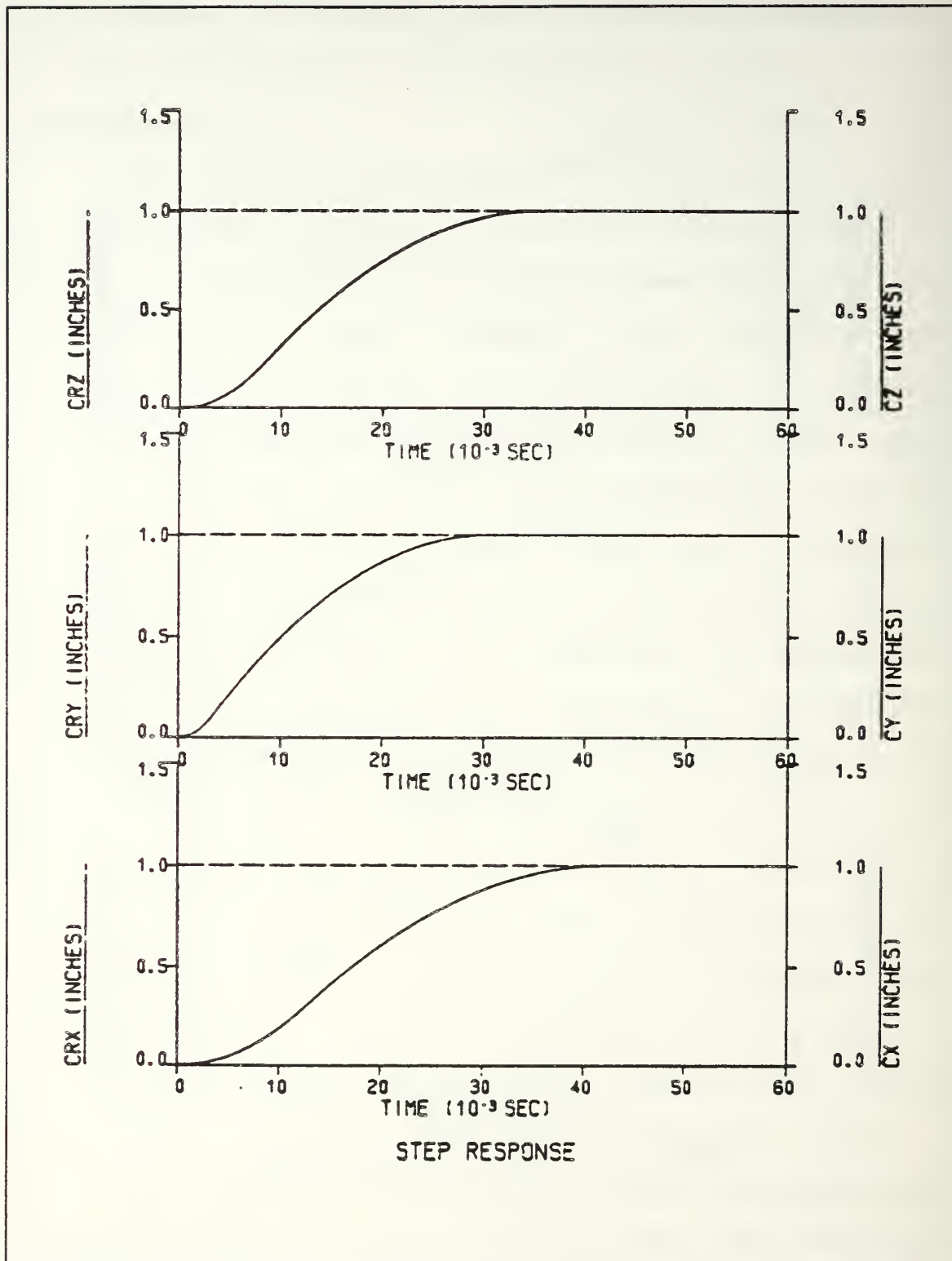
Figure 6.41   Step Response for Unloaded Arm
(With Gravity)
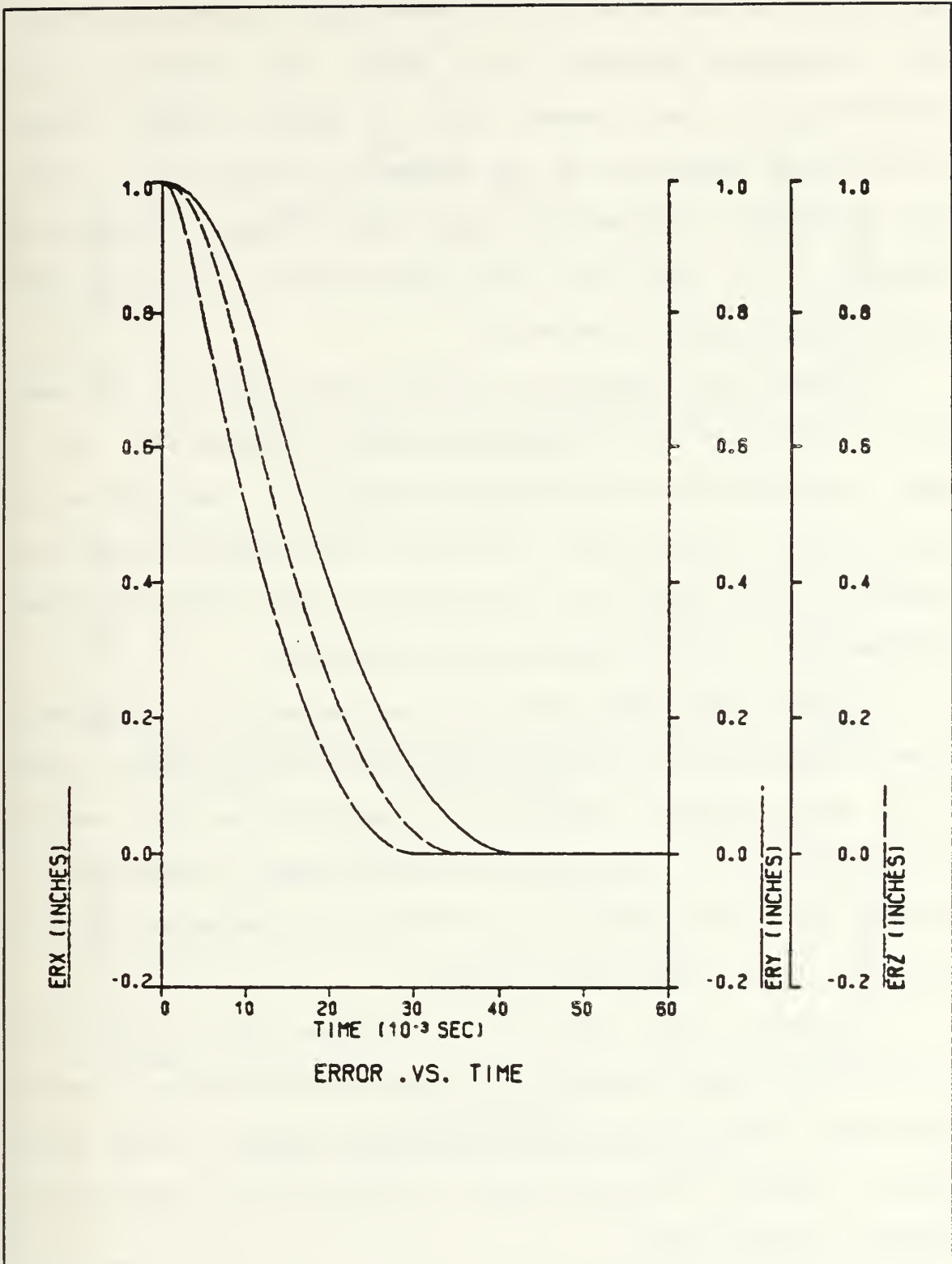
Figure 6.42   Error Curves for Unloaded Arm
(With Gravity)

107

the robot arm operating in a gravity free environment. The only difference observed is a small (3%) change in the capability of the tested arm to carry loads without overshooting depending on the direction of the move. If the arm is moving against gravity, the change is positive (Figures 6.43, 6.44 and 6.45), but if it is moving with gravity the change is negative.

Since the simulation results are very much the same with the results of the robot arm under no gravity, most of them are omitted in this section and just a few indicative figures are included to illustrate the nearly identical behavior of the three link rectangular robot operating under gravitational and no gravitational torques.

After that, the robot arm is tested for disturbance rejection applying the same disturbances mentioned in part 1.b of this chapter. Figures 6.46 and 6.47 are the results obtained using the simulation program listed in Appendix F, and show that the applied disturbances are rejected and all links go to the commanded position.

Finally, the robot arm is tested for slight (10%) variation of motor parameters. Simulation results show no noteworthy change in the characteristics of the system which behaves exactly as does under no gravity as depicted in Figures 6.48 and 6.49.
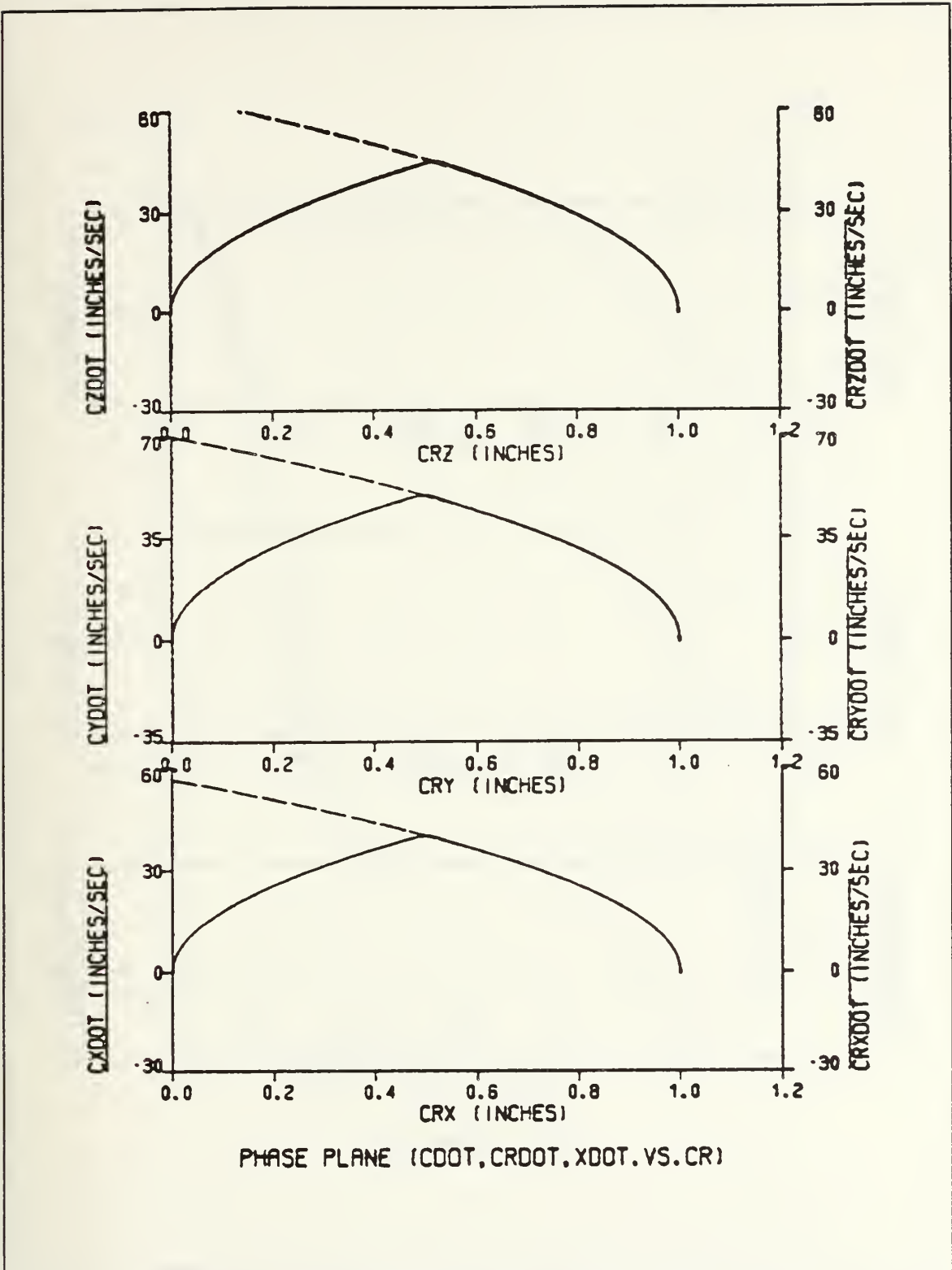
Figure 6.43   Phase Plane Trajectory for Loaded Arm
(Load=0.86 - With Gravity)
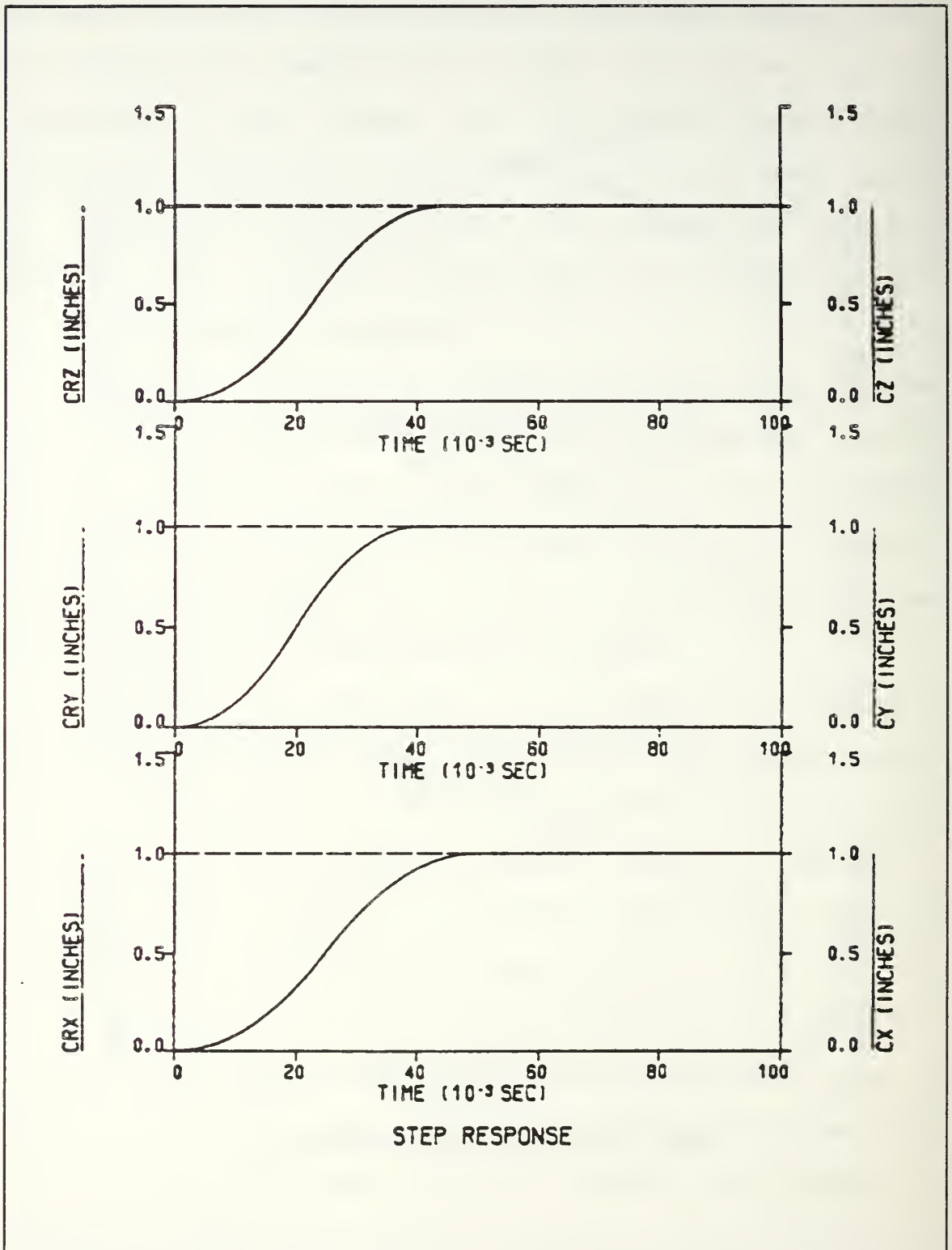
Figure 6.44   Step Response for the Loaded Arm
(Load=0.86 - With Gravity)

110
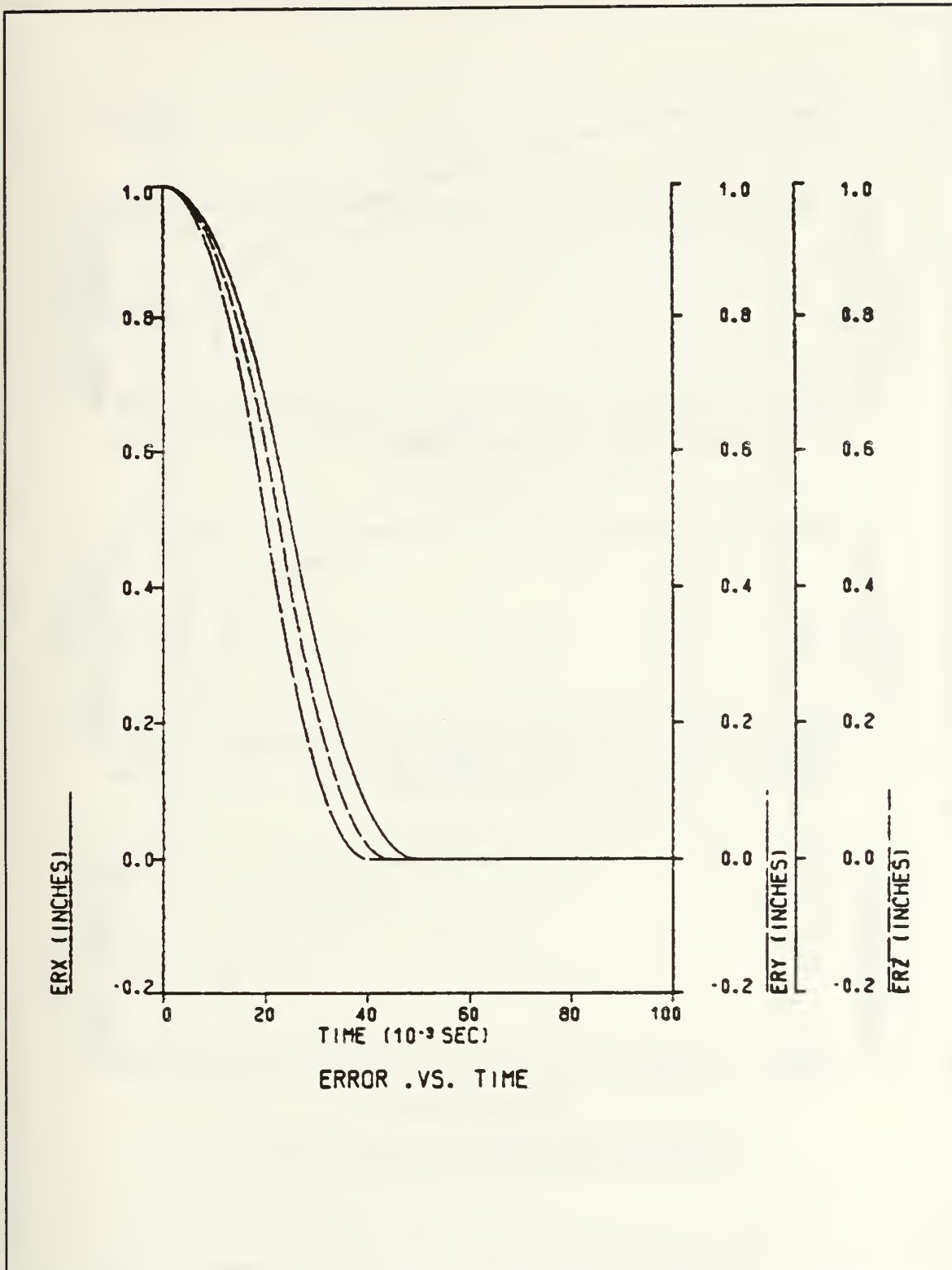
Figure 6.45 Error Curves for Loaded Arm
(Load=0.86 - With Gravity)

111
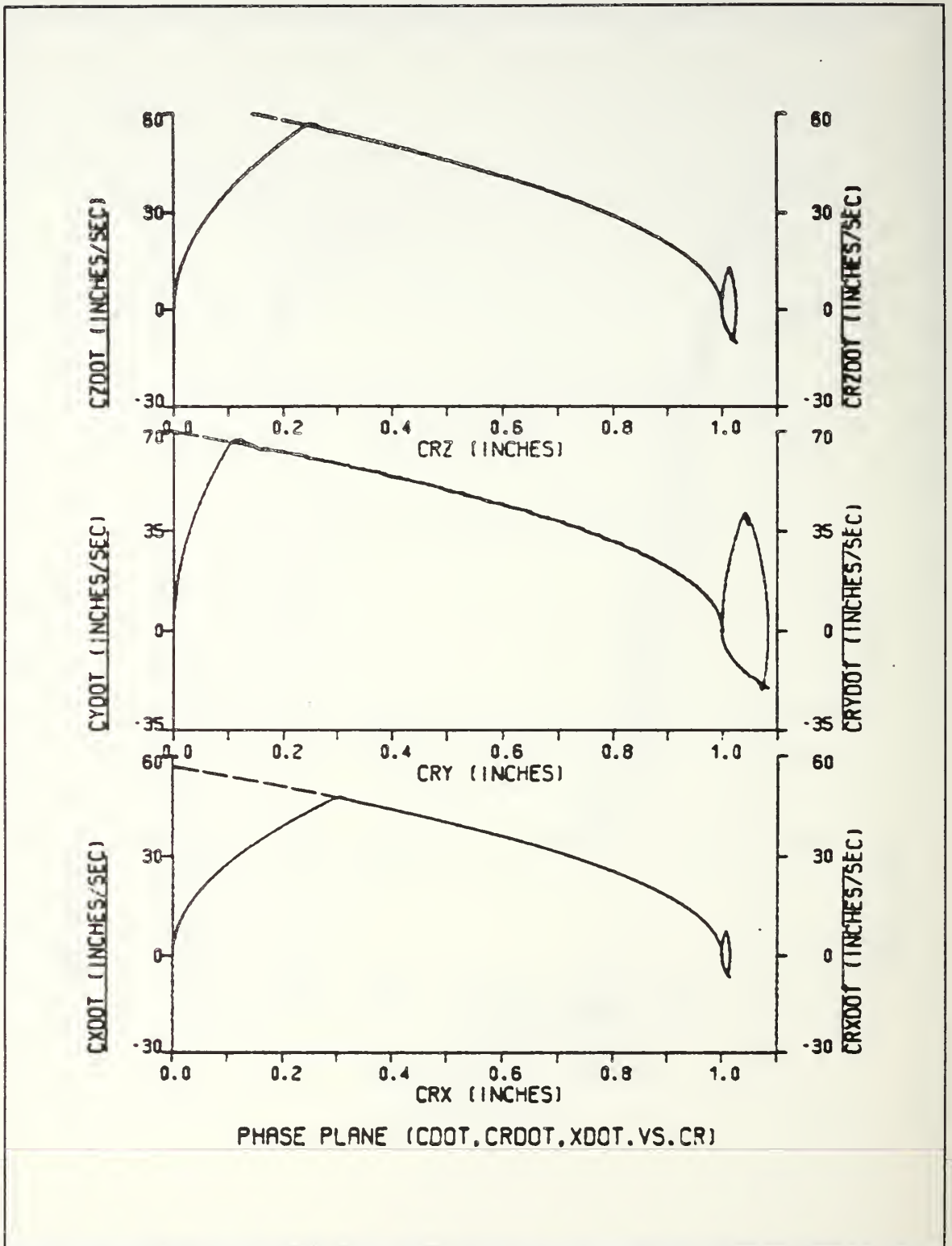
Figure 6.46   Phase Plane Trajectory with Disturbance
Applied at T=50 msec (With Gravity)
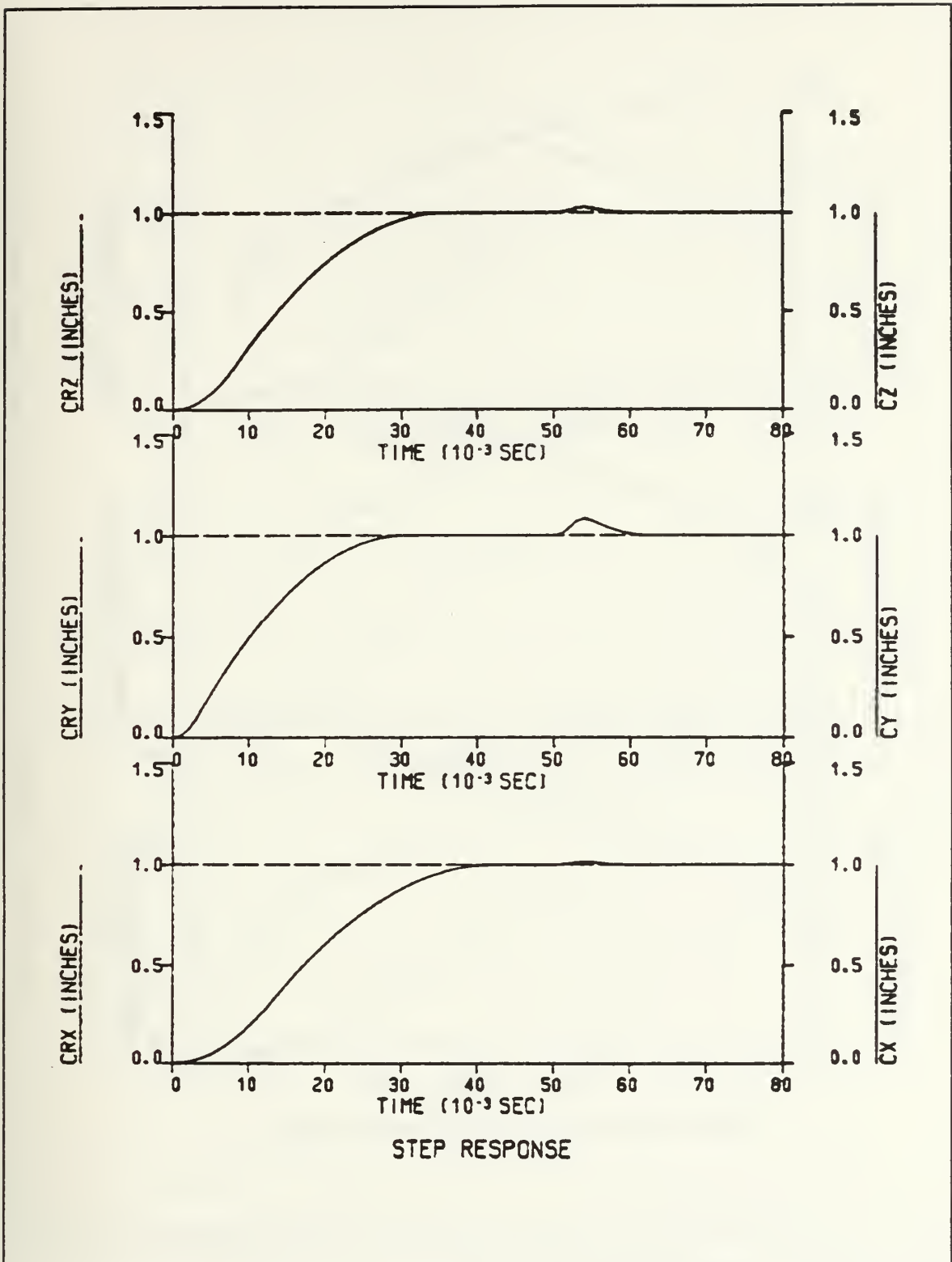
Figure 6.47   Step Response with Disturbance
Applied at T=50 msec (With Gravity)

113
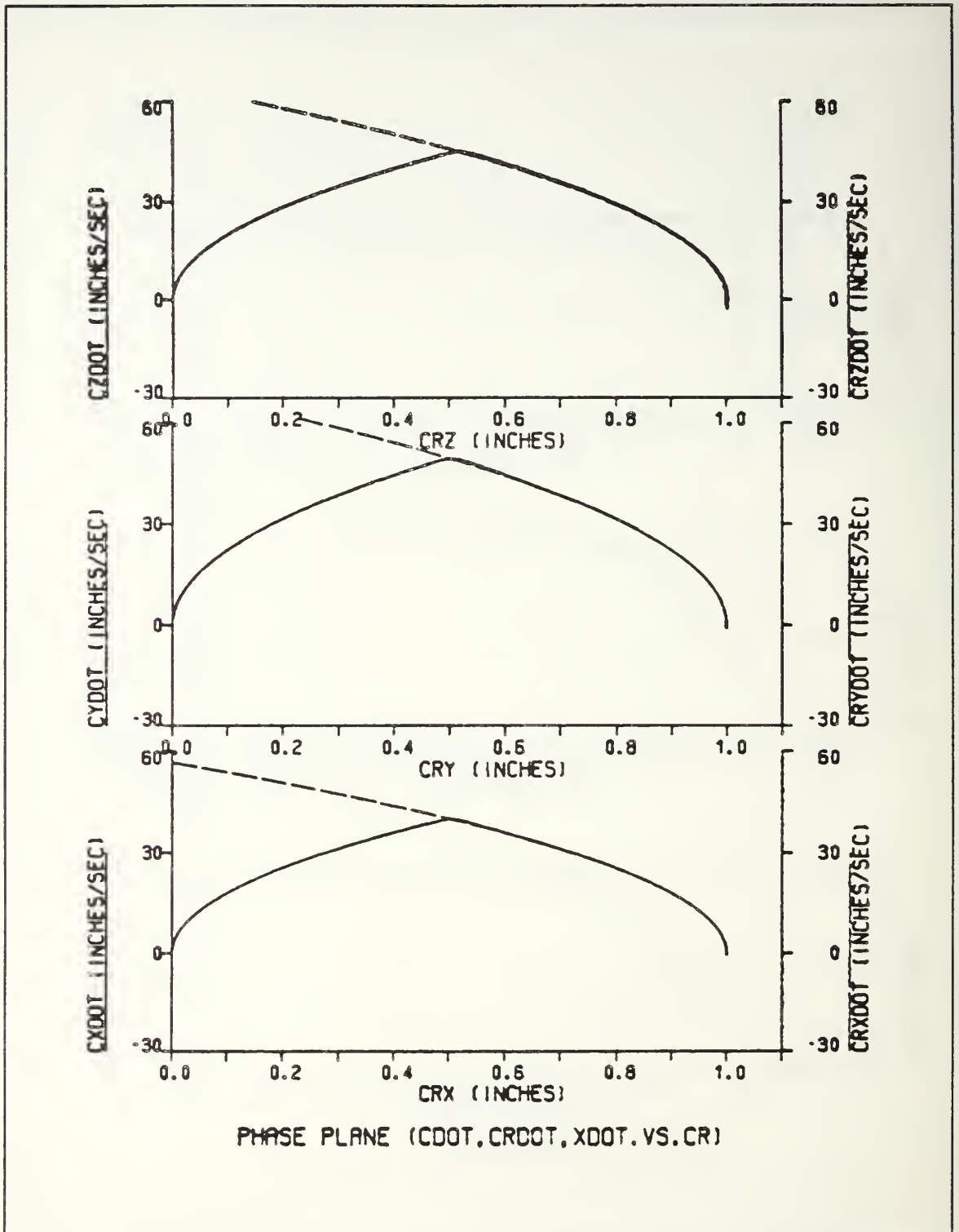
Figure 6.48   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.82 - With Gravity)

114
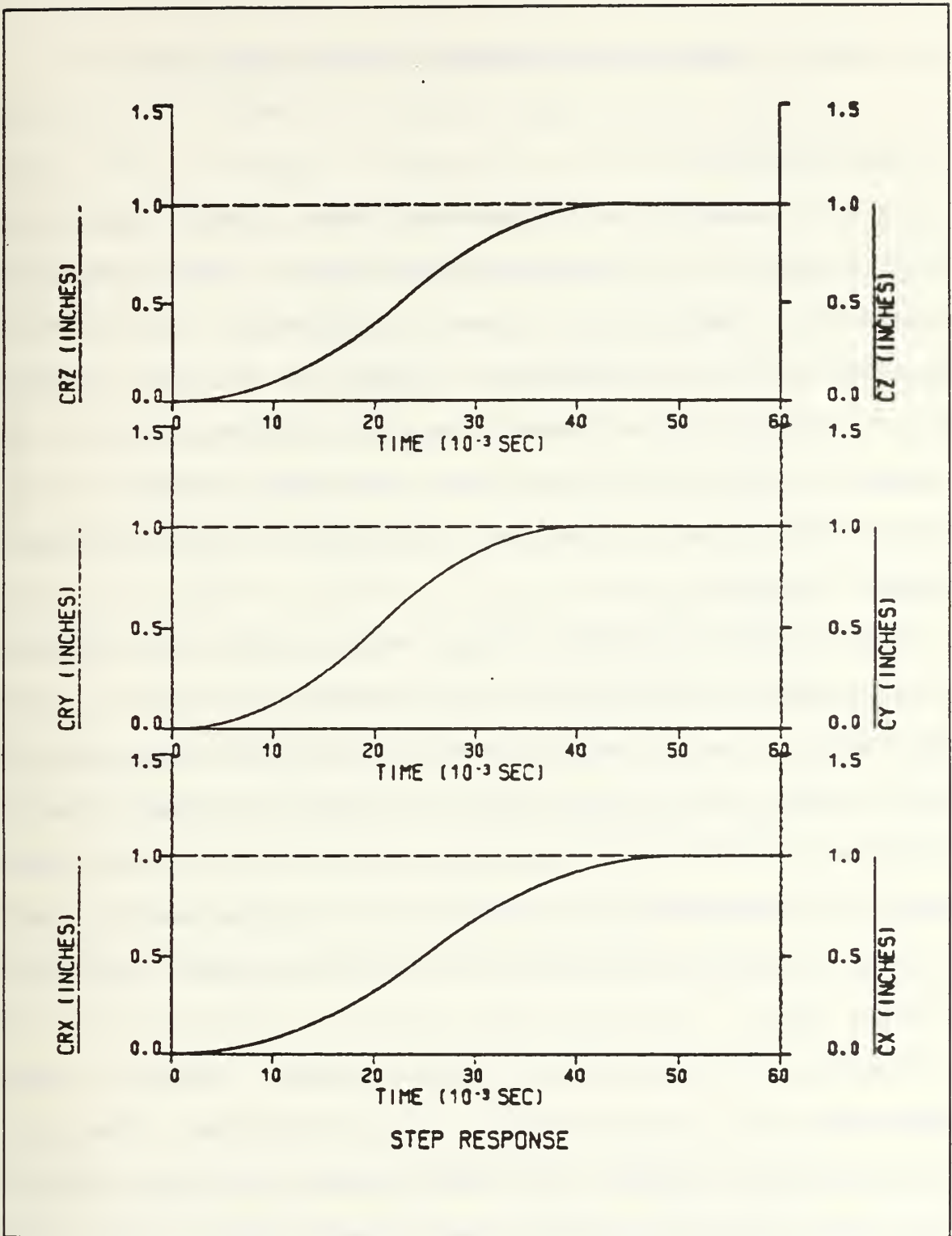
Figure 6.49   Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.82 - With Gravity)

VII. MODELLING THE THREE LINK REVOLUTE ROBOT

A. INTRODUCTION

In this chapter the mathematical model of the three link revolute robot is developed, making use of the Lagrangian mechanics as this method allows us to obtain the dynamic equations for very complicated systems in a rather simple manner. The Lagrangian formulation describes the behavior of a dynamic system in terms of work and energy stored in the system rather than forces and momentums of the individual members involved.

The dynamic equations, which relate forces and torques to positions, velocities, and accelerations, are first obtained and then solved in order to obtain the equations of motion of the robot. Given forces and torques as input these equations specify the resulting motion of the robot. This approach is computationally difficult but gives results that provide insight into both the linear and nonlinear operation of robot arms.

Using the Lagrangian method makes possible the computation of the interaction of gravitational, loading, and centripetal torques and forces dependent on the external forces, the inertias of the various components of the robot arm, and the structure of the system. Coriolis acceleration is derived as part of the overall analysis.

116

B. MODEL DEVELOPMENT

The three link robot, chosen to be tested, is depicted in Figure 7.1. There are three rotary joints connecting the three links. A gripper is mounted at the end of LINK3 which can carry a load. It is assumed that the three links are massless and the equivalent masses, m_1, m_2, m_3, are lumped at the end of the respective links. The lengths of the three links are denoted d_1, d_2, d_3 respectively, and T_1, T_2, T_3 represent the external driving torques to the joints.

Direct-drive motors are used to drive the three joints. "Direct-drive" refers to the fact that the links of the robot are coupled directly to the respective motor shafts without going through any gearing. The advantage of this scheme is that one immediately eliminates the effects of backlash, joint flexibility, etc., which arise from the use of gears. However, there are two significant negative co-sequences. First, the presence of large gear ratios effectively decouples the dynamic of each motor from the rest, which in turn simplifies the control problem [Ref.9]. When direct drive is used, this isolation effect is no longer present, and one is forced to deal with more complicated equations. The second effect concerns the actuator dynamics. In a robot with gears, even a small motion of a link results in several revolutions of the corresponding motor shaft, so that the standard method of modeling a motor as an inertia and a viscous damper is quite
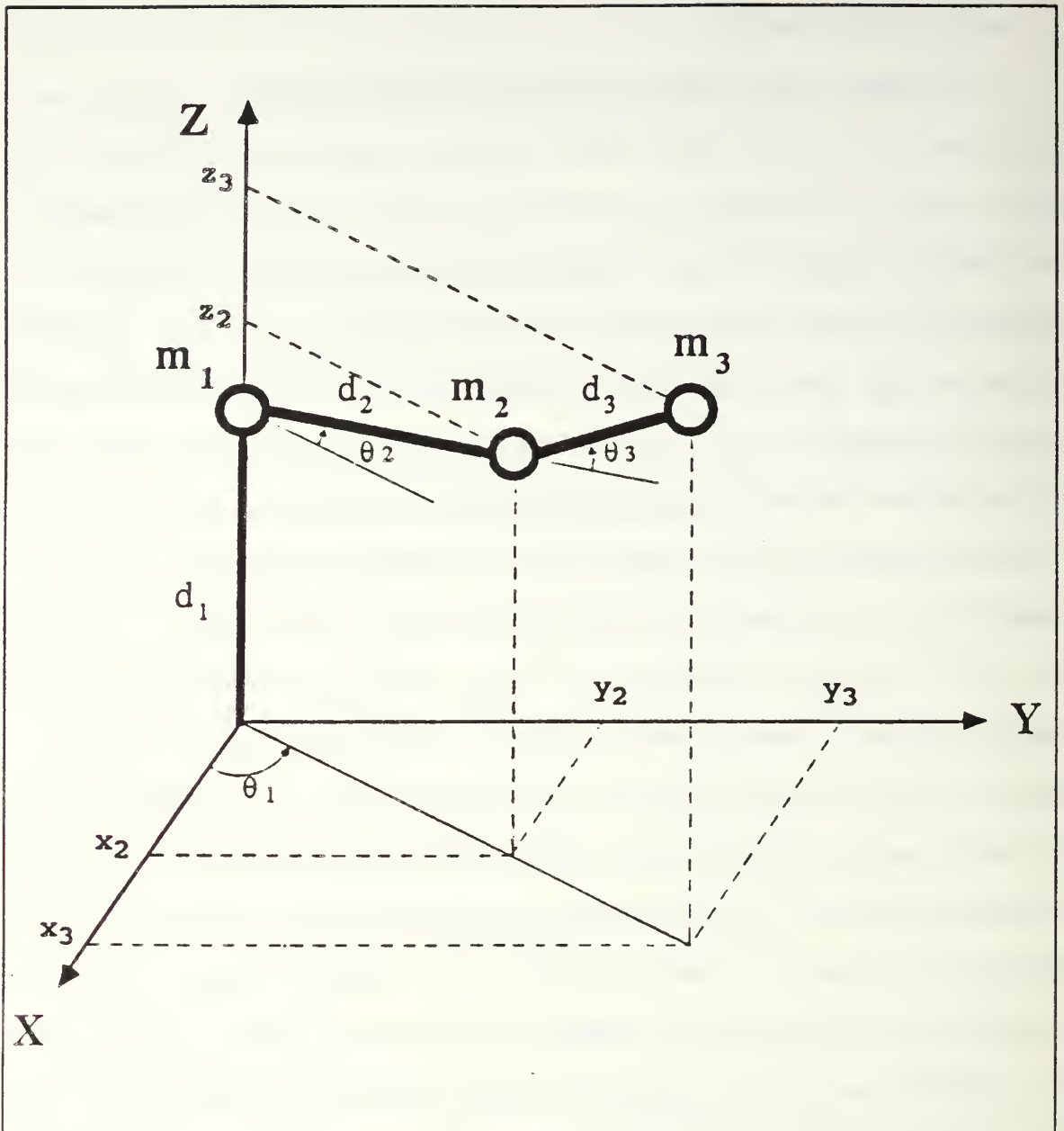
Figure 7.1   Point Mass Representation of a Three Link
Revolute Robot

reasonable. However, this is no longer reasonable when each
motor is turning only a fraction of a revolution, because
there will be significant ripples in the torque produced by
the motor.

118

C. EQUATIONS OF MOTION

The Lagrangian L is defined as the difference between the kinetic energy K and the potential energy V of the system.

$$L = K - V \qquad\qquad (7.1)$$

Generally, the dynamic equations, in terms of the coordinates used to express the kinetic and potential energy, are obtained as

$$F_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i}, \qquad i=1,\ldots,n \qquad (7.2)$$

where

    q_i = the coordinates in which the kinetic and potential energy are expressed

    $\dot{q}_i$ = the corresponding velocity

    F_i = the corresponding force or torque, depending upon whether q_i is a linear or an angular coordinate.

    n = the number of links (degrees of freedom)

These coordinates, forces, and torques are referred to as generalized coordinates, forces, and torques.

In order to apply equation 7.2 to the three link revolute robot, the rotation angles θ_1, θ_2, and θ_3 are chosen as the generalized coordinates. Therefore, F_1, F_2, and F_3 represent the generalized torques about the axes of rotation of the three links and from now on are renamed as T_1, T_2, and T_3. Solving equation 7.2 for the three links,

the following system of nonlinear, second-order differential equations has been derived and use as the basic mathematical model for the simulation of the three link revolute robot. The detailed derivation is presented in Appendix G.

$$(D_{11}+J_{m1})\theta_1 = T_1+D_{112}\dot\theta_1\dot\theta_2+D_{113}\dot\theta_1\dot\theta_3 \qquad (7.3)$$

$$(D_{22}+J_{m2})\theta_2 = T_2-D_{23}\ddot\theta_3+D_{223}\dot\theta_2\dot\theta_3+D_{233}\dot\theta_3{}^2-D_{211}\dot\theta_1{}^2-G_2 \quad (7.4)$$

$$(D_{33}+J_{m3})\theta_3 = T_3-D_{32}\ddot\theta_2-D_{311}\dot\theta_1{}^2-D_{322}\dot\theta_2{}^2-G_3 \qquad (7.5)$$

where

$$D_{11} = m_3(d_2\cos\theta_2+d_3\cos(\theta_2+\theta_3))^2+m_2d_2{}^2\cos^2\theta_2$$

$$D_{22} = (m_2+m_3)d_2{}^2+2m_3d_2d_3\cos\theta_3+m_3d_3{}^2$$

$$D_{23} = m_3d_2d_3\cos\theta_3+m_3d_3{}^2$$

$$D_{32} = m_3d_2d_3\cos\theta_3+m_3d_3{}^2$$

$$D_{33} = m_3d_3{}^2$$

$$D_{112} = 2[(m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2+m_3d_2d_3\sin(2\theta_2+\theta_3)$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]$$

$$D_{113} = 2[m_3d_2d_3\sin(\theta_2+\theta_3)\cos\theta_2$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]$$

$$D_{211} = (m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2+m_3d_2d_3\sin(2\theta_2+\theta_3)$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)$$

$$D_{223} = 2m_3d_2d_3\sin\theta_3$$

$$D_{233} = m_3d_2d_3\sin\theta_3$$

$$D_{311} = m_3d_2d_3\sin(\theta_2+\theta_3)\cos\theta_2$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)$$

$$D_{322} = m_3d_2d_3\sin\theta_3$$

$$G_2 = (m_2+m_3)gd_2\cos\theta_2+m_3gd_3\cos(\theta_2+\theta_3)$$

120

$$G_3 = m_3 g d_3 \cos(\theta_2 + \theta_3)$$

J_{m1} = 0.033 = motor inertia for JOINT 1

J_{m2} = 0.033 = motor inertia for JOINT 2

J_{m3} = 0.033 = motor inertia for JOINT 3

d_1 = 15 inches

d_2 = 10 inches

d_3 = 10 inches

m_1 = 0.268 oz/in/sec^2

m_2 = 0.227 oz/in/sec^2

m_3 = 0.041 oz/in/sec^2

g = 386.4 in/sec^2

Equations 7.3, 7.4, and 7.5 can be written in the form

$$T_1 = (D_{11} + J_{m1})\ddot{\theta}_1 \tag{7.6}$$
$$- D_{112}\dot{\theta}_1\dot{\theta}_2 - D_{113}\dot{\theta}_1\dot{\theta}_3$$

$$T_2 = (D_{22} + J_{m2})\ddot{\theta}_2 + D_{23}\ddot{\theta}_3 \tag{7.7}$$
$$+ D_{211}\dot{\theta}_1{}^2 - D_{233}\dot{\theta}_3{}^2$$
$$- D_{223}\dot{\theta}_2\dot{\theta}_3$$
$$+ G_2$$

$$T_3 = (D_{33} + J_{m3})\ddot{\theta}_3 + D_{32}\ddot{\theta}_2 \tag{7.8}$$
$$+ D_{311}\dot{\theta}_1{}^2 + D_{322}\dot{\theta}_2{}^2$$
$$+ G_3$$

In these equations, the coefficients of the form $D_{ii} + J_{mi}$ ($D_{11} + J_{m1}$, $D_{22} + J_{m2}$, $D_{33} + J_{m3}$) are known as the effective inertia at joints i. Acceleration at joint i

causes torque known as as inertial torque. The coefficients of the form D_{ij} (D_{23}, D_{32}) are known as the coupling inertia between joints i and j, as an acceleration at joint i or j causes a torque at joint j or i, respectively. The terms of the form $D_{ijj}\dot{\theta}_j^2$ ($D_{211}\dot{\theta}_1^2$, $D_{233}\dot{\theta}_3^2$, $D_{311}\dot{\theta}_1^2$, $D_{322}\dot{\theta}_2^2$) represent the centripetal forces acting at joint i due to an angular velocity at joint j. The terms of the form $D_{iij}\dot{\theta}_i\dot{\theta}_j$ ($D_{112}\dot{\theta}_1\dot{\theta}_2$, $D_{113}\dot{\theta}_1\dot{\theta}_3$, $D_{223}\dot{\theta}_2\dot{\theta}_3$) represent the Coriolis force acting at joint i due to angular velocities at both joints i and j. The terms of the form G_i represent the gravitational forces acting at joint i. [Ref.10]

VIII. THE ADAPTIVE MODEL FOR THE THREE LINK REVOLUTE ROBOT

A. INTRODUCTION

The same second order model developed in Chapter III and used in the rectangular robot, will be operated again, open loop, to drive the servo motors of the joints of the revolute robot and control its output, C. For the reasons described in chapter IV the model states and gain constant must be adapted in such a way that the ideal motors imitate the real ones. Since the adaptive model of Figure 4.8, used for the rectangular robot, is used for the revolute robot too, the adaptive algorithm to update the model states and gain constant is identical with the one obtained in Chapter IV and its derivation is not repeated.

In this chapter, the selection of the servo motors is first discussed and then the initial values of the model gain constants (K_m) are obtained.

B. SELECTION OF POSITIONING SERVO MOTORS

The same permanent magnet servo motor used for the rectangular robot is also used for the revolute robot and its parametric data are listed in Table 1 [Ref.6]. The links of the robot are coupled directly to the respective motor shafts without going through any gearing ("direct-drive"

motors). The same motor drive is used for the three joints with the same power supply.

