

# CONTROL SYSTEMS FOR A ROBOT GRIPPER

by

Jonathon C. Kelm

I hereby release the rights and copyright in this thesis to the University of Youngstown, which shall be made available from the OhioLINK ETD Center and the Mang Library Circulation Desk for public access. I also authorize the University or other persons to make copies of this thesis as needed for scholarly research.

Submitted in Partial Fulfillment for the Requirements

For the Degree of

Master of Science in Engineering

in the

Electrical Engineering

Program

Signature:

  
Jonathon C. Kelm, Student

8/3/05  
Date


Approvals:

  
Dr. Faramarz Mousayebi, Thesis Advisor

8/19/05  
Date

  
Dr. Robert H. Foulkes, Committee Member

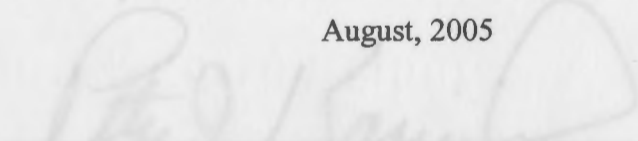
8-15-05  
Date

  
Dr. Philip C. Munro, Committee Member

8/19/05  
Date

YOUNGSTOWN STATE UNIVERSITY

August, 2005

  
Dr. Peter J. Kogutsky, Dean of Graduate Studies

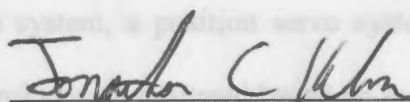
8/18/05  
Date

## CONTROL SYSTEMS FOR A ROBOT GRIPPER

Jonathon C. Kelm

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

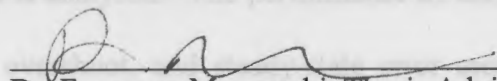


Jonathon C. Kelm, Student

8/3/05

Date

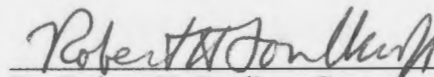
Approvals:



Dr. Faramarz Mossayebi, Thesis Advisor

8/9/05

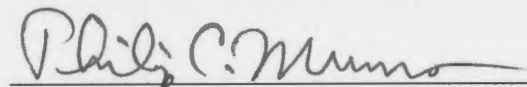
Date



Dr. Robert H. Foulkes, Committee Member

8-15-05

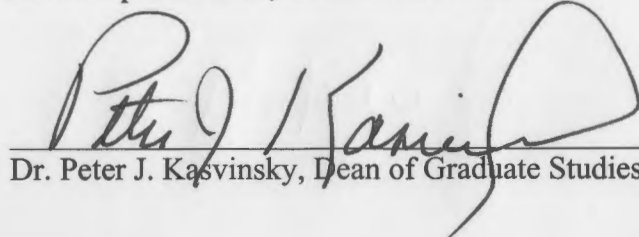
Date



Dr. Philip C. Munro, Committee Member

8/9/05

Date



Dr. Peter J. Kasvinsky, Dean of Graduate Studies

8/18/05

Date

**ABSTRACT**

The fundamental issues in modeling and “executive level” control of a three-link, three-joint robotic arm-gripper system are presented. First, a mathematical model of the mechanism is developed. This model is coupled with a model of a DC permanent magnet motor, as an actuator, to obtain an overall system model. This model is used to design and simulate five local control systems for the independent control of the joints in the arm portion of the arm-gripper system. The control systems developed are a simple feedback system, a position servo system, a position servo system with direct gravity compensation, a state-feedback control system designed based on pole placement, and a PID controller. The effect of gravity moment upon the system response is examined, and the performance of each control system is analyzed. The performance of each controller is quantified by the settling time, percent overshoot, and steady-state error metrics. The position servo system with gravity compensator is found to be the best overall control system approach.

## ACKNOWLEDGEMENTS

It is with sincere gratitude that I acknowledge those who have been supportive of me in my academic endeavors. First, I thank God for giving me the talents and abilities required of me, and for the circumstances in which to use them. My parents, Donald and Janet Kelm, have always been available to offer emotional support and to help me through difficult times. I am grateful to Ashley King for her love and encouragement, and for forever giving me a reason to strive to succeed. Of course, I must thank my advisor, Dr. Faramarz Mossayebi, for his expertise and for his patience with me. I sincerely appreciate the input of my committee members, Dr. Robert Foulkes and Dr. Philip Munro. Finally, I thank all of the other members of the faculty and staff in the Electrical and Computer Engineering department, who have always encouraged me and treated me with respect.

1.1	A Simple Robotic Arm and Gripper System .....	3
1.2	Modeling the Arm Mechanism .....	5
1.3	Modeling the Actuators .....	6
1.4	State-Space Model of Combined Actuator and Joint .....	13
1.5	Combined Actuator and Joint Transfer Function .....	15
1.6	Modeling Gravity Effects .....	18
II.	SYSTEM CHARACTERISTICS .....	21
2.1	Desired System Characteristics .....	21
2.2	Simple Feedback System .....	24
III.	POSITION SERVO SYSTEMS .....	31
3.1	Basic Position Servo System .....	31

	PAGE
1.2    Servo System (TABLE OF CONTENTS) .....	43
IV. STATE SPACE AND PID CONTROL .....	50
4.1    State-Feedback Control Using Pole Placement .....	50
ABSTRACT .....	iii
ACKNOWLEDGEMENTS .....	iv
TABLE OF CONTENTS .....	v
LIST OF SYMBOLS .....	viii
LIST OF FIGURES .....	xi
LIST OF TABLES .....	xiii
INTRODUCTION .....	1
<b>CHAPTER</b> Overall Comparison of Control System Responses .....	69
<b>VI. MODELING THE ARM MECHANISM</b> .....	3
1.1    A Simple Robotic Arm and Gripper System .....	3
1.2    Modeling the Arm Mechanism .....	5
1.3    Modeling the Actuators .....	9
1.4    State-Space Model of Combined Actuator and Joint .....	13
1.5    Combined Actuator and Joint Transfer Function .....	15
1.6    Modeling Gravity Effects .....	18
<b>II. SYSTEM CHARACTERISTICS</b> .....	22
2.1    Desired System Characteristics .....	22
2.2    Simple Feedback System .....	24
<b>III. POSITION SERVO SYSTEMS</b> .....	33
3.1    Basic Position Servo System .....	33

3.2	Servo System with Gravity Compensation .....	45
IV.	STATE-SPACE AND PID CONTROL .....	50
4.1	State-Feedback Control Using Pole-Placement .....	50
4.2	Local PID Controller .....	60
V.	ANALYSIS .....	67
5.1	Simple Feedback Systems Analyses .....	67
5.2	Position Servo Systems Analyses .....	68
5.3	Position Servo with Gravity Compensation Systems Analyses .....	68
5.4	State-Feedback Systems Analyses .....	68
5.5	Local PID Control Systems Analyses .....	60
5.6	Overall Comparison of Control System Responses .....	69
VI.	CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK ....	71
6.1	Joint One .....	71
6.2	Joint Two .....	71
6.3	Joint Three .....	72
6.4	Suggestions for Further Work .....	72
APPENDIX A:	INACCURACIES IN THE MODELS .....	74
A.1	Arm Mechanism Model Inaccuracies .....	74
A.2	Actuator Model Inaccuracies .....	75
APPENDIX B:	SIMULTANEOUS CONTROL OF JOINTS .....	77
B.1	Effect of Inertia Variation .....	77
B.2	Effect of Gravity Moment Variation .....	78
B.3	Effect of Centrifugal Force .....	78

B.4	Modeling the Simultaneous Movement of Joints .....	78
APPENDIX C: C PROGRAM FOR FINDING MAXIMUM $H$ AND $G$ .....		79
APPENDIX D: C PROGRAMS FOR CALCULATING $G$ AND $\Delta\theta$ .....		84
SYMBOL DEFINITION		
D.1	Calculating $G_1$ for $q_2 = q_3 = \pi/4$ rad .....	84
D.2	Calculating $G_2$ for $q_1 = q_3 = \pi/4$ rad .....	85
D.3	Calculating $G_3$ for $q_1 = q_2 = \pi/4$ rad .....	86
APPENDIX E: END-EFFECTOR DESIGN AND CONTROL .....		87
E.1	Important Literature .....	87
E.2	Design of the Hand .....	89
E.3	Grasping vs. Manipulating .....	87
E.4	Dexterity of Manipulation .....	88
E.5	Manipulability .....	89
E.6	Dynamics of Pinching .....	89
REFERENCES .....		90

$F_v$	Actuator viscous damping constant
$G(z)$	Gravitational moment
$G(\infty)$	Steady-state value of the gravity moment
$g$	Acceleration due to gravity (approximately $9.807 \text{ m/s}^2$ )
$G$	Combined actuator and joint transfer function
$G^2$	Second-order combined actuator and joint transfer function
$G^3$	Third-order combined actuator and joint transfer function
$J(z)$	Moment of inertia
$I$	Identity matrix

## LIST OF SYMBOLS

SYMBOL	DEFINITION
$A_a$	Actuator system matrix
$\hat{A}$	Joint system matrix
$\tilde{A}$	Augmented joint system matrix
$B_a$	Actuator system matrix
$\hat{B}$	Joint system matrix
$\tilde{B}$	Augmented joint system matrix
$C$	System output matrix
$C_E$	Actuator electromotive force constant (V/rad/s)
$C_M$	Actuator torque constant (N·m/A)
$E(\infty)$	Steady-state error
$F_v$	Actuator viscous damping constant
$G(Q)$	Gravitational moment
$G(\infty)$	Steady-state value of the gravity moment
$g$	Acceleration due to gravity (approximately 9.807 m/s <sup>2</sup> )
$\mathcal{G}$	Combined actuator and joint transfer function
$\mathcal{G}^{\text{II}}$	Second-order combined actuator and joint transfer function
$\mathcal{G}^{\text{III}}$	Third-order combined actuator and joint transfer function
$H(Q)$	Moment of inertia
$I$	Identity matrix



$i_R$	Actuator rotor current (A)
$J_M$	Actuator rotor moment of inertia ( $\text{kg}\cdot\text{m}^2$ )
$\mathbf{K}$	Feedback gain matrix
$K_{ff}$	Feedforward gain
$K_I$	Current feedback gain
$K_{in}$	Feedback gain for integral of error signal
$K_P$	Position amplifier gain
$K_v$	Velocity feedback gain
$L_R$	Actuator rotor winding inductance (H)
$M$	Load torque at actuator output shaft
$m_G$	Mass of the gripper mechanism (kg)
$N_M$	Actuator torque reduction ratio
$N_v$	Actuator speed reduction ratio
OS	Overshoot
$P$	Driving moment about a joint
$q$	Joint coordinate (actual position in radians)
$q_o$	Joint coordinate (desired position in radians)
$q(\infty)$	Joint coordinate at steady-state
$R_R$	Actuator rotor resistance ( $\Omega$ )
$r_G$	Radius of the gripper mechanism (m)
$r(t)$	System response
$s$	Laplace variable
$T_s$	Settling time

LIST OF FIGURES

$u$	Actuator rotor control voltage (V)	
$V_o$	Voltage corresponding to desired joint position (V)	
<b>FIGURE</b>		<b>PAGE</b>
$V_q$	Voltage corresponding to actual joint position (V)	
1.1	A Simple 3-Link, 3-Joint Robot Arm and Gears	4
$\mathbf{x}_a$	Actuator state vector	
1.2	Schematic Diagram of Permanent Magnet Motor	10
$\hat{\mathbf{x}}$	Joint state vector	
2.1	Simple Feedback System	25
$\tilde{\mathbf{x}}$	Augmented joint state vector	
2.2	Joint 1 Simple Feedback System Response, Neglecting Gravity	27
$y$	System output	
2.3	Joint 2 Simple Feedback System Response, Neglecting Gravity	28
$\zeta$	Damping ratio	
2.4	Joint 3 Simple Feedback System Response, Neglecting Gravity	29
$\lambda$	System matrix eigenvalue	
2.5	Joint 1 Simple Feedback System Response, Considering Gravity	30
$\theta$	Angular displacement (rad)	
2.6	Joint 2 Simple Feedback System Response, Considering Gravity	31
$\dot{\theta}$	Angular velocity (rad/s)	
2.7	Joint 3 Simple Feedback System Response, Considering Gravity	32
$\tau_E$	Electrical time constant	34
$\tau_M$	Mechanical time constant	39
$\omega_n$	Natural frequency (rad/s)	40
3.4	Joint 3 Position Servo Response, Neglecting Gravity	41
3.5	Joint 1 Position Servo Response, Considering Gravity	42
3.6	Joint 2 Position Servo Response, Considering Gravity	43
3.7	Joint 3 Position Servo Response, Considering Gravity	44
3.8	Servo System with Gravity Compensation	46
3.9	Joint 1 Servo with Gravity Compensation Response	47
3.10	Joint 2 Servo with Gravity Compensation Response	48
3.11	Joint 3 Servo with Gravity Compensation Response	49

<b>LIST OF FIGURES</b>		
<b>FIGURE</b>		<b>PAGE</b>
1.1	A Simple 3-Link, 3-Joint Robot Arm and Gripper .....	4
1.2	Schematic Diagram of Permanent Magnet Motor .....	10
2.1	Simple Feedback System .....	26
2.2	Joint 1 Simple Feedback System Response, Neglecting Gravity ...	27
2.3	Joint 2 Simple Feedback System Response, Neglecting Gravity ...	28
2.4	Joint 3 Simple Feedback System Response, Neglecting Gravity ...	29
2.5	Joint 1 Simple Feedback System Response, Considering Gravity .	30
2.6	Joint 2 Simple Feedback System Response, Considering Gravity .	31
2.7	Joint 3 Simple Feedback System Response, Considering Gravity .	32
3.1	Position Servo System .....	34
3.2	Joint 1 Position Servo Response, Neglecting Gravity .....	39
3.3	Joint 2 Position Servo Response, Neglecting Gravity .....	40
3.4	Joint 3 Position Servo Response, Neglecting Gravity .....	41
3.5	Joint 1 Position Servo Response, Considering Gravity .....	42
3.6	Joint 2 Position Servo Response, Considering Gravity .....	43
3.7	Joint 3 Position Servo Response, Considering Gravity .....	44
3.8	Servo System with Gravity Compensation .....	46
3.9	Joint 1 Servo with Gravity Compensation Response .....	47
3.10	Joint 2 Servo with Gravity Compensation Response .....	48
3.11	Joint 3 Servo with Gravity Compensation Response .....	49

4.1	State-Feedback System .....	51
4.2	Joint 1 State-Feedback System Response, Neglecting Gravity .....	54
4.3	Joint 2 State-Feedback System Response, Neglecting Gravity .....	55
4.4	Joint 3 State-Feedback System Response, Neglecting Gravity .....	56
4.5	Joint 1 State-Feedback System Response, Considering Gravity ....	57
4.6	Joint 2 State-Feedback System Response, Considering Gravity ....	58
4.7	Joint 3 State-Feedback System Response, Considering Gravity ....	59
4.8	PID Controller .....	61
4.9	Joint 1 PID Controller System Response .....	64
4.10	Joint 2 PID Controller System Response .....	65
4.11	Joint 3 PID Controller System Response .....	66
4.2	Calculated PID Controller Gain Values .....	63
5.1	Summary of Control System Response Characteristics .....	70

## LIST OF TABLES

TABLE		PAGE
	INTRODUCTION	
1.1	Arm Mechanism Parameters .....	5
1.2	Maximum $H$ and $G$ Values by Joint .....	9
1.3	Actuator Mechanism Parameters .....	12
1.4	Joint Transfer Function Parameters .....	17
1.5	Approximate Effect of Gravity for Joint One, Others at $\pi/4$ rad ...	19
1.6	Approximate Effect of Gravity for Joint Two, Others at $\pi/4$ rad ..	20
1.7	Approximate Effect of Gravity for Joint Three, Others at $\pi/4$ rad	21
3.1	Calculated Simple Servo System Gains .....	38
4.1	Calculated State-Space Gain Values .....	53
4.2	Calculated PID Controller Gain Values .....	63
5.1	Summary of Control System Response Characteristics .....	70

A basic robotic manipulator consists of two somewhat independent parts: an "arm" and an "end-effector" or "gripper." The end-effector is the part of the mechanism that directly manipulates objects; this may be a human-like hand, a surgical tool, a paint sprayer, etc. The arm is the part that moves the end-effector to the proper position.

A layered approach is usually taken to the design of control systems for robot manipulators [1]. When a robot manipulator is given a task, the completion of the task requires interaction between several "levels" of control. At the highest level, some form of artificial intelligence involving computer vision, machine learning, etc., may be used to

## INTRODUCTION

Robotic manipulators have a wide range of applications, including manufacturing, medicine, and space exploration. Robots are able to perform many tasks to a higher degree of precision than humans, and they are often able to work in much more extreme environments.

The design and control of robotic manipulators is a complex topic, requiring knowledge of mechanics, control theory, and computer science. The mechanical properties

of a manipulator affect its speed and precision. Knowledge of control theory is required to

design a control system for a robot that is not only accurate, but also reliable. The control system often requires the use of a digital programmable computer or microcontroller. The

requirements placed on a robot manipulator system vary widely from application to

application. In manufacturing, for example, high speed and low cost may be desired, while medical applications may sacrifice speed and cost for precision.

A basic robotic manipulator may consist of two somewhat independent parts: an "arm" and an "end-effector" or "gripper." The end-effector is the part of the mechanism that directly manipulates objects; this may be a human-like hand, a surgical tool, a paint sprayer, etc. The arm is the part that moves the end-effector to the proper position.

A layered approach is normally taken to the design of control systems for robot manipulators [1]. When a robot manipulator is given a task, the completion of the task requires interaction between several "levels" of control. At the highest level, some form of artificial intelligence involving computer vision, machine learning, etc., may be used to

develop a strategy for completing a task. This task controls a so-called “tactical control level,” normally a digital system such as a microcontroller or computer system, which determines the desired configuration or trajectory for the manipulator based on the needs for carrying out the task. This desired configuration or trajectory is sent to an “executive control level” in the form of internal coordinates of the mechanism. The job of the executive control level is to move the manipulator to the desired position or along the desired trajectory specified by the tactical control level. An important text on kinematics and control of robot manipulators at the tactical level is Mason and Salisbury [2]. This work is concerned with the control of a robotic arm-gripper system at the executive level. Control at higher levels is left for further study.

The arm portion of the system considered in this work consists of three rigid links connected by joints, with each joint being actuated by a permanent-magnet DC motor. The end-effector has a similar design, consisting of two fingers, each of which is made of two rigid links connected by joints.

This thesis is organized as follows. A mathematical model of the arm portion of the mechanism is developed in Chapter 1. Chapter 2 describes the system characteristics and presents simple feedback control approaches for each joint in the arm. In Chapter 3, position servo systems are developed, including approaches with and without compensation for gravity. Chapter 4 describes a state-feedback and a PID controller. Finally, Chapters 5 and 6 present analyses and conclusions, respectively. Additional information that may be useful for further work is presented in the appendices.

## CHAPTER I

### MODELING THE ARM MECHANISM

This chapter presents a simple three-link robotic arm with three joints, each driven by a permanent-magnet DC motor. First, the design of the arm is discussed. Then, a mathematical model of the arm is presented. This model is then coupled with a model of a permanent-magnet DC motor (the actuator) to produce an overall system model.

#### 1.1 A Simple Robotic Arm and Gripper System

Figure 1.1 shows the robotic arm under consideration. The mechanism consists of three rigid links of lengths  $l_A$ ,  $l_B$ , and  $l_C$ , and masses  $m_A$ ,  $m_B$ , and  $m_C$ . The links are connected by joints. The mass of joint 1 does not affect the model and is hence ignored; the other two joints have masses  $m_2$  and  $m_3$ . Control of the fingers in the gripper mechanism can be handled using methods very similar to those for controlling the arm portion, and hence for simplicity the gripper is reduced to a circle of radius  $r_G$  and mass  $m_G$ . Control of the gripper is left for further study, and may be based largely on the work presented in this paper. Appendix E presents some information related to the design and control of the hand.

Table 1.1 gives the numerical values for the arm parameters used for this study.

The configuration of the arm at any point in time is specified by the vector  $Q = [q_1 \ q_2 \ q_3]$ , where  $q_1$  is the angle between link  $A$  and the vertical,  $q_2$  is the angle between link  $A$  and link  $B$ , and  $q_3$  is the angle between link  $B$  and link  $C$ .



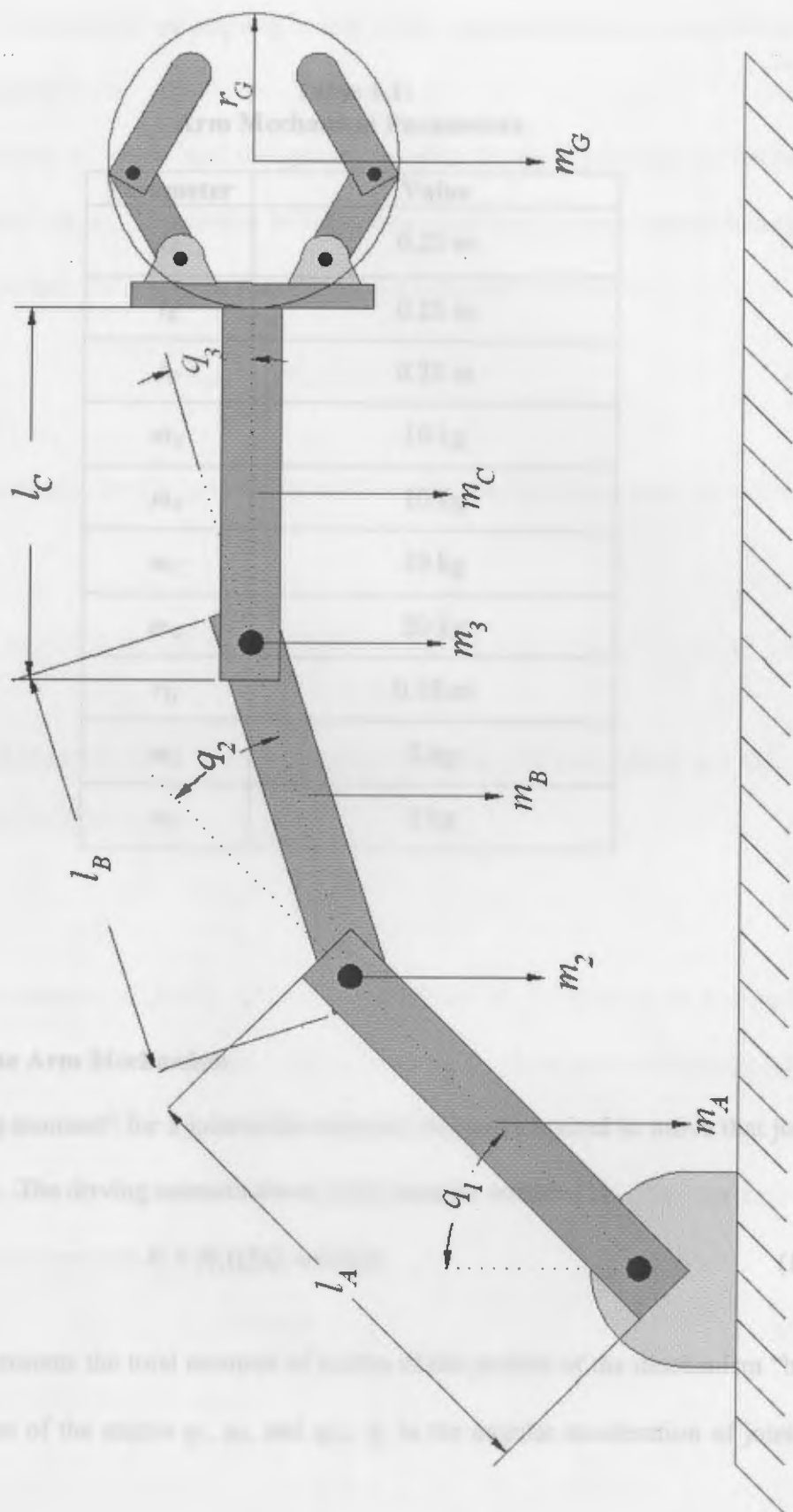


Figure 1.1: A Simple 3-Link, 3-Joint Robot Arm and Gripper

**Table 1.1:**  
**Arm Mechanism Parameters.**

Parameter	Value
$l_A$	0.25 m
$l_B$	0.25 m
$l_C$	0.25 m
$m_A$	10 kg
$m_B$	10 kg
$m_C$	10 kg
$m_G$	20 kg
$r_G$	0.10 m
$m_2$	2 kg
$m_3$	2 kg

## 1.2 Modeling the Arm Mechanism

The “driving moment” for a joint is the moment (torque) required to move that joint to a desired position. The driving moment about joint  $i$  may be written [1]:

$$P_i = H_i(Q)\ddot{q}_i + G_i(Q) \quad (1.1)$$

where  $H_i(Q)$  represents the total moment of inertia of the portion of the mechanism “behind” joint  $i$  (a function of the angles  $q_1$ ,  $q_2$ , and  $q_3$ ),  $\ddot{q}_i$  is the angular acceleration of joint  $i$  (i.e.,

the second time-derivative of angle  $q_i$ ), and  $G_i(Q)$  represents the moment about joint  $i$  contributed by gravity.