C. CALCULATION OF GAIN CONSTANTS (K_m)

To calculate the transfer functions of the servo motors, the effective joint inertias must be first obtained as following

$$J_{EF1} = D_{11}+J_{m1} \qquad\qquad (8.1)$$

$$J_{EF2} = D_{22}+J_{m2} \qquad\qquad (8.2)$$

$$J_{EF3} = D_{33}+J_{m3} \qquad\qquad (8.3)$$

where D_{11}, D_{22} and D_{33} are given in Chapter VII. Substitution of the parameters, for $\theta_2=0°$ and $\theta_3=0°$ gives

$$J_{EF1} = 39.133 \quad oz.in.sec^2$$

$$J_{EF2} = 39.133 \quad oz.in.sec^2$$

$$J_{EF3} = 4.133 \quad oz.in.sec^2$$

Plugging these results and parameters from Table 1 into equation 3.1, the transfer functions of the three servo motors are obtained as

$$G_1(s) = \frac{9.88}{s(s/0.041+1)(s/9100+1)} \quad rad/volt \qquad (8.4)$$

$$G_2(s) = \frac{9.88}{s(s/0.041+1)(s/9100+1)} \quad rad/volt \qquad (8.5)$$

$$G_3(s) = \frac{9.88}{s(s/0.39+1)(s/9100+1)} \quad rad/volt \qquad (8.6)$$

Using these results the gain constants of the model motors (K_m) were determined. The Open Loop Bode plots of the three servo motors are depicted in Figures 8.1, 8.3 and 8.5, respectively. These plots show that the -40 db/decade slope intersects the 0 db axis at ω_1=0.65 rad/sec for the JOINT1 motor, ω_2=0.65 rad/sec for the JOINT2 motor and ω_3=2 rad/sec for the JOINT3 motor. Making use of the above frequencies as gain crossover frequencies for the second order ideal motors, we obtain

K_{m1} = 0.65^2 = 0.4225 rad/volt

K_{m2} = 0.65^2 = 0.4225 rad/volt

K_{m3} = 2^2 = 4 rad/volt

The Open Loop Bode plots of the K_m/s_2 ideal motors are depicted in Figures 8.2, 8.4 and 8.6.
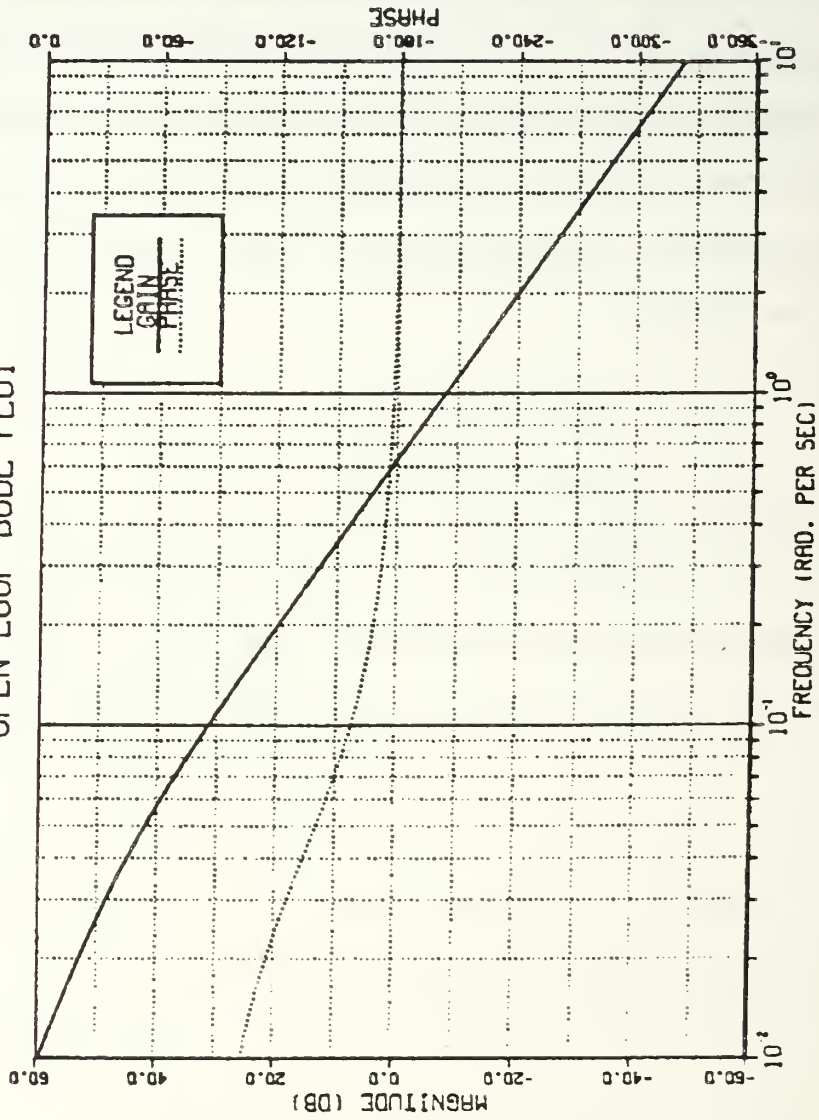
Figure 8.1 Open Loop Bode Plot of the JOINT Servo Motor

126
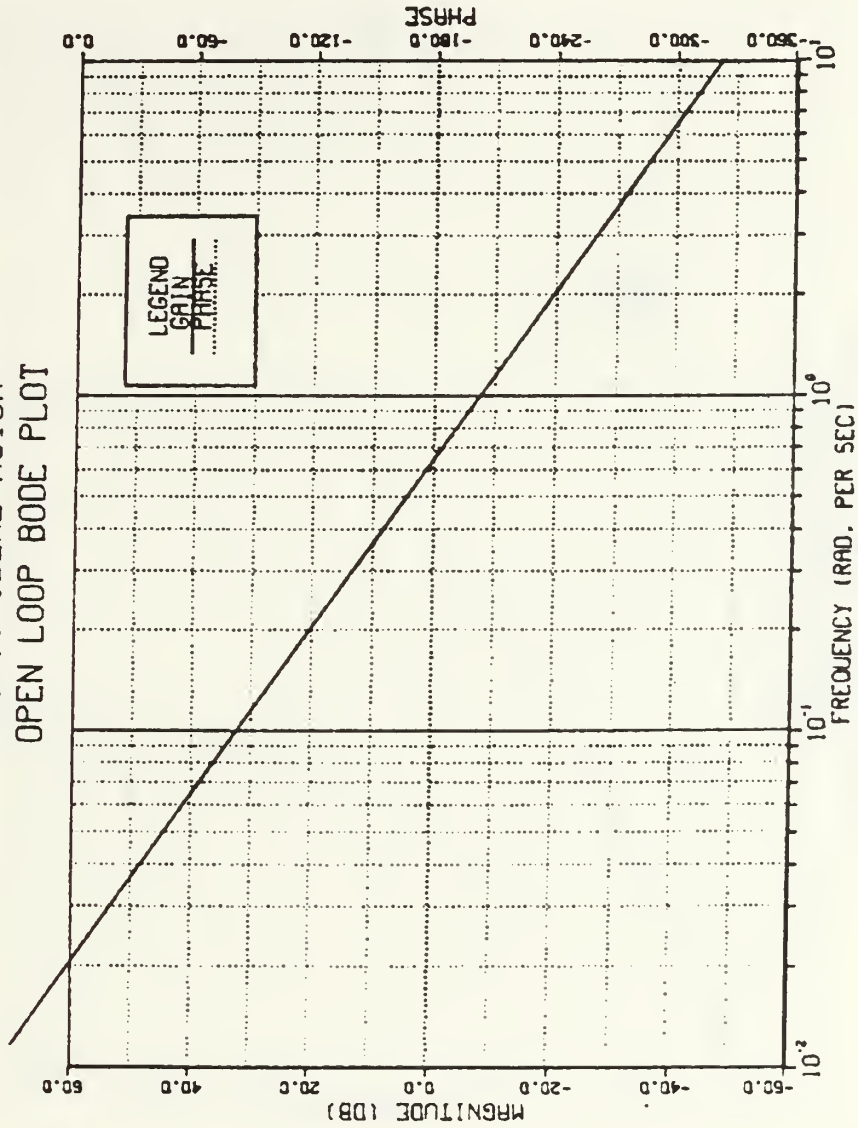
Figure 8.2   Open Loop Bode Plot of the K_{m1}/s^2 Motor
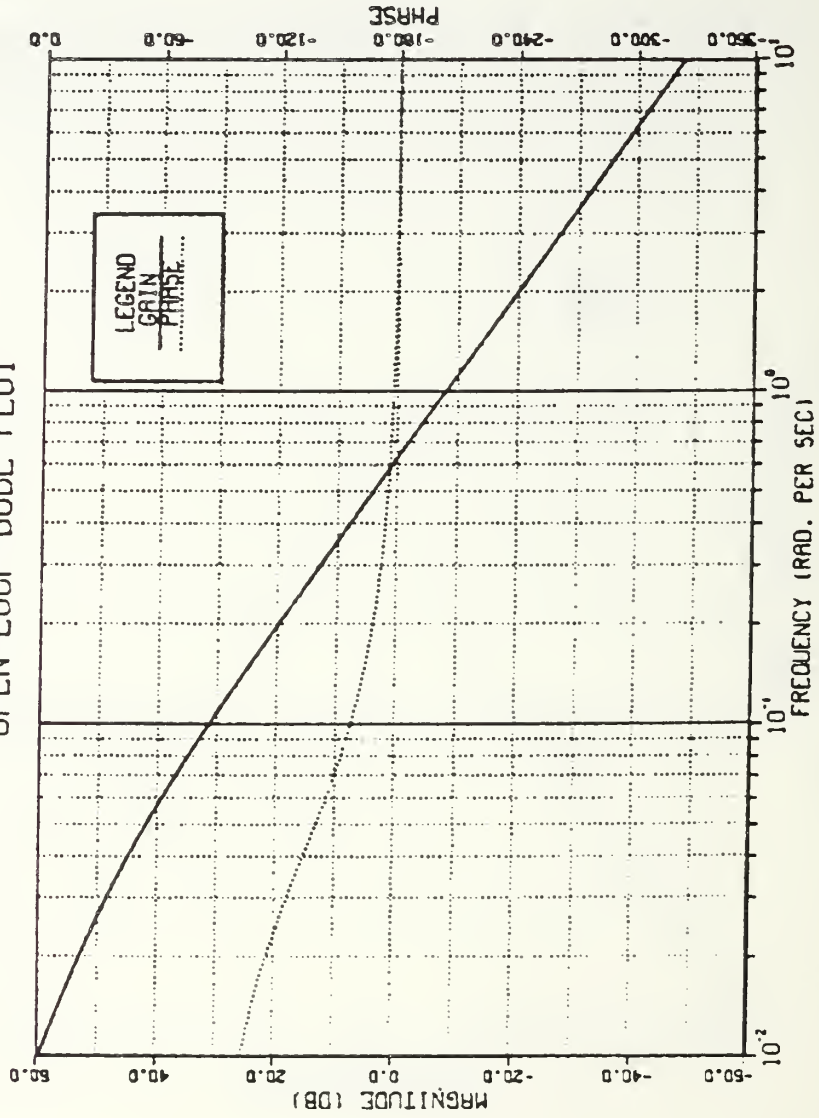
Figure 8.3 Open Loop Bode Plot of the JOINT2 Servo Motor

128
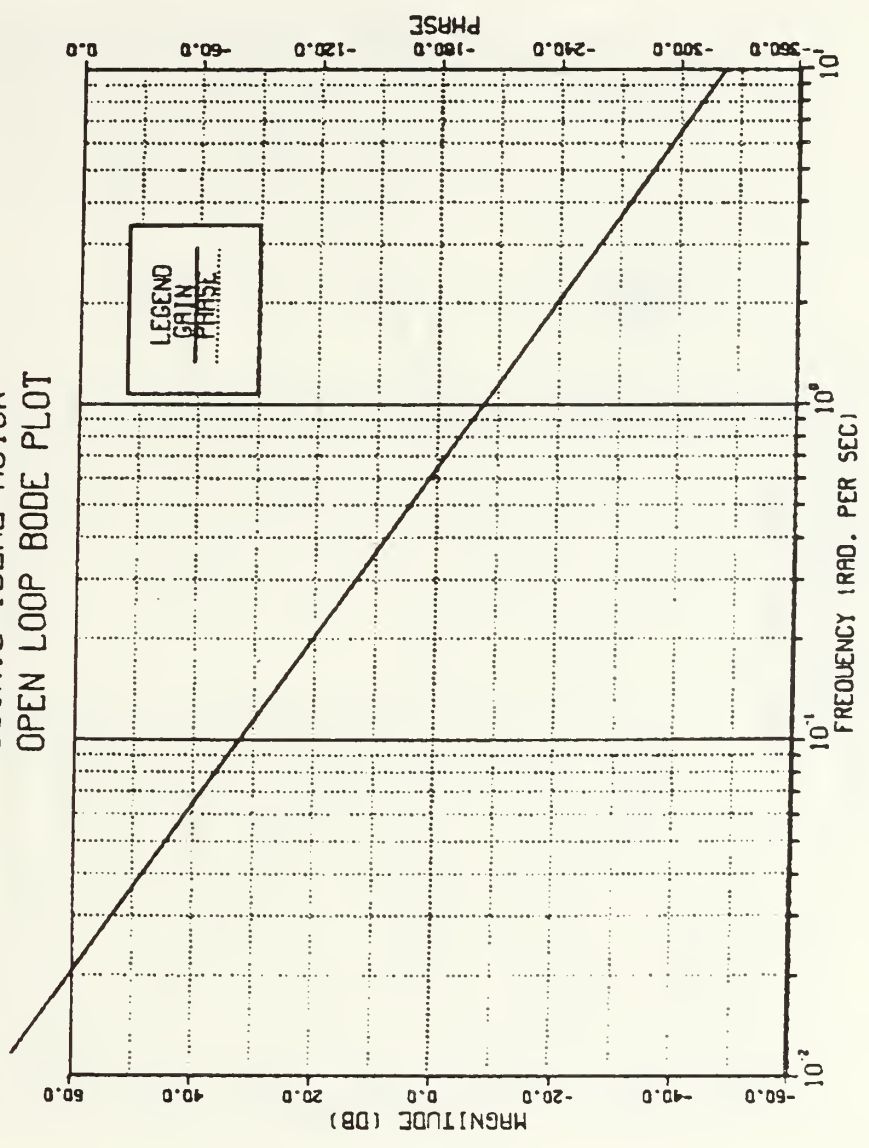
Figure 8.4   Open Loop Bode Plot of the K_{m2}/s^2 Motor

129
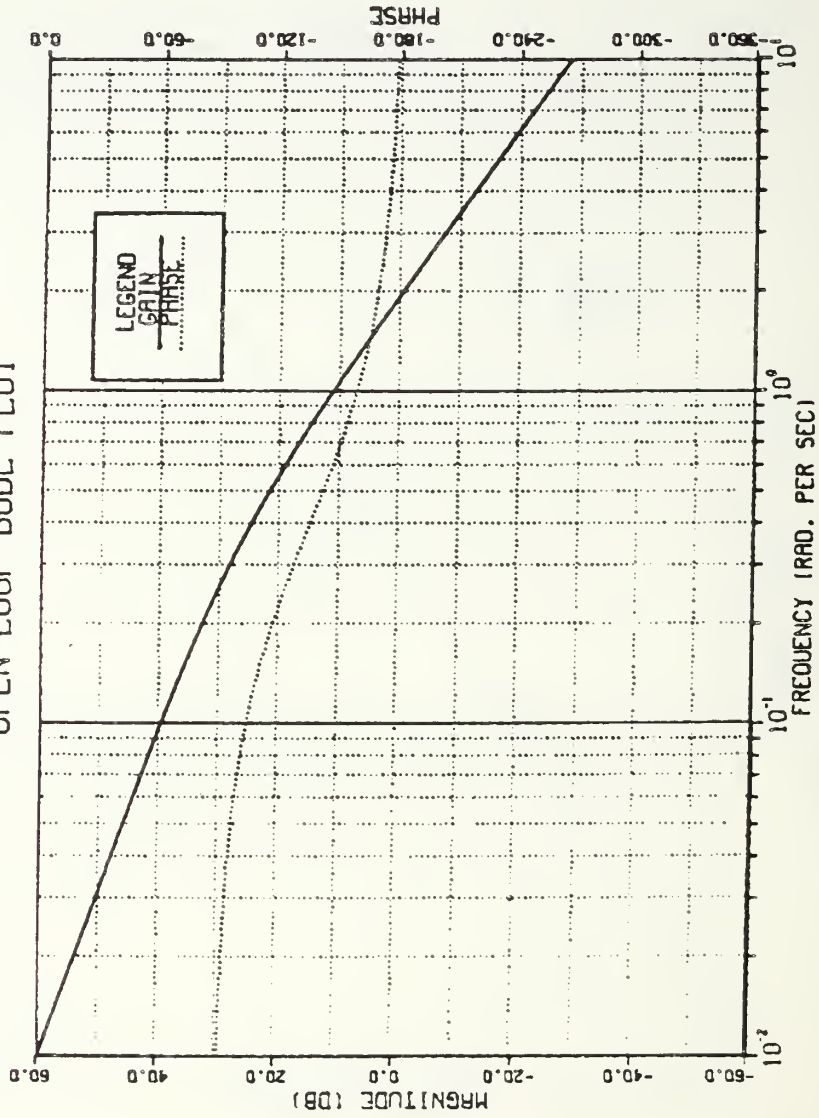
Figure 8.5   Open Loop Bode Plot of the JOINT3 Servo Motor
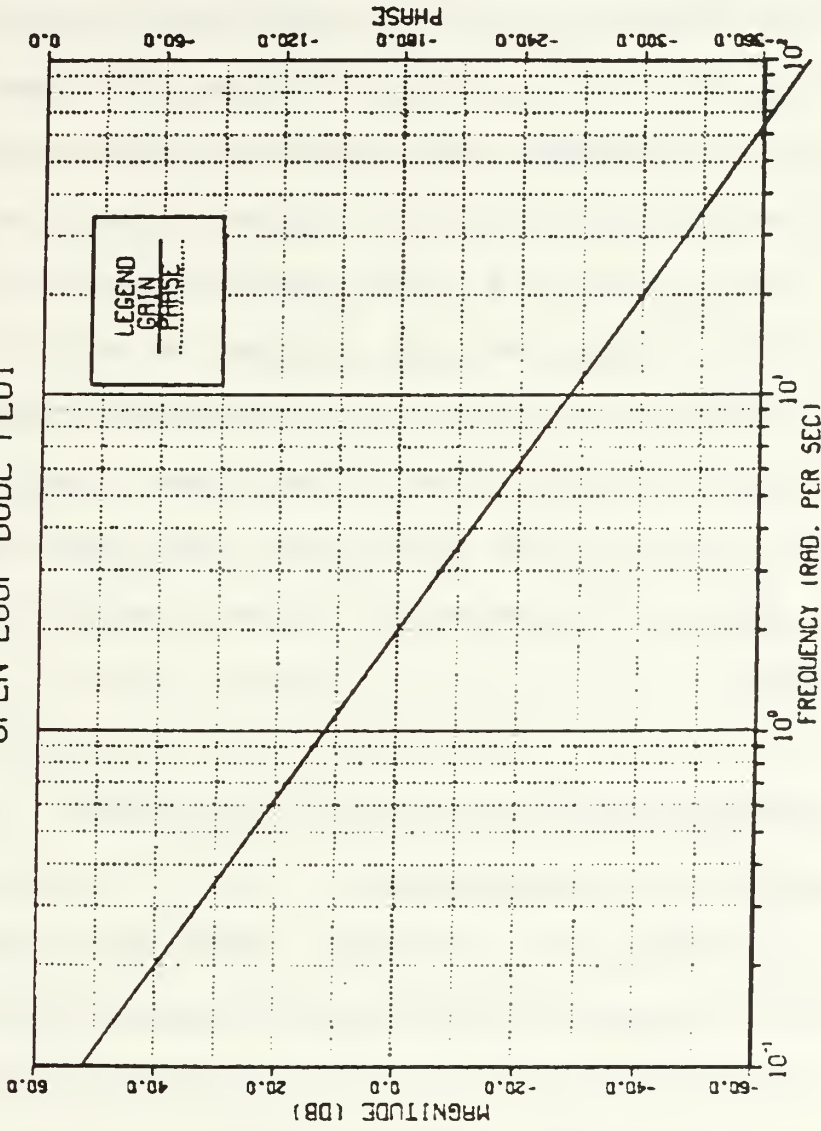
Figure 8.6   Open Loop Bode Plot of the K_{m3}/s^2 Motor

131

IX. SIMULATION OF THE THREE LINK REVOLUTE ROBOT

A. PLANNING THE SIMULATION STUDIES

In this chapter, the model developed in Chapter VII is tested for the Voltage Source Drive. First the model is tested in a gravity-free environment and then the same tests are repeated with gravitational torques taken into account.

The tested arm is a point-to-point arm and for this thesis a free work space is assumed. A unit step input command is applied in all joints at the same time to assure realistic interaction between the three links. For each simulation run the phase plane plot, the step response and the error curves are obtained in order to study the behavior of the system.

B. SIMULATION STUDIES OF THE ADAPTIVE SYSTEM

1. Gravity-free Environment

The three link revolute robot in a gravity-free environment is tested first under different load conditions, then for disturbance rejection and finally the robustness of the system is tested for a slight variation of the parametric data of the servo motors.

a. Different Load Conditions

The adaptive model is tested first without carrying any load and then under different load conditions,

132

using the DSL/VS simulation program listed in Appendix H. Figures 9.1, 9.2 and 9.3 illustrate the phase plane plot, the step response and the error curves, respectively, for the unloaded arm.

Figure 9.1 shows that the model and servo motors have good curve following characteristics for all joints. Observing the phase plane plots for JOINT1 and JOINT2, one would expect JOINT2 to reach the commanded position faster than JOINT1. Contradictorily, the step response of Figure 9.2 shows that JOINT1 reaches steady state a little earlier than JOINT2. The explanation is that the upward motion of the LINK3 creates an interaction torque on JOINT2 servo motor and it takes time (about 40 msec) for the JOINT2 to overcome the reaction torque and start accelerating. The time required for the movement of the arm to be completed is the time required by the slowest link (LINK2) to complete its movement. From Figure 9.2 this time is read 335 msec. The error curves of Figure 9.3 show zero steady state error for the three links, but a more accurate observation of the simulation data reveals a steady state accuracy of the order of 10^{-5} inches.

The simulation results of the loaded arm under different load conditions are depicted in Figures 9.4-9.15. In these figures the effects of the added load can be observed. The phase plane plot of Figure 9.4 shows that at the beginning of the motion, JOINT2 servo motor develops a

Figure 9.1   Phase Plane Trajectory for Unloaded Arm
(No Gravity)

134
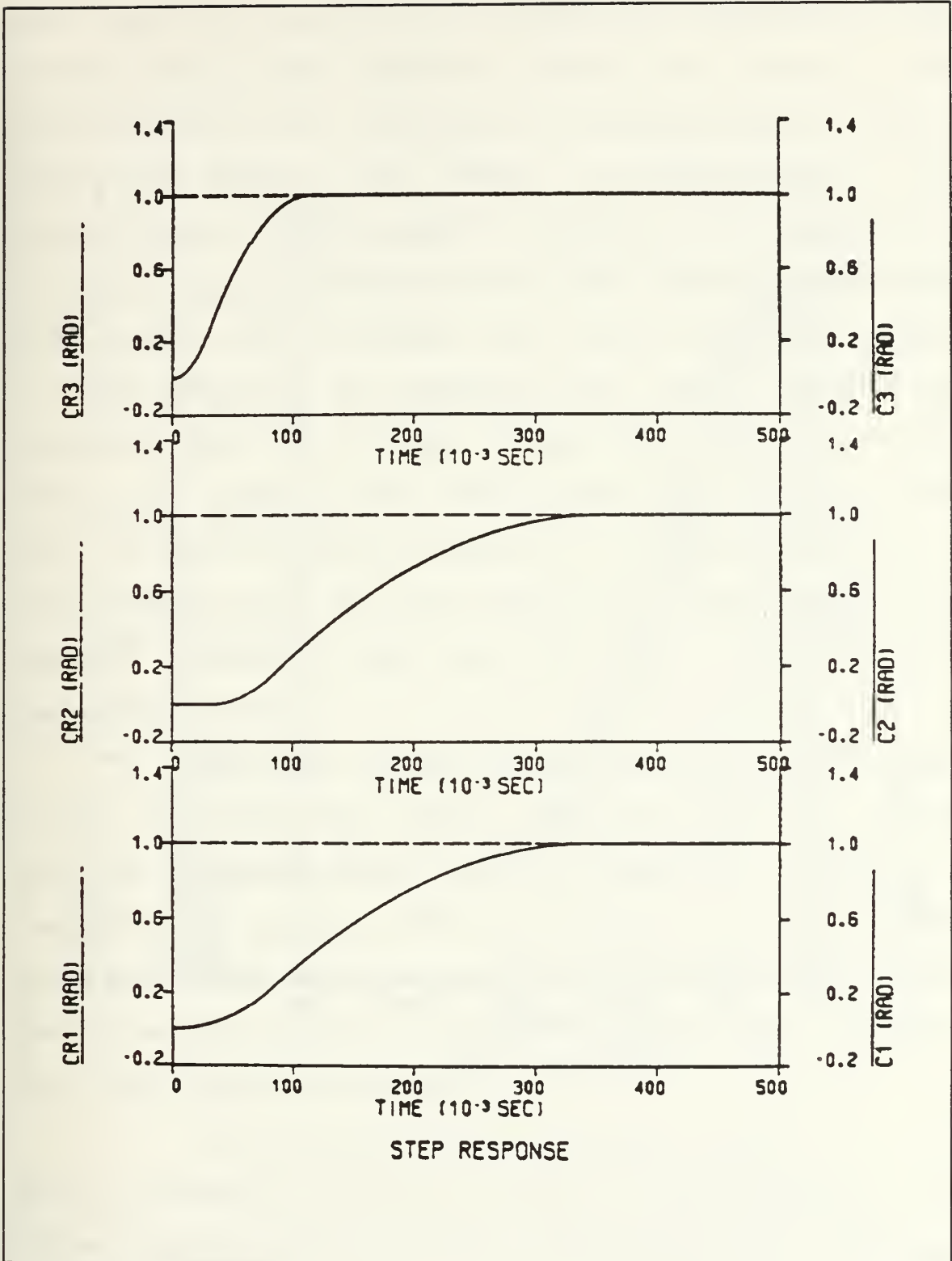
Figure 9.2   Step Response for Unloaded Arm
(No Gravity)
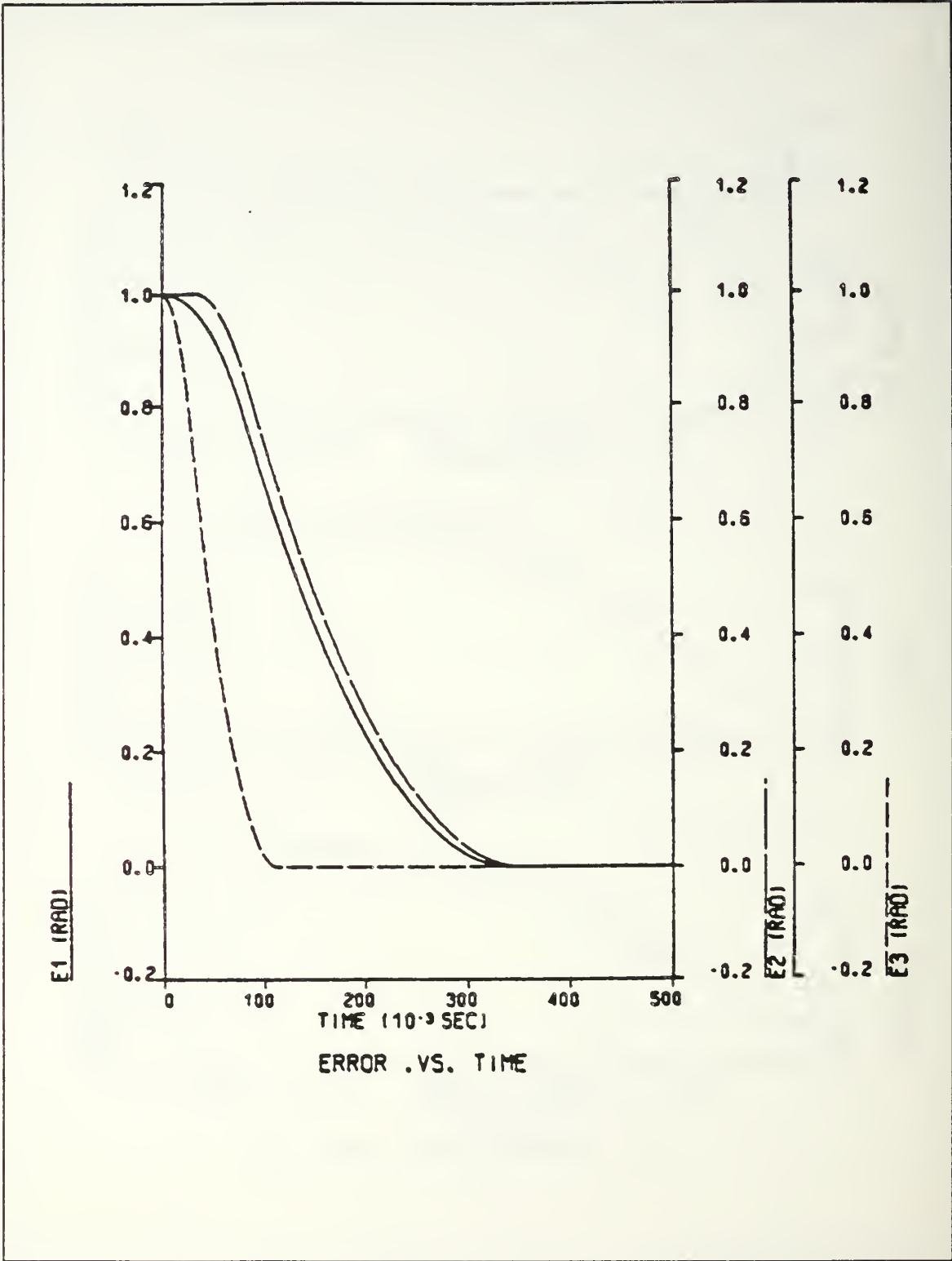
Figure 9.3   Error Curves for Unloaded Arm
(No Gravity)

136

small negative velocity (in the opposite direction of the desired move). This negative velocity is caused by the torque created on the JOINT2 due to coupling inertia between JOINT2 and JOINT3. This torque is proportional to the acceleration of JOINT3 which is large at the beginning of the move. As the acceleration of JOINT3 starts decreasing, JOINT2 servo motor overcomes this torque and starts moving in the direction of the commanded move. When it reaches the commanded curve it can not follow the curve due to the interactive torque of the still accelerating JOINT3 servo motor. As soon as JOINT3 reaches the commanded position, it starts decelerating and the JOINT2 servo motor catches the commanded velocity curve and follows it down to the commanded position.

The step response and the error curves of Figure 9.5 and 9.6 show the initial movement of JOINT2 in the opposite direction. The other two joints move directly to the commanded position. The joint mostly affected by the load is JOINT2 because the added load increases the coupling inertia between JOINT2 and JOINT3 and consequently increases the torque applied on JOINT2.

If the added load increases Figure 9.7 shows that the negative velocity of JOINT2 at the beginning of the motion becomes greater and the settling time for JOINT2 and JOINT3 increase, as depicted in Figure 9.8 and 9.9. But as more load is added, the Coriolis force acting on JOINT1, due
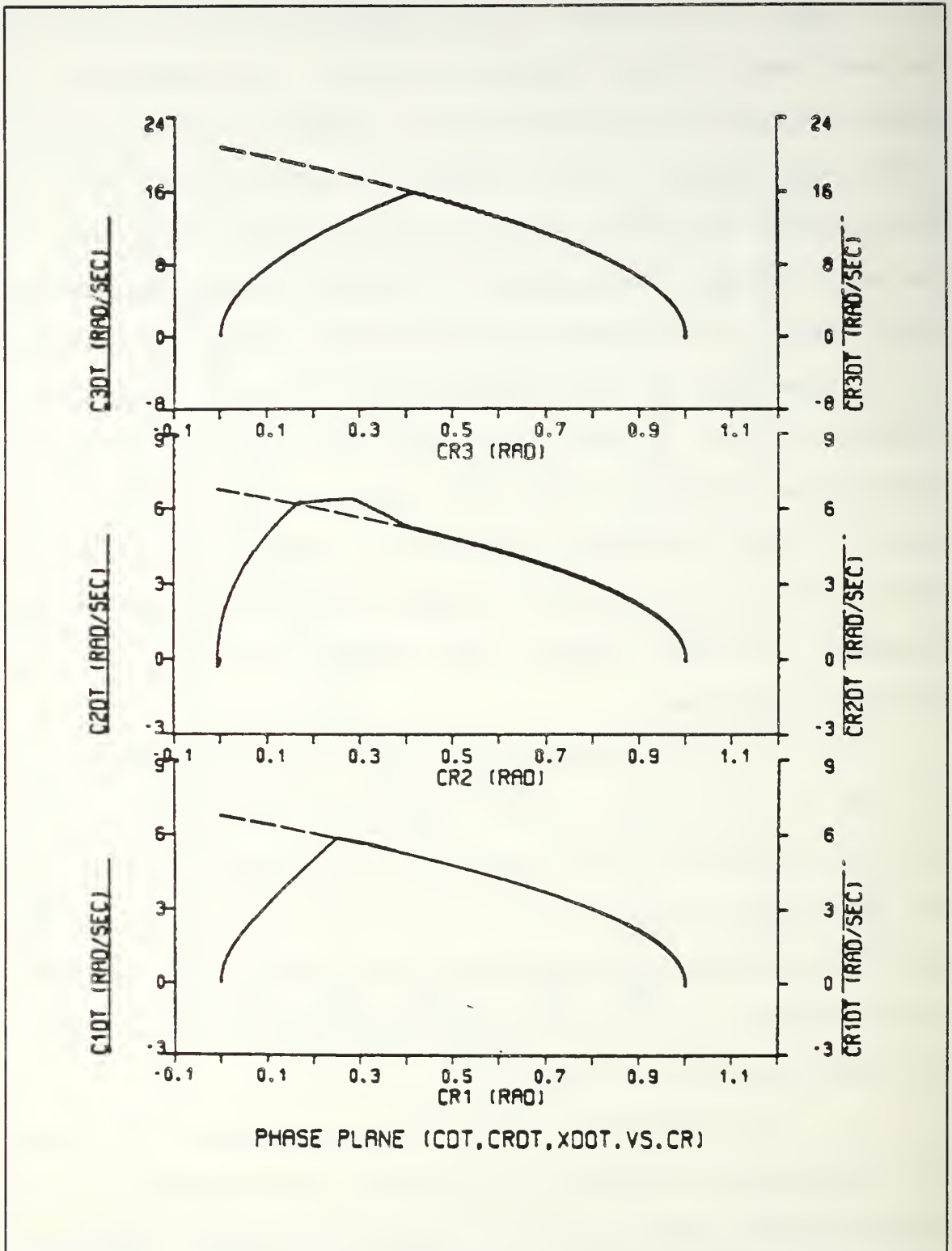
Figure 9.4   Phase Plane Trajectory for Loaded Arm
(Load=0.04 - No Gravity)
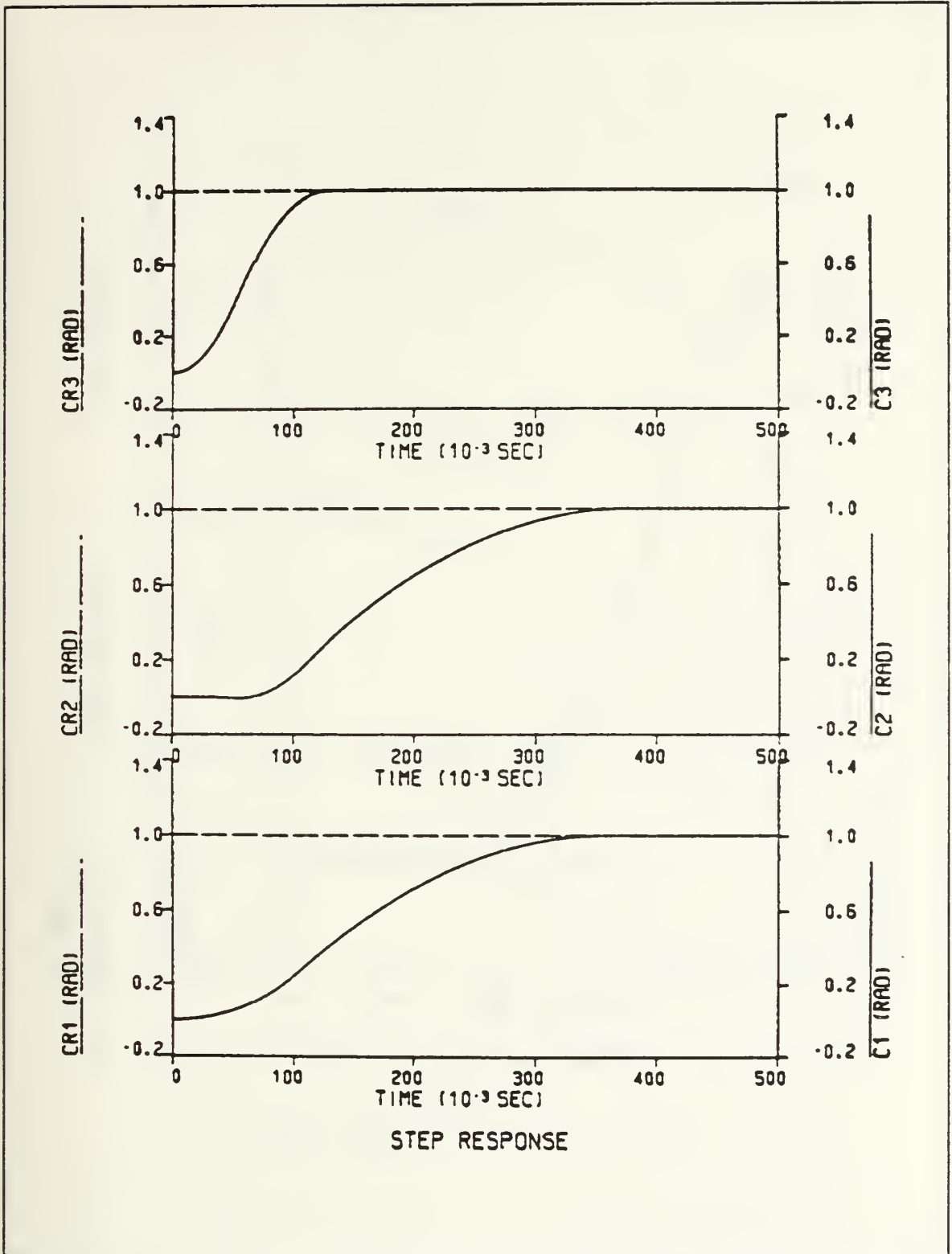
Figure 9.5   Step Response for Loaded Arm
(Load=0.04 - No Gravity)
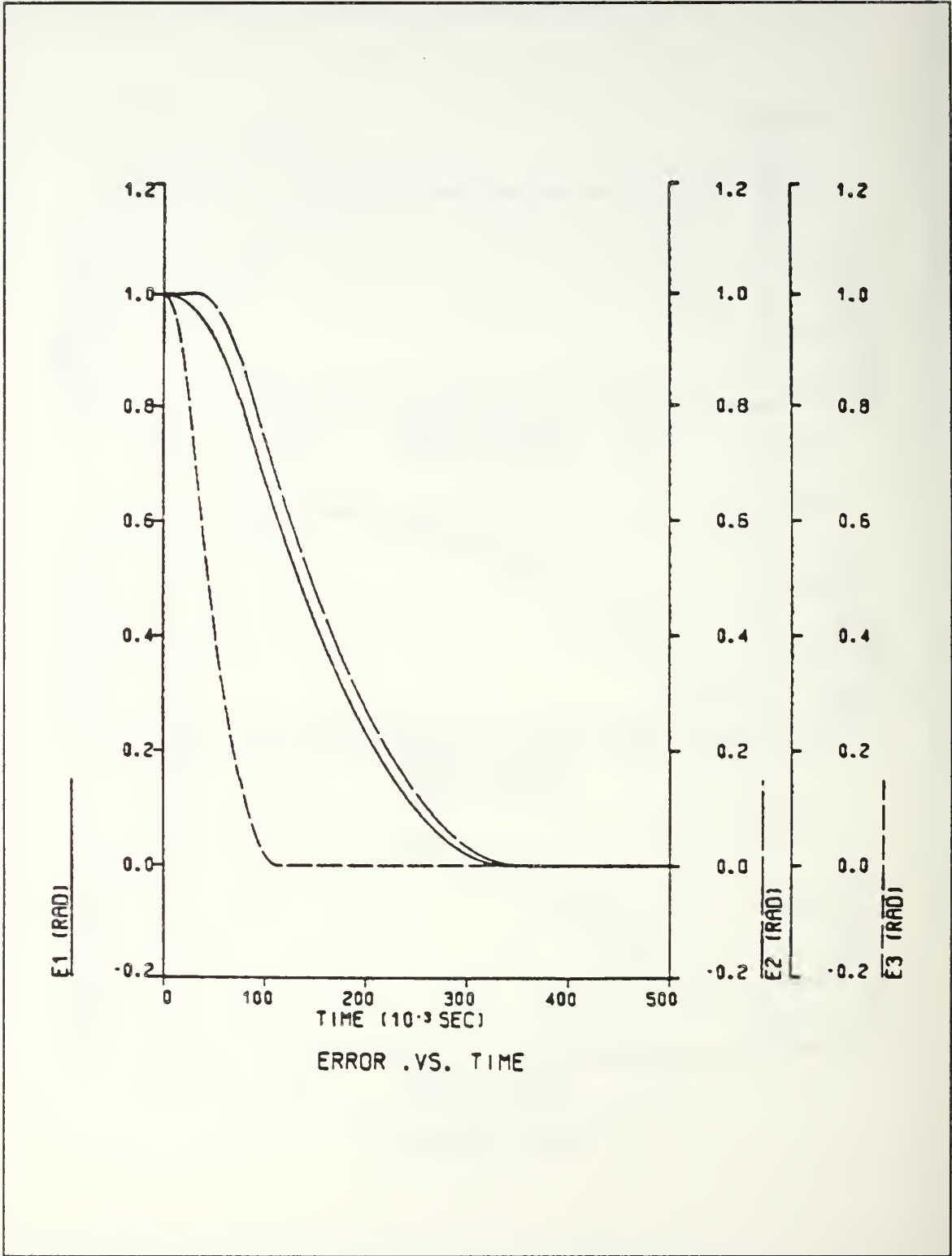
Figure 9.6   Error Curves for Loaded Arm
(Load=0.04 - No Gravity)

140
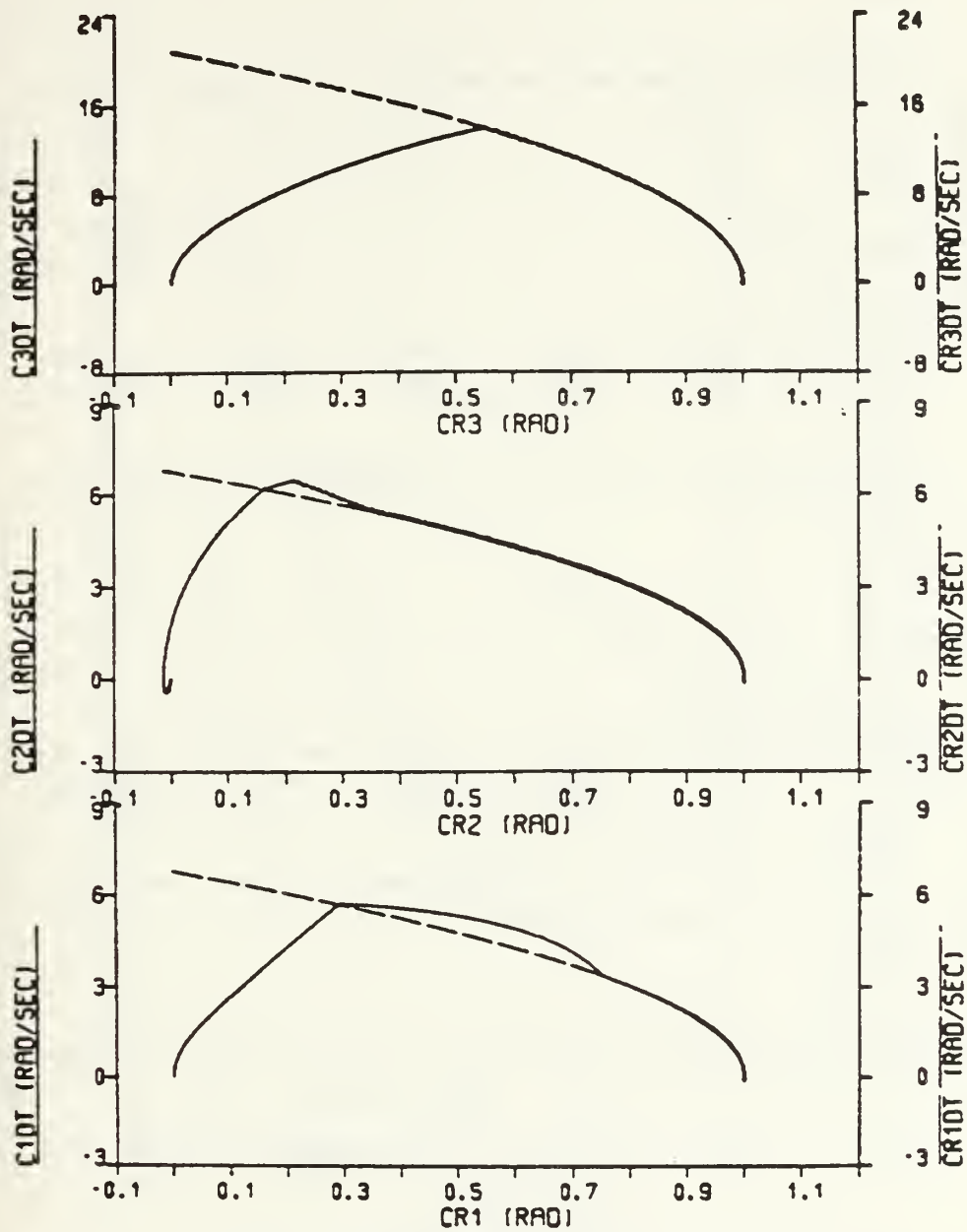
Figure 9.7   Phase Plane Trajectory for Loaded Arm
(Load=0.1 - No Gravity)

141
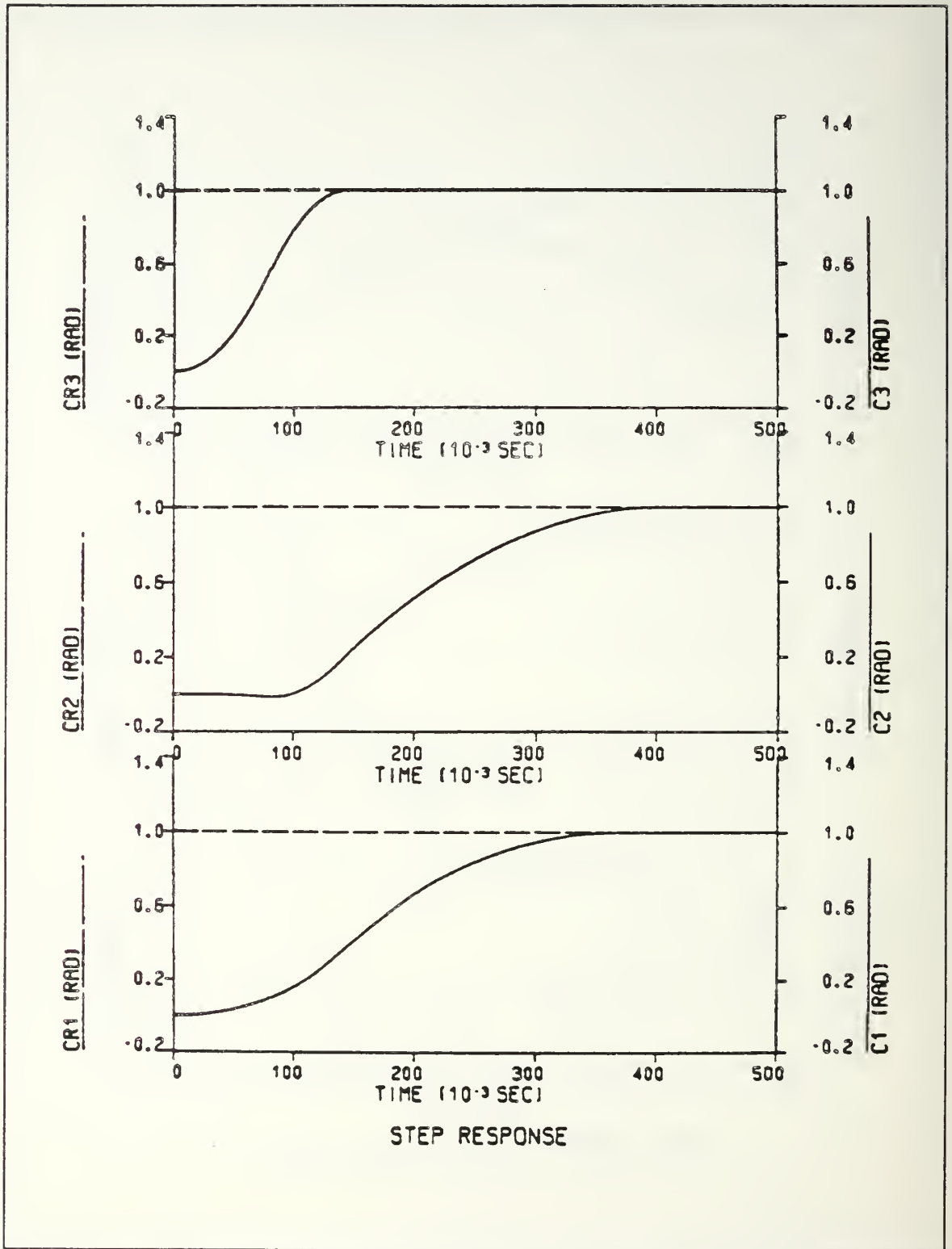
Figure 9.8   Step Response for Loaded Arm
(Load=0.1 - No Gravity)
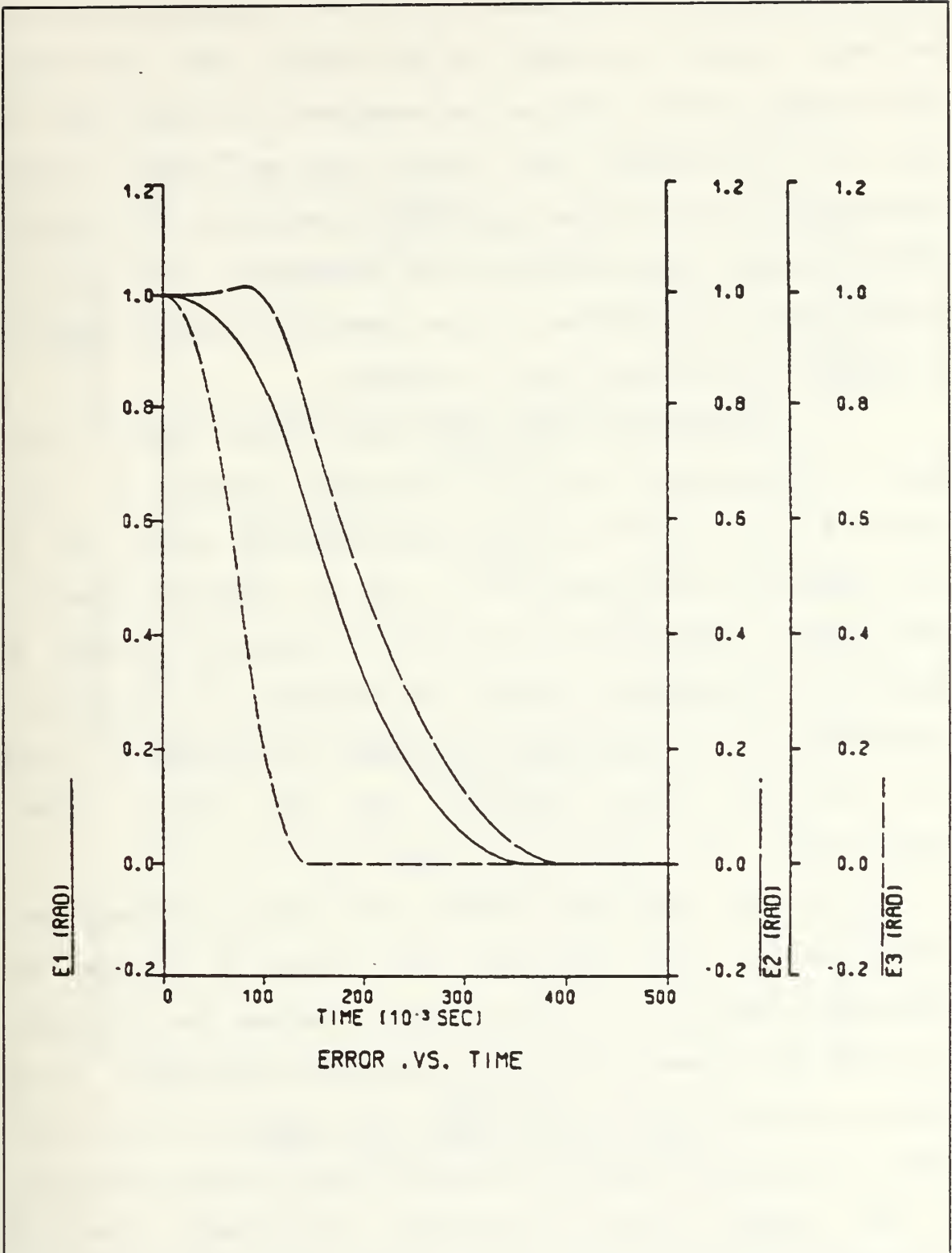
Figure 9.9   Error Curves for Loaded Arm
(Load=0.1 - No Gravity)

143

to the angular velocities at all joints, increases and the increased torque at JOINT1 is the reason that the JOINT1 servo motor can not follow the commanded velocity curve as soon as it reaches the curve. But as JOINT3 starts decelerating, the torque at JOINT1 decreases and finally JOINT1 catches and follows the commanded curve to the desired position. Because of the increased angular velocity of JOINT1 its settling time is reduced.

Figure 9.10 show that the critical load for this robot configuration is 0.157 $oz/in/sec^2$ because for this value of load, JOINT1 catches the commanded curve just at the commanded position. Also it can be observed from the same figure that while the initial negative velocity of JOINT2 is decreased, JOINT2 is capable to follow the commanded curve as soon as it reaches it. Figures 9.11 and 9.12 are the step response and the error curves, respectively, for this critical load.

For any load greater than the critical one, JOINT1 catches the commanded curve after it reaches the commanded position. So, when JOINT1 reaches the commanded position its velocity is not zero but still has a value which is proportional to the load. As a result of this fact, JOINT1 overshoots before it reaches steady state condition. As the added load is further increased (load=0.2 $oz/in/sec^2$), JOINT2 starts overshooting too, as illustrated in Figures 9.13 - 9.15.
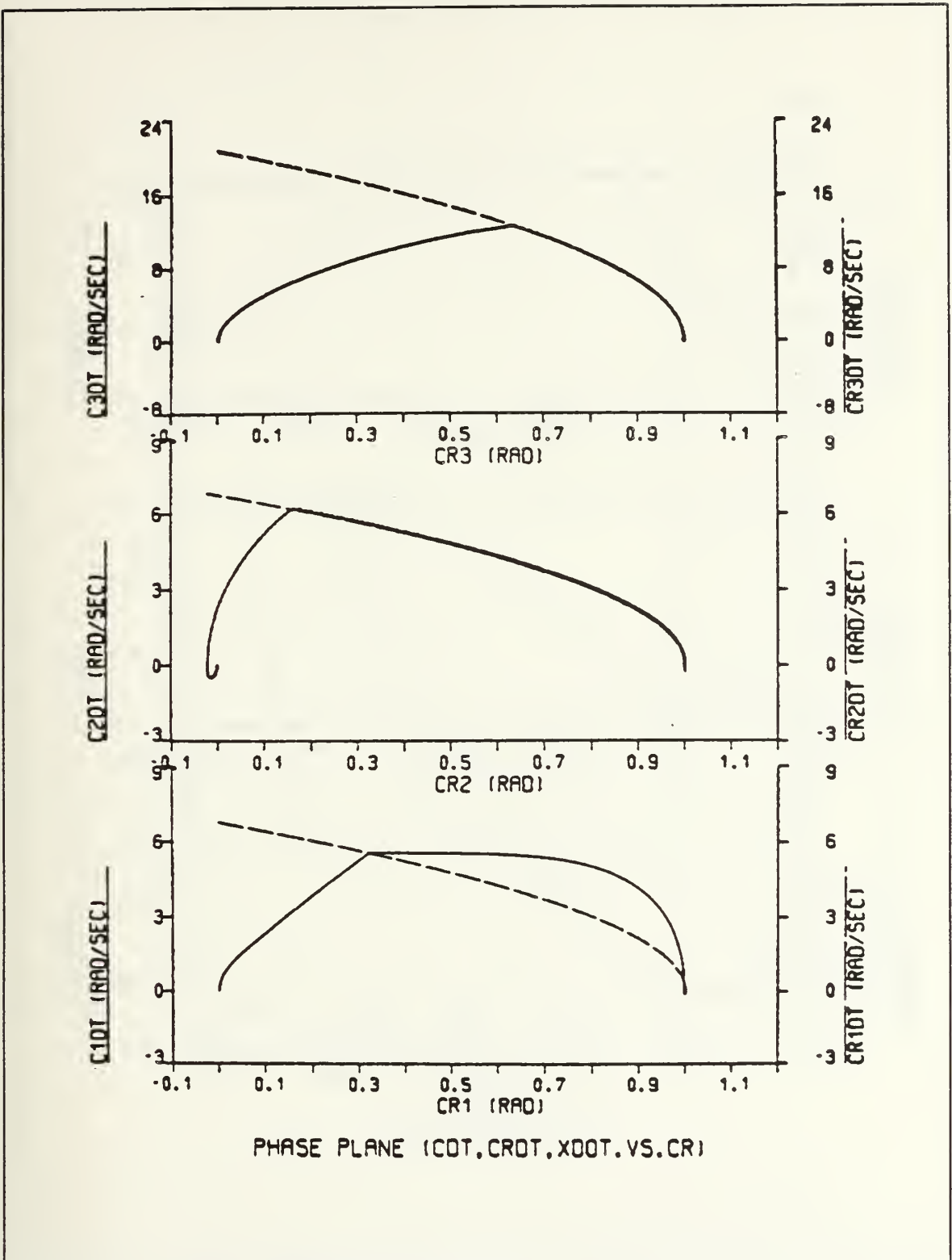
Figure 9.10   Phase Plane Trajectory for Loaded Arm
(Load=0.157 – No Gravity)
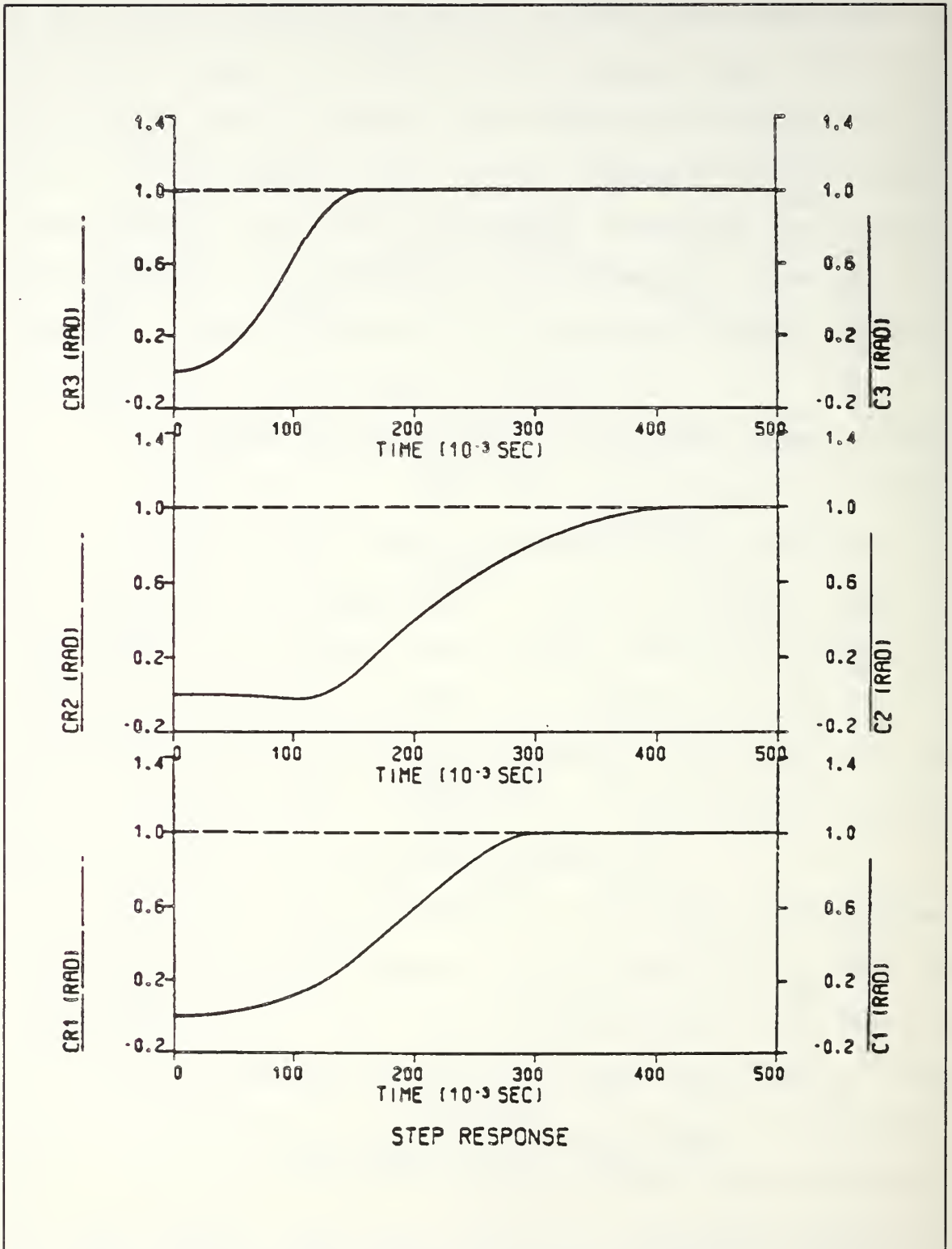
Figure 9.11   Step Response for Loaded Arm
(Load=0.157 - No Gravity)

146
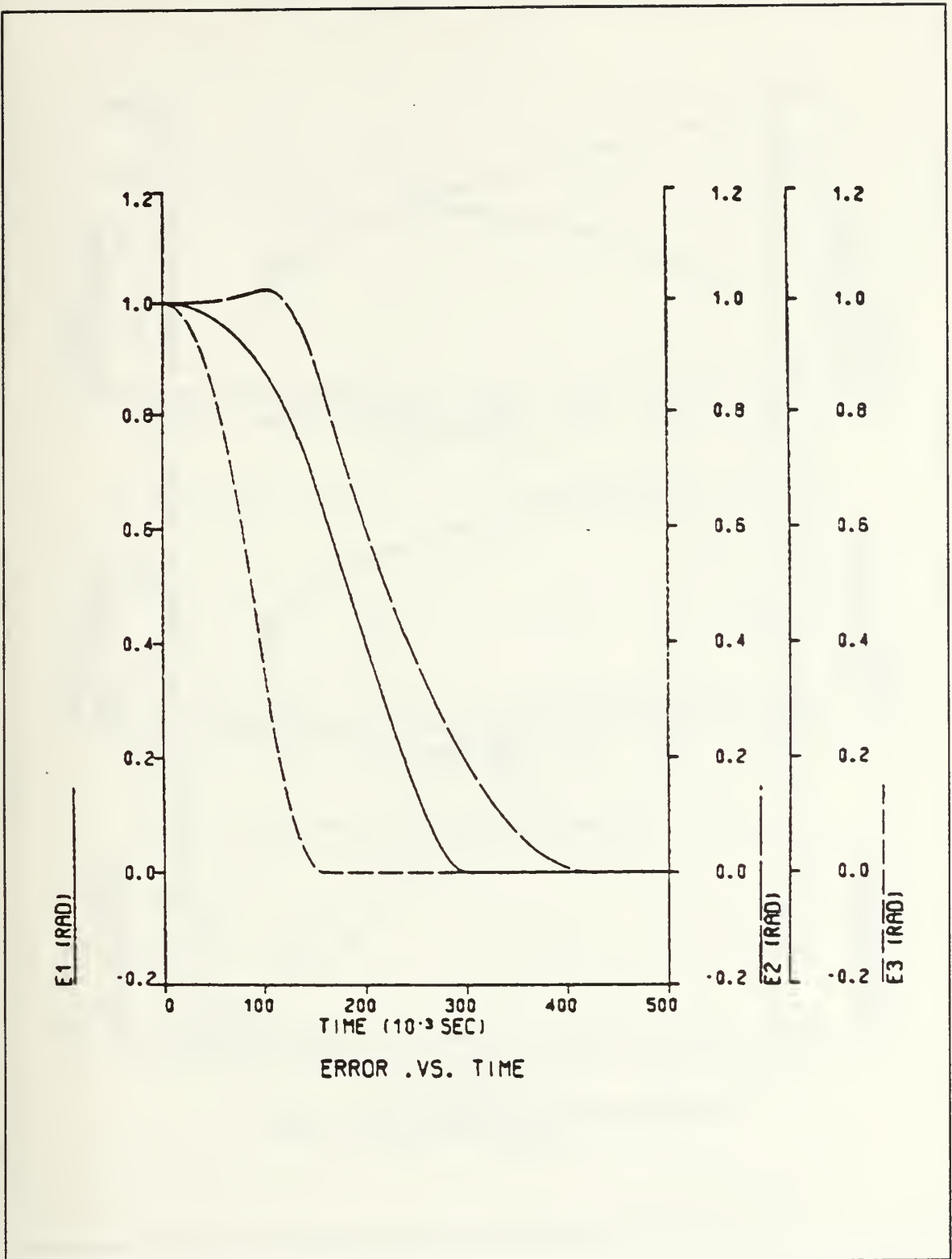
Figure 9.12   Error Curves for Loaded Arm
(Load=0.157 - No Gravity)

147
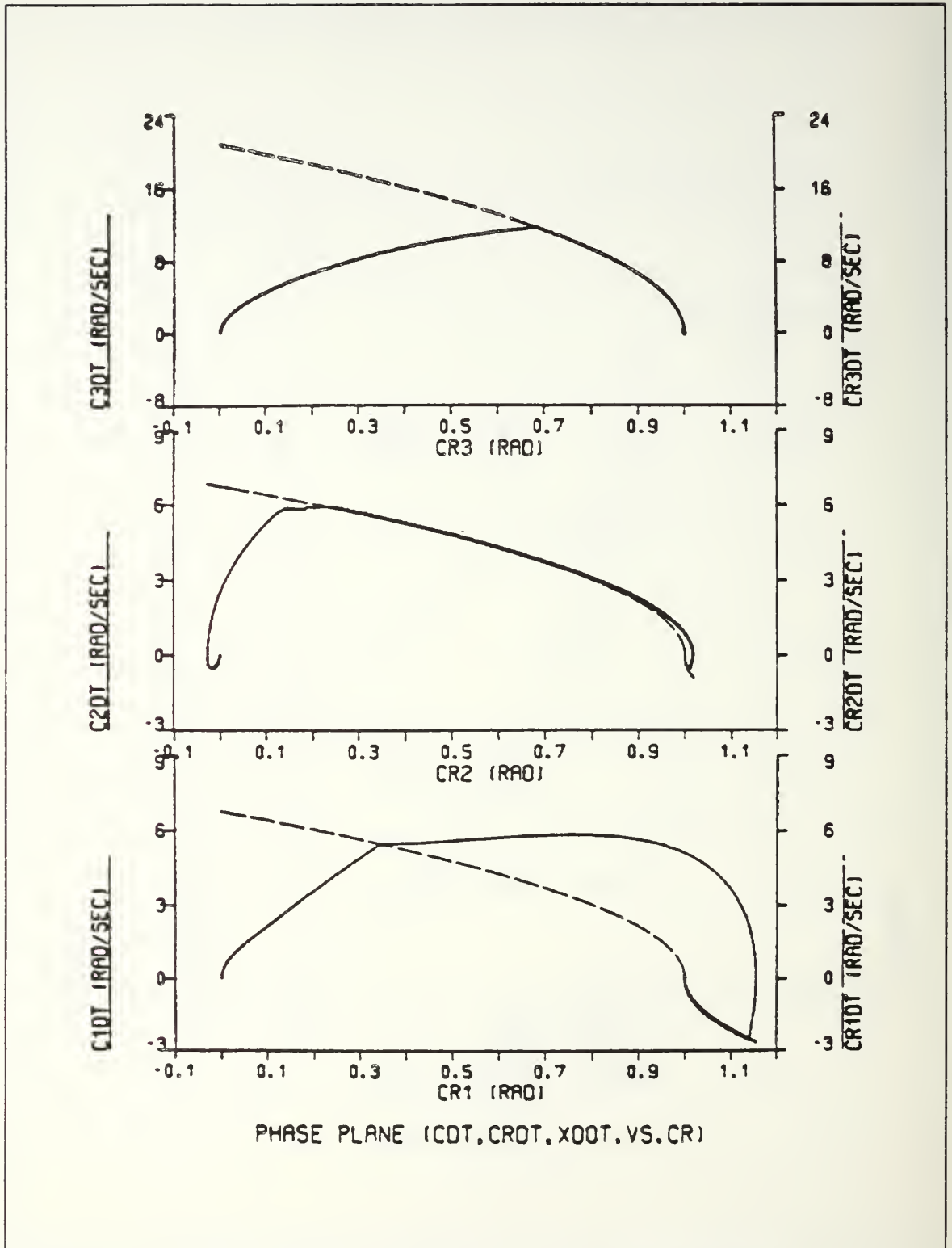
Figure 9.13   Phase Plane Trajectory for Loaded Arm
(Load=0.2 - No Gravity)
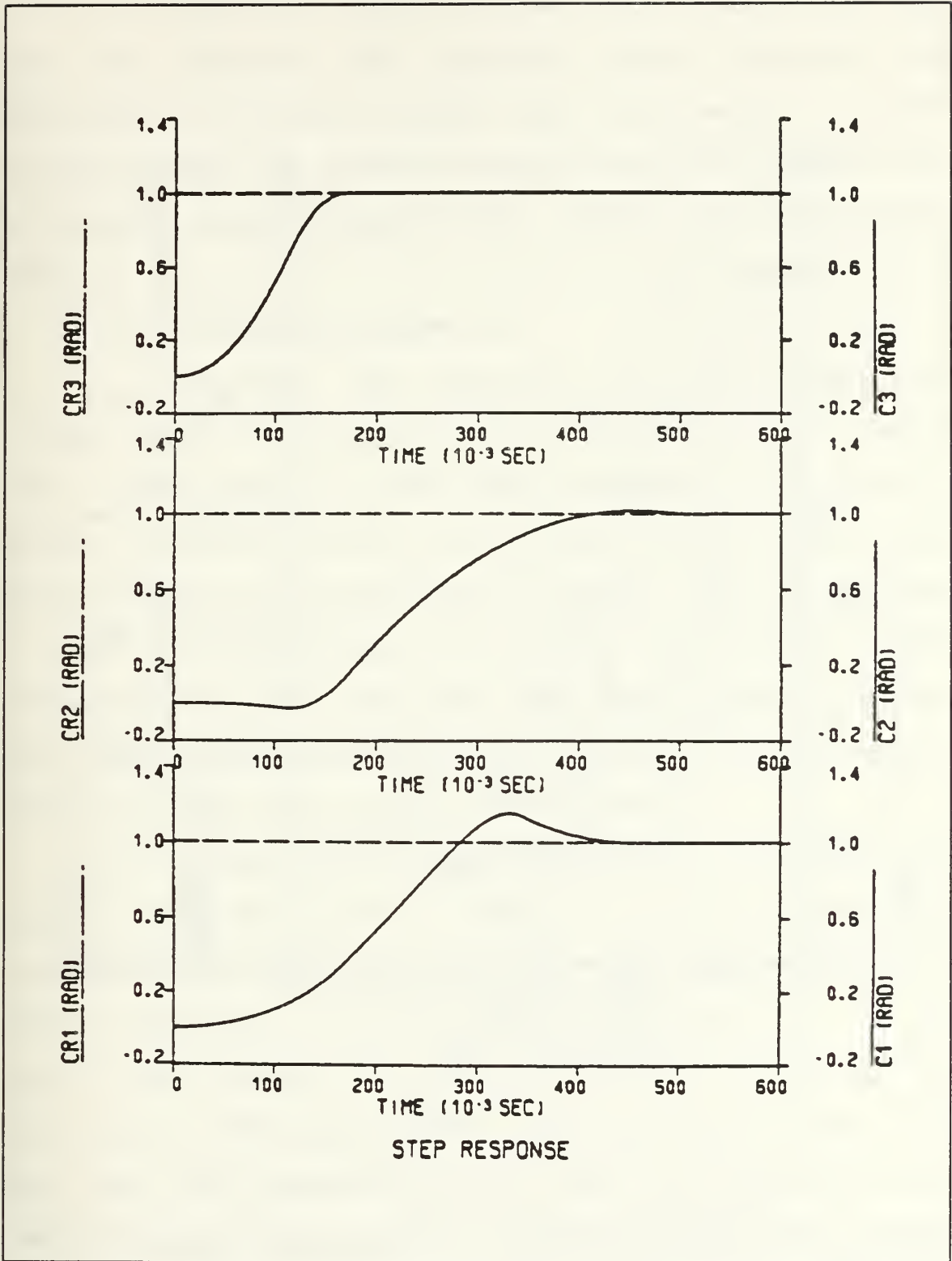
Figure 9.14 Step Response for Loaded Arm
(load=0.2 - No Gravity)

149
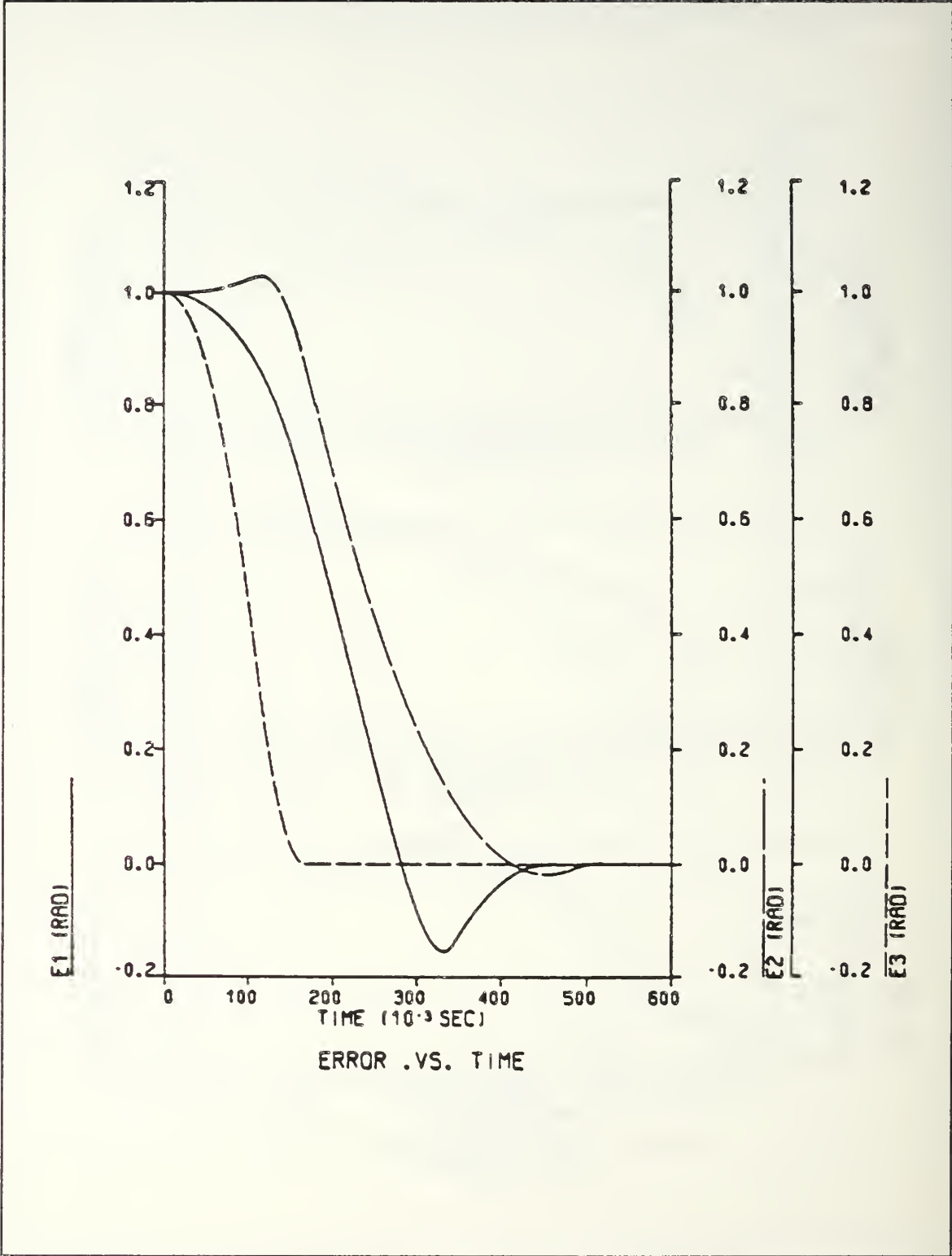
Figure 9.15   Error Curves for Loaded Arm
(Load=0.2 - No Gravity)

150

From the study of the above results for the three link revolute robot operating under different load conditions, it becomes obvious that the greater the added load the greater the initial negative velocity of JOINT2 and the greater overshooting for the JOINT1 and JOINT2 servo motors.

   b.   Disturbance Rejection

   To test the capability of the three link revolute robot to reject random disturbances, an impulse with an amplitude of 50 rad and a duration of 10 msec is applied to three joints in different times during their motion. Making use of the DSL/VS program listed in Appendix I, first the disturbance is applied just before JOINT3 reaches steady state, then the same disturbance is applied just before JOINT1 and JOINT2 reach steady state and finally after all joints have reached steady state.

   In the first case Figure 9.16 shows that the applied disturbance causes JOINT3 to be unable to catch the curve before it reaches the commanded position and therefore JOINT3 overshoots as illustrated in Figure 9.17. In the second case Figure 9.18 shows that due to the applied disturbance JOINT1 and JOINT2 catch the curve just after they pass the commanded position and the resulting small overshooting of these joints is shown in Figure 9.19. In the third case, Figure 9.20 shows that the robot is disturbed
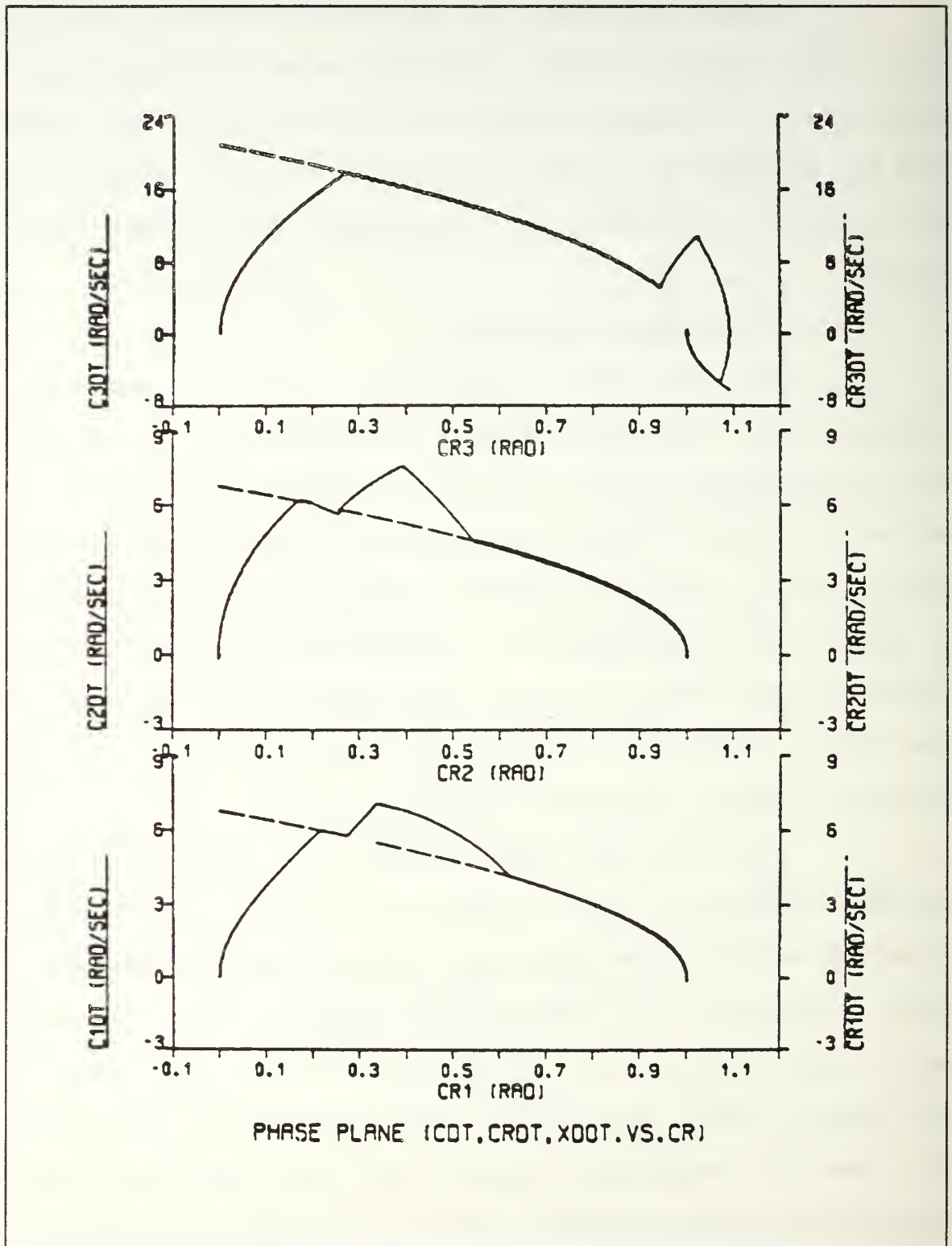
Figure 9.16   Phase Plane Trajectory with Disturbance
Applied at T=90 msec (No Gravity)
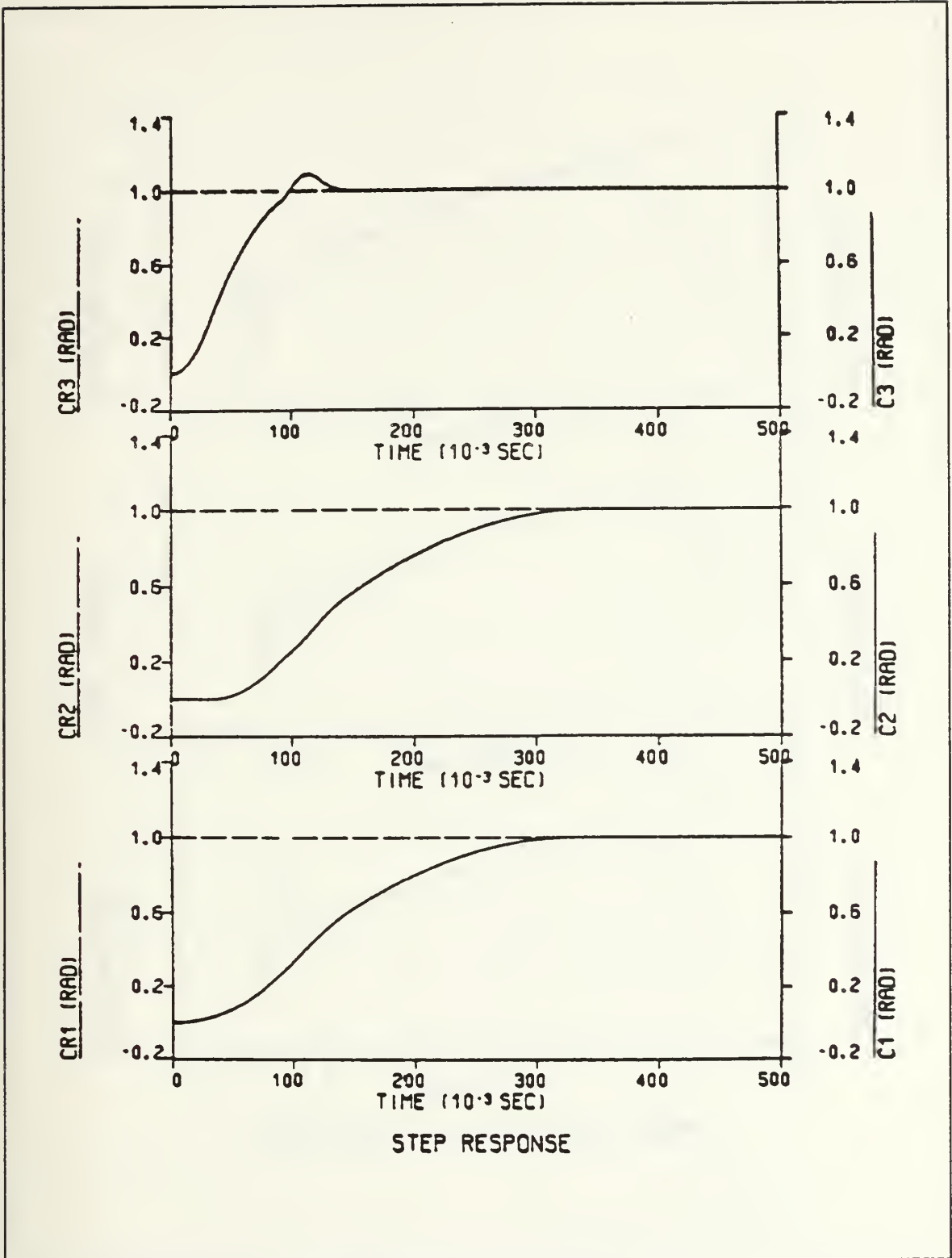
Figure 9.17   Step Response with Disturbance
Applied at T=90 msec (No Gravity)
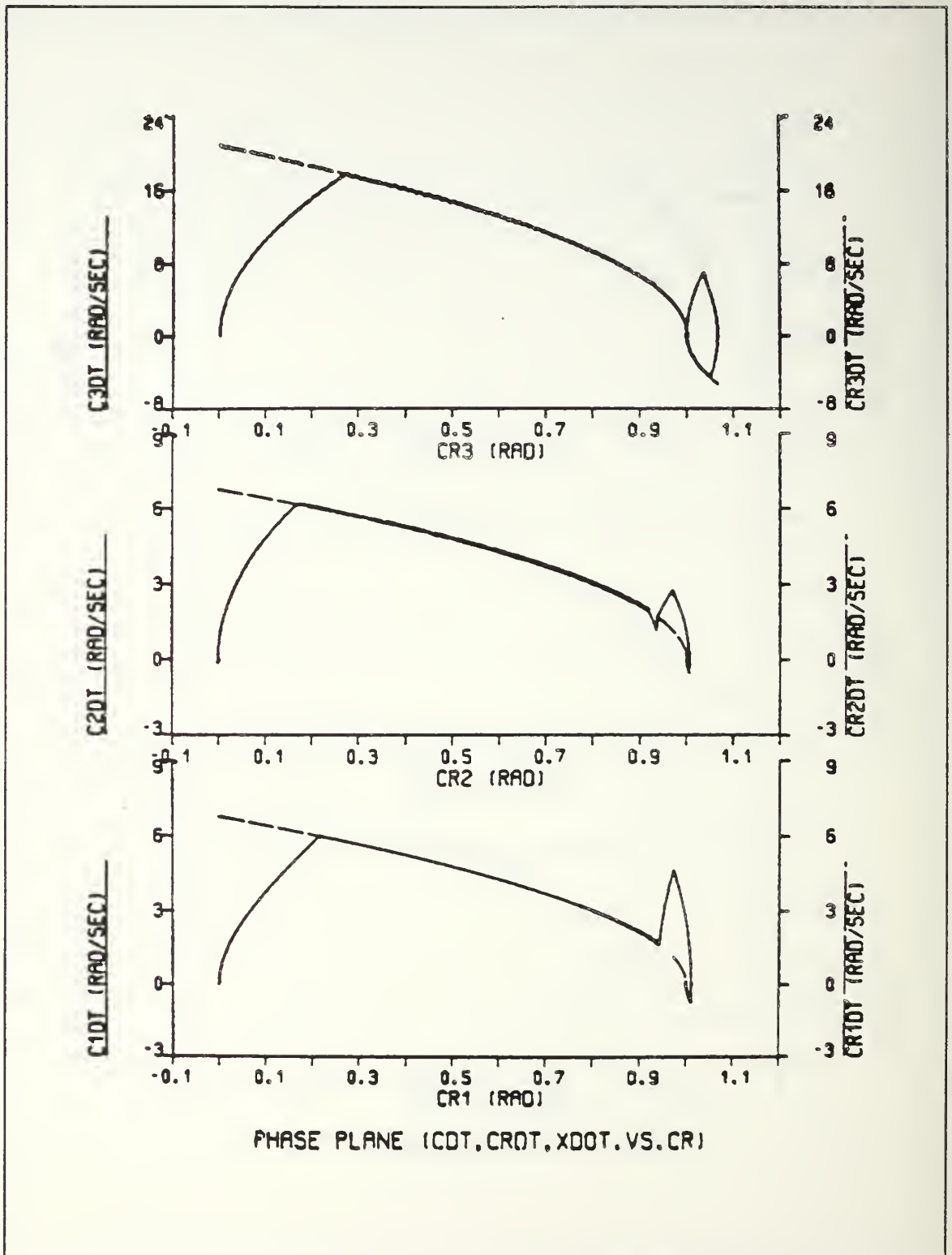
Figure 9.18   Phase Plane Trajectory with Disturbance
Applied at T=270 msec (No Gravity)

154
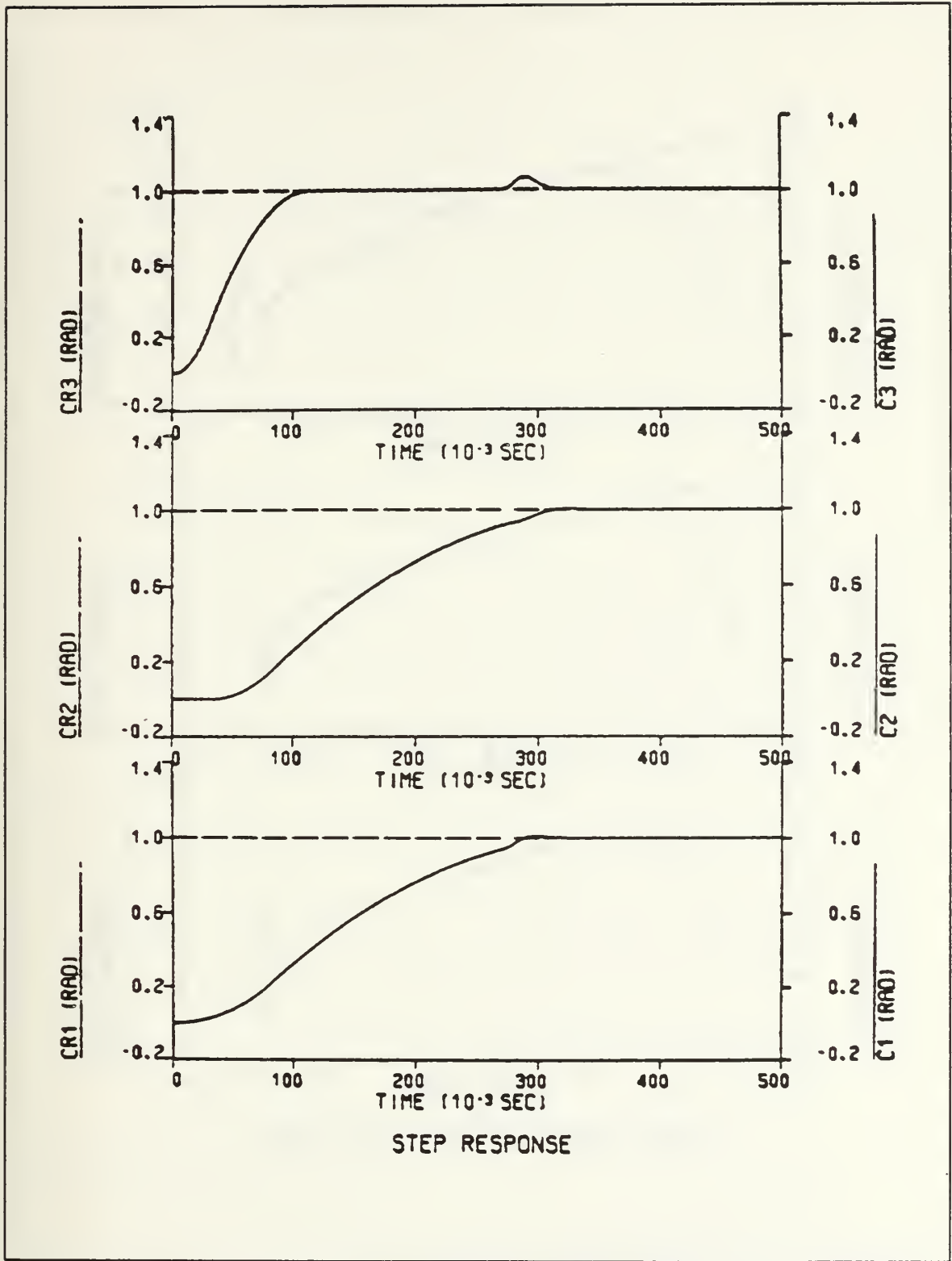
Figure 9.19   Step Response with Disturbance
Applied at T=270 msec (No Gravity)
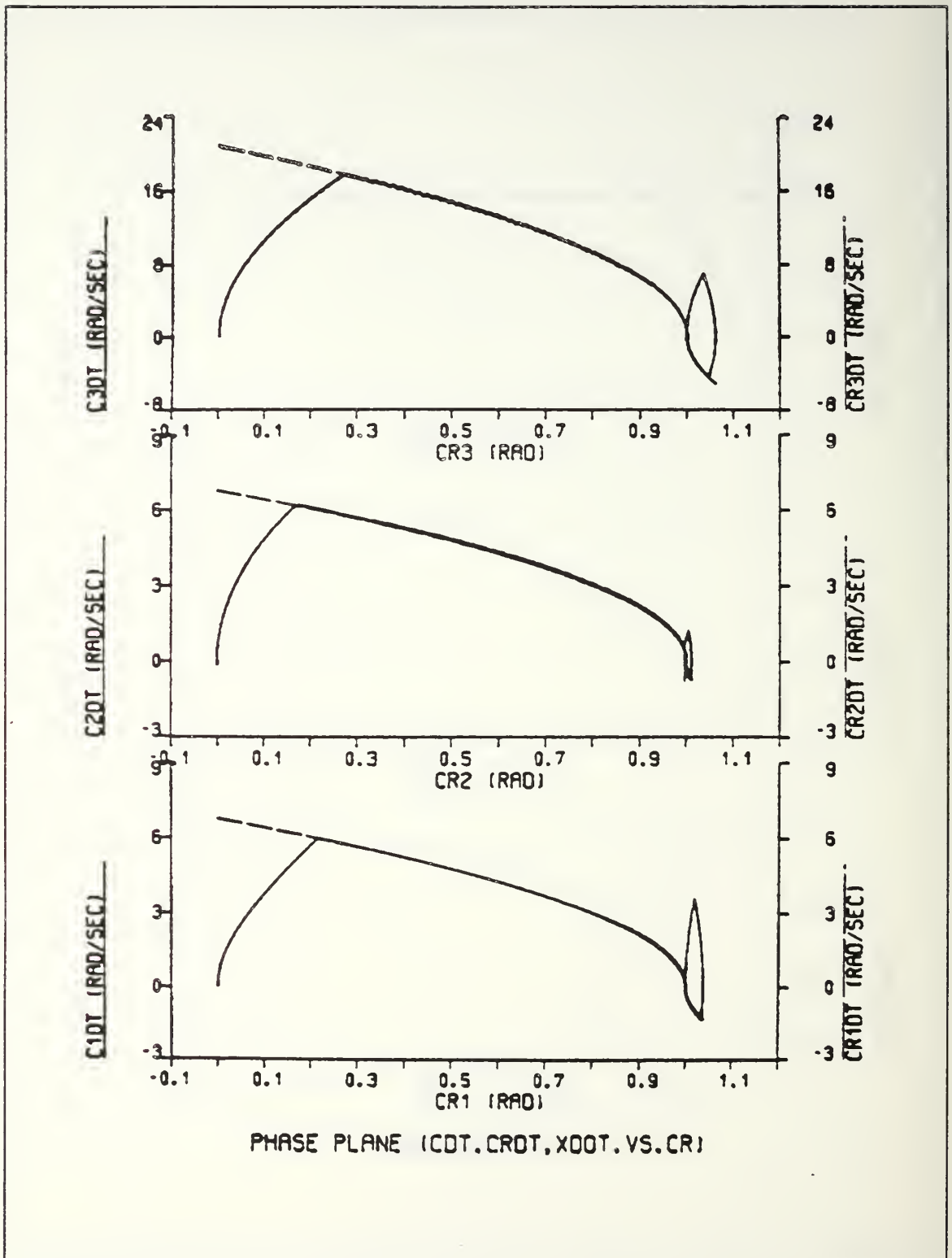
Figure 9.20 Phase Plane Trajectory with Disturbance
Applied at T=370 msec (No Gravity)
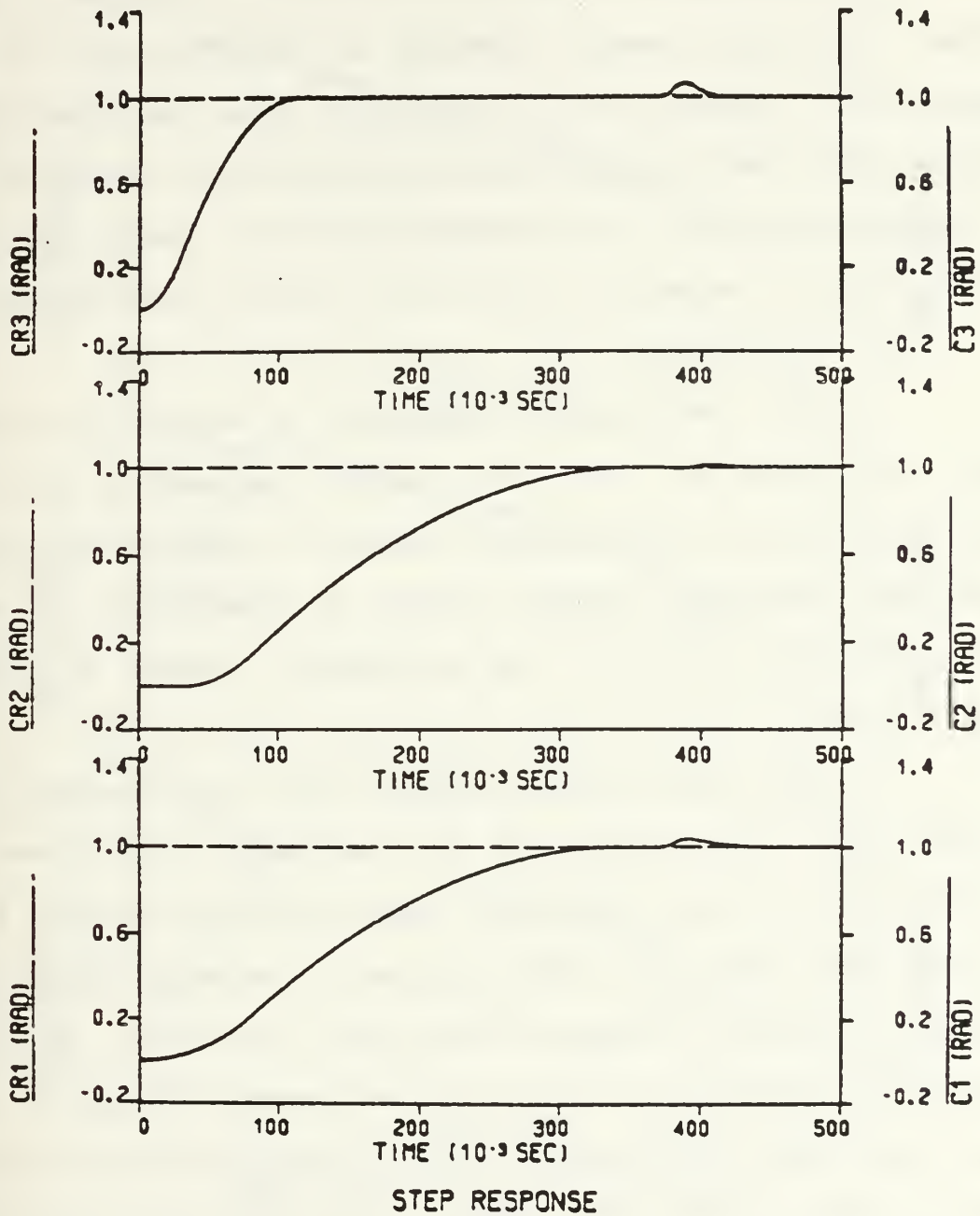
Figure 9.21 Step Response with Disturbance
Applied at T=370 msec (No Gravity)

after all joints have reach steady state condition and the overshooting of all joints can be observed in Figure 9.21.

Although the same disturbance is applied to all joints, the joint mostly affected is JOINT3 which is the lightest one. Finally in all different cases the simulation results show that the applied disturbances are rejected and all joints return to the commanded position. Therefore, the tested robot is capable to reject any random disturbance.

c. Robustness

Next the adaptive algorithm is tested for slight (10%) variations of the servo motors parametric data in order to check if the designed system is robust. For these tests the simulation program listed in Appendix H is used again, changing some of the parametric values for each different run.

First the torque constant (K_t) and the back emf constant (K_v) are changed 10% and the results are shown in Figures 9.22 - 9.25. The phase plane plots of Figures 9.22 and 9.24 show very good curve following characteristics for positive or negative change of the above constants. The step response of Figures 9.23 and 9.24 are nearly identical and only an accurate observation of the simulation data reveals a small variation in the settling time which has the same sign with the variation of the two constants.