The moment of inertia and the gravity moment for each joint can be written using basic kinematics [1, 3, 4]. If each link in the arm is considered to be a slender homogeneous rod pinned at one end, the moment of inertia for the link may be written:

$$H_{\text{link}} = \frac{1}{3} m_{\text{link}} l_{\text{link}}^2. \quad (1.2)$$

The moment of inertia for the gripper, simplified to a circular disk, may be written using equation 1.3:

$$H_{\text{joint}} = \frac{3}{2} m_G r_G^2. \quad (1.3)$$

The parallel axis theorem states that the moment of inertia of a body about any axis parallel to the mass center is given by:

$$H = H_G + md^2 \quad (1.4)$$

where  $H_G$  is the moment of inertia about the mass center of the body,  $m$  is the mass of the body, and  $d$  is the perpendicular distance between the mass center axis and the parallel axis.

Gravity induces a moment (torque) about each joint equal to the sum of the moments about the joint caused by the individual components of the mechanism. The moment about a joint caused by a component of the mechanism is:

$$G = mgd \quad (1.5)$$

$$H_i(Q) = \left(\frac{1}{3} m_1 l_1^2\right) + \left(\frac{1}{3} m_2 l_2^2\right) + \left(\frac{3}{2} m_3 r_3^2 + m_3 (l_1^2 + l_2^2 + 2l_1 l_2 \cos q_1)\right) \quad (1.9)$$

where  $m$  is the mass of the mechanism component,  $g$  is acceleration due to gravity, and  $d$  is the perpendicular distance between the joint axis and the mass center of the mechanism component.

The moments of inertia about each joint can be determined by combining equations 1.2, 1.3, and 1.4. The gravity moments for each joint can be written using equation 1.5.

For joint one the moment of inertia is:

$$H_1(Q) = \left( \frac{1}{3} m_A l_A^2 \right) + \left( \frac{1}{3} m_B l_B^2 + m_B l_A^2 \right) + \left( \frac{3}{2} m_G r_G^2 + m_G d_1^2 \right) + \left( \frac{1}{3} m_C l_C^2 + m_C (l_A^2 + l_B^2 + 2l_A l_B \cos q_2) \right) \quad (1.6)$$

where  $d_1$  is the perpendicular distance from the center of joint 1 to the endpoint of link C, found from:

$$d_1^2 = \left( l_A + \frac{l_B \sin q_3}{\sin(q_2 + q_3)} \right)^2 + \left( l_B + \frac{l_B \sin q_2}{\sin(q_2 + q_3)} \right)^2 + 2 \left( l_A + \frac{l_B \sin q_3}{\sin(q_2 + q_3)} \right) \left( l_B + \frac{l_B \sin q_2}{\sin(q_2 + q_3)} \right) \cos(q_2 + q_3) \quad (1.7)$$

The gravity moment for joint one may be written:

$$G_1(Q) = \left( \frac{1}{2} m_A g l_A + m_2 g l_A + m_B g l_A + m_3 g l_A + m_C g l_A + m_G g l_A \right) \sin q_1 + \left( \frac{1}{2} m_B g l_B + m_B g l_B + m_C g l_B + m_G g l_B \right) \cos(q_1 + q_2) + \left( \frac{1}{2} m_C g l_C + m_G g (l_C + r_G) \right) \cos(q_1 + q_2 + q_3) \quad (1.8)$$

For joint two, the moment of inertia is:

$$H_2(Q) = \left( \frac{1}{3} m_B l_B^2 \right) + \left( \frac{1}{3} m_C l_C^2 \right) + \left( \frac{3}{2} m_G r_G^2 + m_G (l_B^2 + l_C^2 + 2l_B l_C \cos q_3) \right) \quad (1.9)$$

and the gravity moment is:

$$G_2(Q) = \left( -\frac{1}{2} m_A g l_A \right) \cos q_1 + \left( \frac{1}{2} m_B g l_B + m_B g l_B + m_C g l_B + m_G g l_B \right) \cos(q_1 + q_2) + \left( \frac{1}{2} m_C g l_C + m_G g (l_C + r_G) \right) \cos(q_1 + q_2 + q_3) \quad (1.10)$$

Finally, the moment of inertia for joint three is:

$$H_3(Q) = \left( \frac{1}{3} m_C l_C^2 \right) + \left( \frac{3}{2} m_G r_G^2 + m_G l_C^2 \right) \quad (1.11)$$

### 1.3 Modeling the Actuators

and the gravity moment about joint three is:

$$G_3(Q) = \left( -\frac{1}{2} m_A g l_A \right) \sin q_1 + \left( -m_A g l_B - m_2 g l_B - \frac{1}{2} m_B g l_B \right) \sin(q_1 + q_2) + \left( \frac{1}{2} m_C g l_C + m_G g (l_C + r_G) \right) \cos(q_3) \quad (1.12)$$

The maximum values of moment of inertia and gravity moment for each joint will be of interest when designing the joint control systems, since they represent the largest values the control systems must be able to handle. These maximum values were determined by using a computer program (see Appendix C), using the arm parameters as in Table 1.1, and are summarized in Table 1.2.

The equation of mechanical equilibrium can be written using the Law of Conservation of Energy:

$$H_1 \dot{q}_1 + H_2 \dot{q}_2 + H_3 \dot{q}_3 + F_1 \frac{dq_1}{dt} + F_2 \frac{dq_2}{dt} + F_3 \frac{dq_3}{dt} = 0 \quad (1.13)$$

**Table 1.2:**  
**Maximum  $H$  and  $G$  Values by Joint.**

Parameter	Joint 1	Joint 2	Joint 3
$H$ (kg·m <sup>2</sup> )	16	6	2
$G$ (N·m)	285	115	80

### 1.3 Modeling the Actuators

Each joint is driven independently by a single actuator. The actuators are assumed to be permanent magnet DC motors directly connected to their respective joints. The model of a permanent magnet DC motor is well-known and described in the literature [1, 5, 6].

Figure 1.2 shows a schematic diagram of the actuator. The mathematical model is developed by writing the differential equations associated with the electrical and mechanical equilibrium for the motor.

The equation of electrical equilibrium can be written using Kirchoff's Voltage Law, summing the voltages in the loop:

$$i_R R_R + L_R \frac{di_R}{dt} + C_E N_v \frac{d\theta}{dt} - u = 0. \quad (1.13)$$

The equation of mechanical equilibrium can be written using the Law of Conservation of Energy:

$$N_v N_M J_M \frac{d^2\theta}{dt^2} + F_v \frac{d\theta}{dt} + M - C_M N_M i_R = 0. \quad (1.14)$$

To develop a model of the actuator in state-space, let the state vector of the actuator

$$x_a = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ i_p \end{bmatrix} \quad (1.15)$$

Solving equation 1.13 for  $di_p/dt$  in terms of the state variables in (1.15) gives

$$\frac{di_p}{dt} = -\frac{C_p N_p}{L_p} x_1 - \frac{R_p}{L_p} x_2 + \frac{1}{L_p} v \quad (1.16)$$

Similarly, solving equation 1.14 for

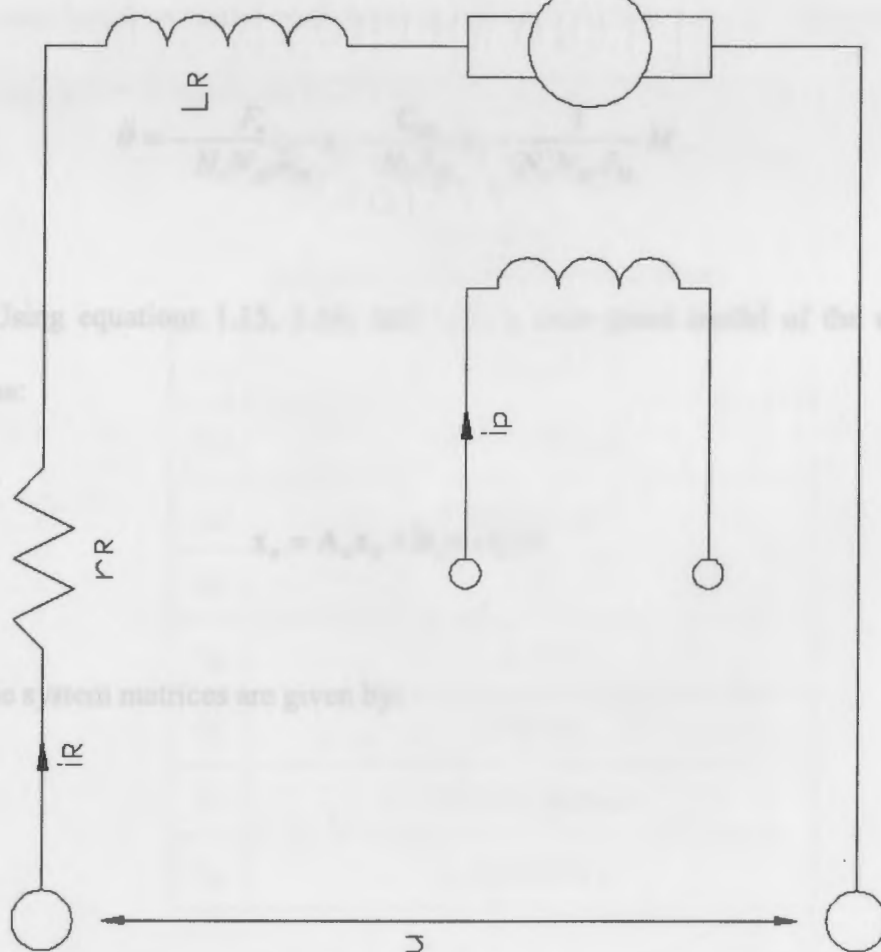


Figure 1.2: Schematic Diagram of Permanent-Magnet Motor.

Using equations 1.15, 1.16, and

written as:

$$\dot{x}_a = A_a x_a + B_a v \quad (1.18)$$

where the system matrices are given by:

To develop a model of the actuator in state-space, let the state vector of the actuator be

$$\mathbf{x}_a = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ i_R \end{bmatrix}. \quad (1.15)$$

Solving equation 1.13 for  $di_R/dt$  in terms of the state variables in 1.15 gives:

$$\frac{di_R}{dt} = -\frac{C_E N_v}{L_R} x_2 - \frac{R_R}{L_R} x_3 + \frac{1}{L_R} u. \quad (1.16)$$

Similarly, solving equation 1.14 for  $\ddot{\theta} = \frac{d^2\theta}{dt^2}$  gives:

$$\ddot{\theta} = -\frac{F_v}{N_v N_M J_M} x_2 - \frac{C_M}{N_v J_M} x_3 - \frac{1}{N_v N_M J_M} M. \quad (1.17)$$

Using equations 1.15, 1.16, and 1.17, a state-space model of the actuator may be

written as:

$$\dot{\mathbf{x}}_a = \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a u + \mathbf{f}_a M \quad (1.18)$$

where the system matrices are given by:

$C_E$	0.0025 V/rad/s
$C_M$	0.0050 N·m/A
$J_M$	0.0001 kg·m <sup>2</sup>
$N_v$	31.17
$N_M$	31.17
$R_R$	1.60 Ω
$F_v$	0.0025 N·m/rad/s
$L_R$	0.0025 H



$$\mathbf{A}_a = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & -F_v/N_v N_M J_M & C_M/N_v J_M \\ 0 & 0 & -C_E N_v/L_R & -R_R/L_R \end{bmatrix} \quad (1.19)$$

$$\mathbf{B}_a = \begin{bmatrix} 0 \\ 0 \\ 1/L_R \end{bmatrix} \quad (1.20)$$

$$\mathbf{f}_a = \begin{bmatrix} 0 \\ -1/N_v N_M J_M \\ 0 \end{bmatrix} \quad (1.21)$$

1.4.3 For a numerical model, the actuator parameters in Table 1.3 were used. These values were chosen based on values used in examples by Vukobratović [1]. The resulting numerical matrices are given in equations 1.22-1.24.

**Table 1.3:**  
**Actuator Mechanism Parameters.**

$C_E$	0.0459 V/rad/s
$C_M$	0.0480 N·m/A
$J_M$	0.00003 kg·m <sup>2</sup>
$N_v$	31.17
$N_M$	31.17
$R_R$	1.60 Ω
$F_v$	0.0058 N·m/rad/s
$L_R$	0.0023 H

From equations 1.18, 1.25, 1.26, and 1.27, the state model for a joint and its actuator may be written:

$$\mathbf{A}_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.198991 & 51.3314 \\ 0 & -622.045 & -695.652 \end{bmatrix} \quad (1.22)$$

In order to develop a model for the combined joint-actuator system, notice that equations 1.25 and 1.27 imply:

$$\mathbf{B}_a = \begin{bmatrix} 0 \\ 0 \\ 434.783 \end{bmatrix} \quad (1.23)$$

where

$$\mathbf{f}_a = \begin{bmatrix} 0 \\ -34.3088 \\ 0 \end{bmatrix} \quad (1.24)$$

#### 1.4 State-Space Model of Combined Actuator and Joint

A state-space model of the system consisting of the mechanical joint and its actuator is developed in this section. It is assumed that the output angle of the actuator is equal to the output angle (coordinate) of the joint to which it is connected. In other words, for each joint  $i$ ,

$$q_i = \theta_i. \quad (1.25)$$

Further, it is required that the load on a joint be equal to the driving moment about the joint (given in equation 1.1):

$$M_i = P_i = H_i(Q)\ddot{q}_i + G_i(Q). \quad (1.26)$$

Let the state vector of the combined joint-actuator system be:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ i_R \end{bmatrix}. \quad (1.27)$$

From equations 1.18, 1.25, 1.26, and 1.27, the total model for a joint and its actuator may be written:

$$\dot{\mathbf{x}} = \mathbf{A}_a \mathbf{x} + \mathbf{B}_a u + \mathbf{f}_a [H_i(Q)\ddot{q}_i + G_i(Q)]. \quad (1.28)$$

In order to develop a model of the combined joint-actuator system, notice that equations 1.25 and 1.27 imply:

$$\dot{q} = \dot{\theta} = \mathbf{T}\mathbf{x} \quad (1.29)$$

where

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0004 & 0.00034 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \mathbf{T} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}. \quad (1.30)$$

Gravity will be treated as an external disturbance upon the system, and hence it is ignored in the system model. Therefore, equation 1.28 may be written [1]:

$$\hat{\mathbf{A}}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0010 & 0 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \hat{\mathbf{B}} = \begin{bmatrix} 0 \\ -14.309 \\ 0 \end{bmatrix} \quad (1.31)$$

$$\dot{\mathbf{x}} = (\mathbf{I} - \mathbf{f}_a H_i(Q)\mathbf{T})^{-1} [\mathbf{A}_a \mathbf{x} + \mathbf{B}_a u] = \hat{\mathbf{A}}\mathbf{x} + \hat{\mathbf{B}}u$$

where

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0001 & 0 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \hat{\mathbf{A}} = (\mathbf{I} - \mathbf{f}_a H_i(Q)\mathbf{T})^{-1} \mathbf{A}_a \quad (1.32)$$

and

$$\hat{\mathbf{B}} = (\mathbf{I} - \mathbf{f}_a H_i(Q)\mathbf{T})^{-1} \mathbf{B}_a. \quad (1.33)$$

The state-space representation given in section 1.4 can be used to derive a transfer function for the electromechanical joint. This transfer function is developed in this section.

Notice from equations 1.22, 1.23, and 1.24 that  $\hat{\mathbf{A}}$  and  $\hat{\mathbf{B}}$  are constant matrices.

Finally, the system output is the position,  $q$ , of the joint; thus, the system output,  $y$ , can be written:

$$y = \mathbf{C}\mathbf{x} \quad (1.34)$$

where

$$\mathbf{C} = [1 \quad 0 \quad 0]. \quad (1.35)$$

For the arm described by the parameters in Tables 1.1 and 1.3, and the maximum values of moment of inertia given in Table 1.2, the following system matrices are obtained using equations 1.32, 1.33, and 1.21:

$$\hat{\mathbf{A}}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0004 & 0.00034 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \hat{\mathbf{B}}_1 = \begin{bmatrix} 0 \\ 0 \\ 434.78 \end{bmatrix} \quad \mathbf{f}_{a,1} = \begin{bmatrix} 0 \\ -34.309 \\ 0 \end{bmatrix} \quad (1.36)$$

$$\hat{\mathbf{A}}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0010 & 0.2482 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \hat{\mathbf{B}}_2 = \begin{bmatrix} 0 \\ 0 \\ 434.78 \end{bmatrix} \quad \mathbf{f}_{a,2} = \begin{bmatrix} 0 \\ -34.309 \\ 0 \end{bmatrix} \quad (1.37)$$

$$\hat{\mathbf{A}}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.0001 & 0.0187 \\ 0 & -622.05 & -695.65 \end{bmatrix} \quad \hat{\mathbf{B}}_3 = \begin{bmatrix} 0 \\ 0 \\ 434.78 \end{bmatrix} \quad \mathbf{f}_{a,3} = \begin{bmatrix} 0 \\ -34.309 \\ 0 \end{bmatrix} \quad (1.38)$$

### 1.5 Combined Actuator and Joint Transfer Function

The state-space representation given in section 1.4 can be used to derive a transfer function for the electromechanical joint. That transfer function is developed in this section.

The system matrices,  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{B}}$ , and  $\mathbf{f}_a$ , are given by equations 1.32, 1.33, and 1.21, respectively. The transfer function  $\mathcal{G}(s)$  of the system can be written [7]:

In the case when the electrical time constant  $\tau_E$  is much less than the mechanical time constant  $\tau_M$ , the terms containing  $\tau_E$  may be neglected and a second-order transfer function

Substituting the appropriate values from the system matrices gives the following expression may be written:

for  $\mathcal{G}(s)$ :

$$\mathcal{G}(s) = \frac{K_D}{C_M C_E N_M N_v + F_v R_R} \left( \frac{L_R (H_i + N_v N_M J_M)}{C_M C_E N_v N_M + F_v R_R} s^2 + \frac{F_v L_R + R_R (H_i + N_v N_M J_M)}{C_M C_E N_v N_M + F_v R_R} s + 1 \right) s \quad (1.40)$$

which can be written [1]:

$$\mathcal{G}^{\text{III}}(s) = \frac{K_D}{\tau_E \tau_M s^3 + (\tau_E F_D + 1) \tau_M s^2 + s} \quad (1.41)$$

where  $K_D$  and  $F_D$  are constants given by

$$K_D = \frac{C_M N_M}{N_M N_v C_M C_E + R_R F_v}, \quad (1.42)$$

Parameter	Joint 1	Joint 2	Joint 3
$F_D$	$F_D = \frac{F_v}{N_M N_v J_M + H_i}$ , 0.00962	0.00962	0.002858
$K_D$	0.69594	0.69594	0.69594
$\tau_E$	0.0014375	0.0014375	0.0014375
$\tau_M$	$\tau_M = L_R / R_R$ , 4.48714	4.48714	1.5102

and

$$\tau_M = \frac{R_R (N_M N_v J_M + H_i)}{N_M N_v C_M C_E + R_R F_v} \quad (1.45)$$

In the case when the electrical time constant  $\tau_E$  is much less than the mechanical time constant  $\tau_M$ , the terms containing  $\tau_E$  may be neglected and a second-order transfer function may be written:

$$G^{\text{II}}(s) = \frac{K_D}{\tau_M s^2 + s} \quad (1.46)$$

The numerical joint transfer functions using the arm parameters in Tables 1.1 and 1.3 and the maximum moment of inertia values from Table 1.2 may be found using equations 1.41 and 1.46 with the values in Table 1.4 calculated for each joint. Equations 1.47, 1.48, and 1.49 show these numerical transfer functions for joints one, two, and three, respectively.

**Table 1.4:**  
**Joint Transfer Function Parameters.**

Parameter	Joint 1	Joint 2	Joint 3
$F_D$	0.000362	0.000962	0.002858
$K_D$	0.69594	0.69594	0.69594
$K_M$	0.744241	0.744241	0.744241
$T_E$	0.0014375	0.0014375	0.0014375
$T_M$	11.92955	4.48714	1.5102

The results are tabulated in Table 1.5 for joint one, Table 1.6 for joint two, and Table 1.7 for joint three. By coordinating the joint positions with the output of each control system in the following chapters, a rudimentary simulation of gravity's effect is performed.

$$\mathcal{G}_1^{\text{III}}(s) = \frac{0.69594}{0.017149s^3 + 11.9296s^2 + s} \quad \mathcal{G}_1^{\text{II}}(s) = \frac{0.69594}{11.92955s^2 + s} \quad (1.47)$$

$$\mathcal{G}_2^{\text{III}}(s) = \frac{0.69594}{0.00645s^3 + 4.48714s^2 + s} \quad \mathcal{G}_2^{\text{II}}(s) = \frac{0.69594}{4.48714s^2 + s} \quad (1.48)$$

$$\mathcal{G}_3^{\text{III}}(s) = \frac{0.69594}{0.002171s^3 + 1.51026s^2 + s} \quad \mathcal{G}_3^{\text{II}}(s) = \frac{0.69594}{1.5102s^2 + s} \quad (1.49)$$

## 1.6 Modeling Gravity Effects

In order to simulate the effect of gravity upon each joint, it was necessary to pick a configuration of the arm in which the other two joints were "locked" into position. This was done for each joint by assuming that the other two joints were fixed at a position of  $\pi/4$ . Computer programs were used to calculate the gravity moments (see Appendix D) using equations 1.8, 1.10, and 1.12. The difference in position (angle) between the system response neglecting gravity and the actual position considering gravity was approximated by the following expression:

$$\Delta\theta = \cos^{-1} \frac{G_i(Q)}{l_{\text{link}} m_{\text{tot}} g}, \quad (1.50)$$

where  $\Delta\theta_i$  is the angular difference,  $l_{\text{link}}$  is the length of the link immediately "behind" the joint, and  $m_{\text{tot}}$  is the total mass of the mechanism "behind" the joint.

The results are tabulated in Table 1.5 for joint one, Table 1.6 for joint two, and Table 1.7 for joint three. By coordinating the joint position with the output of each control system in the following chapters, a rudimentary simulation of gravity's effect is performed.

**Table 1.5:**  
**Approximate Effect of Gravity for Joint One, Others at  $\pi/4$  rad.**

$q_1$ (rad)	$G_1$ (N·m)	$\Delta\theta_1$ (rad)
0	78.0381	0.762405
0.1	82.8462	0.695498
0.2	86.8266	0.63576
0.3	89.9395	0.585443
0.4	92.1537	0.547196
0.5	93.4471	0.523704
0.6	93.8069	0.516999
0.7	93.2293	0.527726
0.8	91.7203	0.554868
0.9	89.2948	0.596169
1.0	85.9771	0.648902



**Table 1.6:**  
**Approximate Effect of Gravity for Joint Two, Others at  $\pi/4$  rad.**

$q_2$ (rad)	$G_2$ (N·m)	$\Delta\theta_2$ (rad)
0	69.3672	0.486695
0.1	53.1067	0.827537
0.2	36.229	1.09096
0.3	18.9027	1.32754
0.4	1.30087	1.55422
0.5	-16.4006	1.78133
0.6	-34.0248	2.01922
0.7	-51.3957	2.28483
0.8	-68.3398	2.62761
0.9	-84.6876	0.0
1.0	-100.276	0.0

**Table 1.7: Approximate Effect of Gravity for Joint Three, Others at  $\pi/4$  rad.**

$q_3$ (rad)	$G_3$ (N·m)	$\Delta\theta_3$ (rad)
0	30.5691	0.897939
0.1	30.1648	0.908435
0.2	28.9558	0.939325
0.3	26.9544	0.988997
0.4	24.1804	1.05529
0.5	20.6616	1.13599
0.6	16.4331	1.22916
0.7	11.5372	1.33336
0.8	6.02282	1.4477
0.9	-0.0549481	1.57192
1.0	-6.63538	1.70649

as small as possible. This is because overshooting may cause the arm to collide with obstacles that lie beyond the target point.

## CHAPTER II

The speed of the control system response can be quantified by the "settling time," which is defined as the time

### SYSTEM CHARACTERISTICS

within 2% of the steady-state value [6]. For the robotic arm system, speed is desirable but not required; speed may be sacrificed. This chapter describes the characteristics and the desired responses of local control systems for each of the three joints in the arm. The term "local" refers to the fact that each joint is controlled independently by its own controller; i.e., only one joint is assumed to be moving at a time, while the other joints are "locked" into position. The effect of removing this requirement is discussed in Appendix B.