Then the mechanical pole of the servo motor is moved toward the origin and the effects of this change in
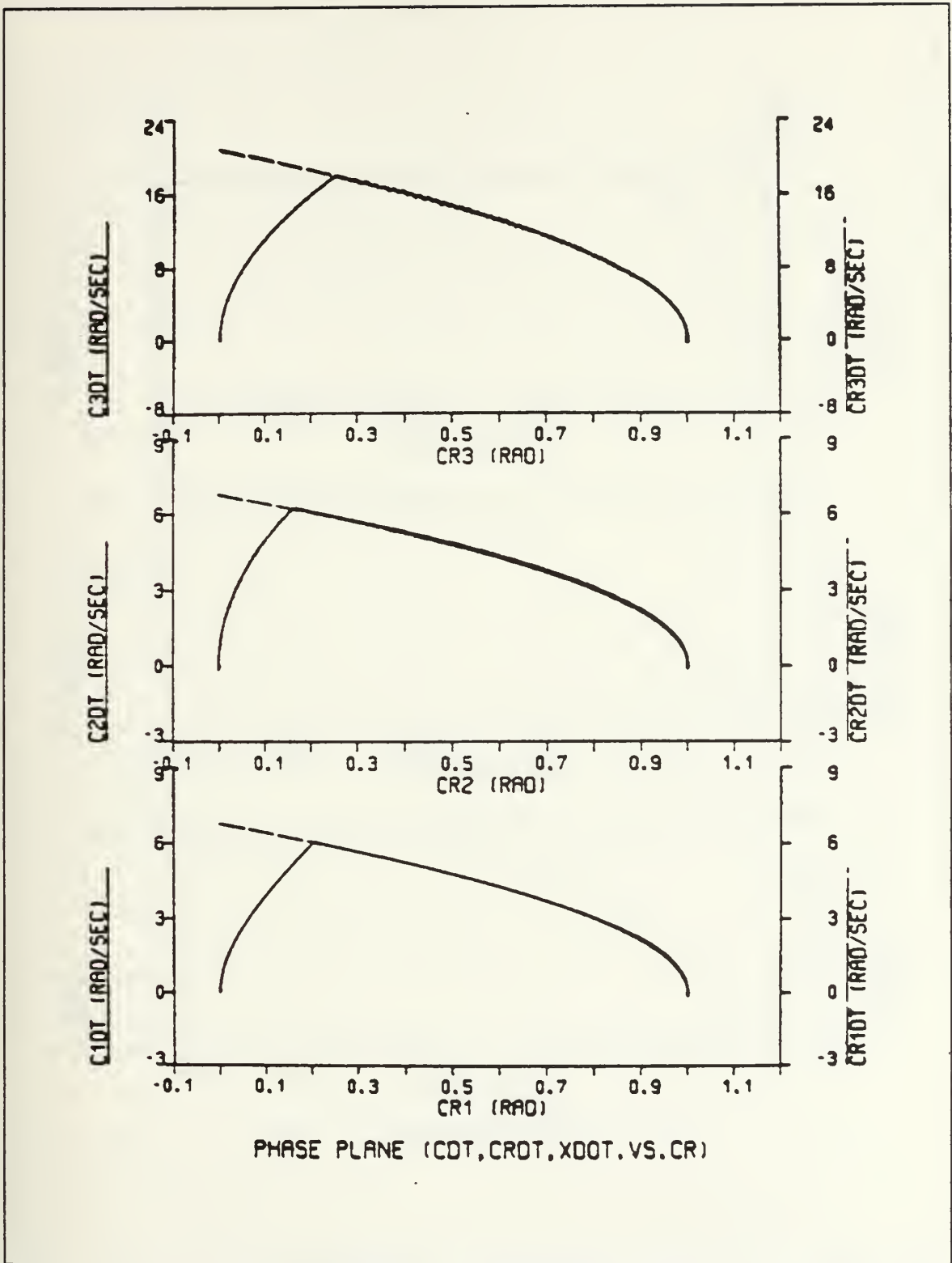
Figure 9.22   Phase Plane Trajectory for Unloaded Arm
with K_t and K_v Increased by 10% (No Gravity)
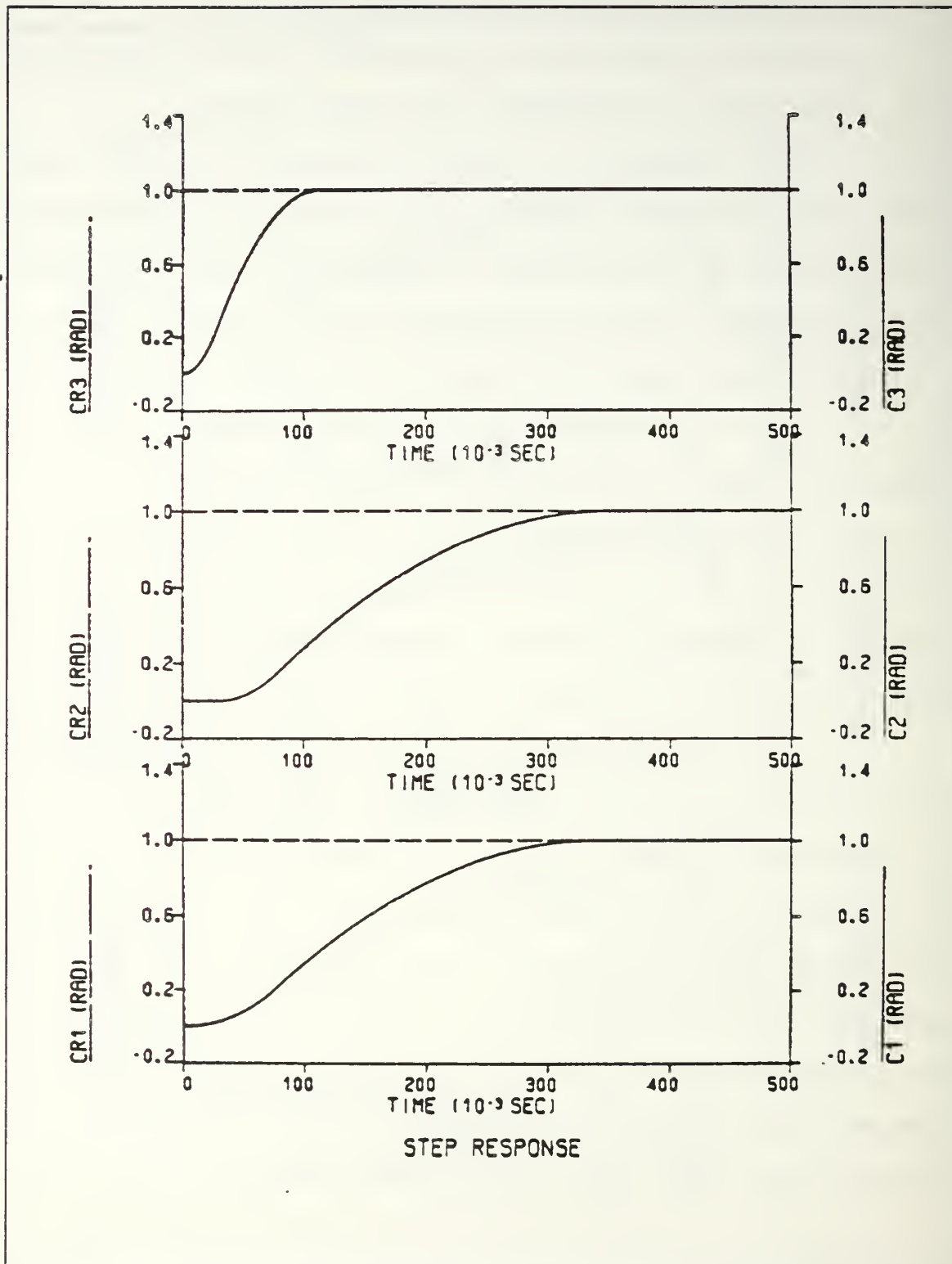
Figure 9.23   Step Response for Unloaded Arm with
K_t and K_v Increased by 10% (No Gravity)
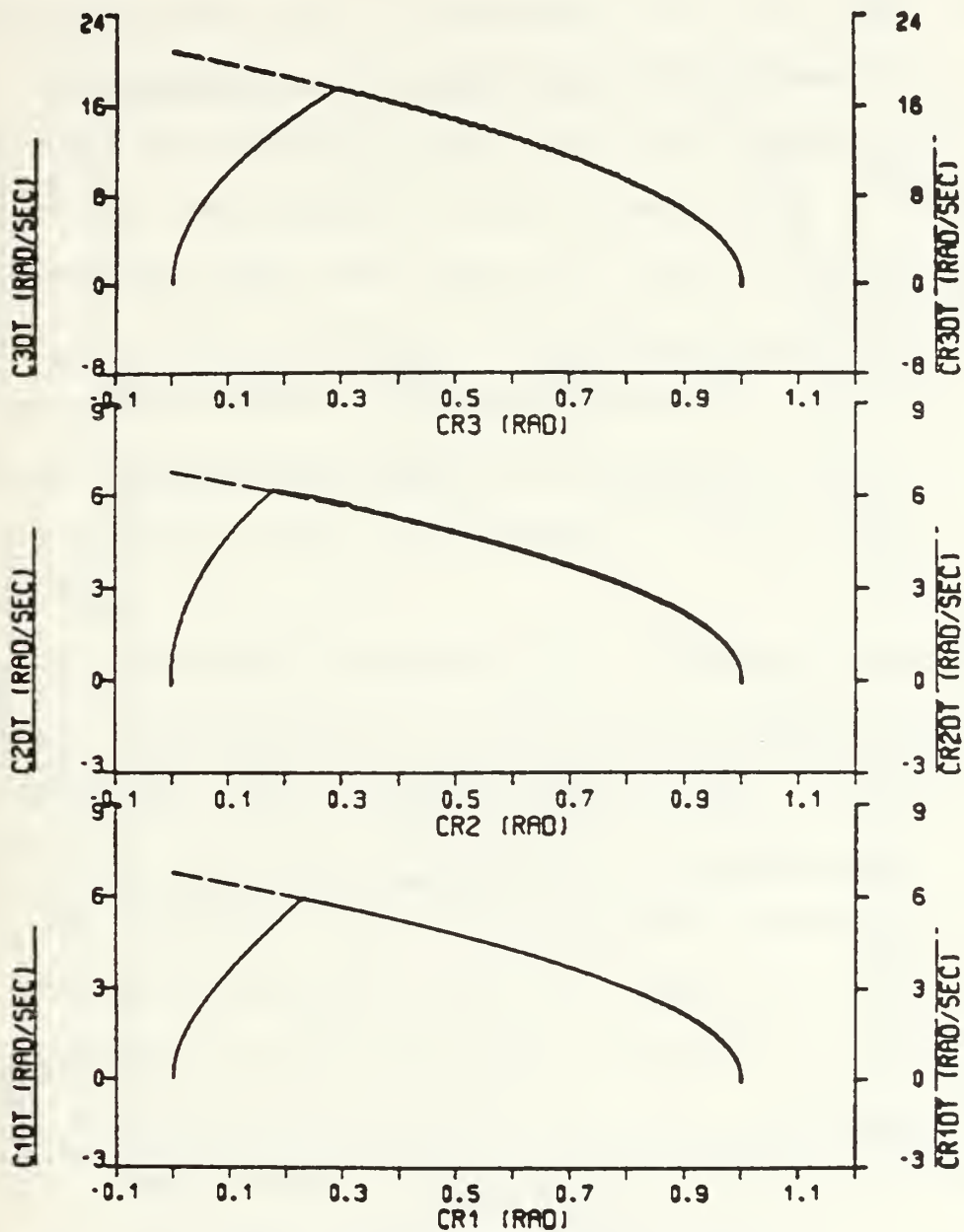
Figure 9.24 Phase Plane Trajectory For Unloaded Arm
with K_t and K_v Decreased by 10% (No Gravity)
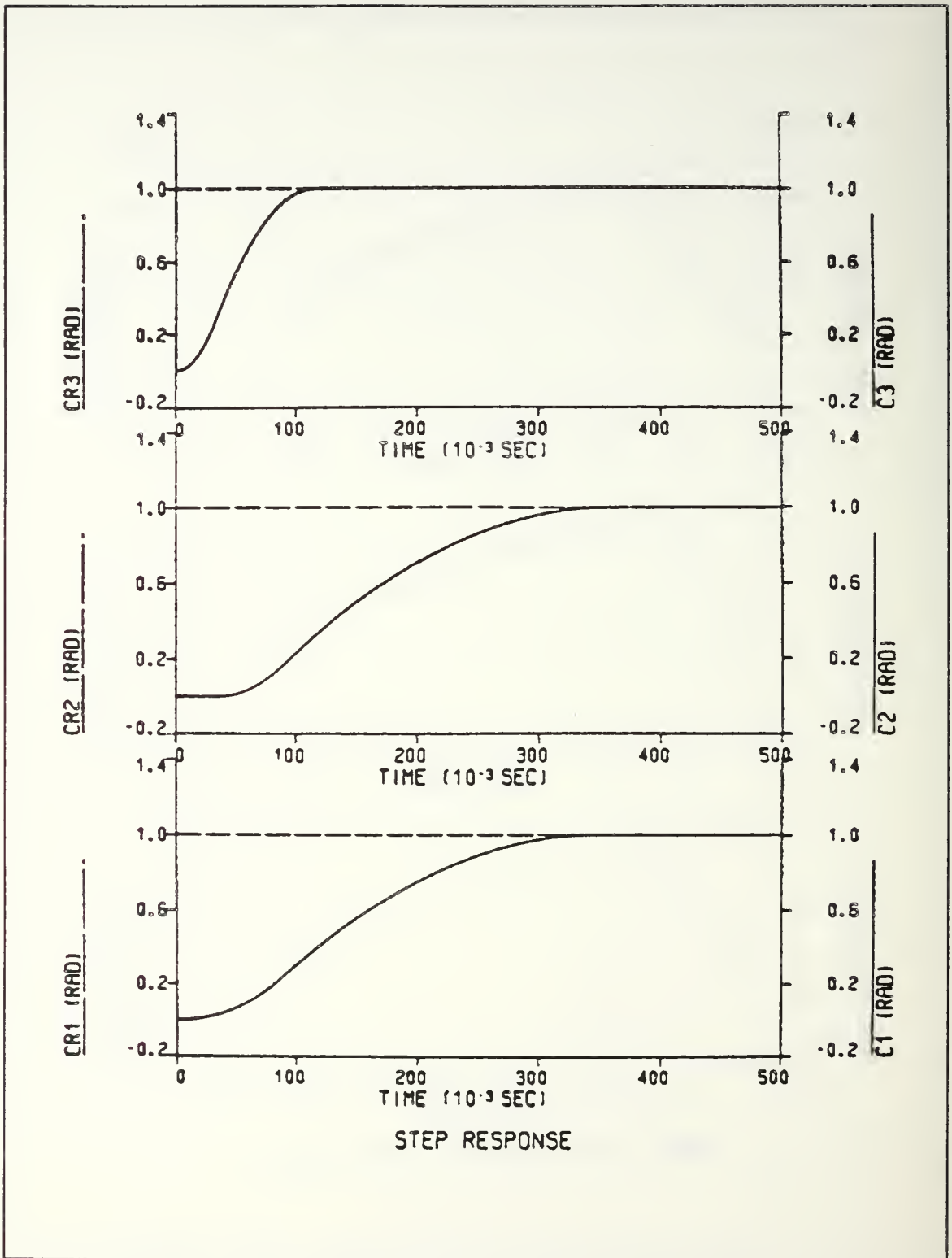
Figure 9.25   Step Response for the Unloaded Arm with
K_t and K_v Decreased by 10% (No Gravity)

the behavior of the system are observed. Maximum movement of the pole is achieved when the resistance (R) is decreased and the inductance (L) is increased with the same amount (10%), simultaneously. In comparison with Figures 9.1 and 9.2, Figures 9.26 and 9.27 show that this movement does not affect at all the behavior of the unloaded arm. But Figures 9.28 - 9.31 show that the loaded arm does not overshoot for loads up to 0.185 oz/in/sec^2. Therefore, as result of this movement of the mechanical pole of the servo motor, the load for which overshooting starts is increased from 0.157 to 0.185 oz/in/sec^2, which corresponds to 18% increase of the critical load.

After the mechanical pole has been moved toward the origin, the constants K_t and K_v are changed again and the effects of this combination of changes are observed. Figures 9.32 - 9.35 show that for 10% increased values of K_t and K_v the critical load is further more increased to the value of 0.215 oz/in/sec^2, while for decreased values of the above constants by the same amount, Figures 9.36 - 9.39 show that the critical load is reduced to 0.160 oz/in/sec^2.

From the results of the above tests, it becomes clear that any small change in the parameters of the servo motors, which is possible in practice, does not affect dramatically the behavior of the system. Therefore, the designed robot arm is robust.
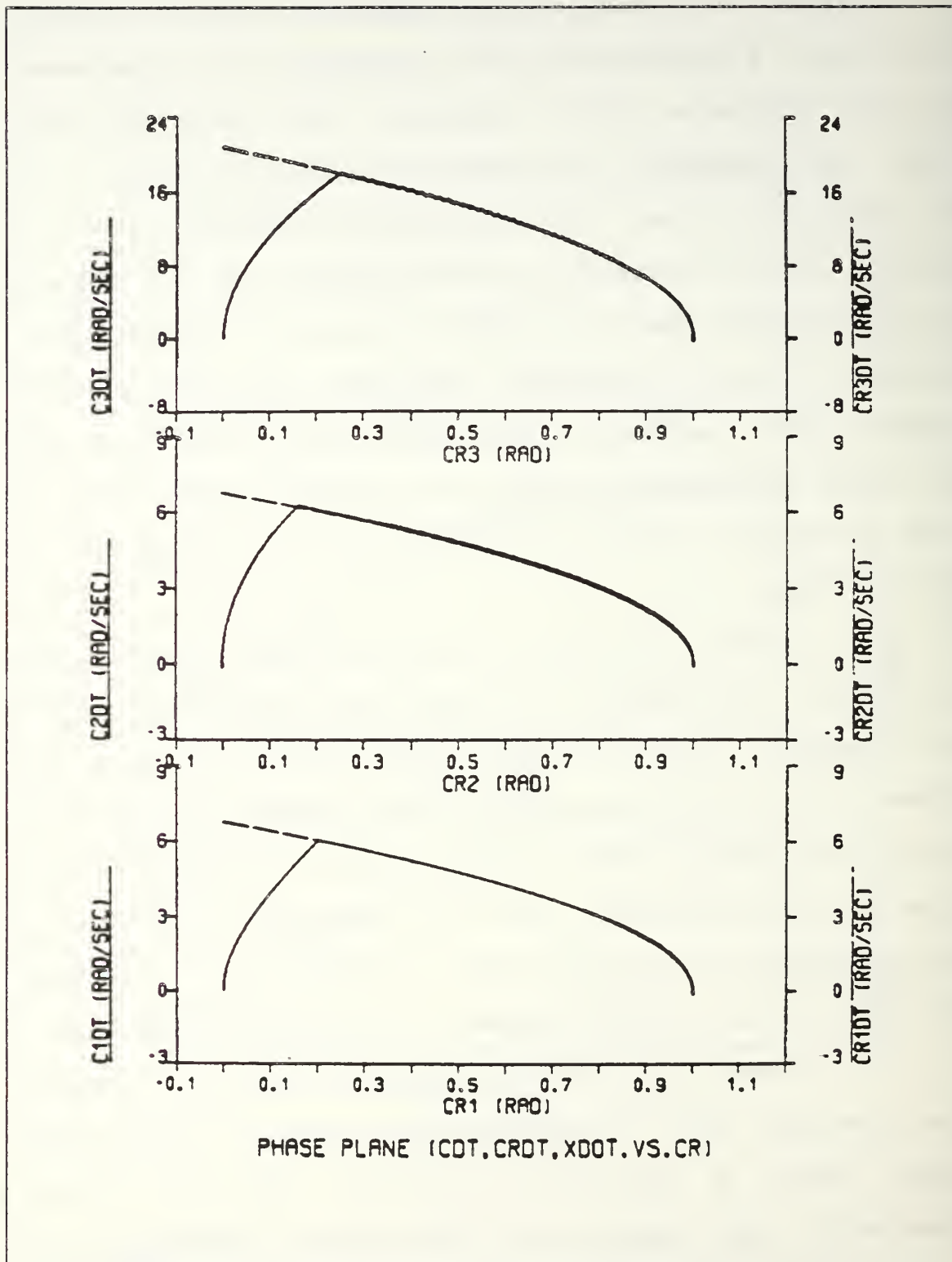
Figure 9.26   Phase Plane Trajectory for Unloaded Arm with
R Decreased and L Increased by 10% (No Gravity)

164
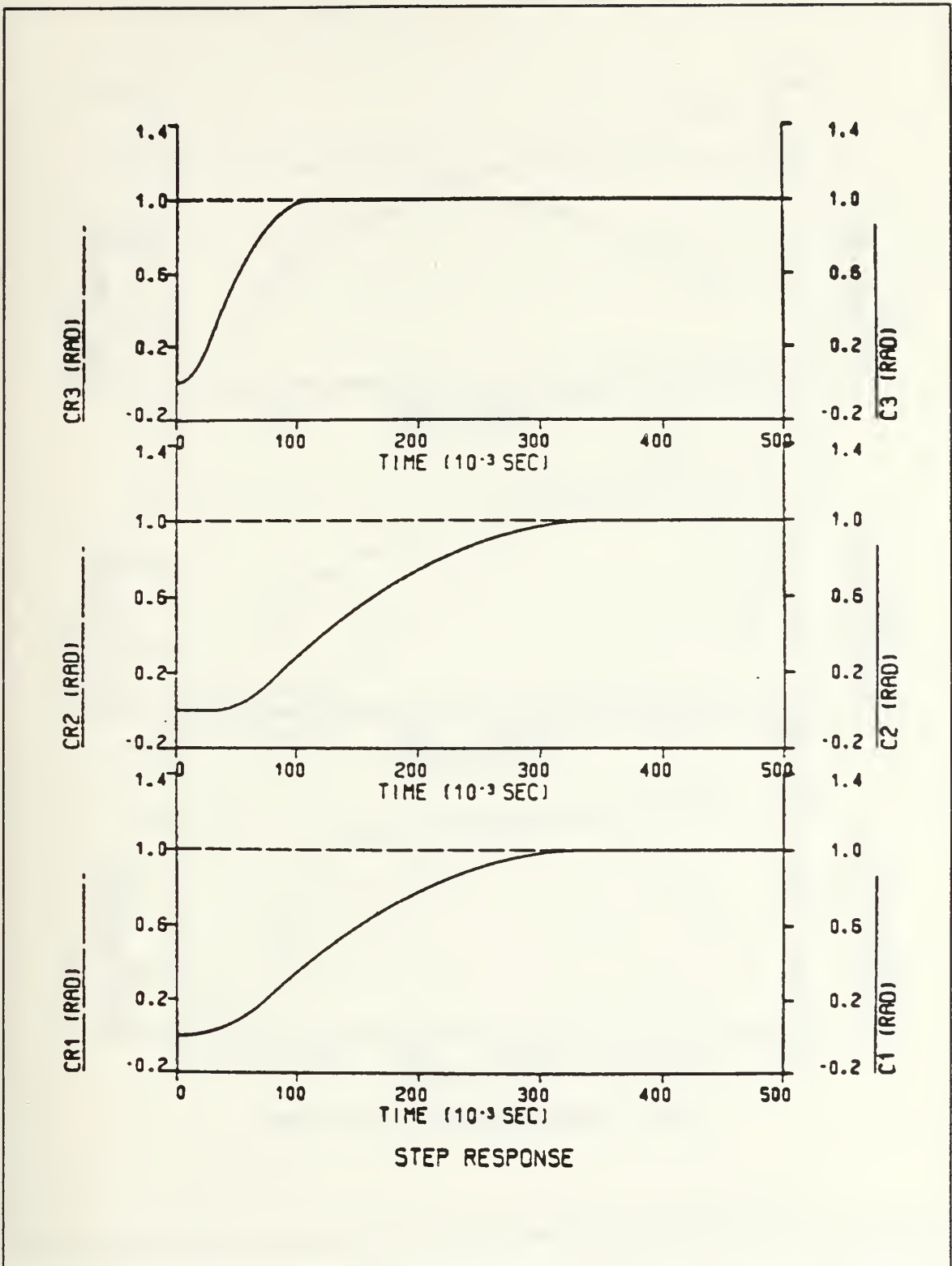
Figure 9.27 Step Response for Unloaded Arm with
R Decreased and L Increased by 10% (No Gravity)
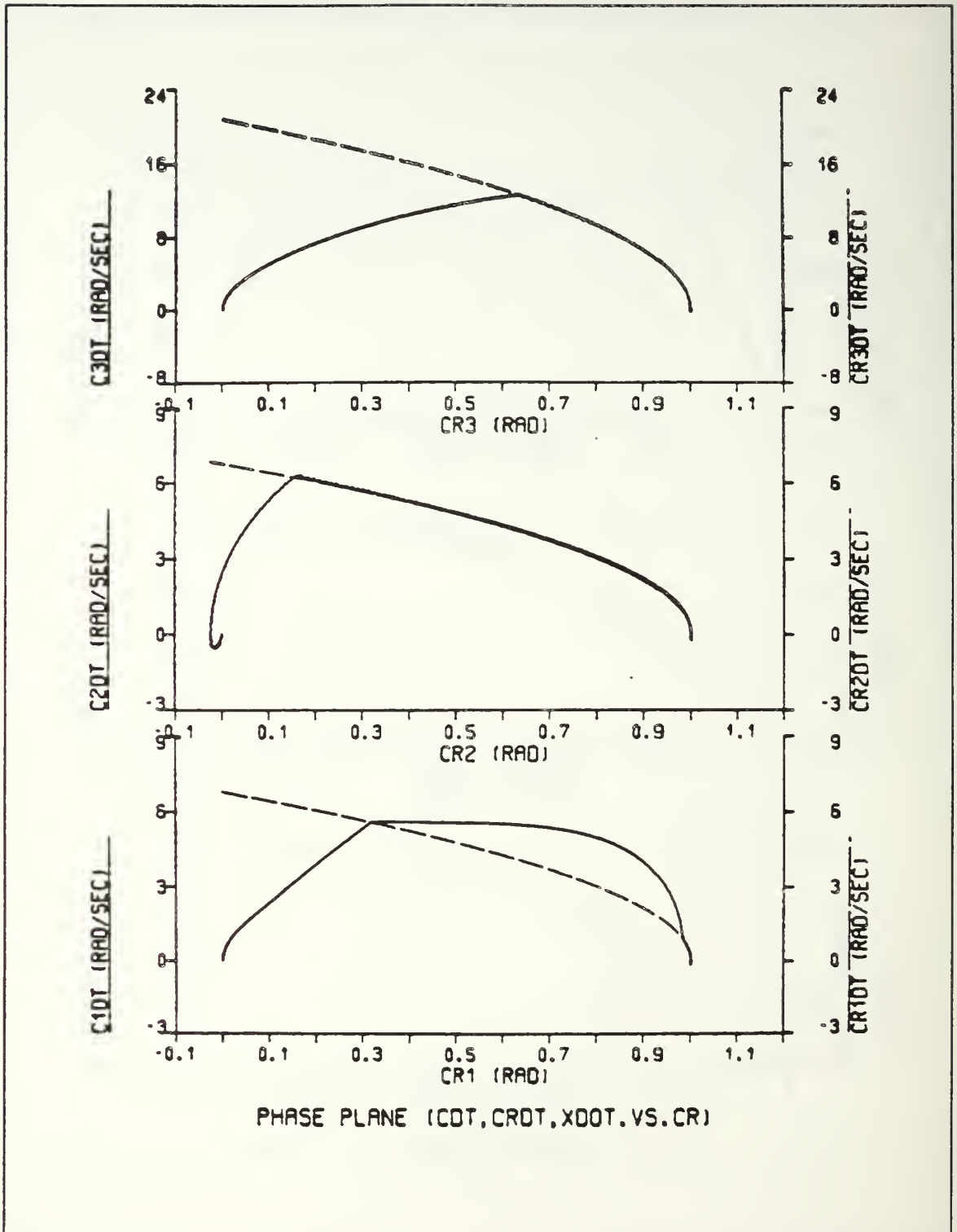
Figure 9.28   Phase Plane Trajectory for Loaded Arm
with R Decreased and L Increased   by 10%
(Load=0.18 - No Gravity)
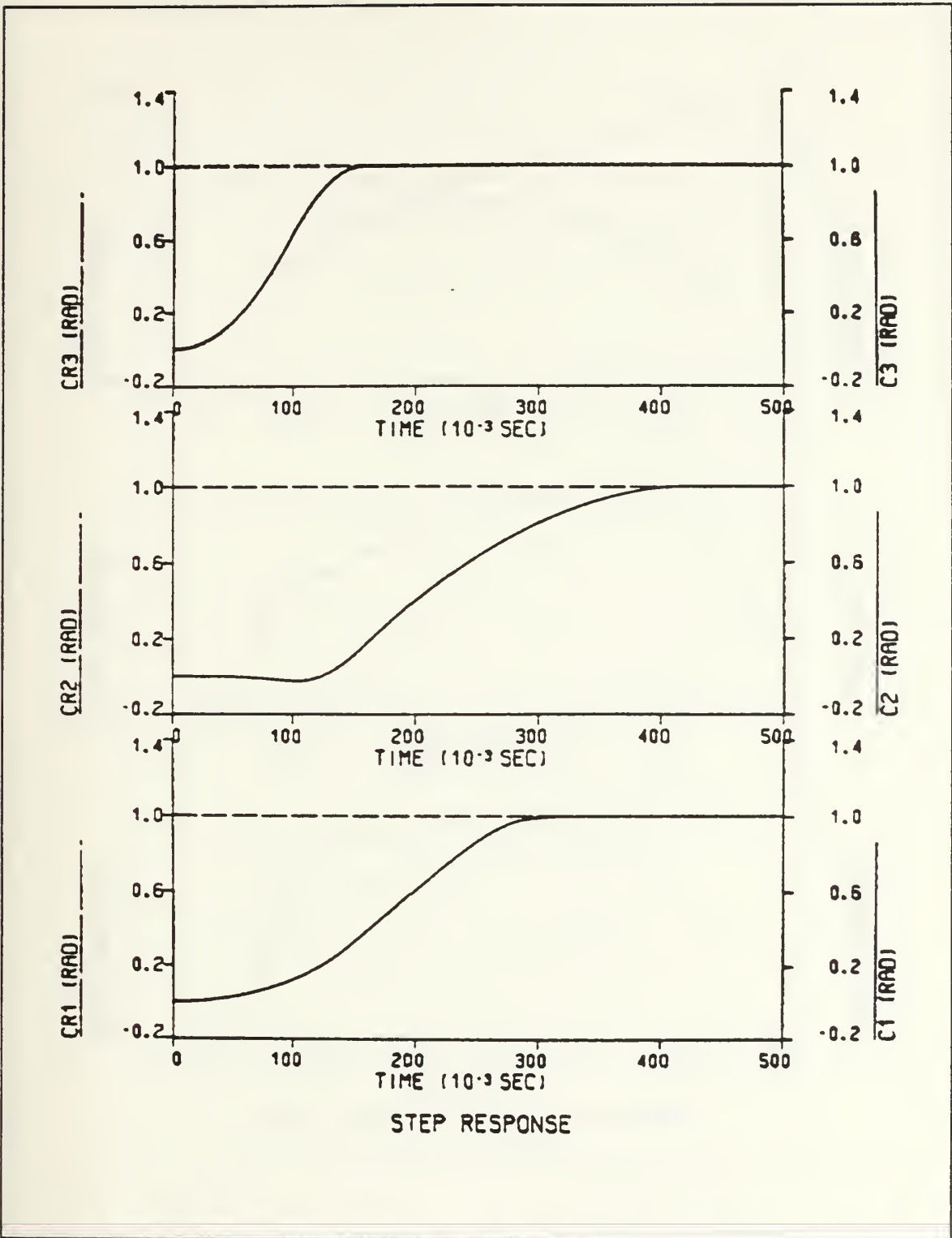
Figure 9.29 Step Response for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.18 - No Gravity)
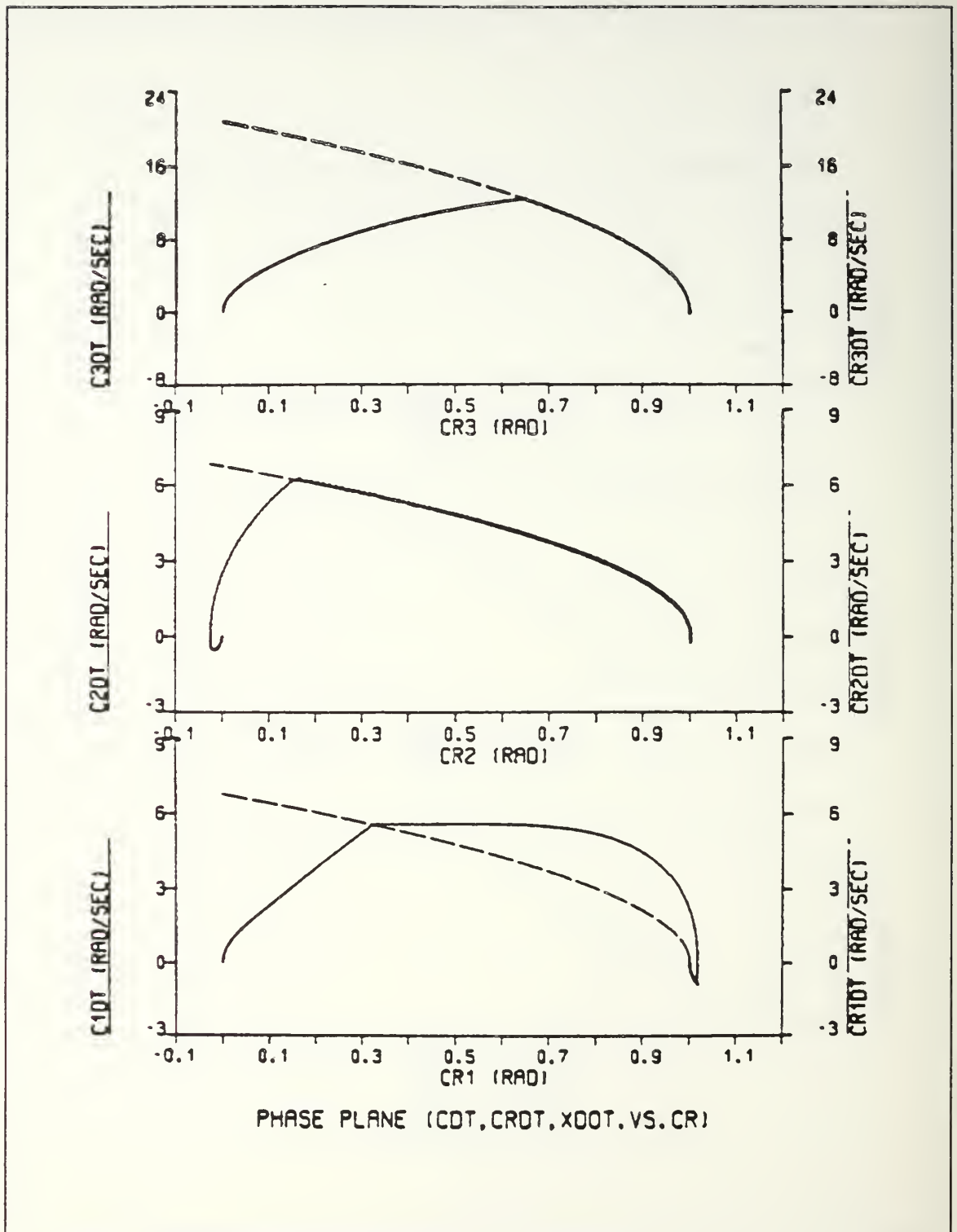
Figure 9.30 Phase Plane Trajectory for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.19 - No Gravity)

168
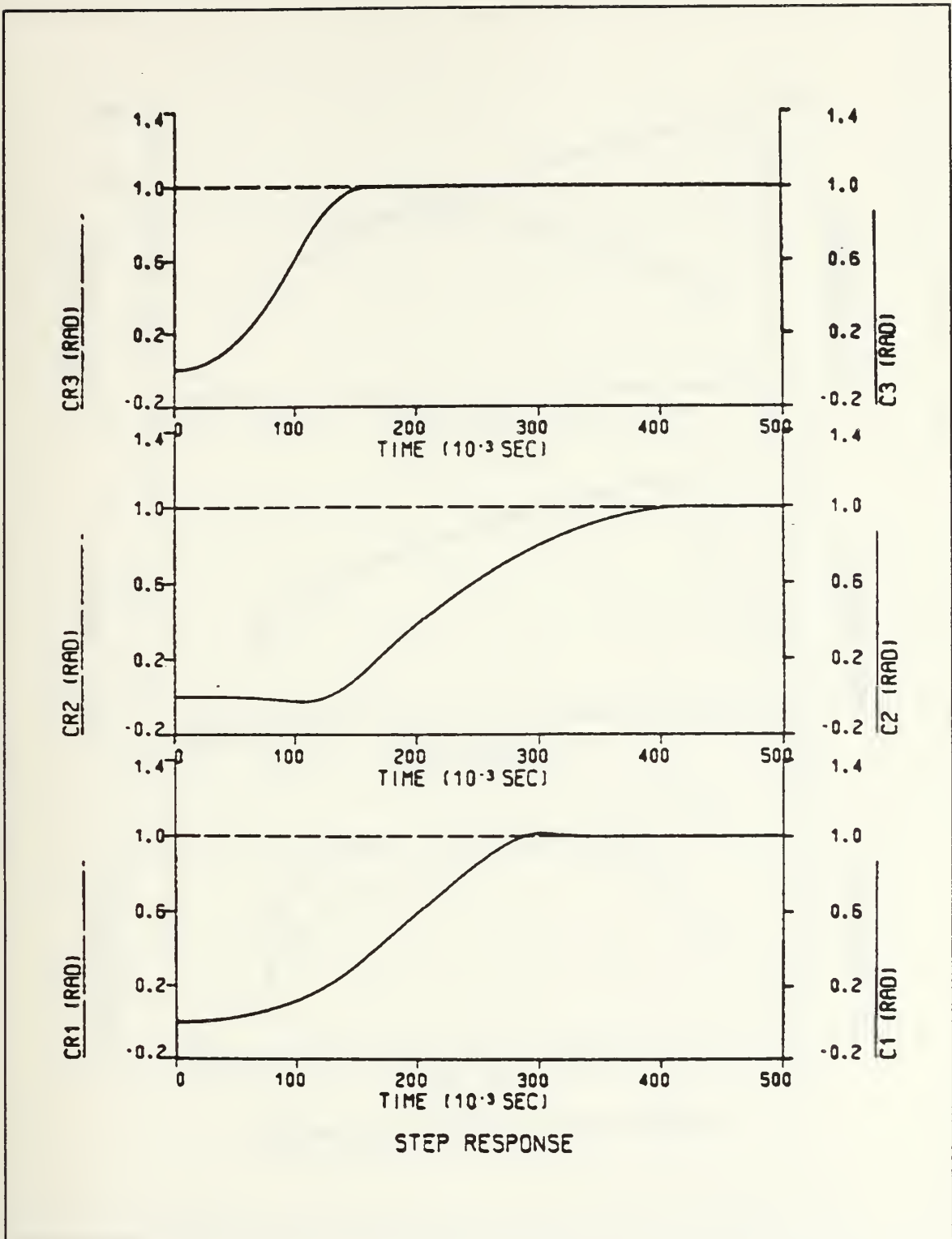
Figure 9.31 Step Response for Loaded Arm with
R Decreased and L Increased by 10%
(Load=0.19 - No Gravity)
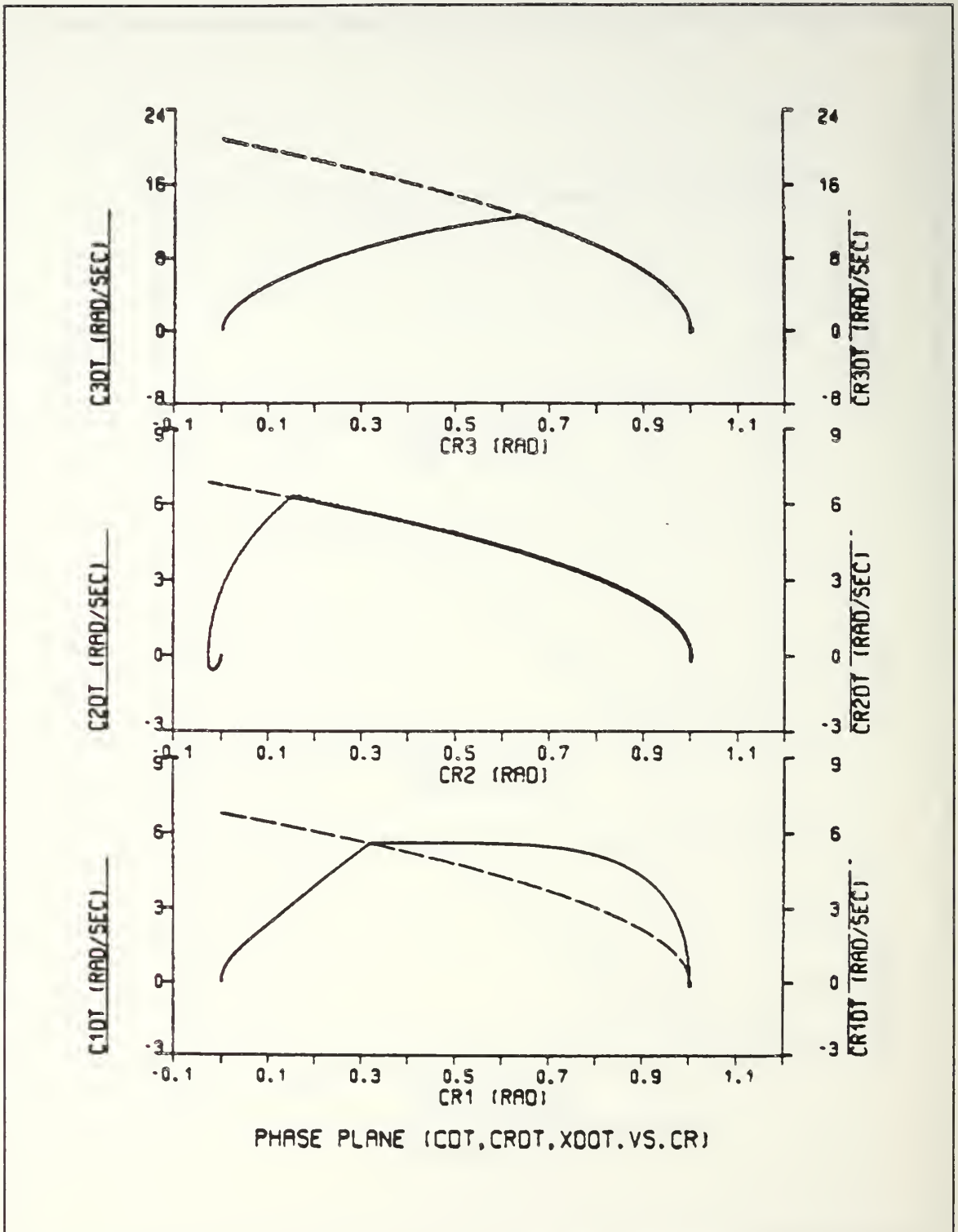
Figure 9.32   Phase Plane Trajectory for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=0.21 - No Gravity)

170
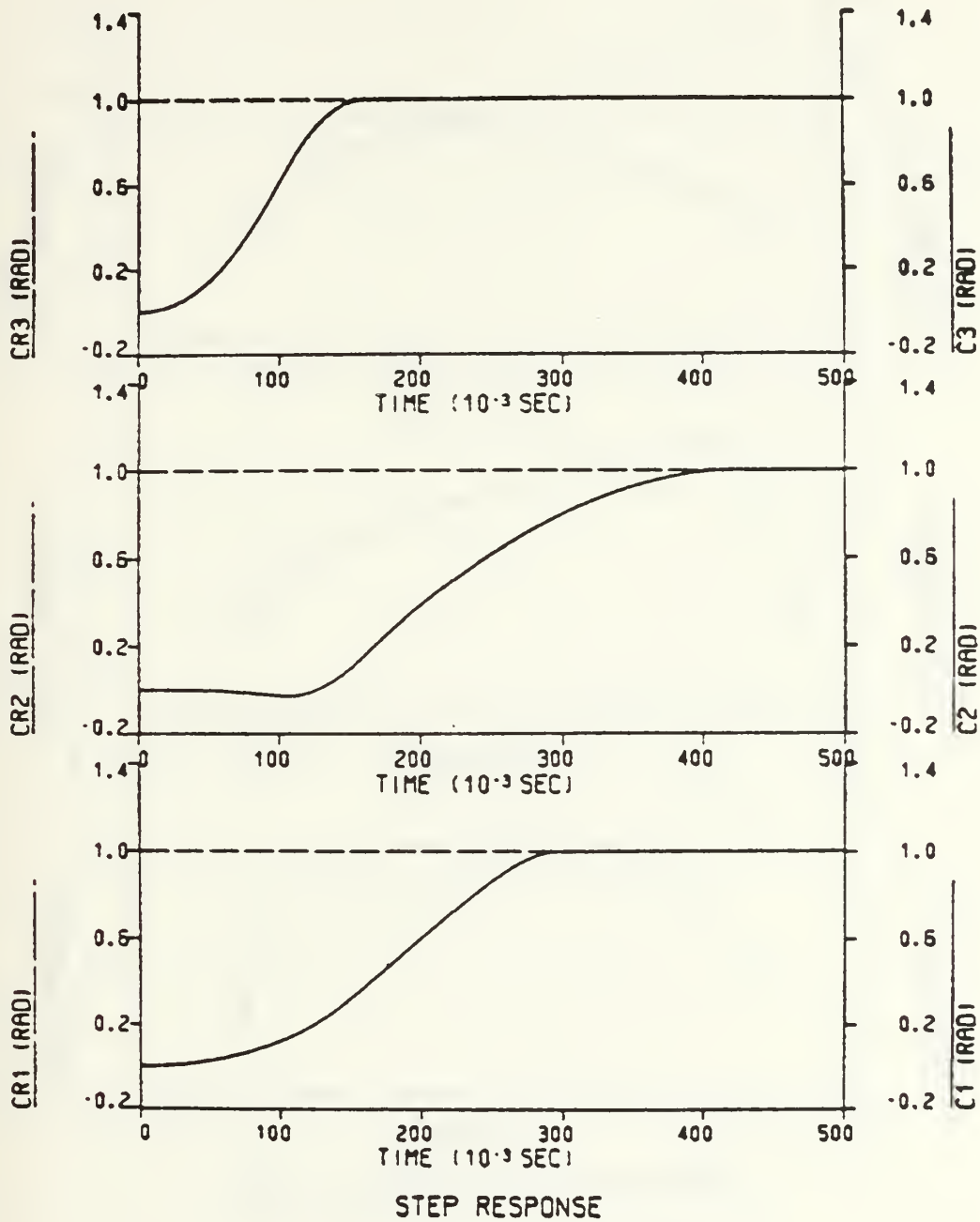
Figure 9.33   Step Response for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=0.21 - No Gravity)
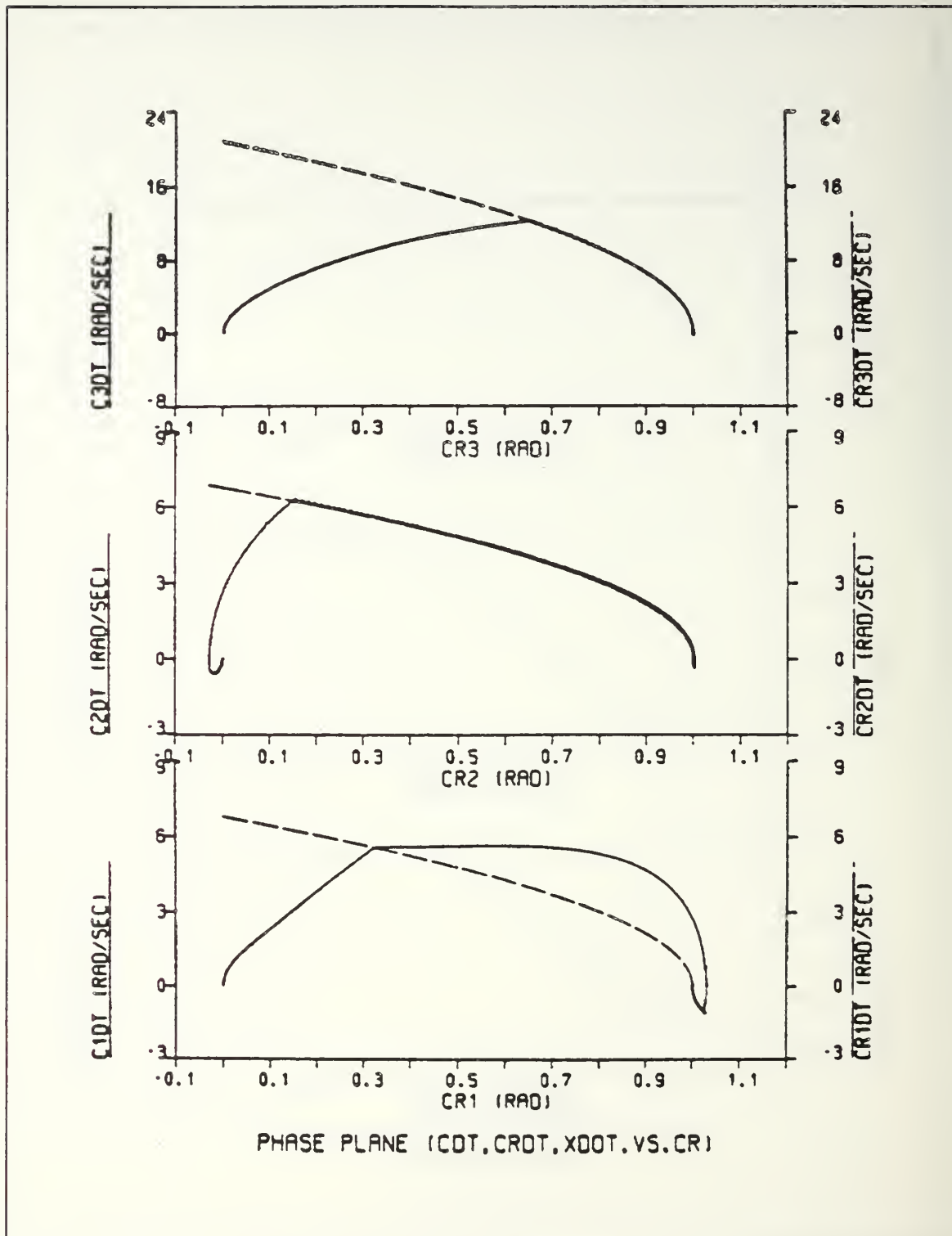
Figure 9.34   Phase Plane Trajectory for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=0.22 - No Gravity)
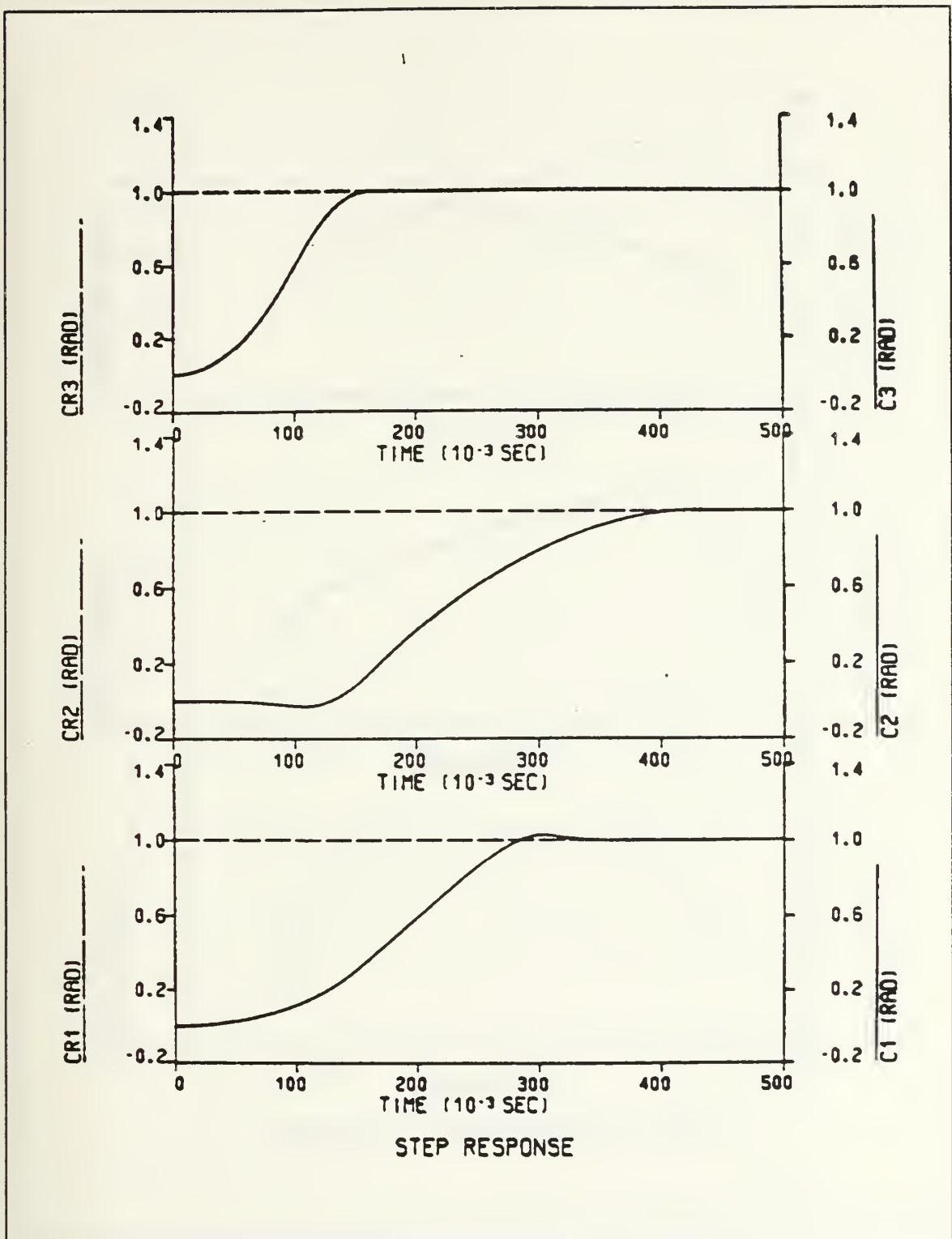
Figure 9.35 Step Response for Loaded Arm with
R Decreased and K_t, K_v, L Increased by 10%
(Load=0.22 - No Gravity)
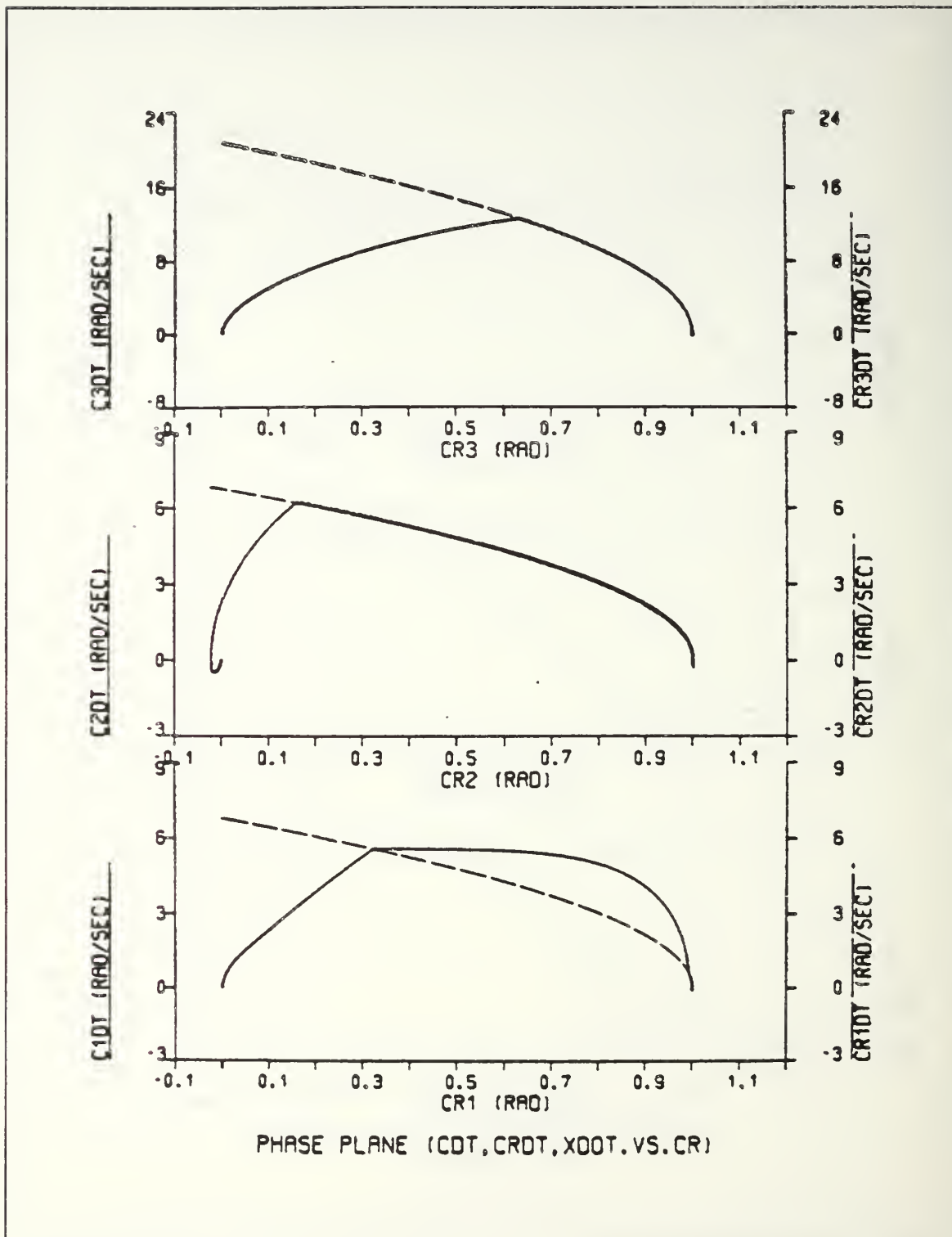
Figure 9.36   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.155 - No Gravity)

174
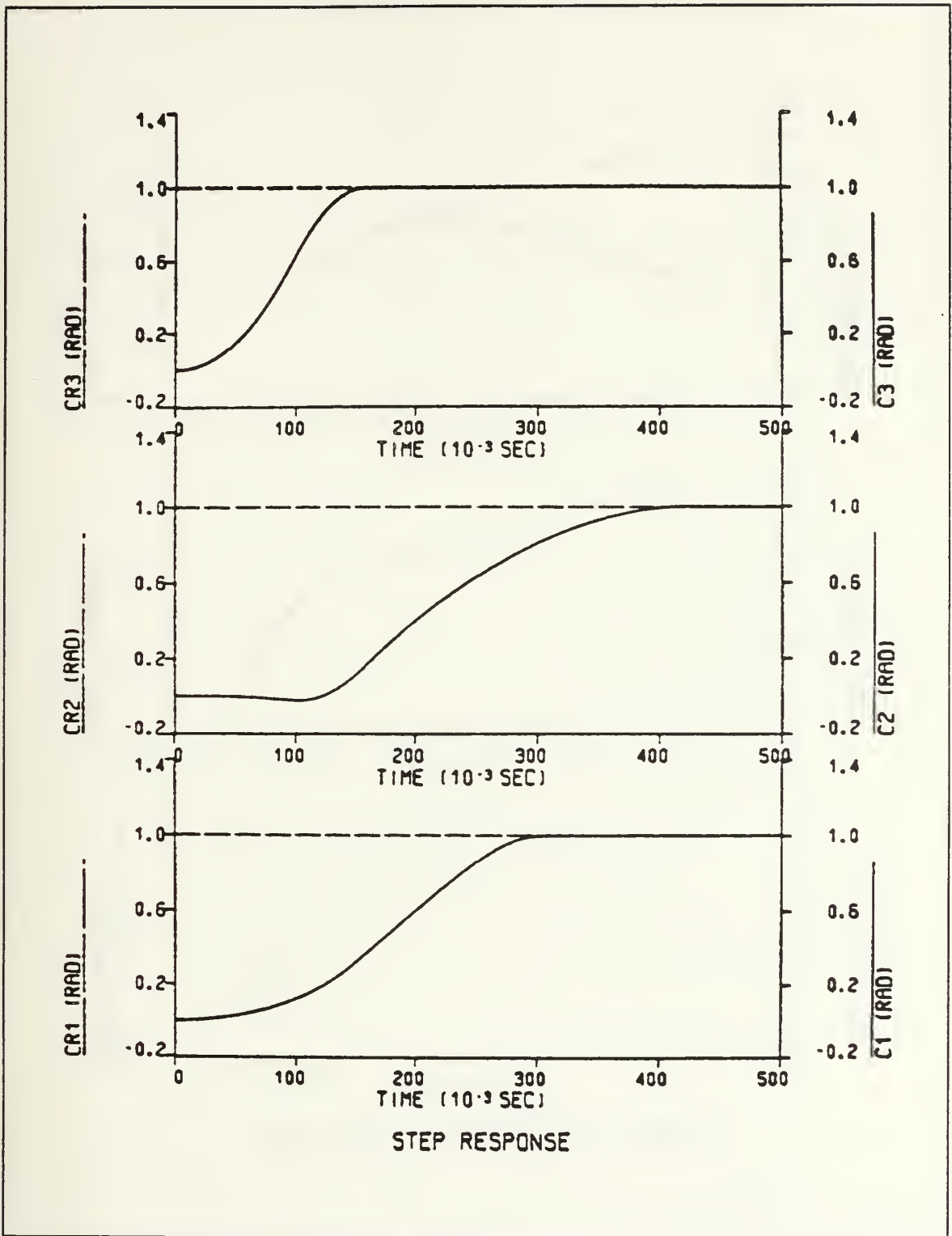
Figure 9.37   Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.155 - No Gravity)

175
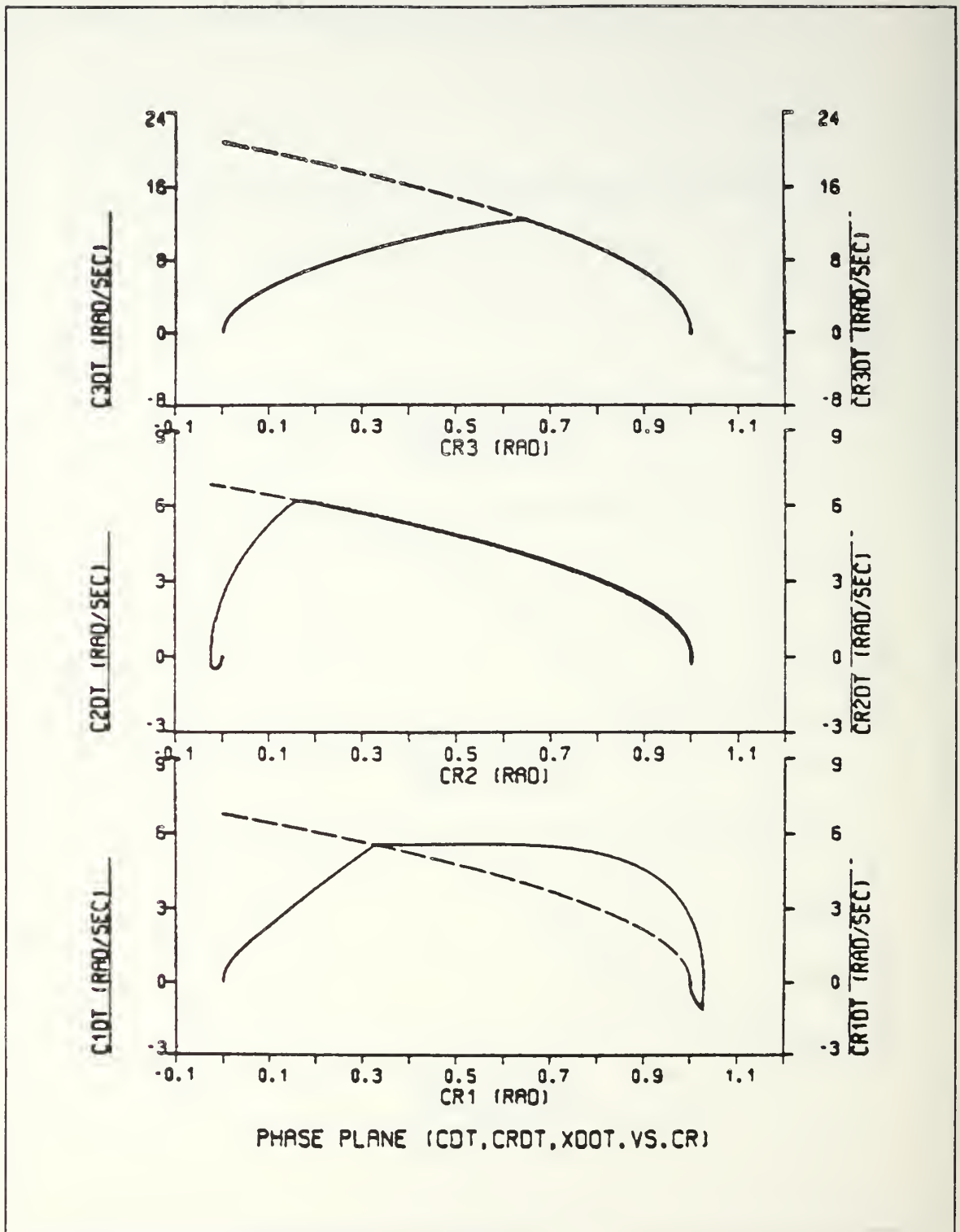
Figure 9.38   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.165 - No Gravity)

176
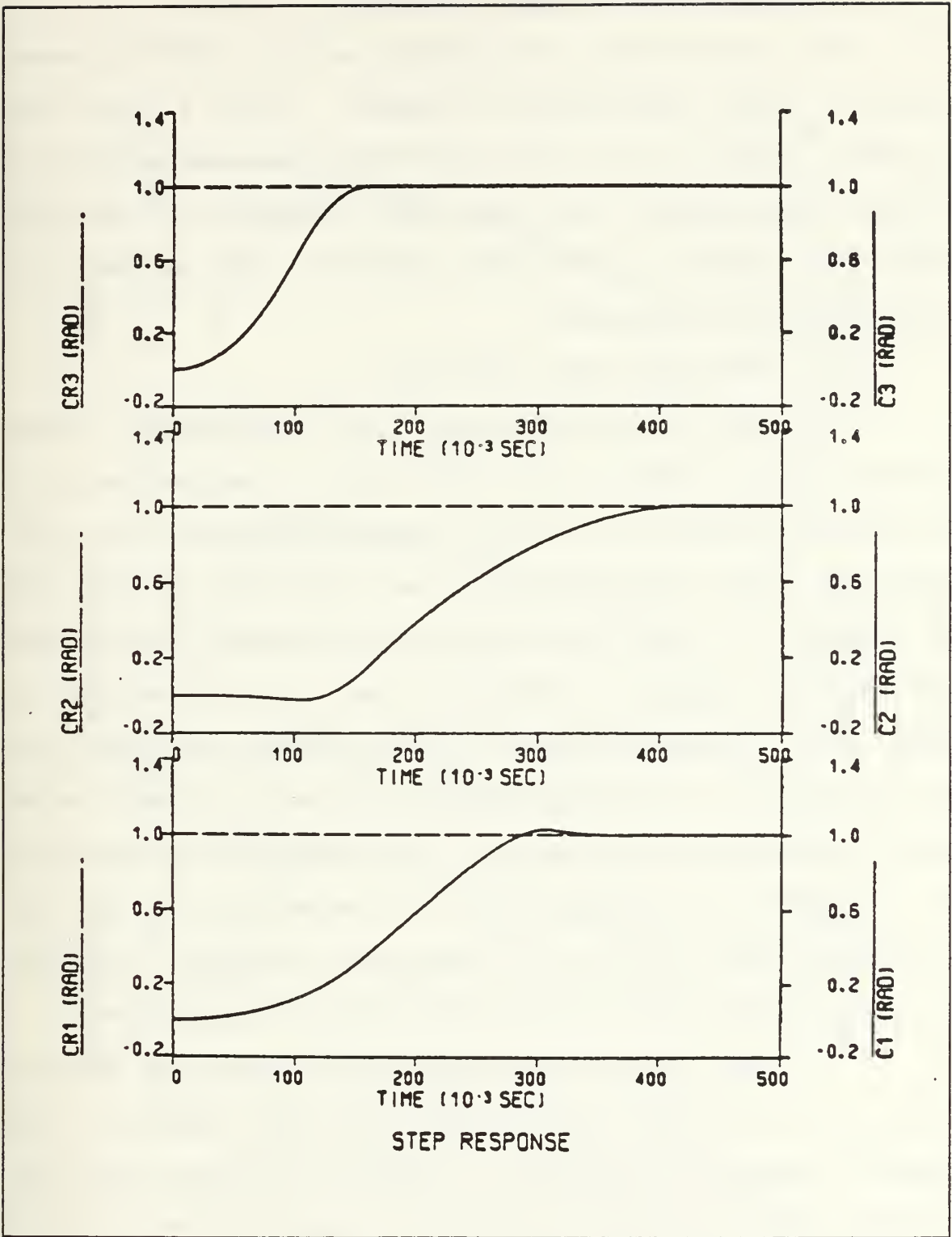
Figure 9.39   Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.165 - No Gravity)

177

2. Gravitational Torques Included

The same tests are applied to the adaptive model operating under gravitational torques. First a unit step is used as input in all joints and then some combinations of positive and negative unit step input commands are used for JOINT2 and JOINT3, to test the system for the existence of excess interactive torques.

a. Different Load Conditions

The phase plane plot of the unloaded arm, which is depicted in Figure 9.40, show that at the beginning of the movement JOINT2 servo motor develops negative velocity, which has larger magnitude than the respective one when the arm operates in a no gravitational environment because the gravitational torque is added to the torque developed on JOINT2 due to coupling inertia between JOINT2 and JOINT3. As the acceleration of JOINT3 starts decreasing, JOINT2 servo motor overcomes the disturbance torque and starts moving in the commanded direction. When it reaches the curve, it follows the curve down to the commanded position. The other two joints show very good curve following characteristics.

The step response and error curves of Figures 9.41 and 9.42 show clearly the motion of JOINT2 in the opposite direction, initially. Figure 9.41 shows also that it takes about 100 msec for JOINT2 to overcome the reaction torque. The time required for the movement to be completed is read as 420 msec, which is 25% increased comparing to the
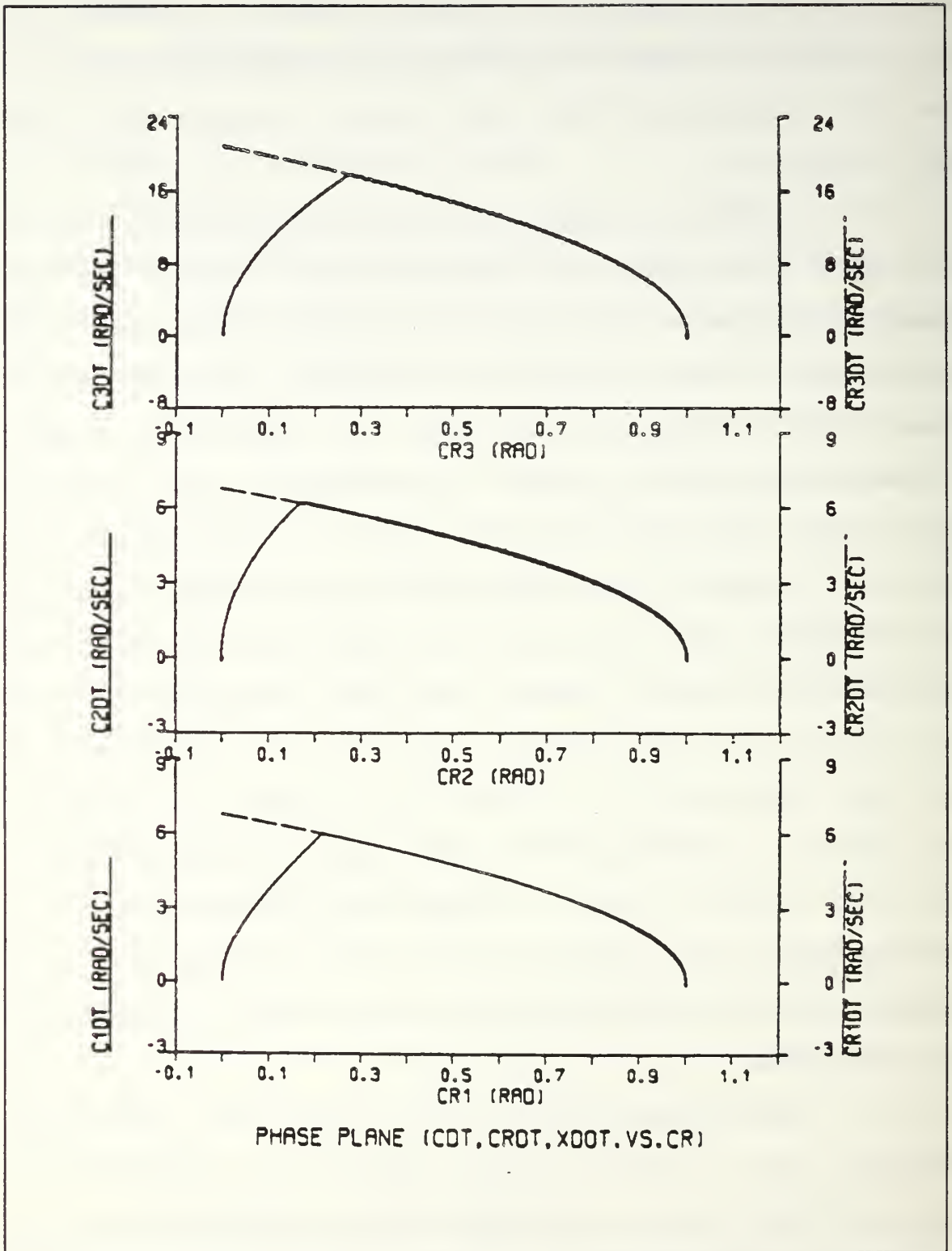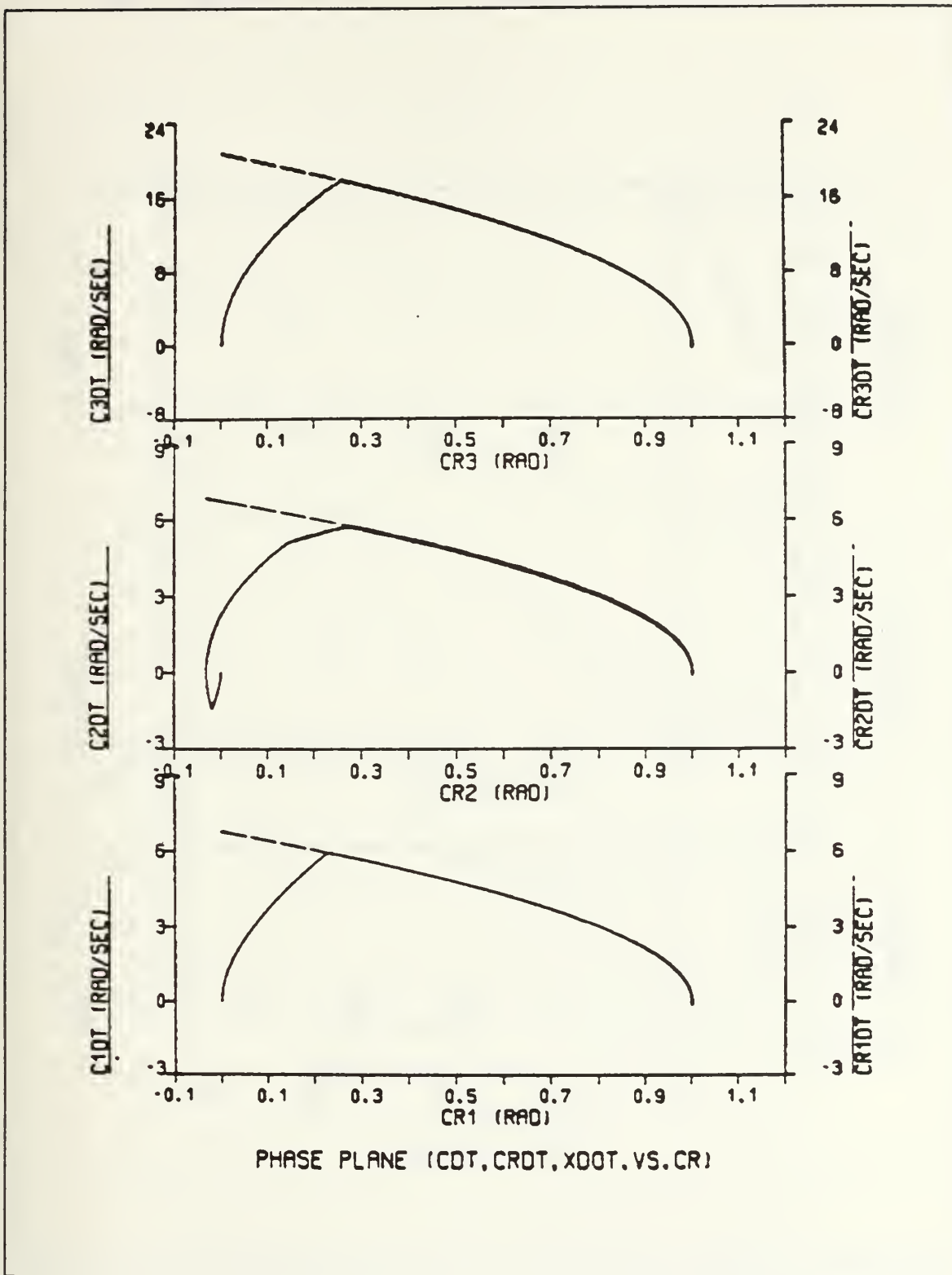
Figure 9.40   Phase Plane Trajectory for Unloaded Arm
(With Gravity)
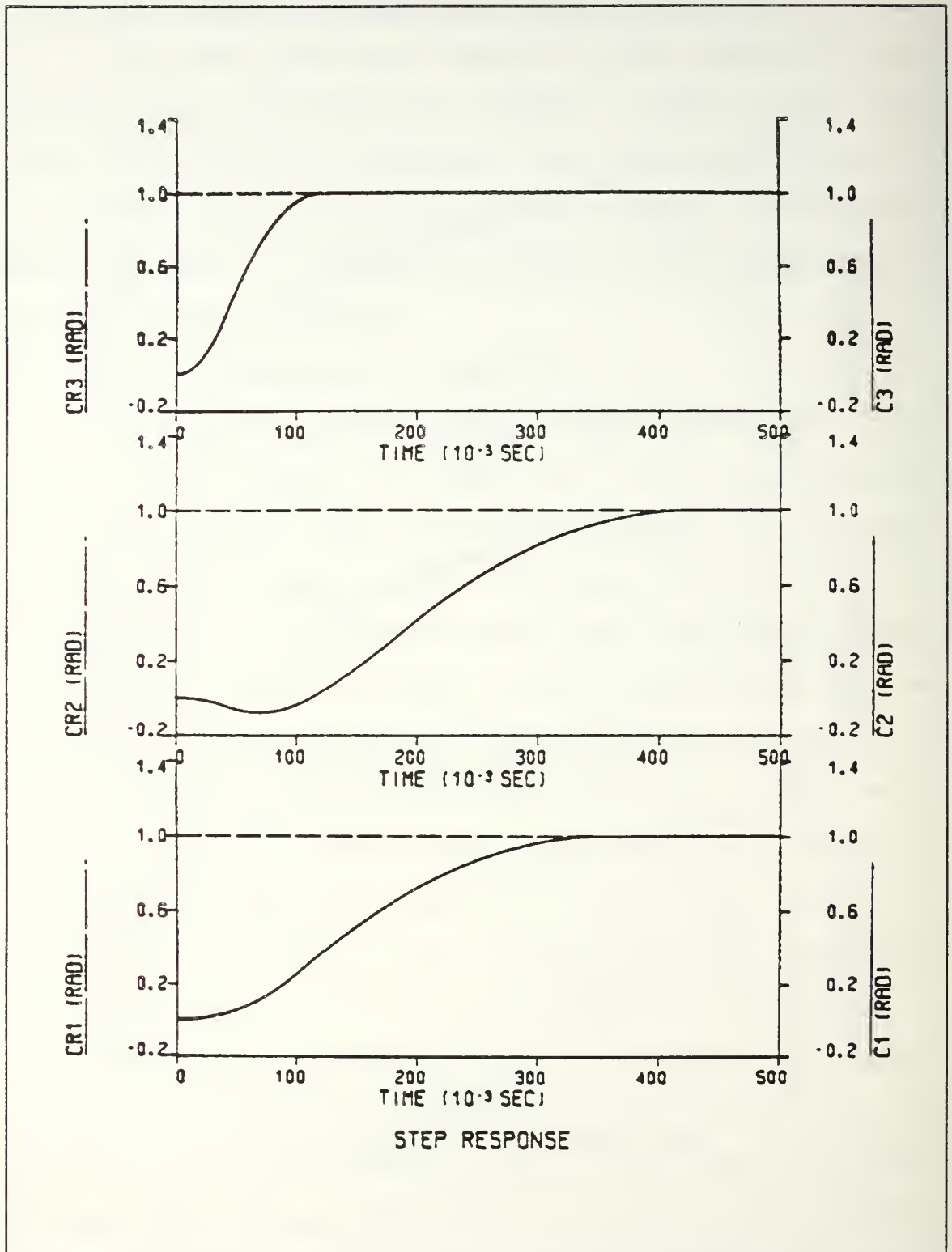
Figure 9.41 Step Response for Unloaded Arm
(With Gravity)

180
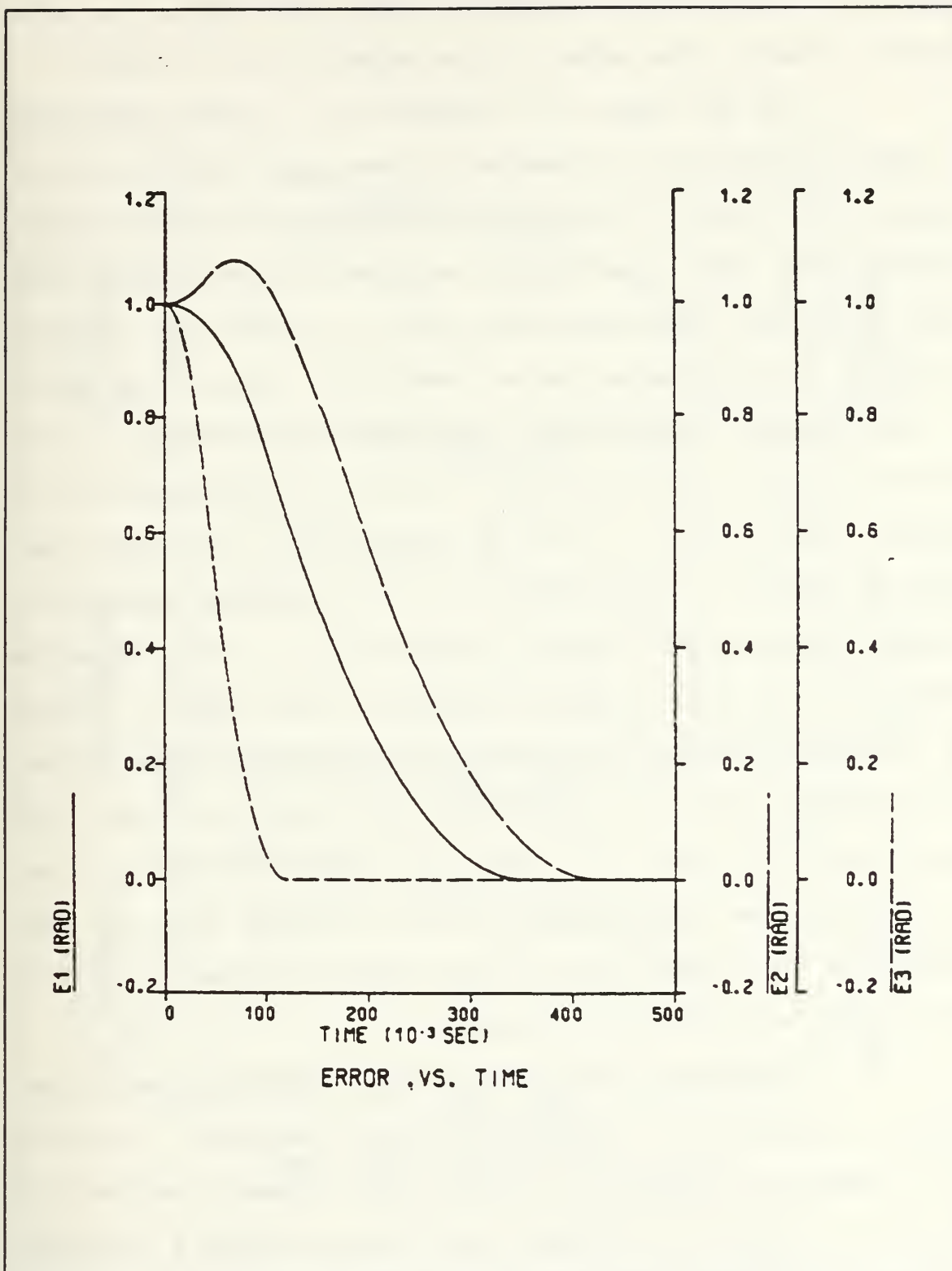
Figure 9.42 Error Curves for Unloaded Arm
(With Gravity)

181

respective time for the no gravity case. The simulation data reveal a steady state error of the order of 10^{-5} inches.

In the case of a loaded arm, a larger magnitude of negative velocity is observed in the phase plane plots of Figures 9.43 and 9.46, because of the greater gravitational torques. The other two joints continue to show very good curve following characteristics and all joints are capable to follow the commanded curve immediately after they catch it. The maximum load that the designed robot arm can carry was found to be 0.195 $oz/in/sec^2$. From the step responses of Figures 9.44 and 9.47 it can be observed that the developed reaction torque at the beginning of the movement results in JOINT2 servo motor moving initially in the opposite direction. The joint mostly affected by the load is JOINT2 and the settling time for this joint increases proportional to the added load. The error curves of Figure 9.45 and 9.48 show that the error of JOINT2 at the beginning of the movement is also proportional to the carried load and the effects of the added load in the settling time of JOINT2 servo motor can be observed clearly.

In some applications the initially developed negative velocity of JOINT2 may cause problems. To avoid this undesired situation, a step input command delayed by 150 msec is applied to JOINT3 servo motor. Figure 9.49 shows that the initially developed negative velocity of JOINT2 servo motor is eliminated and a zig - zag shape in its phase
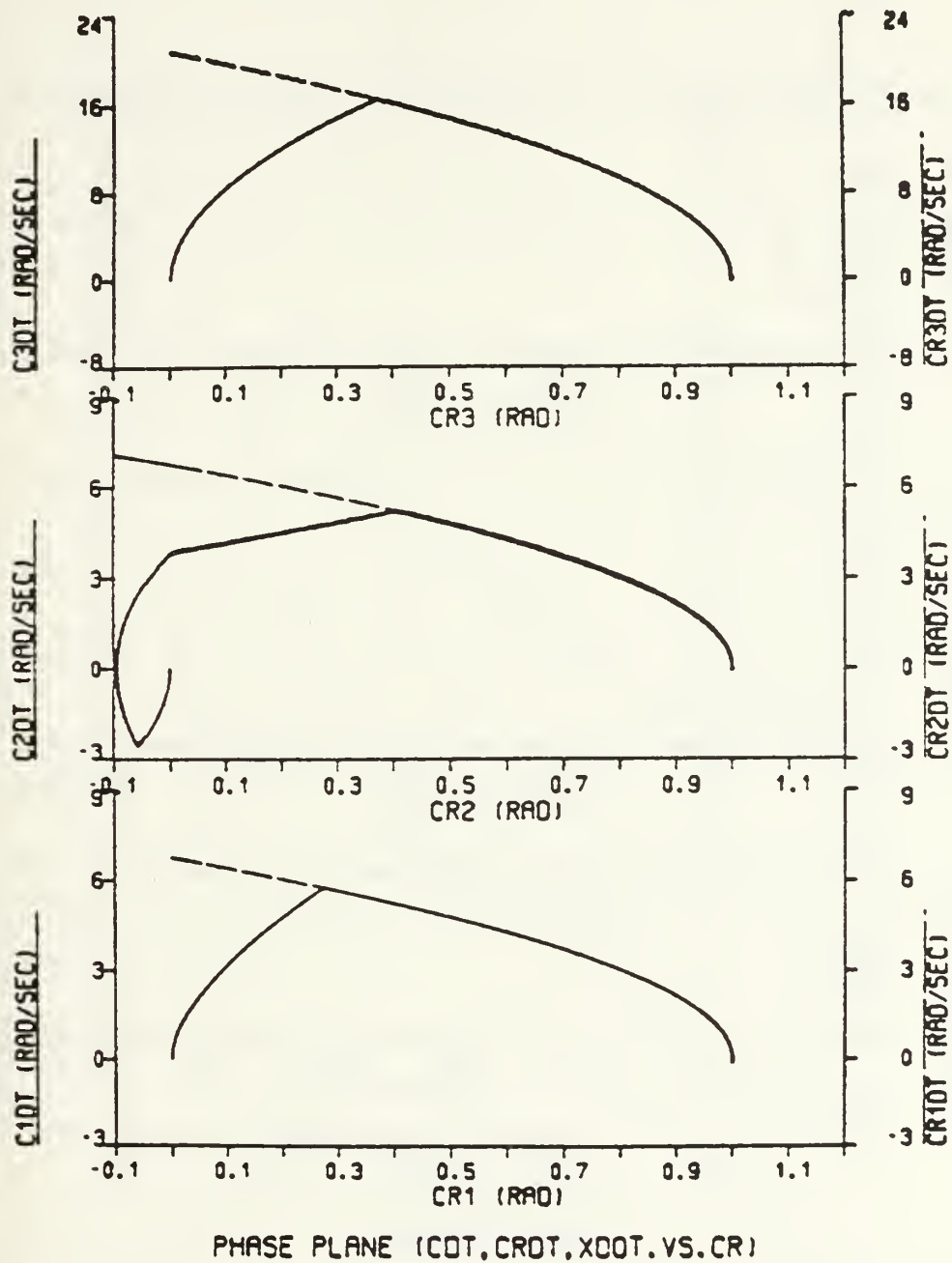
Figure 9.43   Phase Plane Trajectory for Loaded Arm
(Load=0.04 - With Gravity)
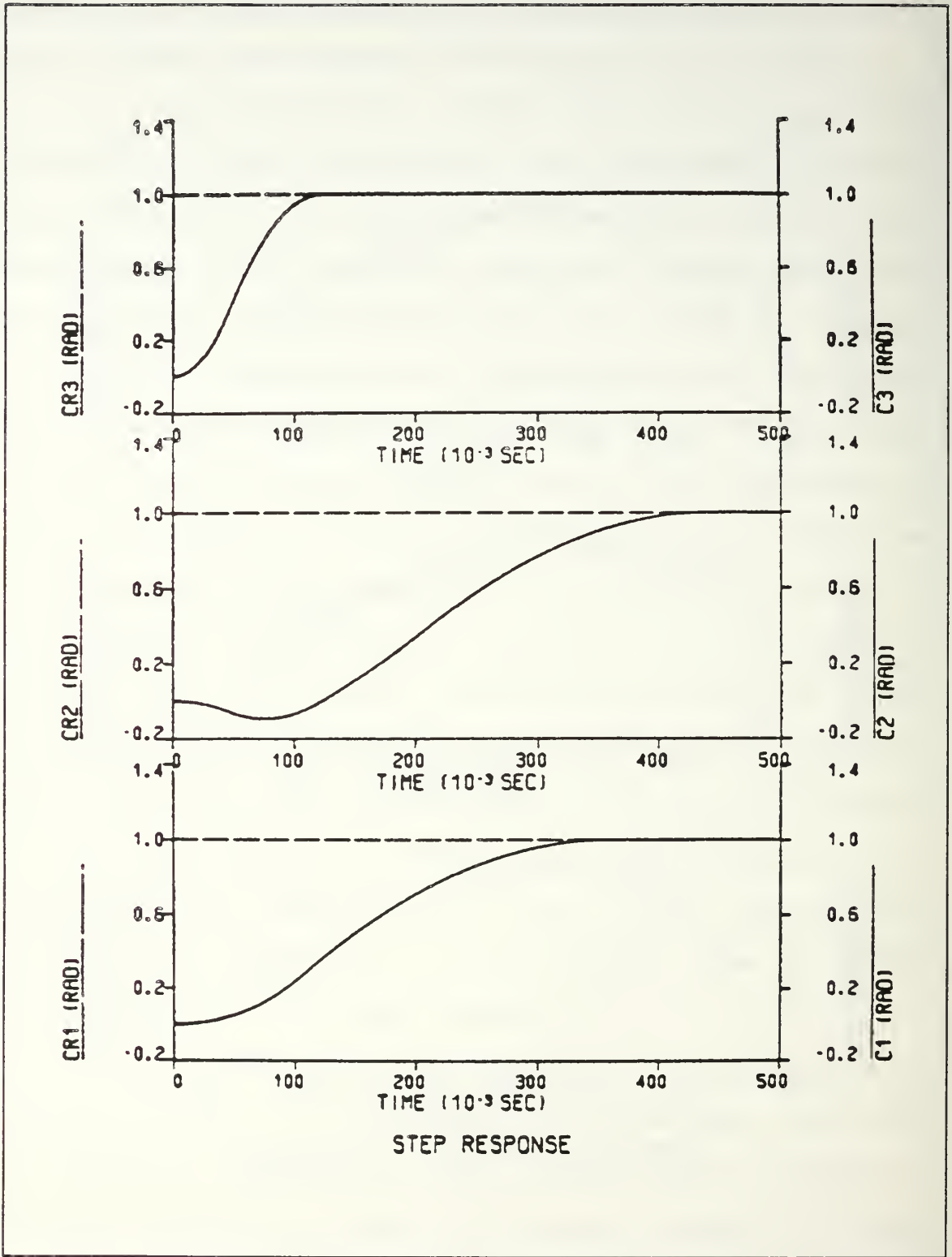
Figure 9.44   Step Response for Loaded Arm
(Load=0.04 - With Gravity)