The nature of the system response waveform (i.e., the magnitude, percent overshoot, of the poles of the system transfer function in the s-domain).

### 2.1 Desired System Characteristics

The performance of each joint's control system can be summarized by examining four aspects of the system's response to a step input: stability, overshoot, speed, and steady-state error.

Obviously, the system is required to be stable. If the joint controller is not designed with stability in mind, driving the joint to a desired position may cause it to continually oscillate about that point, or otherwise fail to converge at a desired steady-state value.

The response of the control system to a step input over time may be divided into "transient" and "steady-state" responses. The percent overshoot is an expression of how far the system will exceed its target position during the transient period before the desired position is reached at steady-state. For a robotic arm system, it is desired to keep overshoot

of the form:

as small as possible. This is because overshooting may cause the arm to collide with obstacles that lie beyond the target position.

The speed of the control system response can be quantified by the “settling time,” which is defined as the time for the response to reach and stay within 2% of the steady-state value [6]. For the robotic arm system, speed is desirable but not required: speed may be sacrificed to achieve minimal overshoot and zero steady-state error.

Finally, steady-state error expresses the difference between the desired response and the actual response. Effectively, the steady-state error tells how precise the positioning of the arm is. It is desired to achieve zero steady-state error. In the chapters that follow, it will be seen that gravity has a most significant impact on the steady-state error.

The nature of the system response waveform (i.e., the stability, percent overshoot, and speed) depends upon the location of the poles of the system transfer function (in the  $s$ -domain), or the location of the eigenvalues of the system matrix (in state-space). For the system to be stable, the system’s poles or eigenvalues must lie in the left-half of the complex plane [6, 8].

Suppose the poles are written in the form

$$\sigma_{1,2} = -\zeta\omega \pm j\omega\sqrt{1-\zeta^2}, \quad (2.1)$$

where  $\omega$  is a quantity known as the system’s “characteristic frequency,” which expresses the frequency of oscillation of the system response, and  $\zeta$  is the “damping ratio,” which expresses how quickly the response’s amplitude reaches the steady-state amplitude.

If  $\zeta > 1$  there are two real poles, and the response will be an “overdamped” response of the form:

$$r(t) = C_1 e^{-(\zeta\omega + \omega\sqrt{\zeta^2 - 1})t} + C_2 e^{-(\zeta\omega - \omega\sqrt{\zeta^2 - 1})t} \quad (2.2)$$

If  $\zeta < 1$  then the response will be “underdamped,” having a decaying sinusoidal form:

$$r(t) = C_1 e^{-\zeta\omega t} \sin(\omega\sqrt{1-\zeta^2}t) + C_2 e^{-\zeta\omega t} \cos(\omega\sqrt{1-\zeta^2}t). \quad (2.3)$$

If  $\zeta = 0$ , then the poles are imaginary and located on the imaginary axis (i.e., they have the form  $\pm j\omega$ ), and the response is an “undamped” sinusoid of the form:

$$r(t) = C_1 \sin(\omega t) + C_2 \cos(\omega t). \quad (2.4)$$

This type of response will cause the system to oscillate, and thus can be considered an unstable response. The “fastest” stable response with no overshoot is a “critically damped” response, for which  $\zeta = 1$  and two real, repeated roots are present. A critically damped response has the form:

$$r(t) = (C_1 t + C_2) e^{-\omega t}. \quad (2.5)$$

The critically damped response is the form strived for in the design of the control systems for the robot arm.

## 2.2 Simple Feedback System

As motivation for examining the approaches to control system design for the arm joints, this section presents the response of the simple closed-loop system for each joint. Figure 2.1 shows the layout. The input to the system is a voltage,  $V_o$ , corresponding to the desired position of the joint.  $V_q$  is a voltage corresponding to the actual joint position, as measured by a potentiometer with gain  $K_o$ . The error (difference) between  $V_o$  and  $V_q$  is

amplified by a feedforward gain,  $K_{ff}$ , and fed into the “plant” (actuator coupled to the joint). The value of  $K_{ff}$  determines how rapidly the system’s response decays toward a steady-state value. Figures 2.2-2.4 depict the responses of joints one, two, and three for varying  $K_{ff}$ , neglecting gravity. The figures exemplify the problems with the basic feedback approach: overshoot and oscillation of the response about the desired point.

As mentioned previously, gravity has a significant effect on the steady-state error of the system. Therefore, neglecting the effect of gravity on the system is unrealistic. As shown in Figures 2.5-2.7, all three joints controlled by a simple feedback system behave unacceptably when gravity effects are considered. This can be seen by examining the steady-state error between the actual response and the desired step response (i.e., achieving a steady-state value of one radian), as well as the long settling time and oscillation of the response.

The simulation results considered clearly indicate that a basic feedback system is not a realistic option for controlling the joints of the arm.

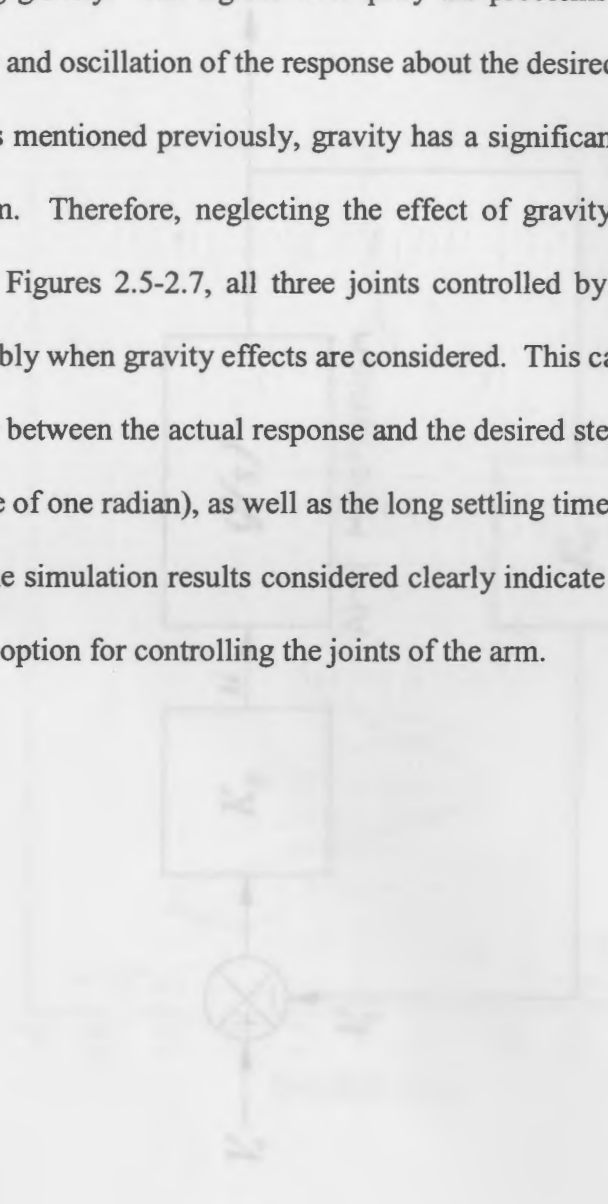


Figure 2.2: Simple feedback control system

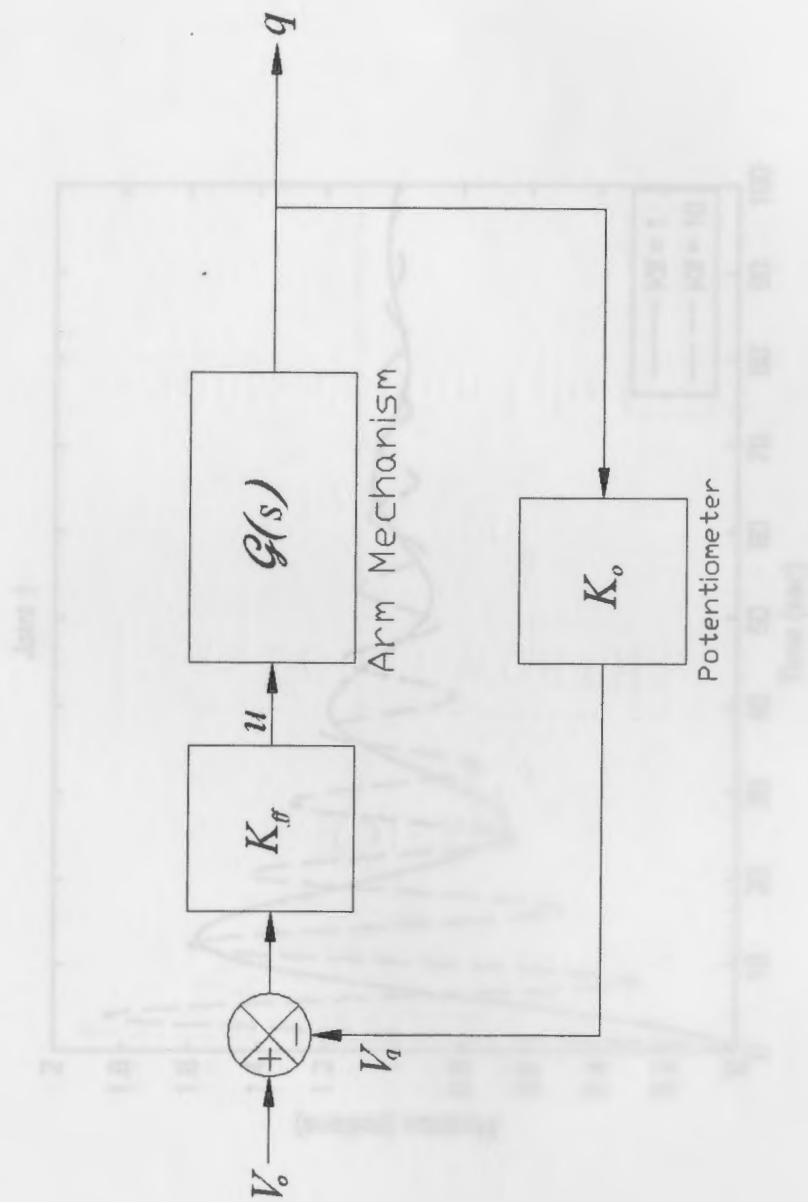


Figure 2.3: Joystick Positioning Gravity.

Figure 2.1: Simple Feedback System.

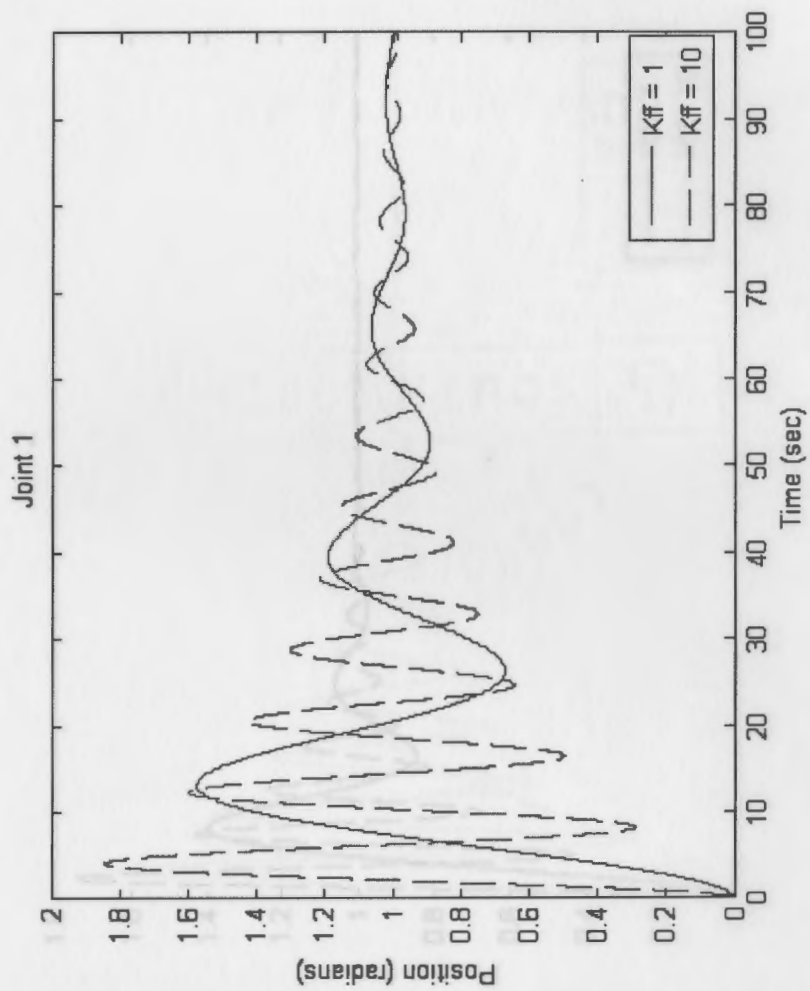


Figure 2.2: Joint 1 Simple Feedback Response, Neglecting Gravity.



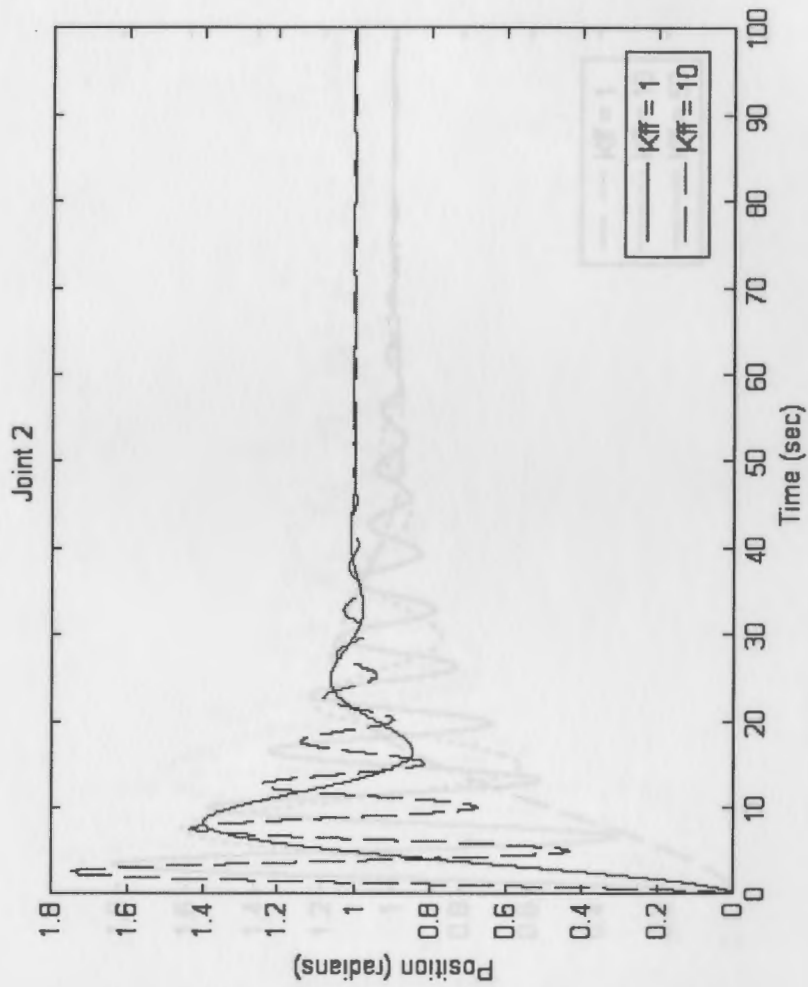


Figure 2.3: Joint 2 Simple Feedback Response, Neglecting Gravity.

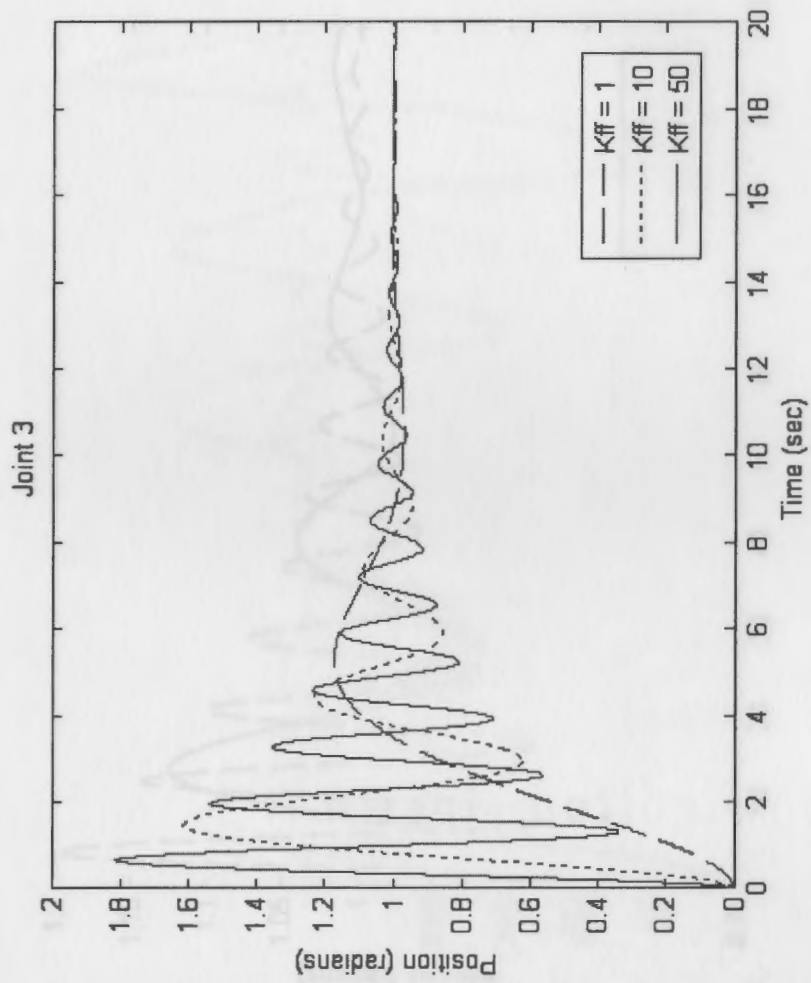


Figure 2.4: Joint 3 Simple Feedback Response, Neglecting Gravity.

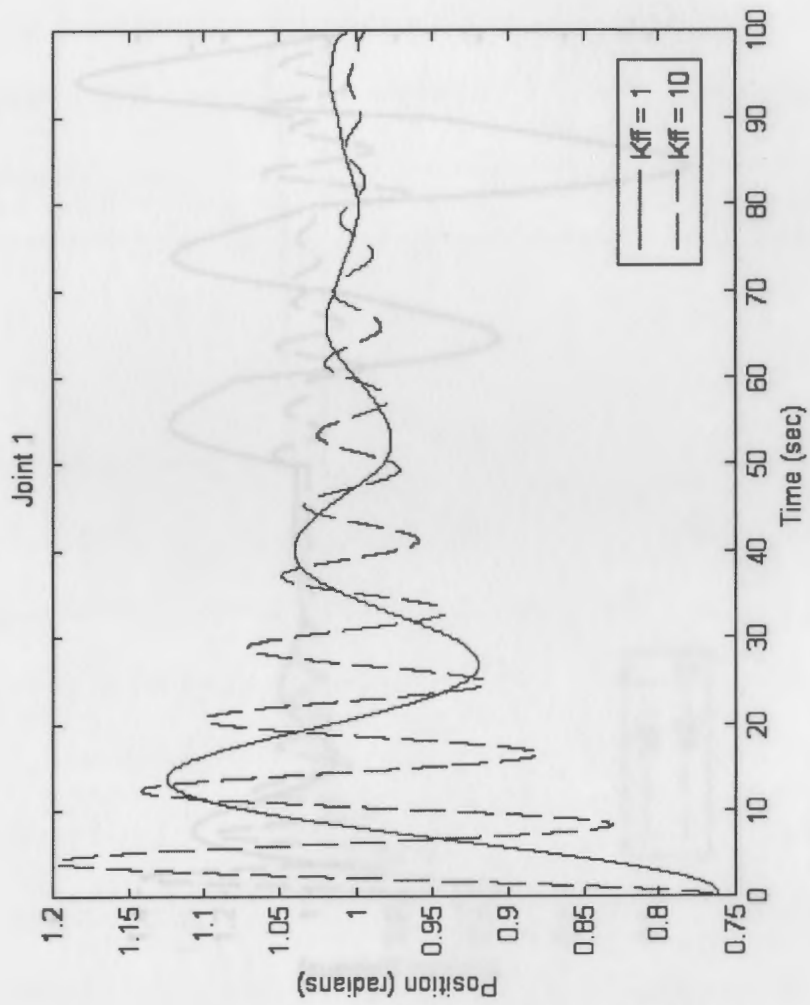


Figure 2.5: Joint 1 Simple Feedback Response, Considering Gravity.

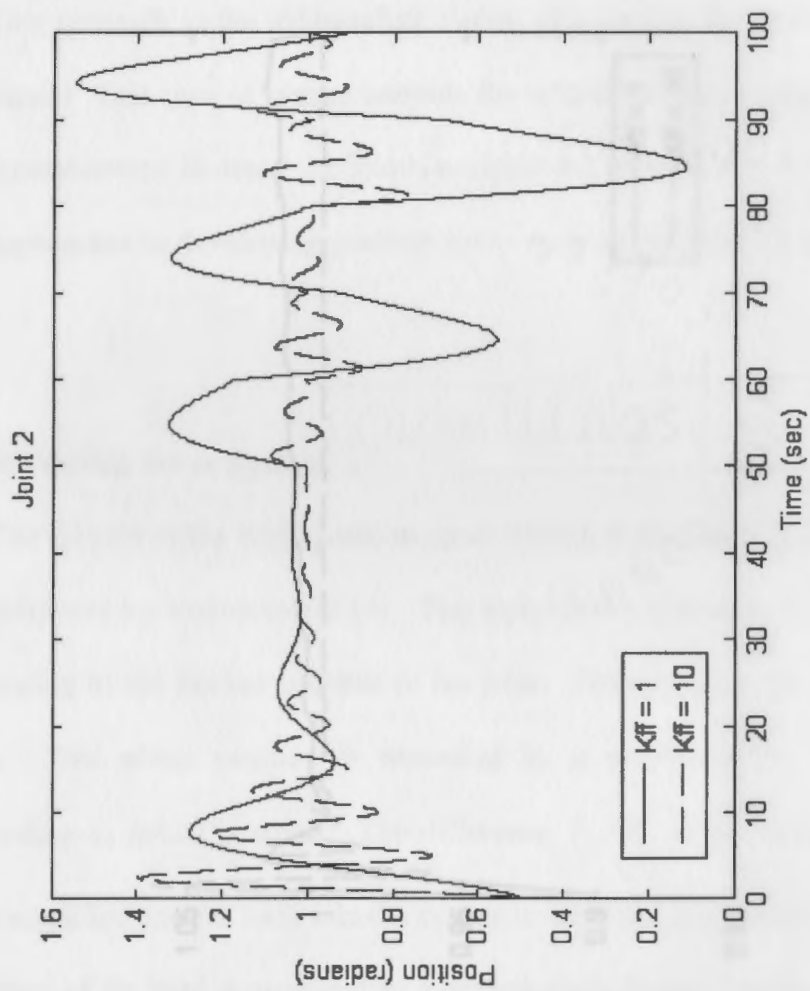


Figure 2.6: Joint 2 Simple Feedback Response, Considering Gravity.

## CHAPTER III

## POSITION SERVO SYSTEMS

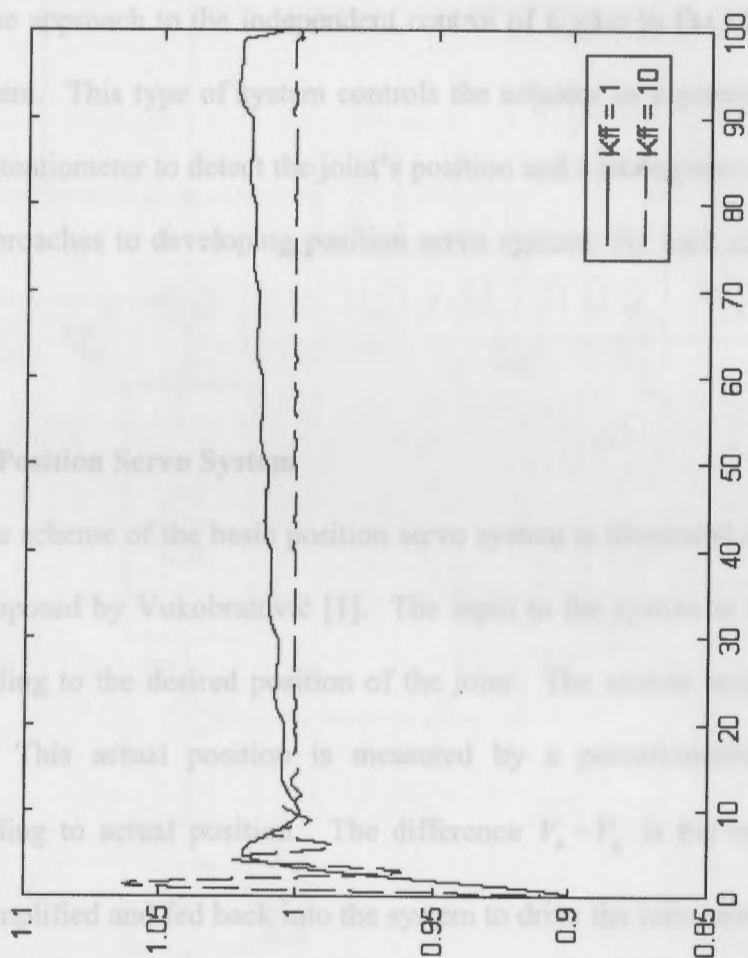


Figure 2.7: Joint Feedback Response, Considering Gravity.

## CHAPTER III

### POSITION SERVO SYSTEMS

One approach to the independent control of a joint in the arm is by using a position servo system. This type of system controls the actuator as a permanent-magnet DC motor, using a potentiometer to detect the joint's position and a tachogenerator to detect its velocity. A few approaches to developing position servo systems for each joint are discussed in this chapter.

#### 3.1 Basic Position Servo System

The scheme of the basic position servo system is illustrated in Figure 3.1, based on a system proposed by Vukobratović [1]. The input to the system is  $V_o$ , a step voltage signal corresponding to the desired position of the joint. The system output is  $q$ , the actual joint position. This actual position is measured by a potentiometer, giving a voltage  $V_q$  corresponding to actual position. The difference  $V_o - V_q$  is the system error. This error signal is amplified and fed back into the system to drive the joint toward the desired position. The position of the joint is measured by a potentiometer coupled to the joint's actuator shaft. The velocity (derivative of the position) is measured using a tachogenerator, this signal is amplified if necessary, and it is fed back into the controller. The effect of gravity is modeled as an external disturbance acting upon the system. The "plant" is the combined joint-actuator

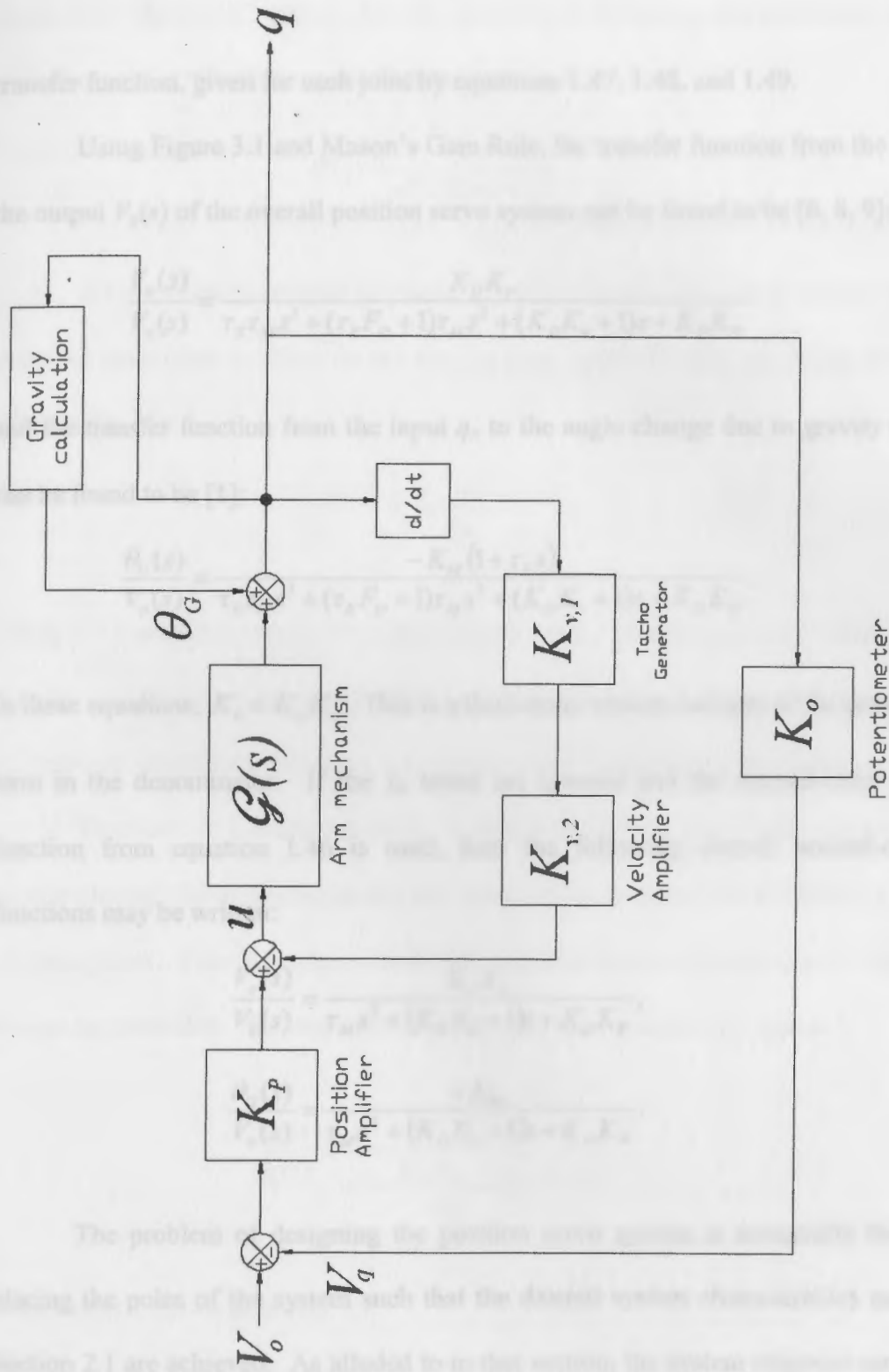


Figure 3.1: Position Servo System.

transfer function, given for each joint by equations 1.47, 1.48, and 1.49.

Using Figure 3.1 and Mason's Gain Rule, the transfer function from the input  $V_o(s)$  to the output  $V_q(s)$  of the overall position servo system can be found to be [6, 8, 9]:

$$\frac{V_q(s)}{V_o(s)} = \frac{K_D K_P}{\tau_E \tau_M s^3 + (\tau_E F_D + 1) \tau_M s^2 + (K_D K_v + 1) s + K_D K_P} \quad (3.1)$$

and the transfer function from the input  $q_o$  to the angle change due to gravity moment,  $\theta_G$ , can be found to be [1]:

$$\frac{\theta_G(s)}{V_o(s)} = \frac{-K_M(1 + \tau_E s)}{\tau_E \tau_M s^3 + (\tau_E F_D + 1) \tau_M s^2 + (K_D K_v + 1) s + K_D K_P} \quad (3.2)$$

In these equations,  $K_v = K_{v_1} K_{v_2}$ . This is a third-order system because of the presence of the  $s^3$  term in the denominator. If the  $\tau_E$  terms are ignored and the second-order joint transfer function from equation 1.46 is used, then the following overall second-order transfer functions may be written:

$$\frac{V_q(s)}{V_o(s)} = \frac{K_D K_P}{\tau_M s^2 + (K_D K_v + 1) s + K_D K_P}, \quad (3.3)$$

$$\frac{\theta_G(s)}{V_o(s)} = \frac{-K_M}{\tau_M s^2 + (K_D K_v + 1) s + K_D K_P}. \quad (3.4)$$

The problem of designing the position servo system is essentially the problem of placing the poles of the system such that the desired system characteristics as discussed in Section 2.1 are achieved. As alluded to in that section, the system response can be specified by deciding upon values of the characteristic frequency  $\omega$  and the damping ratio  $\zeta$  in order to



place two “dominant” poles. For the purpose of designing the controller, the system is therefore approximated by a second-order system of the form:

$$G(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}. \quad (3.5)$$

For a critically damped response,  $\zeta = 1$  is chosen. The characteristic frequency and damping ratio have an effect on the settling time, approximately according to the following relationship [6]:

$$T_s \approx \frac{4}{\zeta\omega}. \quad (3.6)$$

Using  $\zeta = 1$  and choosing  $T_s = 1$  s gives  $\omega \approx 4$  rad/s. Substituting these values into equation 2.1, the two dominant poles will both be placed at  $-4$ . The third pole is free to move about the left-half of real axis; however, it must be located far to the left of the dominant poles.

To place the dominant poles in the desired locations, appropriate values of  $K_P$  and  $K_V$  must be found. First, it is noted that the values of  $\tau_E$ , as tabulated in Table 1.4, are small for all three joints. Therefore, the second-order transfer function in equation 3.3 may be used to design the controller. Writing equation 3.3 in the form given by 3.5 gives:

$$\frac{q(s)}{q_o(s)} \approx \frac{\frac{K_D K_P}{\tau_M}}{s^2 + \frac{K_D K_V + 1}{\tau_M} s + \frac{K_D K_P}{\tau_M}} \quad (3.7)$$

from which it can be seen that

$$\omega = \sqrt{\frac{K_D K_P}{\tau_M}} \quad (3.8)$$

and Table 3.1 summarizes the values of  $K_D$  and  $K_v$  for each of the three joints, using the transfer function parameters in Table 3.1 and the maximum values given in Table 1.2 and a tolerable steady-state error of 0.01 rad.

$$\zeta = \frac{K_D K_v + 1}{2\tau_M \sqrt{K_D K_P / \tau_M}} \quad (3.9)$$

Next, it is noted that the steady-state error can be determined using the Final Value Theorem [6]:

$$E(\infty) = \lim_{s \rightarrow 0} |sG(s)| \quad (3.10)$$

From equations 3.1, 3.2, and 3.10, it is found that:

$$E(\infty) = \frac{|G(\infty)|K_M}{K_D K_P} \quad (3.11)$$

This equation shows that the steady-state error is directly proportional to the steady-state value of the gravity moment, and inversely proportional to the position gain  $K_P$ . In order to minimize steady-state error, it is thus desirable for  $K_P$  to be a large value. Furthermore, equation 3.11 expresses a way to calculate  $K_P$  for a tolerable steady-state error value and known values of  $K_M$  and  $K_D$ . The value of  $K_P$ , then, may be found using equations 3.8 and 3.11:

$$\frac{\omega^2 \tau_M}{K_D} \leq K_P \leq \frac{|G(\infty)|K_M}{K_D e(\infty)} \quad (3.12)$$

Once the value of  $K_P$  is calculated,  $K_v$  may be found by substituting equations 1.42 and 1.45 into equation 3.9 and solving:

$$K_v = \frac{2\zeta \sqrt{R_R(N_M N_v J_M + H_i)C_M N_M K_P - R_R F_v} - N_v C_E}{C_M N_M} \quad (3.13)$$

Table 3.1 summarizes the values of  $K_P$  and  $K_V$  for each of the three joints, using the transfer function parameters in Table 1.4, the maximum values given in Table 1.2, and a tolerable steady-state error of 0.01 rad.

Figures 3.2-3.4 show the responses of the systems neglecting gravity, and Figures 3.5-3.7 show the responses considering the effect of gravity. Notice that gravity has little effect upon joint one. The effect of gravity for joints two and three is more apparent.

**Table 3.1:**  
**Calculated Simple Servo System Gains.**

Parameter	Joint 1	Joint 2	Joint 3
$K_P$	30,478	12,298	8,555
$K_V$	23,950	16.032	271.07

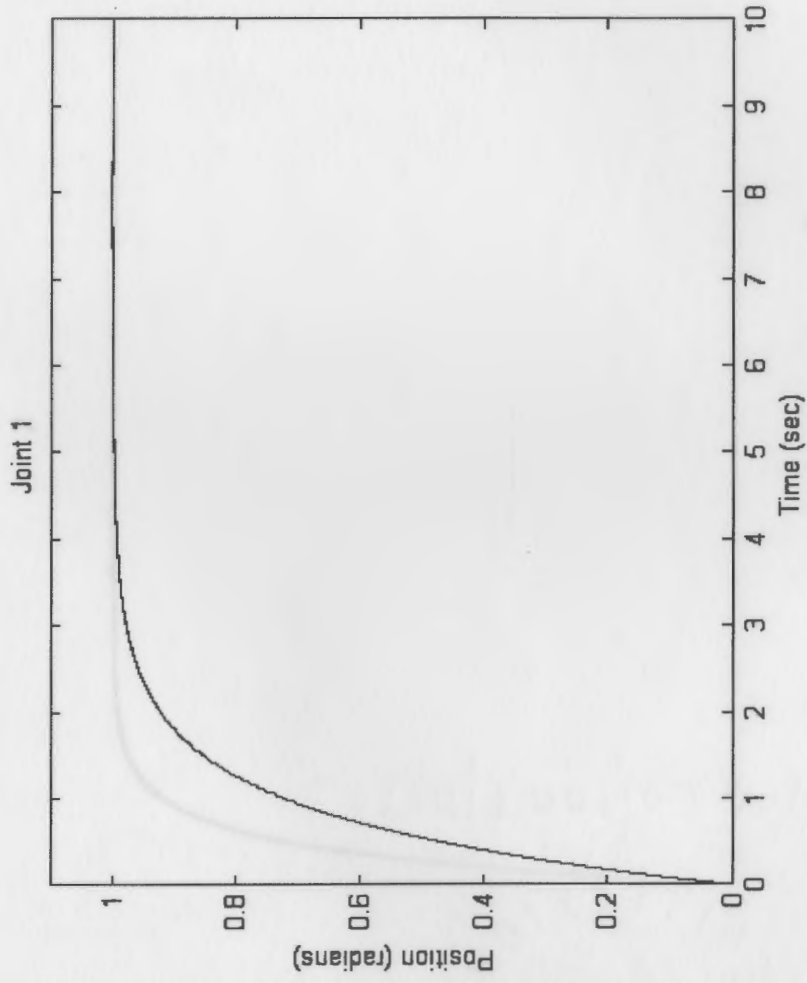
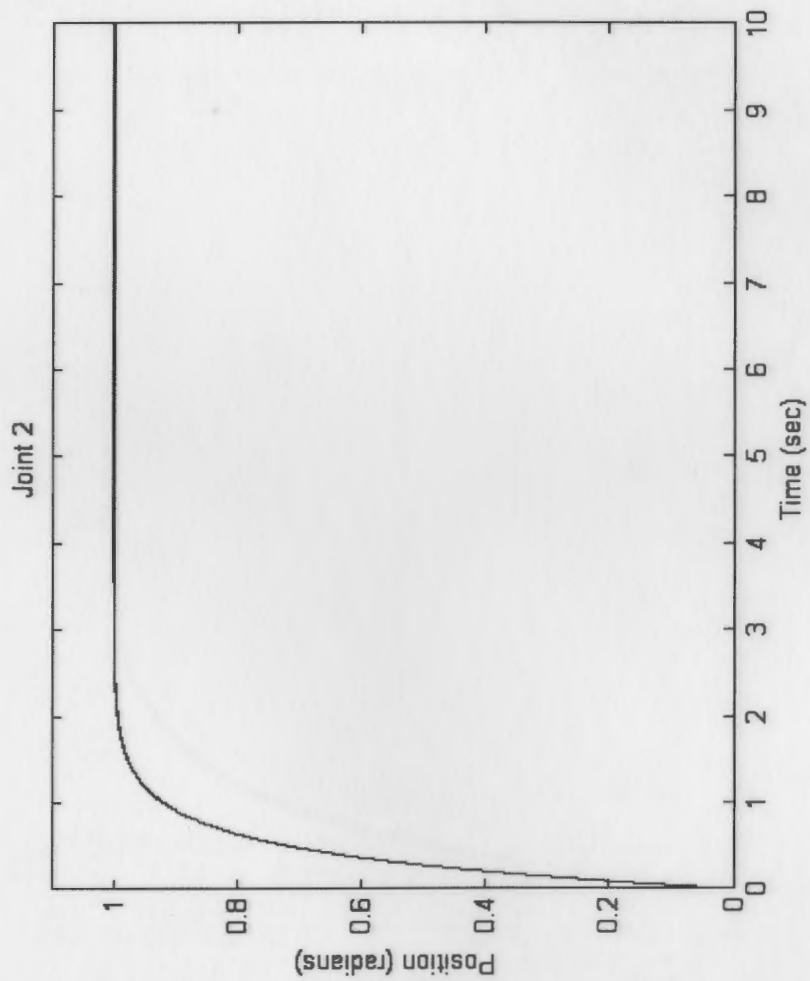


Figure 3.2: Joint 1 Position Servo Response, Neglecting Gravity.



**Figure 3.3: Joint 2 Position Servo Response, Neglecting Gravity.**

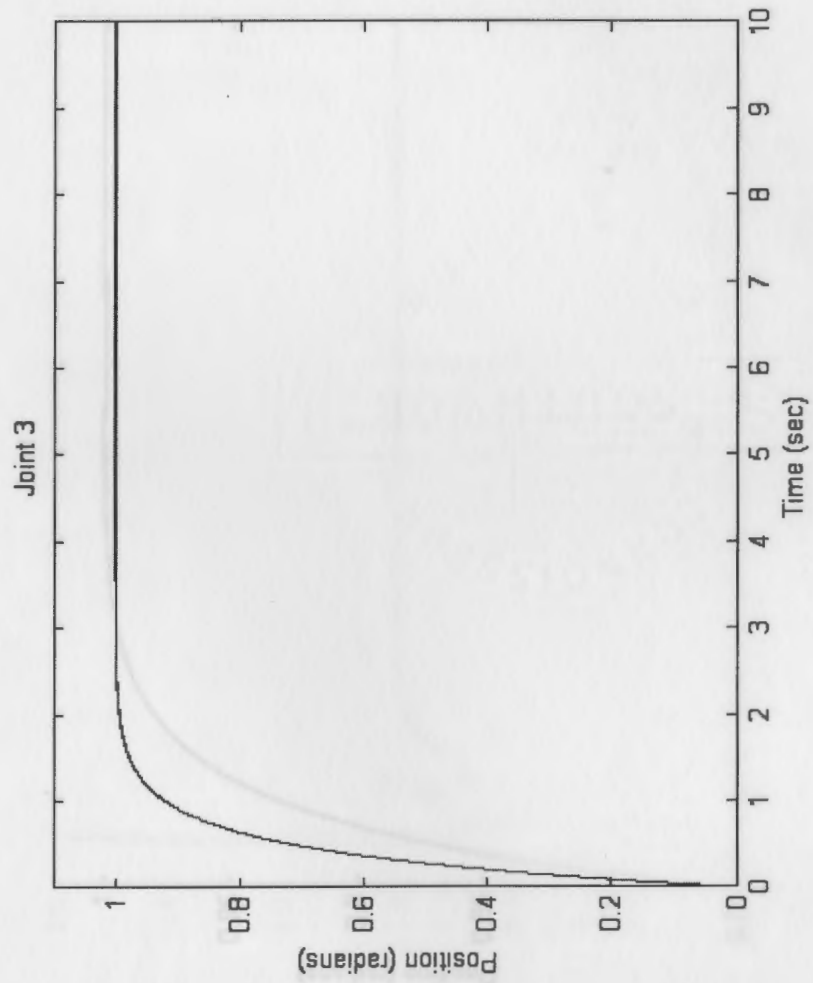


Figure 3.4: Joint 3 Position Servo Response, Neglecting Gravity.

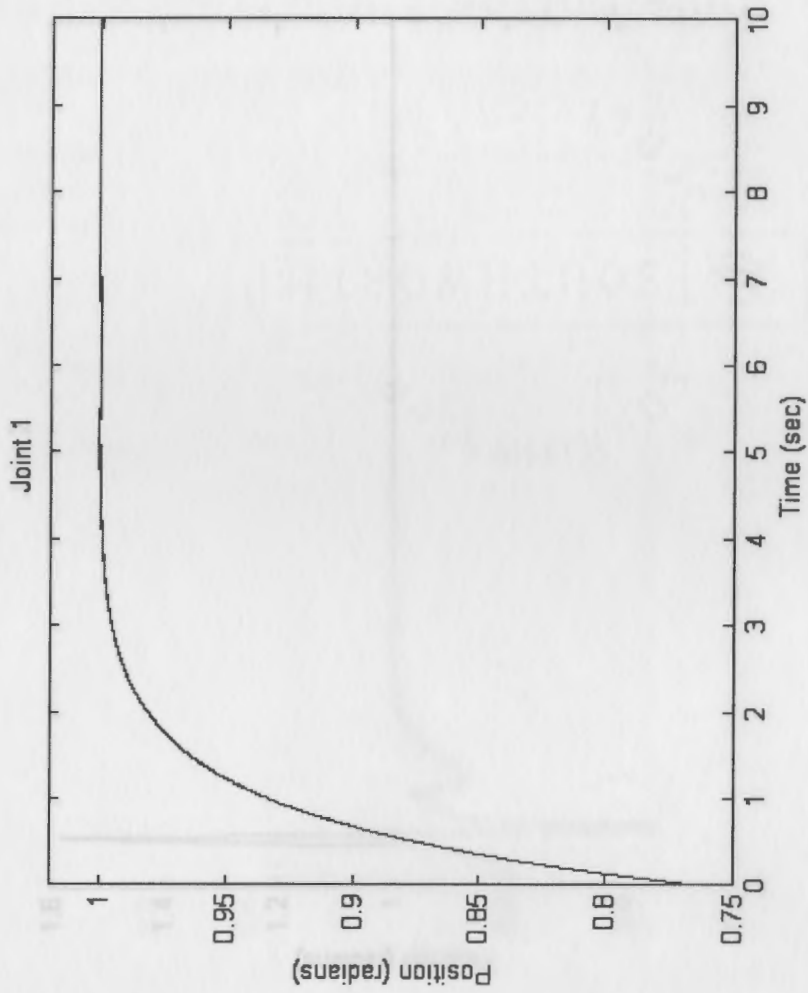


Figure 3.5: Joint 1 Position Servo Response, Considering Gravity.

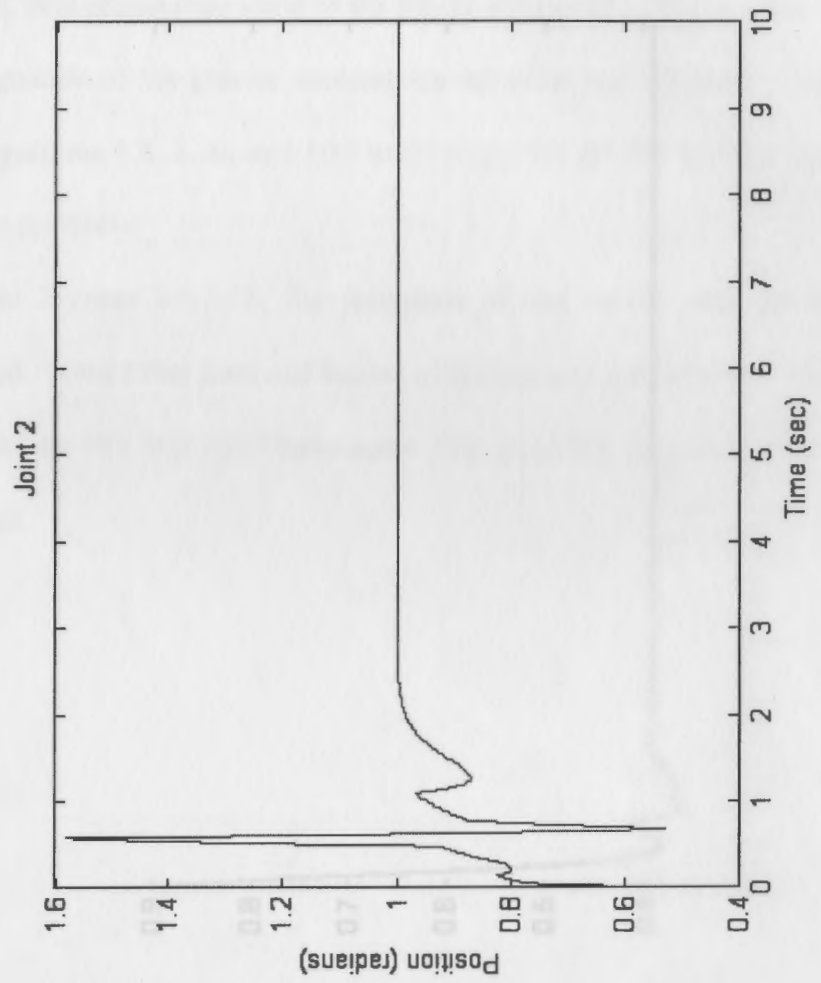


Figure 3.6: Joint 2 Position Servo Response, Considering Gravity.