184
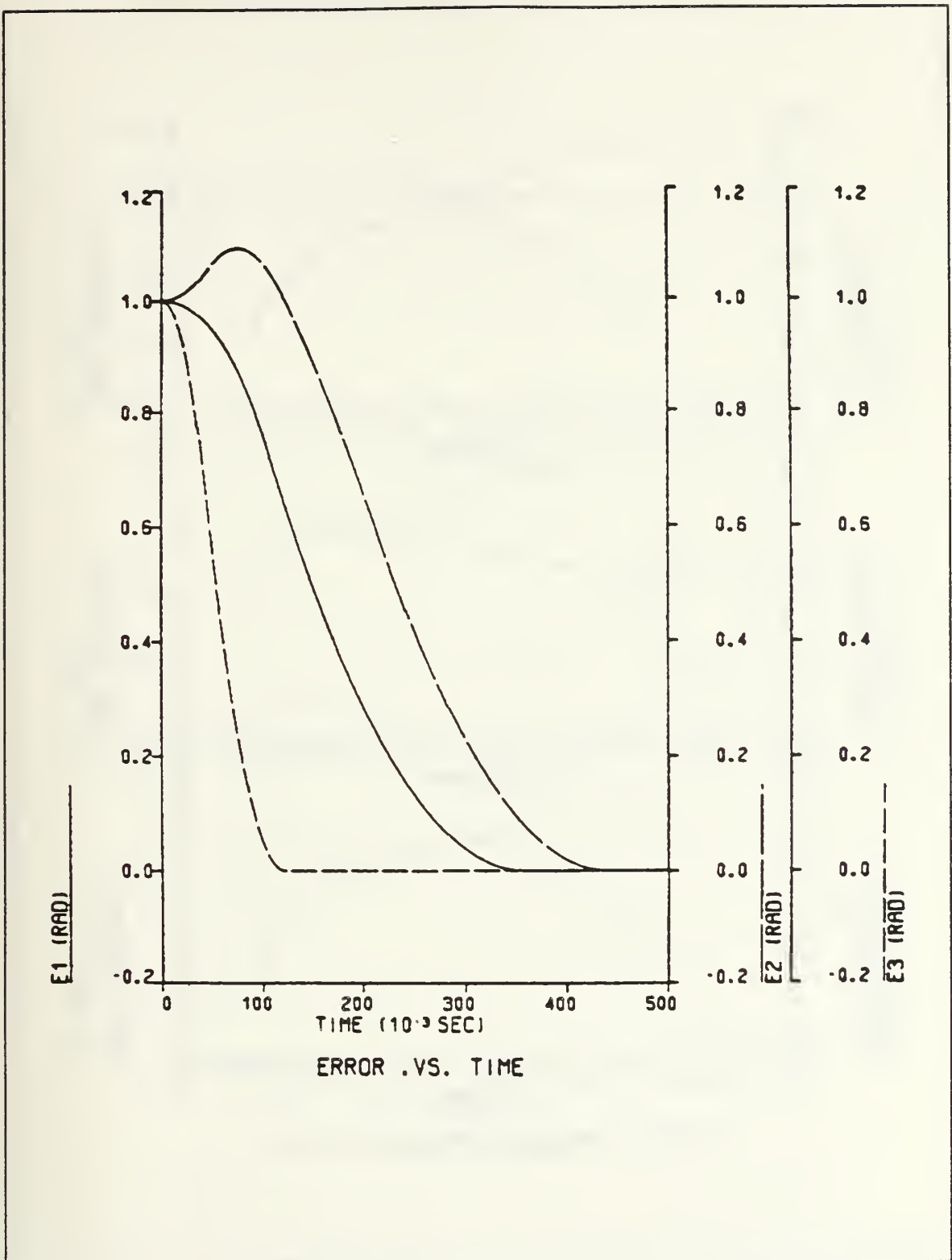
Figure 9.45 Error Curves for Loaded Arm
(Load=0.04 - With Gravity)
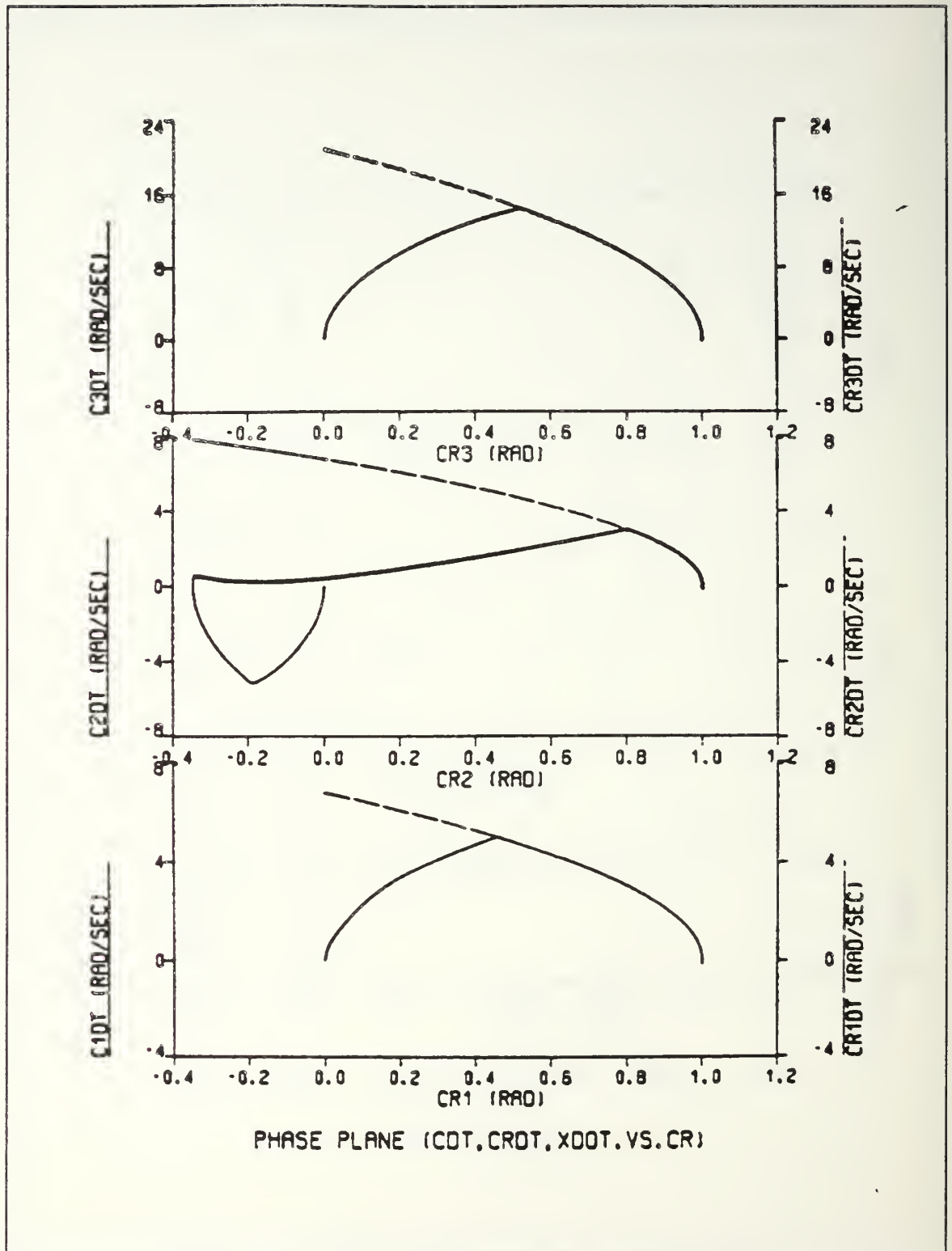
Figure 9.46   Phase Plane Trajectory for Loaded Arm
(Load=0.195 - With Gravity)
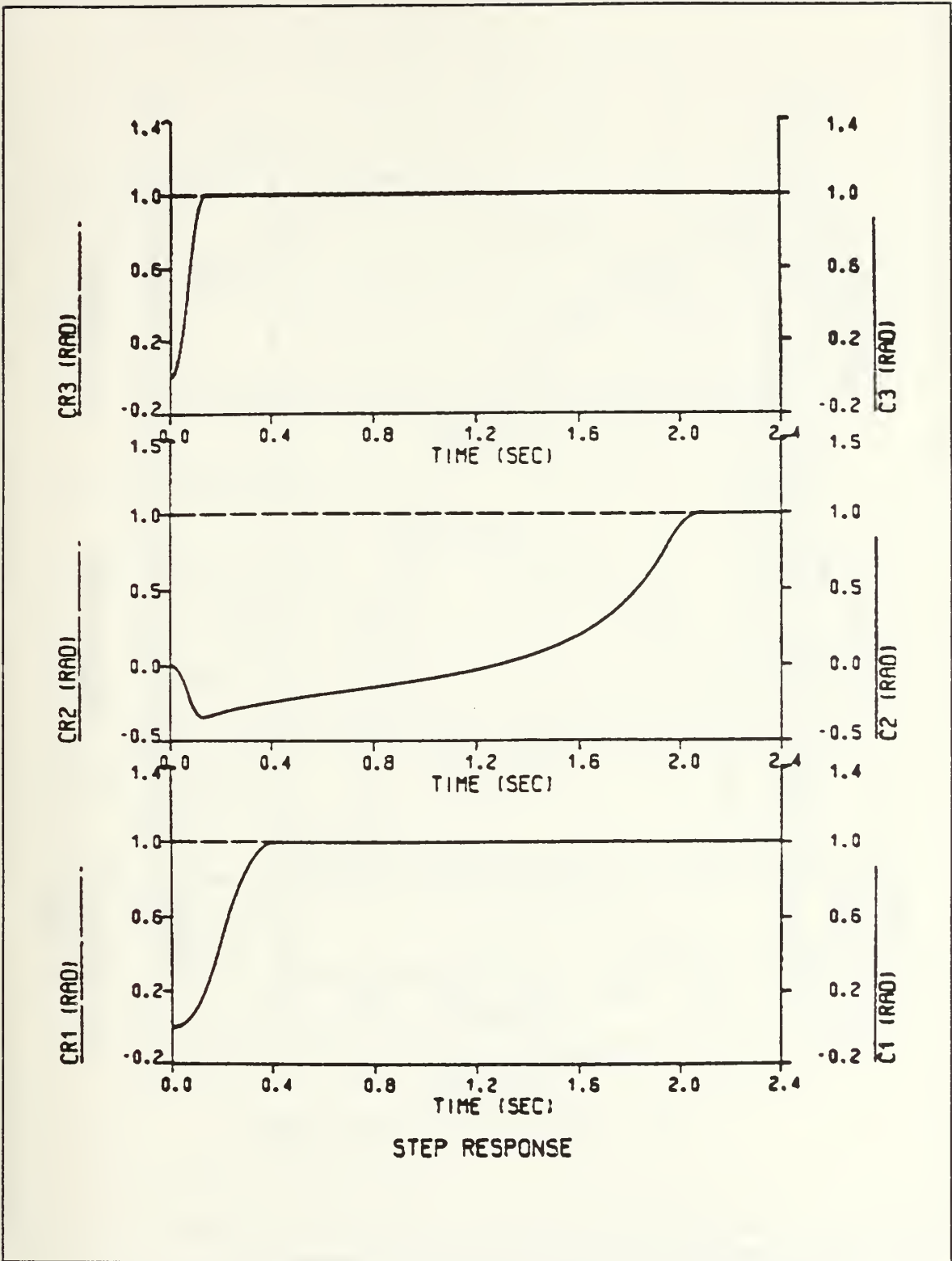
Figure 9.47   Step Response for Loaded Arm
(Load=0.195 - With Gravity)
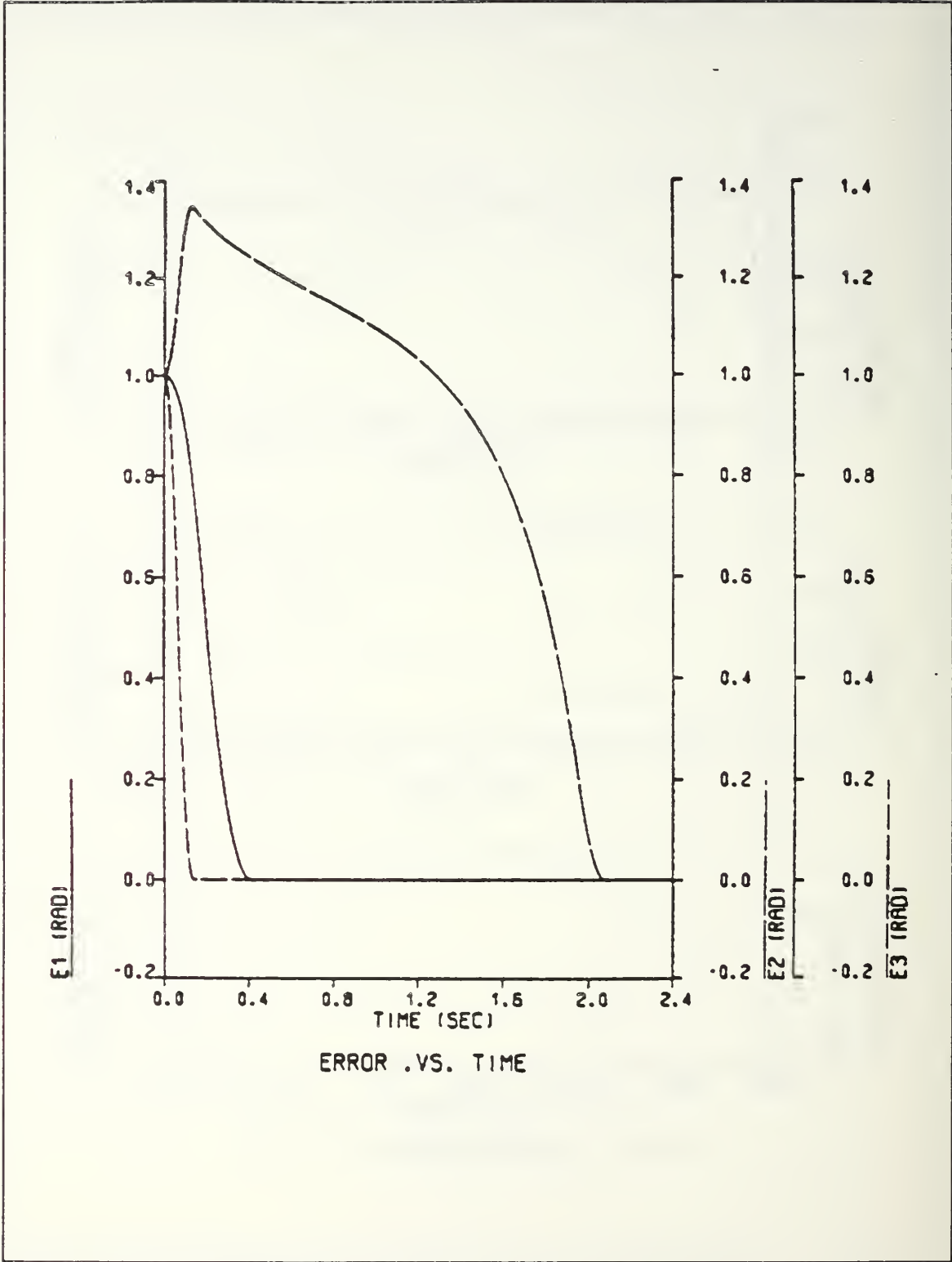
Figure 9.48 Error Curves for Loaded Arm
(Load=0.195 - With Gravity)
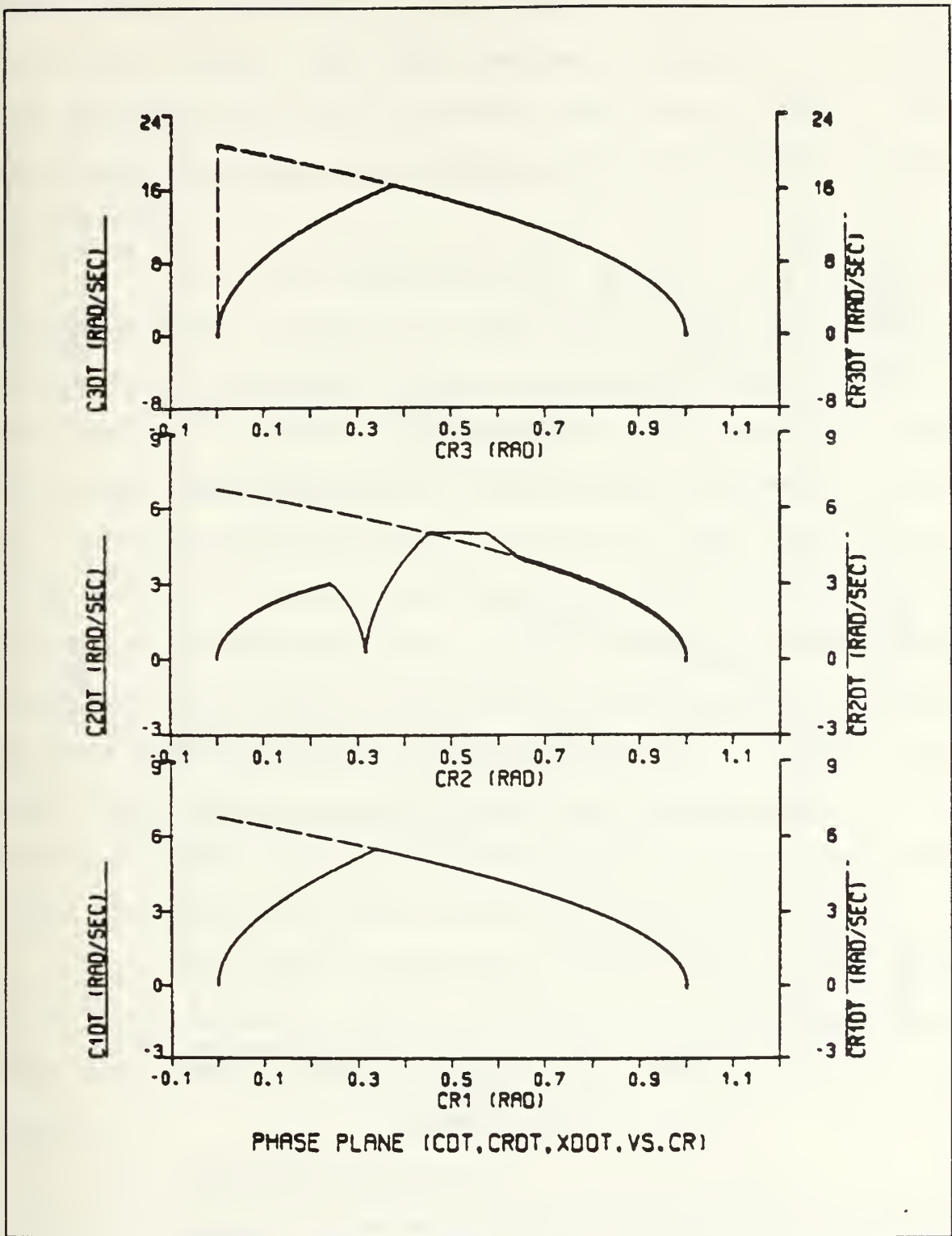
Figure 9.49   Phase Plane Trajectory for Loaded Arm
JOINT3 Servo Motor Input Delayed 150 msec
(Load=0.195 - With Gravity)

189
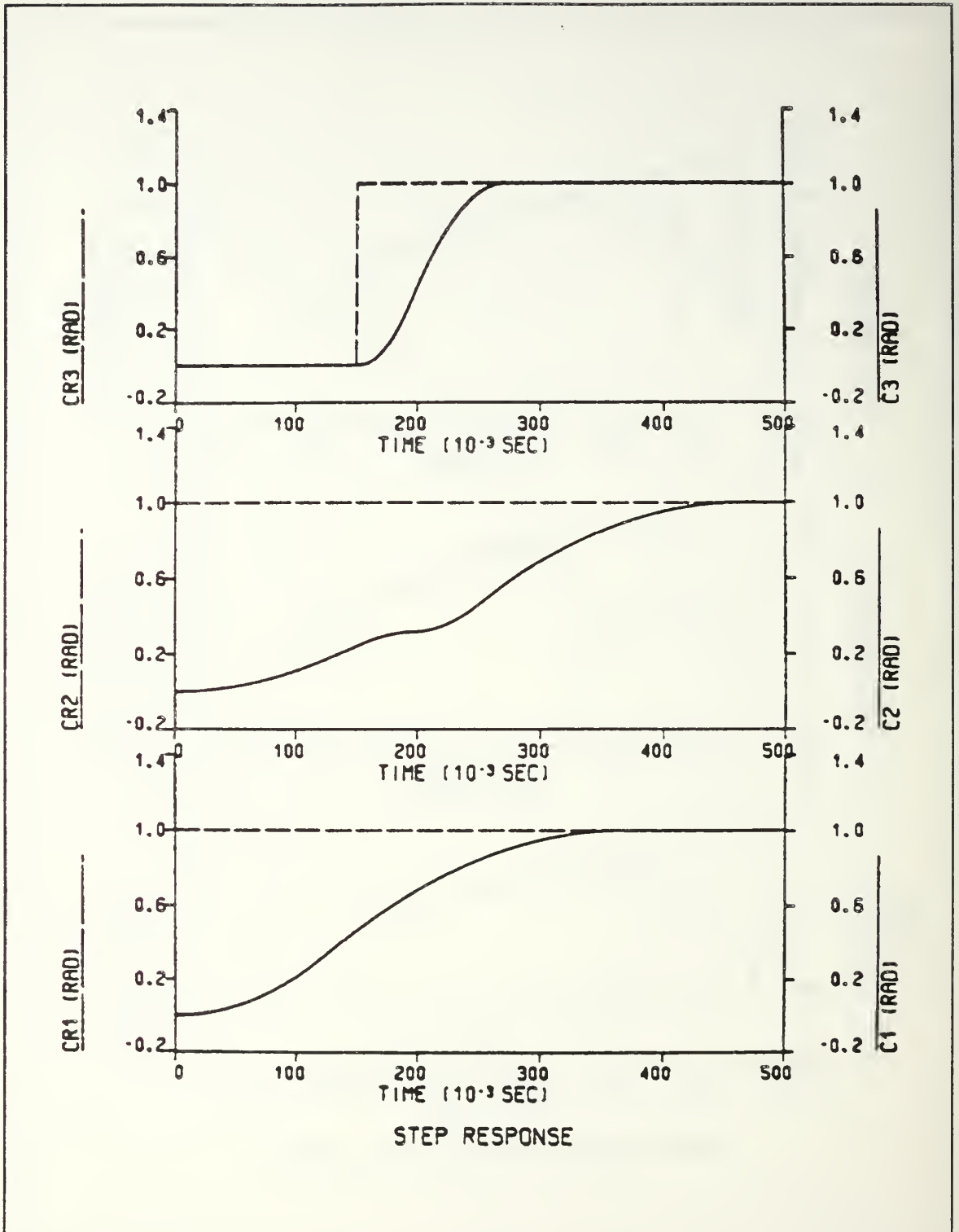
Figure 9.50 Step Response for Loaded Arm
JOINT3 Servo Motor Input Delayed 150 msec
(Load=0.195 - With Gravity)

190

plane plot is observed due to reaction torque produced by this servo motor. The step response of Figure 9.50 shows that the settling time for JOINT2 servo motor remains the same while the movement of JOINT2 in the opposite direction is eliminated.

Then some combinations of positive and negative unit step input commands are applied to JOINT2 and JOINT3 servo motors to simulate the movements of LINK2 and LINK3 in the direction of gravity. Figures 9.51 - 9.54 show that when both LINK2 and LINK3 move in the direction of gravity (move #2), JOINT3 overshoots for a relatively small load (0.06 $oz/in/sec^2$). The phase plane plots of these figures show that for a considerable part of their movement, JOINT1 and JOINT2 move with velocity greater than the commanded one. In the case that LINK2 moves in the direction of gravity and LINK3 moves against gravity (move #3), Figures 9.55 - 9.58 show that JOINT2 overshoots for a slightly larger load but JOINT1 and JOINT3 follows accurately the commanded curve.

The DSL/VS simulation program used to test the three link revolute robot operating in a gravitational environment under different load conditions is listed in Appendix J.

b. Disturbance Rejection

Making use of the DSL/VS simulation program listed in Appendix K, the same disturbances mentioned in part 1.b of this chapter are applied to the robot arm
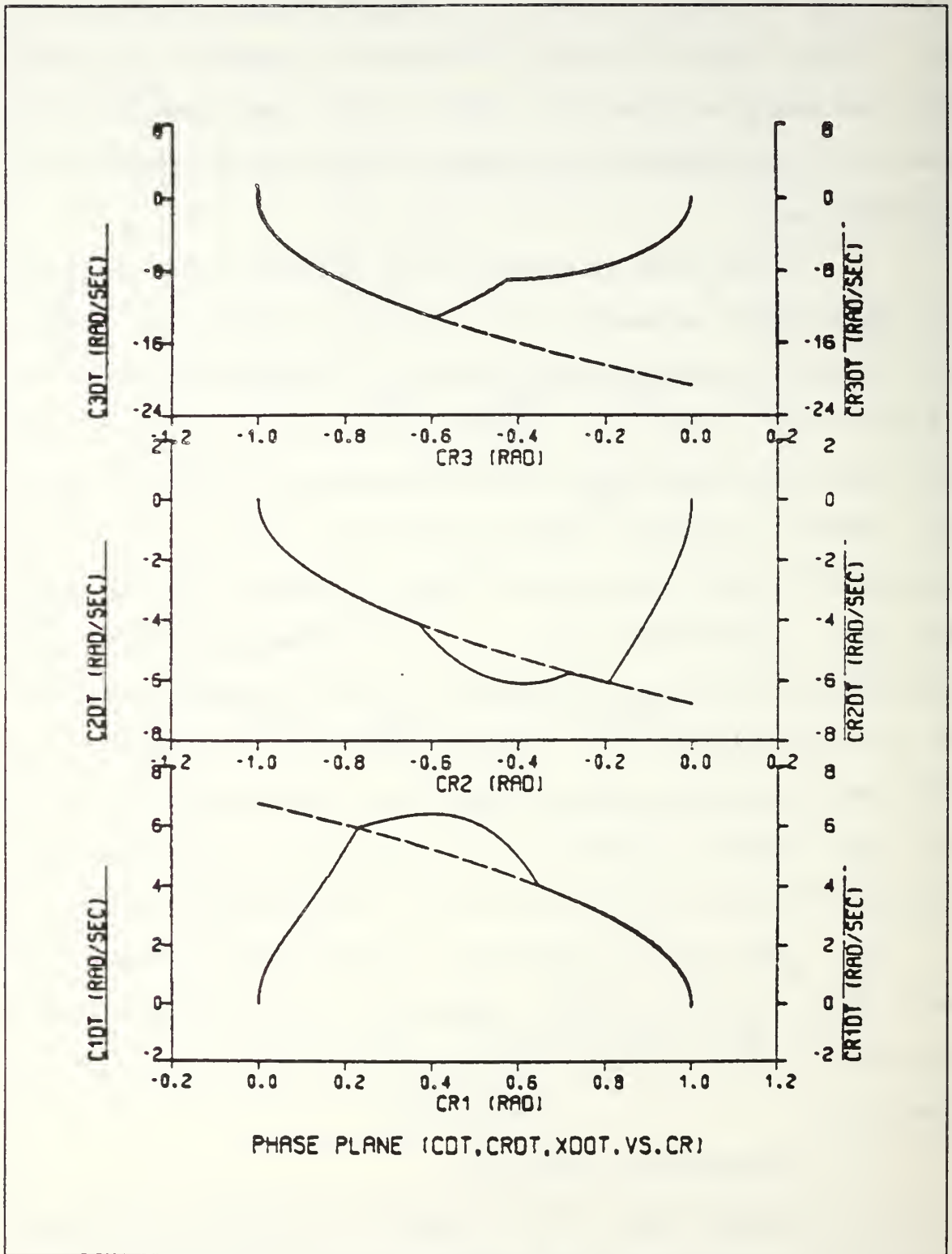
Figure 9.51   Phase Plane Trajectory for Move #2
(Load=0.055 - With Gravity)

192
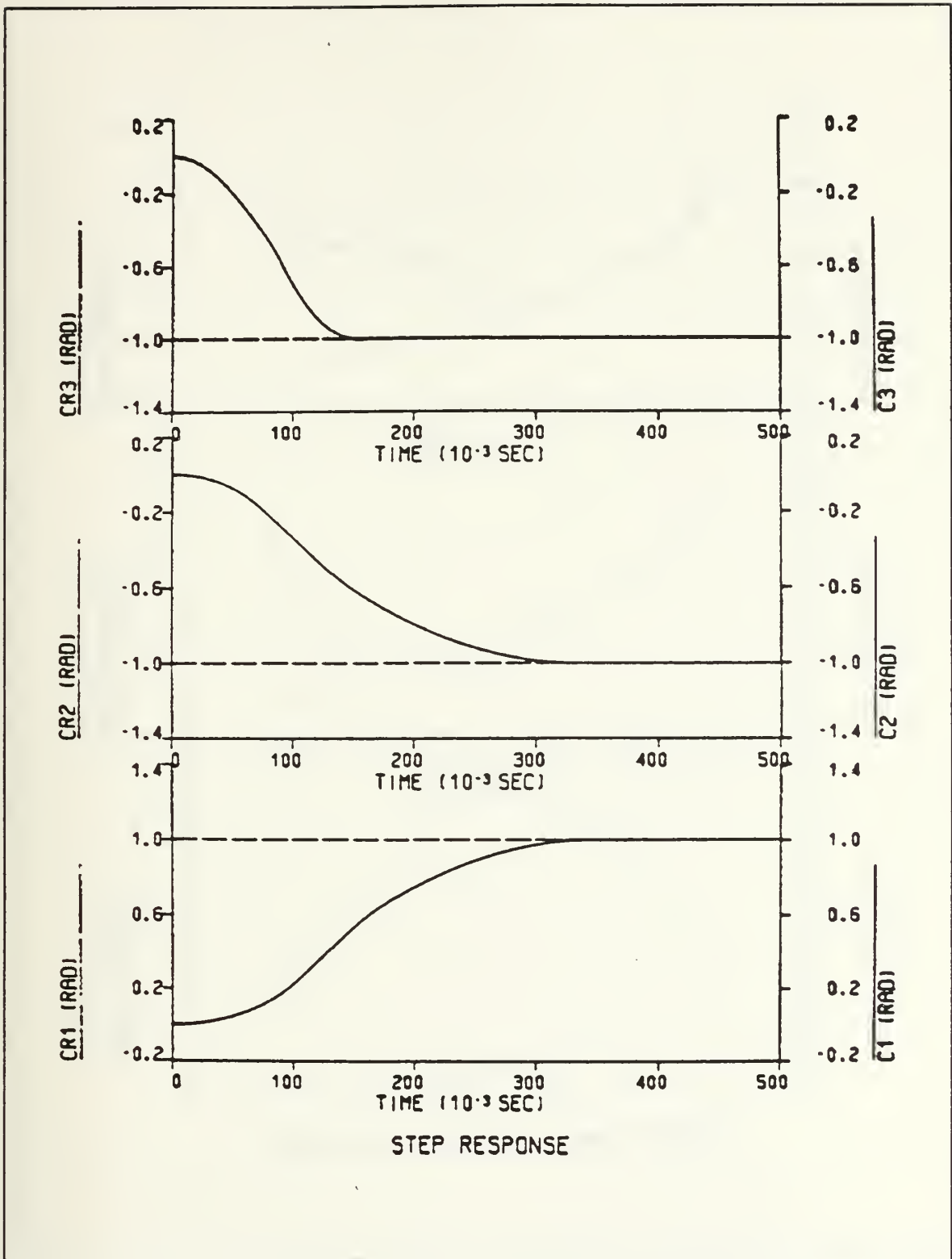
Figure 9.52   Step Response for Move #2
(Load=0.055 - With Gravity)
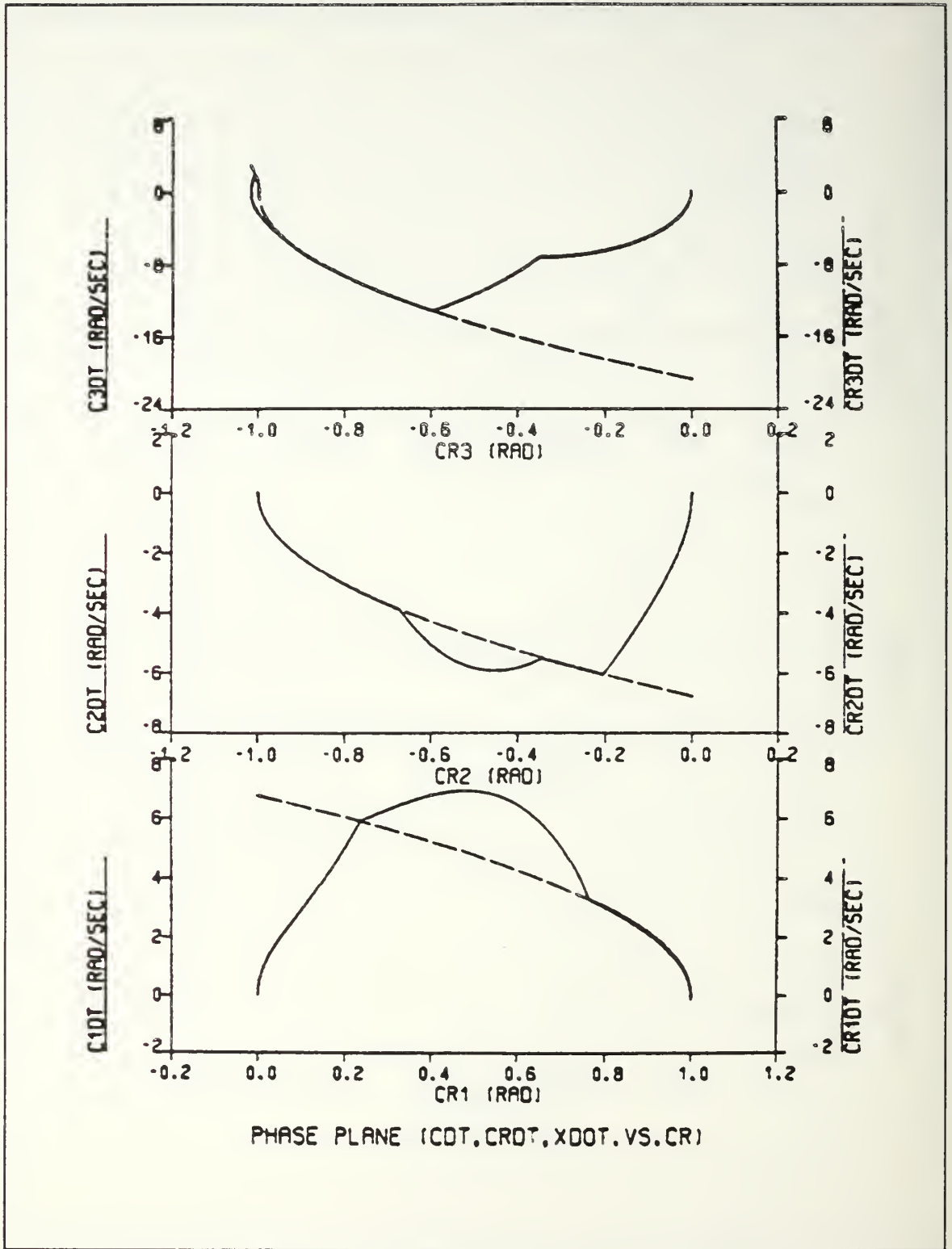
Figure 9.53   Phase Plane Trajectory for Move #2
(Load=0.065 - With Gravity)

194
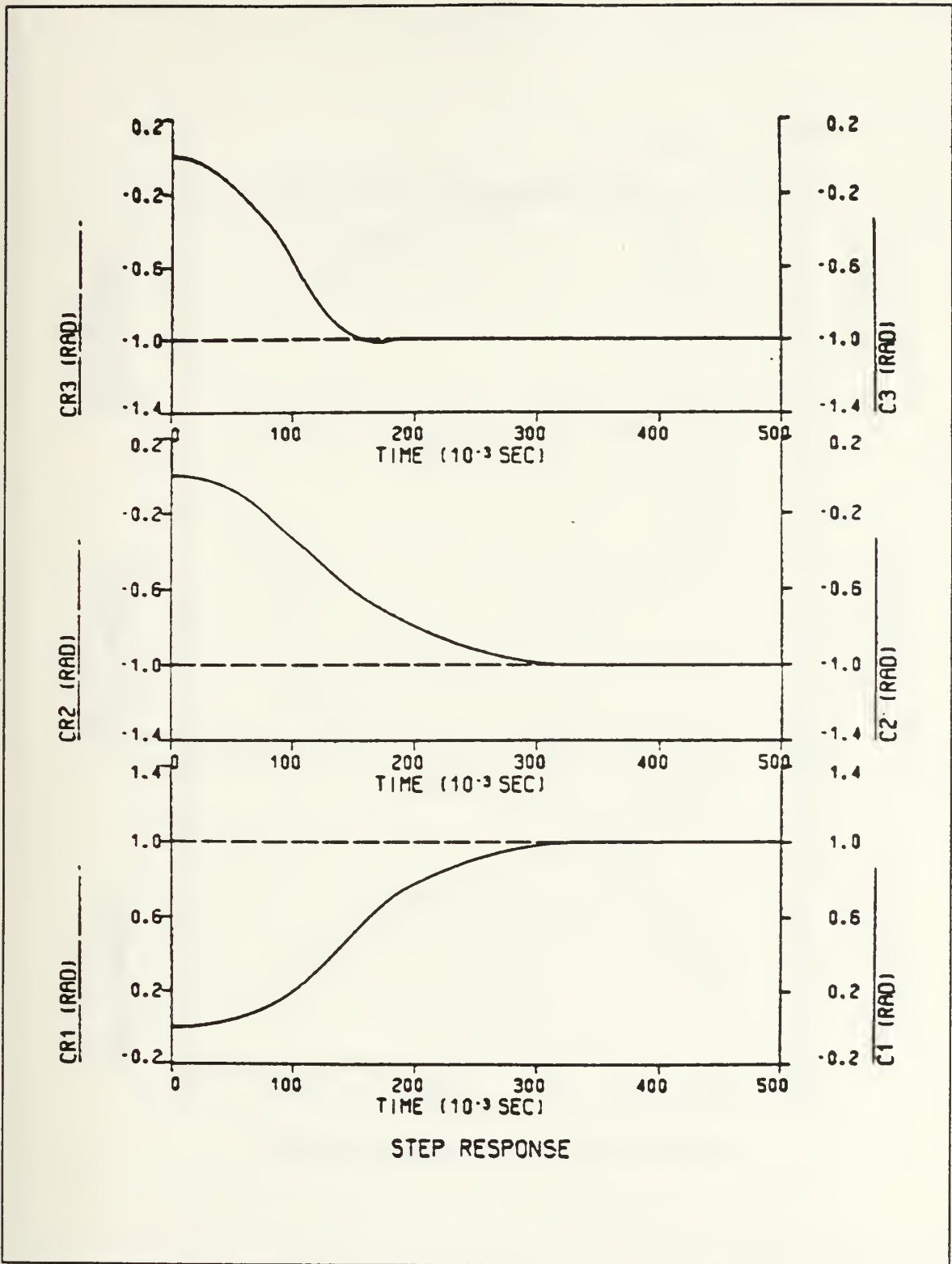
Figure 9.54   Step Response for Move #2
(Load=0.065 - With Gravity)
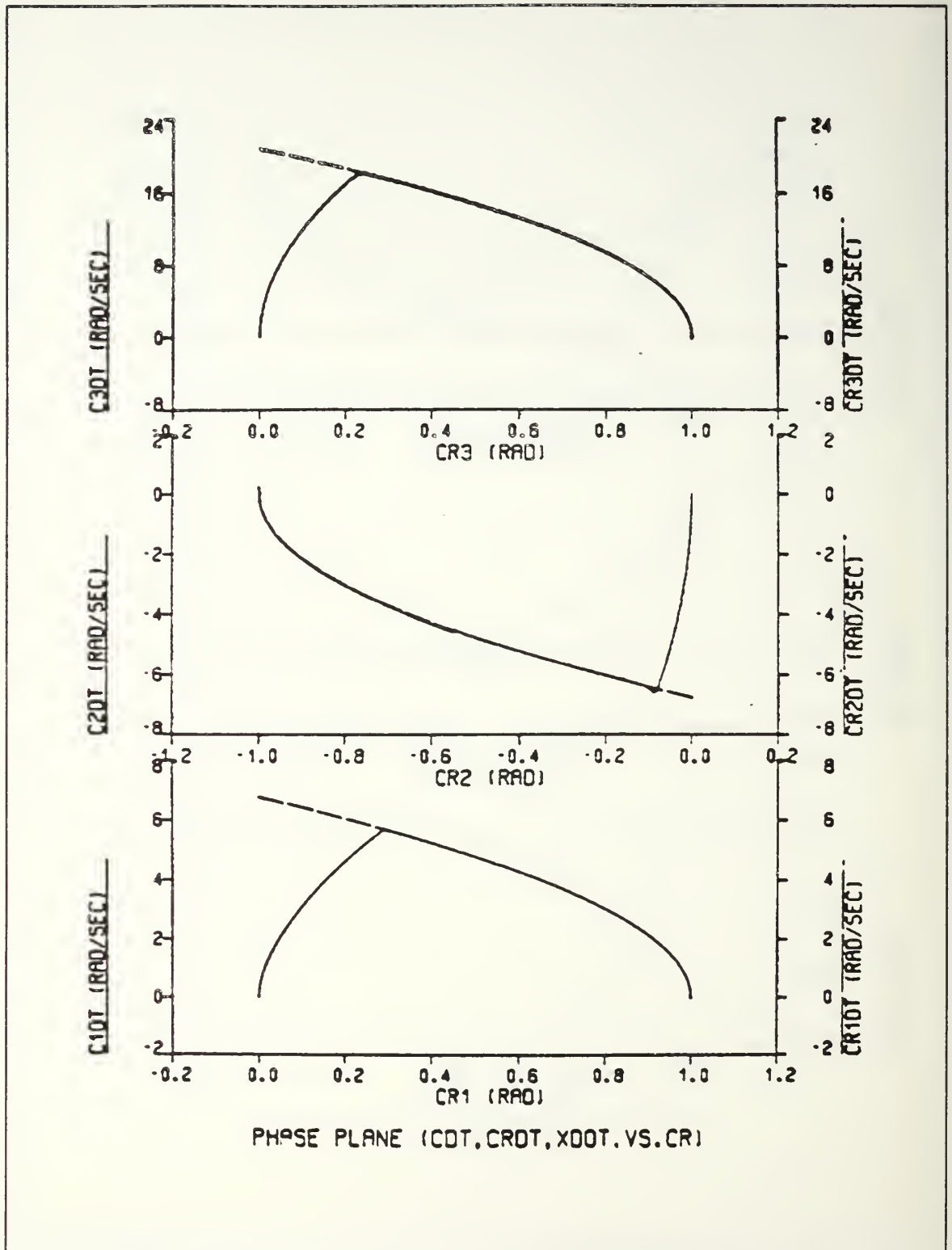
Figure 9.55   Phase Plane Trajectory for Move #3
(Load=0.06 - With Gravity)
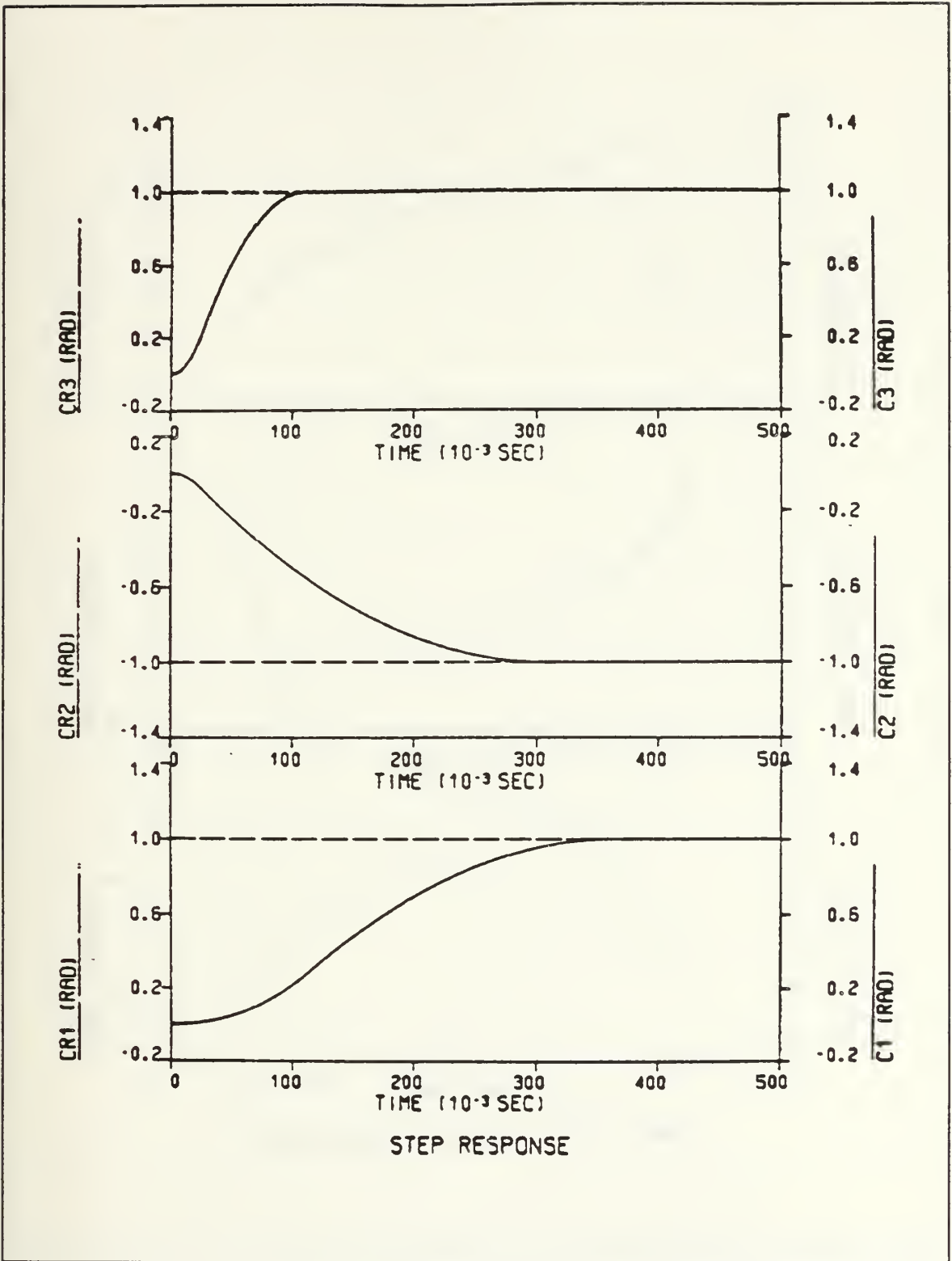
Figure 9.56   Step Response for Move #3
(Load=0.06 - With Gravity)
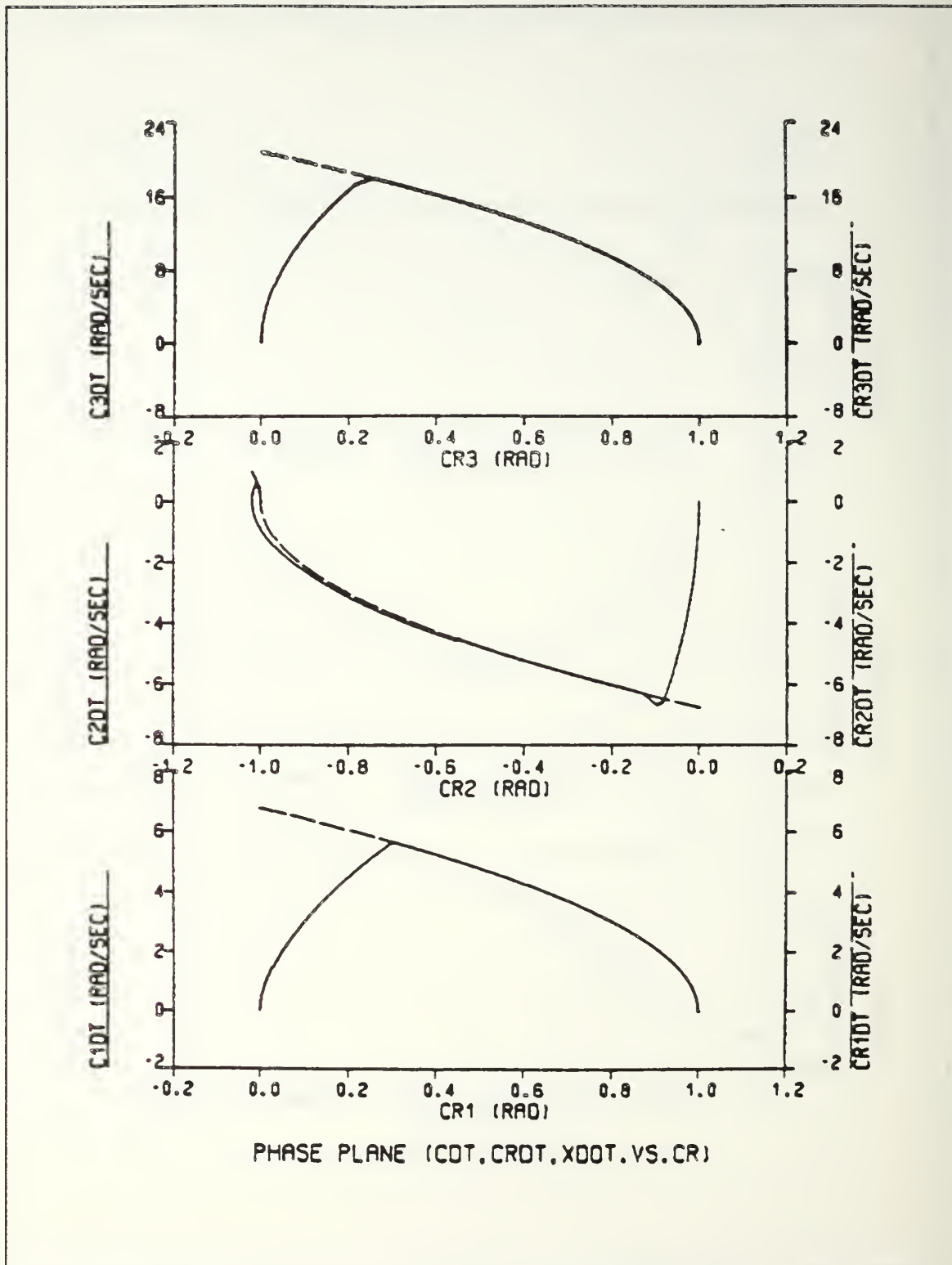
Figure 9.57   Phase Plane Trajectory for Move #3
(Load=0.07 - With Gravity)
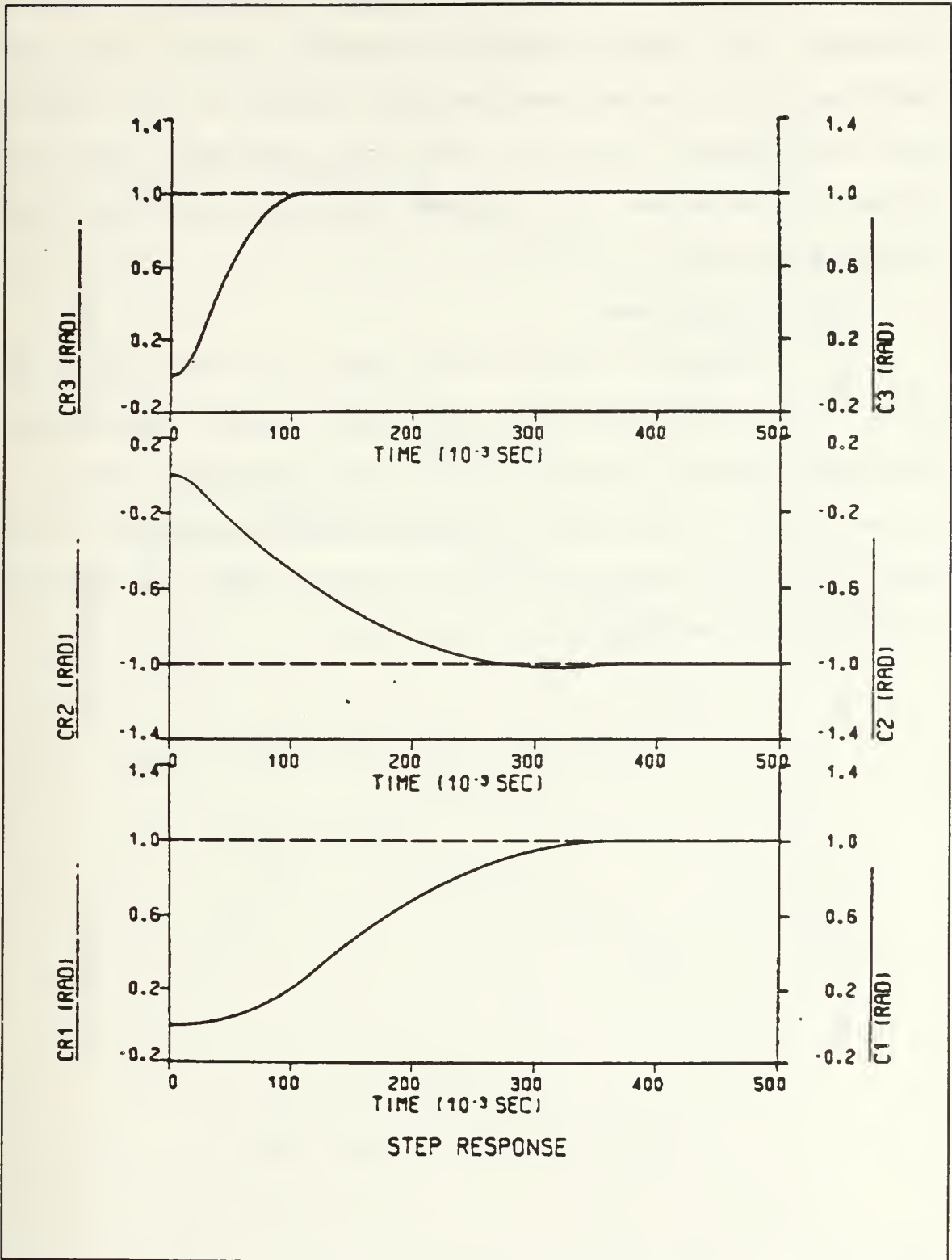
Figure 9.58   Step Response for Move #3
(Load=0.07 - With Gravity)

199

operating under gravitational torques. Simulation results illustrate the nearly identical behavior of the robot arm operating either under gravitational torques or in a gravity free environment. Figures 9.59 and 9.60 show that the applied disturbances are rejected and all joints reach the desired position.