### 3.2 Servo System with Gravity Compensation

The steady-state error of the servo system described in section 3.1 is caused by the gravity moment about the joint (see equation 3.11). Figure 3.6 shows how an additional block diagram element can be introduced in order to directly compensate for the gravity moment, thus eliminating some of the steady-state error. This requires the ability to calculate the magnitude of the gravity moment for the joint, e.g., by programming a microcontroller using equations 1.8, 1.10, and 1.12 to calculate the gravity moment for joints one, two, and three, respectively.

In Figures 3.9-3.11, the responses of the servo with gravity compensation are presented. Notice that joint one has no overshoot and a steady-state error that is very small. Joints two and three have some overshoot but very little steady-state error in their steady-state responses.

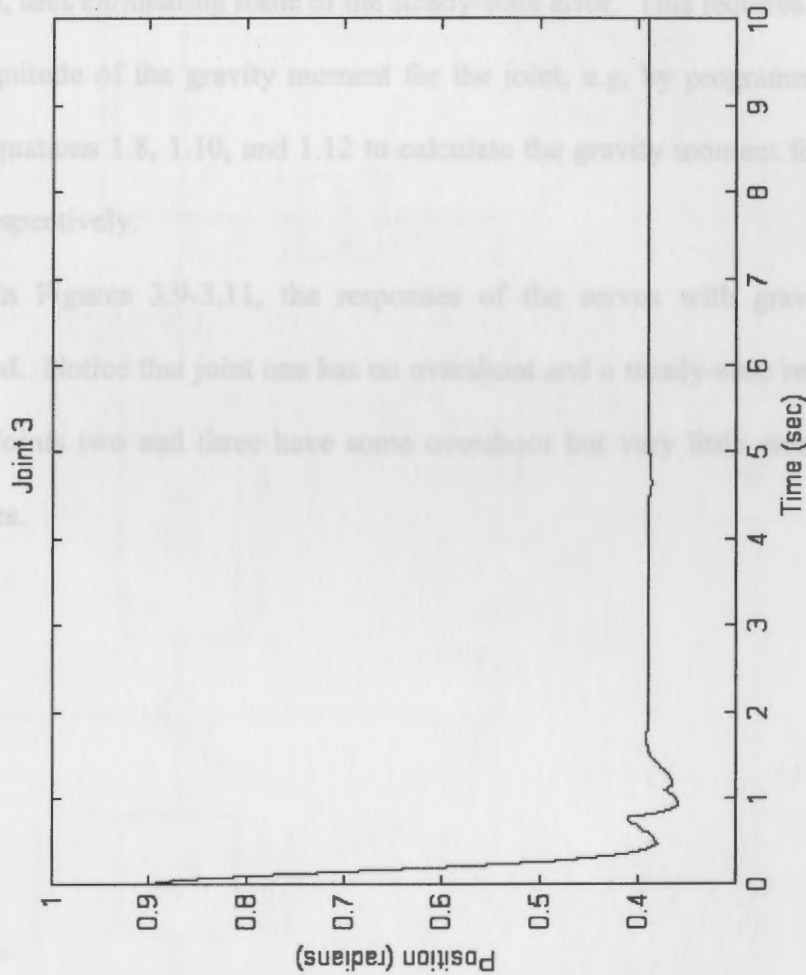


Figure 3.7: Joint 3 Position Servo Response, Considering Gravity.

### 3.2 Servo System with Gravity Compensation

The steady-state error of the servo system described in section 3.1 is caused by the gravity moment about the joint (see equation 3.11). Figure 3.8 shows how an additional block diagram element can be introduced in order to directly compensate for the gravity moment, thus eliminating some of the steady-state error. This requires the ability to calculate the magnitude of the gravity moment for the joint, e.g. by programming a microcontroller using equations 1.8, 1.10, and 1.12 to calculate the gravity moment for joints one, two, and three, respectively.

In Figures 3.9-3.11, the responses of the servos with gravity compensation are presented. Notice that joint one has no overshoot and a steady-state response with very little error. Joints two and three have some overshoot but very little error in their steady-state responses.

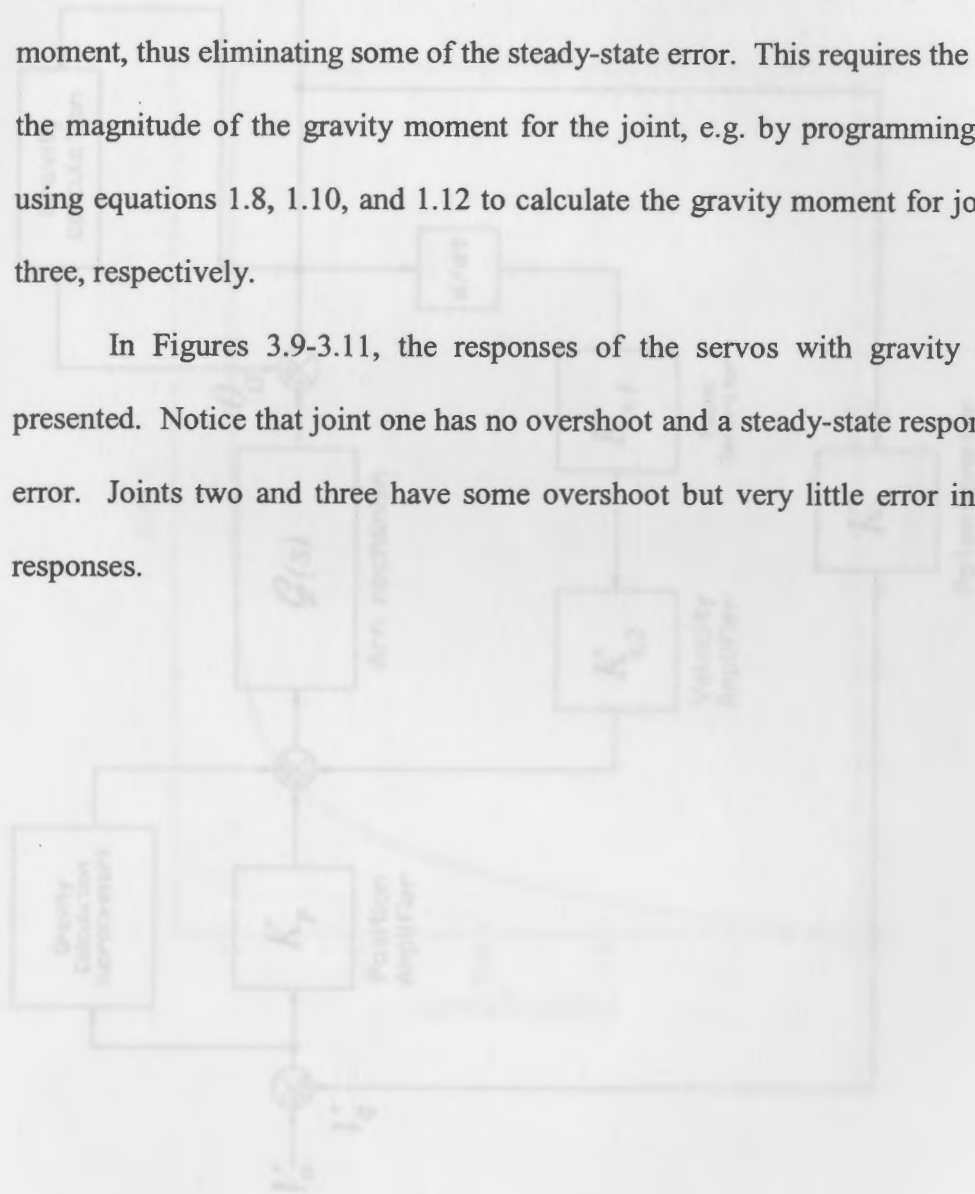


Figure 3.8: Servo System with Gravity Compensation

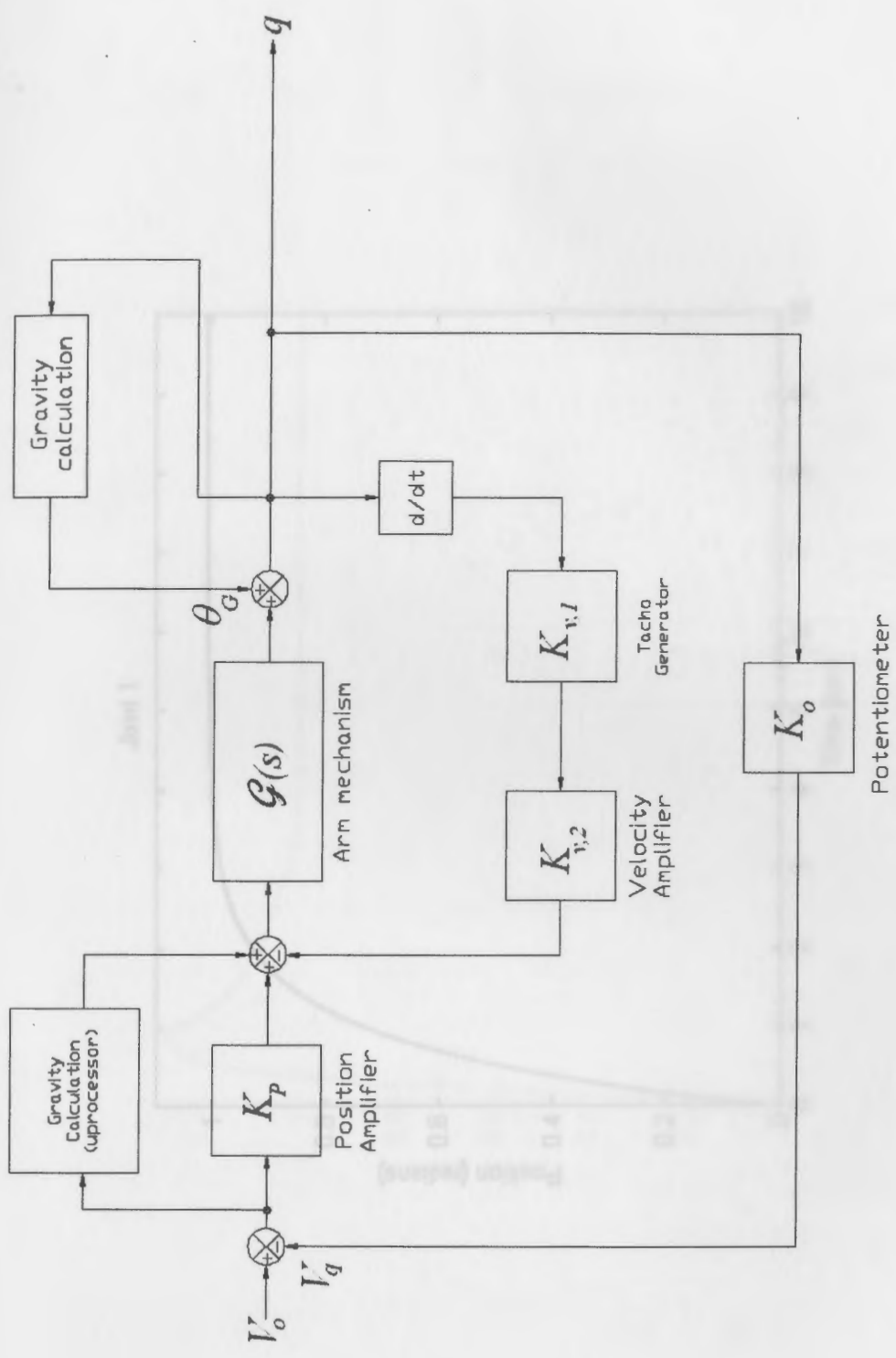


Figure 3.8: Servo System with Gravity Compensation.

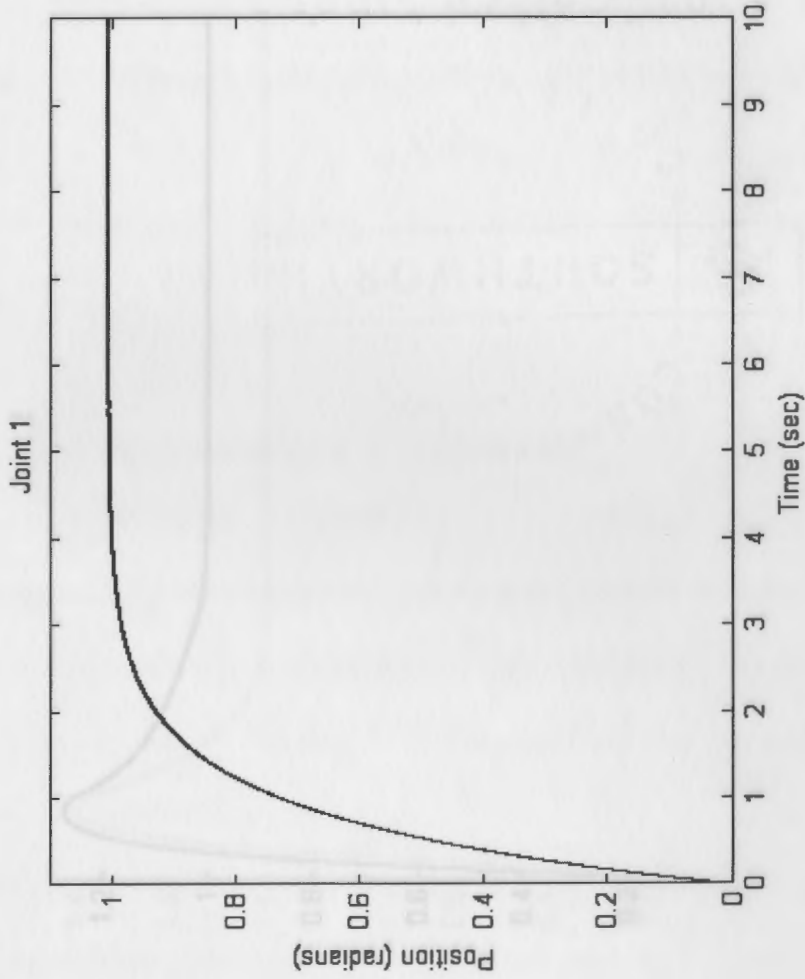


Figure 3.9: Joint 1 Servo with Gravity Compensation Response.

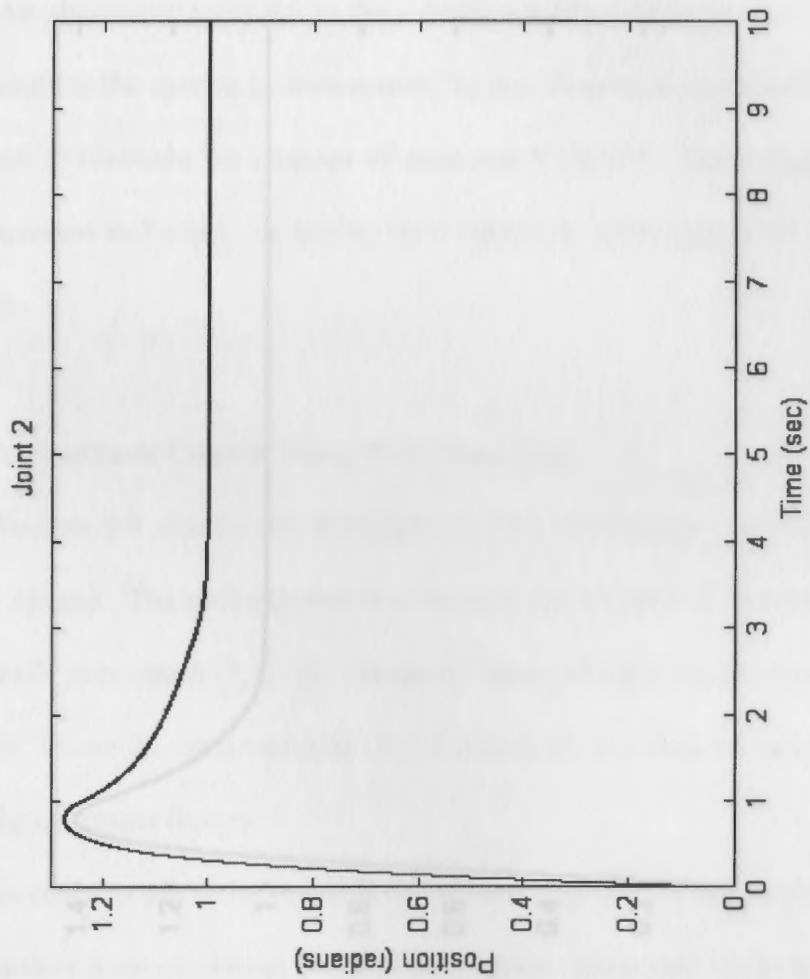


Figure 3.10: Joint 2 Servo with Gravity Compensation Response.

## CHAPTER IV

## STATE-SPACE AND PID CONTROL

An alternative approach to the s-domain method of controller design used in Chapter

3 is to analyze the system in state-space. In this chapter, a state feedback control system developed to illustrate the concept of state-space control. The system is designed using pole placement technique. A similar pole placement technique can be used to design a controller.

## 4.1 State Feedback Control Using Pole Placement

Section 1.4 details the development of a state-space representation of each joint actuator system. The pole placement technique can be used to design a control system for each joint in state-space [4]. The pole placement technique involves the design of a state feedback controller, where the state variables of the system are fed back to the controller. The part of the controller design.

In a state feedback control system, the state variables of the system are fed back to the controller. The system matrix must be chosen such that the closed-loop system is stable. The system poles calculated in section 3.1. Let the desired eigenvalues be  $\lambda_1, \lambda_2, \dots, \lambda_n$ , where  $\lambda_1$  and  $\lambda_2$  are -4 (compare the pole locations calculated in Chapter 3) and  $\lambda_3$  is calculated from:

$$\lambda_3 = -\lambda_1 - \lambda_2 = -8 \quad (4.1)$$

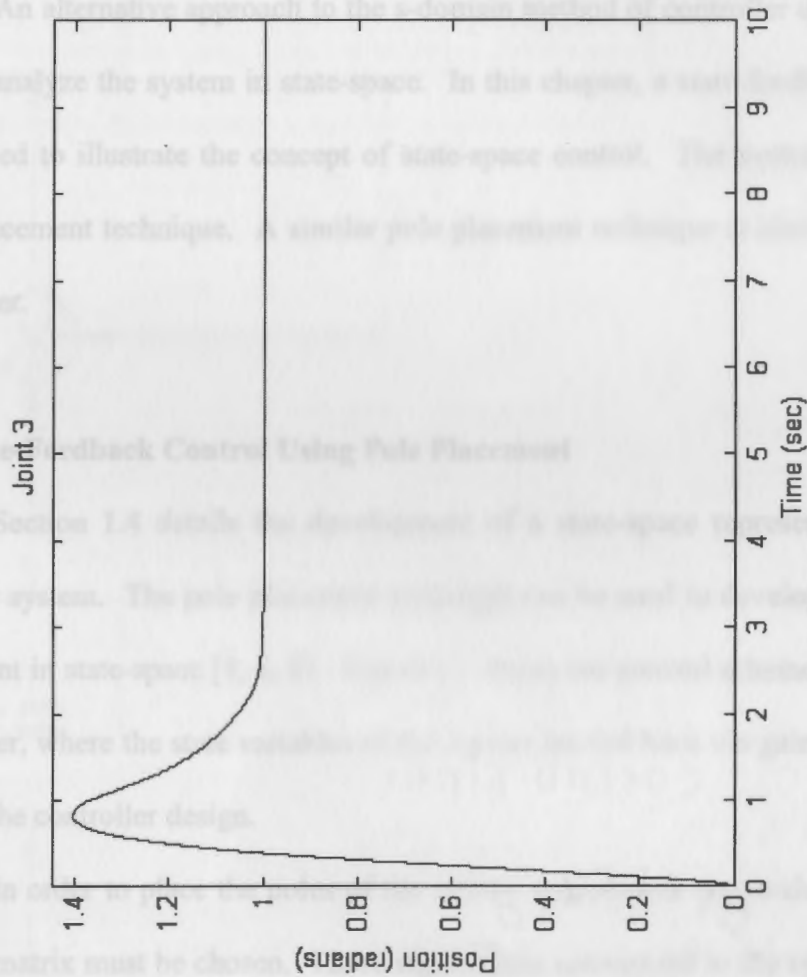


Figure 3.11: Joint 3 Servo with Gravity Compensation Response.

## CHAPTER IV

### STATE-SPACE AND PID CONTROL

An alternative approach to the s-domain method of controller design used in Chapter 3 is to analyze the system in state-space. In this chapter, a state-feedback control system is developed to illustrate the concept of state-space control. The system is designed using a pole placement technique. A similar pole placement technique is also used to design a PID controller.

#### 4.1 State-Feedback Control Using Pole Placement

Section 1.4 details the development of a state-space representation of each joint-actuator system. The pole placement technique can be used to develop a control system for each joint in state-space [1, 6, 8]. Figure 4.1 shows the general scheme of the state-feedback controller, where the state variables of the system are fed back via gains that are specified as part of the controller design.

In order to place the poles of the system, appropriate eigenvalues of the closed-loop system matrix must be chosen. These eigenvalues correspond to the system poles calculated in section 3.1. Let the desired eigenvalues be  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , where  $\lambda_1$  and  $\lambda_2$  are -4 (compare the pole locations calculated in Chapter 3) and  $\lambda_3$  is calculated from:

$$\lambda_3 = a_{22} + a_{33} + 2\zeta\omega \quad (4.1)$$

as suggested by Vukobratović [1]. In these equations,  $\alpha_{22}$  and  $\alpha_{11}$  are the appropriate values from the system matrix  $\hat{A}$ .

The gains required to place the eigenvalues are given by  $K$ , the feedback gain matrix:

$$K = [K_1 \quad K_2 \quad K_3] \quad (4.2)$$

The feedback gains required to place the eigenvalues at the desired locations can be found using the characteristic equation for the system [2]:

$$|sI - (\hat{A} - BK)| = (s - \lambda_1)(s - \lambda_2) = s^2 + a_1s + a_0 \quad (4.3)$$

Solving

$$K_1 = \frac{a_1 + \alpha_{11} - \lambda_1 - \lambda_2 - a_0}{b_1} \quad (4.4)$$

$$K_2 = \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2 - a_1\alpha_{11} + a_0(\lambda_1 + \lambda_2) + \alpha_{11}^2 - \alpha_{11}a_1}{b_1} \quad (4.5)$$

$$K_3 = \frac{\lambda_1\lambda_2 + \alpha_{22}}{b_2} \quad (4.6)$$

In these equations,  $\alpha_j$  refers to the corresponding element of  $\hat{A}$ , and  $b_j$  refers to the corresponding element of matrix  $\hat{B}$ . Table 4.1 gives the values of the gains obtained for the system matrices given in equations 1.20, 1.21, and 1.22.

Figures 4.2-4.4 show the system responses for the three required gains for each joint. The responses considering the effects of gravity are given in Figures 4.5-4.7. The state-feedback system performs well without gravity, but when gravitational effects are included the responses are very undesirable.

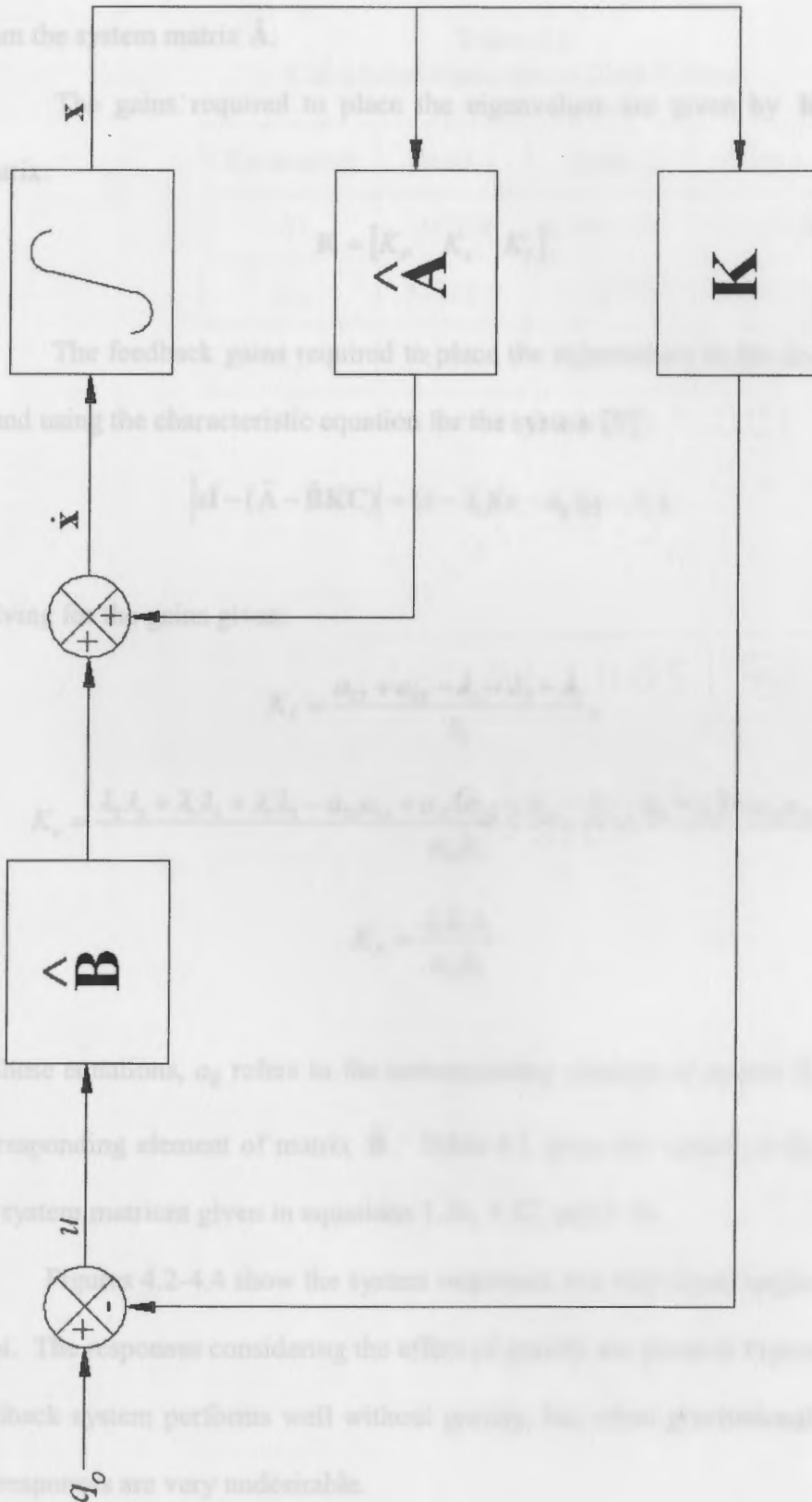


Figure 4.1: State-Feedback System.



as suggested by Vukobratović [1]. In this equation,  $a_{22}$  and  $a_{33}$  are the appropriate values from the system matrix  $\hat{\mathbf{A}}$ .

Table 4.1:  
Calculated State-Space Gain Values.

The gains required to place the eigenvalues are given by  $\mathbf{K}$ , the feedback gain matrix:

Parameter	Joint 1	Joint 2	Joint 3
$K_p$	0.0768	0.76717	0.00004
$K_v$	37322.5	561376	344000
$K_f$			

$$\mathbf{K} = [K_p \quad K_v \quad K_f]. \quad (4.2)$$

The feedback gains required to place the eigenvalues in the desired locations can be found using the characteristic equation for the system [7]:

$$|s\mathbf{I} - (\hat{\mathbf{A}} - \hat{\mathbf{B}}\mathbf{K})| = (s - \lambda_1)(s - \lambda_2)(s - \lambda_3). \quad (4.3)$$

Solving for the gains gives:

$$K_f = \frac{a_{22} + a_{33} - \lambda_3 - \lambda_2 - \lambda_1}{b_3}, \quad (4.4)$$

$$K_v = \frac{\lambda_2\lambda_3 + \lambda_1\lambda_3 + \lambda_1\lambda_2 - a_{22}a_{33} + a_{22}(a_{22} + a_{33} - \lambda_3 - \lambda_2 - \lambda_1) + a_{23}a_{32}}{a_{23}b_3}, \quad (4.5)$$

$$K_p = \frac{\lambda_1\lambda_2\lambda_3}{a_{23}b_3}. \quad (4.6)$$

In these equations,  $a_{ij}$  refers to the corresponding element of matrix  $\hat{\mathbf{A}}$ , and  $b_k$  refers to the corresponding element of matrix  $\hat{\mathbf{B}}$ . Table 4.1 gives the values of the gains calculated for the system matrices given in equations 1.36, 1.37, and 1.38.

Figures 4.2-4.4 show the system responses to a step input, neglecting gravity for each joint. The responses considering the effect of gravity are given in Figures 4.5-4.7. The state-feedback system performs well without gravity, but when gravitational effects are included the responses are very undesirable.

**Table 4.1:**  
**Calculated State-Space Gain Values.**

Parameter	Joint 1	Joint 2	Joint 3
$K_I$	0.0368	$8.74 \times 10^{-8}$	0.000004
$K_v$	37322.5	56.1476	14.4035
$K_P$	74428.4	101.977	28.725

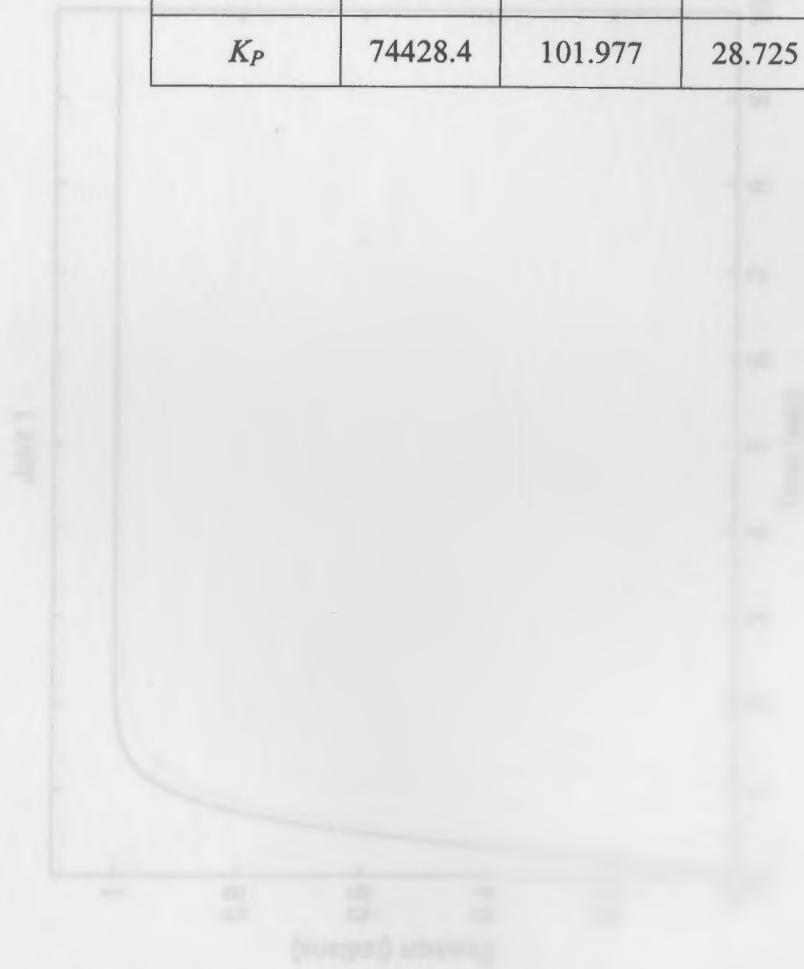


Figure 4.2: Joint 1 State-Feedback Response, Neglecting Gravity.

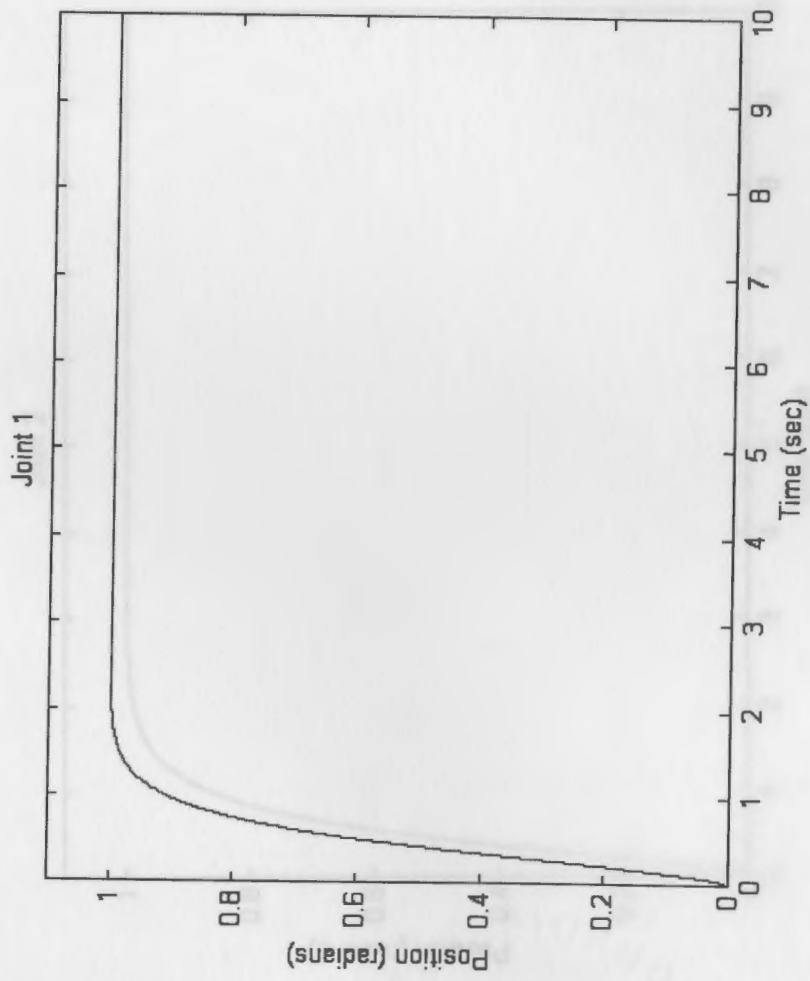


Figure 4.2: Joint 1 State-Feedback Response, Neglecting Gravity.

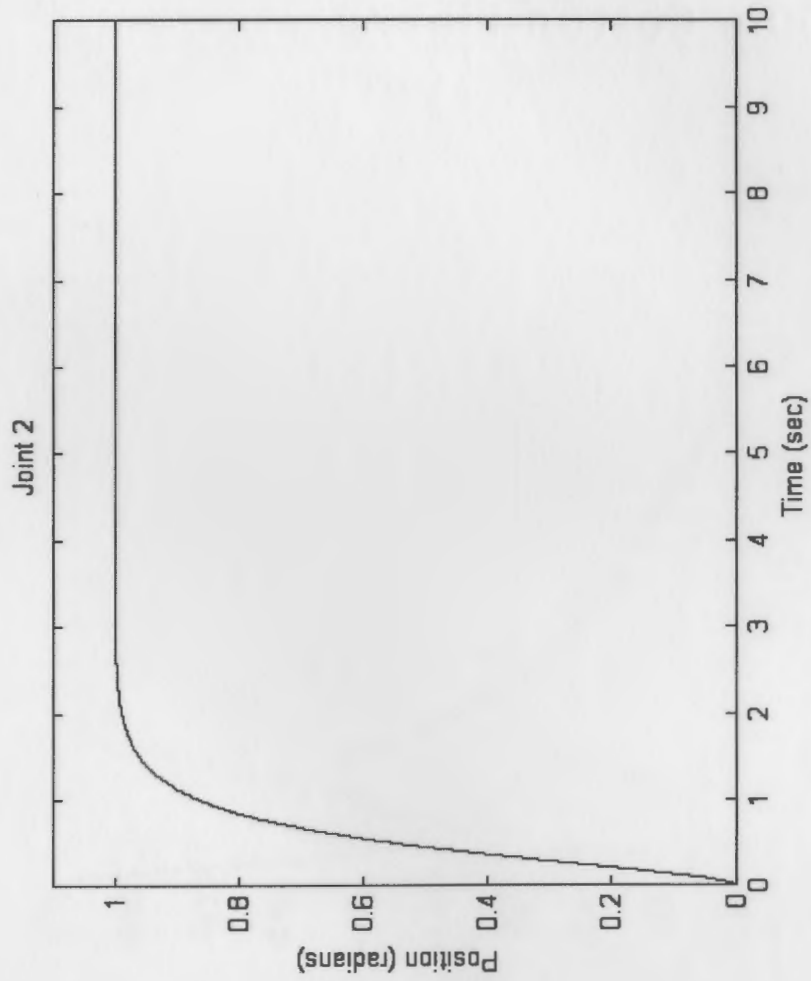


Figure 4.3: Joint 2 State-Feedback Response, Neglecting Gravity.

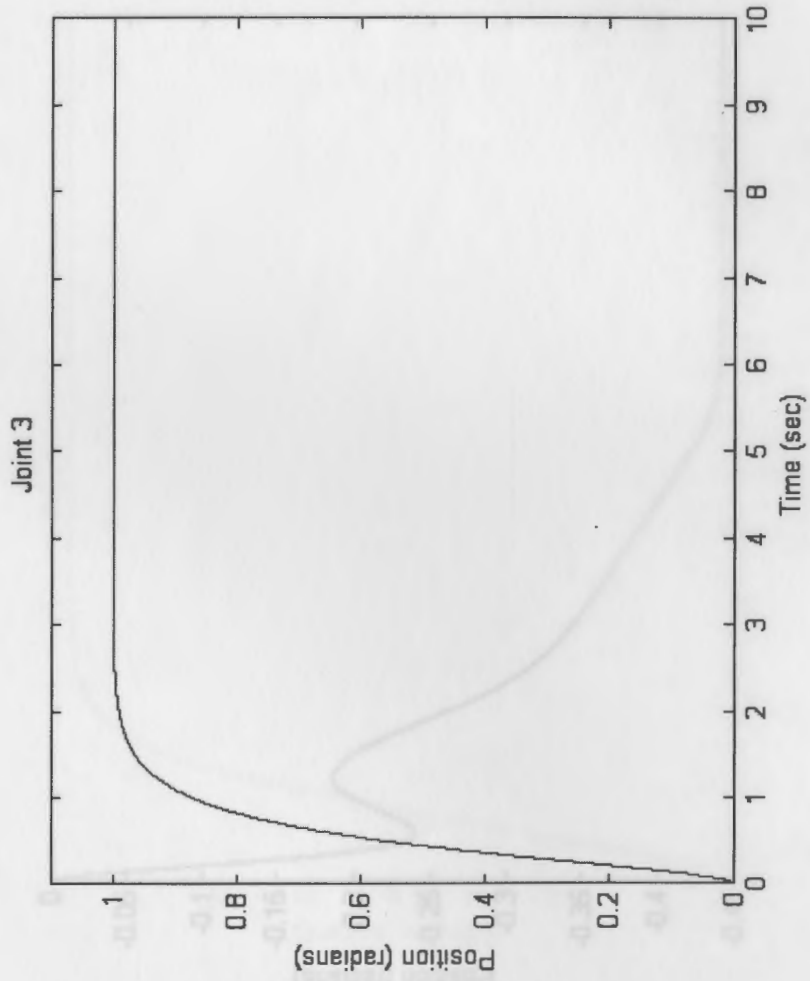


Figure 4.4: Joint 3 State-Feedback Response, Neglecting Gravity.

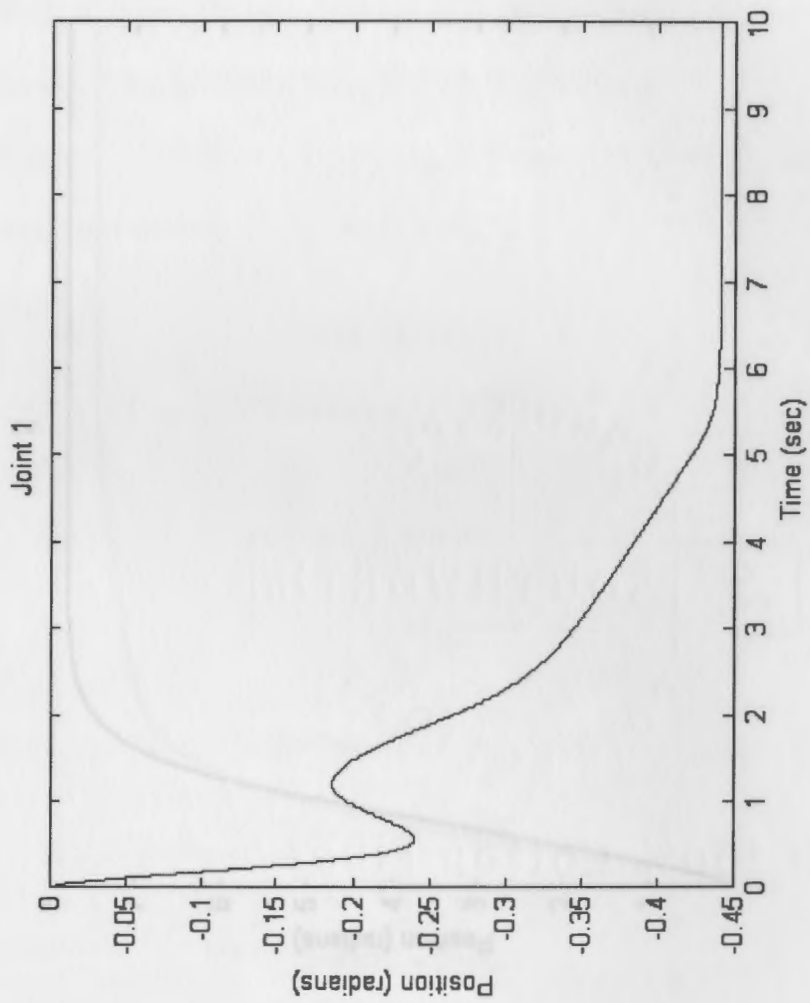


Figure 4.5: Joint 1 State-Feedback Response, Considering Gravity.

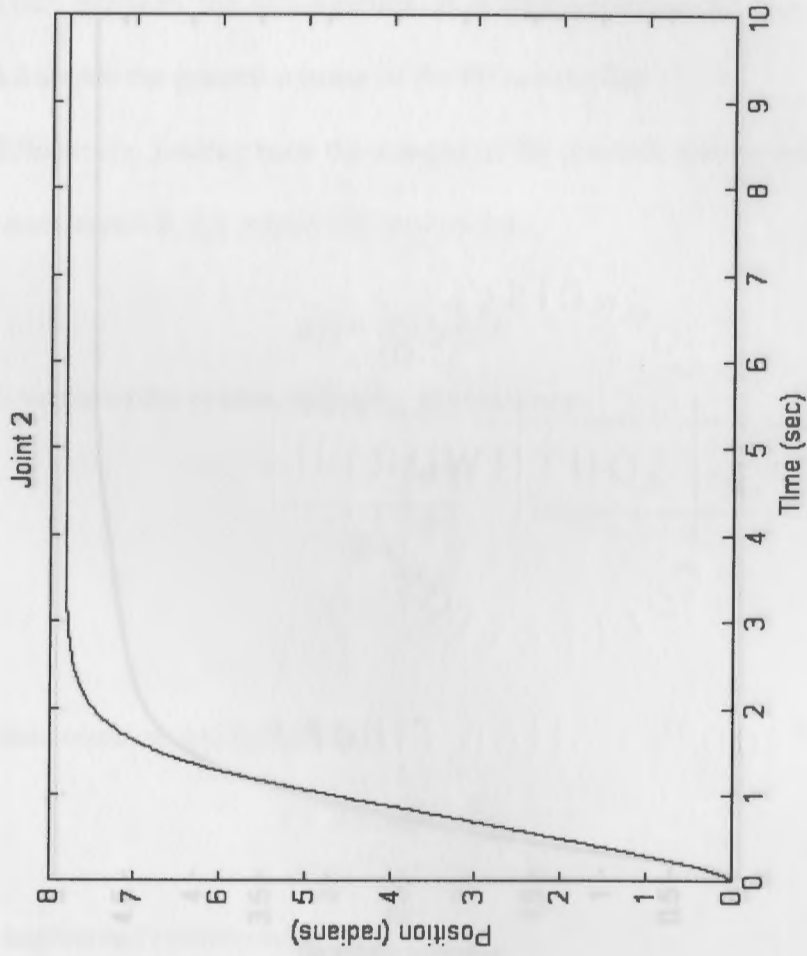


Figure 4.6: Joint 2 State-Feedback Response, Considering Gravity.

## 4.2 Local PID Controller

The steady-state error of the control systems developed thus far, which is caused in large part by gravitational effects, may be eliminated by feeding back the integral of the error signal, developing what is known as a PID (proportional-integral-derivative) controller [1, 8].

This section explains the development of a PID controller for a joint-actuator system.

Figure 4.8 shows the general scheme of the PID controller.

Effectively, feeding back the integral of the position error is equivalent to introducing a fourth state variable  $x(t)$ , where  $x(t)$  is given by

$$x(t) = \int_0^t (y - y_d) dt \quad (4.7)$$

The state vector of the system including  $x(t)$  becomes

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix} \quad (4.8)$$

and the state equation can be written

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (4.9)$$

with the augmented system matrix

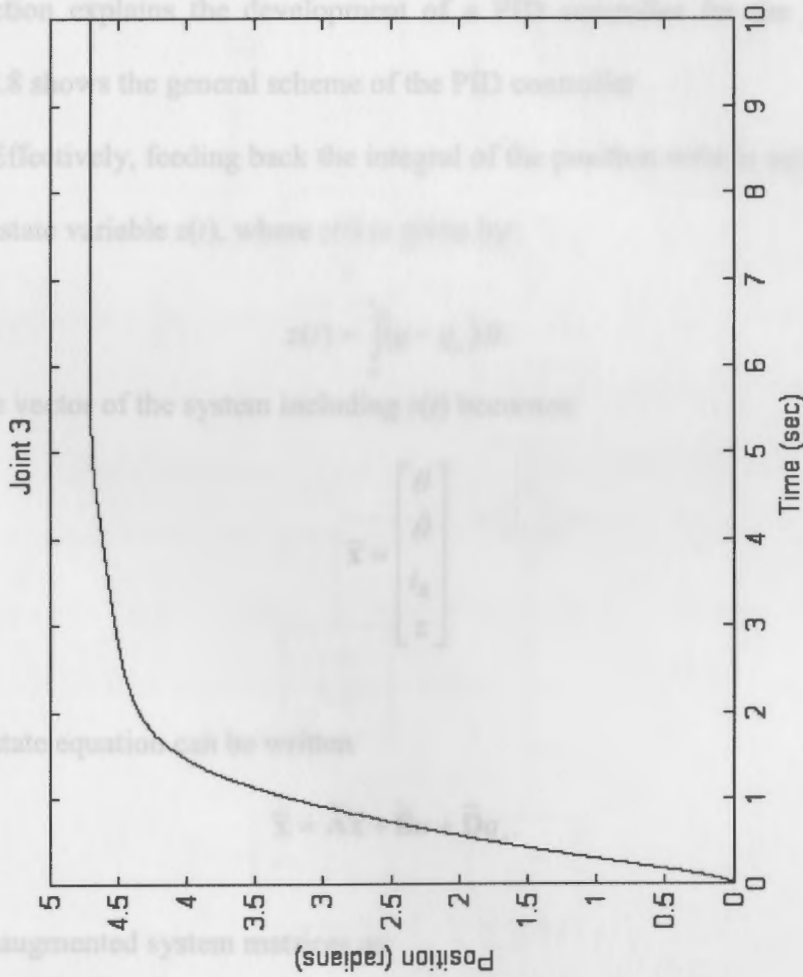


Figure 4.7: Joint 3 State-Feedback Response, Considering Gravity.



## 4.2 Local PID Controller

The steady-state error of the control systems developed thus far, which is caused in large part by gravitational effects, may be eliminated by feeding back the integral of the error signal, developing what is known as a PID (proportional-integral-derivative) controller [1, 8]. This section explains the development of a PID controller for the joint-actuator system. Figure 4.8 shows the general scheme of the PID controller.