c. Robustness

Finally, the revolute robot is tested again for slight (10%) variation of the servo motors parameters. Simulation results do not reveal any noteworthy change in the behavior of the system which is almost identical to the behavior of the system in the no gravity case, as depicted in the indicative Figures 9.61 and 9.62.
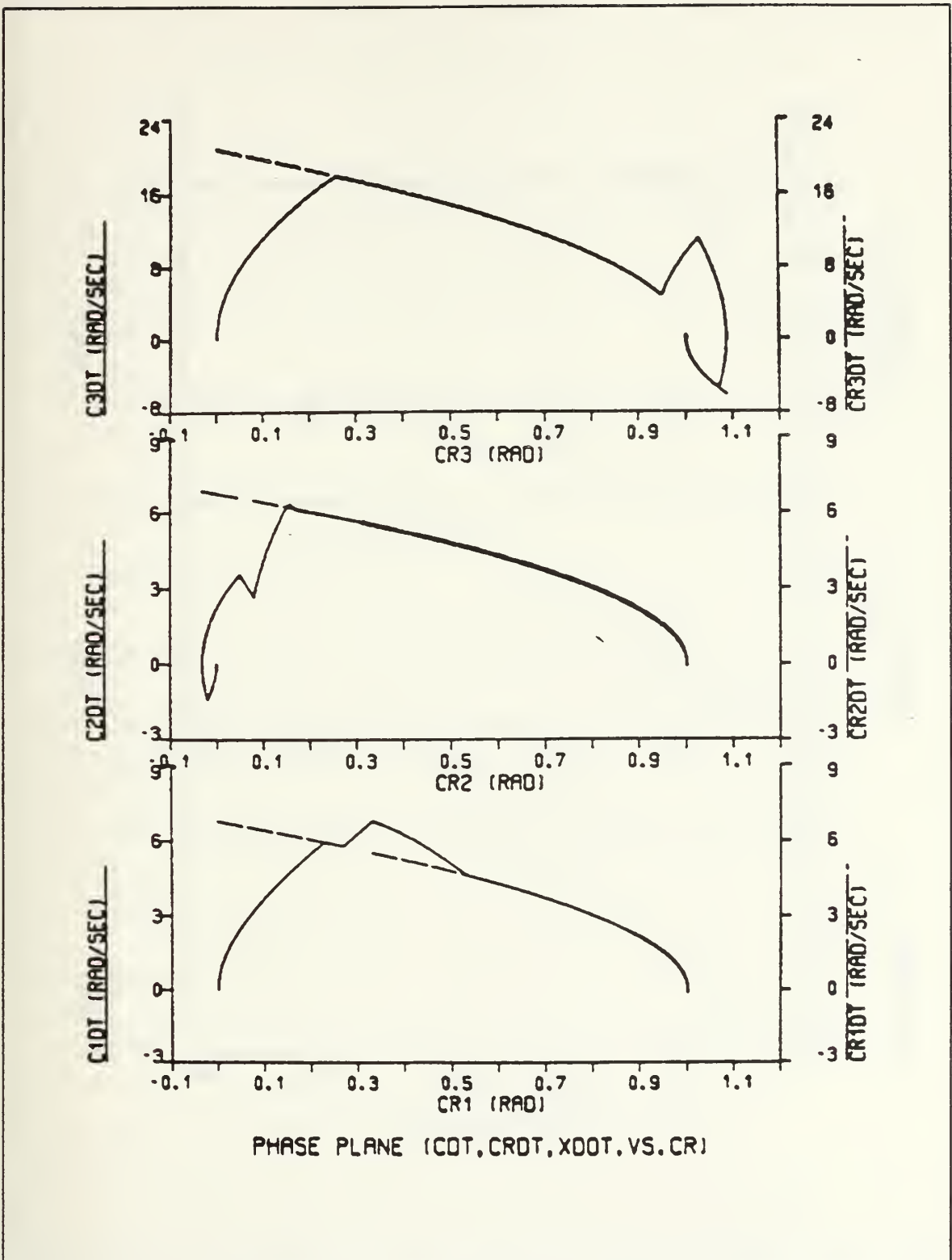
Figure 9.59   Phase Plane Trajectory with Disturbance
Applied T=90 msec (With Gravity)
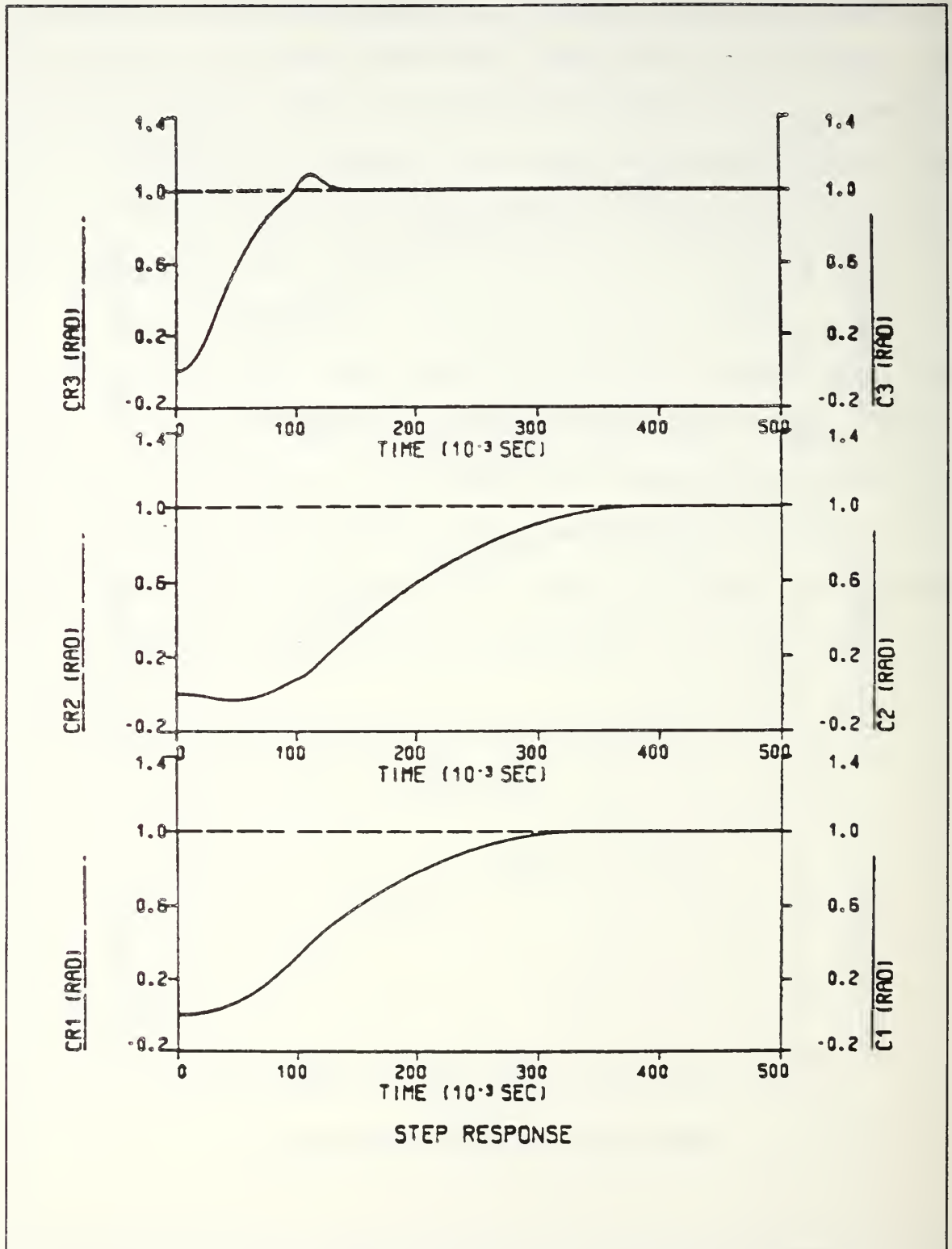
Figures 9.60   Step Response with Disturbance
Applied at T=90 msec (With Gravity)
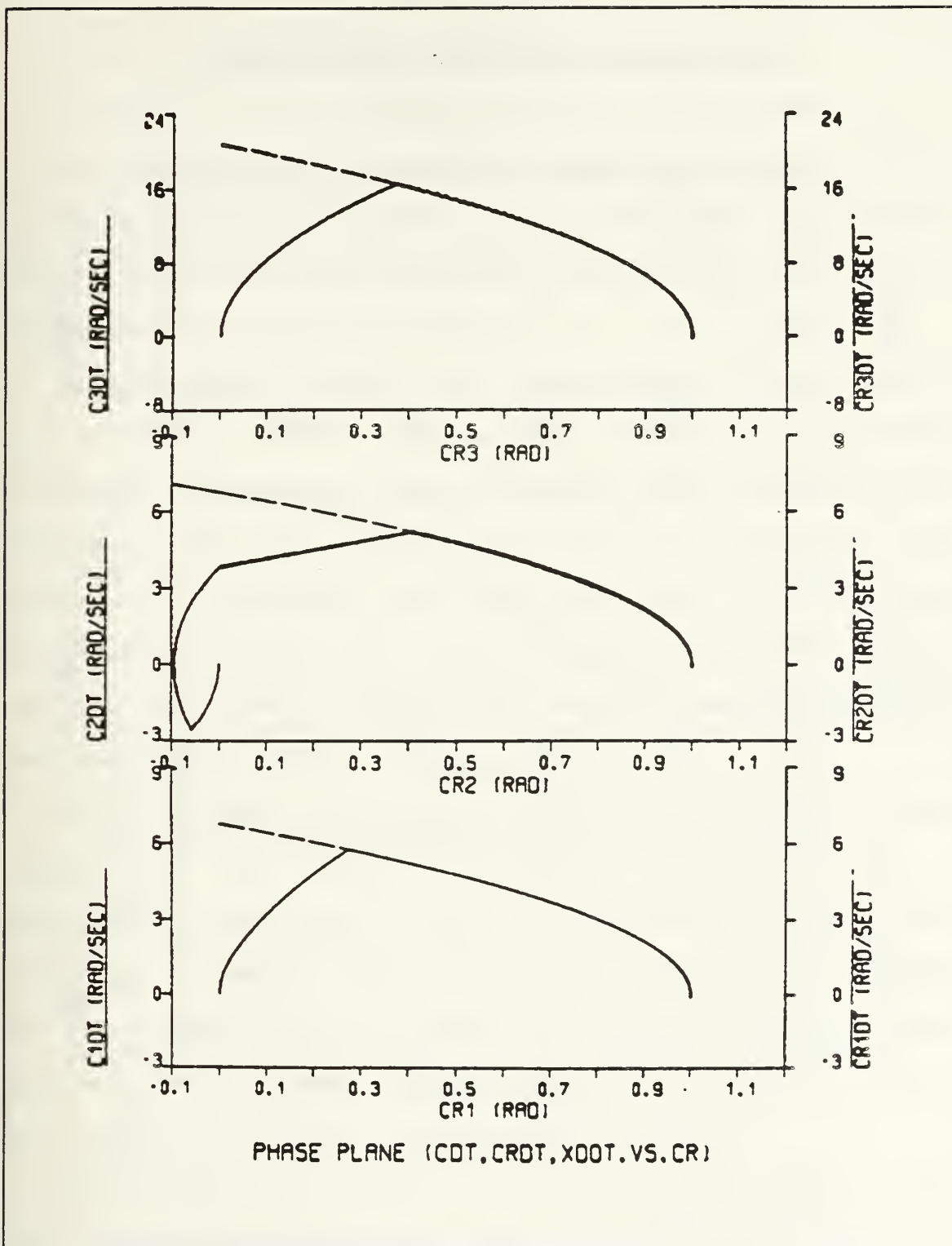
Figures 9.61   Phase Plane Trajectory for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.155 - With Gravity)
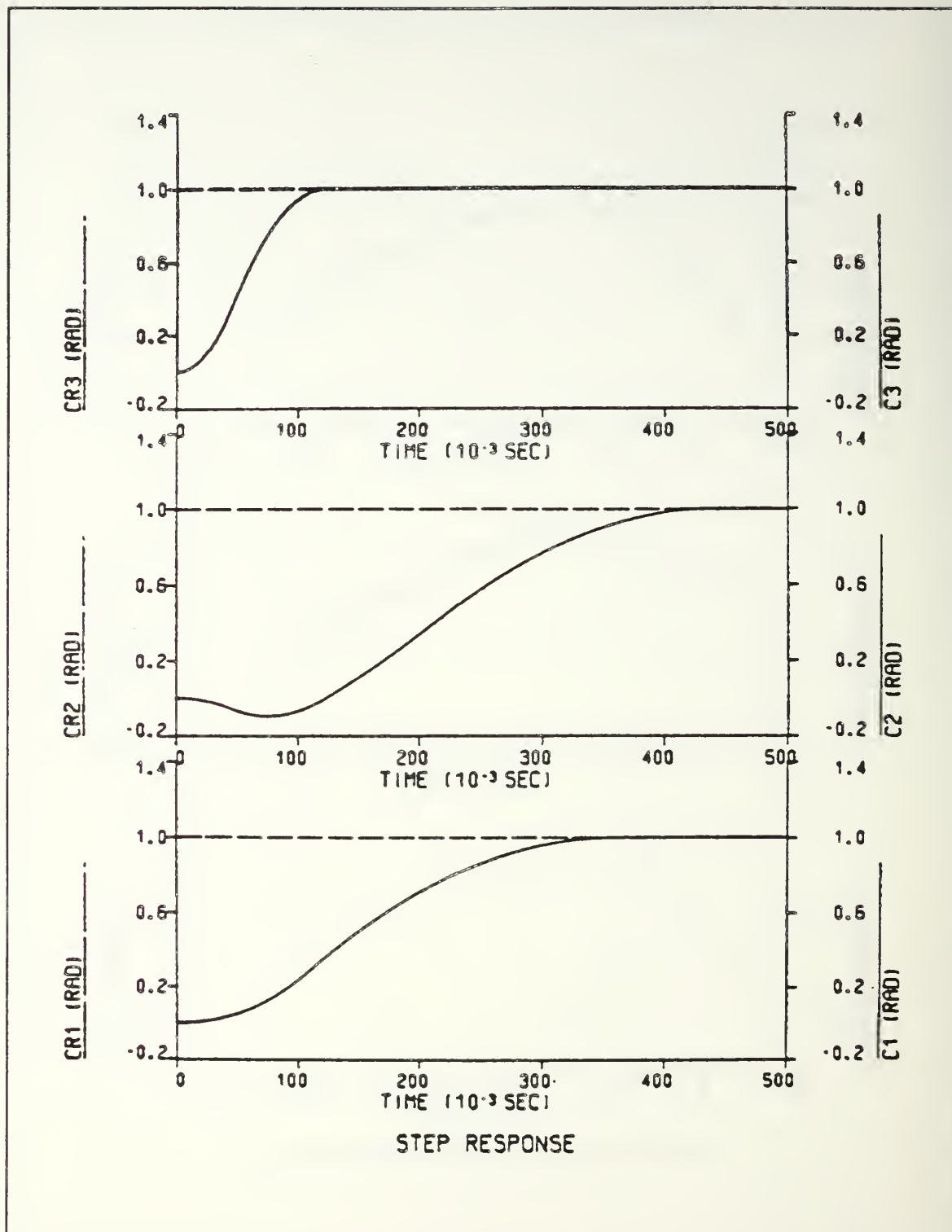
Figure 9.62   Step Response for Loaded Arm with
K_t, K_v, R Decreased and L Increased by 10%
(Load=0.155 – With Gravity)

X. CONCLUSIONS / AREAS FOR FURTHER STUDY

As a result of the research conducted in this thesis, the control of both the three link rectangular and revolute robots, using the curve following technique with an adaptive simulation model appears feasible.

In this study, direct-drive servo motors were used. The use of these motors has several important advantages including no backlash, small friction and high mechanical stiffness, but the complicated dynamics resulting from inertia, interactions and nonlinearites is also more prominent than that of a robot with gears because in this case the disturbance torques are reduced proportionally to the gear ratio.

Both configurations were tested for different load conditions, for random disturbance rejection and for robustness in the case of servo motor parameter variations. Initially, a gravity-free environment was assumed and then the same tests were repeated with gravitational torques included. In all cases actuator dynamics, coupling inertia, centripetal and coriolis forces were included.

From simulation results it was observed that as the added load was increased the robot arms started overshooting which may cause problems in some applications. The capability of the arms to carry larger loads may be improved

if the curve scaling constant, K_1, is designed to be adaptive. Further thesis research in this case is required. The randomly applied disturbances were rejected in all cases and the arms were driven to the commanded position. The slight variation of the servo motor parameters did not show any noteworthy change in the behavior of both robot arms.

A delayed step input command was used to prevent undesired motion of the revolute configuration, caused by interactive torques between links. In order that this technique to be effectively used, an algorithm must be developed which will take into account the initial orientation of the arm and the commanded direction of motion, and will determine the magnitude of the input step command and the optimum delay time.

For this study a rigid arm was assumed for both robot configurations. In rigid arms the speed is greatly limited by the weight of the arm. This weight also increases the size of the required servo motors as well as the power consumption. From this fact, an area for further study is to model an arm which consists of either only flexible links or both rigid and flexible links.

APPENDIX A

DSL PROGRAM FOR THE SIMULATION OF THE SECOND ORDER MODEL

```
TITLE THIS PROGRAM INVESTIGATES THE ABILITY OF THE BASIC
TITLE MODEL TO FOLLOW THE CURVE WHEN VELOCITY CURVE
TITLE FOLLOWING TECHNIQUE IS USED.
PARAM K=1.0,K1=0.6,K2=10000.0,KM=59.29,VSAT=150.0,RO=0.5
PARAM R=1.0
INITIAL
        A=SQRT(2.*KM*VSAT)
        CF=1./RO
        REF=R*CF
        XDOT=0.0
DYNAMIC
        RTH=REF*STEP(0.0)
        E=RTH-C
        IF(E.LT.0.0)XDOT=-A*K1*SQRT(ABS(E))
        IF(E.GE.0.0)XDOT=A*K1*SQRT(E)
DERIVATIVE
        KCDOT=K*CDOT
        XDOTE=XDOT-KCDOT
        V=LIMIT(-VSAT,VSAT,K2*XDOTE)
        CDDOT=V*KM
        CDOT=INTGRL(0.0,CDDOT)
        C=INTGRL(0.0,CDOT)
        CX=C*1./CF
        XDOTX=XDOT*1./CF
        CDOTX=CDOT*1./CF
TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.1,DELT=0.00005
SAVE   (S1)0.001,CX,XDOTX,CDOTX
SAVE   (S2)0.001,CX,R
GRAPH(G1/S1,DE=TEK618,PO=1,.5) CX(LE=6.5,UN='INCHES') ...
        XDOTX(LE=8,NI=8,LO=0.,UN='INCHES/SEC',SC=$AR) ...
        CDOTX(LE=8,NI=8,LO=0.,UN='INCHES/SEC',PO=6.5,SC=$AR)
GRAPH(G2/S2,DE=TEK618,PO=1,1) TIME(LE=6,UN='SEC'),...
        CX(LE=8,NI=6,LO=0,UN='INCHES',SC=0.25),...
        R(LE=8,NI=6,LO=0,SC=0.25,AX=OMIT)
LABEL (G1) PHASE PLANE
LABEL (G1) XDOTX,CDOTX VS CX
LABEL (G2) STEP RESPONSE
LABEL (G2) CX VS TIME
END
STOP
```

207

APPENDIX B

DERIVATION OF MATHEMATICAL MODEL FOR THE THREE LINK RECTANGULAR ROBOT

According to Newton's equation of motion, the force F_i applied to the link i, by the joint motor i, in the direction of the motion of the link i, is

$$F_i = M_i(\dot{v}_i - g) \qquad (B.1)$$

The total torque about the axis of rotation of the motor i is given by the equation

$$T_i = F_i r + J_{mi}\ddot{\theta}_i \qquad (B.2)$$

where r is the radius of the pinion.

Therefore for the three links of the rectangular robot, the equations of motion are

$$F_1 = M_1\ddot{x} \qquad (B.3)$$
$$F_2 = M_2\ddot{y} \qquad (B.4)$$
$$F_3 = M_3(\ddot{z} - g) \qquad (B.5)$$

and the corresponding total torques are

$$T_1 = M_1\ddot{x}r + J_{m1}\ddot{\theta}_1 \qquad (B.6)$$
$$T_2 = M_2\ddot{y}r + J_{m2}\ddot{\theta}_2 \qquad (B.7)$$
$$T_3 = M_3(\ddot{z} - g)r + J_{m3}\ddot{\theta}_3 \qquad (B.8)$$

But the linear motion of the link i is related to the angular motion of the motor i with

$$s_i = \theta_i r \qquad\qquad\qquad\qquad\qquad (B.9)$$

Substituting equation B.9 into equations B.6, B.7, and B.8

$$T_1 = (M_1 r^2 + J_{m1})\ddot{\theta}_1 \qquad\qquad\qquad (B.10)$$

$$T_2 = (M_2 r^2 + J_{m2})\ddot{\theta}_2 \qquad\qquad\qquad (B.11)$$

$$T_3 = (M_3 r^2 + J_{m3})\ddot{\theta}_3 - M_3 g r \qquad\qquad (B.12)$$

APPENDIX C

DSL PROGRAM FOR THE THREE LINK RECTANGULAR ROBOT
UNDER DIFFERENT LOAD CONDITIONS (NO GRAVITY)

```
TITLE SIMULATION PROGRAM FOR THE RECTANGULAR COORDINATE
TITLE CONFIGURATION UNDER DIFFERENT LOAD CONDITIONS
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=59.29,KM2=90.25,KM3=77.44
PARAM VSAT=150.0,M1=0.082,M2=0.041,M3=0.041,MM=0.186
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,RO=0.5,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM RX=1.00 ,RY=1.00 ,RZ=1.00
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* RX,RY,RZ:THE COMMANDED TIP POSITION IN INCHES
* RO: THE RADIUS OF THE PINION GEARING
* T:  THE SAMPLING INTERVAL
INITIAL
        SW1=0
        SW2=0
        SW3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        TL1=0.
        TL2=0.
        TL3=0.
        CF=1./RO
        A1=SQRT(2.*KM1*VSAT)
        A2=SQRT(2.*KM2*VSAT)
        A3=SQRT(2.*KM3*VSAT)
        REF1=RX*CF
        REF2=RY*CF
        REF3=RZ*CF
DERIVATIVE
        TM1=M1+M2+M3+2*MM+LOAD
        TM2=M2+LOAD
        TM3=M2+M3+MM+LOAD
        JM1=TM1*RO
```

210

```
          JM2=TM2*RO
          JM3=TM3*RO
          JTOT1=J1+JM1
          JTOT2=J2+JM2
          JTOT3=J3+JM3
          RTH1=REF1*STEP(0.0)
          RTH2=REF2*STEP(0.0)
          RTH3=REF3*STEP(0.0)
          E1=RTH1-C1
          E2=RTH2-C2
          E3=RTH3-C3
          ERX=RX-CRX
          ERY=RY-CRY
          ERZ=RZ-CRZ
NOSORT
          IF(E1.LT.0.0)  X1DOT=-A1*K1*SQRT(ABS(E1))
          IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
          IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
          IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
          IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
          IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

**********************************************************

          KC1DOT=K*C1DOT
          X1DOTE=X1DOT-KC1DOT
          V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
          C1DDT=V1*KM1
          C1DOT=INTGRL(0.0,C1DDT)
          C1=INTGRL(0.0,C1DOT)
          CX=C1*1./CF
          VM1=V1-KV1*CR1DOT
          MP1=REALPL(0.0,L1/R1,VM1/R1)
          MT1=KT1*MP1
          MT1E=MT1-BM1*CR1DOT-TL1
          CR1DDT=(1./JTOT1)*MT1E
          CR1DOT=INTGRL(0.0,CR1DDT)
          CR1=INTGRL(0.0,CR1DOT)
          CRX=CR1*1./CF
          XXDOT=X1DOT*1./CF
          XXDOTE=X1DOTE*1./CF
          CXDOT=C1DOT*1./CF
          CRXDOT=CR1DOT*1./CF

**********************************************************

          KC2DOT=K*C2DOT
          X2DOTE=X2DOT-KC2DOT
          V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
          C2DDT=V2*KM2
```

211

```
        C2DOT=INTGRL(0.0,C2DDT)
        C2=INTGRL(0.0,C2DOT)
        CY=C2*1./CF
        VM2=V2-KV2*CR2DOT
        MP2=REALPL(0.0,L2/R2,VM2/R2)
        MT2=KT2*MP2
        MT2E=MT2-BM2*CR2DOT-TL2
        CR2DDT=(1./JTOT2)*MT2E
        CR2DOT=INTGRL(0.0,CR2DDT)
        CR2=INTGRL(0.0,CR2DOT)
        CRY=CR2*1./CF
        XYDOT=X2DOT*1./CF
        XYDOTE=X2DOTE*1./CF
        CYDOT=C2DOT*1./CF
        CRYDOT=CR2DOT*1./CF


*************************************************************

        KC3DOT=K*C3DOT
        X3DOTE=X3DOT-KC3DOT
        V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
        C3DDT=V3*KM3
        C3DOT=INTGRL(0.0,C3DDT)
        C3=INTGRL(0.0,C3DOT)
        CZ=C3*1./CF
        VM3=V3-KV3*CR3DOT
        MP3=REALPL(0.0,L3/R3,VM3/R3)
        MT3=KT3*MP3
        MT3E=MT3-BM3*CR3DOT-TL3
        CR3DDT=(1./JTOT3)*MT3E
        CR3DOT=INTGRL(0.0,CR3DDT)
        CR3=INTGRL(0.0,CR3DOT)
        CRZ=CR3*1./CF
        XZDOT=X3DOT*1./CF
        XZDOTE=X3DOTE*1./CF
        CZDOT=C3DOT*1./CF
        CRZDOT=CR3DOT*1./CF


*************************************************************

SAMPLE
NOSORT
        IF (N1.EQ.0.0) GO TO 10
        IF (C1DOT.GT.X1DOT) SW1=1
        IF (SW1.EQ.1) GO TO 20
        KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
        KM1=KS1
20      CONTINUE
        C1=CR1
        C1DOT=CR1DOT
10      N1=N1+1
```

212

```
        CONTINUE
        IF (N2.EQ.0.0) GO TO 30
        IF (C2DOT.GT.X2DOT) SW2=1
        IF (SW2.EQ.1) GO TO 40
        KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
        KM2=KS2
40      CONTINUE
        C2=CR2
        C2DOT=CR2DOT
30      N2=N2+1
        CONTINUE
        IF (N3.EQ.0.0) GO TO 50
        IF (C3DOT.GT.X3DOT) SW3=1
        IF (SW3.EQ.1) GO TO 60
        KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
        KM3=KS3
60      CONTINUE
        C3=CR3
        C3DOT=CR3DOT
50      N3=N3+1
        CONTINUE
SORT


************************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.06,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,XXDOT,XYDOT,XZDOT,CXDOT,CYDOT,CZDOT,...
      CRXDOT,CRYDOT,CRZDOT,CRX,CRY,CRZ
SAVE (S2) 0.001,CX,CRX,RX,CY,CRY,RY,CZ,CRZ,RZ
SAVE (S3) 0.001,ERX,ERY,ERZ
GRAPH (G1/S1,DE=TEK618,PO=1) CRX(LE=6.0,UN='INCHES'),...
      CXDOT(NI=3,LO=0,SC=20,UN='INCHES/SEC'),...
      CRXDOT(NI=3,LO=0,PO=6.0,SC=20,UN='INCHES/SEC'),...
      XXDOT(NI=3,LO=0,SC=20,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
      CRY(LE=6.0,UN='INCHES'),...
      CYDOT(NI=3,LO=0,SC=25,UN='INCHES/SEC'),...
      CRYDOT(NI=3,LO=0,PO=6.0,SC=25,UN='INCHES/SEC'),...
      XYDOT(NI=3,LO=0,SC=25,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
      CRZ(LE=6.0,UN='INCHES'),...
      CZDOT(NI=3,LO=0,SC=20,UN='INCHES/SEC'),...
      CRZDOT(NI=3,LO=0,PO=6.0,SC=20,UN='INCHES/SEC'),...
      XZDOT(NI=3,LO=0,SC=20,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=6,UN='SEC'),...
      CX(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
      CRX(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
      RX(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
```

213

```
        TIME(LE=6.0,NI=6,UN='SEC'),...
        CY(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
        CRY(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
        RY(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
        TIME(LE=6.0,NI=6,UN='SEC'),...
        CZ(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
        CRZ(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
        RZ(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G7/S3,DE=TEK618,PO=1,.5)...
        TIME(LE=5.0,NI=6,UN='SEC'),...
        ERX(LE=8,NI=10,LO=0.0,UN='INCHES',SC=.1),...
        ERY(LE=8,NI=10,LO=0.0,UN='INCHES',SC=.1,PO=5.0),...
        ERZ(LE=8,NI=10,LO=0.0,UN='INCHES',SC=.1,PO=6.2)
LABEL (G1) PHASE PLANE (CDOT,CRDOT,XDOT.VS.CR)
LABEL (G4) STEP RESPONSE
LABEL (G7) ERROR .VS. TIME
END
STOP


************************************************************
```

214

APPENDIX D

DSL PROGRAM FOR THE THREE LINK RECTANGULAR ROBOT
WITH DISTURBANCE (NO GRAVITY)

```
TITLE SIMULATION PROGRAM FOR THE RECTANGULAR COORDINATE
TITLE CONFIGURATION WITH DISTURBANCE (NO GRAVITY)
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=59.29,KM2=90.25,KM3=77.44
PARAM VSAT=150.0,M1=0.082,M2=0.041,M3=0.041,MM=0.186
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,RO=0.5,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM RX=1.00 ,RY=1.00 ,RZ=1.00
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* RX,RY,RZ:THE COMMANDED TIP POSITION IN INCHES
* RO: THE RADIUS OF THE PINION GEARING
* T:  THE SAMPLING INTERVAL
INITIAL
        SW1=0
        SW2=0
        SW3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        TL1=0.
        TL2=0.
        TL3=0.
        CF=1./RO
        A1=SQRT(2.*KM1*VSAT)
        A2=SQRT(2.*KM2*VSAT)
        A3=SQRT(2.*KM3*VSAT)
        REF1=RX*CF
        REF2=RY*CF
        REF3=RZ*CF
DERIVATIVE
        TM1=M1+M2+M3+2*MM+LOAD
        TM2=M2+LOAD
        TM3=M2+M3+MM+LOAD
        JM1=TM1*RO
```

```
            JM2=TM2*RO
            JM3=TM3*RO
            JTOT1=J1+JM1
            JTOT2=J2+JM2
            JTOT3=J3+JM3
            RTH1=REF1*STEP(0.0)+10*(STEP(0.010)-STEP(0.012))
            RTH2=REF2*STEP(0.0)+10*(STEP(0.010)-STEP(0.012))
            RTH3=REF3*STEP(0.0)+10*(STEP(0.010)-STEP(0.012))
            E1=RTH1-C1
            E2=RTH2-C2
            E3=RTH3-C3
NOSORT
            IF(E1.LT.0.0)  X1DOT=-A1*K1*SQRT(ABS(E1))
            IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
            IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
            IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
            IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
            IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

*************************************************************

            KC1DOT=K*C1DOT
            X1DOTE=X1DOT-KC1DOT
            V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
            C1DDT=V1*KM1
            C1DOT=INTGRL(0.0,C1DDT)
            C1=INTGRL(0.0,C1DOT)
            CX=C1*1./CF
            VM1=V1-KV1*CR1DOT
            MP1=REALPL(0.0,L1/R1,VM1/R1)
            MT1=KT1*MP1
            MT1E=MT1-BM1*CR1DOT-TL1
            CR1DDT=(1./JTOT1)*MT1E
            CR1DOT=INTGRL(0.0,CR1DDT)
            CR1=INTGRL(0.0,CR1DOT)
            CRX=CR1*1./CF
            XXDOT=X1DOT*1./CF
            CXDOT=C1DOT*1./CF
            CRXDOT=CR1DOT*1./CF

*************************************************************

            KC2DOT=K*C2DOT
            X2DOTE=X2DOT-KC2DOT
            V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
            C2DDT=V2*KM2
            C2DOT=INTGRL(0.0,C2DDT)
            C2=INTGRL(0.0,C2DOT)
            CY=C2*1./CF
            VM2=V2-KV2*CR2DOT
```

216

```
          MP2=REALPL(0.0,L2/R2,VM2/R2)
          MT2=KT2*MP2
          MT2E=MT2-BM2*CR2DOT-TL2
          CR2DDT=(1./JTOT2)*MT2E
          CR2DOT=INTGRL(0.0,CR2DDT)
          CR2=INTGRL(0.0,CR2DOT)
          CRY=CR2*1./CF
          XYDOT=X2DOT*1./CF
          CYDOT=C2DOT*1./CF
          CRYDOT=CR2DOT*1./CF


**********************************************************

          KC3DOT=K*C3DOT
          X3DOTE=X3DOT-KC3DOT
          V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
          C3DDT=V3*KM3
          C3DOT=INTGRL(0.0,C3DDT)
          C3=INTGRL(0.0,C3DOT)
          CZ=C3*1./CF
          VM3=V3-KV3*CR3DOT
          MP3=REALPL(0.0,L3/R3,VM3/R3)
          MT3=KT3*MP3
          MT3E=MT3-BM3*CR3DOT-TL3
          CR3DDT=(1./JTOT3)*MT3E
          CR3DOT=INTGRL(0.0,CR3DDT)
          CR3=INTGRL(0.0,CR3DOT)
          CRZ=CR3*1./CF
          XZDOT=X3DOT*1./CF
          CZDOT=C3DOT*1./CF
          CRZDOT=CR3DOT*1./CF


**********************************************************

SAMPLE
NOSORT
          IF (N1.EQ.0.0) GO TO 10
          IF (C1DOT.GT.X1DOT) SW1=1
          IF (SW1.EQ.1) GO TO 20
          KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
          KM1=KS1
20        CONTINUE
          C1=CR1
          C1DOT=CR1DOT
10        N1=N1+1
          CONTINUE
          IF (N2.EQ.0.0) GO TO 30
          IF (C2DOT.GT.X2DOT) SW2=1
          IF (SW2.EQ.1) GO TO 40
          KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
          KM2=KS2
```

```
40        CONTINUE
          C2=CR2
          C2DOT=CR2DOT
30        N2=N2+1
          CONTINUE
          IF (N3.EQ.0.0) GO TO 50
          IF (C3DOT.GT.X3DOT) SW3=1
          IF (SW3.EQ.1) GO TO 60
          KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
          KM3=KS3
60        CONTINUE
          C3=CR3
          C3DOT=CR3DOT
50        N3=N3+1
          CONTINUE
SORT


***********************************************************


TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.08,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0001,XXDOT,XYDOT,XZDOT,CXDOT,CYDOT,CZDOT,...
     CRXDOT,CRYDOT,CRZDOT,CRX,CRY,CRZ
SAVE (S2) 0.001,CX,CRX,RX,CY,CRY,RY,CZ,CRZ,RZ
GRAPH (G1/S1,DE=TEK618,PO=1)...
     CRX(LE=6.0,NI=10,UN='INCHES'),...
     CXDOT(NI=3,LO=0,SC=30,UN='INCHES/SEC'),...
     CRXDOT(NI=3,LO=0,PO=6.0,SC=30,UN='INCHES/SEC'),...
     XXDOT(NI=3,LO=0,SC=30,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
     CRY(LE=6.0,NI=10,UN='INCHES'),...
     CYDOT(NI=3,LO=0,SC=30,UN='INCHES/SEC'),...
     CRYDOT(NI=3,LO=0,PO=6.0,SC=30,UN='INCHES/SEC'),...
     XYDOT(NI=3,LO=0,SC=30,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
     CRZ(LE=6.0,NI=10,UN='INCHES'),...
     CZDOT(NI=3,LO=0,SC=30,UN='INCHES/SEC'),...
     CRZDOT(NI=3,LO=0,PO=6.0,SC=30,UN='INCHES/SEC'),...
     XZDOT(NI=3,LO=0,SC=30,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=8,UN='SEC'),...
     CX(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
     CRX(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
     RX(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
     TIME(LE=6.0,NI=8,UN='SEC'),...
     CY(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
     CRY(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
     RY(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
     TIME(LE=6.0,NI=8,UN='SEC'),...
```

```
        CZ(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
        CRZ(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
        RZ(NI=3,LO=0.0,SC=0.5,AX=OMIT)
LABEL (G1) PHASE PLANE (CDOT,CRDOT,XDOT.VS.CR)
LABEL (G4) STEP RESPONSE
END
STOP
```

****

DSL PROGRAM FOR THE THRE LINK RECTANGULAR ROBOT
UNDER DIFFERENT LOAD CONDITIONS
(GRAVITATIONAL TORQUES INCLUDED)

```
TITLE SIMULATION PROGRAM FOR THE RECTANGULAR COORDINATE
TITLE CONFIGURATION UNDER DIFFERENT LOAD CONDITIONS
TITLE (WITH GRAVITY)
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=59.29,KM2=90.25,KM3=77.44
PARAM VSAT=150.0,M1=0.082,M2=0.041,M3=0.041,MM=0.186
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,RO=0.5,LOAD=0.86
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025,G=386.4
PARAM RX=1.00 ,RY=1.00 ,RZ=1.00
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALLING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* RX,RY,RZ: THE COMMANDED TIP POSITION IN INCHES
* RO: THE RADIUS OF THE PINION GEARING
* T: THE SAMPLING INTERVAL
INITIAL
        SW1=0
        SW2=0
        SW3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        TL1=0.
        TL2=0.
        TL3=(M2+M3+MM)*G*RO
        CF=1./RO
        A1=SQRT(2.*KM1*VSAT)
        A2=SQRT(2.*KM2*VSAT)
        A3=SQRT(2.*KM3*VSAT)
        REF1=RX*CF
        REF2=RY*CF
        REF3=RZ*CF
DERIVATIVE
        TM1=M1+M2+M3+2*MM+LOAD
        TM2=M2+LOAD
```

```
          TM3=M2+M3+MM+LOAD
          JM1=TM1*RO
          JM2=TM2*RO
          JM3=TM3*RO
          JTOT1=J1+JM1
          JTOT2=J2+JM2
          JTOT3=J3+JM3
          RTH1=REF1*STEP(0.0)
          RTH2=REF2*STEP(0.0)
          RTH3=REF3*STEP(0.0)
          E1=RTH1-C1
          E2=RTH2-C2
          E3=RTH3-C3
          ERX=RX-CRX
          ERY=RY-CRY
          ERZ=RZ-CRZ
NOSORT
          IF(E1.LT.0.0)  X1DOT=-A1*K1*SQRT(ABS(E1))
          IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
          IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
          IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
          IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
          IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

**********************************************************

          KC1DOT=K*C1DOT
          X1DOTE=X1DOT-KC1DOT
          V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
          C1DDT=V1*KM1
          C1DOT=INTGRL(0.0,C1DDT)
          C1=INTGRL(0.0,C1DOT)
          CX=C1*1./CF
          VM1=V1-KV1*CR1DOT
          MP1=REALPL(0.0,L1/R1,VM1/R1)
          MT1=KT1*MP1
          MT1E=MT1-BM1*CR1DOT-TL1
          CR1DDT=(1./JTOT1)*MT1E
          CR1DOT=INTGRL(0.0,CR1DDT)
          CR1=INTGRL(0.0,CR1DOT)
          CRX=CR1*1./CF
          XXDOT=X1DOT*1./CF
          XXDOTE=X1DOTE*1./CF
          CXDOT=C1DOT*1./CF
          CRXDOT=CR1DOT*1./CF

**********************************************************

          KC2DOT=K*C2DOT
          X2DOTE=X2DOT-KC2DOT
```

221

```
        V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
        C2DDT=V2*KM2
        C2DOT=INTGRL(0.0,C2DDT)
        C2=INTGRL(0.0,C2DOT)
        CY=C2*1./CF
        VM2=V2-KV2*CR2DOT
        MP2=REALPL(0.0,L2/R2,VM2/R2)
        MT2=KT2*MP2
        MT2E=MT2-BM2*CR2DOT-TL2
        CR2DDT=(1./JTOT2)*MT2E
        CR2DOT=INTGRL(0.0,CR2DDT)
        CR2=INTGRL(0.0,CR2DOT)
        CRY=CR2*1./CF
        XYDOT=X2DOT*1./CF
        XYDOTE=X2DOTE*1./CF
        CYDOT=C2DOT*1./CF
        CRYDOT=CR2DOT*1./CF


*********************************************************

        KC3DOT=K*C3DOT
        X3DOTE=X3DOT-KC3DOT
        V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
        C3DDT=V3*KM3
        C3DOT=INTGRL(0.0,C3DDT)
        C3=INTGRL(0.0,C3DOT)
        CZ=C3*1./CF
        VM3=V3-KV3*CR3DOT
        MP3=REALPL(0.0,L3/R3,VM3/R3)
        MT3=KT3*MP3
        MT3E=MT3-BM3*CR3DOT-TL3
        CR3DDT=(1./JTOT3)*MT3E
        CR3DOT=INTGRL(0.0,CR3DDT)
        CR3=INTGRL(0.0,CR3DOT)
        CRZ=CR3*1./CF
        XZDOT=X3DOT*1./CF
        XZDOTE=X3DOTE*1./CF
        CZDOT=C3DOT*1./CF
        CRZDOT=CR3DOT*1./CF


*********************************************************

SAMPLE
NOSORT
        IF (N1.EQ.0.0) GO TO 10
        IF (C1DOT.GT.X1DOT) SW1=1
        IF (SW1.EQ.1) GO TO 20
        KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
        KM1=KS1
20      CONTINUE
        C1=CR1
```

222

```
          C1DOT=CR1DOT
10        N1=N1+1
          CONTINUE
          IF (N2.EQ.0.0) GO TO 30
          IF (C2DOT.GT.X2DOT) SW2=1
          IF (SW2.EQ.1) GO TO 40
          KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
          KM2=KS2
40        CONTINUE
          C2=CR2
          C2DOT=CR2DOT
30        N2=N2+1
          CONTINUE
          IF (N3.EQ.0.0) GO TO 50
          IF (C3DOT.GT.X3DOT) SW3=1
          IF (SW3.EQ.1) GO TO 60
          KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
          KM3=KS3
60        CONTINUE
          C3=CR3
          C3DOT=CR3DOT
50        N3=N3+1
          CONTINUE
SORT

*************************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.1,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,XXDOT,XYDOT,XZDOT,CXDOT,CYDOT,CZDOT,...
     CRXDOT,CRYDOT,CRZDOT,CRX,CRY,CRZ
SAVE (S2) 0.001,CX,CRX,RX,CY,CRY,RY,CZ,CRZ,RZ
SAVE (S3) 0.001,ERX,ERY,ERZ
GRAPH (G1/S1,DE=TEK618,PO=1)...
     CRX(LE=6.0,NI=6,UN='INCHES'),...
     CXDOT(NI=3,LO=-30,SC=30,UN='INCHES/SEC'),...
     CRXDOT(NI=3,LO=-30,PO=6.0,SC=30,UN='INCHES/SEC'),...
     XXDOT(NI=3,LO=-30,SC=30,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
     CRY(LE=6.0,NI=6,UN='INCHES'),...
     CYDOT(NI=3,LO=-35,SC=35,UN='INCHES/SEC'),...
     CRYDOT(NI=3,LO=-35,PO=6.0,SC=35,UN='INCHES/SEC'),...
     XYDOT(NI=3,LO=-35,SC=35,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
     CRZ(LE=6.0,NI=6,UN='INCHES'),...
     CZDOT(NI=3,LO=-30,SC=30,UN='INCHES/SEC'),...
     CRZDOT(NI=3,LO=-30,PO=6.0,SC=30,UN='INCHES/SEC'),...
     XZDOT(NI=3,LO=-30,SC=30,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=5,UN='SEC'),...
     CX(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
```

```
       CRX(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
       RX(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH  (G5/S2,DE=TEK618,OV,PO=1,3.25)...
       TIME(LE=6.0,NI=5,UN='SEC'),...
       CY(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
       CRY(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
       RY(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH  (G6/S2,DE=TEK618,OV,PO=1,6.5)...
       TIME(LE=6.0,NI=5,UN='SEC'),...
       CZ(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
       CRZ(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
       RZ(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH  (G7/S3,DE=TEK618,PO=1,.5)...
       TIME(LE=5.0,NI=5,UN='SEC'),...
       ERX(LE=8,NI=6,LO=-.2,UN='INCHES',SC=.2),...
       ERY(LE=8,NI=6,LO=-.2,UN='INCHES',SC=.2,PO=5.0),...
       ERZ(LE=8,NI=6,LO=-.2,UN='INCHES',SC=.2,PO=6.2)
LABEL  (G1) PHASE PLANE (CDOT,CRDOT,XDOT.VS.CR)
LABEL  (G4) STEP RESPONSE
LABEL  (G7) ERROR .VS. TIME
END
STOP


*************************************************************
```

APPENDIX F

DSL PROGRAM FOR THE THREE LINK RECTANGULAR ROBOT WITH DISTURBANCE (GRAVITATIONAL TORQUES INCLUDED)

```
TITLE SIMULATION PROGRAM FOR THE RECTANGULAR COORDINATE
TITLE CONFIGURATION WITH DISTURBANCE (WITH GRAVITY)
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=59.29,KM2=90.25,KM3=77.44
PARAM VSAT=150.0,M1=0.082,M2=0.041,M3=0.041,MM=0.186
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,RO=0.5,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025,G=386.4
PARAM RX=1.00 ,RY=1.00 ,RZ=1.00
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALLING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* RX,RY,RZ:THE COMMANDED TIP POSITION IN INCHES
* RO: THE RADIUS OF THE PINION GEARING
* T:  THE SAMPLING INTERVAL
INITIAL
        SW1=0
        SW2=0
        SW3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        TL1=0.
        TL2=0.
        TL3=(M2+M3+MM)*G*RO
        CF=1./RO
        A1=SQRT(2.*KM1*VSAT)
        A2=SQRT(2.*KM2*VSAT)
        A3=SQRT(2.*KM3*VSAT)
        REF1=RX*CF
        REF2=RY*CF
        REF3=RZ*CF
DERIVATIVE
        TM1=M1+M2+M3+2*MM+LOAD
        TM2=M2+LOAD
        TM3=M2+M3+MM+LOAD
        JM1=TM1*RO
```

```
          JM2=TM2*RO
          JM3=TM3*RO
          JTOT1=J1+JM1
          JTOT2=J2+JM2
          JTOT3=J3+JM3
          RTH1=REF1*STEP(0.0)+10*(STEP(0.050)-STEP(0.052))
          RTH2=REF2*STEP(0.0)+10*(STEP(0.050)-STEP(0.052))
          RTH3=REF3*STEP(0.0)+10*(STEP(0.050)-STEP(0.052))
          E1=RTH1-C1
          E2=RTH2-C2
          E3=RTH3-C3
NOSORT
          IF(E1.LT.0.0)  X1DOT=-A1*K1*SQRT(ABS(E1))
          IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
          IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
          IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
          IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
          IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

************************************************************

          KC1DOT=K*C1DOT
          X1DOTE=X1DOT-KC1DOT
          V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
          C1DDT=V1*KM1
          C1DOT=INTGRL(0.0,C1DDT)
          C1=INTGRL(0.0,C1DOT)
          CX=C1*1./CF
          VM1=V1-KV1*CR1DOT
          MP1=REALPL(0.0,L1/R1,VM1/R1)
          MT1=KT1*MP1
          MT1E=MT1-BM1*CR1DOT-TL1
          CR1DDT=(1./JTOT1)*MT1E
          CR1DOT=INTGRL(0.0,CR1DDT)
          CR1=INTGRL(0.0,CR1DOT)
          CRX=CR1*1./CF
          XXDOT=X1DOT*1./CF
          CXDOT=C1DOT*1./CF
          CRXDOT=CR1DOT*1./CF

************************************************************

          KC2DOT=K*C2DOT
          X2DOTE=X2DOT-KC2DOT
          V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
          C2DDT=V2*KM2
          C2DOT=INTGRL(0.0,C2DDT)
          C2=INTGRL(0.0,C2DOT)
          CY=C2*1./CF
          VM2=V2-KV2*CR2DOT
```

226

```
        MP2=REALPL(0.0,L2/R2,VM2/R2)
        MT2=KT2*MP2
        MT2E=MT2-BM2*CR2DOT-TL2
        CR2DDT=(1./JTOT2)*MT2E
        CR2DOT=INTGRL(0.0,CR2DDT)
        CR2=INTGRL(0.0,CR2DOT)
        CRY=CR2*1./CF
        XYDOT=X2DOT*1./CF
        CYDOT=C2DOT*1./CF
        CRYDOT=CR2DOT*1./CF


**********************************************************

        KC3DOT=K*C3DOT
        X3DOTE=X3DOT-KC3DOT
        V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
        C3DDT=V3*KM3
        C3DOT=INTGRL(0.0,C3DDT)
        C3=INTGRL(0.0,C3DOT)
        CZ=C3*1./CF
        VM3=V3-KV3*CR3DOT
        MP3=REALPL(0.0,L3/R3,VM3/R3)
        MT3=KT3*MP3
        MT3E=MT3-BM3*CR3DOT-TL3
        CR3DDT=(1./JTOT3)*MT3E
        CR3DOT=INTGRL(0.0,CR3DDT)
        CR3=INTGRL(0.0,CR3DOT)
        CRZ=CR3*1./CF
        XZDOT=X3DOT*1./CF
        CZDOT=C3DOT*1./CF
        CRZDOT=CR3DOT*1./CF


**********************************************************

SAMPLE
NOSORT
        IF (N1.EQ.0.0) GO TO 10
        IF (C1DOT.GT.X1DOT) SW1=1
        IF (SW1.EQ.1) GO TO 20
        KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
        KM1=KS1
20      CONTINUE
        C1=CR1
        C1DOT=CR1DOT
10      N1=N1+1
        CONTINUE
        IF (N2.EQ.0.0) GO TO 30
        IF (C2DOT.GT.X2DOT) SW2=1
        IF (SW2.EQ.1) GO TO 40
        KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
        KM2=KS2
```

227

```
40       CONTINUE
         C2=CR2
         C2DOT=CR2DOT
30       N2=N2+1
         CONTINUE
         IF (N3.EQ.0.0) GO TO 50
         IF (C3DOT.GT.X3DOT) SW3=1
         IF (SW3.EQ.1) GO TO 60
         KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
         KM3=KS3
60       CONTINUE
         C3=CR3
         C3DOT=CR3DOT
50       N3=N3+1
         CONTINUE
SORT


****************************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.08,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0001,XXDOT,XYDOT,XZDOT,CXDOT,CYDOT,CZDOT,...
      CRXDOT,CRYDOT,CRZDOT,CRX,CRY,CRZ
SAVE (S2) 0.001,CX,CRX,RX,CY,CRY,RY,CZ,CRZ,RZ
G R A P H      ( G 1 / S 1 , D E = T E K 6 1 8 , P O = 1 )
CRX(LE=6.0,NI=11,UN='INCHES')...
      CXDOT(NI=3,LO=-30,SC=30,UN='INCHES/SEC'),...
      CRXDOT(NI=3,LO=-30,PO=6.0,SC=30,UN='INCHES/SEC'),...
      XXDOT(NI=3,LO=-30,SC=30,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
      CRY(LE=6.0,NI=11,UN='INCHES'),...
      CYDOT(NI=3,LO=-35,SC=35,UN='INCHES/SEC'),...
      CRYDOT(NI=3,LO=-35,PO=6.0,SC=35,UN='INCHES/SEC'),...
      XYDOT(NI=3,LO=-35,SC=35,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
      CRZ(LE=6.0,NI=11,UN='INCHES'),...
      CZDOT(NI=3,LO=-30,SC=30,UN='INCHES/SEC'),...
      CRZDOT(NI=3,LO=-30,PO=6.0,SC=30,UN='INCHES/SEC'),...
      XZDOT(NI=3,LO=-30,SC=30,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=8,UN='SEC'),...
      CX(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
      CRX(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
      RX(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
      TIME(LE=6.0,NI=8,UN='SEC'),...
      CY(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
      CRY(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
      RY(NI=3,LO=0.0,SC=0.5,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
      TIME(LE=6.0,NI=8,UN='SEC'),...
```

```
        CZ(NI=3,LO=0.0,UN='INCHES',SC=0.5,PO=6.0),...
        CRZ(NI=3,LO=0.0,UN='INCHES',SC=0.5),...
        RZ(NI=3,LO=0.0,SC=0.5,AX=OMIT)
LABEL (G1) PHASE PLANE (CDOT,CRDOT,XDOT.VS.CR)
LABEL (G4) STEP RESPONSE
END
STOP
```

****

APPENDIX G

DERIVATION OF MATHEMATICAL MODEL FOR THE THREE LINK
REVOLUTE ROBOT ARM

First the kinetic energy of each link must be computed
as

$$K_i = 1/2 m_i u_i^2 \qquad\qquad (G.1)$$

and then the potential energy, which is related to the
vertical height of the mass expressed by the z coordinate,
must be calculated as

$$V_i = m_i g z_i \qquad\qquad (G.2)$$

It is obvious that the kinetic energy of mass m_1 is

$$K_1 = 0 \qquad\qquad (G.3)$$

and the potential energy

$$V_1 = m_1 g d_1 \qquad\qquad (G.4)$$

In the case of masses m_2 and m_3 the expressions for the
velocity squared of the masses must be obtained in order to
compute the kinetic energy. Thus, we will first write the
expressions for the Cartesian position coordinates and then
differentiate them to obtain the velocity.