Effectively, feeding back the integral of the position error is equivalent to introducing a fourth state variable  $z(t)$ , where  $z(t)$  is given by:

$$z(t) = \int_0^t (q - q_o) dt. \quad (4.7)$$

The state vector of the system including  $z(t)$  becomes:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \theta \\ \dot{\theta} \\ i_R \\ z \end{bmatrix} \quad (4.8)$$

and the state equation can be written

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}u + \tilde{\mathbf{D}}q_o, \quad (4.9)$$

with the augmented system matrices as:

Figure 4.8: PID Controller

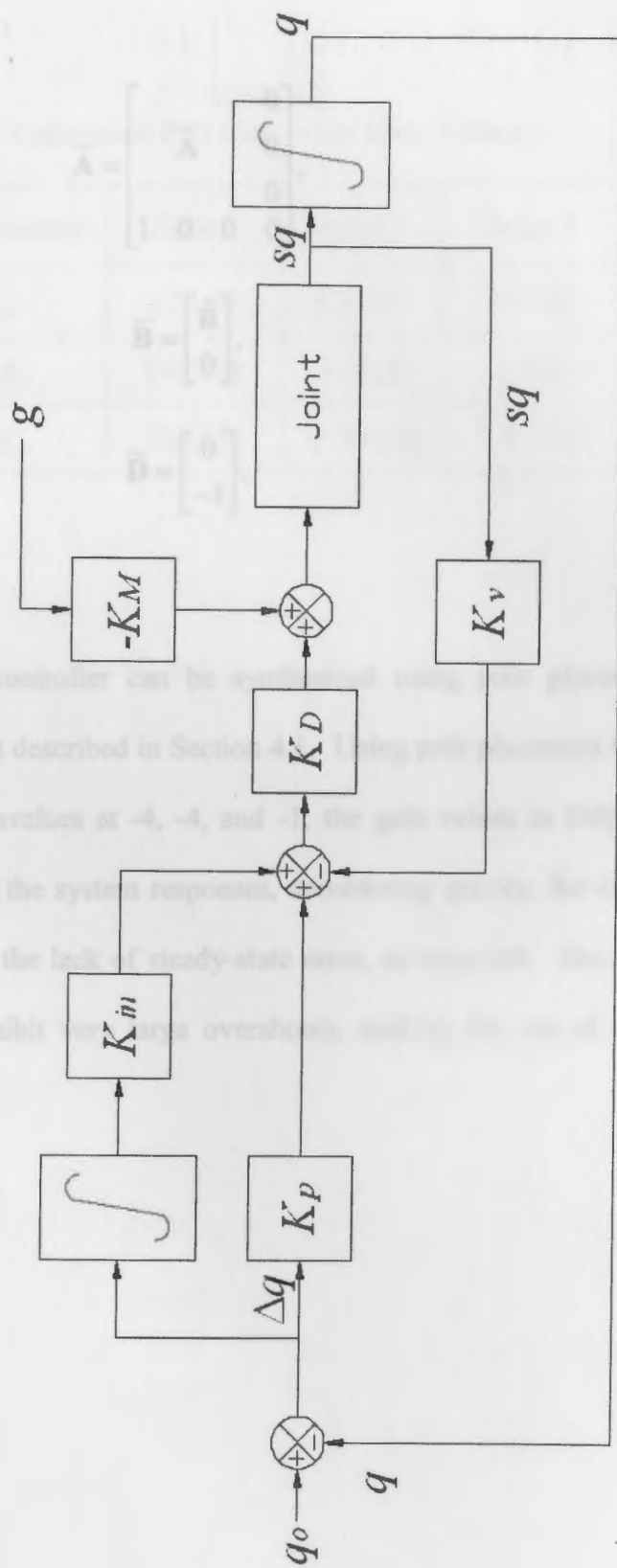


Figure 4.8: PID Controller.

Table 4.2:  
Calculated PID Controller Gain Values.

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 \\ \hat{\mathbf{A}} \\ 0 \\ 0 \\ 1 \quad 0 \quad 0 \quad 0 \end{bmatrix}, \quad (4.10)$$

Parameter		Joint 2	Joint 3
$K_p$	27	0.37037	4.91981
$K_v$	14	-1.12525	2.628
$K_w$	10	0.148148	1.96793

$$\tilde{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{B}} \\ 0 \end{bmatrix}, \quad (4.11)$$

$$\tilde{\mathbf{D}} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}. \quad (4.12)$$

Now the PID controller can be synthesized using pole placement, following a procedure similar to that described in Section 4.1. Using pole placement techniques to place the system matrix eigenvalues at -4, -4, and -1, the gain values in Table 4.2 were found. Figures 4.9-4.11 shows the system responses, considering gravity, for each joint using the PID controller. Notice the lack of steady-state error, as expected. The responses of joints two and three also exhibit very large overshoots, making the use of the PID controller unrealistic.

**Table 4.2:**  
**Calculated PID Controller Gain Values.**

Parameter	Joint 1	Joint 2	Joint 3
$K_p$	270.59	0.37037	4.91981
$K_v$	140.629	-1.12525	2.628
$K_{in}$	108.236	0.148148	1.96793

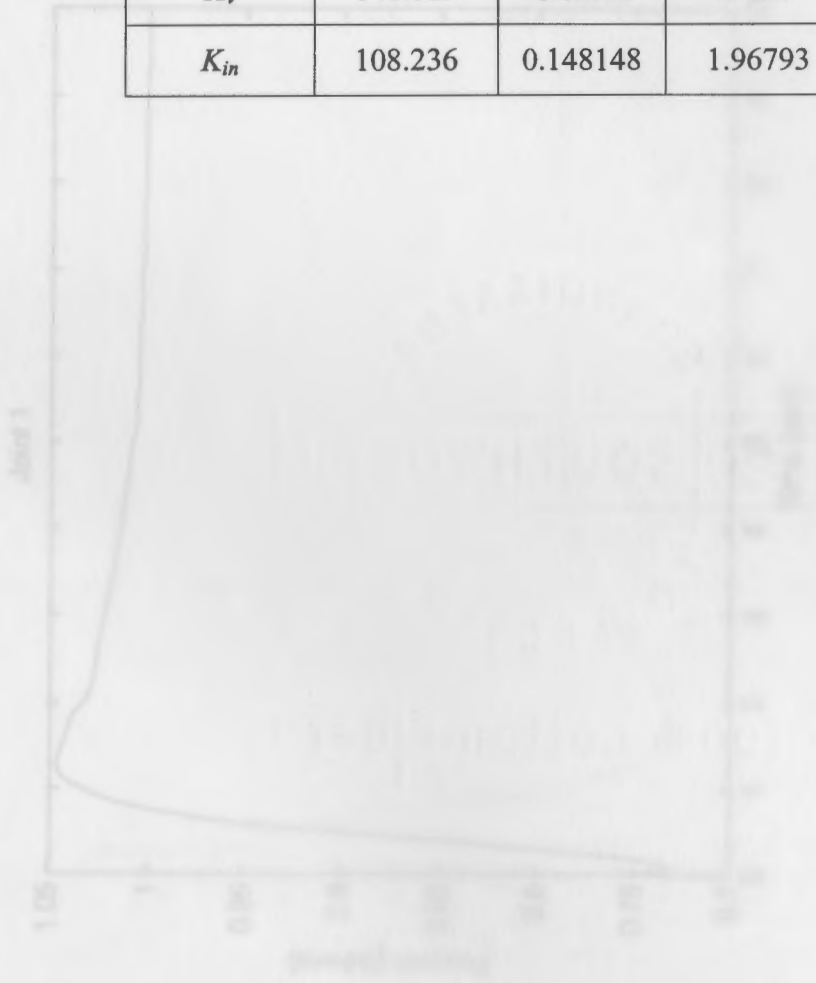


Figure 4.9: Joint 1 PID Controller Response.

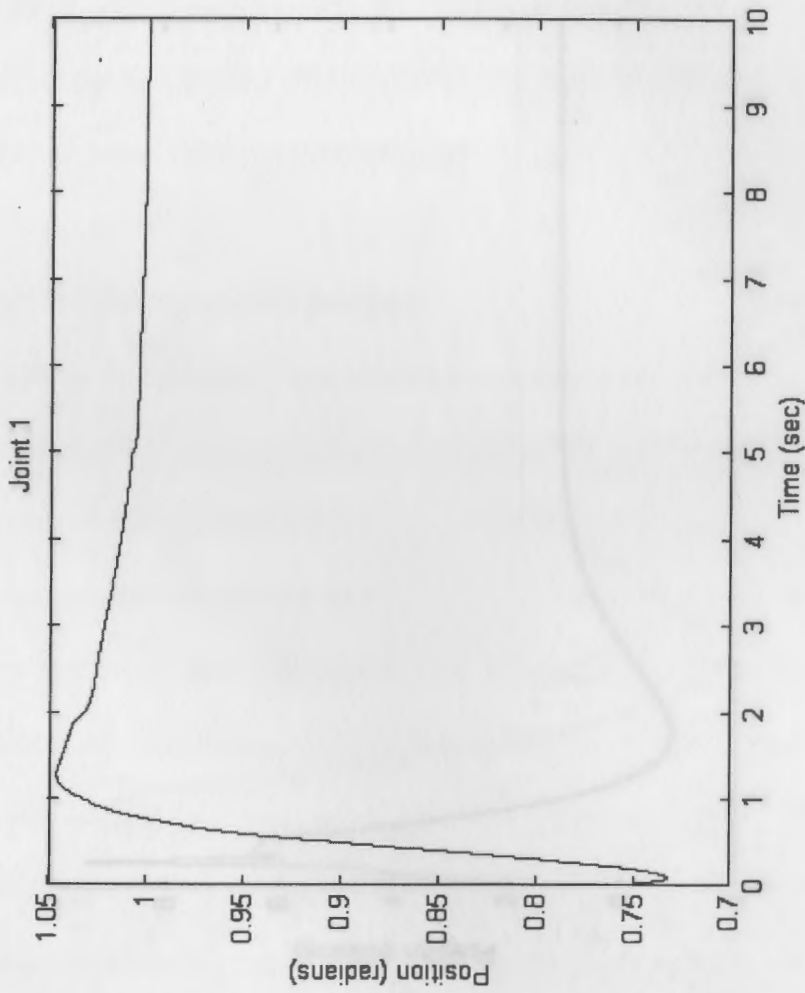


Figure 4.9: Joint 1 PID Controller Response.

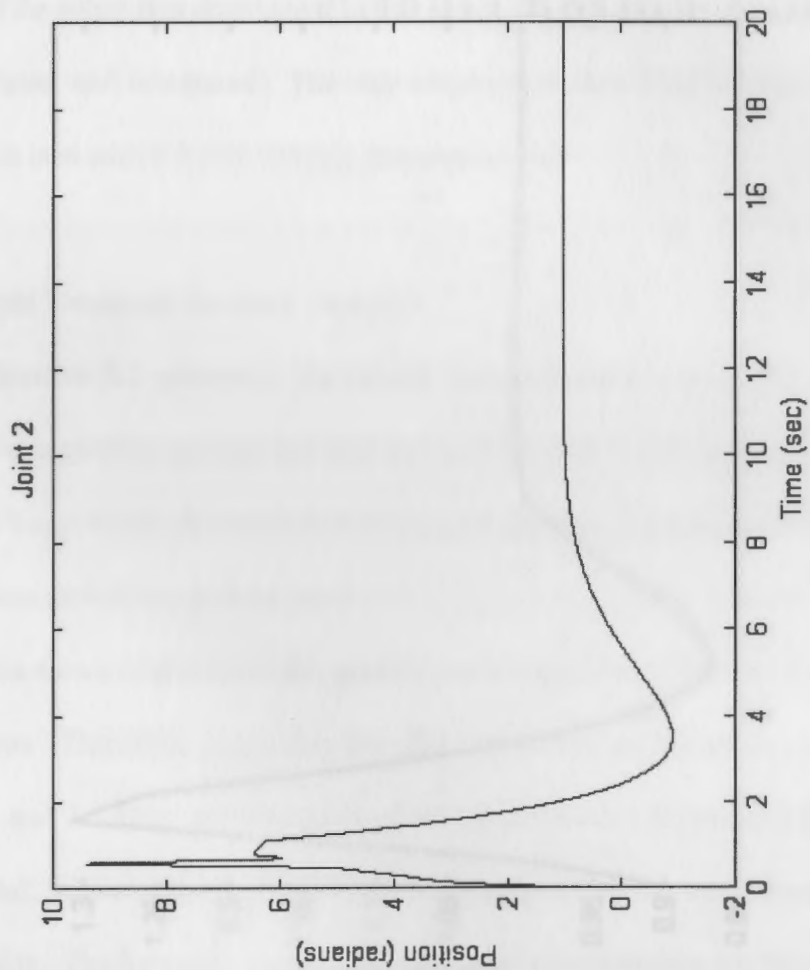


Figure 4.10: Joint 2 PID Controller Response.

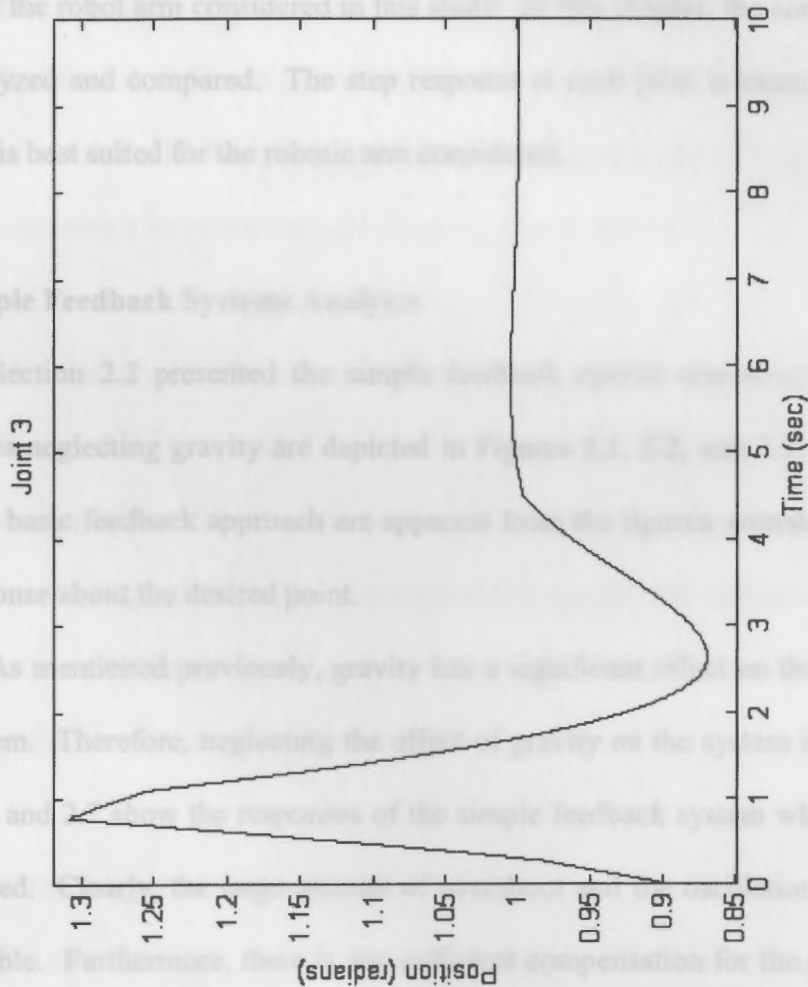


Figure 4.11: Joint 3 PID Controller Response.

## 5.2 Position Servo Systems Analyses CHAPTER V

Section 3.1 presented the design of a basic position servo system for each joint. The responses at each joint to a step input, **ANALYSIS**, are depicted in Figures 3.2, 3.3, and 3.4. The oscillation and overshoot seen in the simple feedback systems are eliminated by this choice. The previous chapters presented various approaches to independently controlling the joints of the robot arm considered in this study. In this chapter, the control system responses are analyzed and compared. The step response at each joint is examined to deduce which scheme is best suited for the robotic arm considered.

### 5.1 Simple Feedback Systems Analyses

Section 2.2 presented the simple feedback system responses for each joint. The responses neglecting gravity are depicted in Figures 2.1, 2.2, and 2.3. The major problems with the basic feedback approach are apparent from the figures: overshoot and oscillation of the response about the desired point.

As mentioned previously, gravity has a significant effect on the steady-state error of the system. Therefore, neglecting the effect of gravity on the system is unrealistic. Figures 2.5, 2.6, and 2.7 show the responses of the simple feedback system when gravity effects are considered. Clearly, the large amount of overshoot and the oscillation of the responses are undesirable. Furthermore, there is not sufficient compensation for the steady-state error due to gravity.

4.2, 4.3, and 4.4, exhibit error-free responses when gravity is neglected. However, since the effect of gravity upon the systems is ignored in the development of the state-space controllers presented, they do not appropriately compensate for gravitational effects. This



## 5.2 Position Servo Systems Analyses

Section 3.1 presented the design of a basic position servo system for each joint. The responses at each joint to a step input, neglecting gravity, are depicted in Figures 3.2, 3.3, and 3.4. The oscillation and overshoot seen in the simple feedback systems are eliminated by this choice of controller (compare Figures 3.2-3.4 with Figures 2.1-2.3). There is very little steady-state error when gravity is neglected.

Figures 3.5, 3.6, and 3.7 show the responses of the position servo systems when gravity is considered. For joint three there is a large amount of steady-state error. The transient responses for joints two and three are also undesirable.

## 3.6 Overall Comparison of Control System Responses

## 5.3 Position Servo with Gravity Compensation System Analyses

Figures 3.9, 3.10, and 3.11 show the responses of the position servo system with gravity compensation described in Section 3.2. The gravity compensator slows down the responses when compared to the uncompensated system, but reduces the steady-state error for joints two and three to zero, at the expense of creating some overshoot (about 28% and 40% for joints two and three, respectively).

## 5.4 State-Feedback Systems Analyses

The state-feedback control systems presented in Section 4.1, having responses given in Figures 4.2, 4.3, and 4.4, exhibit error-free responses when gravity is neglected. However, since the effect of gravity upon the systems is ignored in the development of the state-space controllers presented, they do not appropriately compensate for gravitational effects. This

can be seen by examining Figures 4.5-4.7 and noting the large overshoot and steady-state error of the responses.

Table 5.1

Summary of Control System Response Characteristics

	Joint 1	Joint 2	Joint 3
<b>5.5 Local PID Control Systems Analyses</b>			
Simple servo system neglecting gravity	$T_s = 3.2 \text{ s}$	$T_s = 1.6 \text{ s}$	$T_s = 1.95 \text{ s}$
	OS = 0%	OS = 0%	OS = 0%
	$E = 0\%$	$E = 0\%$	$E = 0\%$
		OS = 100%	OS = 125%
	$T_s = 0\%$	$T_s = 0\%$	$T_s = 0\%$
	$T_s = 3.2 \text{ s}$	$T_s = 1.6 \text{ s}$	$T_s = 2.7 \text{ s}$
<b>5.6 Overall Comparison of Control System Responses</b>			
Servo systems with gravity compensation	OS = 0%	OS = 0%	OS = 0%
	$T_s = 1.8 \text{ s}$	$T_s = 1.8 \text{ s}$	$T_s = 1.8 \text{ s}$
State-space system neglecting gravity	OS = 0%	OS = 0%	OS = 0%
	$E = 0\%$	$E = 0\%$	$E = 0\%$
	$T_s = 3.5 \text{ s}$	$T_s = 2 \text{ s}$	$T_s = 2.5 \text{ s}$
State-space system considering gravity	OS = 44%	OS = 0%	OS = 0%
	$E = 144\%$	$E = 700\%$	$E = 300\%$
	$T_s = 3.1 \text{ s}$	$T_s = 10 \text{ s}$	$T_s = 4.2 \text{ s}$
Local PID controller	OS = 3%	OS = 950%	OS = 30%
	$E = 0\%$	$E = 0\%$	$E = 0\%$

**Table 5.1:**  
**Summary of Control System Response Characteristics.**

System	Joint 1	Joint 2	Joint 3
Simple servo system neglecting gravity	$T_s = 3.2$ s	$T_s = 1.6$ s	$T_s = 1.95$ s
	OS = 0%	OS = 0%	OS = 0%
	$E = 0\%$	$E = 0\%$	$E = 0\%$
Simple servo system considering gravity	$T_s = 2$ s	$T_s = 2$ s	$T_s = 1.5$ s
	OS = 0%	OS = 60%	OS = 125%
	$E = 0\%$	$E = 0\%$	$E = 60\%$
Servo system with compensation for gravity	$T_s = 2.8$ s	$T_s = 3$ s	$T_s = 2.3$ s
	OS = 0%	OS = 28%	OS = 40%
	$E = 0.6\%$	$E = 0\%$	$E = 0\%$
State-space system neglecting gravity	$T_s = 1.5$ s	$T_s = 2.8$ s	$T_s = 1.8$ s
	OS = 0%	OS = 0%	OS = 0%
	$E = 0\%$	$E = 0\%$	$E = 0\%$
State-space system considering gravity	$T_s = 5.5$ s	$T_s = 2$ s	$T_s = 2.5$ s
	OS = 44%	OS = 0%	OS = 0%
	$E = 144\%$	$E \approx 700\%$	$E \approx 360\%$
Local PID controller	$T_s = 3.1$ s	$T_s = 10$ s	$T_s = 4.2$ s
	OS = 5%	OS = 950%	OS = 30%
	$E = 0\%$	$E = 0\%$	$E = 0\%$

## CHAPTER VI

### CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

In this chapter, the system analyses presented in Chapter 5 are used to decide which control system is best for each joint.

#### 6.1 Joint One

Joint one shows the least amount of variation in the approaches to controller design. All of the system responses are similar, which can be seen from Table 5.1. The simple servo system appears to have the “best” response when gravity is considered: it is faster than the other two, and there is less steady-state error for the case considered. See also Figure 3.5, which shows the response of the position servo system with gravity compensation for joint one. This system also performs acceptably.

#### 6.2 Joint Two

Links *A* and *B* are coupled with one another by joint two. The responses of the control systems for joint two are more interesting than those of joint one.

Recall from Section 4.2 and Figure 4.7 that the PID controller’s response is unacceptable for joint 2, due to the presence of a large overshoot. From the simulation results in Table 5.1, it can be seen that the percent overshoot of the uncompensated servo is

unacceptable (it is about 60%). The compensated servo system has a longer settling time than the uncompensated servo, but smaller overshoot (about 28%).

### 6.3 Joint Three

The response of the uncompensated servo for joint three is obviously unacceptable because of the large steady-state error: about 60%. The PID controller appears to have less steady-state error (see also Figure 4.8), but it exhibits overshoot (about 30%) and a longer settling time than that of the servo with gravity compensator. The servo with gravity compensator has a larger overshoot (about 40%), but a relatively fast settling time (about 2.3 s) and better steady-state response than either of the other two controllers.

### 6.4 Suggestions for Further Work

In this work, some of the fundamental issues associated with modeling and controlling a robotic arm were presented. There are many alternate directions that can be taken, many modifications that can be made, and many assumptions and simplifications that can be improved.

Appendix A discusses some of the inaccuracies in the models used in this project. Taking these inaccuracies into consideration would improve the model and make the control system design more realistic.

This paper has considered only the local control of a single joint while the other joints are locked in position. In real use of a robot arm, this type of control may not be desirable or realistic. Appendix B discusses some of the issues involved in the simultaneous control of the joints in the arm.

Although the control of the end-effector, or hand portion of the mechanism, has been largely ignored in this work, Appendix E presents some information useful for extending control to include the hand.

Improvements could be made to the simulation of the gravity effects described in section 1.6.

The models of the arm mechanism and the actuators presented in Chapter 1 are only approximations. There are many other factors that affect the real system, decreasing the accuracy of the model and of the simulations presented in Chapters 1, 2, and 4. This appendix discusses some of the approximations and how they might affect the system.

#### A.1 Arm Mechanism Model Inaccuracies

For simplicity, the links of the robot were treated as rigid bodies. Depending upon the material of which they are made, this may or may not be a valid assumption. Flexible-link manipulators are discussed in some detail in the literature [1, 10].

Equation 1.22 assumed that the joint actuators are directly coupled to the joints such that the position of the actuator,  $\theta$ , is exactly equal to the position of the link,  $q$ . In reality, the relationship between  $\theta$  and  $q$  might be much more complex, due for example to backlash or to practical difficulties in coupling the actuator shaft to the joint shaft. In general, then, equation 1.22 might be replaced by

$$\dot{q} = f(\theta, \dot{\theta}) \quad (A.1)$$

where  $f(\theta, \dot{\theta})$  is some function of the actuator shaft position and the actuator shaft velocity.

## APPENDIX A

### INACCURACIES IN THE MODELS

The models of the arm mechanism and the actuators presented in Chapter 1 are only approximations. There are many other factors that affect the real system, decreasing the accuracy of the model and of the simulations presented in Chapters 2, 3, and 4. This appendix discusses some of the inaccuracies and how they might affect the system.

#### A.1 Arm Mechanism Model Inaccuracies

For simplicity, the links of the robot arm were assumed to be rigid bodies. Depending upon the material of which they are made, this may or may not be a valid assumption. Flexible-link manipulators are discussed in some detail in the literature [1, 10].

Equation 1.22 assumed that the joint actuators are directly coupled to the joints such that the position of the actuator,  $\theta$ , is exactly equal to the position of the joint,  $q$ . In reality, the relationship between  $\theta$  and  $q$  might be much more complex, due for example to friction or to practical difficulties in coupling the actuator shaft to the joint shaft. In general, then, equation 1.22 might be replaced by

$$q = f(\theta, \dot{\theta}), \quad (\text{A.1})$$

where  $f(\theta, \dot{\theta})$  is some function of the actuator shaft position and the actuator shaft velocity.

Similarly, Equation 1.23 assumed that the driving moment required to move a joint is exactly equal to the load on that joint. However, in reality this relationship might be much more complex. Thus, in general, it can be said that

$$P = v(M, Q), \quad (\text{A.2})$$

or, in other words, the driving moment required to move a joint is a function of the load on that joint and the current configuration of the arm.

## A.2 Actuator Model Inaccuracies

The joint actuators were modeled as ideal permanent magnet DC motors, but this type of model is only an approximation.

Throughout this paper, the input to the actuator has been assumed to be the rotor circuit voltage (see Equation 1.18 and Figure 1.2). However, in reality the input signal amplitude is constrained; that is, it must be below some maximal value, say  $u_{\max}$  [1]. This introduces a nonlinearity into the model of the actuators: the input voltage to the actuator is actually  $\mathcal{N}(u)$ , where  $\mathcal{N}(u)$  is a function expressing the saturation voltage of the motor:

$$\mathcal{N}(u) = \begin{cases} -u_{\max} & \text{when } u < -u_{\max} \\ u & \text{when } -u_{\max} \leq u \leq u_{\max} \\ u_{\max} & \text{when } u > u_{\max} \end{cases} \quad (\text{A.3})$$

The effect of this constraint is that there is a more limited domain for the control signal  $u$  than has been considered previously in this paper.

The effect of friction in the model has been ignored thus far. Static friction is a force that opposes the direction of motion of the joint. For the joint to begin to move, it must first overcome the force caused by static friction. Once the joint reaches a certain velocity, the



effect of static friction drops to zero and dynamic friction appears, which also opposes the motion [1, 3]. The effect of both static and dynamic friction upon the joint control systems is that they may increase the steady-state error of the response. The effect of friction can be minimized by good mechanical design and maintenance of the robot, or it may be reduced by adding friction compensation to the control system. One approach to compensating for friction losses is discussed by Vukobratović [1].

... joints of the arm has been considered. In other words, one joint of the arm moves while the others are "locked" into position. This means that moving the arm to a specified position may require several steps, slowing down the positioning. Successive independent motions of the joints also reduce the robot's ability to move along a specified path, since only one joint may be operating at a time. Therefore, it is often desirable for a robot to be able to move all of its joints simultaneously to reach a goal position. In this appendix, some thoughts on the simultaneous control of the joints are presented.

### B.1 Effect of Inertia Variation

The moment of inertia about a joint is the sum of the moments of inertia of the links about that joint; see Equations (1.5), (1.6), and (1.11), and also the preceding discussion. The  $I_{ij}$  are functions of  $q_j$ , where  $q_j$  is the joint angle. If a control system capable of simultaneous control of all of the joints must be able to compensate for the changing moment of inertia. This requires the use of one or more multiplexed control systems, or a system for computing the moment of inertia due to the current position of the robot.

### B.2 Effect of Gravity Moment Variations

The gravity moment about a joint **APPENDIX B** changes as other joints change position; see Equations 1.8, 1.10, and 1.12, and note the presence in expressions for  $G_i$  of functions of  $q_j$  where  $j \neq i$ . **SIMULTANEOUS CONTROL OF JOINTS** is of compensating for these varying gravity moments.

Thus far, local independent control of the joints of the arm has been considered. In other words, one joint of the arm moves while the others are "locked" into position. This means that moving the arm to a specified position may require several steps, slowing down the positioning. Successive independent motions of the joints also reduces the robot's ability to move along a specified path, since only one joint may be moving at a time. Therefore, it is often desirable for a robot to be able to move all of its joints simultaneously to reach a goal position. In this appendix, some thoughts on the simultaneous control of the joints are presented.

### B.4 Modeling the Simultaneous Movement of Arms

#### B.1 Effect of Inertia Variation

The moment of inertia about a joint in the arm changes as the other joints change position; see Equations 1.6, 1.9, and 1.11, and note the presence in expressions for  $H_i$  of functions of  $q_j$  where  $j \neq i$ . A control system capable of simultaneous control of all of the joints must be able to compensate for this changing moment of inertia. This requires the use of one or more microprocessors capable of quickly computing the moment of inertia due to the current position of the robot.

## B.2 Effect of Gravity Moment Variation

The gravity moment about a joint may also change as other joints change position; see Equations 1.8, 1.10, and 1.12, and note the presence in expressions for  $G_i$  of functions of  $q_j$  where  $j \neq i$ . The control system for a joint must also be capable of compensating for these varying gravity moments.

## B.3 Effect of Centrifugal Force

As mentioned in the previous sections, the change in position of joints in the arm affects the moments of inertia and the gravity moments about other joints. This can also be said of the velocity of the joints [1]. When a joint is moving, centrifugal force proportional to the velocity of the joint is exerted, which affects all of the joints. This causes errors in the positioning of the joints.

## B.4 Modeling the Simultaneous Movement of Joints

Aplevich suggests the use of energy-based methods (Euler-Lagrange equations) in modeling the simultaneous movement of joints in a robot arm [7]. The total kinetic energy of the system is found as a function of the joint positions and their velocities. The total potential energy is found as a function of the joint positions, and the Lagrangian (kinetic minus potential energy) is used to find the differential equations of motion of the system.

## APPENDIX C

C PROGRAM FOR FINDING MAXIMUM  $H$  AND  $G$ 

This program uses equations 1.6-1.12 to approximate the maximum values of  $H$  and  $G$  by iterating over values of  $q_1$ ,  $q_2$ , and  $q_3$  between 0 and  $\pi/2$  radians. Note that the program starts with  $q_2$  and  $q_3$  equal to 0.001 since  $1/\sin \theta$  and  $1/\cos \theta$  are undefined for  $\theta = 0$ .

```
#include <iostream>
#include <math.h>

#define m2 2.0
#define m3 2.0
#define ma 10.0
#define mb 10.0
#define mc 10.0
#define la 0.25
#define lb 0.25
#define lc 0.25
#define mp 20.0
#define rp 0.1
#define g 9.81
#define pi 3.14159265

int main(void) {
    double q1_deg=0.0, q2_deg=0.001, q3_deg=0.001, q1, q2,
        q3;
    double inc=1.0;
    double max_h1, max_g1, max_h2, max_g2, max_h3, max_g3;
    double max_h1_q1, max_h1_q2, max_h1_q3, max_g1_q1,
        max_g1_q2, max_g1_q3;
    double max_h2_q1, max_h2_q2, max_h2_q3, max_g2_q1,
        max_g2_q2, max_g2_q3;
    double max_h3_q1, max_h3_q2, max_h3_q3, max_g3_q1,
        max_g3_q2, max_g3_q3;
    double z1, h1, g1, h2, g2, h3, g3;
```

```

q1=(pi/180.0)*q1_deg;
q2=(pi/180.0)*q2_deg;
q3=(pi/180.0)*q3_deg;

// Since q2=q3=0, we will use la+lb+lc to be the
// distance from joint 1 to the end of the gripper
z1 = la+lb+lc;

h1 = ((1.0/3.0)*ma*la*la) +
      ((1.0/3.0)*mb*lb*lb+mb*la*la) +
      ((3.0/2.0)*mp*rp*rp + mp*z1*z1) +
      ((1.0/3.0)*mc*lc*lc + mc*(la*la+lb*lb +
      2*la*lb*cos(q2)));

g1 = (0.5*ma*g*la + m2*g*la * mb*g*la + m3*g*la +
      mc*g*la + mp*g*la)*sin(q1) +
      (0.5*mb*g*lb + mb*g*lb + mc*g*lb +
      mp*g*lb)*cos(q1+q2)
      + (0.5*mc*g*lc + mp*g*(lc+rp))*cos(q1+q2+q3);

max_h1 = h1;
max_g1 = g1;

for (q1_deg=0.0+inc; q1_deg<90.0; q1_deg+=inc) {
  cout << "Calculating q1=" << q1_deg << " (current"
  << " max_h1 = " << max_h1 << ", max_g1 = "
  << max_g1 << ") ... \n";
  for (q2_deg=0.0; q2_deg<90.0; q2_deg+=inc) {
    for (q3_deg=0.0; q3_deg<90.0; q3_deg+=inc) {
      q1=(pi/180.0)*q1_deg;
      q2=(pi/180.0)*q2_deg;
      q3=(pi/180.0)*q3_deg;

      if (q2 == 0.0 && q3 == 0.0) {
        z1 = la+lb+lc;
      }
      else {
        z1 = sqrt(pow((la +
          lb*sin(q3)/sin(q2+q3)),2) +
          pow((lb +
          lb*sin(q2)/sin(q2+q3)),2) +
          2*(la +
          lb*sin(q3)/sin(q2+q3))*(lb +
          lb*sin(q2)/sin(q2+q3)));
      }
    }
  }
}

```

```

h1 = ((1.0/3.0)*ma*la*la) +
      ((1.0/3.0)*mb*lb*lb+mb*la*la)
      + ((3.0/2.0)*mp*rp*rp + mp*z1*z1)
      + ((1.0/3.0)*mc*lc*lc
      + mc*(la*la+lb*lb
      + 2*la*lb*cos(q2)));

```

```

g1 = (0.5*ma*g*la + m2*g*la * mb*g*la +
      m3*g*la + mc*g*la
      + mp*g*la)*sin(q1) + (0.5*mb*g*lb +
      mb*g*lb + mc*g*lb
      + mp*g*lb)*cos(q1+q2)
      + (0.5*mc*g*lc +
      mp*g*(lc+rp))*cos(q1+q2+q3);

```

```

h2 = ((1.0/3.0)*mb*lb*lb) +
      ((1.0/3.0)*mc*lc*lc)
      + ((3.0/2.0)*mp*rp*rp
      + mp*(lb*lb+lc*lc+2*lb*lc*cos(q3)));

```

```

g2 = (-0.5*ma*g*la)*cos(q1) + (0.5*mb*g*lb
      + mb*g*lb + mc*g*lb +
      mp*g*lb)*cos(q1+q2)
      + (0.5*mc*g*lc +
      mp*g*(lc+rp))*cos(q1+q2+q3);

```

```

h3 = ((1.0/3.0)*mc*lc*lc) +
      ((3.0/2.0)*mp*rp*rp + mp*lc*lc);

```

```

g3 = (-0.5*ma*g*la)*sin(q1) + (-ma*g*lb-
      m2*g*lb - 0.5*mb*g*lb)*sin(q1+q2) +
      (0.5*mc*g*lc + mp*g*(lc+rp));

```

```

if (h1 > max_h1 || -h1 > max_h1) {
    max_h1 = h1;
    max_h1_q1 = q1_deg;
    max_h1_q2 = q2_deg;
    max_h1_q3 = q3_deg;
}

```

```

if (g1 > max_g1 || -g1 > max_g1) {
    max_g1 = g1;
    max_g1_q1 = q1_deg;
    max_g1_q2 = q2_deg;
    max_g1_q3 = q3_deg;
}

```

```

cout << "\n" if (h2 > max_h2 || -h2 > max_h2) {
    max_h2 = h2;
    max_h2_q1 = q1_deg;
    max_h2_q2 = q2_deg;
    max_h2_q3 = q3_deg;
}

if (g2 > max_g2 || -g2 > max_g2) {
    max_g2 = g2;
    max_g2_q1 = q1_deg;
    max_g2_q2 = q2_deg;
    max_g2_q3 = q3_deg;
}

if (h3 > max_h3 || -h3 > max_h3) {
    max_h3 = h3;
    max_h3_q1 = q1_deg;
    max_h3_q2 = q2_deg;
    max_h3_q3 = q3_deg;
}

if (g3 > max_g3 || -g3 > max_g3) {
    max_g3 = g3;
    max_g3_q1 = q1_deg;
    max_g3_q2 = q2_deg;
    max_g3_q3 = q3_deg;
}
}
}

cout << "Maximum h1=" << max_h1 << " at q1="
<< max_h1_q1 << ", q2=" << max_h1_q2
<< ", q3=" << max_h1_q3 << "\n";

cout << "Maximum g1=" << max_g1 << " at q1="
<< max_g1_q1 << ", q2=" << max_g1_q2
<< ", q3=" << max_g1_q3 << "\n";
cout << "\n";

cout << "Maximum h2=" << max_h2 << " at q1="
<< max_h2_q1 << ", q2=" << max_h2_q2
<< ", q3=" << max_h2_q3 << "\n";

cout << "Maximum g2=" << max_g2 << " at q1="
<< max_g2_q1 << ", q2=" << max_g2_q2
<< ", q3=" << max_g2_q3 << "\n";

```

```

cout << "\n";

cout << "Maximum h3=" << max_h3 << " at q1="
  << max_h3_q1 << ", q2=" << max_h3_q2
  << ", q3=" << max_h3_q3 << "\n";

cout << "Maximum g3=" << max_g3 << " at q1="
  << max_g3_q1 << ", q2=" << max_g3_q2
  << ", q3=" << max_g3_q3 << "\n";
}

```

In section 11.1, the calculation of the gravity moments is discussed. The U

programs used to calculate the gravity moments and angular differences for each joint as it

moves from 0 to 1 radians are given in the appendix.

#### D.1 Calculating $G_j$ for $q_1 = q_2 = 0$ rad

```

#include <iostream>
#include <math.h>

```

```

#define m1 2.0
#define m2 2.0
#define m3 10.0
#define m4 10.0
#define m5 10.0
#define l1 0.25
#define l2 0.25
#define l3 0.25
#define l4 0.25
#define l5 0.25
#define r1 0.1
#define g 9.81
#define pi 3.14159265

```

```

int main() {

```

```

    double q1, q2, q3;
    double G1, G2, G3;
    double G4, G5;

```

```

    q1=pi/2;
    q2=pi/2;

```

```

    for (q1=0; q1<1.5; q1+=0.1) {
        G1 = (0.5*m1*g*l1 + m2*g*l1 + m3*g*l1 +
              m4*g*l1 + m5*g*l1 + m1*g*l1)*sin(q1)
            + (0.5*m2*g*l2 + m3*g*l2 + m4*g*l2 +

```



## APPENDIX D

C PROGRAMS FOR CALCULATING  $G$  and  $\Delta\theta$ 

In section 1.6, the modeling of gravity's effect upon the systems is discussed. The C programs used to calculate the gravity moments and angular differences for each joint as it moves from 0 to 1 radians are given in this appendix.

D.1 Calculating  $G_1$  for  $q_2 = q_3 = \pi/4$  rad

```
#include <iostream>
#include <math.h>

#define m2 2.0
#define m3 2.0
#define ma 10.0
#define mb 10.0
#define mc 10.0
#define la 0.25
#define lb 0.25
#define lc 0.25
#define mp 20.0
#define rp 0.1
#define g 9.81
#define pi 3.14159265

int main(void) {
    double q1, q2, q3;
    double g1;
    double theta;

    q2=pi/4;
    q3=pi/4;

    for (q1=0.0; q1<1.0; q1+=0.1) {
        g1 = (0.5*ma*g*la + m2*g*la * mb*g*la +
              m3*g*la + mc*g*la + mp*g*la)*sin(q1)
              + (0.5*mb*g*lb + mb*g*lb + mc*g*lb +
```

```

        mp*g*lb)*cos(q1+q2)
        + (0.5*mc*g*lc +
        mp*g*(lc+rp))*cos(q1+q2+q3);
    theta = acos(g1/(la*(m2+m3+mb+mc+mp)*g));
    cout << "q1=" << q1 << "\tg1=" << g1
        << "\ttheta=" << theta << "\n";
}
#include <iostream>
#include <math.h>
}

```

## D.2 Calculating $G_2$ for $q_1 = q_3 = \pi/4$ rad

```

#define m2 2.0
#define m3 2.0
#define ma 10.0
#define mb 10.0
#define mc 10.0
#define la 0.25
#define lb 0.25
#define lc 0.25
#define mp 20.0
#define rp 0.1
#define g 9.81
#define pi 3.14159265

int main(void) {
    double q1q1, q2, q3;
    double g2;
    double theta;

    q1=pi/4;
    q3=pi/4;

    for (q2=0.0; q2<1.0; q2+=0.1) {
        g2 = (-0.5*ma*g*la)*cos(q1) + (0.5*mb*g*lb +
        mb*g*lb + mc*g*lb + mp*g*lb)*cos(q1+q2) +
        (0.5*mc*g*lc +
        mp*g*(lc+rp))*cos(q1+q2+q3);
        theta = acos(g2/(lb*(m3+mc+mp)*g));
        cout << "q2=" << q2 << "\tg2=" << g2
            << "\ttheta=" << theta << "\n";
    }
}

```

## APPENDIX E

D.3 Calculating  $G_3$  for  $q_1 = q_2 = \pi/4$  rad

```

#include <iostream>
#include <math.h>

#define m2 2.0
#define m3 2.0
#define ma 10.0
#define mb 10.0
#define mc 10.0
#define la 0.25
#define lb 0.25
#define lc 0.25
#define mp 20.0
#define rp 0.1
#define g 9.81
#define pi 3.14159265

int main(void) {
    double q1, q2, q3;
    double g3;
    double theta;

    q1=pi/4;
    q2=pi/4;

    for (q3=0.0; q3<1.0; q3+=0.1) {
        g3 = (-0.5*ma*g*la)*sin(q1) + (-ma*g*lb-
            m2*g*lb-0.5*mb*g*lb)*sin(q1+q2)
            + (0.5*mc*g*lc + mp*g*(lc+rp))*cos(q3);
        theta = acos(g3/(lc*mp*g));
        cout << "q3=" << q3 << "\tg3=" << g3
            << "\ttheta=" << theta << "\n";
    }
}

```

The design and control of the end effector, or hand component, of the arm-gripper system in this work was not presented. There is a multitude of literature on the design and control of robotic hands, some of which is referenced in this chapter. Some of the issues involved in the design and control of robotic hands are discussed in the next chapter.

### E.2 Design of the Hand

The design of the hand is a critical part of the design of the arm-gripper system. The design of the hand is discussed in this section. Figure 1.1 shows the hand mounted at the end of the arm. The hand consists of two "fingers" or "phalanges," each of which consists of two rigid links and two joints. At the lowest level, the kinematic analysis of the fingers of the hand is very similar to the analysis of the arm presented in Chapter 1, with respect to moments of inertia and gravitational moments about the joints.

## APPENDIX E

### END-EFFECTOR DESIGN AND CONTROL

The design and control of the end-effector, or hand component, of the arm-gripper system under consideration in this work was not presented. There is a multitude of literature concerned with the design and control of robotic hands, some of which is referenced in this appendix. Then some of the issues involved in the design and control of robotic hands are introduced.

#### E.1 Important Literature

Matthew Mason is a recognized authority on the design and control of robotic manipulators, especially at levels higher than the executive level [2, 11]. Several recent papers also discuss hand design, such as those by Hwang et al. [12], Matsuoka [13], and Pons et al. [14]. Arimoto discusses the use of Hamiltonian mechanics in analysis and control of a hand [4].

#### E.2 Design of the Hand

The design of the hand, or end-effector, presented in this work is similar to the design of the arm itself. Figure 1.1 shows the hand mounted at the end of the arm. The hand consists of two "fingers" or "phalanges," each of which consists of two rigid links and two joints. At the lowest level, the kinematic analysis of the fingers of the hand is very similar to the analysis of the arm presented in Chapter 1, with respect to moments of inertia and gravitational moments about the joints.

### **E.3 Grasping vs. Manipulating**

The rigor of analysis and control required for an end-effector depends upon its intended use. The two main uses of robotic hands is grasping and manipulation. Pons, et al., define “grasping” as “the combination of procedures and operations needed to hold an object in a static position with respect to the hand itself” [14]. In other words, grasping an object means immobilizing it with respect to the hand, while the hand and the object it grasps may be moved about the robot’s workspace using the arm. Manipulation refers to the handling of an object by the hand. Manipulation may be considered “dynamic grasping,” or a series of grasps; in fact, it may be considered a generalization of grasping [14].

Kinematics can be used to determine the minimum number of fingers required for grasping and for manipulation. In a paper by Arimoto [4], it is noted that three fingers are sufficient for grasping (immobilizing) a two-dimensional object with triangular shape and four are sufficient for grasping a parallelepiped. In fact, in general four frictionless fingers are necessary and sufficient for grasping a two-dimensional polygonal object.

### **E.4 Dexterity of Manipulation**

When the velocities of the joints in the hand are related to the velocities of the links using a special Jacobian matrix, a useful matrix called a “grip transform” is formed. The grip transform can be used to determine the overall force exerted on an object grasped by the hand as a function of the joint torques [14].

## E.5 Manipulability

If the purpose of the hand is to manipulate objects in addition to simply grasping them, a useful quantity is the “manipulability” of the hand. Although there is not currently a standardized definition of manipulability, one useful interpretation is due to Cutkosky, and is presented by Pons, et al. Manipulability can be thought of as “the ability to impart arbitrary motions to the object from a given point in the workspace” [14].

## E.6 Dynamics of Pinching

Arimoto suggests the use of Hamiltonian mechanics in the kinematic analysis of robotics hands [4]. In particular, the dynamics of a pair of fingers pinching an object can be derived on the basis of Hamilton’s variational principle. A series of differential equations may be written from the kinematic analysis and used to derive a state-space model of the hand if desired.

- [1] Hibbeler, R.C. *Engineering Mechanics: Dynamics*. Upper Saddle River, Prentice Hall, 2001.
- [2] Mason, Matthew T. and J. Kenneth Salisbury, Jr. *Handbook of Robotics and Manipulation*. Wiley, 2003.
- [3] Hibbeler, R.C. *Engineering Mechanics: Dynamics*. Upper Saddle River, Prentice Hall, 2001.
- [4] Arimoto, E. “Introduction of Hamiltonian Mechanics to the Kinematic Analysis of Robotics Hands” *IEEE Transactions on Systems, Man, and Cybernetics*, 1987.
- [5] Kane, Thomas R. *Dynamics of a Rigid Body*. Wiley, 1985.
- [6] Noc, Norman S. *Classical Mechanics*. Wiley, 1991.
- [7] Goldstein, H., J. Poole, and J. Safir. *The Classical Mechanics*. Wiley, 1991.
- [8] Ogata, Keiroshi. *Modern Control Engineering*. Wiley, 1990.
- [9] Ferick, Tom. “Mason’s Rule” <http://www.masonrule.com/masonrule.html>, 14 July 2003. <<http://www.masonrule.com/masonrule.html>>
- [10] Chernousko, Felix, Nikolai N. Bolotin, and Valery G. Korovin. *Manipulation, Robotic Dynamics, Control, and Identification*. Boca Raton: CRC Press, 1994.
- [11] Mason, Matthew T. *Manipulation of Flexible Structures*. Cambridge: MIT Press, 2001.
- [12] Huang, Cheng-Lian, Matthew Trussell, and Ken Han. “Kinematics of Grasping and Manipulation of a 6-DOF System Subjected to a Multifingered Robot Hand” *Journal of Robotics Systems*, 16(11):945-960, 1999.

## REFERENCES

- [1] Vukobratović, Miomir and Dragan Stokić. Applied Control of Manipulation Robots: Analysis, Synthesis and Exercises. Berlin: Springer-Verlag, 1989.
- [2] Mason, Matthew T. and J. Kenneth Salisbury, Jr. Robot Hands and the Mechanics of Manipulation. Cambridge: The MIT Press, 1985.
- [3] Hibbeler, R.C. Engineering Mechanics: Dynamics. Upper Saddle River: Prentice Hall, 2001.
- [4] Arimoto, S. "Intelligent Control of Multi-fingered Hands." Annual Reviews in Control. 28 (2004): 78-85.
- [5] Adams, Jay L. "Control of a DC Motor: Comparing Classical and Adaptive Techniques." Thesis. Youngstown State University, 2004.
- [6] Nise, Norman S. Control Systems Engineering, Third Edition. New York: John Wiley and Sons, 2000.
- [7] Aplevich, J. Dwight. The Essentials of Linear State-Space Systems. New York: John Wiley and Sons, 2000.
- [8] Ogata, Katsuhiko. Modern Control Engineering. Upper Saddle River: Prentice Hall, 1990.
- [9] Penick, Tom. "Mason's Gain Rule." TEI Controls. 2000. TEI Controls. 14 July 2005. <<http://www.teicontrols.com/notes/AutomaticControlsEE362K/MasonsRule.pdf>>
- [10] Chernousko, Felix, Nikolai N. Bolotnik, and Valery G. Gragetsky. Manipulation Robots: Dynamics, Control, and Optimization. Boca Raton: CRC Press, 1994.
- [11] Mason, Matthew T. Mechanics of Robotic Manipulation. Cambridge: MIT Press, 2001.
- [12] Hwang, Chang-Soon, Masaharu Takano, and Ken Sasaki. "Kinematics of Grasping and Manipulation of a B-Spline Surface Object by a Multifingered Robot Hand." Journal of Robotic Systems. 16 (1999): 445-460.

- [13] Matsuoka, Yoky. "The Mechanisms in a Humanoid Robot Hand." Autonomous Robots. 4 (1997): 199-209.
- [14] Pons, J.L., R. Ceres, and F. Pfeiffer. "Multifingered Dextrous Robotics Hand Design and Control: A Review." Robotica. 17 (1999): 661-674.