For the mass m_2 the Cartesian position coordinates are

$$x_2 = d_2 \cos\theta_2 \cos\theta_1 \tag{G.5}$$

$$y_2 = d_2 \cos\theta_2 \sin\theta_1 \tag{G.6}$$

$$z_2 = d_1 + d_2 \sin\theta_2 \tag{G.7}$$

and the Cartesian components of the velocity are then

$$\dot{x}_2 = -d_2 \sin\theta_2 \cos\theta_1 \dot{\theta}_2 - d_2 \cos\theta_2 \sin\theta_1 \dot{\theta}_1 \tag{G.8}$$

$$\dot{y}_2 = -d_2 \sin\theta_2 \sin\theta_1 \dot{\theta}_2 + d_2 \cos\theta_2 \cos\theta_1 \dot{\theta}_1 \tag{G.9}$$

$$\dot{z}_2 = d_2 \cos\theta_2 \dot{\theta}_2 \tag{G.10}$$

The magnitude of the velocity squared is then

$$u_2{}^2 = \dot{x}_2{}^2 + \dot{y}_2{}^2 + \dot{z}_2{}^2$$

$$= d_2{}^2 (\cos^2\theta_2 \dot{\theta}_1{}^2 + \dot{\theta}_2{}^2) \tag{G.11}$$

and the kinetic energy is

$$K_2 = 1/2 m_2 d_2{}^2 (\cos^2\theta_2 \dot{\theta}_1{}^2 + \dot{\theta}_2{}^2) \tag{G.12}$$

The height of the mass m_2 is expressed by equation G.7, and the potential energy is then

$$V_2 = m_2 g (d_1 + d_2 \sin\theta_2) \tag{G.13}$$

For the mass m_3 the Cartesian position coordinates are

$$x_3 = [d_2 \cos\theta_2 + d_3 \cos(\theta_2 + \theta_3)] \cos\theta_1 \tag{G.14}$$

$$y_3 = [d_2 \cos\theta_2 + d_3 \cos(\theta_2 + \theta_3)] \sin\theta_1 \tag{G.15}$$

$$z_3 = d_1 + d_2 \sin\theta_2 + d_3 \sin(\theta_2 + \theta_3) \tag{G.16}$$

and the Cartesian components of the velocity are then

$$\dot{x}_3 = -[d_2\cos\theta_2 + d_3\cos(\theta_2+\theta_3)]\sin\theta_1\dot{\theta}_1 \tag{G.17}$$
$$-[d_2\sin\theta_2\dot{\theta}_2 + d_3\sin(\theta_2+\theta_3)(\dot{\theta}_2+\dot{\theta}_3)]\cos\theta_1$$

$$\dot{y}_3 = [d_2\cos\theta_2 + d_3\cos(\theta_2+\theta_3)]\cos\theta_1\dot{\theta}_1 \tag{G.18}$$
$$-[d_2\sin\theta_2\dot{\theta}_2 + d_3\sin(\theta_2+\theta_3)(\dot{\theta}_2+\dot{\theta}_3)]\sin\theta_1$$

$$\dot{z}_3 = d_2\cos\theta_2\dot{\theta}_2 + d_3\cos(\theta_2+\theta_3)(\dot{\theta}_2+\dot{\theta}_3) \tag{G.19}$$

The magnitude of the velocity squared is then

$$u_3^2 = \dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2$$
$$= d_2^2[\cos^2\theta_2\dot{\theta}_1^2 + \dot{\theta}_2^2] \tag{G.20}$$
$$+2d_2d_3[\cos\theta_2\cos(\theta_2+\theta_3)\dot{\theta}_1^2 + \cos\theta_3\dot{\theta}_2(\dot{\theta}_2+\dot{\theta}_3)]$$
$$+d_3^2[\cos^2(\theta_2+\theta_3)\dot{\theta}_1^2 + \dot{\theta}_2^2 + 2\dot{\theta}_2\dot{\theta}_3 + \dot{\theta}_3^2]$$

and the kinetic energy is

$$K_3 = 1/2\,m_3d_2^2[\cos^2\theta_2\dot{\theta}_1^2 + \dot{\theta}_2^2]$$
$$+m_3d_2d_3[\cos\theta_2\cos(\theta_2+\theta_3)\dot{\theta}_1^2 + \cos\theta_3\dot{\theta}_2(\dot{\theta}_2+\dot{\theta}_3)] \tag{G.21}$$
$$+1/2\,m_3d_3^2[\cos^2(\theta_2+\theta_3)\dot{\theta}_1^2 + \dot{\theta}_2^2 + 2\dot{\theta}_2\dot{\theta}_3 + \dot{\theta}_3^2]$$

The height of the mass m_3 is expressed by equation G.16 and the potential energy is then

$$V_3 = m_3g[d_1 + d_2\sin\theta_2 + d_3\sin(\theta_2+\theta_3)] \tag{G.22}$$

The Langrangian, $L = K_i - V_i$, is then obtained from equations G.3, G.4, G.12, G.13, G.21, and G.22

$$L = [1/2(m_2+m_3)d_2^2\cos^2\theta_2 + m_3d_2d_3\cos\theta_2\cos(\theta_2+\theta_3)$$
$$+1/2\,m_3d_3^2\cos^2(\theta_2+\theta_3) + 1/2\,J_{m1}]\dot{\theta}_1^2 \tag{G.23}$$
$$+[1/2(m_2+m_3)d_2^2 + m_3d_2d_3\cos\theta_3 + 1/2(m_3d_3^2) + 1/2\,J_{m2}]\dot{\theta}_2^2$$

$$+[1/2(m_3d_3{}^2+J_{m3})]\dot{\theta}_3{}^2$$

$$+m_3(d_3{}^2+d_2d_3\cos\theta_3)\dot{\theta}_2\dot{\theta}_3$$

$$-[(m_1+m_2+m_3)d_1-(m_2+m_3)d_2\sin\theta_2-m_3d_3\sin(\theta_2+\theta_3)]g$$

Taking partial derivatives of equation G.23 with respect to θ_1, θ_2 and θ_3

$$\frac{\partial L}{\partial \theta_1} = 0 \tag{G.24}$$

$$\frac{\partial L}{\partial \theta_2} = -[(m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2+m_3d_2d_3\sin(2\theta_2+\theta_3)$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1{}^2 \tag{G.25}$$
$$-(m_2+m_3)gd_2\cos\theta_2-m_3gd_3\cos(\theta_2+\theta_3)$$

$$\frac{\partial L}{\partial \theta_3} = -[m_3d_2d_3\sin(\theta_2+\theta_3)\cos\theta_2$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1{}^2$$
$$-m_3d_2d_3\sin\theta_3\dot{\theta}_2{}^2 \tag{G.26}$$
$$-m_3d_2d_3\sin\theta_3\dot{\theta}_2\dot{\theta}_3$$
$$-m_3gd_3\cos(\theta_2+\theta_3)$$

Taking partial derivatives of equation G.23 with respect to θ_1, θ_2 and θ_3

$$\frac{\partial L}{\partial \theta_1} = [(m_2+m_3)d_2{}^2\cos^2\theta_2+2m_3d_2d_3\cos\theta_2\cos(\theta_2+\theta_3) \tag{G.27}$$
$$+m_3d_3{}^2\cos^2(\theta_2+\theta_3)+J_{m1}]\dot{\theta}_1$$

$$\frac{\partial L}{\partial \theta_2} = [(m_2+m_3)d_2{}^2+2m_3d_2d_3\cos\theta_3+m_3d_3{}^2+J_{m2}]\dot{\theta}_2 \tag{G.28}$$
$$+[m_3d_2d_3\cos\theta_3+m_3d_3{}^2]\dot{\theta}_3$$

233

$$\frac{\partial L}{\partial \theta_3} = [m_3 d_3{}^2 + J_{m3}]\dot{\theta}_3 + [m_3 d_2 d_3 \cos\theta_3 + m_3 d_3{}^2]\dot{\theta}_2 \qquad (G.29)$$

Taking derivatives of equations G.27, G.28, and G.29 with respect to time

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = [m_3(d_2\cos\theta_2 + d_3\cos(\theta_2+\theta_3))^2$$
$$+ m_2 d_2{}^2\cos^2\theta_2 + J_{m1}]\ddot{\theta}_1$$
$$- 2[(m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2$$
$$+ m_3 d_2 d_3 \sin(2\theta_2+\theta_3) \qquad (G.30)$$
$$+ m_3 d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1\dot{\theta}_2$$
$$- 2[m_3 d_2 d_3 \sin(\theta_2+\theta_3)\cos\theta_2$$
$$+ m_3 d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1\dot{\theta}_3$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = [(m_2+m_3)d_2{}^2 + 2m_3 d_2 d_3\cos\theta_3 + m_3 d_3{}^2 + J_{m2}]\ddot{\theta}_2$$
$$+ [m_3 d_2 d_3 \cos\theta_3 + m_3 d_3{}^2]\ddot{\theta}_3$$
$$- 2m_3 d_2 d_3 \sin\theta_3 \dot{\theta}_2\dot{\theta}_3 \qquad (G.31)$$
$$- m_3 d_2 d_3 \sin\theta_3 \dot{\theta}_3{}^2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_3}\right) = [m_3 d_3{}^2 + J_{m3}]\ddot{\theta}_3$$
$$+ [m_3 d_2 d_3 \cos\theta_3 + m_3 d_3{}^2]\ddot{\theta}_2 \qquad (G.32)$$
$$- m_3 d_2 d_3 \sin\theta_3 \dot{\theta}_2\dot{\theta}_3$$

The corresponding torques are given by

$$T_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i}, \qquad \text{for} \quad i=1,2,3 \qquad (G.33)$$

Substituting equations G.24 - G.32 into equation G.33

$$T_1 = [m_3(d_2\cos\theta_2+d_3\cos(\theta_2+\theta_3))^2$$
$$+m_2d_2{}^2\cos^2\theta_2+J_{m1}]\ddot{\theta}_1$$
$$-2[(m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2+m_3d_2d_3\sin(2\theta_2+\theta_3)$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1\dot{\theta}_2 \qquad\text{(G.34)}$$
$$-2[m_3d_2d_3\sin(\theta_2+\theta_3)\cos\theta_2$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1\dot{\theta}_3$$

$$T_2 = [(m_2+m_3)d_2{}^2+2m_3d_2d_3\cos\theta_3+m_3d_3{}^2+J_{m2}]\ddot{\theta}_2$$
$$+[m_3d_2d_3\cos\theta_3+m_3d_3{}^2]\ddot{\theta}_3$$
$$-2m_3d_2d_3\sin\theta_3\dot{\theta}_2\dot{\theta}_3-m_3d_2d_3\sin\theta_3\dot{\theta}_3{}^2 \qquad\text{(G.35)}$$
$$+[(m_2+m_3)d_2{}^2\sin\theta_2\cos\theta_2+m_3d_2d_3\sin(2\theta_2+\theta_3)$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1{}^2$$
$$+(m_2+m_3)gd_2\cos\theta_2+m_3gd_3\cos(\theta_2+\theta_3)$$

$$T_3 = [m_3d_3{}^2+J_{m3}]\ddot{\theta}_3+[m_3d_2d_3\cos\theta_3+m_3d_3{}^2]\ddot{\theta}_2$$
$$+[m_3d_2d_3\sin(\theta_2+\theta_3)\cos\theta_2 \qquad\text{(G.36)}$$
$$+m_3d_3{}^2\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)]\dot{\theta}_1{}^2$$
$$+m_3d_2d_3\sin\theta_3\dot{\theta}_2{}^2+m_3gd_3\cos(\theta_2+\theta_3)$$

APPENDIX H

DSL PROGRAM FOR THE THREE LINK REVOLUTE ROBOT UNDER
DIFFERENT LOAD CONDITIONS (NO GRAVITY)


```
TITLE MODEL OF REVOLUTE 3-DEG OF FREEDOM ROBOT ARM
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=0.4225,KM2=0.4225,KM3=4.0
PARAM VSAT=150.0,M1=0.268,M2=0.227,M3=0.041
PARAM D1=15.0,D2=10.0,D3=10.0,G=386.4
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM REF1=1.0 ,REF2=1.0 ,REF3=1.0
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:   THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* REF1,REF2,REF3:THE COMMANDED TIP POSITION IN RADS
* T:   THE SAMPLING INTERVAL
INITIAL
        FLAG1=0
        FLAG2=0
        FLAG3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        C1=0.
        C2=0.
        C3=0.
        C1DT=0.
        C2DT=0.
        C3DT=0.
        C1DDT=0.
        C2DDT=0.
        C3DDT=0.
        CR1=0.
        CR2=0.
        CR3=0.
        CR1DT=0.
        CR2DT=0.
        CR3DT=0.
        CR1DDT=0.
```

236

```
          CR2DDT=0.
          CR3DDT=0.
          TL1=0.
          TL2=0.
          TL3=0.
          MP1=0.
          MP2=0.
          MP3=0.
          MT1=0.
          MT2=0.
          MT3=0.
          A1=SQRT(2.*KM1*VSAT)
          A2=SQRT(2.*KM2*VSAT)
          A3=SQRT(2.*KM3*VSAT)
DERIVATIVE
          RR1=REF1*STEP(0.0)
          RR2=REF2*STEP(0.0)
          RR3=REF3*STEP(0.0)
          E1=RR1-C1
          E2=RR2-C2
          E3=RR3-C3
NOSORT
          D11 = M3*(D2*COS(CR2)+D3*COS(CR2+CR3))**2...
                +M2*D2**2*(COS(CR2))**2
          D112= 2*((M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3))
          D113= 2*(M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3))
          D22 = (M2+M3)*D2**2+M3*D3**2+2*M3*D2*D3*SIN(CR3)
          D23 = M3*D3**2+M3*D2*D3*SIN(CR3)
          D211= (M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3)
          D223= 2*M3*D2*D3*SIN(CR3)
          D233= M3*D2*D3*SIN(CR3)
          D32 = M3*D3**2+M3*D2*D3*COS(CR3)
          D33 = M3*D3**2
          D311= M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3)
          D322= M3*D2*D3*SIN(CR3)
          G2  = (M2+M3)*G*D2*COS(CR2)+M3*G*D3*COS(CR2+CR3)
          G3  = M3*G*D3*COS(CR2+CR3)
          TL1 = -D112*CR1DT*CR2DT-D113*CR1DT*CR3DT
          TL2 = D23*CR3DDT+D211*CR1DT**2-D233*CR3DT**2...
                -D223*CR2DT*CR3DT
          TL3 = D32*CR2DDT+D311*CR1DT**2+D322*CR2DT**2
          JTOT1 = D11+J1
          JTOT2 = D22+J2
          JTOT3 = D33+J3
          IF(E1.LT.0.0) X1DOT=-A1*K1*SQRT(ABS(E1))
```

```
          IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
          IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
          IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
          IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
          IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT


**********************************************************

          KC1DOT=K*C1DT
          X1DOTE=X1DOT-KC1DOT
          V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
          C1DDT=V1*KM1
NOSORT
          IF (FLAG1.EQ.1) GO TO 2
          IF (V1.LT.VSAT.AND.TIME.GT.0.00005) FLAG1=1
          NSW1=N1
2         CONTINUE
SORT
          C1DT=INTGRL(0.0,C1DDT)
          C1=INTGRL(0.0,C1DT)
          VM1=V1-KV1*CR1DT
          MP1=REALPL(0.0,L1/R1,VM1/R1)
          MT1=KT1*MP1
          MT1E=MT1-BM1*CR1DT-TL1
          CR1DDT=(1./JTOT1)*MT1E
          CR1DT=INTGRL(0.0,CR1DDT)
          CR1=INTGRL(0.0,CR1DT)


**********************************************************

          KC2DOT=K*C2DT
          X2DOTE=X2DOT-KC2DOT
          V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
NOSORT
          IF (FLAG2.EQ.1) GO TO 4
          IF (V2.LT.VSAT.AND.TIME.GT.0.00005) FLAG2=1
          NSW2=N2
4         CONTINUE
SORT
          C2DDT=V2*KM2
          C2DT=INTGRL(0.0,C2DDT)
          C2=INTGRL(0.0,C2DT)
          VM2=V2-KV2*CR2DT
          MP2=REALPL(0.0,L2/R2,VM2/R2)
          MT2=KT2*MP2
          MT2E=MT2-BM2*CR2DT-TL2
          CR2DDT=(1./JTOT2)*MT2E
          CR2DT=INTGRL(0.0,CR2DDT)
          CR2=INTGRL(0.0,CR2DT)
```

```
**********************************************************

          KC3DOT=K*C3DT
          X3DOTE=X3DOT-KC3DOT
          V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
NOSORT
          IF (FLAG3.EQ.1) GO TO 6
          IF (V3.LT.VSAT.AND.TIME.GT.0.00005) FLAG3=1
          NSW3=N3
6         CONTINUE
SORT
          C3DDT=V3*KM3
          C3DT=INTGRL(0.0,C3DDT)
          C3=INTGRL(0.0,C3DT)
          VM3=V3-KV3*CR3DT
          MP3=REALPL(0.0,L3/R3,VM3/R3)
          MT3=KT3*MP3
          MT3E=MT3-BM3*CR3DT-TL3
          CR3DDT=(1./JTOT3)*MT3E
          CR3DT=INTGRL(0.0,CR3DDT)
          CR3=INTGRL(0.0,CR3DT)


**********************************************************

SAMPLE
NOSORT
          IF (N3.EQ.0) GO TO 30
          IF (N2.EQ.0) GO TO 20
          IF (N3.EQ.0) GO TO 10
          C3=CR3
          C2=CR2
          C1=CR1
          KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
          KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
          KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
          IF (FLAG3.EQ.0) KM3=KS3
          IF (FLAG2.EQ.0) KM2=KS2
          IF (FLAG1.EQ.0) KM1=KS1
          IF (N3.GE.2) CR3DTL=(CR3-CR32L)/(2.*T)
          IF (N2.GE.2) CR2DTL=(CR2-CR22L)/(2.*T)
          IF (N1.GE.2) CR1DTL=(CR1-CR12L)/(2.*T)
          IF(FLAG3.EQ.0) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
          IF(FLAG2.EQ.0) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
          IF(FLAG1.EQ.0) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
          IF (N3.EQ.NSW3.AND.FLAG3.EQ.1) GO TO 30
          IF (N2.EQ.NSW2.AND.FLAG2.EQ.1) GO TO 20
          IF (N1.EQ.NSW1.AND.FLAG1.EQ.1) GO TO 10
          IF (FLAG3.EQ.1) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
          IF (FLAG2.EQ.1) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
          IF (FLAG1.EQ.1) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
30        N3=N3+1
```

239

```
20        N2=N2+1
10        N1=N1+1
          CR3DTL=C3DT
          CR2DTL=C2DT
          CR1DTL=C1DT
          CR32L=CR3LST
          CR22L=CR2LST
          CR12L=CR1LST
          CR3LST=CR3
          CR2LST=CR2
          CR1LST=CR1
SORT

***********************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.5,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,X1DOT,X2DOT,X3DOT,C1DT,C2DT,C3DT,...
       CR1DT,CR2DT,CR3DT,CR1,CR2,CR3
SAVE (S2) 0.001,C1,CR1,RR1,C2,CR2,RR2,C3,CR3,RR3
SAVE (S3) 0.001,E1,E2,E3
PRINT 0.01, E1,E2,E3
GRAPH (G1/S1,DE=TEK618,PO=1)...
       CR1(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
       C1DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
       CR1DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
       X1DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
       CR2(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
       C2DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
       CR2DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
       X2DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
       CR3(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
       C3DT(LE=3,NI=4,LO=-8,SC=8,UN='RAD/SEC'),...
       CR3DT(LE=3,NI=4,LO=-8,PO=6.0,SC=8,UN='RAD/SEC'),...
       X3DOT(LE=3,NI=4,LO=-8,SC=8,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=5,UN='SEC'),...
       C1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
       CR1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
       RR1(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
       TIME(LE=6.0,NI=5,UN='SEC),...
       C2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
       CR2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
       RR2(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
       TIME(LE=6.0,NI=5,UN='SEC'),...
       C3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
       CR3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
```

240

```
        RR3(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH  (G7/S3,DE=TEK618,PO=1,.5)...
        TIME(LE=5.0,NI=5,UN='SEC'),...
        E1(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2),...
        E2(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2,PO=5.0),...
        E3(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2,PO=6.2)
LABEL  (G1) PHASE PLANE (CDT,CRDT,XDOT.VS.CR)
LABEL  (G4) STEP RESPONSE
LABEL  (G7) ERROR .VS. TIME
END
STOP


*************************************************************
```

DSL PROGRAM FOR THE THREE LINK REVOLUTE ROBOT
WITH DISTURBANCE

```
TITLE MODEL OF REVOLUTE 3-DEG OF FREEDOM ROBOT ARM
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=0.4225,KM2=0.4225,KM3=4.0
PARAM VSAT=150.0,M1=0.268,M2=0.227,M3=0.041
PARAM D1=15.0,D2=10.0,D3=10.0,G=386.4
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM REF1=1.0 ,REF2=1.0 ,REF3=1.0
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALLING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* REF1,REF2,REF3:THE COMMANDED TIP POSITION IN RADS
* T:  THE SAMPLING INTERVAL
INITIAL
        FLAG1=0
        FLAG2=0
        FLAG3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        C1=0.
        C2=0.
        C3=0.
        C1DT=0.
        C2DT=0.
        C3DT=0.
        C1DDT=0.
        C2DDT=0.
        C3DDT=0.
        CR1=0.
        CR2=0.
        CR3=0.
        CR1DT=0.
        CR2DT=0.
        CR3DT=0.
        CR1DDT=0.
```

```
          CR2DDT=0.
          CR3DDT=0.
          TL1=0.
          TL2=0.
          TL3=0.
          MP1=0.
          MP2=0.
          MP3=0.
          MT1=0.
          MT2=0.
          MT3=0.
          A1=SQRT(2.*KM1*VSAT)
          A2=SQRT(2.*KM2*VSAT)
          A3=SQRT(2.*KM3*VSAT)
DERIVATIVE
          RR1=REF1*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          RR2=REF2*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          RR3=REF3*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          E1=RR1-C1
          E2=RR2-C2
          E3=RR3-C3
NOSORT
          D11 = M3*(D2*COS(CR2)+D3*COS(CR2+CR3))**2...
                +M2*D2**2*(COS(CR2))**2
          D112= 2*((M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3))
          D113= 2*(M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3))
          D22 = (M2+M3)*D2**2+M3*D3**2+2*M3*D2*D3*SIN(CR3)
          D23 = M3*D3**2+M3*D2*D3*SIN(CR3)
          D211= (M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3)
          D223= 2*M3*D2*D3*SIN(CR3)
          D233= M3*D2*D3*SIN(CR3)
          D32 = M3*D3**2+M3*D2*D3*COS(CR3)
          D33 = M3*D3**2
          D311= M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3)
          D322= M3*D2*D3*SIN(CR3)
          G2  = (M2+M3)*G*D2*COS(CR2)+M3*G*D3*COS(CR2+CR3)
          G3  = M3*G*D3*COS(CR2+CR3)
          TL1 = -D112*CR1DT*CR2DT-D113*CR1DT*CR3DT
          TL2 = D23*CR3DDT+D211*CR1DT**2-D233*CR3DT**2...
                -D223*CR2DT*CR3DT
          TL3 = D32*CR2DDT+D311*CR1DT**2+D322*CR2DT**2
          JTOT1 = D11+J1
          JTOT2 = D22+J2
          JTOT3 = D33+J3
          IF(E1.LT.0.0) X1DOT=-A1*K1*SQRT(ABS(E1))
```

```
          IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
          IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
          IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
          IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
          IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

****************************************************************

          KC1DOT=K*C1DT
          X1DOTE=X1DOT-KC1DOT
          V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
          C1DDT=V1*KM1
NOSORT
          IF (FLAG1.EQ.1) GO TO 2
          IF (V1.LT.VSAT.AND.TIME.GT.0.00005) FLAG1=1
          NSW1=N1
2         CONTINUE
SORT
          C1DT=INTGRL(0.0,C1DDT)
          C1=INTGRL(0.0,C1DT)
          VM1=V1-KV1*CR1DT
          MP1=REALPL(0.0,L1/R1,VM1/R1)
          MT1=KT1*MP1
          MT1E=MT1-BM1*CR1DT-TL1
          CR1DDT=(1./JTOT1)*MT1E
          CR1DT=INTGRL(0.0,CR1DDT)
          CR1=INTGRL(0.0,CR1DT)

****************************************************************

          KC2DOT=K*C2DT
          X2DOTE=X2DOT-KC2DOT
          V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
NOSORT
          IF (FLAG2.EQ.1) GO TO 4
          IF (V2.LT.VSAT.AND.TIME.GT.0.00005) FLAG2=1
          NSW2=N2
4         CONTINUE
SORT
          C2DDT=V2*KM2
          C2DT=INTGRL(0.0,C2DDT)
          C2=INTGRL(0.0,C2DT)
          VM2=V2-KV2*CR2DT
          MP2=REALPL(0.0,L2/R2,VM2/R2)
          MT2=KT2*MP2
          MT2E=MT2-BM2*CR2DT-TL2
          CR2DDT=(1./JTOT2)*MT2E
          CR2DT=INTGRL(0.0,CR2DDT)
          CR2=INTGRL(0.0,CR2DT)

****************************************************************
```

244

```
        KC3DOT=K*C3DT
        X3DOTE=X3DOT-KC3DOT
        V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
NOSORT
        IF (FLAG3.EQ.1) GO TO 6
        IF (V3.LT.VSAT.AND.TIME.GT.0.00005) FLAG3=1
        NSW3=N3
6       CONTINUE
SORT
        C3DDT=V3*KM3
        C3DT=INTGRL(0.0,C3DDT)
        C3=INTGRL(0.0,C3DT)
        VM3=V3-KV3*CR3DT
        MP3=REALPL(0.0,L3/R3,VM3/R3)
        MT3=KT3*MP3
        MT3E=MT3-BM3*CR3DT-TL3
        CR3DDT=(1./JTOT3)*MT3E
        CR3DT=INTGRL(0.0,CR3DDT)
        CR3=INTGRL(0.0,CR3DT)


************************************************************

SAMPLE
NOSORT
        IF (N3.EQ.0) GO TO 30
        IF (N2.EQ.0) GO TO 20
        IF (N3.EQ.0) GO TO 10
        C3=CR3
        C2=CR2
        C1=CR1
        KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
        KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
        KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
        IF (FLAG3.EQ.0) KM3=KS3
        IF (FLAG2.EQ.0) KM2=KS2
        IF (FLAG1.EQ.0) KM1=KS1
        IF (N3.GE.2) CR3DTL=(CR3-CR32L)/(2.*T)
        IF (N2.GE.2) CR2DTL=(CR2-CR22L)/(2.*T)
        IF (N1.GE.2) CR1DTL=(CR1-CR12L)/(2.*T)
        IF(FLAG3.EQ.0) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
        IF(FLAG2.EQ.0) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
        IF(FLAG1.EQ.0) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
        IF (N3.EQ.NSW3.AND.FLAG3.EQ.1) GO TO 30
        IF (N2.EQ.NSW2.AND.FLAG2.EQ.1) GO TO 20
        IF (N1.EQ.NSW1.AND.FLAG1.EQ.1) GO TO 10
        IF (FLAG3.EQ.1) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
        IF (FLAG2.EQ.1) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
        IF (FLAG1.EQ.1) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
30      N3=N3+1
20      N2=N2+1
10      N1=N1+1
```

```
                CR3DTL=C3DT
                CR2DTL=C2DT
                CR1DTL=C1DT
                CR32L=CR3LST
                CR22L=CR2LST
                CR12L=CR1LST
                CR3LST=CR3
                CR2LST=CR2
                CR1LST=CR1
SORT


********************************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.5,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,X1DOT,X2DOT,X3DOT,C1DT,C2DT,C3DT,...
      CR1DT,CR2DT,CR3DT,CR1,CR2,CR3
SAVE (S2) 0.001,C1,CR1,RR1,C2,CR2,RR2,C3,CR3,RR3
GRAPH (G1/S1,DE=TEK618,PO=1)...
      CR1(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C1DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR1DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X1DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
      CR2(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C2DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR2DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X2DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
      CR3(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C3DT(LE=3,NI=4,LO=-8,SC=8,UN='RAD/SEC'),...
      CR3DT(LE=3,NI=4,LO=-8,PO=6.0,SC=8,UN='RAD/SEC'),...
      X3DOT(LE=3,NI=4,LO=-8,SC=8,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=5,UN='SEC'),...
      C1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR1(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
      TIME(LE=6.0,NI=5,UN='SEC'),...
      C2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR2(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
      TIME(LE=6.0,NI=5,UN='SEC'),...
      C3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR3(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
LABEL (G1) PHASE PLANE (CDT,CRDT,XDOT.VS.CR)
LABEL (G4) STEP RESPONSE
END
STOP
```

APPENDIX J

DSL PROGRAM FOR THE THREE LINK REVOLUTE ROBOT
UNDER DIFFERENT LOAD CONDITIONS
(GRAVITATIONAL TORQUES INCLUDED)

```
TITLE MODEL OF REVOLUTE 3-DEG OF FREEDOM ROBOT ARM
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=0.4225,KM2=0.4225,KM3=4.0
PARAM VSAT=150.0,M1=0.268,M2=0.227,M3=0.041
PARAM D1=15.0,D2=10.0,D3=10.0,G=386.4
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM REF1=1.0 ,REF2=1.0 ,REF3=1.0
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALLING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* REF1,REF2,REF3:THE COMMANDED TIP POSITION IN RADS
* T:  THE SAMPLING INTERVAL
INITIAL
         FLAG1=0
         FLAG2=0
         FLAG3=0
         N1=0
         N2=0
         N3=0
         X1DOT=0.
         X2DOT=0.
         X3DOT=0.
         C1=0.
         C2=0.
         C3=0.
         C1DT=0.
         C2DT=0.
         C3DT=0.
         C1DDT=0.
         C2DDT=0.
         C3DDT=0.
         CR1=0.
         CR2=0.
         CR3=0.
         CR1DT=0.
         CR2DT=0.
         CR3DT=0.
```

```
        CR1DDT=0.
        CR2DDT=0.
        CR3DDT=0.
        TL1=0.
        TL2=0.
        TL3=0.
        MP1=0.
        MP2=0.
        MP3=0.
       'MT1=0.
        MT2=0.
        MT3=0.
        A1=SQRT(2.*KM1*VSAT)
        A2=SQRT(2.*KM2*VSAT)
        A3=SQRT(2.*KM3*VSAT)
DERIVATIVE
        RR1=REF1*STEP(0.0)
        RR2=REF2*STEP(0.0)
        RR3=REF3*STEP(0.0)
        E1=RR1-C1
        E2=RR2-C2
        E3=RR3-C3
NOSORT
        D11 = M3*(D2*COS(CR2)+D3*COS(CR2+CR3))**2...
              +M2*D2**2*(COS(CR2))**2
        D112= 2*((M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
              +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
              +M3*D2*D3*SIN(2*CR2+CR3))
        D113= 2*(M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
              +M3*D2*D3*COS(CR2)*SIN(CR2+CR3))
        D22 = (M2+M3)*D2**2+M3*D3**2+2*M3*D2*D3*SIN(CR3)
        D23 = M3*D3**2+M3*D2*D3*SIN(CR3)
        D211= (M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
              +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
              +M3*D2*D3*SIN(2*CR2+CR3)
        D223= 2*M3*D2*D3*SIN(CR3)
        D233= M3*D2*D3*SIN(CR3)
        D32 = M3*D3**2+M3*D2*D3*COS(CR3)
        D33 = M3*D3**2
        D311= M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
              +M3*D2*D3*COS(CR2)*SIN(CR2+CR3)
        D322= M3*D2*D3*SIN(CR3)
        G2  = (M2+M3)*G*D2*COS(CR2)+M3*G*D3*COS(CR2+CR3)
        G3  = M3*G*D3*COS(CR2+CR3)
        TL1 = -D112*CR1DT*CR2DT-D113*CR1DT*CR3DT
        TL2 = D23*CR3DDT+D211*CR1DT**2-D233*CR3DT**2...
              -D223*CR2DT*CR3DT+G2
        TL3 = D32*CR2DDT+D311*CR1DT**2+D322*CR2DT**2+G3
        JTOT1 = D11+J1
        JTOT2 = D22+J2
        JTOT3 = D33+J3
```

248

```
        IF(E1.LT.0.0)  X1DOT=-A1*K1*SQRT(ABS(E1))
        IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
        IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
        IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
        IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
        IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

***********************************************************

        KC1DOT=K*C1DT
        X1DOTE=X1DOT-KC1DOT
        V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
        C1DDT=V1*KM1
NOSORT
        IF (FLAG1.EQ.1) GO TO 2
        IF (V1.LT.VSAT.AND.TIME.GT.0.00005) FLAG1=1
        NSW1=N1
2       CONTINUE
SORT
        C1DT=INTGRL(0.0,C1DDT)
        C1=INTGRL(0.0,C1DT)
        VM1=V1-KV1*CR1DT
        MP1=REALPL(0.0,L1/R1,VM1/R1)
        MT1=KT1*MP1
        MT1E=MT1-BM1*CR1DT-TL1
        CR1DDT=(1./JTOT1)*MT1E
        CR1DT=INTGRL(0.0,CR1DDT)
        CR1=INTGRL(0.0,CR1DT)

***********************************************************

        KC2DOT=K*C2DT
        X2DOTE=X2DOT-KC2DOT
        V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
NOSORT
        IF (FLAG2.EQ.1) GO TO 4
        IF (V2.LT.VSAT.AND.TIME.GT.0.00005) FLAG2=1
        NSW2=N2
4       CONTINUE
SORT
        C2DDT=V2*KM2
        C2DT=INTGRL(0.0,C2DDT)
        C2=INTGRL(0.0,C2DT)
        VM2=V2-KV2*CR2DT
        MP2=REALPL(0.0,L2/R2,VM2/R2)
        MT2=KT2*MP2
        MT2E=MT2-BM2*CR2DT-TL2
        CR2DDT=(1./JTOT2)*MT2E
        CR2DT=INTGRL(0.0,CR2DDT)
        CR2=INTGRL(0.0,CR2DT)
```

```
*************************************************************

        KC3DOT=K*C3DT
        X3DOTE=X3DOT-KC3DOT
        V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
NOSORT

        IF (FLAG3.EQ.1) GO TO 6
        IF (V3.LT.VSAT.AND.TIME.GT.0.00005) FLAG3=1
        NSW3=N3
6       CONTINUE
SORT

        C3DDT=V3*KM3
        C3DT=INTGRL(0.0,C3DDT)
        C3=INTGRL(0.0,C3DT)
        VM3=V3-KV3*CR3DT
        MP3=REALPL(0.0,L3/R3,VM3/R3)
        MT3=KT3*MP3
        MT3E=MT3-BM3*CR3DT-TL3
        CR3DDT=(1./JTOT3)*MT3E
        CR3DT=INTGRL(0.0,CR3DDT)
        CR3=INTGRL(0.0,CR3DT)


*************************************************************

SAMPLE
NOSORT
        IF (N3.EQ.0) GO TO 30
        IF (N2.EQ.0) GO TO 20
        IF (N3.EQ.0) GO TO 10
        C3=CR3
        C2=CR2
        C1=CR1
        KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
        KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
        KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
        IF (FLAG3.EQ.0) KM3=KS3
        IF (FLAG2.EQ.0) KM2=KS2
        IF (FLAG1.EQ.0) KM1=KS1
        IF (N3.GE.2) CR3DTL=(CR3-CR32L)/(2.*T)
        IF (N2.GE.2) CR2DTL=(CR2-CR22L)/(2.*T)
        IF (N1.GE.2) CR1DTL=(CR1-CR12L)/(2.*T)
        IF(FLAG3.EQ.0) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
        IF(FLAG2.EQ.0) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
        IF(FLAG1.EQ.0) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
        IF (N3.EQ.NSW3.AND.FLAG3.EQ.1) GO TO 30
        IF (N2.EQ.NSW2.AND.FLAG2.EQ.1) GO TO 20
        IF (N1.EQ.NSW1.AND.FLAG1.EQ.1) GO TO 10
        IF (FLAG3.EQ.1) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
        IF (FLAG2.EQ.1) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
        IF (FLAG1.EQ.1) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
30      N3=N3+1
```

```
20        N2=N2+1
10        N1=N1+1
          CR3DTL=C3DT
          CR2DTL=C2DT
          CR1DTL=C1DT
          CR32L=CR3LST
          CR22L=CR2LST
          CR12L=CR1LST
          CR3LST=CR3
          CR2LST=CR2
          CR1LST=CR1
SORT

***********************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.5,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,X1DOT,X2DOT,X3DOT,C1DT,C2DT,C3DT,...
      CR1DT,CR2DT,CR3DT,CR1,CR2,CR3
SAVE (S2) 0.001,C1,CR1,RR1,C2,CR2,RR2,C3,CR3,RR3
SAVE (S3) 0.001,E1,E2,E3
PRINT 0.01, E1,E2,E3
GRAPH (G1/S1,DE=TEK618,PO=1)...
      CR1(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C1DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR1DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X1DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
      CR2(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C2DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR2DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X2DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
      CR3(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C3DT(LE=3,NI=4,LO=-8,SC=8,UN='RAD/SEC'),...
      CR3DT(LE=3,NI=4,LO=-8,PO=6.0,SC=8,UN='RAD/SEC'),...
      X3DOT(LE=3,NI=4,LO=-8,SC=8,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=5,UN='SEC'),...
      C1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR1(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
      TIME(LE=6.0,NI=5,UN='SEC),...
      C2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR2(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
      TIME(LE=6.0,NI=5,UN='SEC'),...
      C3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
```

251

```
        RR3(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH  (G7/S3,DE=TEK618,PO=1,.5)...
        TIME(LE=5.0,NI=5,UN='SEC'),...
        E1(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2),...
        E2(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2,PO=5.0),...
        E3(LE=8,NI=7,LO=-.2,UN='RAD',SC=.2,PO=6.2)
LABEL  (G1)  PHASE PLANE (CDT,CRDT,XDOT.VS.CR)
LABEL  (G4)  STEP RESPONSE
LABEL  (G7)  ERROR .VS. TIME
END
STOP


*********************************************************
```

DSL PROGRAM FOR THE THREE LINK REVOLUTE ROBOT WITH DISTURBANCE (GRAVITATIONAL TORQUES INCLUDED)

```
TITLE MODEL OF REVOLUTE 3-DEG OF FREEDOM ROBOT ARM
PARAM K=1.0,K1=0.6,K2=10000.0,KM1=0.4225,KM2=0.4225,KM3=4.0
PARAM VSAT=150.0,M1=0.268,M2=0.227,M3=0.041
PARAM D1=15.0,D2=10.0,D3=10.0,G=386.4
PARAM J1=0.033,J2=0.033,J3=0.033,R1=0.91,R2=0.91,R3=0.91
PARAM KT1=14.4,KT2=14.4,KT3=14.4,L1=.0001,L2=.0001,L3=.0001
PARAM BM1=0.04297,BM2=0.04297,BM3=0.04297,LOAD=0.0
PARAM KV1=0.1012,KV2=0.1012,KV3=0.1012,T=0.00025
PARAM REF1=1.0 ,REF2=1.0 ,REF3=1.0
INTGER SW1,SW2,SW3,N1,N2,N3
* K1: THE CURVE SCALLING CONSTANT
* K2: THE AMPLIFIER GAIN
* KM: THE INITIAL CONSTANT OF THE IDEAL (MODEL) MOTOR
* VSAT: THE SATURATION LIMITS OF THE AMPLIFIER
* K:  THE VELOCITY LOOP FEEDBACK GAIN (OF THE MODEL)
* REF1,REF2,REF3:THE COMMANDED TIP POSITION IN RADS
* T:  THE SAMPLING INTERVAL
INITIAL
        FLAG1=0
        FLAG2=0
        FLAG3=0
        N1=0
        N2=0
        N3=0
        X1DOT=0.
        X2DOT=0.
        X3DOT=0.
        C1=0.
        C2=0.
        C3=0.
        C1DT=0.
        C2DT=0.
        C3DT=0.
        C1DDT=0.
        C2DDT=0.
        C3DDT=0.
        CR1=0.
        CR2=0.
        CR3=0.
        CR1DT=0.
        CR2DT=0.
        CR3DT=0.
        CR1DDT=0.
```

```
          CR2DDT=0.
          CR3DDT=0.
          TL1=0.
          TL2=0.
          TL3=0.
          MP1=0.
          MP2=0.
          MP3=0.
          MT1=0.
          MT2=0.
          MT3=0.
          A1=SQRT(2.*KM1*VSAT)
          A2=SQRT(2.*KM2*VSAT)
          A3=SQRT(2.*KM3*VSAT)
DERIVATIVE
          RR1=REF1*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          RR2=REF2*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          RR3=REF3*STEP(0.0)+50*(STEP(.090)-STEP(.100))
          E1=RR1-C1
          E2=RR2-C2
          E3=RR3-C3
NOSORT
          D11 = M3*(D2*COS(CR2)+D3*COS(CR2+CR3))**2...
                +M2*D2**2*(COS(CR2))**2
          D112= 2*((M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3))
          D113= 2*(M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3))
          D22 = (M2+M3)*D2**2+M3*D3**2+2*M3*D2*D3*SIN(CR3)
          D23 = M3*D3**2+M3*D2*D3*SIN(CR3)
          D211= (M2+M3)*D2**2*COS(CR2)*SIN(CR2)...
                +M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*SIN(2*CR2+CR3)
          D223= 2*M3*D2*D3*SIN(CR3)
          D233= M3*D2*D3*SIN(CR3)
          D32 = M3*D3**2+M3*D2*D3*COS(CR3)
          D33 = M3*D3**2
          D311= M3*D3**2*COS(CR2+CR3)*SIN(CR2+CR3)...
                +M3*D2*D3*COS(CR2)*SIN(CR2+CR3)
          D322= M3*D2*D3*SIN(CR3)
          G2  = (M2+M3)*G*D2*COS(CR2)+M3*G*D3*COS(CR2+CR3)
          G3  = M3*G*D3*COS(CR2+CR3)
          TL1 = -D112*CR1DT*CR2DT-D113*CR1DT*CR3DT
          TL2 = D23*CR3DDT+D211*CR1DT**2-D233*CR3DT**2...
                -D223*CR2DT*CR3DT+G2
          TL3 = D32*CR2DDT+D311*CR1DT**2+D322*CR2DT**2+G3
          JTOT1 = D11+J1
          JTOT2 = D22+J2
          JTOT3 = D33+J3
          IF(E1.LT.0.0) X1DOT=-A1*K1*SQRT(ABS(E1))
```

```
        IF(E1.GE.0.0)  X1DOT=A1*K1*SQRT(E1)
        IF(E2.LT.0.0)  X2DOT=-A2*K1*SQRT(ABS(E2))
        IF(E2.GE.0.0)  X2DOT=A2*K1*SQRT(E2)
        IF(E3.LT.0.0)  X3DOT=-A3*K1*SQRT(ABS(E3))
        IF(E3.GE.0.0)  X3DOT=A3*K1*SQRT(E3)
SORT

****************************************************************

        KC1DOT=K*C1DT
        X1DOTE=X1DOT-KC1DOT
        V1=LIMIT(-VSAT,VSAT,K2*X1DOTE)
        C1DDT=V1*KM1
NOSORT
        IF (FLAG1.EQ.1) GO TO 2
        IF (V1.LT.VSAT.AND.TIME.GT.0.00005) FLAG1=1
        NSW1=N1
2       CONTINUE
SORT
        C1DT=INTGRL(0.0,C1DDT)
        C1=INTGRL(0.0,C1DT)
        VM1=V1-KV1*CR1DT
        MP1=REALPL(0.0,L1/R1,VM1/R1)
        MT1=KT1*MP1
        MT1E=MT1-BM1*CR1DT-TL1
        CR1DDT=(1./JTOT1)*MT1E
        CR1DT=INTGRL(0.0,CR1DDT)
        CR1=INTGRL(0.0,CR1DT)

****************************************************************

        KC2DOT=K*C2DT
        X2DOTE=X2DOT-KC2DOT
        V2=LIMIT(-VSAT,VSAT,K2*X2DOTE)
NOSORT
        IF (FLAG2.EQ.1) GO TO 4
        IF (V2.LT.VSAT.AND.TIME.GT.0.00005) FLAG2=1
        NSW2=N2
4       CONTINUE
SORT
        C2DDT=V2*KM2
        C2DT=INTGRL(0.0,C2DDT)
        C2=INTGRL(0.0,C2DT)
        VM2=V2-KV2*CR2DT
        MP2=REALPL(0.0,L2/R2,VM2/R2)
        MT2=KT2*MP2
        MT2E=MT2-BM2*CR2DT-TL2
        CR2DDT=(1./JTOT2)*MT2E
        CR2DT=INTGRL(0.0,CR2DDT)
        CR2=INTGRL(0.0,CR2DT)

****************************************************************
```

```
              KC3DOT=K*C3DT
              X3DOTE=X3DOT-KC3DOT
              V3=LIMIT(-VSAT,VSAT,K2*X3DOTE)
NOSORT
              IF (FLAG3.EQ.1) GO TO 6
              IF (V3.LT.VSAT.AND.TIME.GT.0.00005) FLAG3=1
              NSW3=N3
6             CONTINUE
SORT
              C3DDT=V3*KM3
              C3DT=INTGRL(0.0,C3DDT)
              C3=INTGRL(0.0,C3DT)
              VM3=V3-KV3*CR3DT
              MP3=REALPL(0.0,L3/R3,VM3/R3)
              MT3=KT3*MP3
              MT3E=MT3-BM3*CR3DT-TL3
              CR3DDT=(1./JTOT3)*MT3E
              CR3DT=INTGRL(0.0,CR3DDT)
              CR3=INTGRL(0.0,CR3DT)


***************************************************************

SAMPLE
NOSORT
              IF (N3.EQ.0) GO TO 30
              IF (N2.EQ.0) GO TO 20
              IF (N3.EQ.0) GO TO 10
              C3=CR3
              C2=CR2
              C1=CR1
              KS3=ABS(2.*CR3)/(((N3*T)**2)*V3)
              KS2=ABS(2.*CR2)/(((N2*T)**2)*V2)
              KS1=ABS(2.*CR1)/(((N1*T)**2)*V1)
              IF (FLAG3.EQ.0) KM3=KS3
              IF (FLAG2.EQ.0) KM2=KS2
              IF (FLAG1.EQ.0) KM1=KS1
              IF (N3.GE.2) CR3DTL=(CR3-CR32L)/(2.*T)
              IF (N2.GE.2) CR2DTL=(CR2-CR22L)/(2.*T)
              IF (N1.GE.2) CR1DTL=(CR1-CR12L)/(2.*T)
              IF(FLAG3.EQ.0) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
              IF(FLAG2.EQ.0) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
              IF(FLAG1.EQ.0) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
              IF (N3.EQ.NSW3.AND.FLAG3.EQ.1) GO TO 30
              IF (N2.EQ.NSW2.AND.FLAG2.EQ.1) GO TO 20
              IF (N1.EQ.NSW1.AND.FLAG1.EQ.1) GO TO 10
              IF (FLAG3.EQ.1) C3DT=(2.*((CR3-CR3LST)/T))-CR3DTL
              IF (FLAG2.EQ.1) C2DT=(2.*((CR2-CR2LST)/T))-CR2DTL
              IF (FLAG1.EQ.1) C1DT=(2.*((CR1-CR1LST)/T))-CR1DTL
30            N3=N3+1
20            N2=N2+1
10            N1=N1+1
```

```
          CR3DTL=C3DT
          CR2DTL=C2DT
          CR1DTL=C1DT
          CR32L=CR3LST
          CR22L=CR2LST
          CR12L=CR1LST
          CR3LST=CR3
          CR2LST=CR2
          CR1LST=CR1
SORT


*************************************************************

TERMINAL
METHOD RKSFX
CONTRL FINTIM=0.5,DELT=0.00005,DELS=0.00025
SAVE (S1) 0.0004,X1DOT,X2DOT,X3DOT,C1DT,C2DT,C3DT,...
      CR1DT,CR2DT,CR3DT,CR1,CR2,CR3
SAVE (S2) 0.001,C1,CR1,RR1,C2,CR2,RR2,C3,CR3,RR3
GRAPH (G1/S1,DE=TEK618,PO=1)...
      CR1(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C1DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR1DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X1DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G2/S1,DE=TEK618,OV,PO=1,3.25)...
      CR2(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C2DT(LE=3,NI=4,LO=-3,SC=3,UN='RAD/SEC'),...
      CR2DT(LE=3,NI=4,LO=-3,PO=6.0,SC=3,UN='RAD/SEC'),...
      X2DOT(LE=3,NI=4,LO=-3,SC=3,AX=OMIT)
GRAPH (G3/S1,DE=TEK618,OV,PO=1,6.5)...
      CR3(LE=6.0,NI=13,LO=-.1,SC=.1,UN='RAD'),...
      C3DT(LE=3,NI=4,LO=-8,SC=8,UN='RAD/SEC'),...
      CR3DT(LE=3,NI=4,LO=-8,PO=6.0,SC=8,UN='RAD/SEC'),...
      X3DOT(LE=3,NI=4,LO=-8,SC=8,AX=OMIT)
GRAPH (G4/S2,DE=TEK618,PO=1) TIME(LE=6.0,NI=5,UN='SEC'),...
      C1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR1(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR1(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G5/S2,DE=TEK618,OV,PO=1,3.25)...
      TIME(LE=6.0,NI=5,UN='SEC'),...
      C2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR2(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR2(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
GRAPH (G6/S2,DE=TEK618,OV,PO=1,6.5)...
      TIME(LE=6.0,NI=5,UN='SEC'),...
      C3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4,PO=6.0),...
      CR3(LE=3,NI=4,LO=-.2,UN='RAD',SC=0.4),...
      RR3(LE=3,NI=4,LO=-.2,SC=0.4,AX=OMIT)
LABEL (G1) PHASE PLANE (CDT,CRDT,XDOT.VS.CR)
LABEL (G4) STEP RESPONSE
END
STOP
```

LIST OF REFERENCES

1. Wikstrom, R. K., *An Adaptive Model Based Disc File Head Positioning Servo System.* Master's Thesis, Naval Postgraduate School, Monterey, California, September 1985.

2. Ozaslan, K., *The Near - Minimum Time Control of a Robot Arm.* Master's Thesis, Naval Postgraduate School, Monterey, California, December 1986.

3. Suza, R. P., *Real Time Programming of a Robot,* Master's Thesis, Naval Postgraduate School, Monterey, California, December 1986.

4. Thaler, G. J. and Stein, W. A, "Transfer Function and Parameters Evaluation for D-C Servomotors", *Applications and Industry,* pp. 410-417, January 1956.

5. Thaler, G. J., *Nonlinear Controls Theory, and Methods for Analysis and Design,* unpublished notes for EC 4350, Nonlinear Controls, Naval Postgraduate School, Monterey, California, October 1987.

6. Luh, J. Y., Fisher, W. D., and Paul, R. C.,"Joint Torque Control by a Direct Feedback for Industrial Robots", *IEEE Transactions on Automatic Control,* V. AC-28, No.2, pp. 153-161, February 1983.

7. Critchlow, Arthur. I., *Introduction to Robotics,* pp. 63-64, Macmillan, NY, 1985.

8. Thaler, G. J. and Martin, H. Dost., *Modelling and Simulation of Dynamic Systems,* unpublished notes for EC 4370, Modelling and Simulation for Control Systems, Naval Postgraduate School, Monterey, California, July 1987.

9. Sweet, L. M. and Good, M. C., "Redefinition of the Robot Motion-Control Problem", *IEEE Contr. Syst. Mag.,* pp. 18-25, August 1985.

10. Paul, R. P., *Robot Manipulators : Mathematics, Programming, and Control,* the MIT press, pp. 161-162, 1981.

BIBLIOGRAPHY

Asada, H and Slotine, J.- J. E., Robot Analysis and Control, Wiley, 1984.

Brady, M., Hollerbach, J. M., Johnson, T., Lozano-Perez, T. and Mason, M. T., Robot Motion, the MIT press, 1982.

Engelberger, J. F., Robotics in Practice, Ama, 1980.

Scott, P. B., The Robotics Revolution : The Complete Guide, NY, 1986.

Wolovich, W. A., Robotics : Basic Analysis and Design, Holt, Rinehart and Winston, 1985.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center   2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 0142   2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Department Chairman, Code 62   1
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943

4. Professor G. J. Thaler, Code 62Tr   5
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943

5. Professor H. A. Titus, Code 62Ti   1
   Department of Electrical & Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943

6. Hellenic Navy General Staff   4
   2nd Branch, Education Department
   Stratopedon Papagou
   GR 155.61 - Holargos, GREECE

7. LT George Kalogiros H.N   5
   Tirtaiou 13 - A. Kalamaki
   GR 174.56 - Athens, GREECE

8. Commander   1
   Naval Surface Weapons Center White Oak
   Silver Springs, Maryland 20910

9. Commander   1
   Naval Weapons Center
   China Lake, California 93555

10. Chief of Naval Operations   1
    Attn: Code OP-03
    Washington, D.C. 20362